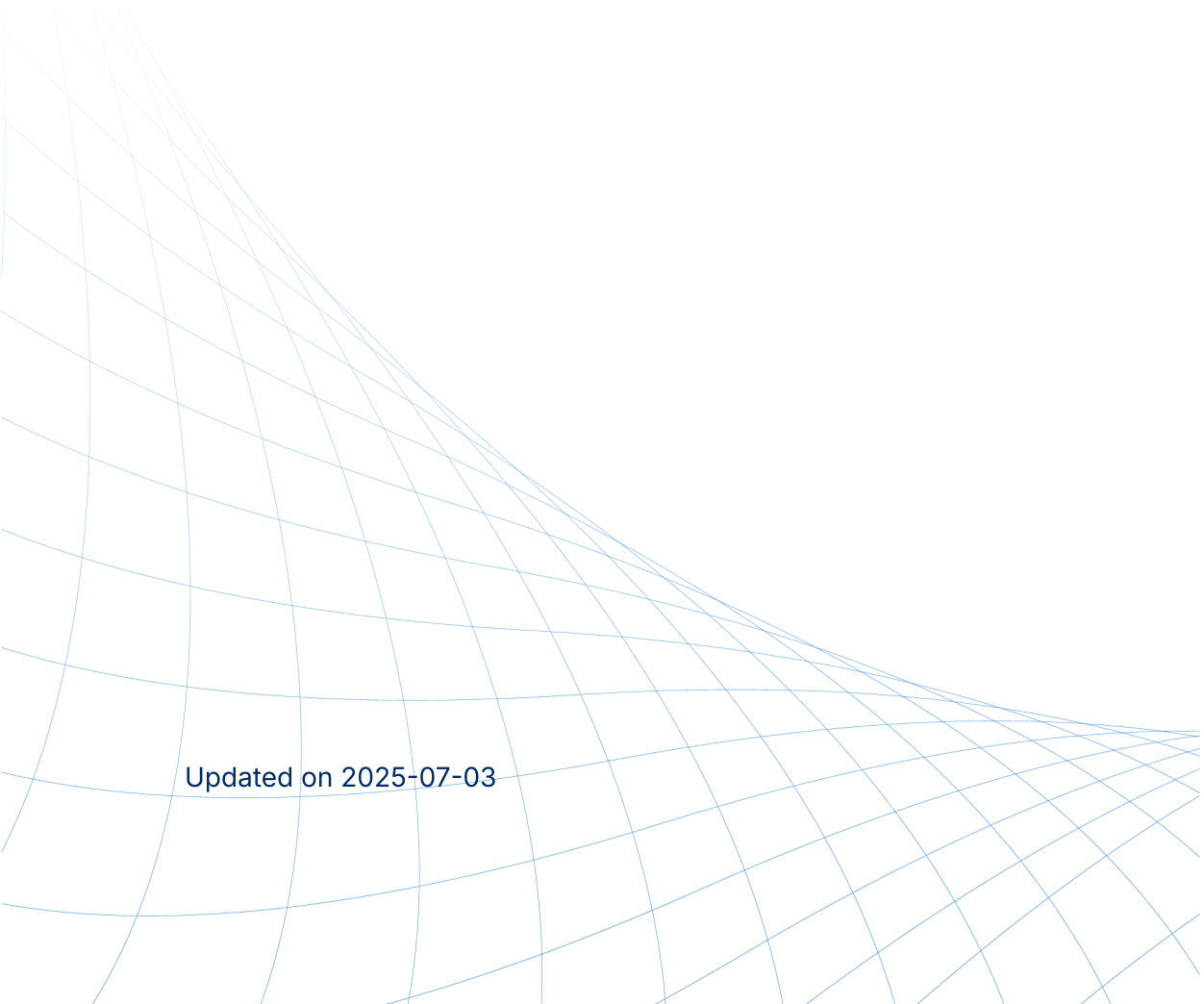
runZero
User Guide
v4.0.250701.0



Updated on 2025-07-03

Contents

- 4.0.250329.0
- 4.0.250328.0
- 4.0.250325.0
- 4.0.250324.1
- 4.0.250324.0
- 4.0.250317.0
- 4.0.250307.0
- 4.0.250305.0
- 4.0.250303.1
- 4.0.250303.0
- 4.0.250228.0
- 4.0.250226.0
- 4.0.250221.0
- 4.0.250219.1
- 4.0.250219.0
- 4.0.250214.0
- 4.0.250213.1
- 4.0.250213.0
- 4.0.250209.0
- 4.0.250208.0
- 4.0.250207.3
- 4.0.250207.2
- 4.0.250207.0
- 4.0.250203.1
- 4.0.250203.0
- 4.0.250130.1
- 4.0.250130.0
- 4.0.250129.0
- 4.0.250127.0
- 4.0.250124.0
- 4.0.250123.0
- 4.0.250122.0
- 4.0.250120.0
- 4.0.250117.0
- 4.0.250116.0
- 4.0.250113.0
- 4.0.250106.0
- 4.0.241223.0
- 4.0.241219.2
- 4.0.241217.0
- 4.0.241213.0
- 4.0.241212.0
- 4.0.241210.0
- 4.0.241209.1
- 4.0.241206.0
- 4.0.241205.1
- 4.0.241205.0
- 4.0.241203.0
- 4.0.241125.0
- 4.0.241123.0

- 4.0.241122.0
- 4.0.241120.0
- 4.0.241118.0
- 4.0.241114.0
- 4.0.241109.0
- 4.0.241106.0
- 4.0.241101.2
- 4.0.241101.1
- 4.0.241101.0
- 4.0.241029.0
- 4.0.241025.0
- 4.0.241023.0
- 4.0.241022.0
- 4.0.241016.0
- 4.0.241015.0
- 4.0.241010.0
- 4.0.241009.0
- 4.0.241003.0
- 4.0.241001.0
- 4.0.240927.0
- 4.0.240926.0
- 4.0.240925.0
- 4.0.240924.1
- 4.0.240924.0
- 4.0.240923.0
- 4.0.240921.0
- 4.0.240919.0
- 4.0.240918.0
- 4.0.240917.2
- 4.0.240917.1
- 4.0.240910.2
- 4.0.240909.0
- 4.0.240907.0
- 4.0.240904.1
- 4.0.240904.0
- 4.0.240902.0
- 4.0.240829.0
- 4.0.240826.0
- 4.0.240825.1
- 4.0.240825.0
- 4.0.240822.0
- 4.0.240820.0
- 4.0.240817.0
- 4.0.240816.0
- 4.0.240814.0
- 4.0.240811.0
- 4.0.240809.0
- 4.0.240807.0
- 4.0.240803.0
- 4.0.240802.0

- 4.0.240731.1
- 4.0.240731.0
- 4.0.240730.0
- 4.0.240729.1
- 4.0.240729.0
- 4.0.240727.0
- 4.0.240726.0
- 4.0.240725.0
- 4.0.240722.0
- 4.0.240719.0
- 4.0.240718.0
- 4.0.240716.0
- 4.0.240715.1
- 4.0.240715.0
- 4.0.240712.0
- 4.0.240707.0
- 4.0.240702.0
- 4.0.240628.0
- 4.0.240627.0
- 4.0.240626.1
- 4.0.240622.0
- 4.0.240621.0
- 4.0.240620.0
- 4.0.240619.2
- 4.0.240619.1
- 4.0.240619.0
- 4.0.240618.0
- 4.0.240616.0
- 4.0.240614.0
- 4.0.240613.0
- 4.0.240612.0
- 4.0.240610.0
- 4.0.240607.0
- 4.0.240606.1
- 4.0.240606.0
- 4.0.240605.0
- 4.0.240603.0
- 4.0.240531.0
- 4.0.240530.0
- 4.0.240529.1
- 4.0.240524.0
- 4.0.240522.0
- 4.0.240519.0
- 4.0.240516.0
- 4.0.240514.0
- 4.0.240508.0
- 4.0.240503.0
- 4.0.240501.0
- 4.0.240429.0
- 4.0.240425.0

- 4.0.240424.0
- 4.0.240423.0
- 4.0.240419.0
- 4.0.240417.0
- 4.0.240411.0
- 4.0.240410.0
- 4.0.240408.0
- 4.0.240405.0
- 4.0.240404.0
- 4.0.240403.0
- 4.0.240402.0
- 4.0.240401.0
- 4.0.240331.0
- 4.0.240329.0
- 4.0.240327.0
- 4.0.240326.0
- 4.0.240325.0
- 4.0.240320.0
- 4.0.240318.0
- 4.0.240314.0
- 4.0.240311.0
- 4.0.240308.0
- 4.0.240306.0
- 4.0.240305.1
- 4.0.240305.0
- 4.0.240304.0
- 4.0.240301.0
- 4.0.240228.0
- 4.0.240226.0
- 4.0.240223.0
- 4.0.240221.0
- 4.0.240218.0
- 4.0.240216.0
- 4.0.240214.0
- 4.0.240213.0
- 4.0.240208.0
- 4.0.240207.0
- 4.0.240206.0
- 4.0.240205.0
- 4.0.240202.0
- 4.0.240131.0
- 4.0.240129.0
- 4.0.240126.0
- 4.0.240124.0
- 4.0.240122.0
- 4.0.240119.0
- 4.0.240117.0
- 4.0.240112.0
- 4.0.240110.0
- 4.0.240109.0

What is runZero?

runZero

runZero is a total attack surface and exposure management platform that combines active scanning, passive discovery, and API integrations to deliver complete visibility into managed and unmanaged assets across IT, OT, IoT, cloud, mobile, and remote environments. runZero can be used as a hosted service (SaaS) or managed on-premise. The runZero stack consists of one more Consoles, linked Explorers that run as light-weight services on network points-of-presence, and a command-line tool that can be used for offline data collection. runZero can be managed through the web interface, via API, or for self-hosted customers, on the command line.

Data Sources

- **Active Scans**: runZero's best-in-class active scan engine is fast, accurate, and safe for all environments, with support for a massive number of protocols and applications.
- **Passive Traffic Sampling**: runZero's passive traffic sampling engine scales with available resources and works with broadcast traffic, SPAN ports, and encapsulated streams. Any runZero Explorer can be used for passive traffic sampling, regardless of location, configuration, or resources.
- **API Integrations**: runZero supports inbound and outbound integrations with major Cloud, Endpoint, CMDB, and Endpoint providers. In addition to the native options, customers can create their own integrations using the Custom Integration API and Custom Integration Scripts.

Live Inventory

- **Assets**: runZero tracks all assets across the environment; including cloud, mobile, endpoint, server, OT, IoT, and everything else in between. Assets are correlated and merged across data sources to provide a multi-perspective snapshot of all organization resources. The asset inventory supports deep search, configurable columns, and simple export.
- **Services**: runZero tracks all identified network services, via active scans, passive discovery, and integrations (where applicable). The services inventory simplifies exposure management tasks and enables deep search and exports.
- **Screenshots**: runZero takes a snapshot of each exposed web service included in active scans. The screenshot inventory allows security teams to visually inspect unknown devices and services.
- **Software**: runZero identifies network-exposed software and imports software records from API integrations. The result is a software inventory that can be used to quickly

identify specific packages and versions across the environment.

- **Vulnerabilities**: runZero reports vulnerabilities based on identified exposures and imports vulnerability data from API integrations. The vulnerability inventory is provided as both a detailed, per-asset inventory, as well as a grouped view that simplifies investigation into specific issues.
- **Certificates**: runZero identifies SSL/TLS certificates, including those that are expired or soon-to-expire. The certificate inventory includes all relevant metadata, such as issuer, expiration date, and certificate chain.
- **Wireless**: runZero active scans also enumerate wireless access points within range of the Explorer running the task. This inventory includes the BSSID, SSID, encryption settings, and signal strength.
- **Users and Groups**: runZero imports user and group information from directory services, including Active Directory, Azure Active Directory (Entra ID), and Google Workspace. The user and group inventories can be used to identify accounts with specific attributes, such as expired passwords and excessive group permissions.

Reports

runZero includes a comprehensive set of reports that cover everything from layer-2 topology maps to outliers and asset risk. In addition to pre-defined reports, most attributes within an asset or service can be used to create a grouped report with a single click. For deep customization, the runZero Export API provides CSV and JSON(L) formats with arbitrary search filters, which can be used to drive analytics platforms like Tableau and PowerBI.

Monitoring and Alerts

runZero provides monitoring and alert capabilities that can trigger based on changes to the inventory, new results for custom search queries, and any system-level event (of dozens). These alerts can be delivered either in-product, by email, or to a webhook destination, including Slack channels. In addition to alerts, rules, and custom queries, goals and custom dashboard widgets can be defined to track progress towards a specific outcome.

Getting started

To get started, you'll need to sign up for a runZero account. The default account is a trial of the full runZero Platform. After the trial expires, you will have the option to convert to the free Community Edition or purchase a subscription.

- Sign up for a runZero account
- Read up on creating an account for help activating your account, changing your password, and adding a profile picture.
- Once your account is set up, there are a couple of paths you can take to deploy runZero.
    - Quickstart
    - Full deployment plan

Quickstart guide

The quickstart path is ideal for those who want to jump into using runZero and explore its core functionalities. This section covers the initial setup, running basic scans, and configuring integrations.

1. Set up an Explorer

To begin asset discovery, you need to deploy an Explorer:

- Navigate to Deploy > Deploy Explorers in the runZero Console.
- Review the system requirements and choose the appropriate binary for your platform.
- Ensure the Explorer connects to your organization. Each download link is organization-specific.

2. Run your first scan

Once the Explorer is installed:

- Go to Scan in the console.
- Click Start standard scan and configure the discovery scope:
  - Choose an Explorer.
  - Set the target subnets or CIDR ranges.
  - Run the scan and monitor progress from the console.

> Note
> We recommend starting with a `/24` or handful of small CIDR blocks to verify connectivity in the network and then scaling out from there.

3. Configure integrations

Enhance your data with third-party integrations:

- Navigate to Integrate.
- Select and configure supported integrations for EDR, Vulnerability Management, Cloud, etc.
  - Documentation for each inbound integration is available.
- Ensure appropriate permissions are set in the integrated systems.
- Run sync tasks to import data and merge with existing asset information.

4. Review scan results

After the scan completes:

- Check results under Inventory > Assets.
- Use the Query library to better understand misconfigurations and other noteable findings in the inventory.
- Review reports and customize dashboards for deeper insights.
- Here are a few sample reports to check out:
  - SNMP default communities
  - SSH authentication methods
  - TLS supported versions

> Note
> If you have no results with the `matches:>0` search, you may need to wait for the metrics to be calculated. You can see the status of the metrics calculations here.

Full deployment plan

For organizations planning a production deployment, careful preparation ensures a smooth and scalable setup. Follow these steps:

1. Plan your deployment

- Determine the goals for asset visibility, network segmentation, or compliance.
- Ensure adequate resources for Explorers and network access points.
- Familiarize yourself with runZero-specific terms in the glossary.

2. Initial setup

- Deploy Explorers across key network segments.
- Configure credentials for access to devices and services (e.g., SNMP, EDR, MDM, etc).
- Use network communication guidelines to verify connectivity.

3. Run discovery and assess

- Perform discovery scans on all required segments.
- Validate data integrity and optimize scan settings as needed.
- Identify any scanning gaps and adjust configurations.

4. Automate and optimize

- Implement scan schedules for continuous monitoring.
- Use integrations for enriched asset data.
- Set up alerting and reporting tailored to your organization’s needs.

5. Rollout and monitor

- Deploy across remaining network segments based on your plan.
- Monitor scan results and network health regularly.
- Document configurations and processes for ongoing management.

For detailed instructions, refer to the production deployment documentation. For assistance, please contact support.

Creating an account

To get started, you'll need to sign up for a runZero account. The default account is a trial of the full runZero Platform. After the trial expires, you will have the option to convert to the free Community Edition or purchase a subscription.

Sign up for a runZero account

Activating your account

After you sign up for an account, we'll email you a link to activate your account. If you don't see an email from us, check your spam folder.

Open the link in the email to go to the **Activation** page. Follow the instructions on the page to activate your account. You'll need to provide your name, set up a password, specify your location, and accept our privacy policy and terms of service.

After activating your account, you'll be taken directly to the runZero Console. Your new account has administrative access, so you will be able to manage sites, organizations, users, and Explorers.

If you have any trouble creating your account, please contact support.

Changing your password

To change your password, go to your account settings. You'll need to provide your current password before you can enter a new one.

All passwords must contain:

- At least 8 characters
- At least 1 uppercase character
- At least 1 lowercase character
- At least 1 number

Updating your profile picture

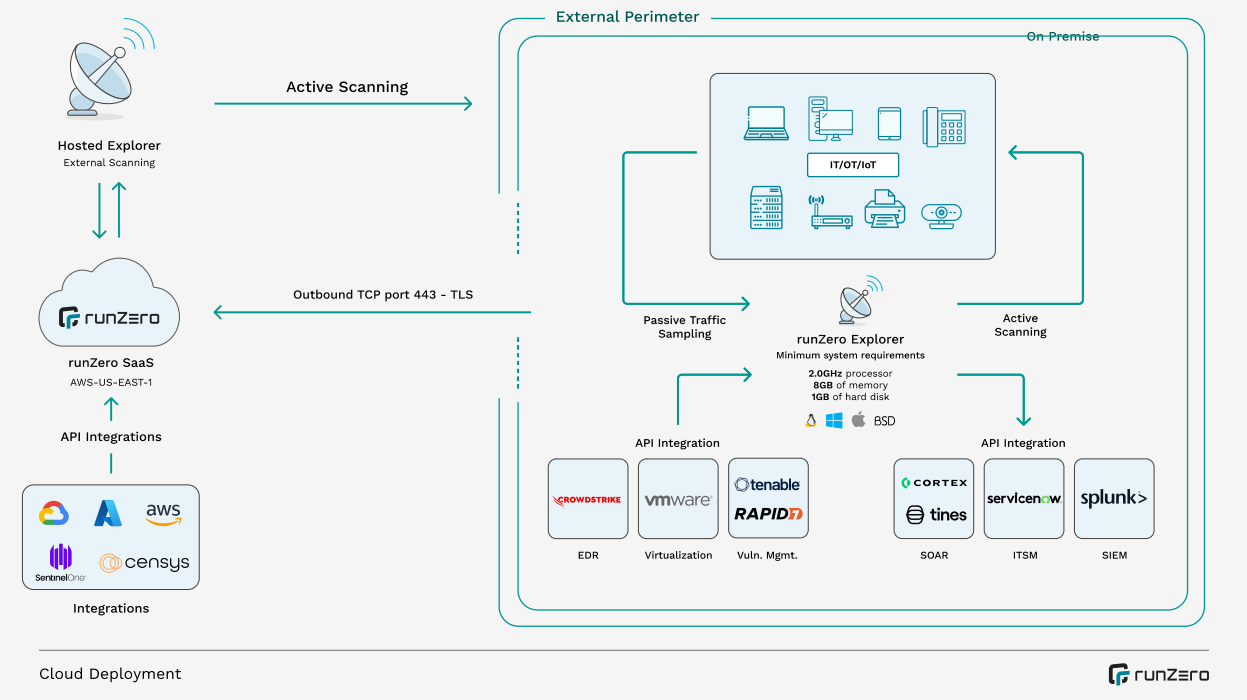
User profile images are managed through Gravatar and associated with your email address. If you don't have an account, sign up for one.

Installing an Explorer

runZero requires the use of at least one Explorer within your environment to enable active and passive network discovery. The Explorer should be installed on a system with reliable connectivity to the network you want to discover. For internal networks, runZero works best when installed on a system with a wired (vs wireless) connection.

For external network discovery, nearly any cloud provider with a reliable connection should do. If the runZero Explorer is installed in a container or virtualized system, ensure that it has direct access to the network (host networking in Docker, bridged networking in VMware, etc). SaaS customers with a Platform license can use the runZero hosted Explorers at no additional cost.

View an interactive version of this diagram



Installation

To install the runZero Explorer, sign in to the runZero Console and switch to the Organization that should be associated with the Explorer. Explorer downloads are then available by selecting Deploy in the left navigator and choosing the Deploy Explorers sub-menu.

Note: The Explorer download link is specific to your active organization and **using the wrong link can result in a new Explorer being associated with the wrong organization**.

Download the correct binary for your system from the Explorer download page. For most systems, select the 64-bit (x86_64) architecture. For macOS, you will need to select 64-bit

Intel (x86_64) or ARM (Apple M*), depending on your hardware. For embedded devices, such as the Raspberry Pi 3+, choose the ARM7 architecture. Windows binaries are signed with a valid Authenticode signature, which should be validated before the executable is launched.

The Explorer installation process requires administrative privileges. On Windows, a UAC prompt may be displayed. On Linux and macOS the downloaded binary should be made executable (`chmod u+x runzero-agent.bin`) and then executed with root privileges (`sudo` or from root shell). In either case, the Explorer should install itself as a system service and start immediately, displaying a new entry in the Explorers page.

System requirements

Windows

- Windows Server 2012 R2+ or Windows 10 Build 1604+
- Processor running at 2.0 GHz or faster
- At least 16GiB of memory (8GiB for small environments)
- At least 1GB of free storage space

Limitations

- The Trellix agent for Windows appears to interfere with network scans and traffic capture. Please use a system without the Trellix agent for active scans or passive network sampling. Alternatively, switching Trellix HIDS agent from “block” to “monitor” may allow successful scans.
- Windows Server 2008, Windows Server 2012, Windows 7, and Windows 8 may be able to run the Explorer in a pinch, but are not officially supported.
- Windows Explorers are limited to a single concurrent scan task due to performance limitations of the raw packet driver.

Linux

- Kernel version 2.6.23 or later
- Processor running at 2.0 GHz or faster
- At least 16GiB of memory (8GiB for small environments)
- At least 1GB of free storage space

Linux ARM devices with limited processing power and memory, such as the Raspberry Pi, can run the runZero Explorer, but may have trouble scanning larger networks, or running integrations with large amounts of data.

MacOS

- macOS 10.11 (El Capitan) or newer
- Processor running at 2.0 GHz or faster

- At least 16GiB of memory (8GiB for small environments)
- At least 1GB of free storage space

macOS systems running Catalina (10.15) or newer need to use the **curl** download method to avoid issues with the new Notary requirements.

BSD variants

- Processor running at 2.0 GHz or faster
- At least 16GiB of memory (8GiB for small environments)
- At least 1GB of free storage space

Requires root access to a system running a recent version of the operating system. FreeBSD 11.2 or newer, recent versions of NetBSD/DragonFly/OpenBSD.

Web screenshots

Google Chrome should be installed on the Explorer system to enable web screenshots. Please note that "snap"-based Chromium installs (Ubuntu 20.04 and newer) don't appear to work properly in headless mode and the official Chrome packages should be used instead.

To install the latest Chrome package on Debian-based Linux installations (including Ubuntu):

```
curl -o chrome.deb https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb && \
    sudo apt install ./chrome.deb
```

To install the latest Chrome package on RedHat-based Linux installations (including Fedora, CentOS, Rocky, and Alma):

```
curl -o chrome.rpm https://dl.google.com/linux/direct/google-chrome-stable_current_x86_64.rpm && \
    sudo dnf install ./chrome.rpm
```

Configuration

> Note
> Existing installations may refer to `Rumble` in the directory name, service name, and binary names. `Rumble, Inc`` was the previous name of our business and new installations will only refer to `runZero`, our new name. going forward.

The Explorer can be configured by setting variables in a `config.txt` file located in the same directory as the executable. On Windows this file should be created in `C:\Program Files\runZero\config.txt`, while other platforms should use `/opt/runzero/bin/config.txt`. In addition to `config.txt`, the legacy `.env` name is also supported. If the file `/etc/runzero/config` is present, this will be preferred as the configuration settings for the Explorer.

The format of this file is `VAR=VAL` with one variable per line.

Configuration file locations

Windows: `C:\Program Files\runZero\config.txt`

Other Platforms: `/opt/runzero/bin/config.txt`

Network communication

The Explorer connects to the runZero Console on TCP port 443 using TLS. This connection is used for Explorer registration, job scheduling, status messages, and submission of completed scan jobs. For completely offline environments, the runZero CLI can be used to create scan data files that can be uploaded later via the Inventory `Import` action. The console is used for automatic updates of the Explorer executable. The specific IP addresses used depend on your deployment model and region. For customers using the SaaS console, the following static IPs are associated with the runZero consoles.

United States

The console hostname is `console.runzero.com`.

IPv4

- 13.248.161.247
- 76.223.34.198

IPv6

- 2600:9000:a415:cd87:fbe5:476a:3533:69f2
- 2600:9000:a716:ee91:85f9:3c9:48c9:59b9

Germany

The console hostname is `console-eu.runzero.com`.

IPv4

- 15.197.131.232
- 3.33.248.90

IPv6

- 2600:9000:a603:e925:542d:6d40:6897:bc3a
- 2600:9000:a70e:635f:71bd:bb0a:8e43:9466

NAT

The Explorer should be run on a system with a full bidirectional network connection. It should **not** be run on a system behind a NAT gateway. This includes virtual machines, which should be set up with bridged network adapters.

Juniper DDoS protection is known to break runZero scans.

Proxy support

Please note that certain web proxies that perform TLS inspection do not handle Websocket communication properly and TLS inspection will need to be disabled for the runZero Explorer to successfully connect. The most popular product with this problem is the Sophos (previously Cyberoam) security appliance. Websense users may need to add a bypass rule for `console.runzero.com`.

Proxy support is handled automatically in most cases. On the Windows platform, proxy information is read from the registry keys (used by Chrome, Edge, and IE).

The proxy can also be configured by setting the `HTTPS_PROXY` environment variable. The value of the `HTTPS_PROXY` environment variable should be a hostname and port (`proxy.example.com:8080`) or just a hostname (`proxy`). Environment variables are read from your configuration file. Please view the Configuration section to see how to set environment variables. The common option is to create a file named `config.txt` in the same directory as the Explorer binary and set the environment variables in the format described below.

It's also possible to use a SOCKS proxy by setting `HTTPS_PROXY` to a `socks5` URL, for example `HTTPS_PROXY=socks5://socks.example.com:1080`.

The Explorer will attempt to use the configured proxy for each probe. If it doesn't succeed, it will try making a direct connection. This means both proxy and non-proxy connection attempts may appear in logs. Using only proxies to try to hide or anonymize Explorer connections is not supported.

TLS configuration

The minimum and maximum version of TLS used by the Explorer for outbound communication to the console can be configured through environment variable and the configuration file. The `TLS_VERSION_MIN` and `TLS_VERSION_MAX` variables can be set to any of `1.0`, `1.1`, `1.2`, and `1.3`. The default configuration is to use a minimum version of TLS 1.2 and a maximum of TLS 1.3. If a maximum version is set to a value lower than the minimum value, the maximum will be set to the minimum value. runZero does not recommend using TLS versions prior to 1.2.

The following example will configure the Explorer to only speak TLS version 1.3.

```
TLS_VERSION_MIN=1.3
TLS_VERSION_MAX=1.3
```

Restarting an Explorer

The easiest way to restart an Explorer is to force a software update from the cloud console. Otherwise, you can find the service on the host machine and restart it by hand.

On Linux or Mac, you can run `/opt/runzero/bin/runzero-agent-[uuid] restart` where `[uuid]` is the ID of the organization the Explorer belongs to.

On Linux systems you can also use `systemd` to restart Explorers. First obtain the name of the Explorer (runzero-agent-[uuid]) service:

```
systemctl | grep runzero-agent
```

Then restart the service using this name:

```
systemctl restart runzero-agent-[uuid]
```

A `kill -9` of the Explorer pid should cause a restart as well.

On macOS, you can use `launchctl` to restart the Explorer:

```
launchctl kickstart -k runzero-agent-[uuid]
```

As with Linux, the `[uuid]` is the organization UUID, which you can find by looking at the `runzero-agent-*` filename in `/opt/runzero/bin`

On Windows, you can use the Services console to restart the Explorer like any other background service.

Removing an Explorer

The easiest way to remove an Explorer is to use the Explorers page and locate the Explorer you want to remove, then click the *Delete Explorer* trash bin icon under the *Actions* column. This will remove the service and terminate the current Explorer process. In addition, you can remove all online Explorers by clicking the *Manage All Explorers* menu and choose the *Uninstall All Online Explorers* option. If you would like to remove an Explorer without using the runZero Console, there are a couple of options.

On the Windows platform, each Explorer will be listed in Programs and Features (as the runZero Agent), and can be uninstalled like any other application.

On all platforms, including Windows, the Explorer can uninstall itself if run with the `uninstall` argument from a root or Administrator shell:

Removal on Windows

```
c:\Program Files\runZero\runzero-agent-[uuid].exe uninstall
```

Removal on Other Platforms

```
/opt/runzero/bin/runzero-agent-[uuid] uninstall
```

Log management

The Explorer logs to a file and to standard output by default. On Windows the default log file location is the installation directory (`C:\Program Files\runZero`) while other platforms log to the files `/var/log/runzero.log` and `/var/log/runzero.err`. The default configuration limits log files to 100MiB, creates three backups, and expires logs after 90 days. These defaults can be be changed by setting the following values in the `$BIN/config.txt` file:

- The `RUNZERO_AGENT_LOG_MAX_SIZE` setting controls the maximum log size in mibibytes. The default is **100**.
- The `RUNZERO_AGENT_LOG_MAX_BACKUPS` setting controls the number of backup files created by log rotation. The default is **3**.
- The `RUNZERO_AGENT_LOG_MAX_AGE` setting controls the maximum age in days, this applies to all files, including backups. The default is **90**.
- The `RUNZERO_AGENT_LOG_COMPRESS` setting determines whether to gzip compress the backups. The default is **false**.
- The `RUNZERO_AGENT_LOG_STDOUT` setting determines whether to write logs to standard output. The default is **true**. On Linux this results in logs being written to the system log when the Explorer is started by systemd or upstart. On macOS this results in separate logs viewable in the Console application under “Log Reports” when the Explorer is started by launchd.
- The `RUNZERO_AGENT_LOG_FILE` setting determines whether to write logs to a log file as described above. The default is **true**, set to `false` to disable log file writing.

The Explorer must be restarted for these settings to take effect.

Certificate Authorities (CAs)

The runZero Explorer uses the system-installed certificate authorities to validate TLS connections in addition to an internal CA certificate bundle. By default, both the system certificate roots, and the bundled roots are considered for all secure TLS connections, including Starlark-based custom integrations. This behavior can be controlled via environment variables (set in the `$BIN/config.txt` file or at the system level):

- The `RUNZERO_TLS_IGNORE_SYSTEM_ROOTCA` setting can be set to **true** to ignore the system CA roots.
- The `RUNZERO_TLS_IGNORE_EMBEDDED_ROOTCA` setting can be set to **true** to ignore the bundled CA roots.
- The `RUNZERO_TLS_ADDITIONAL_ROOTCA` setting can be set to a file path containing additional CA roots in PEM format.

Manual mode

If a supported system service manager, such as systemd or upstart, is not detected, the runZero Explorer will switch to manual mode, running in the foreground, and replacing and re-executing its own binary as new updates become available. For temporary Explorer installations or to run the Explorer in a container environment, the argument “manual” can be specified:

```
$ sudo ./runzero-agent.bin manual
```

Storage locations

The runZero Explorer installs into `%PROGRAMFILES%\runZero` on Windows and `/opt/runzero/bin` on all other platforms. Temporary files are stored in the default operating system locations. These locations can be overridden using the `config.txt` file (see the above Configuration section). Note that the Explorer service needs to be restarted (or force updated) for these changes to take effect. Older installations may still refer to the `Rumble` versions of the previously mentioned directories and `.env` instead of `config.txt` for environment overrides.

On Windows, the temporary file location is chosen from the first non-empty environment value of `TMP`, `TEMP`, or `USERPROFILE`, falling back to the Windows directory. To override this location, set an entry in `config.txt` like the following:

```
TMP=D:\Storage\runZero
```

On all other platforms, the temporary file location is chosen based on the value of `TMPDIR`, falling back to `/tmp` otherwise. To override this location, set an entry in `config.txt` like the following:

```
TMPDIR=/home/storage/runzero
```

Any scans that fail to upload are stored in the runZero Explorer installation directory and can be imported into the platform manually or using the runZero CLI’s `scan --import` and `scan --upload` options.

Container installations

The runZero Explorer can run in standard container environments, but may require additional configuration. To run as a standalone executable, the Explorer can be run with the argument `manual`. For non-persistent containers an Explorer identifier needs to be persisted through an environment variable. This can be done by setting the variable `RUNZERO_AGENT_HOST_ID` to a 32-character hexadecimal string. This identifier is used to uniquely identify the Explorer within an organization.

To generate a suitable identifier, the `openssl` tool may be used:

```
openssl rand -hex 16
01b0283809b24511929d0b062bd36109
```

Here is a sample Containerfile you can edit and use:

```
#
# Sample Containerfile for running the runZero Explorer in a container, with
# screenshot support.
#
FROM debian:stable-slim

WORKDIR /opt/runzero

# Ensure curl is available and install tools for wireless scanning.
#
RUN apt update && apt install -y curl wireless-tools

# Install Chrome for screenshots.
#
RUN curl -o chrome.deb https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb && \
    apt install -y ./chrome.deb

# Set AGENT_URL to be the download URL for your Linux runZero Explorer. To
# find your URL, go to https://console.runzero.com/deploy/download/explorers
# and click on the first URL box to copy it to the clipboard.
#
ENV AGENT_URL=https://console.runzero.com/download/explorer/DT[uniqueToken]/[versionID]/runzero-agent-linux-amd64.bin

# This ID is used to track the Explorer even if the container is rebuilt.
# Set it to a unique 32 character hex ID. You can generate one via:
#
# $ openssl rand -hex 16
#
ENV RUNZERO_AGENT_HOST_ID=[UNIQUE-ID]

# If you need to set environment variables to change the Explorer behavior,
# you can do so via the ENV directive. Example:
#
# ENV RUNZERO_AGENT_LOG_DEBUG=true

ADD ${AGENT_URL} runzero-agent.bin

RUN chmod +x runzero-agent.bin

# For full functionality the runZero CLI needs to send and receive raw
# packets, which requires elevated privileges.
USER root

# The argument `manual` tells runZero not to look for SystemD or upstart.
ENTRYPOINT [ "/opt/runzero/runzero-agent.bin", "manual" ]
```

This containerfile works with podman as well as Docker. Note that because of the requirement for root privileges, you should start the container as root. For the best results, run the container with the `--privileged` option to allow the Explorer to listen to network traffic.

Automated installations

The Explorer will automatically install when executed if root or administrative privileges are available.

On Linux and BSD systems, automatic installation depends on the presence of a supported init service like systemd or upstart. If no supported init service is found, the Explorer will instead run in manual mode, automatically overwriting and re-executing itself with each update. To automatically deploy an Explorer on systems without a supported init service, the Explorer should be executed in the background and with the `nohup` wrapper.

On Windows systems, the Explorer will automatically install when run interactively or when the `updater` parameter is passed to the binary. For environments where MSIs are required, the Explorer MSI wrapper can be used to deploy an Explorer from the runZero Console or a local mirror.

AWS EC2 installations

The runZero Explorer can be run in an AWS EC2 instance. However, there are a number of configuration changes required to avoid packet loss when scanning.

- DNS resolution from EC2 to the AWS DNS server has a fixed cap of 1024 packets per second which cannot be increased. To avoid this, set a custom list of non-AWS nameservers in the scan configuration advanced section.
- Any Security Group without a `0/0` rule results in connection tracking, which has an undocumented limit on connections per instance type. Avoid this by adding `0/0 allow` for ICMP/UDP inbound and outbound for the Explorer instance. Also add a `0/0 allow` for outbound TCP connections. (Inbound TCP is not currently required for runZero scans, but may be needed in the future for callback protocols.)
- Overall packet rates have undocumented limits which depend on instance type. You will need to experiment with sizing your Explorer instances until scans are consistent for a given scan rate. We hope to gather and share data on appropriate instance sizes soon.

The Explorer should work well deployed to a memory optimized, compute optimized, or general compute instance. Since the Explorer can make full use of multi-core systems, you may want to target the number of cores to the number of simultaneous scans. You probably won't want to run larger scans on an instance with less than 32GiB of RAM.

Automated MSI deployments

runZero uses dynamically generated binaries for the runZero Explorer downloads and this doesn't always play well with MSI-based installation methods.

To work around this issue, we have provided a shim MSI package that can be used with automated installers. This package has a valid Authenticode signature and can also be verified using the runZero Verifier.

> Note
> Some components of the application still reference the name "Rumble" for backwards compatibility. All new installations will use runZero for directory, file, and user names.

To use this package, deploy it with the `URL` parameter specified as the organization-specific download URL from the runZero Console Explorers section.

```
msiexec /i runzero-explorer-installer-amd64.msi URL=https://console.runzero.com
/download/explorer/DT[uniqueToken]/[versionID]/runzero-explorer-windows-amd64.exe
```

> Note
> The above command should be entered as one long line with a single URL parameter, but is shown wrapped here.

The MSI shim will verify that the URL contains a valid runZero Explorer binary and install it normally.

Warning: Note that installing with the `/a` parameter will not work and `/i` must be used instead.

Binary downloads

Build	SHA256
runZero Explorer Installer MSI x86 64-bit	sha-256
runZero Explorer Installer MSI x86 32-bit	sha-256

Installing on a Raspberry Pi

The runZero Explorer enables discovery scanning. In most cases, you can deploy an Explorer on an existing system that has connectivity to the network you want to discover. However, there may be times when the traditional deployment model may not work for you. Some locations, like retail stores or customer sites, may not have staff or hardware available to install the Explorer, making remote deployment a bit tricky.

In these types of scenarios, you can install a runZero Explorer on a Raspberry Pi and send the device to the location for them to plug into their network.

What you'll need

- Raspberry Pi 4 Model B (4GB or 8GB), Raspberry Pi Compute Module 4, or Raspberry Pi 400
- At least 1GB free of storage space on your Raspberry Pi's MicroSD card after installing the operating system

Connecting to your Raspberry Pi via SSH

In this section, we're going to show you how to SSH to your Raspberry Pi and install the Explorer from your terminal.

Step 1. Enable SSH on your Raspberry Pi

Before you can connect to your Raspberry Pi, make sure to enable SSH on the device.

You can add a file called ssh.txt at the root of the SD card and reboot the Raspberry Pi. The contents of the file can be empty. On reboot, the Pi looks for the SSH file and enables SSH if it finds the file.

Step 2. Update the password for your Raspberry Pi

If you enable SSH on your Raspberry Pi, you must update your credentials.

Log in as the `pi` user and use the `passwd` command to change the default password. Entering the `passwd` command will prompt you for your current password to authenticate before you can change it.

Step 3. Copy the Explorer instructions

- Go to your console. Verify you are in the right organization. runZero keys your Explorer's download link to the organization you are currently viewing to associate them together.
- Go to the Explorer deployment page and select **Linux Distributions**, and then choose Linux ARM 32-bit V7. Note that Raspbian uses a 32-bit kernel by default, even on 64-bit Raspberry Pi hardware.
- Click the instructions at the bottom of the Linux installation page to place the commands into your clipboard.

Step 4. Install the Explorer

> Note
> Some components of the application still reference the name "Rumble" for backwards compatibility. The documentation will be updated as these are changed.

Paste the installation command into the terminal after connecting to the Raspberry Pi via SSH. The URL in the command links the installation to your active organization and will be different from the example below.

```
me@mac ~ % ssh pi@hostIP
pi@host's password:
pi@raspberrypi:~ $ sudo bash
```

```
root@raspberrypi:/home/pi# cd Downloads
root@raspberrypi:/home/pi# curl -o runzero-explorer.bin https://console.runzero.com/download/explorer/DT[uniqueToken]/|
```

The Explorer automatically installs itself in `/opt/runzero` and sets up a systemd service with the name `runzero-agent-`. The service automatically starts on boot.

Next steps

Your Raspberry Pi is now set up to use as a runZero Explorer box and ready to be sent wherever you need.

You can always view and manage your registered Explorers from your console.

Managing Explorers

The runZero Explorer is a lightweight scan engine that enables network and asset discovery. You should have at least one Explorer deployed. After deployment, you can manage your Explorers from the Deploy page in your runZero web console.

Viewing all Explorers

For each Explorer, you can see:

- The Explorer status (whether it is communicating with runZero)
- The OS it is running on
- Its name
- Any site it is associated with
- Its IP addresses
- The software version it is running
- Whether the version of npcap installed is up-to-date, if the OS is Windows (see upgrading npcap below)
- The CPU architecture of the host machine
- Any tags associated with the Explorer
- The status of its last scan
- Its capabilities, like Chrome support

Screenshot capabilities

To capture screenshots, Chrome must be installed. You can check if an Explorer has screenshot capabilities by looking for the Chrome icon in the *Capabilities* column.

Here's what each icon means:

- Green icon - The Explorer has access to a Google Chrome binary and can take screenshots.
- Red icon - No suitable Chrome binary was found.

Searching for Explorers

You can use the search bar to find Explorers. The query syntax is similar to other search bars in runZero, with keywords to filter by specific fields:

Keyword	Search by	Example
`arch:`	CPU architecture	`arch:amd64`
`name:`	assigned name	`name:scanner.local`
`address:`	IP address	`address:10.0.1.200`
`capability:`	capabilities	`capability:screenshot` or `capability:aws`
`tag:`	assigned tag	`tag:dev`
`npcap_version:`	npcap version	`npcap_version:1.31`

Explorer actions

Each Explorer has a set of action buttons that allow you to:

- **Reinstall an Explorer** - Performs a reinstall or upgrade of the Explorer. The current Explorer will download the latest Explorer code from runZero, and then run the install process.
- **Configure an Explorer** - You can associate the Explorer with a specific site, and add tags to it. You can also set the maximum number of concurrent scans allowed. A single Explorer can be configured to run multiple tasks at once.
- **Reassign an Explorer** - You can reassign an Explorer to a different organization within your account or even to a different runZero client account entirely.
- **Remove an Explorer** - If the Explorer is running, the Explorer will be asked to uninstall itself from the host machine. If the Explorer is not running, you can still tell runZero to forget about it. This is useful if you have decommissioned the machine the Explorer was running on or uninstalled the Explorer manually. If the Explorer runs again after runZero has been told to forget it, it will be readded to the registered Explorers list.

Bulk management operations

Bulk operations allow you to perform a set of actions to multiple Explorers at one time. Bulk actions are available from the *Manage all Explorers* menu.

You can bulk:

- **Update all online Explorers** - Tells all Explorers–that are up and communicating with runZero–to upgrade their software.
- **Forget all offline Explorers** - Clears all Explorers currently offline, and makes runZero forget them. No data will be lost. If any of the Explorers are reactivated, they will be added back to the active list.

- **Uninstall all online Explorers** - Tells all online Explorers to uninstall themselves from their host systems.
- **Automatically assign sites** - Runs through all of the Explorers that are not currently assigned to a specific site. It checks their IP address against the CIDR IP ranges of the registered subnets of all sites in the current organization. If the Explorer's IP address only matches a single site, the Explorer is assigned to that site.

Viewing Explorer details

Clicking on an Explorer's name takes you to a page showing the diagnostic information for that Explorer, including its software version, available memory, and network interfaces.

At the bottom of the page is a diagnostics text area. Clicking the *Update Diagnostics* button will fetch an updated list of all sub-processes active within the Explorer. This is useful to send to runZero support if you are having problems with a particular Explorer.

Traffic sampling

Community Platform

The Explorer details page is also where users can configure traffic sampling.

1. From the **Registered Explorers** page, select the Explorer you wish to configure to perform traffic sampling.
2. In the traffic sampling card, configure the following options:
    - **Site**: Specify the site the assets discovered as a result of Traffic Sampling will be added to.
    - **Discovery scope**: List the IP addresses or CIDR networks that traffic sampling will observe on this Explorer.
    - **Asset tags** (optional): List the tags you want applied to assets discovered through traffic sampling.
    - **Excluded hosts** (optional): List the IP addresses or CIDR networks that traffic sampling will exclude from the results.
    - **Interfaces**: Toggle the switches for the interfaces you want this Explorer to listen on.
3. Click **Save** to save your configuration and initiate the traffic sampling task.

Once configured, traffic sampling can be disabled by returning to this page and toggling off the selected interfaces. Upon saving, the traffic sampling tasks will automatically stop.

Upgrading npcap

On Windows, runZero uses a licensed third-party library called npcap for access to raw network traffic. Other software installed on the Explorer's host machine may also use npcap, and sometimes will have installed obsolete versions of the software. This can cause reliability problems.

runZero will alert you to obsolete versions of npcap by displaying a warning icon in the list of Explorers.

However, runZero cannot yet reliably upgrade npcap for you. runZero can't automatically upgrade npcap/winpcap, as it tends to be shared between applications, and forcing an upgrade from the runZero side can break other services (EDRs, Wireshark, etc).

To upgrade npcap manually:

1. Stop any running runZero services. This can be done using the Windows Services app. You'll need to look for "runZero Network Discovery Explorer".
2. Stop any other running software which uses npcap.
3. Uninstall Winpcap and any npcap installations via the Windows Control Panel.
4. Reboot the computer.

runZero will restart automatically, and install the latest npcap.

Verifying binaries

runZero uses dynamically generated binaries for the runZero Console, CLI, and Explorer downloads. Although Windows binaries have a valid Authenticode signature, all binaries also contain a secondary, internal signature. Dynamic binaries make it easy to deploy Explorers that connect back to the right organization, but present a challenge for independent integrity validation. To enable verification of the internal signature, we offer the **runZero Verifier**. This verification tool can confirm whether a given binary contains a valid internal signature, in addition to any existing Authenticode signatures.

To get started, download the latest version of the verifier from the bottom of this page along with the PGP signature file for the selected architecture.

The runZero Verifier is always signed by PGP Key ID `60EBAAE9AEF08C6D`.

To validate the signature of the runZero Verifier for Windows 64-bit, you will need a GPG client and to run the following commands:

```
C:\> curl -s https://www.runzero.com/.well-known/security.pub.asc | gpg --import
C:\> gpg --verify runzero-verifier-3.1.0-windows-amd64.exe.asc
```

Successful validation will show a valid signature by key ID `9B5DAFF7D43349298A3039BD60EBAAE9AEF08C6D`.

```
gpg: Signature made Sun 07 Aug 2022 11:33:15 AM CDT
gpg:                using RSA key 9B5DAFF7D43349298A3039BD60EBAAE9AEF08C6D
gpg:                issuer "security@runzero.com"
gpg: Good signature from "runZero Security <security@runzero.com>" [unknown]
```

The warning below is expected and does not indicate a problem with the signature:

```
gpg: WARNING: This key is not certified with a trusted signature!
gpg: There is no indication that the signature belongs to the owner.
```

Once the runZero Verifier itself has been validated, it can be used to check the signature of any runZero binary:

```
C:\> runzero-verifier-3.1.0-windows-amd64.exe runzero-explorer-3.1.0-windows-amd64.exe
runzero-explorer-3.1.0-windows-amd64.exe: VALID SIGNATURE
```

A failed validation will show the error `Invalid or missing signature` and the verifier will set exit status to 1.

Binary downloads

Windows

Build	PGP sig	SHA hash
runZero Verifier x86 64-bit	pgp signature	sha-256
runZero Verifier x86 32-bit	pgp signature	sha-256

Linux

Build	PGP sig	SHA hash
runZero Verifier x86 64-bit	pgp signature	sha-256
runZero Verifier x86 32-bit	pgp signature	sha-256

Additional Linux builds

Build	PGP sig	SHA hash
runZero Verifier ARM v5 32-bit	pgp signature	sha-256
runZero Verifier ARM v6 32-bit	pgp signature	sha-256
runZero Verifier ARM v7 32-bit	pgp signature	sha-256
runZero Verifier ARM 64-bit (aarch64)	pgp signature	sha-256
runZero Verifier PPC 64-bit Little Endian	pgp signature	sha-256
runZero Verifier MIPS 32-bit Big Endian	pgp signature	sha-256
runZero Verifier MIPS 32-bit Little Endian	pgp signature	sha-256
runZero Verifier MIPS 64-bit Big Endian	pgp signature	sha-256
runZero Verifier MIPS 64-bit Little Endian	pgp signature	sha-256
runZero Verifier S390X	pgp signature	sha-256

MacOS

Build	PGP sig	SHA hash
runZero Verifier x86 64-bit	pgp signature	sha-256
runZero Verifier ARM 64-bit	pgp signature	sha-256

BSD Variants

FreeBSD

Build	PGP sig	SHA hash
runZero Verifier x86 64-bit	pgp signature	sha-256
runZero Verifier x86 32-bit	pgp signature	sha-256
runZero Verifier ARM v6 32-bit	pgp signature	sha-256
runZero Verifier ARM v7 32-bit	pgp signature	sha-256

NetBSD

Build	PGP sig	SHA hash
runZero Verifier x86 64-bit	pgp signature	sha-256
runZero Verifier x86 32-bit	pgp signature	sha-256
runZero Verifier ARM v5 32-bit	pgp signature	sha-256
runZero Verifier ARM v6 32-bit	pgp signature	sha-256
runZero Verifier ARM v7 32-bit	pgp signature	sha-256

Dragonfly

Build	PGP sig	SHA hash
runZero Verifier 64-bit	pgp signature	sha-256

OpenBSD

Build	PGP sig	SHA hash
runZero Verifier 64-bit	pgp signature	sha-256

Full-scale deployment

As you get started with runZero, we recommend kicking off with our standard deployment plan and adding tasks as needed. The standard deployment plan is broken out into six stages which will help you plan out your requirements, execute the deployment, and optimize your environment based on runZero's best practices.

Identify key success outcomes

Total attack surface visibility

- Active discovery on all internal assets
- Active discovery on all externally facing assets
- Passive discovery and enrichment in key network segments
- Integrate with all cloud providers and other relevant data sources

Additional Resources

- Overview
- RFC1918 coverage playbook
- Scanning OT playbook

Full-spectrum exposure detection

- Rapid Response findings and asset-level pivoting
- Network misconfiguration findings and control coverage gaps
- Vulnerability enrichment and inside-out findings

Additional Resources

- Overview
- Rapid Response blog
- Gaps in EDR playbook
- Gaps in VM playbook

Risk prioritization and insights

- Custom dashboards
- Rules and alerts
- Setting asset criticality and ownership

Additional Resources

- Overview
- Custom Dashboards documentation
- Alerting playbook

Compliance, Reporting, and KPIs

- Comply with asset inventory and discovery requirements of relevant frameworks

- Comply with secure configuration requirements of relevant frameworks
- Comply with malware protection requirements of relevant frameworks
- Comply with vulnerability management requirements of relevant frameworks

Additional Resources

- Overview
- Compliance alignment documentation
- NYDFS dashboard playbook

Planning your deployment

This first set of tasks will help your team identify target results, get ahead of potential blockers, and help you avoid misconfigurations within runZero.

Tasks

- Identify key organizational stakeholders
  - Administrator(s) - who will be setting up runZero?
  - Integration owner(s) - who will provide credentials for each integration?
- All users take the runZero 101 training
- Administrators take the runZero 201 training
- Determine whether self-hosting is required (docs | video)
- Identify known networks and subnets for discovery and other inventory sources (docs | video)
- Define organizations based on RBAC requirements (docs | video)
- Determine Explorer deployment location(s)

Initial configuration

Once you have your plan in place, it's time to execute and run your initial scans. **Please note** that these configuration tasks are in a prioritized order to help you avoid having to reconfigure things down the road.

Tasks

- Deploy self-hosted console (if required) (docs | video)
- Setup organizations
- Set up sites, and define subnets for discovery (video)
  - **Sites do not necessarily correspond to physical locations within runZero.** Sites are used to represent distinct networks that may have overlapping IP space
- Install Explorer(s) (video)
- Run initial scan (docs | video)

Analysis

Now that you have done some initial discovery, it's time to review the results. Reviewing the results and leveraging our reports will help you expand scan scope, better understand your network, as well as help you identify key issues such as misconfigurations.

Tasks

- Review results of initial scan
  - Review dashboard (docs | video)
  - Review the asset inventory (docs | video)
  - Review the asset detail view (docs | video)
- Identify risky assets using the Queries library (docs | clickthrough)
  - Learn query syntax
  - Apply vulnerability records to queries (docs)
- Track long-term initiatives with Goals
- Review reporting
  - Identify gaps in scanning (docs | video)
  - Understand network segmentation (docs | video)

Advanced configuration / Optimization

After you've done your initial analysis, you will want to optimize your scans and configure integrations to further build your complete asset inventory.

Tasks

- Configure inbound integration connections
  - Cloud inventory sources
  - Endpoint protection
  - Vulnerability management
  - Directory services
- Configure SNMP credentials (video)
- Optimize scans by adjusting scan rates and other configurations (docs | video)
  - See our clickthrough of some key additional configuration options as well

Automation

Now that you have optimized your scans and have analyzed your runZero data, you can automate these tasks to avoid manual effort. You can leverage this automation to run scans on a recurring basis, automate queries, and generate alerting for the team.

Tasks

- Schedule recurring scan tasks and any inbound integration tasks
- Automate queries and configure alerts to align with use cases (video)
- Configure outbound integration connections to enrich other IT and security tools
    - CMDB
    - SIEM
    - SOAR
    - **Note:** If you're utilizing a solution that runZero does not offer a standard outbound integration for at this time, be sure to review our API documentation to learn about how to export runZero data.

Rollout

As your runZero deployment comes to a close, you will want to ensure all users have gone through training and ensure anyone that would get value from runZero has access to the platform.

Tasks

- Add users
- Ensure all users are trained on runZero
    - Training and key documentation
        - 101 user training
        - 201 administrator training
        - Search
        - Reporting
        - Exporting assets
    - runZero playbooks
- Identify other teams interested in the asset inventory data, such as:
    - Enterprise security team
        - runZero is typically used by security teams to achieve a complete asset inventory, find gaps in their vulnerability scanning and endpoint protection, as well as discover potential vulnerabilities.
    - IT Operations team
        - runZero is typically used by IT Operations teams to achieve a complete inventory of all assets across on-premise and cloud-based infrastructure. This allows the team to identify misconfigurations as well as report on assets in the environment by leveraging our searching and reporting capabilities.
    - Penetration testing team
        - runZero is typically used by penetration testing teams for conducting reconnaissance both internally and externally, identifying vulnerable targets, and finding ways to get to these vulnerable targets by using our reporting and searching capabilities.

Additional Resources

Now that runZero has been deployed and users have been trained on the platform, please review some of our additional resources to help answer questions you might have as well as maximize the value of runZero:

- runZero playbooks
- Leveraging the API
- Glossary
- runZero FAQs

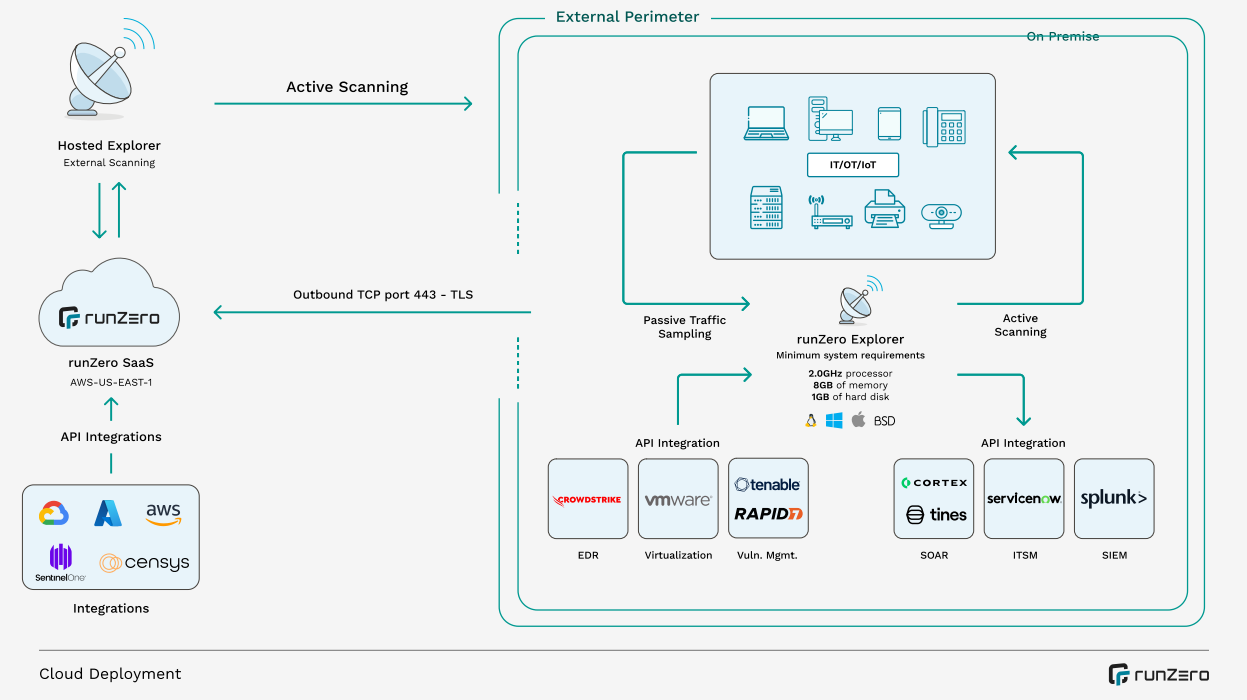
Getting help

If you need assistance at any point in this process, you can book a session with a runZero Customer Success Engineer to discuss further.

Types of networks

It is often helpful to use network examples as a starting point for planning your runZero implementation. This document breaks down a few standard network types and provides potential configurations for each. With that being said, every network has nuance, so it's likely there will be some differences for your implementation.

This is a basic overview of how discovery will be done using Explorers and scanners. By default, one Explorer will be deployed with the goal of running discovery on as much of the network as possible. If needed, more Explorers can be added for areas the primary Explorer cannot get to. You can also use a scanner for offline environments where there is no internet connectivity.



SaaS company

Network characteristics

- Flat on-premises network for offices
- Multi-cloud environment
- Hybrid workforce remote and in office

Sample runZero implementation

- Explorers
  - Corporate network Explorer that is able to get all on-premises network and cloud services private connectivity
  - One Explorer per VPC that isn't routable from on-premises networks
- Organizations
  - Single organization
- Sites
  - Primary corporate site
  - One site per VPC

Large corporation

Network characteristics

- Corporate offices with many connected branches

- Multiple M&A transactions happening at any given time, onboarding new offices every year
- Multiple data centers for internal and externally-facing applications

Sample runZero implementation

- Explorers
  - Corporate network Explorer that is able to get all on-premises networks
  - One Explorer per site with low bandwidth or legacy firewalls, proxies, etc.
  - One Explorer for each M&A transaction to understand new risks and keep data segmented
- Organizations
  - Corporate assets
  - One project per M&A transaction until the deal is complete and assets are merged
- Sites
  - Primary corporate site
  - Potentially separate sites if there is overlapping IP space in branch offices

Retail company

Network characteristics

- Flat on-premise network for corporate offices
- Data centers
- 100s of retail locations

Sample runZero implementation

- Explorers
  - Corporate network Explorer that is able to get all on-premise networks
  - One Explorer per retail location with low bandwidth or legacy firewalls, proxies, etc.
  - One Explorer per data center
- Organizations
  - Single organization
- Sites
  - Primary corporate site
  - One per retail site if they have overlapping IP space

Manufacturing company

Network characteristics

- Flat on-premise network for corporate offices
- OT environment is completely disconnected from the internet
- Field service agents VPN in from varying locations

Sample runZero implementation

- Explorers
  - Corporate network Explorer that is able to get all on-premise networks
- Scanners
  - Manufacturing plant that is not connected to the corporate networks
- Organizations
  - Single organization
- Sites
  - Primary corporate site
  - One per OT site if there is overlapping IP space

Telecommunications company

Network characteristics

- Flat on-premise network for corporate offices
- Segmented data centers
- IoT devices scattered across the country
- Field service agents VPN in from varying locations

Sample runZero implementation

- Explorers
  - Corporate network Explorer that is able to get all on-premise networks
  - One for each segmented lab
  - One for each data center
- Organizations
  - Corporate network
  - Production network
  - One per lab
- Sites
  - One primary site per organization

Managed security service provider

Network characteristics

- Many customers spinning up/down in engagements
- Mostly small, flat networks
- Some customers have OT environments that are sensitive

Sample runZero implementation

- Explorers
  - One per customer

- Scanners
  - Only needed for customers with offline networks
- Organizations
  - Project per customer initially
  - Promote project to organization for long-term use
- Sites
  - One per customer but potentially multiple if a customer has overlapping IP space

Academic organization

Network characteristics

- Many buildings/networks spread around the campus with varying connectivity
- Multiple data centers managed by different departments
- Multiple labs for research and development that are disconnected from the rest of the network and the internet

Sample runZero implementation

- Explorers
  - Main network Explorer that is able to get all on-premise networks
  - One Explorer per building with low bandwidth or legacy firewalls, proxies, etc
  - One Explorer per data center
- Scanners
  - One for each disconnected lab
- Organizations
  - Main assets
  - One for each lab
- Sites
  - One primary per organization but potentially multiple if buildings have overlapping IP space

Organizations

Community Platform

An organization represents a distinct entity; this can be your business, a specific department within your business, or one of your customers. All actions, tasks, Explorers, scans, and other objects managed by runZero are tied to specific organizations and isolated from each other.

Your active organization can be switched by using the dropdown selector at the top right of the runZero Console. If your default role is Viewer or higher, you can select **All Organizations** from the dropdown to view the data for all of your organizations in the Dashboard and Inventory. The Queries page also supports the All Organizations view. Pages that are not compatible with the All Organizations view will be hidden until a single organization is selected.

runZero Community Edition is limited to a single organization.

Use cases for organizations

It is common to segment your assets to multiple organizations based on a few common requirements:

- **RBAC**: Users will only be able to see the assets in the organizations or the children of organizations they're added to.
- **Environments**: It is possible you do not want your development, production, or other assets in one pool and would prefer them to be segmented.
- **Customers**: Service providers generally have one organization per customer.

Creating organizations

Organizations can be created, modified, and deactivated by going to the organizations section within the console. Click the Organizations link under Global Settings.

Organization configurations

- **Name**: Name of the organization.
- **Parent organization**: Allows for sub-organizations, limited to three levels of nesting. Permissions will be inherited based on most privilege.
- **Description**: description of the organization.
- **Stale asset expiration**: Sets the number of days before stale assets are removed from the inventory. This applies to online and offline assets not seen within the configured number of days.
- **Offline asset expiration**: Sets the number of days before offline assets are removed from the inventory. This applies to offline assets not seen within the configured number of days.

- **Stale integration attributes expiration**: Sets the number of days before stale integration attributes are removed from the inventory. Optionally keep the latest set of integration attributes per source, even if it is older than the threshold.
- **Stale vulnerability expiration**: Sets the number of days before stale vulnerabilities are removed from the inventory.
- **Scan data expiration**: Sets the number of days before completed scans are removed from the platform. This applies to the scan task metadata as well as the raw data files and change reports.

Projects

A project is a special type of organization designed for temporary use. They behave like organizations, and can have sites defined within them. The important difference is that projects cannot be parents of organizations or other projects, automatically become read only after 30 days, and are automatically deleted after 90 days. While organizations allow you to customize expiration times, the expiration times for projects are fixed.

Your runZero license includes the addition of up to five times as many project assets as the total number of licensed live assets. For example, if your license includes 1,000 live assets, you can have up to 5,000 project assets total. The total number of assets in organizations and projects can be seen on your license information page in the runZero console.

If you decide that you want to keep a project indefinitely, it can be converted into an organization.

Note that organizations support custom data retention settings that you can edit to your requirements.

Organization and project details

Click on the name of an organization or project to get to the details page. This page includes basic details about the organization or project, as well as its place in the organization hierarchy.

Sites

By default, your account includes a single organization, which itself contains a single site, named Primary. If the only site in an organization is deleted, a replacement will be created automatically. Similarly, if the last organization is removed, a replacement will be created. You can rename organizations and sites at any time.

Every organization has at least one site, but may have multiple sites. A site represents a distinct network segment, usually defined by addressing or accessibility. Sites in runZero do not necessarily correspond to physical sites or locations. Instead, they are used to represent distinct networks that may have overlapping address space. This allows for multiple sites to use the same RFC1918 space, something common in retail, while still being possible to differentiate their assets within the inventory.

Because sites represent separate networks, if you set up two sites and scan the same devices from both, you will end up with two copies of the resulting assets, one set for each site's network.

All analysis actions within runZero occur at the site level. For example, reports such as the switch topology report analyze a single site's devices, so you will likely want to avoid splitting routers and non-router assets into separate sites unless they are truly on separate disconnected networks.

Use cases for sites

For flat networks, where every IP address can reach any other address on the network, a single site is usually enough, and avoids the possibility of accidentally creating duplicate assets by scanning the same devices from multiple sites. Sites are recommended for complicated, sprawling, and highly-segmented environments.

Two circumstances that could lead to multiple sites:

- **Overlapping IP space**: sites will allow you to differentiate identical IPs that are actually different machines if you have overlapping IP space.
- **Highly complex network**: sites would not be required in this case, but they can be used as an organizational tool.

Creating sites

To create a new site, click the **New Site** button on the top of the sites page.

Site configurations

- **Name**: The name of your site.
- **Description**: The description can help identify the purpose of the site.
- **Default scan scope**: The default scan scope will be pre-populated when creating scans for this site.

- **Default scan exclusions**: The default scan exclusions will be pre-populated when creating scans for this site.
- **Registered subnets**: Registered subnets can be used to automatically tag assets, services, screenshots, and software that fall within each subnet.

Subnet tagging

Tagging based on subnet works a little differently to directly tagging assets or tagging them via a task.

If a subnet is defined to have a specific tag, then only assets in that subnet will be given that tag. The tag will be removed from any assets not in the subnet, even if set by a task or set manually.

Importing and exporting sites

Site configurations can be created or updated based on a CSV file import. Import your CSV from the sites page. The CSV format should include the following:

```
name,description,scope,exclusion,subnet_ranges,subnet_tags,subnet_descriptions
```

Your site configurations can also be exported as a CSV from the sites page.

Sites and Explorers

Sites can be tied to specific Explorers, which can help limit traffic between low-bandwidth segments. The site configuration allows a default scan scope to be defined, along with an optional list of excluded scan scopes. These fields can be used to set the scan scope for scans of the site.

If you would like to tie an Explorer to a site, navigate to the Explorers page, click the Explorer you would like to tie to the site, and then click configure. After that, you will see a dropdown with your site options.

Self-hosting runZero

Platform

Background

The self-hosted version of runZero allows you to run the entire platform on-premises or within your own cloud environment. This platform is functionally identical to the hosted service, provides a fully-offline mode, and does not send any inventory data back to runZero.

While self-hosting is less common, here are a few reasons your company might choose to:

- ISO compliance requirement
- Other compliance requirement
- Prefer data on-premise

Self-hosting requires a Platform license and must be explicitly enabled for your account. Please contact your runZero sales representative for further information.

Requirements

Before you get started, make sure your system meets the following requirements.

Hardware requirements

Recommended production system requirements:

- 12 CPU Cores at 2 GHz or faster
- 1TB of local disk storage
- 128 GB of RAM

Minimum production system requirements:

- 4 CPU Cores at 2 GHz or faster
- 100 GB of local disk storage
- 32 GB of RAM (more for large sites)

Minimum testing system requirements:

- 2 CPU Cores at 2 GHz or faster
- 20 GB of local disk storage
- 16 GB of RAM (more for large sites)

Sample customer deployments

Up to 1M assets:

- Two virtual machines (one for console and one for database)
- 32 CPU Cores at 2 GHz or faster
- 1TB of local disk storage
- 128 GB of RAM

Up to 500K assets:

- 32 CPU Cores at 2 GHz or faster
- 1TB of local disk storage
- 128 GB of RAM

Up to 50K assets:

- 16 CPU Cores at 2 GHz or faster
- 1TB of local disk storage
- 64 GB of RAM

Software requirements

- PostgreSQL 13 or newer (16+ preferred, our installer can install this for you)

Supported operating systems

- Ubuntu 18.04 and newer running on x86_64
- Red Hat Enterprise Linux 7.x and newer running on x86_64
- CentOS Linux 7.x and newer running on x86_64
- Oracle Linux 7.x and newer running on x86_64
    - 8.x must be 8.4+ with UEK 5.4+ or kernel 4.18+
    - 7.x must be 7.9+ with UEK 5.4+ or kernel 3.10+
- Debian Linux 9.x and newer running on x86_64

Windows Subsystem for Linux is not supported.

The runZero console can be installed in a minimal install of Ubuntu or any RHEL derivative; you do not need to install a GUI. The default disk partitioning scheme will work, but you may want to consider using LVM if you plan to resize the disk later.

Note about Debian

By default, Debian Linux uses the `su` command instead of `sudo`. It also requires that you use the command `su -` to become root, not just `su`, in order to update the `PATH` to include system administration commands such as `useradd`. If you receive an error during installation saying `executable file not found in $PATH`, this is the most likely reason.

Connectivity

The self-hosted runZero platform requires connectivity to the runZero SaaS backend over TCP port 443 (TLS) to obtain online updates. The specific IP addresses and hostnames depend on your deployment model and region and can be found below.

United States

The console hostname is `console.runzero.com`.

IPv4

- 13.248.161.247
- 76.223.34.198

IPv6

- 2600:9000:a415:cd87:fbe5:476a:3533:69f2
- 2600:9000:a716:ee91:85f9:3c9:48c9:59b9

Germany

The console hostname is `console-eu.runzero.com`.

IPv4

- 15.197.131.232
- 3.33.248.90

IPv6

- 2600:9000:a603:e925:542d:6d40:6897:bc3a
- 2600:9000:a70e:635f:71bd:bb0a:8e43:9466

Platform Address & TLS Configuration

The system running the runZero platform should have a static IP address.

The Explorers need to be able to validate their HTTPS connection to the platform via a TLS certificate. By default, the runZero platform installer will set up a self-signed certificate for the console's IP address, and downloaded Explorers will be preconfigured with the appropriate URL and certificate so they can make a verified connection.

> Note
> Some components of the application still reference the name "Rumble" for backwards compatibility. The documentation will be updated as these are changed.

If instead you plan to use a real certificate from an internal or public CA for your runZero console deployment, you will want to assign the console server a DNS name, and edit `RUNZERO_CONSOLE` in the configuration to contain the fully qualified domain name. The `TLS_CERT` and `TLS_KEY` should be set to point at the appropriate certificate and private key files. The console should then be restarted via `runzeroctl restart`. Newly downloaded Explorers will then be preconfigured with the right URL and certificate.

Because Explorers are preconfigured with the console address during download, if the IP address and/or DNS name are changed at a later date, the Explorers may need to be redeployed to receive the updated address.

Offline mode

The self-hosted version of runZero has the ability to run in offline mode. In this mode, the runZero update service is not used and offline updates must be applied manually. Enable this mode if you're in an isolated network or you don't want your self-hosted runZero console to make any connections to the internet. In addition to disabling online updates, offline mode also disables certain DNS probes that could reflect responses to the internet during a scan.

Installation steps

For offline installs please see offline installation.

For installs that use your own database credentials see Installation with your own PostgreSQL database.

Here's what the installation process will do:

- Set up PostgreSQL and create a passworded user.
- Generate TLS certificates for your IP address located in `/etc/runzero/certs`.
- Generate a configuration file at `/etc/runzero/config` and set some defaults.
- Create a `systemD` service for the runZero platform.
- Create all the necessary cron jobs required for the runzero platform.

Step 1: Download and run the installer

- Go to https://console.runzero.com/deploy/download/platform.
- Copy the command directly from the download page and run it in your terminal. This will download the installer and mark it as executable. The download path for the installer is uniquely keyed.
- Run the installer.
- By default, the console will be installed to `/opt/runzero`. If you need to change this, you can use the `--install-directory` option.

Step 2: Ensure sufficient disk space

By default, the runZero console will store data in the `storage` directory underneath the install directory. By default, that will be `/opt/runzero/storage`. The `storage` directory must have enough space to store all unexpired scan data and asset data. If necessary you can override its location by setting the `RUNZERO_STORAGE_PATH` variable in the runZero configuration file `/etc/runzero/config`.

You will also want to ensure that the temporary directory has plenty of space for temporary files such as software and content updates. If the `TMPDIR` environment variable is set, that directory will be used; otherwise the default is `/tmp`.

Step 3: Initialize the admin user

After you've installed the runZero platform, you will have access to the runZero CLI `runzeroctl`.

To initialize an admin user, run:

```
runzeroctl initial [email address]
```

Step 4: Sign in to your self-hosted console

If everything is set up correctly, you can sign in to your console at `https://YourInternalIPAddress`.

Note that you may need to enable HTTPS to pass through the Linux system firewall. Example commands:

Ubuntu Linux: `sudo ufw allow https/tcp`

RHEL/CentOS/Oracle: `sudo firewall-cmd --add-service=https`

To make a `firewall-cmd` change permanent across reboots, run the command a second time with the `--permanent` flag added.

Installation with your own PostgreSQL database

runZero defaults to installing and configuring a PostgreSQL user and database for you. If you would like to provide your own details this option will allow you to override that behavior.

Requirements

- PostgreSQL 13 or newer (16+ preferred)
- Password authentication must be enabled.
- Two extensions are required: pg_trgm and uuid-ossp. These may be supplied as part of a `contrib` package rather than as part of the main PostgreSQL server install, depending on where you obtain your PostgreSQL packages.

PostgreSQL example to enable extensions and add a database/user

```
CREATE DATABASE runzero TEMPLATE='template0' LC_COLLATE = 'en_US.UTF-8' LC_CTYPE = 'en_US.UTF-8';
CREATE USER runzero WITH PASSWORD 'YOURPASSWORD';
GRANT ALL PRIVILEGES ON DATABASE runzero TO runzero;
\connect runzero;
CREATE EXTENSION IF NOT EXISTS pg_trgm;
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

Steps to install the self-hosted runZero platform with your own database credentials

1. Run this install command as root:

```
./runzero-platform-[VERSION]-linux-amd64.bin install --manual-database
```

2. Edit your runZero configuration and add your database details in `/etc/runzero/config`. The line you need to edit is:

```
DATABASE_URL=postgres://runzero:{DB_PASSWORD}@127.0.0.1:5432/runzero?sslmode=disable
```

Change this to match your credentials. You need to set the user, password, host, port, and database name. Here is the format:

```
DATABASE_URL=postgres://{DB_USER}:{DB_PASSWORD}@{DB_HOST}:{DB_PORT}/{DB_NAME}?{DB_OPTIONS}
```

4. Verify the self-hosted runZero platform can connect to your database with this command. `sudo runzeroctl database verify`
5. Once your database is configured and verified you can restart the self-hosted runZero platform service `sudo systemctl restart runzero-console`

runZero updates

For offline updates please see CLI update with offline mode.

The self-hosted runZero platform must be updated prior to first use. The `runzeroctl` command can be used to download the update and then restart the service after the update is complete.

You can update the platform and scanners at the same time or separately with the CLI update management commands.

Managing users

You can manage users inside your self-hosted runZero platform console at `https://YourInternalIPAddress/team` or via the runZero CLI

Some things you can manage:

- Adding, deleting, and listing users
- Resetting passwords
- Changing default roles
- Viewing details

- Changing organization roles

CLI service management

Start the runZero service

Starts the runZero platform service.

```
runzeroctl start
```

Stop the runZero service

Stops the runZero platform service.

```
runzeroctl stop
```

Restart the runZero service

Restarts the runZero platform service.

```
runzeroctl restart
```

Install the runZero platform

Install the runZero platform service and all required dependencies such as PostgreSQL. Creates a systemd service, generates cron jobs, and generates a configuration file in `/etc/runzero/config`.

```
runzeroctl install
```

Uninstall the runZero platform

Stop and remove the runZero platform service from systemd and removes the generated cron jobs. This does not remove your PostgreSQL database, and it retains your data.

```
runzeroctl uninstall
```

Purge the runZero platform

Stop and remove the runZero platform service from systemd and removes the generated cron jobs. This will delete your runZero database and remove all the runZero directories `/etc/runzero` and `/opt/runzero`.

```
runzeroctl purge
```

You can uninstall and purge everything except the database and your PostgreSQL settings with this flag:

```
runzeroctl uninstall --purge --ignore-database
```

Run the runZero platform manually

Starts the runZero platform manually. Logs will be written to standard output.

```
runzeroctl server
```

Verify your database is reachable

Attempts to connect to your database using your self-hosted runZero platform configuration. It will either succeed or display an error.

```
runzeroctl database verify
```

CLI update management

Update the runZero platform and scanners

Updates the runZero platform service and runZero scanners. You can use the optional parameter `force` to force the update even if the current installation is the latest version.

```
runzeroctl update [--force]
```

Update the runZero platform

Updates just the runZero platform service. You can use the optional parameter `force` to force the update even if the current install is the latest version.

```
runzeroctl update-platform [--force]
```

Update the runZero scanners

Updates just the runZero scanners. You can use the optional parameter `force` to force the update even if the current installation is the latest version.

```
runzeroctl update-scanner [--force]
```

CLI user management

Create the initial administrator account

Creates the initial admin user for a new installation. You must provide an email address.

```
runzeroctl initial <email>
```

List user accounts

Lists all the users along with their email address, full name, and current roles.

```
runzeroctl user list
```

Add a user account

Creates a new user account under the initial administrator user. You must provide an email address.

```
runzeroctl user add <email>
```

Delete a user account

Deletes a user account. You must provide an email address. This cannot be undone.

```
runzeroctl user delete <email>
```

Get user details

Gets the details for a user account. You must provide an email address. Provides information such as full name, date created, last sign in IP, last sign in time, last activity, default organization role, and their current roles. You must provide an email address.

```
runzeroctl user details <email>
```

Set a user role

Sets a user's role to the role provided. Email and role must be provided. The organization is optional. If the organization isn't provided this sets their default role.

```
runzeroctl user set-role <email> [organization name or organization ID]:<role>
```

Reset a user password and MFA

This will generate and apply a new password for the specified user. The password will be printed to the terminal. You must provide an email address. This reset process also clears any associated MFA tokens.

```
runzeroctl user reset <email>
```

CLI organization management

List all organizations

Lists all the organizations by their name and ID.

```
runzeroctl organization list
```

Advanced configuration

The file at /etc/runzero/config can be modified to support a wide variety of configurations.

After making changes, apply them by running `runzeroctl restart`.

Email server

runZero uses an SMTP server for user account invitations and notifications. The default configuration assumes that a SMTP server is available on localhost that does not require authentication:

```
SMTP_SERVER=127.0.0.1:25
SMTP_AUTH_METHOD=none
```

runZero will automatically use STARTTLS with plaintext SMTP servers and validate the certificate. In internal environments where the SMTP server is not using a valid TLS certificate, verification can be disabled by setting:

```
SMTP_TLS_NOVERIFY=true
```

Transport-layer TLS (instead of STARTTLS) can be configured with:

```
SMTP_TLS=true
```

If authentication is required, the following three settings should be configured:

```
SMTP_AUTH_METHOD=plain
SMTP_AUTH_USER=YourUsername
SMTP_AUTH_PASS=YourPassword
```

Emails are sent from `noreply@rumble.run` by default, but this can be changed by setting the `FROM_EMAIL` option:

```
FROM_EMAIL=runzero@yourcompany.int
```

Hostname and port

The RUNZERO_CONSOLE variable is used for creating inbound links, configuring deployed Explorers, and generating the default self-signed TLS certificate. This setting is how both users and deployed Explorers connect to the platform.

```
RUNZERO_CONSOLE=https://{IP ADDRESS OR HOSTNAME}:443
```

Changing this setting may require regeneration of the TLS certificate and redeployment of Explorers.

```
runzeroctl generate-certificate
```

runZero can be configured to run on a different port with the CONSOLE_PORT setting. This port defines where the console listens, but users and Explorers still connect to the RUNZERO_CONSOLE value. In most cases this should match the port specified by the RUNZERO_CONSOLE.

```
CONSOLE_PORT=443
```

TLS configuration

runZero will generate a self-signed TLS certificate and serve all web requests using HTTP over TLS. The standard configuration uses a self-signed certificate stored in the filesystem:

```
TLS=true
TLS_CERT=/etc/runzero/certs/cert.pem
TLS_KEY=/etc/runzero/certs/key.pem
```

The certificate and key file are PEM encoded and can be replaced with any valid certificate. Please ensure that any new certificate lists the value of RUNZERO_CONSOLE in the list of Subject Alternative Names.

Note that the key file must be unencrypted (not password protected). If you have a passphrase set, you can remove it using OpenSSL. For example:

```
# RSA key
openssl rsa -in key.pem -out newkey.pem
# DSA key
openssl dsa -in key.pem -out newkey.pem
```

If a TLS-terminating reverse proxy is used (AWS ELB, nginx, etc), TLS can be disabled at the application level:

```
TLS=false
```

Please note that while the web interface can be accessed over plain HTTP in this scenario, Explorers will refuse to connect to a plain HTTP port, and features like WebAuthn MFA will not work unless the site is accessed through TLS.

Specific TLS versions and ciphers can be configured.

TLS versions are chosen by minimum and maximum:

```
TLS_VERSION_MIN=1.2
TLS_VERSION_MAX=1.3
```

TLS ciphers may be chosen by profile name:

- **default**: A set of strong ciphers, great for most configurations
- **nist80052**: A set of strong ciphers, approved in NIST 800-52r2.
- **nist80052-aes256**: A set of strong ciphers, approved in NIST 800-52r2, restricted to AES-256 variants

Please note that TLS 1.3 ciphers work differently and if a specific set of ciphers is required, both TLS_VERSION_MIN and TLS_VERSION_MAX should be set to `1.2`. For example, to restrict runzero to **only** NIST 800-52r2 approved ciphers using AES-256, the following configuration should be used:

```
TLS_VERSION_MIN=1.2
TLS_VERSION_MAX=1.2
TLS_CIPHERS=nist80052-aes256
```

TLS ciphers may also be chosen using comma-separated list of cipher constants.

Database configuration

runZero uses a PostgreSQL database for all platform data, except for raw scan files, change reports, and images processed from scans. By default, runZero will configure a local PostgreSQL server on the same system, with a random password, and without TLS encryption:

```
DATABASE_URL=postgres://runzero:{DB_PASSWORD}@127.0.0.1:5432/runzero?sslmode=disable
```

If separate database is preferred, any PostgreSQL server running 12.x or newer should work. TLS (sslmode=require) should be enabled when a non-local database server is configured.

The default database pool (connection count) can be modified for high throughput environments:

```
DATABASE_POOL_COUNT=50
```

Proxy configuration

runZero makes outbound connections to receive platform updates (in online mode), to connect to third-party APIs, and to delivery webhooks for notifications. If a proxy server is required, it can be configured with this setting:

```
HTTPS_PROXY=host:port
```

Storage configuration

runZero uses local file storage to store raw scan data, change reports, and images retrieved from assets. This storage directory must be owned by the `rumble` user and be mounted below the /opt/runzero path.

```
RUNZERO_STORAGE_MODE=local
RUNZERO_STORAGE_PATH=/opt/runzero/storage
```

Files within the storage directory are split up into two groups, assets and scans. The names of these can be changed by setting:

```
ASSET_BUCKET=assets
SCAN_BUCKET=scans
```

To use AWS S3 for file storage instead, the following configuration can be set:

```
RUNZERO_STORAGE_MODE=s3
ASSET_BUCKET=company-runzero-assets
SCAN_BUCKET=company-runzero-scans
```

If a non-AWS backend is used that is compatible with the S3 API, use the same AWS and bucket variables above but override `AWS_REGION` and set the `AWS_ENDPOINT_URL_S3` or `RUNZERO_STORAGE_ENDPOINT` parameter to the endpoint as appropriate. Reach out to runZero support if you run into any issues with the endpoint configuration.

If the S3 buckets are in a different region than the environment, set the `RUNZERO_STORAGE_REGION` to the correct region for the buckets.

If S3 is used, AWS must also be configured, with at least the `AWS_REGION` variable set, even if a non-AWS backend is enabled.

Secret configuration

runZero uses three randomly generated secret tokens to secure the platform. These keys are hexadecimal strings generated by 16 bytes of random. Compatible values can be generated by OpenSSL:

```
$ openssl rand -hex 16
```

The authentication key used for local storage HMAC operations. This key can be rotated as long as the service is restarted afterwards:

```
RUNZERO_STORAGE_KEY_SECRET={SECRET_32_HEX}
```

The session secret key is used to sign and validate browser session cookies. This key can be rotated, but doing so will invalidate all existing web sign-ins:

```
SESSION_SECRET={SECRET_32_HEX}
```

The DB key is used for encryption of sensitive fields (user password hashes). This key cannot be rotated, as password authentication will no longer work. If this key is changed, users must reset their password from the command-line or web interface using email before they can sign back in:

```
DB_KEY={SECRET_32_HEX}
```

AWS configuration

The AWS region is required:

```
AWS_REGION=us-east-1
```

The Access Key ID and Secret must be valid and correlated to a user with read-write access to the S3 buckets and read-only access to Secrets Manager.

```
AWS_ACCESS_KEY_ID=AKIA....
AWS_SECRET_ACCESS_KEY=SECRET....
```

AWS Secrets Manager can be used to retrieve almost any configuration setting at startup. The Secrets Manager entries should match the key names of the configuration file. The secret name can be defined with:

```
AWS_SECRETS_MANAGER_KEY=rumble/production
```

In addition the variables above, most AWS CLI environment variables are also available for specific tuning.

The location of the Explorer and scanner binaries can be changed with these settings. Note that these should still live under /opt/runzero or the service will not be able to load them:

```
RUNZERO_RELEASE_DIR=/opt/runzero/agent/
RUNZERO_SCANNER_RELEASE_DIR=/opt/runzero/scanner/
```

Content security policy

In the case of a non-standard S3 configuration (or S3-like deployment, such as minio), the Content Security Policy headers need to be configured to allow external image loads.

The CSP_IMAGES setting can be used to specify one or more (comma-delimited) external image sources:

```
CSP_IMAGES=https://*.custom-storage-backend.lan
```

In additional to CSP_IMAGES, the following additional CSP settings are available:

```
CSP_SCRIPTS=https://*.myscripts.lan
CSP_FONTS=https://*.myfonts.lan
CSP_STYLES=https://*.mystyles.lan
```

Logging

The self-hosted runZero console sends its logging output to standard output. On Linux this is picked up by Linux systemd, and stored in the journal where it can be queried with the `journalctl` tool. For example, to view the most recent hour of logs, with most recent messages first, you can run the following command:

```
journalctl --unit=runzero-console --since=-1hour --reverse
```

The journalctl output can be piped to a text file to send to runZero support.

The systemd logging subsystem can be configured to send log messages to a local syslog daemon as well as the journal. Once logs are in syslog, they can be forwarded across the network to remote logging servers using the standard syslog protocol. Some Linux distributions, such as RHEL, are set up to forward logs from systemd to syslog by default. Other distributions, such as Ubuntu and Debian, don't include a syslog daemon in their default minimal server installs.

To configure systemd to send logs to syslog, you can use the ForwardToSyslog option in `/etc/systemd.conf`. Alternatively, some syslog daemons have an option to read the systemd journal; for example, rsyslog has imjournal.

runZero's logs are output in CEE-enhanced JSON format. This is compatible with rsyslog and syslog-ng, DataDog, and other common logging tools. For rsyslog, the mmjsonparse module can be used to filter the logs based on individual JSON fields, and forward them to ElasticSearch or other JSON databases.

The environment variable `LOG_MAX_LENGTH` can be set in runZero's config file to apply a limit to the length of each log line, in bytes of UTF-8 text. A value of 0 means no limit, other values below 480 are treated as 480. runZero will attempt to preserve the most valuable logging fields when truncating log output, and ensure that the result is still valid JSON. Note that the length limit is applied before any additional information systemd or syslog adds to the start of the line.

The environment variable `LOG_FORMAT` can be set to `text` to disable the CEE and JSON format, and log in plain text. For example:

```
LOG_FORMAT=text
LOG_MAX_LENGTH=512
```

HTTP timeouts

The self-hosted runZero Console allows you to change the built-in web server's HTTP timeouts. These can be changed through three configuration variables.

You are able to set the HTTP idle, read, and write timeout. The default settings are below and don't usually need to be changed.

```
HTTP_IDLE_TIMEOUT_MINUTES=3
HTTP_READ_TIMEOUT_MINUTES=60
```

```
HTTP_WRITE_TIMEOUT_MINUTES=720
```

Concurrent processing

The runZero Console can process more than one completed task at a time if the `RUNZERO_CRUNCHER_INSTANCES` option is set to a value greater than 1. Tasks are only processed concurrently when they exist within different organizations, or the tasks are within the same organization, but different sites, and the task itself does not require cross-site asset merging. Most third-party connectors and integrations require cross-site merging and are not able to take advantage of concurrent site processing within the same organization. Please note that the resource requirements for concurrent task processing scale linearly with the specified instance count.

The example below configures the console to process up to four concurrent tasks across all organizations:

```
RUNZERO_CRUNCHER_INSTANCES=4
```

Custom javaScript

The self-hosted runZero Console allows you to include custom arbitrary JavaScript to be executed on the various runZero Console web pages. To use this feature, add your JavaScript to /opt/runzero/etc/custom.js. If this is your first time configuring this feature, the custom.js file will not exist. You will need to create the file custom.js inside the /opt/runzero/etc folder.

To enable the feature set the enviroment variable below to your configuration.

```
RUNZERO_CUSTOM_JS=true
```

HTTP headers

The security headers sent by the runZero Console can be disabled as needed using the following options:

```
# Disable Strict-Transport-Security
RUNZERO_DISABLE_HSTS=true

# Disable X-Frame-Options
RUNZERO_DISABLE_HXFO=true

# Disable X-Content-Type-Options
RUNZERO_DISABLE_HXCTO=true

# Disable X-XSS-Protection
RUNZERO_DISABLE_HXXP=true

# Disable Referrer-Policy
RUNZERO_DISABLE_HRP=true
```

```
# Disable Content-Security-Policy
RUNZERO_DISABLE_HCSP=true
```

Unofficial CPEs

When runZero can successfully fingerprint an asset's operating system, a CPE will be generated. In cases where the NIST database does not contain an official match, runZero will generate an unofficial CPE by default. This behavior can be disabled by setting `RUNZERO_GENERATE_UNOFFICIAL_CPE` to `false`:

```
RUNZERO_GENERATE_UNOFFICIAL_CPE=false
```

When unofficial CPEs are generated by runZero, they include `r0_unofficial` in the `other` field of the CPE by default. This value can be changed to any alphanumeric-constrained tag (limited to 32 characters):

```
RUNZERO_UNOFFICIAL_CPE_TAG=custom_unoffical_tag
```

Permissions

The self-hosted platform requires **root** access to install and manage from the command-line.

The platform service (`runzero-console`) runs as **root** and spawns a worker subprocess that runs as the **runzero** user account inside of a chroot environment (/opt/runzero). All substantive work happens within this isolated subprocess. Please note that older installations will use `rumble` instead of `runzero` in directory, file, and user names.

The following filesystem locations are used by the self-hosted platform:

/etc/runzero

Path	Owner	Permission	Notes
`/etc/runzero`	root	0700	Configuration files and certificates
`/etc/runzero/config`	root	0600	A plain-text configuration file
`/etc/runzero/certs`	root	0700	A directory containing the TLS certificate and key
`/etc/runzero/certs/cert.pm`	root	0600	The TLS certificate in PEM format
`/etc/runzero/certs/key.pm`	root	0600	The TLS certificate private key in PEM format

/opt/runzero

Path	Owner	Permission	Notes
`/opt/runzero/tmp`	runzero	0755	A temporary directory
`/opt/runzero/storage`	runzero	0700	Contains asset and scan artifacts
`/opt/runzero/console`	root	0755	Contains the platform executable
`/opt/runzero/console/runzero-console.bin`	root	0755	The platform executable
`/opt/runzero/agent`	root	0755	Contains the Explorer binaries
`/opt/runzero/agent/runzero-agent-*`	root	0755	The Explorer binaries
`/opt/runzero/scanner`	root	0755	Contains the CLI binaries
`/opt/runzero/agent/runzero-scanner-*`	root	0755	The CLI binaries
`/opt/runzero/proc`	root	0755	Contains copies of system /proc files
`/opt/runzero/proc/cpuinfo`	root	0644	A copy of /proc/cpuinfo
`/opt/runzero/proc/meminfo`	root	0644	A copy of /proc/meminfo
`/opt/runzero/proc/version`	root	0644	A copy of /proc/version
`/opt/runzero/etc`	root	0755	Contains copies of system files
`/opt/runzero/etc/resolv.conf`	root	0644	A copy of /etc/resolv.conf
`/opt/runzero/etc/hosts`	root	0644	A copy of /etc/hosts
`/opt/runzero/etc/ca-certificates.crt`	root	0644	A copy of the system root CA store
`/opt/runzero/etc/runzero`	runzero	0700	Contains instance identifiers
`/opt/runzero/etc/runzero/cruncher.id`	runzero	0700	A unique ID to identify the cruncher instance
`/opt/runzero/etc/runzero/hub.id`	runzero	0700	A unique ID to identify the hub instance
`/opt/runzero/config`	root	0700	Unused today

Backup and restoration

Your runZero installation and data can be backed up and restored to preserve your configuration.

runZero data backup

A backup of a self-hosted installation can be obtained by archiving the file system and database.

The file system archive includes the following paths:

- `/etc/runzero`
- `/opt/runzero`
- `/lib/systemd/system/runzero-console.service`
- `/etc/systemd/system/multi-user.target.wants/runzero-console.service`
- `/usr/bin/runzeroctl`

A sample file system backup command is:

```
# tar zcvf runzero-backup-fs.tar.gz /etc/runzero/ /opt/runzero/ \
/lib/systemd/system/runzero-console.service \
/etc/systemd/system/multi-user.target.wants/runzero-console.service \
/usr/bin/runzeroctl
```

The PostgreSQL database must be backed up separately. A sample command is shown below:

```
# sudo su - postgres
$ pg_dumpall -f runzero.sql && gzip runzero.sql
```

runZero data restoration

To restore the runZero install, follow these steps.

1. Stop any running runZero service:

```
# runzeroctl stop
```

2. Unpack the filesystem archive:

```
# tar -C / -zxvf /path/to/runzero-backup-fs.tar.gz
```

3. Restore the PostgreSQL database:

```
# sudo su - postgres
$ dropdb runzero; gzip -dc runzero.sql.gz | psql
```

4. Restart the runZero service:

```
sudo systemctl restart runzero-console
```

Support and debugging

The `runzeroctl` command includes a debugging tool which can collect diagnostics from your server and assemble them into a zip file which you can send to support. If you are asked to do this, the command is:

```
runzeroctl diagnostics run-script
```

Data is written to `/opt/runzero/collector`.

Alternatively you can save a copy of the script to the current directory so that you can examine it before running it:

```
runzeroctl diagnostics write-script
```

Manual migrations

Starting with version `4.0.240221.0`, the runZero self-hosted upgrade process will run migrations before restarting the service. If you are running an older version of the software and would like to prevent downtime during the upgrade of a single-node self-hosted installation, the following steps can be used:

1. Obtain the self-hosted download link from the runZero SaaS.
2. Download this file manually to your self-hosted systems:

```
$ curl -o platform.bin https://console.runzero.com/...../runzero-platform.bin
```

3. Mark this file as executable and run it with the `task db:migrate` parameter:

```
$ chmod u+x platform.bin; ./platform.bin task db:migrate
```

4. Once the migrations are done, install the update as usual:

```
$ runzeroctl update
```

Offline mode configuration

Offline installation

The self-hosted runZero platform comes with a few options for your installation. You can utilize these options by adding flags to the install command. The current flags available are `--offline`, `--distro-packages-only`, and `--postgres-rpm-directory`.

`--offline`
: Configures the installation for operation without internet access. Network access may still be necessary during the installation to acquire dependencies. The installer uses the

upstream postgresql.org packages by default and this can be disabled by specifying the options below.

`--distro-packages-only`
: Install the platform without using third-party repositories (not available on RHEL/CentOS 7). The install will try to acquire the PostgreSQL package from the configured operating system repositories (local or remote). If no repository is reachable, these packages can also be specified as a directory on the file system using the option below.

`--postgres-rpm-directory [directory]`
: Install using supplied PostgreSQL RPM files (requires RHEL or CentOS). The installer will use the RPMs in the specified directory to satisfy the PostgreSQL dependencies.

By default, the console will be installed to `/opt/runzero`. If you need to change this, you can use the `--install-directory [directory]` option.

PostgreSQL RPMs required for `--postgres-rpm-directory`

- RHEL/CentOS 9 RPMs.
- RHEL/CentOS 8 RPMs.
- RHEL/CentOS 7 RPMs.

There are four RPMs we require for installing PostgreSQL 16:

- postgresql16
- postgresql16-server
- postgresql16-contrib
- postgresql16-libs

Example install commands for offline mode

The first step is to download the runZero platform

RHEL/CentOS 7, 8, or 9

1. Download the required RPMs above and store them in a directory.
2. Run this install command as root:

```
./runzero-platform-[VERSION]-linux-amd64.bin install --offline
    --postgres-rpm-directory [RPM_DIRECTORY]
```

Ubuntu 18.04+, Debian 10+, or RHEL/CentOS 8 or 9

1. Run this install command as root.

