NOZOMI NETWORKS

# Central Management Console
## User Guide

N2OS v25.2.0 — 2025-07-03

# Legal notices

*Information about the Nozomi Networks copyright and use of third-party software in the Nozomi Networks product suite.*

## Copyright



## Third Party Software

Nozomi Networks uses third-party software, the usage of which is governed by the applicable license agreements from each of the software vendors. Additional details about used third-party software can be found at https://security.nozominetworks.com/licenses.

# Contents

# Chapter 1. Introduction

# CMC overview

*The Central Management Console (CMC) aggregates information from thousands of sites and assets to deliver instant awareness of your networks and their activity patterns.*

## Overview

The *Central Management Console (CMC)* makes it easy to have visibility of all your Nozomi Networks sensors. The *CMC* aggregates information from thousands of sites and assets to deliver instant awareness of your *operational technology (OT)*/*Internet of Things (IoT)* networks and their activity patterns, enabling users to quickly troubleshoot issues and accelerate incident response across your network.

*CMC* can be deployed:

- On-premises
- In a public cloud provider such as *Amazon Web Services (AWS)* or Microsoft Azure
- At the edge on physical or virtual appliances

## Functionalities

The *CMC* has been designed to support complex deployments that cannot be addressed with a single sensor. The *CMC* shares the user experience of the sensor. In comparison to the sensor, some functionalities have been added, and some have been removed. The functionalities that have been added let you manage *hundreds* of sensors. The functionalities that have been removed increase efficiency for real-world, geographic deployments of the Nozomi Networks Solution architectures.

## Sensors

The *CMC* **Sensors** page lets you see and manage all the connected sensors. A graphical representation of all the hierarchical structure of the connected sensors and the sensor Map is presented to allow a quick health check on a user-provided raster map.

Once sensors are connected, they are periodically synchronized with the *CMC*. In particular, the Environment of each sensor is merged into a global Environment and Alerts are received for a centralized overview of the system. Of course, Alerts can also be forwarded to a *security information and event management (SIEM)* directly from the *CMC*, thus enabling a simpler decoupling of components in the overall architecture.

## Synchronization

To synchronize data, the sensors must be running the same major *Nozomi Networks Operating System (N2OS)* release, or one of the two prior major ones. For example, if the *CMC* is running version 23.0.x (the major release is 23.0), sensors can synchronize if running one of the following versions: 23.0.x, 22.5.x or 22.0.x.

Firmware update is also simpler with a *CMC*. Once the new firmware is deployed to it, all connected sensors can be automatically updated.

# Functionalities

*A description of the differences in functionalities between the Central Management Console (CMC) and sensors. The actions shown below are only applicable for CMCs that are in* **All-In-One** *mode, not* **Multicontext** *mode.*

## Node actions

**Table 1. Node actions**

| Action | Sensor | CMC |
|---|---|---|
| Configure node | ✓ | ✗ |
| Show alerts | ✓ | ✓ |
| Show requested traces | ✓ | ✗ |
| Request a trace | ✓ | ✗ |
| Manage Learning | ✓ | ✓ |
| Navigate | ✓ | ✓ |

## Network links

**Table 2. Network links**

| Action | Sensor | CMC |
|---|---|---|
| Configure node | ✓ | ✗ |
| Show alerts | ✓ | ✓ |
| Show requested traces | ✓ | ✗ |
| Request a trace | ✓ | ✗ |
| Show events | ✓ | ✗ |
| Show captured urls | ✓ | ✗ |
| Manage Learning | ✓ | ✓ |

Table 2. Network links (continued)

| Action | Sensor | CMC |
|---|---|---|
| Navigate | ✓ | ✓ |

## Process variables

Table 3. Process variables

| Action | Sensor | CMC |
|---|---|---|
| Configure variable | ✓ | ✗ |
| Variable details | ✓ | ✓ |
| Add to favourites | ✓ | ✓ |
| Navigate | ✓ | ✓ |

# Alerts

*A description of alerts management in a Central Management Console (CMC).*

Alerts management in the *CMC* is similar to alerts management in a sensor. However, in the *CMC*, you can have all the alerts, from all the sensors, in one centralized place.

In a sensor, you can create a query, and therefore an assertion, that includes all the nodes, or links etc., for your complete infrastructure. You can create a *Global Assertion*. This is one or more groups of assertions that can be propagated to all the sensors. The *CMC* has control of these assertions, and sensors cannot edit or delete them.

It is possible to configure the *CMC* to forward alerts to a *SIEM* without the need to configure each sensor. For more details, see **Data integration**.

# Updates

*A description of the release update and rollback operations when you have a Central Management Console (CMC) and one or more sensors.*

The Nozomi Networks update bundle applies to both the *CMC* and Guardian, except in the case of the Docker installation. It works for all physical and virtual sensors.

Once a sensor is connected to a *CMC*, the *CMC* will control all the updates. The software bundle is propagated from the *CMC* and, once the sensor receives the bundle, you can do the update manually, or automatically. You can configure this behavior in the **General settings** section of the **Synchronization settings** page.

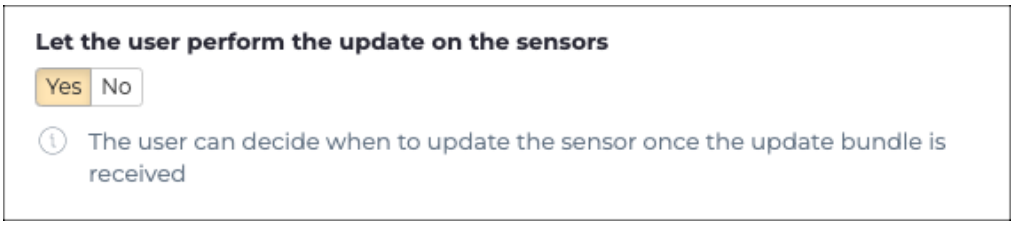



**Figure 1. Let user perform updates section**

If the *CMC* is configured to allow manual updates, the sensor's status bar displays a message that notifies you as soon as the sensor receives the update bundle.


**Figure 2. Update available notification**

The update process from the *CMC* can proceed as described in **Software update and rollback**. After the Central Management Console is updated, each sensor will receive the new software update.

If an error occurs during the update procedure, a message shows next to the related sensor's version number on the **Sensors** page.

To Rollback, first rollback the Central Management Console, and then proceed to rollback all the sensors as described in **Software update and rollback**.

# Chapter 2. Sensors

*The **Sensors** page lets you view all of the sensors that you have in your system.*

The **Sensors** page has these three tabs:

- List
- Map
- Graph

# List

*The **List** page lets you view all of the sensors that you have in your system.*

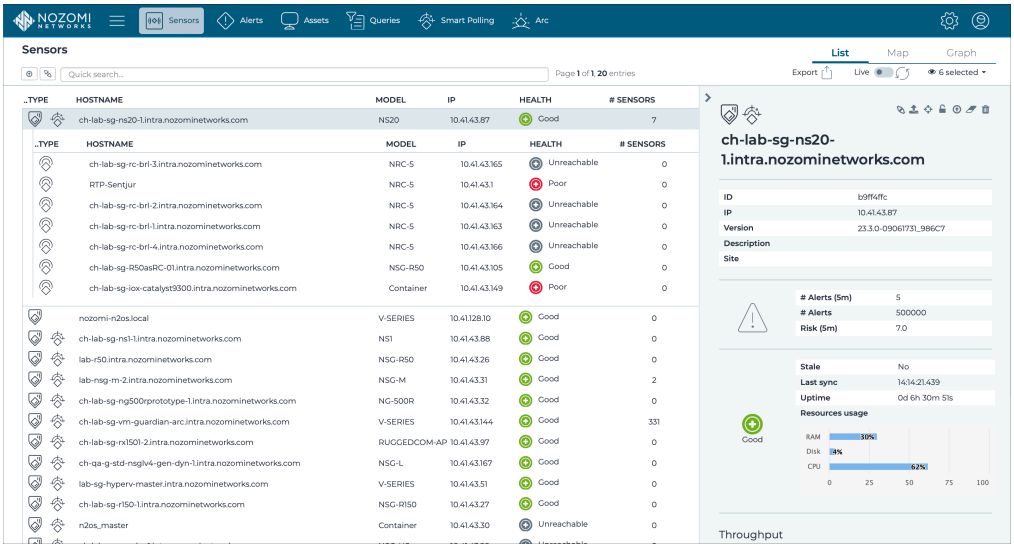



**Figure 3. Sensors list**

## Force update

The force update icon lets you do a force update on the selected sensors.

## Allow/disallow

The allow/disallow icon lets you allow or disallow sensors.

## Quick search

The quick search field lets you easily do a search on the current page.

## Export

The **Export** icon lets you export the current list in either *comma-separated value (CSV)* or Microsoft Excel format.

## Download Arc

The **Download Arc** dropdown lets you select, and download, the applicable Arc package for your *operating system (OS)* and architecture.

### Live

The **Live** toggle lets you change live view on, or off. When live mode is on, the page will refresh periodically.

### Refresh

The **Refresh** icon lets you immediately refresh the current view.

### Column selection

The columns selection icon lets you choose which columns to show or hide.

### Information pane

The information pane to the right of the list of sensors shows additional information for the selected sensor.

It also lets you do these actions shown below.

**Table 4. Sensors list actions**

| | |
|---|---|
| Allow/Disallow sensor | After allowing a sensor, this icon shows: Synchronized data coming from the sensor become part of the Environment of the *CMC*. Alerts coming from the sensor can be seen in the **Alerts** section. |
| Focus on sensor | Allows to filter out only the sensor chosen data, such as Alerts and Environment. |
| Go to sensor | Connect to a remote sensor directly from the *CMC*. Select this to open a new browser tab to the sensor selected login page. The action is hidden if the *CMC* isn't configured to allow this type of communication between sensors and *CMC*. |
| Place in map | This action is used to place the sensor on the map. |
| Toggle version lock | When locked, the sensor will not automatically update its software. |
| Force update | Even if it is locked, the sensor will automatically update its software, with the version installed on the *CMC*. |
| Clear sensor data on this machine | Clear all synchronized data at the *CMC* received from the selected sensor. Use this in combination with the clearing of the data on the sensor, and you will be able to restart the synchronization between the sensor and the *CMC* from an empty state |
| Delete sensor | Clear all data received from the selected sensor and delete it from the list. If the sensor tries to sync with the *CMC* again, it shows as disallowed in the list. |

# Map

*You can use the **Map** page to upload, and view, a map of the sensors in your environment.*

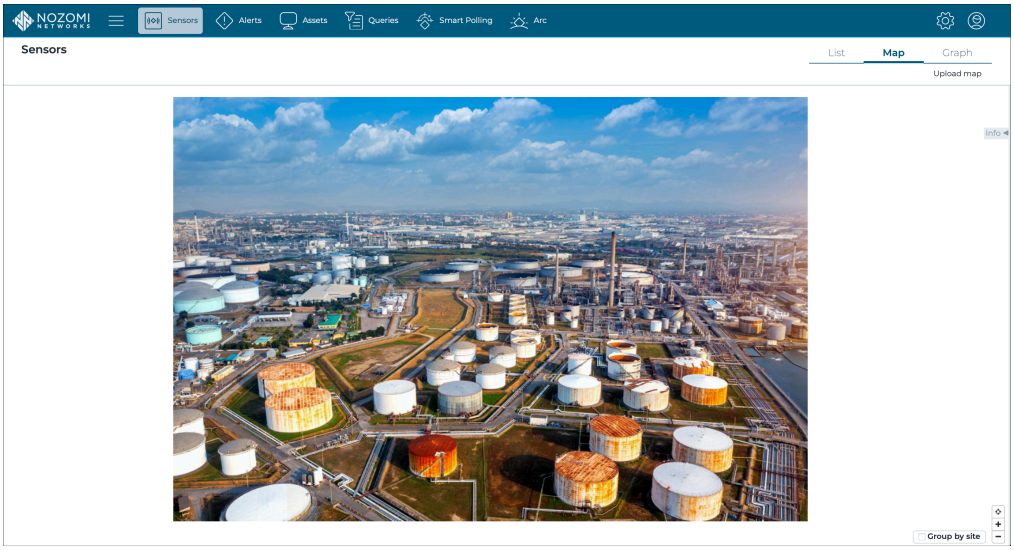



**Figure 4. Sensors map**

## Info(rmation) pane

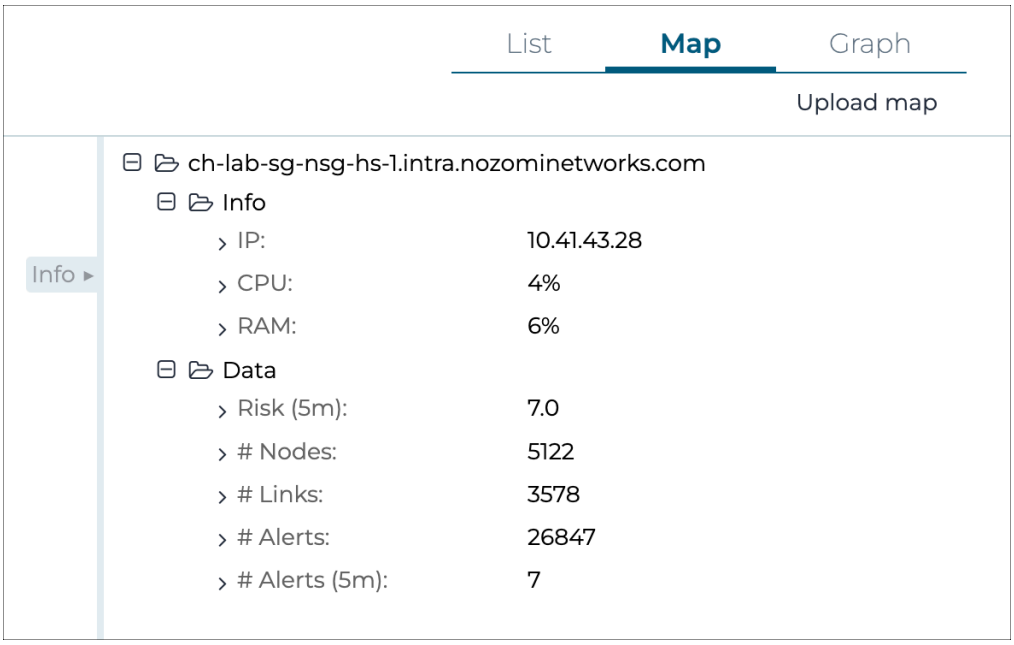



**Figure 5. Info(rmation) pane**

The **Info** pane lets you view information for the related sensor. The *identifier (ID)* of each sensor is used in the map to help you identify it. The marker color of the sensor relates to the risk of its alerts.

In the map view, a red indicator to the right of the sensor's *ID* shows the number of the alerts in the last five minutes. This indicator only shows if there are some alerts for the sensor. If the alerts in last 5 minutes increase, the sensor marker will blink for one minute.

If the site of the sensor has been specified in > **System > General**, it is possible to enable the **Group by site** option, in the bottom right corner of the map view. The sensors with the same site will be grouped to deliver a simpler view of a complex installation.

**Note:**

The sensors map is also available as a widget.

# Graph

*The **Graph** page shows a graphical view of all the sensors in you environment.*


**Figure 6. Sensors graph**

# Do a force update on sensors

*If you have disabled auto updates, you can use the force update icon to do a manual update.*

## Procedure

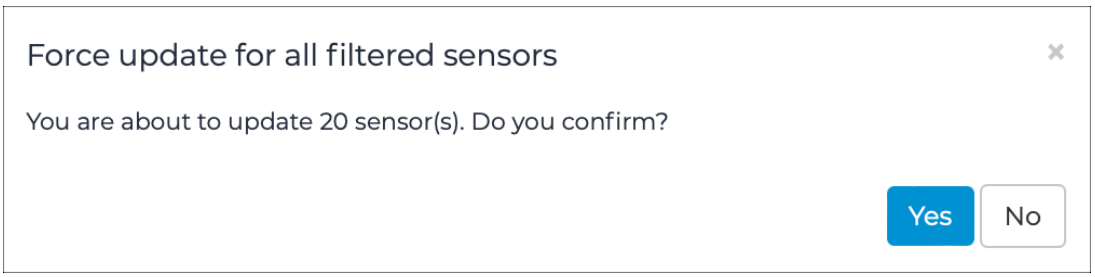
1. In the top navigation bar, select **Sensors**.

   **Result:** The **Sensors** page opens.

2. In the top left section, select .

   **Result:** A dialog shows.

3. To confirm, select **Yes**.



# Allow a sensor

*The allow icon lets you give permission for a new sensor to connect to Central Management Console (CMC).*

## Procedure

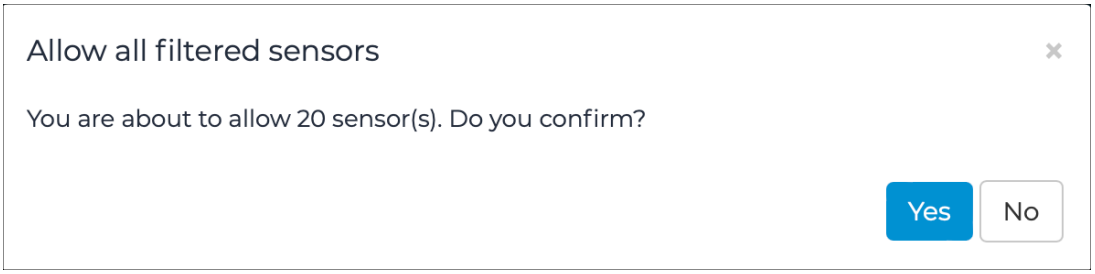
1. In the top navigation bar, select **Sensors**.

   **Result:** The **Sensors** page opens.

2. In the top left section, select the icon.

   **Result:** A dialog shows.

3. To confirm, select **Yes**.



# Export a list of sensors

*You can export a list of the sensors from the current view.*

## Procedure

1. In the top navigation bar, select **Sensors**.

   **Result:** The **Sensors** page opens.

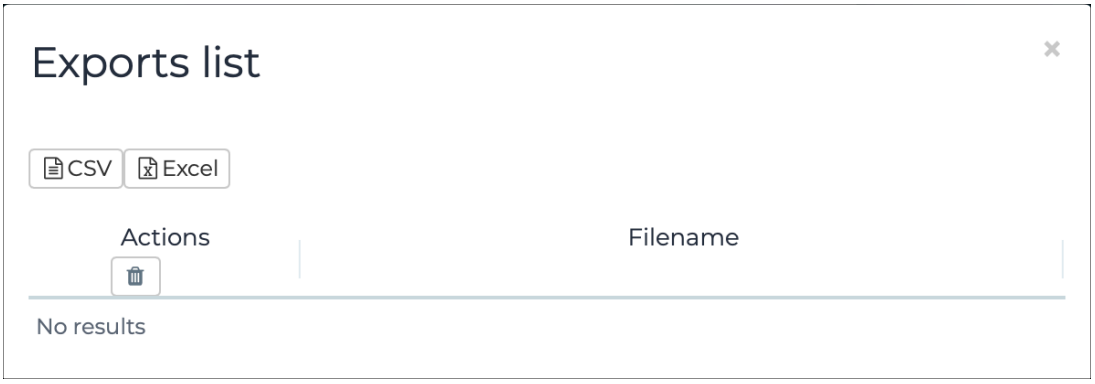
2. In the top right section, select **Export**.

   **Result:** A dialog shows.

3. Choose an export format:

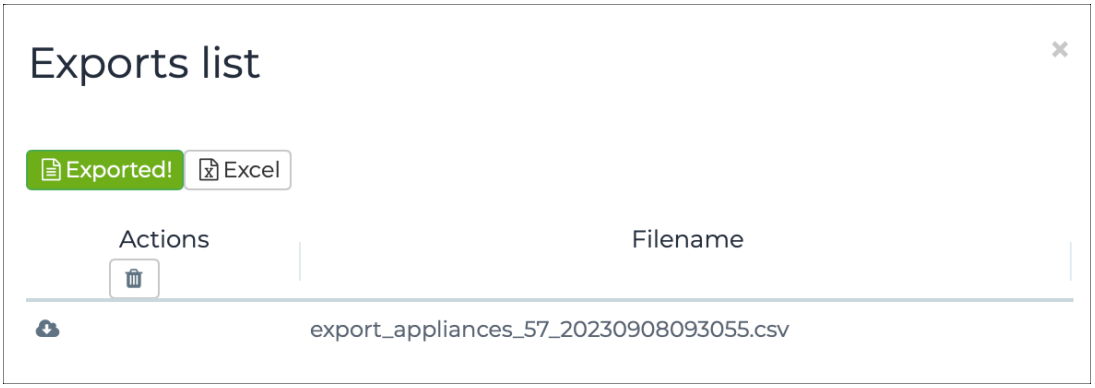
   **Choose from:**

   - Select **CSV** to create a *CSV* file
   - Select **Excel** to create a Microsoft Excel file




   **Result:** The file is created and a `Exported` message shows.

4. To download the file, in the bottom-left of the dialog, select




## Results

The exported file has been downloaded.

# Upload a map

*The **Map** page lets you upload a map of your sensors in your environment.*

## Procedure

1. In the top navigation bar, select **Sensors**.

   **Result:** The **Sensors** page opens.

2. In the top right section, select **Map**.

   **Result:** The **Map** page opens.

3. In the top right section, select **Upload map.**
4. Select the image file that you want to upload.

## Results

The map is uploaded.

# Chapter 3. Alerts

*The **Alerts** page shows all the latest alerts in the system. It lets you view the alerts in different modes, and carry out actions on the alerts.*

**Important:**

To perform actions on alerts, the user must belong to a group with admin permissions. Non-admin users can access alerts only if at least one of the groups that they belong to has alerts permission enabled.


**Figure 7. Alerts page menu**

The top right section of the **Alerts** page has two icons that let you change between these two options:

- Standard mode (on page 30)
- Expert mode (on page 31)

# Standard mode

*You can view alerts in standard mode to give you an overview of the latest anomalies.*

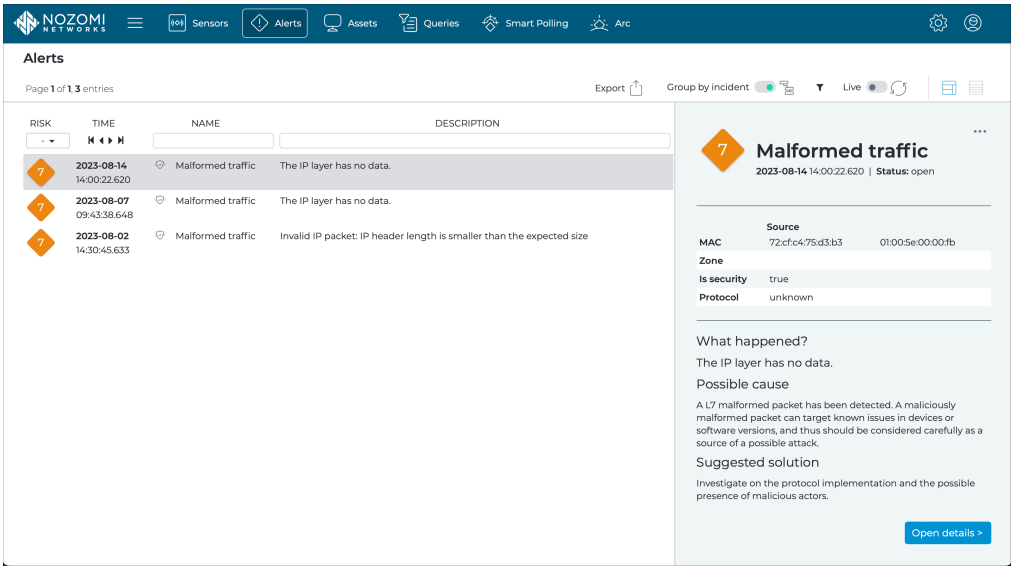



**Figure 8. Standard mode**

## Risk
This shows the risk associated to each alert or incident.

## Time
The time related to each event.

## Name
The name category of the event.

## Description
This shows a detailed description of the related event.

## Analysis
If you can select a row, this pane will show a more detailed analysis of the alert.

# Expert mode

*You can view alerts in expert mode to give you a detailed view of the alerts in the system. This lets you filter, sort, and analyze the information in detail.*

Expert mode shows a comprehensive table layout, with details on the alerts and incidents listed, which include:

- Addresses
- Labels
- The roles of the involved nodes, zones, protocol, and ports used in the involved transactions, and more

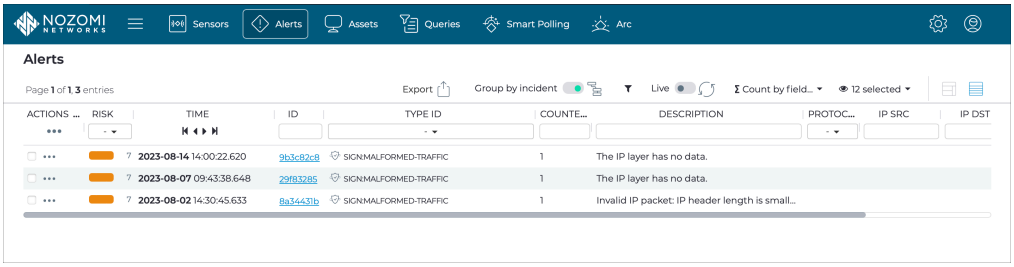



**Figure 9. Expert mode**

## Export

The **Export** icon lets you export the current list in either *CSV* or Microsoft Excel format.

## Group by incident

The **Group by incident** icon lets you group alerts by incident. This will show incidents, and hide all the alerts that belong to it.

## Filter

The filter icon opens a list of items that you let you filter the results.

## Live

The **Live** toggle lets you change live view on, or off. When live mode is on, the page will refresh periodically.

## Refresh

The **Refresh** icon lets you immediately refresh the current view.

