ivanti

Guide

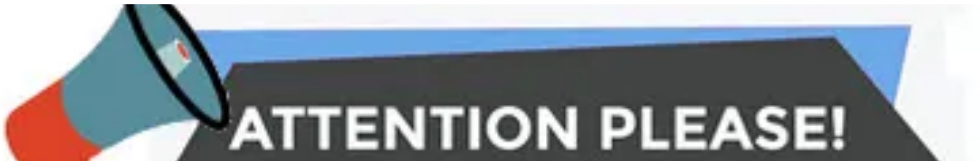
ivanti

CONTENTS

1. **Ivanti Tunnel 4.20.0 for Android Guide** **3**
2. **New features summary** **4**
3. **About Ivanti Tunnel** **5**
   - 3.1. Overview 6
   - 3.2. Ivanti Tunnel for Android support on UEM 7
   - 3.3. Ivanti Tunnel configuration 8
   - 3.4. Tunnel creation with Ivanti Tunnel for Android 9
   - 3.5. App traffic allowed through Ivanti Tunnel VPN (Android native and Android Enterprise) 10
   - 3.6. Send and receive IP packets with Ivanti Tunnel 12
   - 3.7. Support for anti-phishing with Mobile Threat Defense 13
   - 3.8. Controlling VPN traffic 16
4. **Setting up Ivanti Tunnel for Android native** **18**
   - 4.1. Before you configure Ivanti Tunnel for Android native (Ivanti EPMM and Ivanti Neurons for MDM) 19
   - 4.2. Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti EPMM) 21
   - 4.3. Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti Neurons for MDM) 23
5. **Setting up Ivanti Tunnel for Samsung Knox in Ivanti EPMM** **25**
   - 5.1. Before you configure Ivanti Tunnel for Samsung Knox 26
   - 5.2. Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM) 28
   - 5.3. Configuring an IP_ANY AppTunnel service on a Standalone Sentry 31
   - 5.4. Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace (Ivanti EPMM) 32
   - 5.5. Distributing Ivanti Tunnel through Apps@Work 34
   - 5.6. Configuring app VPN in the Samsung Knox container 35
   - 5.7. Configuring VPN chaining 37
6. **Setting up Ivanti Tunnel for Android Enterprise** **38**
   - 6.1. Before you configure Ivanti Tunnel for Android Enterprise (Ivanti EPMM and Ivanti Neurons for MDM) 39
   - 6.2. Configuration tasks overview for Android Enterprise (Ivanti EPMM) 41
   - 6.3. Configuration tasks overview for Android Enterprise (Ivanti Neurons for MDM) 44
7. **Ivanti Tunnel Configuration Fields and Custom Data** **48**
   - 7.1. Ivanti Tunnel for Android native configuration field description 49
   - 7.2. Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace 53
   - 7.3. Ivanti Tunnel configuration field description for Android Enterprise 58
   - 7.4. Example showing the Sentry certificate in the certificate chain 65
8. **Ivanti Tunnel for Android device user experience** **68**
   - 8.1. Ivanti Tunnel installation on devices 69
   - 8.2. Troubleshooting 73
   - 8.3. Anti-phishing enabled user experience 86

1. Ivanti Tunnel 4.20.0 for Android Guide

April 2026



New features summary

2. New features summary

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

Ivanti Tunnel 4.20.0 - New features summary

- **Support for SDK version 5.8.53**: Ivanti Tunnel is upgraded to Zimperium SDK Version 5.8.53.

Ivanti Tunnel 4.19.0 - New features summary

- **16 KB memory support**: Ivanti Tunnel now supports 16 KB page sizes on 64-bit devices.

3. About Ivanti Tunnel

The following provide an overview of Ivanti Tunnel for Android devices:

- Overview
- Ivanti Tunnel for Android support on UEM
- Ivanti Tunnel configuration
- Tunnel creation with Ivanti Tunnel for Android
- App traffic allowed through Ivanti Tunnel VPN (Android native and Android Enterprise)
- Send and receive IP packets with Ivanti Tunnel
- Support for anti-phishing with Mobile Threat Defense

3.1. Overview

Ivanti Tunnel enables VPN capability on Android (native, with no containerization), Android Enterprise (containerized, previously called Android for Work) and Samsung Knox Workspace (containerized) devices.

Ivanti Tunnel interacts with the Unified Endpoint Management (UEM) platform, Standalone Sentry, and Access to allow apps and browsers on Android, Android Enterprise, and Samsung Knox devices to securely access enterprise resources from outside the enterprise network. The enterprise resource can be on premise or in the cloud. It provides the following UEM platforms: Ivanti EPMM and Ivanti Neurons for MDM.

3.2. Ivanti Tunnel for Android support on UEM

The following table describes Ivanti Tunnel for Android support on the Unified Endpoint Management (UEM) platform.

	Ivanti EPMM	Ivanti Neurons for MDM
Android native	Supported	Supported
Android enterprise	Supported	Supported Only work profile and work managed device modes are supported.
Samsung Knox	Supported	Not supported

3.3. Ivanti Tunnel configuration

Configurations for Ivanti Tunnel are created in a Unified Endpoint Management (UEM) platform, which are Ivanti EPMM and Ivanti Neurons for MDM. Ivanti Tunnel receives the configuration from the UEM client. The client for Ivanti EPMM is Mobile@Work, and the client for Ivanti Neurons for MDM is Go.

If you configure Ivanti Tunnel for Android native and Android Enterprise on the same Ivanti EPMM, ensure that the configurations are applied to the correct labels.

3.4. Tunnel creation with Ivanti Tunnel for Android

The following describes how a tunnel session with Ivanti Tunnel for Android is created:

1. Tunnel validates the configuration syntactically.
2. Tunnel establishes a TCP connection with Standalone Sentry on port 443.
3. Ivanti Tunnel and Standalone Sentry mutually authenticate each other using TLS 1.2 using client identity certificates. The Android TLS stack is used for this purpose.
4. Standalone Sentry's certificate presented in the TLS handshake is compared with the Standalone Sentry certificate in the Ivanti Tunnel configuration. This step occurs if certificate pinning is enabled.
5. Tunnel initiates the AppTunnel protocol handshake:
    1. POST with device ID, user ID, and service ID are sent to Standalone Sentry.
    2. Standalone Sentry validates the parameters. For example, Standalone Sentry checks if the user or device is blocked.
    3. Standalone Sentry provides additional configuration parameters: interface IP and DNS server IP.
    4. The TCP connection is switched to the Tunnel protocol.
6. A VPN session is created using Android API VpnService.Builder.
    1. VPN specific configuration is set in the VPN session based on the Ivanti Tunnel configuration created in UEM.
    2. Android creates a TUN interface and the VPN icon is set in the system bar. The VPN icon indicates that the tunnel is established and available. The VPN icon (looks like a key for Android native and Android Enterprise, and like a lock for Samsung Knox) in the status bar indicates that the Tunnel session is available. It does not indicate if traffic from an app currently being used is going through the tunnel. The behavior is similar to that of the Wi-Fi icon.

        > Device users may also see the Tunnel notifications icon, which looks like the Ivanti Tunnel logo. The Tunnel notifications icon does not indicate that Ivanti Tunnel VPN is on. It only indicates that the there are notifications from Ivanti Tunnel.

Traffic from an app is automatically tunneled through Ivanti Tunnel irrespective of when an app is installed. The app may have been installed before Ivanti Tunnel was initiated or after Tunnel was initiated.

3.5. App traffic allowed through Ivanti Tunnel VPN (Android native and Android Enterprise)

When a Tunnel VPN session is created, the Ivanti Tunnel configuration is provided to the Android operating system. The Ivanti Tunnel configuration includes information such as allowed and disallowed apps, routes, and domain name servers. Android enforces access to Ivanti Tunnel, based on the provided configuration. The apps that use Ivanti Tunnel is determined by the allowed and disallowed configuration. You configure either an allowed list or a disallowed list.

- Allowed: Only the apps that are on the allowed list (whitelist) have access to Ivanti Tunnel. Traffic from all other apps is not allowed to go through Ivanti Tunnel and goes through the device network.
- Disallowed: All apps have access to Ivanti Tunnel, except the ones on the disallowed list (blacklist). Traffic from the disallowed list goes through the device network.

Ensure that you have configured either an allowed app list or a disallowed appllowed list is not configured, Ivanti recommends adding at least the following to a disallowed list to avoid OS traffic going through Ivanti Tunnel VPN:

- Mobile@Work if your UEM is Ivanti EPMM (com.mobileiron)
- MobileIron Go if your UEM is Ivanti Neurons for MDM (com.mobileiron.anyware.android)
- Android play store (com.android.vending)
- Google Play Service (com.google.android.gms)
- Carrier Service (com.google.android.ims)
- (For Samsung devices) Samsung Experience Service (com.samsung.android.mobileservice)"

In addition, the following also determine how an app uses Ivanti Tunnel:

- **Tunnel routes and Ivanti Tunnel for Android**
- **DNS servers and Ivanti Tunnel for Android**
- **Always-on Tunnel VPN and Ivanti Tunnel for Android**
- **Connection recovery for Ivanti Tunnel for Android**

Tunnel routes and Ivanti Tunnel for Android

During the creation of the VPN session, configured routes are set to the TUN interface. If the administrator did not configure any routes in Ivanti Tunnel configuration, Tunnel uses 0.0.0.0/0. The configured routes are used in the following ways:

- Only traffic from apps that can use Ivanti Tunnel goes through the configured routes.
- You cannot configure a different set of routes for different allowed apps.
- Traffic from non Android Enterprise apps or to disallowed Android Enterprise apps does not go through the routes configured for Ivanti Tunnel.

DNS servers and Ivanti Tunnel for Android

DNS requests coming from allowed apps are resolved by the domain name servers (DNS) configured for the VPN during the VPN creation session. These servers are different from the DNS for the original Wi-Fi or cellular connection.

In addition, the Ivanti Tunnel SplitDomain feature allows you to use two different domain name servers to resolve DNS requests, based on the requested domain. The two domain name servers typically are the DNS configured for the device network and the DNS

configured for VPN.

Always-on Tunnel VPN and Ivanti Tunnel for Android

On Android 5 and 6 devices, always-on is an Ivanti implementation. The feature is enabled by default. You can configure by using the key appRunningCheckIntervalSec, which configures the check interval.

On Android Enterprise devices running Android N (7.0) and through the most recently released version as supported by Ivanti, Google provides the always-on feature. You can configure the Google implementation of always-on VPN in the Android Enterprise (Android for Work) configuration in Ivanti EPMM and in the Always-on configuration in Ivanti Neurons for MDM.

Connection recovery for Ivanti Tunnel for Android

If a connection fails, Ivanti Tunnel tries to reconnect periodically, by default. Ivanti Tunnel makes three quick attempts at one-second intervals, and then at one-minute intervals. Ivanti Tunnel attempts to reconnect when there is a network status change or there is a configuration change. Ivanti Tunnel will also attempt to reconnect if Standalone Sentry times out due to TCP idle time. If Ivanti Tunnel is idling, Standalone Sentry closes the TCP connection. In this case, Ivanti Tunnel will attempt to reconnect. The recommended idle timeout is one hour.

You can configure connection recovery using the following keys: quickRetryMaxAttempts, quickRetryIntervalSec, slowRetryIntervalSec.