```
./runzero-platform-[VERSION]-linux-amd64.bin install --offline --distro-packages-only
```

Remember to use `su -` on Debian, not just `su`.

Note

Some components of the application still reference the name "Rumble" for backwards compatibility. The documentation will be updated as these are changed.

Enabling offline mode for existing installs

- Open `/etc/runzero/config` with an editor of your choice.
- Look for `OFFLINE=` and change it to `OFFLINE=true`.
- Restart the runZero service `runzeroctl restart`.

CLI update with offline mode

The self-hosted runZero platform must be updated prior to first use.

Update the runZero platform and scanners with an offline update

- Go to https://console.runzero.com/deploy/download/platform.
- Copy the command directly from the download page and run it in your terminal, or you can use one of the following commands to update using the zip archive you downloaded.

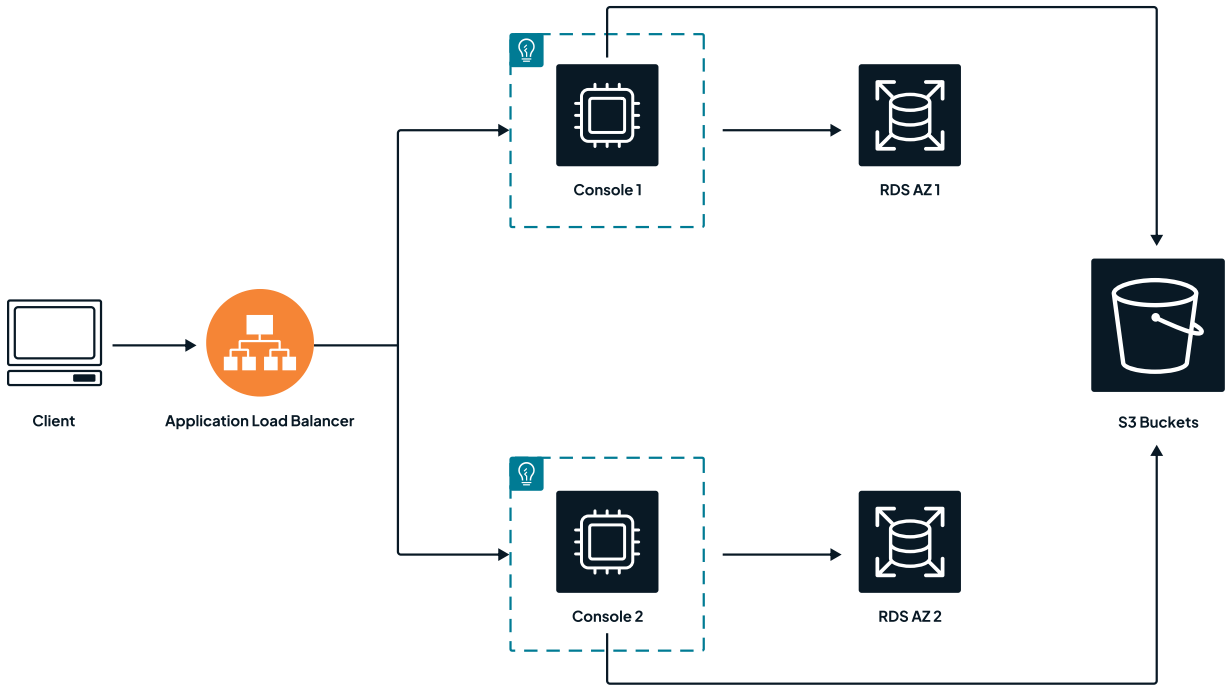
```
runzeroctl update runzero-platform-update-[VERSION].zip

runzeroctl update --offline --zip-file-path runzero-platform-update-[VERSION].zip
```

You will need to change the version to match the zip archive you downloaded.

High-availability configuration

Self-hosted installations of runZero can be configured for high-availability. For this configuration, a load balancer is used to direct traffic to multiple console servers, which use a shared PostgreSQL cluster and storage backend. The following diagram illustrates an example architecture of a high-availability installation using AWS with two availability zones.



In this diagram, an application load balancer (ALB) is terminating TLS and pointing to a target group that consists of two runZero servers, each in separate availability zones. The runZero servers use a multi-availability-zone PostgreSQL RDS instance, also configured for the same two availability zones, and both servers point to the same set of S3 storage buckets.

Installation steps

These are the general steps for installing and configuring a high-availability instance of runZero:

1. Deploy an application load balancer.
2. Install and configure the first server node.
3. Install and configure subsequent server nodes.

Deploy a load balancer with TLS termination

In order to provide a high-availability interface to the runZero cluster, an application load balancer should be used that receives TLS connections and forwards the HTTP requests to the available servers. This load balancer should be highly-available on its own. Any load balancer that can proxy large HTTP requests and websockets should be supported. The load balancer should add an X-Forwarded-For header to the HTTP requests sent onward to the runZero servers. The load balancer should also support detection of unhealthy nodes in order to handle automatic failover. The /health endpoint on the runZero servers can be used to implement health checks on the load balancer.

Prepare the first runZero server node

The general steps for installing and configuring the first runZero node are:

1. Install the runZero platform.
2. Configure the console URL and TLS/XFF settings.
3. Configure the database settings.
4. Configure the shared storage settings.
5. Verify your configuration.
6. Create the initial user account.
7. Add the runZero server to the load balancer and verify the connection.

Install the runZero platform

> Note
> Some components of the application still reference the name "Rumble" for backwards compatibility. The documentation will be updated as these are changed.

1. Go to platform download page in the runZero Console.
2. Copy the URL to the platform installer from the download page. The download path for the installer is uniquely keyed to your license.
3. Download the install using Wget or cURL:

- `wget -O runzero-platform.bin https://console.runzero.com/download/platform-combined/<UNIQUE KEY>/62e41615/runzero-platform-vX.X.X-linux-amd64.bin`
- `curl -o runzero-platform.bin https://console.runzero.com/download/platform-combined/<UNIQUE KEY>/62e41615/runzero-platform-vX.X.X-linux-amd64.bin`

4. Run the following command to make the installer file executable:

```
# chmod u+x runzero-platform.bin
```

5. Run the install command with the manual database option:

```
# ./runzero-platform.bin install --manual-database
```

Configure the console URL and TLS/XFF

To configure your self-hosted installation for high-availability, the settings below should be configured in `/etc/runzero/config`, starting with the console URL and the TLS/XFF settings.

- Set RUNZERO_CONSOLE variable to the hostname or IP address of the load balancer:

```
RUNZERO_CONSOLE=https://{IP ADDRESS OR HOSTNAME}:443
```

- Disable TLS and configure the server to trust X-Forwarded-For headers with the following two lines:

```
TLS=false
RUNZERO_TRUST_XFF=true
```

Configure the database

The next step is to configure the database to be used by runZero.

- To do this, edit the DATABASE_URL value to match this format:

```
DATABASE_URL=postgres://{DB_USER}:{DB_PASSWORD}@{DB_HOST}:{DB_PORT}/{DB_NAME}?{DB_OPTIONS}
```

An example DATABASE_URL for AWS RDS will look like:

```
DATABASE_URL=postgres://rumble:[password]@rumbledb.rds-id.us-east-1.rds.amazonaws.com:5432/rumble?sslmode=require
```

To use a replica for read-only queries, specify DATABASE_REPLICA_URL with the same syntax. For RDS Aurora, specify the read-only endpoint as the replica URL. If no replicas are in use, only the DATABASE_URL is required, and this should point to the writer endpoint for RDS Aurora instances.

If a different type of PostgreSQL clustering is in use, make sure that the DATABASE_URL points to the highly-available endpoint for the active writer.

Configure shared storage

Next, configure the shared storage backend. This is used for storing asset data, raw scan data, and reports. runZero supports both mounted NFS and object storage.

- To use NFS, mount a read-write NFS share to /opt/runzero/storage and specify local storage mode with the following configuration options. The bucket names will be created under the NFS mount base directory.

```
RUNZERO_STORAGE_MODE=local
RUNZERO_STORAGE_PATH=/opt/runzero/storage
ASSET_BUCKET=runzero-assets
SCAN_BUCKET=runzero-scans
```

- To use AWS S3 as the shared storage, either configure an IAM instance role with read-write access to two S3 buckets, or specify the AWS credentials in the configuration file.
    - To specify AWS credentials specifically for runZero, use the following syntax:

```
AWS_ACCESS_KEY_ID=AKIA…
AWS_SECRET_ACCESS_KEY=[SECRET]
```

- Specify that the S3 storage mechanism should be used and set the name of the two buckets:

```
AWS_REGION=<your-region>
RUNZERO_STORAGE_MODE=s3
ASSET_BUCKET=<company>-runzero-assets
SCAN_BUCKET=<company>-runzero-scans
```

If a non-AWS backend is used that is compatible with the S3 API, use the same AWS and bucket variables above but override AWS_REGION and set the AWS_ENDPOINT_URL_S3 parameter to the endpoint as appropriate. Reach out to runZero support if you run into any issues with the endpoint configuration.

In addition the variables above, most AWS CLI environment variables are also available for specific tuning.

Verify the configuration

1. After saving your configuration file, use this command to verify the self-hosted runZero platform can connect to your database:

```
sudo runzeroctl database verify
```

2. Once your settings are configured and verified you can restart the self-hosted runZero platform service:

```
sudo systemctl restart runzero-console
```

Create the initial user account

- Use the `runzeroctl initial <your-email>` command to create the initial user account.

Add the runZero server to the load balancer and verify the connection

- Point the load balancer to the new runZero server's IP on port 80 (unless modified).
- Browse to the load balancer URL over HTTPS using the configured DNS name.
- Verify that the user account configured previously can authenticate correctly.

Installing and configuring subsequent nodes

- Download and install the runZero platform binary on each node as above with the `--manual-database` option specified.
- Copy the configuration file from the first node to `/etc/runzero/config` on each subsequent node.
- After saving your configuration file, use this command to verify that the self-hosted runZero platform can connect to your database:

```
sudo runzeroctl database verify
```

- Restart the platform service by running:

```
sudo systemctl restart runzero-console
```

runZero nodes should be spread across multiple availability zones and added to the load balancer's target group. Verify that requests are reaching each individual node by reviewing the syslog entries on that node.

AWS advanced configuration

AWS Secrets Manager can be used to retrieve configuration settings at startup. The Secrets Manager entries should match the key names of the configuration file. The secret name can be defined with:

```
AWS_SECRETS_MANAGER_KEY=rumble/production
```

To ensure that runZero can access the Secrets Manager, ensure that an AWS_REGION is set in the configuration file, and that the instance has an IAM instance role with permission to read the Secrets Manager key or the AWS credentials have been specified directly in the configuration file. With a Secrets Manager configuration, only the AWS options need to be present in the configuration file and the rest of the settings will be loaded from the Secrets Manager key. This will require copying and defining various settings from the generated configuration file to the Secrets Manager key-value pairs.

Note that many settings need to be identical across the runZero cluster for web requests and authentication to work correctly (including session secrets, DB encryption key, and the console URL).

Self-hosted troubleshooting

The runZero console includes a diagnostics collection script inspired by the need to troubleshoot a self-hosted environment. Collecting the necessary performance statistics, log files, system configuration, and profile debug capture was difficult for customers since there are many different commands and files involved. After checking permissions and dependencies, the diagnostic script gathers and compresses troubleshooting data into a convenient archive file that can be provided to runZero support. The script provides a consistent method for data collection whether it be from a system running your self-hosted console or an Explorer.

Requirements

The diagnostic script must be run from a Linux system. Additionally, the following dependencies must be installed for the script to run successfully:

Dependency	To install (Ubuntu)	To install (RHEL, CentOS, Fedora, Oracle Linux)
`zip`	`apt install zip`	`dnf install zip`
`unzip`	`apt install unzip`	`dnf install unzip`
`lsscsi`	`apt install lsscsi`	`dnf install lsscsi`
`iostat`	`apt install sysstat`	`dnf install sysstat`
`curl`	`apt install curl`	`dnf install curl`

The script must be run as `root` or with `sudo` in order to gather information from the system files and the `/opt/runzero` and `/etc/runzero` directories.

Running the script

Using runzeroctl

To run the script via the `runzeroctl` command, use:

```
runzeroctl diagnostics run-script
```

Output will be placed in `/opt/runzero/collector`.

Manually

To run the script manually, use the command below to write a copy of the script to the current directory:

```
runzeroctl diagnostics write-script
```

The script will be written as `runZero-collector.sh`. To run it:

```
sudo ./runZero-collector.sh
```

The script also has a `--help` option.

This script will perform the following tasks:

- Check for the required permissions.
- Check for the dependency commands (lsscsi / zip / unzip/ iostat / curl).
- Check that the operating system is Linux.
- Check that either a runZero Explorer or runZero console is installed.
- Collect system configuration information.
- Collect system performance statistics.
- Collect database configuration information.
- Collect some basic statistics from the database.
- Collect runZero debug information from the console or log files from the Explorer.
- Create a zip file of the data collected and the log from the script.

Logging

The script will log all actions in `/tmp` and will provide the file name. Upon completion the log file is moved to the `/opt/runzero/collector` directory. This directory does not exist with the installation of a runZero Explorer or runZero console. The script will create the directory if it doesn’t exist and place all files into that directory.

Data retention

runZero allows the data retention periods to be configured at the organization level. The organization settings page provides three ways to control how runZero manages your asset and scan data. Data expiration is processed as a nightly batch job based on the current settings for each organization in your account.

By default, data is retained for up to 1 year in the runZero Platform. This can be increased to up to 3 years for active subscriptions. The runZero Community Edition is limited to 30 days of retention.

Stale asset expiration

Stale asset expiration allows you to make sure that out-of-date information gets purged from runZero. Assets which have not been seen in the specified number of days are automatically deleted. The automatic deletion task runs once per day.

The stale asset expiration limit applies to both online and offline assets. For example, if an asset was online when last scanned, but hasn't been scanned (or imported via a connector) in the specified number of days, then the information is stale and the asset will be deleted. It's like running an asset search query of `last_seen:>Xdays` (where X is replaced by a number), and deleting the resulting assets.

You can disable stale asset expiration by setting the number of days to zero.

Offline asset expiration

Offline asset expiration allows you to clear out information about assets that no longer exist on the network. Assets which are offline and have not been seen in the specified number of days are automatically deleted daily. Online assets are not touched. The automatic deletion task runs once per day.

This feature is particularly useful for guest wireless networks, or networks with short DHCP leases. It's like running an asset search query of `last_seen:>Xdays AND alive:false` (where X is replaced by a number), and deleting the resulting assets.

You can disable offline asset expiration by setting the number of days to zero.

Stale integration attribute expiration

Stale integration attribute expiration allows you to clear out data from integrations that are no longer reporting an asset. Attributes for assets that have not been reported by the integration in more than a specified number of days are automatically deleted. Assets with no remaining data sources are also removed. The automatic deletion task runs once per day.

Optionally, you can keep the latest attribute set per integration, even if it is older than the specified threshold.

You can disable stale integration attribute expiration by setting the number of days to zero.

Stale vulnerability expiration

Stale vulnerability expiration allows you to remove out-of-date vulnerability data. Vulnerabilities that have not been updated in more than a specified number of days are automatically deleted. The automatic deletion task runs once per day.

You can disable stale vulnerability expiration by setting the number of days to zero.

Scan data expiration

This setting controls how long runZero will retain the raw data and metadata for Scan and Import tasks. If your organization has a maximum data retention policy for cloud services, this setting can be used to force the removal of any data older than a given number of days. The default setting of 365 preserves one year of historical scan data.

Event records

Account-level event records (found under Alerts → Events) are always retained for one year. If you need to keep copies of these for longer than one year, these records can be exported and archived through the UX and API.

Data deletion after account termination

Upon termination of your runZero Platform subscription, your data will be deleted according to the table below.

Data Type	Default	Maximum
Stale assets (devices that have not received any new data from active, passive, or integration tasks)	180 days, unless requested earlier by emailing support[at]runzero.com	3 years (solely as set by you in runZero Platform prior to account termination)
Task data (raw data associated with each discovery task)	1 year, unless requested earlier by emailing support[at]runzero.com	3 years (solely as set by you in runZero Platform prior to account termination)
All other data (including data contained in support tickets, administrative data, etc.)	Upon written request by you to runZero at support[at]runzero.com	n/a
Any of the above data types archived as a copy on backup systems	1 year from date original data point is deleted	n/a

If you downgrade from your runZero Platform subscription to the free Community Edition, your data will be deleted according to the table below.

Data Type	Default	Maximum
Stale assets (devices that have not received any new data from active, passive, or integration tasks)	30 days, unless requested earlier by emailing support[at]runzero.com	3 years (solely as set by you in runZero Platform prior to account termination)
Task data (raw data associated with each discovery task)	10-30 days, unless requested earlier by emailing support[at]runzero.com	3 years (solely as set by you in runZero Platform prior to account termination)
All other data (including data contained in support tickets, administrative data, etc.)	Upon written request by you to runZero at support[at]runzero.com	n/a
Any of the above data types archived as a copy on backup systems	1 year	n/a

Managing access

runZero supports multiple concurrent users with a variety of roles. Roles can be set per-user on both a default and per-organization basis. The standard roles are administrator, user, billing, annotator, viewer, and no access. There is also a superuser role available to manage global settings.

Where there are multiple roles defined for a user, the access granted is based on most privilege. For example, if a user has user access by being in a group, but admin access assigned directly, they will be given admin privileges.

Available roles

Superuser

The first user created within the runZero console is considered a superuser. This role allows management of global settings like subscriptions and SSO parameters, and is shown as an access level of “everything”.

If you are a superuser, you can promote someone else to be a superuser. To do this, check the row listing them, and click the *Promote to superuser* button.

If you are using SSO authentication, you should configure at least one superuser with a strong password and MFA that can used as a backup if SSO settings need to be changed in the future.

We strongly recommend having more than one superuser, particularly if you are using MFA. That way if an MFA token is lost or a superuser leaves your organization, another superuser can fix the problem.

Administrator

Administrators can modify any aspect of an organization and have the unique ability to permanently delete bulk data, create additional organizations, and reset settings for other users.

User

Users have full access to an organization and can update sites, modify assets, schedule scans, and generally use most functionality. Users are not permitted to reset other users’ security credentials, bulk delete data, or delete an organization.

Billing

Billing users are unable to see any asset data, but can manage the licensing, billing, and entity settings for the account.

Annotator

Annotators have the same permissions as a viewer, except they have the ability to add tags to assets. Annotators do not have any other write-access within an organization, so they are unable to modify or remove existing tags. Modifications to existing tags must be made by a runZero user or administrator.

Viewer

Viewers have read-only access to an organization. This includes all inventory data and reports. Viewers are not allowed to interact with tasks, modify settings, or update assets. Viewers may not download the runZero CLI or install runZero Explorers, and they do not have access to view API tokens or export tokens.

No Access

The *no access* role is generally used as a default. Accounts with no access as a default are limited to those organizations where they have been granted access. If no organizations are allowed, the user is limited to managing their own account settings.

The no access global role can be used to create a single-organization user, such as a customer or third-party that needs access to the inventory for a specific organization. For consulting use cases, a single-organization user is a way to provide clients with visibility into their environment at no additional cost.

Another use for the *no access* role is to set it as the default for the account when you have no limits on who can sign in using an SSO system. You can then wait for the user to sign in and request access, before granting their newly-created account access to the appropriate organizations.

Inviting users

To add a team member, access the Your team page, and use the *Invite user* button to send an invitation.

The **Your team** menu entry has several submenus.

- The first, *Users*, shows all users in the current client account.
- The second entry, *Restricted*, goes to a page listing users who by default have no access to any organization.
- The next entry is the name of the current organization, as selected from the organization selector at the top of the screen. The page shows only users with access to that organization.
- The *External* entry goes to a page where you can invite users from other runZero client accounts.
- Finally, *Groups*, lists the user groups available. Groups can be used to set access and permissions users have within each organization.

User details

On the Users page, you can click on a user to view their details page. The user details include a list of their effective access to each organization. These details are split into three sections:

- *User access* lists access to organizations that has been directly granted to the user.
- *Group access* lists access to organizations that they are granted because they are in a group that has access to the organization.
- *SSO group access* similarly lists access from being in a group, but in this case for groups set as a result of SSO group roles.

Directly assigned user permissions can be edited using the gear icon button, either at the right side of the appropriate row of the user listing, or using the button at the top right of their user details page.

While editing user permissions using the gear icon, you are editing the explicit assigned roles. To see the resulting access levels, check the user details page.

Account settings

The *Account* page is available to superusers. It contains settings which apply to all users and organizations within the account.

Single-sign on (SSO)

runZero supports the implementation of SSO through SAML2. If you use a SAML2-compatible single sign-on (SSO) implementation, the SSO Settings page can be used to configure an Identity Provider (IdP) and allow permitted users to sign in to the runZero console.

Multi-factor authentication (MFA)

runZero supports multi-factor authentication, also known as two-factor authentication or 2FA. Physical hardware keys such as Google TitanKey and Yubico YubiKey are supported via the WebAuthn standard.

You can configure MFA policies for your account via the Account settings page. If multi-factor authentication is required, users who do not have an MFA token set up will be required to set one up when they next sign in. You can choose between requiring this for all users, or only requiring it for non-SSO users. The latter option is useful if your SSO server enforces MFA use.

Once a user registers one or more MFA tokens, they will be required to use one of the tokens every time they sign in.

Note that **changing the account settings to not require MFA will not alter the MFA status of existing accounts**. Existing accounts will keep any existing MFA tokens they have registered, and will still be required to use one to sign in. To disable MFA for a user, the user must clear the MFA token registration. To do this, they can go to their user settings page and click the red “Unlink” text next to the token ID in the bottom right.

Disabling support access

If you check the box labeled *Disable support access to your account*, runZero support staff will not be allowed to switch to your account.

If you choose to disable support access, this may make it harder for runZero support to answer any questions you have. In some cases we may need you to turn support access back on so that we can help you.

Idle times and sign in duration

You can set a maximum idle session time in minutes. If set, users whose web browsers don’t access runZero for the specified time period will be considered idle, and signed out.

You can also specify a maximum sign in duration. If set, users will be forced to sign in again regularly, at least once every specified period.

Account API keys

The Account API is a REST API which allows account-level operations such as adding and removing organizations and sites, adding users, and accessing the system event log. The *Generate API Key* button on the Account page can be used to generate a token which will allow access to the Account API.

License information

The *License* page shows information about your runZero software license, including how many assets you are licensed for, how many assets you have across all organizations, and when your license renews.

Entity information

The *Entity* page allows you to update information about the legal entity runZero is licensed to. You should ensure that this information is kept up-to-date if your company changes name or location, as we use the information to calculate taxes and ensure compliance with appropriate regulations.

Audit log

The *Audit log* page shows a history of all system events relevant to the superuser, such as login events, that are not visible within the organization Events page.

Managing user groups

User groups help streamline the management of users who need the same set of permissions. A user group explicitly sets the organizational role for users, which determines the tasks they can perform within each organization. You can assign roles at a per-organization level or assign a single role across all organizations. Single sign-on settings can also be applied to groups through SSO group mappings.

What happens if there are conflicting permissions?

runZero will always grant the role with the highest permissions level. For example, let's say an account has a viewer role for all organizations, but they've been added to a user group that has a user role for all organizations. This user will now have user-level permissions for all organizations. If the user group expires, the user's role reverts back to their account-level role.

User groups can also have an optional expiration date, which sets time-bound access to organizations within runZero for specific users. When the expiration date elapses, the user reverts back to their account-level permissions. If no expiration date is set, the user group settings will be persistent.

Creating user groups

runZero administrators and super users can create user groups.

1. Go to Your team > Groups and click Add Group.
2. Enter a name for the user group.
3. Choose the default role you'd like to assign to the group. This setting will establish the access level for every organization you have.
4. Set the per-organization roles, if you need to provide different access levels to specific organizations.
5. Set an expiration date for the user group, if you need to time-bound the permissions. When the expiration date elapses, the role for users part of the group will revert back to their user-level permissions. Otherwise, if you don't specify an expiration date, the user group will be persistent.
6. Go to the **Add users** tab and search for the users you want to add to the group. You can search by username or email.
7. When you are done adding users, save the user group. The user group will be listed on the Groups page.

Adding users to user groups

To add users to multiple groups at the same time, you can use the *Edit group membership* button on the Users page. The *Edit group membership* window will list every user group each user is currently in. Making changes from the *Edit group membership* window will apply to all

users you have selected. Only runZero administrators and superusers can add users to user groups.

1. Go to the Users page.
2. Select the users you want to add to a group.
3. Click the **Edit group membership** button.
4. Choose the user groups you want to add the users to.
5. Save your changes.

Setting an expiration date for a user group

runZero administrators and super users can set an expiration date for a user group.

1. Go to Your team > Groups.
2. Find the user group you want to assign an expiration date. Click the name to open the config page.
3. Set an expiration date for the user group. After the expiration date, the user's role will revert back to their account-level permissions.
4. Save the user group. You'll be able to see the user group's expiration date from the Groups page.

Viewing users in a user group

1. Go to Your team > Groups.
2. Find the *Users* column in the *User groups* table, which shows the user count for the group.
3. Click the user count to query and display the users assigned to the group.

Viewing user groups assigned to a user

1. Go to Your team > Users.
2. In the Groups column for each user is the number of groups that user is a member of.
3. Click on a number to show the corresponding list of groups.

Removing users from a user group

runZero administrators and super users can remove users from a user group.

1. Go to Your team > Groups.
2. Find the user group you want to modify. Click the name to open the config page.
3. Go to the *Users* tab.
4. Remove the users you no longer want assigned to the group. Their permissions will revert back to their account-level ones.
5. Save your changes.

Deleting user groups

runZero administrators and super users can delete user groups.

1. Go to Your team > User groups.
2. Select the user group you want to delete.
3. Click the **Delete** button.
4. Confirm you want to delete the user group.

Searching for users and user groups

When you are on Users page or Groups page, you can use the following keywords to search in the table:

Keyword	Description	Example
`id`	User's ID.	`id:123456789`
`name`	User's name.	`name:john`
`expires_at`	Time or date the user group expires.	`expires_at:>2weeks`
`created_at`	Time or date the user group was created.	`created_at:>2weeks`
`updated_at`	Time or date the user group was last updated.	`updated_at:>1year`
`has_expiration`	Whether the group has an expiration date.	`has_expiration:true`
`created_by_id`	ID of user who created the user group.	`created_by_id:123456789`
`created_by_email`	Email of the user who created the user group.	`created_by_email:user@example.com`
`group_id`	The user group ID.	`group_id:123456789`
`group_name`	The user group's name.	`group_name:group1`

Group IDs can be found in the URL for the group config page `https://console.runzero.com/groups/<groupid>/edit`.

The `group_id` keyword is only available for the users table; for the groups table, use `id`.

Bulk importing users

Community Platform

Instead of manually adding users one at a time, runZero administrators can add multiple users via bulk import. To bulk import users, you will need to create a CSV (comma separated values) file that contains the user information, such as their first name, last name, email, role, and organizational access.

Bulk imports will only add new users; it will not update existing users. If the file contains a user that already exists in the system, the import will not complete. You'll need to remove all duplicate users from your CSV file and import the file again.

Here are the high-level steps for bulk-importing users:

1. Create a CSV file with your user information.
2. Import the CSV file into runZero.
3. Verify users have registered their accounts.

Creating the CSV file for importing users

To create the CSV file, you can use the provided template or create a file from scratch. You'll need to follow the guidelines and include the fields outlined below.

- The first row of the CSV file is the header row and it must be included in the file. It contains the required fields needed for importing users, and those fields must follow a specific order (`first_name`, `last_name`, `email`, `all_orgs_role`, `org_roles`).
- runZero requires the `first_name`, `last_name`, and `email` fields. The organization-level fields, `all_orgs_role` and `org_roles`, are optional.
- To define the `org_roles` field, use the following format: `orgName=role`. You can use pipes to separate multiple definitions.
- Excluding the `all_orgs_role` field from the CSV will result in users being assigned a no-access role in all organizations.
- Excluding the `orgs_role` field from the CSV will result in users being assigned the role defined in the `all_orgs_role` field. If both fields are excluded from the file, all users will be assigned the no-access role. They will need to contact the runZero administrator to modify their account access.

User fields for the CSV file

The following table lists the fields you can include in your CSV file:

Field	Description	Example
`first_name`	The user's first name.	`sarah`
`last_name`	The user's last name.	`smith`

`all_orgs_role`	The role assigned to the user for all organizations. This is optional.	user
`org_roles`	The role assigned to the user for a specific site. Use pipes to separate the organizations. This field is optional.	org1=user

Your resulting CSV file should look like the following example:

```
first_name,last_name, email,all_orgs_role,orgs_role
Sarah,Smith,ssmith@company.org,user,org1=annotator|org2=annotator
```

Importing users into runZero

1. Go to your Team Import page.
2. Choose the CSV file that contains your user information and upload it to runZero.
3. Import the file.

runZero will alert you if there are issues with your file. For example, if the file contains an existing user, the import will not complete. You'll need to remove all duplicate users from your file and import it again.

Verifying users have registered

After you add new users via import, each user receives an email invitation to join your team. The email contains a link to register their email address, which prompts them to enter their name and password for their account. After they create an account, they'll be able to sign in to the runZero console.

You can visit the Teams page to find the status and last sign in for each user. Users who have completed the registration process will show an *Activated* status.

Managing external users

Community Platform

You can invite external users to join your runZero instance and view the organizational data available to them. The ability to add external users is useful for consultants, value-added resellers, and managed service providers who want to be able to share data from runZero with external partners and clients.

If you are a superuser, you can invite any user who has a runZero account to join your account. When you invite the user, they will receive an invitation to join your account via email. After they accept the invitation and sign in to runZero, they will see a new menu next to their organization switcher that lists all the clients they can access. Your client name will display in the list.

> Note
> If the person you want to invite does not have a runZero client account, they will need to sign up for one before you can invite them as an external user. A free runZero Community Edition account can be invited as an external user.

External users are only viewable from the External users page, separate from your other user lists. These users can use their same credentials to sign in and switch between clients and the organizations they've been granted access to.

Inviting an external user into your account

1. Go to **Your team > External** and click Invite external users.
2. Assign the default role you want to assign the user for all organizations.
3. Enter the email address for the user. The user must have an existing runZero account.
4. Customize access for the user at the per-organization level. runZero grants the highest permission levels, when there is a conflict between the global and per-organization permissions.
5. Add a custom email subject or body, if needed. runZero uses a default email template that informs the user you have invited them to your account and provides a link to activate the invitation. The email sender will be your name and the subject line will be "Invitation to join [client] by [your name]."
6. Send the email.

After they activate their invitation, they will be able to sign in to runZero, switch clients, and view your organizational data.

Switching clients and organizations as an external user

External users who have access to another account will see a client switcher, or client dropdown menu, located next to their organization switcher at the top right of the runZero web interface. Changing the client will change the organizations you can access and view.

The account that granted you access has configured the organizations and permissions you have within the client account.

Removing an external user from your account

Removing an external user from your account will remove their access to your data. The user will no longer be able to choose your client. To remove an external user, go to the External users page, select the user account you want to remove, and click **Remove user account**. Confirm you want to remove the user. They will be reverted to their primary client account and will no longer have the option to access your runZero data.

Implementing SSO

If you use a SAML2-compatible single sign-on (SSO) implementation, the SSO Settings page can be used to configure an SSO Identity Provider (IdP) and allow permitted users to sign in to the runZero console.

runZero's SSO implementation is designed to work with common SAML providers with minimal configuration, but there are a few requirements:

- Your users need to authenticate to a single domain such as `example.com`, not to multiple domains or a domain with many subdomains.
- The domain name needs to be configured in the SSO identity provider settings in runZero. This is true even for self-hosted runZero deployments.
- Your SAML IdP should provide something that looks like an email address in the NameID parameter. It doesn't need to be a valid email address, but it should be a unique value that has the same syntax as an email address (`user@example.com`).
- If the NameID does not look like an email address, runZero will check the fields `email`, `user.email`, `emailaddress` and `email address` for a suitable ID.
- runZero will check for the user's full name in the fields `name`, `gecos`, `user.name` and `displayname`. If no full name field is found, runZero will proceed to check for a first name in `first_name`, `firstname`, `given_name`, `user.firstname`, `givenname` or `first name`; and for a last name in `last_name`, `lastname`, `family_name`, `user.lastname`, `surname`, `sn`, or `last name`. These attributes are case insensitive.

Note that you must be a superuser to manage runZero SSO settings.

> Note
> If a user first registers using SSO, they are marked as an SSO-only user, and cannot bypass SSO by setting up a regular password. For this reason, we strongly recommend that you set up and keep a non-SSO superuser account, so that you can update the settings if SSO stops working for any reason.

Specific SSO providers

You can refer to the Azure AD or Okta pages for details on configuring these providers. The basic steps for configuration are:

1. Add runZero as an application.
2. Set up SSO in runZero.
3. Provision users to the runZero app.

For other SSO providers, the following information will help you to configure things.

Identity provider settings

To get to the identity provider settings, choose *Your team* from the left navigator, then click the SSO Settings button at the top right of the page.

The identity provider settings form is where you enter information about your SSO identity provider.

The easiest way to configure SSO is to use XML metadata, if your SSO service provides it. This will usually require that you set up runZero as an application on your SSO provider, then download an XML file. The XML can be then opened in a text editor, and copied and pasted into the box at the bottom of the runZero identity provider settings form. If the XML decodes successfully, the key fields on the form will then be set automatically.

Single Sign On mode

You can choose whether to allow regular runZero accounts, SSO-provisioned accounts, or both.

If both are allowed, you can set up backup administrator accounts in case your SSO provider is unavailable.

Domain name

The domain name is used by runZero to locate the correct SSO IdP settings when users choose to sign in via SSO. This is so that it can redirect them to the correct SSO provider to sign in. For example, if the domain is `example.com`, the runZero sign-in page will expect users to begin by entering an address of the form `username@example.com`.

The domain is also passed back by the SSO provider after sign in, and used to fetch the SSO IdP settings so that runZero can verify and process the data.

The domain is included as part of the ACS URL in the sign in request sent to the SSO provider. If you change the domain, you will need to reconfigure the SSO provider to accept the new ACS URL.

Issuer URL

The issuer URL, also often referred to as the entity ID, is a URL your SSO provider uses to uniquely identify itself in data passed back to runZero. You need to obtain this value from your SSO provider. Note that the value is not normalized to add or remove trailing slashes; it needs to match exactly.

Sign in URL

The sign-in URL is the URL users should be redirected to in order to begin the sign-in process for your SSO provider.

Certificate

The certificate is the PEM encoded CA certificate runZero will use to verify the signature on the data from your SSO provider.

If you paste multiple CA certificates into the box, your SSO provider will need to include a `KeyInfo` element in data passed back to runZero, to specify which certificate to check the signature against.

SSO walkthrough

To configure and debug SSO settings, it helps to understand how the SAML sign-in process works.

SAML SSO authentication is carried out using messages passed by the user's web browser. No direct connection occurs between runZero and the SSO identity provider (IdP) in either direction.

The normal sign-in process starts with the user going to the runZero console sign-in page, choosing **Sign in via SSO**, and entering their email address. The domain name of the email address is used to find the correct SSO identity provider settings in the runZero database.

runZero then redirects the user's web browser to the Sign in URL specified in those SSO IdP settings. Some additional parameters are added to the URL, according to the SAML specifications. These are verified by the SSO IdP.

The user logs in on the SSO IdP website. The IdP then redirects the user's web browser back to runZero, to the Assertion Consumer URL (ACS URL). The redirection includes a digitally signed blob of encoded data about the authenticated user. The ACS URL is shown on the service provider information tab of the SSO settings. Note that it includes the domain name, so that runZero can find the correct set of SSO settings to use to process the data.

runZero then decodes the encoded data, and checks its digital signature using the certificate that was provided as part of the identity provider settings. Assuming the certificate is valid, the decoded information about the user is then trusted by runZero. If necessary, a runZero user account is provisioned. The user is then signed in to begin a new runZero session.

Service provider information

After filling out the identity provider settings, you can switch to the service provider information tab to obtain information you need to provide to your SSO identity provider.

The Assertion Consumer URL, sometimes called the SSO URL, is the URL a person's web browser will be redirected to after they sign in successfully on the SSO identity provider. The URL contains the domain that was set up in the identity provider settings. runZero uses the domain to find the right set of SSO settings, so it can verify the data from the SSO provider. This means that if you change the domain in the identity provider settings for any reason, the ACS URL will change, and your SSO provider will need to be given the new URL.

The user sign-in URL is a URL that you can visit to begin the sign-in process. It displays the computed identity provider URL to aid in debugging.

Common problems

A user signs in on the SSO provider, but runZero refuses the sign in

The most common cause of this problem is that the IdP has been configured with an Assertion Consumer URL that does not include the domain name that was set in the SSO identity provider settings in runZero, or includes the wrong domain name.

A correct ACS URL will look like `https://console.runzero.com/auth/example.com/saml20/process` where `example.com` is the domain name configured in the runZero SSO identity provider settings.

Missing x509 Element

The full error message is "invalid SAML response: error validating response: Missing x509 Element"

This error can occur if you have entered multiple CA certificates into the certificate field of the SSO settings in runZero, but your SSO IdP didn't include a `KeyInfo` element to tell runZero which certificate to check the signature against. You will need to either configure your IdP to include information about which of the CA certificates to use as `KeyInfo`, or use a single CA certificate.

Managing SSO group mappings

Platform

> Only runZero administrators can automatically map users to user groups using SSO attributes and custom rules.

SSO group mapping allows you to map your SAML attributes to user groups in runZero. In runZero, user groups explicitly set the organizational role and determines the tasks users can perform within each organization. When you set up SSO group mappings, you explicitly define the SSO attribute and value you want to use for mapping. If there is a match, runZero will

apply the group settings for the user. As a result, you can ensure that SSO users are mapped to their respective groups in runZero.

For example, your IT team may need to be part of a group with administrator privileges. In this case, you can create a user group with an administrator role and then create an SSO group mapping that maps the SAML attribute that identifies your IT team to the user group. When someone from your IT team logs in to runZero, they will automatically be added with the appropriate access and permissions, all without pre-provisioning their account. After evaluating all SSO group mapping rules, runZero grants the user the highest privilege assigned for each organization.

Creating SSO group mappings

Before you create your SSO group mapping, make sure that you have set up SSO for your organization and created user groups. Both must be set up in order to successfully create SSO group mappings.

Only runZero super users can create SSO group mappings.

1. Go to Your team > SSO settings > Group mappings > Add group mapping.
2. In the **SSO attribute** field, enter the attribute you want to check for matching values. These values are defined in your SSO configuration.
    - For Azure AD SSO, note that the **SSO attribute** field must match the claim name from Azure AD.
3. In the **SSO value** field, provide a comma separated list of values that the attribute could match. When there is a match, runZero will apply the group permissions.
4. Click the **Group** dropdown and choose the user group that will be assigned if there is a match. The dropdown will list all user groups that have been created.
5. Save the SSO group mapping. These settings will apply the next time the user logs in to runZero.

Changes will not apply to users currently signed in. They will need to sign out and sign back in for the changes to take effect. You can forcibly sign out users to apply the SSO group mappings immediately.

Forcing a user to sign out

Changes to user permissions will not apply until the user signs out and logs back in to runZero. If you need to apply permissions immediately after setting up the SSO group mappings, you can forcibly sign out users. This will sign users out of their current session and require them to sign back in again. After they sign in, their updated permissions will be applied. Only superusers and admins who have access to all organizations can force sign-outs.

To forcibly sign out users, go to the Teams page and select the users you want to sign out. Click the sign-out button to log these users out.

Viewing SSO group mappings

To view all SSO group mappings that have been created, you can go to the Group mappings page. From this page, you can create, edit, or delete group mappings as needed.

Viewing SSO group mapping assignments

To see the SSO groups that a user has been assigned to, go to the Users page. From the *Groups* column, you can see the number of user groups and SSO groups the user is a part of. The number of SSO groups will be in parentheses.

Clicking the gear icon under actions will open the user settings for the user. The access summary tab will then display all of the organizations and roles they have assigned.

Deleting a group mapping

1. Go to the Group mappings page.
2. Select the group you want to delete and click the **Delete** button. All users provisioned through the group mapping will revert back to their account-level permissions.

Searching for SSO group mappings

When you are on the Group mappings page, you can use the following keywords to search in the table:

Keyword	Description	Example
`id`	User's ID.	`id:123456789`
`sso_attribute`	User's SSO attribute.	`sso_attribute:department`
`sso_value`	User SSO attribute value.	`sso_value:securityteam`
`created_at`	Time or date user group was created.	`created_at:>2weeks`
`updated_at`	Time or date user group was last updated.	`updated_at:>1year`
`created_by_email`	Email of user who created the group.	`created_by_email:user@example.com`
`group_id`	User group ID.	`group_id:123456789`
`group_name`	User group's name.	`group_name:group1`

Group IDs can be found by going to the group config page and enabling the ID column from the Columns menu above the data grid.

The `group_id` keyword is only available for the Users table; for the groups table, use `id`.

The `group_name` keyword is only available for the Users table.

Setting up Microsoft Entra SSO

Superusers can configure single sign-on to the runZero Console using an external identity provider (IdP), which enables authentication and user access control to the runZero Console from your single sign-on (SSO) solution. By default, runZero has SSO functionality available, but it's not a requirement to sign in to the console. You can make it a requirement or disable it completely.

Here are the high-level steps to set up SSO using Microsoft Entra (formerly Azure AD) to authenticate and manage user access to runZero:

1. Add and configure runZero as an Entra app.
2. Download the SSO configuration metadata in XML format.
3. Set up SSO in runZero.
4. Add users to your runZero app in Entra ID.

Requirements

Before you can set up SSO for the Microsoft Identity Platform:

- Verify that you have administrator privileges for Azure AD.
- Verify that you are a superuser in runZero. Look for the yellow star in your account status.

Step 1: Add and configure runZero as an Azure app

The first thing you need to do is add runZero as a non-gallery application to your Azure AD setup and to configure the settings for runZero as an Azure AD application.

1. In Azure, go to **Enterprise Applications > New Application > Create your own application**.
2. Under the **What are you looking to do with your application?** section, choose the **Non-gallery application** option.
3. Name your application something like `runZero`, and then add it.
4. Go to **Azure Active Directory > Enterprise applications** and open the newly created runZero application.
5. Select the **Single sign-on** tab, and then choose **SAML** as the sign-on method.
6. For the fields on the **Configure App Settings** page, go to https://console.runzero.com/team/sso/sp and copy the necessary service provider details:
    - Entity ID
    - Single sign-on URL
    - SSO callback (ACS) URL
7. Enter the values into the relevant fields in the Azure AD portal.
8. **Do not** set a value for "Sign on URL (Optional)" or "Relay State (Optional)".
9. If you want to use group information from Azure to set up group membership in runZero, configure the IdP to send group information.

Step 2: Download the SSO configuration metadata

While editing your application settings, you can get the download link to obtain the SSO configuration metadata in XML. You'll need this information to set up SSO in runZero.

1. On the **Configure App Settings** page, find the **SAML Signing Certificate** section.
2. Locate the XML download link under the Federation Metadata URL.
3. Download the file. You'll need the contents of this file for the next step.

Step 3: Set up Entra ID SSO in runZero

Now that you have the SSO configuration metadata in XML, you can configure Entra ID SSO settings in runZero.

1. Go to https://console.runzero.com/team/sso/idp to access the SSO IdP provider settings page in runZero.
2. Choose one of the following modes to enable SSO:
    - **Allowed** - Enables SSO, but users still have the option to sign in without SSO.
    - **Required** - Requires users to sign in with SSO. Only superusers can sign in without SSO.
3. Enter the domain name that is associated with SSO authentication. This is likely your company domain (companyabc.com).
4. Choose a default role for SSO users. This is the role all new users will be assigned when their account is created.
5. Copy the XML you downloaded from Azure and paste it into the Metadata XML field on the runZero SSO IdP page.
6. Apply your SSO settings. The remaining IdP fields will auto-configure for you.
    - The issuer URL will look something like `https://sts.windows.net/00000000-0000-0000-0000-000000000000/` where the UUID at the end is your unique Microsoft Active Directory (tenant) ID, listed under **App registrations > Overview > Endpoints**.
    - The sign-in URL will be something like `https://login.microsoftonline.com/00000000-0000-0000-0000-000000000000/saml2` with the zero UUID replaced with your unique tenant ID.
    - The certificate will be Microsoft's PEM encoded certificate, which will be extracted automatically from the XML.
    - On the Microsoft side, the redirection URL for runZero should be `https://console.runzero.com/auth/<domain>/saml20/process`, where <domain> is replaced with the domain specified in the runZero SSO settings.

Step 4: Add users to the runZero app in Azure

Now that you've completed the set up, you can go to the runZero app in Azure portal to add users and assign their access. Any users you add to the runZero app will be viewable from the Team members page in runZero, once they have signed in to runZero.

Step 5: Update SSO group mappings to match any configured Azure groups (if applicable)

If you have created user groups within Azure, you will need to update your SSO group mappings in runZero to associate the groups created in Azure with user groups in runZero. This will ensure that the appropriate access and permissions are added to your users when they sign in to runZero.

When setting up the SSO group mappings, note that the **SSO attribute** field must match the claim name from Entra ID.

Setting up Okta SSO

Superusers can configure single sign-on to the runZero Console using an external SAML identity provider (IdP), such as Okta, which enables authentication and user access control to the runZero Console without typing in credentials.

Here are the high-level steps to set up single sign-on (SSO) using Okta to authenticate and manage user access to runZero:

- Add runZero as an application in Okta.
- Set up SSO in runZero.
- Add users to the runZero app in Okta.

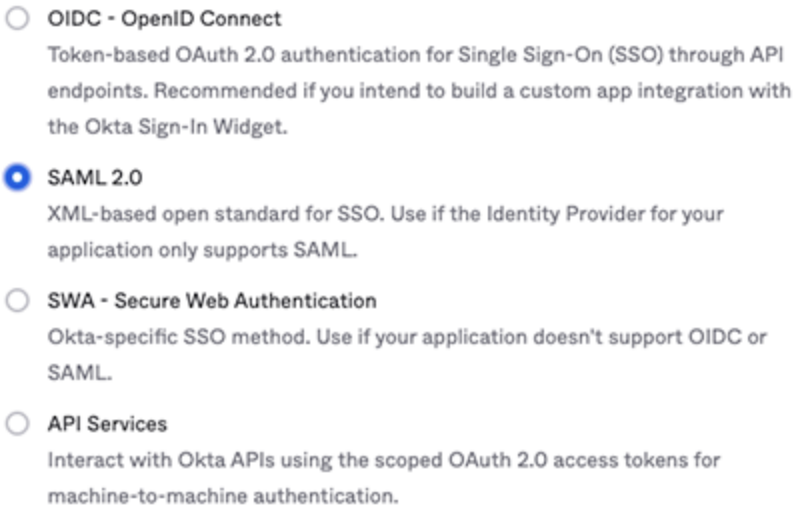
Requirements

Before you can set up Okta SAML:

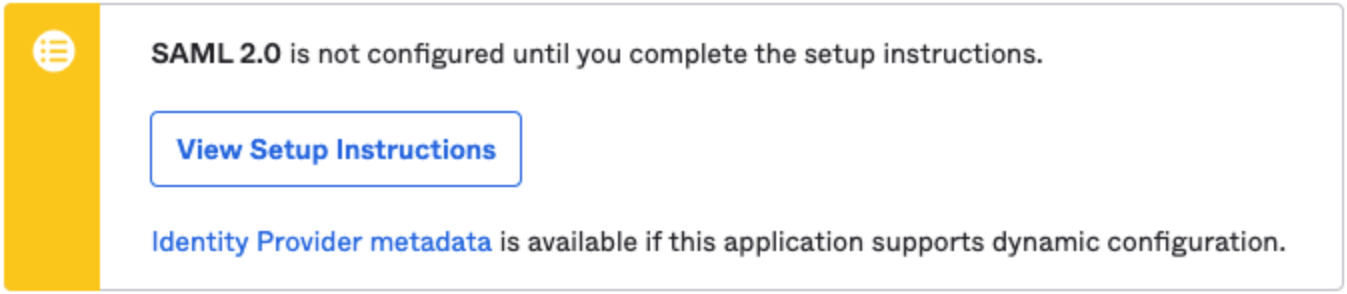
- Verify that you have administrator privileges for Okta.
- Verify that you are a superuser in runZero. Look for the yellow star in your account status.

Step 1: Add and configure runZero as an Okta app

1. Go to **Okta > Applications > Create App Integration**. When the **Create a new app integration** window appears, select `SAML 2.0` for your sign-in method.



2. For the general settings, you'll need to provide a name for the app. Call the app `runZero`. You can also add a logo to make it easier for users to identify the runZero app.
3. For the SAML settings, you'll need to go to the service provider information page in runZero to find the relevant URLs.
    - **Single sign on URL** - In runZero, this is the assertion consumer service (ACS) URL.
    - **Audience URI or SIP Entity ID** - In runZero, the entity ID, or SAML audience, will be `https://console.runzero.com`.
        - **NOTE:** make sure to include the leading 'https://' when entering this field in Okta.
4. For the remaining settings, like the attribute statements, visit the Okta documentation to learn how to configure them.
5. When you finish configuring the SAML settings, Okta will prompt you for some feedback on how you will be using the app. You can opt to provide feedback or skip to complete the set up.
6. After Okta creates the app, you will need to view the SAML 2.0 instructions to complete the set up. Go to the the **Sign On** tab for the runZero app and view the SAML 2.0 instructions. You'll need these details for the next step.



Step 2: Set up SSO in runZero

1. Go to the SSO setup page in runZero. Choose one of the following modes to enable SSO:
    - **Allowed** - Enables SSO, but users still have the option to sign in without SSO.
    - **Required** - Requires users to sign in with SSO. Only superusers can sign in without SSO.
2. Enter the domain name that is associated with SSO authentication. This is likely your company domain (companyabc.com).
3. Choose a default role for SSO users. This is the role all new users will be assigned when their account is created.
4. Copy the fields from Okta into runZero.
    - **Issuer URL** - In this field, enter the **Identity Provider Issuer URL** from Okta. This will look something like `http://www.okta.com/<ID>`
    - **Sign-in URL** - In this field, enter the **Identity Provider Single Sign-On URL** from Okta. This will look something like `http://<okta-instance>/app/<app-name>/<ID>/sso/saml`.
    - **Certificate** - Copy and paste the entire contents of the X.509 certificate from Okta.
5. Apply your SSO settings.

Step 3: Add users to the runZero App in Okta

Now that you've completed the set up, you can go to the runZero app in Okta to add and manage user access. After you've completed this step, your users will be able to go to your SSO sign-in URL to access runZero.

Step 4: Update SSO group mappings to match any configured Okta groups (if applicable)

If you have created user groups within Okta, you will need to update your SSO group mappings in runZero to associate the groups created in Okta with user groups in runZero. This will ensure that the appropriate access and permissions are added to your users when they sign in to runZero.

Managing licenses

As a runZero superuser or billing user, you can access and manage your organization's licensing, plan, and billing information.

Not a superuser or billing user? Please contact your organization's superuser or billing user to get help with licensing and billing information. These users are tagged with a yellow star.

How do I view my license?

If you're a superuser or billing user, go to Account > License to view your runZero licensing information.

Your licensing information will show:

- The number of organizations, sites, Explorers, and user accounts you have.
- The number of recent assets you have across all organizations.
- The number of project assets you have.
- The total number of recent assets and project assets your license supports.

What count as recent assets?

runZero calculates the number of recent assets based on how many have been seen during the last 30 days.

For assets that have been scanned by runZero, if they have been seen by a scan in the last 30 days, they are counted as recent assets.

If third party data indicates a device has been seen in the last 30 days, that will also cause the asset to count as recently seen.

License limits are uniform across all asset types. It doesn't make any difference whether the asset is online or offline, scanned or unscanned, or where the asset data was sourced from. For assets with multiple data sources, with different dates that the device was last seen at, the most recent date applies.

The number of recent assets is calculated in the background. If you delete assets it may take a short while for the number to recalculate.

When does my subscription expire?

To see when your subscription or license expires, go to Account > License.

Find the line: `This is a runZero [edition] subscription that expires at [date and time].`

If your subscription has expired, you will see: `This is a runZero [edition] subscription that expired on [date and time].` You will no longer be able to run discovery scans. To continue using

runZero, you will need to renew your subscription.

Does the Community license expire? Nope, it's part of our free tier and has no expiration date. You can continue to use it as long as you don't have more than 100 recent assets.

How do I renew my subscription?

For help renewing your subscription, please contact us by email or by using this online form.

How do I convert to the Community Edition?

No longer need the full platform? You can convert it to the Community Edition. This plan is on our free tier and supports up to 100 recent assets.

How do I find my invoices?

Super users and billing users can access invoices from the runZero Console. Go to the License page to see all invoices.

How do I change or cancel my subscription?

Please contact us if you need to modify or cancel your subscription.

Network discovery

runZero can gather asset data through unauthenticated active scanning, passive traffic sampling, and inbound integrations.

Active scanning

The runZero Explorer and scanner perform unauthenticated active scanning of your specified networks based on the configurations you set. They leverage various network protocols to discover and fingerprint assets connected to the network.

Active scans can be configured to run once or on a schedule. Scan templates can also be used to ensure consistency across multiple scan tasks.

Traffic sampling

Community Platform

Explorers can be configured to perform passive traffic sampling in your environment in order to identify assets communicating across your networks. Traffic sampling is configured on a per-Explorer basis on the Explorer details page. Explorers will automatically pause traffic sampling activities in order to handle scan tasks, continuing to parse sniffed traffic once the scan completes.

The traffic sampling feature can also ingest a PCAP file to glean useful asset data.

Inbound integrations

Inbound integrations for many different platforms can be configured to enrich your runZero asset context and provide important insights into coverage and capabilities. Integrations can be configured either as connectors or scan tasks.

Active scanning

An active scan identifies all responsive devices on a given network, fingerprints these devices, and populates the asset, services, screenshot, and software inventory.Regular scans of internal and external networks is an important step in network management. Scans are configured by site, Explorer, and scan scope. The scan scope can include IP ranges, domain names, ASNs, and even entire country codes.

When creating a new scan, you have multiple parameters you can set, ranging from scheduling a date to more advanced options. To get started, login to the runZero Console, select Scan fromthe Data sources section of the navigation menu, and choose “Start Standard Scan”. Scans can also be launched from the Inventory views.

Site

runZero organizes information into organizations and sites. Organizations are distinct entities that are useful for keeping data separate and contain a collection of sites. Sites are used to model segmented networks, particularly independent networks which use the same private IP address ranges.

For example, you might have multiple physical locations with their own local networks, all using the 10.0.0.0/8 private IP range. By defining them as sites, you can set up an Explorer for each, and the networks and assets will be treated as completely independent even if similar systems are seen at the same IP addresses in each.

> Note
> runZero treats each site as a separate IP space. This enables overlapping subnets within an organization, but can result in duplication if the same network is scanned in different sites within the same organization.

Since scan analysis occurs at the site level, the boundaries you define for a site set the default scope for scans for that site.

Explorer

Select the Explorer to run the scan from, chosen from the set of registered Explorers for the site. The Explorer you choose must be able to directly communicate with the networks and addresses you define for the discovery scope. The chosen Explorer should ideally be able to reach all addresses in the scope directly, without a firewall in the way. Stateful firewalls and VPN gateways may interfere with the discovery process.

Hosted zone

Community Platform

runZero Platform users can perform scans of public IP space using runZero-hosted scanners. When creating a scan, set the Explorer to **None** and choose a hosted zone from which to scan. When using this option, the discovery scope must use public IP addresses or ranges, or resolve to public IP space.

Discovery scope

The discovery scope defines the IP addresses that will be scanned. The scope uses the site settings when specified as they keyword “defaults”, but may be changed on a per scan basis as well. The scope should include at least one IP address or hostname. IPv4 address ranges can be specified in most standard formats:

- `10.0.0.1`
- `10.0.0.0/24`
- `10.0.0.0/255.255.255.0`
- `10.0.0.1-10.0.0.255`

IPv6 addresses can be specified individually, but IPv6 ranges are not supported.

Hostnames specified in the scope will be resolved at runtime by the assigned Explorer. If the hostname returns multiple IP addresses, all addresses in the response will be scanned. Hostnames can also have masks applied, indicating that the mask should expand to each resolved address of the hostname. For example, if `example.com` resolves to both `1.2.3.4` and `5.6.7.8`, the input of `example.com/24` would become `1.2.3.0/24` and `5.6.7.0/24`. IPv6 addresses returned from hostname resolution will be scanned if the Explorer has a valid IPv6 address and route to the target.

> Note
> runZero load balances the scan across as many subnets as possible, in a quasi-random order.

Discovery keywords

The following keywords are supported for both scan scopes and exclusions.

- **asn4**: The `asn4:<AS number>` keyword can be used to specify IPv4 ranges associated with a given AS number.
- **country4**: The `country4:<ISO code>` keyword can be used to specify IPv4 ranges associated with a given two-character country code.
- **public** and **private**: The `public:<mode>` and `private:<mode>` keywords can be used to specify IPv4 and IPv6 addresses associated with assets in the current organization. The mode parameter can be set to `all`, `primary`, or `secondary` to indicate which IP addresses are used. The `public` keyword selects all non-reserved IP addresses associated with organization

assets. The `private` keyword selects all RFC-1918 and private use IP addresses associated with organization assets.

- **domain**: The `domain:<domain>` keyword is available to cloud-hosted users and uses the syntax `domain:<domain name>` to automatically select publicly-known hostnames for a given domain name.

Scan name

You can assign a name to your Scan task to make it easier to keep track of.

Scan speed

Specify the maximum packet rate for the overall discovery process, in network packets per second. 500 is conservative, 3000 works for most LANs including WiFi, 10000 or more may be helpful for large sites with fast connectivity.

The scan speed directly affects how long the scan will take to complete. An approximate formula is:

time in seconds = hosts × ports × attempts ÷ scan speed

The number of hosts scanned is primarily determined by the discovery scope. The number of ports is around 500 by default, and three attempts are made to connect.

The number of hosts and ports scanned can be affected by the advanced scan options, and speed can also be impacted by maximum host rate and group size; see the descriptions of the advanced scan options below.

Note also that this formula doesn't take into account time taken to take screenshots, follow web server redirects, or process the scan data.

Schedule

You can set a date and frequency for your scan task. Dates and times take into account your browser's advertised timezone.

Scans scheduled to start in the past will be launched immediately and then repeated at the specified time based at the frequency selected.

Scheduling grace period

Specify the number of hours to wait for an available Explorer before giving up on this scan. A zero or negative value will result in the scan retrying indefinitely until an Explorer becomes available.

Scan duration limit

You can specify a number of hours to limit scan duration to; if scanning is still in progress after this time has elapsed, the scan will be canceled. This does not limit processing time.

If you set this to 0, no limit is applied.

Advanced scan options

The Advanced tab can be used to display and modify additional scan settings, such as network exclusions, scan speed, the ports covered by the TCP scan, and which probes are enabled. The default settings should work for most organizations but may need to be tweaked for slow networks or unreliable links.

Maximum host rate

As well as setting an overall scan rate in packets per second, you can also control the maximum rate at which packets are sent to any single host IP address. This is useful when you have devices which are easily overloaded by network traffic. The default should be safe for most systems.

Max group size

When runZero scans your network, it spreads the scan load across many IP addresses at once. The max group size determines how many IP addresses can be actively scanned at once – allowing for the fact that hosts may take some time to respond to probes. The max group size needs to be at least as large as the overall scan speed, or else it would limit the speed of the scan to below the set value. If you provide a value that's lower than the overall scan speed, it will be increased automatically at scan time.

The max group size is mostly useful when dealing with stateful network devices that can only track a limited number of connections at once, as a way to restrict how many active TCP sessions will result from a runZero scan.

Max TTL

The IP standards define a maximum hop count for packets. In IPv4, this is called the Time To Live or TTL, while on IPv6 this is called the Hop Limit. Every device processing a packet must decrease the TTL or Hop Limit one. If this value reaches zero, the route receiving the packet must discard the packet. This setting can be used to set the maximum hop limit for scan traffic.

ToS

The IP standards define a Type of Service or ToS for packets. In IPv4, this is called the Type of Service or ToS, while on IPv6 this is called the Traffic Class or TC. The ToS or Traffic Class is used by switches and routers to prioritize network traffic. The lower bits of the IPv4 ToS are also used for congestion controller. This setting can be used to set the ToS or Traffic Class for scan traffic. Please note that the ToS/Traffic Class settings do not apply to all traffic sent by runZero, but instead are limited to the basic discovery probes. Some protocols, such as SNMP, and integrations, such as VMware, do not set the ToS/Traffic Class fields on their corresponding packets. If all scan traffic must be consistently tagged with the correct ToS or Traffic Class, this can be accomplished through settings on the managed switch port instead.

TCP ports

The **Included TCP ports** and **Excluded TCP ports** fields can be used to override the default scan ports. The string “defaults” will lookup the current default port list at scan time. The current port list is:

1	7	9	13	17	19	21	22	23	25	37
42	43	49	53	69	70	79	80	81	82	83
84	85	88	102	105	109	110	111	113	119	123
135	137	139	143	161	179	222	264	280	384	389
402	407	442	443	444	445	465	500	502	512	513
515	523	524	540	541	548	554	587	617	623	631
636	664	689	705	717	743	771	783	830	873	888
902	903	910	912	921	990	993	995	998	1000	1024
1030	1035	1080	1083	1089	1090	1091	1098	1099	1100	1101
1102	1103	1128	1129	1158	1199	1211	1220	1234	1241	1260
1270	1300	1311	1352	1433	1434	1440	1443	1468	1494	1514
1521	1530	1533	1581	1582	1583	1604	1610	1611	1723	1755
1801	1811	1830	1883	1900	2000	2002	2021	2022	2023	2024
2031	2049	2068	2074	2082	2083	2100	2103	2105	2121	2181
2199	2207	2222	2224	2323	2362	2375	2376	2379	2380	2381
2443	2525	2533	2598	2601	2604	2638	2809	2947	2967	3000
3001	3003	3033	3037	3050	3057	3071	3083	3128	3142	3200
3217	3220	3260	3268	3269	3273	3299	3300	3306	3311	3312
3351	3389	3460	3500	3502	3628	3632	3690	3780	3790	3817
3871	3872	3900	4000	4092	4322	4343	4353	4365	4366	4368
4369	4406	4433	4443	4444	4445	4567	4659	4679	4730	4786
4840	4848	4949	4950	4987	5000	5001	5003	5007	5022	5037
5038	5040	5051	5060	5061	5093	5168	5222	5247	5250	5275
5347	5351	5353	5355	5392	5400	5405	5432	5433	5498	5520
5521	5554	5555	5560	5580	5601	5631	5632	5666	5671	5672
5683	5800	5814	5900	5901	5902	5903	5904	5905	5906	5907
5908	5909	5910	5911	5920	5938	5984	5985	5986	5988	5989
6000	6001	6002	6050	6060	6070	6080	6082	6101	6106	6112
6161	6262	6379	6405	6443	6481	6502	6503	6504	6514	6542
6556	6660	6661	6667	6905	6988	7000	7001	7002	7021	7070

7071	7077	7080	7100	7144	7181	7210	7373	7443	7444	7474	7510
7547	7579	7580	7676	7700	7770	7777	7778	7787	7800	7801	7860
7879	7902	8000	8001	8003	8006	8008	8009	8010	8012	8014	8020
8023	8028	8030	8080	8081	8082	8083	8086	8087	8088	8089	8090
8095	8098	8099	8100	8123	8127	8161	8172	8180	8181	8182	8205
8222	8300	8303	8333	8400	8443	8444	8445	8471	8488	8500	8503
8530	8531	8545	8649	8686	8787	8800	8812	8834	8850	8871	8880
8883	8888	8889	8890	8899	8901	8902	8903	8983	9000	9001	9002
9042	9060	9080	9081	9084	9090	9091	9092	9099	9100	9111	9152
9160	9200	9300	9380	9390	9391	9401	9418	9440	9443	9471	9495
9524	9527	9530	9593	9594	9595	9600	9809	9855	9999	10000	10001
10008	10050	10051	10080	10098	10162	10202	10203	10250	10255	10257	10259
10443	10616	10628	11000	11099	11211	11234	11333	12174	12203	12221	12345
12379	12397	12401	13364	13500	13778	13838	14330	15200	15671	15672	16102
16443	16992	16993	17185	17200	17472	17775	17776	17777	17778	17781	17782
17783	17784	17790	17791	17798	18264	18881	19300	19810	19888	20000	20010
20031	20034	20101	20111	20171	20222	20293	22222	23472	23791	23943	24442
25000	25025	25565	25672	26000	26122	27000	27017	27018	27019	27080	27888
28017	28222	28784	29418	30000	31001	31099	32764	32844	32913	33060	34205
34443	34962	34963	34964	37718	37777	37890	37891	37892	38008	38010	38080
38102	38292	40007	40317	41025	41080	41523	41524	44334	44343	44818	45230
46823	46824	47001	47002	47290	48899	49152	50000	50013	50021	50051	50070
50090	50121	51443	52302	52311	53282	54321	54921	54922	54923	55553	55580
57772	61614	61616	62078	62514	65002	65535					

Prescan modes for large IP spaces

Sometimes, the scope of your IP space is unknown, subnet usage is unknown, and the total number of assets is unknown. These unknowns can make it challenging to optimize your discovery scans for efficiency and speed. And when your IP space is large, like a /16 space with a few thousand IPs in use, a full discovery scan can take more time to complete, since it looks at more than 500 TCP ports and 15 UDP ports on every address. In these types of cases, you may want to tune your scan settings to prefilter ranges and IP addresses before a full scan.

runZero has two prescan modes that you can use to run a faster scan: subnet sampling and host ping.

Subnet sampling

Community Platform

To speed up scans of large subnets you can use the **“Only scan subnets with active hosts”** advanced scan option. If this option is on, a prescan runs against the target space to identify the subnets with an active host. This mode leverages heuristics runZero has collected to identify addresses that are more likely to be responsive across subnets. This process allows runZero to quickly scan larger spaces by identifying the subnets that are in use, before starting full probes. All subnets that are identified as having active hosts are then fully scanned – unless you enable host pings.

There are two tweakable parameters for subnet sampling. The sample rate determines what percentage of addresses in each subnet are prescanned to determine if the subnet should be scanned. The subnet size determines how many IP addresses are in each subnet. By default, the subnet size is 256 addresses, corresponding to a /24 subnet, and 3% of the addresses in each subnet are prescanned.

Host ping

After you have some insights on the subnets that are in use, you may want to limit the full scan to only addresses that respond to the most common ping methods, such as ICMP and some TCP and UDP ports. If you choose the **“Limit scans to pingable hosts”** advanced scan option, only hosts that respond to a ping request will be fully scanned.

The runZero Explorer uses multiple protocols for ping scans:

- Conventional ICMP ping, performed by sending an ICMP echo request and looking for an ICMP echo reply.
- TCP ping, performed by sending a TCP SYN packet to a series of common ports and seeing whether the host responds with RST or TCP SYN/ACK.
- UDP ping, performed by sending a packet to port 65535 and checking for an ICMP response of port unreachable.

The set of ports used for TCP and UDP ping can be adjusted in the LAYER2 section of the Probes and SNMP tab when setting up a scan task.

Note that it is relatively common for enterprise firewalls to be set up to block ping, or for hosts to be set up not to respond to ping requests. Limiting scans to pingable hosts can therefore result in assets being missed entirely, even if their IP addresses are probed. If your goal is to speed up scan times, subnet sampling is usually the better option.

It’s possible to use both subnet sampling and limiting scans to pingable hosts at the same time, but this is not recommended except as a last resort for reducing scan times.

Initial network scans

Background

Once you have an Explorer installed, you can start using it for network discovery. While our goal is to configure scheduled scans that we set and forget, we need to go about our first scans in a more structured manner.

The goals of our first scans are to:

- Verify the Explorer is setup properly and has everything installed
- Validate Explorer connectivity to varying parts of the network
- Determine how long scans will take at varying sizes to help with future scheduling

Your first few scans

To get started, you will want to scan a few smaller ranges to make sure everything is working as expected. Start with a few /24 network blocks from each of the RFC 1918 ranges to make sure everything looks good.

For setting up the first scan:

1. Navigate to **Sites > New Site > Create** a new temporary site within the Organization
2. Navigate to **Tasks > Scan > Standard Scan** to create a scan task
3. Chose the new site you created in step 1
4. Include a range of the RFC1918 IP addresses in the Discovery Scope, plus a small network or two that you know is in use. A suggested value for the RFC1918 range includes:

   ```
   10.0.0.0/24,10.0.255.0/24,10.64.0.0/24,10.64.255.0/24,10.128.0.0/24,10.128.255.0/24,
   10.192.0.0/24,10.192.255.0/24,10.255.0.0/24,10.255.255.0/24,192.168.0.0/24,192.168.64.0/24,
   192.168.128.0/24,192.168.192.0/24,192.168.255.0/24,172.16.0.0/24,172.23.0.0/24,172.31.0.0/24,
   <your networks here>
   ```

5. On the Advanced tab, enable the *Subnet sampling* option
6. Click on Initialize Scan

After these scans are complete, you will want to check for these things:

- Check the `ipv4.traceroute` value for assets in each RFC1918 range to verify you aren't sending traffic to an edge router or firewall.
  - Unused private IPs should have routes stubbed out to prevent traffic from being sent to the default gateway which can create a loop. You can also verify this with traceroutes from the Explorer.
- If your scan results have a large series of somewhat sequential IPs that have only ICMP or a very small number of similar ports open on them, that's probably a proxy or firewall. Check out those IPs to see if any are real. To find assets with only ICMP enabled use the inventory query `alive:t AND service_count:=1 AND service_count_icmp:=1`
  - You can add an allow rule for the Explorer IP to properly scan devices on the other side
  - Another option is to add a second Explorer on the other side of the proxy or firewall
- If you receive reports or alerts about service outages, check for session aware devices such as routers, firewalls, and proxies that are having issues handling the session load. If you run into this, there are multiple ways to approach solving the issue.
  - The simplest solution is to set up another Explorer on the other side of the device and run scans separately
  - Another option is to segment your scans on the existing Explorer, and run smaller, separate scan tasks for the network ranges on the other side of the device with lower packet per second and *max group* sizes to minimize the number of IPs that will be scanned at once
- Check how long each scan took to get an idea for how long larger scans would take
- Verify you see screenshots on ports that accept HTTP/HTTPS requests
  - If you don't see any, you likely need to install Chrome on the machine

- Check for MAC addresses
  - If you aren't seeing them you should configure SNMP
  - If SNMP is configured, you should verify community strings and check for unmanaged switches

Note
Once you have verified that your first scan ran successfully, you can delete the temporary site and set up a real scan.

Full RFC 1918 scans

Once you have completed initial test scans, it's time to expand scanning to cover all subnets with live assets. One method of discovering all subnets with live assets is to run full RFC 1918 scans.

runZero offers a **Full RFC 1918 discovery** scan option that will discover assets across the following private address ranges as a single task.

- 10.0.0.0/8 or 10.0.0.0-10.255.255.255
- 172.16.0.0/12 or 172.16.0.0-172.31.255.255
- 192.168.0.0/16 or 192.168.0.0-192.168.255.255

The **Full RFC 1918 discovery** scan option is only recommended for small networks with limited complexity and should only be leveraged in a single site configuration.

Discovering the entire RFC 1918 private address space in a single scan can takes days, if not weeks, to complete in a large complex network. runZero recommends reviewing the Achieving RFC 1918 coverage playbook for more information on scanning the entire RFC 1918 private address space.

Once you've completed scans of your private address space, review Identifying gaps in scanning to learn more about some of our built-in reports that we offer to help you get a better understanding of your network.

Identifying gaps in scanning

Background

After you have run a full network discovery scan, you can start to better understand your coverage and begin to optimize. By the end of this guide, you will understand how to use the out of the box reports in runZero to understand your gaps in network coverage.

RFC 1918 coverage

The first report to look at is the **RFC 1918 coverage** report. This report shows you which internal IPv4 subnets have been scanned, which likely contain assets, and which are still unknowns.

The scan coverage maps show all the addresses scanned within the `10.0.0.0/8`, `172.16.0.0/12`, and `192.168.0.0/16` ranges. See the legend to understand what percentage of each address space has been scanned. Clicking into any of the scanned subnets gives you access to the subnet grid for deeper asset analysis.

Identify scanned and un-scanned areas with the coverage map: On the flip side, red outlines indicate that there are un-scanned addresses runZero has indirect knowledge of that haven't been scanned directly. For example, this can happen when runZero finds a secondary IP address on a multi-homed device within a scanned subnet. The red boxes highlight the subnets most likely to be in use, but un-scanned.

Scan missing subnets: From the coverage report, you can launch a scan for any missing subnets in a given RFC1918 block – look for the binocular icon.

Scan missed subnets: The missing subnets will be shown in the scan scope and the subnet ping will be enabled by default. You can tune the scan configuration as needed for your environment.

Subnet utilization

The **Subnet utilization** report can provide similar visibility into your network to the RFC 1918 coverage report, but with emphasis on the subnets that contain live assets. This report will enumerate each of the subnets defined in your site definitions, and provide a count of live assets for that site and subnet, along with a utilization percentage. If there are live assets that are outside any site subnet, they will be aggregated into an inferred subnet based on the network mask size you select.

From this report, you can pivot to the asset inventory for a given subnet or initiate a new scan of a subnet. Another benefit of this report is that you can export the results as a CSV. This can be helpful for more complex data analysis and for scheduling recurring scans.

Switch topology

View layer-2 link information extracted from SNMP-enabled switches. This report can be used to find unmapped assets and investigate why they aren't showing up in your scans.

Configuration for this report: The Switch Topology report uses data enumerated via SNMP to map switch ports to assets. In environments where SNMP v1 or v2 with default public or private communities are in use, this enumeration happens automatically. Non-default communities for SNMP v1/v2/v3 can also be provided in the scan configuration. Clicking on a node in this report will expand it to show its connections.

Finding unmapped MACs: This topology view is helpful when trying to understand how a given asset or switch is connected, but also provides a critical data point related to risk; the

number of unmapped assets. An unmapped asset is a MAC address connected to a switch, but not found in an ARP cache or through any of the other techniques runZero uses for remote MAC address discovery.

Re-scan to properly map MAC addresses: For environments where a runZero agent is connected to each network segment, unmapped MACs may highlight VLANs or network segments that are missing from the scan scope. In environments where runZero is scanning assets multiple hops away, the unmapped asset count can provide an estimate of how well the remote segment is being identified.

Unmapped MACs

If you prefer a condensed view to see all unmapped macs, you can also use the Unmapped MACs report. This report shows all the unmapped MACs in your organization, organized by switch, and again by port. This data can be used similarly to the unmapped MACs data from the Switch Topology report.

Managing scan templates

A scan template is a predefined set of scan options and settings. If you have a scan configuration you use often, you can create a scan template to save those settings. The next time you create a scan, you can choose a template instead of manually configuring your settings. Each update you make to the scan template is automatically applied to new and recurring scans based on the template, as well as any queued scans which were set up using the template but have not started yet. With scan templates, you can save time and reduce the likelihood of misconfiguring a scan.

When you create a scan based on a scan template, you will not be able to edit the fields set by the template. However, you can configure the site, scan name, discovery scope, Explorer, and scan schedule, since these are not defined in the template.

Each time a scan runs using values from a template, the scan task is saved with a copy of the parameters. This means the task will list the values used for the scan, even if the template is modified after the scan completes.

Creating a scan template

Scan templates can be created in a few ways in runZero:

- By going to **Tasks > Task library**
- From an existing scan task
- Based on an existing scan template

Creating a scan template from scratch

1. Go to **Tasks > Task library** to view the task library.
2. From the **Task library**, click **Add template**.
3. When the scan configuration form appears, enter a name for the template. Use this name to search and assign the template to a new scan.

4. Configure the scan as you normally would. The following tabs are available for you to access and configure settings:
    - **Standard** - Provide a name, description, and scan rate for the template.
    - **Advanced** - Configure excluded hosts, included ports, tags, host rates, group size, UDP probe max attempts, max TTL for all scan probe packets, ToS, screenshots, and subnet sampling.
    - **Probes and SNMP** - Enable SNMP and additional probes, and provide credentials. When you enable the SNMP probe, the scan will identify devices that are using the SNMP protocol, using the port and credentials provided to find anything with SNMP enabled.
    - **Credentials** - Enable any preconfigured Scanning with credentials you want to use for the scan. The scan uses the credentials if the defined CIDR scope matches the specific scan target.
5. Save the template.

To use this template, go to the *Templates* tab when you configure a scan or create a Template scan from the scan menu on the inventory or tasks pages.

Creating a scan template from a scan task

1. Go to **Tasks** to open the Tasks overview page.
2. Find the scan task you want to use as the basis for your scan template and click the name to view the task details.
3. From the task details, click the **Copy** dropdown menu and choose **Copy for new template**.
4. When the scan configuration form appears, enter a name for the template. Use this name to search and assign the template to a new scan.
5. Configure the scan as you normally would. The following tabs are available for you to access and configure settings:
    - **Standard** - Provide a name, description, and scan rate for the template.
    - **Advanced** - Configure excluded hosts, included ports, tags, host rates, group size, UDP probe max attempts, max TTL for all scan probe packets, ToS, screenshots, and subnet sampling.
    - **Probes and SNMP** - Enable SNMP and additional probes, and provide credentials. When you enable the SNMP probe, the scan will identify devices that are using the SNMP protocol, using the port and credentials provided to find anything with SNMP enabled.
    - **Credentials** - Enable any preconfigured Scanning with credentials you want to use for the scan. The scan uses the credentials if the defined CIDR scope matches the specific scan target.
6. Save the template.

To use this template, go to the *Templates* tab when you configure a scan or create a Template scan from the scan menu on the inventory or tasks pages.

Creating a scan template based on a template

1. Go to the **Tasks > Task library** to view the task library.
2. From the **Task library**, find the template you want to use as the basis for a new template and click the name to open it.

3. From the scan template configuration page, click **Copy**. runZero will duplicate and create a new scan template.
4. When the scan configuration form appears, enter a name for the template. Use this name to search and assign the template to a new scan.
5. Configure the scan as you normally would. The following tabs are available for you to access and configure settings:
    - **Standard** - Provide a name, description, and scan rate for the template.
    - **Advanced** - Configure excluded hosts, included ports, tags, host rates, group size, UDP probe max attempts, max TTL for all scan probe packets, ToS, screenshots, and subnet sampling.
    - **Probes and SNMP** - Enable SNMP and additional probes, and provide credentials. When you enable the SNMP probe, the scan will identify devices that are using the SNMP protocol, using the port and credentials provided to find anything with SNMP enabled.
    - **Credentials** - Enable any preconfigured Scanning with credentials you want to use for the scan. The scan uses the credentials if the defined CIDR scope matches the specific scan target.
6. Save the template.

To use this template, go to the *Templates* tab when you configure a scan or create a Template scan from the scan menu on the inventory or tasks pages.

Applying a scan template to a scan

Instead of manually configuring a scan, you can choose to use a template instead. You can find all templates available for a scan on the **Templates** tab in your scan configuration.

You can also go to the **Task library** page and choose to **Create a scan based on this template** from the template's actions.

Editing a scan template

All updates to a scan template affect every scan that uses it. After you make a change to a template, the next scan will automatically update to use the new settings. Any scan queued to run will use the new settings.

Editing a scan that uses a scan template

Any changes to a template will affect all scans that use them. If a scan uses a template, you will need to edit the template in order modify the scan settings. Otherwise, you will need to delete the scan and create a new scan configuration with your modifications.

When you create a scan based on a scan template, you will not be able to edit the fields set by the template. However, you can configure the standard fields, such as the site, scan name, discovery scope, Explorer, and scan schedule, since these are not defined in the template.

Deleting a scan template

Deleting a template will remove it permanently from runZero. All configurations will no longer be accessible. To delete a scan template, go to the Task library, find the template you want to delete, and click **Delete**. A dialog will prompt you to confirm that you want to delete the template, and the action will be irreversible.

Scanning with credentials

The Credentials page provides a single place to store any secure credentials needed by runZero, including:

- SNMPv3 credentials
- Access secrets for cloud services like AWS and Azure
- API keys for services such as Censys and Miradore

Credentials are stored in encrypted form in the runZero database. Credentials, such as SNMP passwords, are used by runZero Explorers and are transmitted to them in encrypted form. For security reasons, the secret part of any credential cannot be viewed once entered.

When adding a credential, you can choose to make it a global credential that can be used for all organizations or to allow access only by specific organizations. The **Allow all** or **Disallow all** buttons let you quickly apply the same setting across all organizations. Individual organizations can also be toggled to allow or disallow access.

Most credential fields can be edited after the credential is saved. Some fields, like URLs, cannot be edited after saving for security reasons. Sensitive fields, such as passwords or access keys, will be hidden but can be overwritten.

Credential settings

The specific fields and options for a credential depend on the type of credential.

VMware and SNMP credentials, which are used by the runZero Explorer, allow a CIDR allow list to be specified. This can be used to limit which scanned IP addresses the credential will be used with. This feature allows you to avoid sending SNMP or VMware credentials to all scanned hosts on the network, and instead limit them to specific IP addresses or ranges.

Credential verification

Credentials can be verified when created or edited to ensure they can successfully authenticate. Choose **Verify & save** when creating or editing a credential to run the verification before saving. If verification fails, it will display an error message and then give you the option to **Save anyway**. **Save anyway** will save the most recent verification status.

Credential management

Users must have administrator-level permissions to manage credentials. Users with **Administrator** as their default role can fully manage all credentials. Users with per-organization permissions do not have access to global credentials, and are only able to manage credentials in organizations where they have administrator permissions. A shared credential cannot be deleted by an organization administrator.

Scanning with SNMP

SNMP is an open standard network protocol for collecting information about devices on a network.

There are three main versions of the protocol.

SNMP versions 1 and 2

SNMP version 1 was designed in the 1980s as an interim protocol, intended to be replaced by ISO CMIP. It was built to be used across any network common at the time, not just TCP/IP networks, so security was left up to the host network. The protocol defined a community string for arbitrary organization of groups of assets, but didn't specify how access should be granted.

SNMP version 2 attempted to introduce a security model based on "parties," but it wasn't widely adopted so a revised standard was issued as SNMP v2c. SNMP v2c removed party-based security and went back to just using community strings. The previous SNMP v2 specification is considered obsolete.

Unfortunately, both SNMP v1 and SNMP v2 send community strings in plain text across the network. There's no encryption of SNMP v1 or v2 data packets, and because the protocol is based on UDP there's no way to simply add TLS to make a secure connection.

SNMP version 3

SNMP v3 fixes the security problems of SNMP v2 by supporting both password-based authentication and encryption (referred to as privacy protection) as part of the standard. Unfortunately there are multiple algorithms supported for both the encryption and the password authentication, and you have to know which ones your network devices use. Unlike SSH for example, the protocol doesn't include any kind of negotiation of encryption methods.

SNMP v3 devices can operate in three different modes:

- noAuthNoPriv, meaning no authentication is required and there's no privacy protection (no encryption),
- authNoPriv, meaning authentication *is* required but there's still no privacy protection, and
- authPriv, meaning both authentication and privacy protection are required.

runZero's SNMP support

runZero supports SNMP v1, SNMP v2 (the common `v2c` variant), and SNMP v3. Scans can be performed using only v1/v2, only v3, or both.

SNMP scanning is on by default. You can turn it off or customize it using the SNMP tab when setting up a scan or a scan template.

SNMP v1 & v2 scanning

There are two ways to set up community strings for SNMP v1/v2 scanning. The first is to enter them as a comma-separated list on the SNMP tab. By default, runZero supplies the community strings `public` and `private`, as these are common defaults on network-enabled hardware such as printers and NAS servers. If you remove those defaults, runZero will not probe with them.

The second way to set up community strings is to enter them using the credentials feature as the credential type *SNMP v2 Communities*. While communities technically shouldn't be used as credentials because they're sent in plain text across the network, in practice many networks use them that way. By entering your community strings as credentials, you can use the CIDR allow list feature to control which parts of the network the community strings will be sent to, reducing the risk of their being captured by rogue devices.

If you remove all community strings from the SNMP tab and do not set up any SNMP v2 Communities as credentials, no SNMP v1/v2 scanning will be performed.

SNMP v3 scanning

Most devices which provide potentially sensitive information such as serial numbers and software versions will only do so in response to an *authenticated* SNMP v3 query. To perform authenticated SNMP v3 scanning, you will need to set up an *SNMP v3 Credential* record in runZero's credentials feature.

The authentication protocol determines the hashing algorithm used to process the authentication passphrase – that is, how the runZero Explorer logs in to the remote device.

The most common default algorithm that devices use for authentication is `sha` (HMAC-SHA-96), which is required to be supported by the SNMP v3 standards. Newer devices may support more secure variants such as SHA-256, and runZero supports up to SHA-512. HMAC-MD5-96 is also supported, as per the standards, but is best not used on your network because the MD5 algorithm is known to be insecure.

The privacy protocol determines how the data sent to and from the remote device is encrypted to prevent eavesdropping. The privacy passphrase is used as seed data to initialize the encryption.

The most common privacy algorithm, required by the original SNMP v3 standards, is CBC-DES-128. It's selected as `des` in the runZero user interface. Later RFCs added AES-128,

represented as `aes`. Again, runZero supports more secure variants that may be used by newer devices, such as AES256.

As mentioned above, devices may require only authentication, both authentication and privacy, or neither. For some devices there is only space for a single passphrase in the device configuration. This usually means that the same passphrase is used for both the privacy passphrase and the authentication passphrase, if the device is running in authPriv mode.

As with SNMP v2 community strings, you can set a CIDR allow list to determine which IP addresses SNMP v3 credentials will be sent to. This is particularly recommended if your network is not set up to require privacy passphrases.

In addition to credentialed scanning, runZero will gather information from the pre-authentication handshake of SNMP v3.

To perform SNMP v3 scanning across Cisco switches, modify the ACLs to include the following rule:

- Newer IOS versions: `snmp-server group YourGroupName v3 auth context vlan- match prefix`
- Older IOS versions: `snmp-server group YourGroupName v3 auth context vlan-1` (repeated for every VLAN)

Choosing which versions of SNMP are scanned

To disable SNMP v1/v2 and only scan with SNMP v3, remove the community strings from the SNMP tab when setting up a scan or template and ensure that any SNMP v2 community strings stored as credentials are disabled on the Credentials tab.

To disable authenticated SNMP v3 scanning and only scan with SNMP v1/v2, ensure that any SNMP v3 credentials are disabled on the Credentials tab and make sure you have community strings specified, either on the SNMP tab or a credential.

To disable all SNMP scanning, including uncredentialed SNMP v3, switch off the toggle switch on the SNMP tab labeled *Use the SNMP protocol for discovery*.

How scans are performed

runZero will probe devices using all configured SNMP versions and all community strings. This is because devices often respond with different levels of data for SNMP v3 versus SNMP v2, and may respond with different information depending on community string.

SNMP is based on UDP, and UDP doesn't guarantee the order of data packets, so it's impossible to guarantee in what order community strings will be received by remote devices.

SNMP v3 authentication errors are not treated as scan failures. This is because it's common to have multiple sets of SNMP v3 credentials used on different parts of a network, as well as misconfigured devices with incorrect passphrases.

Checking SNMP v3 credentials and SNMP v1/v2 authentication

If you are not getting SNMP data in your assets and think that you should be, a common cause is network firewalls, which often block SNMP traffic entirely. If possible, deploy an Explorer on the other side of the firewall, on the network segment you want to scan with SNMP.

The second most common reason for not getting the expected SNMP data in assets is incorrect SNMP v3 credentials.

- If SNMP v3 authentication was attempted, you will see an *SNMP engine ID* recorded in the asset as `snmp.engineID.raw`. The engine ID is usually a long string of hex digits and is used to calculate access keys.
- If SNMP v3 authentication failed, you will see an error reported in the attribute `snmp.failedAuth`. If the error is a request timeout, that likely indicates that authentication succeeded but the encrypted data channel could not be set up, which means you may have the wrong privacy password or privacy algorithm.
- If SNMP authentication was attempted successfully (v1/v2 or v3), you will see an `snmp.credentials` attribute with the UUID of any credentials record that succeeded.
- If SNMP v1/v2 authentication via community string succeeded, you will see `snmp.secretCommunities` or `snmp.defaultCommunities` attributes, depending on the type of community string.
- If SNMP v1/v2 authentication succeeded using community strings entered directly into the task request rather than credentials records, the `snmp.secretCommunities` and/or `snmp.defaultCommunities` attributes will still be present, but `snmp.credentials` will be a zero UUID.

Debugging credential issues

It can be frustrating to debug SNMP v3 credential problems by trial and error using multiple scan attempts. Instead, you can use the `snmpwalk` utility. It's part of the net-snmp open source project, packaged for most Linux distributions and available for macOS via Homebrew. There are unofficial builds for Windows, or you can install it in the Windows Subsystem for Linux (WSL).

The `snmpwalk` utility will connect to an IP address and dump all of the information it can retrieve. Here's an example of how to use it to connect in `authPriv` mode with both authentication and privacy passphrases:

```
snmpwalk -v3 -l authPriv -a SHA -A "authentication passphrase"
  -x AES -X "privacy passphrase" -u username 10.0.1.25
```

(Text above has been line-wrapped, but it's one long command.) The parameters are:

- `-v3`: switch to SNMP v3 mode.
- `-l authPriv`: run in authPriv mode, with both authentication and privacy required.
- `-a SHA`: authentication algorithm is SHA (SHA-128).
- `-A`: specify the authentication passphrase as the next argument.
- `-x AES`: privacy (encryption) algorithm is AES (AES-128).
- `-X`: specify the privacy passphrase as the next argument.

- `-u username`: specify the login username.
- `10.0.1.25`: the IP address to probe.

If you have the right passphrases and algorithms, you'll get data back, probably a lot of it. If not, you'll get an error message which may help identify the cause.

To check SNMP v2 community strings, you can use a command like this:

```
snmpwalk -v2c -c commstring 10.0.1.25
```

The `-v2c` argument chooses SNMP v2 (SNMP v2c) mode, and the `-c` argument specifies the community string. If you have the wrong community string, `snmpwalk` will run for a while and then print a timeout error.

Reviewing discovered SNMP services

When SNMP services are found during a scan, their protocol versions are tracked at the asset level where you will see `snmp` in addition to `snmp1`, `snmp2`, and `snmp3` depending on which version responded. This makes for performant queries via the asset inventory, such as when querying for `protocol:snmp2`.

Addtionally, SNMP services track how they authenticated and which protocols they used. If a commonly used value for a SNMP v2 community was used (such as `public`, `private`, some vendor defaults, and common values like `password`, `cisco`, and `community`), these will be listed in cleartext in the service details under `snmp.defaultCommunities` and asset details under `snmp.v2DefaultCommunities` with the list of communities that responded. If the SNMP v2 community is not a common value, this will be reported as `snmp.secretCommunities` with a value of `true` in the service details and in an attribute called `snmp.v2SecretCommunities` with a value of `true` in the asset details. If SNMP v3 is used, then `snmp.v3Usernames` will be populated at the asset level. Lastly, an asset attribute called `snmp.auth` is populated indicating whether `v2DefaultCommunity`, `v2SecretCommunity`, or `v3Username` was successful to authenticate to the asset.

Object Identifiers (OIDs)

While some of these OIDs are used only for specific vendors, this is a complete list of all OIDs potentially used on a given asset.

System attributes

- `1.3.6.1.2.1.1.*.0`

IP addresses

- `.1.3.6.1.2.1.4.20.1.3.*`

MAC addresses

- `.1.3.6.1.2.1.2.2.1.6.*`

ARP caches

- `.1.3.6.1.2.1.4.35.1.4.*`
- `.1.3.6.1.2.1.3.1.1.2.*`
- `.1.3.6.1.2.1.4.22.1.2.*`

Routes

- `.1.3.6.1.2.1.4.24.7.1.7.*`

Port (CAM) tables

- `.1.3.6.1.2.1.17.7.1.2.2.1.2.*`
- `.1.3.6.1.2.1.17.4.3.1.2.*`

VLANs

- `.1.3.6.1.2.1.17.7.1.4.3.1.1.*`
- `.1.3.6.1.4.1.9.9.46.1.3.1.1.4.1.*`

Interface names

- `.1.3.6.1.2.1.17.1.4.1.2.*`
- `.1.3.6.1.2.1.31.1.1.1.1.*`

Serial numbers

- `.1.3.6.1.2.1.47.1.1.1.1.11.*`

Device models

- `.1.3.6.1.2.1.47.1.1.1.1.13.*`

Hostnames

- `.1.3.6.1.4.1.9.2.1.3.*`

Model names from Juniper switches

- `.1.3.6.1.4.1.2636.3.1.2.0.*`

Serial numbers from A10 devices

- `.1.3.6.1.4.1.22610.2.4.1.6.2`

Please note that Cisco Catalyst & Nexus switches require per-VLAN enumeration using either community indexing (v2) or contexts (v3). runZero automatically enumerates all ARP caches and port tables for each discovered VLAN.

runZero gathers information from SNMP v3 systems even when credentials are not available. This information includes the engineID as well as values from the following OIDs

- `.1.3.6.1.6.3.15.1.1.1.0`: `usmStatsUnsupportedSecLevels`

- `.1.3.6.1.6.3.15.1.1.2.0`: `usmStatsNotInTimeWindows`
- `.1.3.6.1.6.3.15.1.1.3.0`: `usmStatsUnknownUserNames`
- `.1.3.6.1.6.3.15.1.1.4.0`: `usmStatsUnknownEngineIDs`
- `.1.3.6.1.6.3.15.1.1.5.0`: `usmStatsWrongDigests`
- `.1.3.6.1.6.3.15.1.1.6.0`: `usmStatsDecryptionErrors`

Using custom fingerprints

Community Platform

Customers running a self-hosted instance or using the standalone scanner have the ability to use custom-written fingerprints. This can be useful in adding new fingerprint coverage for very unique or custom assets and services, such as device prototypes or proprietary applications/services. Custom fingerprints can also be used to override existing, similar runZero fingerprints by using a same-or-higher certainty value.

> Note
> When using the runZero standalone scanner with custom fingerprints, you'll need to use the `RUNZERO_EXTERNAL_FINGERPRINTS` value as an environment variable when launching the scanner.

Create new fingerprints

Custom fingerprints follow the structure and format of the open-source Recog fingerprint database. You can author your own fingerprint XML entries in files of similar name and format to those found in Recog. For cases where an asset or service matches both a built-in runZero fingerprint and a custom fingerprint of the same kind, preference will be given to the fingerprint with higher “certainty” value(s) (e.g. `hw.certainty`, `os.certainty`, `service.certainty`). In the event of a certainty “tie” (i.e. same certainty value(s)), the custom fingerprint will be given preference.

Add new fingerprints to your self-hosted runZero instance

To ensure the self-hosted instance of runZero can properly access your custom fingerprints, they will need to exist within the runZero installation directory (/opt/runzero by default). The following steps will get your custom fingerprints setup for use by runZero:

- Create a new directory within the runZero install directory (e.g. `mkdir /opt/runzero/myfingerprints`)
- Update your `/etc/runzero/config` file with the new directory location (see below)
- Copy your custom fingerprint XML files into the new directory (e.g. `cp *.xml /opt/runzero/myfingerprints`)

The `/etc/runzero/config` file needs to be updated with the location of your custom fingerprints directory, which you can do by using your favorite editor to add the following line to the end of your config file (swapping `myfingerprints` with the name of the directory you added):

```
RUNZERO_EXTERNAL_FINGERPRINTS=myfingerprints
```

Note that you only need to do the directory creation and update of `/etc/runzero/config` file once. After that, you can add, remove, or modify your custom fingerprints in the directory as needed and then restart the runZero application to reload the current custom fingerprints.

Restart the runZero service

The runZero self-hosted instance will need to be restarted when custom fingerprints are added, removed, or updated. On restart, the runZero log file can be monitored to verify if the custom fingerprints were successfully applied or if an error occurred. On successful load of custom fingerprints, a log message like the following should be present:

```
@cee:{"level":"info","msg":"loaded (5) external fingerprints","time":"2022-09-12T19:51:49Z"}
```

If issues are encountered when loading or processing custom fingerprints, or if a 10 second timeout is reached, a warning message will be logged and the runZero application will continue running without any custom fingerprints.

Verify your fingerprints

Once your custom fingerprints have been added to your self-hosted runZero instance and the instance restarted, you can verify that the custom fingerprints are performing as-expected in one of the following ways:

- Running a scan task to go scan a relevant asset/service, or
- Importing an existing scan data file of the relevant asset/service

Following completion of the scan task OR the import of scan data, depending on which method you chose, you can then navigate to a specific asset or specific service and verify that your custom fingerprints are acting as-expected.

Passive sampling

As well as active scans, you can set up runZero Explorers to listen passively for network traffic. While network scans will typically provide better information, passive sampling can be useful on networks where you are not permitted to scan. It can also be useful for finding networks that are not routable from your Explorers.

Only one of sampling or scanning can be active on a single Explorer at any time, but you can schedule scan tasks for an Explorer with passive sampling enabled and the system will ensure that the scans still run. When the Explorer has no tasks assigned, it will go back to passive sampling.

Configuring passive sampling

Passive sampling is set up on the Explorer configuration page. Choose *Deploy* from the left navigator and click on an Explorer. The *Passive traffic sampling* box allows you to configure the feature.

The network interfaces available to the Explorer will be shown on the left. You do not need to use an interface connected to a SPAN or TAP port; a regular network interface will work.

The network interface used for passive sampling needs to permit promiscuous mode, where all traffic is passed unfiltered. If it does not, passive sampling will fail and an empty task will be created with an error shown.

The *Discovery scope* specifies the range of IP addresses to create assets for if traffic is seen. You can also specify *Excluded hosts* if there are IP addresses you want to ignore traffic from.

The *Site* option allows you to set the site where newly discovered assets will be created. It has no affect on the range of IP addresses assets are created for.

You can set *Asset tags* on assets discovered through passive traffic sampling. This can be helpful for reviewing the new assets and setting up scans to obtain more information about them.

Platform integrations

Inbound integrations

Enriching runZero results with data from other tools

The runZero platform offers integrations with several sources of asset data, allowing users to enrich their asset inventory and identify assets and subnets that are not effectively managed or protected. By leveraging product APIs and export/import functionality, runZero can pull data from many IT and security tools to extend visibility across your organization's network.

Supported integrations

Cloud and virtualization

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- VMware
- Wiz

Endpoint protection

- CrowdStrike Falcon
- Microsoft 365 Defender
- Microsoft Intune
- Miradore MDM
- SentinelOne
- Tanium API Gateway

Endpoint management

- Microsoft Endpoint Configuration Manager (MECM)

Asset and identity management

- Google Workspace
- Microsoft Active Directory
- Microsoft Entra ID (formerly Azure AD)

Vulnerabilities and risk

- Censys Search & Data
- Qualys VMDR
- Rapid7
  - InsightVM
  - Nexpose
- Shodan
- Tenable
  - Tenable Vulnerability Management
  - Tenable Nessus Professional
  - Tenable Security Center
  - Tenable Nessus (file import)

Network management

- Cisco Meraki Dashboard

Custom integrations

Community Platform

If the solution you want to draw data from isn't available as a current runZero integration, Platform and Community users can leverage the custom integrations feature to add asset data from custom sources. Adding custom asset sources can be accomplished through the API or by leveraging the runZero Python SDK.

Scan probes or connector tasks

Most integrations can be run either as a scan probe or a connector task.

Scan probes run as part of a scan task. The scan task can be used to scan your environment and sync integrations at the same time. To run an integration as a scan probe:

1. Configure a scan task from the *Scan* menu in your inventory or tasks page.
2. Activate the integration under the *Probes* tab.
3. Activate the correct credential under the *Credentials* tab.
4. Configure, activate, or deactivate other scan task configuration options as preferred.

Connector tasks run independent of scan tasks in order to allow more finely tuned scheduling of integration syncs and asset scans. Connector tasks are run from the runZero cloud by default, but can be configured to run from an Explorer in your organization if preferred. To run an integration as a connector task:

1. Configure a connector task from the Integrations page or the *Integrate* menu in your inventory or tasks page.
2. Select an Explorer from the *Explorer* menu (optional).

3. Configure, activate, or deactivate other connector task configuration options as preferred.

Importing integration data

Some integrations can be used by importing data from that platform into runZero. For example, `.nessus` files from Tenable Nessus and `.xml` files from Rapid7 Nexpose can both be ingested without requiring a connection to their APIs.

Automatic asset merge

How runZero maps integration assets to assets:

- For hosts that can be matched to an existing runZero asset, asset-level attributes will be updated, and integration-specific attributes will be added.
- For hosts that cannot be matched with an existing runZero asset, a new asset will be created in the site specified when the integration task is set up.

runZero is able to merge integration data into existing assets by the following, in priority order:

1. MAC address
2. IP address (3-day window)
3. Hostname

Assets from integrations can also be manually merged into runZero assets using the *Merge* button on the Asset Inventory page.

Removing an integration data source

When an integration is removed as a data source, the associated attributes are removed from your runZero assets. Since some asset attribute fields are merged, it is possible that attributes populated by both runZero scans and the integration could be deleted. Rescanning the affected assets will resolve this issue.

Excluding integration attributes

Your account and organizations can be configured to prevent the collection of certain integration attributes; for example, personally identifiable information (PII) such as usernames or email addresses. Please note that some attributes are crucial to runZero's merging and fingerprinting methods, and excluding these attributes may lead to inaccuracies in your inventory.

Excluded attributes can be configured in your Account settings under *Personally Identifiable Information (PII) collection* to apply across all organizations, or configured within individual Organization settings to override the account-level settings.

You can choose to use the default list of attributes or define your own. The attributes must be in the format `@source.resourceType.attributeName`. For example, `@crowdstrike.dev.firstLoginUser`

excludes the attribute “firstLoginUser” from CrowdStrike devices.

Only integrations that import PII support attribute exclusions, at this time those integrations are:

- CrowdStrike
- SentinelOne
- Tanium
- Google Workspace
- Intune
- Miradore

Attribute exclusions are not supported for the Directory Users and Directory Groups inventories. The AzureAD, LDAP, and Google Workspace integrations import directory users and groups by default, but this can be disabled in the task configuration.

Source names and IDs

The table below maps the source name to the source ID for querying assets and vulnerabilities.

ID	Name	Description
-1	`custom`	Custom
1	`runzero`	runZero
2	`miradore`	Miradore
3	`aws`	AWS
4	`crowdstrike`	CrowdStrike
5	`azure`	Azure
6	`censys`	Censys
7	`vmware`	VMware
8	`gcp`	GCP
9	`sentinelone`	SentinelOne
10	`tenable`	Tenable
11	`nessus`	Nessus
12	`rapid7`	Rapid7
13	`insightvm`	InsightVM
14	`qualys`	Qualys
15	`shodan`	Shodan
16	`azuread`	AzureAD

17	`ldap`	LDAP
18	`ms365defender`	MS365Defender
19	`intune`	Intune
20	`googleworkspace`	GoogleWorkspace
21	`sample`	Sample
22	`tenablesecuritycenter`	TenableSecurityCenter
23	`packet`	Packet
24	`wiz`	Wiz
25	`meraki`	Meraki
26	`mecm`	MECM
27	`tanium`	Tanium
28	`simulator`	Simulator
29	`netbox`	NetBox
30	`cip`	CIP
31	`pan`	Palo Alto Networks
32	`prisma`	Prisma

Amazon Web Services

Community Platform

runZero integrates with Amazon Web Services (AWS) to provide better visibility across your cloud environment. This integration imports data from each applicable API to add detailed information to your asset inventory:

- AWS EC2 API
- AWS RDS API
- AWS ELBv1 API
- AWS ELBv2 API
- AWS Lambda API

Syncing with AWS allows you to quickly identify the number of EC2 instances, elastic load balancers, and relational database services you have running, as well as their region, account, and more.

This integration supports the import of all running EC2 instances, RDS instances, and active application, network, gateway, or classic load balancers. It can be configured to connect to a single AWS account or to all accounts in your organization and imports data across multiple regions.

Getting started

The following AWS resource types are supported:

- EC2 instances
- Elastic load balancers
- RDS instances
- Lambda instances

To set up the AWS integration, you'll need to:

1. Configure AWS to allow API access through runZero.
2. Add the AWS credential to runZero, which includes the access key and secret key.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the AWS integration to sync your data with runZero.

Requirements

Before you can set up the AWS integration:

- Make sure you have access to the AWS console.
- Make sure you are using AWS Organizations if you want to connect to multiple accounts.

Step 1: Configure AWS to allow API access through runZero

To connect to a single AWS account:

1. Sign in to the AWS console.
2. Go to **Identity and Access Management (IAM) > Users** and select or create a user that will provide API access to runZero.
3. Click **Add permissions > Attach existing policies directly**. Search for and attach the following policies based on the services you would like to sync:
    - `AmazonEC2ReadOnlyAccess` to sync EC2 and ELB resources
    - `AmazonRDSReadOnlyAccess` to sync RDS instances
    - `AWSLambda_ReadOnlyAccess` to sync Lambda functions
4. From the user summary screen, open the **Security credentials** tab and click on **Create access key**.
5. Save the **Access key ID** and **Secret access key**.
6. If you intend to sync with AWS regions that are not enabled by default, change the region compatibility of the global STS endpoint to be compatible with All AWS Regions. To do this, in the AWS console navigate to IAM → Account Settings → Security Token Service (STS). Then change the Global Endpoint to be compatible with "All AWS Regions."

To connect to all accounts in your organization:

1. Sign in to the AWS console.

2. For each account in your organization, create a role and assign the `AmazonEC2ReadOnlyAccess`, `AmazonRDSReadOnlyAccess`, and `AWSLambda_ReadOnlyAccess` policies. You can do this one at a time for each account or use StackSets to deploy the role if you have a large number of accounts:

   Option 1: Create a role using IAM

   1. Go to **Identity and Access Management (IAM) > Roles** and click **Create role**.
   2. Choose **Another AWS Account** for the type of trusted entity.
   3. For Account ID, enter the ID for your organization's management account.
   4. Click **Next: Permissions**.
   5. Attach the `AmazonEC2ReadOnlyAccess` policy if you want to sync EC2 and ELB resources.
   6. Attach the `AmazonRDSReadOnlyAccess` policy if you want to sync RDS instances.
   7. Attach the `AWSLambda_ReadOnlyAccess` policy if you want to sync Lambda functions.
   8. Click **Next: Tags** and add tags optionally.
   9. Click **Next: Review** and provide a name for the role. (The role must be named the same for all accounts)
   10. Click **Create role**.

   Option 2: Create and deploy a role to multiple accounts using StackSets

   1. Create an IAM account for runZero to use for AWS access, such as `arn:aws:iam:::user`.
   2. While signed in to your organization's management account, go to **CloudFormation > StackSets** and click **Create StackSet**.
   3. Select **Template is ready** and upload a file with the JSON template for the stackset provided below. Replace `<accountID>` with the account ID where the role was created, `<userName>` with the name of the user you created, and `<roleName>` with the name of the role. Click next.
   4. Enter a name for the StackSet. Click next.
   5. Optionally set tags. Click next.
   6. Set the deployment options. Click next.
   7. Review and create the StackSet.

3. For your organization's management account, create an inline policy to allow the `STS AssumeRole` action.

   1. Go to **Identity and Access Management (IAM) > Users** and select or create a user that will provide API access to runZero.
   2. Click **Add inline policy**.
   3. In the JSON tab, enter the inline policy text from below, replacing `<rolename>` with the role name.

4. If you intend to sync with AWS regions that are not enabled by default, change the region compatibility of the global STS endpoint in the management account to be compatible with All AWS Regions. To do this, in the AWS console for the management

account navigate to IAM → Account Settings → Security Token Service (STS). Then change the Global Endpoint to be compatible with “All AWS Regions.”

5. From the user summary screen, open the **Security credentials** tab and click on **Create access key**.
6. Save the **Access key ID** and **Secret access key**.
7. From the user summary screen, open the **Permissions** tab and click **Add permissions**. Attach the `AWSOrganizationsReadOnlyAccess` policy.
8. (Optional) Attach the `AmazonEC2ReadOnlyAccess` policy to the user in your organization’s management account if it has EC2 or ELB instances you would like to sync.
9. (Optional) Attach the `AmazonRDSReadOnlyAccess` policy to the user in your organization’s management account if it has RDS instances you would like to sync.
10. (Optional) Attach the `AWSLambda_ReadOnlyAccess` policy to the user in your organization’s management account if it has Lambda functions you would like to sync.

StackSet template

```
{
  "Resources": {
    "IAMrunZeroRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                "AWS": "arn:aws:iam::<accountID>:user/<userName>"
              }
            }
          ],
          "Version": "2012-10-17"
        },
        "ManagedPolicyArns": [
          {
            "Fn::Join": [
              "",
              [
                "arn:",
                {
                  "Ref": "AWS::Partition"
                },
                ":iam::aws:policy/AmazonEC2ReadOnlyAccess"
              ]
            ]
          },
          {
            "Fn::Join": [
              "",
              [
                "arn:",
                {
                  "Ref": "AWS::Partition"
                },
                ":iam::aws:policy/AmazonRDSReadOnlyAccess"
              ]
            ]
          },
          {
            "Fn::Join": [
              "",
              [
                "arn:",
                {
                  "Ref": "AWS::Partition"
```

```
":iam::aws:policy/AWSLambda_ReadOnlyAccess"
              ]
            ]
          }
        ],
        "RoleName": "<roleName>"
      }
    }
  }
}
```

Inline policy template

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::*:role/<rolename>"
    }
  ]
}
```

Step 2: Add the AWS credential to runZero

1. Go to the Credentials page in runZero and click **Add Credential**.
2. Provide a name for the credential, like `AWS EC2`.
3. Choose **AWS Access & Secret** from the list of credential types.
4. Provide the following information:
    - **AWS access key** - Access key ID obtained from **User > Security credentials > Create access key**.
    - **AWS secret access key** - Secret access key obtained from **User > Security credentials > Create access key**.
    - **AWS role** - Assumed role used to connect to other accounts in your organization. It should be named the same across accounts.
    - Select the region(s) that you want to sync.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
6. Save the credential. You're now ready to set up and activate the connection to bring in data from AWS.