## Count by field

The **∑ Count by field** dropdown lets you select a data field on which to group and count the alerts.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# View alerts in standard mode

*You can view alerts in standard mode to give you an overview of the latest anomalies.*

## Procedure

1. In the top navigation bar, select **Alerts**.

   **Result:** The **Alerts** page opens.

2. In the top right corner, select the standard mode icon.

   **Result:** The Standard mode (on page 30) view opens.

# View alerts in expert mode

*You can view alerts in expert mode to give you a detailed view of the alerts in the system. This lets you filter, sort, and analyze the information in detail.*

## Procedure

1. In the top navigation bar, select **Alerts**.

   **Result:** The **Alerts** page opens.

2. In the top right corner, select the expert mode icon.

   **Result:** The Expert mode (on page 31) view opens.

# Closing alerts

*When you close an alert, or incident, a dialog lets you select a reason, and specify the learning process.*

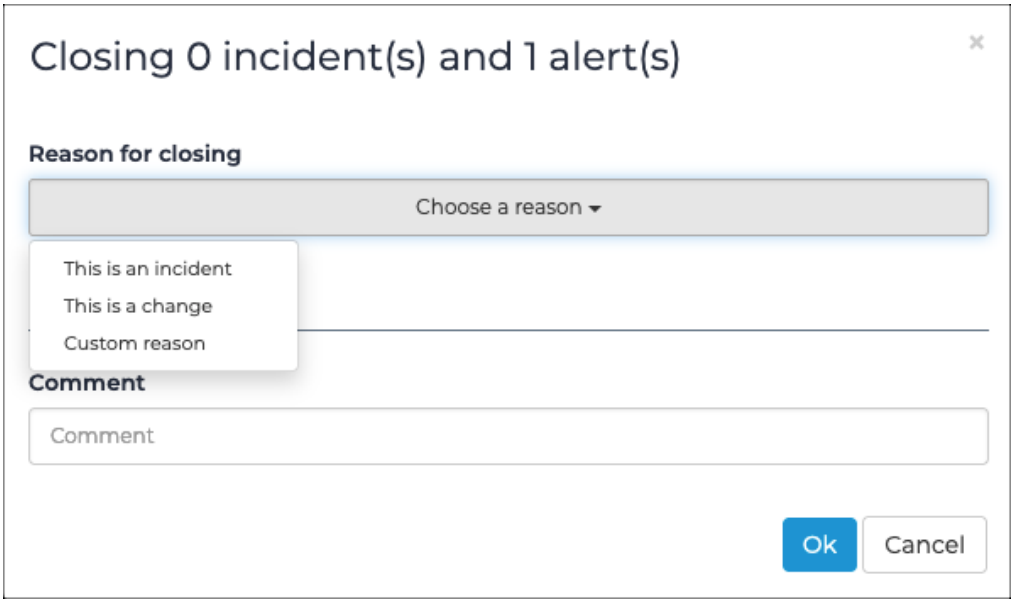



**Figure 10. Alerts closing dialog**

The **Reason for closing** dropdown has these options:

- **This is a change**: If the cause of the alert is an intended change to the network, such as:
  - A new computer being attached
  - New communication between two nodes that were not previously communicating

  Guardian can learn the change that has been detected as part of the environment baseline. When you close an alert in this way, the *intrusion detection system (IDS)* is instructed to learn the related objects. For example, when a `VI:NEW-NODE` alert is closed as a change, Guardian registers that the corresponding node is part of the environment and will not raise subsequent `VI:NEW-NODE` alerts about the same node.
- **This is a change**: If the cause of the alert is an intended change to the network, such as a new computer being attached, or a new communication between two nodes that were not talking before, the change detected by Guardian can be learned as part of the environment baseline. When closing an alert in this way, the IDS is instructed to learn the corresponding objects. For example, when a VI:NEW-NODE alert is closed as a change, Guardian registers that the corresponding node is part of the environment and will not raise subsequent VI:NEW-NODE alerts about the same node.

- **This is an incident**: If the cause of the alert is a configuration error, an attack, a malfunctioning device, or other security incident, the change is not learned as part of the environment baseline. When closing an alert in this way, the IDS is instructed to delete the corresponding objects. For example, a new node entering the network for the first time causes a VI:NEW-NODE alert. If an alert closes as an incident, reference to the new node is deleted. The VI:NEW-NODE alert is raised again in subsequent communication involving the same node.
- **Custom reason**: This lets you write a custom reason for closing an alert. You can enter a text string as the closing reason, with a request to apply one of the two described behaviors.

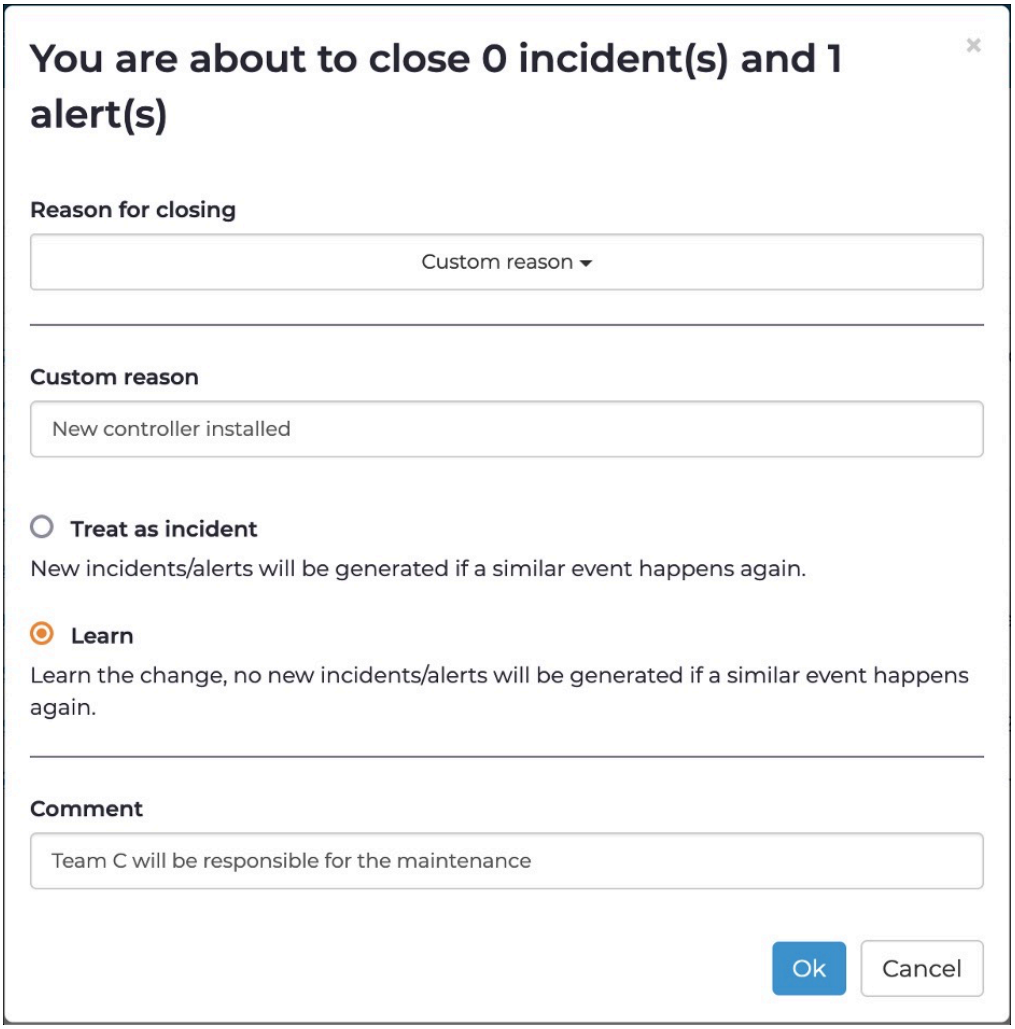



Figure 11. Closing alert for custom reason with comment

You can add a comment so that it shows in the **alert audit** log.

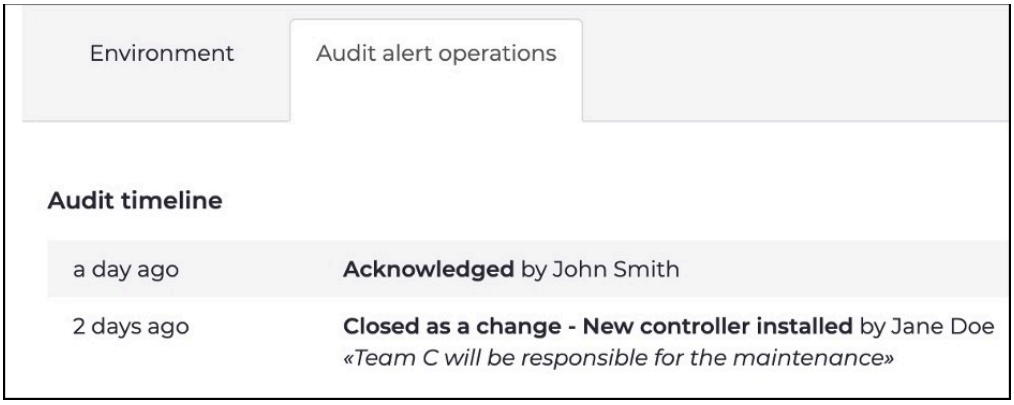



Figure 12. Audit alert operations

# Edit a playbook associated with an alert

*A playbook associated with an alert can be modified. A change you make on the playbook only affects the playbook related to that specific alert.*

## About this task

You can edit a playbook associated with an alert from the **Playbook** tab in the **Alerts** page.

## Procedure

1. In the top navigation bar, select **Alerts**.

   **Result:** The **Alerts** page opens.

2. Find the alert on which to edit the assigned playbook.
3. Select the alert, then select **Playbook** at the bottom of the screen.
4. To edit the playbook, select **Edit**.

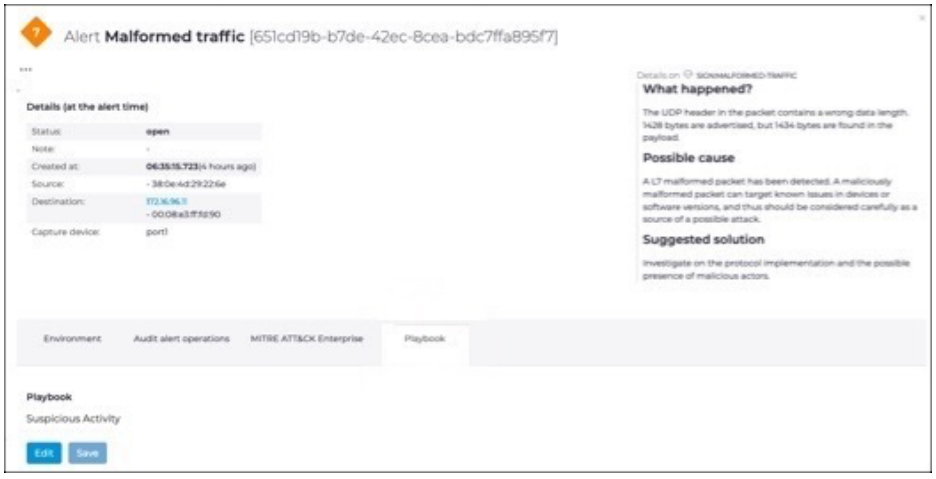



**Figure 13. Playbook tab**

5. Edit the playbook as necessary.
6. Select **Save**.

> **Note:**
> The playbook template from which the alert playbook was generated remains unchanged, as do all other alert playbooks generated from the same playbook template.

# Chapter 4. Assets

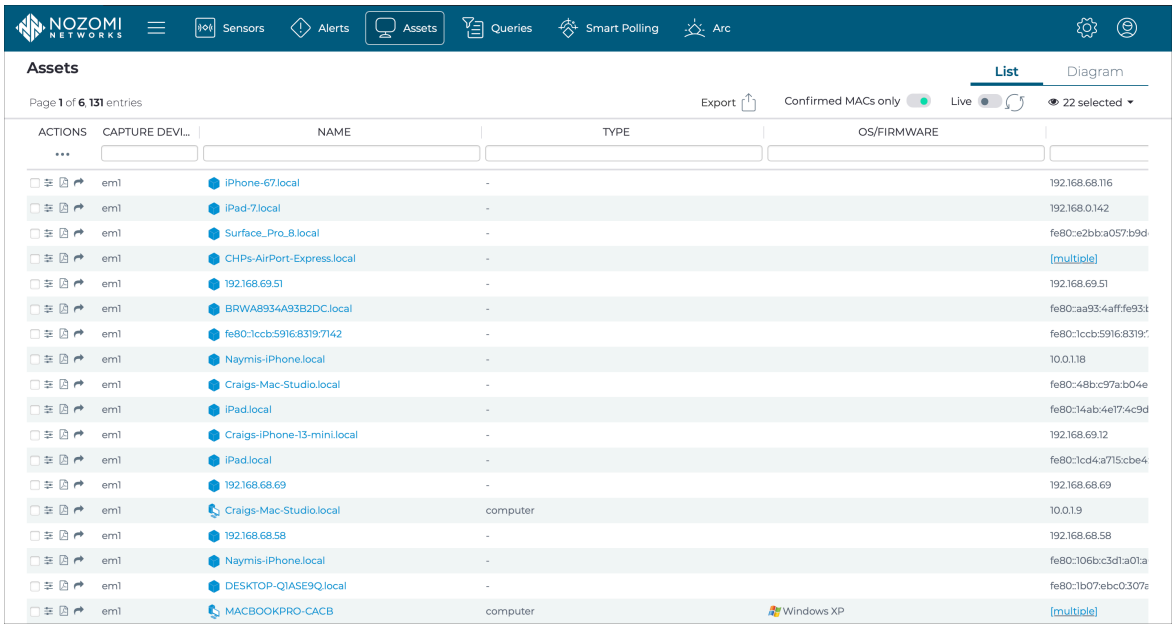
*The **Assets** page shows all the physical components and systems in the local network environment and their associated details. It also lets you perform actions on those assets. Depending on the nodes and components involved, assets can range from a simple personal computer to an operational technology (OT) device.*

The top right section of the **Assets** page has these two tabs:

- List (on page 39)
- Diagram (on page 41)

# List

*The **List** page shows all the assets in table format.*




## Export

The **Export** icon lets you export the current list in either *CSV* or Microsoft Excel format.

## Confirmed MACs only

The **Confirmed MACs only** toggle lets you select only assets that have a confirmed *media access control (MAC)* address.

## Live

The **Live** toggle lets you change live view on, or off. When live mode is on, the page will refresh periodically.

## Refresh

The **Refresh** icon lets you immediately refresh the current view.

### Column selection

The columns selection  icon lets you choose which columns to show or hide.

# Diagram

*The **Diagram** page uses the Purdue model format to display the assets. Assets are shown in separate rows, according to their level.*

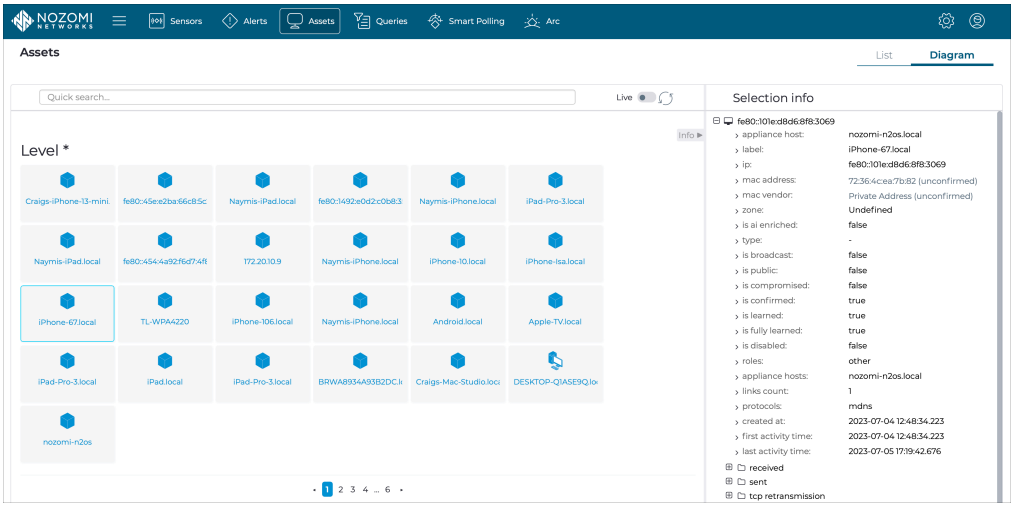



**Figure 14. Diagram page**

## Search bar

The search bar lets you search for a specific item.

## Live

The live toggle lets you immediately refresh the graph.

## Refresh

The refresh icon lets you immediately refresh the graph.

## Levels section

The levels section shows an icon for each asset, and shows on which level of the Purdue model it is.

## Selection info

When you select the link below an asset's icon, the **Selection info** pane shows more details for the selected asset.

# Details window

*The details window lets you view more detailed information for an assets.*

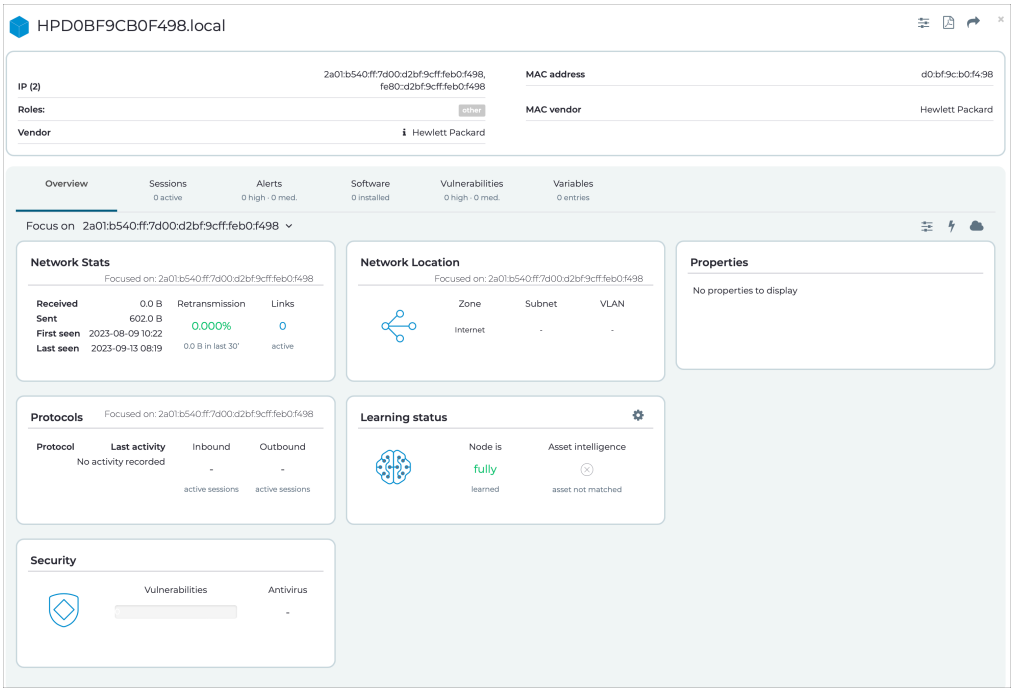



**Figure 15. Asset details window**

This details window shows more details for an asset.

The top section of the screen contains generic data. You can hover your mouse over the information **i** icon to display the source, granularity and confidence of the corresponding piece of data. Data includes:

- *internet protocol (IP)* (address)
- Roles
- Vendor
- *MAC* address
- *MAC* vendor

The details window has these tabs:

- Overview (on page 48)
- Sessions (on page 49)
- Alerts (on page 50)
- Software (on page 51)
- Vulnerabilities (on page 54)
- Variables (on page 55)

## Information icon

When you hover over the icon, you can see information for:

- Source
- Granularity
- Confidence

## Source

| Information source | Description |
|---|---|
| manual | Information that is manually added from the configuration |
| imported data | Imported information |
| passive detection | Information from deep packet inspection |
| asset-kb | Information from Asset Intelligence |
| smart-polling | Information from Smart Polling |

## Granularity

| Level of detailed information | Description |
|---|---|
| manual-or-import | Information manually added or imported |
| complete | Detailed information that has been extracted |
| partial | Detailed, but not complete information |
| generic | A family/generic value is found, but it is not detailed |
| unknown | Unknown |

## Confidence

| Level of confidence in information | Description |
|---|---|
| manual-or-import | Information manually added or imported, with the highest level of confidence at this level |
| high | High level of confidence |
| good | Good level of confidence |
| low | Low level of confidence |
| unknown | Unknown confidence |

# Configure an asset

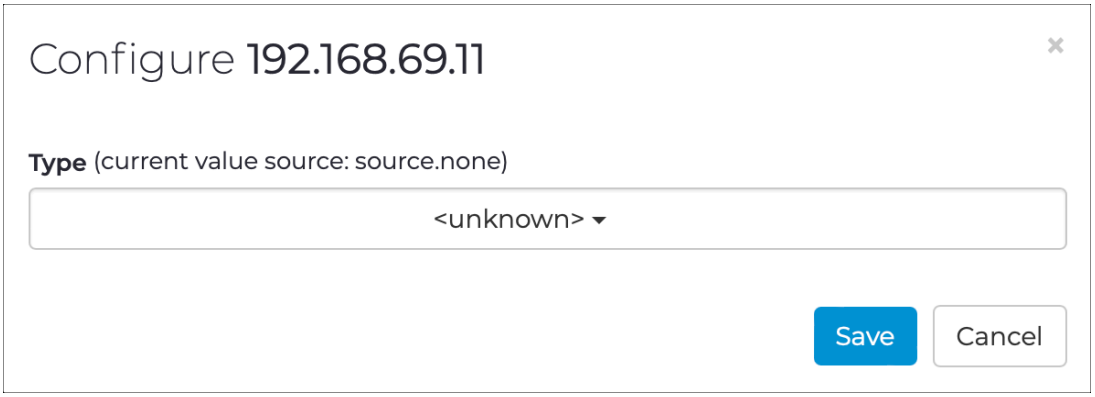
*The **Lists** page lets you configure assets.*

## Procedure

1. In the top navigation bar, select **Assets**.

   **Result:** The **Assets** page opens.

2. In the Actions column, to the left of the applicable asset, select the icon.




3. From the dropdown, select the applicable type.
4. Select **Save**.

## Results

The asset has been configured.

# Generate a PDF report of an asset

*The **Lists** page lets you generate a report in portable document format (PDF).*

## Procedure

1. In the top navigation bar, select **Assets**.

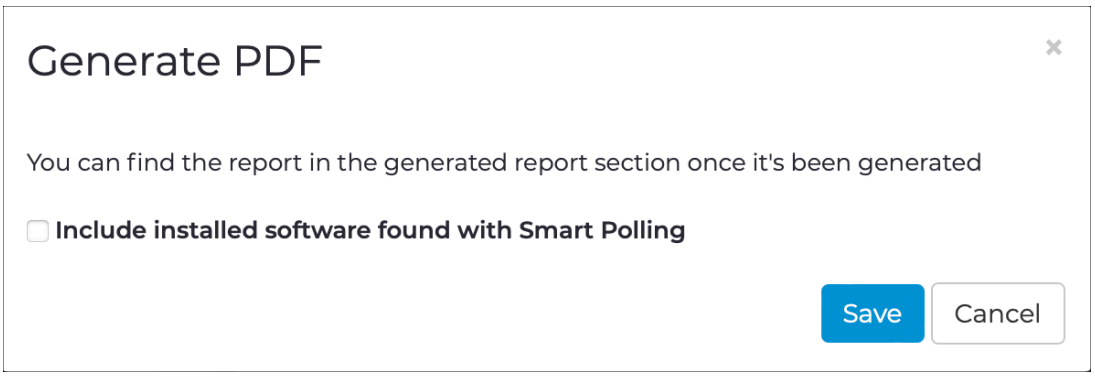
   **Result:** The **Assets** page opens.

2. In the Actions column, to the left of the applicable asset, select the icon.

   **Result:** A dialog shows.

3. **Optional:**

   If necessary, select the **Include installed software found with Smart Polling** checkbox.




4. Select **Save**.

   **Result:** The *portable document format (PDF)* file generates in the background. When it is ready, you can view it on the **Reports** page.

5. To view the report, go to **Reports > Generated**.

# Navigate from an asset

*The **Lists** page lets you use hyperlinks to navigate to entities that are related to an asset.*

## Procedure

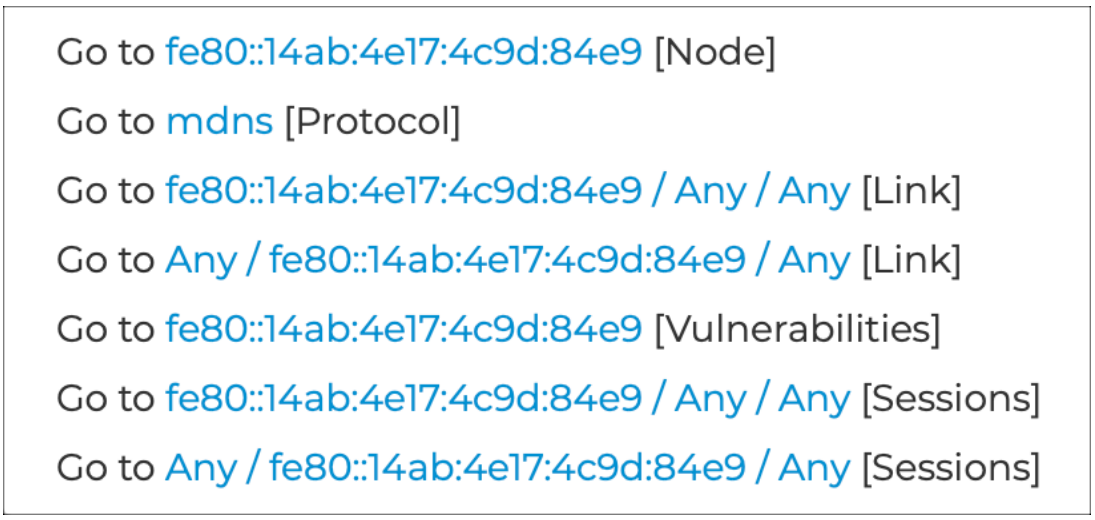
1. In the top navigation bar, select **Assets**.

   **Result:** The **Assets** page opens.

2. In the Actions column, to the left of the applicable asset, select the icon.

   **Result:** A list of related entities shows.

3. Select the hyperlink that you want to navigate to.




## Results

The entity shows in the applicable page.

# View more details for an asset

*Both the **Lists** and **Diagrams** pages lets you view more details for a specific asset.*

## Procedure

1. In the top navigation bar, select **Assets**.

   **Result:** The **Assets** page opens.

2. In the top right section, select either **List** or **Diagram**
3. Choose a method to view more details for an asset:

   **Choose from:**

   - If you chose the **List** page, in the **NAME** column for the applicable asset, select the hyperlink
   - If you chose the **Diagram** page, below the applicable asset, select the hyperlink

## Results

The Details window (on page 42) for the asset shows.