3.6. Send and receive IP packets with Ivanti Tunnel

The following describes how IP packets are sent and received between the app attempting to connect to a backend resource and Standalone Sentry:

1. The Android app posts the IP packets to the TUN interface.
2. The Tunnel plugin/service receives the IP packets from the TUN interface.
3. The packets are sent as payload of the TCP connection to Standalone Sentry.
4. Standalone Sentry sends the IP packets to the end destination.
5. Standalone Sentry receives IP packets from the end destination and sends the packets over the TCP connection to the Ivanti Tunnel plugin and posts it to the TUN interface.
6. The app gets the payload through the TUN interface.

TCP and UDP are supported. IPv4 is supported.

> Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.

Ivanti Tunnel connectivity probe

Ivanti Tunnel sends probes with VPN traffic, after a specified period of idle time, to check if the Tunnel connection to the VPN server is open. If Ivanti Tunnel does not receive a response for at least one of the probe packets, Tunnel closes the current connection and initiates a new connection to the VPN server. The following key-value pairs are available to allow administrators to customize the settings: AtpProbeIdleSec, AtpProbeIntervalSec, AtpProbeCount. For information about the key-value pairs, see **Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace**.

3.7. Support for anti-phishing with Mobile Threat Defense

Ivanti Tunnel VPN supports anti-phishing with a Mobile Threat Defense (MTD) setup that is enabled for anti-phishing.

You must have the following set up:

- On the Threat Management Console, an Anti-phishing policy.
- On a Unified Endpoint Management (UEM) platform, an anti-phishing policy with the Use VPN to analyze malicious URLs option selected.
- On the UEM, add Ivanti Tunnel for distribution to devices.
- Anti-phishing support is provided with Ivanti Tunnel for Android native and Android Enterprise. Samsung Knox is not supported.
- Anti-phishing checks are not applied to Sentry and Access traffic.
- Enable the Phishing Local Classifier and Remote classifier flags in the group on the Mobile Threat Management Console. Otherwise, the detection of phishing URLs is inconsistent across different types of Android devices.

For information about setting up anti-phishing with Threat Defense, see "Advanced phishing protection for managed devices" in the Mobile Threat Defense Solution Guide for Ivanti Neurons for MDM or Ivanti Threat Defense Solution Guide for Ivanti EPMM.

Ivanti Neurons for MDM deployments — See the following:

- **Adding and configuring the Ivanti Tunnel app for Android native (Ivanti Neurons for MDM)**
- **Adding and configuring the Ivanti Tunnel app for Android Enterprise(Ivanti Neurons for MDM)**

Ivanti EPMM deployments — See the following:

- **Distributing through the app storefront (Ivanti EPMM)**
- **Adding and configuring the Ivanti Tunnel for Android Enterprise (Ivanti EPMM)**

Ivanti Tunnel for Android native

No additional Tunnel configurations are required for supporting anti-phishing. However, make sure to add the Ivanti Tunnel app to UEM and distribute the app to devices.

A default Ivanti Tunnel VPN configuration that supports anti-phishing is automatically available in the UEM. On Ivanti EPMM, the default Ivanti Tunnel configuration is available when you upgrade to Ivanti EPMM 11.0.0.0. If the option Use VPN to analyze malicious URLs is enabled in the anti-phishing policy on UEM, the default Ivanti Tunnel VPN configuration is automatically distributed to the devices to which the anti-phishing policy is distributed.

If you already have Ivanti Tunnel for Android native deployment using a custom Tunnel VPN configuration, the configuration is automatically updated to consume the MTD license and keys.

After initially deploying Ivanti Tunnel VPN to use for anti-phishing with Mobile Threat Defense, if you also want to deploy Ivanti Tunnel VPN to use with Access or Sentry, create a custom Tunnel configuration as described in one of the following sections. Use the link appropriate to your UEM:

- **Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti Neurons for MDM)**
- **Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti EPMM)**

The custom configuration automatically consumes the Threat Defense license and key and is also automatically distributed to devices applied to the Threat Defense label. The custom Tunnel configuration is replaces the default Tunnel configuration on devices.

Ivanti Tunnel for Android Enterprise

For Tunnel for Android Enterprise deployments, in addition to the anti-phishing policy configurations on the Threat Management Console and the UEM, do one of the following:

- For a new Tunnel for Android Enterprise deployment, select the Use Tunnel for Anti-phishing only option when configuring Tunnel for Android Enterprise on the UEM.
- If you already have a Tunnel for Android Enterprise deployment, in the Configuration Choices section of the Ivanti Tunnel configuration, add a new configuration and select Use Tunnel for Anti-phishing only.

See one of the following for information about configuring and distribution Tunnel for Android Enterprise. Use the link appropriate to your UEM:

- **Configuration tasks overview for Android Enterprise (Ivanti Neurons for MDM)**
- **Configuration tasks overview for Android Enterprise (Ivanti EPMM)**

Ivanti Tunnel deployment options for anti-phishing with MTD

You can deploy Tunnel VPN for anti-phishing with Mobile Thread Defense (MTD) only or in addition to Tunnel VPN with Sentry and Access. Any combination of the following Tunnel VPN deployments are supported:

- Data traffic to enterprise resources: This requires a Sentry deployment.
- Authentication traffic to enterprise cloud resources: This requires a Access deployment.
- Analyze phishing URLs: This requires an MTD deployment.

> Anti-phishing checks are not applied to Sentry and Access traffic.

Ivanti Tunnel VPN and MTD deployment anti-phishing blocking behavior

The following table provides some Ivanti Tunnel VPN and MTD deployment scenarios and the corresponding anti-phishing blocking behavior.

Ivanti Tunnel VPN + MTD deployment scenario	Anti-phishing blocking behavior
• Route is not configured in Tunnel configuration. The default route is 0.0.0.0/0.	• All traffic is routed to Sentry. • Phishing URLs are not blocked.
• Route is configured in Tunnel configuration. For example, 10.0.0.0/8.	• Traffic in the range 10.0.0.0/8 is routed to Sentry. • Phishing URLs are not blocked for traffic going to Sentry. • Phishing URLs for traffic that is not in the range 10.0.0.0/8 is blocked.
On Android Enterprise in Work Profile mode, in Tunnel configuration, • Route is not configured. The default route is 0.0.0.0/0. • A web browser is configured in the allowed app list. For example, Chrome. • Another web browser is available in the container, but is not configured in the allowed or disallowed app list. For example, Firefox.	• Phishing URLs from Chrome are not blocked. • Phishing URLs from Firefox are blocked.

3.8. Controlling VPN traffic

Ivanti Tunnel VPN on Android native and Android Enterprise devices is always on. App traffic is allowed or disallowed based on the allowed (whitelist) or disallowed (blacklist) list, and the routes the administrator sets up in the Ivanti Tunnel VPN configuration.

The following table compares the behavior between Ivanti Tunnel for Android versus Ivanti Tunnel for iOS.

Function	Behavior on Android	Behavior on iOS
Activating Ivanti Tunnel	When Ivanti Tunnel is first launched on Android native devices, device users must accept the Ivanti Tunnel VPN connection and allow access to the Tunnel certificate. This is not applicable to Android Enterprise and Samsung KNOX devices.	If the Ivanti Tunnel VPN profile is installed on your device, the Ivanti Tunnel VPN connection is automatically turned on when you tap a supported managed app and the app attempts to connect to a backend resource. In rare cases, if the VPN connection is not turned on, you can manually turn on VPN in the Ivanti Tunnel app. Your IT administrator will tell you if you need to turn on VPN in the Tunnel app.
Automatic Tunnel triggering	By default, Ivanti Tunnel VPN is always on for Android native and Android Enterprise. User action is not required after the initial activation. If the user disables Tunnel, Tunnel is not triggered automatically. Users must re-enable Tunnel. In the Knox container, on-demand VPN is triggered by managed apps.	Managed apps or Safari domains can automatically trigger a Tunnel VPN session.
Allowing app traffic	Admin must create an allowed list or create an exclusion list to allow or block app traffic.	Admin must make apps managed and assign them Tunnel to enable traffic through Ivanti Tunnel.
Domain name triggers	Ivanti Tunnel VPN is always on. There is no triggering of VPN on Android devices.	Safari can trigger Tunnel using domain names.
Per-app allow/block list	No per-app information is sent to Standalone Sentry. Sentry cannot enforce allow/block lists at a per-app level.	Ivanti Tunnel sends per-app information to Sentry. Sentry can enforce blocking at a per-app level.
Notifications	Ivanti Tunnel can provide notifications to users for various events (connect/disconnect, allow/block).	When the device is out of compliance, per-app Ivanti Tunnel VPN cannot provide notifications to the user if traffic is blocked.
UDP support	Standalone Sentry supports only limited types of UDP traffic,such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.	Standalone Sentry supports only limited types of UDP traffic,such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.
ICMP support	ICMP is not supported.	ICMP is not supported.
IPv6	IPv6 is not supported.	IPv6 is not supported.

4. Setting up Ivanti Tunnel for Android native

The following address the setup required for app VPN using Ivanti Tunnel for Android native in Ivanti EPMM and Ivanti Neurons for MDM:

- Before you configure Ivanti Tunnel for Android native (Ivanti EPMM and Ivanti Neurons for MDM)
- Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti EPMM)
- Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti Neurons for MDM)

4.1. Before you configure Ivanti Tunnel for Android native (Ivanti EPMM and Ivanti Neurons for MDM)

Before you configure Ivanti Tunnel for Android native, ensure that you have met the requirements and have read the recommendations and limitations listed in this section.

- Required components for Ivanti Tunnel for Android native
- Requirements for Ivanti Tunnel for Android native
- Recommendations for Tunnel for Android native
- Limitations for Ivanti Tunnel for Android native

Required components for Ivanti Tunnel for Android native

The following components are required for Ivanti Tunnel deployment on Android native devices:

- Standalone Sentry with AppTunnel enabled or Access.
- Unified Endpoint Management (UEM) platform: Ivanti EPMM or Ivanti Neurons for MDM.
- Client for Android:
  - Ivanti EPMM: Mobile@Work
  - Ivanti Neurons for MDM: Go

For supported versions see the *Ivanti Tunnel for Android Release Notes* for this release.

Requirements for Ivanti Tunnel for Android native

The following are requirements for deploying Ivanti Tunnel for Android native:

- If your deployment uses Standalone Sentry:
  - You must have installed Standalone Sentry. See the *Standalone Sentry Installation Guide*.
  - To allow Android 7 devices to use Ivanti Tunnel, Standalone Sentry must use a publicly trusted CA certificate.
  - Standalone Sentry must be set up for AppTunnel using identity certificates for device authentication.
  - For information about setting up a Standalone Sentry for AppTunnel, see *Sentry Guide for Ivanti EPMM* and *Sentry Guide for Ivanti Neurons for MDM*.
- If your deployment uses Access, ensure that Access is set up.
  See the *Access Guide* for information on how to set up Access.
- Ensure that the appropriate ports are open.
  See the *Ivanti Tunnel for Android Release Notes*

Recommendations for Tunnel for Android native

The following are recommendations for deploying Ivanti Tunnel for Android native:

- Ivanti recommends that Standalone Sentry use a publicly trusted CA certificate. Android version 7 through the latest versions as supported by Ivanti does not accept self-signed certificates.

- If access to the ActiveSync server is going through Standalone Sentry, configure Tunnel so that email clients are excluded from being routed through Tunnel.