Step 3: Choose how to configure the AWS integration

The AWS integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the AWS integration to sync data

After you add your AWS credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the AWS integration as a connector task

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new AWS-only assets are created.

1. Activate a connection to AWS. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure the credential has access to the organization you are currently in.
3. Enter a name for the task, like `AWS sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. Under **Task configuration**:
    - Choose the site you want to add your assets to. All newly discovered assets will be stored in this site. You can also choose to **Automatically create a new site per VPC** or **Automatically create a new site per account**, and runZero will take care of creating the sites for newly discovered assets or accounts.
    - Choose whether to **automatically delete stale AWS assets**. If you check this option, runZero will automatically delete AWS assets previously seen in AWS that were not found in the most recent sync.
    - Choose whether to include AWS assets that are not currently running. If you check this option, runZero will import AWS asset data for assets that are not in a running state.
6. Under **Service options**, select the services you would like to sync data from. You must choose at least one.
7. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
8. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the AWS integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the AWS toggle to *Yes*, and change any of the default options if needed.
4. On the Credentials tab, set the AWS toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 5: View AWS assets

After a successful sync, you can go to your inventory to view your AWS assets. These assets will have an AWS icon listed in the **Source** column.

To filter by AWS assets, consider running the following queries:

- View all AWS assets:

  ```
  source:aws
  ```

- View all AWS EC2 instances:

  ```
  source:aws AND has:"@aws.ec2.instanceID"
  ```

- View all AWS Elastic Load Balancers:

  ```
  source:aws AND (has:"@aws.elb.loadBalancerArn" OR has:"@aws.elb.loadBalancerName")
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by the AWS APIs.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Amazon Web Services integration unable to connect?

1. Are you getting any data from the AWS integration?
   - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
   - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the AWS integration is unable to connect be sure to check the task log for errors. Some common errors include:
   - 500 - server error, unable to connect to the endpoint
   - 404 - hitting an unknown endpoint on the server
   - 403 - not authorized, likely a credential issue

Enriching scans with EC2

Community Platform

As part of a discovery scan, runZero will automatically enrich scanned assets with data from the AWS EC2 API when available. runZero assets will be updated with internal IP addresses, external IP addresses, hostnames, MAC addresses, and tags, along with other EC2-specific attributes, such as the account ID and instance type.

No additional configuration is needed in runZero to get this data enrichment. However, you may need to modify the permissions associated with the instance's IAM role.

Find Explorers with EC2 enrichment capabilities

To use the EC2 enrichment capabilities, the runZero Explorer must be running on an instance with permissions to describe your EC2 instance list. This can be configured through an IAM role associated with the instance as well as by configuring the AWS credentials for the root user account.

To identify the Explorers with this capability, view your registered Explorers. Any Explorer with the cloud icon indicates that it can enumerate EC2 instances.

Scans run from these Explorers will merge any EC2 instance fields into the asset automatically for any in-scope targets matched against the instance list.

Add permissions to describe instances

To allow for EC2 instance data enrichment, you will need to add the EC2 `ec2:DescribeInstances` permission for the instance role configured for your instance. Visit the Amazon docs to learn how to create and update policies.

Basically, your policy will look like:

```
{
   "Version": "2012-10-17",
   "Statement": [{
     "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
         "ec2:DescribeInstances",
      ],
      "Resource": "*"
   }
   ]
}
```

From the IAM UI, go to **Roles > Permissions > Attach policies** and search for the EC2 service. From the actions, select `DescribeInstances`, which is located under List.

You can also configure credentials on the instance by running `aws configure` as root, instead of using the IAM instance role.

After you save your policy, you'll need to restart your Explorer. The easiest way to do this is to force an update from the Explorer menu.

Keep in mind if your configuration uses one region, but the instance is located in another, it will use the instance's region instead for all API requests.

Attributes runZero gets from the EC2 API

When runZero determines that an IP address is also an EC2 instance, it will enrich the existing runZero data with EC2 metadata. runZero will be able to pull in more hostnames based on AWS asset tags, MAC addresses, internal and external IPs.

Some attributes that runZero is able to get from the EC2 API metadata:

- aws.accountID
- aws.architecture
- aws.availabilityZone
- aws.hypervisor
- aws.imageID
- aws.instanceID
- aws.instanceType
- aws.ipv4
- aws.ipv6
- aws.keyName
- aws.launchTimeTS
- aws.macs
- aws.privateDNS
- aws.privateIP
- aws.publicDNS
- aws.publicIP
- aws.region
- aws.rootDeviceName
- aws.rootDeviceType
- aws.scanner.instanceID
- aws.scanner.instanceType
- aws.state
- aws.subnetID
- aws.tags
- aws.tenancy
- aws.virtualizationType
- aws.vpcID

Additionally, runZero will also report on other things that respond to the scan, but don't match an EC2 entry. You may see things like Amazon RDS, temporary ELBs, and Lambdas in your inventory, but you will only see EC2 metadata on EC2 assets.

Censys Search & Data

Community Platform

runZero supports importing assets from the Censys Search API and the Censys Internet Dataset.

- Importing assets from the Censys Search API
- Importing assets from the Censys Universal Internet Dataset

Censys Search API

To get started with the Censys Search API, you will need to register for a Censys Search account. Once you have done so, you can find your API credentials in the My Account section.

In runZero, go to the Credentials page, and click Add Credential. Select *Censys Search API Key* as the credential type, and enter your API ID and API secret.

You can now go to your asset inventory, click the **Connect** button, and choose **Censys Search API**. Select the credential you just created from the Censys Search credential dropdown.

Configuration

There are two modes for connecting runZero to the Censys Search API.

- **Custom Query** mode - runZero runs a Censys search query you specify, and then imports all of the results into runZero. The search query should be in Censys Search Language. It is a good idea to test your query using the main Censys Search 2.0 interface before running an import task.
- **All Assets** mode - runZero assembles a list of public IP addresses from all of the assets in the selected site, and then uses the API to find Censys Search information for those addresses. The information found is imported into runZero and merged into the appropriate assets.

As with a runZero scan, you'll need to select a site to contain the scan data. The usual task scheduling options are available.

When you have finished editing the Censys Search configuration, click **Activate Connection**.

Censys Universal Internet Dataset

To get started with the Censys Universal Internet Dataset API, you will need a paid Censys Data account and the associated API credential. You can find your API credentials in the My Account section.

The dataset can be downloaded by following the instructions in the Censys documentation. The Search API is used to get a list of files for a given date and those individual files should downloaded into a local directory backed by SSD or NVMe storage.

Creating the database

The raw files are in Apache Avro format and need to be converted into a database for efficient queries.

To process Censys data files, you use the runZero CLI's `censys-db-convert` command. This command takes two parameters:

- The path to a directory containing the `.avro` files from Censys
- The path to write the computed database

```
$ nice runzero censys-db-convert /home/censys/avro /home/censys/db
```

The default configuration requires substantial computing resources:

- At least 8 CPU cores, but 16 or more is better
- At least 64GiB of RAM, but more is better
- At least 3Tb of storage backed by SSD or NVMe (1Tb+2Tb or single volume)

An AWS `m5.4xlarge` with a 3Tb GP2 SSD volume meets these requirements and can process a full dataset (single day) in about 13 hours. The resulting database is about twice the size of the source data (1.3TiB database from 640GiB of Avro). Using the database requires additional disk overhead and over provisioning the storage also improves throughput.

Querying the database

After the Avro files have been converted to a local database, the `censys-db` command can be used to import data into runZero.

The CLI queries the local database, and writes a file in runZero scan format containing the appropriate host records. By default, the file has a name matching `censys-*.rumble.gz` and is written to the current directory. Alternatively you can specify an output filename with the `--output-raw` option, as if performing a runZero scan.

The runZero scan file can be uploaded to the runZero console like any other scan file.

If you have more IP addresses or CIDRs than will fit on a command line, you can use the `--input-targets` option to specify that the CLI should read them from a file. The file is expected to be ASCII text, and contain CIDRs or IP addresses separated by whitespace (which can include newlines).

You can also use the CLI to process data, upload it, and then delete the scan data file if everything succeeded. For example:

```
% runzero censys-db /home/censys/db \
  12.216.190.0/24 --upload --api-key=YOUR_ORGANIZATION_API_KEY \
  --upload-site="Primary site"
```

If you are using self-hosted runZero, you can use the `--api-url` option to specify your console's API endpoint.

The `censys` command also supports the `--verbose` option, which will make it list host addresses as they are written to the output file.

Creating a local Censys Search API server

The computed database can also be used to serve a limited, local version of the Censys Search API using the runZero CLI's `censys-db-server` command.

Due to the size of the database, the system `vm.max_map_count` may need to be increased to avoid a memory map error. The memory map count can be increased by adding the following line to `/etc/sysctl.conf`:

```
vm.max_map_count=262144
```

Once this line is added, reload the sysctl.conf with the following command:

```
$ sudo sysctl -p /etc/sysctl.conf
```

After the `vm.max_map_count` has been updated, start the Censys DB Server with the following command:

```
$ runzero censys-db-server /home/censys/db
```

This will start a local web service on port 55555 by default (changeable via `--port <val>`) that responds to the `/api/v2/hosts/search` and `/api/v2/hosts/<ip>` endpoints. Once this server is running, it can be queried using the runZero Censys Search API connector, and through other HTTP clients, such as curl:

```
$ curl http://127.0.0.1:55555/api/v2/hosts/search?q=ip%3A8.8.8.0/24
$ curl http://127.0.0.1:55555/api/v2/hosts/8.8.8.8
```

Querying the raw Avro files without database processing

runZero also supports direct queries of the unprocessed Avro files. These queries are slow and may take hours or days to complete depending on the query and local storage speed. To query the raw Avro files, you use the runZero CLI's `censys` command. It takes any number of arguments, which can be:

- Names of Avro files, which must end in `.avro`
- CIDRs or IP addresses to search for in the files

The CLI reads the Avro files specified, and writes a file in runZero scan format containing the appropriate host records. By default, the file has a name matching `censys-*.rumble.gz` and is written to the current directory. Alternatively you can specify an output filename with the `--output-raw` option, as if performing a runZero scan.

The runZero scan file can be uploaded to the runZero console like any other scan file.

If you have more IP addresses or CIDRs than will fit on a command line, you can use the `--input-targets` option to specify that the CLI should read them from a file. The file is expected to

be ASCII text, and contain CIDRs or IP addresses separated by whitespace (which can include newlines).

You can also use the CLI to process data, upload it, and then delete the scan data file if everything succeeded. For example:

```
% runzero censys universal-internet-dataset-20210923-000000000000.avro \
  12.216.190.0/24 --upload --api-key=YOUR_ORGANIZATION_API_KEY \
  --upload-site="Primary site"
```

If you are using self-hosted runZero, you can use the `--api-url` option to specify your console's API endpoint.

The `censys` command also supports the `--verbose` option, which will make it list host addresses as they are written to the output file.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Censys Search integration unable to connect?

1. Are you getting any data from the Censys Search integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this integration documentation and follow the steps exactly.
3. If the Censys Search integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

Cisco Meraki Dashboard

Community Platform

runZero integrates with Cisco Meraki Dashboard by importing data from the Cisco Meraki Dashboard API. This integration allows you to sync data about your devices and network clients from Meraki to provide better visibility of your network.

Getting started with Meraki

To set up an integration with Meraki, you'll need to:

1. Generate an API key for your Cisco Meraki Dashboard administrator account.
2. Configure the Meraki credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the integration to pull your data into runZero.

Requirements

Before you can set up the Meraki integration:

- Make sure you have administrator access to the Meraki Dashboard.

Step 1: Generate an API key in Cisco Meraki Dashboard

1. Sign in to Cisco Meraki Dashboard with an administrator account.
2. Check that access to the Cisco Meraki Dashboard API is enabled. This must be enabled for each organization you would like to sync.
    1. Navigate to **Organization > Settings** and enable **API access** under **Dashboard API access**.
3. Navigate to **My profile** to generate an API key. The API key will inherit the same permissions as the account that created it.

Step 2: Add the Meraki API key to runZero

1. Go to the Credentials page in runZero.
2. Choose **Meraki API Key** from the list of credential types.
3. Provide a name for the credential, like `Meraki`.
4. Provide the following information:
    - **Meraki API URL** - The API Endpoint URL used to access the Cisco Meraki Dashboard API.
    - **Meraki API key** - The API key for the Cisco Meraki Dashboard administrator account.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
6. Save the credential.

You're now ready to set up and activate the connection to bring in data from Meraki.

Step 3: Choose how to configure the Meraki integration

The Meraki integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the integration to sync data

After you add your Meraki credential, you'll need to sync your data from Meraki.

Step 4a: Configure the Meraki integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Meraki-only assets are created.

1. Activate a connection to Meraki. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Optionally provide a list of organization IDs or names to include in the import. The list must be comma-separated. We will only import data for the organizations specified.
4. Optionally provide a list of network IDs or names to include in the import. The list must be comma-separated. We will only import data for the networks specified.
5. Optionally provide a comma-separated list of VLANs to exclude from the import. We will not import devices associated with the specified VLANs.
6. If you want to exclude assets that are not associated with a VLAN, set the **Exclude clients with no VLAN** option to *Yes*.
7. Optionally provide a comma-separated list of SSIDs to exclude from the import. We will not import devices connected on the specified SSIDs.
8. Enter a name for the task, like `Meraki Sync` (optional).
9. Choose the Explorer to perform this connector task from (optional).
10. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
11. Enter a description for the task (optional).
12. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
13. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
14. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Meraki integration as a scan probe

You can run the Meraki integration as a scan probe so that the runZero Explorer will pull your Meraki assets into the runZero Console.

In a new or existing scan configuration:

- Ensure that the `MERAKI` option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Set the correct `MERAKI` credential to *Yes* in the *Credentials* tab.

Step 5: View Meraki assets

After a successful sync, you can go to your inventory to view your Meraki assets. These assets will have a Cisco icon listed in the **Source** column.

To filter by Meraki assets, consider running the following queries:

- View all Meraki assets:

  ```
  source:Meraki
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from Meraki.

CrowdStrike Falcon

Community Platform

runZero integrates with CrowdStrike by importing data through the CrowdStrike Falcon API. This integration allows you to sync and enrich your asset inventory, as well as ingesting vulnerability data from Falcon Spotlight and software data from Falcon Discover. Adding your CrowdStrike data to runZero makes it easier to find things like endpoints that are missing an EDR agent.

Getting started

To set up the CrowdStrike integration, you'll need to:

1. Configure CrowdStrike to allow API access through runZero.
2. Add the CrowdStrike credentials, which will include the client ID and client secret, and CrowdStrike base API URL in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the CrowdStrike integration to sync your data with runZero.

Requirements

Before you can set up the CrowdStrike integration:

- Make sure you have access to the CrowdStrike admin portal.

Step 1: Configure CrowdStrike to allow API access to runZero

1. Sign in to CrowdStrike.
2. Go to **Support > API Clients and Keys**. When the **API Key** page appears, choose to add a new API client.
3. Provide the following details for the API client:
   - **Client name**: API client name, such as runZero.

    - **API scope**:
        - To ingest host details, include read permissions for **Hosts** and **Host Groups**.
        - To ingest vulnerability data, include read permissions for **Vulnerabilities**.
        - To ingest software data, include read permissions for **Assets**.
4. When you are done, add the client. An API client created window appears and shows you the client ID and client secret. You'll need them to configure the integration in runZero.
5. Copy the client ID and client secret now. You may not be able to get them later.

Step 2: Add the CrowdStrike credentials to runZero

1. Go to the Credentials page in runZero. Provide a name for the credentials, like `CrowdStrike Falcon`.
2. Choose **CrowdStrike Falcon API key** from the list of credential types.
3. Provide the following information:
    - **CrowdStrike client ID and CrowdStrike client secret** - To generate your client ID and client secret, go to **Support > API Clients and Keys > OAuth2 API clients > Add new API Client** in your CrowdStrike portal.
    - **CrowdStrike API URL** - Your organization-specific base URL, which will depend on your account type. You can find this in the CrowdStrike API Swagger documentation.
        - For a **US-1** account use `api.crowdstrike.com`
        - For a **US-2** account use `api.us-2.crowdstrike.com`
        - For a **US-GOV-1** account use `api.laggar.gcw.crowdstrike.com`
        - For a **EU-1** account use `api.eu-1.crowdstrike.com`
4. If you want other organizations to be able to use these credentials, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
5. Save the credentials. You're now ready to set up and activate the connection to bring in data from CrowdStrike.

Step 3: Choose how to configure the CrowdStrike integration

The CrowdStrike integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the CrowdStrike integration to sync data

After you add your CrowdStrike credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the CrowdStrike integration as a connector task

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new CrowdStrike-only assets are created.

1. Activate a connection to CrowdStrike. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure that it has access to the organization you are currently in.
3. Set the severity and risk levels you want to import (optional).
4. Set the *Fingerprint only* toggle to *Yes* if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).
5. Add an filter for imported assets (optional).

   > If the Crowdstrike API key is configured with access to Falcon Discover or Falcon Spotlight, software and vulnerability data will only be imported for the assets included in the filtered results.

6. Enter a name for the task, like `CrowdStrike sync` (optional).
7. Choose the Explorer to perform this connector task from (optional).
8. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
9. Enter a description for the task (optional).
10. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the *Exclude unknown assets* toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
11. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
12. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the CrowdStrike integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the CrowdStrike toggle to *Yes*, and change any of the default options if needed.
4. On the Credentials tab, set the CrowdStrike toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 5: View CrowdStrike assets and vulnerabilities

After a successful sync, you can go to your inventory to view your CrowdStrike assets and vulnerabilities. These will have a CrowdStrike icon listed in the **Source** column.

To filter by CrowdStrike attributes, consider running the following queries:

- View all CrowdStrike assets

  ```
  source:crowdstrike
  ```

- Find assets that have a CrowdStrike EDR agent installed

  ```
  edr.name:crowdstrike
  ```

- Find Windows assets, excluding servers, that are missing a CrowdStrike EDR agent

  ```
  os:windows and not type:server and not edr.name:CrowdStrike
  ```

- View all CrowdStrike vulnerabilities

  ```
  source:crowdstrike
  ```

- View all CrowdStrike software results

  ```
  source:crowdstrike
  ```

Click into each asset or vulnerability to see its individual attributes. runZero will show you the attributes returned by the CrowdStrike API, with the exception of policies.

Filtering Crowdstrike assets

An optional filter can be applied to Crowdstrike integration tasks. runZero uses Crowdstrike's Falcon Query Language (FQL) for filtering. FQL follows the syntax `<property>:[operator]<value>`. Multiple expressions can be combined for more complex filtering by adding a `+` between expressions. An OR expression can also be leveraged with comma separated expressions.

Properties

The following are some useful CrowdStrike properties that can be used in an FQL expression to filter assets. Details on additional attributes that are available for filtering can be found by reviewing CrowdStrike's API documentation.

CrowdStrike Property	runZero Attribute	Description	Example
`external_ip`	externalIP	The external IP address of the device	18.191.169.203
`first_seen`	firstSeen	A timestamp of when the device was first seen by CrowdStrike	2022-01-08T19:42:34Z
`hostname`	hostname	The hostname of the device	EXPLORER-01
`last_seen`	lastSeen	The timestamp of when the device was last seen by CrowdStrike	2022-09-13T19:14:30Z
`local_ip`	localIP	The local IP address of the device	192.168.1.100

`mac_address`	macAddress	The mac address of the interface communication with CrowdStrike	0a-6e-20-4a-e6-56
`os_version`	osVersion	The operation system version of the device	Ubuntu 20.04
`platform_name`	platformName	The platform running on the device	Linux
`product_type_desc`	productTypeDesc	The type of device	Server

Operators

The following operators can be used in an FQL expression to filter assets.

Operator	Description
!	Not equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
~	Text match. Tokenizes the string, ignoring spaces, case and punctuation
!~	Does not text match. Tokenized the string, ignoring spaces, cases and punctuation
*	Wildcard matching. Matches one or more characters

Example Filters

The following are examples of filters that can be applied to the CrowdStrike sync.

Search Filter	Description
`hostname:'WIN10*'`	Import all devices where the hostname starts with WIN10
`platform_name:'Linux'`	Import all Linux devices
`platform_types_desc:'Server'`	Import all devices that CrowdStrike identifies as a Server
`hostname:'PROD*'+platform_name:'Linux'`	Import all Linux devices with a hostname that starts with PROD
`local_ip:'192.168.1.100'`	Only import the device with a local IP address of 192.168.1.100
`local_ip:!'192.168.1.100'`	Import all devices, excluding 192.168.1.100

	192.168.1.0/24 range
`(local_ip.raw:*'192.168.1.*'), (local_ip.raw:*'192.168.2.*')`	Import all devices with a local IP address in the 192.168.1.0/24 or 192.168.2.0/24 range
`local_ip.raw:!*'192.168.1.*'`	Import all devices, excluding devices with a local IP address in the 192.168.1.0/24 range
`local_ip.raw:!*'192.168.1.*'+local_ip.raw:!*'192.168.2.*'`	Import all devices, exluding devices with a local IP in the 192.168.1.0/24 and 192.168.2.0/24 ranges

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the CrowdStrike integration unable to connect?

1. Are you getting any data from the Crowdstrike integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the CrowdStrike integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. If the integration endpoint is on-premises, verify they are running the integration task from an Explorer with access to the CrowdStrike host.

How can I solve the following CrowdStrike error?

```
Unable to collect software data for CrowdStrike devices: invalid response 403 Forbidden
```

This error occurs if your API client is missing the **Assets** API scope. The integration requires read-only permissions for **Assets** in order to collect software information. Host and vulnerability data should be collected just fine, though. This can be remedied by returning to step 1 of the CrowdStrike documentation above and enabling read permissions for **Assets**.

Custom Integration Scripts

Community Platform

To set up the custom integration script, you will need to:

1. Write the script.
2. Optionally add credentials.
3. Create an integration task.

Step 1: Write integration script

The script can be written in Starlark, a Python-like language with some notable differences:

1. There is no exception handling (`try/catch`)
2. There is no f-string `f'{var}'` formatting - `"{}".format(var)` is the supported method of string interpolation.

Step 1a: Entrypoint

The script needs an entrypoint, a function that gets called by the runZero service and returns the Inventory Assets discovered by the script.

1. The entrypoint must accept a variadic `*args` and `**kwargs` for parameters to be passed in.
2. The entrypoint must return a `list` of `ImportAsset`s to be imported.
3. The entrypoint (function name) of the script will default to `main`, but can be set depending on the integration type scan probe or a connector task.

```
def main(*args, **kwargs):
    asset_one = ImportAsset(
            id=1,
            os='custom import os',
            osVersion='0.0.0.0.0.0.1-pre-alphabetazed',
            manufacturer='Name of manufacturer',
            model='Model of asset',
        )
    asset_two = ImportAsset(
            id=2,
            os='custom import os 2',
            osVersion='0.0.0.0.0.0.2-prezed',
            manufacturer='Name of manufacturer',
            model='model of asset',
        )
    return [asset_one, asset_two]
```

Step 2: Add the custom script Credential to runZero

The credential used for the script is a key-value pair passed into the script function as `kwargs` and can be used as username and password or for anything requiring a secret.

```
def main(*args, **kwargs):
    client_id = kwargs['access_key']
```

```
client_secret = kwargs['access_secret']
```

1. Go to the Credentials page in the runZero console and click **Add Credential**.
2. Choose **Custom Script Secret** from the list of credential types.
3. Provide a name for the credential, like `Script Secret` for the service being integrated with.
4. Provide the following information:
    - **Access Key** - The username or client id that will be passed into the script as a `kwargs` named `access_key`.
    - **Access Secret** - The secret that will be passed into the script as a `kwargs` named `access_secret`.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
6. Save the credential.

Step 3: Create a new task

1. Go to **Tasks**, click the **Integrate** button, and select **Custom Scripts** under **Custom integrations** from the dropdown.
2. Provide a name for the task
3. Select or create a custom integration
4. Select or create credentials for the script
5. Select an explorer
    - Custom integration scripts must be run on a hosted explorer
6. Select Site, Task description, and Schedule as appropriate.
7. Click Activate Connection to start the task.

Using the CLI

The `runzero` CLI includes a `script` sub-command to help in writing and debugging Starlark scripts that use the [[#runZero Types]] and [[#Libraries]] outlined below.

Running scripts

Hello World

```
def main(*args, **kwargs):
        print("Hello world!")
```

Save the file as `print.star` so it can be run from the CLI

```
$ runzero script --filename print.star
Dec 12 12:59:07.099 [INFO] script: Hello world
None
```

Running the script with `args`

```
def main(*args, **kwargs):
        print("Hello {}".format(args[0]))
```

```
$ runzero script --filename print.star --args Dave
Dec 12 13:01:20.707 [INFO] script: Hello Dave
None
```

Running the script with `kwargs`

```
def main(*args, **kwargs):
        print("Hello {}".format(kwargs["name"]))

runzero script --filename print.star --kwargs name=Dave
Dec 12 13:28:56.933 [INFO] script: Hello Dave
None
```

`args` and `kwargs` can be called multiple times to pass in as many arguments as needed

```
def main(*args, **kwargs):
        print("{} {}".format(args[0], args[1]))
        print("{} {}".format(kwargs["access_key"], kwargs["access_secret"]))

runzero script --filename print.star --args Hello --args Dave --kwargs access_key=foo --kwargs access_secret=bar
Dec 12 13:45:50.284 [INFO] script: Hello Dave
Dec 12 13:45:50.284 [INFO] script: foo bar
None
```

REPL

The sub-command `script repl` can be useful for larger scripts with multiple functions. The REPL will allow setting variables, calling functions, printing variables, and anything else that can be done in Starlark.

```
runzero script repl --filename print.star
>>> main(*("hello", "dave"), **{"access_key": "foo", "access_secret": "bar"})
Dec 12 14:53:42.384 [INFO] script: hello dave
Dec 12 14:53:42.384 [INFO] script: foo bar
>>> print("Hello {}".format("dave"))
Dec 12 14:57:28.961 [INFO] script: Hello dave
>>> args = ("hello", "dave")
>>> kwargs = {"access_key": "foo", "access_secret": "bar"}
>>> main(*args, **kwargs)
Dec 12 14:58:57.567 [INFO] script: hello dave
Dec 12 14:58:57.567 [INFO] script: foo bar
>>>
```

`^D` To exit the REPL

runZero Types

Resource types are implemented in Starlark and are equivalent with the Python SDK types:

- `load('runzero.types', 'ImportAsset', 'NetworkInterface', 'Service', 'ServiceProtocolData', 'Software', 'Vulnerability')`

The types can be run from the CLI or loaded into the REPL.

```
runzero script repl --filename print.star
>>> load('json', json_encode='encode', json_decode='decode')
>>> greeter = json_decode('{"greeting":"hello", "name":"dave"}')
>>> print(greeter)
Dec 12 15:13:00.971 [INFO] script: {"greeting": "hello", "name": "dave"}
>>> ^D
```

Libraries

runZero provides some additional libraries for basic functionality to make web requests.

Sample usage can be found on the Starlark library usage examples page.

- requests
  - `load('requests', 'Session', 'Cookie')`
    - `requests`
    - `Session`
    - `Cookie`
- http
  - `load('http', http_post='post', http_get='get', http_patch='patch', http_delete='delete', 'url_encode')`
    - `url_encode`
    - `post`
    - `get`
    - `patch`
    - `delete`
- net
  - `load('net', 'ip_address')`
    - `ip_address`
- json
  - `load('json', json_encode='encode', json_decode='decode')`
    - `encode`
    - `decode`
- time
  - `load('time', 'parse_time')`
    - `parse_time`
- uuid
  - `load('uuid', 'new_uuid')`
    - `new_uuid`
- gzip
  - `load('gzip', gzip_decompress='decompress', gzip_compress='compress')`
    - `gzip_decompress`
    - `gzip_compress`
- base64
  - `load('base64', base64_encode='encode', base64_decode='decode')`
    - `base64_encode`
    - `base64_decode`
-

- crypto
  - `load('crypto', 'sha256', 'sha512', 'sha1', 'md5')`
    - `sha256`
    - `sha512`
    - `sha1`
    - `md5`
- flatten
  - `load('flatten_json', 'flatten')`
    - `flatten`

Starlark library usage examples

Community Platform

This document provides sample usage for common runZero Starlark libraries used in custom integration scripts.

requests

```
load('requests', 'Session', 'Cookie')
load('json', json_decode='decode')

def requests_example():
    session = Session()
    session.headers.set('Accept', 'application/json')
    session.headers.set('User-Agent', 'Mozilla/5.0')
    session.headers.set('User-Agent', None)  # remove header

    url = 'https://hacker-news.firebaseio.com/v0/topstories.json'
    session.cookies.set(url, {"test_cookie": "cookie_value"})

    response = session.get(url)
    if response and response.status_code == 200:
        data = json_decode(response.body)
        print("Top story IDs:", data[:5])
    else:
        print("Failed to fetch stories")
```

http

```
load('http', http_post='post', http_get='get', 'url_encode')

def fetch_example():
    url = "https://hacker-news.firebaseio.com/v0/topstories.json"
    headers = {"Accept": "application/json"}

    response = http_get(url, headers=headers)

    if response and response.status_code == 200:
```

```
else:
        print("Request failed with status:", response.status_code)
```

net

```
load('net', 'ip_address')

def parse_ip_list():
    ips = ["192.168.1.1", "2607:f8b0:4005:805::200e"]
    for ip in ips:
        addr = ip_address(ip)
        print("IP:", addr, "Version:", addr.version)
```

json

```
load('json', json_encode='encode', json_decode='decode')

def test_json_handling():
    data = {"name": "runZero", "features": ["scan", "API", "integrations"]}

    encoded = json_encode(data)
    print("Encoded JSON:", encoded)

    decoded = json_decode(encoded)
    print("Decoded:", decoded["name"])
```

time

```
load('time', 'parse_time')

def parse_example_time():
    time_str = "2025-05-01T15:00:00Z"
    parsed = parse_time(time_str)
    print("Parsed time:", parsed)
```

uuid

```
load('uuid', 'new_uuid')

def create_unique_id():
    uid = new_uuid()
    print("Generated UUID:", uid)
```

gzip

```
load('gzip', gzip_decompress='decompress', gzip_compress='compress')
```

```
original = "Hello, runZero!".encode("utf-8")

    # Compress the data
    compressed = gzip_compress(original)
    print("Compressed length:", len(compressed))

    # Decompress it back
    decompressed = gzip_decompress(compressed)
    print("Decompressed value:", decompressed.decode("utf-8"))
```

base64

```
load('base64', base64_encode='encode', base64_decode='decode')

def b64_example():
    username = "xxx"
    password = "yyy"
    enc = base64_encode(username + ":" + password)
    dec = base64_decode(enc)
    return (enc, dec)
```

crypto

```
load('crypto', 'sha256', 'sha512', 'sha1', 'md5')

def main(*args, **kwargs):
    access_key = kwargs['access_key']
    access_secret = kwargs['access_secret']

    input = "test"

    sha256_hash = sha256(input)
    if str(sha256_hash) != '9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08':
        print("sha256_hash [fail]: {}".format(sha256_hash))
    else:
        print("sha256_hash [pass]: {}".format(sha256_hash))

    sha512_hash = sha512(input)
    if str(sha512_hash) != 'ee26b0dd4af7e749aa1a8ee3c10ae9923f618980772e473f8819a5d4940e0db27ac185f8a0e1d5f84f88bc887fc
        print("sha512_hash [fail]: {}".format(sha512_hash))
    else:
        print("sha512_hash [pass]: {}".format(sha512_hash))

    sha1_hash = sha1(input)
    if str(sha1_hash) != 'a94a8fe5ccb19ba61c4c0873d391e987982fbbd3':
        print("sha1_hash [fail]: {}".format(sha1_hash))
    else:
        print("sha1_hash [pass]: {}".format(sha1_hash))
```

```
if str(md5_hash) != '098f6bcd4621d373cade4e832627b4f6':
        print("md5_hash [fail]: {}".format(md5_hash))
    else:
        print("md5_hash [pass]: {}".format(md5_hash))

    return True
```

flatten (json)

```
load('flatten_json', 'flatten')

def main(*args, **kwargs):
    nested_json = {"foo": "bar", "a": {"b": "c"}}
    flattened_json = flatten(nested_json)
    return True
```

Google Cloud Platform

Community Platform

The Google Cloud Platform (GCP) integration provides visibility into your cloud assets by synchronizing your GCP cloud inventories with runZero. runZero also integrates with other cloud providers, such as Microsoft Azure and Amazon AWS. Similarly to other integrations, you will need to add the Scanning with credentials needed to authenticate to GCP and set up a connector in runZero. runZero will pull in GCP compute instance VMs, pulling in GCP attributes that will be viewable from each asset.

The following GCP asset types are supported:

- Compute Engine instances
- Load balancers
- Cloud SQL

Requirements

- Verify you have a Google Cloud service account with the `Compute Network Viewer` and `Cloud SQL Viewer` roles.
    - This service account will need to be granted access to each project that you want the integration to gather data from.
- Download a key for the GCP service account.
- Verify you have these GCP APIs enabled on each project:
    - Compute Engine
    - Cloud SQL Admin

How to set up the Google Cloud Platform integration

Here are the high-level steps to set up the Google Cloud Platform integration:

- Create a Google Cloud Platform credential in runZero.
- Choose whether to configure the integration as a scan probe or connector task.
- Activate the integration for Google Cloud Platform.
- View your GCP assets.

Step 1: Create Google Cloud Platform credentials

1. Go to the Credentials page and click **Add Credential**.
2. From the **Credentials type** dropdown, choose **GCP Service Account Key**.
3. Provide a name for the credential, like `GCP`.
4. Set the **Include all projects** toggle to *Yes* if you want runZero to gather asset data from all GCP projects that the service account has access to. If set to *No*, the integration will only gather asset data from the project specified in the key file.
5. Click **Choose file** to upload the service account key file you downloaded from GCP.
6. If you want other organizations to be able to use this credential, select the **Make this a global credential** option. Otherwise, you can configure access on a per-organization basis.
7. Save the credential. You're now ready to set up and activate the connection to bring in data from Google Cloud Platform.

Step 2: Choose how to configure the GCP integration

The GCP integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 3: Activate the Google Cloud Platform integration

After you add your GCP credential, you'll need to set up a connector task or scan probe to sync your data.

Step 3a: Configure the GCP integration as a connector task

1. Activate a connection to GCP. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure the credential has access to the organization you are currently in.
3. Enter a name for the task, like `Google Cloud Platform sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. To organize your assets logically, choose the site you'd like to use to add your assets to. You can choose an existing site or add them to a new site when the sync occurs. Assigning your assets to a site helps organize and group your assets. You can automatically generate a new site per GCP project by selecting this option from the task configuration.

6. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
7. Activate the connection when you are done. The sync will run on the defined schedule. You can check the Scheduled tasks to see when the next sync will occur.

Step 3b: Configure the GCP integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the GCP toggle to *Yes*, and change any of the default options if needed.
4. On the Credentials tab, set the GCP toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 4: View your Google Cloud Platform assets

After a successful sync, you can go to your inventory to view your GCP assets. These assets will have a Google icon listed in the **Source** column.

To filter by GCP assets, consider running the following queries:

- View all GCP assets:

```
source:gcp
```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by GCP.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Google Cloud Platform integration unable to connect?

1. Are you getting any data from the GCP integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the GCP integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server

- 403 - not authorized, likely a credential issue

Google Workspace

Community Platform

runZero integrates with Google Workspace to allow you to sync and enrich your asset inventory, as well as gain visibility into users and groups. Adding your Google Workspace data to runZero makes it easier to find unmanaged assets on your network. The Google Workspace integration supports ChromeOS, Mobile, and Endpoint registered asset types.

Requirements

- Verify or create a new Google service account in whichever project is most suitable.
- Create and download a key for the Google service account. Save this JSON file.
- Verify that you have the `Admin SDK` and `Cloud Identity` APIs enabled for the project. Use the search box in the API Library to find each API and then enable it.
- Enable domain-wide delegation in the Google Workspace console
  - Add a new API client using the unique numeric ID of service account as the Client ID
  - Enable the following OAuth scopes for this API client:

    ```
    https://www.googleapis.com/auth/admin.directory.user.readonly,
    https://www.googleapis.com/auth/admin.directory.group.readonly,
    https://www.googleapis.com/auth/admin.directory.device.mobile.readonly,
    https://www.googleapis.com/auth/admin.directory.device.chromeos.readonly,
    https://www.googleapis.com/auth/cloud-identity.devices.readonly
    ```

  - Optionally, enter each OAuth scope individually:
  - `https://www.googleapis.com/auth/admin.directory.user.readonly`
  - `https://www.googleapis.com/auth/admin.directory.group.readonly`
  - `https://www.googleapis.com/auth/admin.directory.device.mobile.readonly`
  - `https://www.googleapis.com/auth/admin.directory.device.chromeos.readonly`
  - `https://www.googleapis.com/auth/cloud-identity.devices.readonly`

How to set up the Google Workspace integration

These are the high-level steps to set up the Google Cloud Platform integration:

- Create a Google Workspace credential in runZero.
- Choose whether to configure the integration as a scan probe or connector task.
- Activate the connection for Google Workspace.
- View your results.

Step 1: Create a Google Workspace credential

1. Go to the Credentials page and click **Add Credential**.
2. From the **Credentials type** dropdown, choose **Google Workspace Client Secret**.

3. Provide a name for the credential, like `Google Workspace`.
4. In the *Admin account email* field, provide the email address of an administrator account with access to the assets, users, or groups you wish to import.
5. If you want to import from an organization other than the one your administrator account belongs to, provide a *Customer ID*. By default, runZero will use the Customer ID associated with the service account. (Optional)
6. Click **Choose file** to upload the service account key file you downloaded from Google Workspace.
7. If you want other organizations to be able to use this credential, select the **Make this a global credential** option. Otherwise, you can configure access on a per-organization basis.
8. Save the credential. You're now ready to set up and activate the connection to bring in data from Google Workspace.

Step 2: Choose how to configure the Google Workspace integration

The Google Workspace integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync. Configuring the integration as a scan probe is useful if you are running self-hosted runZero Platform and your console cannot access Google Workspace. For most situations it will be easier to set up a scheduled connection to sync your data from Google Workspace.

Step 3: Activate the Google Workspace integration

After you add your GCP credential, you'll need to set up a connector task or scan probe to sync your data.

Step 3a: Configure the Google Workspace integration as a connector task

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where the data is organized.

1. Activate a connection to Google Workspace. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure the credential has access to the organization you are currently in.
3. Enter a name for the task, like `Google Workspace sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. To organize your assets logically, choose the site you'd like to use to add your assets to. You can choose an existing site or add them to a new site when the sync occurs. Assigning your assets to a site helps organize and group your assets.
6. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.

7. Activate the connection when you are done. The sync will run on the defined schedule. You can check the Scheduled tasks to see when the next sync will occur.

Step 3b: Configure the Google Workspace integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the GoogleWorkspace toggle to *Yes*, and change any of the default options if needed.
4. On the Credentials tab, set the GoogleWorkspace toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 4: View Google Workspace assets

After a successful sync, you can go to your inventory to view your Google Workspace assets. These assets will have a Google Workspace icon listed in the **Source** column.

To filter by Google Workspace assets, consider running the following queries:

- View all Google Workspace assets:

  ```
  source:googleworkspace
  ```

- View runZero assets not connected to Google Workspace:

  ```
  source:runzero AND NOT source:googleworkspace
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by Google Workspace.

Community Platform

> The Google Workspace integration provides details about users and groups in addition to enriching asset inventory data. Go to Inventory > Users or Inventory > Groups to view the data provided by Google Workspace. Use the query `source:googleworkspace` to filter your results.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Google Workspace integration unable to connect?

1.

1. Are you getting any data from the Google Workspace integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Google Workspace integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

Microsoft 365 Defender

Community Platform

runZero integrates with Microsoft 365 Defender to allow you to sync and enrich your asset, software, and vulnerability inventory. Adding your Microsoft 365 Defender data to runZero makes it easier to find assets missing EDR protection.

Getting started

To set up the Microsoft 365 Defender integration, you'll need to:

1. Configure Microsoft 365 Defender to allow API access through runZero.
2. Add the Microsoft 365 Defender credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the Microsoft 365 Defender integration to sync your data with runZero.

Requirements

Before you can set up the Microsoft 365 Defender integration:

- Make sure you have access to the Microsoft Azure portal.

Step 1: Register an Azure application for Microsoft 365 Defender API access

runZero can authenticate to the Microsoft 365 Defender API using a client secret. Register an application to configure Microsoft 365 Defender API access.

1. Sign in to the Microsoft Azure portal.
2. Go to **Azure Active Directory > App registrations** and click on **New registration**.
    - Provide a name.
    - Select the supported account types.

    - Optionally add a redirect URI.
3. Click **Register** to register the application.
4. Once the application is created, you should see the **Overview** dashboard. Note the following information:
    - Application (client) ID
    - Directory (tenant) ID
5. From the application's details page, go to *API permissions > Add a permission*.
6. Select the second tab called **APIs my organization uses** to view available APIs.
7. Select `WindowsDefenderATP` from the list of Microsoft APIs.
8. Select the permissions type `Application permissions` to configure a **client secret**.
9. Search for and select the following required permissions:
    - Windows Defender ATP API permissions:
        - `Machine.Read.All`
        - `Software.Read.All`
        - `Vulnerability.Read.All`
10. Click **Add permissions** to save the permissions to the application.
11. Navigate to **Azure Active Directory > App registrations** and select the application you created.
12. Go to **Certificates & secrets** and click on **New client secret**.
    - Enter a description.
    - Select the expiration.
13. Click **Add** to create the client secret and save the client secret value.

Step 2: Add an Azure Client Secret credential to runZero

This type of credential can be used to sync all resources in a single directory (across multiple subscriptions).

1. Go to the Credentials page in runZero and click **Add Credential**.
2. Provide a name for the credential, like `Azure Client Secret`.
3. Choose **Azure Client Secret** from the list of credential types.
4. Provide the following information:
    - **Azure application (client) ID** - The unique ID for the registered application. This can be found in the Azure portal if you go to **Azure Active Directory > App registrations** and select the application.
    - **Azure client secret** - To generate a client secret, go to **Azure Active Directory > App registrations**, select your application, go to **Certificates & secrets** and click on **New client secret**.
    - **Azure directory (tenant) ID** - The unique ID for the tenant. This can be found in the Azure portal if you go to **Azure Active Directory > App registrations** and select the application.
    - Select the **Access all subscriptions in this directory (tenant)** option to sync all resources in your directory. Otherwise, specify the **Azure subscription ID** - The unique ID for the subscription that you want to sync. This can be found in the Azure portal if you go to **Subscriptions** and select the subscription.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
6. Save the credential. You're now ready to set up and activate the connection to bring in data from Azure.

Step 3: Choose how to configure the Microsoft 365 Defender integration

The Microsoft 365 Defender integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the Microsoft 365 Defender integration to sync data

After you add your Microsoft 365 Defender credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the Microsoft 365 Defender integration as a connector task

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Microsoft 365 Defender-only assets are created.

1. Activate a connection to Microsoft 365 Defender. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure it has access to the organization you are currently in.
3. Enter a name for the task, like `Microsoft 365 Defender sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. Under **Task configuration**, choose the site you want to add your assets to.
6. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
7. If you want to include assets that have stopped reporting to the Microsoft 365 Defender service in your integration import, switch the **Include inactive assets** toggle to *Yes*. By default, the integration will not include assets that are marked as inactive.
8. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Microsoft 365 Defender integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the Defender365 toggle to *Yes*, and change any of the default options if needed.

4. On the Credentials tab, set the MS365Defender toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 5: View Microsoft 365 Defender assets

After a successful sync, you can go to your inventory to view your Microsoft 365 Defender assets. These assets will have an Active Directory icon listed in the **Source** column.

To filter by Microsoft 365 Defender assets, consider running the following queries:

- View all Microsoft 365 Defender assets:

  ```
  source:ms365defender
  ```

- View runZero assets not connected to Microsoft 365 Defender:

  ```
  source:runzero AND NOT source:ms365defender
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by Microsoft 365 Defender.

Filtering Microsoft 365 Defender assets

An optional filter can be applied to Microsoft 365 Defender integration tasks. runZero uses Microsoft Graph $filter query paramater to filter assets. GraphQL follows the syntax `<property> [operator] <value>`. Multiple expressions can be combined for more complex filtering by adding an `and` or `or` between expressions.

Properties

Any property that runZero imports from Microsoft 365 Defender can be used to apply a filter. The following are some examples.

Defender Property	runZero Attribute	Description	Example
`computerDNSName`	@ms365defender.dev.computerDNSname	The hostname of the device	EXPLORER-01
`osPlatform`	@ms365defender.dev.osPlatform	The operation system of the device	Windows11, Android
`heatlhStatus`	@ms365defender.dev.heatlhStatus	The health status of the Defender agent that is installed	Active, Inactive

		specifying whether the device is registered in Azure AD	true, false
`managedBy`	@ms365defender.dev.managedBy	Description of how the device is managed	Intune

Operators

The following are common operators that can be used in a Microsoft 365 Defender filter.

- Equal to (`eq`)
- Not equal to (`ne`)
- Has (`has`)
- Less than (`lt`)
- Greather than (`gt`)
- Less than or equal to (`le`)
- Greater than or equal to (`ge`)

The following are common functions that can be used in a Microsoft 365 Defender filter. Functions follow the syntax `function(<property>, <value>)`.

- Starts with (`startswith`)
- Ends with (`endswith`)
- Contains (`contains`)

Example Filters

The following are examples of filters that can be applied to a Microsoft 365 Defender integration.

Search Filter	Description
`not(osPlatform eq 'Android')`	Import all assets except those with an Android operating system
`not(osPlatform eq 'Android') and not(osPlatform eq 'iOS')`	Import all assets execpt those with an Android or iOS operating system
`startswith(computerDNSname, 'PROD')`	Import all devices with a hostname that starts with PROD
`not(startswith(computerDNSname, 'DEV'))`	Import all devices except those with a hostname that starts with DEV

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Microsoft 365 Defender integration unable to connect?

1. Are you getting any data from the Microsoft 365 Defender integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Microsoft 365 Defender integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

Microsoft Active Directory

Community Platform

runZero integrates with Microsoft Active Directory (AD) via LDAP to allow you to sync and enrich your asset inventory, as well as gain visibility into domain users and groups. Adding your AD data to runZero makes it easier to find assets that are not part of your domain.

Getting started

To set up the Active Directory integration, you'll need to:

1. Add the AD credential in runZero.
2. Choose whether to configure the integration as a scan probe or connector task.
3. Activate the Active Directory integration to sync your data with runZero.

Requirements

Before you can set up the LDAP integration, make sure you have credentials for an LDAP account.

Step 1: Add the LDAP credential to runZero

1. Go to the Add credential page in runZero. Provide a name for the credentials, like `LDAP`.
2. Choose **LDAP Username & Password** from the list of credential types.
3. Provide the following information:
    - **LDAP username** - The username you want to use with the LDAP integration. The account used for this integration does not require any special permissions. The following username formats are accepted:
        - Distinguished Name (DN): `CN=[username],CN=Users,DC=[domain],DC=[tld]`

        - User Principle Name (UPN): `[username]@[domain].[tld]`
        - Domain\Username: `[domain]\[username]`
    - **LDAP password** - The password for the username to be used with the LDAP integration.
    - **LDAP base DN** - The base distinguished name for LDAP searches. This field requires distinguished name format: `DC=[domain],DC=[tld]`. Note that only entities underneath the specified base DN will be imported into runZero.
    - **LDAP URL** - The URL for your LDAP server. This field supports `IP[:port]` notation as well as `hostname.domain.tld[:port]`. This field requires that the URL entered begins with `ldap://` (for insecure LDAP connections) or `ldaps://` (for secure LDAP connections). For example: `ldaps://ad.example.com:636`
    - **LDAP insecure** - Set this to *Yes* if you want to attempt authentication without a verified thumbprint. By default, runZero will attempt to connect with LDAPS but will fall back to LDAP+StartTLS then LDAP. LDAP without StartTLS will only work if this toggle is set to *Yes*.
    - **LDAP thumbprints** (optional) - A set of `IP[:port]=SHA256:B64HASH` or `hostname.domain.tld=SHA256:B64HASH` pairs to trust for authentication.
        - You will need to scan your LDAP server with runZero in order to obtain the TLS thumbprint. The TLS fingerprints service attribute report lists all previously seen fingerprints. The TLS thumbprints used for self-signed certificates will only work with LDAPS. If you want to use LDAP+StartTLS with a self-signed certificate, you will need to set the **Insecure** option to `Yes`.
        - If *LDAP insecure* is set to *No* and no thumbprints are provided:
            - With a self-signed certificate, the connection will fail because the certificate chain cannot be verified.
            - With a valid certificate from a public CA, the connection will work without thumbprints.
4. If you want all other organizations to be able to use this credential, select the **Make this a global credential** option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential. You're now ready to set up and activate the connection to bring in data from LDAP.

Step 2: Choose how to configure the Active Directory integration

The Active Directory integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 3: Set up and activate the Active Directory integration to sync data

After you add your Active Directory credential, you'll need to set up a connector task or scan probe to sync your data.

Step 3a: Configure the Active Directory integration as a connector task

After you add your LDAP credential, you'll need to set up a connection to sync your data from LDAP. A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new LDAP-only assets are created.

1. Activate a connection to Active Directory. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Enter a name for the task, like `LDAP sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. Under **Task configuration**, choose the site you want to add your assets to.
6. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
7. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 3b: Configure the Active Directory integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the LDAP toggle to *Yes*, and change any of the default options if needed.
4. On the Credentials tab, set the LDAP toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 4: View Active Directory assets

After a successful sync, you can go to your inventory to view your LDAP assets. These assets will have an Active Directory icon listed in the **Source** column.

To filter by LDAP assets, consider running the following queries:

- View all LDAP assets:

  ```
  source:ldap
  ```

- View runZero assets not connected to LDAP:

  ```
  source:runzero AND NOT source:ldap
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by LDAP.

Community Platform

The LDAP integration provides details about users and groups in addition to enriching asset inventory data. Go to Inventory > Users or Inventory > Groups to view the data provided by LDAP.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Microsoft Active Directory integration unable to connect?

1. Are you getting any data from the Microsoft Active Directory integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Microsoft Active Directory integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the Microsoft Active Directory host.

Microsoft Azure

Community Platform

runZero integrates with Microsoft Azure to deliver greater visibility into your cloud assets. This integration imports data through each applicable API to enrich your asset inventory:

- Virtual Machines API
- Virtual Machine Scale Sets API
- Load Balancers API
- AzureSQL API
- Web Apps API

Syncing with Azure allows you to view information about your asset's OS profile, storage profile, and more. This integration imports assets that are in a running state.

Getting started

The following Azure resource types are supported:

- Virtual Machines
- Virtual Machine Scale Sets
- Azure SQL
- Azure Cosmos DB
- Load Balancers
- Function Apps

To set up the Azure integration, you'll need to:

1. Configure Azure to allow API access through runZero.
2. Add an Azure credential to runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the Azure integration to sync your data with runZero.

Requirements

Before you can set up the Azure integration, make sure you have access to the Microsoft Azure portal.

Step 1: Configure Azure to allow API access through runZero

1. Sign in to the Microsoft Azure portal.
2. Go to **Azure Active Directory > App registrations** and click on **New registration**.
    - Provide a name.
    - Select the supported account types.
    - Optionally add a redirect URI.
3. Click **Register** to register the application.
4. Once the application is created, you should see the **Overview** dashboard. Save the following information:
    - Application (client) ID
    - Directory (tenant) ID
5. Give the client access to the subscriptions you want to sync. From the subscription details page, go to **Access Control (IAM)** and select **Add > Add role assignment**. Enter the following:
    - Role: Reader
    - Assign access to: User, group, or service principal
    - Under **Select**, search for the name of the application you created. Click on your application to add it to the **Selected members** list below.
6. Click **Save** to save the role assignment.
7. Navigate to **Azure Active Directory > App registrations** and select the application you created.
8. Go to **Certificates & secrets** and click on **New client secret**.
    - Enter a description.

    - Select the expiration.
9. Click **Add** to create the client secret. Save the following information:
    - Client secret value

Step 2: Add the Azure credential to runZero

The credential used for the Azure integration can be either a client secret or a username & password.

Step 2a: Add an Azure Client Secret credential to runZero

This type of credential can be used to sync all resources in a single directory (across multiple subscriptions).

1. Go to the Credentials page in runZero and click **Add Credential**.
2. Provide a name for the credential, like `Azure Client Secret`.
3. Choose **Azure Client Secret** from the list of credential types.
4. Provide the following information:
    - **Azure application (client) ID** - The unique ID for the registered application. This can be found in the Azure portal if you go to **Azure Active Directory > App registrations** and select the application.
    - **Azure client secret** - To generate a client secret, go to **Azure Active Directory > App registrations**, select your application, go to **Certificates & secrets** and click on **New client secret**.
    - **Azure directory (tenant) ID** - The unique ID for the tenant. This can be found in the Azure portal if you go to **Azure Active Directory > App registrations** and select the application.
    - Select the **Access all subscriptions in this directory (tenant)** option to sync all resources in your directory. Otherwise, specify the **Azure subscription ID** - The unique ID for the subscription that you want to sync. This can be found in the Azure portal if you go to **Subscriptions** and select the subscription.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
6. Save the credential. You're now ready to set up and activate the connection to bring in data from Azure.

Step 2b: Add an Azure Username & Password credential to runZero

This type of credential can be used to sync all resources across directories. Alternatively, you can add one **Azure Client Secret** credential for each Azure directory you want to sync.

1. Go to the Credentials page in runZero and click **Add Credential**.
2. Provide a name for the credential, like `Azure User/Pass`.
3. Choose **Azure Username & Password** from the list of credential types.
4. Provide the following information:
    - **Azure application (client) ID** - The unique ID for the registered application. This can be found in the Azure portal if you go to **Azure Active Directory > App registrations** and select the application.

- **Azure directory (tenant) ID** - The unique ID for the tenant. This can be found in the Azure portal if you go to **Azure Active Directory > App registrations** and select the application.
- **Azure username** - The username for your Azure cloud account. This cannot be a federated user account.
- **Azure password** - The password for your Azure cloud account.

5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
6. Save the credential. You're now ready to set up and activate the connection to bring in data from Azure.

Step 3: Choose how to configure the Azure integration

The Azure integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the Azure integration to sync data

After you add your Azure credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the Azure integration as a connector task

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Azure-only assets are created.

1. Activate a connection to Azure. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure the credential has access to the organization you are currently in.
3. Enter a name for the task, like `Azure sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. Under **Task configuration**, choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
6. Under **Service options**, select the services you would like to sync data from. You must choose at least one.
7. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
8. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Azure integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.

2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the Azure toggle to *Yes*, and change any of the default options if needed.
4. On the Credentials tab, set the Azure toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 5: View Azure assets

After a successful sync, you can go to your inventory to view your Azure assets. These assets will have an Azure icon listed in the **Source** column.

To filter Azure assets, consider running the following queries:

- View all Azure assets:

  ```
  source:azure
  ```

- View all Azure VMs

  ```
  has:"@azure.vm.vmID"
  ```

- View all Azure virtual machine scale set VMs

  ```
  has:"@azure.vmss.vmID"
  ```

- View all Azure load balancers

  ```
  has:"@azure.lb.id"
  ```

- View all AzureSQL instances

  ```
  has:"@azure.azsql.id"
  ```

- View all Azure Cosmos DB instances

  ```
  has:"@azure.cosmos.id""
  ```

- View all Azure Function Apps

  ```
  @azure.functionapp.kind:"functionapp"
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by the Azure APIs.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Microsoft Azure integration unable to connect?

1. Are you getting any data from the Microsoft Azure integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Microsoft Azure integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

Microsoft Endpoint Configuration Manager (MECM)

Community Platform

runZero integrates with Microsoft Endpoint Configuration Manager (MECM), formerly System Center Configuration Manager (SCCM), by importing data from the MECM MSSQL database. This integration allows you to sync data about your devices from MECM, making it easier to find unmanaged devices in your network.

Getting started with MECM

To set up an integration with MECM, you'll need to:

1. Identify or create a database user with read access to the MECM database.
2. Configure the MECM credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the integration to pull your data into runZero.

Step 1: Identify or create a database user for access to MECM

1. Identify an existing database user with read access to the database.
2. Alternatively, create a dedicated read-only database user for this integration. More details on creating a new database user can be found in Microsoft's documentation - Create a database user.

Step 2: Add the MECM database connection string to runZero

1. Go to the Credentials page in runZero.
2. Choose **MECM Database Connection String** from the list of credential types.
3. Provide a name for the credential, like `MECM`.

4. Provide the database connection string, using one of the following formats:
    - `Server=host,port;Database=database-name;User Id=user-id;Password=password;`
      When using this format, the values should not contain a semicolon (;). You can use single or double quotes to escape any special characters.
    - `sqlserver://username:password@host/instance?database=value&param=value`
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
6. Save the credential.

You're now ready to set up and activate the connection to bring in data from MECM.

Step 3: Choose how to configure the MECM integration

The MECM integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the integration to sync data

After you add your MECM credential, you'll need to sync your data from MECM.

Step 4a: Configure the MECM integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new assets are created.

1. Activate a connection to MECM. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Enter a name for the task, like `MECM Sync` (optional).
4. Choose the Explorer to perform this connector task from (optional).
5. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
6. Enter a description for the task (optional).
7. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
8. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
9. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the MECM integration as a scan probe

You can run the MECM integration as a scan probe so that the runZero Explorer will pull your MECM devices into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *MECM* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Set the correct *MECM* credential to *Yes* in the *Credentials* tab.

Step 5: View MECM assets

After a successful sync, you can go to your inventory to view your MECM assets. These assets will have a Microsoft icon listed in the **Source** column.

To filter by MECM assets, consider running the following queries:

- View all MECM assets:

  ```
  source:mecm
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from MECM.

Microsoft Entra ID

Community Platform

runZero integrates with Microsoft Entra ID (formerly Azure AD) to allow you to sync and enrich your asset inventory, as well as gain visibility into Entra ID users and groups. Adding your Entra ID data to runZero makes it easier to find assets that are not part of your domain.

Note that Entra ID is still referred to as Azure AD within the runZero product.

Getting started

To set up the Entra ID integration, you'll need to:

1. Configure Entra ID to allow API access through runZero.
2. Add the Entra ID credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the Entra ID integration to sync your data with runZero.

Requirements

Before you can set up the Entra ID integration, make sure you have access to the Microsoft Azure portal.

Step 1: Register an Azure application for Entra ID API access

runZero can authenticate to the Entra ID API using either a username and password or a client secret. Register an application to configure Entra ID API access.

1. Sign in to the Microsoft Azure portal.
2. Go to *App registrations* and click *+ New registration*.
    - Provide a name.
    - Select the supported account types.
    - Optionally add a redirect URI.
3. Click *Register* to register the application.
4. Once the application is created, go back to the main Azure portal page and select *App registrations*. You should be able to find the application you just registered. (It may help to select the *Owned applications* tab.) Click on the app's name
5. You should see the an overview of the app registration. Note the following information:
    - Application (client) ID
    - Directory (tenant) ID
6. Click on the display name of the application to go to its management pages.
7. Select the *Manage > Authentication* section on the left. Set **Allow public client flows** to *Yes* and then save the configuration.
8. Go to *API permissions* and click *+ Add a permission*.
9. Select *Microsoft Graph* from the list of Microsoft APIs.
10. Select the correct permissions type for your needs:
    - **Username & password**: select *Delegated permissions*
    - **Client secret**: select *Application permissions*
11. Search for and select the following required permissions:
    - *Device.Read.All*
    - *Group.Read.All*
    - *User.Read.All*
12. Click *Add permissions* to save the permissions to the application.
13. If using a client secret, also perform the following steps:
    - Navigate to *Azure Active Directory > App registrations* and select the application you created.
    - Go to *Certificates & secrets* and click on *New client secret*.
        - Enter a description.

        - Select the expiration.
    - Click *Add* to create the client secret and save the client secret value.

Add the Entra ID credential to runZero

Step 2a: Add an Azure Username & Password credential to runZero

1. Go to the Credentials page in runZero and click *Add Credential*.
2. Provide a name for the credential, like `Azure Username & Password`.
3. Choose *Azure Username & Password* from the list of credential types.
4. Provide the following information:
    - **Azure application (client) ID** - The unique ID for the registered application. This can be found in the Azure portal if you go to *App registrations* and select the application.
    - **Azure directory (tenant) ID** - The unique ID for the tenant. This can be found in the Azure portal if you go to *App registrations* and select the application.
    - **Azure username** - The username for your Azure cloud account. This cannot be a federated user account.
    - **Azure password** - The password for your Azure cloud account.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
6. Save the credential. You're now ready to set up and activate the connection to bring in data from Azure.

Step 2b: Add an Azure Client Secret credential to runZero

This type of credential can be used to sync all resources in a single directory (across multiple subscriptions).

1. Go to the Credentials page in runZero and click *Add Credential*.
2. Provide a name for the credential, like `Azure Client Secret`.
3. Choose *Azure Client Secret* from the list of credential types.
4. Provide the following information:
    - **Azure application (client) ID** - The unique ID for the registered application. This can be found in the Azure portal if you go to *App registrations* and select the application.
    - **Azure client secret** - To generate a client secret, go to *App registrations*, select your application, go to *Certificates & secrets* and click on *+ New client secret*.
    - **Azure directory (tenant) ID** - The unique ID for the tenant. This can be found in the Azure portal if you go to *App registrations* and select the application.

- Select the **Access all subscriptions in this directory (tenant)** option to sync all resources in your directory. Otherwise, specify the Azure subscription ID - The unique ID for the subscription that you want to sync. This can be found in the Azure portal if you go to *Subscriptions* and select the subscription.

5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.

6. Save the credential. You're now ready to set up and activate the connection to bring in data from Azure.

Step 3: Choose how to configure the Entra ID integration

The Entra ID integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the Entra ID integration to sync data

After you add your Entra ID credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the Entra ID integration as a connector task

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Entra ID-only assets are created.

1. Activate a connection to Entra ID. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.

2. Choose the credential you added earlier. If you don't see the credential listed, make sure it has access to the organization you are currently in.

3. Optionally provide a filter following the Microsoft Graph API filter syntax. We will only import devices that match the filter.

4. Enter a name for the task, like `Entra ID sync`.

5. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.

6. Under **Task configuration**, choose the site you want to add your assets to.

7. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.

8. If you want to include assets in your integration import that the Entra ID account has marked as inactive, switch the **Include inactive assets** toggle to *Yes*. By default, the integration will not include assets that are marked as inactive.

9. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Entra ID integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.

2. Add or update the scan parameters based on any additional requirements.

3. On the Probes and SNMP tab, choose which additional probes to include, set the **Azure AD** toggle to *Yes*, and change any of the default options if needed.

4. On the Credentials tab, set the **Azure AD** toggle for the credential you wish to use to *Yes*.

5. Click *Initialize scan* to save the scan task and have it run immediately or at the scheduled time.

Step 5: View Entra ID assets

After a successful sync, you can go to your inventory to view your Entra ID assets. These assets will have an Active Directory icon listed in the *Source* column.

To filter by Entra ID assets, consider running the following queries:

- View all Entra ID assets:

  ```
  source:azuread
  ```

- View runZero assets not connected to Entra ID:

  ```
  source:runzero AND NOT source:azuread
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by Entra ID.

Community Platform

The Entra ID integration provides details about users and groups in addition to enriching asset inventory data. Go to Inventory > Users or Inventory > Groups to view the data provided by Entra ID.

Filtering Entra ID assets

An optional filter can be applied to Entra ID integration tasks. runZero uses Microsoft Graph `$filter` query paramater to filter assets. GraphQL follows the syntax `<property> [operator] <value>`. Multiple expressions can be combined for more complex filtering by adding an `and` or `or` between expressions.

Properties

Any property that runZero imports from Entra ID can be used to apply a filter. The following are some examples.

Entra ID Property	runZero Attribute	Description	Example
`displayName`	@azuread.dev.displayName	The hostname of the device	EXPLORER-01
`operatingSystem`	@azuread.dev.operatingSystem	The operation system of the device	Windows
`operatingSystemVersion`	@azuread.dev.operatingSystemVersion	The version of the specified operoating system	10.0.x
`manufacturer`	@azuread.dev.manufacturer	The manufacturer of the device	Dell Inc.
`model`	@azuread.dev.model	The model of the device	Precision 3560
`isManaged`	@azuread.dev.isManaged	Boolean value specifying whether device is managed	true, false
`managementType`	@azuread.dev.managementType	Description of how the device is managed	MDM, MicrosoftSense
`deviceOwnership`	@azuread.dev.deviceOwnership	Description of who owns the device	Company, Personal

Operators

The following are common operators that can be used in an Entra ID filter.

- Equal to (`eq`)
- Not equal to (`ne`)
- Has (`has`)
- Less than (`lt`)
- Greather than (`gt`)
- Less than or equal to (`le`)
- Greater than or equal to (`ge`)

The following are common functions that can be used in an Entra ID filter. Functions follow the syntax `function(<property>, <value>)`.

- Starts with (`startswith`)
- Ends with (`endswith`)
- Contains (`contains`)

Example Filters

The following are examples of filters that can be applied to an Entra ID integration.

Search Filter	Description
`not(operatingSystem has 'Android')`	Import all assets except those with an Android operating system
`not(operatingSystem eq 'iOS') and not(operatingSystem eq 'iPad')`	Import all assets execpt those with an iOS or IPad operating system
`startswith(displayName, 'PROD')`	Import all devices with a hostname that starts with PROD
`not(startswith(displayName, 'DEV'))`	Import all devices except those with a hostname that starts with DEV
`deviceOwnership eq 'Company' or isManaged eq true`	Import all devices that are owned by company or that are configured as managed devices

For more information about filter syntax and additional examples, check Microsoft's Graph API documentation.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Azure Active Directory integration unable to connect?

1. Are you getting any data from the Entra ID integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Entra ID integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

How do I solve the following Entra ID errors?

- `(invalid_client) AADSTS7000218: The request body must contain the following parameter: 'client_assertion' or 'client_secret'`
    1. This error can be corrected by enabling **Allow Public Client Flows** in Entra ID. This can be accomplished by entering the Application details page under *Authentication > Advanced Settings*. From here you can toggle the **Allow Public Client Flows** setting to *Yes*.
    2. It can also be helpful to ensure that application permissions were granted correctly when registering the Azure application for Entra ID API access. To do this, navigate to the API permissions settings page and ensure that each of the API/Permissions have the *type* set to *application*. Also make sure that the permission granted is *Grant Admin Consent for Default Directory*.
- `failed to get Entra ID groups: invalid response: 403 (403 Forbidden)`

  This error is likely due to an issue with credentials. Please review the documentation and check your credentials to ensure everything was entered correctly and no steps were accidentally skipped.

Microsoft Intune

Community Platform

runZero integrates with Microsoft Intune to allow you to sync and enrich your asset inventory. Adding your Microsoft Intune data to runZero makes it easier to find unmanaged assets on your network. Data added includes the discovered apps from Intune. Managed apps (those pushed to devices by Intune) are not currently reported.

Getting started

To set up the Microsoft Intune integration, you'll need to:

1. Configure Microsoft Intune to allow API access from runZero.
2. Add the Microsoft Intune credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the Microsoft Intune integration to sync your data with runZero.

Requirements

Before you can set up the Microsoft Intune integration:

- Make sure you have access to the Microsoft Azure portal.

Step 1: Register an Azure application for Microsoft Intune API access

runZero can authenticate to the Microsoft Intune API using either a username and password or a client secret. Register an application to configure Microsoft Intune API access.

1. Sign in to the Microsoft Azure portal.
2. Go to *App registrations* and click on *+ New registration*.
    - Provide a name.
    - Select the supported account types.
    - Optionally add a redirect URI.
3. Click *Register* to register the application.
4. Once the application is created, go back to the main Azure portal page and select *App registrations*. You should be able to find the application you just registered. (It may help to select the *Owned applications* tab.) Click on the app's name
5. You should see the an overview of the app registration. Note the following information:
    - Application (client) ID
    - Directory (tenant) ID
6. From the application's details page, go to *Manage > API permissions* and choose *+ Add a permission*.
7. Select *Microsoft Graph* from the list of Microsoft APIs.
8.

Select the correct permissions type for your needs:

- **Username & password**: select *Delegated permissions*
- **Client secret**: select *Application permissions*

9. Search for and select the following required permission:

   - `DeviceManagementManagedDevices.Read.All`
   - `User.Read.All`

10. Click *Add permissions* to save the permissions to the application.

11. Click *Grant admin consent* to grant consent for the permissions to the application.

12. If using a client secret, also perform the following steps from the app management pages:

    - Navigate to *App registrations* and select the application you created.
    - Go to *Certificates & secrets* and click on *+ New client secret*.
      - Enter a description.
      - Select the expiration.
    - Click *Add* to create the client secret and save the client secret value.

Step 2: Add the Microsoft Intune credential to runZero

Adding the Microsoft Intune credential requires adding an Azure username and password to and an Azure Client Secret to runZero. The following sub-steps breaks down each task.

Step 2a: Add an Azure Username & Password credential to runZero

1. Go to the Credentials page in runZero and click *Add Credential*.

2. Provide a name for the credential, like `Azure User/Pass`.

3. Choose *Azure Username & Password* from the list of credential types.

4. Provide the following information:

   - **Azure application (client) ID** - The unique ID for the registered application. This can be found in the Azure portal if you go to *App registrations* and select the application.
   - **Azure directory (tenant) ID** - The unique ID for the tenant. This can be found in the Azure portal if you go to *App registrations* and select the application.
   - **Azure username** - The username for your Azure cloud account. This cannot be a federated user account.
   - **Azure password** - The password for your Azure cloud account.

5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.

6. Save the credential. You're now ready to set up and activate the connection to bring in data from Azure.

Step 2b: Add an Azure Client Secret credential to runZero

This type of credential can be used to sync all resources in a single directory (across multiple subscriptions).

1. Go to the Credentials page in runZero and click *Add Credential*.
2. Provide a name for the credential, like `Azure Client Secret`.
3. Choose *Azure Client Secret* from the list of credential types.
4. Provide the following information:
    - **Azure application (client) ID** - The unique ID for the registered application. This can be found in the Azure portal if you go to *App registrations* and select the application.
    - **Azure client secret** - To generate a client secret, go to *App registrations*, select your application, go to *Manage > Certificates & secrets* and click on *New client secret*.
    - **Azure directory (tenant) ID** - The unique ID for the tenant. This can be found in the Azure portal if you go to *App registrations* and select the application.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per organization basis.
6. Save the credential. You're now ready to set up and activate the connection to bring in data from Azure.

Step 3: Choose how to configure the Microsoft Intune integration

The Microsoft Intune integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the Microsoft Intune integration to sync data

After you add your Microsoft Intune credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the Microsoft Intune integration as a connector task

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Microsoft Intune-only assets are created.

1. Activate a connection to Microsoft Intune. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure it has access to the organization you are currently in.
3. Optionally provide a filter to import only devices that match it. You can find more details in the Filtering Intune assets section below.
4. Enter a name for the task, like `Microsoft Intune sync`.
5. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
6. Under **Task configuration**, choose the site you want to add your assets to.
7. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
8. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Microsoft Intune integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the Intune toggle to `Yes`, and change any of the default options if needed.
4. On the Credentials tab, set the Intune toggle for the credential you wish to use to `Yes`.
5. Click *Initialize scan* to save the scan task and have it run immediately or at the scheduled time.

Step 5: View Microsoft Intune assets

After a successful sync, you can go to your inventory to view your Microsoft Intune assets. These assets will have an Active Directory icon listed in the *Source* column.

To filter by Microsoft Intune assets, consider running the following queries:

- View all Microsoft Intune assets:

  ```
  source:intune
  ```

- View runZero assets not connected to Microsoft Intune:

  ```
  source:runzero AND NOT source:intune
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by Microsoft Intune.

Filtering Intune assets

An optional filter can be applied to Intune integration tasks. runZero uses the Intune reports API to retrieve assets from the `DevicesWithInventory` report. This API accepts a filter field to narrow down the results.

Properties

The `DevicesWithInventory` report supports filtering based on the following properties:

- CreatedDate
- LastContact
- CategoryName
- CompliantState
- ManagementAgents
- OwnerType
- ManagementState
- DeviceType
- JailBroken
- EnrollmentType
- PartnerFeaturesBitmask

For the most up-to-date list of properties, please refer to Microsoft's documentation on the DevicesWithInventory report. Note that not all the properties listed on that table are filterable. Scroll just below the table to find a separate list of properties that can be used in filters.

Operators

There is no official documentation on the filter syntax or supported operators for the Intune reports API, but based on our testing, the following operators may be supported. In practice, we've found that some of these, such as `not` and `in`, can produce errors, so your results may vary.

- Equal to (`eq`)
- Not equal to (`ne`)
- Logical negation (`not`)
- In (`in`)

- Has (`has`)
- Less than (`lt`)
- Greather than (`gt`)
- Less than or equal to (`le`)
- Greater than or equal to (`ge`)

The syntax appears to be similar to the `$filter` query parameter used elsewhere in Microsoft Graph. Examples for this syntax can be found in Use the $filter query parameter. We recommend testing filters carefully and starting with basic operators like `eq` and `ne` for best results.

Example Filters

The following are examples of filters that can be applied to an Intune integration.

Search Filter	Description
`DeviceType eq 'android'`	Import all android devices
`LastContact ge '2023-02-23 23:50:01.0000000'`	Import devices that checked in after a specific date and time

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Microsoft Intune integration unable to connect?

1. Are you getting any data from the Microsoft Intune integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Microsoft Intune integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

How do I solve the following Microsoft Intune error:

- `(invalid_client) AADSTS7000218: The request body must contain the following parameter: 'client_assertion' or 'client_secret'`

This error means that you need to enable `Public Client Flows` in Azure. To do so, follow these steps:

1. Navigate to the App Registration page in the Azure portal
2. Choose *Authentication* from the left navigation
3. Select *Advanced Settings*
4. Toggle the *Allow Public Client Flows* switch at the bottom of the page to *Yes*

Miradore MDM

Community Platform

runZero integrates with Miradore mobile device management (MDM) to deliver greater visibility into your mobile assets. This integration imports data from the Miradore API to enrich your asset inventory. Syncing with Miradore allows you to view information about device hardware, OS version, associated user, and more. This integration imports all enrolled devices.

Getting started

To set up the Miradore integration, you'll need to:

1. Sign in to your Miradore web portal and create a new API key.
2. Add the Miradore credential to runZero, which includes the endpoint hostname and API key.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the Miradore integration to sync your data with runZero.

Requirements

Before you can set up the Miradore integration:

- Make sure you have access to the Miradore MDM portal as an administrator.

Step 1: Create a Miradore API key

1. Sign in to your Miradore portal as an administrator.
2. Click on the **System > Infrastructure diagram** link from the left-side navigation.
3. Scroll down to the find the **API** node in the diagram. Click this and choose **Create API key**.
4. Give this new key a name and copy the secret value.

Step 2: Add the Miradore credential to runZero

1. Go to the Credentials page in runZero and click **Add Credential**.
2. Provide a name for the credential, like `Miradore MDM`.

3. Choose **Miradore MDM API Key** from the list of credential types.
4. Provide the following information:
    - **Name** - Give this credential a unique name (ex: Miradore)
    - **Miradore endpoint hostname** - The URL for your Miradore portal.
    - **Miradore API key** - The API key created in step 1.
5. Save the credential.

You're now ready to set up and activate the connection to bring in data from Miradore.

Step 3: Choose how to configure the Miradore integration

The Miradore integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the Miradore MDM integration to sync data

After you add your Miradore credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the Miradore integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Miradore-only assets are created.

1. Activate a connection to Miradore. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure the credential has access to the organization you are currently in.
3. Enter a name for the task, like `Miradore sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. Under **Task configuration**, choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
6. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
7. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Miradore integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the Miradore toggle to *Yes*, and change any of the default options if needed.

4. On the Credentials tab, set the Miradore toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 5: View Miradore assets

After a successful sync, you can go to your inventory to view your Miradore assets. These assets will have a Miradore icon listed in the **Source** column.

To filter by Miradore assets, consider running the following queries:

- View all Miradore assets:

```
source:miradore
```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by the Miradore API.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Miradore integration unable to connect?

1. Are you getting any data from the Miradore integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Miradore integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the Miradore host if it is on-premises.

NetBox CMDB

Community Platform

runZero integrates with NetBox configuration management database (CMDB) using the NetBox REST API to enrich your asset inventory.

NetBox limitations

runZero explicitly supports NetBox version 4. Older versions (3.x) may be compatible, but are not specifically tested or supported.

Since NetBox data entry is free-form, the runZero integration may not be a good fit for your organization, and we strongly recommend testing this integration in a Project first before configuring it for a production Organization.

The NetBox integration will import Devices and Virtual Machines, as well as their associated resources.

The NetBox integration will NOT create entities for anything other than Devices and Virtual Machines.

The Device and Virtual Machine records are imported as runZero assets with the following fields:

- Device or Virtual Machine
  - ID in the form of `device.<ID>` or `vm.<ID>`
  - Name
  - FirstSeenTS and LastSeenTS
  - Serial
  - AssetTag (Device only)
  - Interfaces
    - IP addresses for PrimaryIP / PrimaryIP4 / Primary IP6 / OOB IP
    - Additional IP addresses using the `ip<index>.field` syntax
    - NAT Inside and NAT Outside addresses
    - MAC Addresses (if entered in NetBox)
    - CreatedTS and UpdatedTS
    - ID, DNSName, Name, Comments, and Description
    - Module Name, MTU, Speed (Device only)
    - Tagged and Untagged VLANs
    - Custom Fields
  - Prefixes
    - VLAN and CIDR
    - ID, Comments, and Description
    - Role Name
    - Custom Fields
  - Location
    - ID and Name
    - Latitude and Longitude
    - Chained Location Path
    - Custom Fields
  - Rack
    - ID and Name
    - Position and Rack Face
  - Status
  -

    - Manufacturer
        - ID, Name, and Model
        - Comments, Description, and Part Number
        - Custom Fields
    - Device Role
        - ID, Name, Description
        - Custom Fields
    - Platform
        - ID & Name
        - Custom Fields
    - Site
        - ID, Name, Description, Comments
        - Facility, Physical Address, Shipping Address
        - Region, Group
        - Custom Fields
    - Virtual Chassis
        - ID, Name, Comments, and Description
        - Master ID and Master Name
        - VC Position and Priority
    - Interface Count
    - Inventory Item Count
    - Custom Fields

The following top-level attributes are rollups across all sub-fields:

- Addrs
- Names
- MACs
- VLANs

NetBox asset matching

runZero will use the Name, IP Address, and MAC Addresses to match NetBox Devices and Virtual Machines to runZero assets. If there isn't a direct match, runZero will create a new asset record, unless the "Exclude unknown assets" option is enabled. Since NetBox data is typically static and network assets tend to change over time, there is a good chance that many assets will not match. MAC Addresses are the most reliable way to correlate these, but many organizations do not fill out the MAC Address fields in NetBox.

NetBox asset fingerprinting

runZero will use the `mfg.name` as the hardware vendor and `mfg.model` as the hardware model for Devices. runZero will report Virtual Machine vendors by matching the Cluster Type field to a known list of common hypervisors (Xen, VMware, Proxmox, etc.). The `platform.name` field, with and without the `mfg.name` field as a prefix will be used to find an operating system match.

The NetBox operating system data in `platform.name` will be treated as low-confidence relative to agent-based sources like EDRs and CMDB.

The NetBox hardware values will be treated similarly, with live agent-based sources taking priority.

Getting started

To set up the NetBox integration, you will need:

1. A NetBox instance in the cloud or that is reachable by one of your runZero Explorers.
2. An administrative account in the NetBox instance.

Step 1: Create a NetBox API key

1. Sign in to your NetBox portal.
2. Click on your name in the upper right and select **API Tokens** from the drop-down.
3. Click **Add a Token** and define a new token with no expiration and without Write permissions.
4. Give this new key a name and copy the secret value.

Step 2: Add the NetBox credential to runZero

1. Go to the Credentials page in runZero and click **Add Credential**.
2. Provide a name for the credential, like `NetBox CMDB`.
3. Choose **NetBox API Key** from the list of credential types.
4. Provide the following information:
    - **Name** - Give this credential a unique name (ex:NetBox)
    - **NetBox URL** - The URL for your NetBox portal.
    - **NetBox API key** - The API key created in step 1.
5. Save the credential.

You're now ready to set up and activate the connection to bring in data from NetBox.

Step 3: Configure the NetBox integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new NetBox-only assets are created.

1. We strongly recommend testing NetBox in a a new Project first before applying this to your production Organization. Create a new Project.
2. Activate a connection to NetBox. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
3. Choose the credential you added earlier. If you don't see the credential listed, make sure the credential has access to the organization or project you are currently in.
4. Enter a name for the task, like `NetBox sync`.
5. Leave the default settings to create a one-off sync task.
6. Under **Task configuration**, choose the site you want to add your assets to. NetBox assets that do not match existing assets will use this site by default.

7. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*.
8. Activate the connection when you are done. The task will run. You can always check the Tasks page to track the status.

Step 4: View NetBox assets

After a successful sync, you can go to your inventory to view your NetBox assets. These assets will have a NetBox icon listed in the **Source** column.

To filter by NetBox assets, consider running the following queries:

- View all NetBox assets:

```
source:netbox
```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by the NetBox API.

Review the new NetBox assets and confirm that the NetBox-sourced asset count is roughly in-line with your overall inventory and the total in NetBox itself

If there are far fewer NetBox-sourced assets in runZero, you may want to avoid the final step until NetBox is updated with better correlation data.

Step 5: Enable Unknown Assets in Recurring Sync

Configure a new recurring NetBox task, with *Exclude unknown assets* disabled. This will create new assets for anything in NetBox that runZero can't correlate to in your asset inventory. If this process creates significant duplicates, delete all assets matching the filter `source_count:1 AND source:netbox`, and re-enable the *Exclude unknown assets* option in your recurring task until the NetBox records have better correlation data (MAC addresses, hostnames, etc.).

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the NetBox integration unable to connect?

1. Are you getting any data from the NetBox integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the NetBox integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the URL
    - 404 - hitting an unknown URL on the server
    - 401 - not authorized, likely a credential issue
    - 403 - access denied, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the NetBox host if it is on-premises.

Qualys VMDR

Community Platform

runZero integrates with Qualys VMDR by importing data from the Qualys KnowledgeBase API.

Asset inventory

There is a column on the asset inventory page showing the count of vulnerabilities detected by Qualys for each asset. When a single asset is selected, the vulnerabilities table lists all the results related to that asset. The vulnerability count can be impacted by the type of vulnerability scan as well as the import settings selected.

Vulnerabilities table

The **Vulnerabilities** tab of the inventory lists all vulnerability results that have been imported from Qualys. The table lists every result, and selecting a result will take you to the page for the impacted asset.

Severity and risk scores

Qualys assigns all vulnerabilities a severity rating (Minimal, Medium, Serious, Critical, Urgent). runZero normalizes the severities shown in the vulnerability inventory to be consistent across the runZero Console.

runZero Severity	Qualys Severity
Info	1 / Minimal
Low	2 / Medium
Medium	3 / Serious
High	4 / Critical
Critical	5 / Urgent

runZero will also normalize risk scores assigned by Qualys. A risk score of 0.0 will be shown as `none` in the runZero Console, and all other risk scores will match the assigned severity level.

Getting started with Qualys

To set up the Qualys VMDR integration, you'll need to:

1. Create or obtain user credentials with access to the Qualys API.
2. Configure CVSS scoring in Qualys.
3. Add the Qualys API username, password, and account API URL in runZero.
4. Choose whether to configure the integration as a scan probe or connector task.
5. Activate the Qualys integration to pull your data into runZero.

Requirements

Before you can set up the Qualys VMDR integration:

- Make sure you have access to the Qualys Cloud Platform portal.

Step 1: Add the Qualys credentials to runZero

1. Go to the Credentials page in runZero. Provide a name for the credentials, like `Qualys`.
2. Choose **Qualys Username & Password** from the list of credential types.

3. Provide the following information:
    - **Qualys username** - the username you want to use to connect to the Qualys API.
    - **Qualys password** - the password for your Qualys API username.
    - **Qualys account API URL** - the URL of the Qualys API for the relevant account. The expected format is `https://ip:port` or `https://domain.tld:port`. This URL is unique for each Qualys user.
4. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential. You're now ready to set up and activate the connection to bring in data from Qualys VMDR.

> Note
> The role of the Qualys API user will determine which VM scanned hosts are visible to the integration. Managers may view all VM scanned hosts in the subscription. Auditors have no permission to view VM scanned hosts. Unit Managers may view VM scanned hosts in the user's assigned business unit. Scanners and Readers may view VM scanned hosts in the user's account, but must be assigned to the assets through asset groups in VM/VMDR.

Step 2: Choose how to configure the Qualys integration

The Qualys integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 3: Set up and activate the Qualys integration to sync data

After you add your Qualys credential, you'll need to sync your data.

> Note
> The Qualys Cloud Platform enforces limits on the API calls subscription users can make. Both API controls are limited per subscription based on your service level. The tasks generated by this integration may experience slow performance or failures as a result of the enforced API limits. The Qualys API documentation on this topic can be found here.

Step 3a: Configure the Qualys scan probe

You can run the Qualys VMDR integration as a scan probe so that the runZero Explorer will pull your vulnerability data into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *QUALYS* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.

- Set the correct Qualys credential to *Yes* in the *Credentials* tab.
- Optionally, set the severity and risk levels for ingested vulnerability scan results.

Step 3b: Configure the Qualys connector

A connection requires you to set a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Qualys-only assets are created.

1. Activate a connection to Qualys. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Set the severity and risk levels you want to import (optional).
4. Set the **Fingerprint only** toggle to *Yes* if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).
5. Specify the tags to include in the import. This should be a comma-separated list. Only assets that match any of the provided tags will be imported (optional).
6. Specify the network IDs to include in the import. This should be a comma-separated list. Only assets that match any of the provided network IDs will be imported (optional).
7. Enter a name for the task, like `Qualys sync` (optional).
8. Choose the Explorer to perform this connector task from (optional).
9. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
10. Enter a description for the task (optional).
11. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
12. If you want to include assets that have not been assessed for vulnerabilities, switch the **Include unscanned assets** toggle to *Yes*.
13. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
14. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4: View Qualys assets and vulnerabilities

After a successful sync, you can go to your inventory to view your Qualys assets. These assets will have a Qualys icon listed in the **Source** column.

The Qualys integration gathers details about vulnerabilities detected in addition to enriching asset inventory data. Go to Inventory > Vulnerabilities to view the vulnerability data provided by Qualys VMDR.

To filter by Qualys assets, consider running the following query:

- View all Qualys assets:

  ```
  source:Qualys
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from the Qualys VMDR scan data.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Qualys integration unable to connect?

1. Are you getting any data from the Qualys integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Qualys integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the Qualys host if it is on-premises.

Rapid7

Community Platform

runZero integrates with Rapid7's InsightVM and Nexpose to enrich your asset inventory and gain visibility into vulnerabilities detected in your environment.

Asset inventory

There is a column on the asset inventory page showing the count of vulnerabilities detected by Rapid7 for each asset. When a single asset is selected, the vulnerabilities table lists all the results related to that asset. The vulnerability count can be impacted by the type of vulnerability scan as well as the import settings selected.

Vulnerabilities table

The **Vulnerabilities** tab of the inventory lists all vulnerability results that have been imported from Rapid7. The table lists every result, and selecting a result will take you to the page for the impacted asset.

Severity and risk scores

Rapid7 assigns all vulnerabilities a severity rating (Moderate, Severe, or Critical) based on the vulnerability's CVSSv2 score. runZero normalizes the severities shown the vulnerability inventory to be consistent across the runZero Console.

runZero Severity	Rapid7 Severity	CVSS Range
Info	Moderate	0.0
Medium	Moderate	0.1 - 3.4
High	Severe	3.5 - 7.4
Critical	Critial	7.5 - 10.0

runZero will also normalize risk scores assigned by Rapid7. A risk score of 0.0 will be shown as `none` in the runZero Console, and all other risk scores will match the assigned severity level.

InsightVM

Community Platform

runZero integrates with Rapid7 InsightVM by importing data from the InsightVM API.

Both Rapid7 InsightVM Cloud and on-premises InsightVM are supported. For on-premises use you will need to use the InsightVM connector as a scan probe from a runZero Explorer which has network access to the InsightVM deployment.

The Insight Platform API is distinct from the InsightVM API, and is not supported.

Getting started with InsightVM

To set up the InsightVM integration, you'll need to:

1. Create or obtain user credentials to use with the InsightVM API.
2. Add the InsightVM API username, password, and API URL in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the InsightVM integration to pull your data into runZero.

Requirements

Before you can set up the InsightVM integration:

- Obtain credentials for an InsightVM user with administrator access to the InsightVM portal.
- Scan your InsightVM with a runZero Explorer if you want to use trusted authentication (optional).