# Overview

*The **Overview** page shows a general overview of information for items such as network statistics and location, protocols, and learning status for the related assets.*

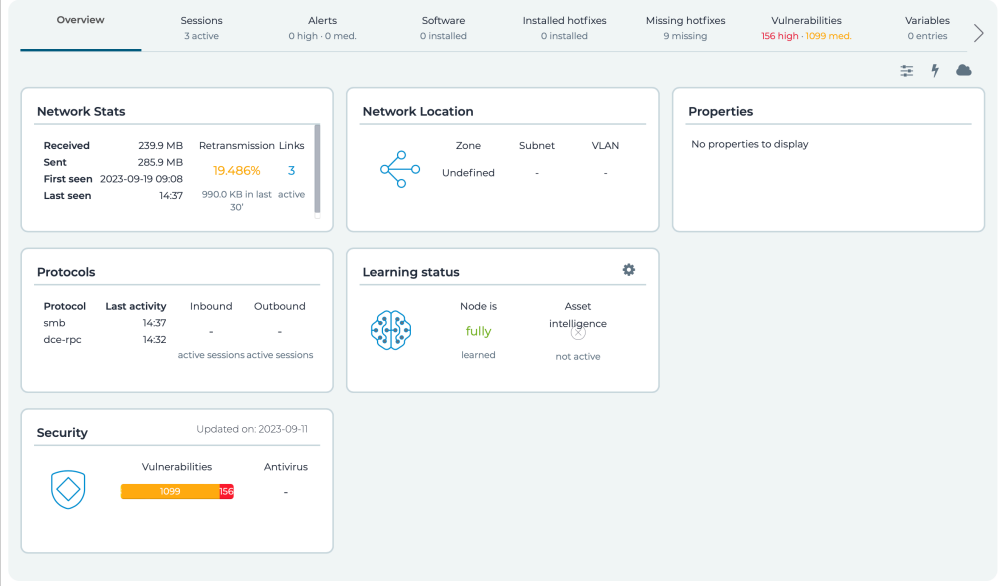



**Figure 16. Overview tab**

## Network Stats

This shows useful statistics for the network activity for the related device.

## Network Location

This shows information for the location for the related device on the network.

## Properties

This shows additional information for the related device.

## Protocols

This shows information for the different protocols that the related device uses.

## Learning status

This shows the learning status of the related asset.

## Security

This shows the security status of the related asset.

# Sessions

*The **Sessions** page shows detailed information for communication sessions between devices.*

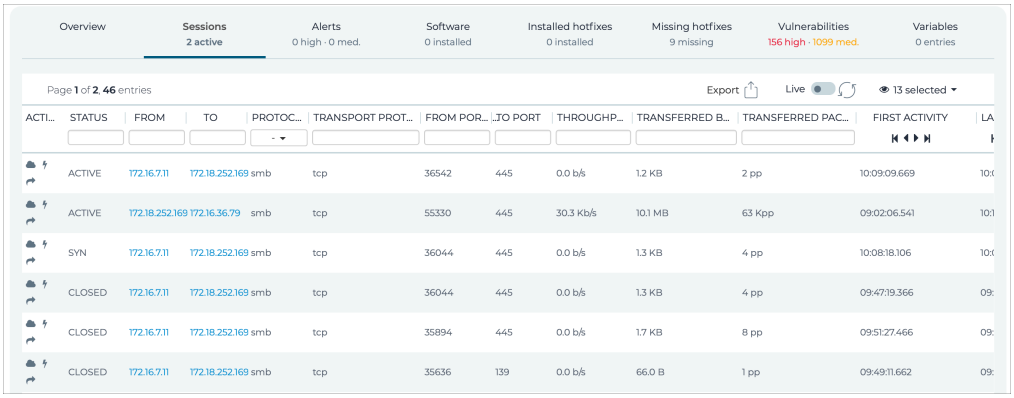



**Figure 17. Sessions tab**

## Export

The **Export** icon lets you export the current list in either *CSV* or Microsoft Excel format.

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Alerts

*The **Alerts** page shows detailed information for all the alerts that have been raised for the related assets.*

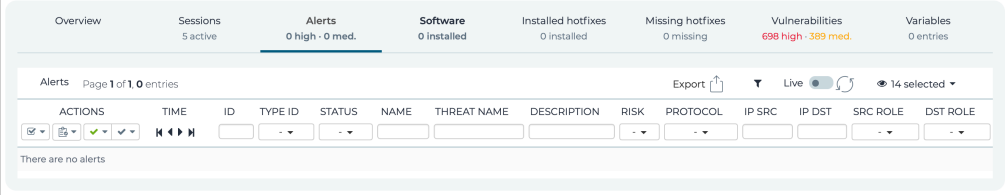



**Figure 18. Alerts tab**

## Export

The **Export** icon lets you export the current list in either *CSV* or Microsoft Excel format.

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Software

*The **Software** page shows a list of software applications that are installed on the related assets.*

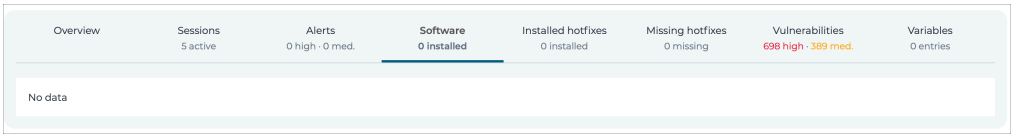



**Figure 19. Software tab**

# Installed hotfixes

*The **Installed hotfixes** page shows a list of hotfixes that are installed on the related assets.*

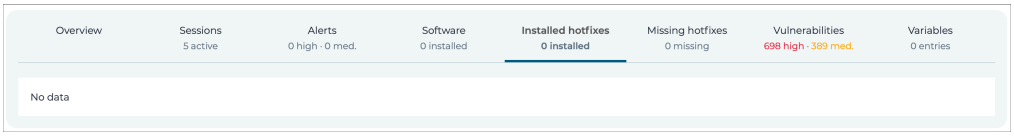



**Figure 20. Installed hotfixes tab**

# Missing hotfixes

*The **Missing hotfixes** page shows a list of hotfixes that could be installed on the related assets to resolve the related vulnerabilities.*

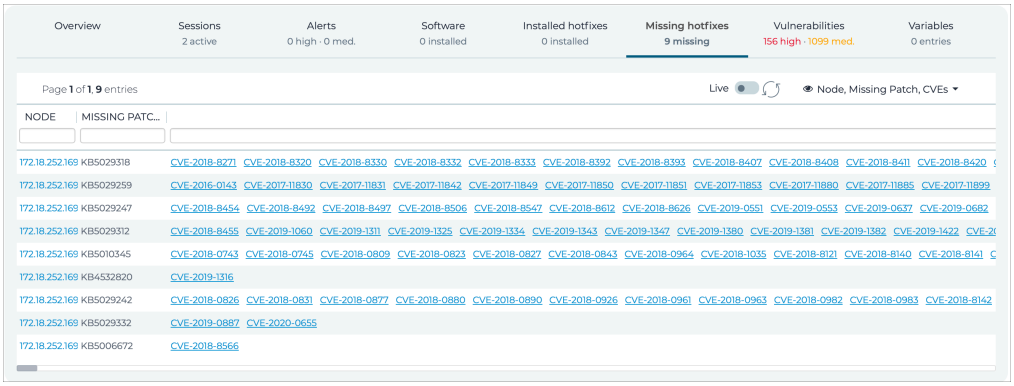



**Figure 21. Missing hotfixes tab**

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Vulnerabilities

*The **Vulnerabilities** page shows a list of vulnerabilities that are present on the related asset.*

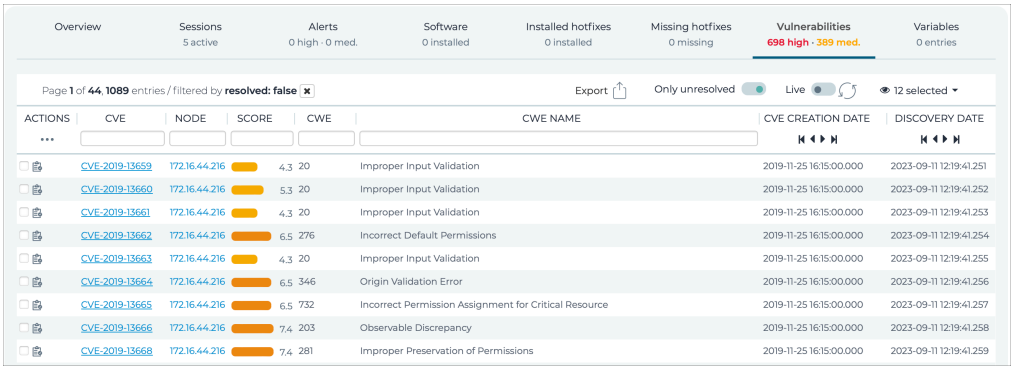



**Figure 22. Vulnerabilities tab**

## Export

The **Export** icon lets you export the current list in either *CSV* or Microsoft Excel format.

## Only unresolved

The **Only unresolved** toggle lets you filter the column to only show unresolved vulnerabilities.

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Variables

*The **Variables** page shows detailed information for the variables hosted by the related asset.*

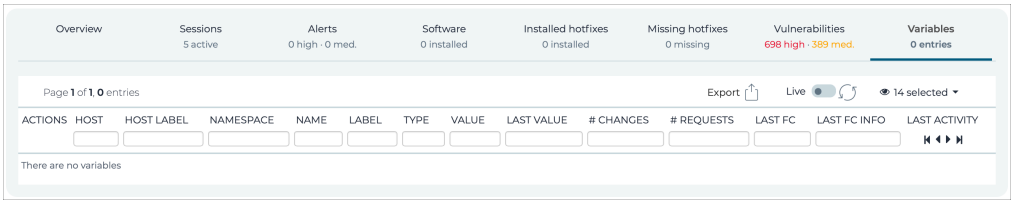



**Figure 23. Variables tab**

## Export

The **Export** icon lets you export the current list in either *CSV* or Microsoft Excel format.

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Chapter 5. Queries

*You can use the Nozomi Networks Query Language (N2QL) syntax to create complex data processes to obtain, filter, and analyze lists of information from the Nozomi Networks software.*

In *Nozomi Networks Query Language (N2QL)*, queries consist of:

- Data sources (on page 61)
- Commands (on page 66)
- Functions (on page 74)

## Data sources

Queries start by calling a data source. For example:

```
nodes | sort received.bytes desc | head
```

This query will show, in table format, the first 10 nodes that received the most bytes. If you add the `pie` command at the end of the query, the results will show in a pie chart format, where each slice has `node id` as the label and the `received.bytes` field as data.
For example:

```
nodes | sort received.bytes desc | head | pie ip received.bytes
```

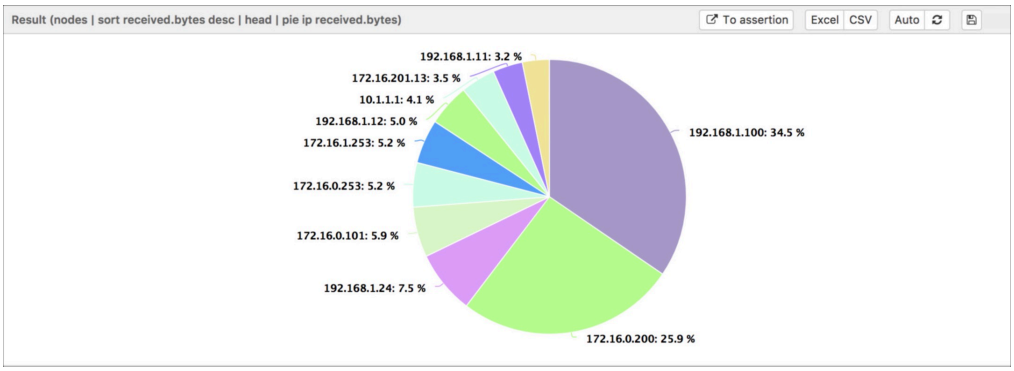



**Figure 24. Queries example**

## Functions

You might not achieved your desired result just using queries. Consequently, query syntax supports functions. With functions, you can apply calculations to the fields and use the results as a new temporary field. For example, the query:

```
nodes | sort sum(sent.bytes,received.bytes) desc | column ip
 sum(sent.bytes,received.bytes)
```

uses the `sum` function to `sort` on the aggregated parameters, which produces a chart with the columns representing the sum of the `sent` and `received` bytes.

## Prefix

The `$` is a prefix that changes the interpretation of the right hand side (rhs) of a `where` clause. By default, the rhs is interpreted as a string. With the `$` prefix, the interpretation of the rhs changes to a field name.

For example, in a query such as:

```
nodes | where id == 17.179.252.2
```

the right side of the `==` is expected to be a constant. If you create a query such as:

```
nodes | where id == id
```

the query tries to match all of the nodes having `id` equal to the string `id`.
If, however, you use the `$`, the second field is interpreted as a field, not a constant:

```
nodes | where id == $id
```

and returns the full list of records.

# Data sources

These are the available data sources with which you can start a query.

| | |
|---|---|
| alerts | Raised events |
| appliances | Downstream connected sensors synchronizing data to this, local one |
| assertions | Assertions saved by the users. An assertion represents an automatic check against other query sources |
| assets | Identified assets. Assets represent a local (private), physical system to care about, and can be composed of one or more Nodes. Broadcast nodes, grouped nodes, internet nodes, and similar cannot be Assets accordingly |
| audit_log | System's log for important operational events, e.g., login, backup creation, etc. |
| captured_files | Files reconstructed for analysis |
| captured_logs | Logs captured passively over the network |
| captured_urls | URLs and other protocol calls captured over the network. Access to files, requests to DNS, requested URLs and other are available in this query source |
| cpe_items | CPE maps definitions |
| cve_files | CVE definitions |
| dhcp_leases | IP to Mac bindings due to the presence of DHCP |
| function_codes | Protocols' function codes used in the environment |
| health_log | System's Health-related events, e.g. high resource utilization or hardware-related issues or events |
| link_events | Events that can occur on a Link, like it being available or not |
| links | Identified links, defined as directional one-to-one associations with a single protocol (i.e. source, destination, protocol) |
| microsoft_hotfixes | Microsoft hotfix information |

| | |
|---|---|
| node_cpe_changes | Common Platform Enumeration changes identified over known nodes. On the event of update of a CPE (on hardware, operating system and software versions), an entry in this query source is created to keep track of software updates or better detection of software |
| node_cpes | Common Platform Enumeration identified on nodes (hardware, operating system and software versions) |
| node_cves | Common Vulnerability Exposures: vulnerabilities associated to identified nodes' CPEs |
| node_points | Data points extracted over time, via Smart Polling or via Arc, from monitored Nodes |
| node_points_last | node_points last samples per each included data point |
| nodes | Identified nodes, where a node is an L2 or L3 (and above) entity able to speak some protocol |
| packet_rules | Packet rules definitions |
| protocol_connections | Identified protocol handhsakes/connections needed to decode process variables |
| report_files | Generated report files available for consultation |
| report_folders | Generated report folders |
| sessions | Sessions with recent network actvity. A Session is a specific application-level connection between nodes. A Link can hold one or more Session at a given time |
| sessions_history | Archived sessions |
| sigma_rules | Sigma rules definitions |
| sp_executions | Executions of Smart Polling plans |
| sp_node_executions | Results of Smart Polling plans executions per node |
| stix_indicators | STIX definitions |
| subnets | Identified network subnets |
| threat_models | Threat Modeling definitions |
| trace_requests | Trace requests in processing |

| | |
|---|---|
| variable_history | Process variables' history of values |
| variables | Identified process variables |
| yara_rules | YARA rules definitions |
| zone_links | A list of protocols exchanged by the defined zones |
| zones | Defined network zones |

# Basic operators

| | |
|---|---|
| **Operator** | `\|` (pipe, AND logical operator) |
| **Description** | Add a where clause with a logical AND, append it using the pipe character (`\|`). For example, the query below returns links that are from 192.168.254.0/24 `AND` going to 172.217.168.0/24. |
| **Example** | **links \| where from in_subnet? 192.168.254.0/24 \| where to in_subnet? 172.217.168.0/24** |

| | |
|---|---|
| **Operator** | `OR` |
| **Description** | To add a where clause with a logical OR, append it using the OR operator. For example, the query below returns links with either the http `OR` the https protocols. |
| **Example** | **links \| where protocol == http OR protocol == https** |

| | |
|---|---|
| **Operator** | `!` (exclamation point, NOT logical operator) |
| **Description** | Put an exclamation point (`!`) before a term to negate it. For example, the query below returns links that do NOT (`!`) belong to 192.168.254.0/24. |
| **Example** | **nodes \| where ip !in_subnet? 192.168.254.0/24 \| count** |

| | |
|---|---|
| **Operator** | `->` |
| **Description** | To change a column name, select it and use the `->` operator followed by the new name. It is worth noting that specific suffixes are parsed and used to visualize the column content differently. For example:<br>• `_time` data is shown in a timestamp format (1647590986549 becomes 2022-03-18 09:09:46.549)<br>• `_bytes` adds KB or MB, as applicable (50 becomes 50.0 B)<br>• `_percent` adds a percentage sign (50 becomes 50%)<br>• `_speed` adds a throughput speed in Mb/s (189915 becomes 1.8 Mb/s)<br>• `_date` converts numbers into a date format (2022-06-22 15:43:31.297 becomes 2022-06-2214:24:09.280 becomes 2022-06-24 (current day))<br>• `_packets` adds pp after the number of packets (50 becomes 50 pp) |
| **Example 1** | `nodes \| select created_at created_at->my_integer \| where my_integer > 946684800000` |
| **Example 2** | `nodes \| select created_at->my_creation_time` |

| | |
|---|---|
| **Example 3** | `nodes \| select tcp_retransmission.bytes->my_retrans_bytes` |

| | |
|---|---|
| **Operators** | ==, =, <, >, <=, and >= |
| **Description** | Queries support the mathematical operators listed above. |

| | |
|---|---|
| **Operator** | `"` (Quotation marks) |
| **Description** | Use quotation marks (`"`) to specify an empty string. Consider these two cases where this technique is useful:<br>• Finding non-empty values. Example 1 below returns assets where the `os` field is not blank.<br>• Specifying that a value in the query is a string (if its type is ambiguous). Example 2 below tells concat to treat the `"--"` parameter as a fixed string to use rather than as a field from the **alerts** table. |
| **Example 1** | `assets \| where os != ""` |
| **Example 2** | `alerts \| select concat(id_src,"--",id_dst)` |

| | |
|---|---|
| **Operator** | `in?` |
| **Description** | `in?` is only used with arrays; the field `type` must be an array. The query looks for the text strings you specify using `in?` and returns arrays that match one of them.<br>The example below uses `in?` to find any node having `computer` or `printer` as elements in the array. |
| **Example** | `assets \| where type in? ["computer","printer_scanner"]` |

| | |
|---|---|
| **Operator** | `include?` |
| **Description** | The query looks for the text string you specify using `include?` and returns strings that match it.<br>The example below uses `include?` to find assets where the **os** field contains the string **Win**. |
| **Example** | `assets \| where os include? Win` |

# Commands

| | |
|---|---|
| **Syntax** | `select <field1> <field2> ... <fieldN>` |
| **Parameters** | • the list of field(s) to output |
| **Description** | The select command takes all the input items and outputs them with only the selected fields |

| | |
|---|---|
| **Syntax** | `exclude <field1> <field2> ... <fieldN>` |
| **Parameters** | • the list of field(s) to remove from the output |
| **Description** | The exclude command takes all the input items and outputs them without the specified field(s) |

| | |
|---|---|
| **Syntax** | **where <field> <==\|!=\|<\|>\|<=\|>=\|in?\|include?\|start_with?\|end_with?\|in_subnet?> <value>** |
| **Parameters** | • field: the name of the field to which the operator will be applied<br>• operator<br>• value: the value used for the comparison. It can be a number, a string, or other data type. Advanced operators can use other data types, such as:<br>◦ a list (using JSON syntax) when using the in? operator, for example: `nodes \| where ip in? ["172.18.41.44"]`<br>◦ another property when using the '$' symbol, for example: `nodes \| where ip != $id` |
| **Description** | The where command will send to the output only the items which fulfill the specified criterion, many clauses can be concatenated using the boolean **OR** operator |
| **Example** | • `nodes \| where roles include? consumer OR zone == office`<br>• `nodes \| where ip in_subnet? 192.168.1.0/24`<br>• `<value>` can also be another `<field>`, as in: `links \| where from_zone == $to_zone \| select from_zone to_zone` |

| | |
|---|---|
| **Syntax** | `sort <field> [asc\|desc]` |
| **Parameters** | • field: the field used for sorting<br>• asc\|desc: the sorting direction |

| | |
|---|---|
| **Description** | The sort command will sort all the items according to the field and the direction specified, it automatically understands if the field is a number or a string |

| | |
|---|---|
| **Syntax** | `group_by <field> [ [avg\|sum] [field2] ]` |
| **Parameters** | • field: the field used for grouping<br>• avg\|sum: if specified, the relative operation will be applied on field2 |
| **Description** | The group_by command will output a grouping of the items using the field value. By default the output will be the count of the occurrences of distinct values. If an operator and a **field2** are specified, the output will be the average or the sum of the **field2** values |

| | |
|---|---|
| **Syntax** | `head [count]` |
| **Parameters** | • count: the number of items to output |
| **Description** | The head command will take the first **count** items, if **count** is not specified the default is 10 |

| | |
|---|---|
| **Syntax** | `uniq [<field1> <field2> ... <fieldN>]` |
| **Parameters** | • an optional list of fields on which to calculate the uniqueness |
| **Description** | The uniq command will remove from the output the duplicated items |

| | |
|---|---|
| **Syntax** | `expand <field>` |
| **Parameters** | • field: the field containing the list of values to be expanded |
| **Description** | The expand command will take the list of values contained in **field** and for each of them it will duplicate the original item substituting the original **field** value with the current value of the iteration |

| | |
|---|---|
| **Syntax** | `expand_recursive <field>` |
| **Parameters** | • field: the field to be recursively expanded |

| | |
|---|---|
| **Description** | The expand_recursive command will recursively parse the content of **field**, expanding each array or json structure until a scalar value is found. It generates a new row for each array element or json field. For each new row, it duplicates the original item substituting the original **field** value with the current value of the iteration and adding a new field that represents the current iteration path from the root |

| | |
|---|---|
| **Syntax** | `sub <field>` |
| **Parameters** | • field: the field containing the list of objects |
| **Description** | The sub command will output the items contained in **field** |

| | |
|---|---|
| **Syntax** | `count` |
| **Parameters** | |
| **Description** | The count command outputs the number of items |

| | |
|---|---|
| **Syntax** | `pie <label_field> <value_field>` |
| **Parameters** | • label_field: the field used for each slice label<br>• value_field: the field used for the value of the slice, must be a numeric field |
| **Description** | The pie command will output a pie chart according to the specified parameters |

| | |
|---|---|
| **Syntax** | `column <label_field> <value_field ...>` |
| **Parameters** | • label_field: the field used for each column label<br>• value_field: one or more field used for the values of the columns |
| **Description** | The column command will output a histogram; for each label a group of columns is displayed with the value from the specified value_field(s). The variant `column_colored_by_label` returns bars of different colors depending on their labels. |

| | |
|---|---|
| **Syntax** | `history <count_field> <time_field>` |
| **Parameters** | • count_field: the field used to draw the Y value<br>• time_field: the field used to draw the X points of the time series |

| | |
|---|---|
| **Description** | The history command will draw a chart representing an historic series of values |

| | |
|---|---|
| **Syntax** | `distance <id_field> <distance_field>` |
| **Parameters** | • id_field: the field used to tag the resulting distances.<br>• distance_field: the field on which distances are computed among entries. |
| **Description** | The distance command calculates a series of distances (that is, differences) from the original series of `distance_field`. Each distance value is calculated as the difference between a value and its subsequent occurrence, and tagged using the **id_field**.<br>For example, assuming we're working with an **id** and a **time** field, entering **alerts \| distance id time** returns a table where each distance entry is characterised by the **from_id**, **to_id**, and **time_distance** fields that represent time differences between the selected alerts. |

| | |
|---|---|
| **Syntax** | `bucket <field> <range>` |
| **Parameters** | • field: the field on which the buckets are calculated<br>• range: the range of tolerance in which values are grouped |
| **Description** | The bucket command will group data in different buckets, different records will be put in the same bucket when the values fall in the same multiple of <range> |

| | |
|---|---|
| **Syntax** | `join <other_source> <field> <other_source_field>` |
| **Parameters** | • other_source: the name of the other data source<br>• field: the field of the original source used to match the object to join<br>• other_source_field: the field of the other data source used to match the object to join |
| **Description** | The join command will take two records and will join them in one record when <field> and <other_source_field> have the same value |

| | |
|---|---|
| **Syntax** | `gauge <field> [min] [max]` |
| **Parameters** | • field: the value to draw<br>• min: the minimum value to put on the gauge scale<br>• max: the maximum value to put on the gauge scale |
| **Description** | The gauge command will take a value and represent it in a graphical way |

| | |
|---|---|
| **Syntax** | `value <field>` |
| **Parameters** | • field: the value to draw |
| **Description** | The value command will take a value and represent it in a textual way |

| | |
|---|---|
| **Syntax** | `reduce <field> [sum\|avg]` |
| **Parameters** | • field: the field on which the reduction will be performed<br>• sum or avg: the reduce operation to perform, it is sum if not specified |
| **Description** | The reduce command will take a series of values and calculate a single value |

| | |
|---|---|
| **Syntax** | size() |
| **Parameters** | • field: the field to calculate the size of |
| **Description** | If the field is an array, then the size function returns the number of entries in the array. If the field contains a string, then the size function returns the number of characters in the string.<br>**Note**: The size function may only be used on the following data sources: alerts, assets, captured_files, links, nodes, packet_rules, sessions, stix_indicators, subnets, variables, yara_rules, zones, and zone_links. |
| **Example:** | `assets \| where size(ip) > 1` |

