ivanti

Sentry 9.14.0 - 9.20.0 Guide for Ivanti Neurons for MDM

March 2024

Contents

Revision history

TABLE 1. REVISION HISTORY

Date	Revision
March 28, 2024	Updated for 9.14.0 - 9.20.0 Ivanti Standalone Sentry.
October 23, 2023	Updated for 9.14.0 - 9.19.0 Ivanti Standalone Sentry.
July 18, 2023	Updated for 9.14.0 - 9.18.0 Ivanti Standalone Sentry.
May 29, 2023	Updated "OAuth for Ivanti Standalone Sentry on Ivanti Neurons for MDM" on page 43.
January 10, 2023	Updated Multi-factor authentication configuration for Ivanti EPMM for Android and iOS Email+ configuration.
December 15, 2022	Updated for 9.14.0 - 9.17.0 Sentry.
September 26, 2022	Updated "Checking TLS compliance" on page 202 with disabled TLS versions.
August 08, 2022	Updated for 9.16.0 Sentry.
April 05, 2022	Updated for 9.14.0 - 9.15.0 Sentry.

About Sentry

The following provide information about Sentry:

Ivanti Standalone Sentry overview

Ivanti Standalone Sentry is a part of deployment that serves as an intelligent gatekeeper to your company's ActiveSync server, such as a Microsoft Exchange Server, or with a backend resource such as a Sharepoint server, or it can be configured as a Kerberos Key Distribution Center Proxy (KKDCP) server. Sentry gets configuration and device information from a unified endpoint management (UEM) platform - Ivanti EPMM or Ivanti Neurons for MDM.

ℹ️   Access to enterprise content in business cloud services such as SalesForce, Box, G Suite, Dropbox, and Office 365 can be secured using Access. Access is a cloud service. An Access deployment requires a unified endpoint management (UEM) platform, Ivanti Standalone Sentry, and Ivanti Tunnel. For information about Access and how to set up the service, see the *Access Guide*.

The following provide additional information about Sentry:
- "Sentry flavors" below

    - "Sentry and UEM platform support" on the next page

    - "ActiveSync with Ivanti Standalone Sentry" on page 9

    - "AppTunnel with Ivanti Standalone Sentry" on page 10

    - "Network traffic support with AppTunnel" on page 10

Sentry flavors

Sentry is available in two flavors: Ivanti Standalone Sentry or Integrated Sentry.

Ivanti Standalone Sentry is a separate appliance that acts as a gateway between devices and your ActiveSync-enabled email servers or backend resource. Ivanti Standalone Sentry can be configured for ActiveSync or AppTunnel, or as Kerberos Key Distribution Center Proxy (KKDCP) server. Ivanti Standalone Sentry can be installed on premise on a Appliance or a virtual appliance. Or, Ivanti Standalone Sentry can be installed in the cloud on AWS and Microsoft Azure.

Integrated Sentry is a Windows service that interacts with the Microsoft Exchange Server.

Ivanti Standalone Sentry gets input from the unified endpoint management (UEM) platform, Ivanti EPMM or Ivanti Neurons for MDM, to do the following:

- Integrated Sentry or the Ivanti Standalone Sentry configured for ActiveSync protects the ActiveSync server from wrongful access from devices.

- Ivanti Standalone Sentry configured for AppTunnel provides authenticated apps secure access to the backend resource.

Sentry and UEM platform support

UEM support varies depending on the type of Sentry. The following tables provides the UEM supported by Sentry.

TABLE 2. SENTRY AND UEM PLATFORM SUPPORT

Sentry	UEM platform
Integrated Sentry	Ivanti EPMM
Ivanti Standalone Sentry (on-premise, AWS, and Microsoft Azure)	Ivanti EPMM, Ivanti Neurons for MDM

Ivanti Neurons for MDM license requirement

Some Ivanti Standalone Sentry features are available based on the license level for Ivanti Neurons for MDM. The following table describes the license required for these features:

TABLE 3. IVANTI NEURONS FOR MDM LICENSE REQUIREMENT

License requirement	Feature
Gold	Advanced traffic control
	Outgoing ciphers
	CIFS service for Docs@Work
	Sharepoint service for Docs@Work
	Help@Work service
	Web@Work service
	Custom HTTP service
	Custom TCP service
Platinum	Tunnel service
	AppTunnel

ActiveSync with Ivanti Standalone Sentry

Ivanti Standalone Sentry enabled for ActiveSync serves as an intelligent gatekeeper to the ActiveSync server. It uses the ActiveSync protocol to communicate with the ActiveSync server and with ActiveSync devices. For information about these interactions, see .

Exchange ActiveSync, also known as ActiveSync, is the protocol that the ActiveSync server uses to communicate over HTTP or HTTPS with devices. The ActiveSync server uses the ActiveSync protocol to do the following:

- synchronize email, contacts, calendar, tasks and notes with a mobile device

- provide for server-device interactions relating to mobile device management and policy controls

In a deployment, these devices are called ***ActiveSync devices***. Ivanti Standalone Sentry and the UEM platform work together to protect the ActiveSync server from wrongful access by these devices.

Communication between Ivanti Standalone Sentry and ActiveSync servers is encrypted using HTTPS. Administrators can enable server TLS and configure outbound SSL. For Office 365 and GMail, Ivanti, Inc recommends that the communication should be configured to use HTTPS, so that confidential information such as user name, password, and email content are never communicated in clear text.

AppTunnel with Ivanti Standalone Sentry

Ivanti Standalone Sentry enabled for AppTunnel provides per-app secure tunneling and access control to protect app data as it moves between the device and corporate backend resources (data-in-motion). App-by-app session security protects the connection between each app container and the corporate network. AppTunnel is particularly useful when an organization does not want to open up VPN access to all apps on the device.AppTunnel is part of an AppConnect app or an Ivanti Tunnel deployment. However, AppTunnel is not a requirement for an AppConnect app deployment.

Ivanti Standalone Sentry and the UEM platform work together to provide secure access to the backend resource. For example:

- UEM provides Ivanti Standalone Sentry with the backend resource configuration for the app.

- When an app attempts to connect to a backend resource, Ivanti Standalone Sentry creates an app tunnel which is a unique combination of user, device, and app. Ivanti Standalone Sentry provides information about the app tunnel to the UEM.

- UEM informs Ivanti Standalone Sentry when an app should not be allowed to access the backend resource. For example, Ivanti Standalone Sentry blocks access to the backend resource if there are security policy violations or the AppTunnel is manually blocked.

Network traffic support with AppTunnel

TABLE 4. SUPPORTED NETWORK TRAFFIC FOR APPTUNNEL

Protocol	Support
HTTP, HTTPS tunneling	• Android and iOS AppConnect apps
TCP tunneling	• Android AppConnect apps • iOS and macOS managed apps with Tunnel configured for app proxy VPN. • Web@Work with Chromium stack enabled. TCP tunnels support HTTP and HTTPS traffic also.
IP tunneling	• Windows 10, iOS, and Android managed apps with Ivanti Tunnel. IP tunnels also support HTTP, HTTPS, TCP, and UDP traffic.

Benefits of Ivanti Standalone Sentry

Benefits of using Sentry in your deployment include the following:

- "Device and user authentication support with Ivanti Standalone Sentry" below
- "Enforcement of security policies with Ivanti Standalone Sentry" below
- "Visibility into which devices are accessing the backend resource with Sentry" below
- "Ability to take action on ActiveSync device with Sentry" on the next page

Device and user authentication support with Ivanti Standalone Sentry

Using Ivanti Standalone Sentry, you can choose how the user authenticates with the ActiveSync server or the backend resource. You can choose password authentication, certificate authentication, or Kerberos Constrained Delegation.

Enforcement of security policies with Ivanti Standalone Sentry

UEM applies security, privacy, lockdown, and sync policies to registered devices. These policies ensure that devices can connect only if they comply to your organization's security requirements. Ivanti Standalone Sentry gets device posture and compliance information from UEM, and allows access based on the device posture.

Visibility into which devices are accessing the backend resource with Sentry

When using Ivanti Standalone Sentry for ActiveSync or AppTunnel, devices access the backend resource through Ivanti Standalone Sentry. Because of this single point of access, Ivanti Standalone Sentry knows which devices and users are accessing backend resources.

Ivanti Standalone Sentry for ActiveSync creates a unique record for each combination of user and device accessing the ActiveSync server. Ivanti Standalone Sentry then associates the ActiveSync record to a device and user in UEM. Without Sentry, a user could configure multiple devices to access the ActiveSync server, and you would have no automated way of knowing about all the devices or managing access for these devices.

Ivanti Standalone Sentry for AppTunnel creates a unique AppTunnel session (connection) for each unique combination of user, app, and device. For example, Ivanti Standalone Sentry creates an AppTunnel for UserA using AppA on DeviceA, and a new AppTunnel for UserA using AppB on DeviceA. Each AppTunnel provides visibility into the user, device and the backend resources being accessed.

Ability to take action on ActiveSync device with Sentry

Ivanti Neurons for MDM

If there are policy violations, registered ActiveSync devices and AppTunnels can be blocked. You can take Allow, Block, and Remove actions on unregistered ActiveSync devices.

Ivanti Standalone Sentry deployment scenarios

In a deployment, Ivanti Standalone Sentry works with the UEM platform secures access to backend resources by preventing wrongful access from devices. The UEM can be Ivanti EPMM on a Physical or Virtual Appliance or it can be an Ivanti Neurons for MDM deployment. This section provides various deployment scenarios with Ivanti Standalone Sentry.

These deployments include:

- "Deployment with Ivanti Standalone Sentry in the DMZ" below

- "Deployment with multiple Ivanti Standalone Sentry servers" on the next page

- "Deployment with Ivanti Standalone Sentry behind a proxy" on page 14

- "Deployment with multiple ActiveSync servers or backend resources" on page 14

Deployment with Ivanti Standalone Sentry in the DMZ

Ivanti Standalone Sentry can be located in the DMZ, along with UEM, but this configuration is not required. You can alternatively:

- Put Ivanti Standalone Sentry in the DMZ and put UEM behind the corporate firewall.

- Put UEM in the DMZ and put Ivanti Standalone Sentry behind the corporate firewall.

- Put both Ivanti Standalone Sentry and UEM behind the corporate firewall.

Deployment with multiple Ivanti Standalone Sentry servers

Use multiple Ivanti Standalone Sentrys in the following situations:

- Your ActiveSync server has more users than one Ivanti Standalone Sentry can support.

An Ivanti Standalone Sentry has an upper limit for the number of registered ActiveSync devices that it can support, depending on its configuration. If your ActiveSync server supports more devices than this limit, use multiple Standalone Sentrys. Configure each Standalone Sentry to point to the same ActiveSync server (or servers if multiple ActiveSync servers back each other up).
For more information about Standalone Sentry capacity, see the *Standalone Sentry On-Premise Installation Guide*.

- You have multiple ActiveSync or backend resources, each of which supports a different organization.

Use one Ivanti Standalone Sentry for each organization. Configure the Ivanti Standalone Sentry to point to the server (or servers if multiple servers back each other up) for that organization.

- You have ActiveSync or backend resources in different locations.

If you have ActiveSync or backend resources in different locations, use an Ivanti Standalone Sentry for each location. By co-locating the Standalone Sentry with the ActiveSync or backend resource, you minimize latency between Sentry and the server. Configure each Sentry to point to its co-located server (or servers if multiple servers back each other up).

FIGURE 1. SENTRY IN DIFFERENT LOCATIONS
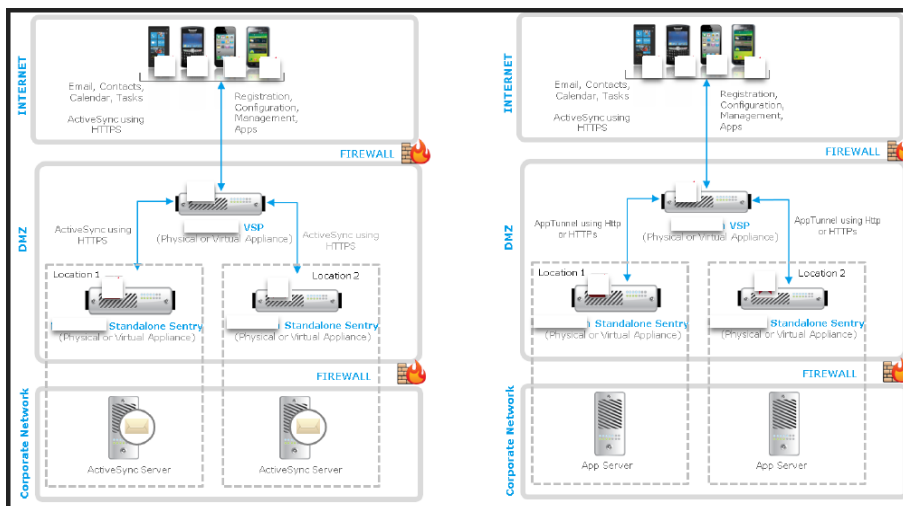
Typically, you use load balancers when using multiple Standalone Sentrys. For information about using load balancers with Standalone Sentry, contact Professional Services.

For more information about deploying Standalone Sentry for high availability and load balancing, see the following knowledge base articles:

- *Sentry HA Networking Overview and Recommendations*

- *Mail Server Resource Consumption*

Deployment with Ivanti Standalone Sentry behind a proxy

You can configure the Ivanti Standalone Sentry to be deployed behind a proxy, for example, an Apache or an F5 server. This allows for SSL termination to occur in front of Sentry even when using certificate based authentication.

By terminating SSL in the DMZ, Standalone Sentry enables an added layer of security, as well as accommodates the DMZ firewall policies.

Leveraging this configuration requires:

- Setting up an Apache or F5 proxy to front-end the Ivanti Standalone Sentry.

- Enabling this feature on Sentry via the UEM UI.

- Additional minor changes to references to hostname in some profiles.

Contact Professional Services or a certified partner to set up this deployment.

Deployment with multiple ActiveSync servers or backend resources

You can configure one Ivanti Standalone Sentry to work with multiple ActiveSync servers or backend resources that are backing each other up. You control when Standalone Sentry switches to another ActiveSync Server or backend resource by setting parameters involving communication failures Ivanti Standalone Sentry and the active ActiveSync servers or backend resource.

Ivanti Neurons for MDM, Standalone Sentry, and device interaction

The following describe Ivanti Neurons for MDM, Standalone Sentry, and device interaction:
- "When an ActiveSync device accesses email" on the next page

  - "When an app accesses the backend resource" on page 16

  - "When Ivanti Neurons for MDM detects a security policy violation" on page 16

When an ActiveSync device accesses email

The following illustrates the interaction between Standalone Sentry, UEM, and the device when the device *first* attempts to access the ActiveSync server.

FIGURE 1. DEVICE FIRST ATTEMPT TO ACCESS THE ACTIVESYNC SERVER

1. Device attempts to access the ActiveSync server.
2. Sentry queries Ivanti Neurons for MDM for registered devices and unregistered tunnels that might match the device.
   Sentry checks for unregistered tunnels to ensure that the device is not already allowed on a different Sentry registered to Ivanti Neurons for MDM.
3. Standalone Sentry correlates the list provided by Ivanti Neurons for MDM and picks the best match based on the following criteria: Active Sync ID, User ID. If a match is found, Sentry does additional checks to ensrue that the device is in compliance before allowing or blocking the device access to the ActiveSync server.
4. Standalone Sentry adds the device to its list of devices.
5. If access is allowed, device continues email processing.
   If access is blocked, the device will not be able to process email through Standalone Sentry.
6. Standalone Sentry checks in with Ivanti Neurons for MDM, at the next check-in interval, to update Ivanti Neurons for MDM with the tunnel (activesync and app) inventory in its list.

The next time a device attempts to access the ActiveSync server, the device is already in the Standalone Sentry's list. Standalone Sentry periodically checks in with Ivanti Neurons for MDM to update the compliance status for the device. The device is either allowed or blocked access based on the compliance status.

If Standalone Sentry cannot communicate with Ivanti Neurons for MDM

Allowing or blocking new device access to the ActiveSync server, if Ivanti Neurons for MDM is not accessible, is configured on Ivanti Neurons for MDM.

1. Based on the setting in Ivanti Neurons for MDM, Standalone Sentry either allows or blocks access to the ActiveSync server.
2. When the connection is reestablished, Standalone Sentry evaluates the status of the device following the steps described in "When an ActiveSync device accesses email" on the previous page.

When an app accesses the backend resource

When using Standalone Sentry for AppTunnel, when an app *first* attempts to access the backend resource, the following occurs:

1. UEM tells Standalone Sentry whether to allow or block the app's access to the backend resource based on:
   - the device's security policy and traffic control rules
   - whether the app is an authorized app
2. Standalone Sentry creates an AppTunnel for the app to access the backend resource based on the AppTunnel status provided by the UEM.
3. The AppTunnel view on the UEM now includes the new AppTunnel.
4. The next time the app attempts to access the backend resource, the app uses the AppTunnel that was created to access the backend resource.

On the first attempt, if Standalone Sentry is temporarily unable to communicate with the UEM due to, for example, a network error, the following occurs:

1. Standalone Sentry allows the app to access the backend resource.
2. At the periodic Sentry check in with Ivanti Neurons for MDM, the UEM sends Standalone Sentry the proper state of the device (allowed, blocked, or wiped).

When Ivanti Neurons for MDM detects a security policy violation

Ivanti Neurons for MDM detects a security policy violation when, for example, a device checks in. At the periodic Ivanti Neurons for MDM-Sentry check in, Standalone Sentry get the updated status for the devices and blocks the device from accessing the ActiveSync server and backend resources if Ivanti Neurons for MDM is configured so that Sentry blocks the device.

When Sentry initializes

When Standalone Sentry starts or restarts, the following occurs:

1. When a device attempts to access the ActiveSync server it is as though it is the first time. See "When an ActiveSync device accesses email" on the previous page.
2. Standalone Sentry retrieves the AppTunnels equal to the Sentry device cache size (number).

Periodic Standalone Sentry check in with Ivanti Neurons for MDM

Standalone Sentry periodically checks in with Ivanti Neurons for MDM to do the following:

- Get the updated compliance status for devices.

- Get any administrator actions taken on tunnels. Example: If a tunnel is blocked, Standalone Sentry retrieves the blocked status when it periodically checks in with Ivanti Neurons for MDM.

- Update Ivanti Neurons for MDM with the tunnel (ActiveSync and app) inventory in its list.

These are separate check ins with Ivanti Neurons for MDM and occur on different schedules.

Persistent device list

Ivanti Standalone Sentry operates using a list of ActiveSync devices that it keeps in its memory. This list is sometimes called the device cache. The information includes each device's state, such as allowed or blocked.

Checking if Ivanti Standalone Sentry can reach UEM

You can check whether Ivanti Standalone Sentry can reach UEM by using the Ivanti Standalone Sentry System Manager. See "Service Diagnosis" on page 160.

New features summary

These are cumulative release notes. If a release does not appear in this section, then there were no associated new features and enhancements.

Ivanti Standalone Sentry features for Ivanti Email+

- **Support to connect Email+ to Sentry using MSAL library**: Starting from Ivanti Standalone Sentry release 9.20.0, Sentry now supports MSAL library for Email+.

Related information from previous releases

If a release does not appear in this section, then there were no associated new features and enhancements.

- [Ivanti Standalone Sentry 9.18.0 - New features summary](#)

- [Ivanti Standalone Sentry 9.17.0 - New features summary](#)

- [Ivanti Standalone Sentry 9.16.0 - New features summary](#)

- [Ivanti Standalone Sentry 9.15.0 - New features summary](#)

- [Ivanti Standalone Sentry 9.14.0 - New features summary](#)

Standalone Sentry configuration overview

Configuration overview

You configure the initial setup of Standalone Sentry as part of the installation process. Additional Ivanti Standalone Sentry configuration for ActiveSync, AppTunnel, certificates, and preferences occurs on the UEM. These settings specify how Sentry connects to the UEM, the ActiveSync server, backend resources, and to devices. Ivanti Standalone Sentry system management occurs on the Sentry System Manager.

- "Initial setup" below

- "Ivanti Standalone Sentry configurations on Ivanti Neurons for MDM" below

- "Ivanti Standalone Sentry System Manager" on the next page

Initial setup

Initial setup is part of the Sentry installation process. You configure the Sentry administrator, network setup, including the Sentry IP address and FQDN, as part of the initial setup. For information see the *Ivanti Standalone Sentry Installation Guide*.

Before continuing with Sentry configuration using the UEM administrative portal, ensure that you have installed Ivanti Standalone Sentry.

Ivanti Standalone Sentry configurations on Ivanti Neurons for MDM

For Ivanti Standalone Sentry configurations on Ivanti Neurons for MDM, see the following:

- "Standalone Sentry for ActiveSync Email" on page 25

- "Standalone Sentry for AppTunnel" on page 61

- "Device and server authentication" on page 75

Ivanti Standalone Sentry System Manager

Settings in the System Manager include Ivanti Standalone Sentry's host name, network address, interfaces, and routes, the certificate for accessing the Sentry System Manager, and log management.

For more information, see:

- "Ivanti Standalone Sentry Settings" on page 89

- "Ivanti Standalone Sentry Security Settings" on page 133

- "Ivanti Standalone Sentry Maintenance Settings" on page 147

- "Troubleshooting" on page 153

- "Monitoring" on page 165

Accessing the Ivanti Standalone Sentry System Manager

The following provides the steps to access the Ivanti Standalone System manager.

Procedure

1. Enter the following URL in browser:

https://*<fully_qualified_hostname>*:8443

The version number will change depending on the version of the Ivanti Standalone Sentry.

2. Enter a user ID and password for the Ivanti Standalone Sentry.

You can enter the administrator ID and password specified during installation of Sentry. You can also enter credentials for any local users created on this Sentry after installation. The user ID is case sensitive.

FIGURE 1. IVANTI STANDALONE SENTRY SYSTEM MANAGER
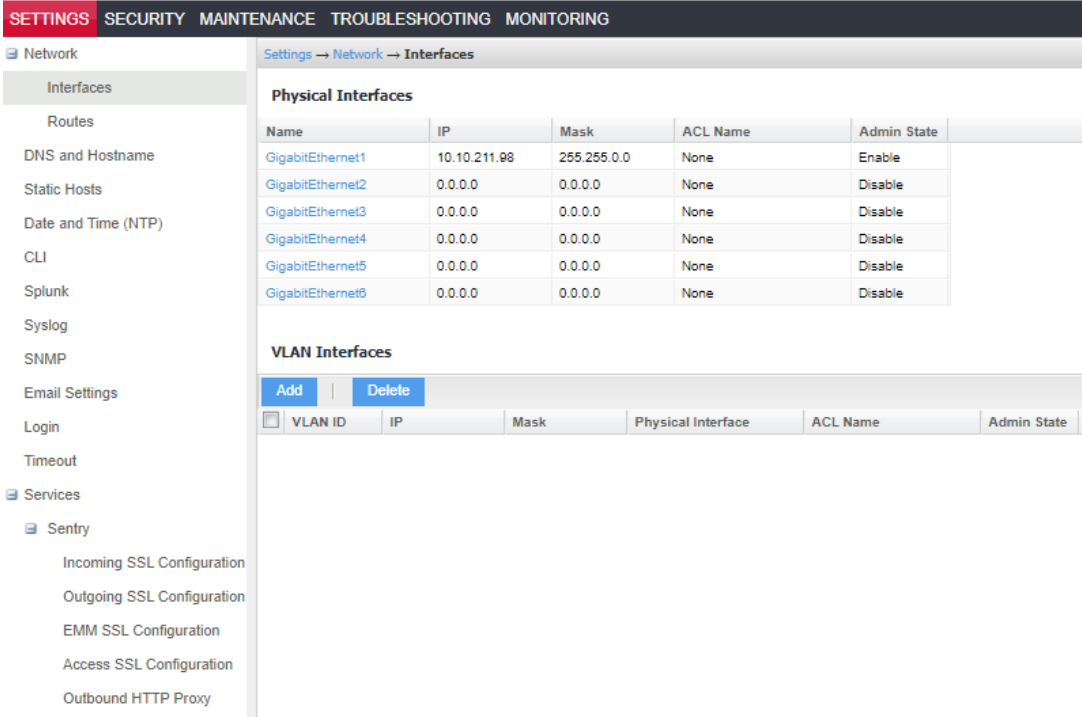
To log out of the Sentry web portal, click **Sign Out** in the upper right corner.

Standalone Sentry for ActiveSync Email

About Ivanti Standalone Sentry for ActiveSync email

Ivanti Standalone Sentry is a part of Ivanti deployment that provides secure access to your company's ActiveSync server, such as a Microsoft Exchange Server.

You configure Ivanti Standalone Sentry for email in the UEM platform. Configuring Standalone Sentry for email is a two-step process.

1. Configure a Ivanti Standalone Sentry for ActiveSync in the UEM platform.
2. Create an Exchange setting in the UEM, which points to the Ivanti Standalone Sentry.

Despite its name, you configure an Exchange app setting regardless of whether your ActiveSync server is a Microsoft Exchange Server or another type of ActiveSync server, such as a Lotus Domino server.

Before you configure Ivanti Standalone Sentry for ActiveSync

1. You must have installed Ivanti Standalone Sentry. See the *Ivanti Standalone Sentry Installation Guide*.
2. If your UEM is Ivanti Neurons for MDM, the Standalone Sentry must be registered to your cloud instance.
3. You must have the required certificate setup in your UEM.

ActiveSync uses either pass through or Identity certificate for device authentication and pass through or Kerberos for server authentication. The Identity certificate can be local or a trusted CA. If you are using an Identity certificate, you will upload the Identity certificate in the Sentry configuration you create on your UEM.

Configuring authentication using SCEP Identity (Ivanti Neurons for MDM only)

For Ivanti Neurons for MDM only, if you intend to use a SCEP identity certificate you must add the certificate to Ivanti Neurons for MDM and create the associated **App Identity Certificate** configuration.

Procedure

1. Add a local or external certificate authority in **Admin > Certificate Authority**.

A Connector installation is required if you are using an external certificate authority.

2. Add an **App Identity Certificate Configuration**. This is the SCEP identity you will use when you configure device authentication in the Standalone Sentry configuration.
3. Create an Identity Certificate setting, in **Configurations > Add > Identity Certificate**. For **Certificate Distribution**, select **Dynamically Generated** and for **Source**, select the certificate you configured in **Admin > Certificate Authority**. You will reference the Identity Certificate configuration when you create the Exchange configuration.

Configuring Ivanti Standalone Sentry for ActiveSync

You configure an Ivanti Standalone Setnry for ActiveSync by configuring the Ivanti Standalone Sentry profile with an ActiveSync Service in Ivanti Neurons for MDM.

Procedure

1. In Ivanti Neurons for MDM, go to Admin > Sentry.
2. Click **+Add Sentry Profile** or click on an existing profile to **Edit**.
3. Depending on the device authentication you will configure, select one of the following:
   - **ActiveSync with basic auth**
   - **ActiveSync and/or App Tunnel with certificates**
   - **ActiveSync and/or App Tunnel with Kerberos**
4. Use the guidelines provided in the following sections to configure Standalone Sentry for ActiveSync.
   - "Configuring Ivanti Standalone Sentry connectivity settings" on the next page
   - "Device Authentication" on the next page
   - "Default unmanaged devices behavior" on the next page
   - "Passive health check options" on page 28
   - "Scheduling options" on page 28
   - "Default HTTP/TCP timeouts" on page 28
   - "Sentry server configuration" on page 29
   - "ActiveSync service" on page 31
5. Click **Save**.
6. Create an Exchange setting that points to the Standalone Sentry.
   See "Configuring Exchange settings for Ivanti Standalone Sentry" on page 34.

Configuring Ivanti Standalone Sentry connectivity settings

The following table describes the Ivanti Standalone Sentry connectivity settings.

TABLE 1. IVANTI STANDALONE SENTRY CONNECTIVITY SETTINGS

Item	Description
Sentry Host / IP	Enter the host name (FQDN) or IP address of the server on which the Ivanti Standalone Sentry is installed. The host name or IP address must be external so that apps that are tunneling data are able to access the Sentry.
Sentry Port	Enter the port where mobile devices will connect to Ivanti Standalone Sentry. Enter 443.

Device Authentication

Device authentication determines how users attempting to connect to the ActiveSync server or backend resource authenticate with Ivanti Standalone Sentry. AppTunnel setup requires an Identity certificate or a Kerberos setup.

See "Device and server authentication" on page 75 for information on selecting and configuring a method of device authentication.

Default unmanaged devices behavior

By default, Sentry blocks unregistered devices from accessing backend resources. Use this setting to change Sentry's behavior to allow unregistered devices access to backend resources. Irrespective of this setting, you will be able to allow or block devices on a per-device basis.

TABLE 2. DEFAULT UNMANAGED DEVICES BEHAVIOR FIELD DESCRIPTION

Item	Description
Allow unmanaged devices to receive email and data	Check to allow unregistered devices access to backend resources.

Passive health check options

These settings determine when a Sentry is marked as 'dead'. If a server fails more than the number set in **Dead Threshold** within the time set in **Failure Window,** then it is marked as dead for the time set in **Dead Time**.

TABLE 3. PASSIVE HEALTH CHECK FIELD DESCRIPTION

Item	Description
Dead Threshold	Specify the number of times that a server connection can fail before the server will be marked "dead". The valid range is 1 through 1000.
Failure Window	Specify the time interval in milliseconds during which the specified number of server connection failures must occur in order for the server to be marked "dead". The valid range is 1 though 86400000 milliseconds (24 hours).
Dead Time	Specify the amount of time in milliseconds that the server should be marked "dead" after the specified number of connection failures. The valid range is 1 through 172800000 milliseconds (48 hours).

Scheduling options

The options provide additional flexibility in managing multiple Sentrys. Specify Priority or Round Robin scheduling if multiple servers are specified.

TABLE 4. SCHEDULING OPTIONS FIELD DESCRIPTION

Item	Description
Priority	The first available server in the specified list will be used, with the first server in the list having highest priority. So if the first server in the list is never unavailable, then the other servers will never be used.
Round Robin	Each server in the list will be used in turn. By default, Round Robin is enabled.

Default HTTP/TCP timeouts

These settings provide additional flexibility to configure Ivanti Standalone Sentry session timeouts. You may want to configure the session timeouts to manage server resources.

Do not make changes to the settings unless specifically instructed in the documentation or by Professional Services.

TABLE 5. DEFAULT HTTP/TCP TIMEOUTS FIELD DESCRIPTION

Item	Description
Socket read/write timeout	Specify the time in milliseconds, the Sentry should check for the socket read/write time out from either the device or the server. Enter a valid integer. The default setting is 10000, and the minimum is 1.
Server connection timeout	Specify the time in milliseconds after which the Sentry will time out when connecting to the server. Enter a valid integer. The default setting is 10000, and the minimum is 1.
Server response timeout	Specify the time in milliseconds after which the Sentry will time out when waiting for an HTTP response from the server. Enter a valid integer. The default setting is 60000, and the minimum is 1.
Device request timeout	Specify the time in milliseconds after which the Sentry will time out when waiting for an HTTP request from the device on a new or existing connection. Enter a valid integer. The default setting is 10000, and the minimum is 1.

Sentry server configuration

Configure the Sentry port, certificate, protocols and cipher suites.

- "HTTPS Port" below

- "Certificate/Key " on the next page

- "Protocols and cipher suites" on the next page

- "Load balancers and ciphers" on the next page

- "Supported protocols" on page 31

HTTPS Port

The default is 443. This is the port Sentry listens on for connections from the mobile devices.