Step 1: Add the InsightVM credentials to runZero

1. Create a new credential via the runZero Credentials page.
2. Provide a name for the credentials, like `InsightVM`.
3. Choose **InsightVM Username & Password** from the list of credential types.
4. Provide the following information:
    - **InsightVM username** - The username you want to use to connect to the InsightVM API. This account requires the `User` role or greater permissions in InsightVM.
    - **InsightVM password** - The password for your InsightVM username.
    - **InsightVM API URL** - The URL of your InsightVM API instance. By default, the InsightVM API uses port 3780. The expected format is `https://ip:3780` or `https://domain.tld:3780`. RunZero automatically appends `api/3/sites?size=500` to this URL.
    - **InsightVM insecure** - Set this to `Yes` if you want to attempt authentication without a verified thumbprint.
    - **InsightVM thumbprints** (optional) - A set of `IP=SHA256:B64HASH` or `domain.tld=SHA256:B64HASH` pairs to trust for authentication.
        - You will need to scan your InsightVM instance with runZero in order to obtain the TLS thumbprint. The TLS fingerprints service attribute report lists all previously seen fingerprints.
        - If *InsightVM insecure* is set to *No* and no thumbprints are provided:
            - With a self-signed certificate, the connection will fail because the certificate chain cannot be verified.
            - With a valid certificate from a public CA, the connection can work without thumbprints.
5. If you want all other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
6. Save the credential.

You're now ready to set up and activate the connection to bring in data from InsightVM.

Step 2: Choose how to configure the Rapid7 integration

The Rapid7 InsightVM integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync. Scan probes will be the right option for most users. Setting up a connector will only work for if you're self-hosting runZero or your InsightVM instance is publicly accessible.

Step 3: Sync your InsightVM data

After you add your InsightVM credential, you'll need to activate the integration to sync your data.

Step 3a: Configure the InsightVM scan probe

You can run the InsightVM integration as a scan probe so that the runZero Explorer will pull your vulnerability data into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *INSIGHTVM* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Set the correct InsightVM credential to *Yes* in the *Credentials* tab.
- Optionally, set the severity and risk levels for ingested vulnerability scan results.

Step 3b: Configure the InsightVM connector

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new InsightVM-only assets are created.

1. Activate a connection to InsightVM. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Set the severity and risk levels you want to import (optional).
4. Set the **Fingerprint only** toggle to *Yes* if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).
5. Enter a name for the task, like `InsightVM sync` (optional).
6. Choose the Explorer to perform this connector task from (optional).
7. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
8. Enter a description for the task (optional).
9. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
10. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
11. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4: View InsightVM assets and vulnerabilities

After a successful sync, you can go to your inventory to view your InsightVM assets. These assets will have a Rapid7 icon listed in the **Source** column.

The InsightVM integration gathers details about vulnerabilities detected in addition to enriching asset inventory data. Go to Inventory > Vulnerabilities to view the vulnerability data provided by InsightVM.

To filter by Rapid7 assets, consider running the following queries:

- View all Rapid7 assets:

```
source:Rapid7
```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from the Rapid7 scan data.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Rapid7 InsightVM integration unable to connect?

1. Are you getting any data from the Rapid7 InsightVM integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Rapid7 InsightVM integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

How can I get a TLS thumbprint for the InsightVM credential?

Here is a set of example commands that would calculate your TLS thumbprint, where is the IP address of your InsightVM instance:

```
$ (echo|openssl s_client -connect <IP address>:3780 -showcerts 2>/dev/null) | openssl x509 -inform PEM -outform DER |
```

An example value returned by this set of commands would look something like this: `X1NWttnkIQprK6zSre/VweKpbRlj7Dt4M6hNfUacytE=` Use the following entry for the corresponding TLS thumbprint: 192.168.0.3:3780=SHA256:X1NWttnkIQprK6zSre/VweKpbRlj7Dt4M6hNfUacytE=

Nexpose

Community Platform

runZero integrates with Rapid7 Nexpose by importing files that were exported from your Nexpose instance.

Getting started with Rapid7 Nexpose

To use the Rapid7 Nexpose integration, you'll need to:

1. Download an XML Export or XML Export 2.0 report from Nexpose.

2. Import the Nexpose files through the inventory pages.

Requirements

Before you can set up the Nexpose integration:

- Make sure you have access to the Nexpose portal.

Step 1: Export Nexpose vulnerability scan report

1. Sign in to Nexpose with the account being used for the runZero integration.
2. Go to the Reports page and select *Create a report*.
3. From the Export tab, select either XML Report or XML Report 2.0.
4. Set the scan, asset, asset group, or site scope.
5. Click *Save & Run the Report*.
6. When the report completes, save the report to a local file.

Step 2: Import the Nexpose files into runZero

1. Go to the Inventory page in runZero.
2. Choose **Import** > **Nexpose XML Export (.xml)** from the list of import types.
3. On the import data page:
    - Choose the site you want to add your assets to.
    - Set tags to apply to the imported assets (optional).
    - Set the severity and risk levels to ingest (optional).
    - Set the **Fingerprint only** toggle to *Yes* if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).

Step 3: View Nexpose assets and vulnerabilities

After a successful sync, you can go to your inventory to view your Nexpose assets. These assets will have a Rapid7 icon listed in the **Source** column.

The Nexpose integration gathers details about vulnerabilities detected in addition to enriching asset inventory data. Go to Inventory > Vulnerabilities to view the vulnerability data provided by Nexpose.

To filter by Rapid7 assets, consider running the following queries:

- View all Rapid7 assets:

  ```
  source:Rapid7
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from the Nexpose scan file.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Rapid7 Nexpose integration unable to connect?

1. Are you getting any data from the Rapid7 Nexpose integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Rapid7 Nexpose integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the Nexpose host.

SentinelOne

Community Platform

runZero integrates with SentinelOne by importing data from the SentinelOne API. This integration allows you to sync and enrich your asset inventory, import software installed on assets, and import vulnerabilities affecting the installed software. Adding your SentinelOne data to runZero makes it easier to find things like endpoints that are missing required software or identify vulnerable endpoints.

Any IP address reported by SentinelOne will be treated as a secondary address, not a primary address, since these IPs can be stale and may not be associated with a specific network or site.

Getting started

To set up the SentinelOne integration, you'll need to:

1. Configure SentinelOne to allow API access through runZero.
2. Add the SentinelOne API key and SentinelOne base API URL in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the SentinelOne integration to sync your data with runZero.

Requirements

Before you can set up the SentinelOne integration:

- Make sure you have access to the SentinelOne admin portal.

Step 1: Configure SentinelOne to allow API access to runZero

1. Sign in to SentinelOne with the account being used for the runZero integration.
2. Go to **User > My User**.
3. **Generate** the API token, then download or copy it. This API key expires and will need to be regenerated every six months.

Step 2: Add the SentinelOne credential to runZero

1. Go to the Credentials page in runZero. Provide a name for the credentials, like `SentinelOne`.
2. Choose **SentinelOne API key** from the list of credential types.
3. Provide the following information:
    - **SentinelOne API URL** - Your organization-specific base URL, which will depend on your account type. It will be something like `organization.sentinelone.net`.
    - **SentinelOne API key** - To generate your API key, go to **User > My User** in your SentinelOne portal. From there, a key can be generated, regenerated, or revoked.
4. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential.

You're now ready to set up and activate the connection to bring in data from SentinelOne.

Step 3: Choose how to configure the SentinelOne integration

The SentinelOne integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the SentinelOne integration to sync data

After you add your SentinelOne credential, you'll need to set up a connector task or scan probe to sync your data.

Step 4a: Configure the SentinelOne integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new SentinelOne-only assets are created.

1. Activate a connection to SentinelOne. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.

2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Enter a name for the task, like `SentinelOne sync`.
4. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
5. Under **Task configuration**, choose the site you want to add your assets to.
6. If you do not want to import software, switch the **Import Software** toggle to *No*.
7. If you do not want to import vulnerabilities, switch the **Import Vulnerabilities** toggle to *No*.
8. If the **Import Vulnerabilities** toggle is set to *Yes*, you can set the desired vulnerability severities to import using the **Severities** checkboxes.
9. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
10. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the SentinelOne integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the SentinelOne toggle to *Yes*, and change any of the default options if needed.
4. On the Credentials tab, set the SentinelOne toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 5: View SentinelOne assets and software

After a successful sync, you can go to your inventory to view your SentinelOne assets. These assets will have a SentinelOne icon listed in the **Source** column.

The SentinelOne integration gathers details about installed software in addition to enriching asset inventory data. Go to Inventory > Software to view the software data provided by SentinelOne.

To filter by SentinelOne assets, consider running the following queries:

- View all SentinelOne assets:

  ```
  source:SentinelOne
  ```

- Find assets that have a SentinelOne agent installed:

  ```
  edr.name:SentinelOne
  ```

- Find Windows assets, excluding servers, that are missing a SentinelOne agent:

  ```
  os:windows and not type:server and not edr.name:SentinelOne
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes returned by the SentinelOne API, with the exception of policies.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the SentinelOne integration unable to connect?

1. Are you getting any data from the SentinelOne integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the SentinelOne integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the SentinelOne host if it is on-premises.

Shodan

Community Platform

runZero integrates with Shodan by importing data from the Shodan API. This integration allows you to sync data about your externally-facing assets and services from Shodan to provide better visibility of your internet footprint and cyber hygiene.

Getting started

To set up the Shodan integration, you'll need to:

1. Add the Shodan API key in runZero.
2. Choose whether to configure the integration as a scan probe or connector task.
3. Activate the Shodan integration to sync your data with runZero.

Requirements

Before you can set up the Shodan integration:

- Make sure you have a Shodan account with the correct license to meet your needs.

Step 1: Add the Shodan credential to runZero

1. Go to the new credential page in runZero. Provide a name for the credential, like `Shodan`.
2. Choose **Shodan Search API key** from the list of credential types.
3. Provide your **Shodan Search API key** - To view your API key, go to your Account page in the Shodan portal. Your API key is available on that page and can be reset if needed.
4. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential. You're now ready to set up and activate the connection to bring in data from Shodan.

Step 2: Choose how to configure the Shodan integration

The Shodan integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 3: Set up and activate the Shodan integration to sync data

After you add your Shodan credential, you'll need to set up a connection or a scan probe to sync your data from Shodan.

Step 3a: Configure the Shodan integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Shodan-only assets are created.

1. Activate a connection to Shodan. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credential you added earlier. If you don't see the credential listed, make sure the credential has access to the organization you are currently in.
3. You can choose whether to specify a Shodan search using Shodan's search syntax, or to have runZero generate a search for all of the public IP addresses of live assets.
4. Enter a name for the task, like `Shodan sync`.
5. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start at the date and time you have set.
6. Under **Task configuration**, choose the site you want to add your assets to.
7. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.

8. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the tasks page to see when the next sync will occur.

Step 3b: Configure the Shodan integration as a scan probe

1. Create a new scan task or select a future or recurring scan task from your Tasks page.
2. Add or update the scan parameters based on any additional requirements.
3. On the Probes and SNMP tab, choose which additional probes to include, set the Shodan toggle to *Yes*, and change any of the default options if needed. As with running the integration as a connector task, you can choose to specify a Shodan search string directly, or choose *assets* mode to have runZero generate a search query to look for all public IP addresses of live assets.
4. On the Credentials tab, set the Shodan toggle for the credential you wish to use to *Yes*.
5. Click **Initialize scan** to save the scan task and have it run immediately or at the scheduled time.

Step 4: View Shodan assets and services

After a successful sync, you can go to your inventory to view your Shodan assets. These assets will have a Shodan icon listed in the **Source** column.

The Shodan integration gathers details about services in addition to enriching asset inventory data. Go to Inventory > Services to view the service data provided by Shodan.

To filter by Shodan assets or services, consider running the following queries:

- View all Shodan assets:

  ```
  source:shodan
  ```

- View all Shodan services:

  ```
  source:shodan
  ```

Click into each asset or service to see its individual attributes. runZero will show you the attributes returned by the Shodan Search API.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Shodan integration unable to connect?

1. Are you getting any data from the Shodan integration?
   - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
   - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.

2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Shodan integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

Tanium API Gateway

Community Platform

runZero integrates with Tanium by importing data from the Tanium Gateway API. This integration allows you to sync data about your endpoints, applications, and vulnerabilities from Tanium to provide better visibility over your network.

Getting started with Tanium

To set up an integration with Tanium, you'll need to:

1. Generate an API token with the necessary permissions.
2. Configure the Tanium credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the integration to pull your data into runZero.

Step 1: Generate an API key in Tanium Dashboard

1. Sign in to Tanium and navigate to **Administration > Roles**.
2. Create a role with the necessary permissions:
    1. Search for the **Gateway User** role.
    2. Select it and click the **Clone** button that appears to create a copy of this role.
    3. On the **Clone Role** screen, enable **Platform Content Permissions > Sensor > Read** and add these Content Sets (via the **n+** button beside the green check):
        - Base
        - Comply Reporting
        - Core AD Query Content
        - Core Content
        - Reserved
        - Tanium Data Service
    4. Save the role.
3. Navigate to **Administration > Personas** and click **New Persona** to create a persona using the role you just created:
    1. Name the persona.
    2. Under **Manage Roles**, search for and apply your new role.
    3. Under **Computer Groups**, add the groups you need, or check **Unrestricted Management Rights** to allow access to all Computer Groups.

    4. Assign a user or service account which has the permissions granted to the persona.
    5. Save the persona.
4. Navigate to **Administration > API Tokens** and click **New API Token**.
    1. Enter a name and select a TTL.
    2. Select the persona you just created from the dropdown (you may need to refresh the page for it to appear).
    3. Enter IP addresses to allow requests from:
        - If you will run the integration via an Explorer or CLI, enter the IP addresses or ranges of your host(s);
        - Otherwise, enter `0.0.0.0/0`.
    4. Save the API token.

Step 2: Add the Tanium API token to runZero

1. Go to the Credentials page in runZero.
2. Choose **Tanium API Token** from the list of credential types.
3. Provide a name for the credential, like `Tanium`.
4. Provide the following information:
    - **Tanium API URL** - Your Tanium API Gateway URL. The full URL will be something like `https://<customername>-api.cloud.tanium.com/plugin/products/gateway/graphql`. If the path (`/plugin/products/gateway/graphql`) is omitted, it will be added automatically when the API is called.
    - **Tanium API token** - The API token (including the `token-` prefix) created in step 1.
5. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
6. Verify and save the credential.

You're now ready to set up and activate the connection to bring in data from Tanium.

Step 3: Choose how to configure the Tanium integration

The Tanium integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the integration to sync data

After you add your Tanium credential, you'll need to sync your data from Tanium.

Step 4a: Configure the Tanium integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Tanium-only assets are created.

1. Activate a connection to Tanium. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Optionally provide a list of computer groups to include in the import. The list must be comma-separated. We will only import data for the computer groups specified.
4. Enter a name for the task, like `Tanium Sync` (optional).
5. Choose the Explorer to perform this connector task from (optional).
6. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
7. Enter a description for the task (optional).
8. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
9. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
10. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Tanium integration as a scan probe

You can run the Tanium integration as a scan probe so that the runZero Explorer will pull your Tanium assets into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *TANIUM* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Set the correct *TANIUM* credential to *Yes* in the *Credentials* tab.

Step 5: View Tanium assets

After a successful sync, you can go to your inventory to view your Tanium assets. These assets will have a Tanium icon listed in the **Source** column.

To filter by Tanium assets, consider running the following queries:

- View all Tanium assets:

  ```
  source:Tanium
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from Tanium.

Tenable

Community Platform

runZero integrates with Tenable Vulnerability Management (previously Tenable.io), Tenable Nessus, and Tenable Security Center to enrich your asset inventory and gain visibility into vulnerabilities detected in your environment. The Tenable Vulnerability Management, Nessus Professional, and Tenable Security Center integrations pull data from the Tenable API, while all versions of Tenable Nessus and Tenable Security Center (previously Tenable.sc) are also supported through Nessus v2 file imports (`.nessus`).

Note that at this time, only the main Tenable Vulnerability Management cloud API endpoint at `https://cloud.tenable.com` is supported as an API integration.

Asset inventory

There is a column on the asset inventory page showing the count of vulnerabilities detected by Tenable for each asset. When a single asset is selected, the vulnerabilities table lists all the results related to that asset. The vulnerability count can be impacted by the type of vulnerability scan as well as the import settings selected.

Vulnerabilities table

The **Vulnerabilities** tab of the inventory lists all vulnerability results that have been imported from Tenable. The table lists every result, and selecting a result will take you to the page for the impacted asset.

Severity and risk scores

Tenable uses the third-party Common Vulnerability Scoring System (CVSS) values from the National Vulnerability Database (NVD) to describe severity associated with vulnerabilities. Tenable assigns all vulnerabilities a severity rating (Info, Low, Medium, High, or Critical) based on the vulnerability's static CVSSv2 or CVSSv3 score. By default, Tenable uses CVSSv2, however users can opt to use CVSSv3 instead. runZero displays the CVSSv2-based severity in the Inventory and Asset views.

To supplement the severity ratings, Tenable calculates a dynamic Vulnerability Priority Rating (VPR) for most vulnerabilities. The VPR is a dynamic companion to the data provided by the vulnerability's CVSS score, since Tenable updates the VPR to reflect the current threat landscape. VPR values range from 0.1-10.0, with a higher value representing a higher likelihood of exploit. runZero displays the VPR-based risk in the Inventory and Asset views. For more details about how Tenable handles severity and risk, please refer to their documentation.

Tenable Vulnerability Management

Community Platform

runZero integrates with Tenable Vulnerability Management (previously Tenable.io) by importing data from the Tenable API.

Getting started with Tenable Vulnerability Management

To set up an integration with Tenable Vulnerability Management, you'll need to:

1. Create an Administrator API key in an access group with `Can View` permission to `Manage Assets`.
2. Configure the Tenable Vulnerability Management credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the integration to pull your data into runZero.

Requirements

Before you can set up the Tenable Vulnerability Management integration:

- Make sure you have administrator access to the Tenable portal.

Step 1: Create an Administrator API key

1. Sign in to Tenable Vulnerability Management with the Administrator account being used for the runZero integration.
2. Go to **My Profile** > **My Account** > **API Keys**.
3. **Generate** the API token, and then download or copy it.

Step 2: Add the Tenable credential to runZero

1. Go to the Credentials page in runZero. Provide a name for the credentials, like `Tenable Vulnerability Management`.
2. Choose **Tenable.io Access & Secret** from the list of credential types.
3. Generate your Tenable access and secret keys via your account page in the Tenable portal, and then provide the following information:
    - **Access key** - Your 64-character Tenable access key.
    - **Secret key** - Your 64-character Tenable secret key.
4. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential.

You're now ready to set up and activate the connection to bring in data from Tenable Vulnerability Management.

Step 3: Choose how to configure the Tenable integration

The Tenable Vulnerability Management integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector

tasks run independently from either the cloud or one of your Explorers, only performing the integration sync. Setting up a connector will work if you're self-hosting runZero or integrating with Tenable Vulnerability Management.

Step 4: Set up and activate the integration to sync data

After you add your Tenable credential, you'll need to sync your data from Tenable Vulnerability Management.

Step 4a: Configure the Tenable integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Tenable-only assets are created.

1. Activate a connection to Tenable Vulnerability Management. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Set the severity and risk levels you want to import (optional). **Note**: Much of the host information provided by Tenable is from Info-level plugins, so if you only import higher levels of severity you may not see much information about assets not scanned by runZero.
4. Set the **Fingerprint only** toggle to `Yes` if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).
5. Optionally provide a list of tags to include in the import. The list should be comma separated and use the format `category:value`. We will import assets that match at least one of the specified tags.
6. To filter by asset source, set the **Filter by source** option to *Yes* and select the target sources.
7. Enter a name for the task, like `Tenable Vulnerability Management sync` (optional).
8. Choose the Explorer to perform this connector task from (optional).
9. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
10. Enter a description for the task (optional).
11. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
12. If you want to include assets that have not been assessed for vulnerabilities, switch the **Include unscanned assets** toggle to *Yes*.
13. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
14. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Tenable integration as a scan probe

You can run the Tenable Vulnerability Management integration as a scan probe so that the runZero Explorer will pull your vulnerability data into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *TENABLE* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Optionally, set the severity and risk levels for ingested vulnerability scan results.
- Set the correct `Tenable` credential to *Yes* in the *Credentials* tab.

Step 5: View Tenable assets and vulnerabilities

After a successful sync, you can go to your inventory to view your Tenable assets. These assets will have a Tenable icon listed in the **Source** column.

The Tenable integration gathers details about vulnerabilities detected in addition to enriching asset inventory data. Go to **Inventory** > **Vulnerabilities** to view the vulnerability data provided by Tenable Vulnerability Management.

To filter by Tenable assets, consider running the following queries:

- View all Tenable assets:

  ```
  source:Tenable
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from the Tenable scan data.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Tenable Vulnerability Management integration unable to connect?

1. Are you getting any data from the Tenable Vulnerability Management integration?
   - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
   - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Tenable Vulnerability Management integration is unable to connect be sure to check the task log for errors. Some common errors include:
   - 500 - server error, unable to connect to the endpoint
   - 404 - hitting an unknown endpoint on the server
   - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the Tenable host if it is on-premises.

How do I solve the following error in Tenable Vulnerability Management:

```
"error-message":"no tenable assets match import criteria",
"level":"error","msg":"could not load scan result data to writer"
```

This is an error we have seen intermittently from Tenable. A solution that usually works is to enable the *Include Unscanned Assets* toggle in the Tenable task configuration. This will disable the filters we apply for live assets that were scanned in the last 30 days.

Nessus Professional

Community Platform

runZero integrates with Nessus Professional by importing data from the Tenable API.

Getting started with Nessus Professional

To set up an integration with Nessus Professional, you'll need to:

1. Create an Administrator API key in an access group with *Can View* permission to *Manage Assets*.
2. Configure the Nessus Professional credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the integration to pull your data into runZero.

Requirements

Before you can set up the Nessus Professional integration:

- Make sure you have administrator access to the Nessus Professional portal.

Step 1: Create an Administrator API key

1. Sign in to Nessus Professional with the Administrator account being used for the runZero integration.
2. Go to **My Profile** > **My Account** > **API Keys**.
3. **Generate** the API token, and then download or copy it.

- You will either need to configure the Tenable credential to skip TLS verification, or provide the TLS thumbprint when creating the runZero credential.

Step 2: Add the Nessus Professional credential to runZero

1. Go to the Credentials page in runZero. Provide a name for the credentials, like `Nessus Professional`.
2. Choose **Nessus Professional Access & Secret** from the list of credential types.
3. Generate your Tenable access and secret keys via your account page in the Tenable portal, and then provide the following information:
    - **Access key** - Your 64-character Tenable access key.

- **Secret key** - Your 64-character Tenable secret key.
- **Nessus API URL** - The API URL for your Nessus Professional instance. The expected format is `https://ip:port` or `https://domain.tld:port`. The default port used by Nessus Professional is `8834`.
- **Nessus insecure** - Set this to *Yes* if you want to attempt authentication without a verified thumbprint.
- **Nessus thumbprints** (optional) - A set of IP=SHA256:B64HASH pairs to trust for authentication.
  - You will need to scan your Nessus instance with runZero in order to obtain the TLS thumbprint. The TLS fingerprints service attribute report lists all previously seen fingerprints.
- If *Nessus insecure* is set to *No* and no thumbprints are provided:
  - With a self-signed certificate, the connection will fail because the certificate chain cannot be verified.
  - With a valid certificate from a public CA, the connection can work without thumbprints.

4. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential.

You're now ready to set up and activate the connection to bring in data from Nessus Professional.

Step 3: Choose how to configure the Nessus Professional integration

The Nessus Professional integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync. Setting up a connector will work if you're self-hosting runZero or your Nessus Professional instance is publicly accessible.

Step 4: Set up and activate the integration to sync data

After you add your Tenable credential, you'll need to sync your data from Nessus Professional.

Step 4a: Configure the Nessus Professional integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Tenable-only assets are created.

1. Activate a connection to Nessus Professional. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Set the severity and risk levels you want to import (optional). **Note**: Much of the host information provided by Tenable is from Info-level plugins, so if you only import higher

levels of severity you may not see much information about assets not scanned by runZero.
4. Set the **Fingerprint only** toggle to *Yes* if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).
5. Enter a name for the task, like `Nessus Professional sync` (optional).
6. Choose the Explorer to perform this connector task from (optional).
7. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.
8. Enter a description for the task (optional).
9. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
10. If you want to include assets that have not been assessed for vulnerabilities, switch the **Include unscanned assets** toggle to *Yes*.
11. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
12. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Nessus Professional integration as a scan probe

You can run the Nessus Professional integration as a scan probe so that the runZero Explorer will pull your vulnerability data into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *NESSUS* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Set the correct *Nessus* credential to *Yes* in the *Credentials* tab.
- Optionally, set the severity and risk levels for ingested vulnerability scan results.

Step 5: View Tenable assets and vulnerabilities

After a successful sync, you can go to your inventory to view your Tenable assets. These assets will have a Tenable icon listed in the **Source** column.

The Tenable integration gathers details about vulnerabilities detected in addition to enriching asset inventory data. Go to Inventory > Vulnerabilities to view the vulnerability data provided by Nessus Professional.

To filter by Tenable assets, consider running the following queries:

- View all Tenable assets:

  ```
  source:Tenable
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from the Tenable scan data.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Nessus Professional integration unable to connect?

1. Are you getting any data from the Nessus Professional integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the Nessus Professional integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the Tenable host if it is on-premises.

How do I solve the following error in Nessus Professional:

- `"error-message":"no tenable assets match import criteria","level":"error","msg":"could not load scan result data to writer"` {#nessuspro-import-error}

This is an error we have seen intermittently from Tenable. A solution that usually works is to enable the *Include Unscanned Assets* toggle in the Tenable task configuration. This will disable the filters we apply for live assets that were scanned in the last 30 days.

Tenable Nessus

Community Platform

runZero integrates with Tenable Nessus using two methods. For all versions of Nessus, runZero can import Nessus files (`.nessus`) that were exported from your Nessus instance. Exports from Tenable Security Center are also supported. For Nessus Professional users, the runZero integration can pull scan data from the Nessus Professional API.

Getting started with Tenable Nessus

To use the Tenable Nessus integration, you'll need to:

1. Export vulnerability scan results as Nessus files.
2. Import the Nessus files through the inventory pages.

Requirements

Before you can set up the Nessus integration:

- Make sure you have access to the Nessus portal.

Step 1: Export vulnerability scan results

1. Sign in to Nessus with the account being used for the runZero integration.
2. Open the scan results you want to be able to import into runZero.
3. Choose **Export** > **Nessus** to download the scan results.

Step 2: Import the Nessus files into runZero

1. Go to the Inventory page in runZero.
2. Choose **Import** > **Nessus scan (.nessus)** from the list of import types.
3. On the import data page:
    - Choose the site you want to add your assets to, and
    - Set the severity levels and minimum risk level to ingest. (**Note**: much of the host information provided by Tenable is from Info-level plugins, so if you only import higher levels of severity you may not see much information about assets not scanned by runZero.)

Step 3: View Nessus assets and vulnerabilities

After a successful sync, you can go to your inventory to view your Nessus assets. These assets will have a Tenable icon listed in the **Source** column.

The Nessus integration gathers details about vulnerabilities detected in addition to enriching asset inventory data. Go to Inventory > Vulnerabilities to view the vulnerability data provided by Nessus.

To filter by Nessus assets, consider running the following queries:

- View all Nessus assets:

  ```
  source:Tenable
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from the Nessus scan file.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Tenable Nessus integration unable to connect?

1.

1. Are you getting any data from the Tenable Nessus integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.

Tenable Security Center

Community Platform

runZero integrates with Tenable Security Center (previously Tenable.sc) by importing data from the Tenable Security Center API.

Getting started with Tenable Security Center

To set up an integration with Tenable Security Center, you'll need to:

1. Create an API key for a user that has access to view and query vulnerabilities in Tenable Security Center.
2. Configure the Tenable Security Center credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the integration to pull your data into runZero.

Requirements

Before you can set up the Tenable Security Center integration:

- Make sure you have administrator access to the Tenable Security Center portal.

Step 1: Create an API key

1. Sign in to Tenable Security Center with an Administrator account.
2. Make sure API key authentication is enabled
3. Go to **Users** > **Users**.
4. Check the box for the user you want to create an API key for. **Note**: The API key will have the same access as the user you select. Make sure the user has access to view and query vulnerabilities in the desired organization.
5. At the top of the table, click the **API Keys** > **Generate API Key** option.
6. Click **Generate** to create the API token, and then download or copy it.

Step 2: Add the Tenable Security Center credential to runZero

1. Go to the Credentials page in runZero. Provide a name for the credentials, like `Tenable Security Center`.
2. Choose **Tenable Security Center Access & Secret** from the list of credential types.

3. Generate your Tenable Security Center API key as directed in Step 1, and then provide the following information:
    - **Access key** - Your 64-character Tenable Security Center access key.
    - **Secret key** - Your 64-character Tenable Security Center secret key.
4. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential.

You're now ready to set up and activate the connection to bring in data from Tenable Security Center.

Step 3: Choose how to configure the Tenable Security Center integration

The Tenable Security Center integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync. If you are integrating with an internal Tenable Security Center instance, we recommend setting up a connector to run from one of your explorers. Otherwise, if you are integrating with an external-facing Tenable Security Center instance, you can set up a connector to run from the cloud. If you are self-hosting runZero, you can run the connector from an explorer or from your runZero host, whichever can reach your Tenable Security Center install.

Step 4: Set up and activate the integration to sync data

After you add your credential, you'll need to sync your data from Tenable Security Center.

Step 4a: Configure the Tenable Security Center integration as a connector task

A connection requires you to specify a schedule which determines when the sync occurs.

1. Activate a connection to Tenable Security Center. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Configure the *Tenable Security Center query mode* setting (optional).
    - Select *Define filters* to define a filter based on vulnerability severity and risk level. **Note**: Much of the host information provided by Tenable is from Info-level plugins, so if you only import higher levels of severity you may not see much information about assets.
    - Select *Use existing query ID* to provide the Tenable Security Center query to use. **Note**: The query must be the *Vulnerability* type and use the *Vulnerability Detail List* tool.
4. Set the **Fingerprint only** toggle to *Yes* if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).
5. Enter a name for the task, like `Tenable Security Center sync` (optional).
6. Choose the Explorer to perform this connector task from (optional).
7. Choose the site you want to configure the connector for.

8. Enter a description for the task (optional).
9. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
10. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Tenable Security Center integration as a scan probe

You can run the Tenable Security Center integration as a scan probe so that the runZero Explorer will pull your vulnerability data into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *TENABLESECURITYCENTER* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Optionally, set the severity and risk levels for ingested vulnerability scan results or provide a query ID.
- Set the correct `TenableSecurityCenter` credential to *Yes* in the *Credentials* tab.

Step 5: View Tenable Security Center assets and vulnerabilities

After a successful sync, you can go to your inventory to view your Tenable Security Center assets. These assets will have a Tenable icon listed in the **Source** column.

The Tenable Security Center integration gathers details about vulnerabilities detected in addition to enriching asset inventory data. Go to **Inventory** > **Vulnerabilities** to view the vulnerability data provided by Tenable Security Center.

To filter by Tenable Security Center assets, consider running the following queries:

- View all Tenable Security Center assets:

  ```
  source:tenablesecuritycenter
  ```

- View all Tenable Security Center vulnerabilities:

  ```
  source:tenablesecuritycenter
  ```

Click into each asset or vulnerability to see its individual attributes. runZero will show you the attributes gathered from the Tenable Security Center API.

VMware

Community Platform

runZero Platform supports synchronization of VMware vCenter and ESXi virtual machine inventories.

Setting up VMware credentials

Unlike other APIs, the VMware synchronization process is configured as part of your regular runZero Explorer scans. The first step is to set up a set of VMware credentials.

On the Scanning with credentials page, click *Add Credential* and choose a credential type of *VMware vCenter/ESXi Username and Password*, and enter the appropriate username and password. The correct username syntax in most cases is `user@domain.com`. The VMware account used requires at least **read-only** access.

The CIDR allow list field can be used to limit which addresses the credentials should be sent to. This helps ensure that they are not passed to unexpected VMware systems that the runZero Explorer encounters on the network.

If runZero has previously found VMware API endpoints, the optional *VMware thumbprints* field will list their IP addresses and TLS fingerprints. You can edit this list to remove any systems you do not want to trust with your VMware credentials. Alternatively, if you do not want to limit authentication to the set list of IP addresses and TLS certificates, you can set *VMware insecure* to *Yes* to approve authenticating with untrusted endpoints.

The organization access for the credentials can be set as for any other stored credentials; see Scanning with credentials.

Performing VMware synchronization

Once you have defined a set of VMware credentials, the second step is to enable VMware synchronization as part of a scan task. Any task which includes scanning the VMware host systems can be used to synchronize VMware VM data.

The Probes tab of the scan setup has a section for enabling and disabling the VMware probe. The probe must be enabled for VMware synchronization to work; it is enabled by default.

On the Credentials tab of the scan setup, use the toggle switch to enable the appropriate set of VMware vCenter/ESXi credentials.

When the scan runs, the Explorer will use the credentials to authenticate with any VMware ESXi or vCenter hosts it finds that the credentials are configured to trust. Data about assets which are VMware VMs will be imported into runZero automatically, and merged with the other information runZero finds by scanning.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the VMware integration unable to connect?

1.

1. Are you getting any data from the VMware integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
3. If the VMware integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue
4. Verify you are running the integration task from an Explorer with access to the VMware host if it is on-premises.

Wiz

Community Platform

runZero integrates with Wiz by importing data from the Wiz API. This integration allows you to sync data about your cloud assets, software, and vulnerabilities from Wiz to provide better visibility of your cloud assets and security posture.

Getting started with Wiz

To set up an integration with Wiz, you'll need to:

1. Create a Service Account in Wiz with permissions to read graph resources, read reports, and create reports.
2. Configure the Wiz credential in runZero.
3. Choose whether to configure the integration as a scan probe or connector task.
4. Activate the integration to pull your data into runZero.

Requirements

Before you can set up the Wiz integration:

- Make sure you have administrator access to the Wiz portal.

Step 1: Create a Service Account in Wiz

1. Sign in to Wiz with an Administrator account.
2. Go to **Settings** > **Access Management** > **Service Accounts** > **Add Service Account**.
3. Enter a descriptive name in the **Name** field.
4. Select *Custom Integration (GraphQL API)* for the **Type**.
5. Leave the **Projects** section blank so the Service Account has access to all projects.

6. Enable `read:resources`, `read:reports`, and `create:reports` for the **API Scopes**.
7. Click **Add Service Account** and copy the **Client ID** and **Client Secret**.
8. Go to **User Settings** > **Tenant** and note the **API Endpoint URL** in the format: `https://api.{{region}}.app.wiz.io/`.

Step 2: Add the Wiz credential to runZero

1. Go to the Credentials page in runZero. Provide a name for the credentials, like `Wiz`.
2. Choose **Wiz Client Secret** from the list of credential types.
3. Create your Wiz service account via the settings page in the Wiz portal, and then provide the following information:
    - **Wiz Client ID** - The client ID of your Wiz service account.
    - **Wiz Client Secret** - The client secret of your Wiz service account.
    - **Wiz Auth URL**- The URL used to authenticate the Wiz service account.
    - **Wiz API URL** - The API Endpoint URL used to access the Wiz API.
4. If you want other organizations to be able to use this credential, select the *Make this a global credential* option. Otherwise, you can configure access on a per-organization basis.
5. Save the credential.

You're now ready to set up and activate the connection to bring in data from Wiz.

Step 3: Choose how to configure the Wiz integration

The Wiz integration can be configured as either a scan probe or a connector task. Scan probes gather data from integrations during scan tasks. Connector tasks run independently from either the cloud or one of your Explorers, only performing the integration sync.

Step 4: Set up and activate the integration to sync data

After you add your Wiz credential, you'll need to sync your data from Wiz.

Step 4a: Configure the Wiz integration as a connector task

A connection requires you to specify a schedule and choose a site. The schedule determines when the sync occurs, and the site determines where any new Wiz-only assets are created.

1. Activate a connection to Wiz. You can access all available third-party connections from the integrations page, your inventory, or the tasks page.
2. Choose the credentials you added earlier. If you don't see the credentials listed, make sure the credentials have access to the organization you are currently in.
3. Set the severity and risk levels you want to import (optional).
4. Set the **Fingerprint only** toggle to *Yes* if you want vulnerability records to be ingested for fingerprint analysis but not stored in your runZero vulnerability inventory (optional).
5. Enter a name for the task, like `Wiz Sync` (optional).
6. Choose the Explorer to perform this connector task from (optional).
7. Choose the site you want to add your assets to. All newly discovered assets will be stored in this site.

8. Enter a description for the task (optional).
9. If you want to exclude assets that have not been scanned by runZero from your integration import, switch the **Exclude unknown assets** toggle to *Yes*. By default, the integration will include assets that have not been scanned by runZero.
10. If you want to exclude assets that have not been assessed for vulnerabilities, switch the **Include unscanned assets** toggle to *No*.
11. Schedule the sync. A sync can be set to run on a recurring schedule or run once. The schedule will start on the date and time you have set.
12. Activate the connection when you are done. The sync will run on the defined schedule. You can always check the Scheduled tasks to see when the next sync will occur.

Step 4b: Configure the Wiz integration as a scan probe

You can run the Wiz integration as a scan probe so that the runZero Explorer will pull your Wiz assets into the runZero Console.

In a new or existing scan configuration:

- Ensure that the *WIZ* option is set to *Yes* in the *Probes and SNMP* tab and change any of the default options if needed.
- Optionally, set the severity and risk levels for ingested vulnerability results.
- Set the correct *Wiz* credential to *Yes* in the *Credentials* tab.

Step 5: View Wiz assets, software, and vulnerabilities

After a successful sync, you can go to your inventory to view your Wiz assets. These assets will have a Wiz icon listed in the **Source** column.

The Wiz integration gathers details about software and vulnerabilities detected in addition to enriching asset inventory data. Go to **Inventory** > **Software** or **Inventory** > **Vulnerabilities** to view the software and vulnerability data provided by Wiz.

To filter by Wiz assets, consider running the following queries:

- View all Wiz assets:

  ```
  source:Wiz
  ```

Click into each asset to see its individual attributes. runZero will show you the attributes gathered from Wiz.

Troubleshooting

If you are having trouble using this integration, the questions and answers below may assist in your troubleshooting.

Why is the Wiz integration unable to connect?

1.

1. Are you getting any data from the Wiz integration?
    - Make sure to query the inventory rather than look at the task details to review all the data available from this integration.
    - In some cases, integrations have a configuration set that limits the amount of data that comes into the runZero console.
2. Some integrations require very specific actions that are easy to overlook. If a step is missed when setting up the integration, it may not work correctly. Please review this documentation and follow the steps exactly.
    - Double-check the API Scopes assigned to the Wiz service account. A valid service account that is missing the required permissions will result in a failed import.
3. If the Wiz integration is unable to connect be sure to check the task log for errors. Some common errors include:
    - 500 - server error, unable to connect to the endpoint
    - 404 - hitting an unknown endpoint on the server
    - 403 - not authorized, likely a credential issue

Outbound integrations

Using runZero data to enrich other tools

In addition to being able to enrich your runZero inventory with data from your other IT and security tools, the runZero platform offers egress integrations with several platforms. By leveraging product APIs and export/import functionality, runZero can provide additional asset context in other IT and security tools.

The following integrations are available to send your runZero data into other platforms:

IT service management

- Atlassian Insight & Jira Service Management
- ServiceNow CMDB

Detection and investigation

- Panther
- Splunk Search
- Sumo Logic
- Tines
- Thinkst Canary

Vulnerabilities and risk

- SecurityGate.io

Atlassian Insight & Jira Service Management

All runZero editions integrate with Jira Service Management via an import in Atlassian Insight. runZero asset data is then imported into the CMDB.

Follow these steps to perform a basic import.

Step 1: Export runZero asset data

You can export data using the **Export** button from the runZero inventory or the Export API.

The following are sample commands for the export API that include common export fields but omit the tags field. You must replace the token `ETxxx...` with your account's export token from the Inventory export API page.

For a CSV export, use this command (one line):

```
curl -o "assets.csv" -H "Authorization: Bearer ETxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  "https://console.runzero.com:443/api/v1.0/export/org/assets.csv"
```

For a JSON export, use this command (one line):

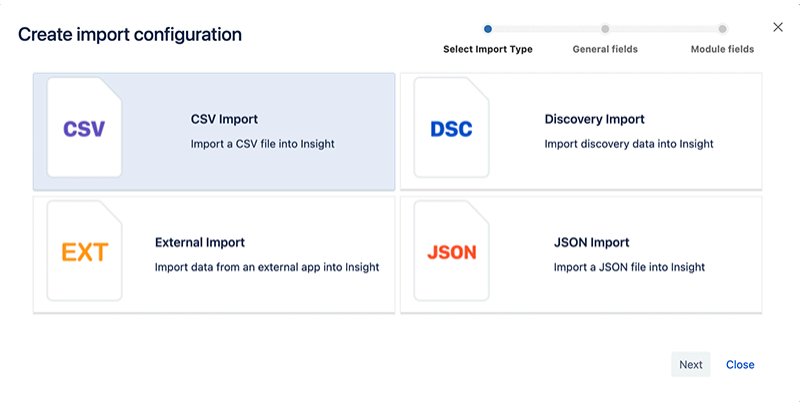
```
curl -o "assets.json" -H "Authorization: Bearer ETxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
  "https://console.runzero.com:443/api/v1.0/export/org/assets.csv"
```

> **Atlassian Insight** does not accept any attribute longer than 255 characters and will produce errors if you are trying to import a file that contains longer fields. If you are seeing such errors, you may need to either trim or omit the fields.

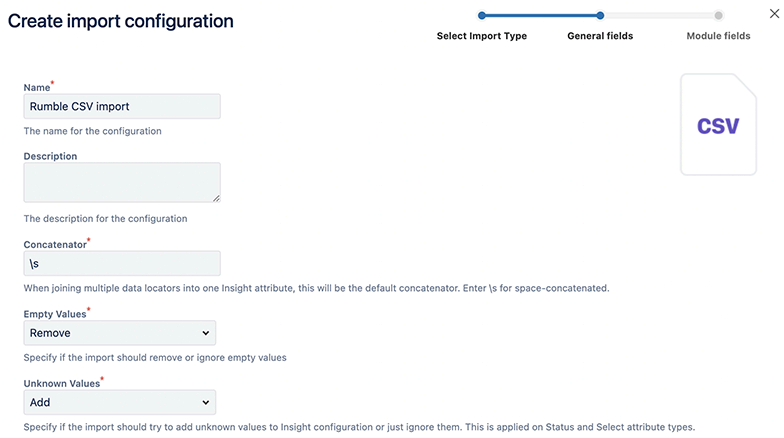
Step 2: Import data into Atlassian Insight

You must use Jira Service Management Premium or Enterprise and have Atlassian Insight installed for the following steps to work.

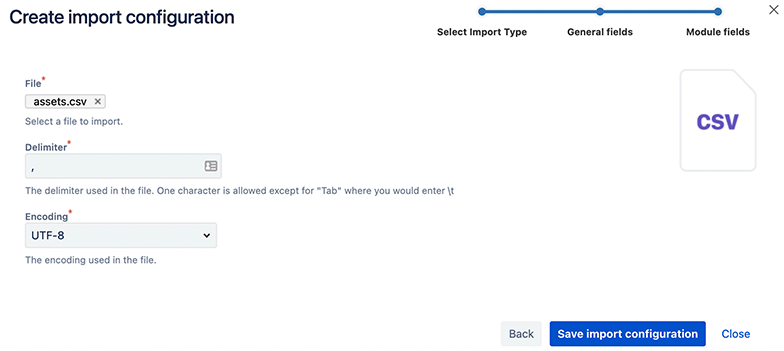
1. In Jira Service Management, go to the Insight menu and click the **+** sign to create an object schema.
2. Name the object schema. In our example, we're calling it `Assets`.
3. Click on **...** next to the object schema and choose **Configuration**.
4. Go to the **Import** tab.
5. Click **Create Configuration**.



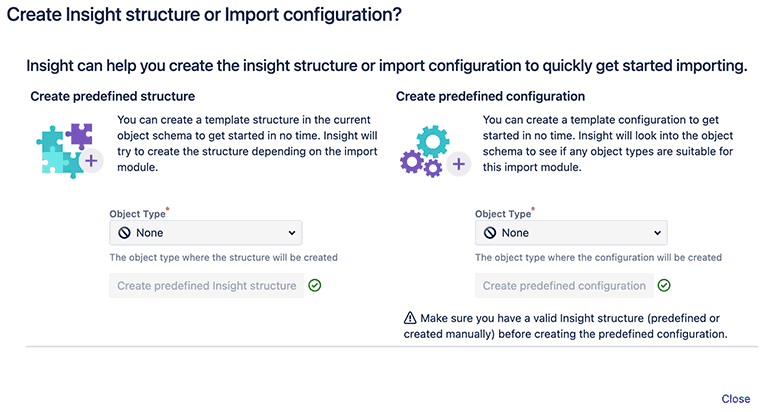
6. Choose CSV or JSON, depending on your previous export and click **Next**.
7. Enter the name for your import.
8. In the **Concatenator** field, enter `\s`.



9. Scroll down and click **Next**.
10. Choose the file you have previously exported form runZero.

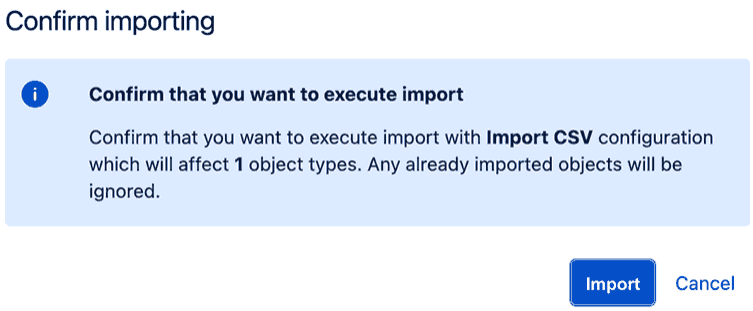


11. Click **Save Import Configuration**.
12. Click **Create predefined Insight structure** and wait for the task to finish.
13. Click **Create predefined configuration** and wait for the task to finish.

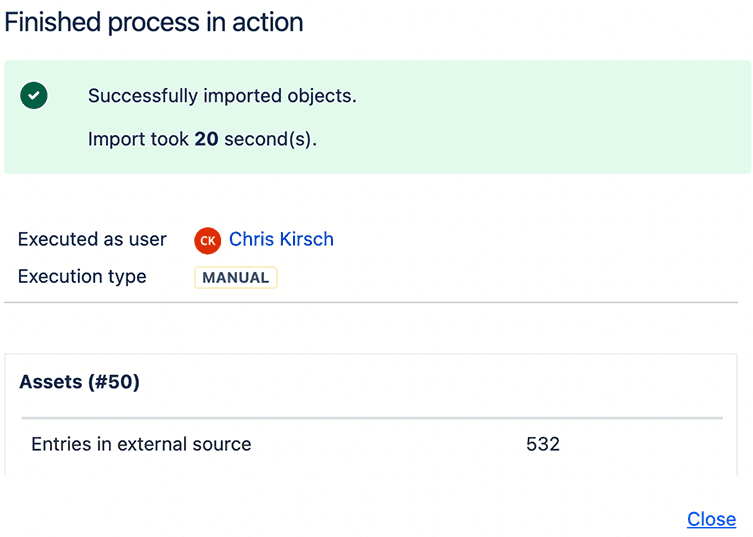


14. Close the window. You will see a new entry with your configured import on the page. Click the settings cog and select **Execute Import**.

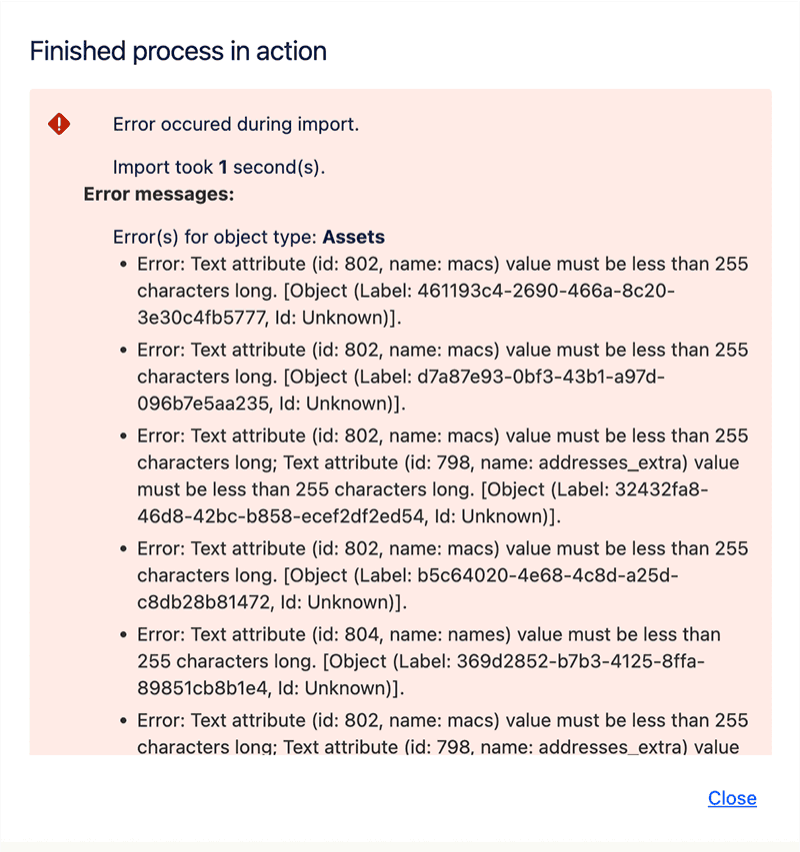
15. Click **Import**.



16. Your import was successful. Close the window.

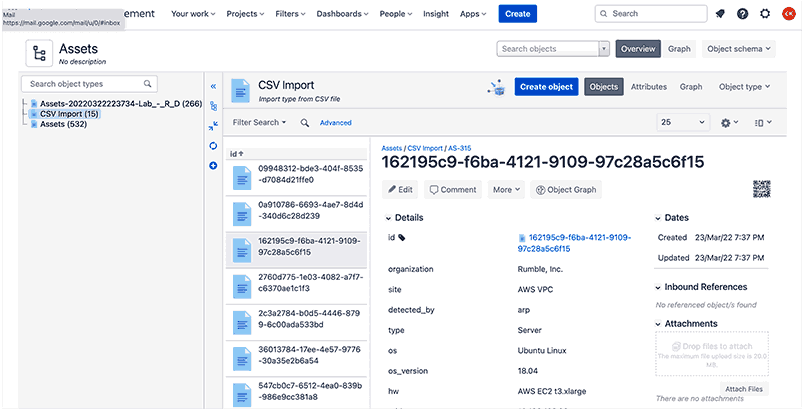


If you are seeing the following error, one of your fields is longer than 255 characters. Manually trim the field to 255 characters or delete the field to successfully import your assets.



Step 3: View your imported assets

1. In Jira Service Management, click on the Insights.
2. Click on your object schema.
3. Click on assets to view details.



Panther

runZero data can be imported into your Panther instance for enhanced logging and alerting.

Requirements

- A Panther account with the required permissions,
- An AWS S3 bucket, and
- Exported .jsonl files from runZero that have been uploaded into your AWS S3 bucket.

Step 1: Adding a custom schema

1. Go to **Configure > Schemas** and select **Create New**.
2. Add a name.
3. Upload a sample log to automatically parse the runZero output schema.

Step 2: Adding a custom log source

1. Go to **Configure > Log Sources** and select **Create New**.
2. Complete the **Basic Information** section.
3. Opt to configure S3 prefixes and schemas now and select the custom schema you created.
4. Configure the IAM role:
    - Opt to configure **Using the AWS Console UI**.
    - Click **Launch Console UI**.
    - Review the stack in AWS, then check the box to approve, and click to deploy the stack.
    - When the deployment completes, navigate to the **Resources** tab and select the **LogProcessingRole** that was created.
    - Copy the ARN from that role into the field on the Panther console.
5. Configure an alarm if logs are not processed (optional).

Once completed, any .jsonl files added to the specified AWS S3 bucket will be automatically ingested and processed by Panther.

SecurityGate.io

All runZero editions integrate with SecurityGate.io to enrich asset visibility in support of your risk assessment program. Setting up the integration requires a few steps in your SecurityGate.io console.

Requirements

- Configuring the SecurityGate.io integration requires a runZero API key.

The SecurityGate.io integration will pull runZero asset data from across all organizations.

Integrate runZero with SecurityGate.io

1. Sign in to your SecurityGate.io console.
2. Go to **My Account** under the **Hello** dropdown menu.
3. Click on **Integration Manager**.
4. Select **Add New Integration**.
5. Choose *Rumble* from the **Integration Partner** dropdown menu.
6. Provide the **Account API Key**, **Server URL**, and **API Version**.
7. Click **Test Connection**. If the test is successful, click **OK** to save the configuration.

Viewing runZero data in SecurityGate.io

After the integration is enabled within SecurityGate.io, runZero data will be available through the Asset Inventory page.

ServiceNow Service Graph

Platform

The Service Graph connector for runZero allows you to bring runZero assets into your ServiceNow CMDB as CIs, and optionally periodically update the CIs with fresh information from runZero scans.

The Service Graph Connector fetches and transforms data using ServiceNow IntegrationHub ETL, and passes it through the Identification and Reconciliation Engine (IRE). This allows specific fields and CI class mappings to be fine-tuned from the ServiceNow console. You can also specify a runZero search query to determine which assets get brought in by the connector.

Important notes

- There is no charge from runZero for use of the connector. However, CI resources created in CMDB by the connector will increase ServiceNow Subscription Unit consumption. CIs created by Service Graph Connectors should be charged at a lower rate than CIs created via other means; you will need to consult ServiceNow to confirm this and obtain current pricing.
- If you are a self-hosted runZero customer, you will need to set up a MID Server to enable ServiceNow to connect to your runZero console. If your console uses a self-signed TLS certificate, or a TLS certificate signed by your own internal CA, you will need to configure the MID server to accept the CA or certificate as trusted. For more information, see ServiceNow Knowledge Base article KB0863673.

- While the Service Graph Connector is packaged and available from the ServiceNow Application Store, setting up connectors to integrate with your existing ServiceNow configuration is unlikely to be a one-click operation. You are strongly advised to have a ServiceNow Consulting and Implementation partner available to assist with the process.
- You can find the Scoped Application Installation and Configuration Guide on the Service Graph Connector for runZero page under Supporting Links and Docs. This has detailed installation instructions with screenshots.

Prerequisites

To use the Service Graph connector for runZero, you need the following:

- A Platform license for runZero.
- A ServiceNow ITOM license including ITOM Discovery and ITOM Visibility.
- Integration Commons for CMDB installed.

The ServiceNow dependencies are checked as part of the connector installation process.

Installation

The first step of the installation is to locate the Service Graph connector for runZero in the ServiceNow Application Store, and follow the usual process to install it into your ServiceNow instance.

The Service Graph connector adds a new menu entry **Service Graph connector for runZero** to the ServiceNow main menu. The module contains entries for **Setup**, **Data Sources**, **Scheduled Imports**, **Support**, and **System Import Sets**. You must have the **admin** role in ServiceNow to configure the connector.

Setup

The **Setup** menu offers a **Guided Setup** process to help you get up and running with the connector. There are three main stages to the setup process:

1. Configure the connection
2. Configure the mappings (optional)
3. Confgure the scheduled import (optional)

Step 1: Configure the connection

Configuring the connection consists of two steps: entering a runZero API key as credentials, and configuring connection options.

Entering the API key

The connector can use a runZero **Organization API** or **Export API** key. In either case, you can obtain the key by going to the **Organizations** page in runZero and clicking on the organization

containing the assets you want to bring into ServiceNow CMDB. Organization tokens begin with `OT`, export tokens begin with `ET`.

In the **API Key Credentials** form, paste the runZero token into the field labeled **API Key**, and click **Update**.

Configuring connection options

For runZero cloud users, the default **HTTP connection** options should be correct.

You can optionally add a **Search** string to the **Attributes** for the connection at the bottom of the form. If you do, this search will be passed to runZero to determine the set of assets to bring in to CMDB via the connector. Any search string should be in runZero search query format. You are strongly encouraged to test your search string in your runZero asset inventory.

The **Connection** alias and **Base path** fields should be left as-is. (The connection alias groups together the connection settings and the API key credentials for the connector to use.)

If you are self-hosting runZero, the **connection options** form is where you configure the MID server to use and specify the URL of your runZero console. Your console's fully-qualified hostname should be placed in the **Host** field. The hostname should match the hostname in the console's TLS certificate in order for the MID server to trust it, even if the TLS certificate is trusted by the MID server. Check the **Use MID server** box and select the MID server to use.

Step 2: Configure the mappings (optional)

The second step of setting up the connector allows you to customize the mapping of runZero data to CI classes, using the IntegrationHub ETL Transform Map Assistant.

This stage can be skipped if you do not have special requirements for data mapping.

Step 3: Configure the Scheduled Import (optional)

The third step of setting up the connector is to set up a schedule for data import. Initially, you will want to test the connector setup with a one-off import. Once everything is configured correctly, you can create recurring imports.

Scheduled imports are configured as regular ServiceNow tasks. This step can be skipped if you don't want to set up a scheduled import at this time.

Notes on data mapping

The object data models for ServiceNow CMDB and runZero are not an exact match. Some runZero data does not fit in any standard attributes available in CMDB, and in other cases runZero does not have data that CMDB expects. Review this list to understand how the connector will map runZero attributes to ServiceNow CMDB attributes:

- IP addresses: ServiceNow CIs have an attribute which takes a single IP address. The connector places the first IP address in the CI attribute IP Address. The remaining

addresses are created as `IP Address` CIs owned by the device CI. Each `IP Address` CI is given a `Description` indicating whether it has been scanned by runZero or not. **Note**: By default, ServiceNow does not show IP addresses owned by a device CI. It only shows IP addresses owned by a network adapter CI which is owned by the device CI. This is a known limitation of ServiceNow's default CMDB forms. runZero doesn't associate IP addresses with network adapters because the necessary information isn't generally known. In that situation, the CMDB connector development guide states that the IP addresses should be associated with the device CI.

- **Names**: ServiceNow CIs have a single `Name` attribute, while runZero assets can have any number of names with runZero attempting to guess which are the best names and put them at the front of the list. The connector places the first name in the CI `Name` attribute. The first name which looks like a fully-qualified domain name is placed in the `Fully qualified domain name` attribute, and is used to compute a DNS Domain for CIs descended from Network Gear. The full list of names is placed in the `Description` attribute for reference.
- **Serial numbers**: ServiceNow CIs have an attribute which takes a single serial number. The connector picks the first serial number for the `Serial Number` attribute, and then also performs a Serial Number Lookup operation to add the complete list of serial numbers with their associated type.
- **Tags**: runZero assets can have any number of tags, which are either single tags such as `tag_name`, or tags with values in the format `tag=value`. These are placed in the `Key Value` class, and this class is then associated with the `Hardware` class.
- **MAC addresses/Network Adapters**: As with IP addresses, ServiceNow has both a single-value `MAC Address` attribute, and an option to look up named MAC addresses to create Network Adapter objects. A runZero asset can have any number of MAC addresses, but since runZero does not run agent software on the scanned systems, it has no way to know the interface name associated with a given MAC address. The first MAC address is imported as the `MAC Address` attribute, and the entire set is imported via lookup as `Network Adapter` objects, with each being given a name made up by the MAC address and IP address (e.g. `00:00:00:00:00:00-127.0.0.1`) or the MAC address and hostname if the IP address is not available (e.g. `00:00:00:00:00:00-hostname`). If neither the IP address or hostname are available, the MAC address alone is used as the network adapter name. In some cases, runZero may not know the MAC address(es) of an asset. In this case, a zero MAC address (`00:00:00:00:00:00`) is added to satisfy ServiceNow requirements for some CI classes.
- **Sites**: runZero sites represent distinct networks, which may or may not correspond to physical sites. They are imported as CMDB `Network Site` objects.
- **Organizations**: The runZero organization has no obvious corresponding attribute in network hardware CIs. The organization name is incorporated into the `Description` of CIs.
- **SNMP data**: The `snmp.sysDesc` attribute from runZero, if it exists, is also placed in the `Description`.

Splunk Search

Community Platform

runZero integrates with Splunk using a dedicated Splunk Addon, compatible with Splunk 7, Splunk 8, and Splunk Cloud. With this add-on, you'll be able to pull new or updated hosts into a Splunk index, where you'll be able to analyze, visualize, and monitor them there.

This add-on uses the Splunk API from the runZero Network Discovery platform. It supports syncing assets into Splunk, with multiple inputs supported, global API key management, and optional search filters for each input. For example, you can track new assets as one input, and SMBv1 enabled assets as another input.

To set up this add-on, you'll need an Export API or Organization API key, which you can generate from your Organization page in the runZero Console.

Get the runZero add-on for Splunk

1. Sign in to Splunk.
2. Go to **Find More Apps**.
3. Search for `runZero Network Discovery`.
4. Install the add-on for runZero.
5. Splunk will prompt you to sign in again. After you log back in again, the add-on will be installed. You'll be able to open the runZero Asset Sync app. Splunk might also prompt you to restart your server.

Asset sync modes

Two asset sync modes are available: New Assets Only and All Updated Assets. You can export asset inventory that contains newly discovered assets or updated assets, since the last poll, in a sync-friendly format for Splunk. You can leverage the same capabilities from the Asset Sync API to pull data in Splunk, such as search filters, fields, and time-based checkpoints.

Once data is pulled into Splunk, you can create Splunk inputs with filters. This allows you to sync specific assets with a certain protocol, discovery date, or open service.

Sumo Logic

All runZero editions integrate with Sumo Logic to enrich asset visibility and help you visualize your asset data. Setting up the integration requires a few steps in your Sumo Logic console. The integration can be set up to support two distinct purposes:

- Complete asset visibility
- Targeted alerting and visualization

Requirements

- A Sumo Logic account
- A runZero account and API key.

Sumo Logic asset export

runZero integrates with Sumo Logic to make your asset inventory available directly in Sumo Logic. This article will show you how to export your runZero inventory into Sumo Logic for use within the SIEM.

Integrating runZero with Sumo Logic

Setting up the connection between Sumo Logic and runZero has three options with different configuration steps.

Option A: Local script

1. Create a Sumo Logic HTTP Source.
2. Configure your host to run the provided script.

Option B: AWS Lambda function

1. Create a Sumo Logic HTTP Source.
2. Configure the AWS Lambda function to run the provided script.

Option C: Sumo Logic script source

1. Install a Sumo Logic collector.
2. Create a Sumo Logic script source.

Once your data is flowing into Sumo Logic, you can start using the data in Sumo Logic.

Option A: Local script

Step 1: Create a Sumo Logic HTTP Source

1. After logging in to Sumo Logic, navigate to **Manage Data** > **Collection**.
2. Click **Add Collector** then select **Hosted Collector**.
    - Provide a name, such as `runZero Collector` and click **Save**.
3. If prompted to add a data source, click **OK**. Otherwise, find your Collector in the list and click **Add Source**.
4. Select the **HTTP Logs and Metrics** source.
    - Provide a name, such as `runZero Assets`, then click **Save**.
5. Copy the URL provided to use in step 2.

Step 2: Configure your host to run the provided script

1. Identify the host you would like to run the script from.

2. Ensure the host has Python3 and Pipenv installed.

3. Save the script below to the host it will be run from.

```
#!/usr/bin/env python3
import json
import requests
import os

# RUNZERO CONF
RUNZERO_EXPORT_TOKEN = os.environ["RUNZERO_EXPORT_TOKEN"]
HEADERS = {"Authorization": f"Bearer {RUNZERO_EXPORT_TOKEN}"}
BASE_URL = "https://console.runZero.com/api/v1.0"

# SUMO LOGIC CONF
HTTP_ENDPOINT = os.environ["SUMO_HTTP_ENDPOINT"]


def main():
    url = BASE_URL + "/export/org/assets.json"
    assets = requests.get(url, headers=HEADERS)
    batchsize = 500
    if len(assets.json()) > 0 and assets.status_code == 200:
        for i in range(0, len(assets.json()), batchsize):
            batch = assets.json()[i:i+batchsize]
            f = open("upload.txt", "w")
            f.truncate(0)
            for a in batch:
                json.dump(a, f)
                f.write("\n")
            f.close()
            r = open("upload.txt")
            requests.post(HTTP_ENDPOINT, data=r.read())
            r.close()
    else:
        print(f"No assets found - status code from runZero API: {assets.status_code}")

if __name__ == "__main__":
    main()
```

4. Create your environment variables by running the following commands:
    - `export RUNZERO_EXPORT_TOKEN=XXX`: Use your runZero export API token, which can be obtained in your runZero console on an organization detail page. Select the organization you wish to export data from, then click **Edit organization** to view the export API token.
    - `export SUMO_HTTP_ENDPOINT=XXX`: Use the Sumo Logic token obtained in step 1.
5. Create your virtual environment to run the script by running `pipenv --python /path/to/python3`.
6. Install the `requests` library in your virtual environment for making API calls:
    - `pipenv shell`
    - `pip install requests`

7. Test the script by running your script from the virtual environment.
    - Use the location from the `pipenv` output to start.
    - Append `/bin/python3` to use Python in the virtual environment.
    - Use the full path to the script.

```
my-server:~/ $ /home/user/.local/share/virtualenvs/runZero-scripts-mVQtFLDO/bin/python3 \
  /home/user/scripts/script.py
```

8. Configure a crontab task to run at the desired cadence.
    - On the hour: `0 * * * * RUNZERO_EXPORT_TOKEN=XXX SUMO_HTTP_ENDPOINT=XXX /path/to/virtual/env/python3 /path/to/script.py`
    - Daily at midnight: `0 0 * * * RUNZERO_EXPORT_TOKEN=XXX SUMO_HTTP_ENDPOINT=XXX /path/to/virtual/env/python3 /path/to/script.py`
    - Weekly at midnight on Monday: `0 0 * * 1 RUNZERO_EXPORT_TOKEN=XXX SUMO_HTTP_ENDPOINT=XXX /path/to/virtual/env/python3 /path/to/script.py`

Option B: AWS Lambda function

Step 1: Create a Sumo Logic HTTP Source

1. After logging in to Sumo Logic, go to **Manage Data** > **Collection**.
2. Click **Add Collector** then select **Hosted Collector**.
    - Provide a name, such as `runZero Collector` and click **Save**.
3. If prompted to add a data source, click **OK**. Otherwise, find your Collector in the list and click **Add Source**.
4. Select the **HTTP Logs and Metrics** source.
    - Provide a name, such as `runZero Assets`, then click **Save**.
5. Copy the URL provided to use in step 2.

Step 2: Configuring the AWS Lambda function to run the provided script

1. Go to your AWS Console and navigate to the Lambda page.
2. Click **Create a function**.
3. Give your function a **name**.
4. Select *Python 3.9* as the **runtime**.
5. Everything else can be left with the default setting. Click **Create function** to move to the next page.
6. Click **Add Trigger** to set up a cron job.
7. Select *EventBridge* to set up a schedule.
8. Use an existing rule or select **Create new rule**.
    - Give it a name and set **Rule type** to `Schedule expression`.

    - Use one of these options or create your own based on desired cadence:
        - Daily: `rate(1 day)`
        - Every 12 hours: `rate(12 hours)`
        - Every 3 hours: `rate(3 hours)`
    - Click **Add** to return to the main Lambda configuration page.

9. Under **Configuration** select *Environment variables*.
10. Enter these two environment variables:
    - `RUNZERO_EXPORT_TOKEN` which can be obtained in your runZero console on an organization detail page. Select the organization you wish to export data from, then click **Edit organization** to view the export API token.
    - `SUMO_HTTP_ENDPOINT` which was obtained in step 1.
11. Click *Save* to return to the main Lambda configuration page.
12. Click the *Code* tab and replace the default code with this script.

```
import json
import urllib3
import os

# RUNZERO CONF
RUNZERO_EXPORT_TOKEN = os.environ["RUNZERO_EXPORT_TOKEN"]
HEADERS = {"Authorization": f"Bearer {RUNZERO_EXPORT_TOKEN}"}
BASE_URL = "https://console.runZero.com/api/v1.0"

# SUMO LOGIC CONF
HTTP_ENDPOINT = os.environ["SUMO_HTTP_ENDPOINT"]


def lambda_handler(event, context):
    http = urllib3.PoolManager()
    url = BASE_URL + "/export/org/assets.json"
    response = http.request("GET", url, headers=HEADERS)
    data = response.data
    assets = json.loads(data)
    batchsize = 500
    if len(assets.json()) > 0 and assets.status_code == 200:
        for i in range(0, len(assets.json()), batchsize):
            batch = assets.json()[i : i + batchsize]
            f = open("upload.txt", "w")
            f.truncate(0)
            for a in batch:
                json.dump(a, f)
                f.write("\n")
            f.close()
            r = open("upload.txt")
            http.request("POST", HTTP_ENDPOINT, data=r.read())
            r.close()
    else:
        print(f"No assets found - status code from runZero API: {assets.status_code}")
```

13. Click **Deploy** to update the code.
14. Click **Test** to verify the code works.

Your asset data export will now be posted to Sumo Logic at the cadence you configured.

Option C: Sumo Logic script source

Step 1: Installing a Sumo Logic collector

Follow the Sumo Logic documentation in order to install a collector.

Step 2: Creating a Sumo Logic script source

Sumo Logic has documentation on script sources as well. Here are the steps to follow to set up the script source once your collector is installed.

1. Navigate to the Collection page in Sumo Logic.
2. Find your collector and click **Add** > **Add Source**.
3. Select `Script` as the **source type**.
4. Input a `Name` and `Source Category`.
5. Select a `Frequency`.
6. Select **Command** type `/usr/bin/python`.
7. Add the following script in the **Script** field.

```
#!/usr/bin/python
import json
import requests
import os

# RUNZERO CONF
RUNZERO_EXPORT_TOKEN = os.environ['RUNZERO_EXPORT_TOKEN']
HEADERS = {'Authorization': 'Bearer ' + RUNZERO_EXPORT_TOKEN}
BASE_URL = 'https://console.runZero.com/api/v1.0'


def main():
    url = BASE_URL + '/export/org/assets.json'
    assets = requests.get(url, headers=HEADERS)
    if assets.status_code == 200:
        for a in assets.json():
            print(json.dumps(a))
    else:
        print(f"No assets found - status code from runZero API: {assets.status_code}")
```

```
if __name__ == '__main__':
    main()
```

8. Click **Save** to allow the source to start working.

Working with the asset data in Sumo Logic

Once your asset data in in Sumo Logic, you can use it in any way you would use any other log source. Here are some sample searches that you could use to create scheduled searches and dashboards.

Search distinct assets

```
_sourceCategory="runzero"
| json field=_raw "id"
| count_distinct(id) as distinct_assets
```

Search assets with more than 3 services running

```
_sourceCategory="runzero"
| json field=_raw "addresses_extra"
| json field=_raw "addresses"
| json field=_raw "id"
| concat("https://console.runzero.com/inventory/", id) as runzero_link
| json field=_raw "service_count"
| where service_count > 3
| count addresses, addresses_extra, service_count,  runzero_link
```

Determine counts of different operating systems

```
_sourceCategory="runzero"
| json field=_raw "os"
| where !isEmpty(os)
| json field=_raw "id"
| count os, id
| count os
```

Sumo Logic alerting

runZero integrates with Sumo Logic to help you visualize your asset data. This helps you track your progress on reducing risk in your asset inventory over time.

Setting up the connection between Sumo Logic and runZero requires:

1. Creating a Sumo Logic HTTP Source
2. Creating a runZero alert template
3. Creating a rule in runZero
4. Handling runZero data in Sumo Logic
5. Creating a Sumo Logic dashboard (optional)

Step 1: Create a Sumo Logic HTTP Source

1. After logging in to Sumo Logic, navigate to **Manage Data** > **Collection**.
2. Click **Add Collector** select **Hosted Collector**, provide a name, such as `runZero Collector` and click **save**.
3. If prompted to add a data source, click **OK**. Otherwise, find your Collector in the list and click **Add Source**.
4. Select the **HTTP Logs and Metrics** source, provide a name, such as `runZero Alerts`, and then click **save**.
5. Copy the URL provided to use in step 2.

Step 2: Create a runZero alert template

1. Create an alert template in runZero and provide the following details:
    - **Name**: Name for template
    - **Template type**: JSON
    - **Subject line for message**: Leave empty
    - **Body of message**: The following JSON example will include the rule name and the search URL in the alert message body

      ```
      {"rule_name":"{{rule.name}}","search_url":"{{search.url}}","found": "{{search.found}}",
      "assets_new": "{{scan.assets_new}}"}
      ```

2. Create an alert channel in runZero and provide the following details:
    - **Name**: Name for alert channel
    - **Channel type**: Webhook
    - **Webhook URL**: The webhook URL you copied from Sumo Logic

Step 3: Create a rule in runZero

Now that you have your alert template and channel created, you will want to identify the triggers to alert on. Some common examples are:

- Asset query results - When there is a match on a query in runZero after a scan completes
- New assets found - When a scan completes with new assets
- Agent offline - When your runZero Explorer stops checking in to the console
- Task failed - When a task fails for any reason.

We will use the **asset query results** selection as an example for the rest of the steps. Review example queries for ideas on what queries you could create.

1. Create a new alert rule.
2. Select `asset-query-results` and click **Configure rule**.
3. Input values for the rule:
    - **Name**: name of the rule.
    - **Conditions**: optional parameters that will trigger the alert when all conditions match.
        - **Query**: The query the assets must match.
        - **Number of matches**: The numeric comparison logic for the **value**.

- **Value**: The threshold of matches to trigger the rule.
- **Limit to organization**: Allows you to limit the alert to a specific organization.
- **Limit to site**: Allows you to limit the alert to a specific site.
- **Action**: `Notify`
  - **Notification channel**: Name of the alert channel you created in step 2.
  - **Notification template**: Name of the alert template you created in step 2.

Step 4: Handle runZero data in Sumo Logic

This search will display the raw runZero data

```
_source="runZero Alerts" and _collector="runZero"
```

This search will show alerts matching the runZero rule name

```
_source="runZero Alerts" and _collector="runZero"
| json field=_raw "found" nodrop
| json field=_raw "rule_name" nodrop
| where rule_name = "<RULE NAME>"
```

This search will create a graph of the data matching the runZero rule name

```
_source="runZero Alerts" and _collector="runZero"
| json field=_raw "assets_new" nodrop
| toLong(assets_new)
| json field=_raw "found" nodrop
| json field=_raw "search_url" nodrop
| json field=_raw "rule_name" nodrop
| where rule_name = "<RULE NAME>"
| timeslice 1m
| sum(found) by _timeslice
| order by _timeslice
```

Step 5: Create a dashboard in Sumo Logic (optional)

Now that you know how to look at the data in Sumo Logic and make a graph, you can follow these steps to create a dashboard. You will first create four rules in runZero, then you will import the Sample Sumo Logic Dashboard below.

Create the runZero rules

You will follow the same actions from step 3 to create each of these rules using the form inputs provided.

Assets running a TLS service

- **Rule type**: asset-query-results
- **Name**: `Assets running a TLS service - Sumo`
- **Query**: `alive:t protocol:tls`

- **Number of matches**: is greater than 0
- **Notification channel**: alert channel created in Step 2
- **Notification template**: alert template created in Step 2

Multihomed assets - Sumo

- **Rule type**: asset-query-results
- **Name**: `Multihomed assets - Sumo`
- **Query**: `alive:t AND multi_home:t`
- **Number of matches**: is greater than 0
- **Notification channel**: alert channel created in Step 2
- **Notification template**: alert template created in Step 2

Assets with OpenSSL - Sumo

- **Rule type**: asset-query-results
- **Name**: `Assets with OpenSSL - Sumo`
- **Query**: `alive:t product:openssl`
- **Number of matches**: is greater than 0
- **Notification channel**: alert channel created in Step 2
- **Notification template**: alert template created in Step 2

New assets to Sumo Logic

- **Rule type**: new-assets-found
- **Name**: `New assets to Sumo Logic`
- **Number of matches**: is greater than 0
- **Notification channel**: alert channel created in Step 2
- **Notification template**: alert template created in Step 2

Sample Sumo Logic Dashboard

1. Navigate to the library in Sumo Logic.
2. Click the **options** button on the folder you'd like to import to and click `Import`.
    - **Name**: runZero Alert Metrics
    - **JSON**: Copy from the sample below
3. Click `Import` to see this dashboard under the folder it was imported to.

```
{
    "type": "DashboardV2SyncDefinition",
    "name": "runZero Asset Metrics",
    "description": "",
    "title": "runZero Asset Metrics",
    "theme": "Dark",
    "topologyLabelMap": {
        "data": {}
    },
    "refreshInterval": 0,
    "timeRange": {
        "type": "BeginBoundedTimeRange",
```

```
"type": "RelativeTimeRangeBoundary",
            "relativeTime": "-3d"
        },
        "to": null
    },
    "layout": {
        "layoutType": "Grid",
        "layoutStructures": [
            {
                "key": "panelPANE-AC8FB3DCBD32DA48",
                "structure": "{\"height\":6,\"width\":12,\"x\":0,\"y\":0}"
            },
            {
                "key": "panel3D084A3284252A4E",
                "structure": "{\"height\":6,\"width\":12,\"x\":12,\"y\":0}"
            },
            {
                "key": "panelPANE-4389DBF794B13B44",
                "structure": "{\"height\":6,\"width\":12,\"x\":0,\"y\":6}"
            },
            {
                "key": "panelPANE-FBE08549B2123A4A",
                "structure": "{\"height\":6,\"width\":12,\"x\":12,\"y\":6}"
            }
        ]
    },
    "panels": [
        {
            "id": null,
            "key": "panelPANE-AC8FB3DCBD32DA48",
            "title": "New assets found",
            "visualSettings": "{\"general\":{\"mode\":\"timeSeries\",\"type\":\"line\",\"displayType
\":\"default\",\"markerSize\":5,\"lineDashType\":\"solid\",\"markerType\":\"none\",\"lineThickness\":
1},\"title\":{\"fontSize\":14},\"axes\":{\"axisX\":{\"titleFontSize\":12,\"labelFontSize\":12},\"axisY
\":{\"titleFontSize\":12,\"labelFontSize\":12,\"logarithmic\":false}},\"legend\":{\"enabled\":true,\"v
erticalAlign\":\"bottom\",\"fontSize\":12,\"maxHeight\":50,\"showAsTable\":false,\"wrap\":true},\"colo
r\":{\"family\":\"Categorical Default\"},\"series\":{},\"overrides\":[]}",
            "keepVisualSettingsConsistentWithParent": true,
            "panelType": "SumoSearchPanel",
            "queries": [
                {
                    "transient": false,
                    "queryString": "_source=\"runZero Alerts\" and _collector=\"runZero\"\n| json fiel
d=_raw \"assets_new\" nodrop\n| json field=_raw \"found\" nodrop\n| json field=_raw \"search_url\" nod
rop\n| json field=_raw \"rule_name\" nodrop\n| where rule_name = \"New Assets to Sumo Logic\"\n| times
lice 1m\n| sum(assets_new) by _timeslice\n| order by _timeslice",
                    "queryType": "Logs",
                    "queryKey": "A",
                    "metricsQueryMode": null,
                    "metricsQueryData": null,
                    "tracesQueryData": null,
                    "spansQueryData": null,
                    "parseMode": "Auto",
```

```
"outputCardinalityLimit": 1000
                }
            ],
            "description": "",
            "timeRange": null,
            "coloringRules": null,
            "linkedDashboards": []
        },
        {
            "id": null,
            "key": "panel3D084A3284252A4E",
            "title": "Multihomed assets found",
            "visualSettings": "{\"general\":{\"mode\":\"timeSeries\",\"type\":\"line\",\"displayType
\":\"default\",\"markerSize\":5,\"lineDashType\":\"solid\",\"markerType\":\"none\",\"lineThickness\":
1},\"title\":{\"fontSize\":14},\"axes\":{\"axisX\":{\"titleFontSize\":12,\"labelFontSize\":12},\"axisY
\":{\"titleFontSize\":12,\"labelFontSize\":12,\"logarithmic\":false}},\"legend\":{\"enabled\":true,\"v
erticalAlign\":\"bottom\",\"fontSize\":12,\"maxHeight\":50,\"showAsTable\":false,\"wrap\":true},\"colo
r\":{\"family\":\"Categorical Default\"},\"series\":{},\"overrides\":[]}",
            "keepVisualSettingsConsistentWithParent": true,
            "panelType": "SumoSearchPanel",
            "queries": [
                {
                    "transient": false,
                    "queryString": "_source=\"runZero Alerts\" and _collector=\"runZero\"\n| json fiel
d=_raw \"assets_new\" nodrop\n| toLong(assets_new)\n| json field=_raw \"found\" nodrop\n| json field=_
raw \"search_url\" nodrop\n| json field=_raw \"rule_name\" nodrop\n| where rule_name = \"Multihomed As
sets\" or rule_name = \"Multihomed Assets - Sumo\"\n| timeslice 1m\n| sum(found) by _timeslice\n| orde
r by _timeslice",
                    "queryType": "Logs",
                    "queryKey": "A",
                    "metricsQueryMode": null,
                    "metricsQueryData": null,
                    "tracesQueryData": null,
                    "spansQueryData": null,
                    "parseMode": "Auto",
                    "timeSource": "Message",
                    "outputCardinalityLimit": 1000
                }
            ],
            "description": "",
            "timeRange": null,
            "coloringRules": null,
            "linkedDashboards": []
        },
        {
            "id": null,
            "key": "panelPANE-4389DBF794B13B44",
            "title": "Assets with a TLS service",
            "visualSettings": "{\"general\":{\"mode\":\"timeSeries\",\"type\":\"line\",\"displayType
\":\"default\",\"markerSize\":5,\"lineDashType\":\"solid\",\"markerType\":\"none\",\"lineThickness\":
1},\"title\":{\"fontSize\":14},\"axes\":{\"axisX\":{\"titleFontSize\":12,\"labelFontSize\":12},\"axisY
\":{\"titleFontSize\":12,\"labelFontSize\":12,\"logarithmic\":false}},\"legend\":{\"enabled\":true,\"v
erticalAlign\":\"bottom\",\"fontSize\":12,\"maxHeight\":50,\"showAsTable\":false,\"wrap\":true},\"colo
```

```
"keepVisualSettingsConsistentWithParent": true,
            "panelType": "SumoSearchPanel",
            "queries": [
                {
                    "transient": false,
                    "queryString": "_source=\"runZero Alerts\" and _collector=\"runZero\"\n| json fiel
d=_raw \"assets_new\" nodrop\n| toLong(assets_new)\n| json field=_raw \"found\" nodrop\n| json field=_
raw \"search_url\" nodrop\n| json field=_raw \"rule_name\" nodrop\n| where rule_name = \"Assets runnin
g a TLS service\" or rule_name = \"Assets running a TLS service - Sumo\"\n| timeslice 1m\n| sum(found)
by _timeslice\n| order by _timeslice",
                    "queryType": "Logs",
                    "queryKey": "A",
                    "metricsQueryMode": null,
                    "metricsQueryData": null,
                    "tracesQueryData": null,
                    "spansQueryData": null,
                    "parseMode": "Auto",
                    "timeSource": "Message",
                    "outputCardinalityLimit": 1000
                }
            ],
            "description": "",
            "timeRange": null,
            "coloringRules": null,
            "linkedDashboards": []
        },
        {
            "id": null,
            "key": "panelPANE-FBE08549B2123A4A",
            "title": "Assets running OpenSSL",
            "visualSettings": "{\"general\":{\"mode\":\"timeSeries\",\"type\":\"line\",\"displayType
\":\"default\",\"markerSize\":5,\"lineDashType\":\"solid\",\"markerType\":\"none\",\"lineThickness\":
1},\"title\":{\"fontSize\":14},\"axes\":{\"axisX\":{\"titleFontSize\":12,\"labelFontSize\":12},\"axisY
\":{\"titleFontSize\":12,\"labelFontSize\":12,\"logarithmic\":false}},\"legend\":{\"enabled\":true,\"v
erticalAlign\":\"bottom\",\"fontSize\":12,\"maxHeight\":50,\"showAsTable\":false,\"wrap\":true},\"colo
r\":{\"family\":\"Categorical Default\"},\"series\":{},\"overrides\":[]}",
            "keepVisualSettingsConsistentWithParent": true,
            "panelType": "SumoSearchPanel",
            "queries": [
                {
                    "transient": false,
                    "queryString": "_source=\"runZero Alerts\" and _collector=\"runZero\"\n| json fiel
d=_raw \"assets_new\" nodrop\n| toLong(assets_new)\n| json field=_raw \"found\" nodrop\n| json field=_
raw \"search_url\" nodrop\n| json field=_raw \"rule_name\" nodrop\n| where rule_name = \"Assets with O
penSSL\" or rule_name = \"Assets with OpenSSL - Sumo\"\n| timeslice 1m\n| sum(found) by _timeslice\n|
order by _timeslice",
                    "queryType": "Logs",
                    "queryKey": "A",
                    "metricsQueryMode": null,
                    "metricsQueryData": null,
                    "tracesQueryData": null,
                    "spansQueryData": null,
                    "parseMode": "Auto",
```

```
"outputCardinalityLimit": 1000
                }
            ],
            "description": "",
            "timeRange": null,
            "coloringRules": null,
            "linkedDashboards": []
        }
    ],
    "variables": [],
    "coloringRules": []
}
```

Tines

runZero integrates with Tines to help you automate workflows related to your asset data. This helps teams leverage runZero to the fullest while optimizing the team's workflows with automation. A video demo is available to show the final outcome of these instructions.

Requirements

- A Tines account
- runZero Export API and Organization API tokens

There are two ways to integrate runZero and Tines:

- Follow the steps to create a custom story in Tines, or
- Use the **runZero sample story** to begin with a story outline.

Tines custom story

A Tines story is a collection of actions that work together towards a specific goal, like a playbook. Tines has a Story Library that contains ready-made automated playbooks, or you can create a your own custom story if they don't have one that matches your needs. That's what we'll need to do here.

Step 1: Creating a Tines story and adding runZero API credentials

1. After logging in to Tines, create a new story.
2. Use the `+` in the **credentials** section to add a **text** credential.
    - **Name**: specify a name for the credential, for example `runzero_export_token`.
    - **Value**: your runZero export API token, which can be obtained from the desired runZero organization page. Export API tokens start with `ET`.
3. Use the `+` in the **credentials** section to add another **text** credential.
    - **Name**: specify a name for the credential, for example `runzero_org_token`.
    - **Value**: your runZero organization API token, which can be obtained from the desired runZero organization page. Organization API tokens start with `OT`.

Step 2: Creating a Tines webhook action

1. Add a **Webhook** action to your story.
2. Click the **Webhook** action and copy the webhook URL, which will look like `https://<tenant-name>.tines.com/webhook/<guid>`.

Step 3: Creating a runZero alert template

1. Create an alert template in runZero:
    - **Name**: name for template
    - **Template type**: JSON
    - **Subject line for message**: leave empty
    - **Body of message**: the following JSON example will include the rule name and the search URL in the alert message body: `{"rule_name":"{{rule.name}}","search_url":"{{search.url}}"}`
2. Create an alert channel in runZero:
    - **Name**: name for alert channel
    - **Channel type**: webhook
    - **Webhook URL**: the webhook URL you copied from Tines

Step 4: Creating a rule in runZero

Now that you have your template ready to go, you will want to identify which triggers to alert on. Some common examples are:

- **Asset query results** - When there is a match on a query in runZero after a scan completes
- **New assets found** - When a scan completes with new assets
- **Agent offline** - When your runZero Explorer stops checking in to the console
- **Task failed** - When a task fails for any reason

We will use the **asset query results** selection for the rest of the steps. Our example will be any asset that has an open Telnet port `port:23`. You can see more example queries.

1. Create an alert rule.
2. Select `asset-query-results` and click *Configure rule*.
3. Input values for the rule:
    - **Name**: name of the rule.
    - **Conditions**: optional parameters that will trigger the alert when all conditions match.
        - **Query**: query the assets must match, such as `port:23` for our example.
        - **Number of matches**: the numeric comparison logic for the **value**.
        - **Value**: the threshold of matches to trigger the rule.
        - **Limit to organization**: allows you to limit the alert to a specific organization in runZero
        - **Limit to site**: allows you to limit the alert to a specific site in runZero
        - **Action**: notify
            - **Notification channel**: name of the alert channel you created in step 2.

- **Notification template**: name of the alert template you created in step 2.

Step 5: Handling the data in Tines

1. The **Webhook** action created in Step 2 will be the entry point for the runZero alert in Tines.
2. Add an **Event Transform** to parse the search from the URL provided in the runZero alert. If you connect the **Webhook** action to this one, some parameters will automatically populate.
    - Mode: extract
    - Matchers:
        - Path: `path_to_alert_search`, for example `receive_alerts_from_runzero.body.search_url` if your **Webhook** action is named "RECEIVE alerts from runZero"
        - Regex: `\?.*`
        - Extract to: `search`
3. Add an Event Transform that takes the output of the previous step. If you connect the previous **Event Transform** action with this one, some parameters will automatically populate.
    - Mode: explode
    - Matchers:
        - Path: `path_to_extraction_transform`, for example `get_search_from_url.search` if your previous **Event Transform** action is named "GET search from URL"
        - To: `individual_item`
4. Add an HTTP Request to make an API call to the runZero Export API.
    - URL: the `path_to_explosion_transform` added to the end of the runZero API URL endpoint, for example `https://console.runZero.com/api/v1.0/export/org/assets.json<<get_search_from_regex.individual_item>>` if your previous **Event Transform** action is named "GET search from regex"
    - Content Type: JSON
    - Method: get
    - Use the **+ Option** button to add **Headers**:
        - Change `header` to `Authorization`
        - Change `value` to `Bearer CREDENTIAL.runzero_export_token`
5. Add an Event Transform that takes the runZero Export API output and loops through each value.
    - Mode: explode
    - Path: `path_to_HTTP_request`, for example `get_assets_from_runzero.body` if your **HTTP Request** action is named "GET assets from runZero"
    - To: `individual_item`

Step 6: Personalized automation in Tines

Now that you have received the alert, parsed the search, and obtained the assets from the runZero Export API, it's time to add your own flare. While you will likely want to do something more crafty, the sample story provided includes these two actions as examples.

1. Send an email to the destination of your choice for each asset. The sample JSON shows how you might customize it to put the context you'd like in the Subject and Body of the

email.
2. The HTTP Request action shows how you might reach back into runZero to add tags to the assets after other data is gathered.

runZero sample story

The sample story below can be imported to Tines to do all of the actions outlined in the steps above. Simply save a JSON file with the contents below, and use the **Import** button in Tines to upload it. After importing, you will need to complete the following steps.

Step 1: Update credentials in Tines

1. Replace the `runzero_export_token` credential:
    - **Name**: specify a name for the credential, such as `runzero_export_token`
    - **Type**: text
    - **Value**: your runZero export API token, which can be obtained from the desired runZero organization page. Export API tokens start with `ET`.
2. Replace the `runzero_org_token` credential:
    - **Name**: specify a name for the credential, such as `runzero_org_token`
    - **Type**: text
    - **Value**: your runZero organization API token, which can be obtained from the desired runZero organization page. Organization API tokens start with `OT`.

Step 2: Create runZero alert template

1. Create an alert template in runZero:
    - **Name**: name for template
    - **Template type**: JSON
    - **Subject line for message**: leave empty
    - **Body of message**: the following JSON example will include the rule name and the search URL in the alert message body: `{"rule_name":"{{rule.name}}","search_url":"{{search.url}}"}`
2. Create an alert channel in runZero:
    - **Name**: name for alert channel
    - **Channel type**: webhook
    - **Webhook URL**: the webhook URL you copied from Tines

Step 3: Create the rule in runZero

Now that you have your template ready to go, you will want to identify which triggers to alert on. Some common examples are:

- Asset query results: when there is a match on a query in runZero after a scan completes.
- New assets found: when a scan completes with new assets.
- Agent offline: when your runZero Explorer stops checking in to the console.
- Task failed: when a task fails for any reason.

We will use the **asset query results** selection for the rest of the steps. Our example will be any asset that has an open Telnet port `port:23`. You can see more example queries.

1. Create a new alert rule.
2. Select `asset-query-results` and click `Configure rule`.
3. Input values for the rule:
    - **Name**: name of the rule.
    - **Conditions**: optional parameters that will trigger the alert when all conditions match.
        - **Query**: query the assets must match, such as `port:23` for our example.
        - **Number of matches**: the numeric comparison logic for the **value**.
        - **Value**: the threshold of matches to trigger the rule.
        - **Limit to organization**: allows you to limit the alert to a specific organization in runZero
        - **Limit to site**: allows you to limit the alert to a specific site in runZero
        - **Action**: notify
            - **Notification channel**: name of the alert channel you created in step 2.
            - **Notification template**: name of the alert template you created in step 2.

runZero sample story JSON


```
{
  "schema_version": 4,
  "standard_lib_version": 6,
  "name": "runZero Sample Story",
  "description": null,
  "guid": "7706b502e51b4c68f0dbe9721c88d665",
  "slug": "runzero_sample_story",
  "exported_at": "2022-10-27T22:22:37Z",
  "agents": [
    {
      "type": "Agents::WebhookAgent",
      "name": "RECEIVE alerts from runZero",
      "disabled": false,
      "guid": "ab1f5bf7be49c943d893ab993ade9421",
      "options": {
        "path": "da6e90c0b276bdda97e6bfa31ad50787",
        "secret": "5e290305e0a483bf843f1213f0f21dda",
        "verbs": "get,post"
      },
      "reporting": {
        "time_saved_value": 0,
        "time_saved_unit": "minutes"
      },
      "monitoring": {
        "monitor_all_events": false,
        "monitor_failures": false,
        "monitor_no_events_emitted": null
      },
```

```
},
    {
      "type": "Agents::HTTPRequestAgent",
      "name": "GET assets from runZero",
      "disabled": false,
      "guid": "d54d8f3f5b0ea857308f45f61a224547",
      "options": {
        "url": "https://console.runZero.com/api/v1.0/export/org/assets.json<<get_search_from_regex.ind
ividual_item>>",
        "content_type": "application_json",
        "method": "get",
        "headers": {
          "Authorization": "Bearer <<CREDENTIAL.runzero_export_token>>"
        }
      },
      "reporting": {
        "time_saved_value": 0,
        "time_saved_unit": "minutes"
      },
      "monitoring": {
        "monitor_all_events": false,
        "monitor_failures": false,
        "monitor_no_events_emitted": null
      },
      "width": null,
      "schedule": []
    },
    {
      "type": "Agents::EventTransformationAgent",
      "name": "GET search from URL",
      "disabled": false,
      "guid": "d62ddd189bb9080d778d900f2292504d",
      "options": {
        "mode": "extract",
        "matchers": [
          {
            "path": "=receive_alerts_from_runzero.body.search_url",
            "regexp": "\\?.*",
            "to": "search"
          }
        ]
      },
      "reporting": {
        "time_saved_value": 0,
        "time_saved_unit": "minutes"
      },
      "monitoring": {
        "monitor_all_events": false,
        "monitor_failures": false,
        "monitor_no_events_emitted": null
      },
      "width": null,
      "schedule": null
```

```
{
      "type": "Agents::EventTransformationAgent",
      "name": "GET search from regex",
      "disabled": false,
      "guid": "41afc5c4d4c0fdb694d14e1eb380688a",
      "options": {
        "mode": "explode",
        "path": "=get_search_from_url.search",
        "to": "individual_item"
      },
      "reporting": {
        "time_saved_value": 0,
        "time_saved_unit": "minutes"
      },
      "monitoring": {
        "monitor_all_events": false,
        "monitor_failures": false,
        "monitor_no_events_emitted": null
      },
      "width": null,
      "schedule": null
    },
    {
      "type": "Agents::EventTransformationAgent",
      "name": "LOOP through assets for follow up actions",
      "disabled": false,
      "guid": "b9ada61162bdf006f09d264845ebd304",
      "options": {
        "mode": "explode",
        "path": "=get_assets_from_runzero.body",
        "to": "individual_item"
      },
      "reporting": {
        "time_saved_value": 0,
        "time_saved_unit": "minutes"
      },
      "monitoring": {
        "monitor_all_events": false,
        "monitor_failures": false,
        "monitor_no_events_emitted": null
      },
      "width": null,
      "schedule": null
    },
    {
      "type": "Agents::EmailAgent",
      "name": "SEND Email Action",
      "disabled": false,
      "guid": "0e9647b114f315d480f1f85c58e481d3",
      "options": {
        "recipients": "youremail@email.com",
        "reply_to": "youremail@email.com",
        "sender_name": "Your Name",
```

```
"body": "Alert: <<receive_alerts_from_runzero.body.rule_name>>\n\\<br />\n\\<br />\nLink to asset: htt
ps://console.runzero.com/inventory/<<loop_through_assets_for_follow_up_actions.individual_item.id>>\n
\\<br />\n\\<br />\nAddresses: <<loop_through_assets_for_follow_up_actions.individual_item.addresses>>
\n\\<br />\n\\<br />\nServices:\n<<NEWLINE_TO_BR(NEAT_JSON(loop_through_assets_for_follow_up_actions.i
ndividual_item.services))>>"
      },
      "reporting": {
        "time_saved_value": 0,
        "time_saved_unit": "minutes"
      },
      "monitoring": {
        "monitor_all_events": false,
        "monitor_failures": false,
        "monitor_no_events_emitted": null
      },
      "width": null,
      "schedule": null
    },
    {
      "type": "Agents::HTTPRequestAgent",
      "name": "TAG assets in runZero",
      "disabled": false,
      "guid": "c3171fed91ec40d39572260295a63ae0",
      "options": {
        "url": "https://console.runZero.com/api/v1.0/org/assets/<<loop_through_assets_for_follow_up_ac
tions.individual_item.id>>/tags",
        "content_type": "application_json",
        "method": "patch",
        "headers": {
          "Authorization": "Bearer <<CREDENTIAL.runzero_org_token>>"
        },
        "payload": {
          "tags": "hello=from_tines"
        }
      },
      "reporting": {
        "time_saved_value": 0,
        "time_saved_unit": "minutes"
      },
      "monitoring": {
        "monitor_all_events": false,
        "monitor_failures": false,
        "monitor_no_events_emitted": null
      },
      "width": null,
      "schedule": []
    }
  ],
  "diagram_notes": [
    {
      "content": "Hello! Thanks for importing the runZero Sample Story. \n\nThis story is a basic exam
ple of what you can do with runZero and Tines. \n\nPrerequisites:\n1. Create a text [credential](http
s://www.tines.com/docs/credentials/text) called `runzero_export_token`. Set the value as your [runZero
```

n](https://console.runzero.com/organizations) page. Export tokens start with `ET`.\n2. Create a text [credential](https://www.tines.com/docs/credentials/text) called `runzero_org_token`. Set the value as your as your [organization API token](docs/leveraging-the-api.md), which can be obtained from the desired [runZero organization](https://console.runzero.com/organizations). Organization API tokens start with `OT`. \n\nThe steps are as follows:\n\n1. **RECEIVE alerts** provides the webhook destination for your runZero alerts \n\n2. **GET search from URL** parses out the search string from the URL provided in the runZero alert \n\n3. **GET search from regex** takes the list provided in step 2 and allows you to use the individual value \n\n4. **GET assets from runZero** uses the runZero Export API to get the list of assets related to the alert \n\n5. **LOOP through assets for follow up actions** takes the list of assets and sends each to the next steps individually \n\n6. **NOTE**: this is the step that you could implement more custom logic in most cases. All of the initial runZero and Tines data transfer is done, but it's your chance to customize this story to fit your use case. \n\n7. **SEND email action** simply sends an email to the destination of your choice \n\n8. **TAG assets in runZero** adds a tag in runZero to each asset showing an example of how you might reach back into runZero after doing other automated activities ",
      "position": [
        165.0,
        180.0
      ],
      "guid": "0e4b9b331dc65316849f7b089bf7bde2",
      "width": 375
    }
  ],
  "links": [
    {
      "source": 0,
      "receiver": 2
    },
    {
      "source": 1,
      "receiver": 4
    },
    {
      "source": 2,
      "receiver": 3
    },
    {
      "source": 3,
      "receiver": 1
    },
    {
      "source": 4,
      "receiver": 6
    },
    {
      "source": 4,
      "receiver": 5
    }
  ],
  "diagram_layout": "{\"ab1f5bf7be49c943d893ab993ade9421\":[555,180],\"d54d8f3f5b0ea857308f45f61a224547\":[555,450],\"d62ddd189bb9080d778d900f2292504d\":[555,270],\"41afc5c4d4c0fdb694d14e1eb380688a\":[555,360],\"b9ada61162bdf006f09d264845ebd304\":[555,555],\"0e9647b114f315d480f1f85c58e481d3\":[555,690],\"c3171fed91ec40d39572260295a63ae0\":[780,690]}",

```
"entry_agent_guid": null,
  "exit_agent_guids": [],
  "exit_agent_guid": null,
  "keep_events_for": 604800,
  "reporting_status": true,
  "send_to_story_access": null,
  "send_to_stories": [],
  "form": null,
  "forms": []
}
```


Thinkst Canary

All runZero editions integrate with Thinkst Canary by providing quick access from the Canary console to your asset data in the runZero Console. Setting up the integration is as simple as one change to your Canary settings.