## Nodes-specific commands reference

| | |
|---|---|
| **Syntax** | `where_node <field> < ==\|!=\|<\|>\|<=\|>=\|in?\|include?\|exclude?\| start_with?\|end_with? > <value>` |
| **Parameters** | • field: the name of the field to which the operator will be applied<br>• operator<br>• value: the value used for the comparison. It can be a number, a string or a list (using JSON syntax), the query engine will understand the semantics. |
| **Description** | The where_node command will send to the output only the items which fulfill the specified criterion, many clauses can be concatenated using the boolean **OR** operator. The where_node command is similar to the where command, but the output will also include all the nodes that are communicating directly with the result of the search.<br>**Note**: This command is only applicable to the nodes table. |

| | |
|---|---|
| **Syntax** | `where_link <field> < ==\|!=\|<\|>\|<=\|>=\|in?\|include?\|exclude?\| start_with?\|end_with? > <value>` |
| **Parameters** | • field: the name of the links table's field to which the operator will be applied.<br>• operator<br>• value: the value used for the comparison. It can be a number, a string or a list (using JSON syntax) the query engine will understand the semantics. |
| **Description** | The where_link command will send to the output only the nodes which are connected by a link fulfilling the specified criterion. Many clauses can be concatenated using the boolean **OR** operator.<br>**Note**: This command is only applicable to the nodes table. |

| | |
|---|---|
| **Syntax** | `graph [node_label:<node_field>]`<br>`[node_perspective:<perspective_name>]`<br>`[link_perspective:<perspective_name>]` |

| | |
|---|---|
| **Parameters** | • node_label: add a label to the node, the label will be the content of the specified node field<br>• node_perspective: apply the specified node perspective to the resulting graph. Valid node perspective values are:<br>◦ roles<br>◦ zones<br>◦ transferred_bytes<br>◦ not_learned<br>◦ public_nodes<br>◦ reputation<br>◦ appliance_host<br>• link_perspective: apply the specified link perspective to the resulting graph. Valid link perspectives are:<br>◦ transferred_bytes<br>◦ tcp_firewalled<br>◦ tcp_handshaked_connections<br>◦ tcp_connection_attempts<br>◦ tcp_retransmitted_bytes<br>◦ throughput<br>◦ interzones<br>◦ not_learned |
| **Description** | The graph command renders a network graph by taking some nodes as input. |

## Link-events-specific commands reference

| | |
|---|---|
| **Syntax** | `availability` |
| **Parameters** | |
| **Description** | The availability command computes the percentage of time a link is UP. The computation is based on the link events UP and DOWN that are seen for the link. |

| | |
|---|---|
| **Syntax** | `availability_history <range>` |
| **Parameters** | • range: the temporal window in milliseconds to use to group the link events |
| **Description** | The availability_history command computes the percentage of time a link is UP by grouping the link events into many buckets. Each bucket will include the events of the temporal window specified by the range parameter. |

| | |
|---|---|
| **Syntax** | `availability_history_month <months_back> <range>` |
| **Parameters** | • months_back: number of months to go back in regards to the current month to group the link events<br>• range: the temporal window in seconds to use to group the link events |
| **Description** | The availability_history command computes the percentage of time a link is UP by grouping the link events into many buckets. Each bucket will include the events of the temporal window specified by the range and months parameters. |

# Functions

Functions are always used in conjunction with other commands, such as `select`. In the following examples, functions are shown in **bold**:

- Combining functions with `select: nodes | select id type` **`color`**`(type)`
- Combining functions with `where: nodes | where` **`size`**`(label) > 10`
- Combining functions with `group_by: nodes | group_by` **`size`**`(protocols)`

Here is the complete list of functions:

| | |
|---|---|
| **Syntax** | `abs(<field>)` |
| **Parameters** | • the field on which to calculate the absolute value |
| **Description** | The abs function returns the absolute value of the field |

| | |
|---|---|
| **Syntax** | `bitwise_and(<numeric_field>,<mask>)` |
| **Parameters** | • numeric_field: the numeric field on which apply the mask<br>• mask: a number that will be interpreted as a bit mask |
| **Description** | The bitwise_and function calculates the bitwise & operator between the numeric_field and the mask entered by the user |

| | |
|---|---|
| **Syntax** | `coalesce(<field1>,<field2>,...)` |
| **Parameters** | • a list of fields or string literals in the format "<chars>" |
| **Description** | The coalesce function will output the first value that is not null |

| | |
|---|---|
| **Syntax** | `color(<field>)` |
| **Parameters** | • field: the field on which to calculate the color |
| **Description** | The color function generates a color in the rgb hex format from a value |
| **Note** | Only available for nodes, links, variables and function_codes |

| | |
|---|---|
| **Syntax** | `concat(<field1>,<field2>,...)` |
| **Parameters** | • a list of fields or string literals in the format "<chars>" |

| | |
|---|---|
| **Description** | The concat function will output the concatenation of the input fields or values |

| | |
|---|---|
| **Syntax** | `date(<time>)` |
| **Parameters** | • time defined as unix epoch |
| **Description** | The date function returns a date from a raw time |

| | |
|---|---|
| **Syntax** | `day_hour(<time_field>)` |
| **Parameters** | • time_field: the field representing a time |
| **Description** | The day_hour function returns the hour of the day plus the sensor's local time offset from UTC, i.e. a value in the range 0 through 23. Be careful when accounting for daylight saving time. Use **day_hour_utc** when absolute precision is desired |

| | |
|---|---|
| **Syntax** | `day_hour_utc(<time_field>)` |
| **Parameters** | • time_field: the field representing a time |
| **Description** | The **day_hour_utc** function returns the hour of the day expressed in UTC for the current time field, i.e. a value in the range 0 through 23 |

| | |
|---|---|
| **Syntax** | `days_ago(<time_field>)` |
| **Parameters** | • time_field: the field representing a time |
| **Description** | The days_ago function returns the amount of days passed between the current time and the time field value |

| | |
|---|---|
| **Syntax** | `dist(<field1>,<field2>)` |
| **Parameters** | • the two fields to compute the distance on |
| **Description** | The dist function returns the distance between field1 and field2, which is the absolute value of their difference |

| | |
|---|---|
| **Syntax** | `div(<field1>,<field2>)` |

| | |
|---|---|
| **Parameters** | • field1 and field2: the two field to divide |
| **Description** | The div function will calculate the division field1/field2 |

| | |
|---|---|
| **Syntax** | `hours_ago(<time_field>)` |
| **Parameters** | • time_field: the field representing a time |
| **Description** | The hours_ago function returns the amount of hours passed between the current time and the time field value |

| | |
|---|---|
| **Syntax** | `is_empty(field) == true \| false` |
| **Parameters** | • field: the field to check to evaluate whether it is empty or not |
| **Description** | The is_empty command takes a field as input and returns only the entries that are either empty / not empty. |
| **Example** | **nodes \| where is_empty(label) == false** |

| | |
|---|---|
| **Syntax** | `is_recent(<time_field>)` |
| **Parameters** | • time_field: the field representing a time |
| **Description** | The is_recent function takes a time field and returns true if the time is not farther than 30 minutes |

| | |
|---|---|
| **Syntax** | `minutes_ago(<time_field>)` |
| **Parameters** | • time_field: the field representing a time |
| **Description** | The minutes_ago function returns the amount of minutes passed between the current time and the time field value |

| | |
|---|---|
| **Syntax** | `mult(<field1>,<field2>,...)` |
| **Parameters** | • a list of fields to multiply |
| **Description** | The mult function returns the product of the fields passed as arguments |

| | |
|---|---|
| **Syntax** | `round(<field>,[precision])` |

| | |
|---|---|
| **Parameters** | • field: the numeric field to round<br>• precision: the number of decimal places |
| **Description** | The round function takes a number and outputs the rounded value |

| | |
|---|---|
| **Syntax** | `seconds_ago(<time_field>)` |
| **Parameters** | • time_field: the field representing a time |
| **Description** | The seconds_ago function returns the amount of seconds passed between the current time and the time field value |

| | |
|---|---|
| **Syntax** | `split(<field>,<splitter>,<index>)` |
| **Parameters** | • field: the field to split<br>• splitter: the character used to separate the string and produce the tokens<br>• index: the 0 based index of the token to output |
| **Description** | The split function takes a string, separates it and outputs the token at the <index> position |

| | |
|---|---|
| **Syntax** | `sum(<field>,...)` |
| **Parameters** | • a list of fields to sum |
| **Description** | The sum function returns the sum of the fields passed as arguments |

# Examples

## Pie chart

*An example on how to create a pie chart to understand the media access control (MAC) vendor distribution in a network.*

We choose nodes as our query source and we start to group the nodes by mac_vendor:

```
nodes | group_by mac_vendor
```

We can see the list of the vendors in our network associated with the occurrences count. To better understand our data we can use the sort command, so the query becomes:

```
nodes | group_by mac_vendor | sort count desc
```

In the last step we use the pie command to draw the chart with the mac_vendor as a label and the count as the value.

```
nodes | group_by mac_vendor | sort count desc | pie mac_vendor count
```

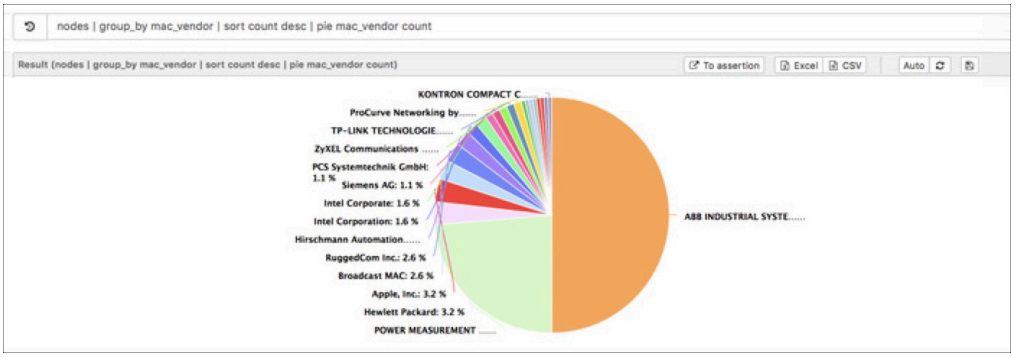



**Figure 25. Pie chart example**

## Column chart

*An example on how to create a column chart with the top nodes by traffic.*

To start, you need to get the nodes and select the:

- `id`
- `sent.bytes`
- `received.bytes`
- `sent.bytes`
- `received.bytes`

To calculate the sum , you need to use the `sum` function. The query is:

```
nodes | select id sent.bytes received.bytes
 sum(sent.bytes,received.bytes)
```

When you execute this query, the sum field has a very long name. You can rename it to be more comfortable with these commands:

```
nodes | select id sent.bytes received.bytes
 sum(sent.bytes,received.bytes)->sum
```

To obtain the top nodes by traffic, you can sort and take the first 10:

```
nodes | select id sent.bytes received.bytes
 sum(sent.bytes,received.bytes)->sum | sort sum desc | head 10
```

Finally, to display the data in a graphical way, you can use the `column` command:

```
nodes | select id sent.bytes received.bytes
 sum(sent.bytes,received.bytes)->sum | sort sum desc | head 10 | column
 id sum sent_bytes received_bytes
```

**Note:**
You can access an inner field of a complex type with the dot syntax, in the example the dot syntax is used on the fields `sent` and `received` to access their bytes sub field.

**Note:**
After accessing a field with the dot syntax, it will gain a new name to avoid ambiguity; the dot is replaced by an underscore. In the example `sent.bytes` become `sent_bytes`

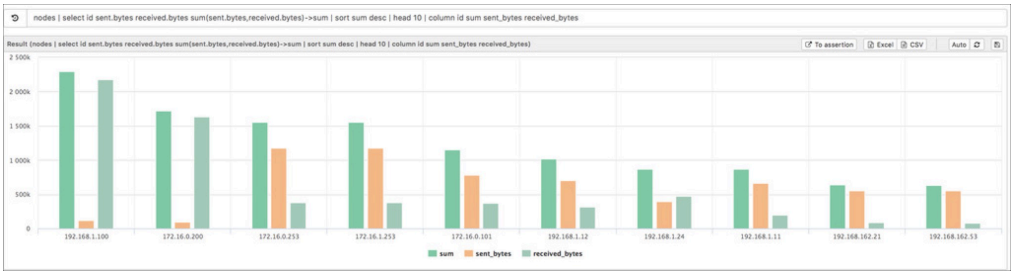



Figure 26. Column chart example

## Where with multiple conditions in OR

*An example of a query to get all the nodes with a specific role, in particular all the nodes which are web or domain name server (DNS) servers.*

With the `where` command, you can separate many conditions with `OR`

**Note:**
Because the roles field contains a list of values, you can use the `include?` operator to check if a value was contained in the list.

```
nodes | where roles include? web_server OR roles include? dns_server |
 select id roles
```


Figure 27. Where with multiple conditions in OR example

## Bucket and history

*An example of a query to calculate the distribution of link events towards an internet protocol (IP) address.*

You can filter all the `link_events` with `id_dst` equal to `192.168.1.11` After this you can sort by time, this is a very important step because bucket and history depend on how the data are sorted.

Then you can use `bucket` to group the data by time. The final step is to use the `history` command to draw a chart, we pass `count` as a value for the Y axis and `time` for the X axis.

The `history` command is particularly suited for displaying a big amount of data, in the image below we can see that there are many hours of data to analyze.

```
link_events | where id_dst == 192.168.1.11 | sort time asc | bucket time
 36000 | history count time
```

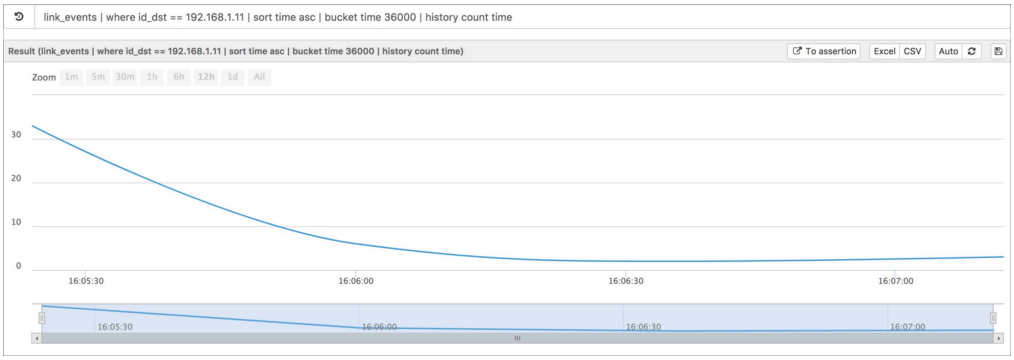



Figure 28. Bucket and history example

## Join

*An example query to join two data sources to obtain a new data source with more information. In particular, how to list the links with the labels for the source and destination nodes.*

You can match the `from` field of the links with the `id` field of the nodes to ask for the links, and join them with the nodes:

```
links | join nodes from id
```

After executing the query above you will get all the links fields, plus a new field called `joined_node_from_id`, it contains the node which satisfies the `link.from == node.id` condition. You can use the dot syntax to access the sub fields of `joined_node_from_id`

Because we also want to get the labels for the `to` field of the `links` you add another `join` and exclude the empty labels of the node referred by `to` to get more interesting data:

```
links | join nodes from id | join nodes to id | where
 joined_node_to_id.label != ""
```

This will obtain a huge amount of data, which is difficult to understand. To only get the relevant information, you can use a `select`:

```
links | join nodes from id | join nodes to id | where
 joined_node_to_id.label != "" | select from joined_node_from_id.label to
 joined_node_to_id.label protocol
```

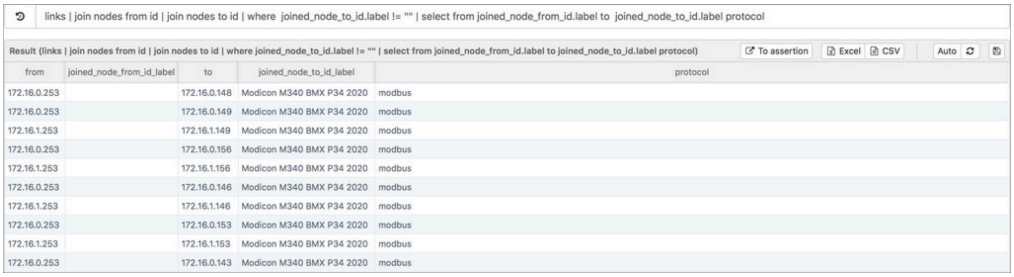



Figure 29. Join example

## Compute the availability history

*An example query to compute the availability history for a link.*

In order to achieve a reliable availability, it is recommended to enable the **Track availability** feature on the desired link.

Start from the `link_events` data source, filtered by source and destination ip in order to precisely identify the target link. Consider also filtering by protocol to achieve a higher degree of precision.

```
link_events | where id_src == 10.254.3.9 | where id_dst == 172.31.50.2
```

The next step is to sort the events by ascending time of creation. Without this step the `availability_history` might produce meaningless results, such as negative values. Then, to compute the `availability_history` with a bucket of 1 minute (60000 milliseconds), you can complete query is as follows:

```
link_events | where id_src == 10.254.3.9 | where id_dst == 172.31.50.2 |
 sort time asc | availability_history 60000
```

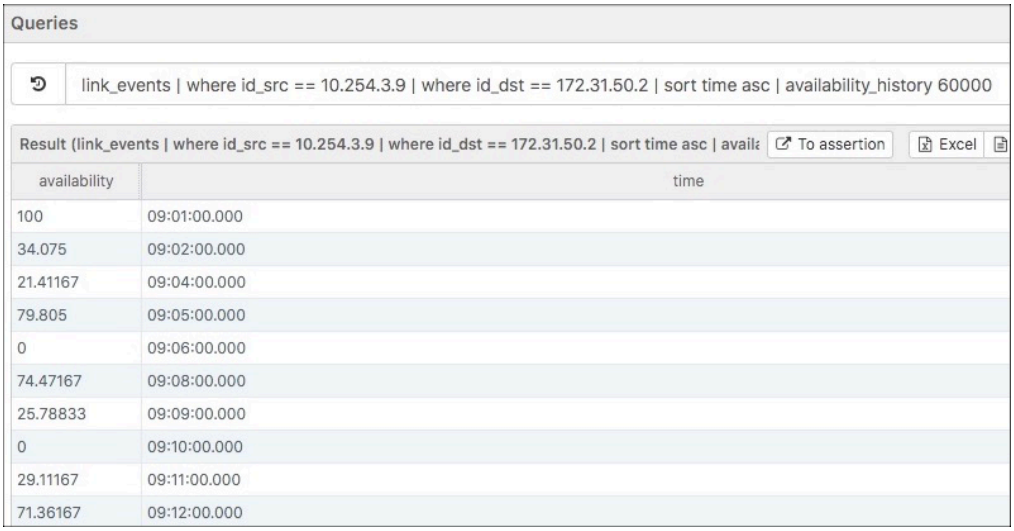



Figure 30. Availability history example

**Note:**
By default, `link_events` generation is disabled. To enable it, you can use the configuration rule described in **Configure links**.

## Complex field types

### Single scalar values

To query single scalar values, apply the commands that are explained in this section.

### Objects

Objects show in braces: {object}

```
{
"source": "ARP",
"likelihood": 1,
"likelihood_level": "confirmed"
}
```

An example on how to query only `confirmed` *MAC* addresses.

**Note:**
Possible values are:

- `confirmed`
- `likely`
- `not confirmed`

Since `mac_address:info` is an object, you can access subfields like `mac_address:info.likelihood_level` to apply the `where` condition:

```
nodes | select mac_address:info mac_address:info.likelihood_level | where
 mac_address:info.likelihood_level == confirmed
```

Since N2OS 24.1 is possible to access complex objects with a different syntax that is compatible with Vantage, using the `/` operator, the query specified above becomes:

```
nodes | select mac_address:info/likelihood_level | where
 mac_address:info.likelihood_level == "confirmed"
```

Note that also the `"confirmed"` literal can now be quoted and the query can be executed in Vantage without any change.

### Arrays

**Note:**
For example, a `parent` in the alerts table.

Arrays show in braces: {array}

```
[
"5b867836-2b41-4c15-ab6f-4ae5f0251e30"
]
```

An example on how to only query alerts that have a parent incident, with a known incident id with the value: `d36d0`

Since the `parents` field is an array, you can use `expand` first to get an entry for each parent, then apply your condition:

```
alerts | expand parents | where expanded_parents include? d36d0
```

### Object arrays

**Note:**
For example, `function_codes` in the links table.

Object arrays are a combination of the above examples. Therefore, they show an object included in a [{..},{..},.. ] :

```
[
{
"name": "M-SEARCH",
"is_learned": true,
"is_fully_learned": true
}
]
```

An example on how to query learned function codes.

Since `function_codes` is an object array, you can use `expand` first, to get an entry for each function code, then use the `.` operator (`function_code.is_learned`) to apply your `where` condition:

```
links | select from to protocol function_codes | expand function_codes |
 where expanded_function_codes.is_learned == true
```

# Chapter 6. Smart Polling in CMC

*The **Smart Polling** page gives you a read-only view of Smart Polling.*

In the *CMC*, Smart Polling plans are read-only. Plans created in Guardian can be synchronized, along with the associated data. By default, synchronization is disabled.

Smart Polling has these tabs:

- Plans (on page 88)
- Node points (on page 89)

# Plans

*The **Plans** page shows a list of Smart Polling plans and lets you manage plans and add new ones.*

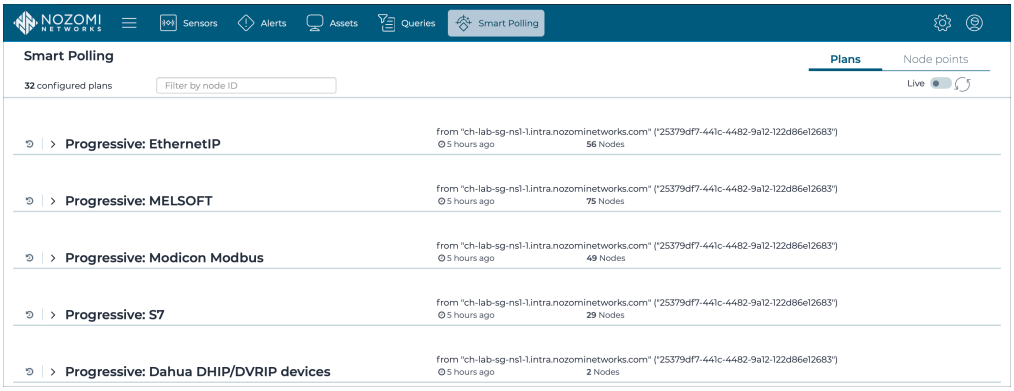
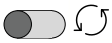



**Figure 31. Plans page**

## Filter

The filter field lets you use the node *ID* to filter the page.

## Live / refresh

The **Live**  icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

# Node points

*The **Node points** page shows all of the node points.*

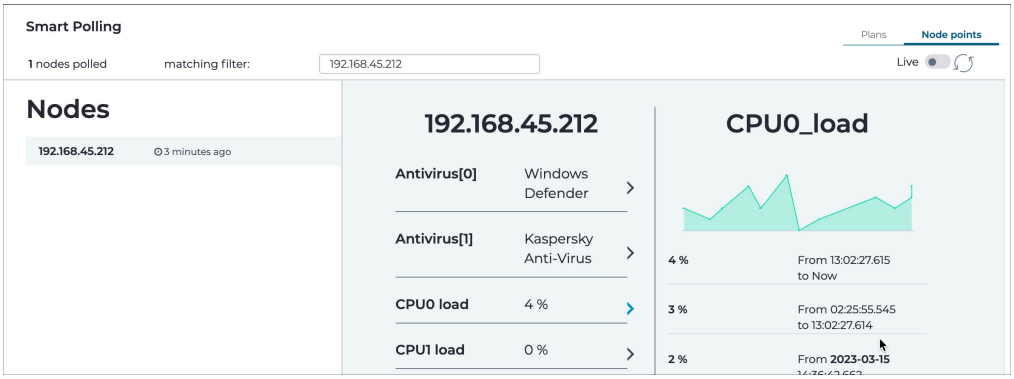



**Figure 32. Node points page**

# View the enriched information history for a node

*The **Node points** page lets you view the enriched information history for a node.*

## Procedure

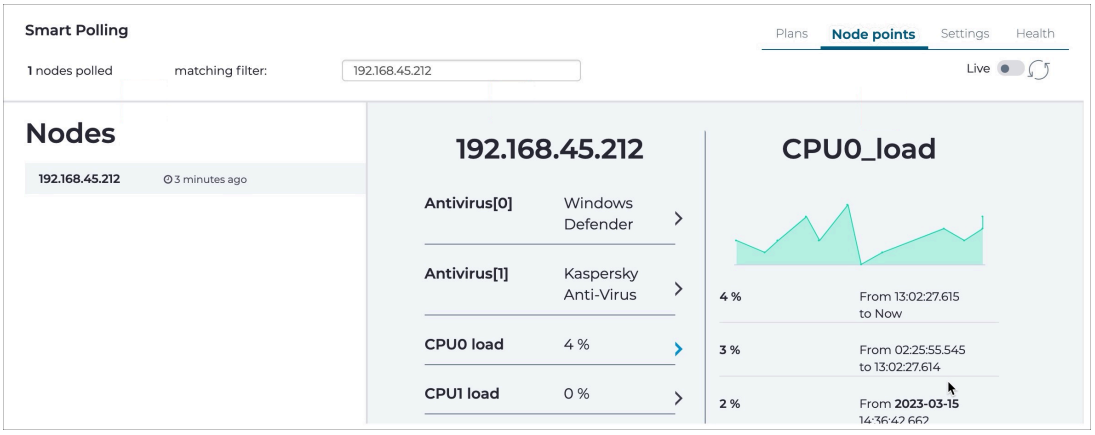
1. In the top navigation bar, select **Smart Polling**.

   **Result:** The **Smart Polling** page opens.

2. In the top right, select **Node points**.

   **Result:** The **Node points** page opens.

3. In the left column, select a node point.




**Result:** The second and third columns show an increased level of detail for the extracted information for that node point

# Chapter 7. Arc overview

**Arc™** *is a host-based sensor that detects and defends against malicious or compromised endpoints, and insider attacks. You can use Arc sensors to aggregate data for analysis and reports, either on-premises, or in the Vantage cloud.*

## General

When detecting cyberthreats, identifying vulnerabilities, or analyzing anomalies in your processes, it is critical to have as much detailed network and system information as possible. More accurate and timely access to data leads to better diagnostics and a faster time to repair.