Certificate/Key

When you first install Standalone Sentry, a self-signed certificate is also installed. Ivanti strongly recommends that you replace the default certificate with a publicly trusted certificate.

Protocols and cipher suites

Ivanti Standalone Sentry uses the ciphers and protocols defined here for incoming traffic from device to Ivanti Standalone Sentry and outgoing traffic from Ivanti Standalone Sentry to backend resources. Ciphers and protocols for incoming traffic is set in the **Incoming** tab. Ciphers and protocols for outgoing traffic is set in the **Outgoing** tab.

You can do the following:

- View the available and selected protocols and cipher suites.

- Setup custom protocol and cipher suite configuration.

- Enable SNI. (Outgoing only)

A default set of cipher suites and protocols are selected. You can customize the selected list of ciphers and protocols to match the security and system needs for your enterprise.

The available and default set of cipher suites and protocols may be updated in a release. Some cipher suites and protocols may be added, while others may be removed. Cipher suites and protocols may be removed if the platform no longer supports these cipher suites and protocols.

If you are set up to use the default cipher suites and protocols, these will be updated to the latest defaults. If you are set up to use a custom list of selected cipher suites and protocols, the custom list is preserved. However, any cipher suites or protocols that are not supported by Standalone Sentry are ignored and noted in **Monitoring** in Ivanti Standalone Sentry System Manager.

Making changes to the default list of cipher suites may impact the performance and security of traffic through Ivanti Standalone Sentry. Therefore, before making any changes to the selected cipher suites, Ivanti recommends that you understand both the performance and security impact of the changes.

Load balancers and ciphers

If you use a load balancer to perform HTTPS/GET checks against your Sentry and your Sentry uses strong ciphers, do the following:

- Make sure the ciphers enabled in your HTTPS/GET check match one of Sentry's strong ciphers.

- If you cannot change the ciphers that your HTTPS/GET check uses, you can change your check to use HTTP/GET to accomplish the same monitoring.

Supported protocols

- TLSv1

- TLSv1.2

- TLSv1.1

SNI

Server Name Indication (SNI) is an extension to TLS. SNI allows multiple hostnames to be served over HTTPS from one IP address. By default, SNI is disabled on Standalone Sentry for outgoing connections.

SNI allows a load balancer to direct incoming traffic to the correct backend server based on the hostname provided by the client, in this case, Standalone Sentry. Some backend server may require that SNI is enabled in the client.

Your Active Directory Federation Services (ADFS) may require SNI for all client communications.

> ℹ️ If SNI is enabled for Outgoing SSL connections, in some cases health check may fail if the backend server does not also support SNI. The workaround is to disable health check for the impacted server.

Advanced Traffic Control and server-side explicit proxy

These settings are not applicable for configuring an ActiveSync service on Ivanti Standalone Sentry.

ActiveSync service

In the Sentry profile, in **Manage Services**, configure **Exchange ActiveSync** to set up an ActiveSync service.

The following table describes the settings for configuring an ActiveSync service.

TABLE 6. FIELD DESCRIPTIONS FOR ACTIVESYNC SERVICE

Item	Description
Service Name	The **Service Name** identifies the ActiveSync service. A service name cannot contain these characters: 'space' \ ; * ? < > " |.
Limit Protocol Version	Choose the ActiveSync protocol version that the device and the ActiveSync server use to communicate with Standalone Sentry. If the device is already registered, you have to push the Exchange profile to the device to force the device to use the new protocol. Alternately, device users can go to iOS device **Settings > Mail > Accounts**, select the enterprise mail account, and toggle to disable and re-enable the mail account.
Server Authentication	Select how the Standalone Sentry authenticates the user to the ActiveSync server. Select Pass Through or Kerberos. The Kerberos option is only available if you selected **ActiveSync and/or App Tunnel with Kerberos**. If you select Kerberos, additional fields are displayed. See "Authentication using an Identity certificate and Kerberos constrained delegation" on page 79.
ActiveSync Servers	Enter the ActiveSync server FQDN and port. You can add multiple ActiveSync servers. For Microsoft Office 365, enter outlook.office365.com. For Gmail, enter m.google.com.

Item	Description
Enable Server TLS	Specify whether the ActiveSync servers require SSL (i.e., port 443). ℹ️ If you are using Google Apps via Standalone Sentry, you must check Enable Server TLS.
Enable Redirect Processing (451)	To disable redirect processing, clear the check box. If Enable Redirect Processing (451) is disabled, the Standalone Sentry does not handle redirection, and passes the redirect URL to the device. See also "451 redirect processing" on the next page.
Enable Active Health Check	The default setting is enabled. Clear the check box to disable the ActiveSync server health check. If enabled, when the ActiveSync server fails for the number of times configured in the Dead Threshold setting and within the number configured in the Failure Window, then the ActiveSync server status shows Unreachable. When the background health check determines that the server is live for the number configured for Live Threshold, the ActiveSync server status shows Reachable.

Configuring Exchange settings for Ivanti Standalone Sentry

The Exchange setting is pushed to the device and points to the Ivanti Standalone Sentry.

Procedure
1. In Ivanti Neurons for MDM, go to **Configurations**.
2. Click **+Add > Exchange**.
3. For Server address, enter one of the following:
   - When using Ivanti Standalone Sentry, set the server address to the Standalone Sentry's address.
   - When using Ivanti Standalone Sentry with Lotus Domino server 8.5.3.1 Upgrade Pack 1, set the server address to *<Ivanti Standalone Sentry's fully qualified domain name>*/traveler.
   - When using Ivanti Standalone Sentry with a Lotus Domino server earlier than 8.5.3.1 Upgrade Pack 1, set the server address to *<Standalone Sentry's fully qualified domain name>*/servlet/traveler.
   - If you are using load balancers, contact Professional Services.
4. Enter the variable for the ActiveSync User.
5. Enter the variable for the ActiveSync user Email.
6. Enter the ActiveSync user Account Password.
7. If Ivanti Standalone Sentry is using Identity or Group certificate for device authentication, select the **Identity Certificate** configuration to generate the identity certificate for the device.
8. After completing the form, click **Next**.
9. Select the distribution option and click **Done**.

Related topics

See "Exchange Configuration" in the *Ivanti Neurons for MDM Guide* or help to complete the fields in the form.

451 redirect processing

If 451 redirect URL is set up on your ActiveSync server, Ivanti Standalone Sentry handles the redirection when a device tries to sync. The redirect URL is not forwarded to the device.

You configure 451 redirect processing on the Ivanti Standalone Sentry by enabling or disabling the **Enable Redirect Processing (451)** field in the **ActiveSync service**. Redirect processing is enabled by default.

Ivanti Standalone Sentry Email+ Notification Service for Ivanti Neurons for MDM

You must configure Ivanti Standalone Sentry on Ivanti Neurons for MDM and key-value pairs of the Email+ application to receive email notifications on the device. The configuration is required for Ivanti Standalone Sentry to act as a Email+ Notification Service to deliver notifications. This feature is available only when it is used with Email+ 3.13.0.

Before you begin

- Ensure that you have Ivanti Neurons for MDM 74 and Sentry 9.10.0 or later.

- Ensure that you have the JWT token of CNS production server.
  A token is a randomly generated string from Ivanti, representing an authorization token for the cloud server.
  The term JWT token is also referred as Authorization Token, Token, and notification_server_ authorization across Ivanti products.

- Standalone Sentry must be configured with a publicly trusted certificate.

- Ensure that the Exchange servers are configured with the service account. The servers must have identity certificate to authenticate the service account.
  For more information on configuring service account on Microsoft Exchange server, see "Configuring a service account" on page 41.
  Also, see Microsoft documentation.

- If Exchange server version support is earlier than TLS v1.2, then the supported protocols should be configured in **Incoming SSL configuration protocols** on MICS.

Procedure

1. On Ivanti Neurons for MDM, click **Admin**

2. Under **Infrastructure**, click **Sentry** > [+ Add Sentry Profile]

3. Select **Email+ Notification Service** and click **Next**.

4. In Global Settings, enter a profile **Name** and **Notification Server Hostname** (Sentry hostname). The port is set at 443.

5. Verify the **Sentry Server Configuration** and click **Next**.

These fields are at default values. Ensure that the TLS certificate is a third party trusted certificate.

6. Under **Services** > **Manage Services**, click **Email+ Notification Service**.
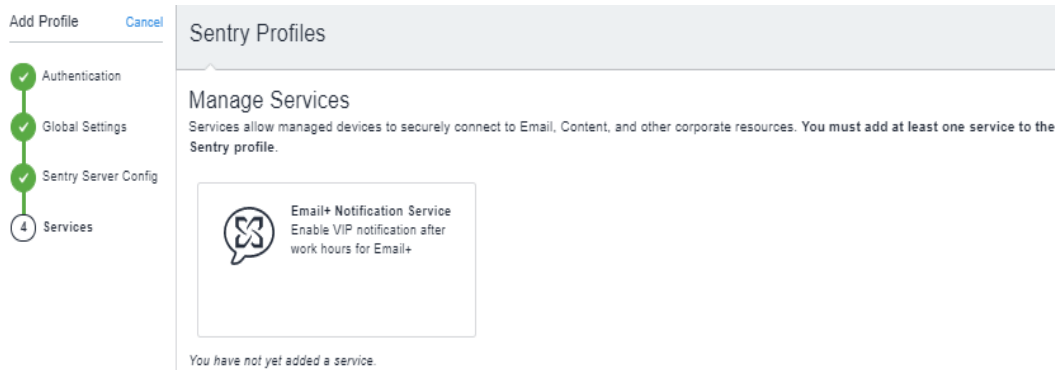
7. Enter the following details:

- **Notification Server Authorization**: The token received after registering for the Ivanti Neurons for MDM Notification Service.

- **Add the Service Account**: Click Add to upload the Exchange Service account certificate.

- **Add the Exchange Server**: Click Add to add the desired Exchange Server. The port is set at 443.

- Retain the other fields at default values.

8. On Standalone Sentry, register Sentry to Ivanti Neurons for MDM using CLI.
   "`registration tenantadminuser`".

9. Under **Sentry** > **Sentry Profiles**, click **Actions** to assign the registered Sentry to the ENS profile created above.

Sentry
Show Description

Download installer   + Add Sentry Profile

Unconfigured Sentry Servers
The following Sentry servers are registered but not assigned to any Sentry Profile.

SENTRY HOSTNAME	SERVER SIZE	VERSION
ip-10-84-98-121	small	Sentry Standalone 9.8.1 Build 21

Sentry Profiles

ENS profile

SENTRY HOSTNAME	SERVER SIZE	VERSION	STATUS	LAST CON
There is no information to display.				

10. On Ivanti Neurons for MDM, click **Apps** > **App Catalog** > **Add** to add the Email+ configuration. Select the appropriate configuration for the Email+ application and under **Apps Configurations**, enter the following details.

11. Click **Done**.

Next steps

You must configure the key-value pairs for Email+ notification services. For more information, see "Additional configurations using key-value pairs" in the *Email+ Guide*.

Configuring a service account

Service account on Microsoft Exchange impersonates other mailboxes when accessing exchange over various supported protocols. Following are the main steps for configuring service account.

- Setting up service accounts on Exchange server

- Configuring a service account on Exchange server

Setting up service accounts on Exchange server

For the purpose of Exchange Notification Proxy (ENP), Microsoft's Exchange Web Services (EWS) protocol is used to access mailbox messages. For example service account is assigned to the following role:

```
ApplicationImpersonation
```

The EWS sends requests with the credentials of a single service account which includes an .XML key.

```
<soap:Header>
<t:RequestServerVersion Version="Exchange2013" />
<!-- The following causes the request to run as alfred@contoso.com -->
<t:ExchangeImpersonation>
<t:ConnectingSID>
<t:SmtpAddress>alfred@contoso.com</t:SmtpAddress>
</t:ConnectingSID>
</t:ExchangeImpersonation>
</soap:Header>
```

This allows a single account to access the mailbox of other accounts.

Configuring a service account on Microsoft Exchange server

Procedure

1. In the Microsoft Exchange Management console, open a browser and type in URL. For example: https://<hostname>/ecp

2. Log in as an Admin, go to **Mail** > **Options** > **Manage My Organization** > **Roles & Auditing**> **Mailboxes** and create a new Role group.

3. Add the **applicationImpersonation** role to the group.

4. Add members to the group.

5. Click **Save** to finish.

For more information on configuring service account on Microsoft Exchange server, see Microsoft documentation.

A device authenticating to Ivanti EPMM with a certificate is also known as certificate-based authentication (CBA) to Ivanti EPMM.

OAuth for Ivanti Standalone Sentry on Ivanti Neurons for MDM

OAuth is supported with Ivanti Standalone Sentry for Office 365. The following scenarios must be compliant for OAuth to function correctly:

- The email client must support OAuth (iOS Native Mail, iOS Email+ and Android Email+)

- UEM must push an OAuth configuration to the email client

- UEM must enable Sentry for OAuth

> **i** Sentry 9.14 supports Azure AD Conditional Access Policy.
> For more information, see Configuring conditional access rules in Azure.

Configuring Sentry on Ivanti Neurons for MDM for OAuth

You must configure Sentry to enable OAuth and provide the endpoints.

Before you begin

- Verify that you have Sentry 9.12.0 or later and Ivanti Neurons for MDM 75 or later.

Procedure

1. Login to Ivanti Neurons for MDM with admin credentials.
2. Click **Admin** > **Sentry** > **Add Sentry Profile**.
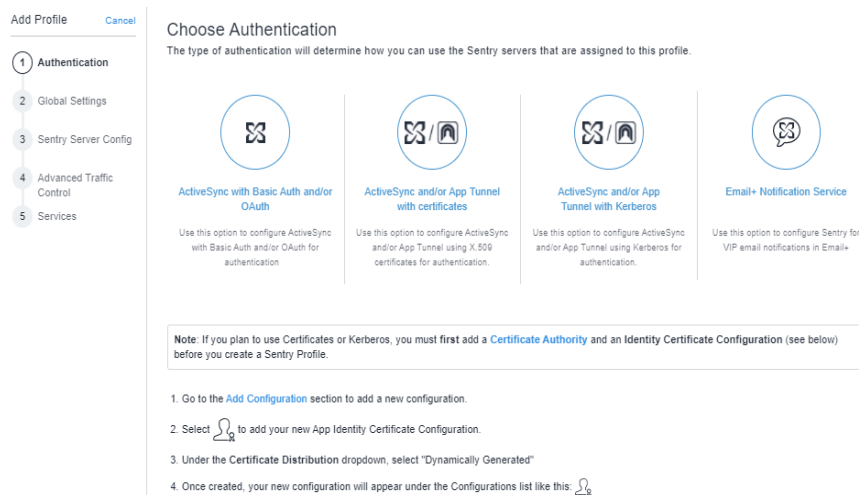
3. Select **ActiveSync with Basic Auth and/or OAuth** and click **Next**.
4. Enter the **Name**, **Hostname**, and **Port** details and click **Next**.
5. Enter the Sentry Server Configuration details and click **Next**.
6. Click **Exchange ActiveSync** and enter the following details:

   - **Destination OAuth2 Authorization Endpoint**:
     https://login.windows.net/common/oauth2/authorize

   - **Destination OAuth2 Token Endpoint**: https://login.windows.net/common/oauth2/token

   - **Sentry Resource**: https://<SentryHostname>

   - **Destination Resource**: https://outlook.office365.com

7. Click **Next** > **Save**.

Configuring iOS native email configuration with OAuth

You must enter the Sentry OAuth Sign In URL.

Before you begin

Verify that you have added the Sentry profile with OAuth settings.

> ℹ️ With iOS 14 and newer versions, password prompts appear on the devices for iOS Native email client when the exchange profile with OAuth configuration is pushed to the devices.

Procedure

1. Login to Ivanti Neurons for MDM with admin credentials.

2. Click **Configurations** > **Add** > **Exchange**.

3. Enter the name and description.

4. In Configuration Setup, select **Enable OAuth for exchange payload**.

5. Enter the following details:

   a. **OAuth Sign In URL**: https://<Sentry HostName>/proxyservice/oauth2/authorize

   b. **OAuth Token Request URL**: https://<Sentry HostName>/proxyservice/oauth2/token

6. To modify the Exchange settings:
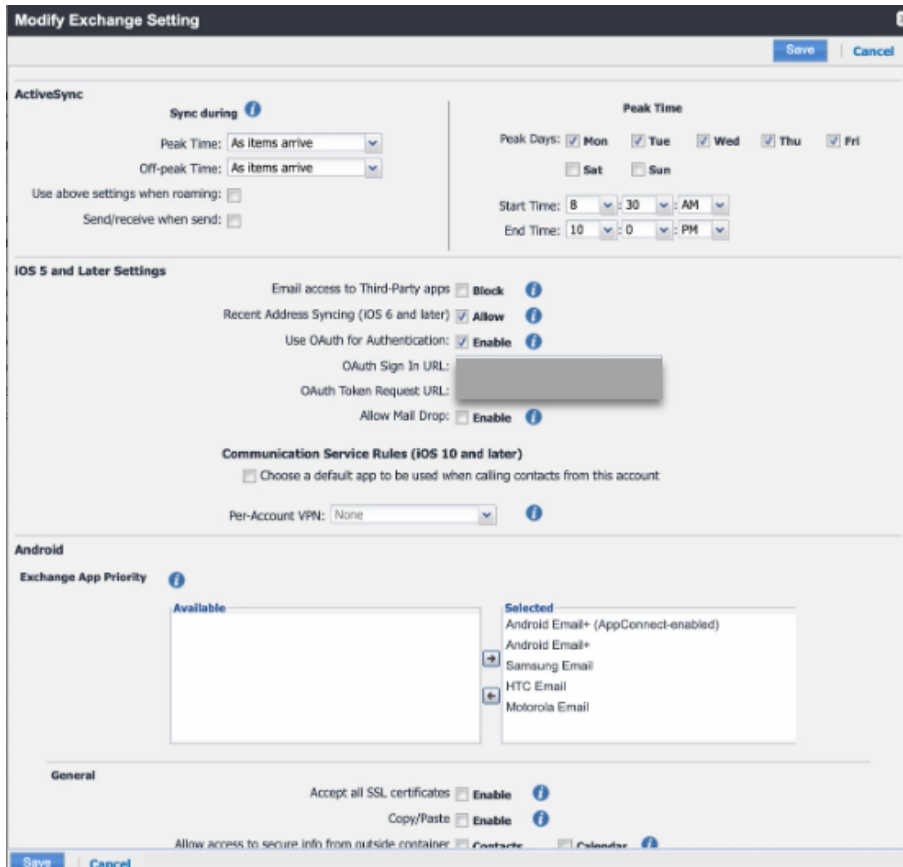
7. Enter the VPN profile's Safari domains for Tunnel requests to go through the domains used in OAuth and multi-factor Authentication:

8. Click **Save**.

Configuring Android and iOS Email+ configuration with OAuth

1. Login to Ivanti Neurons for MDM with admin credentials.

2. Click **Apps**> **App Catalog**> **Add**.

3. Select Email+ 3.8.0. (Android AppConnect) for Android configuration.

4. Enter the details and click **Next** to add the application.

FIGURE 1. ANDROID EMAIL+ CONFIGURATION

← Back to list   App Catalog   Categories   Reviews   Catalog Settings   Distribution Filters

Email+
Not available | Version 3.8.0.0.148-T9.0.0.0.32-148 ∨ |AppConnect ✓ | Delegation Status: App is not deleg

Details   Distribution   **App Configurations**   Reviews   App Config Feedback

App Configurations Summary › **Email+ Configuration**

← Back to list   Edit

Configuration Setup

Name
Email+ Config

Description
N/A

Updated
17 minutes ago

AppConnect Custom Configuration ⓘ

Key	Value
eas_min_allowed_auth_mode	modern_auth
email_watermark	$NULL$
show_dialing_confirmation	true
allow_detailed_notifications	false
gal_search_display_name	true
default_network_timeout	90
email_password	$PASSWORD$
allow_export_contacts	true
email_alert_unsafe_domains	false
allow_logging	false
disable_analytics	false
email_exchange_host	
email_max_attachment	10
email_safe_domains	$NULL$
email_ssl_required	true
email_watermark_parameters	#11888888,20,78,70
migrate_from_email_exchange_host	null
email_max_sync_period	0
email_default_sync_period	2
email_address	
email_exchange_username	
contacts_display_order	first_last
gal_search_minimum_characters	4
email_trust_all_certificates	false
prompt_email_password	false
show_pictures_default	false
allow_export_contacts_to_email	true
limit_contact_export_to	all
email_device_id	$DEVICE_UUID_NO_DASHES$
modern_auth_authority_url	
modern_auth_resource_url	

AppConnect Certificate Configuration

Key	Certificate

Data Loss Prevention

Note: These will override any device configurations sent for users with this app.
☐ Allow Screen Capture

Distribute this App Config
Choose one of these options

👥   👤   ⥄

Everyone with App   **No One**   **Custom**

All Users who have the app   Stage this App Config for later distribution   This config goes to a custom defined set of users and/or user groups

FIGURE 2. IOS EMAIL+ CONFIGURATION

5. Email+ iOS also has AppTunnel definitions that send traffic to specific domains through Ivanti Tunnel.

For more information on adding an application on Ivanti Neurons for MDM, see "Adding an In-house app" in the *Ivanti Neurons for MDM Administrator Guide*.

KVPs for Email+ configuration

- eas_min_allowed_auth_mode: modern_auth

- modern_auth_authority_url: https://<SentryHostname>/proxyservice

- modern_auth_resource_url: https://<SentryHostname>

- email_exchange_host =<sentry_address>

Configuring conditional access policy in Azure AD

You can configure the conditional access rules in Azure for OAuth to function correctly.

1. Login to Azure portal with admin credentials.
   The admin has to be super admin who has premium features to configure Conditional Access rules.

2. Click **Azure AD Conditional Access** > **Named Locations** > **IP Range Locations** > **New IP Range Location**.

3. Click **Add** and enter the IPv4 or IPv6 address range.

4. Configure a name and Sentry IP address with **Subnet** > **Add** > and enable **Mark as Trusted location** > **Create**.

5. On the **Home** tab, click **Conditional Access Policies** > **Create New Policy**.

6. Under **Users and Groups**, select Users and Groups.

7. Search for the appropriate Users or Groups and click **Select**.

8. Under **Cloud apps or actions**, select **apps** > **Office 365**.

9. Under **Conditions** > **Locations** > **Any Location** > Configure "**Yes**" under Include to "**Any Location**".

10. Under list of locations, select **Selected locations** under **Exclude**.

11. Select **Grant access as block access** > **require one of the selected controls**.

12. Select **Enable Policy** > **On** > **Create**.

Multi-factor authentication configuration for Ivanti Neurons for MDM

Configuration of native email on Ivanti Neurons for MDM for multi-factor authentication

Procedure

1. On Sentry 1, configure OAuth. See "OAuth for Ivanti Standalone Sentry on Ivanti Neurons for MDM" on page 43.

2. On Sentry 2, configure VPN on UEM.

3. In Safari domain, add the required MS domains and save the settings.
    * login.windows.net
    * login.microsoftonline.com

SCEP Identity: # OAuth_ID cert

VPN Sub Type: -

Debug Info Recipient: a@aol.com

Legacy App Support: Enabled for iOS 7 & 8 only

DNS Resolver IP:

Search Domains:

Match Domains:

Custom Data:

None

Safari Domains:

Safari Domain
login.windows.net
login.microsoftonline.com

Associated Domains:

None

Excluded Domains:

None

Mail Domains:

None

Contacts Domains:

None

Calendar Domains:

None

Disconnection Timeout: 60

Rules

Rule Type	Value
All	Any

Action: Connect

4. On Sentry 2, configure tunnel services with identity certificate.

Device Authentication Mode
Authentication Mode: SCEP Identity

SCEP Identity Name: # OAuth_ID cert

Enable Certificate revocation list (CRL): false

Default Unmanaged Devices Behavior
Allow unmanaged devices to receive email and data: false

Passive health check Options
Dead Threshold: 3

Failure Window: 60000 ms

Dead Time: 120000 ms

Scheduling Options
Scheduling: Round Robin

Default HTTP/TCP Timeouts
Socket read/write timeout: 10000 ms

Server connection timeout: 10000 ms

Server response timeout: 60000 ms

Device request timeout: 10000 ms

Sentry Server Config
Listener Protocol: HTTPS Only

Https Port: 443

Use Sentry's self-signed cert: false

Certificate Names: Sentry_new_public_cert_use_this_only

Incoming

Protocols: TLS_V_1_2, TLS_V_1_1, TLS_V_1_0

Cipher suites: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_RSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA

Outgoing

Enable SNI: false
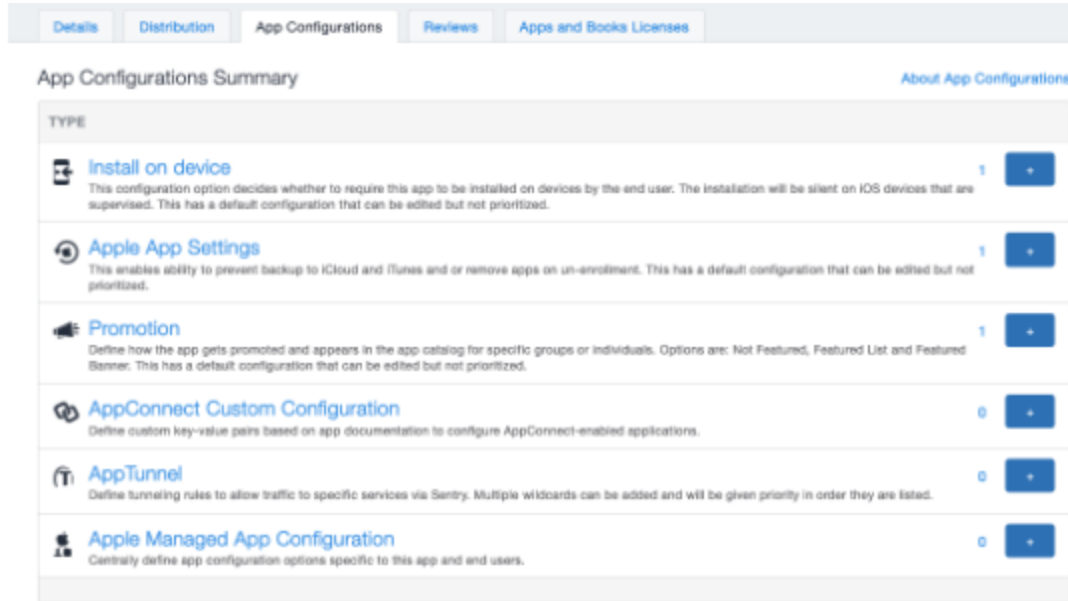
Protocols: TLS_V_1_2, TLS_V_1_1, TLS_V_1_0

Cipher suites: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_256_GCM_SHA384,
TLS_RSA_WITH_AES_128_GCM_SHA256,
TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA

Advanced Traffic Controls Enabled.: false

Service Name: Tunnel_TCP

OS: iOS/macOS

Service Type: MobileIron Tunnel

Server Authentication: Pass through (Basic Authentication)

Proxy Type: All destinations (forward proxy)

5. Configure Ivanti Tunnel application under **Apps Catalog.**
   App Tunnel or App-Connect configuration is not required and distribute the configuration.

6. Register the device with Office 365 user using Go client.

7. Download the Ivanti Tunnel application and complete the device registration.
   The user is now redirected to MS online to enter the password.

8. Enter the password.

The user is now prompted for MFA on the device.
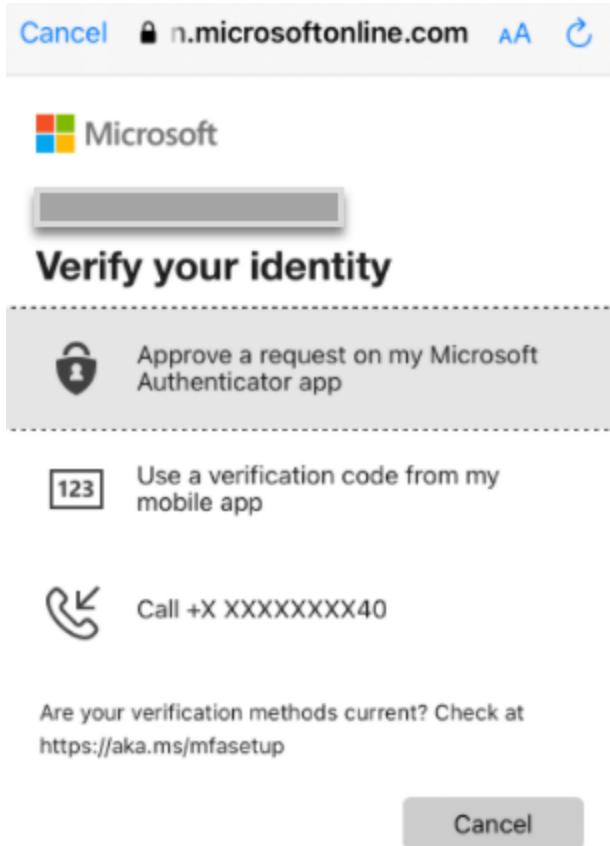
Standalone Sentry for AppTunnel

About Ivanti Standalone Sentry for AppTunnel

Ivanti Standalone Sentry configured for AppTunnel provides users secure access from an app on their device to your company's backend resource such as a SharePoint server. AppTunnel protects app data-in-motion by providing app-by-app session security between an app and the corporate resource.

You configure Ivanti Standalone Sentry for AppTunnel in the UEM. AppTunnel is part of Ivanti Tunnel deployment or an AppConnect app deployment.

Deploying AppTunnel is a two-step process:
1. In the UEM configure a Standalone Sentry for AppTunnel.
2. Configure one of the following for apps:
   - Configure an AppConnect app.

   Or
   - Configure Ivanti Tunnel.

Before you configure Ivanti Standalone Sentry for AppTunnel

1. You must have installed Ivanti Standalone Sentry. See the *Ivanti Standalone Sentry Installation Guide*.
2. If your UEM is Ivanti Neurons for MDM, the Ivanti Standalone Sentry must be registered to your cloud instance.
3. You must have the required certificate setup in your UEM.

AppTunnel uses either an Identity certificate for device authentication and pass through or Kerberos for server authentication. The Identity certificate can be local or a trusted CA. If you are using an Identity certificate, you will upload the identity certificate in the Sentry configuration you create on your UEM. See "Configuring authentication using SCEP Identity (Ivanti Neurons for MDM only)" on the next page.

Related topics

- "About Ivanti Standalone Sentry for AppTunnel" above

- "Configuring Ivanti Standalone Sentry for AppTunnel" below

Configuring authentication using SCEP Identity (Ivanti Neurons for MDM only)

For Ivanti Neurons for MDM only, if you intend to use a SCEP identity certificate you must add the Certificate to Ivanti Neurons for MDM and create the associated App Identity Certificate configuration.

Procedure

1. Add a local or external certificate authority in **Admin > Certificate Authority**.

A Connector installation is required if you are using an external certificate authority.

2. Add an **App Identity Certificate Configuration**. For source, select the certificate you configured in **Admin > Certificate Authority**. This is the SCEP identity you will use when you configure device authentication in the Standalone Sentry configuration.
3. Create an Identity Certificate setting, in **Configurations > Add > Identity Certificate**. For **Certificate Distribution**, select **Dynamically Generated** and for **Source**, select the certificate you configured in **Admin > Certificate Authority**. If the app configuration requires, you will reference the Identity Certificate configuration in the app's configuration that uses an AppTunnel service or the Tunnel service.