Integrate runZero with Thinkst Canary

1. Sign in to your runZero console.
2. Sign in to your Canary console.
3. Go to **Global Settings** under the gear icon.
4. Click on **Integrations**.
5. Toggle the runZero switch.

Accessing runZero data from Canary

After the integration is enabled within the Canary settings, runZero data will be available through any incident. When viewing an incident, click the magnifying glass icon under `source IP` or `reverse IP lookup` to open a search in your runZero console.

Exposure

runZero supports the following exposure management capabilities:

- Complete asset identification
- Attack surface management
- Vulnerability management
- Risk prioritization
- Continuous monitoring

Asset identification

runZero provides comprehensive asset identification by combining active scanning, passive traffic sampling, and integrations with deep fingerprinting and correlation capabilities.

Assets are normalized, deduplicated, and tracked as they move across your environment.

Attack surface management

runZero offers comprehensive attack surface management for external, internal, cloud, IoT, and OT environments.

Vulnerability management

runZero imports vulnerability reports from leading endpoint management and vulnerability management platforms.

In addition to importing data, runZero natively reports vulnerability information in the following categories:

- Applications that expose data without authentication
- Actively exploited vulnerabilities in the wild
- Devices and applications using default logins
- End-of-Life hardware, firmware, and software
- Shared and compromised encryption keys
- Expired and insecure certificates
- Devices from embargoed manufacturers
- Internet exposure of management services
- Internet exposure of internal assets
- Internet exposure of operational technology
- Insecure protocol configurations
- Missing security controls
- Network outliers

runZero & Nuclei

In addition to native capabilities, runZero scans can also make use of the Nuclei open source vulnerability scanner. A modified version of the Nuclei engine is embedded into the runZero Explorer and CLI binaries. These binaries also include a set of curated vulnerability checks (templates) that can be selected from the runZero scan configuration. runZero's Nuclei templates are open source and regularly submitted back to the upstream project.

Early access

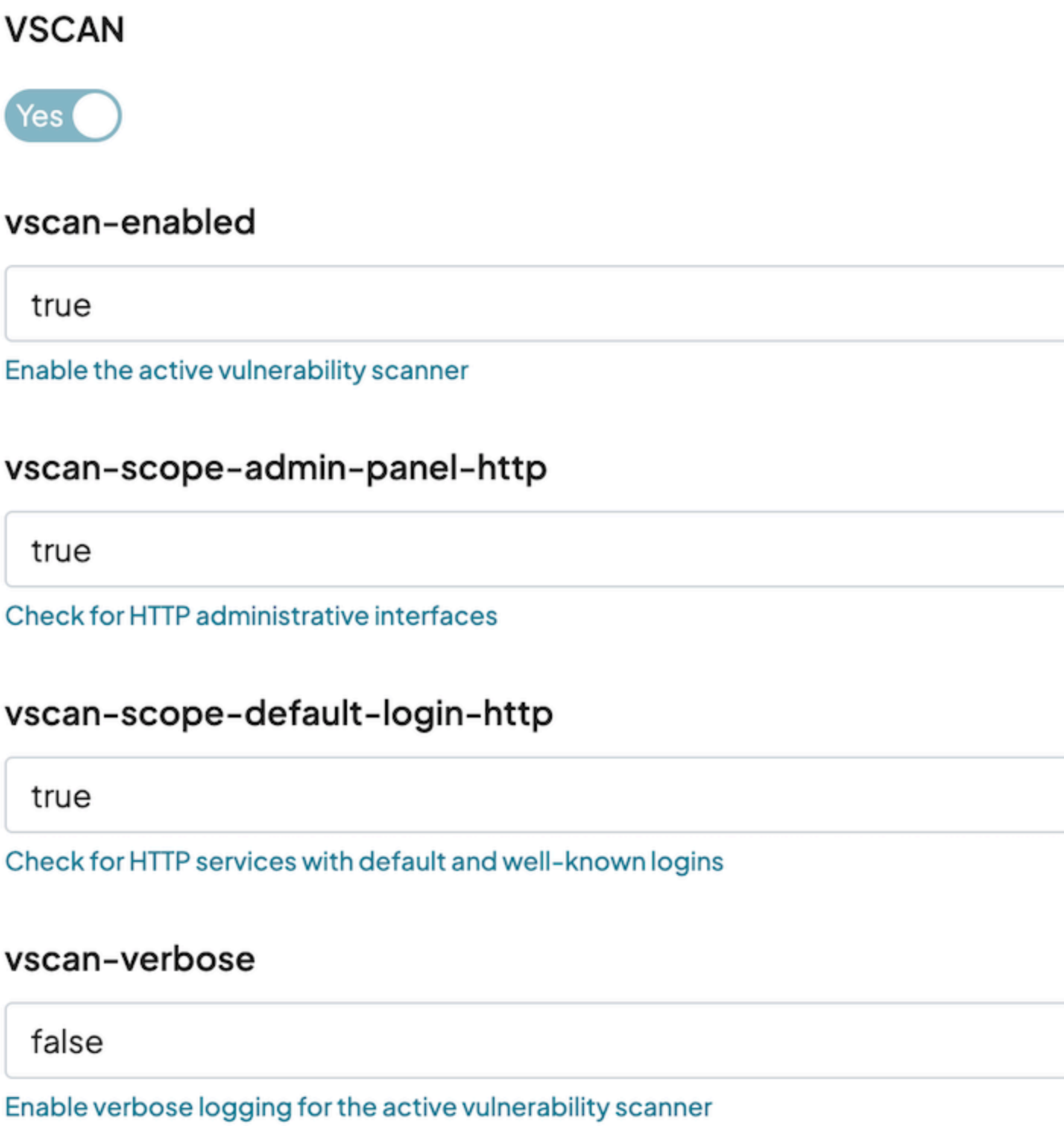
Use of Nuclei within runZero is still early access and not enabled by default.

To enable Nuclei templates within a runZero scan:

1. Ensure that you are using runZero 4.0.250606.0 or newer

2. Create or modify a new Scan Task
3. Under the Probes & SNMP tab, search for the *vscan* probe.
4. Set the *vscan-enabled*, *vscan-scope-admin-panel-http*, and *vscan-scope-default-login-http* values to true.
5. Save and/or launch the scan task.

These settings enable detections for exposed web panels and default and weak credentials within web interfaces.



Approximately 240 templates are enabled today. These templates only run when runZero's precise fingerprinting determines that the specific asset and service is appropriate for the

template. As a result, including these options generally does not noticeably increase the amount of time that a scan takes.

There are a couple caveats:

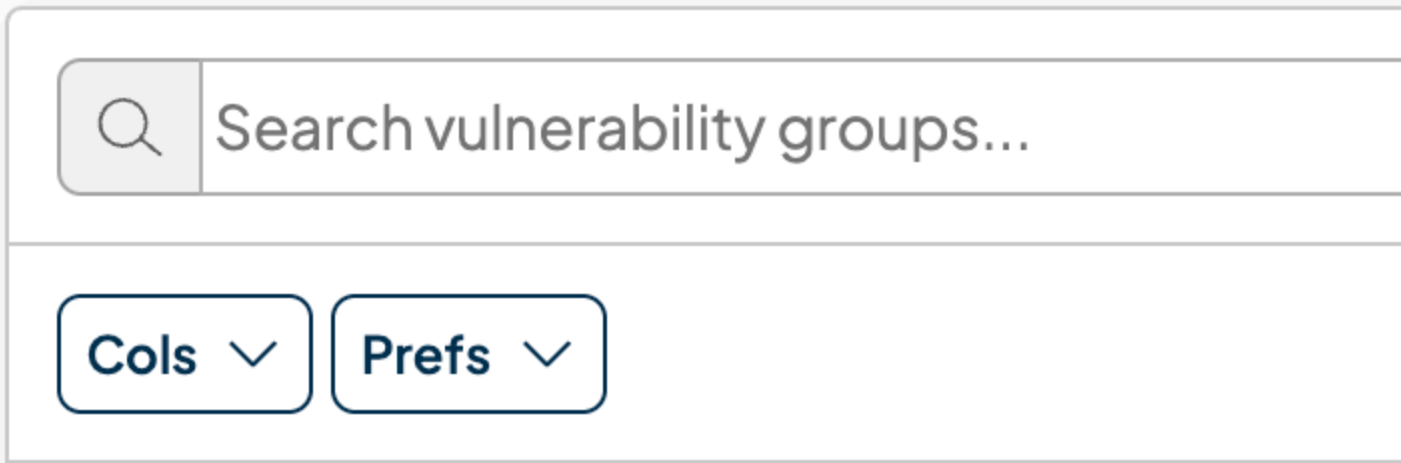
1. Default login tests, even for HTTP interfaces, can result in account lockouts. The default settings of things like Apache Tomcat and CrushFTP can result in the administrative accounts being temporarily locked as the result of a scan.

2. Templates may not run if there is a gap in runZero's fingerprinting or the service presents a variant that we haven't seen before. Our goal is to be as safe as possible and this is one of the tradeoffs compared to running Nuclei as a standalone tool.

If you'd like to see extensive logs for what decisions are made by the scan engine, set the *vscan-verbose* option to `true` and review the task logs after the scan completes.

After the scan completes, any identified vulnerabilities and findings will be shown in the respective product sections.

Vulnerability inventory



Future plans

runZero plans to support a wide assortment of new coverage through the Nuclei engine.

Our near-term plans include:

- Dedicated configuration elements for vulnerability check selection
- Extending default-login templates to cover non-HTTP protocols (SSH, Telnet, FTP)
- Rapid Response templates for emerging vulnerabilities
- Extended coverage of important vulnerabilities

Slightly longer-term, we would like to support customer-provided templates and template libraries.

From an open-source perspective, we have a ton of ideas for improving the Nuclei engine, including linking Nuclei with SSHamble and excrypto.

Risk prioritization

runZero normalizes and assigns risk scores to all assets. These scores are influenced by a combination of threat intelligence, vulnerability information, and exposure measurement. Asset criticality can be set through automated rules or manually through the product interface. Assets can be assigned to specific owners for remediation. Tags are imported from API integrations and can be managed natively within the interface, including rule-based tagging. Exported assets include the risk, criticality, and tag information set through the product interface.

Continuous monitoring

runZero continuously monitors your organization for changes to exposure, at a per-asset and per-service level. Recurring active scans, background passive traffic sampling, and regular sync with your existing infrastructure enable quick detection and reporting of new risks. Alerts can be managed in-product, sent by email, or delivered by webhook to the platform of your choice. All asset data can be synced to external platforms, including popular SIEMs and data lakes.

The certificates inventory

As the runZero Explorer scans your network, it collects encryption certificates that it encounters. These are stored in the certificates inventory, which you can access from the *Inventory* menu, *Certificates* sub-menu.

The primary source of certificates on most networks will be TLS certificates on web servers. However, the Explorer will also collect certificates from other services, such as SMTP, RDP and DNS-SD.

Certificates have a validity period. In the inventory view, the *Valid from* and *Valid until* values are color coded to show if the certificate is not valid yet, has expired, or is soon to expire.

A certificate has an *issuer*, the authority which signed the certificate. The issuer may be the same as the subject, in which case the certificate will be marked as self-signed certificate.

Certificates rely on two kinds of algorithms:

- The public key algorithm, used for decryption and verification.
- The signature algorithm, used to produce a content hash that is signed using the public key algorithm.

The security of a public key algorithm depends on the key size. This is shown in bits. How large a key is considered secure will depend on the algorithm. For example, for RSA a key size should be 2048 bits or larger to be considered secure; whereas for ed25519, all keys are 256 bits, and that's considered secure.

The certificate inventory view has a button for each certificate to hide it from the inventory. This can be useful for embedded devices that have certificates you can't change, that you don't want to see in the inventory. If you later decide you want to see which certificates you've hidden, you can use the search query `hidden:true`.

TLS certificates

TLS certificates have three sets of names for their hosts:

- The Subject, an X.509 distinguished name (DN) made up of one or more attributes, each with a value.
- The Subject Alternative Name lists. These provide the values web browsers check against when connecting to a server. SANs are stored in a certificate in separate fields by type (DNS, IP, email or URI).
- The Common Name, which was once used by web browsers but is now ignored by them. It is typically in X.509 format.

The Names shown in runZero for a TLS certificate are assembled from the four different kinds of SANs, plus the Subject DN.

Each TLS certificate has a subject key ID used to identify the subject of the certificate, and an authority key ID used to identify the signing authority. These are used to link certificates

together in a chain — the authority key ID of each certificate is the subject key ID of the next certificate in the chain.

The issuer for a TLS certificate is included as a DN in the certificate.

TLS certificates have a flag to say if they are intended for use as a certificate authority or (CA). The flag doesn't mean that they *are* a CA, or that they are trusted; just that the certificate has that flag set. CA certificates are usually self-signed.

The values shown to identify TLS certificate algorithms are mostly taken from RFC 5280, and should correspond closely to the values used by OpenSSL.

The self-signed value indicates whether a certificate appears to be self-signed, based on its issuer, subject, and (if present) their IDs. A certificate may be flagged as apparently self-signed but not have a valid signature according to some software, as different software may implement more or less strict checks.

The certificate details page has a button to download the certificate in PEM format. You can also download the entire chain of certificates known to runZero as a single PEM file.

Data analysis

Reviewing results

Task details

After each discovery task completes, the task details page will list a summary of how many assets were updated. To understand the numbers, it's important to remember that runZero will correlate assets across IPs and data sources, which can result in different results than IP-based matching alone.

The change summary on the task details page includes the following statistics:

- Asset changes:
  - **Newly discovered assets** are devices that were found during the task for which no device with matching fingerprints was previously seen.
  - **Assets marked offline** are assets that runZero has previously seen on the scanned network, but that didn't respond on any of the IP addresses during this scan. When this happens, the asset is marked offline. The offline status is a flag on the asset, and doesn't count as a change to the asset. Assets may be marked offline because the device was powered down or disconnected, or because of network problems.
  - **Assets back online** are assets that were marked offline at some point in the past, but the runZero Explorer got a response from them during this scan. The online status is a flag on the asset, and doesn't count as a change to the asset.
  - **Assets changed** is the number of assets where some property of the asset was modified, other than its online status. Examples include changes to the device's IP addresses or hostname, or responses from new ports or protocols.
  - **Assets unchanged** is the number of assets that were seen exactly where runZero found them in the last scan, with no changes to their responses.
  - **Assets ignored** is the number of occasions where the Explorer got a response from probing an IP address, but it turned out to be bogus in some way. This typically happens when a web proxy, stateful firewall, or SIP gateway responds as if it is the asset at every address on a subnet.
  - **Assets updated by task** is the total of **Assets changed** plus **Assets unchanged**. It indicates the number of asset records that are now up-to-date.
- User changes:
  - **Newly discovered users** are users that were seen for the first time during the integration sync.
  - **Users changed** are users that had attributes change during the integration sync.
  - **Users unchanged** are users that did not change during the integration sync.
  - **Users updated by task** is the total number of **Users changed** and **Users unchanged**, indicating how many user records are now up-to-date.
- Group changes:
  - **Newly discovered groups** are groups that were seen for the first time during the integration sync.

- **Groups changed** are groups that had attributes change during the integration sync.
- **Groups unchanged** are groups that did not change during the integration sync.
- **Groups updated by task** is the total number of **Groups changed** and **Groups unchanged**, indicating how many group records are now up-to-date.

Dashboard & inventory views

The dashboard will be populated with results after the first scan completes. The dashboard provides trend data and insights that will help you assess how your inventory is changing over time. You can select a time period and site for the trend data using the selectors at the top right of the dashboard page.

The main asset trends graph shows the number of assets in each of the four main states – live, offline, scanned and unscanned. Beneath the graph are additional asset breakdowns, each of which shows a top 10 of an asset category – asset type, operating system, hardware and tags.

The service trends graph shows how many total services were found in your asset inventory, along with breakdowns for ARP, ICMP, TCP and UDP. Below the service trends graph are breakdowns of the top 10 TCP ports, UDP ports, protocols and products detected.

Clicking the menu button at the top right of each table and selecting “View more” shows a more detailed inventory by category.

Insights from queries

Queries and reports can help you gain valuable insights, but you may wonder where to get started. We recommend trying the pre-built queries in the Query Library first. Some of these queries are a result of runZero’s Rapid Response to emerging threats and are described on our blog.

runZero’s query language allows you to search and filter your asset inventory based on asset fields and value pairs. See the documentation about querying your data. Once you are familiar with the query language you can write your own queries.

You can set queries to run automatically by opening the query and setting the “Automatically track query results on the dashboard”. The query will run when scans complete, and you will be notified of any resulting insights on the dashboard page.

Sample Queries

Asset inventory

- Equipment that is likely 8+ years old: `alive:t mac_age:>8years`
- Assets with end-of-life OS: `os_eol:<now`
- Virtual machines: `has:virtual`
- Devices acting as a router: `router:true`

- Devices that may be bridging: `has_public:t and has_private:t`

Service inventory

- Protocol on a non-standard port example: `protocol:ssh not port:22`
- Publicly addressed assets running RDP or VNC: `has_public:t and (protocol:rdp or protocol:vnc)`
- Authenticated web services that are not encrypted: `(_asset.protocol:http AND not _asset.protocol:tls) AND ( html.inputs:"password:" OR last.html.inputs:"password:" OR has:http.head.wwwAuthenticate OR has:last.http.head.wwwAuthenticate )`
- Older TLS versions in use: `alive:t AND protocol:"=tls" AND ( tls.versionName:"=TLS 1.0" OR tls.versionName:"=TLS 1.1")`

Some other sample queries are described in our blog entries:

- Finding duplicate SSH host keys
- Identifying rogue remote access solutions
- Finding device serial numbers

Reports

After viewing the dashboard and inventory, your next stop should be the runZero Reports page.

Switch topology

This report uses SNMP information to map how the switches on your network are connected. Each switch displays its IP address, name, and the number of assets connected to it. If runZero detected MAC addresses that were not found as part of the scan scope, you will see a number of unmapped assets indicated below the switch.

You can click on a switch to see a pop-up with the number of identified and unmapped assets. From there, you can click to view the unmapped assets, and be taken to a table of unmapped MACs by switch port.

Double-clicking on a switch will expand that part of the diagram and show the individual assets connected to the switch.

The switch topology report won't always be entirely accurate as it's based on which switch claims to have seen each MAC address, and this may not always be the nearest access switch. Our algorithm looks for the switch port with the least number of shared MACs to find the best match, but this may not give the answer you expect, depending on switch cache timeouts and how the switches were scanned.

Subnet utilization

The subnet utilization report lists the subnets scanned on your network, and what percentage of each is in use. For example, if you have scanned 10.0.1.0/24 and found 25 assets, the report

will show that 10% of the available IP addresses in the subnet are in use.

Network bridges

The network bridges report is a way to find devices that bridge multiple network segments. It can be useful to locate unintentional bridging between your internal networks and the Internet.

The report shows your internal networks in green, and external networks in red. It then shows you the multihomed assets which bridge an internal network to an external one.

RFC 1918 coverage

The RFC 1918 coverage report is a way to view how much of the private internal network address space has been scanned for assets. It can help you discover rogue assets, unscanned subnets, and secondary interfaces on scanned devices. More information is in the section on coverage reports.

Unmapped MACs

This report uses SNMP information to list MAC addresses runZero found evidence for, but which weren't encountered as addresses of assets during the network scan. The MAC addresses are grouped by the switch that reported them, along with information about the vendor, manufacture date and switch port of the possible asset, to help identify them.

Outliers

The outliers reports allow you to obtain a summary of how often different values occur in specific attributes of assets and services. The values are sorted from most frequent to least frequent.

For example, the HTTP servers outliers report will list all of the HTTP servers encountered by runZero, starting with the most common.

As well as the one-click outliers reports, you can produce an outliers report for any asset or service attribute.

- Switch topology to identify how your assets are connected and find "unmapped" MAC addresses (in red) that were not included in your scan scope (a summary of which is in the Unmapped MACs report)
- Bridging to visualize what hosts may have both public and private connections
- RFC 1918 coverage that can identify potential blindspots on your network like missing (unscanned) subnets, rogue devices, and "hinted" IPs that are secondary interfaces on unscanned network ranges.
- See the "View all" button at the top right for a list of other reports to investigate outliers

Domain membership report

The domain membership report lists the Active Directory domains encountered by runZero, and lists how many assets are in each.

Analysis reports

Platform

Analysis reports are more advanced reports. They may run as tasks, rather than being generated on-the-fly.

The first analysis report is Compare Sites, which generates searchable reports of the differences between two sites.

The Outlier Overview Report analyzes assets across the organization and summarizes the most unusual values for an assortment of key attributes such as hardware type and SNMP enterprise ID.

The Specific Outlier Report allows you to select an attribute which has outlying values and get a detailed breakdown of those values.

The Organization Overview Report builds a high level summary report of the entire organization. It can optionally include lists of assets found.

Alerts

Platform

As well as manually generated reports and queries, runZero also supports automatic alerts to designated channels for post-scan inventory queries, asset changes, Explorer and scan issues, security operations, or API events.

Available channels are internal notifications in the runZero web console, email, or webhooks that can enable integration with services such as Slack or Mattermost. Alerts use the same query language as the sample queries above, so this is a good way to automate proactive notification for critical events.

Using dashboards

Dashboards provide customizable, visual views into your asset inventory and can be created to serve different use cases such as compliance, vulnerability remediation, or asset visibility.

A variety of visualization widgets are available that show operational information, trends, insights, goals, sources, and most and least seen graphs. You can also create your own custom widget based on queries to get the exact data you are looking to surface, displayed either as a trend line or latest count.

You can filter the dashboard data by site and time buckets based on your needs by using the dropdown buttons at the top of the dashboard and selecting the timeframe and/or site you wish to filter by.

You can also customize the arrangement and size of each widget on the dashboard to suit your particular needs using Edit Mode, and access the widget library to add or remove widgets as desired.

Dashboard selection

The dashboard selector button at the top of the dashboard displays the dashboard's name by default, and when clicked will open up a menu that displays runZero-managed dashboards, any shared dashboards, as well as your personal customized dashboards. To change dashboards, simply click the name of the dashboard you wish to view.

Creating dashboards

To create a new personal dashboard, click the dashboard selector and click the "Create new dashboard" button, which will create the new dashboard and automatically take you into Edit Mode where you can begin adding, resizing and arranging widgets.

Customizing dashboards

Personal dashboards can be customized by clicking the "Edit" button found in the dashboard header, which will put the dashboard into "Edit Mode" and allow resizing, rearranging, or adding and removing widgets.

While in Edit Mode, each of the widgets on the dashboard can be resized according to your needs by clicking and dragging the bottom right corner of the widget.

Widgets can also be rearranged within the dashboard by clicking and dragging on the widget heading area as indicated by the "cross-hair" mouse cursor icon.

When satisfied with the configuration, save your customization settings by clicking the "Save" button so that you can quickly get to the information you'd like to see any time you visit the dashboard.

If you wish to revert the dashboard to its original state, click the “Cancel” button to destroy your changes. This will reset the visibility, arrangement, and size of all widgets to reflect the current view of the dashboard prior to customizing it.

Sharing dashboards

A dashboard may be shared with any number of organizations by clicking the “Share” button. In the dialog, select the organizations to share the dashboard with and click “Done” to save the settings.

When a dashboard has been shared with your organization by another user, the shared dashboard will appear under the “Shared with me” heading in the dashboard selector.

Dashboards that you share with others will still appear within the “My dashboards” section, and can only be modified by you, but may be duplicated by other organization users if desired.

Duplicating dashboards

When viewing a runZero-managed dashboard or one shared with your organization, it cannot be modified directly, but you may make a copy of it to customize as desired by clicking the “Duplicate” button.

Dashboards can also be duplicated by clicking the menu button at the top of the dashboard and selecting “Duplicate dashboard”. In this case, a dialog will open that allows you to set the name of the new dashboard to distinguish it from the original version if desired.

Note: sharing settings from the original dashboard will not be copied to the new dashboard.

Default dashboards

A user may choose to set any dashboard as their preferred dashboard by clicking the dashboard menu button and selecting “Set as my preferred dashboard”.

An organization admin can also set the default dashboard for all users of the selected organization by selecting the “Set dashboard as organization default” option. Within the dialog, select the “Override user-preferred dashboard” checkbox to override the user’s preferred default dashboard if desired. When configured as such, users will only see their preferred dashboard if the organization doesn’t have a default dashboard set to override their preference.

Deleting dashboards

A custom dashboard can be deleted by clicking the menu button at the top of the dashboard, selecting “Delete dashboard” and confirming the intention in the dialog that opens. This cannot be undone.

Widget library

The dashboard widget library is used to add and remove widgets on the dashboard, and can be accessed by clicking the “Widgets” button at the top right of the dashboard.

When you open the widget library, a list of all the available widgets is presented for you to select from. You may also choose to create your own custom widget based on queries.

Widgets included on the dashboard will be marked with a checkmark icon. When clicked, the widget will be deselected and removed from the dashboard when you click the “Save” button.

Conversely, widgets that are not currently visible on the dashboard can be added by clicking on the widget to select it, and then saving the changes.

Widget types

Operational information

- Live assets: number of assets currently alive based on the latest scans
- Active scans: number of scans currently in progress
- Explorers online: number of Explorers reporting as healthy
- Accounts: number of users with access to your console

Trends

- Asset trends: timeline view of asset counts broken out by live, recent, and a couple other states
- Service trends: timeline view of network services seen in your environment

Insights

- Latest insights: automated queries that were ran after your latest scan with result counts

Most and least seen graphs

A variety of asset views by different asset properties. All of these graphs can be toggled between most and least seen based on the counts.

- Asset types
- Operating systems
- Hardware
- RRT latency
- MAC vendors
- Newest MAC age
- TCP ports

- UDP ports
- Products
- Protocols
- Address count
- Extra address count
- Risk
- Criticality
- Tags

Goals

- Goals overview: summary view of all goals that are pinned to display on the dashboard

Asset count and trend tables

A view into the latest count and historical timeline for your assets related to ownership coverage, source types, and custom integrations.

- Asset ownership: view of all ownership types including assets missing owners
- Asset sources: view of all sources of asset data from supported integrations
- Custom integrations: view of all custom sources of asset data from API integrations

Bookmarks

A customizable list of frequently visited pages, favorite reports, or external links. After adding the widget to your dashboard, you can manage the list by using the menu button at the top of the widget and selecting the appropriate menu item.

Bookmarks can be edited or deleted by selecting the “Edit bookmarks” menu item, selecting the bookmark to be edited from the list shown in the dialog, updating the name or URL below, and saving the change. Similarly, a bookmark can be removed by clicking the trash can icon within the list if desired.

Additional bookmarks can be added by selecting the “Add bookmark” menu item, entering a name and URL, and then saving the change. Bookmarks to external sites will open in a new window, while internal console URLs will open in the current window.

When viewing individual reports or browsing the list of reports, you can easily add any report to your bookmarks by clicking the bookmark ribbon icon. After a report is bookmarked, you can remove it by clicking the ribbon icon again, which will open the editing dialog and allow you to remove the bookmark.

Custom widgets

Several widget types can be created and customized using the “Create widget” button in the widget library. These widgets fall into 4 types:

- **Single match count** - Displays the most recent match count for a given query.

- **Single match trend** - Displays a trend chart detailing the history of a given query.
- **Multi-query match count** - Displays the most recent match count for 2 or more given queries, displayed similar to the *Organization overview* widget.
- **Multi-query trend line** - Displays a trend chart detailing the history of 2 or more given queries.

Custom widgets can be customized in several ways. For single-query widgets, you can choose a name, query, and a color to style the results with. In the case of a multi-query widget, you can choose a name for the widget itself, and you can choose a name and color for each individual query as well.

Any custom widgets that display as a trend chart can be exported to JPG, PNG, or SVG formats via the menu button in the top right of the widget on your dashboard.

After creation, custom widget templates can be managed in the widget library by selecting the appropriate icon at the bottom of each widget's preview image. When updating a custom widget, you will be prompted to also update the widget on the dashboard if desired. Similarly, when deleting a custom widget, you will be prompted to also delete the widget from the dashboard if desired.

Drill down

Except for timeline-based charts, each widget's distinct data points can be viewed in detail by clicking on the relevant table row links or buttons, or by clicking on individual segments of donut, bar or column charts.

Printing and exporting

All chart-based widgets on the dashboard support exporting to SVG, PNG, and JPEG file formats, and table-based widgets can be exported to CSV format, by using the menu button at the top of the widget and selecting the file format you wish to export.

The entire dashboard and all of its widgets can be printed by selecting the printer icon at the top of the dashboard. For best results, use a Chromium-based browser or Firefox with the 'Print backgrounds' option enabled to print the dashboard.

Display mode

The dashboard supports displaying in theater (or kiosk) mode, where navigation menus are hidden and only the dashboard content is shown. To enable theater mode, click the menu button at the top of the dashboard and select the "Enter theater mode" option.

When theater mode is active, the dashboard also supports fullscreen display by clicking the *Fullscreen* button at the top of the page. In order to exit fullscreen mode, press the `Escape` key or click the *Exit fullscreen* button to return to theater mode.

To exit theater mode, click the X icon to return to the normal browsing experience.

Using the inventory

The inventory page is the heart of runZero Network Discovery and the key to understanding what is on your network. The inventory displays all assets within the Organization and can be sorted, filtered, and exported to obtain specific views of the environment.

Understanding assets

An asset within runZero is defined as a unique network entity. Assets may have multiple IP addresses and MAC addresses and these addresses may change as the environment is updated. runZero tracks assets based on several heuristics, including MAC address, IP address, hostnames, and fingerprint results for the operating system and running services.

In most cases, runZero can accurately follow assets over time in environments using DHCP, even across remote subnets. For external networks, scans that are initiated with fully qualified hostnames will consolidate assets based on the hostname, which allows for consistent asset tracking for cloud-based external systems with dynamic IP addresses.

Within an organization, assets are isolated by site, and each site can have address space that overlaps with other sites. Sorting the Inventory view based on the site column can help in these scenarios, as can filtering the Inventory based on a specific site name.

The search field allows the inventory to be filtered based on the specified criteria. Please see the search query syntax documentation for specific details.

In addition to viewing assets, the Inventory page provides data export functionality, along with the ability to select assets, and specify the comments field. The *Rescan* action can be used to selectively rescan specific systems from the inventory, while the *Remove Assets* and *Purge Assets* can be used to permanently remove data from the inventory view.

The *Reports* button provides quick access to key reports from the runZero reports page.

Loading assets

Data is loaded into the inventory using the **Scan** and **Import** buttons. The results are analyzed and merged, updating asset information as necessary.

The **Scan** button has two options: Standard Scan and Full RFC 1918 Discovery. The latter is an easy way to set up a fast scan of all private range IP addresses. You can then use the coverage reports to check for assets in unexpected private address ranges.

The **Import** button has two options. Importing runZero scan data allows you to import data. This means you can scan networks that have no connectivity to the internet, and still view the results in the runZero console. It's also useful for reprocessing old scan data so that you can use the site compare feature to see how assets have changed over time.

Bulk asset update

The bulk asset update feature allows you to modify assets by exporting a CSV using the **Export** button, making changes to the data in a spreadsheet program or text editor, and then importing the result back into runZero with the **Import** button. This feature will update existing assets that have a matching `id` value in the organization.

The fields listed below can be updated through the bulk asset update:

- Type
- Operating system
- OS version
- Hardware
- Comments
- Tags
- Owner
- Names
- Domains

The `type`, `os`, `os_version`, and `hardware` fields only accept a single value. The `comments`, `tags`, `owner`, `names`, and `domains` fields each accept multiple values, and a space-delimited list of `field=value` pairs is the standard syntax. The `tags` field can also be specified without `tag=` as just a space-delimited list of values.

Only modifications to the `tags`, `comments`, and `owners` fields will be retained through subsequent scans, any changes to the other supported fields will be overwritten by the latest scan data.

Connecting to other systems

Community Platform

The **Connect** button lets you connect runZero to other systems. The integrations you're able to connect depends on your license level, but may include tools like cloud and viritualization platforms, endpoint protection solutions, identity and access management tools, and vulnerability and risk platforms. These inbound integrations can also be configured as scan probes if required.

Viewing services

The Inventory page has a submenu labeled *Services*. This changes the table of data from an asset-focused view to a service-focused view. For each asset, you will see one row for each service runZero detected.

Like the main asset view, the services view has a full search interface. You can filter services by protocol, port, and many other criteria, using the runZero search language.

Viewing screenshots

If the runZero Explorer has access to Google Chrome, it will attempt to take screenshots of web pages it finds while scanning your network. (This feature can be disabled in the scan options when setting up the scan.)

You can view the screenshots for all of your assets via the *Screenshots* submenu, and click through to the asset records for full details.

Viewing software

The inventory page has a submenu labeled *Software*. This flips the table of data from an asset-focused view to a software-focused view. For each asset, you will see one row for each software detected by runZero or a supported integration.

Like the main asset view, the software view has a full search interface. You can filter software by vendor, product, and many other criteria, using the runZero search language.

Viewing vulnerabilities

The inventory page has a submenu labeled *Vulnerabilities*. This flips the table of data from an asset-focused view to a vulnerability-focused view. For each asset, you will see one row for each vulnerability detected by a supported integration.

Like the main asset view, the vulnerability view has a full search interface. You can filter vulnerabilities by CVSS score, name, CVE, and many other criteria, using the runZero search language.

Viewing certificates

The *Certificates* submenu provides a searchable, sortable certificates inventory of all of the encryption certificates the runZero Explorer encountered while scanning.

Like the other views, the certificates view has a search interface and can be sorted.

Viewing wireless networks

If the machine running the runZero Explorer has a working WiFi adapter and appropriate system tools installed, the Explorer will attempt to scan for nearby wireless networks. The *Wireless* submenu will show the results of the scan.

The tools required are:

- Windows: `netsh.exe` (part of modern Windows releases)
- macOS: Airport Utility
- Linux: `iwlist`, often available via the `wireless-tools` package.

Viewing users and groups

The inventory pages for users and groups contain data imported from directory services such as Active Directory. The user directory and group directory can be populated using third party integrations listed under *Directory services* in the *Integrate* drop-down menu.

Understanding assets

runZero treats assets as unique network entities from the perspective of the system running the Explorer. An asset may have multiple IP addresses, MAC addresses, and hostnames and it may move around the network as these attributes are updated. runZero tries hard to follow assets by correlating new scan data with the existing inventory, using multiple attributes.

An asset is always associated with a single site. If the same system happens to be covered by multiple sites, these will be treated as different assets, and will only be correlated against assets within their respective site. This separation by site allows the same network to be scanned from multiple perspectives and compared in a single view within the organization.

After each scan, all assets within the corresponding site are updated. If a system is identified that doesn't match an existing asset, a new asset will be created. If an asset is part of the site and it is not found during a scan, it will be marked as offline. If an asset is not correlated, due to substantial changes to the fingerprint (for example, a new network adapter was installed and the firewall was enabled), the previous asset will be marked as offline, and a new asset will be created to track the new configuration. This can lead to some level of duplication within a site, but these duplicates are usually marked as offline, and can be safely ignored or removed from the inventory by hand.

Asset fields

The following asset fields are available.

Primary addresses

runZero will report at least one and often multiple primary IP addresses for a given asset. These addresses can encompass multiple network interfaces but will only be displayed as a primary address if runZero has scanned it. This requires that the address is within the scan scope of one or more runZero scans.

Secondary addresses

runZero may report one or more secondary addresses, based on network response probes. These are IP addresses that were detected on the asset but were not within the scan scope. Secondary address detection is critical when trying to identify systems that bridge networks that should be isolated.

Hostnames

runZero may report one or more hostnames. These names can be obtained from the initial DNS lookup (when hostnames are provided in the scan scope), from DNS PTR lookups during the scan, and by extracting names advertised within network probe responses.

Operating System (OS)

runZero attempts to fingerprint, and failing that, guess at the operating system running on each asset. If limited information is available, this field may be empty.

Type

runZero attempts to determine the general device type through analysis of fingerprints and running services.

Hardware

runZero attempts to determine the physical (or virtual) hardware if enough information is present.

MAC addresses

runZero may be able to enumerate one or more MAC addresses from the asset. MAC addresses are pulled from ARP if available, but also several network services that can return MAC address information across routed segments.

Services

runZero tries to detect approximately 100 TCP services by default, along with several useful UDP services. These services are in addition to ARP and ICMP. The services field contains a list of the most recently recorded services for the asset.

Round Trip Time (RTT)

runZero records the amount of time certain probes take in order to get a rough sense of the latency between the Explorer and the asset.

Detected by

runZero records which probe was used to identify an asset. For assets that are on remote subnets and have firewalls in place, this field indicates what service was used to obtain a response.

Alive status

runZero tracks whether a given asset was found during the most recent scan where its site was in scope. If the asset was not found, it will be marked as offline until a following scan

detects it again.

First seen

runZero tracks the initial timestamp when an asset was first identified.

Last seen

runZero tracks the last timestamp when an asset responded to a probe during a scan.

Outlier score

runZero computes an outlier score for all assets in your inventory. The outlier score has a value from 0 to 5 (inclusive). It is a heuristic that aims to indicate how unusual the asset is, compared to all of the others in the inventory.

Outlier scores are computed by examining key properties of the asset and its services, working out which values are unusual (infrequent) across the organization, and then computing how many unusual properties each asset has. The more unusual properties, the higher the asset's outlier score will be.

Understanding findings

Findings simplify vulnerabilities, misconfigurations and best practices into a prioritized, curated and aggregated list that helps you identify and remediate the most critical risk in your environment. runZero Findings are available from the *Findings* menu, and from the Risk Management dashboard.

What are findings?

Findings highlight the risks attackers are most likely to target. This enables security teams to focus remediation efforts on risks with real operational impact.

Findings group similar vulnerabilities and misconfigurations together, providing a more holistic view of the risk they represent. Each finding contains a description of the risk, remediation steps, risk rankings, and an individual list of all assets and entities affected by the risk.

Finding categories

Each finding is placed into a category based on the type of risk it represents. Categories include:

- Internet Exposure: identifies assets and services that are potentially unintentionally exposed to the internet
- Certificates: expired or soon-to-expire certificates as well as widely shared private keys
- Vulnerability: actively exploited vulnerabilities or critical vulnerabilities that runZero believes are critical to address
- End-of-Life: operating systems, hardware and applications that have reached End-of-Life (EOL) or End-of-Service (EOS), and are no longer supported by the vendor
- Open Access: network services such as unauthenticated databases and sensitive applications that are accessible without authentication
- Compliance: assets and services that violate security best practices
- Best Practice: general best practices that cover insecure authentication, service misconfiguration and obsolete protocols
- Rapid Response: emerging and novel threats covered in detail by runZero Rapid Response blog posts

Findings list

Findings can be found by using the main navigation menu directly below the Inventory item. They can be searched, sorted and exported similarly to other views within the console.

Finding details

When clicking on an individual finding's name to see its details, you will see an overview of the risks discovered, any associated external resources, and pertinent remediation information to

help you address the risk. Below the finding summary, you will find a list of all the asset instances that the finding applies to, so that you can quickly prioritize remediation.

How are Findings different from Vulnerabilities?

A vulnerability is a specific security issue, usually closely tied to a specific CVE or security advisory. Vulnerabilities are found by the runZero Explorer or imported via one of the many supported integrations. There can be hundreds of thousands of vulnerabilities for any given asset, and even more across your entire environment.

A finding is a curated, aggregated and prioritized list of risks that are most likely to be targeted by attackers. A single finding may group together several similar vulnerabilities. Findings are not always tied to a CVE and may include misconfigurations, best practices, and other security issues that may not make sense as a vulnerability.

Asset risk and criticality

Community Platform

runZero is able to help users assign and evaluate risk and criticality levels to the assets in their inventory. This can help prioritize risk mitigation or vulnerability remediation efforts by allowing users to quickly identify the assets in their organization with the highest levels of risk or criticality.

Defining risk and criticality

The **risk** level assigned automatically to assets in your inventory is inferred from the risk associated with vulnerabilities or risky configurations on that asset and defaults to the value `none`. Vulnerability risk level may be defined by the vulnerability management solution the vulnerability records are ingested from, or by the risk level assigned to a query vulnerability. The risk level can be overridden, in which case the override is retained until the asset or vulnerability is deleted. For vulnerabilities ingested from integrations, this may occur when the source no longer reports the vulnerability on that asset.

The **criticality** level is assigned manually and defaults to the value `unset`. This value is intended to be used to denote the criticality or importance of an asset to your organization. As an example, you may choose to assign business-critical systems such as database and web servers a `critical` level, but normal enduser systems a `medium` level.

Assigning asset risk and criticality

Both asset risk and criticality can be assigned via the asset inventory. Asset criticality can also be assigned with alert rules.

Superusers, administrators, and users can add or modify asset risk and criticality levels, and can reset risk assignment or remove criticality assignment from assets.

Risk and criticality in the asset inventory

Follow these steps to set risk and criticality through the asset inventory:

1. Select all the assets you wish to update, applying a query filter if needed.
2. Click the *Modify asset risk* or *Modify asset criticality* button to open the relevant popup.
3. Select the level of risk or criticality you wish to apply to the asset(s).
4. Click *Override risk* or *Set criticality* to apply your changes.

Applying criticality with rules

To automatically apply asset criticality values to assets after a scan, create an alert rule by going to **Alerts** > **Rules** and clicking the *Create rule* button:

1. Select an inventory query you wish to use, such as the `asset-query-results` rule type, then click *Configure rule*.
2. Configure any desired settings.
3. Set the **Action** to *Modify asset*.
4. Select an option from the *Asset criticality* menu.
5. Save the rule.

This rule will now add the specified asset criticality level to all assets that match the rule when a scan completes.

Asset risk report

The **Asset risk report** provides visibility into the risk and criticality levels across your asset inventory. To run the Asset risk report, go to **Reports** > **Asset risk report** and click the **Asset risk report** button. Configure the following fields:

1. **Sites**: Select a site of assets to include in the report, or leave the default *All Sites*.
2. **Minimum risk**: Choose the minimum asset risk level to include.
3. **Minimum criticality**: Choose the minimum asset criticality to include.
4. **Top vulnerabilities per asset**: Set this field to an integer between 0 and 20. If the value is set to an integer between 1 and 20, the report will list up to that number of the top vulnerabilities detected on each asset. The top vulnerabilities are identified by sorting the vulnerability results for each asset by risk rank, then risk score, then severity rank, then severity score.
5. Click **Create report** to generate the results.

The resulting report is grouped by asset criticality level and then sorted by risk level. The results can be exported as JSON Lines (.jsonl), a JSON document (.json), or CSV (.csv).

Managing ownership

Platform

runZero is able to help users track ownership with the ability to configure different types of owners and assign owners to runZero assets and vulnerability records. Ownership coverage can also be tracked as a goal.

Ownership types

Superusers can manage the available types of ownership on the **Account** > **Ownership types** page. Custom ownership types can be configured to meet your needs. Some common ownership types may include **Security owner**, **IT owner**, or **Compliance owner**.

The ownership type requires configuring three fields:

- **Name**: the name of the ownership type.
- **Reference**: whether the ownership type should be correlated with the user inventory, group inventory, or neither.
- **Visibility**: whether the ownership type is visible through the asset inventory and asset details pages.

The default `Asset Owner` ownership type, when visible, will be automatically populated with ownership-related data that runZero can glean from your configured integrations. The name of this ownership type can be changed by a superuser.

The list of ownership types can be prioritized by dragging the types into the preferred order. This will dictate the order in which the types are displayed in the inventory and asset details pages. Only types marked `visible` will be displayed.

Assigning owners to assets and vulnerabilities

Once created, custom owners can be assigned via the inventory or through an alert rule.

Superusers, administrators, and users can add or modify owner values, and can remove owners from assets or vulnerability records. Annotators can only add owner values, but cannot modify or remove owners.

Ownership in the inventory

Follow these steps to assign owners through the asset or vulnerability inventory:

1. Select all the assets or vulnerability records you wish to update, applying a query filter if needed.
2. Click the *Manage asset ownership* or *Manage vulnerability ownership* button to open the ownership popup. **Note**: Ownership values applied to an asset will be inherited by

unowned vulnerability records on that asset. Vulnerability records with owners defined will not inherit the ownership value assigned to the asset.
3. Click *Add ownership type* and choose which type(s) of owner you wish to apply to the selected assets or vulnerability records.
4. Add the owner value to the field.
5. Click *Save* to apply your changes.

Applying owners with rules

To automatically apply ownership values to assets after a scan, create an alert rule by going to **Alerts > Rules** and clicking the *Create rule* button:

1. Select an inventory query you wish to use, such as the `asset-query-results` rule type, then click *Configure rule*.
2. Configure any desired settings.
3. Set the **Action** to *Modify asset*.
4. Specify a value for the *Set [ownership type]* field for the ownership type(s) you wish to apply. **Note**: Ownership values applied to an asset will be inherited by unowned vulnerability records on that asset. Vulnerability records with owners defined will not inherit the ownership value assigned to the asset.
5. Save the rule.

This rule will now add the specified owner type and value to all assets that match the rule when a scan completes.

Managing tasks

You can view and manage discovery scans and other background actions from the Tasks overview page. The *Active* and *Completed* task sections will show standard tasks, such as scans and imports, along with their current progress and summarized results. You can search or filter the tasks using different attributes.

Task status values

Tasks can have the following status values:

- **New**: The task has been created and is waiting to be picked up by a scanner or connector.
- **Active**: A scan task is in progress and the Explorer is scanning the network.
- **Scanned**: The network scan part of a scan task has completed.
- **Connecting**: A connector task is connecting to the remote system and downloading data.
- **Connected**: A connector task has finished downloading and the data is waiting to be processed.
- **Processing, queued**: Task data has been collected from a scan or a connector task, and is queued for processing.
- **Processing**: Task data is being processed.
- **Processed**: Task data has been processed and runZero data updated.
- **Stopping**: The task was requested to stop and is in the process of doing so.
- **Stopped**: The task successfully stopped.
- **Canceled**: An error occurred which meant that the task could not continue.
- **Paused**: A repeating task has been paused.

Tabs

Tabs on the Tasks overview page allow you to view a filtered subset of active, processing, scheduled, failed, completed, or recurring tasks. Each tab includes a search bar so that you can search your complete task history.

- **Active** shows all tasks currently scanning or connecting
- **Processing** shows all tasks currently processing
- **Scheduled** shows all non-recurring tasks that are scheduled to run
- **Failed** includes all tasks that were unable to finish due to an error or cancellation
- **Completed** lists all completed tasks in chronological order based on when the task finished processing
- **Recurring** shows all recurring tasks

Task details

Clicking on a task will open a card at the top of the page with details. The concentric circles show the progress of the task, with the outer circle tracking the scan or connection progress

and the inner circle tracking the processing progress. The card also includes relevant timestamps and user creation data, as well as site, Explorer, and hosted zone settings. Completed tasks will also include the change summary, and recurring tasks will include a list of past runs.

Scheduled tasks

Scheduled tasks are one-off actions that will be started in the future, while recurring tasks are actions that happen on a regular basis. Recurring scan tasks generate a new standard task on each iteration of their schedule. Both scheduled and recurring scans will only launch if their associated Explorer is online and no other scan tasks are running.

The Tasks page allows *Scheduled* and *Recurring* tasks to be removed and *Active* tasks to be stopped. Please note that stopping an active *Scan* task may take a few moments, as the status of the task is not updated until the Explorer confirms that the scan was terminated.

System tasks

runZero has system tasks that you may notice while searching through your tasks. Here are some examples you might see:

- **Report-based**: a few reports will create tasks when initialized, including:
  - Organization overview report
  - Site and organization comparison
  - External assets report
- **Outlier calculation**: this generates the outlier score seen in the asset details page and will show up with the name `Outlier calculation`
- **Query**: all queries that are pinned to the Dashboard as Insights will show up with the name Query

Dismissing failed tasks

Tasks that were unable to finish due to an error or cancellation can be removed from the runZero Console by viewing the task details, clicking the “Dismiss” button and confirming the desired action. Dismissing failed tasks will archive the tasks so that they are no longer accessible.

Reprocessing tasks

You can use the Reprocess Tasks tool on the tasks overview to delete existing assets in the current organization and repopulate asset data from previous scan, import, and integration tasks using runZero’s latest algorithms. This can potentially fix duplicated or incorrectly merged assets. To reprocess tasks:

1. On the Tasks Overview page, open the hamburger menu at the top right of the table and select “Reprocess tasks”;
2. Review the warnings in the modal that opens;

3. Enter a number of days to reprocess (this defaults to 7);
4. Enter the confirmation text;
5. Click “Continue”.

The process will take some time to delete the organization’s assets and queue tasks that were created within the number of days specified for reprocessing. When it has finished queueing tasks, it will display the number of tasks to be reprocessed. You will then be able to monitor their progress in the console as normal.

Please note:

All assets in the organization will be deleted.

Any tags, risk, criticality, comments, or other manually applied changes will be removed.

During reprocessing, asset information may be incomplete and alert rules may be triggered.

If your organization has a very large number of assets, it is possible for the request to time out before completing asset deletion. In this case, you can retry the request.

The entire process may take several hours to complete.

Tracking goal progress

Community Platform

With runZero goals, users are able to create and monitor progress toward achieving security initiatives. All goal types are supported by the robust query language on the backend. All types of inventory queries are supported by the goal tracking feature.

There are three types of goals:

- **Saved query** - a goal based on the results of a user-defined or runZero system query.
- **Asset ownership** - a goal that measures progress toward assigning owners to assets in your inventory.
- **Asset risk** - a goal that measures progress toward reducing the number of assets with a minimum risk level in your inventory.

Goal creation

New goals can be created by users whose default role is user or greater from the Goals page in the console. Users with viewer-level or greater access will be able to view the goals page and see the goals that apply to organizations they have access to.

To create a new goal, click the *New goal* button on the *Goals* page in the console.

Creating saved query goals

1. On the *Goal type* tab, select the **Saved query** type.
2. On the *Build goal* tab, complete any desired fields:
    - **Name**: Provide a name for your goal.
    - **Description**: Provide a description for the goal.
    - **Notes**: Add relevant notes about this goal.
    - **Organization access**: By default, a goal will be created for the currently selected organization, but you may select additional organizations as desired. Goals are editable by any user with User role access to every organization associated with the goal, and can be read by any user with Viewer role access to at least one organization.
    - **Query**: Determine the query you want to use to track progress for this goal by selecting the query from the table. If needed, use the search field to find the query you wish to use.
    - **Threshold**: Select whether goal progress will be calculated as *Greater than or equal* or *Less than or equal*, whether goal progress will be calculated as a percent or fixed number, and then set the desired threshold value.
    - **Target date**: Provide an optional target date for achieving this goal.
3. Click *Save* when you're ready, or make changes as needed.

Creating asset ownership goals

1. On the *Goal type* tab, select the **Asset ownership** type.
2. On the *Build goal* tab, complete any desired fields:
    - **Name**: Provide a name for your goal.
    - **Description**: Provide a description for the goal.
    - **Notes**: Add relevant notes about this goal.
    - **Organization access**: By default, a goal will be created for the currently selected organization, but you may select additional organizations as desired. Goals are editable by any user with User role access to every organization associated with the goal, and can be read by any user with Viewer role access to at least one organization.
    - **Ownership type**: Select the ownership type you wish to track for this goal. New ownership types can be created by users with a default role of administrator or superuser from the Ownership page.
    - **Target coverage**: Set the target coverage for the selected ownership type.
    - **Target date**: Provide an optional target date for achieving this goal.
3. Click *Save* when you're ready, or make changes as needed.

Creating asset risk goals

1. On the *Goal type* tab, select the **Asset risk** type.
2. On the *Build goal* tab, complete any desired fields:
    - **Name**: Provide a name for your goal.
    - **Description**: Provide a description for the goal.
    - **Notes**: Add relevant notes about this goal.
    - **Organization access**: By default, a goal will be created for the currently selected organization, but you may select additional organizations as desired. Goals are editable by any user with User role access to every organization associated with the goal, and can be read by any user with Viewer role access to at least one organization.
    - **Minimum risk**: Select the minimum level of risk you wish to track for this goal.
    - **Minimum criticality**: Set the minimum criticality of the assets tracked by this goal (optional).
    - **Target percent**: Set the percent threshold you aim to be under. As an example, you might define a goal to have less than 25% of your critical assets with a risk level of medium or higher by setting **target percent** to `25%`, **minimum criticality** to `critical`, and **minimum risk** to `medium`.
    - **Target date**: Provide an optional target date for achieving this goal.
3. Click *Save* when you're ready, or make changes as needed.

Goal progress calculation

Once a goal has been created, the progress will be calculated once a day and when tasks complete. As a result, you may see a different number of metric calculations in the historical line chart depending on the number of tasks that are completed on a given day.

The *overall status* noted in the *Goal details* card on a goal details page reflects the progress toward goal completion for all the configured organizations that you have access to. This may differ for a user with more or less per-organization access.

Note: When a goal applies to multiple organizations, progress is calculated per-organization and extrapolated across all applicable organizations.

Goal events and notifications

The events system includes the following events related to goals:

- `goal-created`: This event fires when a goal is created and saved.
- `goal-updated`: This event fires when a goal is modified and saved.
- `goal-completed`: This event fires when the progress of a goal crosses the completion threshold.
- `goal-lapsed`: This event fires when the progress for a previously completed goal falls below the completion threshold.
- `goal-removed`: This event fires when a goal is deleted.

These events can be used with channels and rules to receive notifications about a change in goal status. After a notification channel has been created, create a rule for the desired event and select the channel you wish to receive the notification through.

Understanding network segmentation

runZero multi-homed asset detection

Network segmentation is a critical security control for many businesses, but verifying that segmentation is working correctly can be challenging, especially across large and complex environments. Common techniques to validate segmentation, such as reviewing firewall rules and spot testing from individual systems can only go so far, and comprehensive testing, such as running full network scans from every segment to every segment, can be time intensive and are hard to justify on a regular basis.

For businesses subject to the PCI DSS requirements, validating cardholder data environment (CDE) segmentation is an important part of the security audit process. The PCI guidance on scoping and segmentation describes a common CDE administration model.

The network bridge detection in runZero is opportunistic and far from perfect, but it may highlight areas where segmentation is broken, and can cut down on the number of surprises encountered in a future security audit.

Using the bridge report

The bridge report shows external networks in red and internal networks in green. This view is not a typical network map, but instead shows possible paths that can be taken through the network by traversing multi-homed assets. Assets where runZero only detected a single IP address are not shown in order to keep the graph readable.

Zooming in will show asset and subnet details. Clicking a bridged node once will highlight the networks it is connected to, and clicking it a second time will either take you to the asset page. Clicking a network once will highlight the connections to bridged nodes, and clicking a second time will perform a CIDR-based inventory search.

Bridge detection is useful when validating network segmentation and ensuring that an attacker can’t reach a sensitive network from an untrusted network or asset. Examples of this include laptops plugged into the internal corporate network that are also connected to a guest wireless segment and systems connected to an untrusted network, such as a coffee shop’s wireless network that also have an active VPN connection to the corporate network.

runZero detects network bridges by looking for extra IP addresses in responses to common network probes (NetBIOS, SNMP, MDNS, UPnP, and others) and only reports bridges when there is at least one asset identified with multiple IP addresses. Typical hardening steps, such as desktop firewalls and disabled network services will usually prevent multi-homed assets from being detected by runZero. The screenshot below shows how to search for multi-homed assets in the runZero inventory.

Using the asset route pathing report

Platform

Network segmentation is a foundational security control that can be easily undermined by network misconfigurations and multi-homed machines. runZero Platform users can now visualize potential network paths between any two assets in an organization using the asset route pathing report.

This report generates a graph of multiple potential paths by analyzing IPv4 and IPv6 traceroute data in combination with subnet analysis of detected multi-homed assets–without requiring access to the hosts or network equipment. This unique methodology identifies surprising and unexpected paths between assets that may not be accounted for by existing security controls or reviews.

With a view of potential paths, security professionals can verify whether a low-trust asset, such as a machine on a wireless guest network, can reach a high-value target, such as a database server within a cardholder data environment (CDE). The new feature highlights potential network segmentation violations and opportunities for an attacker to move laterally from one segment to another.

Managing alerts

Community Platform

runZero can trigger automatic alerts when certain events occur through a combination of Channels and Rules.

runZero currently supports Internal, Email, and Webhook channel types.

Internal channels store events within the Alerts list within the runZero Console. Internal alerts support explicit acknowledgement. Internal alerts can be bulk acknowledged and cleared from within the runZero Console.

Email channels can be configured to deliver mail to one or more recipients. These email messages contain a summary of the alert and a link to the specifics within the runZero Console. Email is sent from the runZero infrastructure using the Sendgrid service.

Webhook channels allow runZero to post alerts to internet-reachable web services. The post request contains a standard text message for use with platforms like Slack and Mattermost, but also additional fields containing the full alert details. Webhooks are a great way to tie runZero alerts into third-party platforms.

To trigger an alert on a channel, a Rule must be created. Rules define which events lead to alert on which channels. The name of the rule will be included in the alert content and should describe the type of event that it monitors.

The following are some example event types that can be used to create rules:

- Scan completed
- New assets found
- Assets back online
- Assets now offline
- Assets changed

Scan completion and *assets changed* rules can be noisy but may be useful to keep a running log of network changes over time. For a typical monitoring use case, a rule would be created to trigger on *Assets now offline*, *Assets back online*, and *New assets found*, automatically alerting an email alias or a Slack channel.

Alert rules, when combined with recurring scans, can be a simple way to track network changes over time.

Using the rules engine

Community Platform

The Rules Engine is an automation framework for monitoring, alerts, and workflow management. You can use the Rules Engine to customize alerts for the events that matter most to your organization and automate repetitive tasks. At the heart of the Rule Engine are

rules. A rule defines the action that is taken based on a set of conditions. You can create rules to proactively alert your team when there are changes to things like Explorers, assets, scans, organizations, and sites. You can also automate tagging and modification of asset fields based on the results of a query.

Some ways you can use the Rules Engine to help automate your workflow:

- Alert your team when new policy violations are identified.
- Modify asset fields when the assets match specific criteria.
- Bulk tag assets that match a specific query.
- Get a Slack notification when a query returns new results.
- Monitor when an Explorer goes offline in the runZero console.
- Know when there are changes to organizations, sites, and users.

Key concepts

Rules can help you stay on top of events as they happen and get better visibility across your network, assets, and your runZero deployment. To build a rule, you need to define four things: events, organization access, conditions, and actions. A rule determines that when a specific event happens, and certain conditions are met, the system will automatically perform the configured action.

Events

Each rule begins with an event. The event sets off the trigger and puts your rule into motion. An event can be based on a query or a system-defined event. runZero offers a library of system-defined events you can use to create your rules. Choosing any of these events will show the conditions and actions available.

Organization access

Each rule can be used by any number of organizations. Rules are editable by any user with User role access to every organization associated with the rule, and can be read by any user with Viewer role access to at least one organization. By default, a rule will be triggered for the currently selected organization.

Setting organizational access on templates and channels only specifies who can view and edit, not which rules can use them. You can set up a rule for an organization with a template and/or channel whose access does not include that organization, as long as you have at least Viewer role access to one or more organizations in the template or channel’s organization access list.

Conditions

A condition narrows the scope of your rule. Unless the condition is met, the rule will not execute the action. You will only see conditions that apply for the event you have chosen. Generally, conditions specify sites, organizations, and asset attributes for the event.

Actions

An action executes your rule, if the event occurs, and the conditions meet all the criteria. An action can be a notification to a channel, or it can be a modification to an asset. What you will need to configure depends on the action type. For notifications, you'll need to specify the notification channel and template. For asset modification, you can edit fields like the OS vendor, OS product, OS version, hardware vendor, hardware product, hardware version, asset tags, and asset type.

Channels

A channel provides a way for you to communicate when a specific event has occurred. You can create multiple channels to support different types of communication needs. For example, you may want to create a Slack channel for one team, and an email list for another. It depends on what communication channels you prefer, and who you are trying to reach.

Much like rules, channels can also be utilized by any number of organizations, and managed by any user with User role access to all associated organizations. Similarly, any user with Viewer role access to one or more organizations will be able to view the channel details only, and may not edit or create new channels.

The body of the message uses default text from runZero. Customizations for messaging is currently unavailable.

Create a rule

Rules set the criteria for actions to take place. To create a rule, you need to choose an event, define the conditions, and choose a resulting action.

Step 1: Open the Rules Engine

- From the Alerts menu, select the Rules submenu.
- Click **Create Rule** to open the editor.

Step 2: Choose an event type and configure organization access

- Provide a descriptive name for the rule. Something that quickly that tells you what the rule does.
- Choose an event you want to use as your trigger.
- You can browse the list of available predefined events. Use the left-hand categories to narrow down the list, or the search field to quickly filter by keyword.
- Choosing 'asset-query-results' or 'service-query-results' will allow you to modify the fields for the resulting assets.
- After you've chosen an event and configured organization access, click **Next**.

Step 3: Define the conditions

- The conditions you can configure depend on the event you have selected.

- If you have an asset or service based query selected, you'll need to provide a query for the rule. This query will run against the site after the scan completes. Note that assets with data from non-runZero sources must be recent (seen in the last 30 days) to be included in the scope of the search, and runZero-scanned assets must be live.
- You may also set the scope to a specific site or Explorer, and sometimes, depending on the event, minimum asset counts or task type.

Step 4: Choose an action, and optionally select a specific channel or template

- Actions can execute a notification to the channel of your choice or modify assets. For example, you can choose to send notifications via email when orphaned devices are found.

Step 5: Turn on and save the rule

- Turn the rule on if you want to activate it immediately by selecting the "Enable this rule" checkbox. Otherwise, you can save the rule and turn it on later.
- Save the rule when you're done.

Keep in mind

Using scan and asset event types can be noisy, but they are useful for tracking network changes over time. To help you focus on the events that matter most, track assets that go offline, assets that come back online, and newly discovered assets.

Monitoring the status of rules

The rules submenu of the Alerts page displays a list of all rules that have been created. For each rule, you can see:

- Whether the rule is enabled.
- The event that triggers processing of the rule.
- The organizations the rule applies to, if the rule has been limited to specific organizations.
- When the rule was last triggered.
- Whether the rule resulted in an action being processed or not.
- When the rule was created and the username that created it.

A status of "skipped" means that last time the rule was processed, its preconditions weren't met, so no action was taken. A status of "processed" means that the rule's preconditions were met, and its action has been processed.

If there is an error processing a rule or sending a notification, the action status of the rule will be set to "error". The error message can be seen as a tooltip on the error status.

Creating alert templates

Community Platform

With the Rules Engine, you can define rules that alert you on specific events, such as changes to scans, assets, and Explorers. To customize the alert messages, you can create custom templates to standardize and format alerts triggered from rules. With custom templates, you can include more context and data for your alerts.

Templates can output in HTML, JSON, and text for use in emails, internal notifications, or webhooks. You can customize the contents of these templates as needed.

Like rules and channels, templates can also be utilized by any number of organizations, and managed by any user with User role access to all associated organizations. Similarly, any user with Viewer role access to one or more organizations will be able to view the template details only, and may not edit or create new templates.

Template building basics

You can define the contents of an alert message using the Mustache templating language. As long as you know a little bit about how the Mustache syntax works, you can build custom HTML and JSON templates to reference and pull in runZero data.

- For Slack notifications, you can use Slack's Block Kit and their interactive Block Kit Builder to construct a rich message in JSON format, and then use Mustache to insert the relevant data.
- For Microsoft Teams, you can use the Adaptive Cards format to build rich messages, and again insert data via Mustache.

Our templates have two fields for template data: subject and body. Both subject and body can be customized using the Mustache syntax.

Inserting a data in a template

A standard set of objects is passed to the {{template engine}}, you just have to indicate the fields from the objects you want to insert into your template. Use the Mustache template syntax, `{{variable}}`, to include alert values when a rule matches certain conditions.

For HTML and JSON templates, values inserted using `{{ }}` are automatically escaped according to the appropriate rules. To avoid escaping a particular value, use triple curly braces, like this: `{{{task.name}}}`.

For JSON templates, the special self variable `{{.}}` at the top level will output all available variables and their values as JSON.

Special rules for JSON

For JSON templates, variables are rendered differently depending on whether they are single values or multiple values:

- A string or number gets rendered to the raw value. A string will have any embedded quotes escaped, but it won't be wrapped in quotes. This is so you can put multiple strings or numbers into a single JSON string in your template.
- An object or array gets rendered to the full JSON representation of the object or array. This is for convenience, so that you can dump arrays and objects to JSON without having to loop through values. However, it means that if you try to put an array value into a JSON string by surrounding it with quotes, the result won't be valid JSON.

Consider the following template running on an asset with multiple IP addresses:

```
{
"addresses_1": {{addresses}},
"addresses_2": "{{#addresses}}{{.}} {{/addresses}}",
"addresses_3": "{{addresses}}"
}
```

Here's the result:

```
{
"addresses_1": ["10.0.1.4","10.1.7.5"],
"addresses_2": "10.0.1.4 10.1.7.5 ",
"addresses_3": "["10.0.1.4","10.1.7.5"]"
}
```

The `addresses_1` substitution works as you might want — `{{addresses}}` is an array, so it gets replaced with a proper JSON array containing the addresses.

The second substitution also works. In that case, the template loops through each address in the array, and puts the values into a single string. The values don't get put in quotes, but they will have any quotes inside them escaped.

The third substitution doesn't work properly — the addresses get turned into a JSON array of strings, but that gets surrounded by the quotes, resulting in invalid JSON.

Inserting a value

To insert a value, put the variable name in double curly brackets, like this `{{variablename}}`.

The following example shows how to insert the console address:

```
The runZero Console is at {{console}}.
```

Inserting a value from an object

To insert a value from a specific object, separate the object name and field name with a dot, like this `{{object.fieldname}}`.

The following example shows how to insert the organization name:

```
The organization name is {{organization.name}}.
```

Inserting multiple values from an object

To insert multiple values from an object easily, use a section. You will need to start the section with `{{#objectname}}` and close it with a matching `{{/objectname}}` The following example shows how to insert the results from a scan that include total assets and number of assets changed:

```
Here are the results:
{{#scan}}
Scan found {{assets_total}} assets and changed {{assets_changed}} of them.
{{/scan}}
```

What happens if a field contains multiple values?

If a section refers to a field which contains multiple values, the template engine will loop through the values in the field, processing the section inside for each individual value.

The following example loops through all of the assets in `report.new`, and for each one, outputs its `names` and `addresses` fields. If there is nothing stored in `report.new`, the section between the tags will not be rendered.

```
{{#report.new}}
{{names}} {{addresses}}
{{/report.new}}
```

Note that within a loop, you can still refer to values from the outer object. If a named value isn't found in the current loop object, the template engine will check the outer object. For example, this can be useful for referring to the `{{console}}` variable, which provides the root runZero console URL.

Using boolean values

You can use boolean values with the `{{#field}}{{/field}}` tags. If the value of the field is false, the section between the tags is not rendered.

For example:

```
{{#query.truncated}}
(Additional results were found but not included in this report)
{{/query.truncated}}
```

Objects and fields reference

To include runZero data and details in your alerts, you can build your template using the following objects and fields.

globals

Field	Contents	Example
`console`	The base URL of the runZero web console.	https://console.runzero.com

event

The following fields are available in the `event` object:

Field	Contents	Example
`action`	The action which triggered the event	task-completed
`created_at`	When the event was created	2021-04-02 12:50:26 -0500
`id`	The UUID of the event	b4b871db-bdf1-4a42-b82d-18ae99972228
`source_name`	Name of the thing which caused the event	Weekly security scan
`source_type`	Type of thing which caused the event	task
`success`	Whether the event succeeded	true
`target_name`	Name of the object targeted by the event	Head Office
`target_type`	The type of the object targeted by the event	site

task

For events triggered by a task, the following fields are available in the `task` object:

Field	Contents	Example
`created_at`	When the task was created	2021-04-02 12:50:26 -0500
`created_by`	The user who created the task	user@example.com
`description`	The description of the task	Weekly scan of main network
`error`	The text of any error message for the task	explorer unavailable after 4h
`id`	The UUID of the task	b4b871db-bdf1-4a42-b82d-18ae99972228
`name`	The name of the task	Weekly Scan
`start_time`	When the task started	2021-04-02 12:50:26 -0500

`status`	The status of the task	processed
`type`	The type of task	scan
`updated_at`	When the task was last updated	2021-04-02 12:50:26 -0500
`url`	A URL linking to the task details	

organization

The following fields are available in the `organization` object:

Field	Contents	Example
`id`	The UUID of the organization	86f12ee1-f0f1-419a-8799-63fff555777a
`name`	The name of the organization	IT Dept.

site

The following fields are available in the `site` object:

Field	Contents	Example
`id`	The UUID of the site	49f9323a-fea1-4afc-b490-2414c3aaaeee
`name`	The name of the site	Head Office

rule

The following fields are available in the `rule` object:

Field	Contents	Example
`action`	The action the rule said to take	notify
`created_at`	When the rule was created	2021-03-08 12:43:59 -0600
`created_by`	The user who created the rule	user@example.com
`event`	The event triggering the rule	scan-completed
`id`	The UUID of the rule	b2269a2c-69a1-4652-bcdf-899938886c17
`name`	The name of the Rule	Alert on scan
`updated_at`	When the rule was last updated	2021-04-01 17:09:40 -0500

scan

For events triggered by a scan task, the following additional fields are available in a `scan` object:

Field	Contents	Example
explorer_id	The UUID of the runZero Explorer which carried out the scan	0fd44a62-d827-41c0-b26c-4837222d8888
assets_changed	The number of assets changed as a result of the scan	9
assets_ignored	The number of assets ignored by the scan	2
assets_new	The number of new assets detected by the scan	2
assets_offline	The number of assets which were previously online but now offline	1
assets_online	The number of assets previously offline but now online again	2
assets_total	The total number of assets for the site scanned (including offline)	11
assets_unchanged	The number of assets unchanged by the scan	1
assets_updated	The total number of assets up-to-date as a result of the scan (changed + unchanged)	15
duration	Duration of the scan in seconds	26
end_time	When the scan ended	2021-04-02 12:50:26 -0500
excludes	Any IP addresses excluded from the scan	10.0.1.123
id	The UUID of the scan task	894a112c-3fb9-4301-8da7-8ce7fffb4443
name	The name of the scan task	Weekly security scan
rate	The scan rate	1000
recv_bytes	How many bytes were received during the scan	45176
recv_error	How many receive errors were detected	1
recv_packets	How many data packets were received during the scan	555
scheduled_time	When the scan was scheduled to run	2021-04-02 12:48:00 -0500
sent_bytes	How many bytes were sent during the scan	44740
sent_error	How many send errors were detected	0
sent_packets	How many data packets were sent during the scan	577
start_time	When the scan actually started	2021-04-02 12:49:55 -0500

`tags`	An array of tags associated with the scan task	
`targets`	The CIDR address ranges scanned	10.0.1.0/24
`type`	The type of operation	scan

explorer

For events triggered by a scan task, the following additional fields are available in an `explorer` object (runZero 2.1.8+):

Field	Contents	Example
`id`	The UUID of the runZero Explorer which carried out the scan	0fd44a62-d827-41c0-b26c-4837222d8888
`name`	The name of the Explorer which carried out the scan	MM34B-2
`internal_ip`	The internal IP address of the Explorer which carried out the scan	10.0.1.200

search

For events triggered by a search query rule, the following additional fields are available in a `search` object:

Field	Contents	Example
`url`	A link to perform the same search	https://console.runzero.com/inventory/?search=...
`found`	The number of matches found	3
`comparator`	The operation used to compare the number of matches	>=
`value`	The value the number of matches was compared against	1

report

For “scan completed” events, a `report` object contains the following results from the scan:

Field	Contents
`truncated`	Whether the set of objects was truncated due to large numbers of assets
`changed`	An array of changed assets (see below)
`new`	An array of new assets (see below)

`online`	An array of assets now online (see below)

The limit on the number of objects passed to the template is 25 for email notifications, 10 for Webhook notifications.

asset (from a scan report)

Each asset returned as part of a scan report has the following fields:

Field	Contents	
`addresses`	The IP address(es) of the asset	10.0.1.123
`alive`	Whether the asset responded to probes	true
`created_at`	The timestamp when the asset record was created	2021-04-02 12:50:26 -0500
`criticality_rank`	The criticality rank from (0-5)	1
`detected_by`	The method by which the asset was detected	arp
`domains`	Any domains the asset was found in	WORKGROUP
`first_seen`	The timestamp when the asset was first seen	2021-04-02 12:50:26 -0500
`hw`	A summary of the asset hardware	HP LaserJet Pro
`id`	The UUID of the asset	b38295fb-bef1-fa42-b82d-18ae99972228
`last_seen`	The timestamp when the asset was most recently seen	2021-04-02 12:50:26 -0500
`macs`	Any MAC addresses detected for the asset	00:56:55:00:91:04
`modified_risk_rank`	The risk rank after applying user overrides	2
`names`	Any names detected for the asset	LP538N
`os`	A summary of the asset's operating system	Linux
`outlier_score`	The normalized outlier score (0-5)	2
`risk_rank`	The normalized risk rank (0-4)	4
`service_count`	How many services the asset is running	4

`type`	The type of asset	Router
`updated_at`	The timestamp when the asset record was last updated	2021-04-02 12:50:26 -0500
`vulnerability_count`	How many vulnerabilies were identified on the asset	4

query

For query events, the following data is provided in the `query` variable:

Field	Contents
`count`	The number of rows matching the search query
`assets`	If it was an asset search, the array of assets matching the query
`services`	If it was a service search, the array of services matching the query
`wlans`	If it was a wireless network search, the array of wireless networks matching the query
`truncated`	Whether the set of assets was truncated due to the query returning a large number

The limit on the number of assets passed to the template from a query is 25 for email notifications, 10 for Webhook notifications.

asset (from a query)

For assets returned from a query rule, the following fields are available:

Field	Contents	
`addresses`	The IP address(es) of the asset	10.0.1.123
`alive`	Whether the asset responded to probes	true
`comments`	Any comments set in the asset record	
`created_at`	The timestamp when the asset record was created	2021-04-02 12:50:26 -0500
`detected_by`	The method by which the asset was detected	arp
`domains`	Any domains the asset was found in	WORKGROUP
`first_seen`	The timestamp when the asset was first seen	2021-04-02 12:50:26 -0500
`hw`	A summary of the asset hardware	ThinkPad X1
`id`	The UUID of the asset	b38295fb-bef1-fa42-b82d-

		18ae99972228
`last_seen`	The timestamp when the asset was most recently seen	2021-04-02 12:50:26 -0500
`macs`	The list of MAC addresses associated with the asset	[F4:F5:E8:89:92:31, 00:D0:2D:9F:47:77]
`names`	Any names detected for the asset	laptop.local
`organization`	The organization the asset belongs to	IT
`os`	A summary of the asset's operating system	Windows 10
`service_count`	How many services the asset is running	4
`site`	The site the asset was detected at	New York
`tags`	An array of tags set on the asset	
`type`	The type of asset	Thermostat
`updated_at`	The timestamp when the asset record was last updated	2021-04-02 12:50:26 -0500

service (from a query)

For services returned from a query rule, the following fields are available, some of which are taken from the associated asset:

Field	Contents	Example
`id`	The UUID of the service (not the asset)	b38efadb-61f1-f332-b92d-18ae99972228
`created_at`	When the service record was created	2021-04-02 12:50:26 -0500
`summary`	A summary of the service	ciscoSystems
`port`	The TCP/UDP port the service is on	53
`vhost`	The vhost of the service	ftp.example.com
`address`	The TCP/IP address of the service	192.168.33.44
`transport`	The transport	udp
`protocol`	The name of the protocol, if known	ssh
`organization`	The name of the organization the service's asset belongs to	HR
`site`	The name of the site the service's asset belongs to	Lab
`alive`	Whether the asset offering the service was alive	true

last_seen	When the asset was last seen	2021-04-02 12:50:26 -0500
first_seen	When the asset was first seen	2021-02-11 09:38:17 -0500
type	The asset type offering the service	Laptop
os	The OS offering the service	Linux
hw	The hardware offering the service	APC UPS
addresses	A list of other IP addresses associated with the asset offering the service	[192.168.0.2, 192.168.0.3]
macs	The list of MAC addresses associated with the asset	[F4:F5:E8:89:92:31, 00:D0:2D:9F:47:77]
names	Any names associated with the asset	[fw-3, fw-3a]
tags	Tags set on the asset	
domains	Any domain associated with the asset	LOCAL
service_count	A count of how many services the asset offers	4
comments	Any comments set on the asset	

wlan (wireless LAN, from a query)

For wireless lans (wlans) returned from a query rule, the following fields are available:

Field	Contents	Example
id	The UUID of the wireless LAN in runZero's database	f938934b-ae23-f112-b23d-18ae99972228
last_seen	When the wlan was last seen	2021-04-02 12:50:26 -0500
essid	The ESSID of the network	Free WiFi
bssid	The ESSID of the network	c4:41:1e:99:88:77
type	The type of wireless network	infrastructure
authentication	The authentication used to access the network	WPA2-PSK
encryption	The encryption used to protect data	AES
signal	The signal strength as a percentage	86
channels	The channel of the network	11
organization	The name of the organization the service's asset belongs to	HR

Example: Alert when scan completes

Let's take an example of something you might want to get alerted on: completed scans. We can create a rule that emails us when a scan completes, and provides us with some details, such as the number of new, online, offline, and modified assets. We'll build these details into our template.

Step 1. Create a template.

You can create a template from the Alerts page.

- Name the template something like `Email the team when a scan completes.`
- For the template type, choose `HTML`, since we want to use this template for emails.
- For the subject line, enter something that's descriptive, like `runZero scan {{scan.name}} completed at {{scan.end_time}}`. You can use the Mustache syntax for the subject.

Step 2. Create the body message

Now, let's create the email body. We want the email to tell us how many new, online, offline, and modified assets there are, as well as give us details on the new assets discovered.

The body looks like this:

```
<h1>{{site.name}}</h1>

<h2>Scan Results</h2>
{{#scan}}
<ul>
<li>{{assets_new}} new assets</li>
<li>{{assets_online}} online assets</li>
<li>{{assets_offline}} offline assets</li>
<li>{{assets_changed}} modified assets</li>
</ul>
{{/scan}}

<h2>New assets</h2>
<ul>
{{#report.new}}
<li>{{names}} {{addresses}} {{os}}</li>
{{/report.new}}
{{^report.new}}
<li>No new assets were discovered.</li>
{{/report.new}}
</ul>

<p><a href="{{task.url}}">View the scan results</a></p>
```

See that caret (^)? It represents inverted sections. These sections only render when the list is empty or the value is empty or false.

Step 3. Save the template and create a rule

The template is now available for you to choose when you create a rule.

Sample JSON templates

A JSON array of new assets found by a scan

```
[
  {{#report.new}}
    {
      "addresses": "{{addresses}}",
      "alive": "{{alive}}",
      "detected_by": "{{detected_by}}",
      "domains": "{{domains}}",
      "first_seen": "{{first_seen}}",
      "hw": "{{hw}}",
      "names": "{{names}}",
      "os": "{{os}}",
      "type": "{{type}}"
    },
  {{/report.new}}
  null
]
```

The trailing `null` in the array is so that the array is valid JSON, because JSON doesn't allow trailing commas.

All available attributes

Output all available variables and their attributes as a JSON object:

```
{{.}}
```

Alternatively, you can build your own JSON array with all attributes:

```
{
  "event": {
    "action": "{{event.action}}",
    "created_at": "{{event.created_at}}",
    "id": "{{event.id}}",
    "source_name": "{{sevent.ource_name}}",
    "source_type": "{{event.source_type}}",
    "success": "{{event.success}}",
    "target_name": "{{event.target_name}}",
    "target_type": "{{tevent.arget_type}}"
  },
  "organization": {
    "name": "{{organization.name}}",
```

```
},
  "site": {
    "name": "{{site.name}}",
    "id": "{{site.id}}"
  },
  "report": {
    "truncated": "{{report.truncated}}",
    "changed": "{{report.changed}}",
    "new": "{{report.new}}",
    "offline": "{{report.offline}}",
    "online": "{{report.online}}"
  },
  "rule": {
    "action": "{{rule.action}}",
    "created_at": "{{rule.created_at}}",
    "created_by": "{{rule.created_by}}",
    "event": "{{rule.event}}",
    "id": "{{rule.id}}",
    "name": "{{rule.name}}",
    "updated_at": "{{rule.updated_at}}"
  },
  "scan": {
    "explorer_id": "{{scan.explorer_id}}",
    "assets_changed": "{{scan.assets_changed}}",
    "assets_ignored": "{{scan.assets_ignored}}",
    "assets_new": "{{scan.assets_new}}",
    "assets_offline": "{{assets_offline}}",
    "assets_online": "{{scan.assets_online}}",
    "assets_total": "{{scan.assets_total}}",
    "assets_unchanged": "{{scan.assets_unchanged}}",
    "assets_updated": "{{scan.assets_updated}}",
    "duration": "{{scan.duration}}",
    "end_time": "{{scan.end_time}}",
    "excludes": "{{scan.excludes}}",
    "id": "{{scan.id}}",
    "name": "{{scan.name}}",
    "rate": "{{scan.rate}}",
    "recv_bytes": "{{scan.recv_bytes}}",
    "recv_error": "{{scan.recv_error}}",
    "recv_packets": "{{scan.recv_packets}}",
    "scheduled_time": "{{scan.scheduled_time}}",
    "sent_bytes": "{{scan.sent_bytes}}",
    "sent_error": "{{scan.sent_error}}",
    "sent_packets": "{{scan.sent_packets}}",
    "start_time": "{{scan.start_time}}",
    "tags": "{{scan.tags}}",
    "targets": "{{scan.targets}}",
    "type": "{{scan.type}}"
  },
  "explorer": {
    "id": "{{explorer.id}}",
    "name": "{{explorer.name}}",
    "internal_ip": "{{explorer.internal_ip}}"
```

```
"search": {
    "url": "{{search.url}}",
    "found": "{{search.found}}",
    "comparator": "{{search.comparator}}",
    "value": "{{search.value}}"
  },
  "query": {
    "count": "{{query.count}}",
    "assets": "{{query.assets}}",
    "services": "{{query.services}}",
    "wlans": "{{query.wlans}}",
    "truncated": "{{query.truncated}}"
  }
}
```

Event attributes only

```
{
  "event": {{event}}
}
```

or

```
{
  "action": "{{event.action}}",
  "created_at": "{{event.created_at}}",
  "id": "{{event.id}}",
  "source_name": "{{sevent.ource_name}}",
  "source_type": "{{event.source_type}}",
  "success": "{{event.success}}",
  "target_name": "{{event.target_name}}",
  "target_type": "{{tevent.arget_type}}"
}
```

Organization attributes only

```
{
  "organization": {{organization}}
}
```

or

```
{
  "name": "{{organization.action}}",
  "id": "{{organization.id}}"
}
```

Site attributes only

```
{
  "site": {{site}}
}
```

or

```
{
  "name": "{{site.action}}",
  "id": "{{site.id}}"
}
```

Report attributes only

```
{
  "report": {{report}}
}
```

or

```
{
  "truncated": "{{report.truncated}}",
  "changed": "{{report.changed}}",
  "new": "{{report.new}}",
  "offline": "{{report.offline}}",
  "online": "{{report.online}}"
}
```

Rule attributes only

```
{
  "rule": {{rule}}
}
```

or

```
{
  "action": "{{rule.action}}",
  "created_at": "{{rule.created_at}}",
  "created_by": "{{rule.created_by}}",
  "event": "{{rule.event}}",
  "id": "{{rule.id}}",
  "name": "{{rule.name}}",
  "updated_at": "{{rule.updated_at}}"
}
```

Scan attributes only

```
{
  "scan": {{scan}}
}
```

or

```
{
  "explorer_id": "{{scan.explorer_id}}",
```

```
"assets_ignored": "{{scan.assets_ignored}}",
  "assets_new": "{{scan.assets_new}}",
  "assets_offline": "{{assets_offline}}",
  "assets_online": "{{scan.assets_online}}",
  "assets_total": "{{scan.assets_total}}",
  "assets_unchanged": "{{scan.assets_unchanged}}",
  "assets_updated": "{{scan.assets_updated}}",
  "duration": "{{scan.duration}}",
  "end_time": "{{scan.end_time}}",
  "excludes": "{{scan.excludes}}",
  "id": "{{scan.id}}",
  "name": "{{scan.name}}",
  "rate": "{{scan.rate}}",
  "recv_bytes": "{{scan.recv_bytes}}",
  "recv_error": "{{scan.recv_error}}",
  "recv_packets": "{{scan.recv_packets}}",
  "scheduled_time": "{{scan.scheduled_time}}",
  "sent_bytes": "{{scan.sent_bytes}}",
  "sent_error": "{{scan.sent_error}}",
  "sent_packets": "{{scan.sent_packets}}",
  "start_time": "{{scan.start_time}}",
  "tags": "{{scan.tags}}",
  "targets": "{{scan.targets}}",
  "type": "{{scan.type}}"
}
```

Explorer attributes only

```
{
  "explorer": {{explorer}}
}
```

or

```
{
  "id": "{{explorer.id}}",
  "name": "{{explorer.name}}",
  "internal_ip": "{{explorer.internal_ip}}"
}
```

Search attributes only

```
{
  "search": {{search}}
}
```

or

```
{
  "url": "{{search.url}}",
  "found": "{{search.found}}",
  "comparator": "{{search.comparator}}",
```

```
"value": "{{search.value}}"
}
```

Query attributes only

```
{
  "query": {{query}}
}
```

or

```
{
  "count": "{{query.count}}",
  "assets": "{{query.assets}}",
  "services": "{{query.services}}",
  "wlans": "{{query.wlans}}",
  "truncated": "{{query.truncated}}"
}
```

Asset, service, or wireless query alert to SIEM or SOAR

```
{
  "organization": {
    "name": "{{organization.name}}",
    "id": "{{organization.id}}"
  },
  "site": {
    "name": "{{site.name}}",
    "id": "{{site.id}}"
  },
  "rule": {
    "action": "{{rule.action}}",
    "created_at": "{{rule.created_at}}",
    "created_by": "{{rule.created_by}}",
    "event": "{{rule.event}}",
    "id": "{{rule.id}}",
    "name": "{{rule.name}}",
    "updated_at": "{{rule.updated_at}}"
  },
  "search": {
    "url": "{{search.url}}",
    "found": "{{search.found}}",
    "comparator": "{{search.comparator}}",
    "value": "{{search.value}}"
  }
}
```

Asset, service, or wireless query alert to Slack

```
{
  "blocks": [
```

```
"type": "section",
      "text": {
        "type": "mrkdwn",
        "text": ":red_circle: *runZero Alert* - {{rule.name}}"
      }
    },
    {
      "type": "divider"
    },
    {
      "type": "section",
      "text": {
        "type": "mrkdwn",
        "text": "*Rule information*\n\n_Name_: {{rule.name}}\n_Type_: {{rule.event}}\n_Link_: https://
console.runzero.com/alerts/rule/{{rule.id}}"
      }
    },
    {
      "type": "section",
      "text": {
        "type": "mrkdwn",
        "text": "*Match information*\n\n_Organization_: {{organization.name}}\n_Site_: {{site.name}}\n
_Match count_: {{search.found}}\n_Search_: {{search.value}}\n_Link_: https://console.runzero.com/alert
s/rule/{{search.url}}"
      }
    }
  ]
}
```

Asset, service, or wireless query alert to Microsoft Teams

```
{
  "type": "message",
  "attachments": [
    {
      "contentType": "application/vnd.microsoft.card.adaptive",
      "contentUrl": null,
      "content": {
        "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
        "type": "AdaptiveCard",
        "version": "1.5",
        "body": [
          {
            "type": "TextBlock",
            "text": "runZero Alert - {{rule.name}}"
          },
          {
            "type": "TextBlock",
            "text": "**Rule information**\n\n_Name_: {{rule.name}}\n\n_Type_: {{rule.event}}\n\n_Link
_: [Rule](https://console.runzero.com/alerts/rule/{{rule.id}})",
            "wrap": true
          },
```

```
"type": "TextBlock",
            "text": "**Match information**\n\n_Organization_: {{organization.name}}\n\n_Site_: {{site.
name}}\n\n_Match count_: {{search.found}}\n\n_Search_: {{search.value}}\n\n_Link_: [Search](https://co
nsole.runzero.com/alerts/rule/{{search.url}})",
            "wrap": true,
            "spacing": "Medium"
          }
        ]
      }
    }
  ]
}
```

Data type accepted by each channel

When you create a rule, the channel you select must accept the data type used by the template. For example, if you want to send a Slack notification, the template must be in plain text or JSON.

Here are the data types accepted by each channel:

Channel	Data type
Email	Plain text, HTML, JSON
Webhook	Plain text, JSON
Internal notification	Plain text

Managing templates

Go to Alerts > Templates. From this page, you can create, edit, and delete templates. Before deleting templates, make sure they are not in use by any rules. If you delete a template that is in use by a rule, the rule will revert to the default template.

Querying your data

runZero provides many ways to query your data. Generally, queries can be broken into two concepts:

- Filters or parameters used in the search bars on pages across the console, or
- System and custom queries for which match metrics are calculated as tasks complete.

Both allow you to leverage the extensive query language to quickly find the information you're looking for.

Filtering and searching data

The various inventory pages are likely the main place you'd look to use these queries, but many other pages include the same type of search bar that can be used to filter results. The following documentation pages will help you craft a query that meets your needs:

- Syntax for crafting compound queries
- Examples to guide query development
- Keywords that can be used across the inventory pages
- Keywords that can be used across other components of the console

System and custom queries

The Queries page serves as an inventory of all your saved queries. Saved queries can either be system queries published by runZero or custom queries created by you or your team.

Standard query attributes

Queries have the following standard attributes:

- **Name**: A name for the saved query.
- **Description**: A description of the saved query.
- **Search query**: The query or parameters to search for matches to.
- **Search live assets only**: When toggled to *Yes* (default), `alive:t` is added to the search query to only include assets marked alive.
- **Type**: The inventory the query will search.
- **Category**: The category the query falls under.
- **Severity**: A severity level for the query.
- **Automatically track query results on the dashboard**: When toggled to *Yes*, the query and count of matches will be included in a component on the dashboard.

Vulnerability record attributes

Community Platform

System and custom queries can also be used to create and associate vulnerability records with matching assets. This will be enabled by default on some system queries. To have a query create vulnerability records, switch the **Apply a vulnerability record to matching assets** toggle to *Yes*, then complete the following fields:

- **Vulnerability ID**: Choose a unique ID to track this vulnerability within runZero.
- **CVEs**: Include a list of CVEs relevant to this vulnerability record (optional).
- **Solution**: Provide context for how this vulnerability could be remediated on assets (optional).
- **Risk**: Select a risk level to associate with the vulnerability. This impacts the asset risk.
- **Exploitable**: Specify whether an exploit is available for the vulnerability.
- **CVSS v3 base score**: The CVSS v3 base score (0.00 to 10.00).
- **CVSS v3 temporal score**: The CVSS v3 temporal score (0.00 to 10.00).
- **CVSS v2 base score**: The CVSS v2 base score (0.00 to 10.00).
- **CVSS v2 temporal score**: The CVSS v2 temporal score (0.00 to 10.00).
- **CPE 2.3 identifier**: Specify a relevant Common Platform Enumeration identifier in URI format (v2.3) to associate with the reported vulnerability (optional).

Creating and editing queries

Custom queries can be created by users with the default role of administrator or higher from the Queries page by clicking the **New query** button. Once created, custom queries can be edited and copied. System queries cannot be edited directly, but can be copied if you wish to make changes.

Search query syntax

runZero supports a deep searching across the Asset, Service, and Wireless Inventory, across organizations and sites, and through the Query Library. The runZero Export API uses the same inventory search syntax to filter results.

Query syntax

Boolean operators

Search queries can be combined through `AND` and `OR` operators and be grouped using parenthesis.

AND

For example, a Asset Inventory query of `os:"Windows 10" AND protocols:http AND protocols:smb2` will show only those assets where Windows 10 was identified and both SMB and a web server were discovered. Search values that contain spaces must be placed in double quotes.

OR

By contrast, the example query of `os:"Windows 10" AND protocols:http OR protocols:smb2` will search for Windows 10 running a web server `or` any assets with the SMB service exposed. In addition to `AND` and `OR`, the `NOT` operator can be used to filter a query. For example, the query `os:"Windows 10" AND NOT protocols:http` will show Windows 10 systems without a web server. If the negation should happen as the first term the `AND` should be dropped. The query `NOT protocol:http AND os:"Windows 10"` is equivalent to the previous search, with the terms reversed.

Wildcard and fuzzy searches

Most keywords are a fuzzy match by default. To force an exact match, prefix match, or suffix match, the `=` prefix can be applied to the search term, with the `%` character used as a wildcard. To search an operating system name of just `Windows`, the Asset Inventory query would be `os:="Windows"`, while to specify a prefix match of `Ubuntu Linux`, the query `os:="Ubuntu Linux%"` can be used.

Single-character wildcard

The `%` wildcard matches any number of characters. To match exactly one character, use the `_` single-character wildcard. For example, `os:="Window%"` will match both `Windows` and `Windows 10`, while `os:="Window_"` will match `Windows` but not `Windows 10`.

Time and date values

Time and date (timestamp) fields can be searched using `<` (less than) and `>` (greater than) operators to compare against the current time. You can also use `-` to compare to a relative time in the past, for example `<-3years` would look for timestamps that occurred before three years ago. Supported units:

- hours
- minutes
- seconds
- months
- years

A special value of `now` can also be used.

For example, an asset search of `first_seen:<1year` would search for assets first detected this year. Other examples:

```
first_seen:<3days

first_seen:>2019-08-01

first_seen:>8/1/2019

last_seen:<1week

last_seen:<2months
```

```
last_seen:<1year

created_at:>2weeks

created_at:<30minutes

updated_at:>1year

updated_at:<12hours

os_eol:<now

os_eol:>4weeks

os_eol:<-2years

os_eol_extended:>now

os_eol_extended:>90days
```

Empty values

To search for an empty value, the `=` prefix can be used with no value after. For example, the query `os:=` will find assets with no identified operating system.

Note that this only works for single-valued attributes such as `os` and `type`; it won't work for multi-value attributes such as `names` or `addresses`.

Asset and service inventory searches

Asset and Service attributes support two special search types in addition to the documented keywords:

- Asset Inventory searches treat unknown keywords as filters against individual Asset **attributes**.
- Service Inventory searches treat unknown keywords as filters against individual Service **data** values.

In situations where an Asset keyword conflicts with a Service data key, or an Asset attribute conflicts with a Service keyword, the prefixes `_asset.` and `_service.` can be used to disambiguate.

Searches are handled slightly differently. Service queries can filter against Asset attributes (`os:linux`) and Service attributes (`banner:Password`), but the Asset queries are limited to summary information about services (`protocol:ssh`).

Query examples

There are endless ways to combine terms and operators into effective queries, and the examples below can be used as-is or adjusted to meet your needs.