Arc gives you enhanced endpoint data collection and asset visibility for your networks. This enhanced visibility gives you more:

- Vulnerability assessment capabilities
- Endpoint protection
- Traffic analysis capabilities
- Accurate diagnostics of in-progress threats and anomalies

Arc lets you easily identify compromised hosts that have:

- Malware
- Rogue applications
- Unauthorized *universal serial bus (USB)* devices
- Suspicious user activity

## Operating Systems (OS)

Arc sensors are endpoint executables that run on hosts on these *OSs*:

- Microsoft Windows
- Linux
- Apple macOS

The data that is collected can be sent to either Guardian or Vantage.

## Use cases and deployment scenarios

Arc lets you:

- Incorporate air-gapped devices into the analysis and reporting system
- Gain deeper intelligence or insight on critical endpoint devices
- Continuously monitor endpoints
- Automatically deploy sensors across thousands of devices
- Use a low-impact process to scan air-gapped networks
- Deploy with solutions

## Continuous monitoring

Because the Arc sensor is on the host, it can monitor traffic continuously, even when the device is not sending or receiving traffic.

## User-specific activity monitoring

With more access to endpoint data, Arc lets you connect network traffic and anomalies with specific users. This helps to identify potential insider threats and makes corrective actions both easier and quicker.

### Local behavioral analysis (Sigma rules)

Sigma is a common open-source standard that lets you analyze log files to identify malicious events. They are not necessarily related to network artifacts, and as such, would not be detected without residing on a machine. Nozomi Networks Labs curates all the Sigma rules that are loaded into Arc. A *Threat Intelligence (TI)* active license is needed to receive curated rules from the upstream Nozomi endpoint.

### Temporary deployment

It is not necessary to keep the Arc executable on a host after you have collected information. This means that you can remove it after data has been collected to conserve host resources, and maintain a clean host environment.

## Architecture

*It is important to understand the different architecture possibilities that are available with Arc.*

You can connect Arc:

- To Guardian
- To Vantage

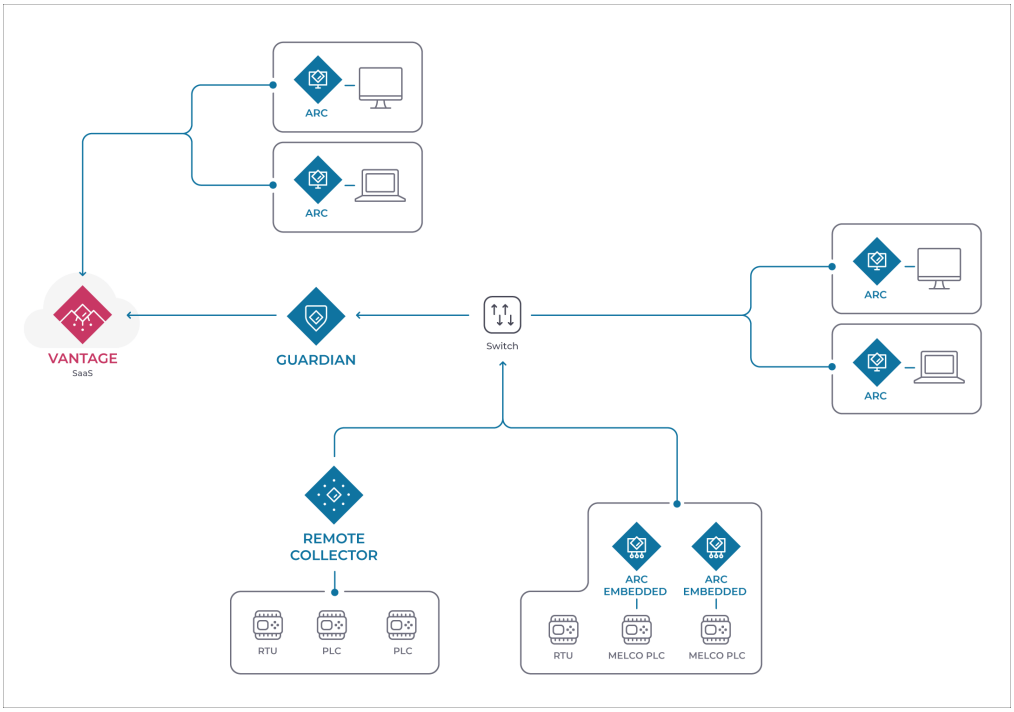



**Figure 33. Arc architecture example**

# Arc in CMC

*The **Arc** button in the Central Management Console (CMC) Web UI lets you access the different pages for Arc.*

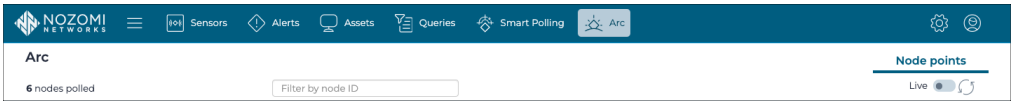



**Figure 34. Arc button in CMC Web UI**

When you select **Arc** in the *CMC* Web *user interface (UI)*, you get access to the **Node points** page.

# Node points

*The **Node points** page shows data points that are collected over time, and represent the state of the target machine.*

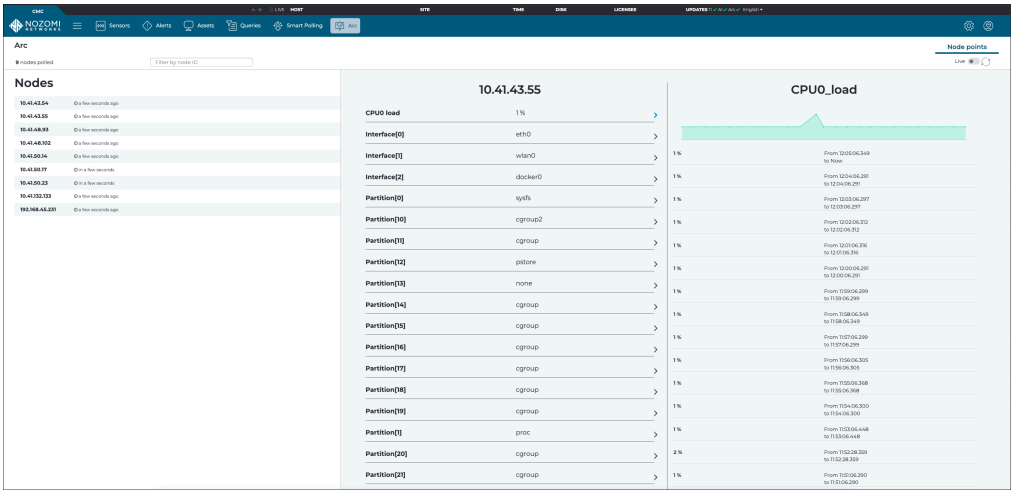



**Figure 35. Node points page in CMC**

## Node points count

This shows the number of the nodes polled.

## Filter by node ID

This field lets you use the node *ID* to filter the nodes.

## Live toggle

The **Live** toggle lets you change live view on, or off. When live mode is on, the page will refresh periodically.

## Refresh

The icon lets you immediately refresh the current view.

## Nodes

The list of nodes that show at least one node point.

# Chapter 8. Reports

*The **Reports** page lets you manage, generate, schedule and view reports.*

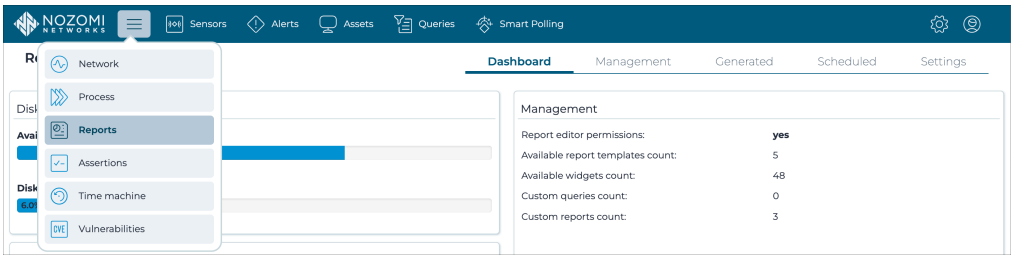



**Figure 36. Reports page**

The **Reports** page has these tabs:

- Dashboard (on page 100)
- Management (on page 101)
- Generated (on page 104)
- Scheduled (on page 105)
- Settings (on page 106)

# Dashboard

*The Dashboard page shows an overview of information related to reports, which includes disk availability, report settings, generated reports, report management, and scheduled reports.*

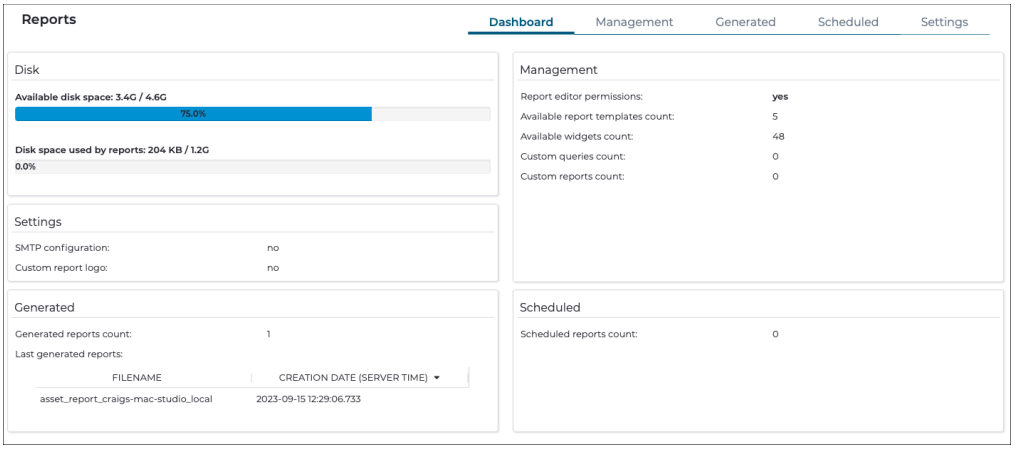



**Figure 37. Dashboard page**

## Disk

This section shows:

- Available disk space
- Disk space used by reports

## Management

This section shows a summary of information from the Management (on page 101) page.

## Settings

This section shows a summary of information from the Settings (on page 106) page.

## Generated

This section shows a summary of information from the Generated (on page 104) page.

## Scheduled

This section shows a summary of information from the Scheduled (on page 105) page.

# Management

*The **Management** page lets you manage all your reports.*

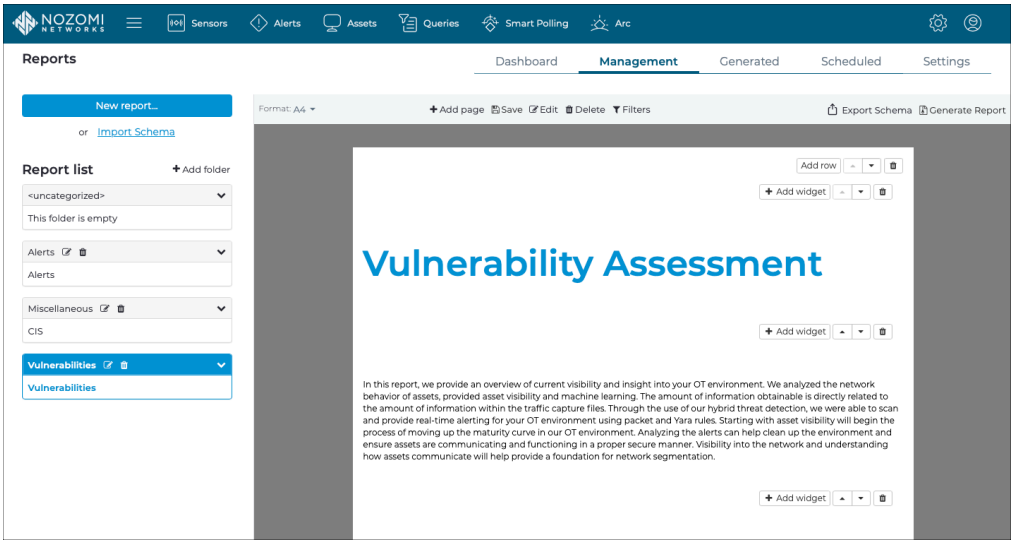



**Figure 38. Management page**

## New report

This button lets you Create a report (on page 107).

## Import schema

This button lets you Import a schema (on page 114).

## Report list

This section shows a list of created and saved reports. You can add folders (on page 112) to group the reports in.

## Add page

This button lets you add a page to the bottom of the current report.

## Save

This button lets you save the changes to the current report.

## Edit

This button lets you edit the current report.

## Delete

This button lets you delete the current report.

## Filters

This button lets you filter the contents of the current report.

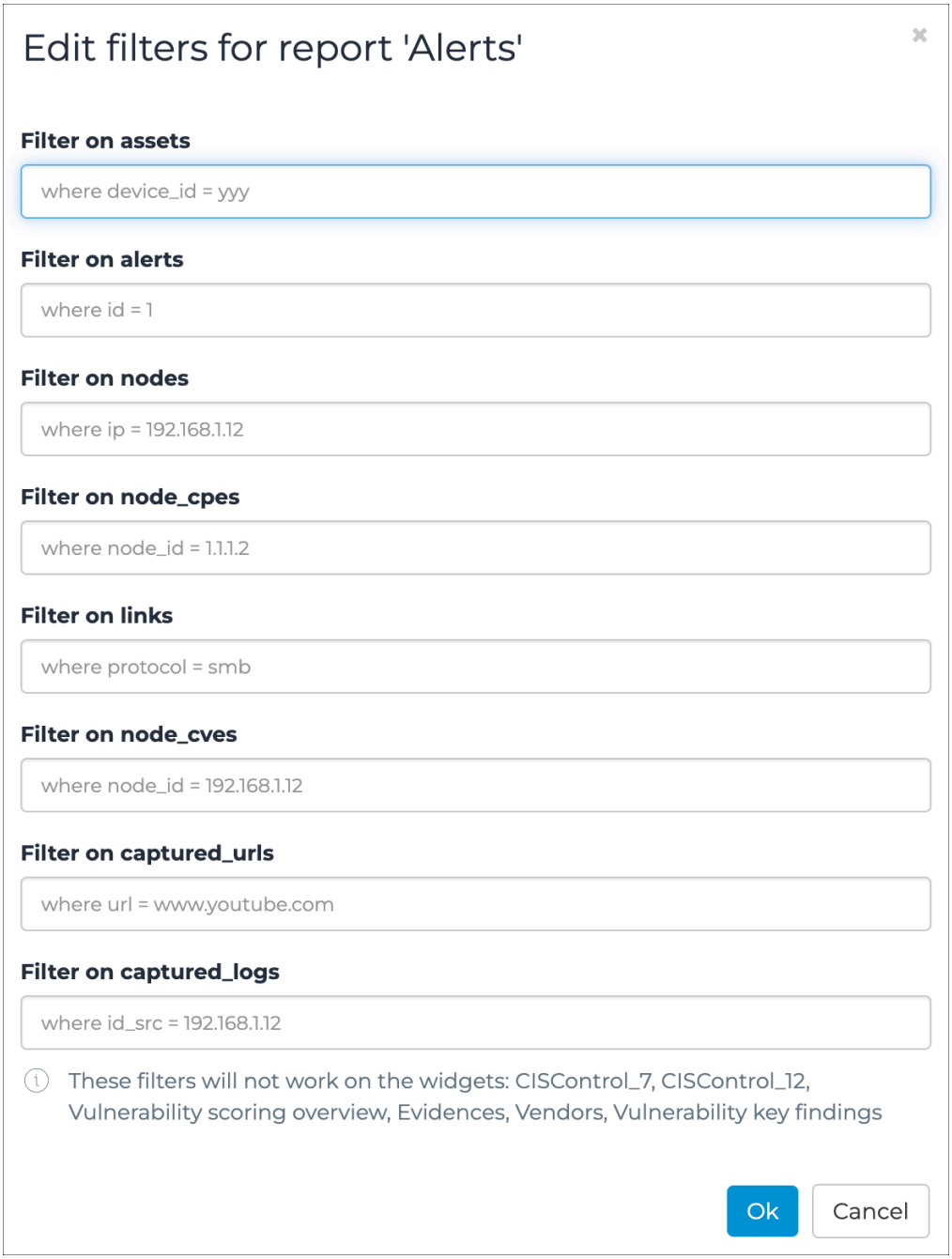



**Figure 39. Filter dialog**

### Export schema

This button lets you export a schema (on page 115) for the current report in *JavaScript Object Notation (JSON)* format.

### Generate report

The generate report  icon lets you generate a report (on page 109) in one of these formats:

- *PDF*
- *CSV*
- Microsoft Excel

# Generated

*The **Generated** page lets you view, download, edit, and delete generated reports.*

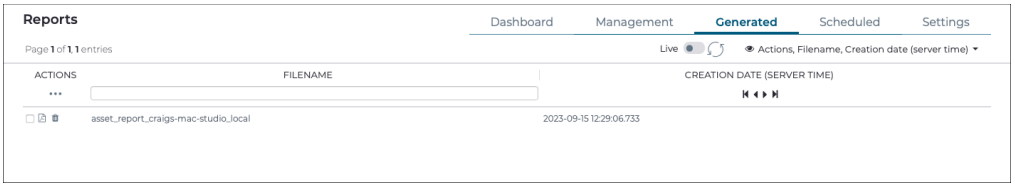



**Figure 40. Generated page**

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Scheduled

*The **Scheduled** page lets you view, download, edit, and delete scheduled reports.*

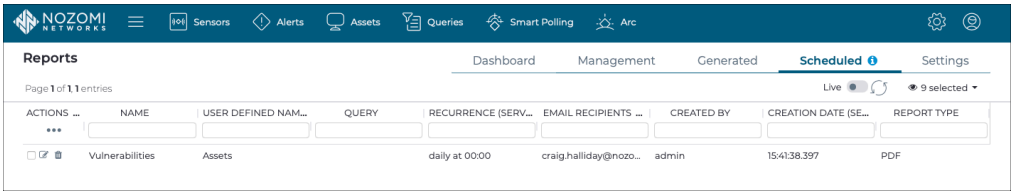



**Figure 41. Scheduled page**

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Settings

*The **Settings** page lets you change report settings, upload custom logos, and configure simple mail transfer protocol (SMTP) settings.*

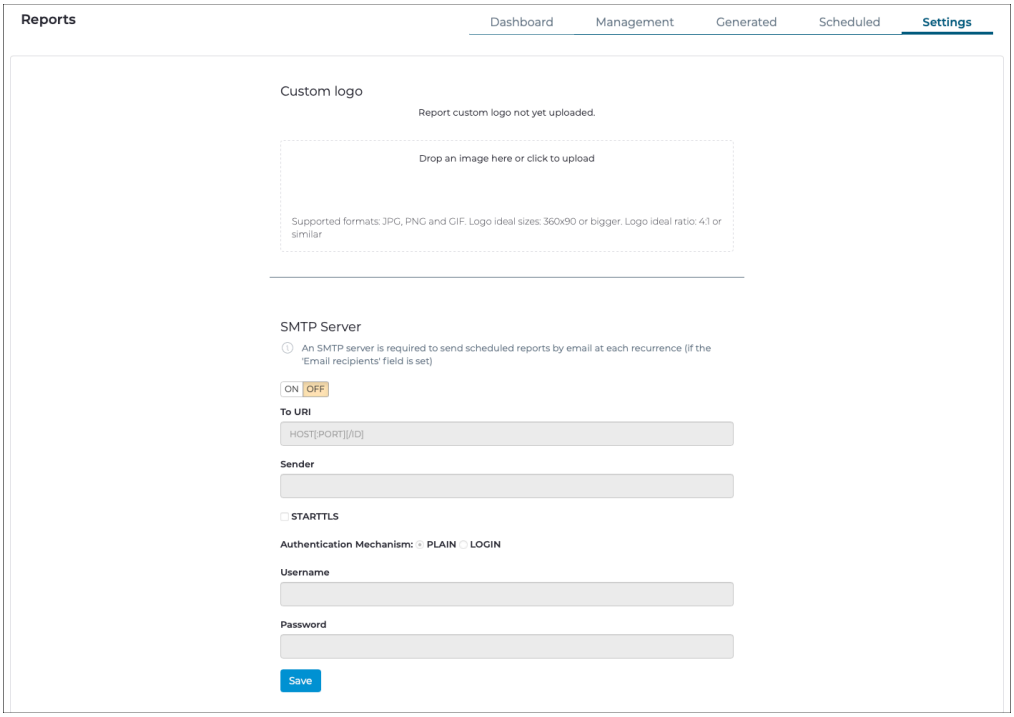



**Figure 42. Settings page**

## Custom logo

This section lets you upload a custom logo (on page 116) that will show in your reports.

## SMTP server

If you want to send emails that contain scheduled reports, you need to configure (on page 118) the *simple mail transfer protocol (SMTP)* server settings.

# Create a report

*The **Management** page lets you create a new report.*

## Procedure

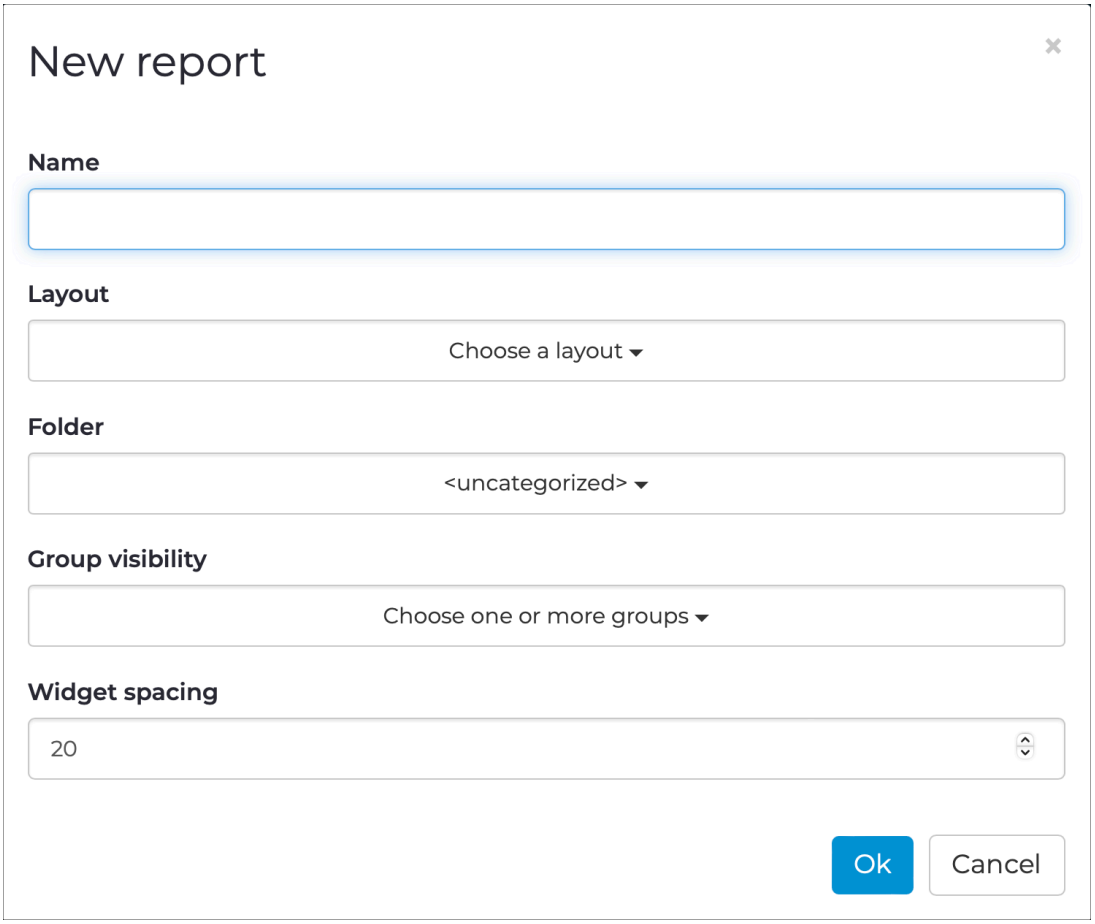
1. In the top navigation bar, select ☰ **icon > Reports**.

   **Result:** The **Reports** page opens.

2. Select **Management**.
3. In the section on the left, select **New report**.

   **Result:** A dialog shows.

4. In the **Name** field, enter a name for the report.




5. From the **Layout** dropdown, select a layout for the report.
6. From the **Folder** dropdown, select a folder for the report.
7. From the **Group visibility** dropdown, select the group(s) that will be able to view the report.

8. From the **Widget spacing** dropdown, enter a value.
9. Select **Ok**.

### Results

The report has been created.

# Generate a report

*You can generate both scheduled, or on-demand, reports, in multiple file formats.*

## Procedure

1. In the top navigation bar, select ☰ **icon > Reports**.

   **Result:** The **Reports** page opens.

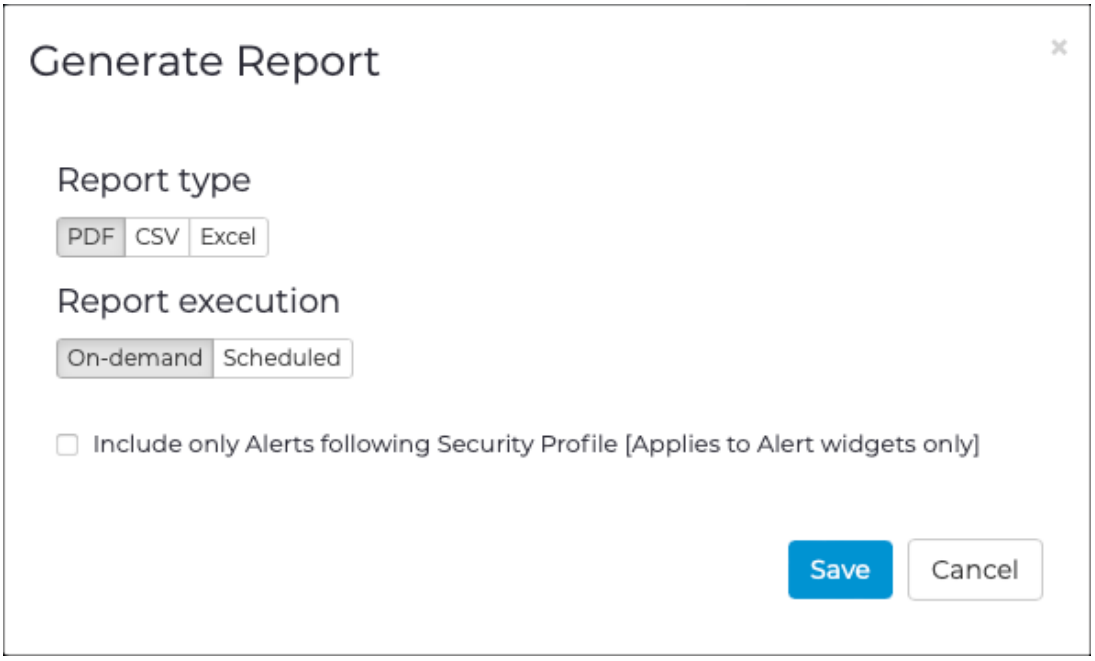
2. Select **Management**.
3. In the section on the left, select the report that you want to generate.
4. In the top right, select **Generate Report**.