Limitations for Ivanti Tunnel for Android native

The following are limitations of Ivanti Tunnel for Android native:

- Front-end load balancer to Standalone Sentry is expected to work but has not been tested.
- Performance depends on the apps using Standalone Sentry. As a best practice, monitor Standalone Sentry usage and add more Standalone Sentry servers a
- Server authentication through Standalone Sentry with Kerberos is not supported.
- Standalone Sentry supports only limited types of UDP traffic,such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.

4.2. Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti EPMM)

The following configuration tasks are required to set up Ivanti Tunnel. These configuration tasks are performed in the Ivanti EPMM Admin Portal.

1. **Creating Ivanti Tunnel VPN configuration (Ivanti EPMM)**.
2. **Distributing through the app storefront (Ivanti EPMM)**.

Before you begin

- If you are configuring app VPN, you must have created an IP_ANY AppTunnel service in Standalone Sentry. For information on setting up an IP_ANYTunnel service see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti EPMM.
- Ensure that you have created a certificates enrollment setting in Ivanti EPMM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.

Creating Ivanti Tunnel VPN configuration (Ivanti EPMM)

Create Ivanti Tunnel (Android) VPN configuration in the Ivanti EPMM Admin Portal.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Policies & Configs > Configurations.
2. Click Add New > VPN.
3. Enter a name and description for the VPN settings.
4. For Connection Type, select Tunnel (Android).
5. Add the necessary configurations and click Save.
6. Apply the appropriate label to the app to distribute it to Android devices.

Next steps

If you are distributing the app through the app storefront, go to **Distributing through the app storefront (Ivanti EPMM)**.

Related topics

- **Ivanti Tunnel for Android native configuration field description**.
- **Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace**.

Distributing through the app storefront (Ivanti EPMM)

Ivanti Tunnel can be added to the app storefront for distribution.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Apps > App Catalog.
2. Click Add+.
3. Click Google Play.
4. Enter "Tunnel" for Application Name, and click Search.

5. Select the line for Tunnel app.
6. Click Next.
7. Select “5.0” for Min. OS Version.
8. Click Next.
9. Select Silently Install.
10. Click Finish.
11. Apply the appropriate label to the app to distribute it to Android devices.

Related topics

For more information on adding and editing Android apps to the app catalog, see “Managing Mobile apps for Android” in the *Apps@Work Guide*.

4.3. Configuration tasks overview for Ivanti Tunnel for Android native (Ivanti Neurons for MDM)

The following configuration tasks are required to set up Tunnel. These configuration tasks are performed in Ivanti Neurons for MDM.

1. **Creating Ivanti Tunnel VPN configuration for Android native (Ivanti Neurons for MDM)**
2. **Adding and configuring the Ivanti Tunnel app for Android native (Ivanti Neurons for MDM)**

Before you begin

- If you are configuring app VPN, ensure the following:
  - You have created a Tunnel service for Android in Standalone Sentry. For information on setting up Standalone Sentry with a Tunnel service, see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti Neurons for MDM.
  - Standalone Sentry is set up to use identity certificates for device authentication.
  - You have created an Identity Certificate configuration in Ivanti Neurons for MDM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.
- If you are configuring Ivanti Tunnel for securing authentication traffic with Access, ensure that you have setup Access. For information about setting up Access see the *Access Guide*. As part of the Access setup, you will have created a Tunnel service.

Creating Ivanti Tunnel VPN configuration for Android native (Ivanti Neurons for MDM)

Create a Tunnel VPN configuration in Configurations.

Procedure

1. In Ivanti Neurons for MDM, go to Configurations > +Add.
2. Search for Ivanti Tunnel.
3. Click the Tunnel configuration.
   The Create Tunnel Configuration page displays.
4. Enter a name for the configuration and click Android.
   The configuration fields for Tunnel VPN for Android are displayed.
5. Add the necessary configurations and click Next.
6. Choose a distribution option for the configuration and click Done.
   The configuration is distributed to the subset of the devices to which the app is distributed. Select the same distribution option that you selected for the Tunnel for Android app.

Next steps

Adding and configuring the Ivanti Tunnel app for Android native (Ivanti Neurons for MDM).

Related topics

- For a description of the configuration fields, see **Ivanti Tunnel for Android native configuration field description**.
- For a description of the key-value pairs, see **Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace**.

Adding and configuring the Ivanti Tunnel app for Android native (Ivanti Neurons for MDM)

Upload the Tunnel app to Ivanti Neurons for MDM from Google Play and configure it to make it available to Android devices. You can download the app from Google Play.

Procedure

1. In the Ivanti Neurons for MDM portal, go to Apps >App Catalog.
2. Click +Add next to App Catalog.
3. Select **Google Play** from the catalog pulldown menu.
4. Use the search to locate the Tunnel app in the Google Play store.
5. Select the Ivanti Tunnel app and click **Next**.
   A description and screen shots of the app are displayed.
6. Make changes, as needed, and click Next.
7. Select an app delegation option, and click Next.
8. Select a distribution option and click Next.
   The configuration will be distributed to the devices in the group you selected.
9. Click Install Application configuration settings to configure the install options.
   1. Edit the Name and Description of the settings if necessary.
   2. Install on Device: Enable Install on devices, if you want to require that the app is installed on devices.
   3. Silently install on Samsung KNOX and Zebra devices: This option is not applicable to Android native apps.
   4. Do not show app in end user App Catalog: Select if you do not want the app displayed in the app catalog on users' devices.
10. Click Next.
11. Click Promotion distribution configuration settings and select a promotion option.
    The promotion option determines how the app appears in the app catalog on the device.
12. Click Next and then click Done.

Related topics

See the *Ivanti Neurons for MDM Guide* or help for more information on adding apps to the Ivanti Neurons for MDM app catalog.

5. Setting up Ivanti Tunnel for Samsung Knox in Ivanti EPMM

The following address the setup required for app VPN using Ivanti Tunnel for Samsung Knox Workspace in Ivanti EPMM:

- Before you configure Ivanti Tunnel for Samsung Knox
- Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM)
- Configuring an IP_ANY AppTunnel service on a Standalone Sentry
- Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace (Ivanti EPMM)
- Distributing Ivanti Tunnel through Apps@Work
- Configuring app VPN in the Samsung Knox container
- Configuring VPN chaining

5.1. Before you configure Ivanti Tunnel for Samsung Knox

Before you configure Tunnel, ensure that you have met the requirements and have read the recommendations and limitations listed in this section.

- Required components for Ivanti Tunnel for Samsung Knox
- Requirements for Ivanti Tunnel for Samsung Knox
- Recommendations for Ivanti Tunnel for Samsung Knox
- Limitations for Ivanti Tunnel for Samsung Knox

Required components for Ivanti Tunnel for Samsung Knox

The following components are required for deploying Ivanti Tunnel for Samsung Knox:

- Standalone Sentry with AppTunnel enabled.
- Ivanti EPMM with the following:
  - Enabled for Samsung Knox. Ensure that the Samsung general policy is configured with the license for Samsung Knox.
  - Users have Samsung Knox-capable device.
- Ivanti Tunnel for Android.
- Android client: Mobile@Work.

> Ivanti Tunnel and Mobile@Work for Android are available from the Google Play store.

For supported versions see the *Ivanti Tunnel for Android Release Notes* for this release.

Requirements for Ivanti Tunnel for Samsung Knox

The following are required for deploying Ivanti Tunnel for Samsung Knox:

- Set up Ivanti EPMM for Samsung Knox. For more information, see the "Samsung Knox support" section in the *Ivanti EPMM Device Management Guide for Android*.
- Install Standalone Sentry. See the *Standalone Sentry Installation Guide*.
- Set up Standalone Sentry for AppTunnel using identity certificates for device authentication.
  For information about setting up a Standalone Sentry for AppTunnel, see the "Working with Standalone Sentry for AppTunnel" section in the Sentry Guide for Ivanti EPMM.
- Add the apps that will use the Ivanti Tunnel VPN to the app catalog on Ivanti EPMM and to the Samsung Knox container.
  For information about adding apps to the Ivanti EPMM app catalog see the "Adding Google Play apps for Android" and "Apps on Samsung Knox devices" sections in the *Ivanti EPMM Apps@Work Guide*.

Recommendations for Ivanti Tunnel for Samsung Knox

Android 7 devices do not accept self-signed certificates. Therefore, Ivanti recommends that Standalone Sentry use a publicly trusted CA certificate.

Limitations for Ivanti Tunnel for Samsung Knox

The following are limitations of Ivanti Tunnel for Samsung Knox:

- Front-end load balancer to Standalone Sentry is expected to work but has not been tested.
- Performance depends on the applications using Standalone Sentry. As a best practice, monitor Standalone Sentry usage and deploy additional Sentry servers as needed for horizontal scaling.
- The Certificate Enrollment created for Standalone Sentry setup for AppTunnel must use RSA key length 2048 due to a Knox limitation.
- Routes configured in the Knox VPN configuration in Ivanti EPMM are ignored by Samsung Knox Workspace. Route lists are not supported in the Knox Workspace. All traffic from an app that uses Ivanti Tunnel VPN goes over Tunnel.
- Server authentication through Standalone Sentry with Kerberos is not supported.
- Standalone Sentry supports only limited types of UDP traffic,such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.

5.2. Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM)

Configuration for Ivanti Tunnel VPN is done in the Ivanti EPMM Admin Portal. Do the following to setup Ivanti Tunnel in the Samsung Knox container:

1. **Configuring an IP_ANY AppTunnel service on a Standalone Sentry**.
2. **Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace (Ivanti EPMM)**.
3. **Distributing Ivanti Tunnel through Apps@Work**.
4. **Configuring app VPN in the Samsung Knox container**.

The VPN configuration for Ivanti Tunnel is done in two separate configurations in the Ivanti EPMM Admin Portal: the VPN configuration for Tunnel (Samsung Knox Workspace) and the Samsung KNOX Container configuration. The Ivanti Tunnel for Samsung Knox workspace VPN configuration sets the DNS and app behavior. The Samsung Knox container configuration determines whether the VPN configuration is applied per-app individually or to all apps in the container (per-container).

The VPN configuration also determines whether the connection is always-on or on-demand. With always-on VPN, Ivanti Tunnel is started when the Samsung Knox Workspace starts, and the connection stays on. Traffic from an app in the Knox Workspace can go through the Ivanti Tunnel VPN. With on-demand VPN, a Tunnel VPN connection is started when an app that uses Tunnel is launched, and the connection stays on till the last app that can use the Tunnel VPN is killed.

> Ivanti Tunnel must be available in the Samsung Knox Workspace. Sometimes an app can be available in the Knox container as well as outside the container. Only the app in the Knox container can use Tunnel.

The following table describes Tunnel behavior depending on the combination of whether VPN is on-demand or always-on and if the VPN configuration is applied per-app or per-container.