Network configurations and access

- Multihomed assets with public and private IP addresses:

```
alive:t AND has_public:t AND has_private:t
```

- Multihomed assets connected only to private networks

```
multi_home:t AND has_public:f
```

- Default SSH configuration using passwords for authentication:

```
alive:t AND protocol:"ssh" AND ssh.authMethods:"=password"
```

- Microsoft FTP servers:

```
alive:t AND protocol:"ftp" AND banner:"=%Microsoft FTP%"
```

- Remote access services/protocols:

```
protocol:rdp OR protocol:vnc OR protocol:teamviewer
```

- Assets with public IPs running remote access services:

```
has_public:t OR has_public:t AND alive:t AND (protocol:rdp OR protocol:vnc OR protocol:teamviewer)
```

- Open ports associated with cleartext protocols:

```
port:21 OR port:23 OR port:80 OR port:443 OR port:139 OR port:445 OR port:3306 OR port:1433 OR port:161 OR
port:8080 OR port:3389 OR port:5900
```

- Telnet on nondefault ports:

```
protocol:telnet AND NOT port:23
```

- Windows assets offering SMB services:

```
os:windows AND protocol:smb1 OR protocol:smb2
```

- Switch assets accepting Username and Password authentication:

```
type:switch AND (_asset.protocol:http AND NOT _asset.protocol:tls) AND ( html.inputs:"password:" OR
last.html.inputs:"password:" OR has:http.head.wwwAuthenticate OR has:last.http.head.wwwAuthenticate )
```

- Assets more than 8 hops away:

```
attribute:"ip.ttl.hops" AND ip.ttl.hops:>"8
```

Asset lifecycle and hardware

- Assets created as a result of arbitrary responses:

```
has_mac:f AND has_name:f AND os:= AND hardware:= AND detected_by:icmp AND service_count:<2
```

- End of Life assets:

```
os_eol:<now
```

- Assets where both OS support and extended support are expired:

```
os_eol:<now AND os_eol_extended:<now
```

- Assets where OS support is EOL but still covered by extended support:

```
os_eol:<now AND os_eol_extended:>now
```

- EOL Linux operating systems:

```
os:linux AND os_eol:<now
```

- EOL Windows operating systems:

```
os:windows AND os_eol:<now
```

- Assets discovered within the past two weeks:

```
first_seen:"<2weeks"
```

- All available serial number sources

```
protocol:snmp has:snmp.serialNumbers OR hw.serialNumber:t OR ilo.serialNumber:t
```

- Asset serial numbers from SNMP:

```
protocol:snmp has:snmp.serialNumbers
```

- Older Windows OSes:

```
os:"Windows Server 2012" OR os:"Windows 7"
```

- Older Linux OSes:

```
OS:linux AND os_eol:<now
```

- BACnet devices:

```
type:bacnet
```

- Hikvision DVRs:

```
type:dvr AND os:hikvision
```

- IoT Devices:

```
type:"IP Camera" OR type:"thermostat" OR type:"Amazon Device" OR hw:"Google Chromecast" OR
type:"Game Console" OR type:"Robotic Cleaner" OR type:"Nest Device" OR type:"Network Audio" OR
type:"Smart TV" OR type:"VR Headset" OR type:"Voice Assistant""
```

- Video-related assets:

```
type:"IP Camera" OR type:"DVR" OR type:"Video Encoder"
```

Misconfigurations

- SMBv1:

```
protocol:"smb1"
```

- Remote access with common services:

```
protocol:rdp OR protocol:vnc OR protocol:teamviewer OR protocol:spice OR protocol:pca
```

- Switches with default configurations for web access:

```
type:switch AND (_asset.protocol:http AND NOT _asset.protocol:tls) AND ( html.inputs:"password:" OR
last.html.inputs:"password:" OR has:http.head.wwwAuthenticate OR has:last.http.head.wwwAuthenticate )
```

- Default SSH configurations using passwords for authentication:

```
alive:t AND protocol:"ssh" AND ssh.authMethods:"=password"
```

- Switches using Telnet or HTTP for remote access:

```
type:switch AND protocol:telnet OR protocol:http
```

- Microsoft FTP servers:

```
alive:t AND protocol:"ftp" AND banner:"=%Microsoft FTP%"
```

- Virtual machines that are not syncing time with the host:

```
@vmware.vm.config.tools.syncTimeWithHost:"False"
```

Weak configurations

- Telnet (vs. SSH):

```
protocol:telnet
```

- FTP on ports 10-21 (vs. FTPS on port 990):

```
protocol:ftp
```

- FTP on ports 20-21 (vs. SCP on port 22):

```
protocol:ftp
```

- HTTP on port 80 (vs. HTTPS on port 443):

```
protocol:http
```

- SSH versions < 2.0:

```
protocol:ssh AND NOT banner:"SSH-2.0"
```

- TLS:

```
tls.versionName:"=TLSv1.3" OR tls.versionName:"=TLSv1.2" OR tls.versionName:"=TLSv1.1" OR
tls.versionName:"=TLSv1.0"
```

- LDAP on port 389 (vs. LDAPS on port 636):

```
protocol:ldap OR port:389
```

- Wireless access points without WPA authentication:

```
not authentication:WPA
```

- Online assets with SSH accepting password authentication:

```
alive:t AND has:"ssh.authMethods" AND protocol:"ssh" AND (ssh.authMethods:"=password" OR
ssh.authMethods:"=password%publickey")
```

- Detect OpenSSL version 3.0 - 3.0.6:

```
product:openssl AND version:3.0
```

EDR / MDM

- CrowdStrike coverage gaps:

```
not edr.name:crowdstrike AND (type:server OR type:desktop OR type:laptop)
```

- Assets with CrowdStrike Agent status “Not Provisioned”:

```
@crowdstrike.dev.provisionStatus:"NotProvisioned"
```

- Assets with CrowdStrike Agent mode “Reduced Functionality”:

```
@crowdstrike.dev.reducedFunctionalityMode:"yes"
```

- Assets with CrowdStrike Agent status “Normal”:

```
@crowdstrike.dev.status:"normal"
```

- SentinelOne coverage gaps:

```
not edr.name:Sentinelone AND (type:server OR type:desktop OR type:laptop)
```

- Assets with SentinelOne Agent requiring patch:

```
(alive:t OR scanned:f) AND has:"@sentinelone.dev.appsVulnerabilityStatus" AND
@sentinelone.dev.appsVulnerabilityStatus:"=patch_required"
```

- Assets missing either CrowdStrike or SentinelOne EDR agents:

```
NOT edr.name:crowdstrike AND (type:server OR type:desktop OR type:laptop) OR NOT edr.name:sentinelone AND
(type:server OR type:desktop OR type:laptop)
```

- Miradore coverage gaps:

```
not source:Miradore AND (os:google android OR os:apple ios) AND type:mobile
```

- Microsoft Defender coverage gaps:

```
not edr.name:"Defender" AND os:Windows
```

- Assets not managed by a Microsoft product:

```
source:runzero AND NOT (source:ms365defender OR source:intune OR source:azuread)
```

- Find mobile devices on the network:

```
(os:google ANDroid OR os:apple ios) AND type:mobile
```

- Known FCC security threats, like Kaspersky:

```
alive:t AND edr.name:Kaspersky
```

Virtual machine configurations

- Virtual machines with less than 8 GB of memory:

```
@vmware.vm.config.hardware.memoryMB:<"8192"
```

- VMs with less than 16GB of memory:

```
@vmware.vm.runtime.maxMemoryUsage:"16384"
```

- Virtual machines that are not syncing time with the host:

```
@vmware.vm.config.tools.syncTimeWithHost:"False"
```

- Virtual machines that are configured with floppy drives:

```
@vmware.vm.config.extra.floppy0.autodetect:"true"
```

- Virtual machines running VMware tools:

```
@vmware.vm.config.extra.guestinfo.vmtools.versionString:"_"
```

- Virtual machines running Windows:

```
source:VMware AND os:Windows
```

- Virtual machines running Linux:

```
source:VMware AND os:Linux
```

Vulnerability concerns

- Rapid7 - fails PCI compliance:

```
test.pciComplianceStatus:"fail"
```

- Tenable - High and Critical severity vulnerabilities that are on CISA's Known Exploited list:

```
plugin.xrefs.type:"CISA-KNOWN-EXPLOITED" AND (severity:high OR severity:critical)
```

- Tenable - Critical severity vulnerabilities where exploits are available:

```
plugin.exploitabilityEase:"Exploits are available" AND severity:critical
```

- Tenable - High and Critical severity vulnerabilities where exploits are not required

```
plugin.exploitabilityEase:"No exploit is required" AND (severity:critical OR severity:high)
```

Wireless results

- Search ESSID for authentication exceptions:

```
essid:"<ESSID>" AND NOT authentication:"wpa2-enterprise"
```

- Find unknown BSSIDs broadcasting known ESSID (exclude known BSSIDs in query for gap analysis)

```
essid:="<ESSID>" AND NOT bssid:"<MAC address>"
```

Inventory keywords

The data across your runZero inventories can be queried and filtered using the search syntax in conjunction with the available inventory keywords. Keywords and example values are documented for the following inventories:

- Assets
- Services
- Software
- Vulnerabilities

- Certificates
- Wireless networks
- Users
- Groups

Asset inventory

When viewing assets, you can use the following keywords to search and filter.

User-specified fields

Comments

Use the syntax `comment:<text>` to search comments on an asset.

```
comment:"contractor laptop"

comment:"imaging server"
```

Tags

Use the syntax `tag:<term>` to search tags added to an asset. The term can be the tag name, or the tag name followed by an equal sign and the tag value. Tag value matches must be exact.

```
tag:"group"

tag:"group=production"
```

Organization name or ID

Use the syntax `organization:<term>` to filter by organization name or ID.

```
organization:runZero

organization:"Temporary Project"

organization:f1c3ef6d-cb41-4d55-8887-6ed3cfb3d42d
```

Site name or ID

Use the syntax `site:<term>` to filter by site name or ID.

```
site:Primary

site:"Branch Office"

site:ad67d649-041b-439d-af59-f200053a8899
```

Explorer name or ID

Use the syntax `explorer:<term>` to filter by Explorer name or ID.

```
explorer:DESKTOP-AB451F

explorer:8b927a8e-d405-40e9-aa47-d6afc9bff237
```

Hosted zone

Use the syntax `hosted_zone:<zone name>` to filter by the hosted runZero Explorer that found the asset. Using this filter after a hosted scan can be a good way to locate externally facing assets.

Owner

Use the syntax `owner:<term>` to filter by owner name.

```
owner:user@runzero.com
owner:"Security Team"
```

Ownership status

Use the syntax `owner_count:<number>` to filter by owner count. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
owner_count:>0
owner_count:0
```

Use the syntax `has_owner:<boolean>` to find assets with owners or assets that are missing owners.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_owner:t
has_owner:f
```

Use the syntax `ownership_type:<term>` to filter assets by ownership type name. This will return assets that have an owner assigned for the specified ownership type.

```
ownership_type:"Asset Owner"
ownership_type:"Security Owner"
```

Asset fields

Asset ID

The ID field is the unique identifier for a given asset, written as a UUID. Use the syntax `id:<uuid>` to filter by ID field.

```
id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Operating system

The operating system field is a string describing the detected operating system software. This field is searched using the syntax `os:<text>`. The OS version, if available, can be searched using `os_version:<number>`.

```
os:"Windows"

os:"Ubuntu Linux"

os_version:8
```

OS CPE

The operating system Common Platform Enumeration (CPE) field is a string describing the detected operating system software aligned to the CPE naming scheme. This field is searched using the syntax `os.cpe23:<text>`. In cases where runZero was able to fingerprint the operating system but the NIST database does not contain an official matching entry, an unofficial CPE will be generated and include `r0_unofficial` in the `other` field of the CPE.

```
os.cpe23:"ubuntu"

os.cpe23:="cpe:/o:canonical:ubuntu_linux:22.04.1"

os.cpe23:="cpe:/o:alma:linux:-::~~~~~r0_unofficial"
```

Type

The type field is a string describing the detected system type, such as Desktop, Laptop, Server, BMC, or Mobile. Use the syntax `type:<text>` to search this field.

```
type:Desktop

type:BMC

type:"Game Console"
```

Hardware

The hardware field is a string describing the detected physical hardware, such as `macMini` or `Nintendo Switch`. Use the syntax `hardware:<text>` to search this field.

```
hardware:Switch

hardware:macMini
```

Hostnames

The hostnames associated with an asset are obtained from DNS and exposed services. Use the syntax `name:<text>` to search these names.

```
name:"www"
```

```
name:"TV"
```

To search an asset where any asset has a specific prefix or suffix, use the `:=` exact match operator, and use `%` as a wildcard:

```
name:="FTP.%"
```

```
name:="%-09"
```

Use the syntax `name_count:<number>`to search the hostname count. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
name_count:>1
```

Use the syntax `name_overlap:<boolean>` to find assets sharing the same name. The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
name_overlap:t
```

Domains

The domains associated with an asset are obtained from DNS and exposed services. Use the syntax `domain:<domainname>` to search the domain names.

```
domain:"amazon.com"
```

```
domain:"corp.lan"
```

```
domain:"WORKGROUP"
```

The domain count can be searched using the syntax `domain_count:<number>`. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
domain_count:>1
```

Addresses

Use the syntax `address:<ip>` to search the addresses (both primary and secondary) associated with an asset, `primary_address:<ip>` to search only the primary addresses associated with an asset, or `secondary_address:<ip>` to search only the secondary addresses associated with an asset. These keywords also allow for CIDR mask matching, as well as wildcard matches using '%'. A comma-separated list of addresses will be used as an efficient multiple-match.

```
address:192.168.0.1
```

```
address:10.0.0
```

```
address:10.1.2.0/24
```

```
address:%.0.1
```

```
address:10.%.254
```

```
address:10.0.0.1,10.0.0.2,10.0.0.3
```

Use the syntax `address_count:<term>` and `address_extra_count:<number>` to search address primary and secondary counts. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
address_extra_count:0
```

Use the syntax `address_overlap:<boolean>` to find assets sharing primary IP addresses. This can be further filtered to single sites using the `site` keyword. The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
address_overlap:t
```

Use the syntax `address_extra_overlap:<boolean>` to find assets sharing secondary IP addresses. This can be further filtered to single sites using the `site` keyword. The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
address_extra_overlap:t
```

Networks

Use the syntax `net:<cidr>` to search the addresses (both primary and secondary) associated with an asset by CIDR mask.

```
net:192.168.0.0/24
```

Default SNMP communities

Use the syntax `has:snmp.v2DefaultCommunities` to search for assets with a default SNMP community (public, private, and other defaults).

```
has:snmp.v2DefaultCommunities
snmp.v2DefaultCommunities:public
```

Public address

Use the keyword `has_public` and syntax `has_public:<boolean>` to locate any asset with a non-reserved IP address. This often corresponds to public-facing systems, though public IPs can also be used internally behind a firewall. Note that public IPv6 addresses are included by this filter; to search for only public IPv4 addresses, you can use `has_public_v4`.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_public:true
```

Private address

Use the keyword `has_private` and syntax `has_private:<boolean>` to locate any asset with a private IP address.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_private:false
```

IPv6 address

Use the keyword `has_ipv6` and the syntax `has_ipv6:<boolean>` to locate any asset with an identified IPv6 address.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_ipv6:false
```

Link-local IPv6 address

Use the keyword `has_link_local` and syntax `has_link_local:<boolean>` to locate any asset with an identified IPv6 link local (`fe80::`) address.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_link_local:true
```

MAC address

Use the syntax `mac:<term>` to search MAC addresses associated with an asset.

```
mac:00:5c:04
```

```
mac:00:00:1c
```

Use the syntax `mac_count:<number>` to search the MAC address count. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
mac_count:>2
```

If you use exact search (`:=`) you can also search for full MAC addresses in Cisco format or dash-separated format:

```
mac:=00-10-fa-c2-bf-d5

mac:=0010.fac2.bfd5
```

Use the syntax `mac_overlap:<boolean>` to find assets sharing the same MAC address. The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
mac_overlap:t
```

MAC address vendors

The vendor associated with the MAC addresses of an asset can be searched using the syntax `mac_vendor:<text>`.

```
mac_vendor:Apple

mac_vendor:"Intel Corporate"
```

To search only the vendor associated with the newest MAC address, use the syntax `newest_mac_vendor:<text>`

```
newest_mac_vendor:Apple
```

The MAC address vendor count can be searched using the syntax `mac_vendor_count:<number>`. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
mac_vendor_count:0
```

MAC address age

Use the syntax `mac_age:<term>` to search the allocation date of the newest MAC address associated with an asset. The term supports the standard runZero [time comparison syntax] [time].

```
mac_age:>1year

mac_age:<6months

mac_age:2019-12-31
```

Outlier score

Use the syntax `outlier_score:<value>` to search the calculated outlier score of assets. The outlier score is in the range 0 to 5 inclusive. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
outlier_score:>2

outlier_score:0
```

Upstream switch IP address

Use the syntax `switch.ip:<address>` to search the IP address of the upstream switch assets are connected to.

```
switch.ip:192.168.1.1
```

```
switch.ip:fe80::81f2:1c9d:5ac9:5420
```

Upstream switch name

Use the syntax `switch.name:<hostname>` to search the hostname of the upstream switch assets are connected to.

```
switch.name:"SWITCH-1"
```

```
switch.name:office
```

Upstream switch port

Use the syntax `switch.port:<address>-<port number string>` to search the port on the upstream switch assets that are connected to.

```
switch.port:192.168.1.1-25
```

```
switch.port:10.1.2.3-0/1/2
```

Upstream switch shared port

Use the syntax `attribute:switch.portShared` to find assets which connect to a switch port that reports multiple MAC addresses.

```
attribute:switch.portShared
```

Attributes

Use the syntax `attribute:<term>` to search the asset attribute fields, such as the port used to detect the TTL.

```
attribute:"ip.ttl.port"
```

```
attribute:"cpe:/a:isc:bind:9.11.3"
```

```
attribute:"9.11.3"
```

To determine if an asset has any attribute defined, use the `has:<attribute-name>` keyword. The `has` keyword can be inverted to find missing fields with `not has:<term>`.

```
has:"ip.ttl.port"
```

```
not has:"rdns.names"
```

In addition to the standard fields, the following special attributes are available:

- `has:screenshot` returns assets where at least one screenshot was obtained.
- `has:icons` returns assets where at least one icon was obtained (HTTP, UPnP, or similar).
- `has:uplink` returns assets seen in the CAM table of a network switch.
- `has:downlink` returns assets where the CAM table was queried at least one other asset was connected.
- `has:unmapped` returns assets where the CAM table was queried at least one other asset was connected but not identified by IP.

The attribute can be specified as a term directly. If the attribute name conflicts with an existing term, the prefix `_asset.` can be specified to disambiguate the query.

```
ip.ttl.port:80

rdns.names:"router"

_asset.ip.ttl.hops:"1"
```

Foreign attributes from third-party inbound integrations can be queried using the syntax `@<integration>.<source>.<attribute>:<term>`. The table below includes the correct prefix for each integration.

Integration	Prefix
Miradore	`@miradore.dev.`
AWS EC2	`@aws.ec2.`
AWS ELB & ELBv2	`@aws.elb.`
AWS RDS	`@aws.rds.`
CrowdStrike	`@crowdstrike.dev.`
Azure Load Balancer	`@azure.vm.`
Azure VM	`@azure.vm.`
Azure Scale Set VM	`@azure.vmss.`
Censys	`@censys.host.`
VMWare	`@vmware.vm.`
GCP Load Balancer	`@gcp.lb.`
GCP E2-Micro VM	`@gcp.vm.`
GCP CloudSQL	`@gcp.cloudsql.`
SentinelOne	`@sentinelone.dev.`
Tenable.io & Nessus	`@tenable.dev.`
Rapid7 Nexpose & InsightVM	`@rapid7.dev.`

Qualys VMDR	`@qualys.dev.`
Shodan	`@shodan.dev.`
Azure AD	`azuread`
Active Directory (LDAP)	`@ldap.computer.`
Microsoft 365 Defender	`@ms365defender.dev.`
Microsoft Intune	`@intune.dev.`
Google Workspace ChromeOS	`@googleworkspace.chromeos.`
Google Workspace Endpoint	`@googleworkspace.endpoint.`
Google Workspace Mobile	`@googleworkspace.mobile.`

```
@aws.ec2.region:="us-east-2"
```

```
@crowdstrike.dev.agentVersion:="6.49.16201.0"
```

```
@googleworkspace.chromeos.model:="HP Chromebook"
```

Asset services

Service ports

The TCP and UDP services associated with an asset can be searched by port number using the syntax `port:<number>`.

```
port:80
```

```
port:161
```

Service TCP ports

Use the syntax `tcp:<number>` to search the TCP services associated with an asset by port number.

```
tcp:443
```

To search for assets with a specific list of TCP ports open, you can use the syntax `service_ports_tcp:=<list>`. Values should be in ascending numerical order, and separated by commas.

```
service_ports_tcp:=80,443
```

Service UDP ports

Use the syntax `udp:<number>` to search UDP services associated with an asset by port number.

```
udp:53
```

To search for assets with a specific list of UDP ports open, you can use the syntax `service_ports_udp:=<list>`. Values should be in ascending numerical order, and separated by commas.

```
service_ports_udp:=53,123
```

Service protocols

Use the syntax `service_protocols:<term>` (or `protocol:<term>` for short) to search the identified service protocols associated with an asset.

```
protocol:http
```

```
service_protocol:telnet
```

The protocol count can be searched using the syntax `protocol_count:<number>`. This search supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
protocol_count:>1
```

Service products

Use the syntax `service_products:<term>` (or `product:<term>` for short) to search for the identified service products associated with an asset.

```
product:openssh
```

```
service_products:nginx
```

The product count can be searched using the syntax `product_count:<number>`. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
product_count:>3
```

Service counts

Use the following keywords to search the number of services associated with an asset can be searched by port number:

- `service_count_tcp:<number>`
- `service_count_udp:<number>`
- `service_count_icmp:<number>`
- `service_count_arp:<number>`

These keywords support numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

Examples include:

```
service_count_tcp:>=5
```

```
service_count_arp:0
```

```
service_count_udp:<=1
```

Asset tracking fields

Timestamps

Use the following syntaxes to search the asset timestamp fields (`first_seen`, `last_seen`, `created_at`, `updated_at`, `os_eol`, `os_eol_extended`):

- `first_seen:<term>`
- `last_seen:<term>`
- `created_at:<term>`
- `updated_at:<term>`
- `os_eol:<term>`
- `os_eol_extended:<term>`

The term supports the standard runZero [time comparison syntax][time].

```
first_seen:<3days

first_seen:>2019-08-01

first_seen:>8/1/2019

last_seen:<1week

last_seen:<2months

last_seen:<1year

created_at:>2weeks

created_at:<30minutes

updated_at:>1year

updated_at:<12hours

os_eol:<now

os_eol:>4weeks

os_eol_extended:>now

os_eol_extended:>90days
```

Online status

Use the syntax `online:<boolean>` or the inverse syntax `offline:<boolean>` to search the online status of an asset.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
online:t

online:1

offline:0
```

Operating system support status

The syntax `os_eol_expired:<boolean>` can be used to find identify assets based on whether their operating systems are End of Life (EOL). This field evaluates both the `os_eol` and `os_eol_extended` values to only return assets with expired coverage.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
os_eol_expired:t

os_eol_expired:1

os_eol_expired:no
```

Detection method

The detected by attribute of an asset can be searched using the syntax `det:<term>` or `detected_by:<term>`. The term is one of `arp`, `icmp`, `<portnumber>-tcp`, or `<portnumber>-udp`. In the case of multiple detections, the priority goes `arp`, `icmp`, and then the first detected service.

```
det:arp

detected_by:80-tcp

det:53-udp
```

Time to Live (TTL) comparisons

Use the syntax `ttl:<term>` and `lowest_ttl:<term>` to search the lowest TTL of an asset. TTL is the estimated number of hops between the scan source and the asset.

This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
lowest_ttl:>3
```

Round Trip Time (RTT) comparisons

Use the syntax `rtt:<term>` and `lowest_rtt:<term>` to search the lowest RTT for an asset. RTT is the round-trip response time of a given probe measured in nanoseconds (1,000,000 == 1ms).

This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
lowest_rtt:>50000000
```

Multiple MAC address status

Use the syntax `multi_mac:<boolean>` to determine if an asset has multiple MAC addresses.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
multi_mac:t
```

Any MAC address status

Use the syntax `has_mac:<boolean>` to find assets with any MAC addresses.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_mac:yes
```

```
has_mac:f
```

Multiple IP address status

Use the syntax `multi_home:<boolean>` to determine if an asset has multiple IP addresses.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
multi_home:t
```

Multiple hostname status

Use the syntax `multi_name:<boolean>` to find assets with multiple hostnames.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
multi_name:yes
```

```
multi_name:false
```

Software installations

Use the syntax `software:<term>>` to find assets with associated software.

The term has three forms:

- `software:<product>` will look for any assets with a software product that matches the term.
- `software:<product>/<version>` will look for any assets with a software product and version that matches the term exactly.
- `software:<vendor>/<product>/<version>` will look for any assets with a software vendor, product, and version that matches the term exactly.

All three forms allow the use of `%` as a wildcard (beginning, middle, or end of the term).

Version terms can be specified with comparators to search for versions that are less than or greater than a specific version. For example `software:"Google/Google Chrome/>=135"` will returns all assets with Google Chrome version 135 or greater.

```
software:IIS
```

```
software:Microsoft/IIS/10.0
```

Service inventory

When viewing services, you can use the keywords in this section to search and filter.

Ports

The TCP and UDP services associated with a service can be searched by port number using the syntax `port:<number>`. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
port:<=25
```

TCP ports

Use the syntax `tcp:<number>` to search TCP service associated with a service by port number.

```
tcp:53
```

To search for all services on assets with a specific list of TCP ports open, you can use the syntax `service_ports_tcp:=<list>`. Values should be in ascending numerical order, and separated by commas.

```
service_ports_tcp:=80,443
```

UDP ports

Use the `udp:<number>` syntax to search UDP services associated with a service by port number.

```
udp:443
```

To search for all services on assets with a specific list of UDP ports open, you can use the syntax `service_ports_udp:=<list>`. Values should be in ascending numerical order, and separated

by commas.

```
service_ports_udp:=53,123
```

Transport

Use the syntax `transport:<term>` to search the transport associated with a service by name.

```
transport:tcp
```

```
transport:udp
```

```
transport:icmp
```

Protocol

Use the syntax `service_protocols:<term>` (or `protocol:<term>` for short) to search the protocols associated with services.

```
protocol:http
```

```
protocol:telnet
```

Assets with product

Use the syntax `service_products:<term>` (or `product:<term>` for short) to search for the identified service products associated with an asset, and return all services for the matching assets.

```
product:openssh
```

```
service_products:nginx
```

Virtual Host (vHost)

Use the syntax `vhost:<text>` to search for virtual hosts associated with a service by name .

```
vhost:"www"
```

Address

Use the keyword `service_address` to match against the service IP address.

```
service_address:192.168.0.1
```

Public address

Use the keyword `service_has_public` and syntax `service_has_public:<boolean>` to locate any service with a non-reserved I address.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*

- `false`, `f`, `0`, and `no` represent *false*

```
service_has_public:true
```

Private address

Use the keyword `service_has_private` and syntax `service_has_private:<boolean>` to locate any service with a private IP address.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
service_has_private:false
```

IPv6 address

Use the keyword `service_has_ipv6` and the syntax `service_has_ipv6:<boolean>` to locate any service with an identified IPv6 address.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
service_has_ipv6:false
```

Link-local IPv6 address

Use the keyword `service_has_link_local` and syntax `service_has_link_local:<boolean>`to locate any service with an identified IPv6 link local (`fe80::`) address.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
service_has_link_local:true
```

Assets with outlier score

You can use the syntax `outlier_score:<value>` to search the calculated outlier score of assets, and return all services on those assets. The outlier score is in the range 0 to 5 inclusive. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
outlier_score:>2
```

```
outlier_score:0
```

Assets with MAC address vendors

To search the vendors associated with the MAC addresses of an asset, and return all services on those assets, use the syntax `mac_vendor:<text>`.

```
mac_vendor:Apple
```

```
mac_vendor:"Intel Corporate"
```

To search only the vendor associated with the newest MAC address, use the syntax `newest_mac_vendor:<text>`

```
newest_mac_vendor:Apple
```

Assets with MAC address age

To search the ages of the newest MAC addresses associated with each asset, and return all services associated with those assets, use the syntax `mac_age:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
mac_age:>1year
```

```
mac_age:<6months
```

```
mac_age:2019-12-31
```

Attributes

You can search all service attributes with the syntax `<attribute>:<term>`. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

If the attribute name conflicts with an existing term, the prefix `_service.` can be added to disambiguate the query.

Note that service attributes can be slow and it is often better to prefix `_asset.protocol:<term>` filter in front of the service attribute query. For example, to search for SSH banners, use the syntax `_asset.protocol:ssh AND banner:<term>`.

```
banner:password
```

```
service.product:"OpenSSH"
```

```
html.title:"Apache2 Ubuntu Default Page"
```

```
http.code:>=500
```

```
screenshot.image.size:=>100000
```

```
_service.arp.macVendor:Xerox
```

To determine if a service has an attribute at all, use the `has` keyword. The `has` keyword can be inverted to find missing fields, with `not has:<term>`.

```
has:"http.head.server"

not has:"html.title"
```

Software inventory

When viewing software groups, you can use the keywords in this section to search and filter.

Vendor

The vendor associated with a software can be searched by name using the syntax `vendor:<name>`.

```
vendor:oracle
```

Product

The product associated with a software can be searched by name using the syntax `product:<name>`.

```
product:java
```

Version

The version associated with a software can be searched by name using the syntax `version:<name>`.

```
version:1.2.3
```

Software instance inventory

When viewing software instances on assets, you can use the keywords in this section to search and filter.

Source

The source reporting the software installed can be searched or filtered by name using the syntax `source:<name>`.

```
source:runzero
```

Vendor

The vendor associated with a software can be searched by name using the syntax `vendor:<name>`.

```
vendor:oracle
```

Product

The product associated with a software can be searched by name using the syntax `product:<name>`.

```
product:java
```

Vulnerabilities inventory

When viewing vulnerability groups, you can use the keywords in this section to search and filter.

Name

The name field can be searched using the syntax `name:<term>`.

```
name:"Cisco IOS Software DHCP Remote Code Execution Vulnerability"
```

```
name:"PHP < 5.3.12 / 5.4.2 CGI Query String Code Execution"
```

CVE

The CVE field can be searched using the syntax `cve:<term>`.

```
cve:CVE-2021-44228
```

```
cve:CVE-2016-2183
```

KEV

Membership in a Known Exploited Vulnerability (KEV) list can be searched using the syntax `kev:<term>`.

```
kev:t
```

will search for vulnerabilities that appear on a KEV list.

Specific KEV lists can be searched by name.

- `kev:cisa` will search for vulnerabilities listed as actively exploited in the CISA Known Exploited Vulnerabilities Catalog.
- `kev:vulncheck` will search for vulnerabilities listed as actively exploited in the VulnCheck Catalog.
- `kev:true` will search for vulnerabilities in either of the above lists.

Severity

The severity field can be searched using the syntax `severity:<term>`.

```
severity:info
```

```
severity:medium
```

Risk

The `Risk` and `Risk Score` fields can be searched using either numeric or keyword values. Risk score is an integer from zero through four, where `0` is Info level risk and `4` indicates Critical risk.

```
risk:"Critical"
```

```
risk:2
```

Vulnerability instance count

The `Asset count` field can be searched using the syntax `count:<text>`.

```
count:>0
```

Site name or ID

Use the syntax `site:<term>` to filter by site name or ID.

```
site:Primary
```

EPSS score

The EPSS score can be searched using the syntax `epss_score:<term>`. The term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
epss_score:>0.5
```

```
epss_score:<=0.1
```

```
epss_score:=0.9
```

Timestamps

Use the following syntaxes to search the vulnerability timestamp field (`created_at`):

- `created_at:<term>`

The term supports the standard runZero [time comparison syntax][time].

```
created_at:>2weeks
```

```
created_at:<30minutes
```

Vulnerability instance inventory

When viewing vulnerability instances on assets, you can use the following keywords to search and filter information.

Vulnerability ID

The ID field is the unique identifier for a given vulnerability, written as a UUID. Use the syntax `id:<uuid>` to filter by the ID field.

```
id:a124a141-e518-4735-9878-8e89c575b1d2
```

Source

The source reporting the vulnerability detected can be searched or filtered by name using the syntax `source:<name>`.

```
source:tenable
```

Severity

The severity field can be searched using the syntax `severity:<term>`.

```
severity:info
```

```
severity:medium
```

Severity score

The severity score can be searched using the syntax `severity_score:<term>`. The term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
severity_score:<5.0
```

```
severity_score:>=9.0
```

Risk

The risk field can be searched using the syntax `risk:<term>`.

```
risk:none
```

```
risk:critical
```

Risk score

The risk score can be searched using the syntax `risk_score:<term>`. The term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
risk_score:>7.0

risk_score:=10.0
```

Modified risk

Vulnerabilities whose risk has been modified, either manually or by way of alert rule trigger `has_modified_risk:<boolean>`.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_modified_risk:true

has_modified_risk:1

has_modified_risk:no
```

Category

The category field can be searched using the syntax `category:<term>`.

```
category:Local

category:Remote
```

Name

The name field can be searched using the syntax `name:<term>`.

```
name:"Cisco IOS Software DHCP Remote Code Execution Vulnerability"

name:"PHP < 5.3.12 / 5.4.2 CGI Query String Code Execution"
```

Description

The description field can be searched using the syntax `description:<term>`.

```
description:"The remote device is missing a vendor-supplied security patch."

description:"remote code execution"
```

Solution

The solution field can be searched using the syntax `solution:<term>`.

```
solution:patch

solution:upgrade
```

CVE

The CVE field can be searched using the syntax `cve:<term>`.

```
cve:CVE-2021-44228
```

```
cve:CVE-2016-2183
```

KEV

Membership in a Known Exploited Vulnerability (KEV) list can be searched using the syntax `kev:<term>`.

```
kev:t
```

will search for vulnerabilities that appear on a KEV list.

Specific KEV lists can be searched by name.

- `kev:cisa` will search for vulnerabilities listed as actively exploited in the CISA Known Exploited Vulnerabilities Catalog.
- `kev:vulncheck` will search for vulnerabilities listed as actively exploited in the VulnCheck Catalog.
- `kev:true` will search for vulnerabilities in either of the above lists.

Exploitable

Vulnerabilities that are exploitable can be searched using the syntax `exploitable:<boolean>`.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
exploitable:true
```

```
exploitable:1
```

```
exploitable:no
```

CVSSv2 score

The CVSSv2 fields, `cvss2_base_score` and `cvss2_temporal_score`, can be searched using the syntax `cvss2_base_score:<term>` and `cvss2_temporal_score:<term>`. The term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
cvss2_base_score:>6.5
```

```
cvss2_base_score:<=3.0
```

```
cvss2_temporal_score:=10.0

cvss2_temporal_score:<5.0
```

CVSSv3 score

The CVSSv3 fields, `cvss3_base_score` and `cvss3_temporal_score`, can be searched using the syntax `cvss3_base_score:<term>` and `cvss3_temporal_score:<term>`. The term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
cvss3_base_score:>6.5

cvss3_base_score:<=3.0

cvss3_temporal_score:=10.0

cvss3_temporal_score:<5.0
```

EPSS score

The EPSS score can be searched using the syntax `epss_score:<term>`. The term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
epss_score:>0.5

epss_score:<=0.1

epss_score:=0.9
```

Address

The address field can be searched using the syntax `address:<term>`.

```
address:192.168.0.1
```

Transport

The transport field can be searched using the syntax `transport:<term>`.

```
transport:tcp

transport:udp
```

Port

The port can be searched using the syntax `port:<term>`. The term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
port:22

port:443
```

Operating system support status

The syntax `os_eol_expired:<boolean>` can be used to find identify vulnerabilities on assets based on whether their operating systems are End of Life (EOL). This field evaluates both the `os_eol` and `os_eol_extended` values to only return vulnerabilities on assets with expired coverage.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
os_eol_expired:t

os_eol_expired:1

os_eol_expired:no
```

Finding code

The finding code field is the unique identifier for a given finding. Use the syntax `finding_code:<term>` to filter by the code field.

```
finding_code:rz-finding-internet-exposed-database
```

Finding name

Use the syntax `finding_name:<text>` to search by finding name.

```
finding_name:"Internet Exposed Database"
```

Finding risk

The `Finding Risk` and `Finding Risk Score` fields can be searched using either numeric or keyword values. Risk score is an integer from zero through four, where `0` is Info level risk and `4` indicates Critical risk.

```
finding_risk:"Critical"

finding_risk:2
```

Detection timestamps (first detected at, last detected at)

The timestamp fields, `first_detected_at` and `last_detected_at`, can be searched using the syntax `first_detected_at:<term>` and `last_detected_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
first_detected_at:>2weeks

first_detected_at:<30minutes

last_detected_at:>1month
```

```
last_detected_at:2hours
```

Publication timestamps (published at)

The timestamp field, `published_at`, can be searched using the syntax `published_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
published_at:>2weeks

published_at:<30minutes
```

Timestamps (created at, updated at)

The timestamp fields, `created_at` and `updated_at`, can be searched using the syntax `created_at:<term>` and `updated_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
created_at:>2weeks

created_at:<30minutes

updated_at:>1month

updated_at:2hours
```

Attributes

You can search all vulnerability attributes with the syntax `<attribute>:<term>`. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

If the attribute name conflicts with an existing term, the prefix `_vulnerability.` can be added to disambiguate the query.

```
plugin.hasPatch:true

severityID:3

_vulnerability.state:REOPENED
```

To determine if a vulnerability has any attribute defined, use the `has:<term>` keyword. The `has` keyword can be inverted to find missing fields with `not has:<term>`.

```
has:plugin.vpr.score

not has:output
```

Certificate inventory

When viewing certificates, you can use the following keywords to search and filter.

General certificate fields

Name

Use the syntax `name:<text>` to search for certificates by name.

```
name:example.com
```

Validity

Use `valid_from:<time>` and `valid_until:<time>` to search for certificates by when they are valid.

```
valid_from:>2025-01-01
```

```
valid_until:<2026-01-01
```

Public key algorithm

Use `pk_algorithm:<text>` to search for certificates by public key algorithm.

```
pk_algorithm:RSA
```

Public key size

Use `pk_size:<number>` to search for certificates by public key size. You will usually want to specify the public key algorithm as well, as different algorithms have different key size ranges.

```
pk_algorithm:RSA and pk_size:<2048
```

Signature algorithm

Use `sig_algorithm:<text>` to search for certificates by signature algorithm.

```
sig_algorithm:SHA1
```

Self-signed

Use `self_signed:true` to search for self-signed certificates.

Certificate authority

Use `is_ca:true` to search for certificates that are certificate authorities (CAs).

Subject

Use `subject:<text>` to search for certificates by X.509 subject DN.

```
subject:"CN=Server Name/O=Company Name"
```

Common name

You can search for certificates by common name using `cn:<name>`. This is equivalent to searching the subject DN for just the CN field.

```
cn:"Server Name"
```

Subject alternative name

You can search the four sets of Subject Alternative Names (SANs) using the following keywords:

```
san_dns_name:example.com
san_ip_address:10.0.1.23
san_email_address:postmaster@example.com
san_uri:https://example.com
```

Issuer

Use `issuer:<text>` to search for certificates by X.509 issuer DN.

```
issuer:"CN=Certificate Authority Name"
```

Subject key ID

To search by X.509 subject key ID, use `subject_key_id:<text>`. Values are accepted with or without colons in.

```
subject_key_id:"12:90:EF:DD:E1:27:A4:47:3E:32:57:AF:44:75:92:8E:8C:C2:0A:C0"
subject_key_id:1290EFDDE127A4473E3257AF4475928E8CC20AC0
```

Authority key ID

To search by X.509 authority key ID, use `authority_key_id:<text>`. Values are accepted with or without colons in.

```
authority_key_id:"12:90:EF:DD:E1:27:A4:47:3E:32:57:AF:44:75:92:8E:8C:C2:0A:C0"
authority_key_id:1290EFDDE127A4473E3257AF4475928E8CC20AC0
```

Hash

You can find certificates based on their SHA1, SHA256 or BK hash values.

```
sha1:<hash value>
sha256:<hash value>
bkhash:<hash value>
```

Serial number

Use `serial_number:<text>` to search for certificates by serial number.

```
serial_number:123456
```

Timestamps

Use the following syntaxes to search certificate inventory timestamp fields:

- `created_at:<term>`
- `updated_at:<term>`
- `last_seen:<term>`

The term supports the standard runZero time comparison syntax `[time comparison][time]`, for example:

```
last_seen:<1week

last_seen:<2months

last_seen:<1year
```

Hidden Certificates

Use `hidden:true` to search for certificates that have been hidden from the inventory.

Wireless inventory

When viewing WiFi networks, you can use the keywords in this section to search and filter.

SSID and ESSID

The SSID/ESSID field can be searched using the syntax `ssid:<text>`.

```
ssid:"Guest Network"

ssid:"Corporate"
```

BSSID (MAC)

The BSSID field can be searched using the syntax `bssid:<text>` or `mac:<text>`.

```
bssid:"00:01:02:03:04:05"

mac:"00:01:%"
```

Vendor

The vendor field can be searched using the syntax `mac_vendor:<text>`.

```
mac_vendor:"Google"

mac_vendor:"Netgear"

mac_vendor:"Cisco"
```

Family

The family field can be searched using the syntax `family:<term>`.

```
family:"010304"
```

Channels

The channels field can be searched using the syntax `channel:<term>`.

```
channel:"11"
```

Type

The network type field can be searched using the syntax `type:<text>`.

```
type:"infrastructure"
```

Interface

The network interface field can be searched using the syntax `interface:<text>`.

```
interface:"wlan0"
```

Encryption

The encryption field can be searched using the syntax `encryption:<term>`.

```
encryption:"aes"
```

```
encryption:"none"
```

Authentication

The authentication field can be searched using the syntax `authentication:<term>`.

```
authentication:"wpa2-psk"
```

```
authentication:"open"
```

Timestamps

The timestamp fields (first_seen, last_seen, created_at) timestamps can be searched using the syntax `first_seen:<term>`, `last_seen:<term>` and `created_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
first_seen:<30seconds
```

```
first_seen:>2019-08-01
```

```
last_seen:<1week

last_seen:<2months

created_at:>2weeks

created_at:<30minutes
```

Signal

The signal field can be searched using the syntax `signal:<number>` or `sig:<number>`. The term can include the operators `>`, `=`, `<=`, and `=`. The default operator is `=`.

```
signal:">75"

signal:"<=25"

signal:99
```

Organization name or ID

Use the syntax `organization:<term>` to filter by organization name or ID.

```
organization:runZero

organization:"Temporary Project"

organization:f1c3ef6d-cb41-4d55-8887-6ed3cfb3d42d
```

Site name or ID

The site name or ID can be used as a filter with the syntax `site:<term>`

```
site:Primary

site:"Branch Office"

site:ad67d649-041b-439d-af59-f200053a8899
```

Explorer name or ID

The Explorer name or ID can be used as a filter with the syntax `explorer:<term>`

```
explorer:DESKTOP-AB451F

explorer:8b927a8e-d405-40e9-aa47-d6afc9bff237
```

Wireless ID

The ID field is the unique identifier for a given wireless network, written as a UUID. This field is searched using the syntax `id:<uuid>`.

```
id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Last task ID

The Last Task ID field defines which task most recently reported the wireless network and is written as a UUID. This field is searched using the syntax `task:<uuid>`.

```
task:39ab0e71-3cf1-4176-b6b0-4ed495288229
```

Wireless attributes

All wireless attributes can be searched using the syntax `<attribute>:<term>`.

```
radio_type:"802.11n"
```

Users inventory

When viewing the Users inventory, you can use the following keywords to search and filter users.

Source

The source reporting the users can be searched or filtered by name using the syntax `source:<name>`.

```
source:ldap
```

Name fields

There are multiple name fields found in the user attributes that can be searched or filtered using the same syntax. Use the syntax `<name_field>:<text>` to search a field for matches.

The following name fields can be searched this way:

- `name`
- `display_name`
- `first_name`
- `last_name`

```
name:j

display_name:"john doe"

first_name:john

last_name:doe
```

Description

Use the syntax `description:<text>` to search the results in this field.

```
description:shared
```

Groups

The `group_id` field is the unique identifier for a given group, written as a UUID. To search for users that are part of a group based on the group's ID, use the syntax `group_id:<uuid>`.

```
group_id:f8a26321-cbce-4fb5-a9ca-ffa809489dd5
```

Email

Use the syntax `email:<address>` to search for users by email address.

```
email:john@example.com
```

Phone

Use the syntax `phone:<phone number>` to search for users by phone number.

```
phone:888-555-1234
```

Title

Use the syntax `title:<text>` to search for users by title.

```
title:CISO
```

Location

Use the syntax `location:<text>` to search for users by location.

```
location:TX
```

Last logon

Use the syntax `last_logon_at:<date>` to search for users by last logon time. This field is compatible with the date comparison operators.

```
last_logon_at:<1year
```

First seen

Use the syntax `first_seen_at:<date>` to search for users by when they were first seen in the organization. This field is compatible with the date comparison operators.

```
first_seen_at:<1day
```

Last seen

Use the syntax `last_seen_at:<date>` to search for users by when they were last seen in the organization. This field is compatible with the date comparison operators.

```
last_seen_at:>2years
```

Site name or ID

Use the syntax `site:<term>` to filter by site name or ID.

```
site:Primary

site:"Branch Office"

site:ad67d649-041b-439d-af59-f200053a8899
```

Organization name or ID

Use the syntax `organization:<term>` to filter by organization name or ID.

```
organization:runZero

organization:"Temporary Project"

organization:f1c3ef6d-cb41-4d55-8887-6ed3cfb3d42d
```

Groups inventory

When viewing the Groups inventory, you can use the following keywords to search and filter groups.

Source

The source reporting the groups can be searched or filtered by name using the syntax `source:<name>`.

```
source:ldap
```

Name fields

There are two name fields found in the group attributes that can be searched or filtered using the same syntax. Use the syntax `<name_field>:<text>` to search a field for matches.

The following name fields can be searched this way:

- `name`
- `display_name`

```
name:admin
```

```
display_name:"Domain Administrators"
```

Description

Use the syntax `description:<text>` to search the results in this field.

```
description:admin
```

Users

The `user_id` field is the unique identifier for a given user, written as a UUID. To search for groups that have a specific member based on the user's ID, use the syntax `user_id:<uuid>`.

```
user_id:f8a26321-cbce-4fb5-a9ca-ffa809489dd5
```

Email

Use the syntax `email:<address>` to search for users by email address.

```
email:john@example.com
```

First seen

Use the syntax `first_seen_at:<date>` to search for users by when they were first seen in the organization. This field is compatible with the date comparison operators.

```
first_seen_at:<1day
```

Last seen

Use the syntax `last_seen_at:<date>` to search for users by when they were last seen in the organization. This field is compatible with the date comparison operators.

```
last_seen_at:>2years
```

Site name or ID

Use the syntax `site:<term>` to filter by site name or ID.

```
site:Primary
```

```
site:"Branch Office"
```

```
site:ad67d649-041b-439d-af59-f200053a8899
```

Organization name or ID

Use the syntax `organization:<term>` to filter by organization name or ID.

```
organization:runZero
```

```
organization:"Temporary Project"

organization:f1c3ef6d-cb41-4d55-8887-6ed3cfb3d42d
```

Interface keywords

The data across your runZero account can be queried and filtered using the search syntax in conjunction with the available interface-specific keywords. Keywords and example values are documented for the following sections of the runZero Console:

- Scan templates
- Findings
- Tasks
- Analysis reports
- Explorers
- runZero users and groups
- Sites and organizations
- Credentials
- Queries
- Events

Scan templates

When viewing scan templates, you can use the keywords in this section to search and filter.

ID

The ID field is the unique identifier for a given template, written as a UUID. Use the syntax `id:<uuid>` to filter by ID field.

```
id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Name

Use the syntax `name:<text>` to search by scan template name.

```
name:WiFi

name:"Data Center"
```

Timestamps

Use the following syntaxes to search the scan template timestamp fields (`created_at`, `updated_at`):

- `created_at:<term>`
- `updated_at:<term>`

The term supports the standard runZero [time comparison syntax][time].

```
created_at:>2weeks

created_at:<30minutes

updated_at:>1year

updated_at:<12hours
```

Created by

The email address for the user that created the template can be searched using the syntax `created_by_email:<term>`.

```
created_by_email:user@example.com
```

Findings

When viewing findings, you can use the keywords in this section to search and filter.

Finding code

The finding code field is the unique identifier for a given finding. Use the syntax `finding_code:<uuid>` to filter by the code field.

```
finding_code:rz-finding-internet-exposed-database
```

Name

Use the syntax `name:<text>` to search by finding name.

```
name:"Internet Exposed Database"
```

Description

The `Description` field can be searched using the syntax `description:<text>`.

```
description:"indicated databases"
```

Solution

The `Solution` field can be searched using the syntax `solution:<text>`.

```
solution:"indicated databases"
```

Risk

The `Risk` / `Risk Rank` value can be searched using either numeric or keyword values. Risk rank is an integer from zero through four, where `0` is Info level risk and `4` indicates Critical risk.

```
risk:"Critical"

risk_rank:>2
```

Category

The finding `Category` field can be searched using the syntax `category:<text>`.

```
category:"End-of-Life"
```

Vulnerability instance count

The `Instance` field can be searched using the syntax `vulnerability_count:<text>`.

```
vulnerability_count:>0
```

Organization and site names

The names of organizations or sites affected can be searched using the following search terms:

- `organization_name:<text>`
- `site_name:<text>`

The IDs are unique and are written as UUIDs.

```
organization_id:0eacf412-6e69-11ec-88b9-f875a414a63a
```

Organization and site IDs

The IDs of organizations or sites affected can be searched using the following search terms:

- `organization_id:<uuid>`
- `site_id:<uuid>`

The IDs are unique and are written as UUIDs.

```
organization_id:0eacf412-6e69-11ec-88b9-f875a414a63a
```

Timestamps

Use the following syntaxes to search the finding timestamp fields (`last_detected_at`, `created_at`, `updated_at`):

- `created_at:<term>`
- `updated_at:<term>`
- `last_detected_at:<term>`

The term supports the standard runZero [time comparison syntax][time].

```
last_detected_at:>2weeks
```

```
created_at:<30minutes

updated_at:>1year

updated_at:<12hours
```

Tasks

When viewing all tasks, you can use the keywords in this section to search and filter them.

Name

The Name field can be searched using the syntax `name:<text>`.

```
name:"test scan"
```

Description

The Description field can be searched using the syntax `description:<text>`

```
description:"full scan"
```

Created by

The `Created By` field can be searched using the syntax `created_by:<term>`.

```
created_by:"admin"
```

Type

The task type can be searched using `type:<text>`.

```
type:scan
```

Status

The task status can be searched using `status:<text>`.

```
status:error
```

Error

The task error message can be searched using `error:<text>`.

```
error:"no disk space"
```

Recurrence frequency

The frequency tasks recur at (the “Freq” column) can be searched using `recur_frequency:<text>` or `freq:<text>`. The term `recurring:<boolean>` or `recur:<boolean>` can be used to search based on whether tasks recur at all.

```
recur_frequency:hourly

freq:daily

freq:continuous

recur:true
```

To search for tasks with a frequency of `Nth Weekday of Month`, you can use (for example) `freq:nth_weekday,2 freq:monday` to find tasks which repeat on the second monday of each month.

Timestamps (created at, updated at)

The timestamp fields, `created_at` and `updated_at`, can be searched using the syntax `created_at:<term>` and `updated_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
created_at:>2weeks

created_at:<30minutes

updated_at:>1month

updated_at:2hours
```

Next/last run time

You can search by next recurrence and last recurrence using the terms `recur_last:<term>` and `recur_next:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
recur_last:<2hours

recur_next:>1day
```

Start time

You can search by start time using the syntax `start_time:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
start_time:<2hour
```

Grace period

The grace period can be searched using the syntax `grace_period:<term>` or just `grace:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
grace:<2hour
```

Site name or ID

Use the syntax `site:<term>` to filter by site name or ID.

```
site:Primary
```

```
site:"Branch Office"
```

```
site:ad67d649-041b-439d-af59-f200053a8899
```

Template ID

Use the syntax `template_id:<term>` to filter by scan template ID.

```
template_id:de657459-041b-439d-af59-ff1f153a7722
```

Source

The data source for tasks can be searched using the term `source:<text>` or `source_id:<number>`.

```
source:censys
```

Sources are:

ID	Name	Description
1	`runzero`	runZero
2	`miradore`	Miradore
3	`aws`	Amazon Web Services
4	`crowdstrike`	CrowdStrike
5	`azure`	Microsoft Azure
6	`censys`	Censys
7	`vmware`	VMWare
8	`gcp`	Google Cloud Platform
9	`sentinelone`	SentinelOne
10	`tenable`	Tenable.io & Nessus
12	`rapid7`	Rapid7 Nexpose & InsightVM
14	`qualys`	Qualys VMDR
15	`shodan`	Shodan
16	`azuread`	Azure AD

17	`ldap`	Active Directory (LDAP)
18	`ms365defender`	Microsoft 365 Defender
19	`intune`	Microsoft Intune
20	`googleworkspace`	Google Workspace
21	`sample`	runZero traffic sampling
22	`tenablesecuritycenter`	Tenable Security Center
23	`packet`	runZero packet capture import
24	`wiz`	Wiz

Credential ID

You can search for tasks that use a specific set of credentials using `credential_id:<id>`.

```
credential_id:d7931a68-6e56-11ec-ad72-f875a414a63a
```

Parameters

Tasks can be searched for task parameters using `params:<text>`. This can be useful for searching for scan tasks that had specific probes enabled.

```
params:bacnet
```

Asset counts

Completed tasks can be searched by the asset counts found in their results. The available search terms are:

- New assets: `assets_new:<number>`
- Assets back online: `assets_back_online:<number>`
- Assets marked offline: `assets_marked_offline:<number>`
- Assets changed: `assets_changed:<number>`
- Assets unchanged: `assets_unchanged:<number>`
- Assets ignored: `assets_ignored:<number>`
- Assets updated: `assets_updated:<number>`

These terms support numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
assets_new:>0
```

```
assets_unchanged:>=1
```

Analysis reports

When viewing generated analysis reports, you can use the keywords in this section to search and filter.

Name

The Name field can be searched using the syntax `name:<text>`.

```
name:"main"
```

Description

The Description field can be searched using the syntax `description:<text>`

```
description:"compare secondary"
```

Type

The report type can be searched using the syntax `type:<text>`

```
type:outliers
```

Report ID

The ID field is the unique identifier for a given analysis report, written as a UUID. This field is searched using the syntax `id:<uuid>`.

```
id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Timestamps

The timestamp when a report was generated can be searched using the syntax `created_at:`.

The term supports the standard runZero [time comparison syntax][time].

```
created_at:>2019-08-01
```

```
created_at:<1week
```

Created by

The `Created By` field can be searched using the syntax `created_by:<term>`.

```
created_by:jsmith
```

Explorers

When viewing deployed Explorers, you can use the keywords in this section to search and filter.

Name

The Name field can be searched using the syntax `name:<text>`.

```
name:"main"
```

Site

The site can be searched using the syntax `site:<text>`.

```
site:Primary
```

Up

Whether the Explorer is up can be searched using the syntax `up:<boolean>`.

```
up:true
```

Address

The IP address(es) the Explorer is deployed on can be searched using the syntax `address:<IP address>`.

```
address:10.0.1.200
```

Version

The software version of Explorers can be searched using `version:<text>`.

```
version:2.9.7
```

Npcap version

The version of the npcap library for Windows Explorers can be searched using `npcap_version:<text>`.

```
npcap_version:1.60
```

Architecture

The machine architecture Explorers are deployed on can be searched using `architecture:<text>`.

```
architecture:amd64
```

OS

The operating system Explorers are deployed on can be searched using `os:<text>`. Note that macOS is recorded as `darwin`, the underlying Unix core of macOS.

```
os:windows
```

```
os:darwin
```

Capability

The capabilities of the Explorers can be searched using the syntax `capability:<keyword>`. Two keywords are supported:

- `screenshot` for Explorers which can screenshot web pages
- `ec2` for Explorers which can describe AWS EC2 instances

Example:

```
capability:screenshot
```

Tags

Use the syntax `tag:<term>` to search tags added to an Explorer. The term can be the tag name, or the tag name followed by an equal sign and the tag value. Tag value matches must be exact.

```
tag:"admin"
```

```
tag:"group=cloud"
```

runZero users and groups

User search keywords

When viewing users, you can use the keywords in this section to search and filter.

Email

Use the syntax `email:<address>` to search for someone by email address.

```
email:john@example.com
```

Name

Use the syntax `name:<text>` to search for someone by name.

```
name:john
```

```
name:"John Smith"
```

Superuser

To search for people based on whether they have superuser access, use the term `superuser:<boolean>`.

```
superuser:true
```

```
superuser:f
```

Access

Use the syntax `access:<term>` to search for users with a specific access level. Possible access levels are `admin`, `user`, `annotator`, `viewer`, `billing` and `none`.

```
access:admin
```

Status

To search for users by invitation status, use the term `status:<text>`. Possible status values are `activated`, `pending` and `expired`.

```
status:pending
```

SSO

To search for people based on whether they can only sign in via SSO, use the term `sso:<boolean>`.

```
sso:true
```

MFA

To search for people based on whether they have enrolled an MFA token, use the term `mfa:<boolean>`.

```
mfa:f
```

Group ID

The `group_id` field is the unique identifier for a given group, written as a UUID. To search for users that are part of a group based on the group's ID, use the syntax `group_id:<uuid>`.

```
group_id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Group name

To search for users that are part of a group based on the group's name, use the syntax `group_name:<text>`.

```
group_name:administrators
```

```
group_name:"Temp annotators"
```

User groups

To search for users based on whether they are part of a group or not, use the term `has_group:<boolean>`. Use boolean value `t` or `true` to show all users who are members of a group.

```
has_group:t
```

Conversely, use boolean value `f` or `false` to show users who are not members of a group.

```
has_group:f
```

Group count

Use the syntax `group_count:<number>`to search the group membership count. This search term supports numerical comparison operators (`>`, `>=`, `<`, `<=`, `=`).

```
group_count:>1
```

```
group_count:=0
```

Group search keywords

When viewing your groups, you can use the keywords in this section to search and filter.

ID

The ID field is the unique identifier for a given group, written as a UUID. Use the syntax `id:<uuid>` to filter by ID field.

```
id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Name

Use the syntax `name:<text>` to search by group name.

```
name:administrators
```

```
name:"Temp annotators"
```

Access

Use the syntax `access:<term>` to search for groups with a specific access level. Possible access levels are `admin`, `user`, `annotator`, `viewer`, `billing` and `none`.

```
access:admin
```

Timestamps (created at, updated at)

Filter groups by their timestamp fields, `created_at` and `updated_at`, using the syntax `created_at:<term>` and `updated_at:<term>`. The terms support the standard runZero [time comparison syntax][time].

```
created_at:<30days
```

```
updated_at:<1week
```

Expiration

Filter groups by their expiration timestamp, `expires_at`, using the syntax `expires_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
expires_at:<30days
```

```
expires_at:>8/1/2019
```

The `expired` property describes whether or not a group has expired. Search this property using the `expired:<boolean>` syntax.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
expired:true
```

```
expired:0
```

Use the syntax `has_expiration:<term>` to find any assets with an expiration date.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
has_expiration:true
```

```
has_expiration:0
```

Email

The `created_by_email` property holds the email address for the user that created the group. It can be searched using the syntax `created_by_email:<term>`.

```
created_by_email:user@runzero.com
```

Group mapping search keywords

When viewing your SSO group mappings, you can use the keywords in this section to search and filter.

ID

The ID field is the unique identifier for a given group mapping, written as a UUID. Use the syntax `id:<uuid>` to filter by ID field.

```
id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

SSO attribute

The `sso_attribute` is the name of the attribute field to check for matching values. Use the syntax `sso_attribute:<text>` to search by `sso_attribute`.

```
sso_attribute:department
```

SSO value

The `sso_value` is the value or comma-separated list of values to match. Use the syntax `sso_value:<text>` to search by `sso_value`.

```
sso_value:security
```

```
sso_value:"admins, administrators"
```

Group ID

The `group_id` field is the unique identifier for a given group, written as a UUID. To search for group mappings related to a group based on the group's ID, use the syntax `group_id:<uuid>`.

```
group_id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Group name

To search for group mappings related to a group based on the group's name, use the syntax `group_name:<text>`.

```
group_name:administrators
```

```
group_name:"Temp annotators"
```

Timestamps (created at, updated at)

Filter group mappings by their timestamp fields, `created_at` and `updated_at`, using the syntax `created_at:<term>` and `updated_at:<term>`. The terms support the standard runZero [time comparison syntax][time].

```
created_at:<30days
```

```
updated_at:<1week
```

Email

The `created_by_email` property holds the email address for the user that created the group. It can be searched using the syntax `created_by_email:<term>`.

```
created_by_email:user@runzero.com
```

Sites and organizations

Site search keywords

When viewing sites, you can use the keywords in this section to search and filter.

Name

The Name field can be searched using the syntax `name:<text>`.

```
name:"Primary"
```

Description

The Description field can be searched using the syntax `description:<text>`.

```
description:"wireless"
```

```
description:"vlan 50"
```

Scope

The Scope field can be searched using the syntax `scope:<term>` .

```
scope:"10.10.10."
```

Excludes

The Excludes field can be searched using the syntax `excludes:<term>` .

```
excludes:"192.168.0."
```

Timestamps (created at, updated at)

The timestamp fields (`created_at`, `updated_at`) timestamps can be searched using the syntax `created_at:<term>` and `updated_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
created_at:>2weeks
```

```
created_at:<30minutes
```

```
updated_at:>1month
```

```
updated_at:2hours
```

Organization search keywords

Name

The Name field can be searched using the syntax `name:<text>`.

```
name:"main"
```

Description

The Description field can be searched using the syntax `description:<text>`

```
description:"branch office"

description:"pci"
```

Timestamps (created at, updated at)

The timestamp fields (`created_at`, `updated_at`) timestamps can be searched using the syntax `created_at:<term>` and `updated_at:<term>`. The term supports the standard runZero [time comparison syntax][time].

```
created_at:>2weeks

created_at:<30minutes

updated_at:>1month

updated_at:2hours
```

Credentials

When viewing saved credentials, you can use the keywords in this section to search and filter.

Credential fields

Credential ID

The ID field is the unique identifier for a given credential, written as a UUID. This field is searched using the syntax `id:<uuid>`.

```
id:cdb084f9-4811-445c-8ea1-3ea9cf88d536
```

Credential name

The credential name can be searched using the syntax `name:<text>`.

```
name:"AWS read-only account"

name:"Miradore API key"
```

Credential type

The credential type can be searched using the syntax `name:<text>`.

```
type:aws_access_secret

type:miradore_api_key_v1
```

Credential global property

The global property describes the level of access for all organizations. If a credential is global, all organizations have access to it. The global property can be searched using the syntax

`global:<boolean>`.

The term is a boolean value:

- `true`, `t`, `1`, and `yes` represent *true*
- `false`, `f`, `0`, and `no` represent *false*

```
global:true
```

```
global:0
```

Credential timestamps

Credential timestamp fields (`created_at` and `last_used_at`) can be searched using the syntax:

- `created_at:<term>`
- `last_used_at:<term>`

The term supports the standard runZero [time comparison syntax][time].

```
created_at:<3days
```

```
created_at:>2019-08-01
```

```
created_at:>8/1/2019
```

```
created_at:<1week
```

```
created_at:<2months
```

```
last_used_at:<1year
```

```
last_used_at:>2weeks
```

```
last_used_at:<30minutes
```

```
last_used_at:>1year
```

```
last_used_at:<12hours
```

```
last_used_at:0
```

Credential created by

The created_by_email holds the email address for the user that created the credential. It can be searched using the syntax `created_by_email:<term>`.

```
created_by_email:user@example.com
```

Queries

When viewing saved queries, you can use the keywords in this section to search and filter.

Name

The `Name` field can be searched using the syntax `name:<text>`.

```
name:"smb2"
```

Description

The `Description` field can be searched using the syntax `description:<text>`.

```
description:"smb version 1"
```

```
description:"wep"
```

Type

The `Type` field can be searched using the syntax `type:<term>` .

```
type:"services"
```

Category

The `Category` field can be searched using the syntax `category:<term>`.

```
category:"security"
```

```
category:"audit"
```

Severity

The `Severity` field can be searched using the syntax `severity:<term>`.

```
severity:"info"
```

```
severity:"critical"
```

Created by

The `Created By` field can be searched using the syntax `created_by:<term>`.

```
created_by:"runzero"
```

Timestamps (created at, updated at)

The timestamp fields, `created_at` and `updated_at`, can be searched using the syntax `created_at:<term>` and `updated_at:<term>`. The term supports the standard runZero [time comparison syntax] [time].

```
created_at:>2weeks
```

```
created_at:<30minutes

updated_at:>1month

updated_at:2hours
```

Events

When viewing system events under alerts, you can use the keywords in this section to search and filter.

Note that event records are retained for one year.

Action

Use the syntax `action:<text>` to search by the action which caused the event.

```
action:agent-reconnected
```

Created timestamp

The timestamp fields `created_at` can be searched using the syntax `created_at:<term>`. The term supports the standard runZero time comparison syntax.

```
created_at:>2weeks

created_at:<30minutes

updated_at:>1month

updated_at:2hours
```

Details

The details in the event record can be searched using the syntax `details:<text>`. This can be useful for searching for IP addresses.

```
details:192.168.0.1
```

Source and target name

The source (src) column can be searched using the syntax `src:<text>` or `source:<text>`. The target (tgt) column can be searched using `tgt:<text>` `target:<text>`.

```
src:crowdstrike

target:primary
```

Source and target type

The source type (shown at the start of the src column) can be searched using the syntax `src_type:<text>` or `source_type:<text>`.

Similarly, the target type can be searched using `tgt_type:<text>` or `target_type:<text>`.

```
src_type:task
```

```
target_type:site
```

Organization, site, source and target IDs

The IDs of organizations, sites, sources and targets mentioned in event details can be searched using the following search terms:

- `organization_id:<uuid>`
- `site_id:<uuid>`
- `source_id:<uuid>` or `src_id:<uuid>`
- `target_id:<uuid>` or `tgt_id:<uuid>`

The IDs are unique and are written as UUIDs.

```
organization_id:0eacf412-6e69-11ec-88b9-f875a414a63a
```

Automating queries

Community Platform

runZero's query language allows you to search and filter your asset inventory, based on asset fields and values. runZero includes a query library of prebuilt searches which can be browsed from the Queries page. You can apply these queries after a scan to investigate discovery findings.

In addition to a flexible query language, the same search syntax can be used to track and monitor events across your network, based on any combination of fields. You can save your custom queries to reuse over and over again. Review the query syntax documentation for a refresher on how to build a query.

If there are certain queries you always want to run after a scan, you can turn on the **Auto** option. After the query runs, you will be able to view its results in the Queries table.

From the Queries table, any filter that has checkmark in the *Auto* column means that it is active. The *Matches* column is next to it. You can click on the numbered result in the *Matches* column to view the results of the query. If there were no matches, the filter will show 0.

Turn on automatic search queries

1. Select Queries from the left navigator.

2. Open or create the query you want to turn on.
3. Turn on the **Automatically track query results on the dashboard** option.
4. Save the query.

Data export

Exporting asset data

The inventory view provides a few ways to export asset data. The `Export` menu offers Export All options in both CSV and JSON format, and when a search query has been provided, options to export just the search results as both CSV and JSON.

The CSV format can be opened with tools like Microsoft Excel and easily imported into other applications but does not contain the full details of certain fields, such as Services. The JSON format contains a complete export but may take additional processing to use with other tools.

If you have a specific export/import scenario in mind, please contact support and let us know. The Export API leverages both the CSV and JSON formats and supports arbitrary search queries in the same syntax as the inventory.

Exporting HP iLO data

Community Platform

HP Integrated Lights-Out (iLO) provides remote management, configuration, and monitoring capabilities for HP servers. These capabilities centralize operations for your server environments and streamline tasks like rebooting servers, booting into single user mode, and bypassing authentication.

Being able to identify and find the serial number for HP iLO devices is useful for tracking warranties for support and contract management. If you have runZero Platform, you can export the runZero data as a CSV to feed into warranty tracking tools.

How to export HP iLO CSV data

In your runZero Console, go to your inventory. From the **Export** menu, choose the **HP iLO CSV format**. This method downloads all HP iLO data from the runZero inventory to a CSV file.

If you want to refine the results in your exported data, you can filter the inventory first. For example, if you only want to export iLOs that have the ProLiant DL360p Gen8 hardware, you can query:

```
alive:t AND ilo.hardware:"=ProLiant DL360p Gen8"
```

And then, from the Export menu, choose the **HP iLO CSV format** from the Export Search Results submenu.

HP iLO CSV export data

When you export HP iLO data, the CSV file will contain the following fields:

- IP address
- MAC address
- Name
- Serial number
- Product ID
- Model
- Version
- Health status
- IRS
- iLO product name
- Serial number
- Boot block
- HW revision
- FW revision
- PWRM
- Auth local
- Auth Kerberos

- Auth LDAP
- Auth license
- Security message
- Alive
- ID
- Site ID
- Last seen

There is a lot of data in the export that you may not need for your warranty tracker. Fields, such as serial numbers, physical hardware information, health status, and firmware version may be the most useful to import into your tracker.

Advanced reports

Switch topology

Platform

The runZero switch topology report allows you to view a graph of the switches and routers on your network, and see how they are interconnected. It will also show which assets are connected to each switch.

> Note
> The switch topology report requires a source of layer 2 topology information. This is typically obtained from SNMP, or by connecting an integration that provides topology information, such as Cisco Meraki.

Generating the switch topology report

Once you have obtained topology data, you can launch the switch topology report.

The initial view shows a graph of network switches, with the links between them shown as lines. Beneath each switch you can see a count of how many known assets were detected as having that switch as their upstream network connection.

You may also see a count of unmapped assets listed. These are assets inferred to exist by MAC addresses which were found in the topology data, but the MAC address isn't known to belong to any scanned asset. Lists of unmapped MACs are also available in tabular form from the Unmapped MACs report.

At the bottom of the page you will see the total number of assets in the graph, and the total number of unmapped MACs.

The report will attempt to lay out the graph appropriately. You can drag nodes around to make it clearer

Clicking on a node representing a switch will display a pop-up window with more information about it. This includes a link to view the asset details for the switch, and a link to view a list of the unmapped assets.

Double-clicking a node will expand it to show the individual assets connected to it. Clicking an asset will show a pop-up window with a link to view the asset details.

Filtering the switch topology report

The switch topology report is limited in the number of assets it can display. To focus on the set of assets you are interested in, you can use the filter box at the top of the report page. It

accepts search strings in the standard runZero search language. For example, you could filter to a given subnet using a search string such as `cidr:10.1.6.0/24`.

There is also a drop-down to switch quickly between sites.

Additional buttons in the filter box allow you to collapse or expand all of the switch nodes at once.

The *Export view* button will render a PNG file of the graph as currently displayed, and download it.

Limitations of the switch topology report

The switch topology report may not always be entirely accurate because of limitations on the data runZero can gather.

In the case where there is SNMP data available, runZero will pull a snapshot of the SNMP data from each device when it is scanned, then use that to build topology. However, in many cases a single infrequent snapshot is not enough to show a complete picture in complicated environments, and links may end up missing. In addition, only recent SNMP data is used — if devices have not been scanned in the last 9 days, their SNMP topology data will not be used.

When there is no SNMP information, runZero will attempt to compute topology based on which switch claims to have seen the MAC, which may not always be the nearest access switch. Our algorithm looks for the port with the least number of shared MACs to find best match, but that depends on the switch cache timeouts and how the switch was scanned, so there may be links shown that don't exist as direct physical connections.

Cisco Catalyst devices

SNMPv3 on Cisco Catalyst devices will not let you pull the bridge port information that we need unless you specifically enable per-VLAN access.

SNMP v3 access to VLAN ARP/FDB tables requires this access rule:

Version	Command
Newer IOS:	`snmp-server group YourGroupName v3 auth context vlan- match prefix`
Older IOS:	`snmp-server group YourGroupName v3 auth context vlan-1` (repeated for every VLAN)

Note that even after this is done, runZero will need to send a separate SNMP request for every VLAN. This can significantly slow down scans with SNMP enabled on a network with many Catalyst devices.

Asset route pathing

Platform

The asset route pathing report generates a visualization of the potential network paths between a source asset and destination asset in an organization. Following the paths, you can see assets connected between the target and source destinations. These assets represent opportunities an attacker could potentially leverage to break into your target asset.

The runZero Explorer performs a traceroute between itself and the source, and then another with the target. runZero then compares the data to infer shared points between the assets. runZero does not get any paths from a direct traceroute. This is runZero's best effort — based on the scan data it has — to identify the assets that it sees as viable points between two assets.

You can share this report with your IT and security teams to highlight assets that could be leveraged as pivot points to your critical assets. Armed with this information, they can identify systems that may need to be hardened. They can assess whether or not the appropriate critical controls are in place to prevent unauthorized access to those assets.

Customers with highly segmented environments can use this report to quickly identify paths from low security assets to critical assets. For example, this report can indicate whether a device in a wireless guest network can reach a system within the PCI cardholder data environment.

Terminology

Before diving into the asset route pathing report, here are some terms you need to know:

- **Hop** - Any node between the source and destination.
- **Node** - Any asset or hop. Nodes can be an IP, an asset, or unknown.
- **Asset** - Any device that is part of your runZero inventory.
- **Network path** - runZero does not get any paths from a direct traceroute. Instead, runZero uses the data it has to identify and display potential network paths between the target and source.

Generating the asset route pathing report

1. Launch the asset route pathing report.

2. When the *Asset route pathing* page appears, you will need to select a source asset and a destination asset. Use the search to filter assets by keyword or the table pagination to browse all of your assets.

3. After you have a source and destination asset selected, start the trace.

4. In the generated results, you will see the potential paths between the source and destination asset.

Analyzing the report

Locate your source asset and target asset. If there are hops between the two assets, you should review them and secure the paths between them. Take a look at the services running on those systems that may provide potential entry points for attacks and harden them.

Nodes in the asset route pathing report are color-coded to help you identify the source asset and destination asset.

The report uses the following colors:

- Green - The source asset
- Red - The destination asset
- Orange - A multi-homed asset that may act as a pivot point
- Blue - A standard layer-3 routing hop
- Gray - Asset is unknown

A hop labeled `Unknown` indicates an intermediate hop in the layer-3 path that did not respond with ICMP errors for TTL exceeded packets.

Sharing the report

There are a couple of ways to share the results from the asset route pathing report. You can either export a PNG or dotfile of the report or share a direct link to the report.

- **Export a PNG** - A PNG export will take a snapshot of your report. To share the asset route pathing report, click **Export view**. A PNG will download to your computer.
- **Export a dotfile** - You can export a dotfile and feed it into a Graphviz engine or open source visualization tool. The file allows you to render the image in formats like SVG, PSD, and PNG.
- **Share a link** - When viewing the generated asset route pathing report, copy the URL. You can share the URL directly with other team members who have a runZero account and access to the organization.

Exporting a dotfile

A dotfile is a text file that can be fed into Graphviz engines or open source visualization tools. With the dot file, you can render the image in other formats, like SVG, PSD, and PNG.

1. Launch the asset route pathing report.

2. When the *Trace path* page appears, you will need to select a source asset and a destination asset. Use the search to filter assets by keyword or the table pagination to browse all of your assets.

3. After you have a source and destination asset selected, generate the report.

4. After the visualization appears, click the **Export report** button. A window appears prompting you to enter a name and download location for the file.
5. Enter a name for the file and choose where you want to save it on your computer.
6. Save the file.

FAQs

Why are there hops in the report that aren't in my inventory?

Not every hop is a runZero asset. runZero fills in IP addresses for some asset hops, based on information that is pulled out of the traceroute data. Sometimes, runZero is able to extract asset information from the traceroute data that isn't part of the inventory, which is why you may be seeing them in the report.

Can runZero determine how two assets are talking to each other?

No, runZero can only identify potential paths–not if they are routable. runZero does not test or validate the paths.

Why isn't the asset route pathing report available?

The asset route pathing report is only available for runZero Platform customers.

Scan coverage

Coverage reports help you understand potential blind spots on your network by identifying which IP spaces have been scanned, which ones contain assets, and which ones still are unknown. With this information, you can find things like missing subnets, rogue devices, and misconfigurations.

To access the coverage reports, go to **Reports** on the main menu and scroll down to **RFC 1918 coverage**.

RFC1918 coverage report

The RFC1918 coverage report helps you better track and identify the subnets that are in use on your internal network, the ones that have been scanned, and the ones that haven't been scanned.

TCP/IP version 4 reserves three ranges of IP addresses for private use. Specified in RFC1918, they are:

CIDR	Address range	Number of addresses
10.0.0.0/8	10.0.0.0 - 10.255.255.255	16,777,216
172.16.0.0/12	172.16.0.0 - 172.31.255.255	1,048,576
192.168.0.0/16	192.168.0.0 - 192.168.255.255	65,536

Most companies use these address ranges for their internal IPv4 networks, connecting them to the Internet via Network Address Translation (NAT).

To help you visualize and assess your RFC1918 coverage, the RFC1918 report includes:

- Coverage maps
- Subgrids
- Statistics

RFC1918 coverage maps

A common network security and administration goal is to scan all of the available private IP addresses, to detect which subnetworks are in use on your internal network. Because of the large number of addresses, this can take a long time, leaving the problem of tracking which addresses have been scanned and which have not.

To solve this problem, runZero's Coverage report shows a graphical map of the RFC1918 private address spaces, showing which pieces have been scanned and to what percentage of completion.

runZero will sometimes detect that a device has additional IP addresses which are not part of the range being scanned. This can indicate that the device is present on an unscanned part of the private IP address space. The coverage reports show this by drawing a red border around the appropriate grid cell.

You can hover the mouse cursor over a cell to see a tooltip showing the CIDR address range the cell represents, how many unscanned hosts are believed to be in that range, and what percentage of the entire range has been scanned.

RFC1918 sub-grids

For the 192.168.0.0/16 map, each cell on the grid represents a /24 (256 addresses). Clicking a cell will take you to the subnet analysis report for that range and list the assets found.

For a map that shows a large address range, such as 10.0.0.0/8, each cell represents an entire /16 range of 65,536 addresses. To help narrow down the search for assets and unscanned hosts, you can click on any cell that represents a /16 range to go a grid map that showing just that range. From a /16 grid, you can use the link at top right to go back to the full range map.

On the /16 sub-grids, each cell is a /24, so clicking one takes you to the subnet analysis for that specific cell's address range, like on the 192.168.0.0/16 map.

Statistics

At the top of the coverage report page, you can see statistics showing how much of the RFC1918 address space you have scanned. Another box breaks the coverage down by the three blocks of reserved addresses.

Clicking the magnifying glass icon in the summary box will create a sample scan task, covering the unscanned address ranges. The refresh buttons create scan tasks to rescan all of the appropriate range.

Site comparison

Platform

The Site and organization comparison feature lets you generate a side-by-side analysis of two sites, so you can understand:

- How assets change over time such as their TCP/UDP services, TCP/UDP ports, and service protocols. You can leverage this data to evaluate historical changes for assets for a specific point in time.
- How exposure changes based on scanning your network from different locations. For example, if you use public IP addresses internally and externally, you may want to scan those addresses from inside and outside your network to understand your potential exposure.

The report provides a summary view of differences. It only captures certain attributes that were added or removed from an asset, such as IP addresses, TCP ports, TCP service counts, UDP ports, UDP service counts, service protocols, and service counts. It does not track every modification to an asset, such as fingerprint or service banner changes.

After you run the report, the data presented in the report will be static. Any changes to your current inventory may result in assets no longer being accessible from the report.

Generate a site comparison

Generating the site comparison requires selecting a current site and a comparison site. The sites can be in different organizations. You can also select “All sites” to compare all sites in an organization with a different site and organization.

When the report runs it will assemble the two sets of assets specified, compare them using runZero’s asset matching algorithms, and generate a set of differences. You can then browse the summarized results in the report.

To generate a site comparison:

1. Verify that your current organization is one containing an inventory of assets you want to compare.
2. Go to the site comparison page.
3. When the site comparison configuration page appears, the current organization will be set to the one you currently have selected. You can change the current site, if needed. For the comparison, choose the organization and site you want to run the analysis against.
4. Run the report. A task will be created to perform the comparisons, and you will be taken to the task page.
5. When the task is complete, the report will appear in the list of recent analysis reports at the top of the Reports page. You can then view and search the results.

View how assets change over time

To analyze how assets have changed over time, you can compare the data from an old scan task with your most recent inventory. Setting a point-in-time comparison requires creating a new project that you can import your old scan data into.

Here's how you can set up a point-in-time analysis:

1. Go to the organization or project that contains the task scan data you'd like to use.
2. Go to your completed tasks and locate the task that contains the data for the point in time you'd like to compare.
3. From the task page, download the task data. It will be in a file with a name starting `scan_` and ending `.json.gz`. This is the file you'll import into your new project. You don't need to uncompress the file, unless you're curious to look at the JSON data.
4. Next, create a new project for your import. You can create an organization if you intend to perform the analysis regularly.
5. After you create the project, go to the Inventory page and import your scan task data into it.

Now, you're ready to compare your current inventory with a previous version of it. Go to the site comparison page. You'll need to select the organization and site for your current inventory as the site to compare against.

After the report runs, it shows a table that highlights the differences between the two sites, which in this case will represent two points in time, going from past to present.

You can also run the comparison by selecting your current organization first, and then choosing the project with the past data as the comparison site. The results will be the same, but with their sense reversed – that is, services which show as added when going from past to present, will show as removed when going from present to past.

View how exposure differs between networks

You can run the site comparison report to compare how exposure varies based on where you scan your network from. By comparing two inventories from two different perspectives, you can obtain better insights on your attack surface, which can help with active defense or risk reduction. For example, if you use external IPs internally and externally, you may want to scan those addresses from inside your network and outside your network. Then, you can run the site comparison report to compare the results from those two sites.

Here's how you can set up a diff for exposures between networks:

1. Set up a site with an Explorer on one network.
2. Set up another site with an Explorer on a different network. This site can be in the same organization as the first site.
3. Run a scan of the same address range with each Explorer. Verify you have the correct site selected for each scan.
4. After the scans complete, run the site comparison to generate the diff. The report will show a table that highlights the differences between the two sites.

Analyze the results in the site comparison report

The site comparison generates a table to show how assets' addresses, names, services, ports, and protocols differ between sites. Red text, denoted by the minus (-) sign, indicates that attributes were removed going from left to right. Green text, denoted by the plus (+) sign, indicates attributes were added.

Here are some things you should know about the results:

- **Address and Other Address** - The first address column contains the asset's addresses for the organization currently selected when the report was requested. The second address column, *Other Address*, contains the asset's addresses for the organization and site that was compared against the current organization.
- **Name and Other Name** - The first name column contains the asset's names in the organization currently selected when the report was generated. The second address column, *Other Name*, contains the asset's names for the organization and site compared against.
- **TCP and UDP services** - These columns show how the total number of TCP and UDP services differ between sites.
- **TCP and UDP ports** - These columns show the ports that have been added or removed between sites.

You can click on the green info (i) icon to view a more detailed comparison of the asset.

Clicking on one of the asset addresses in the report will bring up the full current asset record, if the asset still exists in the appropriate organization and site.

Search the site comparison report

You can search the report using the runZero search query language. In the following descriptions, the main set refers to the assets in the organization that was current when you generated the report (i.e., the address and name columns). The comparison set refers to the assets in the organization and site that you chose to compare against (i.e., the other address and other name columns).

Keyword	Meaning
`address:`	Search for assets in the main set with a specified IP address. Use `none` to find assets that are missing from the main set.
`net:` or `cidr:`	Filter assets by their network CIDR range in the main set.
`other_address:`	Search for assets in the comparison set with a specified IP address. Use `none` to find assets that are missing from the comparison set.
`other_net:` or `other_cidr:`	Filter assets by their network CIDR range in the comparison set.
`id:`	Search by asset ID in the main set.

`tcp:`	Search for a TCP port change by number.
`udp:`	Search for a UDP port change by number.
`protocol:`	Search for a TCP or UDP port change by service name.

Organization overview

Community Platform

The Organization Overview Report captures a point-in-time snapshot of the asset data within your organization and sites. The report organizes data from your asset inventory into relevant sections and summarizes the major findings. The Organization Overview Report is useful for sharing with teams and leaders who may not have access to runZero and need an at-a-glance look into their network.

The report helps you quickly assess high-level metrics across multiple categories for your organization and sites, such as your asset types, operating systems, hardware, protocols, and products. The report also includes a summary of your RFC 1918 coverage, subnet utilization, and switches. For organizations with less than 50,000 assets, you can include additional information from your inventory via asset details and screenshots.

You can schedule the report to run at a specified start time and at a specified frequency. This report can run once, daily, weekly, monthly, or on the Nth weekday of each month. When a report is configured to run recurrently, one or more email addresses can be notified when the report has run.

runZero generates this report in HTML, but you can use your browser's Print to PDF feature to save the file as a PDF for distribution.

Generating the Organization Overview Report

1. Go to the Reports page.
2. Open the Organization Overview Report configuration form.
3. Enable **Include asset details** if you want additional information about each asset. The asset details will include the asset type, operating system, hardware, outlier score, first seen date, last seen date, site, addresses, names, MAC addresses, protocols, products, and services. This option only applies to organizations with less than 50,000 assets.
4. Enable **Include screenshots** if you want to see captured screenshots in the report. This option only applies to organizations with less than 50,000 assets.
5. Configure **Start time** and **Report frequency** if you want the report to run at a specified time or on a recurring schedule.
6. Provide one or more email addresses in the **Email addresses** field if you want email notifications when the report is run.
7. Generate the report. When the report generation completes, it automatically opens in your browser. Save the report as a PDF to share and distribute as needed.

Email notifications

You can include a comma-separated list of email addresses when configuring the report to run on a schedule. The specified email addresses will receive notifications when the report has finished each run of the configured schedule. The notification email will contain a link to

the report in the runZero Console. Please note that only registered users with access to the relevant organization will be able to access the completed report.

External assets

Platform

The External Asset Report captures a point-in-time snapshot of the external asset data within your organization and sites. The report organizes data from your asset inventory into relevant sections and summarizes the major findings. The External Asset Report is useful for sharing with teams and leaders who may not have access to runZero and need an at-a-glance look into their public-facing assets.

The report helps you quickly assess high-level metrics across multiple categories for external assets in your organization and sites, such as asset types, operating systems, hardware, protocols, and products. The report also includes a summary of top Certificate Authorities and GeoIP countries. For organizations with less than 50,000 assets, you can include additional information from your inventory via asset details and screenshots.

You can schedule the report to run at a specified start time and at a specified frequency. This report can run once, daily, weekly, monthly, or on the Nth weekday of each month. When a report is configured to run recurrently, one or more email addresses can be notified when the report has run.

runZero generates this report in HTML, but you can use your browser's Print to PDF feature to save the file as a PDF for distribution.

Generating the External Asset Report

1. Go to the Reports page.
2. Open the External Asset Report configuration form.
3. Enable **Include asset details** if you want additional information about each asset. The asset details will include the asset type, operating system, hardware, outlier score, first seen date, last seen date, site, addresses, names, MAC addresses, protocols, products, and services. This option only applies to organizations with less than 50,000 assets.
4. Enable **Include screenshots** if you want to see captured screenshots in the report. This option only applies to organizations with less than 50,000 assets.
5. Enable **Include unscanned subnet details** if you want to see results from external subnets that have not been scanned by runZero.
6. Enable **Include TLS certificate details** if you want to see details gathered about TLS certificates.
7. Use the **External IP exlusions** if you are using public IP address space internally and do not want those assets included in the report.
8. Configure **Start time** and **Report frequency** if you want the report to run at a specified time or on a recurring schedule.
9. Provide one or more email addresses in the **Email addresses** field if you want email notifications when the report is run.
10. Generate the report. When the report generation completes, it automatically opens in your browser. Save the report as a PDF to share and distribute as needed.

Email notifications

You can include a comma-separated list of email addresses when configuring the report to run on a schedule. The specified email addresses will receive notifications when the report has finished each run of the configured schedule. The notification email will contain a link to the report in the runZero Console. Please note that only registered users with access to the relevant organization will be able to access the completed report.

Additional resources

Leveraging the API

runZero provides three primary APIs as well as integration-specific endpoints:

- The Export API provides read-only access to a specific organizations.
- The Organization API provides read-write access to a specific organizations (Professional and Platform licenses).
- The Account API provides read-write access to all account settings and organizations (Platform license).

To get started, you will need an API key / token or API client credentials.

API keys and tokens

The console supports five types of runZero API key, with different levels of access to runZero APIs.

Export tokens

An export token only allows access to export data via the export API endpoints under the `/api/v1.0/export` path. It cannot access any other APIs.

An export token has information about its organization encoded into it. It can only be used to export data for that organization. There is no need to specify the organization when using an export token.

Export tokens can be recognized by their `ET` prefix.

To generate an export token, go to the Organizations page, click on the desired organization to view its details page. Then, click **Edit organization** and scroll down to the export tokens section and use the button to generate or regenerate the token.

Download tokens

A download token allows access to download the runZero Explorer. This can be useful if you want to automate downloads for deployment across many machines, or if you need to containerize the Explorer. It cannot be used to access any other API.

Like export tokens, the download token has information about the organization encoded into it. This determines the organization the downloaded Explorer will be associated with.

Download tokens can be recognized by their `DT` prefix.

To obtain the download API token, go to the Organizations page, click on the desired organization to view its details page. Then, click **Edit organization** and scroll down to the download token section.

Organization API tokens

An organization API token allows read and write access to data within a particular organization, using the organization API. Most of the organization endpoints are under the `/api/v1.0/org` path. Organization tokens can also be used to access the export APIs.

An organization API token has information about the organization encoded into it. It can only be used to access that organization. There is no need to specify the organization when making a call using an organization API token.

Organization API tokens can be recognized by their `OT` prefix.

To generate an organization API token, go to the Organizations page, click on the desired organization to view its details page. Then, click **Edit organization** and scroll down to the organization API tokens section. You can use the buttons to generate named organization API tokens and revoke them.

Account API tokens

Platform

An account API token allows read and write access to the account API endpoints. These provide access to perform account-level operations such as creating and deleting organizations and users. Account API tokens can also be used for organization level access and for the export API.

Account API tokens require a Platform license and are generated from the Account settings page. To use an account API token with the Organization or Export API, specify the additional parameter `_oid=[organization-id]` in the query parameters.

An account API token has no specific organization encoded into it. When using an account API token with an API call that applies to an organization, the organization must be specified. This can be done by appending an `_oid` value to the URL query parameters. The `_oid` should be the unique ID of the organization. This can be found on the organization's information page.

Account API tokens can be recognized by their `CT` prefix (client token).

To generate an account API token, go to the Account settings page under Account in the left navigator. You can create and delete account API tokens as needed.

API client credentials

Platform

The organization and account tokens above are used as bearer tokens for API access. The API client credentials are different; they are used to obtain account API tokens programmatically, via OAuth2.

Account API client credentials are managed from the API clients page. Your REST client should use the OAuth 2.0 authorization type and Client Credentials grant type. See the OpenAPI specification for the access token details.

Authentication

Once you have a token or some API client credentials, you can authenticate against the runZero API.

For export, organization and account tokens, your REST client should use the token with the `Authorization: Bearer` standard header to authenticate.

For API client credentials, you must use the generated client ID and client secret to make an OAuth2 call to generate an access token. For example:

```
curl -X POST -H "Content-Type: application/x-www-form-urlencoded" \
  -d "grant_type=client_credentials&client_id=<CLIENT_ID>&client_secret=<CLIENT_SECRET>" \
  https://console.runzero.com/api/v1.0/account/api/token`
```

API rate limiting

API calls are rate limited. You can make as many API calls per day as you have licensed assets in your account. For example, if you have 1,000 licensed assets, you can make 1,000 API calls per day. Each API call returns rate limit information in the HTTP headers of the response:

- `X-API-Usage-Total` - Total number of calls made to the API
- `X-API-Usage-Today` - Number of calls made to the API today
- `X-API-Usage-Limit` - Your daily API call limit, shared across all API keys
- `X-API-Usage-Remaining` - The number of API calls remaining from your daily limit

In addition, there's a limit of 2,000 requests per 5 minutes per source IP address. If this is exceeded, an HTTP 429 response will be sent with an error of "request limit exceeded", and your client code should delay the next request.

Additional documentation

Please see the Swagger documentation and runZero OpenAPI specification for details on the individual API calls.

Using the CLI

Platform

The runZero Command Line Interface (CLI) provides various utility functions. For licensed users, it also allows standalone network scanning.

Scanner

The `scan` command has the same options as the runZero Explorer, and similar performance characteristics. The output file named `scan.runzero.gz` can be uploaded to the runZero Console through the Inventory *Import* menu. This

The CLI scanner works best with root privileges on Linux/macOS and Administrator privileges on Windows. Although the CLI will function without privileged access, many probe types will be unavailable. The `sudo` command can be used to run the CLI as root on Linux and macOS, while the tool is best run from an elevated command shell on Windows. On the Windows platform, the runZero CLI will look for an existing `npcap` installation and try to install it if the software is not found. This behavior can be disabled with the `--nopcap` flag.

> Note
> Some components of the application still reference the name "Rumble" for backwards compatibility. The documentation will be updated as these are changed.

The runZero CLI defaults to a semi-interactive terminal interface that writes multiple output files to a directory. The default directory name is `runzero-[current-date]`. To switch to plain text output, use the `--text` option. To skip artifact generation and only produce the raw JSON output file, use the flags `--text -o disable --output-raw scan.runzero`.

Input can be provided as arguments on the command-line or by specifying an input file using the `--input` (or `-i`) parameter. Input can consist of specific IPv4 addresses or IPv4 CIDRs. Supported formats include:

- `10.0.0.1`
- `10.0.0.0/24`
- `10.0.0.0/255.255.255.0`
- `10.0.0.1-10.0.0.255`
- `example.com`
- `example.com/24`

For hostnames, each IPv4 address in the response will be expanded with the optional mask.

The example below downloads and runs the CLI on a Linux x86_64 host. This URL will be different for your installation. The current download links for your organization are available from the CLI page of the runZero Console. If you are using a self-hosted console or a region other than our US-based SaaS, you can find the download link under the Deploy navigation menu.

```
$ wget https://console.runzero.com/download/cli/[unique-link]/runzero-cli-linux-amd64.bin
$ chmod +x runzero-cli-linux-amd64.bin
$ sudo runzero scan 192.168.0.0/24 -o output-dir
```

Please note that the hexadecimal values in the download URL are specific for your account and organization.

Performance

The default speed of runZero scans is limited to 1,000 packets per second with a single pass. This setting works great for reliable wired networks without stateful firewalls between the scanning system and the destination networks. This rate can be changed via the `--rate` (or `-r`) option, with a reasonable maximum being `10000` for most networks. On slow, unreliable networks, a rate of `300` with `--passes` set to `3` may provide better results.

A second parameter, `--max-host-rate` limits how many packets are sent per second to each individual host. This defaults to `40`, which is low, but may be necessary when scanning low-power embedded devices. In cases where a small number of hosts (or a single host) should be scanned quickly, the `--max-host-rate` parameter can be increased to match the `--rate`.

Examples

The following example demonstrates a scan of 65,535 TCP ports on all hosts in the `192.168.0.0/24` subnet running at 10,000 packets per second:

```
$ sudo runzero scan 192.168.0.0/24 -r 10000 --tcp-ports 1-65535 -o output-dir
```

The following example demonstrates a scan on all hosts in the 192.168.0.0/24 and 10.0.0.0/24 subnets running at 5,000 packets per second:

```
$ sudo runzero scan 192.168.0.0/24 10.0.0.0/24 -r 5000 -o output-dir
```

The following example demonstrates a scan on all hosts in the 192.168.0.0/24 and 10.0.0.0/8 subnets running at a max host rate of 20 packets per host:

```
$ sudo runzero scan 192.168.0.0/24 10.0.0.0/8 --max-host-rate 20 -o output-dir
```

The following example demonstrates a scan on all hosts in the 192.168.0.0/24 subnet and the domain “example.com” running at 7,500 packets per second:

```
$ sudo runzero scan 192.168.0.0/24 example.com -r 7,500 -o output dir
```

The following example demonstrates a scan on all hosts in the 10.0.0.0/8 subnet and a particular ASN4 value at a default speed of 1,000 packets per second.

```
$ sudo runzero scan 10.0.0.0/8 asn4:[ID] -o output dir
```

The following example demonstrates a scan on all hosts in the 192.168.0.0/24 subnet with the max TTL set at 128 and a scan rate of 2,500 packets per second:

```
$ sudo runzero scan 192.168.0.0/24 -r 2,500 --max-ttl 128 -o output-dir
```

The following example demonstrates a scan based on an input file:

```
$ sudo runzero scan -i /path/to/input-file.txt -o output dir
```

Here is an example input file:

```
www.example.com
192.168.0.0/24
```

Automatic web screenshots

The `--screenshots` option defaults to `true` and tells runZero to obtain a screenshot of all web services identified during the scan. This feature depends on the system running the Explorer having a local installation of the Google Chrome or Chromium browsers. The acquired screenshots will be reported as a base64 string, stored in the “screenshot.image” field of the containing service scan result.

To disable automatic web screenshots, set the `--screenshots` option to false (`--screenshots=false`).

Scanner defaults

Standard ports scanned

1	7	9	13	17	19	21	22	23	25	37
42	43	49	53	69	70	79	80	81	82	83
84	85	88	102	105	109	110	111	113	119	123
135	137	139	143	161	179	222	264	280	384	389
402	407	442	443	444	445	465	500	502	512	513
515	523	524	540	541	548	554	587	617	623	631
636	664	689	705	717	743	771	783	830	873	888
902	903	910	912	921	990	993	995	998	1000	1024
1030	1035	1080	1083	1089	1090	1091	1098	1099	1100	1101
1102	1103	1128	1129	1158	1199	1211	1220	1234	1241	1260
1270	1300	1311	1352	1433	1434	1440	1443	1468	1494	1514
1521	1530	1533	1581	1582	1583	1604	1610	1611	1723	1755
1801	1811	1830	1883	1900	2000	2002	2021	2022	2023	2024
2031	2049	2068	2074	2082	2083	2100	2103	2105	2121	2181
2199	2207	2222	2224	2323	2362	2375	2376	2379	2380	2381
2443	2525	2533	2598	2601	2604	2638	2809	2947	2967	3000
3001	3003	3033	3037	3050	3057	3071	3083	3128	3142	3200
3217	3220	3260	3268	3269	3273	3299	3300	3306	3311	3312
3351	3389	3460	3500	3502	3628	3632	3690	3780	3790	3817
3871	3872	3900	4000	4092	4322	4343	4353	4365	4366	4368
4369	4406	4433	4443	4444	4445	4567	4659	4679	4730	4786
4840	4848	4949	4950	4987	5000	5001	5003	5007	5022	5037
5038	5040	5051	5060	5061	5093	5168	5222	5247	5250	5275
5347	5351	5353	5355	5392	5400	5405	5432	5433	5498	5520
5521	5554	5555	5560	5580	5601	5631	5632	5666	5671	5672

5683 5800 5814 5900 5901 5902 5903 5904 5905 5906 5907 5908
5909 5910 5911 5920 5938 5984 5985 5986 5988 5989 6000 6001
6002 6050 6060 6070 6080 6082 6101 6106 6112 6161 6262 6379
6405 6443 6481 6502 6503 6504 6514 6542 6556 6660 6661 6667
6905 6988 7000 7001 7002 7021 7070 7071 7077 7080 7100 7144
7181 7210 7373 7443 7444 7474 7510 7547 7579 7580 7676 7700
7770 7777 7778 7787 7800 7801 7860 7879 7902 8000 8001 8003
8006 8008 8009 8010 8012 8014 8020 8023 8028 8030 8080 8081
8082 8083 8086 8087 8088 8089 8090 8095 8098 8099 8100 8123
8127 8161 8172 8180 8181 8182 8205 8222 8300 8303 8333 8400
8443 8444 8445 8471 8488 8500 8503 8530 8531 8545 8649 8686
8787 8800 8812 8834 8850 8871 8880 8883 8888 8889 8890 8899
8901 8902 8903 8983 9000 9001 9002 9042 9060 9080 9081 9084
9090 9091 9092 9099 9100 9111 9152 9160 9200 9300 9380 9390
9391 9401 9418 9440 9443 9471 9495 9524 9527 9530 9593 9594
9595 9600 9809 9855 9999 10000 10001 10008 10050 10051 10080 10098
10162 10202 10203 10250 10255 10257 10259 10443 10616 10628 11000 11099
11211 11234 11333 12174 12203 12221 12345 12379 12397 12401 13364 13500
13778 13838 14330 15200 15671 15672 16102 16443 16992 16993 17185 17200
17472 17775 17776 17777 17778 17781 17782 17783 17784 17790 17791 17798
18264 18881 19300 19810 19888 20000 20010 20031 20034 20101 20111 20171
20222 20293 22222 23472 23791 23943 24442 25000 25025 25565 25672 26000
26122 27000 27017 27018 27019 27080 27888 28017 28222 28784 29418 30000
31001 31099 32764 32844 32913 33060 34205 34443 34962 34963 34964 37718
37777 37890 37891 37892 38008 38010 38080 38102 38292 40007 40317 41025
41080 41523 41524 44334 44343 44818 45230 46823 46824 47001 47002 47290
48899 49152 50000 50013 50021 50051 50070 50090 50121 51443 52302 52311
53282 54321 54921 54922 54923 55553 55580 57772 61614 61616 62078 62514
65002 65535

Scan outputs

The runZero CLI generates a directory of output files by default. This directory includes the following items.

- `scan.runzero.gz`: The raw scan data compressed via gzip, this can be imported or reprocessed via `--import`
- `assets.jsonl`: The new optimized format for correlated, fingerprinted assets.
- `nmap.xml`: A Nmap XML compatible data file that can be imported into various security tools.
- `urls.txt`: A list of discovered web services in URL format.
- `protocols.csv`: A list of protocols with their ports and URLs.
- `assets.html`: A rudimentary HTML report with screenshots.
- `screenshots`: A directory of raw screenshot images, headers in JSON format, and HTML bodies.
- Various lists including `addresses.txt`, `addresses_all.txt`, `hostnames.txt`, and `domains.txt`

Raw Scan Data

The runZero CLI raw data is stored in a file named `scan.runzero.gz` within the output directory. This file contains JSONL-formatted records. An example ARP response record is shown below.