   **Result:** A dialog shows.

5. In the **Report type** section, choose a format for the report:

   **Choose from:**

   - *PDF*
   - *CSV*
   - Excel




6. In the **Report execution** section, choose the type of execution:

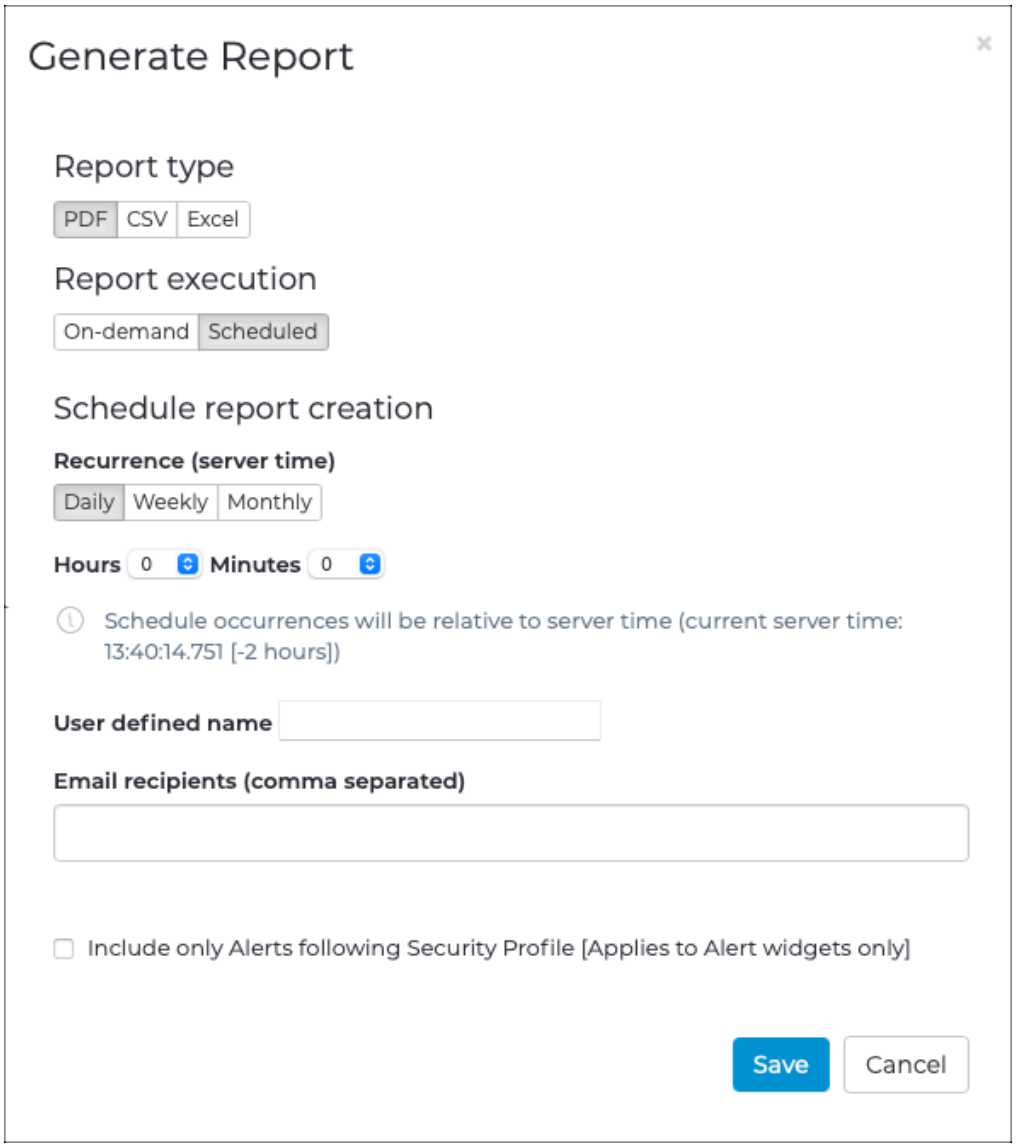
   **Choose from:**

   - On-demand
   - Scheduled

7. If you chose **On-demand**, select **Save**.

   **Result:** The report starts to generate. When the generation is complete, the report will show in the Generated (on page 104) page.

8. If you chose **Scheduled**, do the steps below.
   a. In the **Recurrence (server time)** section, enter the settings that you want.




   b. **Optional:** In the **User defined name** field, enter a name for the report.

   c. **Optional:** In the **Email recipients (comma separated)** field, enter the email addresses of the people that you would like to receive the reports.

   d. **Optional:** If necessary, select the **Include only Alerts following Security Profile [Applies to widgets only]** checkbox.

### Results

The report has been generated, or scheduled, as applicable.

## Download a report

*The **Generated** page lets you download reports.*

### Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Generated**.
3. To the left of the applicable report, select the download icon.

   **Result:** The download starts.

### Results

The report has been downloaded to your downloads folder.

## Delete a report

*The **Generated** page lets you delete generated reports.*

### Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Generated**.
3. To the left of the applicable report, select the download icon.

### Results

The report has been deleted.

# Add a folder

*The **Management** page lets you add folders for you to organize your reports.*

## Procedure

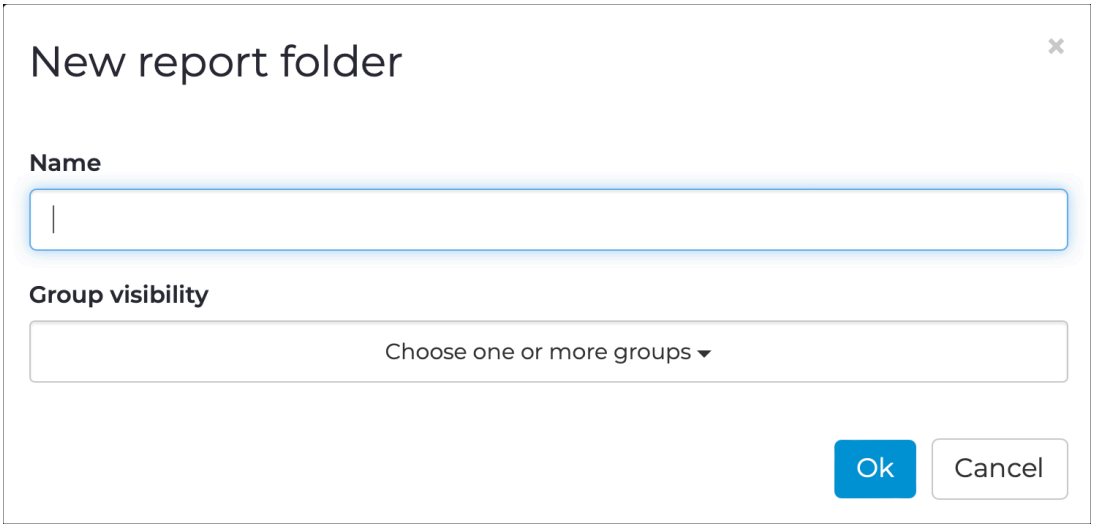
1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Management**.
3. In the **Reports list** section on the left, select **Add folder**.

   **Result:** A dialog shows.

4. In the **Name** field, enter a name for the folder.




5. From the **Group visibility** dropdown, select the group(s) that will be able to view the reports.
6. Select **Ok**.

## Results

The folder has been added.

# Edit a folder

*The **Management** page lets you delete reports.*

## Procedure

1. In the top navigation bar, select ☰ **icon > Reports**.

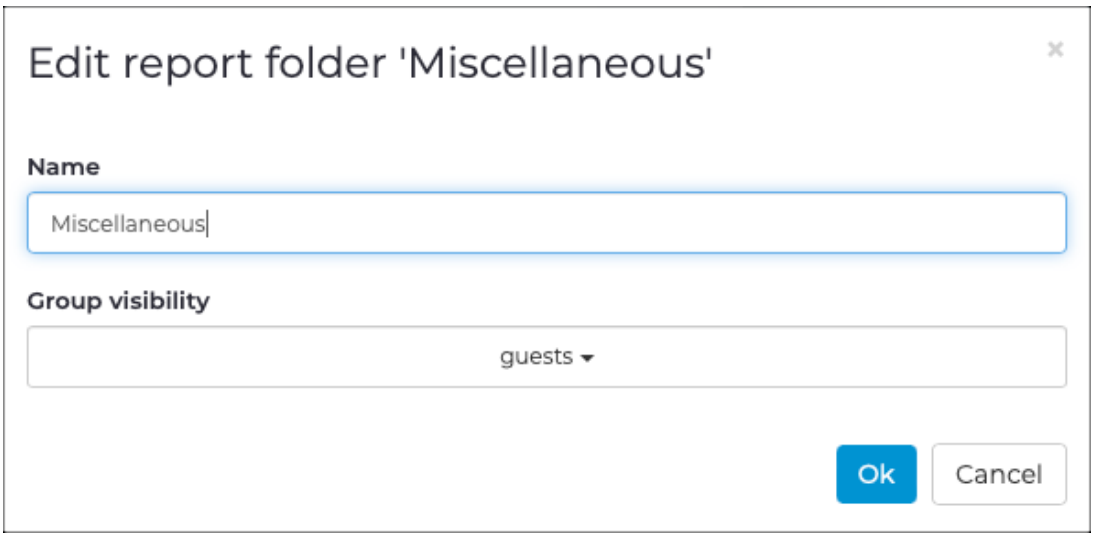
   **Result:** The **Reports** page opens.

2. Select **Management**.
3. In the section on the left, to the right of the applicable folder's name, select the download ✎ icon.

   **Result:** A dialog shows.

4. **Optional:**

   If necessary, in the **Name** filed, edit the name of the folder.




5. **Optional:** If necessary, in the **Group visibility** filed, edit the visibility of the folder.
6. Select **Ok**.

## Results

The report has been edited.

# Delete a folder

*The **Management** page lets you delete reports.*

## Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Management**.
3. In the section on the left, to the right of the applicable folder's name, select the download icon.

## Results

The report has been deleted.

# Import a schema

*The **Management** page lets you import a schema that has previously been exported.*

## Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Management**.
3. In the section on the left, select **Import Schema**.
4. Select the schema to import.

## Results

The schema has been imported.

# Export a schema

*The **Management** page lets you export a schema.*

## Procedure

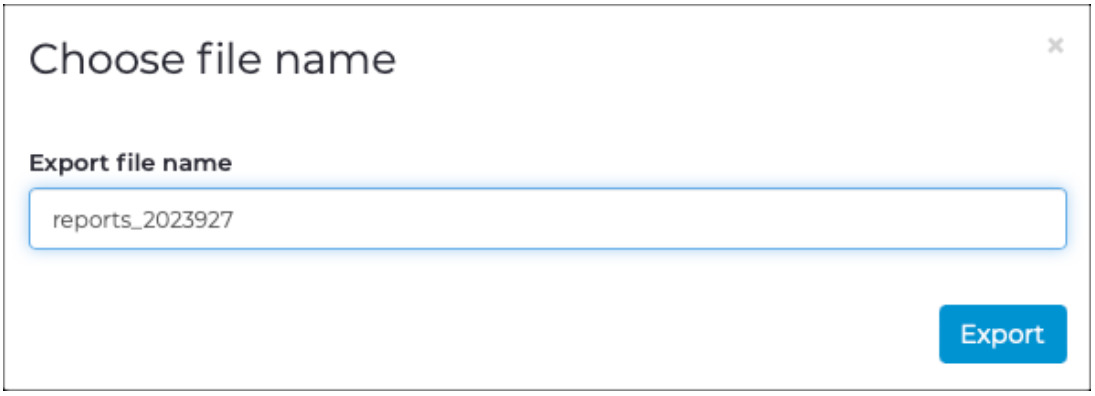
1. In the top navigation bar, select ☰ **icon > Reports**.

   **Result:** The **Reports** page opens.

2. Select **Management**.
3. In the top right, select **Export Schema**.

   **Result:** A dialog shows.

4. In the **Export file name** field, enter a name for the schema.




5. Select **Export**.

## Results

The schema has been exported in *JSON* format.

# Upload a custom logo

*You can add a custom logo that will show in your reports.*

## Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Settings**.

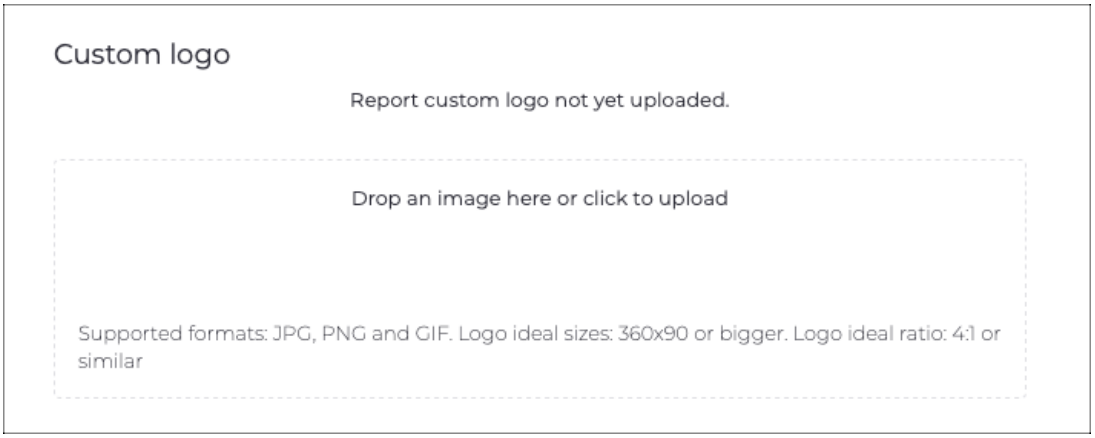
   **Result:** The Settings (on page 106) page opens.

3. Choose a method to upload a custom logo:

   **Choose from:**

   - Drag your image file into the **Drop an image here or click to upload** field
   - Click in the **Drop an image here or click to upload** field

4. If you chose the second method, select the correct file to upload.




**Note:**
Logos should be 360x90 pixels or bigger, with an ideal aspect ratio of 4:1, or similar.

**Note:**
Supported formats are:

- *graphics interchange format (GIF)*
- *joint photographic experts group (JPEG)*
- *portable network graphics (PNG)*

5. Wait for the file to upload.
6. Select **Save**.

### Results

Your custom logo will now be added to your reports.

# Configure SMTP settings

*If you want to send emails that contain scheduled reports, you need to configure the simple mail transfer protocol (SMTP) server settings.*

## Procedure

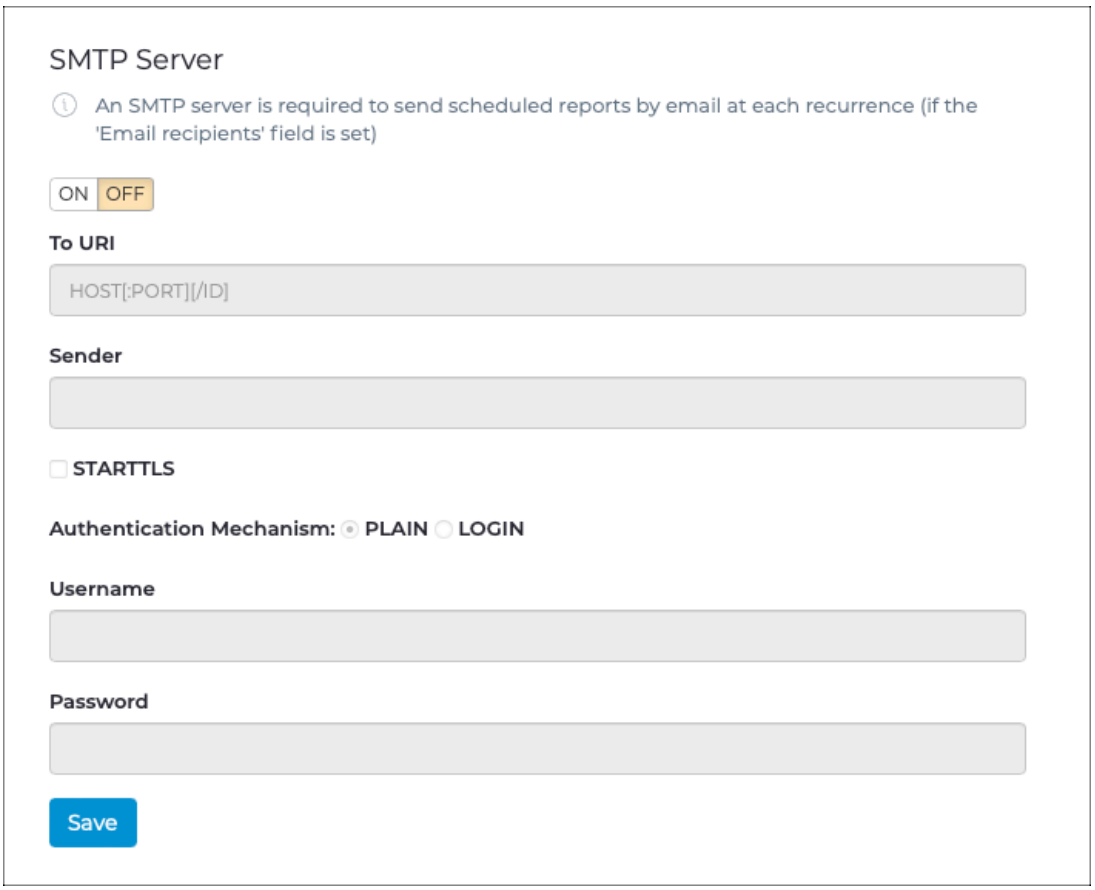
1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Settings**.

   **Result:** The Settings (on page 106) page opens.

3. In the **SMTP Server** section, set the toggle to **ON**.




4. In the **To URI** field, enter the host URI information. For example, `HOST[:PORT][/ID]`
5. In the **Sender** field, enter the sender identification information.

6. To use encryption, select the **STARTTLS** checkbox.

> **Note:**
> If you do not select this option, reports will be sent without encryption.

7. To start the authentication process, choose an Authentication Mechanism:

**Choose from:**

- **PLAIN**
- **LOGIN**

> **Note:**
> The default setting is **PLAIN**.

8. If you chose **LOGIN**, enter your credentials.
    a. In the **Username** field, enter your username.
    b. In the **Password** field, enter your password.
9. Select **Save**.

## Results

Emails for scheduled reports that have email recipients will now be sent at the next scheduled occurrence.

# Filter a report globally

*You can filter a report globally, which is the default filter. This lets you use a specific category to apply filters to the entire report.*

## Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

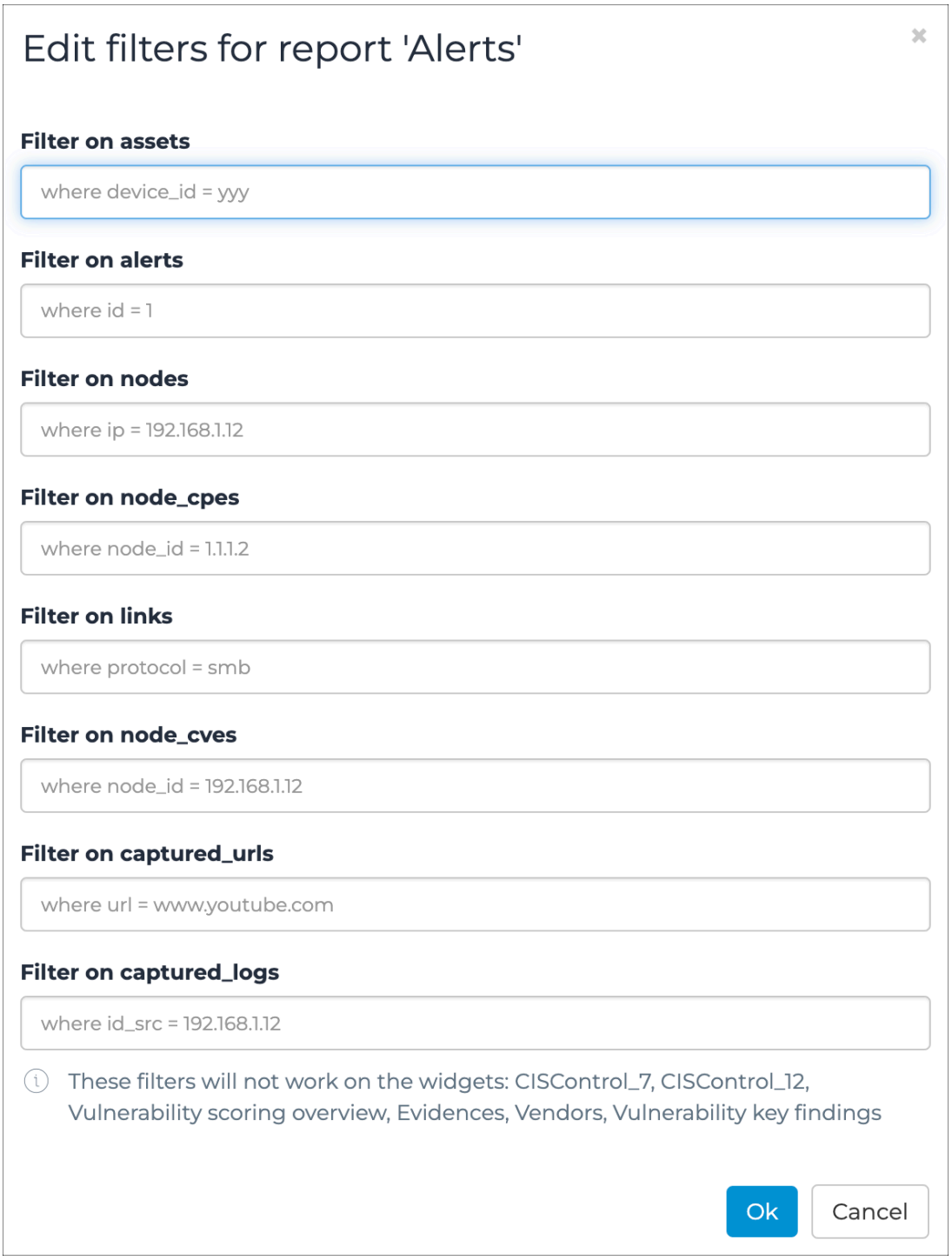
2. Select **Management**.

   **Result:** The Management (on page 101) page opens.

3. In the top section, select **Filters**.

   **Result:** A dialog shows.

4. Select the category on which to filter, then enter your filter query in the related field.



5. Select **Ok**.

> **Note:**
> At the bottom of the dialog is a list of widgets on which filters will not work.

### Results

The filter(s) has (have) been applied to the report.

# Add a widget to a report

*The **Management** page lets you add widgets to reports.*

## Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Management**.

   **Result:** The Management (on page 101) page opens.

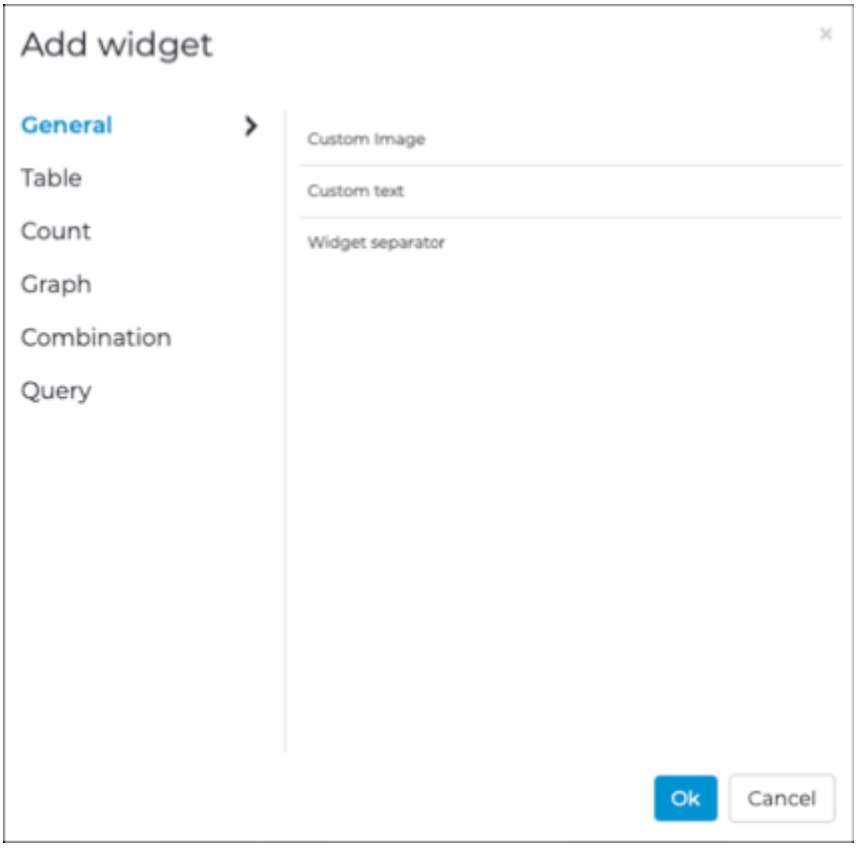
3. In the **Reports list** section on the left, select the applicable report.

   **Result:** The report opens.

4. On the right side of the report, choose the section that you want to add the widget to. Select **Add widget**.

   **Result:** A dialog shows.

5. In the left pane, choose the type of widget that you want to add.




6. From the list, select the widget that you want.

### Results

The widget has been added to the report.