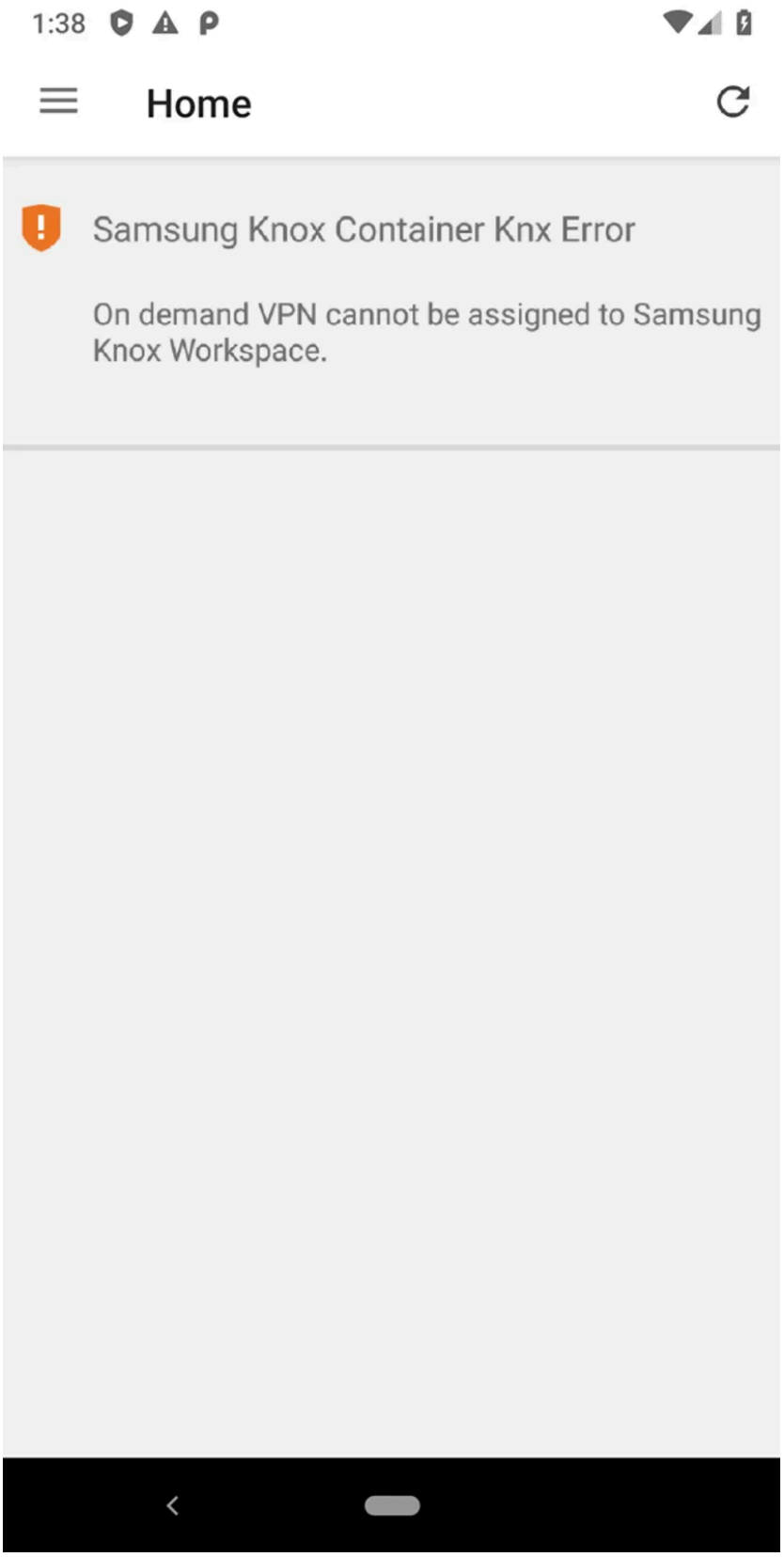
	On-demand	Always-on
Per-container	Not a valid configuration for Samsung Knox. This combination is configurable in Ivanti EPMM, however Tunnel will not work. See **Error messages for a per-container and on-demand Tunnel VPN setup**.	Tunnel starts when the Samsung Knox Workspace container starts. All apps in the container can use Tunnel VPN.
Per-app	Tunnel starts when an app that can use Tunnel is launched. Tunnel stops when there are no apps running that can use Tunnel VPN. The per-app list, which is the list of apps that can use Tunnel VPN, is set in the Knox container configuration.	Tunnel starts when the Samsung Knox Workspace container starts. Only traffic from apps that are configured to use Tunnel are allowed through Tunnel.

Error messages for a per-container and on-demand Tunnel VPN setup

A per-container and on-demand combination VPN configuration is not supported. However, you can configure per-container and on-demand VPN in the Ivanti EPMM Admin Portal. After the device syncs with Ivanti EPMM, error messages are seen in Mobile@Work on the device and in the device profile in the Admin Portal.

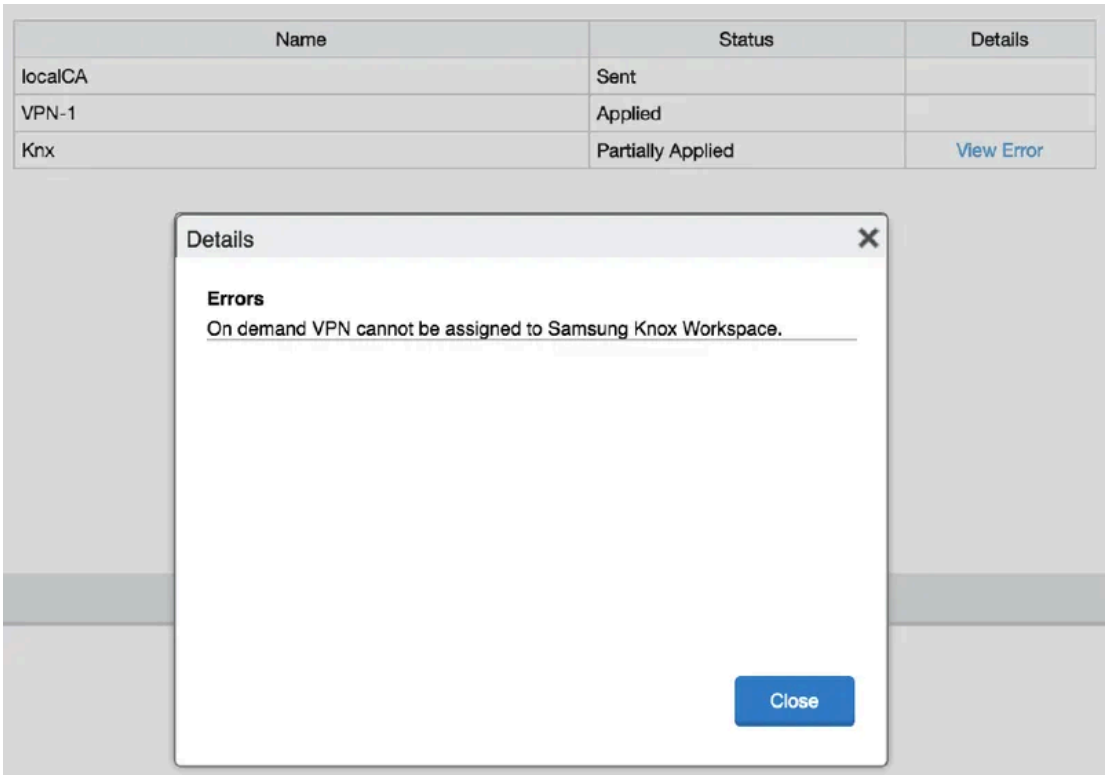
Mobile@Work displays the error as seen in the following figure.

Figure 1. Device configuration status error



In the Admin Portal, in Devices & Users > Devices, the Configurations tab for a device displays a link to View Error for the Samsung Knox container configuration.

Figure 2. View configuration error in the Admin Portal


Name
Status
Details
localCA
Sent
VPN-1
Applied
Knx
Partially Applied
View Error
Details
Errors
On demand VPN cannot be assigned to Samsung Knox Workspace.
Close

5.3. Configuring an IP_ANY AppTunnel service on a Standalone Sentry

Configure an IP_ANY AppTunnel service on a Standalone Sentry enabled for AppTunnel. Ivanti Tunnel creates the tunnel through which traffic is tunneled to the backend resource.

> If you already have an IP_ANY AppTunnel service configured on a Standalone Sentry enabled for AppTunnel, you can skip this section.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Settings > Sentry.
2. Click Edit to open the Standalone Sentry settings.
3. In the AppTunnel Configuration section under Services, click the plus icon to add the following service:
    - **Service name**: <IP_ANY>
    - **Server Auth**: Pass Through
    - All other fields: default
4. Click Save.

Next steps

Go to **Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace (Ivanti EPMM)**.

Related topics

For information about setting up an AppTunnel service in Standalone Sentry, see the “Working with Standalone Sentry for AppTunnel” section in the *Sentry Guide for Ivanti EPMM.*

5.4. Creating Ivanti Tunnel VPN configuration for Samsung Knox Workspace (Ivanti EPMM)

The Ivanti Tunnel (Samsung Knox Workspace) VPN configuration determines, DNS, and app behavior.

Before you begin

- Enable Standalone Sentry for AppTunnel.
- Set up Standalone Sentry to use identity certificates for device authentication.
- Create a certificates enrollment setting in Ivanti EPMM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Policies & Configs > Configurations.
2. Click Add New > VPN.
3. Enter a name and description for the VPN settings.
4. Configure the following:

Item	Description
Connection Type	Select Tunnel (Samsung Knox Workspace)
Sentry	Select the Standalone Sentry on which you have enabled AppTunnel.
Identity Certificate	Select the Certificate Enrollment setting you created for Sentry setup for AppTunnel.
VPN Chaining	Disable: Default. Inner: Select to enable VPN chaining.
VPN on Demand	Select the check box to enable VPN on demand. If unchecked, the VPN connection is always on.

5. Routes List is not supported in the Samsung Knox Workspace. Routes configured here will be ignored.
6. In DNS Resolver IPs, configure the list of DNS for Tunnel.
   Each entry is separated by ';'. IPv4 only.
   The DNS configured here are different from the DNS for the original Wi-Fi or cellular connection. If needed, the administrator should set the appropriate routes to ensure that DNS routes the requests to the appropriate destination.
7. In DNS Search Domain List, enter a list of search domains for DNS resolver separated by a semicolon (;).
8. In Custom Data, add key-value pairs to configure the app.
9. Click Finish.
10. Apply the appropriate label to the app to distribute it to Samsung Knox devices.

Next steps

To distribute the app through the app storefront, go to **Distributing Ivanti Tunnel through Apps@Work**.

Related topics

- See **Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace** for a description of the custom data key-value pairs.
- See **Configuring VPN chaining** for information about how to configure VPN Chaining.
- See also, **Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM)**.

5.5. Distributing Ivanti Tunnel through Apps@Work

Adding Ivanti Tunnel to the app storefront allows you to determine which Samsung Knox devices will get the app.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Apps > App Catalog.
2. Click Add+.
3. Click Google Play.
   The app wizard appears.
4. Type “Tunnel” for Application Name, and click Search.
5. Select the listing for Ivanti Tunnel.
6. Click Next.
7. Select “5.0” for Min. OS Version.
8. Click Next.
9. Select Silently Install.
10. Click Finish.
11. Apply the appropriate label to the app to distribute it to Samsung Knox devices.

Next steps

Add the Tunnel app the Samsung Knox container and to configure VPN for the apps that will use Ivanti Tunnel. See **Configuring app VPN in the Samsung Knox container**.

Related topics

- For more information on adding and editing Google Play apps to the app catalog, see “Managing Mobile apps for Android” in the *Apps@Work Guide*.