Configuring Ivanti Standalone Sentry for AppTunnel

If you configure AppTunnel on a Ivanti Standalone Sentry that was already configured for ActiveSync, and you change the device authentication options, ensure that the associated Exchange profile matches the device authentication options.

Procedure

1. In Ivanti Neurons for MDM, go to Admin > Sentry.

2. Click **+Add Sentry Profile** or click on an existing profile to **Edit**.

3. Depending on the device authentication you will configure, select one of the following:

   - ActiveSync and/or App Tunnel with certificates

   - ActiveSync and/or App Tunnel with Kerberos

4. Use the guidelines provided in the following sections to configure Ivanti Standalone Sentry for AppTunnel.

- "Ivanti Standalone Sentry connectivity global settings" below

- "Device Authentication" below

- "Default unmanaged devices behavior" below

- "Passive health check options" on the next page

- "Scheduling options" on the next page

- "Default HTTP/TCP timeouts" on the next page

- "Sentry server configuration" on the next page

- "Advanced Traffic Control and server-side explicit proxy" on the next page

- "AppTunnel service" on page 69

5. Click **Save**.

6. Assign the profile to a registered Ivanti Standalone Sentry.
   Only one profile can be assigned to a Ivanti Standalone Sentry.

Ivanti Standalone Sentry connectivity global settings

See "Configuring Ivanti Standalone Sentry connectivity settings" on page 27.

Device Authentication

Device authentication determines how users attempting to connect to the ActiveSync server or backend resource authenticate with Standalone Sentry. AppTunnel setup requires an Identity certificate or a Kerberos setup.

See "Device and server authentication" on page 75 for information on selecting and configuring a method of device authentication.

Default unmanaged devices behavior

By default, Sentry blocks unregistered devices from accessing backend resources.

See .

Passive health check options

These settings determine when a server is marked as 'dead'.

See .

Scheduling options

The options provide additional flexibility in managing multiple Sentrys.

See .

Default HTTP/TCP timeouts

Do not make changes to the settings unless specifically instructed in the documentation or by Professional Services.

See .

Sentry server configuration

Configure the Sentry port, certificate, protocols and cipher suites.

See .

Advanced Traffic Control and server-side explicit proxy

-

-

Advanced traffic control

Advance Traffic Control (ATC) allows you to manage access to backend resources based on which app the traffic is coming from and the destination IP address or domain name. ATC provides administrators additional control and flexibility in how traffic to backend resources are managed. You can specify whether traffic to the backend resource is through a proxy server, allowed direct access, or blocked.

Example: You may want to direct Safari traffic to go through a certain proxy server and all other traffic to go directly to backend resources. In this case, you would configure the Safari bundle ID in the Application BundleID and select the proxy server to direct Safari traffic, and set the Default Action to Allow.

> If you are using a Ivanti Standalone Sentry version 8.5.0 or earlier and configure ATC rules, Ivanti Standalone Sentry will ignore the ATC rules.

TABLE 5. RULE TYPE SUPPORTED BY APP

Rule type	App - Services
IP-based rule	• Ivanti Tunnel for Android and Windows.
Domain-based rule	• Ivanti Tunnel for iOS • AppConnect apps for iOS and Android devices. • SharePoint for Docs@Work • CIFS for Docs@Work • Web@Work

Server-side explicit proxy

Ivanti Standalone Sentry supports sending traffic through an HTTP proxy server to access corporate resources. The proxy server is located behind the firewall and sits between the Sentry and corporate resources. This deployment allows you to access corporate resources without having to open the ports that Sentry would otherwise require.

Consider the following:

- This configuration is only supported for AppTunnel traffic.

- Proxy is configured for each AppTunnel service. You may configure proxy for some AppTunnel services and not for other AppTunnel services on the same Sentry.

- The same proxy server may be configured on multiple Sentrys.

Ivanti Standalone Sentry filters HTTP traffic through a TCP tunnel that uses server-side explicit proxy. For HTTP traffic through a TCP tunnel, if server-side explicit proxy is configured, Standalone Sentry will treat the explicit proxy as HTTP proxy. The HTTP request URL will be modified to include the target host.

In all other cases, Ivanti Standalone Sentry treats the explicit proxy server as a TCP proxy server. Sentry will send a HTTP CONNECT request to the explicit proxy, followed by TCP data.

TABLE 6. FIELD DESCRIPTIONS FOR ADVANCED TRAFFIC CONTROL (ATC)

Item	Description
Advanced Traffic Control	Select to enable advanced traffic control.
Server-side Proxy List Traffic is directed to the proxy servers listed here based on the backend resource and action defined in **Traffic Control Rules.**	
Name	Enter a unique name for the proxy server. The name for the proxy server will be available for selection in the **Proxy** field for Traffic Control Rules.
Hostname	Enter the IP address or FQDN for the proxy server.
Port	Enter the port number for the proxy server.
+	Click to add a proxy server.
Traffic control rules Specify whether traffic from an app to the backend resource is through a proxy server, allowed direct access, or blocked. You can create multiple rule sets. Each rule set can have multiple rules. However, all rules in a rule set can only be one type, either IP based or domain based. You cannot create a mix of IP based and domain based rules in a rule set. • Rules in a rule set are matched based on the order in which they are listed. This is especially important for domain names with wildcards. For example, if the Block action is selected for *.company.com, and the Proxy action is selected for *.internal.company.com, and the rule for *.company.com is listed first, then all company.com domains will be blocked. Use the up and down arrows to order the rules. • If AppTunnel traffic is blocked due to traffic control rules, the AppTunnel entry is not reported in the Sentry tab in device details for a device.	

Standalone Sentry for AppTunnel

TABLE 6. FIELD DESCRIPTIONS FOR ADVANCED TRAFFIC CONTROL (ATC) (CONT.)

Item	Description
Rule type	Select one of the following: • IP based: The rule is matched based on the IP address of the destination host. IP based rules are supported only for IP traffic from Windows and Android devices going through Ivanti Tunnel. The destination has to be an IP address or an IP range. Example: 10.0.0.0/8, 10.5.0.0/16, 10.5.2.0/24, 10.5.5.2/32 • Domain based: The rule is matched based on the domain of the destination host and the bundle ID of the app. Domain based rules are supported only for HTTP/S and TCP traffic from iOS devices going through Ivanti Tunnel. The destination has to be a domain name.
Destination Host	Enter the IP address or domain name of the backend resource: • IP based rules: Enter an IP address or range. The IP address range must conform CIDR format. Example: 192.168.0.15/24 • Domain based rules: Port numbers are not supported. Wildcards are supported. Only the suffix after the * wildcard is matched. Example: *.acme.com.
Application Bundle ID	Enter the app bundle ID for which you are creating the domain-based rule. The bundle ID can include "*" for wildcard matching. The bundle ID can be used in conjunction with Destination Host. If you are using the bundle ID with a destination host, then the rule will be applied only to traffic from the app directed to the destination host.
Action	Select **Proxy**, **Allow**, or **Block**.
Proxy	If you selected **Proxy** for **Action**, then select the proxy server for the backend resource.
Default Action	The default action is applied if traffic control rules is not defined for a backend resource.
Proxy	If you choose **Proxy** as the default action, select the proxy server for traffic to the backend resource.

Copyright © 2024, Ivanti, Inc. All Rights Reserved. Privacy and Legal.

Page 68 of 210

AppTunnel service

Select one of the app services to create an AppTunnel service for that app.

-

-

-

Service types and supported traffic

The following table describes the service type available when configuring AppTunnel and the traffic type supported by the service.

TABLE 7. SERVICE TYPE AND SUPPORTED APPTUNNEL TRAFFIC

Service type	Supports traffic type
Sharepoint for Docs@Work	• iOS: HTTP, HTTPS • Android: HTTP, HTTPS
CIFS for Docs@Work	• iOS: CIFS • Android: CIFS
Web@Work	• iOS: HTTP, HTTPS • Android: HTTP, HTTPS
Ivanti Tunnel	• iOS: HTTPS, TCP, IP (split tunneling for UDP traffic) • Windows: HTTPS, TCP, IP (including UDP). • Android: HTTPS, TCP, IP (including UDP).
Help@Work	• iOS: TCP (includes HTTP and HTTPS)
Custom HTTP	• HTTP and HTTPS • Any service other than TCP and IP
Custom TCP	• TCP

Field description for AppTunnel service

The following table describes the fields available for configuring an AppTunnel service. The fields available for configuration changes depending on the type of service you are configuring.

TABLE 8. FIELD DESCRIPTIONS FOR APPTUNNEL SERVICE

Item	Description
Service Name	The **Service Name** identifies the AppTunnel service. The service name is referenced in the AppConnect app configuration for configuring tunneling for the AppConnect app. The app is restricted to accessing the backend resources listed in the **Server List** field. A service name cannot contain these characters: 'space' \ ; * ? < > " |.
Service Type (only Ivanti Tunnel iOS or macOS)	IP: Select to configure packet-tunnel provider type VPN. TCP: Select to configure app-proxy VPN.
Enable DFS	Select the check box if you are configuring a DFS site in Docs@Work. Only for Docs@Work for CIFS service.
Server Authentication	Select the authentication scheme for the Standalone Sentry to use to authenticate the user to the backend resource: **Pass through (Basic Authentication)** The Sentry passes through the authentication credentials, such as the user ID and password (basic authentication) or NTLM, to the backend resource. For TCP and IP tunneling, select Pass Through. The Sentry passes through all TCP or IP packets to the backend resource. - Kerberos The Sentry uses Kerberos Constrained Delegation (KCD). KCD supports Single Sign On (SSO). SSO means that the device user does not have to enter any credentials when the AppConnect app accesses the backend resource. The **Kerberos** option is only available if: **You selected ActiveSync and/or App Tunnel with Kerberos.** **Kerberos authentication is now supported with Exchange ActiveSync, Web@Work, SharePoint for Docs@Work, and CIFS for Docs@Work services.** If you select Kerberos, additional fields are displayed. See "Authentication using an Identity certificate and Kerberos constrained delegation" on page 79.

TABLE 8. FIELD DESCRIPTIONS FOR APPTUNNEL SERVICE (CONT.)

Item	Description
All destinations (forward proxy)	Select if you want all traffic from the app to go through Standalone Sentry.
Specific destinations (reverse proxy)	Select if you want to restrict traffic to only specific servers. Standalone Sentry sends through only requests to the configured servers.
Servers (reverse proxy)	Add the backend resource's fully qualified domain name (FQDN) and port number on the backend resource that the Sentry can access. Example:sharepoint1.companyname.com:443 Acceptable characters in a host name are letters, digits, and a hyphen. The name must begin with a letter or digit. You can enter multiple resources. The Sentry uses a round-robin distribution to load balance. That is, it sets up the first tunnel with the first resource, the next with the next resource, and so on.
Enable Server TLS	Select **Enable Server TLS** if the servers listed in the **Servers** field require SSL. ℹ Although port 443 is typically used for https and requires SSL, the backend resource can use other port numbers requiring SSL.
ATC Rule Set	Select the ATC rule set if you want to manage the AppTunnel service traffic through ATC rules. You must also have configured Advanced Traffic Control (ATC) rules. Only the rule type supported by the app will be listed.
Add Context Headers	Select the check box to forward additional device context information to your corporate backend resource. This allows your corporate backend resources to further validate the device. ℹ Context headers are not supported for TCP tunneling.

About context headers

As an administrator you may require your corporate backend resources to further validate the devices accessing these resources. In these cases, Standalone Sentry forwards context information in the header. You enable context headers in a service configuration. So, you can decide to enable context headers for some services and not enable context headers for other services.

> ℹ️ If server-side explicit proxy is configured, the request to the proxy (HTTP CONNECT) includes the context headers.

TABLE 9. INFORMATION IN CONTEXT HEADERS

Header Name	Description
X-Ivanti-DEVICE-UUID	Device UUID The Device UUID can be used in API calls to Ivanti Neurons for MDM to collect more information about the device.
X-Ivanti-USER-UPN	User Principal name
X-Ivanti-USER-DN	User DN (if available)
X-Ivanti-USER-CERT	User Identity certificate The certificate is represented in a Privacy Enhanced Mail (PEM) encoding without the header or the trailer information.

Configuring apps

After configuring AppTunnel on Ivanti Standalone Sentry, configure the apps to use AppTunnel. See the following documents for information on configuring apps:

TABLE 1. DOCUMENTATION REFERENCE

Product	Documents
Ivanti Neurons for MDM	Ivanti Neurons for MDM Guide or Help documentation.
Ivanti Tunnel	Ivanti Tunnel for Android Guide. Ivanti Tunnel for iOS Guide. Ivanti Tunnel for macOS Guide. Ivanti Tunnel for Windows Guide.
Email+	Email+ for iOS Guide. Email+ for Android Guide.
Docs@Work	Docs@Work for iOS Guide. Docs@Work for Android Guide.

Device and server authentication

Overview

Ivanti Standalone Sentry supports device authentication using user name and password, certificate-based authentication, or Kerberos Constrained Delegation. Device authentication involves configuring:

- device authentication (how the device authenticates to the Ivanti Standalone Sentry)

- server authentication (how the Ivanti Standalone Sentry authenticates the device to the server).

Device authentication

Device authentication specifies how the device authenticates to the Ivanti Standalone Sentry. The following table describes the devices authentication methods supported by Ivanti Standalone Sentry.

TABLE 1. SUPPORTED DEVICE AUTHENTICATION

Device Authentication	Description
Pass Through	Only available if you are using the Sentry for ActiveSync only. The Sentry passes through the authentication provided by the device, for example, user name and password, NTLM.
Group Certificate	Available for ActiveSync and AppTunnel. Requires the following: • A trusted group certificate for device authentication. • A authentication method like user name and password or NTLM for authenticating the device to the server. ⓘ KCD is not supported with Group Certificates.
Identity Certificate	Available for ActiveSync and AppTunnel. Requires the following: • A certificate issued by a Trusted Root Authority for device authentication • A user name and password or a properly configured Kerberos implementation for authenticating the device to the server.

Server authentication

Server authentication specifies how the Sentry authenticates the device to the backend resource. This can be the ActiveSync server or a backend resource. The following table describes the supported server authentication. These are supported for both ActiveSync and AppTunnel.

TABLE 2. SUPPORTED SERVER AUTHENTICATION

Server Authentication	Description
Pass Through	The Sentry passes through the authentication provided by the device. For example: user name and password, NTLM. ⓘ This is the only authentication option you can use with Microsoft Office 365. This is also the only authentication option available for TCP and IP tunneling.
Kerberos	Only available if you choose Identity Certificate for device authentication. Requires a properly configured Kerberos implementation.

Configuring device and server authentication

You specify the device and server authentication in the Sentry configuration when you create a Sentry profile in Admin > Sentry. Click **Add Sentry Profile** or edit an existing Sentry profile.

Select one of the following:

- **ActiveSync with basic auth**: Only for Sentry for ActiveSync. Sentry passes through the user name and password provided by the device user.

- **ActiveSync and/or AppTunnel with certificates**: Select if you plan to use Identity certificates (X.509) for device authentication. Pass through is the only option available for server authentication

- **ActiveSync and/or App Tunnel with Kerberos**: Select if you plan to use Kerberos to authenticate the device to the server.

If you do device authentication with Identity certificates, you can specify different server authentication types for the ActiveSync configuration and for each AppTunnel service. For example, you can specify Pass Through for the ActiveSync server and Kerberos Constrained Delegation (KCD) for the servers listed for an AppTunnel service.

Procedure
1. Obtain the certificates required for your implementation.
2. In **Admin > Sentry**, click **Add Sentry Profile** or edit an existing Sentry profile.
3. For **Authentication** select one of the following authentication options, depending on your implementation:
   - **ActiveSync with basic auth**
   See "ActiveSync with basic auth and pass through" on the next page for next steps.

- **ActiveSync and/or AppTunnel with group certificates**

See "Authentication using a group certificate and pass through" below for next steps.

- **ActiveSync and/or AppTunnel with Identity Certificate**

See "Authentication using a SCEP Identity certificate and pass through " on the next page for next steps.

OR

See "Authentication using an Identity certificate and Kerberos constrained delegation" on the next page for next steps.

ActiveSync with basic auth and pass through

If you select **ActiveSync with basic auth** for authentication, then Ivanti Standalone Sentry passes through the user name and password provided by the device user. No additional configuration on Ivanti Neurons for MDM is required.

Authentication using a group certificate and pass through

If you select **ActiveSync and/or AppTunel with certificates**, additional configuration fields display for **Authentication**.

For device authentication with group certificate, **Pass Through** is the only option available for server authentication.

To complete the configuration:

1. In the Ivanti Standalone Sentry profile with **ActiveSync and/or AppTunnel with certificates,** in **Global Settings**, for **Device Authentication Mode**, select **Use a single certificate for 2-factor auth (basic + cert)**.
2. Click **Upload Certificate**.

Additional fields are displayed.

3. Enter the **Certificate name** and **Password** for the certificate.
4. Upload the certificate (usually a .cer or PKCS 12 file) you trust.
5. Click **Save**.

Though certificate is uploaded, it does not persist until you click **Save**.

6. If you want to validate the certificates presented by the device against the Certificate Revocation List (CRL) published by the CA, then select **Check Certificate Revocation List (CRL)**.
   - CRL check should be enabled only if the certificate chain presented by the device or the Trusted-Front-End to Ivanti Standalone Sentry contains information to download CRL over HTTP.
   - Only HTTP- and HTTPS-based CRLs are supported. Some CAs create LDAP-based CRLs by default that will not work with Sentry.
   - For CRL validation to work, Sentry requires network connectivity to the CRL Distribution Point (CDP), usually the CA that issued the certificate, through an HTTP or HTTPS port.

7. **Server Authentication** defaults to **Pass Through** in any service, ActiveSync or AppTunnel, that you configure.
8. Click **Save**.

The Sentry restarts.

Authentication using a SCEP Identity certificate and pass through

If you select **ActiveSync and/or AppTunel with certificates**, additional configuration fields display for **Authentication**. This section describes the configuration when you choose **Use SCEP Identity** to authenticate the device to the Sentry and Sentry uses **Pass Through** for authenticating the device to the ActiveSync server or a backend resource.

Before you begin

You must have completed the steps described in "Configuring authentication using SCEP Identity (Ivanti Neurons for MDM only)" on page 62.

Procedure

1. In the Ivanti Standalone Sentry profile with **ActiveSync and/or AppTunnel with certificates**, in **Global Settings**, for **Device Authentication Mode**, select **SCEP Identity**.
2. Select the **App Identity Certificate Configuration** for the Identity certificate you want devices to use to authenticate with Sentry.
3. If you want to validate the certificates presented by the device against the Certificate Revocation List (CRL) published by the CA, then select **Check Certificate Revocation List (CRL)**.
   - CRL check should be enabled only if the certificate chain presented by the device or the Trusted-Front-End to Standalone Sentry contains information to download CRL over HTTP.
   - Only HTTP- and HTTPS-based CRLs are supported. Some CAs create LDAP-based CRLs by default that will not work with Sentry.
   - For CRL validation to work, Sentry requires network connectivity to the CRL Distribution Point (CDP), usually the CA that issued the certificate, through an HTTP or HTTPS port.
4. **Server Authentication** defaults to **Pass Through** in any service, ActiveSync or AppTunnel, that you configure.
5. Click **Save**.

The Sentry restarts.

Authentication using an Identity certificate and Kerberos constrained delegation

This section describes the configuration when you choose **Identity Certificate** to authenticate the device to the Sentry and **Kerberos** for how Sentry authenticates the device to the ActiveSync server or backend resource.

- For ActiveSync, Sentry supports Kerberos authentication only with Microsoft Exchange Servers.

- If you are configuring tunneling to a DFS server, in the Kerberos distribution center, map the SPN of the CIFS service domain to one of its domain controllers. See "Configuring Kerberos authentication for DFS" on page 82.

- Kerberos initialization in Standalone Sentry occurs only during tomcat start up. Ivanti Standalone Sentry obtains the ticket-granting ticket (TGT) during Kerberos initialization. If the initialization fails during tomcat start up, Ivanti Standalone Sentry automatically continues to retry until a service ticket from the KDC is received. The retry interval starts at one minute and maxes out at one-hour intervals. Failed initialization attempts are reported with a WARN level log in Ivanti Standalone Sentry System Manager in Monitoring. To manually initialize Kerberos, use the debug command, debug sentry kerberos init. The command has no impact if Kerberos initialization has already been completed.

Before you begin

You must have set up your Kerberos environment. See *Authentication Using Kerberos Constrained Delegation* on the Support site.

ℹ️   The *Authentication Using Kerberos Constrained Delegation document* has setup instruction on also Ivanti EPMM. Please ignore the setup instruction for Ivanti EPMM.

Procedure

1. In the Ivanti Standalone Sentry profile with **ActiveSync and/or AppTunnel with Kerberos**, in **Global Settings**, for **Device Authentication Mode**, select **Use SCEP Identity**.
2. Select the **App Identity Certificate Configuration** for the Identity certificate you want devices to use to authenticate with Sentry.
3. If you want to validate the certificates presented by the device against the Certificate Revocation List (CRL) published by the CA, then select **Check Certificate Revocation List (CRL)**.

    - CRL check should be enabled only if the certificate chain presented by the device or the Trusted-Front-End to Standalone Sentry contains information to download CRL over HTTP.
    - Only HTTP- and HTTPS-based CRLs are supported. Some CAs create LDAP-based CRLs by default that will not work with Sentry.
    - For CRL validation to work, Sentry requires network connectivity to the CRL Distribution Point (CDP), usually the CA that issued the certificate, through an HTTP or HTTPS port.
4. In the Kerberos section, add Realm and Key Distribution Center (KDC).
    - **Realm:**The Kerberos administrative domain with CIFS sahres. The realm is usually the company domain name, in all uppercase characters.

- **Key Distribution Center**: The Key Distribution Center is the network service that supplies session tickets and temporary session keys. This is generally the Active Directory domain controller host name.

5. Click **Next** to enter or update Sentry Server Configuration information.
6. Click **Next** to add or edit a service and configure server authentication.
7. For **Server Authentication** for a service, ActiveSync or AppTunnel that you configure, select **Kerberos**.

When you select **Kerberos** for **Server Authentication, Specific destinations (reverse proxy)** is selected by default. The **Derive Service Principal Name (SPN) from fully qualified Server Name** option appears.

If you select **All Destinations (forward proxy)**, the **Certificate Field Mapping** section also appears. Click **+ Add** to add an entry for certificate mapping.

Item	Description
Derive Service Principal Name (SPN) from fully qualified Server Name	Select to derive the Service Principal Name of the KCD from the FQDN of the ActiveSync server. When you select forward proxy, SPN is automatically derived from the FQDN of the ActiveSync server. Therefore, this option is not available when you select forward proxy.
Certificate Field Mapping	Specify the certificate fields that Standalone Sentry can use to derive users' UPN and Realm for Kerberos authentication. Select the field from which Ivanti Standalone Sentry will derive the User UPN or User DN. For WP8.1 devices, for User UPN, select either **DNS Name** or **RFC 822 Name**, and for User DN select **Certificate Subject**. For each corresponding field selected, select **User UPN** or **User DN**. This field is required in a cross-realm Kerberos environment.

8. If you selected reverse proxy, then you must also add the FQDN and port for the allowed servers.

This must be the FQDN of the backend resource. Wildcards are not supported. If you did not select **Derive Service Principal Name (SPN) from fully qualified Server Name**, then you must also add the SPN.

9. Click **Next,** then Click **Save**

Ivanti Standalone Sentry restarts when the profile is pushed to Ivanti Standalone Sentry.

Cross-realm Kerberos support

Support for cross-realm Kerberos on the Ivanti Standalone Sentry is enabled by default, and does not require any actions from the administrator.

Cross-realm S4U2Self is supported. Cross-realm S4U2Proxy is not supported.

Configuring Kerberos authentication for DFS

Authentication to DFS servers using Kerberos requires additional setup in the KDC and the Standalone Sentry system manager. To support Kerberos authentication for DFS, map the SPN of the CIFS service domain to one of its domain controllers (DC). If your Kerberos environment has multiple domain controllers (DC), to avoid authentication failure, add the DC you are mapping to as a static host in the Ivanti Standalone Sentry system manager.

If your Kerberos environment has multiple domain controllers (DC), note that you can only map the SPN of the CIFS service domain to one DC.

Before you begin

Setup Ivanti Standalone Sentry for authentication using Kerberos.See "Authentication using an Identity certificate and Kerberos constrained delegation" on page 79

Procedure
1. Map the SPN of the domain to one of its Domain Controllers (DC).
2. On the KDC, associate the Standalone Sentry service account to the CIFs service.
3. If the domain contains multiple DCs, add a static host for the DC in the Standalone Sentry system manager:

Sign in to the Ivanti Standalone Sentry system manager.

Go to **Settings > Static Hosts**.

Click **Add**.

Configure the following:

IP address: IP address of the DC.

FQDN: FQDN of the DC entered in Step 1.

Alias: short name of DC followed by space.

Example:

IP Address: 192.168.10.5

FQDN: win2k8.texas.enterprise.com

Alias: win2k8 texas.enterprise.com

Click **Save.**

Related topics

- "Static Hosts" on page 96.

Working with connections through Ivanti Standalone Sentry

Ivanti Standalone Sentry connections on Ivanti Neurons for MDM

Devices connecting to backend resources through Ivanti Standalone Sentry can be managed (registered with Ivanti Neurons for MDM) or unmanaged (not registered with Ivanti Neurons for MDM).

A record of users attempting to connect from managed devices is provided in **Devices > Devices**. A record of the users attempting to connect from unmanaged devices is provided in **Devices > Unmanaged Connections**.

Related topics

- "Connections from managed devices" below.

- "Unmanaged connections" on page 87.

Connections from managed devices

Typically, devices register with an Ivanti Neurons for MDM, which provides the devices with the configurations that allow users on the devices to access backend resources.

To view the users on managed devices that connect to your ActiveSync server or any other backend resource through Standalone Sentry, in Ivanti Neurons for MDM, go to **Devices**, click on a device name, then click on the **Sentry** tab. Information displayed on this page is updated when Standalone Sentry syncs with Ivanti Neurons for MDM.

Each unique combination of user (email account) on the device and app connecting to the backend resource displays as a separate record in the in the **Sentry** tab. If there are multiple accounts on the device, then each User is listed. If multiple apps are being used, then each app (User-Agent) is displayed as a separate record.

- "Information displayed in each record for a connection from a managed device" below.

- "Taking action on a record for a connection from a managed device" below.

Information displayed in each record for a connection from a managed device

The following table describes the information displayed for each record in the Ivanti Neurons for MDM in **Devices > Sentry**.

TABLE 10. INFORMATION DISPLAYED FOR STANDALONE SENTRY CONNECTIONS

Column	Description
User	The device user.
Sentry Hostname	The Standalone Sentry hostname through which the device is connecting with the ActiveSync server or the backend resource.
User-Agent	The app connecting to the backend resource. This could be the email app or any other app connecting to the ActiveSync server or backend resource.
Version	The app version.
Bundle Id	Bundle Id of the app connecting to the backend resource.
Status	Indicates whether the device is **Allow** or **Block**. The status for a connection is blocked if the device is out of compliance.
Actions	Administrators can **Allow, Block,** or **Delete** a record.

Taking action on a record for a connection from a managed device

Actions applied on a Ivanti Standalone Sentry connection record only impacts the user associated with the app in that record. The user using another app or the same user on another device is not impacted. The only action you can take on a record is delete.

Deleting a record removes the record from the list in **Devices > Sentry**. However, if the device attempts to connect to the backend resource

Procedure

1. In Ivanti Neurons for MDM, go to **Devices > Devices.**
2. Click the name for a record.
3. Click on the **Sentry** tab.
4. Click **Delete** for the record you want to delete.

Unmanaged connections

Sometimes users can access backend resources through Ivanti Standalone Sentry from devices that are not managed by Ivanti Neurons for MDM. In these cases these connections are displayed in **Devices > Unmanaged Connections.**

Related topics

- "Information displayed in each record for a connection from an unmanaged device" below.

- "Taking action on a record for an unmanaged connection " on the next page.

Information displayed in each record for a connection from an unmanaged device

The following table describes the information displayed for each record in the Ivanti Neurons for MDM in **Devices > Unmanaged Connections**.

TABLE 11. INFORMATION DISPLAYED FOR CONNECTIONS FROM UNMANAGED DEVICES

Column	Description
User	The device user.
Connection Source	The device type from which the connection is made.
ActiveSync Id	ActiveSync Id of user connecting to the backend resource. Only connections to the ActiveSync server can be from unmanaged devices.
Override Action	Displays the **Action** taken on the unmanaged connection by the administrator. The actions displayed are **Allow** or **Block.**
Status	Indicates whether the device is **Allow** or **Block**. The status for a connection is **Allow** if the **Allow unmanaged devices to receive email and data** setting is enabled or the connection was previously allowed by the administrator, and **Block** if it is disabled.
Created At	The date and time the connection was created.
Sentry Hostnames	The Ivanti Standalone Sentry hostname through which the device is connecting with the ActiveSync server or the backend resource.

Taking action on a record for an unmanaged connection

Actions applied on an Ivanti Standalone Sentry connection record only impacts the user associated with the device in that record. The user using another app or the same user on another device is not impacted. You can take the following actions on connections from unmanaged devices:

- **Allow**: Apply the action on a record that was manually blocked or blocked by configuration. Allows the user on the device in the record to access the ActiveSync server.

  - **Block**: Blocks the user on the device in the record from accessing the ActiveSync server.

  - **Delete**: Deleting a record removes the record from the list. However, if the device attempts to connect to the backend resource it is allowed or blocked depending on whether the **Allow unmanaged devices to receive email and data** setting is enabled or disabled.

Procedure
1. In Ivanti Neurons for MDM, go to **Devices > Unmanaged Connections.**
2. Select a record on which you want to take action.
3. Click **Action** and select the action to apply.

Ivanti Standalone Sentry Settings

The following describe the Ivanti Standalone Sentry settings in the Ivanti Standalone Sentry System Manager:

Overview of Ivanti Standalone Sentry settings

The settings tab in the Ivanti Standalone Sentry System Manager contains links for configuring Ivanti Standalone Sentry. To log in to the Sentry System Manager, go to https://*fully_qualified_hostname*:8443.

TABLE 12. CONFIGURATION LINKS IN THE SETTINGS TAB

Setting	Description
Network: Interfaces	Change physical interface settings. Add VLAN interfaces. Change VLAN interfaces.
Network: Routes	Change the default gateway. Route through different gateways.
DNS and Hostname	Change DNS servers.
Static Hosts	Edit the host list for the Standalone Sentry.
Date and Time (NTP)	Change the time source used by the Standalone Sentry.
CLI	Change the Enable Secret set during installation. Enable or Disable ssh access. Change ssh settings.
Splunk	Configure Splunk server.
Syslog	Configure Syslog servers.
SNMP	Configure SNMP servers.
Email Settings	Configure SMTP servers.
Services	Enable or Disable Sentry services.
Services: Sentry	Allow or block New device access when UEM is unreachable.
Services: Sentry: Cipher Suites & Protocols	Customize Cipher Suites and Protocols settings.