```
{
  "type": "result",
  "host": "192.168.0.1",
  "port": "0",
  "proto": "arp",
  "probe": "arp",
  "name": "192.168.0.1",
  "info": {
    "mac": "f0:9f:c2:11:1a:13",
    "macDateAdded": "2014-12-17",
    "macVendor": "Ubiquiti Networks Inc."
  },
  "ts": 1551584126253853200
}
```

The `info` field is a JSON map of strings to strings. Multiple values are encoded using the `tab` character (0x09), which are otherwise escaped as `\t` (along with `\r` and `\n` for carriage return and line feed bytes and `\x00` for null bytes). runZero scans may return more than one record of the same type for the same host if multiple responses were received.

In addition to the `result` type, there are also records for `status` messages, `stats`, and an initial `config` type that contains the scan parameters.

runZero Command Line Interface (CLI)

The runZero CLI supports a wide range of commands and options. As well as offline scans, you can run third-party integrations and custom scripts on the command line.

The `--help` output provides basic documentation on the available options.

Most commands below accept the following global flags:

`--verbose`
: Display verbose output.

`--very-verbose`
: Display very verbose output.

Import Censys data files

```
runZero censys [avro files] [targets] [flags]
```

Flags:

`-i, --input-targets string`
: Read search targets from the specified input file

`--output-raw string`
: Write results to the specified output file

Import Censys data from a local database

```
runZero censys-db [path-to-database] [targets] [flags]
```

Flags:

`-i, --input-targets string`
: Read search targets from the specified input file

`--output-raw string`
: Write results to the specified output file

Convert Censys Avro files into a local database

```
runZero censys-db-convert [path-to-avro-directory] [path-to-db-directory] [flags]
```

Flags:

`--shard-max-records int`
: Specify the maximum record count per shard (default 500000)

`--test-mode`
: Replace keys in the source data with random values

`--test-multiplier int`
: Multiply the source data by a given factor (default 1)

Serves a Censys database from a web server

```
runZero censys-db-server [path-to-database] [flags]
```

Flags:

`--port int`
: The TCP port for the web server to listen on (default 55555)

Generate the autocompletion script for the specified shell

```
runZero completion [command]
```

Available commands:

`bash`: Generate the autocompletion script for bash

`fish`: Generate the autocompletion script for fish

`powershell`: Generate the autocompletion script for powershell

`zsh`: Generate the autocompletion script for zsh

Help about any command

```
runZero help [command] [flags]
```

Display license information

```
runZero license [flags]
```

Start a runZero active scan, passive discovery, or integration task

```
runZero scan <options> [targets] [flags]
```

Flags:

`--api-key string`
: Specify the runZero API key

`--api-no-verify`
: Disable TLS verification for API communication

`--api-url string`
: Specify the runZero API server hostname (default "https://console.runzero.com/api/v1.0")

`--arp-fast`
: Enables fast mode by ARP scanning at the scan rate vs host rate

`--atg-ports string`
: The destination ports for ATG probes (default "10001")

`--aws-instances-access-key string`
: The access key for the AWS account

`--aws-instances-assume-role-name string`
: The role to assume for all accounts in the organization for cross-account access

`--aws-instances-delete-stale`
: Automatically delete stale AWS assets

`--aws-instances-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--aws-instances-include-stopped`
: Include assets that are not currently running

`--aws-instances-regions string`
: The comma-separated list of regions for the AWS account

`--aws-instances-secret-access-key string`
: The secret access key for the AWS account

`--aws-instances-service-options string`
: The comma-separated list of services to sync data from (defaults,ec2,elb,elbv2,rds,lambda) (default “defaults”)

`--aws-instances-site-per-account`
: Automatically create a new site per account

`--aws-instances-site-per-vpc`
: Automatically create a new site per VPC

`--aws-instances-token string`
: The session token for the AWS account

`--azure-client-id string`
: The application ID (client ID) for the Azure account

`--azure-client-secret string`
: The client secret for the Azure account

`--azure-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--azure-multi-subscription`
: Access all subscriptions in the directory (tenant) for the Azure account

`--azure-password string`
: The password for the Azure account

`--azure-service-options string`
: The comma-separated list of services to sync data from (defaults,vm,vmss,azsql,cosmos,lb,functionapp) (default “defaults”)

`--azure-site-per-subscription`
: Automatically create a new site per subscription

`--azure-subscription-id string`
: The subscription ID for the Azure account

`--azure-tenant-id string`
: The directory ID (tenant ID) for the Azure account

`--azure-username string`
: The username for the Azure account

`--azuread-client-id string`
: The application ID (client ID) for the Azure account

`--azuread-client-secret string`
: The client secret for the Azure account

`--azuread-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--azuread-filter string`
: An optional filter. Only import devices that match this filter.

`--azuread-include-inactive`
: Include assets that are marked as inactive in the AzureAD account

`--azuread-password string`
: The password for the AzureAD account

`--azuread-service-options string`
: The comma-separated list of services to sync data from (defaults,dev,user,group) (default “defaults”)

`--azuread-tenant-id string`
: The directory ID (tenant ID) for the Azure account

`--azuread-username string`
: The username for the AzureAD account

`--bacnet-ports string`
: The destination ports for BACnet probes (default “46808,47808,48808”)

`-b, --baseline string`
: Use the specified file as an asset baseline for tracking

`--bedrock-ports string`
: The destination ports for Bedrock probes (default “19132”)

`--bjnp-printer-ports string`
: The UDP ports to send Canon printer discovery requests (default “8611”)

`--bjnp-scanner-ports string`
: The UDP ports to send Canon scanner discovery requests (default “8612”)

`--censys-api-url string`
: The API endpoint to use for Censys Search (default “https://search.censys.io”)

`--censys-client-id string`
: The Client ID to use for Censys Search authentication

`--censys-client-secret string`
: The Client Secret to use for Censys Search authentication

`--censys-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--censys-mode string`
: The search mode (assets or query). The assets option queries the scan targets (default “assets”)

`--censys-query string`
: The search string to use in query mode

`--coap-port uint`
: The destination port for CoAP probes (default 5683)

`-c, --config string`
: Specify the config file name to load. It must be in a JSON format

`--cpu string`
: Write a cpu profile after the scan completes

`--crestron-port uint`
: The destination port for Crestron probes (default 41794)

`--crowdstrike-api-url string`
: The URL used for the CrowdStrike account's API access

`--crowdstrike-client-id string`
: The client ID for the CrowdStrike account

`--crowdstrike-client-secret string`
: The client secret for the CrowdStrike account

`--crowdstrike-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--crowdstrike-filter string`
: An optional Falcon Query Language (FQL) filter for imported assets

`--crowdstrike-fingerprint-only`
: Import vulnerabilites for fingerprinting purposes only

`--crowdstrike-risks string`
: Minimum risk of imported vulnerabilities (None, Low, Medium, High, Critical) (default "None,Low,Medium,High,Critical")

`--crowdstrike-severities string`
: Severity levels of imported vulnerabilities (Info, Low, Medium, High, Critical) (default "Info,Low,Medium,High,Critical")

`--custom-integration-entry-function-name string`
: Function to call that will return ImportAssets (default "main")

`--custom-integration-id string`
: UUID of custom integration from console

`--custom-integration-script-args string`
: Arguments for the script

`--custom-integration-script-kwargs string`
: Keyword arguments for the script

`--custom-integration-script-source string`
: Source code of the custom integration script

`--dahua-dhip-ports string`
: The destination ports for Dahua DHIP discovery probes (default "37810")

`--defender365-client-id string`
: The application ID (client ID) for the Azure account

`--defender365-client-secret string`
: The client secret for the Azure account

`--defender365-exclude-notonboarded`
: Exclude assets that have not been fully onboarded

`--defender365-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--defender365-filter string`
: Exclude assets using a Graph API $filter string

`--defender365-include-inactive`
: Include assets that have stopped reporting to the Microsoft 365 Defender service

`--defender365-tenant-id string`
: The directory ID (tenant ID) for the Azure account

`--disabled-probes string`
: Specifically exclude these probes, comma-delimited

`--dnp3-address-probe-timeout int`
: Time limit (in seconds) for DNP address discovery. (default 30)

`--dnp3-banner-address-discovery string`
: One of ‘require’, ‘prefer’, or ‘ignore’. (default “ignore”)

`--dnp3-destination-address-discovery-range string`
: A numeric range of addresses to attempt to discover. (default “0-32”)

`--dnp3-explorer-address int`
: Source DNP3 address for the explorer. (default -1)

`--dns-disable-google-myaddr`
: Disables resolution of upstream DNS via Google myaddr service

`--dns-disable-meraki-detection`
: Disables detection of Meraki DNS interception

`--dns-port uint`
: The destination port for DNS probes (default 53)

`--dns-resolve-name string`
: The target hostname for DNS queries (‘off’ to disable) (default “www.google.com”)

`--dns-trace-domain string`
: The subdomain to use for trace requests (‘off’ to disable) (default “helper.rumble.network”)

`--dtls-ports string`
: The destination ports for DTLS probes (default “443,3391,4433,5246,5349,5684”)

`--echo-report-errors`
: Report errors from intermediate in-scope hosts

`--ethernetip-cip-enumeration-method string`
: (BETA) Set this to the preferred CIP enumeration method. (default “none”)

`--ethernetip-udp-ports string`
: The destination ports for EtherNet/IP UDP probes (default “44818”)

`--exclude string`
: Specify scan exclusions

`--excludefile string`
: Read exclusions from an input file

`--filter-base64`
: Filter base64-encoded fields

`-f, --fingerprints string`
: Use the specified directory as an alternate fingerprint database

`--fingerprints-debug`
: Enable debug output for the fingerprint processor

`--fins-port uint`
: The destination port for FINS probes (default 9600)

`--gcp-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--gcp-key-path string`
: Path to GCP service account key file

`--gcp-service-options string`
: The comma-separated list of services to sync data from (defaults,vm,lb,cloudsql) (default “defaults”)

`--gcp-site-per-project`
: Automatically create a new site per project

`--genudp-payload-base64 string`
: The generic udp payload as base64

`--genudp-payload-hex string`
: The generic udp payload as hexadecimal

`--genudp-payload-text string`
: The generic udp payload as plain text

`--genudp-ports string`
: The destination ports for the generic udp probe

`--googleworkspace-client-email string`
: The email address of the service account

`--googleworkspace-client-id string`
: The ID of the service account

`--googleworkspace-customer-id string`
: An optional customer ID for multi-tenant environments (default “my_customer”)

`--googleworkspace-delegate string`
: The email address of an admin account with directory access

`--googleworkspace-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--googleworkspace-private-key string`
: The PEM encoded private key

`--googleworkspace-private-key-id string`
: The ID of the private key

`--googleworkspace-project-id string`
: The project ID of the service account

`--googleworkspace-service-options string`
: The comma-separated list of services to sync data from (defaults,chromeos,mobile,endpoint,user,group) (default “defaults”)

`--goroutines string`
: Write a goroutine dump after the scan completes

`--heap string`
: Write a heap profile after the scan completes

`--hiddiscoveryd-port uint`
: The destination port for HID discoveryd probes (default 4070)

`--host-ping`
: Only scan hosts that respond to a ping scan using the host-ping settings

`--host-ping-max-attempts int`
: Set the maximum number of attempts for each probe (default 2)

`--host-ping-max-ttl int`
: Set the default TTL on host-ping probe packets (default 255)

`--host-ping-passes int`
: Set the number of passes for the host-ping phase (default 1)

`--host-ping-probes string`
: Launch a subset of the probes for the host-ping, comma-delimited (default “arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns”)

`--host-ping-tcp-ports string`
: The list of TCP ports to host-ping using the syn and connect probes (default “22,80,135,179,443,3389,5040,7547,62078”)

`--host-ping-tos int`
: Set the default ToS on host-ping probe packets

`--igel-discovery-ports string`
: The destination ports for IGEL discovery probes (default “30005”)

`--ike-port uint`
: The destination port for IKE probes (default 500)

`-I, --import stringArray`
: Import existing scan data from the specified input files (‘scan.rumble’ format)

`--import-pcap stringArray`
: Import pcap packet capture from the specified input files (‘.pcap’ or ‘.pcapng’ format)

`-i, --input-targets string`
: Read scan targets from the specified input file

`--insightvm-api-url string`
: The URL used for the InsightVM account’s API access

`--insightvm-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--insightvm-fingerprint-only`
: Import vulnerabilites for fingerprinting purposes only

`--insightvm-insecure`
: Set this to true to authenticate to untrusted endpoints (self-signed or no IP SAN) (default true)

`--insightvm-password string`
: The password for the InsightVM account

`--insightvm-risks string`
: Risk levels of imported vulnerabilities (None, Low, Medium, High, Critical) (default “None,Low,Medium,High,Critical”)

`--insightvm-severities string`
: Severity levels of imported vulnerabilities (Info, Low, Medium, High, Critical) (default “Info,Low,Medium,High,Critical”)

`--insightvm-thumbprints string`
: A set of IP=SHA256:B64HASH pairs to trust for authentication

`--insightvm-username string`
: The username for the InsightVM account

`--intune-client-id string`
: The application ID (client ID) for the Azure account

`--intune-client-secret string`
: The client secret for the Azure account

`--intune-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--intune-filter string`
: An optional filter. Only import devices that match this filter.

`--intune-password string`
: The password for the Intune account

`--intune-tenant-id string`
: The directory ID (tenant ID) for the Azure account

`--intune-username string`
: The username for the Intune account

`--ipmi-port uint`
: The destination port for IPMI probes (default 623)

`--ipp-browse-port uint`
: The destination port for IPP-browse probes (default 631)

`--iscsi-discover`
: Enable iSCSI target discovery probe (default true)

`--kerberos-port uint`
: The destination port for kerberos probes (default 88)

`--knxnet-ports string`
: The destination ports for knxnet probes (default “3671”)

`--l2t-port uint`
: The destination port for L2T probes (default 2228)

`--l2tp-ports string`
: The destination ports for L2TP probes (default “1701”)

`--lantronix-port uint`
: The destination port for Lantronix probes (default 30718)

`--layer2-add-targets`
: Set this false to skip scanning discovered targets (default true)

`--layer2-force`
: Set this to true to force discovery even without local targets

`--layer2-max-retries uint`
: The desired number of retries (default 3)

`--layer2-tcp-ports string`
: The TCP ports to ping for local device discovery (default “22,80,135,179,443,3389,5040,7547,62078”)

`--layer2-udp-trace-port uint`
: The UDP port number to use for UDP trace requests (default 9)

`--ldap-base-dn string`
: The base DN used for LDAP searches

`--ldap-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--ldap-insecure`
: Set this to true to authenticate to untrusted endpoints (self-signed or no IP SAN)

`--ldap-legacy-tls`
: Set this to true to authenticate over legacy TLS versions (< 1.2)

`--ldap-password string`
: The password for the LDAP account

`--ldap-service-options string`
: The comma-separated list of services to sync data from (defaults,computer,user,group) (default “defaults”)

`--ldap-thumbprints string`
: A set of IP=SHA256:B64HASH pairs to trust for authentication

`--ldap-url string`
: The URL used for the LDAP server

`--ldap-username string`
: The username for the LDAP account

`--max-attempts int`
: Set the maximum number of attempts for each probe (default 3)

`-G, --max-group-size int`
: Set the maximum number of targets to process in each group (default 4096)

`-R, --max-host-rate int`
: Set the maximum packet rate per target (including ARP broadcast) (default 40)

`--max-scan-duration int`
: Set the maximum scan duration in seconds before aborting

`--max-sockets int`
: Set the maximum number of concurrent sockets (default 2048)

`--max-ttl int`
: Set the default TTL on probe packets (default 255)

`--mdns-port uint`
: The destination port for MDNS probes (default 5353)

`--mecm-database-connection-string string`
: The connection string for your MECM Microsoft SQL Server database

`--mecm-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--memcache-port uint`
: The destination port for memcached probes (default 11211)

`--meraki-api-key string`
: The access key for the Meraki.io account

`--meraki-api-url string`
: The URL used for the Meraki.io account's API access (default "https://api.meraki.com/api/v1")

`--meraki-exclude-no-vlan-clients`

`--meraki-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--meraki-excluded-ssids string`

`--meraki-excluded-vlans string`

`--meraki-networks string`
: An optional list of network names or IDs. Only import devices in the specified networks.

`--meraki-organizations string`
: An optional list of organization names or IDs. Only import devices in the specified organizations.

`--miradore-api-key string`
: The API key for the Miradore account

`--miradore-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--miradore-hostname string`
: The Miradore web console hostname (url)

`--modbus-identification-level string`
: Identification level, one of ‘basic’, ‘regular’, or ’extended’. (default “regular”)

`--mssql-port uint`
: The destination port for MSSQL Browser probes (default 1434)

`--nameservers string`
: One or more nameservers to use for DNS resolution

`--natpmp-port uint`
: The destination port for NATPMP probes (default 5351)

`--nessus-access-key string`
: The access key for the Nessus Professional account

`--nessus-api-url string`
: The URL used for the Nessus Professional account’s API access

`--nessus-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--nessus-fingerprint-only`
: Import vulnerabilites for fingerprinting purposes only

`--nessus-insecure`
: Set this to true to authenticate to untrusted endpoints (self-signed or no IP SAN) (default true)

`--nessus-risks string`
: Risk levels of imported vulnerabilities (None, Low, Medium, High, Critical) (default “None,Low,Medium,High,Critical”)

`--nessus-secret-key string`
: The secret key for the Nessus Professional account

`--nessus-severities string`
: Severity levels of imported vulnerabilities (Info, Low, Medium, High, Critical) (default “Info,Low,Medium,High,Critical”)

`--nessus-thumbprints string`
: A set of IP=SHA256:B64HASH pairs to trust for authentication

`--netbios-port uint`
: The destination port for NetBIOS Name Service probes (default 137)

`--netbox-api-key string`
: The access key for the NetBox instancet

`--netbox-api-url string`
: The URL to the NetBox instance

`--netbox-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--netbox-include-no-ip`
: Include NetBox assets that have no associated IP addresses

`--nowait`
: Exit the user interface immediately upon completion

`--ntp-port uint`
: The destination port for NTP probes (default 123)

`--openvpn-ports string`
: The destination ports for OpenVPN probes (default “1194”)

`--oracledb-fingerprint`
: Enable Oracle DB version fingerprinting using a TNS connect sequence (default true)

`-o, --output string`
: Output directory for scan results and analysis (‘disable’ to skip)

`--output-raw string`
: Set the raw output file for scan data

`--overwrite`
: Overwrite and replace the output directory if it already exists

`--passes int`
: Set the number of passes for each probe (default 1)

`--pca-port uint`
: The destination port for PCAnywhere probes (default 5632)

`--pcworx-ports string`
: The destination ports for PCWORX probes (default “1962”)

`--probes string`
: Launch a subset of the probes, comma-delimited (default “defaults”)

`--psdisco-ports string`
: The destination ports for playstation discovery probes (default “987,9302”)

`--qualys-api-url string`
: The URL used for the Qualys account’s API access

`--qualys-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--qualys-fingerprint-only`
: Import vulnerabilites for fingerprinting purposes only

`--qualys-include-unscanned`
: Include assets that have not been assessed for vulnerabilities

`--qualys-password string`
: The password for the Qualys account

`--qualys-risks string`
: Risk levels of imported vulnerabilities (None, Low, Medium, High, Critical) (default “None,Low,Medium,High,Critical”)

`--qualys-severities string`
: Severity levels of imported vulnerabilities (Info, Low, Medium, High, Critical) (default “Info,Low,Medium,High,Critical”)

`--qualys-tags string`
: An optional list of tags. Only import devices that match any of the specified tags

`--qualys-network-ids string`
: An optional list of network IDs. Only import devices that match any of the specified network IDs

`--qualys-username string`
: The username for the Qualys account

`-r, --rate int`
: Set the maximum packet rate for the overall scan (default 1000)

`--rdns-max-concurrent int`
: The maximum number of concurrent DNS lookups (default 64)

`--rdns-timeout uint`
: The DNS PTR lookup timeout in seconds (default 3)

`--rpcbind-port uint`
: The destination port for RPCBind probes (default 111)

`--rpcbind-port-nfs uint`
: The destination port for NFS probes (default 2049)

`--s7comm-request-extended-information`
: If true, request extended device information.

`--sadp-ports string`
: The destination ports for Hikvision SADP discovery probes (default “37020”)

`--sample-duration string`
: Specify the duration in seconds to sample network traffic (or ‘0’ for non-stop) (default “300”)

`--sample-excludes string`
: Specify host exclusions

`--sample-interfaces string`
: Specify a comma-separated list of network interfaces (or ‘all’ for everything)

`--sample-targets string`
: Specify the discovery scope (default “10.0.0.0/8 172.16.0.0/12 192.168.0.0/16 169.254.0.0/16”)

`-S, --screenshots`
: Capture screenshots from scan target web services (default true)

`--sentinelone-api-url string`
: The URL used for the SentinelOne account’s API access

`--sentinelone-api-key string`
: The API key for the SentinelOne accounts API access

`--sentinelone-exclude-software-import`
: Do not import software records from the SentinelOne account

`--sentinelone-exclude-vulnerability-import`
: Do not import vulnerability records from the SentinelOne account

`--sentinelone-severities string`
: Severity levels of imported vulnerabilities (False Positive, LOW, MEDIUM, HIGH, CRITICAL) (default “False Positive,LOW,MEDIUM,HIGH,CRITICAL”)

`--sentinelone-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--servicetag-port uint`
: The destination port for Solaris Service Tag probes (default 6481)

`--shodan-api-key string`
: The key used for the Shodan account's API access

`--shodan-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--shodan-mode string`
: The search mode (assets or query). The assets option queries the scan targets (default "assets")

`--shodan-query string`
: The search string to use in query mode

`--sip-port uint`
: The destination port for SIP probes (default 5060)

`--snmp-comms string`
: The comma-separated list of SNMP v1/v2c communities (default "public,private")

`--snmp-disable-bulk`
: If true, do not use bulk walking operations

`--snmp-max-repetitions uint`
: The maximum number of repetitions in a bulk walk operation (default 16)

`--snmp-max-retries int`
: The maximum number of retries for an SNMP operation (default 1)

`--snmp-poll-interval uint`
: The minimum number of seconds between polling each host after initial discovery (default 300)

`--snmp-port uint`
: The destination port for SNMP probes (default 161)

`--snmp-timeout uint`
: The maximum number of seconds for each individual SNMP operation (default 5)

`--snmp-v3-auth-passphrase string`
: The authentication passphrase

`--snmp-v3-auth-protocol string`
: The authentication protocol (none, md5, sha, sha224, sha256, sha384, sha512) (default "none")

`--snmp-v3-context string`
: The optional SNMP v3 context to supply

`--snmp-v3-privacy-passphrase string`
: The privacy passphrase

`--snmp-v3-privacy-protocol string`
: The privacy protocol (none, des, aes, aes192, aes256, aes192c, aes256c) (default "none")

`--snmp-v3-username string`
: The username to use for SNMP v3 authentication

`--snmp-walk-timeout uint`
: The maximum number of seconds for each SNMP walk operation (default 60)

`--ssdp-port uint`
: The destination port for UPnP/SSDP probes (default 1900)

`--ssh-fingerprint`
: Enable fingerprinting using partial authentication (default true)

`--ssh-fingerprint-username string`
: The username to use for partial authentication SSH fingerprinting (default "*STATUS*")

`--steam-ports string`
: The destination ports for Steam discovery probes (default "27036")

`--subnet-ping`
: Only scan subnets that have at least one active response using the subnet-ping settings

`--subnet-ping-max-attempts int`
: Set the maximum number of attempts for each probe (default 1)

`--subnet-ping-max-ttl int`
: Set the default TTL on subnet-ping probe packets (default 255)

`--subnet-ping-mode string`
: Set the subnet-ping discovery profile: auto (default "auto")

`--subnet-ping-net-size int`
: Set the subnet size to use for the subnet ping (default 256)

`--subnet-ping-passes int`
: Set the number of passes for the subnet-ping phase (default 1)

`--subnet-ping-probes string`
: Launch a subset of the probes for the subnet-ping, comma-delimited (default "arp,echo,syn,connect,netbios,snmp,ntp,sunrpc,ike,openvpn,mdns")

`--subnet-ping-sample-rate int`
: Set the sample rate of addresses within each subnet as a percentage (default 4)

`--subnet-ping-tcp-ports string`
: The list of TCP ports to subnet-ping using the syn and connect probes (default "22,80,135,179,443,3389,5040,7547,62078")

`--subnet-ping-tos int`
: Set the default ToS on subnet-ping probe packets

`--syn-disable-bogus-filter`
: Disable bogus service detection and filtering

`--syn-forwarding-check`
: Perform an IP forwarding check as part of the scan (default true)

`--syn-forwarding-check-target string`
: An external IPv4 address for the forwarding check (default:runzero) (default "13.248.161.247")

`--syn-max-retries uint`
: The maximum number of retries trace and SYN requests (default 2)

`--syn-report-resets`
: Set this to true to report RST responses (default true)

`--syn-reset-sessions`
: Reset middle-box/firewall sessions automatically (default true)

`--syn-reset-sessions-delay uint`
: Minimum delay in milliseconds between a SYN and a session reset

`--syn-reset-sessions-limit uint`
: Maximum number of in-flight sessions before forcing session resets (default 50)

`--syn-traceroute`
: Perform a multi-protocol traceroute as part of the scan (default true)

`--syn-udp-trace-port uint`
: The UDP port number to use for UDP trace requests (default 9)

`--tanium-api-token string`
: The API token for the Tanium account

`--tanium-api-url string`
: The URL used for the Tanium account's API access

`--tanium-computer-groups string`
: Filter endpoints to members of the specified computer groups

`--tanium-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--tanium-severities string`
: Severity levels of imported vulnerabilities (Low, Medium, High, Critical) (default "Low,Medium,High,Critical")

`--tcp-excludes string`
: The list of TCP ports to always exclude

`-p, --tcp-ports string`
: The list of TCP ports scan using the syn and connect probes (see below for default)

`--tcp-skip-protocol`
: Set this to skip protocol detection on TCP ports

`--tenable-access-key string`
: The access key for the Tenable.io account

`--tenable-api-url string`
: The URL used for the Tenable.io account's API access

`--tenable-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--tenable-fingerprint-only`
: Import vulnerabilites for fingerprinting purposes only

`--tenable-include-unscanned`
: Include assets that have not been assessed for vulnerabilities

`--tenable-risks string`
: Risk levels of imported vulnerabilities (None, Low, Medium, High, Critical) (default “None,Low,Medium,High,Critical”)

`--tenable-secret-key string`
: The secret key for the Tenable.io account

`--tenable-severities string`
: Severity levels of imported vulnerabilities (Info, Low, Medium, High, Critical) (default “Info,Low,Medium,High,Critical”)

`--tenable-sources string`
: An optional list of sources. Only import devices that match the specified sources

`--tenable-tags string`
: An optional list of tags. Only import devices that match the specified tags

`--tenablesecuritycenter-access-key string`
: The access key for the Tenable Security Center account

`--tenablesecuritycenter-api-url string`
: The URL used for the Tenable Security Center account’s API access

`--tenablesecuritycenter-batch-size string`
: The number of records to request at a time. (between 2000 and 10000) (default “2000”)

`--tenablesecuritycenter-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--tenablesecuritycenter-fingerprint-only`
: Import vulnerabilites for fingerprinting purposes only

`--tenablesecuritycenter-insecure string`
: Set this to true to authenticate to untrusted endpoints (self-signed or no IP SAN)

`--tenablesecuritycenter-query-id string`
: The ID of an existing vulnerability query in the Tenable Security Center account

`--tenablesecuritycenter-query-mode string`
: Set to ‘filters’ to provide ‘severities’ and ‘risks’ values to import. Set to ‘query-id’ to provide a value for ‘query-id’. (default “filters”)

`--tenablesecuritycenter-risks string`
: Risk levels of imported vulnerabilities (None, Low, Medium, High, Critical) (default “None,Low,Medium,High,Critical”)

`--tenablesecuritycenter-secret-key string`
: The secret key for the Tenable Security Center account

`--tenablesecuritycenter-severities string`
: Severity levels of imported vulnerabilities (Info, Low, Medium, High, Critical) (default “Info,Low,Medium,High,Critical”)

`--tenablesecuritycenter-sync-since string`
: Specify an initial date to sync data from.

`--tenablesecuritycenter-thumbprints string`
: A set of IP=SHA256:B64HASH pairs to trust for authentication

`--text`
: Force text-only mode (no console ui)

`--tftp-ports string`
: The destination ports for TFTP probes (default “69”)

`--tos int`
: Set the default ToS on probe packets

`--ubnt-port uint`
: The destination port for Ubiquiti probes (default 10001)

`--upload`
: Automatically upload results to the runZero Console

`-u, --upload-site string`
: Specify the Site ID or Name to upload the raw scan results to if –upload is specified (default “Primary”)

`--vmware-insecure`
: Set this to true to authenticate to untrusted endpoints (self-signed or no IP SAN) (default true)

`--vmware-password string`
: The password to use for VMware SDK authentication (read-only)

`--vmware-thumbprints string`
: A set of IP=SHA256:B64HASH pairs to trust for authentication

`--vmware-username string`
: The username to use for VMware SDK authentication (read-only)

`--webmin-ports string`
: The destination ports for webmin probes (default “10000”)

`--wiz-api-url string`
: The URL used for the Wiz account’s API access

`--wiz-auth-url string`
: The URL used for the Wiz account’s authentication

`--wiz-client-id string`
: The client ID for the Wiz account

`--wiz-client-secret string`
: The client secret for the Wiz account

`--wiz-exclude-unknown`
: Exclude assets that cannot be merged into an existing asset

`--wiz-fingerprint-only`
: Import vulnerabilites for fingerprinting purposes only

`--wiz-include-unscanned`
: Include assets that have not been assessed for vulnerabilities (default true)

`--wiz-risks string`
: Risk levels of imported vulnerabilities (None, Low, Medium, High, Critical) (default “None,Low,Medium,High,Critical”)

`--wiz-severities string`
: Severity levels of imported vulnerabilities (Info, Low, Medium, High, Critical) (default “Info,Low,Medium,High,Critical”)

`--wlan-list-poll-interval uint`
: The minimum number of seconds between polls of the access point list (default 300)

`--wsd-port uint`
: The destination port for WSD probes (default 3702)

`--xdmcp-ports string`
: The destination ports for XDMCP probes (default “177”)

Run a custom starlark script

```
runZero script --filename filename [--args a] [--args b] [--kwargs a=b] [--kwargs b=c] [flags]

runZero script [command]
```

Available commands:

`repl`: Run a custom starlark script in a REPL

Flags:

`--args stringArray`
: args to pass into script

`-f, --filename string`
: file of script to load and run

`--kwargs stringToString`
: kwargs to pass into script (default [])

Upgrade to the latest version of the runZero Scanner

```
runZero upgrade [flags]
```

Flags:

`--force`
: Apply the update without checking the version

Perform an internal signature verification

```
runZero verify [flags]
```

Print the version number of runZero

```
runZero version [flags]
```

Glossary

As you read through the documentation, you will see commonly used terms. These are the definitions for those terms, so you can familiarize yourself with them ahead of time.

Terms

Alerts
: Alerts are triggered when a certain events occurs based on rules defined in the Rules Engine.

Analysis reports
: Analysis reports are reports which run as tasks, rather than being generated on-the-fly. These reports are static, so any changes to your inventory may result in assets no longer being accessible from the report.

Asset
: An asset is a unique network entity from the perspective of the system running the Explorer.

Automatic queries
: Automatic queries are certain queries you always want to run after a scan. After the query runs, you will be able to view its results in the queries table.

Dashboard
: The dashboard provides trend data and insights that will help you assess how your inventory is changing over time.

Discovery scan
: A discovery scan finds, identifies, and builds an inventory of all the connected devices and assets on your internal network.

Explorer
: The Explorer is a lightweight scan engine that enables network and asset discovery.

Insights
: Insights are queries that run automatically after each scan. They will populate on your dashboard.

Inventory
: The inventory displays all assets within the Organization and can be sorted, filtered,

Organization
: An organization represents a distinct entity; this can be your business, a specific department within your business, or one of your customers.

Outliers
: Outliers show how often different values occur in specific attributes of assets and services.

Queries
: Queries are filters that can be applied to your Inventory to find assets of interest.

Rules
: A rule defines the action that is taken based on a set of conditions. You can create rules to proactively alert your team when there are changes to things like Explorers, assets, scans, organizations, and sites.

Scheduled scans
: Scheduled scans allow you to set a date and frequency for your scan task.

Self-hosted
: The self-hosted version runZero allows you to run the entire platform on-premises or within your own cloud environment.

Site
: A site represents a distinct network segment, usually defined by addressing or accessibility.

Frequently asked questions

Here you can browse the solutions to some common runZero issues and the answers to some frequently asked questions (FAQs).

For more solutions and FAQs, check out the knowledgebase on the runZero support portal.

Issues and FAQs

Still can't find your answer? Let us know.

Identical assets in inventory

Why are there so many identical assets in my inventory?

Some enterprise routers and firewalls, like Cisco ASA devices, are designed to reply to all unexpected attempts on a particular port with a TCP reset (RST). On top of that, some routers listen to SIP traffic on all addresses and automatically respond to it. runZero tries to automatically detect and avoid most of the SIP helper implementations, but can't always do so without possibly losing real results.

runZero will generally detect when a router or firewall is replying to every connection attempt and avoid creating assets based on those responses. However, if you have a network appliance that runZero doesn't detect is spoofing response, there may be a substantial number of identical assets that will appear in your inventory.

Here are a couple of workarounds if you can't prevent your device from replying to all connections:

- Exclude the ports the device responds to from the scan configuration.
- Exclude all or part of the router's IP address range from the scan.

If you need help deleting unwanted records, please contact our support team.

Scanning routers

How do I run runZero without crashing my router?

If your router is crashing while being scanned, the likely issue is that your router is stateful and it is keeping track of every connection going through it. Since our scanning process involves thousands of attempted connections, your router likely ran out of available stateful sessions. This usually occurs when a router is using Network Address Translation (NAT) or is acting as a stateful security firewall.

If this happens, here's what you can do:

- Avoid scanning across routed networks (wired and WiFi, multiple VLANs, etc) by deploying additional Explorers.
- Reduce the Max group size in your scan configuration. This limits the number of targets runZero can scan at once, which correlates to the number of connections the router sees. Default is 4096.
- Reduce the scan speed. This will give failed connections more time to expire before new ones are attempted.
- If a router can run in bridge mode, and you don't need its NAT features, bridge mode will likely be more reliable. For example, if you have an ISP-provided router connected to a WiFi mesh system, you will likely want to run the mesh system in bridge mode and let the ISP router handle all routing including NAT; you should then be able to scan across your WiFi network without crashing the base stations.

Scanning VMWare virtual machines

How do I scan VMware virtual machines without crashing the host?

runZero can be used to scan VMware virtual machines. However, there are some precautions you should take.

VMnet interfaces normally use Network Address Translation (NAT) to route traffic between the host system and the virtual machines. The VMware software effectively operates as a stateful router. As explained above, this can cause problems when runZero tries to open thousands of connections.

For scanning VMware systems, the best option is to deploy a runZero Explorer inside VMware, on a virtual machine connected to the VMnet you want to scan. That Explorer should be able to scan all VMs on the same VMnet without VMware needing to track all of the connections.

Explorer not capturing screenshots

Why didn’t the runZero Explorer capture screenshots?

The runZero Explorer needs a working install of Google Chrome to obtain screenshots. To check for Google Chrome, the Explorer looks in the following locations on each OS.

Windows

The runZero Explorer looks for Chrome on Windows in:

```
c:\Program Files (x86)\Google\Chrome\Application\chrome.exe
```

The Explorer also checks the following environment variables:

- `ProgramFiles(x86)`
- `ProgramFiles`
- `ProgramW6432`

Each may list another directory, in which case the Explorer looks in `\Google\Chrome\Application\chrome.exe` under each of those directories as well.

To find what the environment variables are set to, open a Windows command prompt and entering the command `set`.

For a default Windows 10 install, the default value of `ProgramFiles` and `ProgramW6432` is `c:\Program Files`, which means the Explorer also checks the following location:

```
c:\Program Files\Google\Chrome\Application\chrome.exe
```

This is the default location for Chrome on a 64 bit Windows 10 system.

MacOS

On macOS, the Explorer checks for Google Chrome in the following locations:

- `/Applications/Google Chrome.app/Contents/MacOS/Google Chrome`
- `/Applications/Google Chrome Canary.app/Contents/MacOS/Google Chrome Canary`
- `/Applications/Chromium.app/Contents/MacOS/Chromium`

Linux

On Linux systems, the Explorer checks for Google Chrome at the following locations:

- `/usr/bin/google-chrome`
- `/usr/bin/google-chrome-beta`
- `/usr/bin/google-chrome-unstable`
- `/usr/local/bin/chrome`
- `/usr/bin/chrome`

- `/opt/google/chrome/google-chrome`
- `/usr/bin/chromium`
- `/usr/bin/chromium-browser`

Chrome is installed, but screenshots still don’t work

If Google Chrome is installed in one of the standard locations, but isn’t being found, it could be a permissions issue. It is also possible for Chrome to fail to run for other reasons, such as a corrupt Chrome profile. The next step to take is to check the Explorer log file which will have more detail about the scan operation.

Explorer security model

runZero Explorers are responsible for running network scans, sampling traffic for passive discovery, and implementing integrations within their deployed environment. This document describes the security model of runZero Explorers.

Development

runZero Explorers are built using the Go programming language. This language provides cross-platform compatibility and memory safety by default. The runZero engineering team updates the Explorer code and it’s dependencies on a continuous basis to ensure that any dependency-level vulnerabilities are resolved quickly. All engineers are required to use MFA to access our Git repository. All code changes go through manual review before being merged to our development branch, deployed to a test environment, reviewed again before being merged to our staging branch, and deployed to a staging environment. During the build process, Explorer binaries are signed with a private Ed25519 key and the signature is stamped into the end of the binary. For Windows binaries, an Authenticode signature is also applied, using an extended-validation certificate stored in a cloud-based HSM. Once all automated tests pass and any changes to behavior are confirmed, the staging environment binaries are tagged for release, and moved to the production deployment location in AWS S3.

Deployment

runZero Explorers are initially provisioned by hand through direct downloads from the runZero Console (whether cloud or self-hosted). During this download process the link token and organization ID is stamped into the binary while still preserving the code signatures. The binary will run on the deployed system, connect back to its stamped Console URL, validate TLS, and register. The registration process starts with the Explorer generating a unique host ID using the operating system’s random source. The ID is stored with limited permissions on the Explorer’s file system (non-Windows) or registry (Windows) and will be used to confirm that Explorer’s identity for future connections.

During the first connection from a new Explorer, the Console records the Explorer-generated Host ID, and establishes a database record for that instance. Other Explorers, even those running on the same system, are not able to claim to be this unique Explorer record without also possessing and verifying the Host ID. New Explorers, by default, do not receive any data from the Console and are not sent any tasks (including those with credentials) unless

manually selected by the user. As a result, its relatively safe to have untrusted Explorers registered, as long as they aren't used for tasks that include sensitive data (typically integrations with credentials).

Note that once an Explorer has the Organization ID, the link token (Download Token) is not needed for the Explorer to connect and register. runZero plans to enforce Explorer use of the current Download Token in the near future, but this is not yet a requirement, as there are few if any security risks exposed by unauthorized Explorer registrations. Once the current Download Token becomes a requirement for registration, older Explorers will not be able to reconnect after the Download Token is rotated. Active Explorers should always receive the latest Download Token as part of the upgrade request.

Commands

The runZero Console provides the ability to send new tasks and specific requests to Explorers. Tasks are limited to active scans, passive traffic sampling, or integrations. Integrations can use custom code, written in the Starlark language, and while they have access to the network, they do not have access the Explorer host machine or file system. Outside of tasks, Explorers accept requests to uninstall, update, and post diagnostics (limited to stack traces and Explorer attributes like operating system and hardware specifications). Explorers do not implement any commands that allow network tunneling or host access to the Explorer's machine.

Credentials are only sent to an Explorer if they are associated with an integration task and if the integration task scope matches the CIDR or other configuration for those credentials. Credentials are never sent to runZero-hosted External Explorers regardless of the task configuration. Credential parameters are sent across in the same TLS connection as the request, but are also encrypted to a unique per-process AGE encryption key. Even if the runZero communication is logged through a TLS proxy and pinned CA, credentials cannot be recovered without also extracting the private, in-process-only, and temporary AGE key.

Updates

The runZero Console verifies that Explorers are running the latest version before scheduling a task (scan, passive, or integration). If the Explorer is out of date, it will be updated to the latest version first, and then the task will be scheduled. If the update takes longer than an internal threshold, or fails due to things like a read-only filesystem on the Explorer side, the task will be scheduled anyways after this threshold is reached.

The update process receives a URL to the new binary. The Explorer connects to this URL, verifies TLS parameters, and downloads the binary to a temporary directory. The Explorer then verifies that the binary has a valid code signature embedded and rejects the update if this verification fails. If the code signature is valid, the Explorer launches the new binary in update mode, which typically replaces the running service with the new executable.

It is possible and supported for the update to change the Organization or even Console for the Explorer service. This feature is used to move Explorers between installations and organizations. In either case, the TLS connection must be valid, and the binary must have a

valid code signature. This verification process also rejects version downgrades by comparing the signed version metadata in the binary against the running version.

Uninstallation

runZero Explorers are automatically uninstalled when their associated runZero Organization is deleted. Explorers can also be uninstalled from the runZero Console or by using the “uninstall” CLI option of the Explorer binary. Explorers try to clean up as many artifacts as possible during the uninstallation process. Any artifacts remaining post-uninstall can be manually removed as needed (common on Windows systems where the OS locks the program files and directory while they run).

Defense in Depth

runZero strives to protect customer data by layering multiple defenses into the runZero architecture:

- An attacker that is able to push code to our internal repository would still need to pass review before it can be merged into our protected staging branch.
- An attacker that is able to compromise our binary distribution point (AWS S3) would still need a valid signing key before any customer updates could be affected. Downgrade attacks are prevented by Explorer-time version comparisons of authenticated and signed metadata.
- An attacker that obtains the download link for a customer can register new Explorers, but is not able to obtain any data from the Console unless those Explorers are specifically chosen for a task.
- An attacker that is able to send commands to an Explorer (by compromising a Console) is limited to a few relatively-safe commands and is not able to force downgrades or arbitrary code execution on the Explorer host.
- An attacker that is able to MiTM the TLS communication with the Console is not able to decrypt credentials used in integration tasks.
- An attacker that is able to schedule a custom integration task with a malicious Starlark script can interact with the network from the perspective of the Explorer, but is not able to interface with the Explorer host.

In all of these situations, runZero also operates extensive logging and alerting, so that any malicious action can be investigated, and the impact of any issue readily assessed.

Protocols scanned by runZero

What protocols does runZero scan for?

runZero supports the following list of protocols:

acpp
ads
amqp
backupexec
bgp
brother-scanner
ceph
chromecast
citrix
common-socket-connection
consul
dahua-dhip
dhcp
docker
drobo-nasd
eero-ebid
epm
etcd2
fins
ganglia
hiddiscoveryd
ics-trace
igel-discovery
infinispan
ipmi
irc
java-object
jdwp
kasa
l2t
lantronix
lockdownd
mdns
mikrotik-bandwidth
mongodb
mqtt
mysql
ndmp
netbios-dgm
nfs
openvpn
pca
postgresql
prosoft
radius
redis
riak-http
rpcbind
s7comm
servicetag
activemq
airplay
arp
bacnet
bitdefender-app
cassandra
chargen
cip
click
companion-link
couchdb
daytime
dnp3
dotnet-remoting
dtls
eerogw
epmd
ethernetip
fortigate-to-fortimanager
giop
http
ident
ike
influxdb
ipp
iscsi
java-rmi
jetdirect
kerberos
l2tp
ldap
lpd
memcache
minecraft
mountd
mssql
mysqlx
neo4j
netbios-ns
ntp
oracledb
pcworx
pptp
psdisco
raritan-csc
rexec
rlogin
rsyncd
sadp
sip
adb
ajp
atg
bedrock
bjnp
cdp
checkmk
ciscosmi
coap
comtrol
crestron
dcerpc
dns
drbd
echo
elasticsearch
erldp
finger
ftp
gpsd
http2
igel
imap
intermapper
ipp-browse
jabber
jdbc-hsqldb
jms
knxnet
landesk
lexmark
matter
meshcop
modbus
mountd
munin
natpmp
netbios
netop-remote-control
opcua
panasonictv
pop3
printerid
qotd
rdp
riak
roomalert
rtsp
securemote
slp

smb1	smb2	smb3
smtp	snmp	socks
sonicwall-sgms	spice	spotify-connect
ssdp	ssh	steam
subversion	sunrpc	syslog
tcpmux	teamviewer	telnet
tftp	thinprint	time
tls	ubnt	upnp
uscan	uscans	vault
vmauthd	vnc	vsdp
waveu	wbsm	webmin
wiznet	wsd	wsman
xdmcp	zabbix-agent	zookeeper

Ports scanned by runZero

What TCP ports does runZero scan?

runZero scans the following TCP ports by default:

1	7	9	13	17	19	21	22	23	25	37
42	43	49	53	69	70	79	80	81	82	83
84	85	88	102	105	109	110	111	113	119	123
135	137	139	143	161	179	222	264	280	384	389
402	407	442	443	444	445	465	500	502	512	513
515	523	524	540	541	548	554	587	617	623	631
636	664	689	705	717	743	771	783	830	873	888
902	903	910	912	921	990	993	995	998	1000	1024
1030	1035	1080	1083	1089	1090	1091	1098	1099	1100	1101
1102	1103	1128	1129	1158	1199	1211	1220	1234	1241	1260
1270	1300	1311	1352	1433	1434	1440	1443	1468	1494	1514
1521	1530	1533	1581	1582	1583	1604	1610	1611	1723	1755
1801	1811	1830	1883	1900	2000	2002	2021	2022	2023	2024
2031	2049	2068	2074	2082	2083	2100	2103	2105	2121	2181
2199	2207	2222	2224	2323	2362	2375	2376	2379	2380	2381
2443	2525	2533	2598	2601	2604	2638	2809	2947	2967	3000
3001	3003	3033	3037	3050	3057	3071	3083	3128	3142	3200
3217	3220	3260	3268	3269	3273	3299	3300	3306	3311	3312
3351	3389	3460	3500	3502	3628	3632	3690	3780	3790	3817
3871	3872	3900	4000	4092	4322	4343	4353	4365	4366	4368
4369	4406	4433	4443	4444	4445	4567	4659	4679	4730	4786
4840	4848	4949	4950	4987	5000	5001	5003	5007	5022	5037
5038	5040	5051	5060	5061	5093	5168	5222	5247	5250	5275
5347	5351	5353	5355	5392	5400	5405	5432	5433	5498	5520
5521	5554	5555	5560	5580	5601	5631	5632	5666	5671	5672
5683	5800	5814	5900	5901	5902	5903	5904	5905	5906	5907
5908	5909	5910	5911	5920	5938	5984	5985	5986	5988	5989
6000	6001	6002	6050	6060	6070	6080	6082	6101	6106	6112
6161	6262	6379	6405	6443	6481	6502	6503	6504	6514	6542

6556	6660	6661	6667	6905	6988	7000	7001	7002	7021	7070	7071
7077	7080	7100	7144	7181	7210	7373	7443	7444	7474	7510	7547
7579	7580	7676	7700	7770	7777	7778	7787	7800	7801	7860	7879
7902	8000	8001	8003	8006	8008	8009	8010	8012	8014	8020	8023
8028	8030	8080	8081	8082	8083	8086	8087	8088	8089	8090	8095
8098	8099	8100	8123	8127	8161	8172	8180	8181	8182	8205	8222
8300	8303	8333	8400	8443	8444	8445	8471	8488	8500	8503	8530
8531	8545	8649	8686	8787	8800	8812	8834	8850	8871	8880	8883
8888	8889	8890	8899	8901	8902	8903	8983	9000	9001	9002	9042
9060	9080	9081	9084	9090	9091	9092	9099	9100	9111	9152	9160
9200	9300	9380	9390	9391	9401	9418	9440	9443	9471	9495	9524
9527	9530	9593	9594	9595	9600	9809	9855	9999	10000	10001	10008
10050	10051	10080	10098	10162	10202	10203	10250	10255	10257	10259	10443
10616	10628	11000	11099	11211	11234	11333	12174	12203	12221	12345	12379
12397	12401	13364	13500	13778	13838	14330	15200	15671	15672	16102	16443
16992	16993	17185	17200	17472	17775	17776	17777	17778	17781	17782	17783
17784	17790	17791	17798	18264	18881	19300	19810	19888	20000	20010	20031
20034	20101	20111	20171	20222	20293	22222	23472	23791	23943	24442	25000
25025	25565	25672	26000	26122	27000	27017	27018	27019	27080	27888	28017
28222	28784	29418	30000	31001	31099	32764	32844	32913	33060	34205	34443
34962	34963	34964	37718	37777	37890	37891	37892	38008	38010	38080	38102
38292	40007	40317	41025	41080	41523	41524	44334	44343	44818	45230	46823
46824	47001	47002	47290	48899	49152	50000	50013	50021	50051	50070	50090
50121	51443	52302	52311	53282	54321	54921	54922	54923	55553	55580	57772
61614	61616	62078	62514	65002	65535						

What UDP ports does runZero scan?

runZero scans the following UDP ports by default:

53	69	88	111	123	137	161	443	500	623	987
1194	1434	1701	1900	2049	2228	3391	3671	3702	4433	5060
5246	5349	5351	5353	5632	5683	5684	9302	10000	10001	11211
19132	30718	37810	41794	46808	47808	48808	65535			

Scanning IoT and OT

Can I safely scan my IoT or OT environments?

Some organizations have IoT or OT equipment sensitive to high traffic rates or malformed packets that may have experienced issues with other scanning tools in the past, resulting in a “don’t scan” rule to be in effect.

runZero is different, and should be able to scan in these environments. runZero provides a lightweight active scan engine called an Explorer that can be deployed almost anywhere. Since the scan is active, there are no tap or span ports that need to be configured, nor device level agents that need to be installed, so you don’t have to modify your environment.

The runZero Explorer was built with sensitive OT environments in mind. It is not based on any other commercial or open source tools such as nmap or masscan. The Explorer only sends normal traffic, nothing malformed that might potentially crash a fragile system. Some of the controls in place also include:

- packets-per-second scan rates with sensible default values:
  - 1000 packets per second for overall maximum scan rate (adjustable; scan traffic is balanced across all hosts in the scan range)
  - 40 packets per second for per-host maximum scan rate (adjustable)
- IP and TCP port exclusions
- UDP service probes can be enabled or disabled individually
- The scan balances SYNs and ACKs and watches for port consumption issues on both the client & target
- Configurable max group size that limits the number of targets runZero can scan at once, which correlates to the number of connections stateful devices such as firewalls or routers receive
- Only those TCP and UDP ports that provide actionable intelligence for fingerprinting a device are checked, not all 65535. This list is adjustable in case specialized equipment runs on a non-standard port (see the Port List).
- Per port / protocol considerations engineered to avoid issues. Ex: Sending characters to port 9100 on a printer could print “garbage”. runZero will collect a banner from some ports such as these but never actively probe them.

Some OT/ICS vendors which runZero can fingerprint upon discovery include:

- Allen-Bradley
- BARIX
- Cisco
- Control Solutions
- Control Techniques
- GE
- GENEREX
- GLC Controls
- Lantronix
- Linor Koda
- Mitsubishi
- Moxa
- PLC
- Pressac
- Rittal
- Rockwell
- Schneider Electric
- Siemens

Many organizations opt to deploy the Explorer to the same system that runs their vulnerability scans since there may already be allow-lists, full network connectivity, and considerations made for session table capacity on any session-aware middle boxes such as firewalls, proxies, or small routers. It may also be advisable to deploy additional Explorers at remote sites to gather additional detail and avoid altogether any need to consider middle boxes.

Browsers supported by the runZero Console

Which browsers are supported when accessing the runZero Console?

We maintain compatibility with the following browser versions or newer:

- *Chrome 110* (released February 6, 2023)
- *Edge 110* (released February 8, 2023)
- *Safari 16.0* (released September 11, 2022)
- *Firefox 115* (released July 3, 2023)
- *Opera 96* (released February 21, 2023)

Internet Explorer is *not* supported.

Mobile browser support is experimental and is not guaranteed to work.

Common sign-in issues

User account has been locked from too many failed sign-in attempts.

Cause: The account you are signing into is configured for password authentication and has been locked due to repeated incorrect sign-in attempts.

Remediation: Contact your runZero administrator and have them follow these steps:

1. In the runZero console, visit the Team page
2. Select the locked account
3. Use the “Unlock user accounts” option in the “Reset” dropdown menu.

If contacting an administrator is not an option, please reach out to support@runzero.com.

Certificate information for single sign-on has expired and must be updated in order to sign in

Cause: The SAML certificate configured for the target runZero tenant has expired.

Remediation:

1. Acquire a new valid certificate from your identity provider (for example, Okta, Google Workspace, etc)
2. Sign into runZero using a superuser account that is configured for link-based or password-based authentication

3. Paste the new certificate contents into the “Certificate” field of the Identity provider settings page

- The “Certificate” field can be found by navigating to the Team page in the product, then clicking the “SSO settings” button in the page header.

If you are the administrator for your account and are unable to sign in at all, please contact support@runzero.com and provide the required certificate for further assistance.

Invalid SAML response

Cause: The data that your identity provider sent to the runZero console was in an invalid format.

Remediation: Review the SAML configuration in your identity provider (for example, Okta, Google Workspace, etc)

The information provided by your identity provider was incomplete or not valid.

Cause: The data that your identity provider sent to the runZero console did not include one or more required assertion elements. Required elements are `Subject` and `NameID`.

Remediation: Review the configuration of your identity provider to ensure it’s configured to send a valid Subject and NameID to the runZero console.

The email address `{email address}` is already in-use by an existing user account.

Cause: The email address provided by the SSO identity provider already exists in a different tenant in the runZero console.

Remediation: Change the email address of the existing runZero account or delete the existing runZero account. There are a few possible ways to fix this:

- Sign into the runZero console with the email address mentioned in the error message and change the associated email address to something different.
- Contact the administrators of the tenant where your email address is registered and request that they delete your account from the Team page.
- Contact runZero support via email at support@runzero.com and request that they delete your existing account.

Once the email address is no longer in use, you will be able to sign in with SSO.

The identity provider has provided an invalid NameID

Cause: Your SSO identity provider (for example, Okta, Google Workspace, etc) has provided the runZero console with a NameID attribute that is not formatted like an email address. The runZero console uses this attribute as your email address in the product.

Remediation: Review the configuration of your identity provider and ensure it is configured to provide an email address in the NameID attribute, or contact your administrators to do so.

runZero data formats

runZero consumes and produces a handful of data formats. This page provides examples of these formats and describes the fields and use cases for each.

Formats

- Scan data (sample)
- Asset data (sample)
- Change reports (sample)

Scan data

The raw output produced by the runZero Explorer and the runZero CLI is the **scan data**. This is newline-delimited JSON – JSONL – that represents the unprocessed output of the scan engine. This format is returned when downloading the task data for an Explorer-run scan and correlates to the `scan.runzero.gz` file created by the CLI. The runZero Inventory view is built by processing scan data in chronological order to create the current state at a given point in time.

Scan data can be imported into an existing site through the Inventory Import menu of the web console and through the `--import` parameter of the CLI. Each line of the file is a JSON object that specifies a `type` and a 64-bit Unix timestamp.

The example below is the raw scan data for a single Apple Mac Mini:


```
{"type":"config","ts":1597259738842951567,"probes":["arp","bacnet","dns","dtls","echo","ike","ipmi","m
dns","memcache","mssql","natpmp","netbios","ntp","openvpn","pca","rdns","rpcbind","sip","snmp","ssd
p","syn","tftp","ubnt","wlan-list","wsd"],"addresses":["192.168.0.1","192.168.30.1","192.168.40.1"],"n
etworks":["192.168.0.1/24","192.168.30.1/24","192.168.40.1/24"],"params":{"arp-fast":"false","bacnet-p
ort":"47808","clock-offset":"0","dns-port":"53","dns-resolve-name":"www.google.com","dns-trace-domai
n":"helper.rumble.network","dtls-ports":"443,3391,4433,5246,5349,5684","excludes":"","ike-port":"50
0","ipmi-port":"623","max-group-size":"4096","max-host-rate":"40","max-sockets":"512","mdns-port":"535
3","memcache-port":"11211","mssql-port":"1434","nameservers":"","natpmp-port":"5351","netbios-port":"1
37","nopcap":"false","ntp-port":"123","openvpn-ports":"1194","passes":"1","pca-port":"5632","probe
s":"arp,bacnet,dns,dtls,echo,ike,ipmi,mdns,memcache,mssql,natpmp,netbios,ntp,openvpn,pca,rdns,rpcbind,
sip,snmp,ssdp,syn,connect,tftp,ubnt,wlan-list,wsd","rate":"1000","rdns-max-concurrent":"64","rpcbind-p
ort":"111","rpcbind-port-nfs":"2049","screenshots":"true","sip-port":"5060","skip-broadcast":"true","s
```

uth-passphrase":"","snmp-v3-auth-protocol":"none","snmp-v3-context":"","snmp-v3-privacy-passphras
e":"","snmp-v3-privacy-protocol":"none","snmp-v3-username":"","ssdp-port":"1900","syn-max-retrie
s":"2","syn-udp-trace-port":"65535","tcp-ports":"1300,5554,8020,20034,47001,41080,2601,2604,2638,5060,
7181,10202,4679,2181,34205,13,2323,5601,18881,50070,139,1129,2199,2375,4444,902,1440,2103,32913,1311,9
524,8028,8883,13364,37718,512,3200,5683,10203,81,1091,5222,8081,13838,37777,1,5672,8095,65535,21,540,5
48,1102,27080,28017,34443,40007,6060,6542,8300,27888,4786,9443,2049,3050,5984,46823,12221,1352,6405,26
122,7210,41025,1103,1530,1883,8834,443,9100,45230,1234,3128,5432,12397,111,993,3780,5250,6112,524,524
7,20031,1211,1755,5985,6070,8880,1241,3690,6002,1035,4000,8080,9081,2362,23,587,921,8903,31001,143,259
8,3273,6101,8812,10628,25,113,513,1720,2533,6905,32764,38080,5040,20010,6001,6660,8471,82,2222,5093,62
62,6379,8545,384,5168,20222,7579,998,3057,3217,6106,9391,9,2380,5520,9060,19300,30718,49,84,161,5900,1
0001,8009,19,617,2100,5580,38292,85,6667,10443,42,2121,5986,23791,515,1199,10008,16993,631,2083,8443,9
527,13500,27017,30000,41523,554,5061,4659,8333,9855,5355,7001,1000,1220,5521,11234,20000,6988,3351,754
7,7,1900,7778,9160,31099,1030,10616,7902,8090,12174,1533,135,5631,623,5038,9300,19888,14330,109,1433,1
5672,1581,3790,5632,9999,80,2381,4840,7800,61616,1101,1128,1494,3311,9092,11000,110,995,1098,5800,523,
8087,10098,28784,407,7777,9090,19810,34963,50000,502,1100,8161,8180,9152,11099,2379,8023,88,1582,2010
1,16102,16992,1583,5814,5938,20111,11211,636,27000,1158,5400,5920,7443,9530,20171,8800,9099,7474,8222,
10000,2082,8902,50013,689,771,7080,8098,8686,22,1024,12345,105,9080,9111,47002,4433,44818,4848,6080,70
71,8303,62078,705,873,6000,7077,8503,9495,34964,5666,17200,5433,7801,11333,12401,12203,25025,264,2525,
3628,9809,26000,50090,9000,2967,137,4730,5051,8899,10050,52302,8400,53,389,402,4443,7700,62514,1090,53
53,6082,6661,40317,8089,17185,912,5405,28222,465,4445,6503,8014,57772,23472,1080,7021,8088,22222,5000,
9084,18264,8888,6050,7144,41524,69,1811,44334,102,6502,1521,2809,9471,888,5351,5498,123,34962,8205,904
2,2947,3389,5555,10080,10162,9200,500,1089,7510,15200,2207,3500,8008,9418,3460,6504,7770,25000,55553,1
79,783,8012,9595,46824,7580,5560,49152,83,903,1604,3632,4322,4567,445,8030,9390,3817,8901,10051,9002,2
7019,910,1099,3000,3299,222,7787,37,2000,3306,48899,7879,79,1723,3037,3312,8000,23943","tftp-ports":"6
9","ubnt-port":"10001","verbose":"true","wlan-list-poll-interval":"300","wsd-port":"3702"},"scan_targe
ts":{"networks":["192.168.0.5/32"],"enable_dns":true,"enable_ip6":false,"inputs":["192.168.0.5"],"dns_
timeout":2000000000,"concurrency":24},"version":"1.10.0 (build 20200804052508) [eae4e551f9f0ce5ab3bf0a
1410b2ed5098db097e]"}
{"type":"status","ts":1597259758851278069,"level":"info","source":"connect","msg":"waiting on TCP prob
es to complete"}
{"type":"stats","ts":1597259769858733899,"stats":{"elapsed":31,"progress":94,"rateLimitTime":242464458
95,"recv":546,"recvBytes":37404,"recvError":10,"recvRate":17,"resultCount":22,"secondsLeft":1,"sent":5
03,"sentBytes":36833,"sentError":0,"sentRate":16,"startTime":1597259738843056937}}
{"type":"result","ts":1597259738844372618,"host":"192.168.0.5","port":"0","proto":"icmp","probe":"ech
o","info":{"icmp.addrs":"192.168.0.5","icmp.rtts":"541214","ip.flags":"DF","ip.id":"0","ip.tos":"0","i
p.ttl":"64"}}
{"type":"result","ts":1597259739077542717,"host":"192.168.0.5","port":"137","name":"MACMINI-EE7C7B","p
roto":"udp","probe":"netbios","info":{"netbios.domain":"WORKGROUP","netbios.mac":"f0:18:98:ee:7c:7
b","netbios.macDateAdded":"2017-12-23","netbios.macVendor":"Apple, Inc."}}
{"type":"result","ts":1597259739219939211,"host":"192.168.0.5","port":"137","proto":"udp","probe":"net
bios","info":{"netbios.addrs":"192.168.0.5"}}
{"type":"result","ts":1597259739471688048,"host":"192.168.0.5","port":"0","proto":"arp","probe":"ar
p","info":{"arp.mac":"f0:18:98:ee:7c:7b","arp.macDateAdded":"2017-12-23","arp.macVendor":"Apple, In
c.","source":"arp"}}
{"type":"result","ts":1597259739826114224,"host":"192.168.0.5","port":"5353","name":"Developers-Mac-mi
ni","proto":"udp","probe":"mdns","info":{"mdns.replies":"5.0.168.192.in-addr.arpa.=PTR,Developers-Mac-
mini.local."}}
{"type":"result","ts":1597259740197709484,"host":"192.168.0.5","port":"445","proto":"tcp","probe":"con
nect","info":{"ntlmssp.dnsComputer":"Developers-Mac-mini.local","ntlmssp.dnsDomain":"local","ntlmssp.n
egotiationFlags":"0x62898235","ntlmssp.netbiosComputer":"DEVELOPERS-MAC-MINI","ntlmssp.netbiosDomai
n":"MACMINI-EE7C7B","ntlmssp.ntlmRevision":"15","ntlmssp.targetName":"MACMINI-EE7C7B","ntlmssp.timesta
mp":"0x01d670dd06500880","ntlmssp.version":"6.1.7600","protocol":"smb1\tsmb2\tsmb3","smb.capabilitie
s":"0x00000066","smb.dialect":"0x0302","smb.guid":"ff12583f-5ba1-53b8-8ff3-a48a394056f7","smb.nativeL-

M":"@(#)PROGRAM:smbd PROJECT:smbx-499.60.1","smb.nativeOS":"Darwin","smb.sessionID":"0x9551b7cb0000000
1","smb.signing":"required","source":"mdns"}}

The data contains four types of object:

- **Scan config:** The `{"type":"config"}` object contains the full set of parameters for the scan as well as the version of the scan engine, and on Windows, the version of npcap installed. This record is used to determine the scan targets, which is used by the analysis engine to determine whether a given IP address was in scope.
- **Scan status:** The `{"type":"status"}` object contains diagnostic output from the scan engine. This can highlight issues that occurred while the scan was running.
- **Scan stats:** The `{"type":"stats"}` object represents point in time statistics for the scan. This will include the number of packets sent, received, and the progress estimate.
- **Scan result:** The `{"type":"result"}` object is target response for a specific probe. This can include TCP SYN+ACK replies, ICMP replies, or the result of application-layer probes, such as SNMP query responses, or HTTP screenshots. Scan Results are analyzed and correlated to create to the Asset Data format.

The scan stats sub-fields are defined below:

Field	Description
cpu	CPU Core Percent * 100. 100% of one core would be 100000.
elapsed	The number of seconds since the scan started.
fdcount	The number of open file descriptors.
memory	The current memory usage in bytes.
progress	The estimated progress as a percentage (90 = 90%).
rateLimitTime	The number of Unix nanoseconds spent idling in the rate limiter.
recv	The number of packets received from the network.
recvBytes	The number of bytes received from the network.
recvError	The number of errors receiving from the network.
recvRate	The average packet receive rate for the scan.
resultCount	The total number of findings from the scan.
routines	The number of internal goroutines in the scan engine.
secondsLeft	The estimated seconds left to complete the scan.
sent	The number of packets sent the network.
sentBytes	The number of bytes sent the network.
sentError	The number of errors sending to the network.

startTime	The Unix timestamp in nanoseconds of when the scan started.

The scan result object type contains the following fields in addition to `type` and `ts`:

Field	Description
host	The IP address associated with the response.
name	An optional hostname returned as part of this response.
port	The TCP or UDP port or zero for other protocols.
proto	The transport protocol, one of `arp`, `icmp`, `tcp`, or `udp`.
probe	The specific internal probe name that returned this response.
info	The result details object where all keys and values are strings.

The `info` object contains probe-specific response data. The key names are typically in the format of `probe.subfield`, with a few exceptions, and the values are always strings, even for numeric and array content. Multiple values for a key are represented as a tab-delimited array. Empty values are never reported for info keys. A given scan may return multiple result objects for a single probe, sometimes with duplicate values. These responses are correlated, deduplicated, and merged during the next phase of processing.

Asset data

The correlated and fingerprinted assets shown in the web console Inventory view and in the `assets.jsonl` file produced by the runZero CLI are the **asset data**. This data represents the state of each unique asset at a point in time and is built up by processing one or more sets of scan data.

runZero supports a few variants of the asset data, including line-delimited JSON (JSONL), standard JSON documents, and a simplified CSV export. The JSONL format is the easiest to work with as it supports incremental processing without having to load the entire response into memory.

The example below is the correlated asset data for a scan of a single Apple Mac Mini:


```
{"id":"b73f8e09-78a6-4d2b-979d-e63908f28251","created_at":1597259778,"updated_at":1597259778,"organiza
tion_id":"b7fb13a7-701d-4ca5-b0e6-6f28f06cc866","site_id":"52d60c51-8dee-4f09-94e5-2dee30050a25","aliv
e":true,"last_seen":1597259750,"first_seen":1597259738,"detected_by":"arp","type":"Desktop","os":"Appl
e macOS","os_version":"10.15","hw":"Apple Mac Mini (Late 2018)","addresses":["192.168.0.5"],"addresses
_extra":["fe80::1c9d:c567:8db1:d79b"],"macs":["f0:18:98:ee:7c:7b"],"mac_vendors":["Apple, Inc."],"name
s":["MACMINI-EE7C7B","DEVELOPERS-MAC-MINI","DEVELOPERS-MAC-MINI.LOCAL"],"tags":{},"domains":[],"servic
es":{"192.168.0.5/0/arp/":{"arp.mac":"f0:18:98:ee:7c:7b","arp.macDateAdded":"2017-12-23","arp.macVendo
r":"Apple, Inc.","source":"arp","ts":"1597259739"},"192.168.0.5/0/icmp/":{"icmp.addrs":"192.168.0.
5","icmp.rtts":"541214","ip.flags":"DF","ip.id":"0","ip.tos":"0","ip.ttl":"64","ts":"1597259738"},"19
2.168.0.5/137/udp/":{"netbios.addrs":"192.168.0.5","netbios.domain":"WORKGROUP","netbios.mac":"f0:18:9
8:ee:7c:7b","netbios.macDateAdded":"2017-12-23","netbios.macVendor":"Apple, Inc.","protocol":"netbio
s","ts":"1597259739"},"192.168.0.5/22/tcp/":{"banner":"SSH-2.0-OpenSSH_7.9","ip.flags":"DF","ip.i
```

ce.family":"OpenSSH","service.vendor":"OpenBSD","service.version":"7.9","source":"mdns","ssh.hostKey.data":"AAAAB3NzaC1yc2EAAAADAQABAAABAQCjGYTFcSp2Fs/R8dboLYiQ6PPrulZYanYH3SCYYr5QgC1SIF3AURGYTMnUDAS+tTI/Pquwowkgiq3rtfsQMAsCrahbPahwiOLTupsuLNp3evXYYSf8ZQFyBN8iz5cys06u+yczqWG7Fu8mgpS8zwCwN7yRrbFWd8+Hp6GgfUU4Z6jUQoZu7iajpbSXlTA9OYKXQIZOm8qc4mPLT/uHw9nxNmExWA1V/2ZeoS59NGSV8zFMKb52SOXKhkvHAIUVh5NJDAudxK4uP4eG6dxr8btYtVKIOYKlsLdSBSfHvSCvVVlb7DKJBiMXG+qspt33Zd73o4S9ICh2OaSbVt7h/NZ3","ssh.hostKey.md5":"75:9b:a2:e6:10:da:72:8a:11:91:3f:a1:43:14:7f:2e","ssh.hostKey.sha256":"SHA256:xVJfddKBJ9E5jstVCj0zY8763Rnxy2pqpzaLZXO+cHc","ssh.hostKey.type":"ssh-rsa","syn.rtt":"542019","tcp.options":"MSS:05b4","tcp.ts":"2009546838","tcp.urg":"0","tcp.win":"65535","ts":"1597259740"},"192.168.0.5/3031/tcp/":{"source":"mdns","ts":"1597259740"},"192.168.0.5/3283/tcp/":{"source":"mdns","ts":"1597259740"},"192.168.0.5/445/tcp/":{"ip.flags":"DF","ip.id":"0","ip.tos":"0","ip.ttl":"64","ntlmssp.dnsComputer":"Developers-Mac-mini.local","ntlmssp.dnsDomain":"local","ntlmssp.negotiationFlags":"0x62898235","ntlmssp.netbiosComputer":"DEVELOPERS-MAC-MINI","ntlmssp.netbiosDomain":"MACMINI-EE7C7B","ntlmssp.ntlmRevision":"15","ntlmssp.targetName":"MACMINI-EE7C7B","ntlmssp.timestamp":"0x01d670dd06500880","ntlmssp.version":"6.1.7600","protocol":"smb1\tsmb2\tsmb3","smb.capabilities":"0x00000066","smb.dialect":"0x0302","smb.guid":"ff12583f-5ba1-53b8-8ff3-a48a394056f7","smb.nativeLM":"@(#)PROGRAM:smbd PROJECT:smbx-499.60.1","smb.nativeOS":"Darwin","smb.sessionID":"0x9551b7cb00000001","smb.signing":"required","source":"mdns","syn.rtt":"624414","tcp.options":"MSS:05b4","tcp.ts":"2009544186","tcp.urg":"0","tcp.win":"65535","ts":"1597259740"},"192.168.0.5/5353/udp/":{"hw.device":"Desktop","hw.family":"Mac mini","hw.product":"Mac mini (Late 2018)","hw.vendor":"Apple","mdns.addrs":"fe80::1c9d:c567:8db1:d79b\t192.168.0.5","mdns.device.model":"Macmini8,1","mdns.device.osxvers":"19","mdns.ports":"eppc/tcp=3031\tnet-assistant/udp=3283\trfb/tcp=5900\tsftp-ssh/tcp=22\tsmb/tcp=445\tssh/tcp=22","mdns.replies":"5.0.168.192.in-addr.arpa.=PTR,Developers-Mac-mini.local.\tDeveloper\\226\\128\\153s\\ Mac\\ mini._device-info._tcp.local.=TXT,model=Macmini8,1 osxvers=19\tDevelopers-Mac-mini.local.=A,192.168.0.5\tDevelopers-Mac-mini.local.=AAAA,fe80::1c9d:c567:8db1:d79b\t_eppc._tcp.local.=PTR,Developer\\226\\128\\153s\\ Mac\\ mini._eppc._tcp.local.\t_net-assistant._udp.local.=PTR,Developer\\226\\128\\153s\\ Mac\\ mini._net-assistant._udp.local.\t_rfb._tcp.local.=PTR,Developer\\226\\128\\153s\\ Mac\\ mini._rfb._tcp.local.\t_sftp-ssh._tcp.local.=PTR,Developer\\226\\128\\153s\\ Mac\\ mini._sftp-ssh._tcp.local.\t_smb._tcp.local.=PTR,Developer\\226\\128\\153s\\ Mac\\ mini._smb._tcp.local.\t_ssh._tcp.local.=PTR,Developer\\226\\128\\153s\\ Mac\\ mini._ssh._tcp.local.","mdns.services":"ssh/tcp\tsftp-ssh/tcp\teppc/tcp\trfb/tcp\tsmb/tcp\tnet-assistant/udp","os.cpe23":"cpe:/o:apple:mac_os_x:10.15","os.family":"Mac OS X","os.product":"Mac OS X","os.vendor":"Apple","os.version":"10.15","protocol":"mdns","ts":"1597259740"},"192.168.0.5/5900/tcp/":{"ip.flags":"DF","ip.id":"0","ip.tos":"0","ip.ttl":"64","protocol":"vnc","source":"mdns","syn.rtt":"625419","tcp.options":"MSS:05b4","tcp.ts":"2009549134","tcp.urg":"0","tcp.win":"65535","ts":"1597259749","vnc.version":"RFB 003.889"}},"credentials":{},"rtts":{"icmp/echo":[541214]},"attributes":{"_macs.ipmap":"f0:18:98:ee:7c:7b=192.168.0.5","ip.ttl.hops":"0","ip.ttl.host":"192.168.0.5","ip.ttl.port":"22","ip.ttl.source":"64","ip.ttl.source.icmp":"64","ip.ttl.win":"65535","match.db":"mdns-device-info-txt","match.score":"90","ntlmssp.dnsComputer":"Developers-Mac-mini.local","ntlmssp.dnsDomain":"local","ntlmssp.version":"6.1.7600","os.cpe23":"cpe:/o:apple:mac_os_x:10.15","os.family":"Mac OS X","os.product":"Mac OS X","os.vendor":"Apple","os.version":"10.15","smb.guid":"ff12583f-5ba1-53b8-8ff3-a48a394056f7","smb.nativeLM":"@(#)PROGRAM:smbd PROJECT:smbx-499.60.1","smb.nativeOS":"Darwin"},"service_count":9,"service_count_tcp":5,"service_count_udp":2,"service_count_arp":1,"service_count_icmp":1,"lowest_ttl":0,"lowest_rtt":541214,"last_agent_id":"ca811190-329c-4da3-8cbe-3fd2ddff2663","last_task_id":"de5a4176-3614-4b71-8939-95b9108124aa","newest_mac":"f0:18:98:ee:7c:7b","newest_mac_vendor":"Apple, Inc.","newest_mac_age":151398720000000000,"comments":null,"service_ports_tcp":["22","445","3031","3283","5900"],"service_ports_udp":["137","5353"],"service_protocols":["mdns","netbios","smb1","smb2","smb3","ssh","vnc"],"service_products":["openbsd openssh"],"org_name":"Test Lab","site_name":"MAC","agent_name":"TENTACULAR"}

Asset Data uses a number of data types for top-level fields, including string arrays, objects, strings, and integers. runZero tracks multiple IP addresses and MACs per asset and these are represented as arrays. For asset-level attributes and services, these are stored as objects with additional structure. Assets are uniquely identified by the `id` field (a V4 UUID) and nearly every other field can be changed between scans, as assets move around the network, change IPs, and open and close services.

Every asset belongs to an organization and a site within that organization.

The core asset data fields are defined below.

Field	Description
id	The unique ID of this asset defined as a v4 UUID.
created_at	The asset created time represented as a 64-bit Unix timestamp in seconds.
updated_at	The asset last update time represented as a 64-bit Unix timestamp in seconds.
organization_id	The organization identifier defined as a v4 UUID.
site_id	The site identifier defined as a v4 UUID.
alive	A boolean indicating whether this asset was found during the last scan of the site.
last_seen	The time the asset last responded represented as a 64-bit Unix timestamp in seconds.
first_seen	The time the asset first responded represented as a 64-bit Unix timestamp in seconds.
detected_by	The protocol used to first detect that this asset was alive during the last scan.
type	A classification that represents a guess of the asset's purpose.
os_vendor	The operating system vendor name as determined by the fingerprinting engine.
os_product	The operating system product name as determined by the fingerprinting engine.
os_version	The operating system version as determined by the fingerprinting engine.
os	The operating system name as determined by the fingerprinting engine.
hw_vendor	The hardware vendor name as determined by the fingerprinting engine.
hw_product	The hardware product name as determined by the fingerprinting engine.
hw_version	The hardware version as determined by the fingerprinting engine.
hw	The hardware definition as determined by the fingerprinting engine.
addresses	An array of IP (v4/v6) addresses for the asset that were within the scan scope.
addresses_extra	An array of IP (v4/v6) addresses for the asset that were outside the scan scope.
macs	An array of MAC addresses associated with this asset.

mac_vendors	An array of MAC address vendors associated with this asset.
names	An array of unique hostnames associated with this asset (uppercase).
domains	An array of unique domain names associated with this asset (uppercase).
tags	A text representation of the user-specified tags associated with this asset.
attributes	An object containing a map of key-value string attributes for this asset.
services	An object containing each associated service with the key representing the service description.
credentials	An object containing a map of any associated credentials (SNMP v2/v3).
rtts	An object containing a map of round-trip measurement times in milliseconds.
service_count	A count of TCP, UDP, ARP, and ICMP services.
service_count_tcp	A count of TCP services.
service_count_udp	A count of UDP services.
service_count_arp	A count of ARP services (0 or 1).
service_count_icmp	A count of ICMP services.
software_count	A count of software results.
vulnerability_count	A count of vulnerability results.
lowest_ttl	The lowest observed source TTL for this asset.
lowest_rtt	The lowest observed source RTT for this asset.
last_agent_id	The v4 UUID of the Explorer responsible for the last scan of this asset.
last_task_id	The v4 UUID of the task responsible associated with the last scan of this asset.
last_task_id	The v4 UUID of the task responsible associated with the last scan of this asset.
newest_mac	The “newest” MAC address by registration date.
newest_mac_vendor	The “newest” MAC address vendor by registration date.
newest_mac_age	The “newest” MAC address registration date as a Unix timestamp in nanoseconds.
comments	User-specified comments associated with this asset.
service_ports_tcp	An array of strings representing the unique TCP ports found on this asset.

service_ports_udp	An array of strings representing the unique UDP ports found on this asset.
service_protocols	An array of strings representing the unique protocols found on this asset.
service_products	An array of strings representing the unique products found on this asset.
scanned	A TRUE or FALSE value indicating whether the asset has been scanned by runZero.
source_ids	The ID of the data source, mapped to this table.
eol_os	The operating system End-of-Life time represented as a 64-bit Unix timestamp in seconds.
eol_os_ext	The operating system extended End-of-Life time represented as a 64-bit Unix timestamp in seconds.
outlier_score	The 0-5 score range indicating how unusual an asset is compared to the rest of the inventory.
outlier_raw	The heuristic score indicating how unusual an asset is compared to the rest of the inventory.
sources	The name of the data source, mapped to this table.
org_name	The name of the organization associated with this asset.
site_name	The name of the site associated with this asset.
agent_name	The name of the Explorer associated with this asset.
agent_external_ip	The external IP address of the Explorer associated with this asset.
hosted_zone_name	The name of the hosted zone associated with this asset.
subnets	The registered subnets associated with the site this asset is in.

The `services` field contains string keys that contain the unique service identifier with values stored as strings. Multiple values may be stored as tab-delimited strings in the service values. A typical service key looks like `192.168.0.5/22/tcp/`. The components of the service key name consist of `address`, `port`, `transport`, and `virtual host` (which can be blank).

Change reports

The runZero platform calculates a **change report** after processing each scan. This is a JSON document available for download from the Task Details page with the following structure.

```
{
  "assets":{
        "new":{ "<asset-UUID>": { "Asset Data Fields":"" } },
        "online":{ "<asset-UUID>": { "Asset Data Fields":"" } },
```