5.6. Configuring app VPN in the Samsung Knox container

Update the Samsung Knox container configuration:

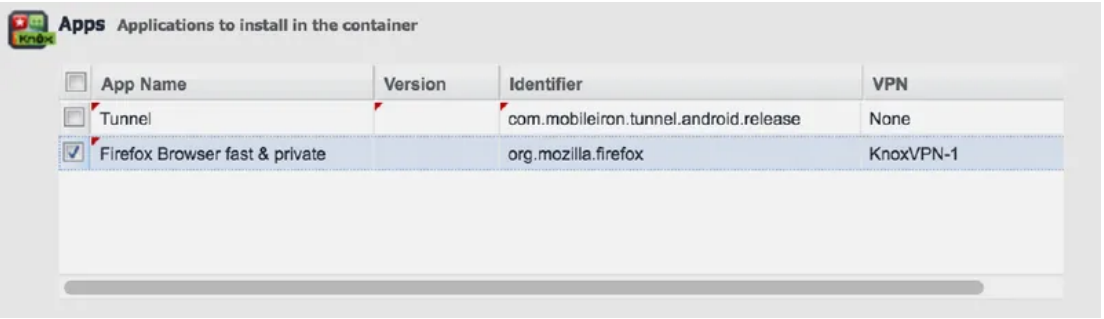
- Add Tunnel to the Samsung Knox container configuration so that the app is available in the container on Samsung Knox devices.
- Configure the apps in the container to use Tunnel VPN.

The Samsung Knox container configuration determines which VPN configuration is used and whether the VPN configuration is applied per app or per container.

Assigning different VPN configurations to apps is not supported. Example: Assigning VPN1 to App1 and VPN2 to App2 is not supported. Only one VPN configuration is supported in the Samsung Knox container. Two separate VPN configurations are allowed only for VPN chaining.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Policies & Configs > Configurations.
2. Select the Samsung Knox container configuration and click Edit.
   Figure 1. Apps configuration



1. In the Apps section do the following:
   1. Click + to add Ivanti Tunnel.
   2. For App Name, select the Tunnel app from the drop down list.
      All other fields for the app are set to default values. Do not make any changes to the default values.
   3. Similarly, if needed, add other apps to make the apps available in the Samsung Knox container.
2. Configure the apps to use Ivanti Tunnel VPN. Do one of the following:
   - **Configuring per-app VPN**.
   - **Configuring per-container VPN**.
3. Click Save.

Related topics

See also, **Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM)**.

Configuring per-app VPN

If you configure per-app VPN, only apps to which the Ivanti Tunnel VPN configuration is applied can use Tunnel VPN.

Procedure

1. In the Samsung Knox container configuration, scroll down to the Apps section.
2. For apps that will use Ivanti Tunnel VPN, in the VPN field, select the Tunnel (Samsung Knox Workspace) VPN configuration from the drop down list.
   Only the specified apps can use Tunnel VPN.
   Configure VPN in the Apps section only if a VPN configuration is not specified in the Apps Settings section.
3. Click Save.

Related topics

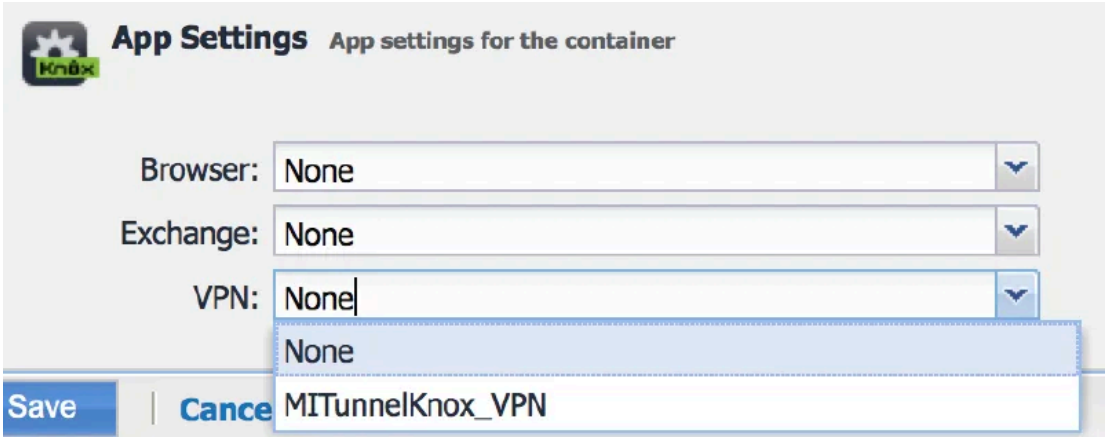
See also, **Configuring app VPN in the Samsung Knox container**.

Configuring per-container VPN

If you configure per-container VPN, all apps in the Knox container can use Ivanti Tunnel VPN.

Procedure

1. In the Samsung Knox container configuration, scroll down to the App Settings section.
   Figure 2. Apps settings configuration



1. For VPN in App Settings, select the Ivanti Tunnel (Samsung Knox Workspace) VPN configuration from the drop down list.
   The selected VPN configuration is applied to all apps in the Samsung Knox container.
   Configure VPN in App Settings only if a VPN configuration is not specified for any app in the Apps section. If you configure VPN in the App Settings, the VPN selection in Apps automatically resets to None.
2. Click Save

Related topics

See also, **Configuring app VPN in the Samsung Knox container**.

5.7. Configuring VPN chaining

VPN chaining is the nesting of a VPN tunnel in another VPN tunnel. VPN chaining provides additional security by hiding the Ivanti Tunnel VPN end destination. With Ivanti Tunnel you can configure VPN chaining with OpenVPN as the outer tunnel and Tunnel as the inner tunnel. VPN chaining can be configured for per-app only.

Before you begin

- Configure Tunnel for Samsung Knox Workspace as described in **Configuration overview for Ivanti Tunnel for the Samsung Knox container (Ivanti EPMM)**.
- Configure an OpenVPN VPN setting in the Ivanti EPMM Admin Portal. For more information, see the "Configuring new VPN settings" and the "OpenVPN" sections in the *Ivanti EPMM Device Management Guide* for Android.
  Use the OpenVPN setting on Ivanti EPMM only to configure Samsung "OpenVPN net.openvpn.knox.connect" for Samsung Knox devices. The configuration is available only to limited customers as approved by Samsung. Contact Samsung to get the correct OpenVPN package. It is supported only on devices with the Samsung Knox option selected in the VPN setting.

Procedure

1. In the Ivanti EPMM Admin Portal, go to Policies & Configs > Configurations.
2. Select and Edit the Ivanti Tunnel VPN configuration for Samsung Knox Workspace.
   1. In the Ivanti Tunnel VPN configuration for Samsung Knox Workspace, for VPN Chaining, select Inner.
   2. Click Save.
3. Select and Edit the OpenVPN configuration.
   1. In the OpenVPN configuration, for VPN Chaining, select Outer.
   2. Click Save.
4. Select and Edit the Samsung Knox container configuration.
   Figure 1. Apps configuration



1. In the Apps section of the Samsung Knox container configuration, do the following:
   1. For VPN for Tunnel, select the OpenVPN configuration with outer VPN chaining (Configured in step 3).
   2. For apps that will use VPN chaining, select the Ivanti Tunnel VPN configuration with inner VPN chaining (Configured in step 2).
2. Ensure that the configurations are applied to a label that contains the devices for which you want to allow VPN chaining with Ivanti Tunnel.

6. Setting up Ivanti Tunnel for Android Enterprise

The following address the setup required for app VPN using Ivanti Tunnel for Android Enterprise in Ivanti EPMM and Ivanti Neurons for MDM:

- Before you configure Ivanti Tunnel for Android Enterprise (Ivanti EPMM and Ivanti Neurons for MDM)
- Configuration tasks overview for Android Enterprise (Ivanti EPMM)
- Configuration tasks overview for Android Enterprise (Ivanti Neurons for MDM)

6.1. Before you configure Ivanti Tunnel for Android Enterprise (Ivanti EPMM and Ivanti Neurons for MDM)

Before you configure Ivanti Tunnel, ensure that you have met the requirements and have read the recommendations and limitations listed in this section.

- Required components for deploying Ivanti Tunnel for Android Enterprise
- Requirements for deploying Ivanti Tunnel for Android Enterprise
- Recommendations for deploying Ivanti Tunnel for Android Enterprise
- Limitations for Ivanti Tunnel for Android Enterprise
- Shared-kiosk mode

Required components for deploying Ivanti Tunnel for Android Enterprise

The following components are required for Ivanti Tunnel deployment on Android Enterprise devices:

- Standalone Sentry with AppTunnel enabled or Access
- UEM with the following:
  - UEM enabled for Android Enterprise
  - Users have Android Enterprise-capable device.
    UEM is Ivanti EPMM or Ivanti Neurons for MDM.
- Client for Android Enterprise:
  - Ivanti EPMM: Mobile@Work
  - Ivanti Neurons for MDM: Go

> Tunnel for Android Enterprise and Mobile@Work for Android are available from the Google Play store.

For supported versions see the *Ivanti Tunnel for Android Release Notes* for this release.

Requirements for deploying Ivanti Tunnel for Android Enterprise

The following are required for deploying Ivanti Tunnel for Android Enterprise:

- Your Ivanti Neurons for MDM must be set up for Android Enterprise. For more information, see:
  - Ivanti EPMM: *Ivanti EPMM Device Management Guide for Android and Android Enterprise.*
  - Ivanti Neurons for MDM: *Getting Started with Android for Work.*
- If your deployment uses Standalone Sentry:
  - You must have installed Standalone Sentry. See the *Standalone Sentry Installation Guide*.
  - Standalone Sentry must be set up for AppTunnel using Identity certificates for device authentication.
    For information about setting up a Standalone Sentry for AppTunnel, see:
    Standalone Sentry Guide for Ivanti EPMM and Standalone Sentry Guide for Ivanti Neurons for MDM.

- If your deployment uses Access, ensure Access is set up.
  See the *Access Guide* for information on how to set up Access.
- Ensure that the appropriate ports are open.
  See the *Ivanti Tunnel for Android Release Notes*.

Recommendations for deploying Ivanti Tunnel for Android Enterprise

The following are recommendations for deploying Ivanti Tunnel for Android Enterprise:

- Ivanti recommends that Standalone Sentry use a publicly trusted CA certificate. Android version 7 through the latest versions as supported by Ivanti does not accept self-signed certificates.
- If your deployment includes Android 5 and 6 devices, and if Standalone Sentry uses a self-signed certificate, see *Using a Self-signed certificate with Standalone Sentry and Tunnel* **knowledge base article** in the Support and Knowledge Base portal. The configuration sections describe the use of Ivanti EPMM UI. However for Ivanti Neurons for MDM as well, create a certificate setting and upload the Sentry server certificate to Ivanti Neurons for MDMand distribute the certificate setting to devices.
- If access to the ActiveSync server is going through Standalone Sentry, configure Ivanti Tunnel so that email clients are excluded from being routed through Tunnel.

Limitations for Ivanti Tunnel for Android Enterprise

The following are limitations of Ivanti Tunnel for Android Enterprise:

- Deployments that use a trusted front-end such as Apache/F5 to terminate SSL or the use of backend proxy from Standalone Sentry to upstream applications are not supported. (Cloud only)
- Front-end load balancer to Standalone Sentry is expected to work but has not been tested.
- Performance depends on the apps using Standalone Sentry. As a best practice, monitor Standalone Sentry usage and add more Standalone Sentry servers as needed for horizontal scaling.
- Real-time audio/video apps may not work. UDP functionality and scale will vary depending on the app. Performance of real-time audio and video apps has not been tested.
- Server authentication through Standalone Sentry with Kerberos is not supported.