# Filter a report with a widget

*You can use widgets to filter a report.*

## Procedure

1. In the top navigation bar, select icon > **Reports**.

   **Result:** The **Reports** page opens.

2. Select **Management**.

   **Result:** The Management (on page 101) page opens.

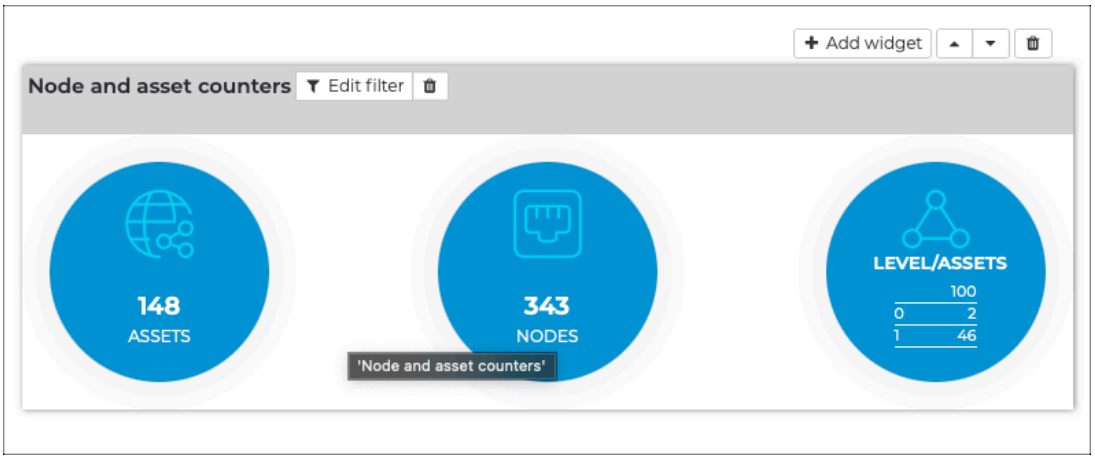
3. In the **Reports list** section on the left, select the applicable report.

   **Result:** The report opens.

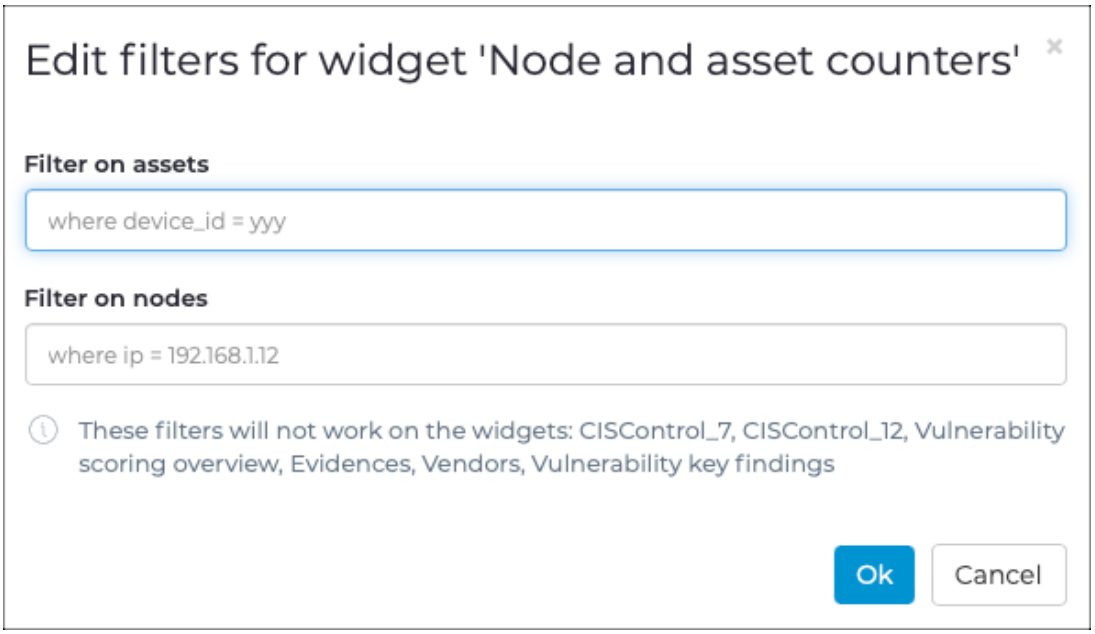
4. Find the widget that you would like to use and hover your mouse over it.

   **Result:** At the top of the widget, more buttons show.

5. Select **Edit filter**.



6. Select the category on which to filter, then enter your filter query in the related field.




7. Select **Ok**.

> **Note:**
> At the bottom of the dialog is a list of widgets on which filters will not work.

### Results

The filter(s) has (have) been applied to the report.

# Chapter 9. Assertions

*The **Assertions** page shows all the assertions and lets you configure them.*

A valid assertion is a normal query with a special command appended at the end. Assertions can be saved in a specific order and can be continuously executed in the system.

Queries are based on the *N2QL*, which you can use to ensure that certain conditions are met on the observed system. An assertion is typically either an empty value, or a specific value. When an unexpected value appears, or when the value is different than the expected, the system alerts the user.

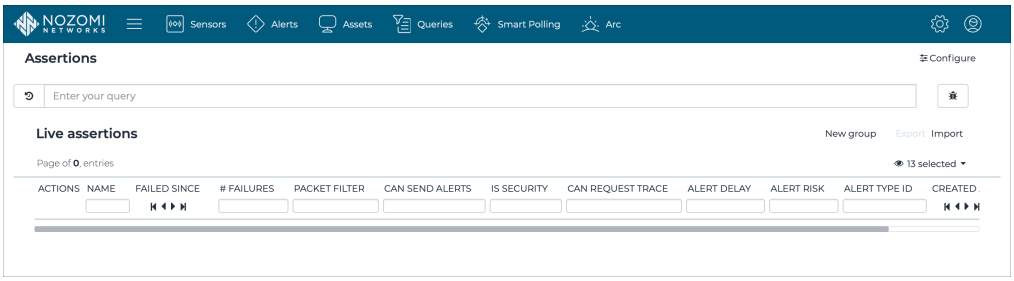



**Figure 43. Assertions page**

## Configure

This lets you configure the execution interval in seconds.

## History button

This button shows a history of the previous queries that have been entered in the query field.

## Query field

This field is where you enter your query.

## Debug button

Because assertions with logical operators and brackets can quickly become complex, the debug icon decomposes the query, and executes each part to show intermediate results.

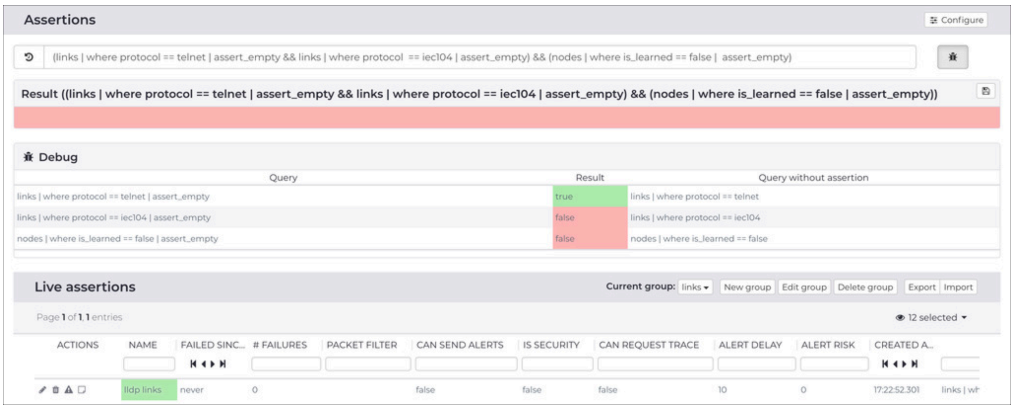



**Figure 44. Complex assertion being debugged**

### New group
This lets you create a group to combine assertions to make viewing and management easier.

### Export
This lets you export assertion groups in *JSON* format.

### Import
This lets you import assertion groups in *JSON* format.

# Assertion operators

**Table 5. Assertion operators**

| Operator | Description |
|---|---|
| `assert_all <field> <op> <value>` | The assertion is satisfied when each element in the query result set matches the given condition. |
| `assert_any <field> <op> <value>` | The assertion is satisfied when at least one element in the query result set matches the given condition. |
| `assert_empty` | The assertion is satisfied when the query returns an empty result set. |
| `assert_not_empty` | The assertion is satisfied when the query returns a non-empty result set. |

# Save an assertion

*You can save assertions to have them continuously executed in the system.*

## Procedure

1. In the top navigation bar, select icon > **Assertions**.

   **Result:** The **Assertions** page opens.

2. In the query field, enter a query.
3. Select **Enter**.
4. To save the assertion, select **Save**.

   **Result:** A dialog shows.

5. In the **Name** field, enter a name for the query.


6. In the **Description** field, enter a description.

7. To assign the assertion to a group, in the **Group** field, select one of the following
   a. From the dropdown menu, select an existing group.
   b. Select **Save**.
   c. To create a new group, select **New group**.
      **Result:** A dialog shows.
   d. In the **Group name** field, enter a group name for the assertion.
   e. Select **Save**.
8. Choose from one of these options:

   **Choose from:**
   - **Is security ?**
   - **Is operational ?**
9. In the **Assertion Check Interval** field, choose the interval in seconds at which the assertion will be rechecked.

   **Note:**
   You can select an interval between 10 seconds and 1 day.
10. **Optional:** If you want the assertion to trigger an alert, select the **Can send alerts** checkbox.
11. From the **Choose the asserted table's specific fields to include in the Description** dropdown, select the fields to include in the assertion description.
12. In the **Query** field, enter the assertion query.
13. Select **Save**.

    **Result:** The saved assertion will be listed at the bottom of the page with a green or red color to indicate the result.

# Edit an assertion

*This lets you edit the details for an existing assertion.*

## Procedure

1. In the top navigation bar, select icon > **Assertions**.

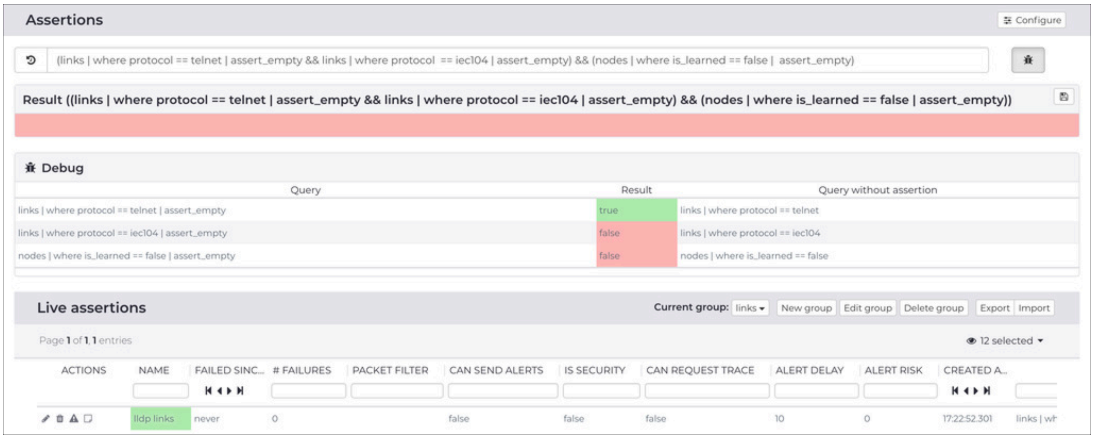
   **Result:** The **Assertions** page opens.

2. In the query field, enter a query.
3. To execute the query, elect **Enter**.

> **Note:**
> You can use the logical operators && (and) and || (or) to combine multiple assertions. Round brackets ( ) change the logical grouping as in a mathematical expression.

4. **Optional:** To the right of the query field, select the debug icon.

> **Note:**
> Because assertions with logical operators and brackets can quickly become complex, the debug icon decomposes the query, and executes each part to show intermediate results.




**Result:** The debug section shows.

# Configure an assertion

*This lets you configure the execution interval of the assertions, in seconds.*

## Procedure

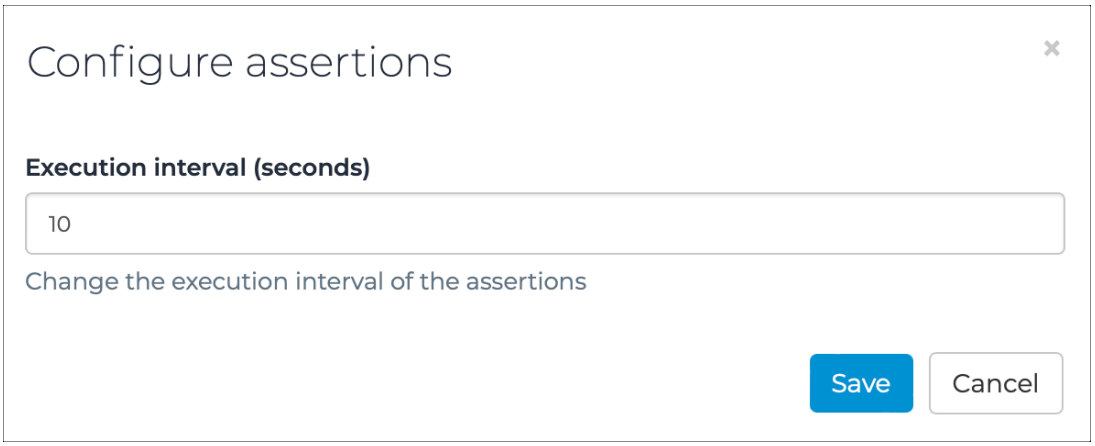
1. In the top navigation bar, select icon > **Assertions**.

   **Result:** The **Assertions** page opens.

2. In the top right, select **Configure**.

   **Result:** A dialog opens.

3. In the **Execution interval (seconds)** field, enter an interval.




4. Select **Save**.

# Configure an assertion on links

*This lets you configure the scope of the assertion for the related links.*

## Procedure

1. In the top navigation bar, select icon > **Network**.

   **Result:** The **Network** page opens.

2. Select **Links**.

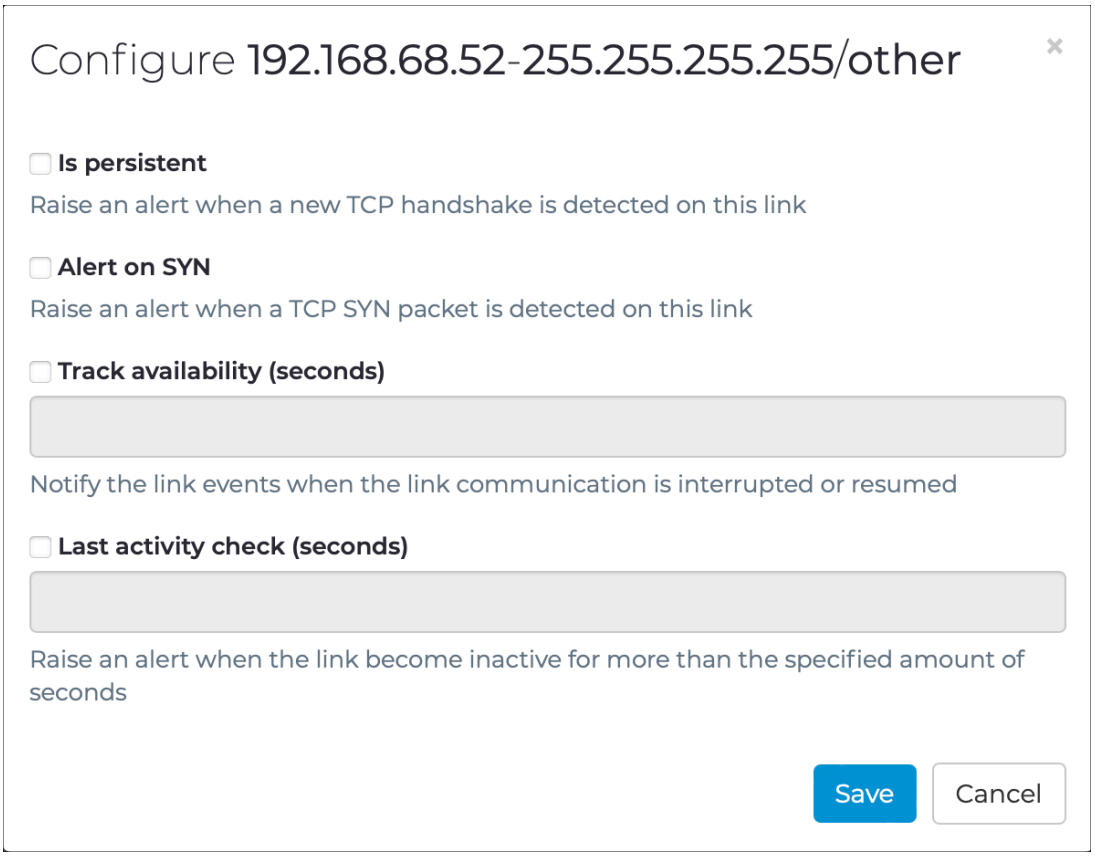
   **Result:** The **Links** page opens.

3. To the left of the applicable link, select the configure icon.

   **Result:** A dialog opens.

4. **Optional:**

   Select **Is persistent**.




**Note:**
When selected, this check raises a new alert whenever a *transmission control protocol (TCP)* handshake is successfully completed on the link.

5. **Optional:** Select **Alert on SYN**.

> **Note:**
> When selected, this check raises a new alert whenever a client sends a TCP SYN on the link.

6. **Optional:** Select **Track availability (seconds)**.

> **Note:**
> When selected, a link is considered non-functioning if it is unresponsive for the specified time.

7. **Optional:** Select **Last activity check (seconds)**.

> **Note:**
> When selected, this check raises an alert when the link is not receiving data for more than the specified time.

8. Select **Save**.

### Results

The assertion has been configured.

# Configure an assertion on variables

*This lets you configure the scope of the assertion for the related variables.*

## Procedure

1. In the top navigation bar, select icon > **Process**.

   **Result:** The **Process** page opens.

2. Select **List**.

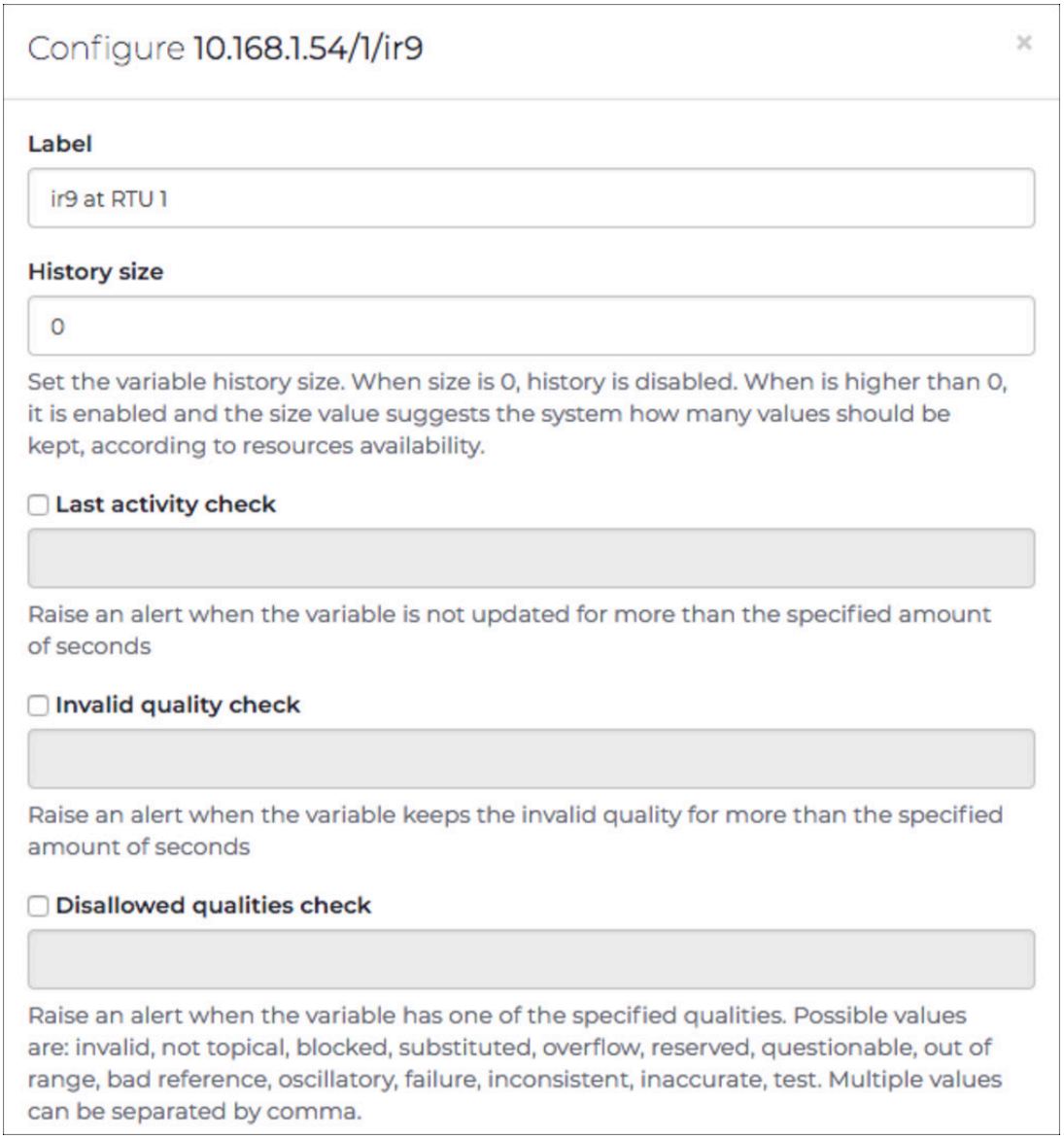
   **Result:** The **List** page opens.

3. To the left of the applicable link, select the configure icon.

   **Result:** A dialog opens.

4. **Optional:**

   In the **Label** field, enter a label for the assertion.




5. **Optional:** In the **History size** field, enter a value.

> **Note:**
> This sets the *variable* history size. When the size is 0, history is disabled. When it is higher than 0, it is enabled, and the size value suggests how many values that the system should keep, depending on the available resources.

6. **Optional:** In the **Last activity check** field, enter a value.

   **Note:**
   When selected, this check raises an alert when the *Variable* is either not measured or is changed for more than the specified number of seconds.

7. **Optional:** In the **Invalid quality check** field, enter a value.

   **Note:**
   When selected, this check raises an alert when the *Variable* maintains an invalid quality for more than the specified amount of seconds.

8. **Optional:** In the **Disallowed quality check** field, enter a value.

   **Note:**
   When selected, this check raises an alert when the *Variable* gains one of the specified qualities.

9. Select **Save**.

## Results

The assertion has been configured.

# Chapter 10. Vulnerabilities

*The Nozomi Networks software continuously discovers vulnerabilities in monitored assets.*

The Nozomi Networks software continuously discover vulnerabilities. To do this, it matches the *Common Platform Enumeration (CPE)* of a device with the National Vulnerability Database, and other data sources.

The Vulnerabilities page has these tabs:

- Assets (on page 146)
- List (on page 147)
- Stats (on page 148)

# Assets

*The **Assets** page shows a list of assets with known vulnerabilities, along with a summary of the severity of the vulnerability.*

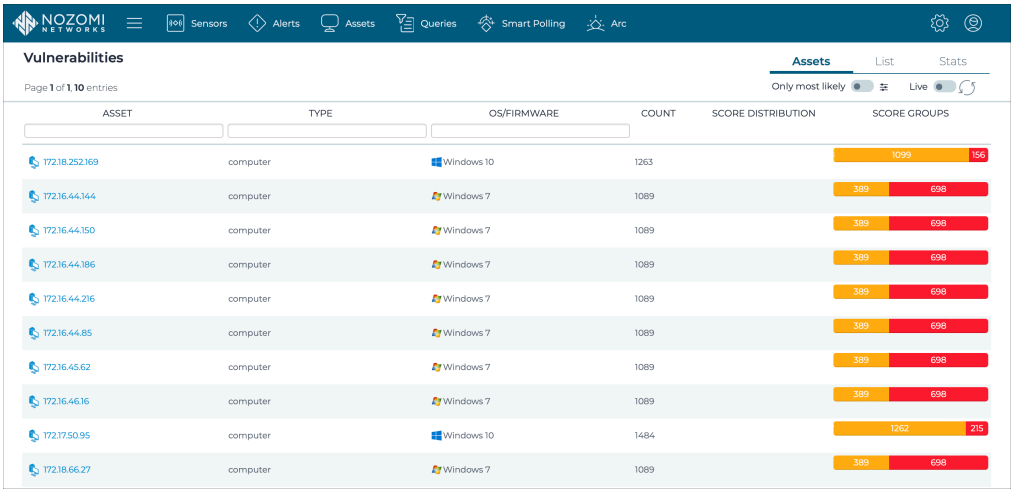



**Figure 45. Assets page**

## Only most likely

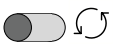
This toggle lets you filter the view to show only the assets the match the criteria that you have set for likelihood threshold.