Interfaces

The **Settings > Interfaces** screen enables you to change parameters for the network interface points for Ivanti Standalone Sentry:

- physical and VLAN interfaces

- static routes

> ℹ️ Physical and VLAN interface fields are not editable for a Ivanti Standalone Sentry installed on Microsoft Azure. These are assigned by the Microsoft Azure infrastructure.

You configure a physical network interface as part of the installation process. You can use the Interfaces screen to:

- • Edit the physical interface settings specified during installation

- • Add physical interfaces

- • Add VLAN interfaces

- • Change VLAN interfaces

Physical interface mapping to M2600 NIC ports

The following table provides a mapping of the physical interface name in the Ivanti EPMM System Manager to the physical NIC port in the M2600 appliance. The six Gigabit Ethernet interfaces are available only on an M2600 appliance.

TABLE 13. PHYSICAL INTERFACE MAPPING TO M2600 NIC PORTS

Physical interface	M2600 NIC port
GigabitEthernet1	I - eth0 (NIC-3)
GigabitEthernet2	J - eth1 (NIC-4)
GigabitEthernet3	K- eth2 (NIC-5)
GigabitEthernet4	L- eth3 (NIC-6)
GigabitEthernet5	C- eth4 (NIC-1)
GigabitEthernet6	D- eth5 (NIC-2)

Changing physical interfaces

You can change the physical interfaces for Ivanti Standalone Sentry in the Sentry System Manager.

Procedure
1. Click the interface name.

FIGURE 1. MODIFY PHYSICAL INTERFACES

2. Change any or all of the following fields:

Field	Description
IP	Enter the IP address of the physical network interface. Unless you are configuring a standalone implementation for a small trial, you should specify at least one physical interface.
Mask	Enter the netmask of the physical network interface.
ACL Name	Select an Access Control List for this interface.
Admin State	To enable this interface for use with the system, click Enable. To temporarily prevent use of this interface with the system, click Disable.

3. Click **Save**.

Adding VLAN interfaces

Virtual Local Area Network (VLAN) interfaces are optional interfaces you can configure on UEM to manage bandwidth and load balancing. You can add a VLAN interface in the Ivanti Standalone Sentry System Manager.

Procedure
1. Click **Add VLAN**.

FIGURE 2. ADDING VLAN

2. Use the following guidelines to complete the configuration:

Field	Description
VLAN ID	Specify a number between 2 and 4094.
IP Address	Enter the IP address for this VLAN interface.
Mask	Enter the netmask for this VLAN interface.
Physical Interface	Select the physical interface that corresponds to this VLAN interface.
ACL Name	Select an Access Control List for this interface. See "Access Control Lists" on page 139.
Admin State	To enable this interface, click Enable. To temporarily suspend use of this VLAN, click Disable.

3. Click **Save**.

Deleting a VLAN interface

You can delete the Ivanti Standalone Sentry Virtual Local Area Network (VLAN) interface in the Sentry System Manager.

Procedure

1. Select the VLAN you want to remove.
2. Click **Delete VLAN**.

Routes

The **Settings > Network > Routes** screen enables you to create and maintain static network routes within the enterprise.

FIGURE 1. STANDALONE SENTRY ROUTES
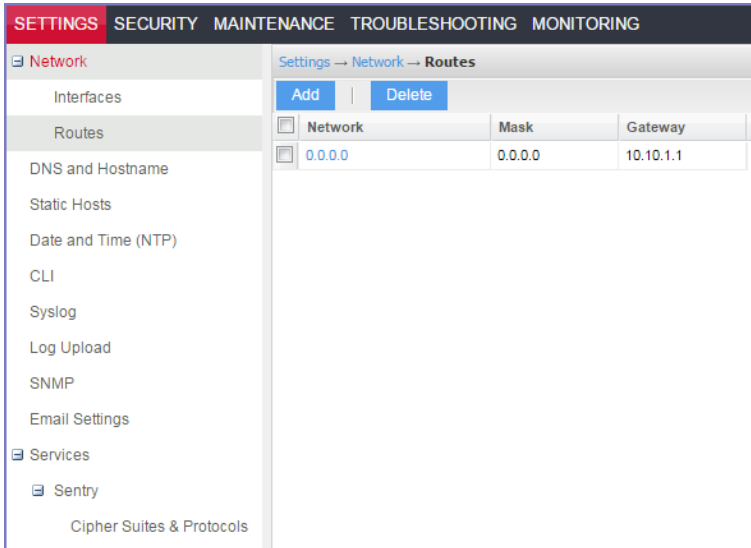
Adding network routes

You can add network routes in the Ivanti Standalone Sentry System manager in **Settings > Network > Routes**.

Procedure

1. In the Ivanti Standalone Sentry System Manager go to **Settings > Network > Routes.**
2. Click **Add**.

FIGURE 2. ADDING NETWORK ROUTES

3. Use the following guidelines to complete the fields:

Field	Description
Network	Enter the network IP address.
Mask	Enter the subnet mask.
Gateway	Enter the IP address for the gateway.

4.   Click **Save**.

Deleting network routes

You can delete the network routes in the Ivanti Standalone Sentry System manager.

Procedure
1.   In the Ivanti Standalone Sentry System Manager go to **Settings > Network > Routes.**
2.   Select the entry.
3.   Click **Delete**.

DNS and Hostname

The DNS and Hostname screen displays the hostname, default domain, and DNS information entered during installation. Use this screen to:

- Change the hostname

- Change the default domain

- Change or add DNS servers

> For an Ivanti Standalone Sentry installed on Microsoft Azure and AWS, the DNS and hostname fields are automatically assigned by the infrastructure and those fields are not editable.

FIGURE 1. DNS AND HOSTNAME
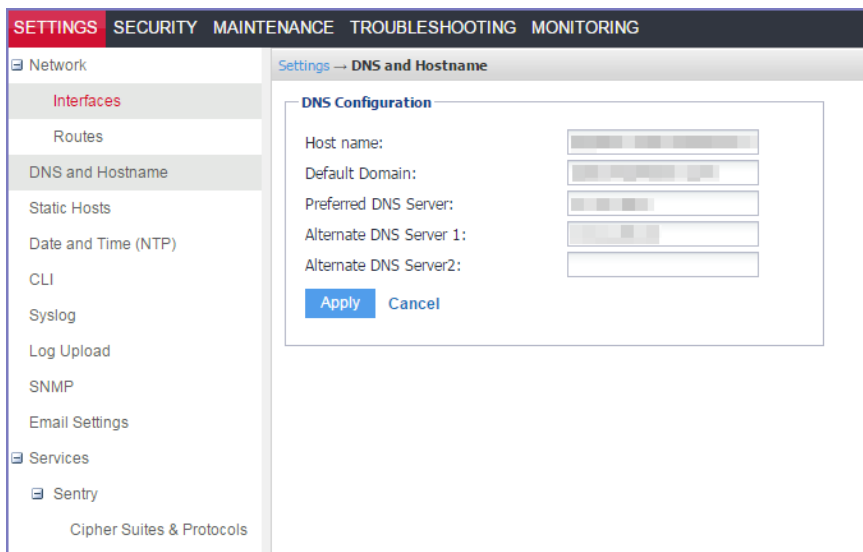
The following table describes the fields for DNS and hostname.

TABLE 14. DNS AND HOSTNAME FIELD DESCRIPTION

Field	Description
Host name	Specify the fully-qualified host name for the appliance.
Default Domain	Specify the default domain for the appliance.
Preferred DNS Server	Specify the IP address of the primary DNS server to use.
Alternate DNS Server 1	Specify the IP address of an optional alternate DNS server.
Alternate DNS Server 2	Specify the IP address of an optional alternate DNS server.

Static Hosts

The Static Hosts page enables you to edit the hosts file for the Ivanti Standalone Sentry appliance.

FIGURE 1. STATIC HOSTS
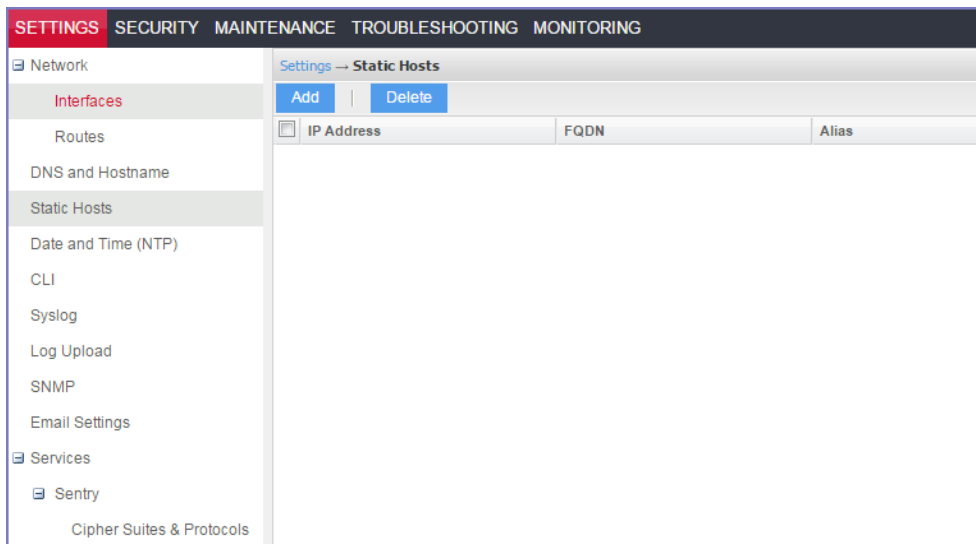
Adding hosts

You can add an entry to the hosts file on the Ivanti Standalone Sentry appliance in the Sentry System Manager.

Procedure

1. In the Ivanti Standalone Sentry System Manager go to **Settings > Static Hosts.**

2. Click the **Add** button.

FIGURE 2. ADDING STATIC HOSTS

3. Use the following guidelines to complete the displayed fields:

Field	Description
IP Address	The IP address for the host you are adding.
FQDN	The fully-qualified domain name for this host, as in myserver.mycompany.com.
Alias	The alias for this host. Up to 29 aliases are allowed.

No more than 799 characters are allowed for each host entry. Each host entry includes the IP address, FQDN, and the alias.

4. Click **Save**.

Editing hosts

To edit an entry in the hosts file on the Sentry appliance, click the IP address for the host displayed in the **Static Hosts** screen in the Ivanti Standalone Sentry System Manager in **Settings > Static Hosts**.

Deleting hosts

To delete an entry in the hosts file on the Sentry appliance, select the IP address for the host displayed in the **Static Hosts** screen in the Ivanti Standalone Sentry System Manager in **Settings > Static Hosts**.
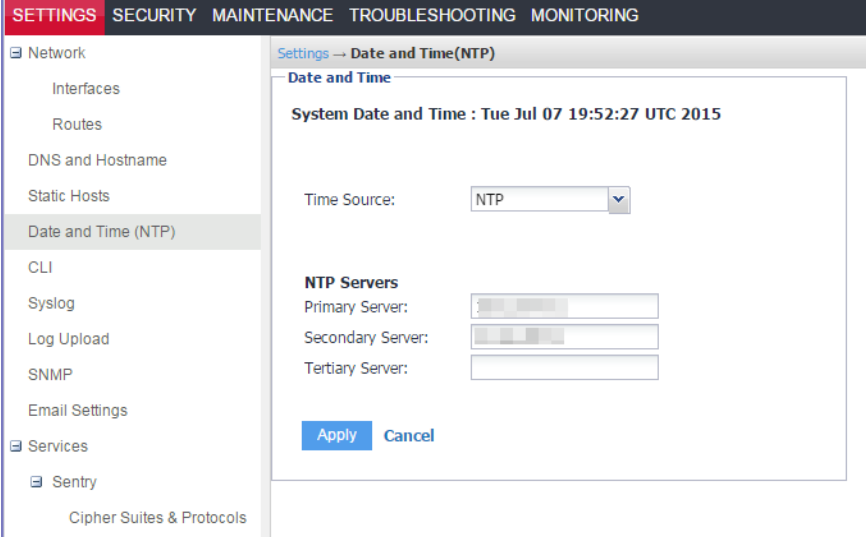
Procedure

1. In the **Static Hosts** screen, select the host to be deleted.
2. Click **Delete**.

Date and Time (NTP)

The Date and Time screen displays any NTP information specified during installation. This is an optional portion of the configuration, but is highly recommended due to the effect of database timestamps on the behavior of the system, as well as on the quality of reporting. Currently, only UTC time display is supported. If you choose to use a local time source, instead, then you can specify the date in this screen.

FIGURE 1. DATE AND TIME (NTP)

The following table describes the fields for setting the system date and time.

TABLE 15. DATE AND TIME (NTP)

Field	Description
Time Source	Select NTP if you intend to specify one or more NTP servers. Select Local if you intend to set the system time for the Server.
If you select NTP	
Primary Server	Specify the IP address or fully-qualified host name for the NTP server to use.
Secondary Server	Specify the IP address or fully-qualified host name for the first failover NTP server to use.
Tertiary Server	Specify the IP address or fully-qualified host name for the second failover NTP server to use.
If you select Local	
Date	Enter the current date.
Time	Enter the current time.

CLI

The CLI screen displays the command line interface access settings specified during configuration. Use this screen to alter these settings.

For information about using the CLI, see "Command Line Interface".

The following table describes the CLI settings.

TABLE 16. CLI FIELD DESCRIPTIONS

Field	Description
Enable Secret	Click the **Change Enable Secret** link to specify the password required to access important functions in the CLI.
Confirm Enable Secret	Re-enter the specified password to confirm. This field displays only if you click the **Change Enable Secret** link.
CLI Session Timeout	Specify the duration of inactivity on the SSH connection that should cause the session to time out.
SSH	Select **Enable** if you want to allow SSH access to the Administration tool.
Max SSH Sessions	Specify the maximum number of simultaneous SSH sessions to allow.

Splunk

You can configure a Splunk entry Ivanti Standalone Sentry so that Standalone Sentry periodically sends Sentry health and audit log data to the Splunk Enterprise server set up on your network. Logs are forwarded to the Splunk receiver and to the local log location.

Overview of the steps for setting up Splunk on Ivanti Standalone Sentry

Following is an overview of the steps for setting up Splunk on Standalone Sentry:

1. "Enabling the Splunk forwarder service in Ivanti Standalone Sentry" below.
2. "Adding a Splunk receiver entry in Ivanti Standalone Sentry" below.
3. "Configuring Ivanti Standalone Sentry data to export to Splunk" on the next page.
4. "Tasks in Splunk server to set up Ivanti Standalone Sentry" on the next page

Enabling the Splunk forwarder service in Ivanti Standalone Sentry

Enable the Splunk forwarder service so that it can push data to the Splunk receiver.

> **i**   The Splunk forwarder service can also be enabled using CLI.

Procedure

1. In Ivanti Standalone Sentry System Manager, go to **Settings > Services**.
2. For **Splunk Forwarder,** select **Enable**.
3. Click **Apply > OK** to save the changes.

The status for Splunk Forwarder displays as **Running.**

Next steps

Go to "Adding a Splunk receiver entry in Ivanti Standalone Sentry" below.

Adding a Splunk receiver entry in Ivanti Standalone Sentry

You add the Splunk receiver in the Standalone Sentry System Manager in **Settings > Splunk**.

Procedure

1. In Ivanti Standalone Sentry System Manager, go to **Settings > Splunk**.
2. Click **Add** to open the **Add Splunk Receiver** window.
3. Configure the fields.

Fields	Description
Splunk Receiver	Add the IP address or the hostname of your Splunk Enterprise Server.
Port	Add the port of your Splunk Enterprise Server.
Enable SSL	(Optional) Click the check box to enable SSL.

4. Click **Apply > OK** to save the changes.

Next steps

Go to "Configuring Ivanti Standalone Sentry data to export to Splunk" below.

Configuring Ivanti Standalone Sentry data to export to Splunk

Use the Ivanti Standalone Sentry command line interface (CLI) to configure the data to export to Splunk.

Procedure

1. SSH to Ivanti Standalone Sentry.
2. In configuration mode, enter `sentry audit` to enable miauditlogs log data for export.
3. In configuration mode, enter `sentry health-monitor` to enable mihealth log data for export.
4. Enter end to exit configuration mode.

Next steps

Go to "Tasks in Splunk server to set up Ivanti Standalone Sentry" below.

Related topics

See "Log representation and format" on page 124 for the representation and the format of the data captured in audit and health logs.

Tasks in Splunk server to set up Ivanti Standalone Sentry

Do the following on the Splunk server:

1. Ensure that Splunk listener is on the same port as the one configured in the Splunk entry in Standalone Sentry.
2. Enable the miauditlog and mihealth indexes, which are sentry_miaudit and sentry_mihealth respectively.

Syslog

You can send Sentry syslog data to a remote log server you have set up on your network. Logs are then written to both the syslog location and the local log location.

Adding a syslog entry

You add the syslog server in the Ivanti Standalone Sentry System Manager in **Settings > Syslog**.

Procedure
1. In Sentry System Manager, go to **Settings > Syslog.**
2. Click **Add**.
3. Enter the requested information.
4. Click **Apply.**

Related topics

See the field descriptions for a Syslog entry in "Field descriptions for a syslog entry" below.

Editing a syslog server entry

Settings > Syslog lists the syslog server you have configured on Sentry. You can edit the syslog server setting you have configured.

Procedure
1. In Sentry System Manager, go to **Settings > Syslog.**
2. Click on the IP address or hostname of the syslog server you want to edit.
3. Update the settings as needed.

You cannot update the server address or hostname.
4. Click **Apply** to save and apply the changes.

Related topics

See the field descriptions for a Syslog entry in "Field descriptions for a syslog entry" below.

Field descriptions for a syslog entry

The following table describes the settings for syslog.

TABLE 17. SYSLOG FIELDS DESCRIPTION

Field	Description
Server	Enter the IP address or host name for the remote log server.
Port	The default is port 514. Monitor listens on port 514. If you are using Monitor, use the default port 514 for both TCP and UDP.
Protocol	Select UDP or TCP. Select UDP or TCP, depending on whether your syslog server is set up to receive UDP or TCP data.
Facility Type	Select the appropriate facility type to select the logs to report to syslog server. **General**: Select to send mi.log and miservicewatch.log data. The mi.log file contains sentry.log and mics.log data. The miservicewatch.log contains data from **Troubleshoot > Service Diagnosis** in the Sentry System Manager. **Audit**: Select to send audit logs. **Health Monitor**: Select to send health monitoring logs.
Log Level	Select a log level from the drop down list. The log level is listed based on the priority and severity of the log message. **Emergency** **Alert** **Critical** **Error** **Warning** **Notice** **Info** **Debug** Emergency has the highest priority and Debug the lowest priority. All log messages at that log level and higher priority are included in the log file. For Facility type **Audit** and **Health Monitor**, **Info** is the only log level available. ⓘ If the log level configured for the syslog server is higher than the log level configured on Sentry, Sentry only sends Alert/Error/Warning messages to the syslog server.
Admin State	Select **Enable** from the dropdown list to apply these settings to your current configuration. Select **Disable** to suspend use of the configured log server.

Adding Monitor as a syslog server

Monitor allows IT and system administrators to monitor the health of all their mission-critical UEM components and services. Monitor organizes and displays monitoring data pushed from Sentry, providing you a comprehensive view of system status and alerts.

Before you begin

- You must have set up Monitor. For information on how to set up Monitor, see the *Monitor Configuration Guide.*

Procedure:
1. In Sentry System Manager, go to **Settings > Syslog**.
2. Click **Add**.
3. Provide values for the Add Syslog dialog fields:

Server: Enter the IP address of Monitor.

Port: Use the default port 514.

Protocol: Select the desired protocol.

Facility Type: Select General.

Only General is supported.

Log Level: Select the desired log level.

Admin State: Select **Enable**.
4. Click **Apply**.

SNMP

Simple Network Management Protocol (SNMP) is a protocol used for network management for collecting information about network entities, such as servers and devices, on an Internet Protocol (IP) network. Various third-party SNMP systems are available that provide SNMP-based management and tools.

Sentry provides the following SNMP capabilities:

- Link up and down traps

Sentry sends these two SNMP traps (events) to a specified SNMP trap receiver using the SNMP v2c protocol.

- An SNMP server can request information from Sentry related to these MIBs:

- The HOST-RESOURCES_MIB
- disk I/O (UCD-DISKIO-MIB)

- Support for SNMP v2c and v3 protocols to pull MIB information from Sentry to the SNMP server.
It is recommended to use v3 protocol if you transmit information across unsecured links.

Use the SNMP screen to manage SNMP trap receivers.

Configuring SNMP on Sentry

The following provides the general workflow to configure SNMP:

1. "Configuring the SNMP trap receiver server" below to which Sentry sends SNMP traps.

2. "Enabling the SNMP service with the v3 protocol" on page 107 from whom Sentry accepts requests.

3. "Enabling the SNMP service with the v2c protocol" on page 110 between Sentry and your SNMP server.

Configuring the SNMP trap receiver server

Configure the server to which Sentry sends SNMP traps. This server can also get MIB information from Sentry.

Procedure

1. Log into Ivanti Standalone Sentry.

2. Go to **Settings > SNMP** to open the SNMP details pane.

3. Click **Add** to open the **Add SNMP Trap Receiver** window.

4. Edit the fields, as necessary.
Refer to the **Add SNMP Trap Receiver** table for details.

5. Click **Apply > OK** to save the changes.

Add SNMP trap receiver field description

The following table summarizes fields and descriptions in the **Add SNMP Trap Receiver** window:

TABLE 18. FIELDS AND DESCRIPTIONS IN THE ADD SNMP TRAP RECEIVER WINDOW

Fields	Description
Server	Enter the IP address or server name for your SNMP trap receiver. For example: trapreceiver.myCompanyDomain.com
Port	By default, port 162 is configured. Edit this field if you are using a different port.
Community	Enter the string which names the SNMP community on your SNMP trap receiver.
Version	Sentry sends SNMP traps using SNMP protocol v2c.
Admin State	Select **Enable** to enable the SNMP service for this SNMP server.

Editing a trap receiver

To edit an SNMP trap receiver, navigate to **Settings > SNMP** in the Ivanti Standalone Sentry System Manager.

Procedure

1. In the Ivanti Standalone Sentry System Manager, navigate to **Settings > SNMP.**

2. In the SNMP screen, select the link for the trap receiver you want to edit:

3. Make your changes.

4. Click **Save**.

Deleting SNMP trap receiver servers

To delete one or more SNMP trap receiver, navigate to **Settings > SNMP** in the Ivanti Standalone Sentry System Manager.

Procedure

1. In the Ivanti Standalone Sentry System Manager, navigate to **Settings > SNMP** to open the SNMP details pane.

2. Select one or more of the servers you want to delete.

Click the box next to **Server** to select all servers in the list.

3. Click **Delete > Yes**.

Enabling the SNMP service with the v3 protocol

Set up the SNMP v3 user from whom Ivanti Standalone Sentry accepts requests. In addition, you can enable or disable sending traps to any configured SNMP trap receiver.

Procedure

1. Log into Ivanti Standalone Sentry System Manager.

2. Go to **Settings > SNMP** to open the SNMP details pane.

3. In the **SNMP Control** section**,** select **Enable** for **SNMP Service** to enable the SNMP service on Standalone Sentry.

4. Go to the **Protocol** option and verify that **v3** is selected.
    The v3 option is selected, by default.

FIGURE 1. SNMP SERVICE

5. Click **Add** to open the **Add SNMP v3 User** window.

6. Enter the SNMP v3 user fields, as necessary.

7. Click **Save** to add this user to the **SNMP v3 Users** table.

8. Go to **Link Up/Down Trap**.

9. Select **Enable**.
    Select **Disable** to stop Sentry from sending SNMP traps to any SNMP trap receiver.

10. Click **Apply > OK** to save the changes.

Related topics

Refer to the "SNMP" on page 104 for details.

SNMP v3 User field description

The following table describes the SNMP v3 user fields.

TABLE 19. SNMP v3 USER FIELD DESCRIPTION

Fields	Description
User Name	Enter the username without any spaces (example: miuser).
Security Level	Select a security level for authentication. The options are: • **noAuthNoPriv**: Without Authentication or Privacy. • **authNoPriv**: With Authentication and without Privacy • **authPriv**: With Authentication and with Privacy
Auth Protocol	Select an authentication protocol. This can be selected only if the **Security Level** is selected as **authNoPriv** or **authPriv**.
Auth Password	Enter a password for authentication. The password must contain at least characters.
Privacy Protocol	Select a privacy protocol. This can be selected only if **Security Level** is selected as **authPriv**.
Privacy Password	Enter a privacy password with minimum of 8 characters.

Deleting SNMP v3 users

The following describes how to delete one or more SNMP v3 users.

Procedure

1. Log into Sentry.
2. Go to **Settings > SNMP** to open the SNMP details pane.
3. Go to the **SNMP Control** group.
4. Select one or more of the users you want to delete.

Click the box next to **User Name** to select all users in the list.

5. Click **Delete > Yes**.

Enabling the SNMP service with the v2c protocol

Set up the SNMP v2c communication between Sentry and your SNMP server. You also enable or disable sending traps to any configured SNMP trap receiver.

Procedure

1. Log into Sentry.

2. Go to **Settings > SNMP** to open the SNMP details pane.

3. Go to the **SNMP Control section > SNMP Service**.

4. Select **Enable** to enable the SNMP service on Standalone Sentry.

5. Go to the **Protocol** option and select **v2c**.

6. Change the value of **Read Only Community** if necessary.
   The standard SNMP community string name is **public**. This is the community string that the SNMP server uses to pull MIB information from Sentry.

7. Go to the **Link Up/Down Trap** option and select **Enable**.
   Select **Disable** to stop Sentry from sending SNMP traps to any SNMP trap receiver.

8. Click **Apply > OK** to save the changes.

Editing the Read Only Community string

The community string is available for v2 protocol. The default community string for the SNMP is set to public. To change this string, navigate to **Settings > SNMP** in the Standalone Sentry System Manager.

Procedure
1. In the Standalone Sentry System Manager, navigate to **Settings > SNMP.**
2. Edit the default string.
3. Click **Apply**.

Email Settings

Use the **Email Settings** page to configure the SMTP server. This configuration is required for Sentry monitoring alert notifications by email.
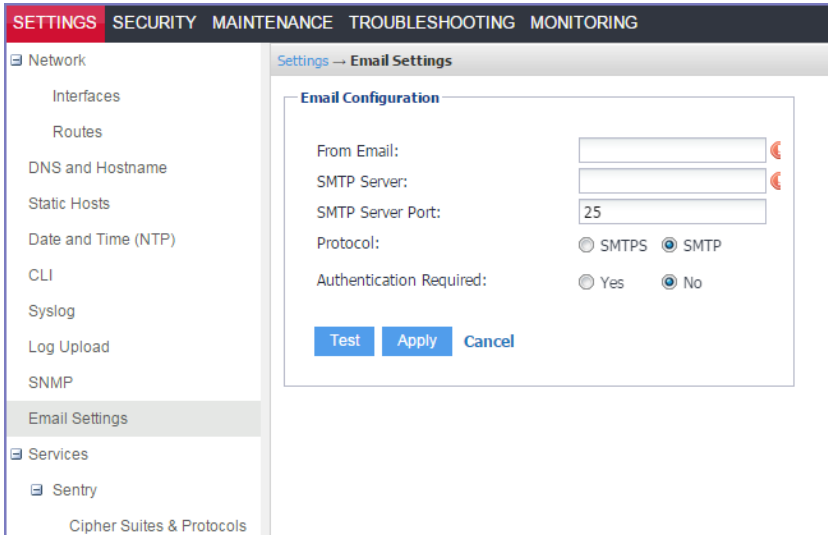
Configuring the SMTP server information for Ivanti Standalone Sentry notifications

Configure the SMTP server information required for Sentry alert notifications.

Procedure
1. In the Sentry System Manager, go to **Settings**.
2. Click **Email Settings** in the left navigation pane.
3. Enter the requested information.
4. Click **Test**.
5. Enter an email address and body for the test email.
6. Click **OK**.
7. Confirm that the email arrives as expected.
8. Click **Save**.

Related topics

For a description of the fields for configuring SMPT, see "Field descriptions for SMTP settings" below.

Field descriptions for SMTP settings

The following table describes the fields for configuring SMTP settings.

TABLE 20. FIELD DESCRIPTIONS FOR SMTP SETTINGS

Field	Description
From Email	Specify the email address to use in the From field for all administrative email notifications.
SMTP Server	Specify the IP address or fully-qualified host name for the SMTP server the Ivanti Server will use.
SMTP Server Port	Specify the port configured for the SMTP server.
Protocol	If th SMTP server you are configuring is a secured server, that is, it uses the SMTPS protocol, then select the SMTPS button. Otherwise, leave SMTP selected.
Authentication Required	Specify whether this SMTP server requires authentication. In most cases, this field will be set to Yes.
User Name	If you select Yes for Authentication Required, then this field displays. Enter the user name required for SMTP authentication.
Password	If you select Yes for Authentication Required, then this field displays. Enter the password required for SMTP authentication.
Confirm Password	If you select Yes for Authentication Required, then this field displays. Confirm the password required for SMTP authentication.

Login

Use the **Login** page to add disclaimers. This banner appears on the log in page for Sentry.

The banner does not appear on the Sentry log in page until the disclaimer is updated in the **Sentry Settings**.

Procedure

1. In the Sentry System Manager, go to **Settings**.
2. Click **Login** in the left navigation pane.
3. Enter the text to display in the Login Banner.

4. Click **Apply**.
5. Click **Yes** to confirm the configuration.

Timeout

Use the **Timeout** page to configure the idle session timeout for Sentry System Manager. You can set the timeout for up to 90 minutes.

> ⓘ Idle Session Timeout will be applicable for the next sessions only.

Procedure

1. In the Sentry System Manager, go to **Settings**.
2. Click **Timeout** in the left navigation pane.
3. Select the Idle Session Timeout for the System Manager portal.
4. Click **Apply**.

5. Click **Ok**.

Services

Use the **Services** screen to enable or disable the Sentry service. Select **Enable** or **Disable,** then click **Apply**. The status displays to the right of the setting. **Running** indicates that the Sentry service is enabled. **Not Running** indicates that the Sentry service is disabled.