Shared-kiosk mode

Ivanti Tunnel for Android supports shared-kiosk mode. Before you deploy Ivanti Tunnel, ensure that shared-kiosk mode is set up and deployed. After the UEM is set up for shared-kiosk mode, follow the configuration tasks for setting up Ivanti Tunnel for Android Enterprise.

For a better user experience, Ivanti recommends updating the UEM client version to Go 72 for Android or Mobile@Work 10.8.0.0 for Android through the most recently released version as supported by Ivanti.

To set up shared-kiosk mode:

- For Ivanti EPMM, see "Android shared-Kiosk mode overview" in the Ivanti EPMM Device Management Guide for Android and Android Enterprise devices.
- For Ivanti Neurons for MDM, see "Setting up Kiosk Mode for Android" in the *Ivanti Neurons for MDM Guide*.

6.2. Configuration tasks overview for Android Enterprise (Ivanti EPMM)

The following configuration tasks are required to set up app VPN with Ivanti Tunnel. These configuration tasks are performed in the Ivanti EPMM Admin Portal:

1. **Adding and configuring the Ivanti Tunnel for Android Enterprise (Ivanti EPMM)**
2. **Creating an Always-On VPN configuration (Ivanti EPMM, optional)**

Adding and configuring the Ivanti Tunnel for Android Enterprise (Ivanti EPMM)

Add the Ivanti Tunnel app to Ivanti EPMM from Google Play and configure it as follows to make it available to Android Enterprise devices.

Before you begin

- If you are configuring app VPN, you must have created an IP_ANY AppTunnel service in Standalone Sentry. For information on setting up an IP_ANYTunnel service see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti EPMM.
- If you are configuring Tunnel to support anti-phishing with MTD, you must have an MTD setup enabled for anti-phishing. See **Support for anti-phishing with Mobile Threat Defense**

Procedure

1. In the Ivanti EPMM Admin Portal, go to Apps > App Catalog.
2. Click Add+.
3. Click Google Play.
4. Enter Tunnel for Application Name, and click Search.
5. Select the line for Tunnel app.
6. Click Next.
7. Select "5.0" for Min. OS Version.
8. Click Next.
9. Select Install this app for enterprise.
   Additional fields are exposed.
10. Select Silently Install.
11. Select Enable Access only if you have an Access as a service deployment.
    Selecting this option enables authentication traffic through Access. The option is available only if Access as a service is set up with Ivanti EPMM.
12. Scroll down to Configuration Choices.
13. Do one of the following:
    - Select Use Tunnel for Anti-phishing only, to enable Tunnel VPN to analyze phishing URLs.
      Do not select this option if you have any restrictions configured. Selecting the option removes any configured restrictions and hides the Default Configuration for Tunnel section. To configure anti-phishing when you have an existing Tunnel

deployment, add a new Tunnel configuration and select the option.

OR

- Expand Default Configuration for Tunnel to configure the restrictions for the app.

Select either Use Tunnel for Anti-phishing only or configure the restrictions under Default Configuration for Tunnel. To deploy Tunnel for MTD and for Sentry or Access, create two separate configurations.

14. Click Finish.

Next steps

Go to **Creating an Always-On VPN configuration (Ivanti EPMM, optional)**.

Related topics

- See **Ivanti Tunnel configuration field description for Android Enterprise** for a description of the restrictions.
- For information about how to set up Access as a service with Ivanti EPMM, see the *Access Guide*.
- For information about adding and configuring an Android Enterprise app, see "App configuration for Android Enterprise apps," in the *Apps@Work Guide*.
- For information about setting up anti-phishing with Threat Defense, see "Advanced phishing protection for managed devices" in the Threat Defense Solution Guide for Ivanti Neurons for MDM.
- **Support for anti-phishing with Mobile Threat Defense**.
- **Adding multiple Ivanti Tunnel configurations**.

Adding multiple Ivanti Tunnel configurations

You can create multiple Ivanti Tunnel configurations and assign the configuration to a label. One reason you may need to create multiple Tunnel configurations is when you configure Tunnel to support anti-phishing with an MTD deployment as well as for deployment with Sentry or Access.

When you add Ivanti Tunnel for Android Enterprise to the App Catalog, a default Tunnel configuration for MTD is automatically available. The Ivanti Tunnel MTD configuration is pushed to devices when you select Use Tunnel for Anti-phishing only. However, selecting the option removes all other restrictions. Therefore, to also configure Ivanti Tunnel for Sentry or Access add a separate Tunnel configuration.

> If you have an existing Ivanti Tunnel configuration to use with Sentry or Access, add a new Tunnel configuration for anti-phishing and vice versa.

Procedure

1. On Cloud, go to Apps > App Catalog.
2. Highlight and click the Ivanti Tunnel app for Android Enterprise.
3. On the App Configurations tab, for Managed Configuration for Android click + to add a new configuration.
4. Select Use Tunnel for Anti-phishing only or configure restrictions in Managed Configurations.
5. Select a distribution for the new configuration.
6. Click Save.

Creating an Always-On VPN configuration (Ivanti EPMM, optional)

The Ivanti Tunnel app can be configured for Always-On VPN status for devices using Android 7 through the most recently released version as supported by Ivanti EPMM 9.3.

With Always-On VPN, the VPN connection is always on. Any app in the Android Enterprise container can go through the tunnel.

If a connection fails, Tunnel tries to reconnect periodically. Tunnel makes three quick attempts at one-second intervals, and then at one-minute intervals.

Ivanti Tunnel attempts to reconnect when there is a network status change or there is a configuration change. Tunnel will also attempt to reconnect if Standalone Sentry times out due to TCP idle time. If Tunnel is idling, Standalone Sentry closes the TCP connection. In this case, Tunnel will attempt to reconnect. The recommended idle timeout is one hour.

Procedure

1. Go to **Policies & Configs** > **Configurations** and click the **Add New** pull down menu.
2. Select **Android** > **Android Enterprise** to display the **New Android enterprise (all modes) Setting** screen.
3. Select the **Always-On VPN** check box to display the **App Identifier** pull-down menu.
   The pull-down menu lists only apps that are configured to be installed as Android Enterprise apps.
4. Select a VPN app to apply the Always-On setting. Click **Save**.

> In Device Details, the Android Enterprise setting displays as Partially Applied with an error message if the selected app is not installed on the device, the app is not a VPN app, or the VPN app does not support Always-on.

6.3. Configuration tasks overview for Android Enterprise (Ivanti Neurons for MDM)

The following configuration tasks are required to set up Tunnel. These configuration tasks are performed in the Cloud Admin Portal.

1. **Adding and configuring the Ivanti Tunnel app for Android Enterprise(Ivanti Neurons for MDM)**.
2. **Creating an Always-On VPN configuration (Cloud, optional)**.

Adding and configuring the Ivanti Tunnel app for Android Enterprise(Ivanti Neurons for MDM)

Upload the Ivanti Tunnel app to Ivanti Neurons for MDM from Google Play and configure it to make it available to Android Enterprise devices. You can download the app from Google Play.

Before you begin

- Ensure that you have met the requirements detailed in **Before you configure Ivanti Tunnel for Android Enterprise (Ivanti EPMM and Ivanti Neurons for MDM)**.
- If you are configuring app VPN,
  - You must have created a Tunnel service for Android in Standalone Sentry. For information on setting up Standalone Sentry with a Tunnel service, see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti Neurons for MDM.
  - Standalone Sentry must be set up to use identity certificates for device authentication.
  - Ensure that you have created a Identity Certificate configuration in Ivanti Neurons for MDM. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.
- If you are configuring Ivanti Tunnel for securing authentication traffic with Access, you must have setup Access. For information about setting up Access see the *Access Guide*. As part of the Access setup, you will have created a Tunnel service. If you are configuring Ivanti Tunnel to support anti-phishing with MTD, you must have an MTD setup enabled for anti-phishing. See **Support for anti-phishing with Mobile Threat Defense**

Procedure

1. In the Ivanti Neurons for MDM portal, go to Apps >App Catalog.
2. Click +Add next to App Catalog.
3. Select **Google Play** from the catalog pull-down menu.
4. In the search text box, enter Tunnel to locate the app in the Google Play store.
5. Click on the Tunnel icon in the search results.
   A description and screen captures of the app are displayed.
6. Click Select.
   Options to add categories and a description are displayed.
7. Make changes as needed and click Next.
   The App Delegation screen displays noting that AFW is enabled.
8. Click Next.

9. Select a distribution option and click Next.
   The configuration will be distributed to the devices in the group you selected.
   The App Configurations screen displays.
10. Click + for Managed Configurations for Android to configure settings for the app.
11. Enter a name and description for the configuration.
12. Select Blocks the user for uninstalling the app if you do not want device users to uninstall the app.
13. Do one of the following:
    - Select Use Tunnel for Anti-phishing Only, to enable Tunnel VPN to analyze phishing URLs.
      Do not select this option if you have any restrictions configured. Selecting the option removes any configured restrictions and hides the Managed Configurations section. To configure anti-phishing when you have an existing Tunnel deployment, add a new configuration and select the option.
      OR
    - Expand Managed Configurations to configure the restrictions for the app.

Select either Use Tunnel for Anti-phishing Only or configure the restrictions under Managed Configurations. To deploy Tunnel for MTD and for Sentry or Access, create two separate configurations.

14. Select a distribution option for the configuration and click Next.
15. Click Install Application configuration settings to configure the install options.
    1. Edit the Name and Description of the settings if necessary.
    2. Install on Device: Enable Install on devices, if you want to require that the app is installed on devices.
    3. Silently install on Samsung KNOX and Zebra devices: This option is not applicable to Android Enterprise apps.
    4. Do not show app in end user App Catalog: Select if you do not want the app to display in the app catalog on users' devices.
16. Click Next.
17. Click Promotion distribution configuration settings and select a promotion option.
    The promotion option determines how the app appears in the app catalog on the device.
18. Click Next and then click Done.

Next steps

Go to **Creating an Always-On VPN configuration (Cloud, optional)**.

Related topics

- See **Ivanti Tunnel configuration field description for Android Enterprise** for a description of the restrictions.
- For information about how to set up Access as a service with Ivanti EPMM, see the *Access Guide*.
- For information about adding and configuring an Android Enterprise app, see "Adding Google Play Store app for Android Enterprise," in the *Ivanti Neurons for MDM Guide*.
- For information about setting up anti-phishing with Threat Defense, see "Advanced phishing protection for managed devices" in the Mobile Threat Defense Solution Guide for Cloud.
- **Support for anti-phishing with Mobile Threat Defense**.
- **Adding multiple Tunnel configurations**.

Adding multiple Tunnel configurations

You can create multiple Tunnel configurations and assign the configuration to a distribution group. One reason for creating multiple Tunnel configurations is when you configure Tunnel to support anti-phishing with an MTD deployment as well as for deployment with Sentry or Access.

Use the procedure described here to create additional Tunnel configurations for anti-phishing or for tunneling to Sentry or Access.

> If you have an existing Tunnel configuration to use with Sentry or Access, add a new Tunnel configuration for anti-phishing and vice versa.