## Likelihood threshold settings

**Likelihood threshold** is a value between 0.1 and 1.0 where 1.0 represents the maximum likelihood of the *Common Vulnerabilities and Exposures (CVE)* to be present. Likelihood is the confidence of the software's correct assignment of a *CPE* to the hardware of the monitored asset. The higher the likelihood, the higher the software's confidence that the vulnerabilities assigned to an asset are in fact are relevant to that asset. **Likelihood threshold** is the minimum likelihood a vulnerability needs in order for it to be shown in this page when the **Only most likely** toggle is set to on. As a guideline, we suggest that you use:

- 0.8 for a high level of confidence
- 0.5 for a medium level of confidence
- 0.3 for a low level of confidence

## Live / refresh

The **Live**  icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

# List

*The **List** page shows a comprehensive list of vulnerabilities in the environment. This lets you perform global, in-depth analysis.*

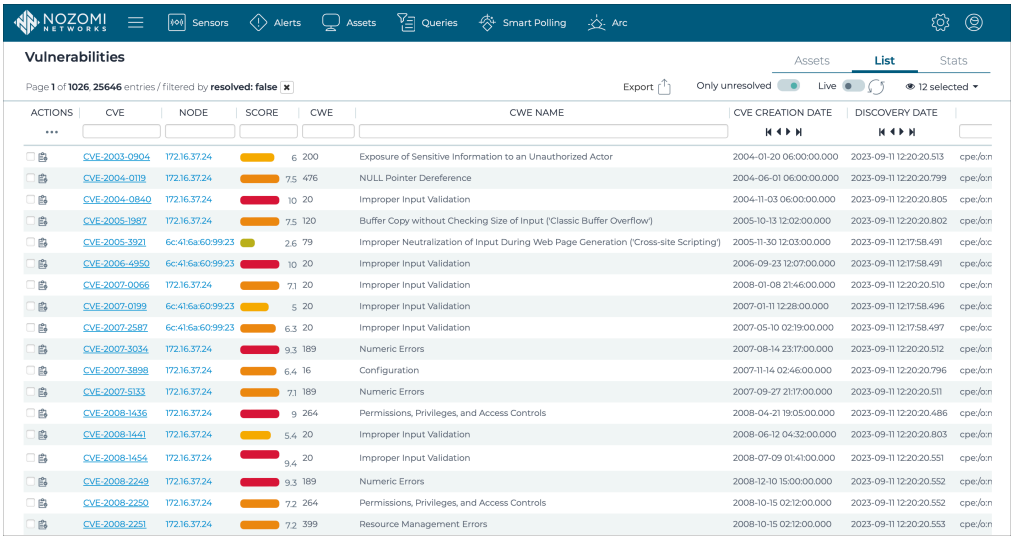



**Figure 46. List page**

## Export

The **Export** icon lets you export the current list in either *CSV* or Microsoft Excel format.

## Only resolved

This lets you show only `Unresolved` vulnerabilities. Vulnerability status options are:

- `Unresolved`
- `Mitigated`
- `Accepted`

`Mitigated` and `Accepted` lead to a resolution status that equals true.

## Live / refresh

The **Live** icon lets you change live view on, or off. When live mode is on, the page will refresh approximately every five seconds.

## Column selection

The columns selection icon lets you choose which columns to show or hide.

# Stats

*The **Stats** page shows high level information in a graphical format that shows the top common platform enumerations (CPEs), common vulnerabilities and exposures (CVEs), and common weakness enumerations (CWEs).*

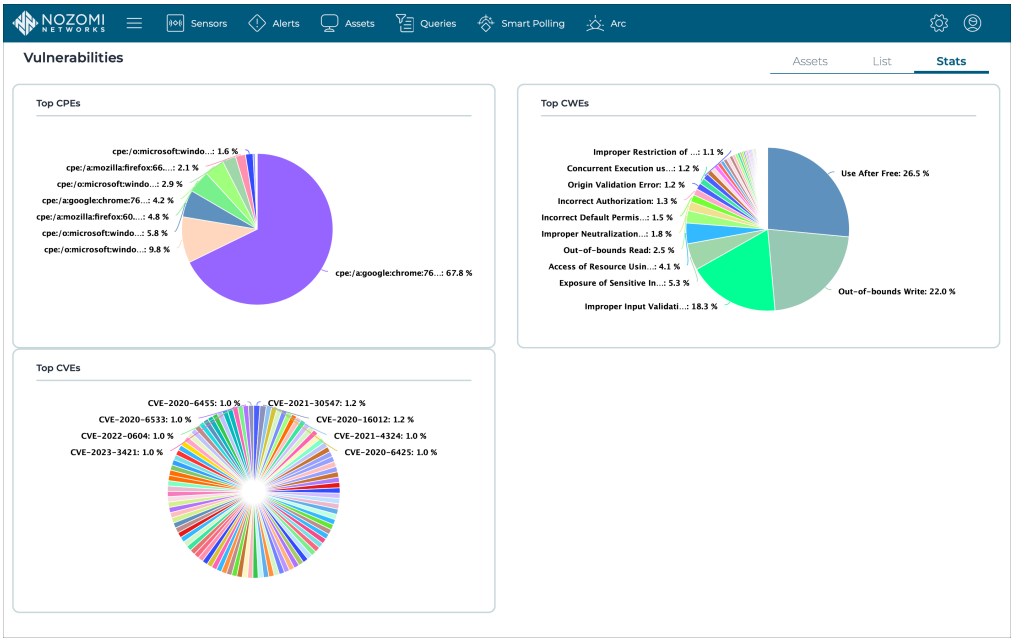



**Figure 47. Stats page**

## Top CPEs

This section shows the:

- Title of vulnerability
- Percentage of the total vulnerabilities
- Actual count of that vulnerability

## Top CWEs

This section shows the:

- Title of vulnerability
- Percentage of the total vulnerabilities
- Actual count of that vulnerability

## Top CVEs

This section shows the:

- Title of vulnerability
- Date of vulnerability type
- Percentage of the total vulnerabilities
- Actual count of that vulnerability

# Details page

*The details page shows you all the details for the related vulnerability.*

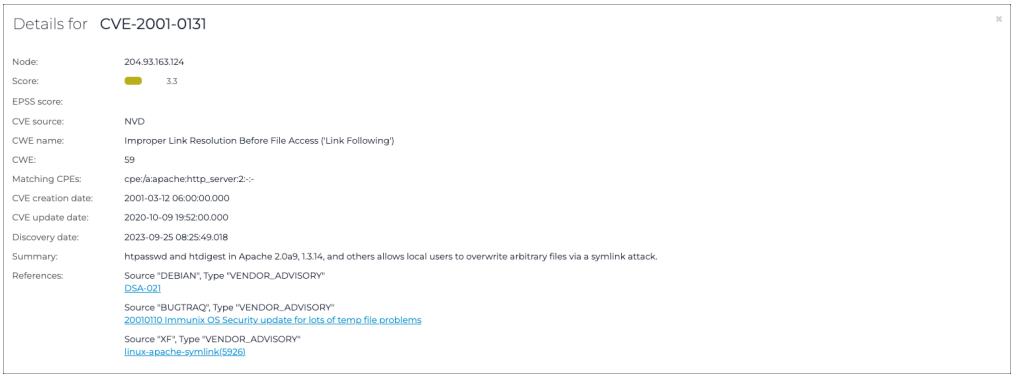



**Figure 48. Details page**

# Chapter 11. Administration

# Administration page

*The administration page lets a user with administrator privileges configure settings and do other tasks.*

For more details, see the **CMC - Administrator Guide**.

# Chapter 12. Personal settings

*The personal settings page lets you do actions that are specific to your profile.*

## General

You can select the icon to access the personal settings menu.

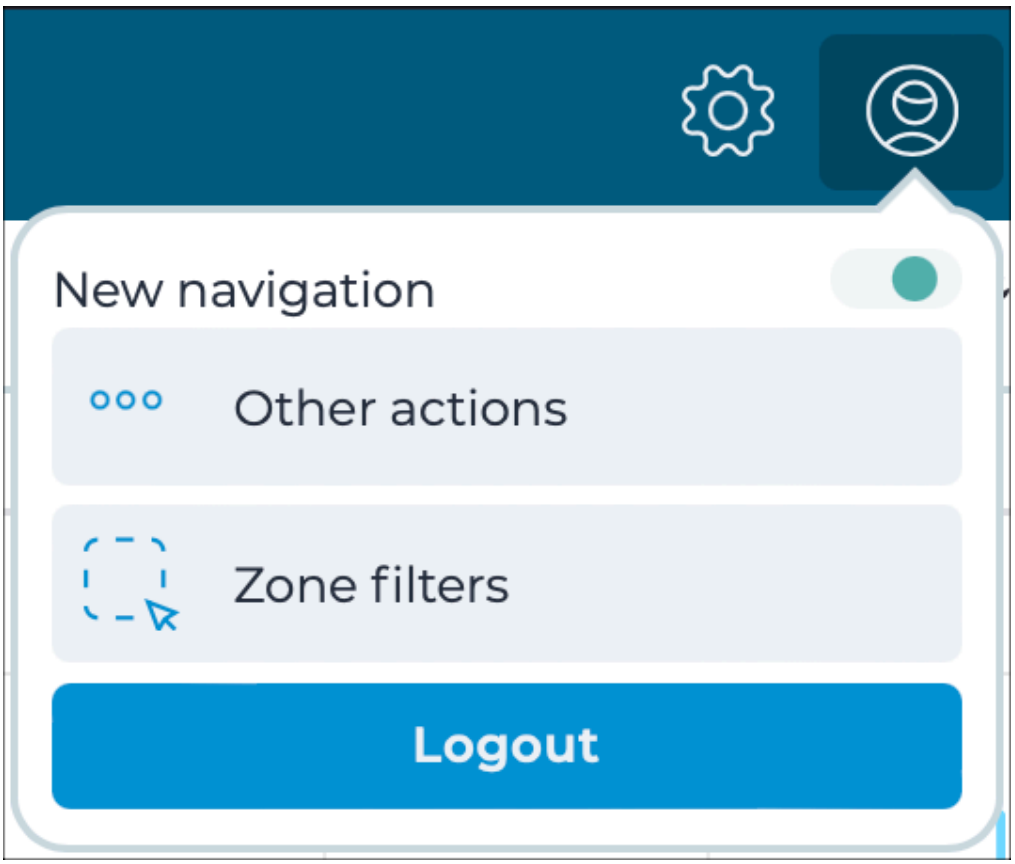



**Figure 49. Personal settings menu**

The personal settings page lets you Clear your personal settings (on page 158)

## Logout

A button in the personal settings menu lets you log out of the software.

# Clear your personal settings

*The profile settings menu lets you clear your personal settings that are stored in the local browser local.*

## Procedure

1. In the top navigation bar, select

   **Result:** A menu shows.

2. Select **Other actions**.
3. Select **Clear personal settings**.

   **Result:** A dialog shows.

4. Select **OK**.


## Results

Your personal settings have been cleared from the local browser.

# Chapter 13. Troubleshooting

# Troubleshooting

*A list of the most useful troubleshooting tips for the Central Management Console (CMC).*

If the sensor is not appearing at all in the *CMC*:

- Make sure that firewall(s) between the sensor and the *CMC* permit traffic on *TCP*/443 port (*hypertext transfer protocol secure (HTTPS)*), with the sensor as **Source** and the *CMC* as the **Target**
- Make sure that the tokens are correctly configured both in the sensor and the *CMC*
- Make sure that in the `/data/log/n2os/n2osjobs.log` file for connection errors

The **Sensor ID** is stored in the `/data/cfg/.appliance-uuid` file. Do not edit this file after the sensor is connected to the Vantage, or the *CMC*, since it is the unique identifier of the sensor inside Vantage and the *CMC*. In case a forceful change of the **Appliance ID** is needed, you will need to remove the old data from Vantage or the *CMC* by removing the old **Appliance ID** entry.

If an issue occurs during the setup of a sensor, you should:

- Delete the sensor, or
- Clear the sensor's data from Vantage or the *CMC*

# Glossary

### *Amazon Machine Image*

An AMI is a type of virtual appliance that is used to create a virtual machine for the Amazon Elastic Compute Cloud (EC2), and is the basic unit of deployment for services that use EC2 for delivery.

### *Amazon Web Services*

AWS is a subsidiary of the Amazon company that provides on-demand cloud computing platforms governments, businesses, and individuals on a pay-as-you-go basis.

### *Application Programming Interface*

An API is a software interface that lets two or more computer programs communicate with each other.

### *Assertion Consumer Service*

An ACS is a version of the SAML standard that is used to exchange authentication and authorization identities between security domains.

### *Asset Intelligence™*

Asset Intelligence is a continuously expanding database of modeling asset behavior used by N2OS to enrich asset information, and improve overall visibility, asset management, and security, independent of monitored network data.

### *Berkeley Packet Filter*

The BPF is a technology that is used in some computer operating systems for programs that need to analyze network traffic. A BPF provides a raw interface to data link layers, permitting raw link-layer packets to be sent and received.

### *Central Management Console*

The Central Management Console (CMC) is a Nozomi Networks product that has been designed to support complex deployments that cannot be addressed with a single sensor. A central design principle behind the CMC is the unified experience, that lets you access information in a similar way as on the sensor.

### *Central Processing Unit*

The main, or central, processor that executes instructions in a computer program.

### *Certificate Authority*

A certificate, or certification authority (CA) is an organization that stores, signs, and issues digital certificates. In cryptography, a digital certificate certifies the ownership of a public key by the named subject of the certificate.

### *Classless Inter-Domain Routing*

CIDR is a method for IP routing and for allocating IP addresses.

### *Command-line interface*

A command-line processor uses a command-line interface (CLI) as text input commands. It lets you invoke executables and provide information for the actions that you want them to do. It also lets you set parameters for the environment.

### *Comma-separated Value*

A CSV file is a text file that uses a comma to separate values.

### *Common Event Format*

CEF is a text-based log file format that is used for event logging and information sharing between different security devices and software applications.

### *Common Platform Enumeration*

CPE is a structured naming scheme for information technology (IT) systems, software, and packages. CPE is based on the generic syntax for Uniform Resource Identifiers (URI) and includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name.

### *Common Vulnerabilities and Exposures*

CVEs give a reference method information-security vulnerabilities and exposures that are known to the public. The United States' National Cybersecurity FFRDC maintains the system.

### *Common Weakness Enumeration*

CWE is a category system for software and hardware weaknesses and vulnerabilities. It is a community project with the aim to understand flaws in software and hardware and create automated tools that can be used to identify, fix, and prevent those flaws.

### *Configuration file*

A CFG file is a configuration, or config, file. They are files that are used to configure the parameters and initial settings for a computer program.

### *Domain Name Server*

The DNS is a distributed naming system for computers, services, and other resources on the Internet, or other types of Internet Protocol (IP) networks.

### ESXi

VMware ESXi (formerly ESX) is an enterprise-class, type-1 hypervisor developed by VMware for deploying and serving virtual computers. As a type-1 hypervisor, ESXi is not a software application that is installed on an operating system (OS). Instead, it includes and integrates vital OS components, such as a kernel.

### Extensible Markup Language

XML is a markup language and file format for the storage and transmission of data. It defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

### Federal Information Processing Standards

FIPS are publicly announced standards developed by the National Institute of Standards and Technology for use in computer systems by non-military American government agencies and government contractors.

### File Allocation Table

FAT is a file system architecture used in computers for managing disk space. It maintains a table to track the allocation of files on a disk, supporting efficient data storage, access, and management

### File Transfer Protocol

FTP is a standard communication protocol that is used for the transfer of computer files from a server to a client on a computer network. FTP is built on a client–server model architecture that uses separate control and data connections between the client and the server.

### Fully qualified domain name

An FQDN is a complete and specific domain name that specifies the exact location in the hierarchy of the Domain Name System (DNS). It includes all higher-level domains, typically consisting of a host name and domain name, and ends in a top-level domain.

### Gigabit per second

Gigabit per second (Gb/s) is a unit of data transfer rate equal to: 1,000 Megabits per second.

### Gigabyte

The gigabyte is a multiple of the unit byte for digital information. One gigabyte is one billion bytes.

### Graphical User Interface

A GUI is an interface that lets humans interact with electronic devices through graphical icons.

### Graphics Interchange Format

GIF is a bitmap image format that is widely used on the internet.

### High Availability

High Availability is a mode that permits the CMC to replicate its own data on another CMC.

### Host-based intrusion-detection system

HIDS is an internal Nozomi Networks solution that uses sensors to detect changes to the basic firmware image, and record the change.

### Hypertext Markup Language

Hypertext Markup Language (HTML) is the standard markup language used to structure and display content on the web. It defines the structure of web pages using elements represented by tags.

### Hypertext Transfer Protocol

HTTP is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen in a web browser.

### Hypertext Transfer Protocol Secure

HTTPS is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet. In HTTPS, the communication protocol is encrypted using Transport Layer Security (TLS) or, formerly, Secure Sockets Layer (SSL). The protocol is therefore also referred to as HTTP over TLS, or HTTP over SSL.

### Identifier

A label that identifies the related item.

### Identity Provider

An IdP is a system entity that creates, maintains, and manages identity information. It also provides authentication services to applications within a federation, or a distributed network.

### *Internet Control Message Protocol*

ICMP is a supporting protocol in the internet protocol suite. Network devices use it to send error messages and operational information to indicate success or failure when communicating with another IP address. ICMP differs from transport protocols such as TCP and UDP in that it is not typically used to exchange data between systems.

### *Internet of Things*

The IoT describes devices that connect and exchange information through the internet or other communication devices.

### *Internet Protocol*

An Internet Protocol address, or IP address, identifies a node in a computer network that uses the Internet Protocol to communicate. The IP label is numerical.

### *Intrusion Detection System*

An intrusion detection system (IDS), which can also be known as an intrusion prevention system (IPS) is a software application, or a device, that monitors a computer network, or system, for malicious activity or policy violations. Such intrusion activities, or violations, are typically reported either to a system administrator, or collected centrally by a security information and event management (SIEM) system.

### *JavaScript Object Notation*

JSON is an open standard file format for data interchange. It uses human-readable text to store and transmit data objects, which consist of attribute–value pairs and arrays.

### *Joint Photographic Experts Group*

JPEG, or JPG, is a method of lossy compression that is used for digital images. The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality.

### *Lightweight Directory Access Protocol*

LDAP is an open, vendor-neutral, industry standard application protocol that lets you access and maintain distributed directory information services over an internet protocol (IP) network.

### *Lightweight Directory Access Protocol Secure*

LDAP over SSL or Secure LDAP is the secure version of LDAP.

### *Managed Security Service Provider*

In computing, MSSP are network security services that have been outsourced to a service provider. A company that provides this type of service is known as an MSSP.

### *Management Information Base*

A MIB is a collection of information organized hierarchically. It is used by network management systems to monitor and control network devices through protocols like SNMP.

### *Media Access Control*

A MAC address is a unique identifier for a network interface controller (NIC). It is used as a network address in network segment communications. A common use is in most IEEE 802 networking technologies, such as Bluetooth, Ethernet, and Wi-Fi. MAC addresses are most commonly assigned by device manufacturers and are also referred to as a hardware address, or physical address. A MAC address normally includes a manufacturer's organizationally unique identifier (OUI). It can be stored in hardware, such as the card's read-only memory, or by a firmware mechanism.

### *Megabyte*

The megabyte is a multiple of the unit byte for digital information. One megabyte is one million bytes.

### *Message Queuing Telemetry Transport*

Message Queuing Telemetry Transport is a lightweight, publish-subscribe network protocol designed for constrained devices and low-bandwidth, high-latency, or unreliable networks. It is commonly used in Internet of Things (IoT) applications.

### *Network Address Translation*

NAT is a method of mapping an internet protocol (IP) address space into another one. This is done by modifying network address information in the IP header of packets while in transit across a traffic routing device.

### *Network-Attached Storage*

NAS is a dedicated file storage system that provides local-area network (LAN) users with centralized, shared access to data through standard protocols such as NFS or SMB.

### Network Interface Controller

A network interface controller (NIC), sometimes known as a network interface card, is a computer hardware component that lets a computer connect to a computer network.

### Network Time Protocol

The NTP is a networking protocol to synchronize clocks between computer systems over variable-latency, packet-switched data networks.

### Nozomi Networks Operating System

N2OS is the operating system that the core suite of Nozomi Networks products runs on.

### Nozomi Networks Query Language (N2QL)

N2QL is the language used in queries in Nozomi Networks software.

### Open Virtual Appliance

An OVA file is an open virtualization format (OVF) directory that is saved as an archive using the .tar archiving format. It contains files for distribution of software that runs on a virtual machine. An OVA package contains a .ovf descriptor file, certificate files, an optional .mf file along with other related files.

### Operating System

An operating system is computer system software that is used to manage computer hardware, software resources, and provide common services for computer programs.

### Operational Technology

OT is the software and hardware that controls and/or monitors industrial assets, devices and processes.

### Packet Capture

A pcap is an application programming interface (API) that captures live network packet data from the OSI model ( layers 2-7).

### Packet Capture Next Generation

A pcapNg is the latest version of a pcap file, an application programming interface (API) that captures live network packet data from the OSI model ( layers 2-7).

### Portable Document Format

PDF is a Adobe file format that is used to present documents. It is independent of operating systems (OS), application software, hardware.

### Portable Network Graphics

PNG is a raster graphics file format that supports lossless data compression.PNG was developed as an improved, non-patented replacement for graphics interchange format (GIF).

### Privacy-Enhanced Mail

PEM is a standard file format that is used to store and send cryptographic keys, certificates, and other data. It is based on a set of 1993 IETF standards.

### Programmable Logic Controller

A PLC is a ruggedized, industrial computer used in industrial and manufacturing processes.

### Protected Extensible Authentication Protocol

PEAP is a protocol that encloses the Extensible Authentication Protocol (EAP) within an encrypted and authenticated Transport Layer Security (TLS) tunnel.

### Random-access Memory

Computer memory that can be read and changed in any order. It is typically used to store machine code or working data.

### Representational State Transfer

Representational State Transfer (REST) is an architectural style for designing networked applications. It uses stateless, client-server communication via standard HTTP methods (GET, POST, PUT, DELETE) to access and manipulate web resources represented in formats like JSON or XML.

### Seamless Message Protocol

Seamless Message Protocol is a communication protocol used primarily in industrial automation for device-level communication. It enables data exchange between controllers and devices over Ethernet networks.

### Secure Copy Protocol

SCP is a protocol for the secure transfer of computer files between a local host and a remote host, or between two remote hosts. It is based on the secure shell (SSH) protocol.

### Secure Shell

A cryptographic network protocol that let you operate network services securely over an unsecured network. It is commonly used for command-line execution and remote login applications.

### Secure Sockets Layer

A secure sockets layer ensures secure communication between a client computer and a server.

### *Security Assertion Markup Language*

SAML is an open standard, XML-based markup language for security assertions. It allows for the exchange of authentication and authorization data different parties such as a service provider and an identity provider.

### *Security Information and Event Management*

SIEM is a field within the computer security industry, where software products and services combine security event management (SEM) and security information management (SIM). SIEMs provide real-time analysis of security alerts.

### *Server Message Block*

Is a communication protocol which provides shared access to files and printers across nodes on a network of systems. It also provides an authenticated interprocess communication (IPC) mechanism.

### *Simple File Transfer Protocol*

SFTP was proposed as an unsecured file transfer protocol with a level of complexity intermediate between TFTP and FTP. It was never widely accepted on the internet.

### *Simple Mail Transfer Protocol*

SMTP is an internet standard communication protocol that is used for the transmission of email. Mail servers and other message transfer agents use SMTP to send and receive mail messages.

### *Simple Network Management Protocol*

SNMP is an Internet Standard protocol for the collection and organization of information about managed devices on IP networks. It also lets you modify that information to change device behavior. Typical devices that support SNMP are: printers, workstations, cable modems, switches, routers, and servers.

### *Simple Text Oriented Messaging Protocol*

STOMP is a simple text-based protocol, for working with message-oriented middleware (MOM). It provides an interoperable wire format that allows STOMP clients to talk with any message broker supporting the protocol.

### *Spanning Tree Protocol*

Spanning Tree Protocol is a network protocol that prevents loops in Ethernet networks by creating a spanning tree topology. It selectively blocks redundant paths and ensures a loop-free logical topology.

### *Structured Threat Information Expression*

STIX™ is a language and serialization format for the exchange of cyber threat intelligence (CTI). STIX is free and open source.

### *Supervisory control and data acquisition*

SCADA is a control system architecture which has computers, networked data communications and graphical user interfaces for high-level supervision of processes and machines. It also covers sensors and other devices, such as programmable logic controllers (PLC), which interface with process plant or machinery.

### *Text-based User Interface*

In computing, a text-based (or terminal) user interfaces (TUI) is a retronym that describes a type of user interface (UI). These were common as an early method of human–computer interaction, before the more modern graphical user interfaces (GUIs) were introduced. Similar to GUIs, they might use the entire screen area and accept mouse and other inputs.

### *Threat Intelligence™*

Nozomi Networks **Threat Intelligence™** feature monitors ongoing OT and IoT threat and vulnerability intelligence to improve malware anomaly detection. This includes managing packet rules, YARA rules, STIX indicators, Sigma rules, and vulnerabilities. **Threat Intelligence™** allows new content to be added, edited, and deleted, and existing content to be enabled or disabled.

### *Transmission Control Protocol*

One of the main protocols of the Internet protocol suite.

### *Transport Layer Security*

TLS is a cryptographic protocol that provides communications security over a computer network. The protocol is widely used in applications such as: HTTPS, voice over IP, instant messaging, and email.

### *Uniform Resource Identifier*

A URI is a unique string of characters used to identify a logical or physical resource on the internet or local network.

### *Uniform Resource Locator*

An URL is a reference to a resource on the web that gives its location on a computer network and a mechanism to retrieving it.

### *Uninterruptible Power Supply*

A UPS is an electric power system that provides continuous power. When the main input power source fails, an automated backup system continues to supply power.

### *Universally unique identifier*

A UUID is a 128-bit label that is used for information in computer systems. When a UUID is generated with standard methods, they are, for all practical purposes, unique. Their uniqueness is not dependent on an authority, or a centralized registry. While it is not impossible for the UUID to be duplicated, the possibility is generally considered to be so small, as to be negligible. The term globally unique identifier (GUID) is also used in some, mostly Microsoft, systems.

### *Universal Serial Bus*

Universal Serial Bus (USB) is a standard that sets specifications for protocols, connectors, and cables for communication and connection between computers and peripheral devices.

### *User Interface*

An interface that lets humans interact with machines.

### *Variable*

In the context of control systems, a variable can refer to process values that change over time. These can be temperature, speed, pressure etc.

### *Virtual DOM*

A virtual DOM, or vdom, is a lightweight JavaScript representation of the Document Object Model (DOM). It is used in declarative web frameworks such as Elm, React, and Vue.js. It enables the updating of the virtual DOM is comparatively faster than updating the actual DOM.

### *Virtual Hard Disk*

VHD is a file format that represents a virtual hard disk drive (HDD). They can contain what is found on a physical HDD, such as disk partitions and a file system, which in turn can contain files and folders. They are normally used as the hard disk of a virtual machine (VM). They are the native file format for Microsoft's hypervisor (virtual machine system), Hyper-V.

### *Virtual Local Area Network*

A VLAN is a broadcast domain that is isolated and partitioned in a computer network at the data link layer (OSI layer 2).

### *Virtual Machine*

A VM is the emulation or virtualization of a computer system. VMs are based on computer architectures and provide the functionality of a physical computer.

### *ZIP*

An archive file format that supports lossless data compression. The format can use a number of different compression algorithms, but DEFLATE is the most common one. A ZIP file can contain one or more compressed files or directories.