```
"changed":{ "<asset-UUID>": { "Asset Data Fields":"" } },
        "summary":{
          "changed":#,
          "new":#,
          "total":#,
          "unchanged":#
        } },
  "directory_users":{
        "new":{ "<user-UUID>": { "User Data Fields":"" }},
        "changed":{ "<user-UUID>": { "User Data Fields":"" }},
        "summary":{
          "changed":#,
          "new":#,
          "total":#,
          "unchanged":#
        } },
  "directory_groups":{
        "new":{ "<group-UUID>": { "Group Data Fields":"" } },
        "changed":{ "<group-UUID>": { "Group Data Fields":"" } },
        "summary":{
          "changed":#,
          "new":#,
          "total":#,
          "unchanged":#
        } },
        "truncated": <true/false>
}
```

The `new`, `online`, `offline`, and `changed` objects each contain keys consisting of the modified asset IDs with the values represented in the **asset data** format. The `summary` field indicates overall change statistics for this task. The `truncated` field is set to true if the change report is incomplete due to reaching the maximum change threshold (1000 asset changes today).

runZero data dictionary

runZero discovers and catalogs the following fields from known protocols and services:

Protocol	Attribute name	Data type	Single-value
acpp	`banner`	string	false
activemq	`activemq.jvm.vendor`	string	true
activemq	`activemq.jvm.version`	string	true
activemq	`activemq.os`	string	true
activemq	`activemq.wireformat.data`	string	true
activemq	`activemq.wireformat.version`	string	true
adb	`adb.access`	string	true

ads	ads.deviceName	string	true
ads	ads.version	string	true
airplay	airplay.build	numeric	true
airplay	airplay.canRecordScreenStream	boolean	true
airplay	airplay.deviceID	string	true
airplay	airplay.features	numeric	true
airplay	airplay.featuresEx	string	true
airplay	airplay.firmwareBuildDate	string	true
airplay	airplay.firmwareRevision	numeric	true
airplay	airplay.hardwareRevision	numeric	true
airplay	airplay.hasUDPMirroringSupport	boolean	true
airplay	airplay.initialVolume	numeric	true
airplay	airplay.keepAliveLowPower	boolean	true
airplay	airplay.keepAliveSendStatsAsBody	boolean	true
airplay	airplay.macAddress	string	true
airplay	airplay.manufacturer	string	true
airplay	airplay.model	string	true
airplay	airplay.name	string	true
airplay	airplay.osBuildVersion	string	true
airplay	airplay.osinfo	string	true
airplay	airplay.pi	uuid	true
airplay	airplay.pk	string	true
airplay	airplay.playbackCapabilities	string	false
airplay	airplay.protocolVersion	numeric	true
airplay	airplay.ptpinfo	string	true
airplay	airplay.receiverHDRCapability	string	true
airplay	airplay.screenDemoMode	boolean	true
airplay	airplay.sdk	string	true
airplay	airplay.senderAddress	string	true
airplay	airplay.sourceVersion	numeric	true
airplay	airplay.statusFlags	string	true

airplay	airplay.supportedFormats	string	true
airplay	airplay.volumeControlType	numeric	true
airplay	airplay.vv	numeric	true
airplay	apache.serverInfo	string	true
amqp	amqp.protocolVersion	string	true
amqp	amqp.saslRequired	string	true
amqp	amqp.tlsRequired	string	true
atg	atg.module	string	true
atg	atg.software	string	true
atg	atg.version	string	true
bacnet	bacnet.bbmd	string	false
bacnet	bacnet.fdt	string	false
bacnet	bacnet.instanceID	numeric	true
bacnet	bacnet.pid	numeric	true
bacnet	bacnet.vendorID	numeric	true
bacnet	bacnet.vendorIDLookup	string	true
bedrock	bedrock.contents	string	true
bedrock	bedrock.guid	string (hex-encoded)	true
bedrock	bedrock.uptime	numeric	true
bitdefender-app	bitdefender.deviceID	string	true
bitdefender-app	bitdefender.hostname	string	true
bitdefender-app	bitdefender.model	string	true
bitdefender-app	bitdefender.os	string	true
bitdefender-app	bitdefender.type	string	true
bjnp	bjnp.ipv4	string	true
bjnp	bjnp.ipv6	string	true
bjnp	bjnp.macAddress	string	true
bjnp	bjnp.type	string	true
brother-scanner	brother.scanner	string	true
cacti	cacti.version	string	true

cassandra	cassandra.cluster	string	true
cassandra	cassandra.version	string	true
chargen	chargen.osGuess	string	true
checkmk	banner	string	true
checkmk	checkmk.agentOS	string	true
checkmk	checkmk.arch	string	true
checkmk	checkmk.buildDate	string	true
checkmk	checkmk.hostname	string	true
checkmk	checkmk.version	string	true
cisco-phone	ciscocp.apploadid	string	true
cisco-phone	ciscocp.bootloadid	string	true
cisco-phone	ciscocp.hostname	string	true
cisco-phone	ciscocp.mac	string	true
cisco-phone	ciscocp.messagewaiting	string	true
cisco-phone	ciscocp.model	string	true
cisco-phone	ciscocp.phonedn	string	true
cisco-phone	ciscocp.revision	string	true
cisco-phone	ciscocp.serial	string	true
cisco-phone	ciscocp.series	string	true
cisco-phone	ciscocp.udi	string	true
cisco-phone	ciscocp.version	string	true
ciscosmi	ciscosmi.reply	string (hex-encoded)	true
coap	coap.contentUnknown	string	true
coap	coap.options	string	true
coap	coap.responseCode	string	true
coap	coap.type	string	true
coap	coap.version	string	true
cockpit	cockpit.os	string	true
cockpit	host.name	string	false
comtrol	dnsServer	string	true

comtrol	hostname	string	true
comtrol	ipAddress	string	true
comtrol	macAddress	string	true
comtrol	manufacturer	string	true
comtrol	modelName	string	true
comtrol	modelNumber	string	true
comtrol	netmask	string	true
comtrol	osVersion	string	true
comtrol	serialNumber	string	true
confluence	confluence.baseURL	string	true
confluence	confluence.build	string	true
confluence	confluence.version	string	true
consul	consul.config.version	string	true
couchdb	couchdb.version	string	true
crestron	crestron.banner	string	true
crestron	crestron.buildDate	string	true
crestron	crestron.hostname	string	true
crestron	crestron.id	string	true
crestron	crestron.mac	string	true
crestron	crestron.model	string	true
crestron	crestron.version	string	true
dahua-dhip	dhip.alarmInputChannels	numeric	true
dahua-dhip	dhip.alarmOutputChannels	numeric	true
dahua-dhip	dhip.deviceClass	string	true
dahua-dhip	dhip.deviceType	string	true
dahua-dhip	dhip.find	string	true
dahua-dhip	dhip.httpPort	numeric	true
dahua-dhip	dhip.init	numeric	true
dahua-dhip	dhip.ipv4.address	string	true
dahua-dhip	dhip.ipv4.dhcp	boolean	true
dahua-dhip	dhip.ipv4.gateway	string	true

dahua-dhip	dhip.ipv4.netmask	string	true
dahua-dhip	dhip.ipv6.address	string	true
dahua-dhip	dhip.ipv6.dhcp	boolean	true
dahua-dhip	dhip.ipv6.gateway	string	true
dahua-dhip	dhip.ipv6.linkLocalAddress	string	true
dahua-dhip	dhip.mac	string	true
dahua-dhip	dhip.machineName	string	true
dahua-dhip	dhip.manufacturer	string	true
dahua-dhip	dhip.method	string	true
dahua-dhip	dhip.port	numeric	true
dahua-dhip	dhip.remoteVideoInputChannels	numeric	true
dahua-dhip	dhip.serialNo	string	true
dahua-dhip	dhip.vendor	string	true
dahua-dhip	dhip.version	string	true
dahua-dhip	dhip.videoInputChannels	numeric	true
dahua-dhip	dhip.videoOutputChannels	numeric	true
daytime	banner	string	true
daytime	daytime.osGuess	string	true
daytime	daytime.timestamp	string	true
dhcp	dhcp.bootFile	string	true
dhcp	dhcp.broadcast	boolean	true
dhcp	dhcp.class	string	false
dhcp	dhcp.clientSystemArchitecture	string	false
dhcp	dhcp.clientid.enterprise	string	false
dhcp	dhcp.flags	numeric	true
dhcp	dhcp.seconds	numeric	true
dhcp	dhcp.serverName	string	true
dhcp	host.ip	string	false
dhcp	host.mac	string	true
dhcp	host.name	string	false
dns	dns.authors.bind	string	true

dns	dns.hostname.bind	string	true
dns	dns.id.server	string	true
dns	dns.opcode	string	true
dns	dns.replies	string	false
dns	dns.version.bind	string	true
dns	dns.version.server	string	true
docker	docker.architecture	string	true
docker	docker.dockerRootDir	string	true
docker	docker.kernelVersion	string	true
docker	docker.memTotal	string	true
docker	docker.name	string	true
docker	docker.ncpu	string	true
docker	docker.operatingSystem	string	true
docker	docker.ostype	string	true
docker	docker.systemTime	string	true
docker	docker.version	string	true
dtls	dtls.alert	string (hex-encoded)	true
dtls	dtls.encapsulation	string	true
dtls	dtls.unknown	string (hex-encoded)	true
echo	banner	string	true
elasticsearch	elasticsearch.version.number	string	true
epm	epm.addrLen	numeric	true
epm	epm.address	string	true
epm	epm.assocGroup	numeric	true
epm	epm.dataRep	string (hex-encoded)	true
epm	epm.maxRecvFrag	numeric	true
epm	epm.maxSendFrag	numeric	true
epm	epm.version	string	true
epo	epo.guid	string	true
epo	epo.hostname	string	true
epo	epo.server	string	true

epo	epo.version	string	true
epo	mcafeeAgent.hostname	string	true
epo	mcafeeAgent.version	string	true
etcd2	etcd2.access	string	true
etcd2	etcd2.keys	string	true
fins	fins.model	string	true
fins	fins.version	string	true
fortigate-to-fortimanager	tls.cipherSuiteNames	string	false
fortigate-to-fortimanager	tls.cipherSuites	string	false
fortigate-to-fortimanager	tls.extensions	string	false
fortigate-to-fortimanager	tls.version	string	true
fortigate-to-fortimanager	tls.versionName	string	true
ftp	banner	string	true
googlewifi	googleWifi.software.version	string	true
googlewifi	googleWifi.system.countryCode	string	true
googlewifi	googleWifi.system.hardwareID	string	true
googlewifi	googleWifi.system.modelID	string	true
googlewifi	googleWifi.wan.gateway	string	true
googlewifi	googleWifi.wan.localIP	string	true
hiddiscoveryd	hiddiscoveryd.address	string	true
hiddiscoveryd	hiddiscoveryd.hostname	string	true
hiddiscoveryd	hiddiscoveryd.mac	string	true
hiddiscoveryd	hiddiscoveryd.model	string	true
hiddiscoveryd	hiddiscoveryd.unpatchedVertXploit	string	true
hiddiscoveryd	hiddiscoveryd.version	string	true
hiddiscoveryd	hiddiscoveryd.versionDate	string	true
http	html.copyright	string	true
http	html.favicon	string	true
http	html.generator	string	true
http	html.title	string	true

http	http.code	numeric	true
http	http.head.acceptRanges	string	true
http	http.head.contentLength	numeric	true
http	http.head.contentType	string	true
http	http.head.date	string	true
http	http.head.etag	string	true
http	http.head.expires	string	true
http	http.head.lastModified	string	true
http	http.head.location	string	true
http	http.head.server	string	true
http	http.head.setCookie	string	true
http	http.head.vary	string	true
http	http.head.xframeoptions	string	true
http	http.message	string	true
http	http.method	string	true
http	http.path	string	true
http	http.uri	string	true
http	http.url	string	true
http	landesk.configPath	string	true
http	landesk.providerVersion	string	true
http	landesk.serverVersion	string	true
ident	ident.error	string	true
ident	ident.opSys	string	true
ident	ident.osGuess	string	true
ident	ident.username	string	true
igel-discovery	igel.firmwareVersion	string	true
igel-discovery	igel.hostname	string	true
igel-discovery	igel.ipAddress	string	true
igel-discovery	igel.macAddress	string	true
igel-discovery	igel.osType	string	true
igel-discovery	igel.productID	string	true

igel-discovery	`igel.productName`	string	true
iis	`http.owa.version`	string	true
iis	`http.owa.version.full`	string	true
iis	`rdg.authScheme`	string	true
iis	`rdg.transport`	string	true
ike	`ike.exchangeType`	string	true
ike	`ike.flags`	string	true
ike	`ike.initiatorSPI`	string (hex-encoded)	true
ike	`ike.messageID`	string	true
ike	`ike.messageLength`	string	true
ike	`ike.nextPayload`	string	true
ike	`ike.payload`	string	true
ike	`ike.responderSPI`	string (hex-encoded)	true
ike	`ike.sha1`	string (hex-encoded)	true
ike	`ike.vendorID`	string	true
ike	`ike.version`	string	true
influxdb	`influxdb.build`	string	true
influxdb	`influxdb.databases`	string	false
influxdb	`influxdb.version`	string	true
ipmi	`ipmi.channel`	numeric	true
ipmi	`ipmi.completionCode`	numeric	true
ipmi	`ipmi.connVersions`	string	true
ipmi	`ipmi.connVersionsRaw`	string	true
ipmi	`ipmi.oemData`	string	true
ipmi	`ipmi.oemID`	numeric	true
ipmi	`ipmi.oemName`	string	true
ipmi	`ipmi.passAuth`	string	true
ipmi	`ipmi.passAuthRaw`	string	true
ipmi	`ipmi.srcAddr`	numeric	true
ipmi	`ipmi.srcLun`	numeric	true
ipmi	`ipmi.tgtAddr`	numeric	true

ipmi	ipmi.tgtLun	numeric	true
ipmi	ipmi.userAuth	string	true
ipmi	ipmi.userAuthRaw	string	true
iscsi	host.ip	string	false
iscsi	iscsi.acknowledge	string	true
iscsi	iscsi.bidiReadResidualCount	numeric	true
iscsi	iscsi.bufferOffset	numeric	true
iscsi	iscsi.continue	string	true
iscsi	iscsi.currentStage	string	true
iscsi	iscsi.dataDigest	string	false
iscsi	iscsi.dataSN	numeric	true
iscsi	iscsi.expCmdSN	numeric	true
iscsi	iscsi.expDataSN	string	true
iscsi	iscsi.final	string	true
iscsi	iscsi.flags.readResidualOverflow	string	true
iscsi	iscsi.flags.readResidualUnderflow	string	true
iscsi	iscsi.flags.residualOverflow	string	true
iscsi	iscsi.flags.residualUnderflow	string	true
iscsi	iscsi.flags.status	string	true
iscsi	iscsi.headerDigest	string	true
iscsi	iscsi.immediateDelivery	string	true
iscsi	iscsi.initiatorTaskTag	string	true
iscsi	iscsi.isid	string	false
iscsi	iscsi.logout.response	string	true
iscsi	iscsi.lun	string	true
iscsi	iscsi.lunAddressMode	string	true
iscsi	iscsi.maxCmdSN	numeric	true
iscsi	iscsi.nextStage	string	true
iscsi	iscsi.opcode	string	false
iscsi	iscsi.residualCount	numeric	true
iscsi	iscsi.response	string	false

iscsi	iscsi.statSN	numeric	true
iscsi	iscsi.status	string	false
iscsi	iscsi.statusClass	string	true
iscsi	iscsi.statusDetail	string	true
iscsi	iscsi.time2Retain	string	true
iscsi	iscsi.time2Wait	string	true
iscsi	iscsi.transferTaskTag	string	true
iscsi	iscsi.transit	string	true
iscsi	iscsi.tsih	string	true
iscsi	iscsi.versionMax	string	true
iscsi	iscsi.versionMin	string	true
jetdirect	pjl.id	string	true
jira	jira.baseURL	string	true
jira	jira.build	string	true
jira	jira.version	string	true
jms	banner	string	true
jms	jms.services	string	false
jms	jms.tcp.ports	string	false
jms	jms.version	string	true
kasa	kasa.deviceAlias	string	true
kasa	kasa.deviceId	string	true
kasa	kasa.deviceType	string	true
kasa	kasa.hardwareAddress	string	true
kasa	kasa.hardwareId	string	true
kasa	kasa.hardwareVersion	string	true
kasa	kasa.model	string	true
kasa	kasa.obdSource	string	true
kasa	kasa.oemId	string	true
kasa	kasa.softwareVersion	string	true
kerberos	kerberos.error	string	true
kerberos	kerberos.errorCode	string	true

kerberos	kerberos.microseconds	string	true
kerberos	kerberos.realm	string	true
kerberos	kerberos.serverTS	numeric	true
kerberos	kerberos.servicePrincipal	string	true
kerberos	kerberos.ticket.realm	string	true
kerberos	kerberos.version	string	true
knxnet	knxnet.channel	numeric	true
knxnet	knxnet.hpaiDataEP	string	true
knxnet	knxnet.status	numeric	true
knxnet	knxnet.tunnelData	string (hex-encoded)	true
knxnet	knxnet.tunnelEP	string	true
knxnet	knxnet.type	string	true
l2t	l2t.attrCount	numeric	true
l2t	l2t.type	numeric	true
l2t	l2t.version	numeric	true
l2tp	l2tp.assignedTunnelID	numeric	true
l2tp	l2tp.bearerCaps	numeric	true
l2tp	l2tp.controlCode	numeric	true
l2tp	l2tp.errorCode	numeric	true
l2tp	l2tp.errorMsg	string	true
l2tp	l2tp.firmwareRevision	numeric	true
l2tp	l2tp.framingCaps	numeric	true
l2tp	l2tp.hostName	string	true
l2tp	l2tp.nr	numeric	true
l2tp	l2tp.ns	numeric	true
l2tp	l2tp.protoVersion	string	true
l2tp	l2tp.recvWindowSize	numeric	true
l2tp	l2tp.resultCode	numeric	true
l2tp	l2tp.sessionID	numeric	true
l2tp	l2tp.tunnelID	numeric	true
l2tp	l2tp.vendorName	string	true

lantronix	`lantronix.fwinfo`	string (hex-encoded)	true
lantronix	`lantronix.mac`	string	true
lantronix	`lantronix.macVendor`	string	true
lantronix	`lantronix.serial`	string	true
lantronix	`lantronix.type`	string	true
lantronix	`lantronix.version`	string	true
ldap	`ldap.notes`	string	false
ldap	`ldap.searchresult`	string	true
lexmark	`banner`	string	true
lockdownd	`lockdownd.cpuArchitecture`	string	true
lockdownd	`lockdownd.deviceName`	string	true
lockdownd	`lockdownd.hardwareModel`	string	true
lockdownd	`lockdownd.productName`	string	true
lockdownd	`lockdownd.productType`	string	true
lockdownd	`lockdownd.productVersion`	string	true
lockdownd	`lockdownd.supportedDeviceFamilies`	string	true
lpd	`lpd.banner`	string	true
mdns	`mdns.addrs`	string	false
mdns	`mdns.authors.bind`	string	true
mdns	`mdns.hostname.bind`	string	true
mdns	`mdns.id.server`	string	true
mdns	`mdns.replies`	string	false
mdns	`mdns.version.bind`	string	true
mdns	`mdns.version.server`	string	true
mikrotik-bandwidth	`banner`	string	true
minecraft	`minecraft.currentplayers`	string	true
minecraft	`minecraft.maxplayers`	string	true
minecraft	`minecraft.motd`	string	true
minecraft	`minecraft.version`	string	true
modbus	`modbus.applicationName`	string	true

modbus	modbus.function	string	true
modbus	modbus.identifier	string	true
modbus	modbus.modelName	string	true
modbus	modbus.productCode	string	true
modbus	modbus.productName	string	true
modbus	modbus.revision	string	true
modbus	modbus.unitID	numeric	true
modbus	modbus.vendor	string	true
modbus	modbus.vendorURL	string	true
mqtt	mqtt.broker.authRequired	string	false
mqtt	mqtt.broker.supportedProtocols	string	false
mssql_browser	mssql.browser.response	string	false
mssql_browser	mssql.tcp.ports	string	false
munin	munin.capabilities	string	false
munin	munin.metrics	string	false
munin	munin.nodes	string	false
munin	munin.tlsRequired	string	true
munin	munin.version	string	true
mysql	mysql.error	string	true
mysql	mysql.version	string	true
mysqlx	mysqlx.error	string	true
mysqlx	mysqlx.errorCode	string	true
mysqlx	mysqlx.errorSeverity	string	true
natpmp	natpmp.externalIP	string	true
natpmp	natpmp.lastChange	string	true
natpmp	natpmp.responseCode	numeric	true
ndmp	ndmp.reason	string	true
ndmp	ndmp.status	string	true
ndmp	ndmp.timestampTS	numeric	true
ndmp	ndmp.version	string	true

neo4j	neo4j.neo4jVersion	string	true
netbios-dgm	cifs.comment	string	true
netbios-dgm	cifs.os.version	string	true
netbios-dgm	cifs.serverType	string	true
netbios-dgm	host.name	string	true
netbios-dgm	netbios.nodeType	string	true
netbios-ns	netbios.addrs	string	false
netbios-ns	netbios.domain	string	true
netbios-ns	netbios.domainController	string	true
netbios-ns	netbios.mac	string	true
netbios-ns	netbios.macDateAdded	string	true
netbios-ns	netbios.macVendor	string	true
netbios-ns	netbios.name	string	true
netbios-ns	netbios.primarydomainController	string	true
netbios-ns	netbios.username	string	true
nfs	nfs.maxVersion	numeric	true
nfs	nfs.minVersion	numeric	true
nfs	nfs.unknownReply	string (hex-encoded)	true
ntp	ntp.interval	numeric	true
ntp	ntp.readVar	string	true
ntp	ntp.referenceID	string	true
ntp	ntp.skew	string	true
ntp	ntp.skewMS	numeric	true
ntp	ntp.stratum	numeric	true
ntp	ntp.swVersion	string	true
ntp	ntp.timestamp	numeric	true
ntp	ntp.version	numeric	true
opcua	opcua.applications	string	false
opcua	opcua.clientThumbprint	string	true
opcua	opcua.endpoints	string	false
opcua	opcua.error	string	true

opcua	opcua.errorReason	string	true
opcua	opcua.errorText	string	true
opcua	opcua.messageType	string	true
opcua	opcua.protocol	numeric	true
opcua	opcua.securityPolicy	string	true
opcua	opcua.supportedMessageSecurityModes	string	false
opcua	opcua.supportedSecurityPolicies	string	false
opcua	opcua.supportedUserTokenTypes	string	false
opcua	opcua.unknownNodeId	string	false
openvpn	openvpn.reply	string (hex-encoded)	true
oracledb	oracledb.tns.error	string	true
oracledb	oracledb.tns.version	string	true
oracledb	oracledb.tns.vsn	string	true
orion	orion.components	string	true
orion	orion.version	string	true
pca	command	string	true
pca	pca.caps	string	true
pca	pca.name	string	true
pca	pca.status	string (hex-encoded)	true
pcworx	pcworx.firmwareDate	string	true
pcworx	pcworx.firmwareTime	string	true
pcworx	pcworx.firmwareVersion	string	true
pcworx	pcworx.modelNumber	string	true
pcworx	pcworx.plcType	string	true
pega	pega.version	string	true
postgresql	postgresql.auth.details	string	true
postgresql	postgresql.auth.method	string	true
postgresql	postgresql.error.code	string	true
postgresql	postgresql.error.file	string	true
postgresql	postgresql.error.line	string	true

postgresql	postgresql.error.routine	string	true
postgresql	postgresql.error.severity	string	true
postgresql	postgresql.error.text	string	true
pptp	pptp.bearerCapabilities	string	true
pptp	pptp.errorCode	numeric	true
pptp	pptp.framingCapabilities	string	true
pptp	pptp.fwRevision	numeric	true
pptp	pptp.hostname	string	true
pptp	pptp.maxChannels	numeric	true
pptp	pptp.resultCode	numeric	true
pptp	pptp.vendor	string	true
prosoft	firmwareDate	string	true
prosoft	moduleName	string	true
prosoft	moduleRevision	string	true
prosoft	moduleSerial	string	true
prosoft	moduleStatus	string	true
psdisco	psdisco.appName	string	true
psdisco	psdisco.appTitleID	string	true
psdisco	psdisco.code	string	true
psdisco	psdisco.id	string	true
psdisco	psdisco.name	string	true
psdisco	psdisco.protoVersion	string	true
psdisco	psdisco.requestPort	numeric	true
psdisco	psdisco.status	string	true
psdisco	psdisco.sysVersion	string	true
psdisco	psdisco.type	string	true
pwa_manifest	pwaManifest.id	string	true
pwa_manifest	pwaManifest.name	string	true
pwa_manifest	pwaManifest.shortName	string	true
pwa_manifest	pwaManifest.startUrl	string	true
qotd	banner	string	true

qualys	qualys.correlationID	string	true
radius	radius.replyMessages	string	true
rdp	rdp.auth.nla	string	true
rdp	rdp.auth.rdp	string	true
rdp	rdp.auth.ssl	string	true
rdp	rdp.auth.sspeua	string	true
rdp	rdp.auth.tls	string	true
redis	redis.protectedMode	string	true
rexec	banner	string	true
riak	riak.nodename	string	true
riak	riak.version	string	true
riak-http	riak-http.version	string	true
roomalert	roomalert.ipAddress	string	false
roomalert	roomalert.macAddress	string	false
roomalert	roomalert.model	string	false
roomalert	roomalert.osVersion	string	false
rtsp	rtsp.head.*	string	true
sadp	sadp.activated	string	true
sadp	sadp.analogChannelNum	string	true
sadp	sadp.bootTime	string	true
sadp	sadp.cmdPort	string	true
sadp	sadp.deviceDesc	string	true
sadp	sadp.deviceSerial	string	true
sadp	sadp.deviceType	string	true
sadp	sadp.dhcp	string	true
sadp	sadp.digitalChannelNum	string	true
sadp	sadp.dspVersion	string	true
sadp	sadp.httpPort	string	true
sadp	sadp.ipv4.address	string	true
sadp	sadp.ipv4.gateway	string	true

sadp	sadp.ipv6.address	string	true
sadp	sadp.ipv6.gateway	string	true
sadp	sadp.ipv6.maskLen	string	true
sadp	sadp.mac	string	true
sadp	sadp.oemInfo	string	true
sadp	sadp.passwordResetAbility	string	true
sadp	sadp.softwareVersion	string	true
sadp	sadp.uuid	string	true
securemote	securemote.hostname	string	true
securemote	securemote.server	string	true
servicetag	serviceTag.agentURN	string	true
servicetag	serviceTag.agentVersion	string	true
servicetag	serviceTag.registryVersion	string	true
servicetag	serviceTag.sysinfo.cpuInfo.name	string	true
servicetag	serviceTag.sysinfo.cpuMfg	string	true
servicetag	serviceTag.sysinfo.hostID	string	true
servicetag	serviceTag.sysinfo.hostname	string	true
servicetag	serviceTag.sysinfo.mfg	string	true
servicetag	serviceTag.sysinfo.platform	string	true
servicetag	serviceTag.sysinfo.release	string	true
servicetag	serviceTag.sysinfo.serialNumber	string	true
servicetag	serviceTag.sysinfo.system	string	true
sip	sip.head.*	string	true
slp	slp.version	numeric	true
smb3	ntlmssp.dnsComputer	string	true
smb3	ntlmssp.dnsDomain	string	true
smb3	ntlmssp.negFlags	string	true
smb3	ntlmssp.netbiosComputer	string	true
smb3	ntlmssp.netbiosDomain	string	true
smb3	ntlmssp.ntlmRevision	numeric	true
smb3	ntlmssp.targetName	string	true

smb3	ntlmssp.version	string	true
smb3	ntlmssp.versionInvalid	string	true
smb3	smb.accessControl	string	true
smb3	smb.capabilities	string	true
smb3	smb.cipherAlg	string	false
smb3	smb.cipherAlgCnt	string	true
smb3	smb.compAlg	string	false
smb3	smb.compAlgCnt	string	true
smb3	smb.compFlags	numeric	true
smb3	smb.dialect	string	true
smb3	smb.guid	string	true
smb3	smb.hashAlg	string	false
smb3	smb.hashAlgCnt	string	true
smb3	smb.hashSaltLen	string	true
smb3	smb.nativeLM	string	true
smb3	smb.nativeOS	string	true
smb3	smb.netName	string	true
smb3	smb.netbiosComputer	string	true
smb3	smb.netbiosDomain	string	true
smb3	smb.posixExtensions	string	true
smb3	smb.rdmaTransformCnt	string	true
smb3	smb.rdmaTransforms	string	false
smb3	smb.sessionID	string	true
smb3	smb.signing	string	true
smb3	smb.signingAlg	string	false
smb3	smb.signingAlgCnt	string	true
smb3	smb.supportsEncryptedPasswords	string	true
smb3	smb.transportSecurity	string	true
smtp	banner	string	true
snmp	snmp.defaultCommunities	string	true
snmp	snmp.interfaceAddrs	string	false

snmp	snmp.interfaceMacs	string	false
snmp	snmp.interfaces	string	false
snmp	snmp.secretCommunities	string	true
snmp	snmp.version	string	true
socks	banner	string	true
splunk	splunk.build	string	true
splunk	splunk.version	string	true
spotify-connect	brandDisplayName	string	true
spotify-connect	clientID	string	true
spotify-connect	deviceID	string	true
spotify-connect	deviceType	string	true
spotify-connect	modelDisplayName	string	true
spotify-connect	remoteName	string	true
spotify-connect	scope	string	true
spotify-connect	status	string	true
spotify-connect	statusString	string	true
ssdp	ssdp.head.*	string	false
ssh	banner	string	true
steam	steam.authKeyIDs	string	true
steam	steam.broadcastingActive	boolean	true
steam	steam.clientID	numeric	true
steam	steam.connectPort	numeric	true
steam	steam.contentCachePort	numeric	true
steam	steam.deviceID	numeric	true
steam	steam.downloadLANPeerGroup	numeric	true
steam	steam.eUniverse	numeric	true
steam	steam.enabledServices	numeric	true
steam	steam.gamesRunning	boolean	true
steam	steam.hostname	string	true
steam	steam.instanceID	numeric	true
steam	steam.ipAddresses	string	true

steam	steam.is64Bit	boolean	true
steam	steam.isSteamDeck	boolean	true
steam	steam.macAddresses	string	true
steam	steam.minVersion	numeric	true
steam	steam.osType	numeric	true
steam	steam.publicIPAddress	string	true
steam	steam.remotePlayActive	boolean	true
steam	steam.screenLocked	boolean	true
steam	steam.steamIDs	string	true
steam	steam.steamVersion	numeric	true
steam	steam.supportedServices	numeric	true
steam	steam.timestamp	numeric	true
steam	steam.version	numeric	true
steam	steam.vrActive	boolean	true
sunrpc	rpcbind.acceptState	string	true
sunrpc	rpcbind.error	string	true
sunrpc	rpcbind.programs	string	true
sunrpc	rpcbind.versionRange	string	true
syslog	host.name	string	true
tcp	banner	string	true
tcp	data	string (hex-encoded)	true
tcp	tcp.flags	string	false
tcp	tcp.options	numeric	true
tcp	tcp.ts	numeric	true
tcp	tcp.urg	numeric	true
tcp	tcp.win	numeric	true
tcpmux	banner	string	true
teamviewer	teamviewer.response	string (hex-encoded)	true
telnet	banner	string	true
tftp	tftp.error	numeric	true

tftp	tftp.mode	string	true
tftp	tftp.opcode	string	true
time	time.skew	string	true
time	time.skewMS	numeric	true
time	time.timestamp	string	true
time	time.value	numeric	true
tls	tls.authorityKeyID	string (hex-encoded)	true
tls	tls.caUnknown	string	true
tls	tls.certificate	string	true
tls	tls.certificates	string	true
tls	tls.cipher	string	true
tls	tls.cipherName	string	true
tls	tls.cn	string	true
tls	tls.crl	string	false
tls	tls.emails	string	false
tls	tls.expired	string	true
tls	tls.fp.bkhash	string	true
tls	tls.fp.caSha1	string	false
tls	tls.fp.sha1	string	true
tls	tls.fp.sha256	string	true
tls	tls.issuer	string	true
tls	tls.issuingURL	string	false
tls	tls.names	string	false
tls	tls.notAfter	string	true
tls	tls.notAfterTS	numeric	true
tls	tls.notBefore	string	true
tls	tls.notBeforeTS	numeric	true
tls	tls.ocsp	string	false
tls	tls.requiresClientCertificate	string	true
tls	tls.selfSigned	string	true
tls	tls.serial	string	true

tls	tls.signatureAlgorithm	string	true
tls	tls.subject	string	true
tls	tls.subjectKeyID	string (hex-encoded)	true
tls	tls.uri	string	false
tls	tls.version	string	true
tls	tls.versionName	string	true
ubnt	ubnt.addrs	string	false
ubnt	ubnt.configStatus	string	true
ubnt	ubnt.directConnectDomain	string	true
ubnt	ubnt.essid	string	true
ubnt	ubnt.firmware	string	true
ubnt	ubnt.hostName	string	true
ubnt	ubnt.interfaceMap	string	false
ubnt	ubnt.macs	string	false
ubnt	ubnt.modelFull	string	true
ubnt	ubnt.modelShort	string	true
ubnt	ubnt.protoVersion	string	true
ubnt	ubnt.sourceMAC	string	false
ubnt	ubnt.unifiVersion	string	true
ubnt	ubnt.uptime	string	true
ubnt	ubnt.webMgmtPort	numeric	true
ubnt	ubnt.webMgmtTLS	string	true
ubnt	ubnt.wmode	string	false
upnp	upnp.controlURL	string	true
upnp	upnp.deviceType	string	true
upnp	upnp.eventSubURL	string	true
upnp	upnp.friendlyName	string	true
upnp	upnp.manufacturer	string	true
upnp	upnp.manufacturerURL	string	true
upnp	upnp.modelDescription	string	true
upnp	upnp.modelName	string	true

upnp	upnp.modelNumber	string	true
upnp	upnp.modelURL	string	true
upnp	upnp.presentationURL	string	true
upnp	upnp.scpdURL	string	true
upnp	upnp.serialNumber	string	true
upnp	upnp.udn	string	true
upnp	upnp.upc	string	true
upnp	upnp.url	string	true
upnp	upnp.urlBase	string	true
upnp	upnp.wifiMac	string	true
upnp	upnp.wiredMac	string	true
uscan	uscan.makeAndModel	string	true
uscan	uscan.manufacturer	string	true
uscan	uscan.serialNumber	string	true
uscan	uscan.version	string	true
vault	vault.version	string	true
vnc	vnc.version	string	true
wbsm	wbsm.active	string	false
webmin	webmin.port	numeric	true
webmin	webmin.scheme	string	true
webmin	webmin.server	string	true
wsd	wsd.addrs	string	false
wsd	wsd.types	string	true
wsman	wsman.body	string	true
xdmcp	xdmcp.address	string	true
xdmcp	xdmcp.hostname	string	true
xdmcp	xdmcp.manufacturer	string	true
xdmcp	xdmcp.port	string	true
xdmcp	xdmcp.status	string	true
zabbix-agent	zabbix.agentVersion	string	true

zookeeper	`zk.access`	string	true
zyxel	`zyxel.builddate`	string	true
zyxel	`zyxel.dhcpstate`	string	true
zyxel	`zyxel.firmware`	string	true
zyxel	`zyxel.firstlogin`	string	true
zyxel	`zyxel.gateway`	string	true
zyxel	`zyxel.hostname`	string	true
zyxel	`zyxel.ip`	string	true
zyxel	`zyxel.mac`	string	true
zyxel	`zyxel.maxport`	string	true
zyxel	`zyxel.model`	string	true
zyxel	`zyxel.subnetmask`	string	true
zyxel	`zyxel.uptime`	string	true

Release notes

Recent runZero release notes

4.0.250701.0

`2025-07-01`

- Fingerprint improvements.

4.0.250630.0

`2025-06-30`

- The “delete stale” integration option description for the AWS and Wiz integrations has been revised to clarify that it enables the deletion of all AWS or Wiz assets not seen by the currently running task.
- An issue that caused `last_seen` attributes to show up in event templates as a number instead of a date has been resolved.
- An issue that prevented community users from creating an organization when no organizations exists has been resolved.
- An issue that caused metrics analysis tasks to be repetitively scheduled in a loop has been resolved.
- An issue that caused slow queries to show up as errors in task logs has been resolved.

4.0.250627.1

`2025-06-27`

- An issue that could result in some metrics tasks containing many “duplicate query in metric data” errors has been resolved.

4.0.250627.0

`2025-06-27`

- An issue that prevented downloading the self-hosted version of runZero in the EU region has been resolved.
- The metrics calculation process has been improved.
- Discovery of embedded IP-to-serial devices has been improved.
- Fingerprint improvements.

4.0.250626.0

2025-06-26

- Email Alerts now have improved support for JSON-formatted attachments.
- An issue loading dashboards with header widgets has been resolved.
- An issue preventing asset information from appearing in the RFC1918 report in some cases has been resolved.
- An issue causing Tenable.io integration tasks to incorrectly import INFO level vulnerabilties when configured to omit importing all vulnerabilities (i.e. “fingerprint-only”) has been resolved.
- Phantom device detection has been improved.
- Fingerprint improvements.

4.0.250625.0

2025-06-25

- An issue causing projects to not correctly analyze vulnerabilities, findings, and query match counts has been resolved.

4.0.250623.1

2025-06-23

- An issue that could cause some scans to fail when scanning certain devices has been resolved.

4.0.250623.0

2025-06-23

- An issue that sometimes prevented Microsoft Intune tasks from completing has been resolved.
- Issues that could result in certain printers experiencing issues during a runZero scan have been resolved.
- An issue that could cause tasks to hang at 99% or fail with error task lost to explorer restart in certain uncommon situations has been resolved.
- An issue that prevented stale vulnerabilities from being deleted in certain circumstances has been resolved.
- Fingerprint and performance improvements.

4.0.250622.0

2025-06-22

- An issue that caused `agent-offline` events to be sent repeatedly every four hours has been resolved. Only one `agent-offline` event will be sent each time an explorer goes offline.

4.0.250620.0

2025-06-20

- The NetBox integration now supports filtering by site names and CIDRs.
- First-page loading speed for software, software groups, vulnerability groups, and findings instances tables has been improved.
- An issue that caused some software records to be duplicated has been resolved.
- An issue that could cause metric and vulnerability tasks to error has been resolved.
- An issue that could prevent sending an invitation email to new users has been resolved.
- An issue that caused `agent-offline` events to be sent even though explorers reconnected has been resolved.
- An issue that caused scan task tags to be applied incorrectly in some cases has been resolved.
- Merge logic improvements.
- Fingerprint improvements.

4.0.250616.0

2025-06-16

- The goal details page has been redesigned to provide more information on goal progress.
- The goal creation workflow UI has been redesigned to streamline the process of editing and creating goals.
- Users with an inherited or explicit role of User or above within the selected organization can now manage goals, consistent with the functionality in alert rules, channels and templates.
- Goals that were previously configured to be “global” have been updated to apply to all currently-existing organizations and will no longer be associated with new or future organizations unless explicitly configured as such, consistent with the functionality in alert rules, channels and templates.
- Task statistics now display total count of software and vulnerability records that were created.
- An issue that caused the display name for the default asset ownership type to be shown as a nil UUID has been fixed.
- Invalid device detection improvements.
- Fingerprint improvements.

4.0.250611.0

2025-06-11

- Operating System End of Life (EOL) information for Microsoft Windows LTSC has been improved.
- Data retention can now be configured per site.
- Wiz integration tasks now have an option to delete stale Wiz assets after each sync.
- The Tenable.io integration has new separate options for disabling vulnerability import and disabling software import.
- Asset search has been updated to support filtering by missing `ownership_type` and missing `mac_countries`.
- When taking screenshots, Chrome is launched with the `--disable-breakpad` option.
- An issue with the Meraki integration that caused First Seen and Last Seen to be set incorrectly has been resolved.
- An issue that caused incorrect display of `ownership_type` information on dashboards has been resolved.
- Merge logic improvements.
- Fingerprint improvements.

4.0.250610.0

2025-06-10

- An issue that could cause scans to stall when interrogating Microsoft SQL Server using TDS 8.0 has been resolved.
- An issue that could lead to out-of-memory conditions on self-hosted consoles with large connector tasks has been resolved.
- Fingerprint improvements.

4.0.250606.1

2025-06-06

- An issue that could cause finding generation to fail for sites in certain situations has been resolved.
- Fingerprint improvements.

4.0.250606.0

2025-06-06

- runZero scans can now check for default logins, you can find details in our documentation.
- The Prisma integration has been updated to support importing assets from AWS.
- Vulnerability displays will no longer show empty information cards when no further information is available.

- Performance improvements for certificate inventory searches, particularly for subject, authority and SAN DNS names.
- Merge logic improvements.
- Fingerprint improvements.

4.0.250604.1

`2025-06-04`

- Security: A bug that could allow MFA bypass during the password reset process has been resolved. This issue was identified internally and did not affect users who authenticate through SSO.
- An issue that prevented the `mac_countries` keyword from matching correctly has been resolved.

4.0.250604.0

`2025-06-04`

- TOTP is now supported as a new MFA option in addition to existing WebAuthn and Passkey support.
- Scan tasks created via templates now carry over the `site:scope` keyword from the template.
- An issue that caused dashboard widgets to display incorrectly when no data is available has been resolved.
- An issue that prevented screenshots from being captured for some services has been resolved.
- Passive sampling tasks that are interrupted by active scans are no longer shown as failed.
- Link-local IPv6 and APIPA IPv4 addresses are no longer shown first in the addresses column of the inventory views.
- MAC addresses with the LAA bit set are no longer resolved to OUI vendors when the source is known to use random values.
- MAC addresses from virtual machine prefixes no longer assert a MAC Country field.
- Merge logic improvements.
- Fingerprint improvements.

4.0.250530.0

`2025-05-30`

- The Vulnerabilities Inventory now includes a column for any associated Finding.
- The confirmation dialog for deleting an Explorer now indicates if tasks will be affected, with a link to view any affected tasks.
- The Asset Inventory now provides a `MAC Countries` column that lists which countries are associated with device MAC addresses.
- The Asset Attributes now include values for `mac.mfgCountries` and `mac.mfgAddresses`.

- The Asset Inventory now accepts searches by MAC Country using the syntax `mac.mfgCountries:US`.
- An issue that caused some integration data not to expire according to organization settings has been resolved.
- View-only users are no longer shown a `Share` action when this functionality is not available to them.
- Outdated browsers will now receive a warning during the sign-in process.
- Tag values are now fully UTF-8 safe and round-trip correctly via tasks.
- Fingerprint improvements.

4.0.250529.1

`2025-05-29`

- The runZero console's content security policy header has been temporarily relaxed to resolve an issue where FireFox ESR would refuse to load icon images.
- A banner will now notify superusers when the SAML certificate being used for single sign-on (SSO) will expire soon.
- An issue that prevented PII attributes from being excluded when integrations were configured to run on an explorer has been resolved.
- Performance improvements.
- Fingerprint improvements.

4.0.250529.0

`2025-05-29`

- Fingerprint improvements.

4.0.250527.0

`2025-05-27`

- TLS certificates no longer used by any service are now automatically deleted from your certificate inventory.
- The design of the progress bar has been updated.
- An issue preventing tables on the risk management dashboard from sorting has been resolved.
- An issue that caused the GCP integration to log errors instead of warnings when some projects didn't have certain features enabled has been resolved.
- An issue causing invalid country flag display in the certificate inventory has been fixed.
- An issue that caused some Azure integrations to fail has been resolved.
- An issue where the error was not reported when scan results failed to upload has been fixed.
- Identification of progressive web applications has been improved.
- Phantom device detection improved.
- Merge logic improvements.
- Fingerprint improvements.

4.0.250524.0

2025-05-24

- An issue that could cause CrowdStrike tasks to fail has been resolved.

4.0.250521.0

2025-05-21

- An issue that was causing findings to not be filtered by asset when following a link from the assets details page has been resolved.
- An issue that was allowing CrowdStrike auth tokens to expire has been resolved.
- An issue that could prevent stale integration attributes from being cleaned up according to organization settings has been resolved.
- The Prisma integration has been updated to fix incorrect request methods and improve logging.
- Fingerprint improvements.

4.0.250516.0

2025-05-16

- Normalization of the software version information from the CrowdStrike integration has been improved.
- Detection of web-interception mechanisms has been improved.
- An issue that could cause integration tasks to merge assets incorrectly in some cases has been resolved.
- An issue that was causing user-set asset criticality values to be overwritten as “Unset” has been resolved.
- An issue that caused Wiz integration tasks run on explorers to fail when importing a large number of assets has been resolved.
- An issue that prevented referencing `organization.name` in event templates was resolved.
- An issue that could cause alert rules to save the current organization in their scope unexpectedly when editing a rule has been resolved.
- Fingerprint improvements.

4.0.250514.0

2025-05-14

- Custom integration scripts can now utilize gzip compression and decompression.
- An issue that caused undesired center justification in table columns has been resolved.
- An issue preventing Alerting Rules from being edited after creation has been resolved.
- An issue that could allow integration tasks to merge assets incorrectly in some cases has been resolved.

- An issue that caused intermittently inaccurate risk levels for “Top findings” in the Risk Management dashboard has been resolved.
- An issue that caused the names of risk levels to not be capitalized on the Risk Management Dashboard has been resolved.
- An issue that caused assets to be recreated when importing AWS assets with the `Automatically delete stale AWS assets` option enabled has been resolved.
- Fingerprint improvements.

4.0.250513.0

`2025-05-13`

- User API endpoints now include information about default, assigned, and effective roles.
- The Microsoft 365 Defender integration now imports the `avMode` attribute and only sets the EDR name when `avMode` is Active.
- The Qualys integration now allows filtering assets by Network IDs.
- Custom integration scripts can now call crypto hashing functions including `sha256`, `sha512`, `sha1` and `md5`.
- Custom integration scripts can now export asset data to a json.gz file that can be imported into a runZero organization.
- Dashboards now include a footer showing the last time data used in the dashboard was updated.
- An issue that resulted in inaccurate information on the finding details page’s Overview section when viewing “My organizations” has been resolved.
- An issue that prevented recalculating metrics in “My Organizations” mode has been resolved.
- An issue that prevented some Risk Management dashboard widgets from displaying results in “My Organizations” mode has been resolved.
- An issue that could cause passive-scanned assets to lose IP addresses when merging with integration assets has been resolved.
- An issue that prevented importing Asset CSVs containing spaces in owner values has been resolved.
- Fingerprint improvements.

4.0.250508.0

`2025-05-08`

- The custom integration script editor now includes autocomplete hints for Starlark and runZero-provided libraries.
- Performance of the certificate inventory has been improved.
- An issue that could cause content updates to fail for some self-hosted customers has been resolved.
- An issue that could prevent integration data from updating correctly in some circumstances has been resolved.
- An issue that limited the number of applications collected from the SentinelOne integration has been resolved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.250506.0

2025-05-06

- An issue preventing the deletion and updating of Explorers from the details view Manage menu has been resolved.
- An issue preventing the finding details view from displaying related runZero blog references has been resolved.
- The vulnerability details view now includes links to runZero Rapid Response posts as well as related references.
- An issue that could cause the network bridges and asset route pathing reports to cut off parts of the graph has been resolved.
- An issue that could cause some integration attributes to be dropped when merging assets has been resolved.
- Fingerprint improvements.

4.0.250505.0

2025-05-05

- Personally identifiable information (PII) or other attributes may now be excluded from data collection for some integrations. See our excluding integration attributes documentation for more details.
- The vulnerability collection performance of the SentinelOne integration has been improved.
- Custom integration scripts now consider a `None` return value as success.
- Custom integration scripts can now use limited `session`-like functionality for making HTTP requests.
- Custom integration scripts can now use base64 encoding and decoding functions.
- The npcap version is now listed on the Explorer details page when the Explorer is installed on Windows.
- The npcap installers have been updated to version 1.82.
- An issue that prevented errors from being returned when exporting data has been resolved.
- An issue that caused CVE overlay data (for example, KEV membership) to not be reflected accurately has been resolved.
- Fingerprint improvements.

4.0.250430.0

2025-04-30

- The SentinelOne integration has been updated to improve performance.
- An issue that prevented exporting vulnerabilities has been resolved.
- Fingerprint improvements.

4.0.250428.0

2025-04-28

- Two new keywords have been added to the scan configuration discovery and exclude scope fields: `site:scope` which expands to the default site scope, and `site:exclusions` which expands to the exclusions set in the site configuration.
- The Tenable Integration Task form has been updated.
- Event log performance was improved.
- Phantom device detection improvements.
- An issue that prevented persistence of inventory table preferences has been resolved.
- An issue that could prevent NetBox assets from merging on IP address has been resolved.
- An issue that could prevent alert channel details from loading in some situations has been resolved.
- An issue that prevented searching Integration attribute data using wildcards in certain cases has been resolved.
- Fingerprint improvements.

4.0.250422.0

2025-04-22

- An issue that resulted in duplicate `assets-expired` events in the event log has been resolved.
- Fingerprint improvements.

4.0.250421.0

2025-04-21

- An issue resulting in reporting an incorrect OS version for fingerprinting of CrowdStrike integration data has been resolved.
- An issue leading to duplicate asset icons and screenshots has been resolved.
- Fingerprint improvements.

4.0.250418.0

2025-04-18

- The SentinelOne integration has been updated to support importing vulnerabilities.
- An issue causing the “Findings by category” header on the runZero Risk dashboard to render incorrectly has been resolved.
- An issue that caused vulnerabilities without categories to display details incorrectly has been resolved.
- An issue that caused the directory users and groups to not be linked for Google Workspace, Azure AD, and LDAP has been resolved.

- An issue that could cause an explorer to be marked as inactive after a reinstall has been resolved.
- User interface improvements.
- Fingerprint improvements.

4.0.250415.0

2025-04-15

- CSV export of certificates was improved for better compatibility with spreadsheet software.
- The findings export now includes the fields `instance_count` and `risk_rank_value`. The fields `vulnerability_count` and `risk_score` have been removed, and the field `risk_rank` now shows the risk label.
- The NetBox integration no longer filters assets that are older than the Organization's stale asset threshold.
- Minor UX enhancement to the Certificate Details page.
- An issue causing the "Internet accessible assets" widget on the Risk Management dashboard to display an incorrect current count has been resolved.
- Fingerprint improvements.

4.0.250414.0

2025-04-14

- Searching the asset inventory for software now supports less-than or greater-than software version queries. See the Asset Inventory search keywords documentation for more information.
- Certificates inventory can now be searched by `last_seen`, `valid_from`, and `valid_until` keywords.
- The API endpoints for `/account/users` and `/account/users/{user_id}` now include the names and IDs of groups that users are members of.
- The inventory export APIs now support gzip Content-Encoding compression when requested via the Accept-Encoding header.
- An issue causing the "High risk findings" widget on the Risk Management dashboard to display an incorrect current count has been resolved.
- An issue causing the Queries list to display incorrect finding codes has been resolved.
- An issue causing duplicate `assets-expired` events to be written to the audit log has been resolved.
- An issue where servers were incorrectly identified as printers in rare cases has been resolved.
- Asset discovery improvements.
- Fingerprint improvements.

4.0.250410.1

2025-04-10

- An issue that could cause importing of vulnerability information from Microsoft Defender to fail has been resolved.
- Fingerprint improvements.

4.0.250410.0

2025-04-10

- The Microsoft 365 Defender integration now has task options to filter importing vulnerabilities by severity.
- The Microsoft 365 Defender integration can now optionally exclude importing software and vulnerabilities.
- The Rapid Response dashboard widget has been updated to clarify when assets are potentially impacted vs actually impacted.
- An issue that caused assets with no vulnerabilities or findings to be shown as Info risk instead of None has been resolved.
- An issue that prevented the Microsoft 365 Defender integration from pulling software without CPEs has been resolved.
- An issue that could cause some vulnerabilities to disappear after successive Microsoft 365 Defender tasks has been resolved.
- An issue that prevented importing certain assets from Wiz has been resolved.
- An issue that could cause integration attributes to be removed from assets has been resolved.
- In accordance with feedback from our annual security audit, a low-severity improvement has been made to the runZero Console's content security policy (CSP).
- Asset merging improvements.

4.0.250409.0

2025-04-09

- The NetBox integration now handles virtual machines, interfaces, and clusters more consistently.
- The NetBox integration now uses fuzzy matching for OS and HW mappings.
- The alerts rules page data table now displays columns for channel and template.
- The alert rule details page now displays the channel and template attributes.
- An issue that could cause errors when gathering screenshots has been resolved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.250407.0

2025-04-07

- Performance of the Asset Details page has been improved.

- An issue that could cause tasks assigned to hosted explorers to be delayed has been resolved.

4.0.250405.0

2025-04-05

- Fingerprint improvements.

4.0.250404.1

2025-04-04

- An issue that could result in printers printing garbage output when their IPv6 addresses are scanned has been resolved.

4.0.250404.0

2025-04-04

- The Microsoft 365 Defender integration has been updated to support pulling software and vulnerabilities.
- New public APIs for findings and certificates inventory are now available. See the API documentation for more information.
- An issue that failed to detect and report encrypted protocols running on non-standard ports has been resolved.
- An issue that caused a page reload when navigating to a Certificate Details page has been resolved.
- An issue that prevented searching the Vulnerability Inventory by Asset for CVEs has been resolved.
- Fingerprint improvements.

4.0.250402.0

2025-04-02

- The NetBox integration now supports importing virtual machines and asset criticality, along with other fixes.
- An issue preventing an icon from showing on the Certificate Details page was resolved.
- Fingerprint improvements.

4.0.250401.0

2025-04-01

- CSRF protection was improved.

- User Public API endpoints now include a field, `mfa_enabled`, indicating if the user has enabled and is required to use MFA to log in.
- An issue preventing the scanner from fingerprinting services on sensitive ports has been resolved.
- An issue that caused some browsers to freeze when viewing the Risk Management dashboard has been resolved.
- Fingerprint improvements.

4.0.250331.0

`2025-03-31`

- Performance of the Risk Management dashboard has been improved.
- An issue that caused the VMware ESXi OS version information to be truncated on some assets has been resolved.

4.0.250330.0

`2025-03-30`

- An issue that could result in excessive logging with misconfigured Windows adapters has been resolved.

4.0.250329.0

`2025-03-29`

- An issue that prevented services from populating in certain circumstances has been resolved.

4.0.250328.0

`2025-03-28`

- Users can now create custom widgets for their dashboards with text of their choosing, rendered as a subset of Markdown/CommonMark.
- The collection performance of the Google Workspace integration has been improved.
- Vulnerability details are now displayed in their own page with a linkable URL.
- An issue that could incorrectly remove integration attributes has been resolved.
- An issue that incorrectly tracked a VMware ESXi asset's full name attribute has been resolved.
- An issue that caused the findings page to take a long time to load has been resolved.
- An issue that could cause console errors when searching for certificates using numeric values has been resolved.
- An issue that resulted in errors when searching the directory users and groups by organization ID has been resolved.
- An issue that prevented Info-level findings from displaying in the Findings overview dashboard widget has been resolved.

- An issue that caused incorrect “What’s Changed” values on the Risk Management dashboard has been resolved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.250325.0

2025-03-25

- The code for detecting self-signed certificates was improved for better accuracy.
- The collection performance of the Google Workspace integration has been improved.
- A bug that resulted in duplicate findings listed for a single organization has been resolved.
- Fingerprint improvements.

4.0.250324.1

2025-03-24

- An issue resulting reporting an incorrect source for fingerprinting of CrowdStrike integration data has been resolved.
- A bug resulting in scans triggering printer output has been resolved.
- Fingerprint improvements.

4.0.250324.0

2025-03-24

- Introducing Risk Findings, a new feature that provides a comprehensive view of risk. Findings group vulnerabilities, misconfigurations and best practices into a curated list of actionable items, allowing you to prioritize and remediate the most critical risk in your environment. To learn more, see our Risk Findings documentation.
- Introducing the Risk Management dashboard, a new dashboard showing an overview of risk in your environment.
- Introducing Certificates Inventory, a new inventory type that allows you to quickly view and search all of the TLS and SSH certificates in your environment. To learn more, see our Certificate Inventory documentation.
- The “None” Risk Rank has been renamed to “Info”.
- SentinelOne integration now processes hostnames with spaces correctly.
- Custom integrations now process hostnames with spaces correctly. Multiple hostnames must be separated by a tab character.
- An issue that resulted in duplicate records for some vulnerabilities has been resolved.
- Fingerprint improvements.

4.0.250317.0

2025-03-17

- A dashboard list page has been added, showing all dashboards the current user has access to.
- Added support for passing an optional `timeout=<seconds>` parameter to `http.get` and `http.post` requests in custom integration scripts.
- Added support for using `HEAD`, `PATCH`, `PUT`, and `DELETE` HTTP methods in custom integration scripts.
- Added support for searching Meraki `firstSeen` and `lastSeen` attributes as timestamps.
- Added new `snmp.interfaceAddrsMap`, `snmp.interfaceAliasesMap`, `snmp.interfaceNamesMap`, and `snmp.interfaceMacsMap` attributes containing SNMP network data indexed by interface index.
- Users will now be shown a helpful message if they cannot login via SSO due to an expired SAML certificate.
- Dashboard widgets now show the date range covered in the widget.
- An issue that caused Rapid7 InsightVM vulnerabilities to not sync the Exploitable flag correctly has been resolved.
- Fingerprint improvements.

4.0.250307.0

2025-03-07

- Several minor graphical issues in the user interface have been resolved.
- An issue preventing filling in additional comments in the query builder feedback menu has been resolved.
- The AWS integration now imports tags for RDS instances.
- The Nessus integration will now continue processing data from other scans, even if an error occurs while ingesting data from one scan.
- Performance improvements.
- Fingerprint improvements.

4.0.250305.0

2025-03-05

- The calculation of the last seen value for Active Directory computers has been improved.
- An issue that caused dashboard widgets to align incorrectly at certain browser sizes has been resolved.
- An issue preventing sorting team user tables by ID has been resolved.
- Fingerprint improvements.

4.0.250303.1

2025-03-03

- Performance improvements for KEV-based vulnerability queries.

4.0.250303.0

2025-03-03

- A bug that prevented self-hosted deployments from the EU SaaS console from installing has been resolved.
- Fingerprint improvements.

4.0.250228.0

2025-02-28

- Merge logic for the Tanium integration when multiple environments are present has been improved.
- IP address collection from the Tanium integration has been improved.
- An issue that resulted in most users on the team datagrid showing as “pending” status has been resolved.
- A bug with host name expansion of scan targets has been resolved.
- Fingerprint improvements.

4.0.250226.0

2025-02-26

- The API now supports setting organization vulnerability expiration configuration parameters.
- An issue that could cause some stale data expiration settings set via the API not to take effect has been resolved.
- HTTP actions in custom integration scripts no longer have a timeout.
- Fingerprint improvements.

4.0.250221.0

2025-02-21

- Performance improvements.
- Fingerprint improvements.

4.0.250219.1

2025-02-19

- An issue that could prevent metrics and query counts from updating has been resolved.
- Fingerprint improvements.

4.0.250219.0

2025-02-19

- runZero scans now record the last time that they detected an asset in the asset attributes.
- Performance improvements.
- Fingerprint improvements.

4.0.250214.0

2025-02-14

- Fingerprint improvements.

4.0.250213.1

2025-02-13

- Self-hosted installations configured with SSO-only logins now automatically redirect to the IdP.

4.0.250213.0

2025-02-13

- An issue in hostname collection that could result in invalid asset hostnames and merges has been resolved.
- An issue preventing drill-down from dashboards' most- and least-seen charts has been resolved.
- The organization API now allows the stale integration attribute setting for an organization to be modified.
- Matching assets from the SentinelOne integration has been improved.
- The `type:` asset search keyword now performs a fuzzy search by default, similar to other search keywords.
- Log events for tasks starting and failing are now labeled with the task name.
- The default HTTP timeout for custom integration script requests has been extended to 5 minutes.
- Performance improvements.
- Asset merging improvements.
- Fingerprint improvements.

4.0.250209.0

2025-02-09

- Performance improvements.
- Fingerprint improvements.

4.0.250208.0

2025-02-08

- Performance improvements.

4.0.250207.3

2025-02-07

- Performance improvements.

4.0.250207.2

2025-02-07

- An issue that could result in an error being displayed when creating a new project for some editions of runZero has been resolved.
- An issue that could cause a metrics recalculation task to issue spurious warnings has been resolved.
- Fingerprint improvements.

4.0.250207.0

2025-02-07

- Organizations now support setting thresholds to automatically expire stale integration and vulnerability data.
- Merge logic for the Tenable Security Center integration has been improved.
- Merge logic for Windows assets with multiple interfaces has been improved.
- The display of errors for query widgets on the dashboard has been improved.
- Filtering of invalid data from the Qualys integration has been improved.
- All available templates are now shown on the scan templates page.
- The APIs used to fetch CrowdStrike applications have been updated to improve collection performance.
- An issue that caused stale protocols to remain on services has been resolved.
- An issue that prevented services from different vhosts on the same IP/port/protocol combination from being displayed on the asset details page has been resolved.
- An issue that prevented seeing the full asset comment in the asset datagrid has been resolved.
- An issue that could cause invalid Steam protocol probe responses has been resolved.

- An issue that could cause higher-than-expected asset risk for certain SSL-related vulnerabilities has been resolved.
- Fingerprint improvements.

4.0.250203.1

2025-02-03

- An issue that could cause scan processing to fail with an error has been fixed.
- An issue that prevented the retrieval of Tanium vulnerability data after the paging data limit is exceeded has been resolved.
- Fingerprint improvements.

4.0.250203.0

2025-02-03

- An issue that caused missing hostname/IP combinations for some assets on the Switch Topology report has been resolved.
- An issue that caused runZero to report SMB v2 as available on certain Samba configurations has been resolved.
- An issue that caused hosted zone tasks to get stuck in a Scheduled state has been resolved.
- An issue that prevented running scans using a hosted zone for some community users has been resolved.
- An issue that sometimes caused task details to show a negative task duration has been resolved.
- Fingerprint improvements.

4.0.250130.1

2025-01-30

- A bug preventing specific organization administrators from modifying asset tags has been resolved.
- A bug that prevented using the quick-bookmark buttons on the reports pages has been resolved.
- The quick-bookmark buttons on the reports pages no longer indicate whether or not they’re already bookmarked.
- Merge avoidance logic for certain integration combinations has been improved.
- Asset merging improvements.

4.0.250130.0

2025-01-30

- A bug that caused dashboard duplication and creation to work incorrectly has been resolved.

- A bug preventing filtering in the dashboard share menu has been resolved.
- Fingerprint improvements.

4.0.250129.0

2025-01-29

- Users can now create multiple dashboards, share them to organizations in which they have User privileges or higher, and set a preferred dashboard in their profile settings. Any personal or runZero managed dashboard can be selected as a preferred dashboard.
- Organization administrators can now set a default dashboard on a per-organization basis. Any dashboard shared to the organization or any runZero managed dashboard can be selected as a default dashboard.
- An issue that prevented query links on dashboards from respecting the “Search live assets” attribute has been resolved.
- An issue that prevented the change report from being visible on the Task Details page has been resolved.
- An issue preventing the scanner from collecting arp cache data from Palo Alto Networks devices using self-signed certificates has been resolved.
- Merge logic for assets observed via both Wiz and AWS has been improved.
- Fingerprint improvements.

4.0.250127.0

2025-01-27

- Self-hosted instances now check for updated content and queries every 5 minutes in online mode.
- Recurring tasks can now be set to a multiple of minutes.
- An issue that caused runZero to report SMB v1 as available on certain Samba configurations has been resolved.
- An issue that prevented Explorer-run InsightVM tasks from running with certain self-signed certificates has been resolved.
- An issue that prevented some integrations from working when a non-standard port number was specified has been resolved.
- An issue that prevented runZero from connecting to InsightVM installations that use a TLS certificate with a negative serial number has been resolved.
- An issue that prevented certain Crowdstrike vulnerabilities from being associated with an asset has been resolved.
- Fingerprint improvements.

4.0.250124.0

2025-01-24

- If site subnets have been defined, newly created scans will target the dynamic “defaults” scope.

- The custom integration script editor is now resizable.
- An issue that prevented connecting to some versions of InsightVM has been resolved.
- An issue that was causing broken links in the Switch Topology Report has been resolved.
- Merge avoidance logic for certain RDP related corner cases has been improved.
- Fingerprint improvements.

4.0.250123.0

2025-01-23

- An issue preventing the Meraki integration from retrying requests has been resolved.
- A bug that would cause an asset's extra addresses to be missing has been resolved.
- Operating System End of Life (EoL) coverage and accuracy has been improved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.250122.0

2025-01-22

- The iSCSI protocol is now supported for asset discovery.
- Reporting of Meraki integration errors has been improved.
- An issue that caused some dashboard widgets to not update properly has been resolved.
- An issue that prevented custom dashboard widgets displaying system queries to non-admin users has been resolved.
- An issue that prevented Palo Alto Networks credentials from appearing in the scan configuration has been resolved.
- Fingerprint improvements.

This release also includes the following fixes for low-severity findings from our annual third-party source code audit and security assessment:

- New password hashes, login tokens, reset password tokens, and new account invite tokens are now stored using the argon2id one-way hashing algorithm. Prior to this release, hashes were generated using the bcrypt hashing algorithm.

4.0.250120.0

2025-01-20

- An issue that could result in inaccurate metrics display has been resolved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.250117.0

2025-01-17

- Operating System End of Life (EoL) coverage and accuracy has been improved.
- Fingerprint improvements.

4.0.250116.0

2025-01-16

- Layer 2 topology calculations have been improved.
- The task details page has been improved.
- The asset details page now includes the date the asset record was created in the runZero database. This information is also available via a new optional column in the asset inventory.
- Intune integration performance has been improved.
- Operating System End of Life (EoL) information is now available for Linux Mint.
- Credentials are now allowed to be re-used across multiple recurring tasks. runZero still recommends limiting credentials to a single recurring tasks in most situations to avoid duplicate asset ingestion.
- An issue that could result in inaccurate query metric representation on the dashboard has been resolved.
- An issue that prevented the Meraki integration from paginating Meraki resources has been resolved.
- An issue that caused the `NO_PROXY` environment variable to be ignored on self-hosted consoles has been resolved.
- Fingerprint improvements.

This release also includes the following fixes for low-severity findings from our annual third-party source code audit and security assessment:

- An issue that allowed Explorer information to be listed across organizations within the same tenant if the requester had knowledge of the Explorer's ID has been resolved.

4.0.250113.0

2025-01-13

- Vulnerability reporting from the Inside Out Attack Surface Management feature is more accurate and adjusted for severity based on the type of exposure.
- Fingerprinting devices via the Matter IoT protocol is now supported.
- The Service Location Protocol (SLP) is now supported for device probing.
- The Tenable integration now records MAC addresses even if they don't have an associated IP address.
- Unmapped MACs are now grouped by interface in the Layer2 information section of the asset details page.

- A flatten_json module with a flatten method can be used when authoring Custom Integration Scripts.
- An issue that prevented organization roles from being saved when creating or updating a group via the API has been resolved.
- An issue that prevented ingesting some assets from Tanium has been resolved.
- An issue that impacted the ability to retry timed-out requests in some connectors has been resolved.
- An issue that could cause a task to repeatedly retry when the task data was improperly formatted has been resolved.
- An issue that prevented setting some asset values in custom integration scripts has been resolved.
- An issue that prevented selecting 'no parent' when editing a project with a consulting license has been resolved.
- Fingerprint improvements.

4.0.250106.0

`2025-01-06`

- Custom Integration Scripts can now run directly on runZero Explorers and be triggered by runZero tasks. To learn more, see our custom integration scripts documentation.
- Vulnerability records are now created for potentially exposed internal assets (Inside Out Attack Surface Management) and misuse of shared encryption keys.
- AWS integration task configuration forms have a new look and feel.
- An issue that could result in a scanning deadlock when using maximum scan durations has been resolved.
- Fingerprint improvements.

4.0.241223.0

`2024-12-23`

- An issue that would cause devices discovered by the Tenable integration to not properly merge has been resolved.
- An issue that caused the list of Explorers to not be sorted correctly when configuring alert rules has been resolved.
- Fingerprint improvements.

4.0.241219.2

`2024-12-19`

- The runZero CLI is now available for download for all license tiers. Specific functionality is still based on your license and entitlements.
- Integrations run through an Explorer now use proxy settings in all cases.
- Explorer upgrades now strictly validate versions and update URLs.
- Added export APIs for export tasks.
- Scan tasks created via console now support an optional scan duration limit.

- The Getting Started Guide has been revamped with additional content.
- Intune logging has been improved.
- Custom multi-query widgets' data sources list can now be reordered with drag and drop.
- An issue preventing users from changing their name or email when SSO is required but the user is not enrolled in SSO has been resolved.
- An issue that could cause the alert rule inventory query preview button to unexpectedly URL-encode search strings has been resolved.
- Fingerprint improvements.

4.0.241217.0

`2024-12-17`

- Merge avoidance logic for integration data has been improved.
- An issue that would cause all software for the entire organization to be displayed in the software section of the asset screen has been fixed.
- An issue that caused the "Copy as a new scan template" button to be displayed for tasks that the action is not available for has been resolved.
- An issue where stale IP addresses resolved through DNS were not periodically removed was resolved.
- Fingerprint improvements.

4.0.241213.0

`2024-12-13`

- An issue that would cause exporting software without a filter to fail has been fixed.
- An issue that caused an application error when uploading an invalid IDP metadata.xml in SSO Settings has been fixed.
- Fingerprint improvements.

4.0.241212.0

`2024-12-12`

- Alert rules for inventory query event types now include a button to preview the configured query in the inventory.
- The loading overlay on the data tables throughout the product has been improved for more clarity.
- An issue that prevented the delivery of scan alerts through Slack has been resolved.
- An issue where the alert error tooltip message wasn't being rendered has been resolved.
- An issue preventing the discovery scope field on the task inspection card from appearing has been resolved.
- An issue causing the task inspection card to sometimes take longer than expected to load has been resolved.

- An issue that prevented Wiz connectors from working with Wiz API credentials scoped to specific projects has been resolved.
- Fingerprint improvements.

4.0.241210.0

2024-12-10

- An issue that caused some event rules, channels and templates to be hidden has been resolved.
- An issue that prevented alert rules from saving the query condition has been resolved.
- An issue that prevented event templates, channels, and rules from being removed when an organization is removed has been resolved.
- An issue that prevented some form “Back” buttons from functioning correctly has been resolved.
- Fingerprint improvements.

4.0.241209.1

2024-12-09

- The navigation menus have been redesigned for ease of use. User settings and sign out buttons are now located in the top right of the application.
- Alerts, rules, channels and templates are now scoped to one or more organizations allowing organization-level users to edit alert rules. See our alerts documentation for more information.
- Asset merge avoidance logic for custom integration data has been improved.
- A bug in merge logic for Tenable Security Center data has been resolved.
- Fingerprint improvements.

4.0.241206.0

2024-12-06

- An issue that prevented some forms from functioning correctly has been resolved.

4.0.241205.1

2024-12-05

- An issue that prevented some dashboard drill down pages from displaying has been resolved.

4.0.241205.0

2024-12-05

- Scans now probe Palo Alto Networks firewalls for ARP cache information.
- A bug that allowed configuring the Wiz integration with no API URL has been resolved.
- A bug in hostname collection that could result in invalid asset hostnames and mergers has been resolved.

Our annual third-party source code audit and security assessment is in progress. This release includes fixes for the following issues:

- Content-Security-Policy headers have been made more strict.
- An XSS vulnerability was identified in the Asset Ownership form.
- A few minor weaknesses were identified in the password reset flow.

4.0.241203.0

2024-12-03

- Assets discovered via CIP backplane enumeration are now better displayed.
- Scan discovery scope has been added to the task inspection card on the task overview page.
- Improved discoverability of Fortinet appliances using the FortiGate to FortiManager (FGFM) protocol.
- Detection of bulk responses from Fortinet network filtering and interception products has been improved.
- An issue which delayed sample tasks from starting once a scan completed has been fixed.
- An issue that prevented exporting asset attribute reports for foreign attributes has been fixed.
- An issue that caused Tenable tasks to occasionally ignore their filter settings has been resolved.
- An issue that could cause inconsistent task inspection card state on the task overview page has been resolved.
- An issue that could cause Explorers with identical host IDs to replace Explorers in another organization has been resolved.
- An issue that could cause invalid events to be shown on the events page has been resolved.
- An issue resulting in assets retaining invalid serial numbers from filtered services has been resolved.
- Fingerprint improvements.

4.0.241125.0

2024-11-25

- Fingerprint improvements.

4.0.241123.0

2024-11-23

- A bug that could result in excessive error reporting under low memory conditions has been resolved.
- Fingerprint improvements.

4.0.241122.0

2024-11-22

- Assets with more than 128 ports open are no longer excluded from asset lists.
- The load time of dashboards when assets have many tags has been improved.
- The speed of loading the explorers list has been improved.
- An issue that would cause tasks to report spurious download errors has been corrected.
- An issue that could prevent rDNS names from being assigned as an asset name has been resolved.
- Fingerprint improvements.

4.0.241120.0

2024-11-20

- Tenable merge rules have been refined to reduce duplicate assets.
- Connection-related error messages for the Active Directory (LDAP) integration have been improved.
- Fingerprint improvements.

4.0.241118.0

2024-11-18

- Intune data collection speed has been improved.
- Qualys integration logging has been improved.
- A bug occasionally causing unprocessed sample tasks to overload the task queue has been fixed.
- Fingerprint improvements.

4.0.241114.0

2024-11-14

- runZero now supports the Hikvision SADP protocol.
- Microsoft Azure and Intune connections now complete faster.

- Recent tasks can now be easily reprocessed to take advantage of updates to asset merge logic.
- An issue that could prevent Shodan devices from being merged into existing assets has been resolved.
- An issue that could cause explorers to unregister due to operational issues with runZero's platform has been resolved.
- An issue that caused `api-export` events to be logged as `api-organization` events has been fixed. The `api-export` events generated between versions 4.0.241022.0 and 4.0.241114.0 were logged as `api-organization` events.
- Fingerprint improvements.

4.0.241109.0

`2024-11-09`

- A bug that could prevent Qualys jobs from completing in some cases has been resolved.
- Fingerprint improvements.

4.0.241106.0

`2024-11-06`

- An issue that could cause Assets to have duplicate foreign data attribute sets has been resolved.
- The CLI scanner `--output-raw` option now produces gzipped output and disables output directory creation.
- The CLI scanner now supports the `link4` and `link6` scan targets for local network ranges.
- The CLI scanner help output now omits redundant host-ping/subnet-ping options.
- Fingerprint improvements.

4.0.241101.2

`2024-11-01`

- A bug that could prevent enumeration of buggy TLS ECDH implementations has been resolved.
- The scanner now reports SNMP interface aliases in addition to names.

4.0.241101.1

`2024-11-01`

- An issue that would prevent assets scanned over certain VPNs from merging correctly has been resolved.

4.0.241101.0

2024-11-01

- The event details modal now displays links to source and target objects.
- The events data grid page now includes an Organization column.
- The Tanium integration now retrieves endpoints' Custom Tags when available.
- The switch topology export options have been expanded to include the entire graph.
- IP address ingestion via the CrowdStrike integration has been improved.
- The metrics recalculation actions found on the task overview and dashboard have been improved.
- Fingerprint improvements.

4.0.241029.0

2024-10-29

- An issue that could cause the GCP integration to attempt to retrieve resources from deleted projects has been resolved.
- Fingerprinting of Comtrol IO-Link devices is now supported.
- The FortiGate to FortiManager (FGFM) protocol is now supported for asset discovery.
- Fingerprint improvements.

4.0.241025.0

2024-10-25

- Enhanced the task details page view for recurring tasks.
- Information about whether individual users are required to use SSO is now displayed more clearly.
- An issue involving processing of UTF-8 BOM sequences in CSV files has been resolved.
- An issue causing broken links in the Switch Topology report has been resolved.
- An issue preventing access to the standard query library from the EU region has been resolved.
- An issue that could cause assets with stale service data to be fingerprinted incorrectly has been resolved.
- Fingerprint improvements.

4.0.241023.0

2024-10-23

- Backplane enumeration of OT devices using CIP over EtherNet/IP is now supported.
- CSV exports can now include Unicode characters.
- An issue that caused an error after editing organization settings has been fixed.
- An issue that prevented "SSO Required" login restrictions from being enforced on existing user accounts has been resolved.

4.0.241022.0

2024-10-22

- runZero now supports the creation of multiple export tokens.
- Newly created export tokens now show creation information and allow setting a description.
- Windows binaries are now exclusively signed with our runZero code signing certificate. The old Rumble code signing certificate has been retired.
- The service inventory view “Summary” column has been renamed “Service response” to better represent the data.
- A bug involving use of asset tags in alert templates has been fixed.
- A bug in parsing tags set to have no value has been fixed.
- A bug causing tags to get dropped from event rule data has been fixed.
- A bug in formatting tag changes in the event log has been fixed.
- A bug that prevented very long Explorer names from being fully visible on the Explorer details page has been resolved.
- A bug impacting fingerprinting when an asset had certain integration sources has been resolved.
- A bug in asset hostname collection from integration data has been resolved.
- A bug causing Windows Subsystem for Linux (WSL) guests observed in MS 365 Defender data to be merged with their hosts has been resolved.
- Merge avoidance logic for integration data has been improved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.241016.0

2024-10-16

- An issue causing current organization to be inconsistent when opening links in the console has been resolved.
- An issue causing the task card on the Explorer details page to show tasks from other Explorers when multiple Explorers with the same name are present in the organization has been resolved.
- An issue involving email invites from users with punctuation characters in their names was fixed.
- An issue that prevented exporting vulnerabilities from the UI when filtering by site has been resolved.
- An issue that prevented viewing recurring task details when no subtasks existed has been resolved.
- Fingerprint improvements.

4.0.241015.0

2024-10-15

- runZero now integrates with NetBox.
- Added new duration and average duration columns to the Completed and Recurring task list pages. This allows viewing and sorting tasks by duration.
- Added a quick link to login with SSO for self-hosted installs.
- The dashboard menu now includes an option to recalculate dashboard metrics.
- Individual assets can now be refingerprinted using the latest fingerprint database directly from the asset details page.
- A bug preventing users from being redirected to a newly-created organization or project after creating one has been resolved.
- A bug preventing the “Switch to” button in the organization table from working has been resolved.
- An issue causing invalid asset links in the organization comparison report has been resolved.
- Fingerprint improvements.

4.0.241010.0

2024-10-10

- An issue that prevented logging in via SSO when a first name or last name was missing has been resolved.
- An issue that allowed clicking on disabled project settings has been resolved.
- Fingerprint improvements.

4.0.241009.0

2024-10-09

- The active console region is now displayed on the login page.
- Improved memory efficiency when exporting assets to Splunk via the runZero Splunk Add-on (requires v3.1.0 or greater of the add-on).
- A bug preventing querying for assets with multiple CVE matches from the vulnerability inventory page has been resolved.
- Explorers older than v4.0 have been phased out and can no longer connect to the console.
- Fingerprint improvements.

4.0.241003.0

2024-10-03

- A bug resulting in incorrect Software Inventory population in certain limited situations has been resolved.
- A bug resulting in incorrect asset Type assertions in limited situations has been resolved.
- Fingerprinting of Apple macOS from CrowdStrike data has been improved.

- Asset merging improvements.
- Fingerprint improvements.

4.0.241001.0

`2024-10-01`

- An issue that prevented NOT and OR operators in queries on the site/organization report has been resolved.
- A bug resulting in incorrect Operating System End of Life (EoL) values for Red Hat Enterprise Linux has been resolved.
- A bug that could require some users to enter their email address twice on login has been resolved.
- A new search keyword `first_seen_task` allows searching for assets first seen by a particular task.
- The serial number coverage in the asset CSV export was expanded to include additional protocols and devices.
- Fingerprint improvements.

4.0.240927.0

`2024-09-27`

- Explorer TLS settings are now configurable via `TLS_VERSION_MIN` and `TLS_VERSION_MAX` parameters.
- Software and Vulnerability inventory queries can now be saved to the query library.
- Vulnerability groups now support searching by site ID or site name.
- A bug that prevented the task status icon and associated error/warning logs from updating when selecting different tasks has been resolved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.240926.0

`2024-09-26`

- runZero scans now include the CUPS (IPP) Browser protocol as a new probe on UDP/631.
- A bug that could lead to incorrect matching between Tenable sources has been resolved.
- Any error messages from the SSO process are now prominently displayed.
- Fingerprint improvements.

4.0.240925.0

2024-09-25

- A bug resulting in malformed query when pivoting from grouped vulnerabilities with multiple CVEs has been resolved.
- A bug that resulted in sending invalid JSON in some events that reference organization.id or site.id has been resolved.
- A bug that could cause Wiz connections to report that results were not found even when using correct service account credentials has been resolved.
- Fingerprint improvements.

4.0.240924.1

2024-09-24

- A bug that could lead to an error message in scan logs from short rpcbind replies has been resolved.
- The Site ID and Organization ID fields in event messages are now formatted as strings and not byte arrays.

4.0.240924.0

2024-09-24

- A bug causing single-sign-on to fail with the error “Email address … is already in use” has been resolved.
- A bug preventing the OS CPE value from being displayed in the Asset inventory has been resolved.
- The Oracle Solaris Service Tag protocol is now supported for asset discovery.
- Fingerprint improvements.

4.0.240923.0

2024-09-23

- Introduced a new login screen.
- runZero now integrates with Tanium API Gateway.
- The API now supports the bulk removal of a custom integration source from a list of assets.
- Begin signing Windows binaries with our new runZero, Inc. code signing certificate. We are currently dual signing with the old and new certificates.
- The speed of navigating to subsequent pages in inventory tables has been improved.
- Improved performance of the Wiz integration.
- Minor UI enhancement to better provide event rule errors via tooltip within table.
- An issue preventing event channels from displaying in the Channels list if the user who created them no longer exists has been resolved.
- Upgraded npcap to v1.80.
- A bug that could prevent Wiz vulnerability data from being processed has been resolved.

- A bug in UUID handling in event rules was fixed.
- A bug that prevented importing some Wiz assets that were created more than 180 days ago has been resolved.
- A bug that resulted in incorrect directory user and group membership counts has been resolved.
- The Wiz integration now properly syncs when the Wiz Service Account credential is limited to specific projects.
- Fingerprint improvements.

4.0.240921.0

2024-09-21

- Asset merging improvements.
- Fingerprint improvements.

4.0.240919.0

2024-09-19

- Asset merging improvements.
- Fingerprint improvements.

4.0.240918.0

2024-09-18

- A race condition that could lead to incorrect asset matching has been resolved.
- A bug that could lead to integration attributes not being updated has been resolved.
- A bug that prevented all-organization admins from managing alerts has been resolved.
- The PCWORX protocol is now supported.
- Fingerprint improvements.

4.0.240917.2

2024-09-17

- An issue that could cause Crowdstrike tasks to fail and retry has been fixed.
- Fingerprint improvements.

4.0.240917.1

2024-09-17

- runZero now integrates with Microsoft Endpoint Configuration Manager (MECM).
- The self-hosted platform now supports ARM64 (aarch64) on Linux.
- Imported scan data now reports the correct scan times in the task view.

- CrowdStrike device last seen fields can now be queried as relative timestamps.
- The performance of the CrowdStrike integration has been improved.
- A bug that could prevent self-hosted from installing on newer versions of Alma Linux has been resolved.
- A bug in the display of the access summary of some users has been resolved.
- A bug that prevented querying directory user and group attributes with relative time queries has been fixed.
- Fingerprint improvements.

4.0.240910.2

`2024-09-10`

- An issue that could prevent login link authentication from working has been resolved.
- An issue that left temporary files in Explorer temp directories has been resolved.
- An issue that prevented My Orgs from working with a large number of organizations has been resolved.

4.0.240909.0

`2024-09-09`

- The login process has been redesigned for a smoother user experience.
- An issue that could cause confusing navigation behavior when viewing different organizations in separate browser tabs has been resolved.
- Asset merging improvements.
- Fingerprint improvements.

4.0.240907.0

`2024-09-07`

- Asset correlation has been improved for Meraki, ChromeOS, and SentinelOne sources.
- Fingerprint improvements.

4.0.240904.1

`2024-09-04`

- An issue that could result in tasks that import software records failing has been fixed.

4.0.240904.0

`2024-09-04`

- A bug that could cause daily recurring tasks to incorrectly be scheduled after modification has been resolved.

- Assets can now be identified using the Automatic Tank Gauge protocol.
- Fingerprinting of Dell iDRAC devices has been improved.
- The RFC1918 scan options are now available from the RFC 1918 reports page.
- Asset merging logic has been improved.
- Performance of foreign data integrations has been improved.
- Fingerprint improvements.

4.0.240902.0

2024-09-02

- An issue that could lead to incorrect correlation due to hardcoded device-side MAC addresses has been resolved.
- Bogus network responses for PPTP and FTP services are now ignored.
- Fingerprint improvements.

4.0.240829.0

2024-08-29

- An issue with certain versions of Chrome that could cause the creation of large numbers of temporary files has been fixed.
- A bug that could result in setting an incorrect asset Type based on integration data has been resolved.
- An issue that could cause recurring tasks to create a new subtask when modifying properties other than “Start time” or “Scan frequency” has been resolved.
- Time and date values in searches now support relative times in more cases.
- Improved handling of API request retries for integrations.
- JSON alert templates now render arrays and objects as JSON arrays and JSON objects, without needing to loop through fields or values.
- Fingerprint improvements.

4.0.240826.0

2024-08-26

- A bug that could cause custom integration attributes to be deleted during asset merging has been fixed.
- A bug that could result in large numbers of attributes attached to assets in some situations has been fixed.
- The performance of the CrowdStrike integration has been improved.
- Fingerprint improvements.

4.0.240825.1

2024-08-25

- A bug that could result in integration source attributes not aging out during merges has been resolved.

4.0.240825.0

2024-08-25

- Scan and passive discovery tasks now complete faster for large sites.
- CrowdStrike integration tasks now complete faster.
- Fingerprint improvements.

4.0.240822.0

2024-08-22

- Operating System End of Life (EoL) coverage has been improved for Cisco IOS XE, IBM AIX, Juniper Junos OS, and Palo Alto Networks PAN-OS.
- Integration-source asset processing now avoids matching assets with excessive attribute sets.
- Self-hosted installations now track performance profiles per task automatically.
- The asset inventory now supports the foreign_attribute_count keyword.
- Fingerprint improvements.

4.0.240820.0

2024-08-20

- A new system query for assets past OS Extended End of Life has been added to the library.
- Passive sampling tasks can now identify Avast, Bitdefender, Carbon Black, ESET, Kaspersky, McAfee, SentinelOne, and Trellix AV/EDR products.
- The Alerts page has been redesigned for ease of use.
- Asset merging performance has been improved.
- Fingerprint improvements.

4.0.240817.0

2024-08-17

- A bug that could result in bad matches due to blank foreign IDs has been resolved. Assets that had conflicting source data due to blank foreign ID matching will rebuild as part of normal job processing.

4.0.240816.0

2024-08-16

- The self-hosted installer now supports custom installation and temporary directory paths.
- The self-hosted installer now supports systems with disabled or restricted sudo.
- The self-hosted console now supports text-format logging via the LOG_FORMAT=text configuration parameter.
- Asset merging performance has been improved.
- Fingerprint improvements.

4.0.240814.0

2024-08-14

- The Meraki integration now supports filtering the imported assets by organization name and/or ID.
- The Qualys integration now supports filtering the imported assets by tags.
- The Operating System icons in the Asset Inventory view have been improved.
- License utilization is now available as a percentage on the license information page.
- Directory group CSV exports now include the directory_group_user_count field at the end of the existing column set.
- The Switch topology report has been redesigned for ease of use.
- A bug that could cause multi-homed hosts to be missing links in the Switch topology report has been resolved.
- Fingerprinting logic has been improved so as to better account for certain source combinations.
- Fingerprint improvements.

4.0.240811.0

2024-08-11

- A bug that prevented software vendor searches by prefix with wildcards from working was fixed.
- Fingerprint improvements.

4.0.240809.0

2024-08-09

- The Alert Templates page has been redesigned for ease of use.
- A bug which caused valid JSON event rule templates to be rejected has been fixed.
- A bug causing MAC and IP address mapping information to be dropped from custom integration device data was fixed.
- Fingerprint improvements.

4.0.240807.0

2024-08-07

- The Alert Rules page has been redesigned for ease of use.
- Fingerprint improvements.

4.0.240803.0

2024-08-03

- Azure and GCP subscription IDs are now also stored in the top-level asset attributes.
- Fingerprint improvements.

4.0.240802.0

2024-08-02

- A bug that could prevent the Tenable Security Center from importing data has been resolved.
- The dashboard now supports filtering trending widgets by a customizable date range.
- Improved detection of invalid services.
- Fingerprint improvements.

4.0.240731.1

2024-07-31

- A bug that could lead to HTTP service data ordering being incorrect has been resolved.

4.0.240731.0

2024-07-31

- A bug that could reduce performance of large task processing has been resolved.
- Fingerprint improvements.

4.0.240730.0

2024-07-30

- Fingerprint improvements.

4.0.240729.1

2024-07-29

- A bug that could prevent the CrowdStrike integration from running from an Explorer has been resolved.
- The matching engine for integration-sourced assets is now faster, more accurate, and better at merging related devices.
- SSH enumeration now results in more consistently-named fields.

4.0.240729.0

2024-07-29

- The Meraki integration now supports filtering by VLAN and SSID.
- Fingerprint improvements.

4.0.240727.0

2024-07-27

- A bug that prevented vulnerability group exports from applying the search filter has been resolved.
- SSH enumeration now captures all host keys as well as server extensions.
- Fingerprint improvements.

4.0.240726.0

2024-07-26

- A bug that prevented checkbox states from persisting in some cases has been resolved.
- Fingerprint improvements.

4.0.240725.0

2024-07-25

- Discovery of devices using the TwinCAT ADS protocol is now supported.
- Asset risk, vulnerability, and outlier fields are now available for use in Event templates.
- Temporary directory selection for Explorers has been improved.
- A bug preventing the display of integration data fetch durations has been resolved.
- Fingerprint improvements.

4.0.240722.0

2024-07-22

- A bug that could result in vulnerabilites not being calculated when software entries were not present has been fixed.
- Name-based asset matching has been significantly improved and now uses more sources and trusts PTR records less.
- Fingerprint improvements.

4.0.240719.0

2024-07-19

- A bug regarding the Tenable Security Center integration risk filter has been resolved.
- Merging of VMware assets has been improved.
- Fingerprint improvements.

4.0.240718.0

2024-07-18

- A bug that caused the Goals Overview dashboard widget to display an incorrect number of days worth of data instead of the selected timeframe has been resolved.
- Network topology calculation is now faster and runs as part of the metrics analysis task and not inline with normal task processing.
- Additional Crowdstrike device data is available for users with access to Crowdstrike's Discover API.
- CrowdStrike, InTune, Tenable, and Wiz integrations are now faster at processing large datasets.
- The Asset ID and Organization ID are now shown on their respective details pages.
- Fingerprint improvements.

4.0.240716.0

2024-07-16

- The CLI scanner now correctly supports the `--import-pcap` option.
- Hosts with only some of their addresses excluded will now match existing assets during merge.
- Meraki-connector sourced assets now report the wired-side MAC for better correlation.
- Fingerprint improvements.

4.0.240715.1

`2024-07-15`

- Connectors now use fast-fallback to IPv4 for non-responsive IPv6 endpoints.
- A performance regression with topology calculation has been resolved.
- The Tenable connector now supports filtering by source and tag.

4.0.240715.0

`2024-07-15`

- Fingerprint improvements.

4.0.240712.0

`2024-07-12`

- Performance of the Crowdstrike integration has been improved.
- A bug that prevented inventory table preferences from persisting throughout the product has been resolved.
- Fingerprint improvements.

4.0.240707.0

`2024-07-07`

- Support for searching for assets and vulnerabilities by VulnCheck KEV membership has been added.
- The CrowdStrike integration now retrieves more detailed information.
- An issue that could prevent users with community licenses from initiating hosted scans has been fixed.
- An issue that could cause VMware guest operating systems to be incorrectly fingerprinted has been fixed.
- Fingerprint improvements.

4.0.240702.0

`2024-07-02`

- An issue that could cause asset type to be set to `Desktop` incorrectly has been fixed.
- An issue that could cause certain virtual machine types to not merge properly has been fixed.
- An issue that could cause certain version comparison queries to not be parsed correctly has been fixed.
- Fingerprint improvements.

4.0.240628.0

2024-06-28

- Version fields across the product are now sortable semantically and can be filtered using the operators `>`, `>=`, `<`, `<=`, `=`.
- The Meraki integration now supports filtering on specific networks by name or ID.
- The scanner now supports the Canon BJNP protocol.
- Fingerprint improvements.

4.0.240627.0

2024-06-27

- EPSS scores for vulnerabilities are now searchable with the `epss_score` keyword.
- The vulnerability information page now shows more information about CISA KEV membership and EPSS scores for vulnerabilities that have relevant information.
- The Asset Ownership report now supports up to 15,000 owners at a time.
- Major performance improvements in vulnerability search.
- Fingerprint improvements.

4.0.240626.1

2024-06-26

- The Meraki integration now populates the switch topology report.
- VMware guests will now link correctly when observed between different ESXi servers and vCenter endpoints.
- The Intune integration now supports an optional filter for devices.
- The search option for the Azure AD integration has been deprecated.
- A bug causing custom widgets to drill down into inventory views with an incorrect `alive:t` filter despite the query's configuration has been resolved.
- Fingerprint improvements.

4.0.240622.0

2024-06-22

- A bug that could lead to incomplete MSSQL enumeration has been resolved.
- A bug that could result in the wrong IP address being assigned to a CrowdStrike record has been resolved.
- Fingerprint improvements.

4.0.240621.0

2024-06-21

- A bug in the Organization Overview report has been fixed and the report speed was improved.
- Custom widgets based on queries have been added to the dashboard. Users can create custom widgets from the widget library on the dashboard, or from the query library.
- Improved discovery and data collection from Microsoft SQL Server endpoints.

4.0.240620.0

2024-06-20

- Fingerprint improvements.

4.0.240619.2

2024-06-19

- Improved logging for CrowdStrike connection errors.

4.0.240619.1

2024-06-19

- A bug that could prevent CrowdStrike credentials from successfully validating has been resolved.
- Fingerprint improvements.

4.0.240619.0

2024-06-19

- A bug that could prevent Azure integrations from being created has been resolved.
- Fingerprint improvements.

4.0.240618.0

2024-06-18

- Passive traffic sampling is now more accurate at detecting syslog clients.
- The scanner now supports providing scan options via a JSON formatted configuration file.
- Fingerprint improvements.

4.0.240616.0

2024-06-16

- The Export API endpoints now support POST requests with application/x-www-form-urlencoded parameters. This allows for larger search queries and field filters to be specified.
- Fingerprint improvements.

4.0.240614.0

2024-06-14

- A bug that could result in stalled scans in some situations has been fixed.
- x.509 serial number values in tls.serial will no longer have the leading zero removed.
- Fingerprint improvements.

4.0.240613.0

2024-06-13

- A bug that could prevent non-Windows installations of the runZero Explorer from restarting has been resolved.
- A bug that could result in stale MAC addresses accruing on Tenable assets has been resolved.
- A bug that could result in long timeouts for CrowdStrike tasks with invalid credentials has been resolved.
- A bug that prevented custom integration attribute links from returning results with mix-cased integration names has been resolved.
- Fingerprinting for Azure VMs now prefers the Azure HW assertion over other sources.
- Fingerprint improvements.

4.0.240612.0

2024-06-12

- A bug that could cause the Meraki integration to error has been resolved.
- A bug that could cause incorrect data to display on the dashboard's most and least seen widgets when toggling the view has been resolved.
- Fingerprint improvements.

4.0.240610.0

2024-06-10

- A visual bug making some toggles in the UI appear incorrectly has been resolved.

- A bug that could prevent Intune devices from being synced has been resolved.
- Fingerprint improvements.

4.0.240607.0

2024-06-07

- Improved discovery and data collection from Microsoft SQL Server endpoints.
- Fingerprint improvements.

4.0.240606.1

2024-06-06

- A bug that could cause the Intune integration to skip syncing certain devices has been resolved.

4.0.240606.0

2024-06-06

- A bug that could result in new Explorer installations on Windows not including npcap has been resolved.
- A bug that could result in connector tasks being stuck in “stopping” status has been resolved.
- Users with no access permissions are no longer allowed to view the account’s superusers.
- Organization hierarchies are now supported up to four levels deep.
- Fingerprint improvements.

4.0.240605.0

2024-06-05

- Support for searching for assets and vulnerabilities by CISA KEV membership has been added.
- Performance improvements.
- Fingerprint improvements.

4.0.240603.0

2024-06-03

- The Defender integration now supports filtering assets that have not been fully onboarded.
- The Defender integration now supports the Graph API filter parameter when running as a scanner probe.

- The Events view is no longer limited to the previous 30 days of records.
- The Explorer now uses consistent file names during the upgrade process.
- A bug that prevented the Defender and Intune configuration from validating when specifying a new Azure credential has been resolved.
- Fingerprint improvements.

4.0.240531.0

2024-05-31

- Discovery of devices using the XDMCP protocol is now supported.
- A bug that could cause incorrect OS CPE generation has been resolved.
- OS version information in Fortinet FortiOS CPE values has been improved.
- Operating System End of Life (EoL) information is now available for Fortinet FortiOS.
- Asset merge logic has been improved.
- Fingerprint improvements.

4.0.240530.0

2024-05-30

- A bug that could show a “user not found” error in API-submitted import jobs has been resolved.
- Fingerprint improvements.

4.0.240529.1

2024-05-29

- runZero now integrates with Meraki. This initial support syncs Devices and Clients to your runZero inventory.
- A bug that could result in an “invalid query” message shown in the self-hosted query library has been resolved.
- A bug that could result in incorrect display of Punycode-encoded hostnames has been resolved.
- A bug that could lead to incorrectly assigned MAC addresses due to cross-VLAN mDNS relays in traffic sampling has been resolved.
- A bug that could lead to invalid MAC address attributes from Defender 365 sources has been resolved.
- A bug that could lead to runZero scan results being attached to not-onboarded Defender 365 assets instead of onboarded assets has been resolved.
- A bug that could result in assets being marked as Laptops instead of Desktops has been resolved.
- A bug that could result in multiple passive sampling tasks being scheduled on the same Explorer has been resolved.
- Fingerprint improvements.

4.0.240524.0

2024-05-24

- The dashboard now supports theater/kiosk mode and fullscreen display options.
- The dashboard widget library now includes a customizable bookmarks widget, that can be used to jump to your favorite reports and views in runZero or to external web sites.
- A bug that could prevent users with organization-specific roles from editing asset tags has been resolved.
- Fingerprint improvements.

4.0.240522.0

2024-05-22

- Performance improvements.
- Fingerprint improvements.

4.0.240519.0

2024-05-19

- The domain: scan target keyword now returns substantially more results for most domains.
- The scanner now treats in-scope addresses found by SNMP as primary addresses.
- The scanner no longer adds reflected IP addresses in L2TP hostname responses.
- The scanner no longer merges specific Netgear switches unintentionally.
- The AzureAD (EntraID) connector now supports the $search and $filter parameters for the Microsoft Graph API.
- The LDAP connector now syncs additional fields, including employeeID, ms-Mcs-AdmPwdExpirationTime, and ms-LAPS-PasswordExpirationTime.
- The CrowdStrike connector now provides better OS fingerprinting during multi-source asset processing.
- The Qualys connector is now more resilient with transient network and service timeouts.
- The Qualys connector now prioritizes Agent-based operating system fingerprints.
- The Custom Integration SDK can now ingest ipAddresses, ipAddressesExtra, and macAddresses fields directly without the presence of a NetworkInterface structure.
- A bug that could prevent the Tenable connector from exporting data has been resolved.
- A bug that could result in stale asset attributes after passive discovery has been resolved.
- A bug that could result in stale service summary columns has been resolved.
- Fingerprint improvements.

4.0.240516.0

2024-05-16

- Fingerprint improvements.

4.0.240514.0

2024-05-14

- Filtering of bogus responses, particularly from interception features of Fortinet gear, has been greatly improved.
- Improved logging for Azure and Intune integrations.
- Fingerprint improvements.

4.0.240508.0

2024-05-08

- A bug that could result in unexpected Wiz authentication errors being included in task logs has been resolved.

4.0.240503.0

2024-05-03

- Creating hosted zone scan tasks via API no longer fails if the site has no non-hosted explorers.
- Fingerprint improvements.

4.0.240501.0

2024-05-01

- Fingerprint improvements.

4.0.240429.0

2024-04-29

- Improved handling of large vulnerability results in the CrowdStrike integration.
- Fingerprint improvements.

4.0.240425.0

2024-04-25

- Fingerprint improvements.

4.0.240424.0

2024-04-24

- Fingerprint improvements.

4.0.240423.0

2024-04-23

- A bug that prevented SSO users from setting a password when SSO was disabled at the runZero account level has been resolved.
- Operating System End of Life (EoL) information is now available for SUSE Enterprise Linux and Apple tvOS.
- Fingerprint improvements.

4.0.240419.0

2024-04-19

- Fingerprinting of assets based on Microsoft 365 Defender data has been improved.
- Fingerprint improvements.

4.0.240417.0

2024-04-17

- Accessibility improvements.
- A bug that could result in errors when deleting a site has been resolved.
- A bug that could cause Wiz tasks to error has been resolved.
- Fingerprint improvements.

4.0.240411.0

2024-04-11

- runZero customers can now sync asset, software, and vulnerability data from Wiz.
- Fingerprint improvements.

4.0.240410.0

2024-04-10

- The runZero dashboard has been improved to better respond to browser window resizing.

- Fingerprint improvements.

4.0.240408.0

2024-04-08

- Data collection from slow SSH services has been improved.
- Fortinet devices are now less likely to cause duplicate assets when traffic is collected using traffic sampling.
- The runZero Explorer now silently skips non-ethernet-like utun (tunnel) interfaces on macOS.
- A bug preventing the “User details” page for external users from loading has been resolved.
- A bug that could lead to errors when changing email address was fixed.
- A bug that could lead to errors when deleting a user was fixed.
- Fingerprint improvements.

4.0.240405.0

2024-04-05

- The profile settings page has been redesigned.
- Names can now be given to multi-factor authentication tokens when enrolling new tokens.

4.0.240404.0

2024-04-04

- Fingerprint improvements.

4.0.240403.0

2024-04-03

- A bug that prevented proper click through from the Query Insights dashboard widget to the appropriate inventory view was fixed.
- Matching of MAC addresses of Fortinet firewall devices was improved.
- Fingerprint improvements.

4.0.240402.0

2024-04-02

- Fingerprint improvements.

4.0.240401.0

2024-04-01

- The layout of the runZero dashboard is now fully customizable.
- The runZero dashboard now supports exporting views as CSV and PNG.
- Fingerprint improvements.

4.0.240331.0

2024-03-31

- Integration task processing is now much faster for assets with large numbers of MAC addresses.
- A bug that could result in assets accumulating link-local IPv6 addresses has been resolved.
- Fingerprint improvements.

4.0.240329.0

2024-03-29

- The “Contact runZero support” menu has been redesigned.
- A bug that could cause the services attribute report to fail has been resolved.
- A bug that could cause hostnames with spaces to be turned into multiple hostnames when imported from the AzureAD connector has been resolved.
- Improved logging for the Intune integration.
- UI improvements.
- Fingerprint improvements.

4.0.240327.0

2024-03-27

- Tenable connector data processing is now significantly faster for devices with large numbers of MAC addresses.
- A bug that could result in the self-hosted updater showing a SQL error during startup has been resolved.
- A bug that could cause scans running on Windows Explorers to accidentially terminate unrelated processes has been resolved.
- Fingerprint improvements.

4.0.240326.0

2024-03-26

- The CrowdStrike connector now only imports actively installed software.
- The CrowdStrike connector now handles large software and vulnerability results reliably.
- The CrowdStrike connector now better filters system accounts from the lastInteractiveUser attribute.
- Fingerprint improvements.

4.0.240325.0

2024-03-25

- Fingerprint improvements.

4.0.240320.0

2024-03-20

- Fingerprint improvements.

4.0.240318.0

2024-03-18

- Task ID is now visible when inspecting a task on the task overview page and on the task details page.
- An issue with calculating mid-scan progress for connector tasks running on Explorers has been resolved.
- A bug that could cause service start issues after upgrading self-hosted runZero instances has been resolved.
- Fingerprint improvements.

4.0.240314.0

2024-03-14

- Colors throughout the product have been tweaked to improve accessibility, legibility, and consistency.
- Tables in the product can now be configured to prefer a mono-spaced variant of the table font.
- Tables throughout the product now allow users to choose text casing preference, available via the “Prefs” dropdown.
- An issue that could prevent updates to Directory Users / Groups has been resolved.
- A bug that could cause the “concurrency” setting on Explorers to be incorrectly changed when editing an Explorer’s settings has been resolved.
- Accessibility improvements.

- Fingerprint improvements.

4.0.240311.0

2024-03-11

- An issue with processing malformed header data from RTSP responses has been resolved.
- The runZero CLI now completes faster for local networks.
- Self-hosted customers can now unbind SSO from a user account using the runzeroctl user reset command.
- Self-hosted customers can now change the SSO mode using the runzeroctl sso-mode *mode* command.
- Accessibility improvements.
- Fingerprint improvements.

4.0.240308.0

2024-03-08

- A bug that could cause short keywords to not show any autocomplete suggestions in the query builder has been resolved.
- Long fields in Nmap XML exports of asset data are no longer truncated.
- Probing devices using EtherNet/IP is now supported over UDP.
- Fingerprint improvements.

4.0.240306.0

2024-03-06

- An issue that could prevent new self hosted installations or updating existing installations has been resolved.
- Fingerprint improvements.

4.0.240305.1

2024-03-05

- An issue that could result in incorrect asset merging in certain situations has been resolved.
- An issue that could result in delayed analysis for busy Organizations has been resolved.
- Fingerprint improvements.

4.0.240305.0

2024-03-05

- Fingerprint improvements.

4.0.240304.0

2024-03-04

- Fingerprint improvements.

4.0.240301.0

2024-03-01

- A new “serialNumbers” column has been added to the asset CSV export. This field contains serial numbers observed during scanning, along with the protocol used to discover the serial number.
- An issue that could cause incorrect attack surface assignment to assets discovered by traffic sampling has been fixed.
- A bug which caused some task errors and warnings to fail to display has been fixed.
- Fingerprint improvements.

4.0.240228.0

2024-02-28

- A bug that could prevent sites from being created per project for the Google Cloud Platform integration has been resolved.
- Fingerprint improvements.

4.0.240226.0

2024-02-26

- A bug impacting Operating System End of Life (EoL) assertions for certain versions of Microsoft Windows and Linux distributions has been resolved.

4.0.240223.0

2024-02-23

- A bug that could cause organization statistics to become out of date in organizations with frequent and concurrent tasks has been resolved.
- Operating System End of Life (EoL) information is now available for Apple iOS and iPadOS as well as CentOS Stream.
- Operating System Extended End of Life (EoL) generation has been improved.

- Fingerprint improvements.

4.0.240221.0

2024-02-21

- The vulnerability inventory is now much faster for large organizations.
- Fingerprinting of devices via BGP is now supported.
- Tenable integration performance has been improved.
- A bug that could cause the asset and service attribute reports to fail has been resolved.
- A bug causing some credential form fields to disappear when modifying an existing credential has been resolved.
- An issue with the query format of site-filtered insights has been resolved.
- Fingerprint improvements.

4.0.240218.0

2024-02-18

- Software inventory is now calcuated as part of metrics, reducing task processing time.
- A bug that prevented the Organization picker from working on some pages has been resolved.
- Saved queries in the search suggestions menu are now ordered by when they were last updated.
- Improved asset correlation logic for devices with wired and wireless interfaces.
- Improved OS detection logic when considering multiple data sources.
- Fingerprint improvements.

4.0.240216.0

2024-02-16

- Improved correlation behavior for assets with information from NTLMSSP or Qualys.
- Search query and query builder autocomplete results have been improved for shorter sets of input.
- A bug preventing the parent-organization-picker from appearing on the organization create and edit pages has been resolved.
- Fingerprint improvements.

4.0.240214.0

2024-02-14

- Improved protocol detection during traffic sampling.
- The alert event type emitted after a client switch has changed from “login” to “client-switched”.
- The “Site” column has been removed from the software groups table.

- An issue where the software inventory sometimes failed to update after a task has been resolved.

4.0.240213.0

2024-02-13

- The Software Inventory is now much faster for large organizations.
- An issue that could result in stale service attributes persisting through rescans has been resolved.
- The `LOG_FORMAT` and `LOG_MAX_LENGTH` configuration values were renamed to `RUNZERO_LOG_FORMAT` and `RUNZERO_LOG_MAX_LENGTH` respectively. The old values will continue to work but are deprecated.
- The request timeout for the Qualys integration has been decreased.
- TCP stack based OS fingerprinting has been improved.
- Fingerprint improvements.

4.0.240208.0

2024-02-08

- An issue with adding addresses for Custom Integration assets without MACs has been resolved.
- The request timeout for the Qualys integration has been increased.

4.0.240207.0

2024-02-07

- Additional data points for result count and sent/received data have been added to the Tasks CSV export.
- An issue with the display format of site subnet tags on assets has been resolved.

4.0.240206.0

2024-02-06

- Improved performance on the Software inventory table.
- Additional fields added to Query Builder autocomplete.
- An issue that prevented Site Subnet information from exporting with Assets has been resolved.
- An issue with data missing from the default email template for alerts has been resolved.

4.0.240205.0

2024-02-05

- Filtering of hostnames collected from TLS X.509 certificates has been improved.
- An issue that could cause overlapping subnets to apply another Site's subnet tags has been resolved.
- An issue that could result in incorrect asset correlation between HP iLOs and their servers has been resolved.
- Fingerprint improvements.

4.0.240202.0

2024-02-02

- Performance of Tenable.io connector tasks when only a subset of Severity/Risk values are selected has been improved.
- An issue that allowed users with the Administrator role to downgrade their own permissions has been resolved.
- An issue that could prevent Nessus attributes from being fully hydrated by runZero has been resolved.
- Fingerprint improvements.

4.0.240131.0

2024-01-31

- Fingerprint improvements.

4.0.240129.0

2024-01-29

- A query builder is now available, accessible from most datagrids by clicking the "Query builder" button to the right of the search bar.
- An issue which caused some out-of-date service information to remain on assets has been resolved.
- An issue which caused service information to be incorrectly removed from assets that were offline during a scan has been resolved.

4.0.240126.0

2024-01-26

- Discovery of devices using the DNP3 protocol is now supported.
- Operating System End of Life (EoL) information is now available for Oracle Linux.
- Page break locations in the overview report have been improved.
- Operating System End of Life (EoL) generation for Red Hat Enterprise Linux and CentOS Linux has been improved.

- Assets with no known address are now labeled with “Unknown” for their address rather than “Unscanned”.
- The bundled npcap driver has been updated to version 1.79.
- An issue that could prevent last task details from correctly displaying on the Sites datatable has been resolved.
- An issue that prevented the expansion of dropdown menu sub-menus using keyboard navigation has been resolved.
- An issue that could result in certain OS fingerprinting data not being updated has been resolved.
- An issue that could prevent creating new Azure Credentials via the Azure connector configuration page has been resolved.
- An issue causing Tenable.io integration tasks to import vulnerability data even when no severity or risk levels were selected has been resolved.
- Fingerprint improvements.

4.0.240124.0

`2024-01-24`

- An issue that could result in hidden fields on the SNMP v3 Credentials form has been resolved.
- Fingerprinting of Red Hat Enterprise Linux derivatives when limited data is available has been improved.
- Additional fingerprint improvements.

4.0.240122.0

`2024-01-22`

- The datagrid search bar has been improved to show recent queries and available queries from the query library.
- Fingerprinting of Red Hat Enterprise Linux and derivatives from Tenable product data has been improved.

4.0.240119.0

`2024-01-19`

- Fixed an issue that prevented the “Edit user permissions” modal from working correctly.
- Fingerprint improvements.

4.0.240117.0

`2024-01-17`

- Fixed an issue where custom integration task data could not be re-imported.
- Fixed an issue where Nessus imports could fail due to Nessus response size.

- Fingerprinting of Red Hat Enterprise Linux derivatives such as CentOS, Rocky Linux, and Oracle Linux has been improved.
- Fingerprint improvements.
- Accessibility improvements.

4.0.240112.0

2024-01-12

- Site column has been added to all tasks lists in the task overview.
- Fingerprint improvements.

4.0.240110.0

2024-01-10

- The Nmap XML export now uses the minimum and maximum asset last_seen timestamps as the start and stop times.
- An issue that could prevent stale services from being cleared from updated Assets has been resolved.
- A resource leak that affects self-hosted customers with transparent huge pages (THP) enabled has been resolved.
- Fingerprint improvements.

4.0.240109.0

2024-01-09

- Tenable Security Center tasks now only retrieve records updated since the previous sync.
- Fingerprint improvements.

4.0.240105.0

2024-01-05

- A bug that prevented the API for creating passive sampling tasks from working as documented was fixed.
- A bug that could cause inventory grids to disappear when using Firefox and resizing the window below a certain point has been resolved.
- Improved error handling for Tenable, Tenable Security Center, and CrowdStrike integrations.
- Fingerprint improvements.

4.0.240103.0

2024-01-03

- Improved correlation for assets sourced from the Censys and Shodan integrations.
- A bug that incorrectly logged certain task failures as 'explorer failed to queue task' has been resolved.
- Fingerprint improvements.

Archived release notes

Release notes prior to 2024 can be found in the release notes archive.

Index

B

C

D

E

F

G

H

I

O

P

Q

R

S

T

U

V

W

X