Procedure

1. On the Ivanti EPMM Admin Portal, go to Apps > App Catalog.
2. Select the Tunnel app for Android Enterprise and click Edit.
3. Scroll down to Configuration Choices.
4. Click Add+ to add a new Tunnel configuration.
5. Select Use Tunnel for Anti-phishing only or expand Configuration for Tunnel to configure the restrictions for the app.
6. If you configured restrictions for the Tunnel app, apply the new configuration to a label.
7. Click Save.

Creating an Always-On VPN configuration (Cloud, optional)

The Ivanti Tunnel app can be configured for Always-On VPN status for devices using Android 7 through the most recently released version as supported by Ivanti.

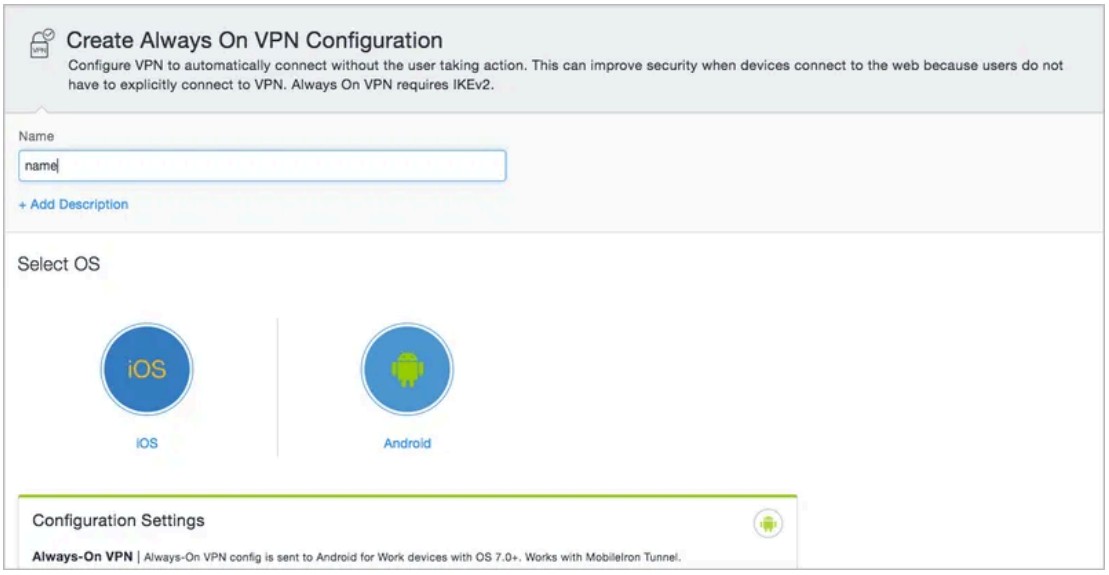
With Always-On VPN, the VPN connection is always on. Any app in the Android Enterprise container can go through the tunnel.

If a connection fails, Tunnel tries to reconnect periodically. Tunnel makes three quick attempts at one-second intervals, and then at one-minute intervals.

Tunnel attempts to reconnect when there is a network status change or there is a configuration change. Tunnel will also attempt to reconnect if Standalone Sentry times out due to TCP idle time. If Tunnel is idling, Standalone Sentry closes the TCP connection. In this case, Tunnel will attempt to reconnect. The recommended idle timeout is one hour.

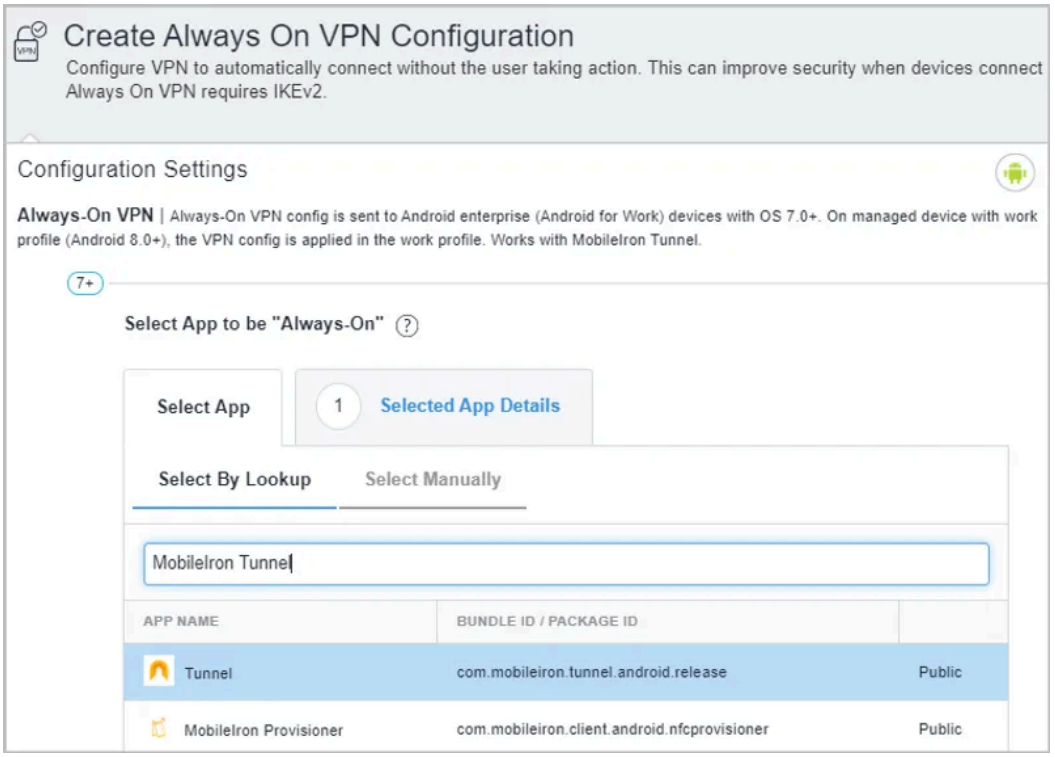
Procedure

1. In Cloud, go to Configuration and click **+Add**.
2. Click **Always On VPN**.
3. Enter a name for the configuration.
4. Select the **Android** operating system.
   Figure 1. Always on VPN configuration



1. In Configuration Settings, enter Tunnel in the search box.
2. Select the Tunnel app. Click **Next**.

Figure 2. Select app for always-on



1. Select a distribution group, and click **Done**.

7. Ivanti Tunnel Configuration Fields and Custom Data

The following describe the configuration fields and key-value pairs for configuring Ivanti Tunnel for Android:

- Ivanti Tunnel for Android native configuration field description
- Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace
- Ivanti Tunnel configuration field description for Android Enterprise
- Example showing the Sentry certificate in the certificate chain

7.1. Ivanti Tunnel for Android native configuration field description

The following table provides field descriptions for the Ivanti Tunnel configuration. There are some variations in field names between Ivanti EPMM and Ivanti Neurons for MDM.

Item	Description
Name	Enter a name for the Tunnel VPN profile.
Description	Enter a description for the profile.
Connection Type (Ivanti EPMM)	Select Tunnel (Android). Only fields relevant to Tunnel for Android are displayed.
Choose OS to create Tunnel Configuration (Ivanti Neurons for MDM)	Click Android. Fields relevant to Tunnel for Android are displayed.
Enable Access (Ivanti EPMM)	Select to enable authentication traffic through Access. The option is available only if Access as a service is set up with Ivanti. For information about how to set up Access as a service with Ivanti EPMM, see the *Access Guide.*
Profile selection mode to use for this configuration (Ivanti Neurons for MDM)	Select one of the following: • Sentry Profile Only: Select if Tunnel traffic goes only through Standalone Sentry. • Access Profile Only: Select if Tunnel traffic goes to Access. This option is available only if an Access as a service deployment is set up with Cloud. • Sentry + Access Profile: Select if Ivanti Tunnel VPN supports both traffic to Access for authentication to enterprise cloud resources and through Standalone Sentry to on-premise enterprise resources. This option is available only if an Access as a service deployment is set up with Ivanti Neurons for MDM.
Sentry (Profile)	**Ivanti EPMM:** Select the Standalone Sentry on which you created the IP_ANY tunnel service. **Cloud:** Select the Standalone Sentry profile on which you created the Tunnel service for Android. The option is not available if the profile mode is Access Profile Only.
Sentry Service (Ivanti Neurons for MDM)	Select the Tunnel service you created for Android. The option is not available if the profile mode is Access Profile Only.
Identity Certificate (Ivanti EPMM)	Select the Certificate Enrollment setting you created for Sentry setup for AppTunnel.
Client Cert. Alias (Ivanti Neurons for MDM)	Select the Identity Certificate configuration you created for Standalone Sentry setup. If the profile mode is Access only or Sentry + Access, select the same certificate you select for SCEP Identity.
SCEP Identity (Ivanti Neurons for MDM)	Select the Identity Certificate configuration you created for Ivanti Tunnel. This field is applicable if the profile mode is Access only or Sentry + Access.
Debug Info Recipient (Ivanti Neurons for MDM) For Ivanti EPMM, the setting is configured using key-value pairs in Custom Data.	Enter a valid email address. The device debug logs are sent to the configured email address. When users tap Email Debug Info, the To field is auto filled with the configured email address.

Item	Description
UI Notification Level (Ivanti Neurons for MDM) For Ivanti EPMM, the setting is configured using key-value pairs in Custom Data.	The user will see error notifications or all Tunnel related notifications, based on the level of notifications you configure. • Never show notifications: Notifications or errors are not displayed, except if an error occurs upon establishing Tunnel. • Error notifications only: Only errors notifications are displayed. This is the default setting if the key-value is configured. • All notifications: Error notifications and connect/disconnect confirmations are displayed. There are no notifications to indicate that an app is blocked or allowed.
Debug Log (Ivanti Neurons for MDM) For Ivanti EPMM, the setting is configured using key-value pairs in Custom Data.	Select the log level. The client app can override the VPN profile.
Tunneled Applications (Ivanti EPMM)	Select one, either Add Allowed Apps or Add Disallowed Apps, to configure the apps that can use Tunnel. If you select an app from the app catalog, the package name is automatically added. Otherwise, enter the app name and the package name. If the list is empty, all apps are allowed through Tunnel VPN.
Add Allowed apps	Use this setting if you want only the listed apps to use Tunnel VPN. Only apps in the App Catalog can be added to the app list. This setting creates a whitelist. For Cloud, • enter a semicolon (;) separated list. • if Allowed Apps List is configured, the Disallowed Apps List setting is grayed out and vice versa.
Add Disallowed apps	Use this setting if you do not want the listed apps to use Ivanti Tunnel VPN. Only apps that are not listed will use Tunnel VPN. This setting creates a blacklist. For Cloud, • enter a semicolon (;) separated list. • if Allowed Apps List is configured, the Disallowed Apps List setting is grayed out and vice versa.
Routes List / Added Routes	Configure the network routes that are allowed through Ivanti Tunnel. Use CIDR format. Each entry in the list is separated by ';'. IPv4 only. This enables split tunneling where only specific traffic can be taken through Tunnel. The routes configured only impact apps that use Ivanti Tunnel. Example: 10.0.0.0/8;101.210.48.9/32 In an Access deployment, if routes are not configured, then authentication traffic that is federated through Access goes to Access and all data-traffic goes to Sentry. Ivanti recommends configuring a route list so that only traffic destined to on-

Item	Description
	premise enterprise resources goes through Standalone Sentry and all other data traffic goes directly to the destination.
DNS Resolver IP	Configure the list of DNS for Tunnel. Each entry is separated by ';'. IPv4 only. The DNS configured here are different from the DNS for the original Wi-Fi or cellular connection. If needed, the administrator should set the appropriate routes to ensure that DNS routes the requests to the appropriate destination.
Search Domain	Enter a list of search domains for DNS resolver separated by a semicolon (;)
Custom Data Add key-value pairs to configure the app. See **Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace** for a description of the restrictions.	