FIGURE 1. ENABLE OR DISABLE SENTRY SERVICE
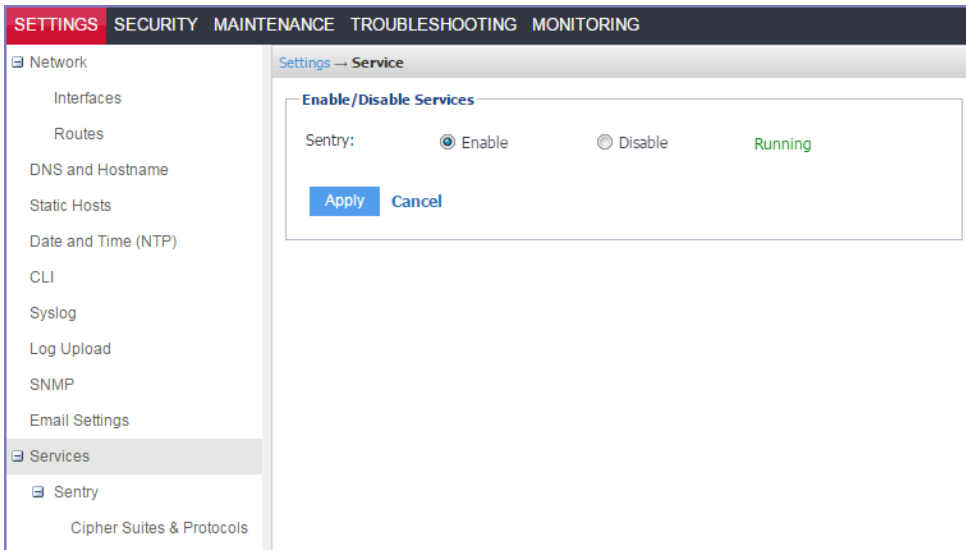
Sentry

Use the **Settings > Services > Sentry** screen to change the default setting for whether new devices are allowed or not allowed to access the ActiveSync server or backend resource when UEM is inaccessible.
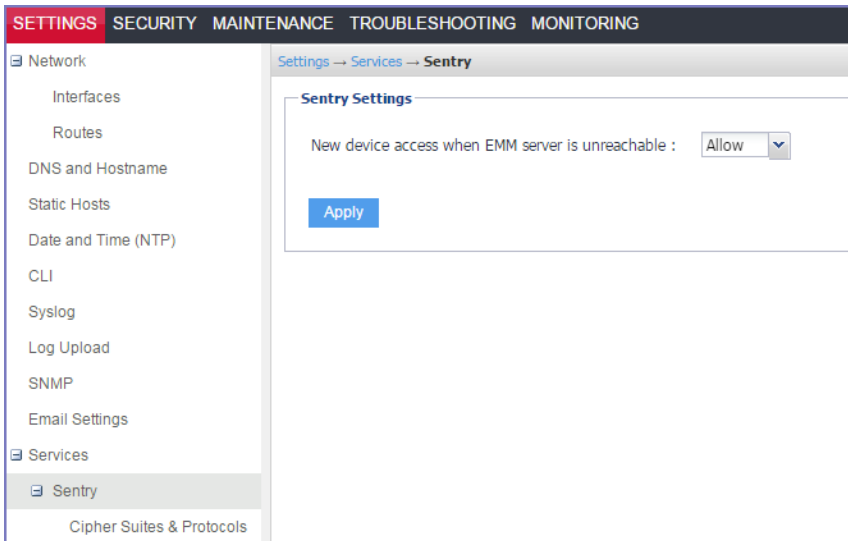
New device access

Use the **New device access when EMM server is unreachable** setting to allow or block new devices or devices not in the Ivanti Standalone Sentry cache, access to the ActiveSync server or backend resource if UEM is not reachable.

By default Ivanti Standalone Sentry allows new devices access to the server if the UEM is not reachable.

> Changing the **New device access when UEM server is unreachable** setting does not restart Ivanti Standalone Sentry.

To block new devices from accessing the server when UEM is unreachable, select **Block** from the drop down list and click **Apply**.

Incoming SSL configuration

For Ivanti Neurons for MDM, cipher suites and protocols configuration is done in the Sentry profile on Ivanti Neurons for MDM.

Outgoing SSL configuration

For Ivanti Neurons for MDM, cipher suites and protocols configuration is done in the Sentry profile on Ivanti Neurons for MDM.

The **Outgoing SSL Configuration** page Allows administrators the flexibility to configure Standalone Sentry to use cipher suites and protocols to match the security and system needs of your enterprise.

UEM SSL Configuration

Use the **UEM SSL Configuration** page to configure the client role parameters for communication from Sentry to UEM. You can configure ciphers and protocols for outgoing traffic from Sentry to UEM.

- "Enabling Strict TLS" below settings for Standalone Sentry SSL connections to UEM.

- "Enabling Server Name Indication (SNI)" on the next page.

- View the **Available** and **Selected** protocols and cipher suites. See "Cipher Suites and Protocols" on page 119.

- Set up custom protocol and cipher suite configuration. See "Cipher Suites and Protocols" on page 119.

The **EMM SSL Configuration page** allows the administrator the flexibility to configure Ivanti Standalone Sentry to use cipher suites and protocols to match the security and system needs of your enterprise.

> When mutual authentication is enabled between Ivanti EPMM and Sentry, then that Sentry is enabled with Strict TLS Configuration.

Enabling Strict TLS

You can enable strict TLS for outgoing traffic from Ivanti Standalone Sentry to UEM. Strict TLS is not enabled by default for the UEM server. However, it is enabled for Ivanti Neurons for MDM. When you enable strict TLS, the Java Trust Store is enabled by default. You can also use the custom trust store option to upload additional certificates that Ivanti Standalone Sentry must use.

Procedure
1. In the Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **EMM SSL Configuration**.
2. In the **Strict TLS Settings** section, check **Enable Strict TLS**.

Additional options are now available.

Item	Description
Enable Default Java Trust Store	Selected by default if strict TLS is enabled. Certificates and Certificates Authorities in the Java Trust Store are used to trust the SSL connection to UEM.
Allow and Log untrusted servers	Select to allow Ivanti Standalone Sentry to connect to UEM that does not use a trusted certificate in Java or custom trust store.
Enable Custom Trust Store	Select to upload certificates to the Ivanti Standalone Sentry trust store. Ivanti Standalone Sentry uses the certificates in the custom store to trust UEM. Generally used if UEM uses self-signed certificates.

3. Click **Apply**.
4. Click **Yes**.

The new TLS settings are applied and Ivanti Standalone Sentry restarts. It may take up to one minute for Ivanti Standalone Sentry to restart. Traffic is disrupted till Standalone is up and running again.

5. Click **OK**.

Enabling Server Name Indication (SNI)

Server Name Indication (SNI) is an extension to TLS. SNI allows multiple hostnames to be served over HTTPS from one IP address. By default, SNI is disabled on Ivanti Standalone Sentry for outgoing connections for the UEM server. However, SNI is enabled (read-only) for Ivanti Neurons for MDM UEM server. SNI allows a load balancer to direct incoming traffic to the correct UEM server based on the hostname provided by the client, in this case, Standalone Sentry. Some UEM servers may require that SNI is enabled in the client. Your Active Directory Federation Services (ADFS) may require SNI for all client communications.

> ℹ️ If SNI is enabled for EMM SSL connections, in some cases health check may fail if the backend server does not also support SNI. The workaround is to disable health check for the impacted server.

Procedure
1. In Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **EMM SSL Configuration**.
2. Click **Enable SNI**.
3. Click **Apply**.

Cipher Suites and Protocols

Ivanti Standalone Sentry includes a set of cipher suites and protocols. A default set of cipher suites and protocols is available in the Selected column. You can customize the Selected list of ciphers and protocols to match the security and system needs for your enterprise.

The available and default set of cipher suites and protocols may be updated in a release. Some cipher suites and protocols may be added, while others may be removed. Cipher suites and protocols may be removed if the platform no longer supports these cipher suites and protocols.

If you are set up to use the default cipher suites and protocols, these will be updated to the latest defaults when you upgrade to a new version of Standalone Sentry. If you are set up to use a custom list of Selected cipher suites and protocols, the custom list is preserved when you upgrade your Standalone Sentry. However, any cipher suites or protocols that were removed will also be removed from the Selected and Available columns. New cipher suites and protocols will be added to the Available column.

Making changes to the selected list of cipher suites may impact the performance and security of traffic through Ivanti Standalone Sentry. Therefore, before making any changes to the Selected cipher suites, Ivanti recommends that you understand both the performance and security impact of the changes.

- SSLv2Hello is a pseudo-protocol that allows Java to initiate the handshake with an SSLv2 'hello message.' This does not cause the use of the SSLv2 protocol, which is not supported by Java. SSLv2Hello requires that TLSv1 protocol is also selected.

SSLv2Hello is required by some load balancers and SSL off loaders for proper functioning. If your environment does not need it, it is recommended to remove this from the protocol list for improved security.

Procedure
1. In Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **EMM SSL Configuration**. Ciphers and protocols are configured in the **Sentry to Backend Ciphers, SNI, and Protocols Configuration** section.

The **Use Default Cipher Suites and Protocols (recommended)** option is selected by default.
2. Select **Use Custom Configuration**.
3. Click **Proceed** to continue.
4. Select the protocols and cipher suites to move from the **Available** to **Selected** column or vice-versa as necessary.

The default cipher suites and protocols are colored blue.
5. Click **Apply** to save the changes.

> When Use Default Cipher Suites and Protocols (recommended) is selected, the cipher suites and protocols can be moved between the Available and Selected columns. However, the configuration is not changed. You must also select the Use Custom Configuration option to make changes to the default configuration.

Switching back to default configuration

You can revert your settings to default configuration if you do not wish to use the custom configuration.

Procedure
1. In Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **EMM SSL Configuration**.
2. In the **Sentry to EMM Ciphers, SNI, and Protocols Configuration** section, select **Use Default Cipher Suites and Protocols (recommended)**.
3. Click **Apply** to save the changes.

The cipher suites and protocols are reset to the default settings.

Clicking on **Reset to Default** resets the **Available** and **Selected** columns to default settings. However, the default settings will not be applied. To apply the default settings, you must select **Use Default Cipher Suites and Protocols (recommended)**, and then click **Apply**.

Access SSL Configuration

Use the Access SSL Configuration page to configure the client role parameters for communication from Sentry to Access. You can configure ciphers and protocols for outgoing traffic from Sentry to Access.

- "Managing Strict TLS" on the next page settings for Ivanti Standalone Sentry SSL connections to Access.

- "Server Name Indication (SNI)" on the next page.

- View the **Available** and **Selected** protocols and cipher suites. See "Cipher suites and protocols" on page 122.

- Set up custom protocol and cipher suite configuration. See "Customizing cipher suites and protocols" on page 122.

The **Access SSL Configuration page** allows the administrator the flexibility to configure Ivanti Standalone Sentry to use cipher suites and protocols to match the security and system needs of your enterprise.

Managing Strict TLS

You can enable strict TLS for outgoing traffic from Standalone Sentry to Access. Strict TLS is enabled by default. With strict TLS enabled, the Java Trust Store is enabled by default. You can also use the custom trust store option to upload additional certificates that Ivanti Standalone Sentry must use.

Procedure
1. In the Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **Access SSL Configuration**.
2. In the **Strict TLS Settings** section, select or deselect **Enable Strict TLS** to enable or disable Strict TLS appropriately.

Additional options are now available.

Item	Description
Enable Default Java Trust Store	Selected by default if strict TLS is enabled. Certificates and Certificates Authorities in the Java Trust Store are used to trust the SSL connection to UEM.
Allow and Log untrusted servers	Select to allow Standalone Sentry to connect to UEM that does not use a trusted certificate in Java or custom trust store.
Enable Custom Trust Store	Select to upload certificates to the Ivanti Standalone Sentry trust store. Standalone Sentry uses the certificates in the custom store to trust UEM. Generally used if UEM uses self-signed certificates.

3. Click **Apply**.
4. Click **Yes**.

The new TLS settings are applied and Standalone Sentry restarts. It may take up to one minute for Ivanti Standalone Sentry to restart. Traffic is disrupted till Standalone is up and running again.

5. Click **OK**.

Server Name Indication (SNI)

Server Name Indication (SNI) is an extension to TLS. SNI allows multiple hostnames to be served over HTTPS from one IP address. By default, SNI is enabled on Ivanti Standalone Sentry for outgoing connections. SNI allows a load balancer to direct incoming traffic to the correct Access server based on the hostname provided by the client, in this case,Ivanti Standalone Sentry. Access servers require that SNI is enabled in the client. Your Active Directory Federation Services (ADFS) requires SNI for all client communications.

ⓘ   If SNI is enabled for Access SSL connections, in some cases health check may fail if the Access server does not also support SNI. The workaround is to disable health check for the impacted server.

Cipher suites and protocols

Ivanti Standalone Sentry includes a set of cipher suites and protocols. A default set of cipher suites and protocols is available in the Selected column. You can customize the **Selected** list of ciphers and protocols to match the security and system needs for your enterprise.

The available and default set of cipher suites and protocols might be updated in a release. Some cipher suites and protocols might be added, while others may be removed. Cipher suites and protocols might be removed if the platform no longer supports these cipher suites and protocols.

If you are set up to use the default cipher suites and protocols, these are updated to the latest defaults when you upgrade to a new version of Ivanti Standalone Sentry. If you are set up to use a custom list of Selected cipher suites and protocols, the custom list is preserved when you upgrade your Ivanti Standalone Sentry. However, any cipher suites or protocols that were removed are also removed from the Selected and Available columns. New cipher suites and protocols are added to the Available column.

Making changes to the selected list of cipher suites may impact the performance and security of traffic through Ivanti Standalone Sentry. Therefore, before making any changes to the Selected cipher suites, Ivanti recommends that you understand both the performance and security impact of the changes.

- SSLv2Hello is a pseudo-protocol that allows Java to initiate the handshake with an SSLv2 'hello message.' This does not cause the use of the SSLv2 protocol, which is not supported by Java. SSLv2Hello requires that r TLSv1 protocol is also selected.

  SSLv2Hello is required by some load balancers and SSL off loaders for proper functioning. If your environment does not need it, it is recommended to remove this from the protocol list for improved security.

Customizing cipher suites and protocols

You can customize the cipher suites and protocols configuration.

Procedure

1. In Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **Access SSL Configuration**. Ciphers and protocols are configured in the **Sentry to CMS Ciphers, SNI, and Protocols Configuration** section.

The **Use Default Cipher Suites and Protocols (recommended)** option is selected by default.

2. Select **Use Custom Configuration**.

---

3. Click **Proceed** to continue.
4. Select the protocols and cipher suites to move from the **Available** to **Selected** column or vice-versa as necessary.

The default cipher suites and protocols are colored blue.

5. Click **Apply** to save the changes.

> When Use Default Cipher Suites and Protocols (recommended) is selected, the cipher suites and protocols can be moved between the Available and Selected columns. However, the configuration is not changed. You must also select the Use Custom Configuration option to make changes to the default configuration.

Switching back to default configuration

You can revert your settings to default configuration if you do not wish to use the custom configuration.

Procedure

1. In Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **Access SSL Configuration**.
2. In the **Sentry to CMS Ciphers, SNI, and Protocols Configuration** section, select **Use Default Cipher Suites and Protocols (recommended)**.
3. Click **Apply** to save the changes.

The cipher suites and protocols are reset to the default settings.

Clicking **Reset to Default** resets the **Available** and **Selected** columns to default settings. However, the default settings are not applied. To apply the default settings, you must select **Use Default Cipher Suites and Protocols (recommended)**, and then click **Apply**.

Outbound HTTP Proxy

Use the Outbound HTTP Proxy page to configure the proxy parameters for communication from Sentry to Ivanti Neurons for MDM or Access.

Configuring Outbound HTTP Proxy

The **Outbound HTTP Proxy** page allows the administrator the flexibility to configure Ivanti Standalone Sentry with outbound HTTP proxy server settings. The traffic from Sentry passes through the proxy to Ivanti Neurons for MDM or Access. Sentry will use the User Name and Password for authentication if requested by the proxy.

Procedure

1. In Ivanti Standalone Sentry System Manager, go to **Settings** > **Services** > **Sentry** > **Outbound HTTP Proxy.**

2. Select **Ivanti Neurons for MDM** to configure the outbound proxy server settings for Ivanti Neurons for MDM.

Enter the **Proxy Host**.

Enter the **Proxy Port**.

Enter the **User Name**.

Enter the **Password**.

3. Select **Access** to configure the outbound proxy server settings for Access.

Enter the **Proxy Host**.

Enter the **Proxy Port**.

Enter the **User Name**.

Enter the **Password**.

4. Click **Apply**.

Log representation and format

The following provide the representation and format of the data captured in audit and health logs:

- "Audit log representation and format" below

- "Health log representation and format" on page 130

Audit log representation and format

An audit entry is created for each request from a device. A corresponding response entry is created for each request. The audit logs are in JSON format.

The following provide the format for audit log entries:

- "Audit log entry for a request" on the next page

- "Audit log entry for a response" on page 126

- "Audit log entry for IP VPN response to tunnel establishment request" on page 126

Audit log entry for a request

The following provides a description of the fields in the audit log entry for a request.

TABLE 21. FIELD DESCRIPTIONS FOR A REQUEST IN AUDIT LOG

Field	Description
publishTime	Actual time of log capture. Logging time might vary based on async strategies.
entryID	Unique for every audit entry. GUID.
useCaseID	ID of use-case to which this entry belongs to. This ID is used for relating Request/Response.
entryType	REQUEST.
userID	EMM User ID.
deviceID	Device identification.
deviceType	Type of device - iPhone, iPad etc.
serviceType	ActiveSync, CIFS, Access, APP_TUNNEL, TCP_TUNNEL, IP_TUNNEL.
serviceName	
clientHost	
clientPort	
requestUrl	URL used by device.
httpMethod	HTTP method used for this request.
applicationId	
forwardedFor	If proxy is forwarding request, this will have actual client host identifier.
contextHeaders	
serverHost	Details of downstream server.
serverPort	
action	ALLOW	BLOCK	NONE (Sentry compliance action taken - NONE - no compliance[Access])

Audit log entry for a response

The following provides a description of the fields in the audit log entry for a response.

TABLE 22. FIELD DESCRIPTIONS FOR A RESPONSE IN AUDIT LOG

Field	Description
publishTime	Actual time of log capture. Logging time might vary based on async strategies.
entryID	Unique for every audit entry. GUID.
useCaseID	ID of use-case to which this entry belongs to. This ID is used for relating Request/Response.
entryType	RESPONSE.
userID	EMM user ID.
deviceID	Device identification.
deviceType	Type of device.
serviceType	ActiveSync, CIFS, Access, APP_TUNNEL, TCP_TUNNEL, IP_TUNNEL.
serviceName	Name of service.
clientHost	Immediate client end-point; if coming via proxy, this could be proxy end-point.
clientPort	
httpStatus	HTTP Response code.
sentryHost	Standalone Sentry hostname.
sentryPort	Standalone Sentry port.
sentryAddress	Standalone Sentry IP address.

Audit log entry for IP VPN response to tunnel establishment request

The following provides a description of the fields in the audit log entry for a request to establish an IP VPN tunnel.

TABLE 23. FIELD DESCRIPTIONS FOR IP VPN RESPONSE TO IVANTI TUNNEL ESTABLISHMENT REQUEST IN AUDIT LOG

Field	Description
publishTime	Actual time of log capture. Logging time might vary based on async strategies.
entryID	Unique for every audit entry. GUID.
useCaseID	ID of use-case to which this entry belongs to. This ID is used for relating Request/Response.
entryType	RESPONSE.
userID	EMM User ID.
deviceID	Device identification.
serviceType	IP_TUNNEL.
clientHost	Immediate client end-point; if coming via proxy, this could be proxy end-point.
clientPort	
serverPort	
httpStatus	HTTP Response code.

Audit log entry for IP VPN internal connection

The following provides a description of the fields in the audit log entry for an internal IP VPN tunnel connection.

TABLE 24. FIELD DESCRIPTIONS FOR AN IP VPN INTERNAL CONNECTION ENTRY IN AUDIT LOGS

Field	Description
publishTime	
entryID	Unique for every audit entry. GUID.
useCaseID	ID of use-case to which this entry belongs to. This ID is used for relating Request/Response.
entryType	IP_VPN_CONN.
userID	
deviceID	
serviceType	IP_TUNNEL.
clientHost	
clientPort	
serverHost	
serverPort	
action	Compliance action like ALLOW, BLOCK, NONE.
type	Connection type: UDP or TCP.
sentryHost	Standalone Sentry hostname.
sentryPort	Standalone Sentry port.
sentryAddress	Standalone Sentry IP address.

Examples for audit log entries

Following are examples of audit log entries:

- "IPVPN audit log example" on the next page

    - "ActiveSync audit log example" on the next page

    - "HTTP tunnel audit log example" on the next page

    - "TCP tunnel audit log example" on page 130

IPVPN audit log example

```
2017 Nov  1 04:13:59 eapp123.auto.ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
43fbd6d7-258d-4d55-aa81-
cf1ba11533b4","entryType":"RESPONSE","userId":"hdhindsa","deviceId":"22002","serviceTy
pe":"IP_
TUNNEL","clientHost":"/24.5.120.210","clientPort":44258,"publishTime":"11/01/2017
4:13:59","entryId":"E-6ec1eeda-5d25-4d3b-8107-
5101c188830f","serverPort":443,"httpStatus":"200"}


2017 Nov  1 04:14:06 eapp123.auto.ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
43fbd6d7-258d-4d55-aa81-cf1ba11533b4","entryType":"IP_VPN_
CONN","userId":"hdhindsa","deviceId":"22002","serviceType":"IP_
TUNNEL","clientHost":"/24.5.120.210","clientPort":44258,"publishTime":"11/01/2017
4:14:06","entryId":"E-4190ad90-4391-47b1-b2b3-
298aec6aec5a","serverHost":"autodns001.auto.ivanti.com","serverPort":53,"action":"ALLO
W","type":"UDP"}


2017 Nov  1 04:14:06 eapp123.auto.ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
43fbd6d7-258d-4d55-aa81-cf1ba11533b4","entryType":"IP_VPN_
CONN","userId":"hdhindsa","deviceId":"22002","serviceType":"IP_
TUNNEL","clientHost":"/24.5.120.210","clientPort":44258,"publishTime":"11/01/2017
4:14:06","entryId":"E-b30097d0-f888-4437-b49d-
232d4f364815","serverHost":"216.58.192.10","serverPort":443,
"sentryHost":"10.10.57.239","sentryPort":446, "sentryAddress":"10.25.35.237",
"action":"ALLOW","type":"TCP"}
```

ActiveSync audit log example

```
2017 Nov  7 21:23:39 app101.auto.ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
ee3608c9-4c88-4b93-8221-
bd69cb4da900","entryType":"REQUEST","userId":"testuser0851","deviceId":"HroLBGueAofSIk
AcECcHMTTqd2","deviceType":"MD723LL","serviceType":"ACTIVE_
SYNC","serviceName":"ActiveSync","clientHost":"/10.11.80.93","clientPort":61693,"publi
shTime":"11/07/2017 21:23:38","entryId":"E-ee3608c9-4c88-4b93-8221-
bd69cb4da900","serverHost":"ex2013.auto19.ivanti.com","serverPort":443,"requestUrl":"/
Microsoft-Server-ActiveSync","httpMethod":"POST","action":"ALLOW"}


2017 Nov  7 21:23:41 app101.auto.ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
ee3608c9-4c88-4b93-8221-
bd69cb4da900","entryType":"RESPONSE","userId":"testuser0851","deviceId":"HroLBGueAofSI
kAcECcHMTTqd2","serviceType":"ACTIVE_
SYNC","clientHost":"/10.11.80.93","clientPort":61693,"publishTime":"11/07/2017
21:23:39","entryId":"E-49b382b2-07c9-4a82-87d3-
3f1f45751879","serverHost":"ex2013.auto19.ivanti.com","serverPort":443,"sentryHost":"1
0.10.57.239","sentryPort":446, "sentryAddress":"10.25.35.237", "httpStatus":"200"}
```

HTTP tunnel audit log example

```
2017 Nov  3 23:06:57 eapp074.auto.Ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
dd7086fc-9599-4581-a8bc-
5a9057ce085b","entryType":"REQUEST","userId":"testuser7331","deviceId":"62b6ae69-9ca8-
4176-85dd-11a7ecaee130","deviceType":"iPhone 6","serviceType":"APP_
TUNNEL","serviceName":"<ANY>","clientHost":"/10.11.205.8","clientPort":1821,"publishTi
me":"11/03/2017 23:06:57","entryId":"E-dd7086fc-9599-4581-
a8bc5a9057ce085b","serverHost":"wiki.ivanti.com","serverPort":443,"requestUrl":"https:
//wiki.ivanti.com/login.action?os_
destination=%2Findex.action&permissionViolation=true","httpMethod":"GET","applicationI
d":"com.ivanti.securebrowser","action":"ALLOW"}


2017 Nov  3 23:06:57 eapp074.auto.Ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
dd7086fc-9599-4581-a8bc-
5a9057ce085b","entryType":"RESPONSE","userId":"testuser7331","deviceId":"62b6ae69-
9ca8-4176-85dd-11a7ecaee130","serviceType":"APP_
TUNNEL","clientHost":"/10.11.205.8","clientPort":1821,"publishTime":"11/03/2017
23:06:57","entryId":"E-c0cd7a3d-1832-4b85-b28c-
7385d2b0eb0c","serverHost":"wiki.ivanti.com","serverPort":443,
"sentryHost":"10.10.57.239","sentryPort":446", "sentryAddress":"10.25.35.237",
"httpStatus":"200"}
```

TCP tunnel audit log example

```
2017 Nov  3 23:06:07 eapp074.auto.ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
bd77654c-42dc-48f3-9b2c-
9aa2d5d63650","entryType":"REQUEST","userId":"testuser7331","deviceId":"62b6ae69-9ca8-
4176-85dd-11a7ecaee130","serviceType":"TCP_TUNNEL","serviceName":"<TCP_
ANY>","clientHost":"/10.11.205.8","clientPort":1391,"publishTime":"11/03/2017
23:06:07","entryId":"E-bd77654c-42dc-48f3-9b2c-
9aa2d5d63650","serverHost":"googleads.g.doubleclick.net","serverPort":443,"application
Id":"com.google.chrome.ios","action":"ALLOW"}




2017 Nov  3 23:06:07 eapp074.auto.ivanti.com SENTRY_AUDIT:  INFO  {"usecaseId":"U-
bd77654c-42dc-48f3-9b2c-
9aa2d5d63650","entryType":"RESPONSE","userId":"testuser7331","deviceId":"62b6ae69-
9ca8-4176-85dd-11a7ecaee130","serviceType":"TCP_
TUNNEL","clientHost":"/10.11.205.8","clientPort":1391,"publishTime":"11/03/2017
23:06:07","entryId":"E-4fa74e1f-e0df-4093-9cd1-
a716aa0697ff","serverHost":"googleads.g.doubleclick.net","serverPort":443,
"sentryHost":"10.10.57.239","sentryPort":446", "sentryAddress":"10.25.35.237",
"httpStatus":"200"}
```

Health log representation and format

The following provide the representation and format for Sentry health logs:

- "/var/log/mihealth_export/openPorts.log " on the next page

- "/var/log/mihealth_export/hardware.log" on the next page

- "/var/log/mihealth_export/cpu.log" below

- "/var/log/mihealth_export/vmstat.log" on the next page

/var/log/mihealth_export/openPorts.log

sourcetype: sentry_mihealth_openPorts

```
 Proto    Port
 tcp     9090
 ...
 udp    10012
```

REGEX = ([^\s]+)\s+([0-9]+)

FORMAT = Proto::"$1" Port::"$2"

/var/log/mihealth_export/hardware.log

sourcetype: sentry_mihealth_hardware

```
 KEY                 VALUE
 CPU_TYPE            Intel(R) Xeon(R) CPU E5504 @ 2.00GHz
 CPU_CACHE           4096 KB
 CPU_COUNT           1
 HARD_DRIVES         sda (Virtual disk) 200 GB;
 NIC_TYPE            <notAvailable>
 NIC_COUNT           1
 MEMORY_REAL         2054232 kB
 MEMORY_SWAP         4128764 kB
```

/var/log/mihealth_export/cpu.log

sourcetype: sentry_mihealth_cpu

```
 CPU    pctUser    pctNice   pctSystem   pctIowait    pctIdle
 all      0.00        1.01        1.01        0.00       97.98
 0        0.00        1.01        1.01        0.00       97.98
```

REGEX = all\s+(\d*\.*\d*)\s+(\d*\.*\d*)\s+(\d*\.*\d*)\s+(\d*\.*\d*)\s+(\d*\.*\d*)

FORMAT = pctUser::$1 pctNice::$2 pctSystem::$3 pctIowait::$4 pctIdle::$5

/var/log/mihealth_export/vmstat.log

/usr/bin/vmstat

sourcetype: sentry_mihealth_vmstat

```
time=2017-09-05 10:24:01, r=5, b=0, swpd=10268, free=80444, buff=109964, cache=845276,
si=0, so=0, bi=5, bo=12, in=115, cs=208, us=1, sy=0, id=99, wa=0, st=0
```

Ivanti Standalone Sentry Security Settings

The following describe the security settings in the Ivanti Standalone Sentry System Manager:

Overview of Ivanti Standalone Sentry security settings

The Security tab in System Manager contains links for configuring aspects of Sentry access. The following table summarizes the tasks associated with each link.

TABLE 25. CONFIGURATION LINKS IN THE SECURITY TAB

Settings	Description
Identity Source: Local Users	Create, delete, and manage local users.
Identity Source: Password Policy	Configure complex passwords for Standalone Sentry.
Certificate Mgmt	View and manage certificates for Portal HTTPS
Access Control Lists: Networks & Hosts	Create and manage entries for networks and hosts
Access Control Lists: Network Services	Create and manage entries for network services
Access Control Lists: ACLs	Compile access control lists

Local Users

All users in the Ivanti Standalone Sentry System Manager database are local users having the following privileges, which cannot be changed:

- Command Line Interface (CLI)

- System Manager access

Adding local users for System Manager

You can add local users in the Ivanti Standalone Sentry System Manager.

Procedure
1. Go to **Security** > **Local Users**.
2. Click **Add**. The Add New User window displays.
3. Use the following guidelines to complete the form:

Field	Description
User ID	Enter the unique identifier to assign to this user.
First Name	Enter the user's first name.
Last Name	Enter the user's last name.
Password	Enter a password for the user. For password requirements, see "Password policy" on the next page
Confirm Password	Confirm the password for the user.
Group	This field is not configurable.
Email	Enter the user's email address.
EDIPI	Enter the unique identifier assigned to this user. This is a mandatory field to configure CAC.

4. Click **Apply**.
5. Click **Save**.

Editing local users for System Manager

You can change the information for local users in the Ivanti Standalone Sentry System Manager.

Procedure
1. Go to **Security > Local Users**.
2. Select the user ID of the entry to display the information for that user.
3. Make your changes.
   You cannot change the user ID.
4. Click **Apply**.
5. Click **Save**.

Deleting local users for System Manager

You can delete local users in the Ivanti Standalone Sentry System Manager.

Procedure

1. Go to **Security > Local Users**.
2. Select the checkbox for the user you want to delete.
3. Click **Delete**.
   You cannot delete the user ID you logged in with.
4. Click **Save**.

Password policy

Password policy lets you configure complex passwords for Ivanti Standalone Sentry.

Configuring password policy

To configure the settings in the Standalone Sentry System Manager, go to **Security > Identity Source**.

Before you begin

- Verify that you have added local users for System Manager.

Procedure

1. On the Security tab, expand **Identity Source**.
2. Select **Password Policy**.
3. Configure the password policy for local users appropriately.

Policy	Values
Minimum Number of Character Classes in Password	1 to 4. Passwords must contain at least one upper case character, one lower case character, and one numeric character by default.
Lower Case	Enable or Disable
Upper Case	Enable or Disable
Numeric	Enable or Disable
Special Character	Enable or Disable. Password can contain special characters only from this set "!=({[_:-;~,)}]@#^|$".

Policy	Values
Minimum Password Length	Passwords must have at least 6 characters. The length is set to 8 by default.
Maximum Password Length	Password length can extend up to 128 characters. The length is set to 32 by default.
Number of Failed Attempts	Failed password attempts are limited from 1 to 16. The number of attempts is set to 5 by default.
Auto-Lock Time	0-3600 seconds. You can set the time for password auto-lock.
Enforce Passcode History (Last 4 passwords)	Enable or Disable.

4. Click **Apply**.
5. Click **Yes** to confirm the change.
6. Click **OK** to save the password policy configuration.

Certificate Management

Use the Certificate Management feature in the Sentry System Manager in **Security > Certificate Mgmt** to manage the certificate required for browsers to access the Ivanti Standalone Sentry System Manager.

You can perform the following tasks from the Certificate Management screen:

- Generate a self-signed certificate

- Generate a certificate signing request (CSR) for a certificate authority (CA)

- Upload a certificate.

> When you update a certificate, you are prompted to confirm that you want to proceed because the HTTP service needs to be restarted, resulting in service disruption.

Generating a self-signed certificate for the Ivanti Standalone Sentry portal

If you use a self-signed certificate, a browser that is connecting to the Sentry System Manager is warned that the Sentry certificate is not from a trusted source. Therefore, Ivanti recommends that you use a certificate from a trusted Certificate Authority (CA).

To generate a self-signed certificate, in the Sentry System Manager go to **Security > Certificate Mgmt.**

Procedure
1. Click the **Manage Certificate** link for **Portal HTTPS**.
2. For **Certificate Options**, select **Generate Self-Signed Certificate** from the dropdown list.

FIGURE 1. GENERATE SELF-SIGNED CERTIFICATE
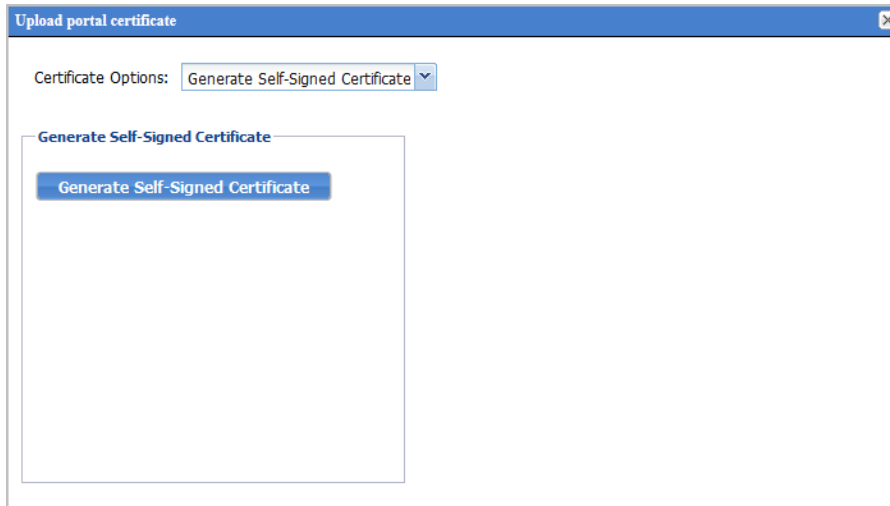
3. Click the **Generate Self Signed Certificate** button.

Generating a certificate signing request (CSR)

To get a certificate from a trusted Certificate Authority (CA), use the **Security > Certificate Mgmt** page to generate a certificate signing request (CSR) to the CA. Once you receive the signed certificate, you can use the same page to upload it to Sentry.

Procedure
1. Click the **Manage Certificate** link for **Portal HTTPS**.
2. For **Certificate Options**, select **Generate CSR** from the dropdown list.
3. Use the following guidelines to complete the displayed form:

Field	Description
Common Name	Enter the server host name.
E-Mail	Enter the email address of the contact person in your organization who should receive the resulting certificate.
Company	Enter the name of the company requesting the certificate.

Field	Description
Department	Enter the department requesting the certificate.
City	Enter the city in which the company is located.
State	Enter the state in which the company is located.
Country	Enter the two-character abbreviation for the country in which the company is located.
Key Length	Select 2048 or 3072 to specify the length of each key in the pair. Longer keys provide stronger security, but may impact performance.

4. Click **Generate**.

A message similar to the following displays.

FIGURE 2. CERTIFICATE REQUEST
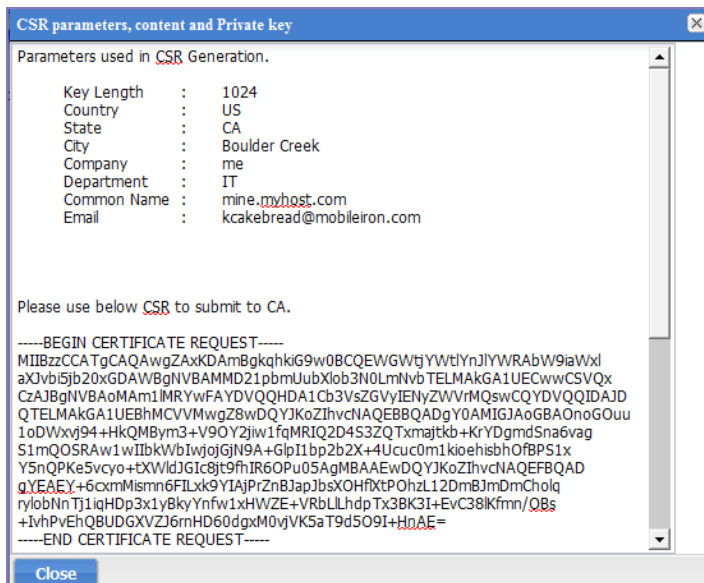
5. Copy the content between BEGIN RSA PRIVATE KEY and END RSA PRIVATE KEY to another text file.
6. Click **Close.**
7. Submit the file you created in step 5 to the certifying authority.

Uploading certificates

To upload the CA certificate from the certifying authority in the Ivanti Standalone Sentry System Manager go to **Security > Certificate Mgmt.**

Procedure

1. Click the **Manage Certificate** link for **Portal HTTPS**.
2. For **Certificate Options**, select **Upload Certificate**.
3. Select the certificates as indicated in the following table:

Certificate	File to Select
Key file	The file created in "Generating a certificate signing request (CSR)" on page 137.
Server certificate	The CA certificate file you received from the certifying authority.
CA certificate	The generic CA certificate file.

4. Click the **Upload Certificate** button.

Viewing certificates

To view a certificate, in the Ivanti Standalone Sentry System Manager go to **Security > Certificate Mgmt** and click the **View Certificate** link for Portal HTTPS.

Access Control Lists

Use the Access Control Lists screen in the Ivanti Standalone Sentry System Manager in **Security > Access Control Lists** to compile and manage the rules that define inbound and outbound access for network hosts and services.

FIGURE 1. ACCESS CONTROL LISTS
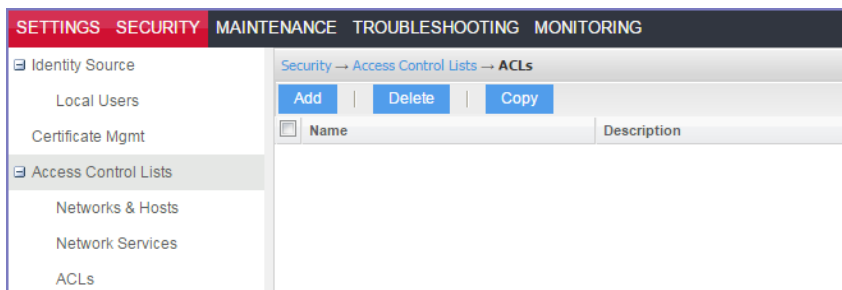
Each access control list (ACL) consists of one or more access control entries (ACEs). Configuring ACLs requires the following tasks:

1. Configure entries for each network and host requiring an ACL.
2. Configure entries for any network services requiring an ACL.
3. Create an ACL.

Adding an ACL

To configure an access control list in the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists**.

Procedure

1. In the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists**.
2. Click **Add**.
3. In the **Name** field, enter a name to identify the ACL.
4. In the **Description** field, enter text to clarify the purpose of the ACL.
5. Click **Save**.

The lower portion of the screen is now enabled.

FIGURE 2. ADD ACL

6. Click **Add** to add an access control entry (ACE) to the ACL.

Each ACE consists of a combination of the network hosts and services you configured for use in ACLs.

7. Use the following guidelines to complete the form:

Field	Description
Source Network	Select the network from which access will originate. This list is populated with the networks and hosts you created for use with ACLs. See ""Networks and Hosts" on page 142" on "Networks and Hosts" on page 142.

Field	Description
Destination Network	Select the network being accessed. This list is populated with the networks and hosts you created for use with ACLs. See ""Networks and Hosts" on the next page" on "Networks and Hosts" on the next page.
Service	Select the network service to which this entry permits or denies access. This list is populated with the services you created for use with ACLs. See ""Network Services" on page 143" on "Network Services" on page 143.
Action	Select Permit or Deny from the dropdown list.
Connections Per Minute	Enter the number of connections to allow per minute.
Description	Enter text to describe the purpose of this entry.

8. Click **Save**.

Editing an ACL

To edit an access control list in the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists**.

Procedure

1. In the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists**.
2. Click the name in the ACLs list.

3. To delete an ACE, click its checkbox and click **Delete**.

4. To add an ACE, click **Add**.
5. To insert an ACE, select the ACE above which you want to insert a new ACE and click Insert.
6. Click **Save**.

Copying an ACL

To create a copy of an existing ACL in Ivanti Standalone Sentry, go to **Security > Access Control Lists**.

Procedure

1. In the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists**.
2. Select the ACL to be copied.
3. Click the **Copy** button.
4. Enter a name for the new ACL.
5. Click **OK**.

Deleting an ACL

To delete an existing ACL in Ivanti Standalone Sentry, go to **Security > Access Control Lists**.

Procedure

1. In the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists**.
2. Select the ACL to be deleted.
3. Click **Delete**.

Networks and Hosts

Use the Networks and Hosts screen to manage the servers and subnets you will use to compile Access Control Lists (ACLs).

FIGURE 1. NETWORK AND HOSTS
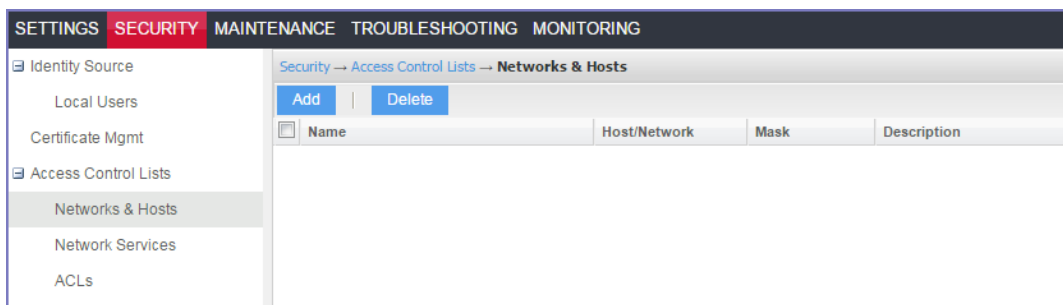
Adding a host or subnet for compiling ACLs

To add a host or subnet for compiling ACLs in the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists > Network & Hosts.**

Procedure
1. Click **Add**.

FIGURE 2. ADD NETWORK HOSTS

2. Use the following guidelines for completing the displayed form:

Field	Description
Name	Enter a name to use to identify this host or network.
Description	Enter additional text to provide supporting information about this host or network.
Type	Select Subnet or Host from the dropdown menu.
Network/Host	Enter the IP address for this network or host.

3. Click **Save**.

This host or network will now be available for ACLs configured in the ACLs screen.

Network Services

Use the Network Services screen to manage available services. Ivanti prepopulates this screen with common services.

FIGURE 1. NETWORK SERVICES FOR ACL
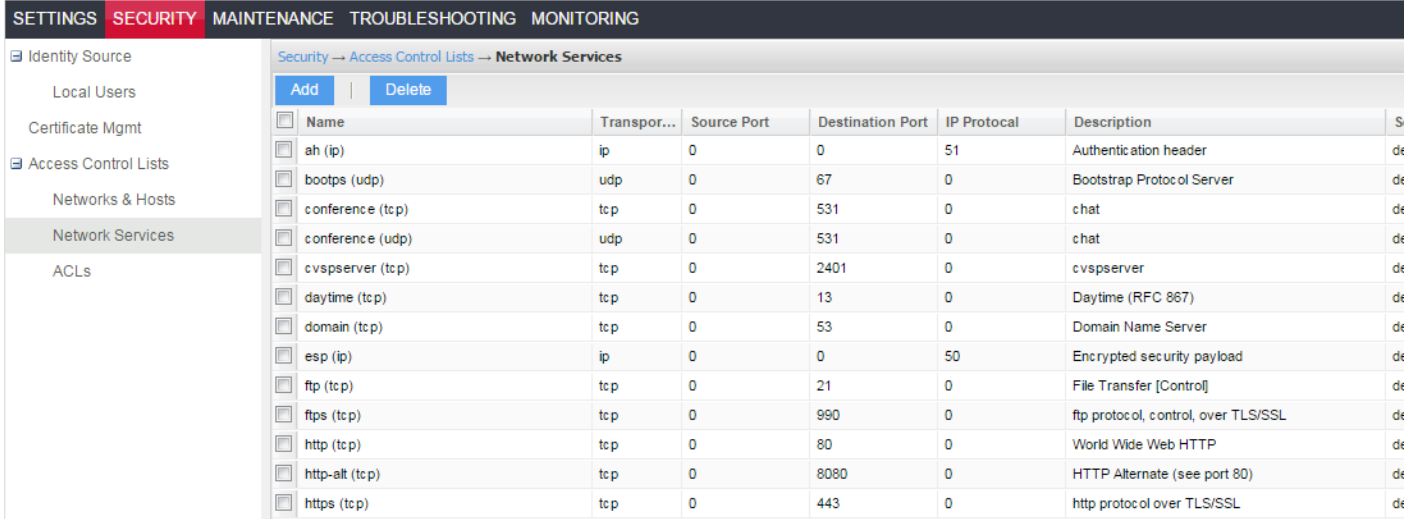
Adding a network service

To add a network service for compiling ACLs in the Ivanti Standalone Sentry System Manager, go to **Security > Access Control Lists > Network Services.**

Procedure

1. Click **Add**.

FIGURE 2. ADD NETWORK SERVICE

2. Use the following guidelines to complete the form:

Field	Description
Name	Enter a name to use to identify this service.
Description	Enter additional text provide supporting information about this service.

Field	Description
Type	Select TCP, UDP, or IP from the dropdown menu.
Source Port	Enter the number of the source port for this service. Enter 0 to allow any source port.
Destination Port	Enter the number of the destination port for this service. Enter 0 to allow any destination port.

3. Click **Save**.

Access Control Lists: ACLs

See ""Access Control Lists" on page 139" on "Access Control Lists" on page 139.

Ivanti Standalone Sentry Maintenance Settings

The following describe the maintenance settings in the Ivanti Standalone Sentry System Manager:

Overview of Sentry maintenance features

The **Maintenance** tab in the Ivanti Standalone Sentry System Manager provides basic maintenance features for Standalone Sentry appliance. The following table summarizes these features.

TABLE 26. CONFIGURATION LINKS IN THE MAINTENANCE TAB

Setting	Description
Software Updates	Upgrade Standalone Sentry software.
Export Configuration	Save the system configuration file.
Import Configuration	Import a saved system configuration file.
Clear Configuration	Clear the current system settings.
Reboot	Restart the Standalone Sentry.

Updating Ivanti Standalone Sentry software

> ℹ️ If you are upgrading using a URL and not using the **Default** setting, use the CLI upgrade method. See "Upgrading using CLI" on page 205.

Before you begin

See the *Ivanti Standalone Sentry Release and Upgrade Notes* for release specific information.

Procedure

1. In Sentry System Manager, go to **Maintenance > Software Updates**.

2. **Software Version**: Check the Ivanti Standalone Sentry version.

3. Set up the Software Repository Configuration.

    a. Enter the credentials assigned by Support.

    b. For **URL, Default** is selected.

    c. Click **Apply**.

    d. Click **OK** to dismiss the success popup.

4. (Optional) If you are using a proxy server to support.mobileiron.com, set up **Software Repository Proxy Configuration**.

    a. **Hostname/IP**: Enter the proxy server hostname or IP address.

    b. **Port**: Enter the port number on the proxy server for Sentry.

    c. (optional) If needed, enter the credentials for the proxy server.

    d. Click **Apply**.

    e. Click **OK** to dismiss the success popup.

5. Click **Check Updates**.
   The available updates are listed.

6. Click **Download Now** if you want to download the update now and complete the installation at a later time.

7. Refresh the screen and click **Check Updates**.
   After the download is complete, the status for the update changes to **Downloaded**.

8. Click **Stage for Install** when you are ready to install.
   If you had already downloaded the selected update, the system stages the update for installation.
   If you did not previously download the selected update, it is downloaded and staged for installation.
   After the software update has been staged for installation, the status for the update changes to
   Reboot to Install. You can now install the update by rebooting the system. If the status of an update

is not **Reboot to Install**, rebooting the system will not install the update.

9. Click **Reboot** in the left navigation pane to install the software update.

Verifying that the upgrade is complete

The following allow you to verify that the Ivanti Standalone Sentry update is complete.

Procedure

1. In Ivanti Standalone Sentry System Manager, go to **Maintenance > Software Updates**.

Confirm that the version displayed is the current version**.**

2. In Ivanti Standalone Sentry System Manager, go to **Troubleshooting > Service Diagnosis**.

Confirm that status for the services listed shows **Success**.

3. Enroll a test device and validate email flow by sending and receiving email on the device.

Software update status

The following tables describes the status shown for each software upgrade:

TABLE 27. SOFTWARE UPGRADE STATUS

Status	Description
Not Downloaded	The update is not yet downloaded. **Next Step:** Click **Download Now** or **Stage for Install** to download the update.
Download in progress	The download is in progress. Refresh the browser to update the status.
Downloaded	The software update has been downloaded. **Next Step:** Click **Stage for Install** to stage the update for installation. The software update must be staged before installing.
Reboot to install	The software update was successfully downloaded and update is staged for the installation. **Next Step:** Click **Reboot** in the left navigation pane to install the software update.

Exporting the configuration

To back up the system configuration, you can export the Ivanti Standalone Sentry configuration settings to XML format.

Procedure

1. Select **Export Configuration**.

FIGURE 1. EXPORT CONFIGURATION
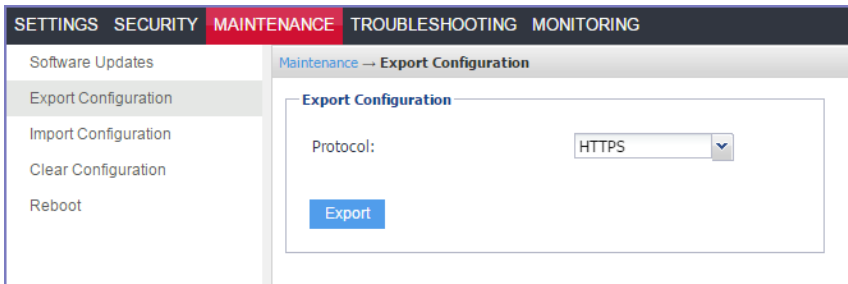
2. Click **Export**.

Importing a configuration

You can import an Ivanti Standalone Sentry configuration from a local XML file or FTP site.

Procedure

1. In the Ivanti Standalone Sentry system manager, go to **Maintenance**.
2. Select **Import Configuration**.

FIGURE 1. IMPORT CONFIGURATION
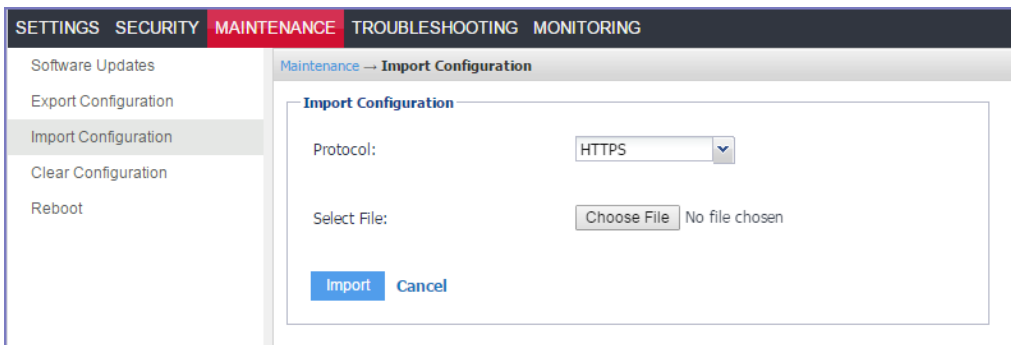
3. Click **Browse** to select an import file.
4. Click **Import**.

Copyright © 2024, Ivanti, Inc. All Rights Reserved. Privacy and Legal.

Clearing the configuration

Clear Configuration allows you to clear unsaved configuration settings and return to the default configuration.

Procedure

1. In the Ivanti Standalone Sentry system manager, go to **Maintenance**.
2. Click **Clear Configuration**.

FIGURE 1. CLEAR CONFIGURATION
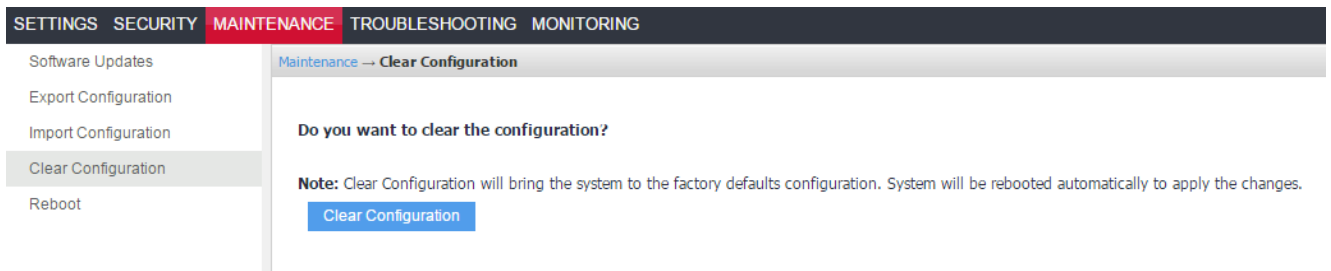
3. Click the **Clear Configuration** button.

The appliance is automatically rebooted to apply the changes.

Rebooting

You can reboot the Ivanti Standalone Sentry to clear the current configuration settings and restart all server modules.

Procedure

1. In the Ivanti Standalone Sentry system manager, go to **Maintenance**.
2. Select **Reboot** in the navigation pane.

FIGURE 1. REBOOTING
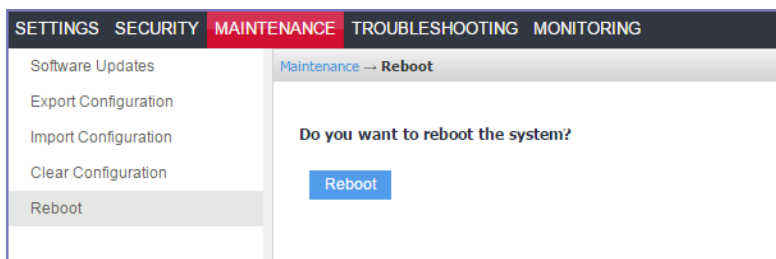
3. Click the **Reboot** button.

Troubleshooting

The following describe the troubleshooting settings in the Ivanti Standalone Sentry System Manager:

Overview of the Ivanti Standalone Sentry Troubleshooting tab

Use the **Troubleshooting** tab to investigate possible problems with Ivanti Standalone Sentry operation. In most cases, you will use this tab under the direction of Customer Support.

TABLE 28. CONFIGURATION LINKS IN THE TROUBLESHOOTING TAB

Settings	Description
Logs	Configure and upload log files.
Network Monitor	Produce a TCP dump for Sentry.
Service Diagnosis	Check the health of related servers.
Sentry Statistics	Produce an operational report.

Logs

The Logs page allows you to the following:

- Log management

- View module logs

- Export logs

Log management

You can enable or disable logging and control the details collected in Sentry logs for the following services:

- MICS (MobileIron Configuration Service)

- Sentry

For the Sentry service, you can do the following:

- Specify whether to exclude interactions between Sentry and the device or between Sentry and the backend resource.

- Specify levels of increasing detail.

- Filter logs based on specific attributes, such as Device ID.

FIGURE 1. TROUBLESHOOTING
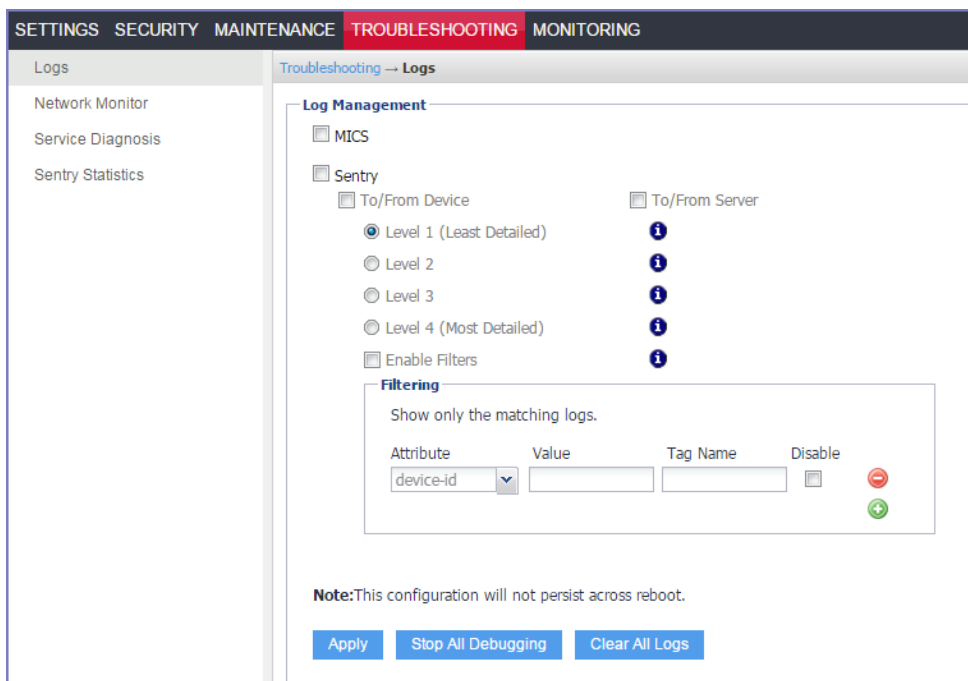
The following table describes the options for managing logs.

TABLE 29. OPTIONS FOR MANAGING LOGS

Item	Description
MICS	Includes messages related to the MobileIron Configuration System module that supports the Sentry.
Sentry	Includes messages related to Sentry operation.
To/From Device	Includes messages related to interactions between Sentry and the registered ActiveSync devices.
To/From ActiveSync Server	Includes messages related to interactions between Sentry the configured ActiveSync servers.
Level 1	Includes HTTP response/request lines and a few supporting operational messages. This level of provides the least detailed logging.
Level 2	Includes Level 1 content, HTTP headers, and additional operational messages. Sufficient when network issues are suspected. Primarily used for troubleshooting AppTunnel issues.
Level 3	Includes Level 1 and Level 2 content and the information associated with the messages. Enabled when you know that there are no network issues, but there may be a an issue with ActiveSync. This is the most common level requested by support. The logs contain WBXML. Log data includes email, calendar, and contact information.
Level 4	Includes all available log data. Enabled if parsing errors or missing data is suspected. Enable, only when requested by Support. Log data includes email, calendar, and contact information

Related topics

- "Turning logging on or off" on the next page

  - "Filtering log entries" on the next page

  - "Disabling filters" on page 157

  - "Deleting filters" on page 157

  - "Viewing logs" on page 158

Turning logging on or off

You can turn logging on or off by either selecting or deselecting the MICS or Sentry options under Log Management.

Procedure

1. Select or deselect the **MICS** checkbox to turn on or off MICS logging.
2. Select or deselect the **Sentry** checkbox to turn on or off Sentry logging.
3. If you turn on Sentry logging, select log options, level, and filters.
4. Click **Apply**.

Filtering log entries

The Filtering section of the Log Management screen enables you to isolate entries based on the following attributes:

- device-id—filters the logs based on the device id

- device-ip—filters the logs based on a specific ip address

- user-id—filters the logs based on a specific user id

    User id must be an exact match, but it is case insensitive.

The following example shows how to restrict the display to entries containing user ID johnd.

FIGURE 2. FILTER LOG ENTRIES

If multiple filters are specified, Sentry performs a logical OR operation; Sentry selects the log lines matching at least one of the filters.

Procedure

1. Select the **Enable Filters** checkbox.
2. Click the green **+** button to display a filter entry.
3. Select an attribute from the **Attribute** drop-down list.
4. In the **Value** field, enter the value you want to match.

The field is free-form and case-sensitive.

5. In the **Tag Name** field, enter a string that identifies the filter.

The tag name is added to the log output. Tag names are especially useful when you apply multiple filters. Adding a tag name is optional.

6. Click **Apply**.

Disabling filters

Disabling a filter removes the effect of the filter. Select the Disable checkbox to disable the filter. Clear it to re-enable the filter.

Deleting filters

If you do not intend to reuse a filter, you can delete it. To delete a filter, click the red **–** button.

View logs

The Troubleshooting > Logs page enables you to view the contents of logs directly from the console. Logging must be turned on.

TABLE 30. DESCRIPTION OF AVAILABLE LOGS

Log Name	Description
MICS	MobileIron Configuration Service log entries Sentry System Manager logs include IP, DNS, debugging, and upgrade configuration.
Sentry	Sentry operation log entries
System	Sentry status log entries
Syslog	A superset of the information in the MICS log Includes Sentry system level information and WARN and above application logs.
Catalina	Application loading status Includes the Sentry application server (tomcat) console logs.
Catalina2	Application loading status Includes the Sentry System Manager (MICS) application server (tomcat2) console logs.

Viewing logs

The **View Module Logs** section in the **Troubleshooting > Logs** page contains links to logs for Sentry modules.

Procedure

1. In the View Module Logs section, click the link for the log you want to view.

FIGURE 3. LOG VIEW
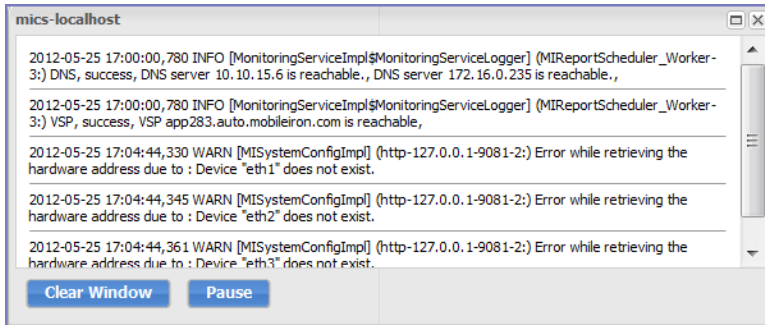
The displayed window shows the most recent log entries. The window scrolls dynamically as the Server adds entries to the log.

2. Click x to close the log view window.

If you close the log view window and then re-open it, the displayed window shows only log entries made since you closed the window.

To remove existing log entries from the log view window and view only new log entries, click the **Clear Window** button.

Exporting logs

The Export Logs section allows you to download logs.

Downloading logs

The following procedure describes how to download Sentry logs. Logs are downloaded to your local drive.

Procedure

1. Go to **Troubleshooting > Logs**.
2. Scroll down to the **Export Logs** section.
3. In the **Export Logs** section, select **Download** from the **Type** drop-down list
4. If you have received a support ticket number associated with this upload, enter it in the **Support Ticket Number** field.

5. Click **Download**.

Network Monitor

The **Network Monitor** page enables you to produce a TCP dump for an Ivanti Standalone Sentry physical interface. The information provided might assist in troubleshooting device connectivity problems. Click **Download** to store the results in a pcap file.

Use the following guidelines to complete this screen:

TABLE 31. NETWORK MONITOR FIELD DESCRIPTIONS

Option	Description
Interface	Select the physical interface for which you want to want to produce a TCP dump. If you have configured multiple interfaces, select **All** to get a TCP dump for all physical interfaces at one time.
Filter	not implemented.
Max. packet size	not implemented.
Max no. of Packets	not implemented.
Start	Click to start TCP dump. A growing file size indicates that the TCP dump is running.
Stop	Click to stop TCP dump. Click stop after reproducing the issue.
Download	Click to download TCP dump. If you click **Download** before starting the TCP dump you may not have any data to download. After starting a TCP dump, you may choose to download later after reproducing the issue. A growing file size indicates that the TCP dump is running.

Service Diagnosis

You can use the **Service Diagnosis** page under **Troubleshooting** to check the health of the following services:

- EAS

- NTP

- DNS

- UEM (Ivanti EPMM or Ivanti Neurons for MDM)

FIGURE 1. SERVICE DIAGNOSIS

FIGURE 1. SERVICE DIAGNOSIS

Click **Verify All** to recheck the listed services, or click **Verify** next to a specific service to verify just that service.

Clicking **Verify** next to the **UEM** (Ivanti EPMM or Ivanti Neurons for MDM) entry causes Ivanti Standalone Sentry to make another attempt to contact the UEM server. The resulting **Message** field of the **UEM** entry indicates whether the server is reachable.

Some reasons that the UEM may not be reachable include:

- Network errors.

- Actions taken by Technical Support for troubleshooting.

ActiveSync server status

You can check the ActiveSync server status by doing one of the following:

- In the Admin Portal, go to **Settings > Service Diagnostic**.

- In the Ivanti Standalone Sentry System Manager, go to **Troubleshooting > Service Diagnosis**.

Sentry Statistics

You can download statistics for Sentry operation. These statistics encompass the entire Sentry implementation and all connecting devices. They are most useful for charting true activity peaks so that you can schedule maintenance appropriately. Also, technical support can use these statistics for troubleshooting issues.

When Sentry Statistics is enabled or when Start is selected, the settings persists across Sentry restart. The settings in Sentry Statistics are saved as the following properties in /mi/alcor/config/v2/local/alcor-local.properties:

```
alcor.local.config.enable.sentry.global.statistics.report=true
alcor.local.config.statistics.log.interval.min=5
```

Sentry statistics collection is a continuous process. If Sentry statistics is started, the statistics are written to global-stats.csv. When Ivanti Standalone Sentry restarts, the data is archived.

Use this page for the following:

- Downloading Sentry Statistics

- Viewing Sentry Utilization

FIGURE 1. SENTRY STATISTICS
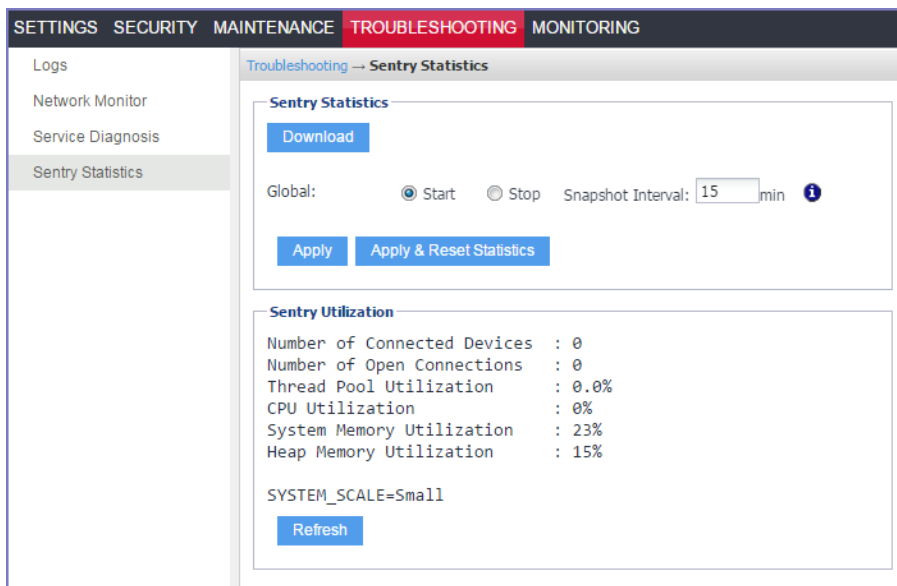
Download Sentry Statistics

Click the Download button to download a ZIP file containing the global statistics and device statistics for the Sentry. The ZIP file contains two CSV files:

- global-stats.csv

Provides overall Sentry statistics, useful for charting peak activity and for troubleshooting.

- all-device-stats.csv

Provides statistics for each device, useful for troubleshooting issues on a specific device.

Change Statistics collection

You can make the following changes to Sentry Statistics collection:

- start

- stop

- reset

- change the interval

TABLE 32. STATISTICS COLLECTION OPTIONS

Item	Description
Start	Select to start Sentry statistics collection.
	When the **Sentry Statistics** view shows **Start** as selected, statistics collection is written to global-stats.csv. This setting is not persistent. When Standalone Sentry restarts, the setting defaults to Stop. Statistics in global-stats.csv are archived
Stop	Select to turn off Sentry statistics collection.
	When the **Sentry Statistics** view shows **Stop** as selected, statistics collection is no longer written to global-stats.csv.
Apply & Reset Statistics	Click to clear the existing statistics file and restart statistics collection.

Changing the log interval

Sentry statistics are recorded to global-stats.csv every 5 minutes by default when statistics collection is enabled. You can change the statistics collection interval.

Procedure
1. Delete the current interval from the **Log Interval** field.
2. Enter a new interval.
3. Click **Apply**.

ⓘ   Changes to Snapshot Interval are persistent after a Standalone Sentry restart.

Sentry Utilization

To view the latest Sentry resource utilization information, click **Refresh**.

There is up to a one minute lag in updating the information.

System utilization alerts

The Ivanti Standalone Sentry monitors system utilization at 30 minute intervals. An alert is raised if utilization exceeds the default threshold level.

If you configured a syslog server in the Sentry System Manager, these alerts can be made available on the syslog server.

Alerts are generated for the following parameters

TABLE 33. SYSTEM UTILIZATION ALERT PARAMETERS

Parameter	Default Threshold
Thread Pool Utilization	80%
CPU Utilization	70%
System Memory Utilization	70%

Monitoring

The following describe the monitoring settings in the Ivanti Standalone Sentry System Manager:

Overview of the Ivanti Standalone Sentry Monitoring tab

Use the **Monitoring** tab to view email and AppTunnel alerts. The alerts include, HTTP (email and apps), Kerberos, and ActiveSync status errors. Alerts that are warning and above are shown.

TABLE 34. STANDALONE SENTRY MONITORING

Alerts Viewer	View alerts
Alert Configuration	Configure notifications

Alert Viewer

Use this page to view Sentry alerts. Sentry reports multiple types of alerts. Each alert has its own alert ID, and each alert ID is linked to an article in the Knowledge Base if one is available. The Knowledge Base article gives more information on the alert and the related error codes. For ActiveSync traffic, separate alert IDs are provided and each alert ID for ActiveSync traffic corresponds to an ActiveSync command, such as Sync, Ping, or Sendmail, supported by Standalone Sentry.

- Set the logging level at 3 or more (Sentry System Manager > **Troubleshooting > Logs**). The **Monitoring** tab will not show any alerts if logging level is 2 or less.

- The page is not automatically refreshed. Reload the page to refresh the alerts.

- Alerts for ping status 2 are also seen. Ping status 2 is not an error.

- After Ivanti Standalone Sentry is restarted, only the most recent 1000 lines will be displayed.

Filtering Ivanti Standalone Sentry alerts

Filtering allows you to narrow down the alerts to a specified set. The filter is applied only to the alerts displayed in the Alert Viewer page.

Procedure

1.   In the Ivanti Standalone Sentry System Manager, go to **Monitoring > Alert Viewer**.

2.   Enter a text string in the text box.

Regular expressions are supported. Case is ignored. Example: Error and ERROR will return the same results.

3.   Click **Filter**.

Only alerts containing the text string are displayed.

Click **Clear** to clear the filter and return the complete set of alerts.

Click **Reload** to return an updated set of alerts based on the current filter settings. Reload does not clear the filter. Refreshing the browser will clear the filter.

Alert Configuration

Use this page to configure notifications for Ivanti Standalone Sentry alerts, such as email address to which notifications are sent.

FIGURE 1. ALERT CONFIGURATION

Configuring Sentry alert notifications

Sentry alert notification is configured in the Ivanti Standalone Sentry System Manager in **Monitoring > Alert Configuration.**

Procedure

1. In the Sentry System Manager go to **Monitoring > Alert Configuration**.
2. Use the guidelines in the table to configure notifications.

Item	Description
Send Notifications	Select the checkbox to enable alert notifications.
Email List	Enter the email address to send the alerts.
	Enter multiple email addresses as a comma separated list.
Alerts Per Hour	Enter a number.
	This is the number of alerts in an hour that can be emailed to you.
	For example, if you enter the number 1, you will get one alert once every hour. If there are more than one alerts for that hour, only the first alert is emailed to you. Subsequent alerts within the hour are not emailed. The clock is reset at the top of each hour.

Item	Description
Batch Time Interval (min.)	Enter a number. Batch notifications are emailed at the interval set in this field.
Default	Select the default action for email alert notification. The default action is applied to alerts that do not have a specific email notification action configured. **Discard**: An email notification is not sent for the alerts. **Realtime Notification**: Email notification is sent immediately after the alert. Apply this action to alerts that require immediate attention. **Batch Notification**: Alerts are combined into a single email notification. Use batch notification for non-critical alerts.

3. Click **Apply.**

> ℹ️ You must also configure **Email Settings** in the Sentry System Manager to receive alert notifications by email.

Managing alert notification

Configure a notification action to a Ivanti Standalone Sentry alert in **Monitoring > Alert Notifications** in the Ivanti Standalone Sentry system manager.

Procedure
1. In the **Alert Notification Management** section, click **Add**.
2. In the **Add Alert ID** pop up, enter an **Alert ID**.
3. From the drop down List, select a notification action.

Item	Description
Discard	A notification is not sent for the alert.
Realtime Notification	Alert notification is sent in real time. Apply this action to alerts that require immediate attention.
Batch Notification	The Alert is batched into a single notification. Apply this action to non-critical, but important alerts.

4. Click **Apply**.

Command Line Interface

The Ivanti Standalone Sentry command line reference provides commands to configure many features that are also available in the Ivanti Standalone Sentry System Manager user interface.

Ivanti Standalone Sentry CLI also includes commands that are specific to a unified endpoint management (UEM) platform. The UEM platforms are:

- Ivanti EPMM

- Ivanti Neurons for MDM

If you execute a UEM platform-specific CLI command in a deployment that does not use that UEM platform, an error message displays.

Example:

```
config# debug sentry check-in
This command is not applicable for the EMM Server
config#
```

The following CLI commands are specific to Ivanti Standalone Sentry:

Purging the cache

Ivanti does not recommend purging the cache unless required for debugging purposes. These commands are accessible from CONFIG mode.

The purge features available through the CLI are described below:

TABLE 35. PURGING THE CACHE

Feature	Command
Purge the device cache	debug sentry device-cache purge all
Purge the device cache of a specific device	debug sentry device-cache purge entry *<device-id> <user-id>*
Purge the Kerberos cache	debug sentry kerberos-cache purge *<upn-string>*

- **To purge the device cache,** enter the following command:

```
debug sentry device-cache purge all
```

Example of purging the device cache:

```
config# debug sentry device-cache purge all
```

- *To purge the device cache of a specific device*, enter the following command :

```
debug sentry device-cache purge entry <device-id> <user-id>
```
   - device-id
   The id of the device for which you want information.
   - user-id
   The User associated with the device.
   Example:

```
config#debug sentry device-cache purge entry Appl7S032TF7A4S testuser2674
config#
```

- **To purge the Kerberos cache for a specific UPN,** enter the following command:

```
debug sentry kerberos-cache purge <upn-string>
```
- upn-string
  The UPN of the Kerberos user for which you want to purge the cached information.

Example of purging the cache for a Kerberos user:

```
config# debug sentry kerberos-cache purge user@ironmobile.com
Purged 1 entries from cache
```

Purging CRL cache

You can purge CRL cache entry by using CRL ID:

```
/usr/bin/curl - XPOST http://localhost:<port>/asproxy/crl-cache?action=purge&crl-id=<crl_id>
```

Logging

The logging commands are accessible from CONFIG mode. Log configuration is not persistent after a reboot.

The logging features available through the CLI are described below:

TABLE 36. LOGGING

Feature	Command
Enable Sentry logging	debug sentry log enable {device | server | both}
Specify filters	debug sentry log filter *<tag>* {device-id |device-ip | user-id | bundle-id | service-name | config-uuid} *<value>*
Specify verbosity	debug sentry log verbosity {level_1_lowest|level_2|level_3|level_4_highest}
Disable Sentry logging	no debug sentry log
Delete log filters	no debug sentry log filter *<tag>*
Disable initialization log messages	no debug sentry init-log
Enable initialization log messages	debug sentry init-log [level_1_lowest|level_2|level_3|level_4_highest]

- **To enable Sentry logging,** enter config mode and type the following command:

```
debug sentry log enable {device|server|both}
```

- device
Optional. Logging is enabled to and from devices.
- server
Optional. Logging is enabled to and from the ActiveSync Server.
- both
Optional. Logging is enabled to and from both devices and the ActiveSync Server. This value is the default if no parameter is specified.

Example of enabling logging to and from devices:

```
config# debug sentry log enable device
Successful
```

- **To create a filter,** enter the following command after logging is enabled:

```
debug sentry log filter <tag> {device-id | device-ip | user-id | bundle-id | service-
name | config-uuid} <value>
```
- tag
The tag name that you are assigning to the filter
- device-id | device-ip | user-id | bundle-id | service-name | config-uuid
The attribute—device id, device ip, or user id—that corresponds to this filter
- value
Specify the value of the attribute
Example of creating a filter for a specific user:

```
config# debug sentry log filter KensDevice user-id ksmith
Successful
```

You can verify that the filter was created by using the following command, in EXEC mode, to view a list of filters:

```
config# end
#show sentry log filter
                    TAG    ENABLED     TYPE          VALUE
            KensDevice      true    user-id         ksmith
```

- **To set the log detail level,** enter the following command after logging is enabled:

```
debug sentry log verbosity {level_1_lowest|level_2|level_3|level_4_highest}
```
- level_1_lowest
Includes HTTP response/request lines and a few supporting operational messages. level_1_lowest is the default setting.
- level_2
Includes Level 1 content, HTTP headers, and additional operational messages.
- level_3
Includes Level 1 and Level 2 content and the information associated with the messages.
- level_4_highest
Includes all available log data.
Example of setting the log detail level:

```
config# debug sentry log verbosity level_1_lowest
Successful
```

- ***To disable Sentry logging,*** enter the following command:

```
no debug sentry log
```
Example of disabling the Sentry logging:

```
config# no debug sentry log
Successful
```

- ***To delete Sentry log filters,*** enter the following command:

```
no debug sentry log filter <tag>
```
  - tag
    The tag of the filter(s) you want to disable.
  If there are multiple filters with the same tag, they will all be deleted.
  To delete or disable specific filters, use the web user interface. For information about deleting filters using the web interface, see "Deleting filters" on page 157. For information about disabling filters using the web interface, see "Disabling filters" on page 157.
  Example of deleting filters using the CLI:

```
config# no debug sentry log filter KensPhone
Successful
```

- ***To disable log messages during Standalone Sentry initialization,*** enter the following command:

```
no debug sentry init-log
```
  Example of disabling log messages during Standalone Sentry initialization:

```
config# no debug sentry init-log
Restart tomcat service to take this effective.
```

- ***To enable log messages during Standalone Sentry initialization,*** enter the following command:

```
debug sentry init-log [level_1_lowest|level_2|level_3|level_4_highest]
```
  To set log verbosity, enter one of the following options:
  - level_1_lowest
    Includes HTTP response/request lines and a few supporting operational messages.
  - level_2
    Includes Level 1 content, HTTP headers, and additional operational messages.
  - level_3
    Includes Level 1 and Level 2 content and the information associated with the messages.
  - level_4_highest
    Includes all available log data. level_4_highest is the default setting.
  Example of setting the log detail level:

```
config# debug sentry init-log level_2
Restart tomcat service to make this effective.
```

Configuring garbage collection (GC)

Garbage Collection (GC) logs are enabled by default. The GC logs are automatically added to show-tech.

TABLE 37. CONFIGURING GARBAGE COLLECTION

Feature	Command
Enable GC logging and rotation	sentry gc-log [*file-count*] [*file-size*] *file-count*: The number of GC log files to use when rotating logs. Enter a number between 1 and 100. The default is 5. *file-size*: The size of GC log file at which point the log will be rotated. Enter a file size between 8K and 300M. The default is 20M.
Disable Sentry GC logging and rotation	no sentry gc-log Requires a restart of Sentry services for changes to take effect.

Monitoring Sentry

The CLI provide commands to monitor and capture additional information about Sentry server. These commands help debug issues in complex deployments. Enter these commands in CONFIG mode.

The monitoring features include enabling, disabling, port change for packet captures, duration of packet captures, monitoring threads percentage, and monitoring trigger time.

TABLE 38. MONITORING SENTRY

Feature	Command
Enable Sentry Monitor	debug sentry monitor enable
Disable Sentry Monitor	no debug sentry monitor enable
Port change for packet captures	debug sentry monitor capture port [*port_number*] If the port number is not provided, the default port 443 is used.
Duration of packet captures performed during data collection	debug sentry monitor capture time [*capture_time*] Capture time is measured in seconds. If *capture_time* is not provided, the default capture time of 600 seconds is used.
Sentry server is monitored for the number of running threads over the threshold	debug sentry monitor threads percentage [*percentage*] If the *percentage* is not provided, the default value for percentage is 90.
Sentry monitoring triggers a Sentry restart after the configured time.	debug sentry monitor trigger [*time_in_ minutes*] If the time is not set, the default time of 600 seconds is used.

Configuring a syslog server

Configuring a remote log server to send Sentry syslog data is a two step process and requires the following:

1. "Adding a syslog server" below
2. "Enabling log data" on page 177

To view Sentry facility configuration see:

- "Displaying syslog configuration" on page 178

Adding a syslog server

To add or edit a syslog server, type the following command in CONFIG mode:

```
syslog <server> [port] <protocol> <facility> <log-level> [state]
```

To delete a syslog server, type the following command in CONFIG mode:

```
no syslog <server> [port]
```

TABLE 39. ADDING A SYSLOG SERVER

Parameter	Description
server	IP address or hostname of the syslog server.
port	Syslog server port. Use port 514 if you are adding Monitor. If the port number is not provided, the default port 514 is used.
protocol	Protocol of the syslog server. The options are: • UDP • TCP

TABLE 39. ADDING A SYSLOG SERVER (CONT.)

Parameter	Description
facility	Type of log messages sent to the syslog server. The options are: • general • health-monitor • audit
log-level	Minimum severity level of log messages to be sent. The options are: • emerg • alert • crit • err • warning • notice • info • debug CLI does not limit log-level by the facility choice.
state	State of the syslog server. The options are: • enable • disable If state is not specified, syslog is enabled by default.

Enabling log data

After adding a syslog server, you need to also enable the log data for the facility you selected for the syslog server. Sentry forwards the log data that is enabled to the syslog server. General log data is enabled by default. No additional action is required if you chose General facility when you added the syslog server.

To enable log data for the facility, enter the following command in CONFIG mode:

```
sentry {audit | health-monitor}
```

TABLE 40. ENABLING LOG DATA

Feature	Command
Enable sentry audit log data	sentry audit
Enable sentry health monitoring	sentry health-monitor
Disable sentry audit	no sentry audit
Disable sentry health monitoring	no sentry health-monitor

Displaying syslog configuration

To view syslog server facility configuration use the following commands in EXEC or PRIVILEGED mode:

TABLE 41. DISPLAYING SYSLOG CONFIGURATION

Feature	Command
Display syslog configuration	show logging
Display sentry audit configuration	show sentry audit config
Display sentry health monitoring	show sentry health-monitor

Example

```
sentry# show logging
+---------------------------+------+---------+----------------+----------+------
--
 Hostname / IP Address      + Port  + Protocol + Facility Type   + Log Level + State
+---------------------------+------+---------+----------------+----------+------
--
 app1111.auto1.mycompany.com   514    UDP        health-monitor    info
enable
```

Reporting

The CLI provides commands to support the reporting features. The commands are accessible from exec mode.

The reporting features include configuration, cache reporting for systems and devices, statistics for systems and devices, and Kerberos-related reporting.

- "Displaying Sentry configuration" below

- "Displaying information for entries in the device cache" below

- "Displaying information about Kerberos modules" on page 183

- "Displaying Sentry statistics" on page 184

- "Displaying information about servers" on page 188

- "Displaying Sentry system resources" on page 189

- "Displaying Sentry log configuration" on page 189

- "Displaying Sentry log filters" on page 190

Displaying Sentry configuration

You can request a report of the entire configuration or filter by a string of text so that the output only displays rows matching the filter-string text.

To show the Sentry log configuration, enter the following command:

```
show sentry config-properties [filter-string]
```

- filter-string
  Optional. Specify text to filter the output.

Example of a request for output of the Sentry log configuration for asproxy.client:

```
#show sentry config-properties asproxy.client
asproxy.client.port = 80
asproxy.client.tls.port = 443
```

If you do not specify a filter, the output includes all configuration information.

Displaying information for entries in the device cache

You can display information for entries in the Sentry device cache. You can also display information about the in-memory device list, the persistent device list, and connectivity to UEM.

Caution: The purpose of these commands is to assist Technical Support with troubleshooting. Do not depend on the output's format for use with any programs.

TABLE 42. DISPLAYING INFORMATION FOR ENTRIES IN THE DEVICE CACHE

Feature	Command
Display a list of entries in the Sentry device cache.	show sentry device-cache dump {all | active-sync | app-tunnel}
Display detailed information for a specific entry.	show sentry device-cache entry *<tunnel-id>* *<user-id>*
Display the entries associated with a device id	show sentry device-cache device *<device-id>*
Display the entries associated with a user id	show sentry device-cache user *<user-id>*
Display the entries with the specified number of minimum connections	show sentry device-cache min-connection *<value>*
Show the entries that need validation from UEM.	show sentry device-cache validation-pending {yes | no}
Show the status of the Sentry-device cache.	show sentry device-cache status
Show the connection status to UEM.	show sentry status

- • **To display the entries in the device cache**, type the following command:

```
show sentry device-cache dump {all | active-sync | app-tunnel}
```
- all
    Displays all entries in the cache in table format.

```
#show sentry device-cache dump all
```

- active-sync
    Displays the ActiveSync entries in the cache in table format.

```
#show sentry device-cache dump active-sync
```

- app-tunnel
    Displays the app tunnel entries in the cache in table format.

```
#show sentry device-cache dump app-tunnel
```

- • **To display detailed information about a specific entry in the device cache,** type the following command:

```
show sentry device-cache entry <tunnel-id> <user-id>
```
- tunnel-id
    The Tunnel ID of the entry in the device cache for which you want information. You can view the Tunnel-ID in the output from `show sentry device-cache dump`.
- user-id
    The User ID associated with the entry.

- ***To display all entries associated with a specific device,*** type the following command:

```
show sentry device-cache device <device-id>
```
- device-id
  The device-id for which you want to display all entries.
  If you provide a partial device id, the rows for all matching devices are displayed.

  Example of a request for a report for a specific device-id:

```
sentry# show sentry device-cache device 0V55EEVSRT5UVFA05AUBLF9O4S

S: Tunnel State {A:Allowed, B:Blocked, P:Policy Pending, W:Wipe Pending}
Vs: EMM Validation State {Y:validated, N:not-validated}
Cn: Connection count
Ver: AppTunnel Version or ActiveSync Version
Time: Timestamp of the last request or connection
Application[/ID]: Application[/ActiveSync Device ID (if application is 'ActiveSync')]
Tunnel-ID: Generic Tunnel ID


Index User          S Vs Cn Ver  Time                    Application[/ID]
          Tunnel-ID
1    testuser3351   A Y   0 14.1 2015-08-27 18:30:57 GMT
ActiveSync/0V55EEVSRT5UVFA05AUBLF9O4S 0v55eevsrt5uvfa05aublf9o4s


sentry#
```

- ***To display all entries associated with a specific user,*** type the following command:

```
show sentry device-cache user <user-id>
```
- user-id
  The user-id for which you want to display all entries.
  If you provide a partial userid, the rows for all matching users are displayed.

  Example of a request for a report for a specific user-id:

```
sentry# show sentry device-cache user testuser3351

S: Tunnel State {A:Allowed, B:Blocked, P:Policy Pending, W:Wipe Pending}
Vs: EMM Validation State {Y:validated, N:not-validated}
Cn: Connection count
Ver: AppTunnel Version or ActiveSync Version
Time: Timestamp of the last request or connection
Application[/ID]: Application[/ActiveSync Device ID (if application is 'ActiveSync')]
Tunnel-ID: Generic Tunnel ID


Index User                      S Vs Cn Ver  Time
Application[/ID]          Tunnel-ID
```

```
1     testuser3351                   A  Y   0 14.1 2015-08-27 18:30:57 GMT
ActiveSync/0V55EEVSRT5UVFA05AUBLF9O4S 0v55eevsrt5uvfa05aublf9o4s
sentry#
```

- **To display the entries with the specified minimum number of connections,** enter the following command:

```
show sentry device-cache min-connection <value>
```
- value
  The number of connections, at a minimum, for which you want to display the associated entries.
  Example of a request for a list of devices that have the minimum number of connections:

```
#show sentry device-cache min-connection 1
```

- **To display the entries that require EMM validation,** enter the following command:

```
show sentry device-cache validation-pending {yes | no}
```
  Example of a request for a list of devices that require validation from UEM:

```
#show sentry device-cache validation-pending yes
```

- **To display information about the persistent device list and the in-memory device list,** enter the following command:

```
show sentry device-cache status
```
  Example of the command:

- **To display information about the Standalone Sentry's connection to the UEM server ,** enter the following command:

```
show sentry status
```
  Example of the command and its output:

```
#show sentry status
EMM server type : Ivanti EPMM
Connectivity to Ivanti EPMM: Connected
  Last connectivity change detected by : Periodic connectivity check
  Last connectivity change time        : Fri Aug 03 18:55:16 UTC 2012

Ivanti EPMM periodic connectivity check status
  Current time     : Fri Aug 03 21:32:49 UTC 2012
  Last successful : Fri Aug 03 21:25:17 UTC 2012
  Last failed      : Never
  Next scheduled   : Fri Aug 03 21:40:17 UTC 2012

EMM server fail-open status : Allow
  Last fail-open status change detected by : Sentry initialization
```

```
Last fail-open status change time        : Tue Aug 02 23:05:08 UTC 2012
```

The Standalone Sentry detects changes to EMM connectivity in one of the following ways:
- The Standalone Sentry checks EMM connectivity on a regular basis. This is known as the periodic connectivity check.
- The Standalone Sentry checks EMM connectivity when the Sentry initializes.
- The administrator can manually check EMM connectivity by using the Verify button on the Troubleshooting > Service Diagnosis page of the Standalone Sentry Web Portal.

Displaying information about Kerberos modules

You can display the Kerberos and UPN information. The new CLI commands for Kerberos reporting are described below.

TABLE 43. DISPLAYING INFORMATION ABOUT KERBEROS MODULES

Feature	Command
Display the SPN, timeout, and cache size for Kerberos	show sentry kerberos
Display Kerberos UPN information	show sentry kerberos cache dump [*upn-filter*]
Display Kerberos information related to a specific UPN	show sentry kerberos cache upn *<upn-string>*

- **To display Kerberos information for the Sentry,** enter the following command:

```
show sentry kerberos
```
Example of a request for Kerberos information:

```
#show sentry kerberos
 sentry-spn = HTTP/sentry.company.com
 cache-ticket-idle-timeout = 48
 cache-size = 1
```

- **To display information for all UPNs in the Kerberos cache,** enter the following command:

```
show sentry kerberos cache dump [upn-filter]
```
- upn-filter
  Optional. Full or partial UPN to filter on. Shows only the rows matching this string.
  Example of a request to display Kerberos UPN information for UPNs that match the upn-filter:

```
# show sentry kerberos cache dump user
Indx User-UPN                   Created(m)  IdleTime(m)
1    user@ironmobile.com          1010         7
```

- *To display Kerberos information for a specific UPN in the Kerberos cache,* enter the following command:

```
show sentry kerberos cache upn <upn-string>
```
   - upn-string
      The full UPN of the UPN for which you want to display information.
   Example of a request to display Kerberos UPN information for a specific UPN:

```
# show sentry kerberos cache upn user@ironmobile.com
Kerberos Cache for UPN: user@ironmobile.com
   [0]user-upn = user@ironmobile.com
   [0]idle-time-min = 7
   [0]creation-time = Fri Jan 13 01:44:11 UTC 2012
```

Displaying Sentry statistics

Displays statistics for Sentry. You can specify parameters for global statistics or statistics for a specific device.

TABLE 44. DISPLAYING SENTRY STATISTICS

Feature	Description
Display Sentry statistics for a specific device	show sentry statistics entry *<device-id>* *<user-id>*
Display Sentry global statistics for a specific device	show sentry statistics global [device|system] [*filter-string*]
Display complete global Sentry system statistics	show sentry statistics global system [*filter-string*]
Display complete global Sentry statistics for a server	show sentry statistics global server [*filter-string*]

- *To display complete Sentry statistics for a specific device,* type the following command:

```
show sentry statistics entry <device-id> <user-id>
```
   - device-id
   The id of the device for which you want information.
   - user-id
   The User associated with the device.
   Example:
```
sentry#show sentry statistics entry ApplDN6FM6SZDKPH testuser2885
 d-connection = 4
 s-connections = 4
 d-bytes-rcvd = 8572
 s-bytes-sent = 8368
```

```
  s-bytes-rcvd = 2704
  d-bytes-sent = 2704
  d-http-requests = 20
  s-http-requests = 16
  s-http-responses = 16
  d-http-responses = 16
  d-http-449 = 0
  s-http-449 = 0
  permits = 16
  pendings = 0
blocks = 0

  wipes = 0
  s-http-3xx-redirects = 0
  s-http-451-redirects = 0
  d-http-451-redirect-drops = 0
  cmd-none = 0
  cmd-options = 0
  cmd-provision = 0
  cmd-sync = 0
  cmd-folder-sync = 0
  cmd-ping = 16
cmd-get-attachment = 0
cmd-item-operations = 0
cmd-unknown = 0
d-err-conn-timeout = 0
 s-err-conn-timeout = 0
 d-err-so-timeout = 0
 s-err-so-timeout = 0
 s-err-cmd-ping-timeout = 0
 s-err-cmd-sync-timeout = 0
 d-err-cmd-timeout = 0
 s-err-cmd-timeout = 0
 d-err-reset = 0
 s-err-reset = 0
 d-so-close = 0
 s-so-close = 0
http-status-200 = 0
 http-status-401 = 16
 http-status-404 = 0
 http-status-409 = 0
```

```
 http-status-5xx = 0
 http-status-other = 0
err-conn-pooling = 0
 d-unclassified = 0
 s-unclassified = 0
 ping-sync-throttled = 0
kerberos-auth-error = 0
 attachments-encrypted = 0
 attachment-encrypt-failures = 0
 attachments-converted = 0
 attachments-replaced = 0
 attachment-replaced-failures = 0
 attachments-fwd-restored = 0
 attachments-fetched = 0
 attachments-embedded = 0
 attachments-renamed = 0
 attachments-size-MB = 0
 attachments-size-bytes = 0
 decryption-failures = 0
 d-http-503-s2c = 0
 d-http-503-c2s = 0
 d-http-400-c2s = 0
active-sync-status-reports = 0
sentry#
```

- ***To display complete global Sentry statistics for devices,*** type the following command:

```
show sentry statistics global device <filter-string>
```
   - filter-string
     The full or partial string of one of the fields in the statistics report. The filter-string can either be a
     field name or a value in the field.
   Example of a request to display global statistics for devices, filtered on http-status:

```
# show sentry statistics global device http-status
 http-status-200 = 388
 http-status-401 = 32
 http-status-404 = 0
 http-status-409 = 0
 http-status-5xx = 0
 http-status-other = 0
```

- ***To display complete global Sentry system statistics,*** type the following command:

```
show sentry statistics global system [filter-string]
```
-   filter-string
    The full or partial string of one of the fields in the statistics report. The filter-string can either be a field name or a value in the field.

Example of a request to display global Sentry statistics, filtered on peak:

```
# show sentry statistics global system peak


peak-heap-mem-used-MB = 389
 peak-date-heap-mem-used-MB = Thu Aug 27 22:32:30 UTC 2015
 peak-buff-cached-mem-used-MB = 1189
 peak-date-buff-cached-mem-used-MB = Thu Aug 27 22:33:30 UTC 2015
 peak-process-virtual-mem-used-MB = 2988
 peak-date-process-virtual-mem-used-MB = Thu Aug 27 22:32:30 UTC 2015
 peak-process-resident-mem-used-MB = 1049
 peak-date-process-resident-mem-used-MB = Thu Aug 27 22:34:30 UTC 2015
 peak-cpu-% = 14
 peak-date-cpu-% = Thu Aug 27 22:32:30 UTC 2015
 peak-mem-% = 39
 peak-date-mem-% = Thu Aug 27 22:32:30 UTC 2015
 peak-running-threads = 1
 peak-date-running-threads = Thu Aug 27 22:33:21 UTC 2015
 peak-device-cache-size = 2
 peak-date-device-cache-size = Thu Aug 27 22:32:30 UTC 2015
 peak-user-url-cache-size = 0
 peak-date-user-url-cache-size = Thu Aug 27 22:32:30 UTC 2015
 peak-kerb-servtkt-cache-size = 0
 peak-date-kerb-servtkt-cache-size = Thu Aug 27 22:32:30 UTC 2015
sentry#
```

The full global statistics report can be downloaded in CSV format using the user interface. See "Sentry Statistics" on page 161.

-   *To display complete global Sentry statistics for a server*, type the following command:

```
show sentry statistics global server <filter-string>
```
-   filter-string
    The full or partial string of one of the fields in the statistics report. The filter string can either be a field name or a value in the field.

Example:

```
sentry#show sentry statistics global server
 hc-connections = 6
 hc-bytes-sent = 816
 hc-bytes-rcvd = 1116
 hc-http-requests = 6
```

```
hc-http-responses = 6
hc-err-conn-timeout = 0
hc-err-so-timeout = 0
hc-err-reset = 0
hc-so-close = 0
hc-unclassified = 0
hc-http-status-200 = 0
hc-http-status-401 = 6
hc-http-status-404 = 0
hc-http-status-other = 0
```
   Example with filter string:
```
sentry#show sentry statistics global server err
hc-err-conn-timeout = 0
hc-err-so-timeout = 0
hc-err-reset = 0
sentry#
```

Displaying information about servers

- *To display server details and connection status*, type the following command in EXEC mode:

```
show sentry server status
```
   Example:
```
sentry# show sentry server status


Current Time : Thu Aug 27 19:21:37 UTC 2015


Service Name : <ANY>
Service Type : App Tunnel
Server Scheduling : PRIORITY
Server                              Declared Last            Failure
Name/IP                             Status   Failed          Count
-----------------------------------------------------------------------
                                    Live     Never                   0


Service Name : default
Service Type : Active-Sync
Server Scheduling : PRIORITY
Active Background Health Check : Enabled
Server                              Declared Last            Last
 Failure
```

```
Name/IP                          Status    Successful          Failed
 Count
-----------------------------------------------------------------------------------
----------
ex2010sp3.enterprise.com         Live      08/27/2015 19:20:52  Never
0


Service Name : <TCP_ANY>
Service Type : App Tunnel
Server Scheduling : PRIORITY
Server                           Declared Last                 Failure
Name/IP                          Status    Failed              Count
----------------------------------------------------------------------
                                 Live      Never                      0
sentry#
```

Displaying Sentry system resources

- *To display Sentry system resources*, type the following command:

```
show sentry utilization
```
  Example:

```
sentry#show sentry utilization
Number of Connected Devices  : 0
Number of Open Connections   : 0
Thread Pool Utilization      : 0.0%
CPU Utilization              : 0%
System Memory Utilization    : 23%
Heap Memory Utilization      : 15%
sentry#
```

Displaying Sentry log configuration

You can display the Sentry log configuration. To change the log configuration, see the commands in "Logging" on page 171.

To display the Sentry log configuration, type the following command:

```
show sentry log
```

Example of a request to display log configuration information:

```
# show sentry log
    log-from-to = both
```

```
       enable = true
       verbosity = level3
```

Displaying Sentry log filters

You can display the log filters that are currently configured on Sentry. To configure the log filters, see "Logging" on page 171.

To display the Sentry log filters, type the following command:
```
show sentry log filter
```

Example of a request to display the log filters:
```
 # show sentry log filter
           TAG    ENABLED    TYPE                    VALUE

       KensPhone      true   user-id                 ksmith
```

Displaying Sentry GC log configuration

You can display the garbage collection (GC) currently configured on Sentry. To configure GC, see "Configuring garbage collection (GC)" on page 174.

To display the Sentry GC configuration, type the following command:
```
show sentry gc-log
```

Clearing the redirect URL

To clear a redirect URL from the Sentry, enter the following command:

TABLE 45. CLEARING THE REDIRECT URL

Feature	Command
Clear redirect URL	debug sentry device-cache clear-redirect-url {all | entry *<device-id> <user-id>*}

To clear the redirect URL for a specific device, enter the following CLI command:
```
debug sentry device-cache clear-redirect-url entry <device-id> <user-id>
```

- <device-id>

The device id of the device for which you want to delete the redirect URL.

- <user-id>

The User associated with the device.

Example:

```
sentry/config# debug sentry device-cache clear-redirect-url entry
Appl7S032TF7A4S testuser2674
sentry/config#
```

To clear the redirect URL for all devices, enter the following CLI command:

```
debug sentry device-cache clear-redirect-url all
```

Example

```
sentry/config# debug sentry device-cache clear-redirect-url all
```

Configuring access to UEM

To configure new device access to the UEM server when the UEM server is not reachable, enter one of the following commands:

TABLE 46. CONFIGURING ACCESS TO UEM

Feature	Command
Allow new devices to access the server.	sentry emm-fail-open
Block new devices from accessing the server.	no sentry emm-fail-open

- *To allow new devices to access the server when the UEM server is not reachable*, type the following command in CONFIG mode:

```
sentry emm-fail-open
```
Example:
```
sentry/config# sentry emm-fail-open
sentry/config#
```

- *To block new devices from accessing the server when the UEM server is not reachable*, type the following command in CONFIG mode:

```
no sentry emm-fail-open
```
Example:
```
sentry/config# no sentry emm-fail-open
sentry/config#
```

Enabling and disabling iptables

The iptables service is enabled by default. Any changes to the configuration is persistent. A write is not required to save any changes in the configuration.

> ℹ️   If the iptables service is disabled, you cannot configure ACLs in the Sentry System Manager.

To enable or disable the iptables service at system startup, enter one of the following commands in CONFIG mode

TABLE 47. ENABLING AND DISABLING IPTABLES

Feature	Command
Enable the iptables service.	service iptables enable
Disable the iptables service.	no service iptables

- *To enable the iptables service*, type the following command in CONFIG mode:

```
service iptables enable
```
   Example:

```
sentry/config# service iptables enable
sentry/config#
```

- *To disable the iptables service*, type the following command in CONFIG mode:

```
no service iptables
```
   Example:

```
sentry/config# no service iptables
sentry/config#
```

- *To view whether the iptables service is enabled or disabled at system startup*, type one of the following commands in EXEC mode:

```
show service
 or
show running-config
```
   Example:

```
sentry#show service

 +-----------+----------+---------------
  Servicename + Enabled   + Max.Sessions
 +-----------+----------+---------------
  ssh             yes         5
  ntp             yes
```

```
iptables        yes
```

Example:

```
sentry#show running-config
Display running configuration
interface GigabitEthernet 1
  ip address 10.10.27.14 255.255.0.0
  no shutdown
  end
interface GigabitEthernet 2
  no ip address
  shutdown
  end
interface GigabitEthernet 3
  no ip address
  shutdown
  end
interface GigabitEthernet 4
  no ip address
  shutdown
  end
ip route 0.0.0.0 0.0.0.0 10.10.1.1
no dbconfig
service ssh 5
service ntp
no service iptables
ip name-server 10.10.15.6 0
ip name-server 10.11.50.31 1
ip domain-name auto.ivanti.com
ntp 172.16.0.235 1
hostname app264.auto.ivanti.com
timeout 0
system user miadmin ***
sentry#
```

- *To view the iptables service status*, type the following command in EXEC PRIVILEGED mode:

```
#service iptables status
```
Example:
```
sentry# service iptables status
```

curl

A new **curl** CLI command is added to allow you to run cURL operation from EXEC Privileged mode.

The cURL features available through CLI are described in the following table:

TABLE 48. CURL COMMANDS

Feature	Command
cURL to the ActiveSync destination server	curl active-sync *\<dest\>* [*port*] [*scheme*] [*protocol*] *\<user\>*
cURL to the AppTunnel destination server	curl app-tunnel *\<dest\>* [*port*] [method] [*scheme*] [*protocol*] [*user*]

- **To cURL to the ActiveSync destination server,** enter the following command:

```
curl active-sync <dest> [port] [scheme] [protocol] [user]
```
- dest
  Required. The IP address or hostname of the destination server.
- port
  Optional. The port for the destination server. The default used is 443.
- scheme
  Optional. The HTTP scheme. Enter HTTP or HTTPS. The default used is HTTPS.
- protocol
  Optional. The SSL protocol version. Enter TLSv1, or SSLv2. The entry is ignored if SSL is not required. The default used is TLSv1.
- user
  Optional. Enter the username for the destination server.

Example:

```
#curl active-sync activesyncserver.domainname.com
```

- **To cURL to the AppTunnel destination server,** enter the following command:

```
curl app-tunnel <dest> [port] [method] [scheme] [protocol] [user]
```
- dest
  Required. The IP address or hostname of the destination server.
- port
  Optional. The port for the destination server. The default used is 443.
- method
  Optional. The HTTP method. Enter GET, HEAD, or OPTIONS. The default used is GET.
- scheme
  Optional. The HTTP scheme. Enter HTTP or HTTPS. The default used is HTTPS.

- protocol
  Optional. The SSL protocol version. Enter TLSv1, or SSLv2. The entry is ignored if SSL is not required. The default used is TLSv1.
- user
  Optional. Enter username for the destination server.

Example:

```
#curl app-tunnel appserver.domainname.com
```

Regenerating the Ivanti Standalone Sentry self-signed certificate

You can regenerate the Ivanti Standalone Sentry self-signed certificate using the command line interface (CLI). You can regenerate only the self-signed certificate or both self-signed and CA certificates.

Impact of regenerating the Ivanti Standalone Sentry self-signed certificate

Regenerating the self-signed certificate will impact email and app tunnel deployments. The self-signed certificate will have to be re-pushed to the devices.

Regenerating the CA certificate, in addition, impacts Ivanti Tunnel. For Ivanti Tunnel, the CA certificate must be manually uploaded to the device. .

How to regenerate the Ivanti Standalone Sentry self-signed certificate

To regenerate the Ivanti Standalone Sentry certificates, enter the following CLI command in configuration mode:

```
certificate {portal}
```

TABLE 49. REGENERATING THE STANDALONE SENTRY SELF-SIGNED CERTIFICATE

Feature	Command
Regenerate Standalone Sentry self-signed portal certificate	certificate portal

To regenerate Ivanti Standalone Sentry self-signed portal certificate, enter the following CLI command:

```
certificate portal
```

Example

```
config# certificate portal
Services will be disrupted.
Would you like to proceed? [y/n]:
```

If Standalone Sentry does not use a self-signed certificate

If Standalone Sentry does not use a self-signed certificate, then the `certificate {portal | sentry}` command will return the following message:

"Non Self-Signed Certificate in use. No Action performed"

Checking Kerberos Key Distribution Center (KDC) connectivity

To check connectivity and reachability to a KDC host use the following CLI command:

```
debug sentry kerberos kdc
```

This allows you to check that the port on the KDC host is reachable and ensure that the port is not blocked by firewall.

Executing the `debug sentry kerberos kdc` CLI command causes a TCP connection to the specified KDC host. If a port is not specified, the default KDC port 88 is used. The TCP connection is dropped immediately after establishing a connection without either sending or receiving any data.

Checking connectivity to a KDC host

To check connectivity to a KDC host, enter the following CLI command in configuration mode:

```
debug sentry kerberos kdc <hostname> [port]
```

- hostname

The hostname for the KDC server.

- port

The port for the KDC server. If port is not specified, the default port 88 is used.

Successful example

```
sentry/config# debug sentry kerberos kdc win2k8.acmetwo.acme.com
Connecting to KDC win2k8.acmetwo.acme.com, port 88
Connection successful.
Address: win2k8.acmetwo.acme.com/192.0.2.0:88
```

```
sentry/config#
```

Failure example

```
sentry/config# debug sentry kerberos kdc win2k8.acmeone.acme.com

Connecting to KDC win2k8.acmeone.acme.com, port 88

Connection failed.

java.net.UnknownHostException: win2k8.acmeone.acme.com

    at java.net.AbstractPlainSocketImpl.connect
(AbstractPlainSocketImpl.java:184)

    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)

    at java.net.Socket.connect(Socket.java:589)

    at java.net.Socket.connect(Socket.java:538)

    at java.net.Socket.<init>(Socket.java:434)

    at java.net.Socket.<init>(Socket.java:211)

    at com.ivanti.alcor.controller.SentryAdminController.debugKerberosKDC
(SentryAdminController.java:1085)
```

Verifying Kerberos configuration

To verify the Keberos constrained delegation (KCD) setup, use the following CLI command:

```
 debug sentry kerberos request-ticket
```

The CLI command issues a Kerberos ticket for a particular user. These tickets are issued for testing and debugging only and are not cached or reused.

TABLE 50. VERIFYING KERBEROS CONFIGURATION

Feature	Command
Request a kerberos ticket on behalf of a user with a host:port combination	debug sentry kerberos request-ticket host-port *<upn> <realm> <hostname> [port]* • *upn:* user's UPN • *realm:* user's REALM • *hostname*: backend server's hostname • *port::* backend server's port The default value for port is 443.
Request a kerberos ticket on behalf of a user with an SPN	debug sentry kerberos request-ticket spn *<upn> <realm> <spn>* • *upn:* user's UPN • *realm:* user's REALM • *spn:* service principal name

Initializing Kerberos

To initialize Keberos, use the following CLI command:

```
debug sentry kerberos init
```

The CLI command initializes Kerberos. If Kerberos is already initialized, executing the command has not impact.

Stopping and restarting Ivanti Standalone Sentry services

You can stop, start, or restart the following Ivanti Standalone Sentry services from the Ivanti Standalone Sentry command line interface (CLI):

- iptables

- tomcat

- tomcat2

Restarting a service, stops the service and then restarts the service automatically.

TABLE 51. STOPPING AND RESTARTING STANDALONE SENTRY SERVICES

Feature	Command
Stop a service	service {iptables | tomcat | tomcat2} stop
Start a service	service {iptables | tomcat | tomcat2} start
Restart a service	service {iptables | tomcat | tomcat2} restart
View service status	service {iptables | tomcat | tomcat2} status

This allows the administrator to stop, start, or restart a Standalone Sentry service without having to reboot Standalone Sentry and prevents disruption to other services that are running.

Impact of stopping and restarting Ivanti Standalone Sentry services

- Stopping or restarting tomcat2 will impact access to Ivanti Standalone Sentry system manager UI and CLI commands. The Ivanti Standalone Sentry system manager will not be available and you will be able to execute only a subset of the CLI commands.

- Stopping or restarting tomcat or iptables services will impact Ivanti Standalone Sentry traffic till the service is back up and running.

- After stopping a service through the CLI, restarting Standalone Sentry also restarts the service.

How to stop and restart Ivanti Standalone Sentry services

- ***To stop a service,*** enter the following command:

```
service [iptables | tomcat | tomcat2] stop
```

Example for stopping tomcat::

```
sentry# service tomcat stop
```

- ***To start a service,*** enter the following command:

```
service [iptables | tomcat | tomcat2] stop
```

Example for starting tomcat::

```
sentry# service tomcat start
```

- ***To restart a service,*** enter the following command:

```
service [iptables | tomcat | tomcat2] restart
```

Example for restarting tomcat::

```
 sentry# service tomcat restart
```

- **To view service status,** enter the following command:

```
service [iptables | tomcat | tomcat2] status
```

Example for viewing tomcat::

```
 sentry# service tomcat status
```

Configuring kernel parameters

To configure kernel parameters, enter the following command in CONFIG mode:

```
kparam {rp_filter | log_martians | kernel_panic | tcp_sack | tcp_keepalive_time | tcp_
keepalive_probes | tcp_keepalive_intvl} [kvalue]
```

TABLE 52. CONFIGURING KERNEL PARAMETERS

Feature	Command
Set this value to filter the kernel parameters.	kparam rp_filter [*kvalue*] *kvalue* is 0, 1 or 2.
Set this value to allow any unsigned integer value.	kparam log_martians [*kvalue*] *kvalue* is 0 or 1.
Set this value to allow safe recovery of any kernel malfunction.	kparam kernel_panic [*kvalue*] *kvalue* is an integer equal to or greater than 0.
Set this value to configure TCP SACK.	kparam tcp_sack [*kvalue*] *kvalue* is *kvalue* is 0 or 1.

Using the Splunk forwarder service

You can enable and perform other actions for the Splunk forwarder service from the Standalone Sentry command line interface (CLI).

The following table lists the commands.

TABLE 53. SPLUNK FORWARDER SERVICE CLI COMMANDS IN STANDALONE SENTRY

Action	Command	Mode
Enable the Splunk forwarder service	service splunk-forwarder enable	CONFIG
Disable the Splunk forwarder service	no service splunk-forwarder	CONFIG
Start Splunk forwarder service	service splunk-forwarder start	EXEC PRIVILIGED
Stop Splunk forwarder service	service splunk-forwarder stop	EXEC PRIVILIGED
Status of the Splunk forwarder service	service splunk-forwarder status	EXEC PRIVILIGED
Restart the Splunk forwarder service	service splunk-forwarder restart	EXEC PRIVILIGED
Verify if the service is enabled or disabled	show service	EXEC

Changing TLS protocols

To change the TLS protocol version, use the following CLI command in CONFIG mode:

```
httpd protocol protocol-list
```

You can configure the following TLS versions:

- TLSv1

- TLSv1.1

- TLSv1.2.

Enter the versions as a comma-separated list. The updates will be applied to port 8443 and 9090 only. By default, TLSv1 is disabled and TLSv1.1 and TLSv1.2 are enabled on ports 8443 and 9090.

Example:

```
sentry/config# httpd protocol tlsv1.1,tlsv1.2

Changes will issue restart of httpd service and Sentry system service might be
distrupted.

Would you like to proceed? [y/n]: y

sentry/config# do show httpd protocol

+--------+--------------------------

  Port  +  TLS Protocols Enabled

+--------+--------------------------

  8443      TLSv1.1,TLSv1.2
  9090      TLSv1.1,TLSv1.2
```

```
sentry/config#
```

Checking TLS compliance

For improved security, Ivanti recommends that TLS v1.2 is used and TLS v1.0 and v1.1 are disabled. You can check which servers that Sentry connects with support TLS v 1.2 using one of the following methods from the Standalone Sentry command line interface (CLI):

- "Using CLI command to check TLS compliance" below

- "Running TLS compliance utility" on the next page

Both methods return an OK or FAILED value for each server that is checked.

OK indicates that Ivanti Standalone Sentry is able to successfully connect with the server on TLS v1.2.

FAILED indicates that Ivanti Standalone Sentry cannot connect with the server on TLS v1.2.

The results are also recorded into a log file /var/log/TLSTrafficTool-*timestamp*.log. The log file is included in ShowTech-All. In case of failure, additional error message content as provided by OpenSSL displays and is recorded in the log file. Ivanti recommends upgrading the failed servers to support TLS v1.2.

Using CLI command to check TLS compliance

You can use a CLI command instead of the utility.

Use the following Ivanti Standalone Sentry command in EXEC PRIVILEGED mode to check TLS compliance:

```
tlscheck {all | server <server> [port]}
```

The command checks the servers that Sentry connects with and returns an OK or FAILED value for each server it checks.

To check TLS compliance for all servers that Ivanti Standalone Sentry connects with, enter the following command:

```
tlscheck all
```

To check TLS compliance for specified servers that Ivanti Standalone Sentry connects with, enter the following command:

```
tlscheck server server [port]
```

where:

- *server* is the IP address or the hostname of the server

- *port* is the port on which the server listens. If the port is not specified, 443 is used.

Running TLS compliance utility

Ivanti provides an utility that you can execute from the Standalone Sentry CLI that checks if Sentry can successfully connect with the server on TLS v1.2.

From the Ivanti Standalone Sentry command line interface, enter the following command in EXEC PRIVILEGED mode:

```
#install rpm url url_for_the_rpm
```

The command executes a script that checks the servers that Sentry connects with and returns an OK or FAILED value for each server it checks. The script uninstalls after each run.

Enabling and disabling SSL HSTS

Enabling HSTS (RFC 6797) enforces secure HTTPS connection between a web browser and Ivanti Standalone Sentry. By default, HSTS is disabled.

Before enabling HSTS ensure the following:

- Ivanti Standalone Sentry uses a root or intermediate certificate from a publicly trusted CA.

- You have policies and processes to ensure that the certificate is current.

- Port 443 is open.

Enabling SSL HSTS

To enable SSL HSTS, use the following CLI command in CONFIG mode:

```
httpd hsts enable [preload]
```

> ℹ️ HSTS Preloading is not enabled by default and it can be enabled by setting the `preload` option to `yes`. The value of the preload is either `yes` or `no` and it is `no` by default.

If SSL HSTS is enabled, the following header is added to the HTTP response:

```
Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

If SSL HSTS is enabled with the preload option set to "yes", then the following header is added to the HTTP response:

```
Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
```

Disabling SSL HSTS

To disable SSL HSTS, use the following CLI command in CONFIG mode:

```
no httpd hsts
```

> ℹ️ After disabling HSTS, also clear HSTS for the Ivanti Standalone Sentry FQDN from your browser cache. Otherwise, the browser continues to attempt to load the Standalone Sentry FQDN with a secure connection and you will not be able to access the site.

Viewing SSL HSTS

To view the current status of SSL HSTS, use the following CLI command in EXEC mode:

```
show httpd hsts
```

For more information on HSTS, see https://tools.ietf.org/html/rfc6797

Enabling and disabling web service HSTS

Enabling HSTS (RFC 6797) enforces secure HTTPS connection between web services that talk to the web browsers and Ivanti Standalone Sentry. By default, HSTS is Enabled.

Restart Tomcat for enable/disable HSTS operations to take effect.

Before enabling HSTS ensure the following:

- Ivanti Standalone Sentry uses a root or intermediate certificate from a publicly trusted CA.

- You have policies and processes to ensure that the certificate is current.

- Port 443 is open.

Use the CLI command line to run the following commands:

- To enable HSTS, use the following CLI command in CONFIG mode:

  - ```
    sentry hsts enable
    ```

- To disable HSTS, use the following CLI command in CONFIG mode:

  - ```
    no sentry hsts
    ```

- To show HSTS in the current setting, use the following CLI command in CONFIG mode:

  - ```
    do show sentry hsts
    ```

> After disabling HSTS, also clear HSTS for the Standalone Sentry FQDN from your browser cache. Otherwise, the browser continues to attempt to load the Standalone Sentry FQDN with a secure connection and you will not be able to access the site.

- To view the current status of HSTS, use the following CLI command in EXEC mode:

  - ```
    show sentry hsts
    ```

Examples

- **Enabling HSTS**

```
sentry@app431.auto.ivanti.com/config# sentry hsts enable
Are you sure you want to enable Sentry Strict-Transport-Security (HSTS rfc6797)? {yes|[no]} : yes
Please restart tomcat service for changes to take effect.
sentry@app431.auto.ivanti.com/config#
```

- **Disabling HSTS**

```
sentry@app431.auto.ivanti.com/config# no sentry hsts
Disabling Strict-Transport-Security (HSTS rfc6797) for Sentry.
Please restart tomcat service for changes to take effect.
sentry@app431.auto.ivanti.com/config#
```

For more information on HSTS, see https://tools.ietf.org/html/rfc6797.

Upgrading using CLI

The following describe the steps for upgrading the Ivanti Standalone Sentry version using CLI:

1. "Configuring the update repo" on the next page
2. "Initiating the upgrade" on the next page
3. "Rebooting Ivanti Standalone Sentry" on page 207

4. "Verifying that the upgrade is complete" on the next page

Configuring the update repo

The following procedure describes the steps to update the repo.

Procedure

1. Log into the Ivanti Standalone Sentry command line interface (CLI) using the administrator account you created during installation.
2. Enter the following command to switch to EXEC Privileged mode:
`enable`

3. Enter the password for enabling the EXEC Privileged mode.

The command line prompt changes:
`#`

4. Enter the following command to enable CONFIG mode:
`configure terminal`

5. Enter the following command to specify the URL and credentials for the repo:

`software repository https://support.ivanti.com/mi/sentry/<version>/ <username> <password>`

where <username> and <password> are your company's software download/documentation credentials as provided by Support.

For the upgrade URL, see the *Ivanti Standalone Sentry Release Notes* for the release.
   - The CLI upgrade will to fail if the trailing '/' after the version number is missing.
     Example: Enter ...`sentry/9.13.0/`
     If the trailing '/' is missing, you will see the following error message:
     `Unable to find applicable update packages in software repository. Please check URL.`
6. Enter the following command to exit CONFIG mode:
`end`

Initiating the upgrade

The following procedure describes the steps to initiate the upgrade.

Procedure

1. In EXEC Privileged mode enter the following command:
`software checkupdate`

2. Confirm that there are no errors displayed.
3. Enter the following command to download the latest available updates:
`software update`

Rebooting Ivanti Standalone Sentry

The following procedure describes the steps to reboot Ivanti Standalone Sentry.

Procedure

1. After all the listed updates are installed, in EXEC Privileged mode, enter the following command to reload the appliance:

```
reload
```

The following message displays:

```
System configuration may have been modified. Save? [yes/no]
```

2. Enter **yes**.

The following message displays:

```
Proceed with reload? [yes/no]
```

3. Enter **yes**.

Verifying that the upgrade is complete

The following procedure describes the steps to verify the upgrade.

Procedure

In EXEC Privileged mode enter:

```
 show version
```

The Ivanti Standalone Sentry version should be the version to which you upgraded.

Configuring a proxy server for upgrades

In cases where you may have a proxy server sitting between Ivanti Standalone Sentry and https://www.ivanti.com/support, you can configure the upgrade to go through the proxy server. The following table describes the commands for configuring a proxy server for software upgrades:

TABLE 54. CONFIGURING PROXY SERVER FOR SOFTWARE UPGRADES

Feature	Command in CONFIG mode
Configure a proxy server for software upgrades	software outbound-proxy *\<hostname\> \<port\>* [*username*] [*password*] • *hostname :* The hostname or IP address of the proxy server. • *port:* Port number on the proxy server for Sentry. • *username*: Username to authenticate to the proxy server if authentication is required. • *password*: Password to authenticate to the proxy server if authentication is required.
Disable proxy server configuration	no software outbound-proxy
Feature	**Command in EXEC or PRIVILEGED mode**
Display proxy server information	show software outbound-proxy

Example for configuring a proxy server for software upgrades:

```
sentry/config# software outbound-proxy proxyserver.company.com 8080
```

Upgrading multiple Ivanti Standalone Sentry

You can upgrade multiple Ivanti Standalone Sentry at the same time using the Sentry CLI.

Before you begin

• Download the upgradeConfig.json file from the support site to a location where Ivanti Standalone Sentry can access the file.

• Edit the upgradeConfig.json file with the following details:

- targetVersion: Enter the version of the Sentry that you want to upgrade to.
- emailTo: Enter the email (comma separated email IDs) to receive upgrade notifications.
- sentryGroup: Group the existing Sentrys in your deployment.
- sentryList (on-prem): Enter the FQDN of the Sentry that you want to upgrade.
- sentryList (Ivanti Neurons for MDM): Enter the IP Address for the Sentry on Ivanti Neurons for MDM that you want to upgrade.
- sentryUpgradeGroupList: Enter the Sentry groups that you want to upgrade.

Procedure

1. Log into the CLI using the administrator account you created during installation.
2. Enter the following command to switch to EXEC Privileged mode:

```
enable
```

3. Enter the password for enabling the EXEC Privileged mode.

The command line prompt changes:

```
#
```

4. Enter the following command to enable CONFIG mode:

```
configure terminal
```

5. Enter the following command to configure software autoupgrade:

```
software autoupgrade <urlstring> [username] [password]
```

where

- *upgrade URL string* is the location where the upgradeConfig.json file is hosted.
- *username* and *password* are credentials to access the upgradeConfig.json file.

6. Enter the following command to exit:

```
end
```

Related topics

- "Disabling auto-upgrade" on the next page
- "Viewing auto-upgrade details" below

Viewing auto-upgrade details

Do the following to view the auto-upgrade configuration details.

Procedure

1. Log into the Ivanti Standalone Sentry command line interface (CLI) using the administrator account you created during installation.
2. Enter the following command to switch to EXEC Privileged mode:

```
enable
```

3. Enter the password for enabling the EXEC Privileged mode.

The command line prompt changes:

```
#
```

4. Enter the following command to view auto-upgrade configuration:

```
show software autoupgrade
```

The auto-upgrade configuration details are displayed.

5. Enter the following command to exit:

```
end
```

Disabling auto-upgrade

When auto-upgrade is enabled, Ivanti Standalone Sentry checks for upgrades every thirty minutes. Do the following to disable auto-upgrade.

Procedure
1. Login to the CLI using the administrator account you created during installation.
2. Enter the following command to switch to EXEC Privileged mode:

```
enable
```

3. Enter the password for enabling the EXEC Privileged mode.

The command line prompt changes:

```
#
```

4. Enter the following command to enable CONFIG mode:

```
configure terminal
```

5. Enter the following command to disable auto-upgrade:
    ```
    no software autoupgrade
    ```

The auto-upgrade configuration details are removed.

6. Enter the following command to exit:

```
end
```