7.2. Custom data key-value pairs for Ivanti Tunnel for Android native and Samsung Knox Workspace

The following table provides a description of the custom data key-value pairs.

Key	Value: Enter	Description
Manage Tunnel timeout		
TcpIdle TmoMs	*An integer*	The Tunnel TCP session idle timeout, on Standalone Sentry, in milliseconds. Tunnel sends this value to Standalone Sentry during the initial handshake in header X-App-TcpIdleTimeoutMs. If this key-value pair is not configured, the default value is 3600000 milliseconds (one hour). Frequently, in production environments, there are firewalls and load balancers between the device and Standalone Sentry. Each network element may have a different idle timeout, shorter than the timeout for Standalone Sentry. Ivanti recommends that the value for TcpIdleTmoMs is less than the idle timeout for all the other network elements. As an alternative, consider configuring TCP keep-alive.
VPN connection		
AllowBypass (Android native only)	• true • false	true: Allows all apps to bypass this VPN connection. Apps may use methods such as setProcessDefaultNetwork(Network) to send and receive directly over the underlying network or any other network for which they have permissions. false: Default, if the key-value pair is not configured. All traffic from apps is forwarded through the VPN interface. Apps cannot bypass the VPN.
SplitDomainsList	*List of domain suffixes separated a semicolon (;)*	Example: acme.com; google.com DNS requests with domains matching the values are sent to the DNS for the VPN. DNS requests with non-matching domains are sent to the device's DNS. Example: All DNS queries that match *.company.com are handled by the VPN DNS server, but all other queries are handled by the device network DNS i.e. not the VPN DNS server. The DNS handler for the Tunnel plugin decides which DNS request will be sent to which DNS server, based on the configured domains: • All sub domains are matched. Example: example.com matches example.com, staf.example.com, and jira.example.com • The configured domain is considered completed with top domains. Anything to the right of the top domain is omitted. Example: example.com does not match example.com.akamai.com • Only complete domains are matched. Example: example.com does not match myexample.com • '*' and '?' are not valid characters for the configuration. The filtering is done on an IP packet level, therefore, DNS resolver functionality is not provided. The default behavior sends all DNS requests to the DNS for the VPN.
SplitUDPPortList	List of UDP ports separated by a semicolon (;)	List of UDP ports to send through Ivanti Tunnel VPN. All other UDP packets are sent directly to destination. If the key-value pair is not configured, all UDP packets are sent through Ivanti Tunnel VPN. Example

Key	Value: Enter	Description
		53;161-162;200-1024
MTU	*An integer*	Tunnel MTU. The default value if the key-value is not configured is 1400
quickRetryMaxAttempts	*An integer*	Number of attempts to reconnect to VPN. The default if the key-value pair is not configured is 3.
quickRetryIntervalSec	*An integer*	Time between attempts to reconnect to VPN in seconds. The default if the key-value pair is not configured is 1.
slowRetryIntervalSec	*An integer*	Time between attempts to reconnect to VPN in seconds. The default if the key-value pair is not configured is 60.
TcpKeepCount	*An integer*	The value configured specifies the number of unacknowledged probes for TCP keep-alive to send before the connection is considered as dead. The default value, if the key-value pair is not configured, is 20. The key is part of the Android operating system specifications.
TcpKeepIntervalSec	*An integer*	The value configured specifies the TCP keep-alive interval between subsequent failed keep-alive probes in seconds. The default value, if the key-value pair is not configured, is 2 seconds. The key is part of the Android operating system specifications.
AtpProbeIdleSec	An integer	Sets the minimum idle time, in seconds, after which probe packets are sent out with outbound Tunnel traffic. If Tunnel does not receive a response for at least one of the probes sent, the existing connection is dropped and a new connection is established with the server. The minimum idle time is based on the last inbound response received by Tunnel. For example, if the value is 60 seconds, if Tunnel does not receive any inbound traffic for 60 seconds, probe packets are sent with the next outbound Tunnel traffic. Default value if the key-value pair is not configured: 60 seconds
AtpProbeIntervalSec	An integer	Sets the interval, in seconds, between probe packets sent after the minimum idle time specified in AtpProbeIdleSec. Default value if the key-value pair is not configured: 1 second
AtpProbeCount	An integer	Sets the total count of the probe packets sent after the minimum idle time specified in AtpProbeIdleSec. Default value if the key-value pair is not configured: 5
Certificates		
DisablePinning	• true • false	false: Default, if the key-value pair is not configured. Certificate pinning is enabled. true: Certificate pinning is disabled. Disabling certificate pinning is not recommended for security reasons. The Standalone Sentry server certificate is automatically pushed to the device.

Key	Value: Enter	Description
Troubleshooting		
UINotificationLevel	• 0 • 1 • 2	The user will see error notifications or all Tunnel related notifications, based on the level of notifications you configure. Configure one of the following levels of user notifications that the Tunnel app will provide: • 0: Notifications or errors are not displayed, except if an error occurs upon establishing Tunnel. • 1: Only errors notifications are displayed. This is the default setting if the key-value is configured. • 2: Error notifications and connect/disconnect confirmations are displayed. There are no notifications to indicate that an app is blocked or allowed.
DebugLog	• 0 • 6 • 4 • 3 • 2	Controls the amount of logging. The client app can override the VPN profile. • 0: Default setting if the key-value pair is not configured. Minimal level of logs are collected. • 6: ERROR level • 4: INFO level. • 3: DEBUG level • 2: VERBOSE level
AllowCapture	• false • true	Allows users to capture traffic in a PCAP file. false: Device users are not allowed to trigger inner traffic capture. true: Device users are allowed to trigger inner traffic capture and email the PCAP file. The default, if the key-value pair is not configured, is false. The PCAP file may contain sensitive information.
debugInfoRecipient	*Email address*	The device debug logs are sent to the configured email address. When users tap Email Debug Info, the To field is auto filled with the value configured for debugInfoRecipient.
EnableUserControl	• true • false	**true**: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is available to the device user. **false**: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is not available to the device user. Default value if the key-value pair is not configured: true The key-value pair is not applicable to Tunnel deployed in the Samsung Knox workspace. By default, device users in the Samsung Knox workspace do not have the option to enable or disable Tunnel VPN.
DefaultMaxNumLogs	*An integer*	Sets the maximum number of log files. The default if the key-value pair is not configured is 8.
DefaultMaxPcapSize	An integer	Sets the maximum pcap file size in bytes. The default if the key-value pair is not configured is 2097152.

Key	Value: Enter	Description
DefaultMaxNumPcaps	An integer	Sets the maximum number of pcap files. The default if the key-value pair is not configured is 10.
AnalyticsEnabled	• true • false	**true**: Enables collection of analytics data for Mixpanel. **false**:Collection of analytics data is disabled. Default value if the key-value pair is not configured: true.
SendDeviceID	• true • false	**true**: Ivanti Tunnel provides the device ID to Access. The device ID is reported on Access in Reports > Errors. **false**: Ivanti Tunnel does not provide the device ID to Access. The key-value pair is useful in identifying devices that encounter connection errors when authenticating through Access. Default value if the key-value pair is not configured: false
Tethering		
ExcludeTethering	• true • false	**true**: Ivanti Tunnel VPN continues to work on the tethered host device without impacting the tethering client connection. **false**: Ivanti Tunnel VPN may impact the tethering client connection. Default value if the key-value pair is not configured: false This key-value pair may be required for Ivanti Tunnel for Android native only. If the KVP is configured to true, ensure that internal IP ranges do not overlap with the IP ranges used by the tethering client. Avoid the following IP ranges: 192.168.42.0/23 (192.168.42.0 ~ 192.168.43.255) 192.168.44.0/22 (192.168.44.0 ~ 192.168.47.255) 192.168.48.0/23 (192.168.48.0 ~ 192.168.49.255) Tethering traffic from client devices does not go through the VPN of the host device.

7.3. Ivanti Tunnel configuration field description for Android Enterprise

The following table provides a description of the configuration fields for Ivanti Tunnel enterprise.

Restriction	Description
Ivanti Tunnel profile mode (Ivanti Neurons for MDM)	Select one of the following: • Sentry Profile Only: Select if Tunnel traffic goes only through Standalone Sentry. • Access Profile Only: Select if Tunnel traffic goes to Access. This option is available only if an Access as a service deployment is set up with Cloud. • Sentry + Access Profile: Select if Ivanti Tunnel VPN supports both traffic to Access for authentication to enterprise cloud resources and through Standalone Sentry to on-premise enterprise resources. This option is available only if an Access as a service deployment is set up with Ivanti Neurons for MDM.
Sentry Server	Specify the FQDN for the Sentry server that is configured with the IP_ANY service. Configure Sentry Server if you selected one of the following Ivanti Tunnel profile modes: • Sentry Profile Only • Sentry + Access Profile
AllowedAppList	Optional. Use only if DisallowedAppList is empty. Applies only to apps in the Android Enterprise work profile. Provide a list of apps in the Android Enterprise profile that are allowed to use the Ivanti Tunnel VPN connection by supplying the app package names, separated by ';'. Example Example: com.salesforce.chatter;com.appexample.two If AllowedAppsList has one or more entries, only the apps in the list are allowed to use VPN. This is a whitelist.
DisallowedAppList	Optional. Use only if AllowedAppList is empty. Applies only to apps in the enterprise work profile. Provide a list of applications in the Android Enterprise profile to be prevented from using Tunnel by supplying the app package names separated by ';'. Example: com.salesforce.chatter;com.appexample.two If AllowedAppList is empty, then all apps can use VPN except the apps in the DisallowedAppList. This is a blacklist. Configuration conditions: • Both DisallowedList and DNSResolverIP Configured: • DNS resolved by non-DNSResolverIP for DisallowedList apps, independent of SplitDomainsList priority. Anti-phishing URLs not blocked for DisallowedList apps as traffic bypasses tunnel.
AllowBypass	Select to allow all apps to bypass this Ivanti Tunnel VPN.
AddedRoutes	Enter the network routes that are allowed through Ivanti Tunnel. Use CIDR format. Each entry in the list is separated by a semicolon (;). IPv4 only. This enables split tunneling where only specific traffic can be taken through Tunnel. The routes configured only impact apps that use Tunnel. Example: 10.0.0.0/8;101.210.48.9/32 In an Access deployment, if routes are not configured, then authentication traffic that is federated through Access goes to Access and all data-traffic goes to Sentry. Ivanti recommends configuring a route list so that only traffic destined to on-premise enterprise resources goes through Standalone Sentry and all other data traffic goes directly to the destination.
ExcludedRoutes	This API excludes a network route from the VPN interface. Calling this method overrides previous calls to addRoute(IpPrefix) for the same destination. This functionality leverages the new Android API level 33 to allow exclusion of specified IP routes from VPN traffic, directing them through the device's native network.

Restriction	Description
	If multiple routes match the packet destination, route with the longest prefix takes precedence.
DNSResolverIP	Enter the list of DNS for Ivanti Tunnel. Each entry is separated by a semicolon (;). IPv4 only. The DNS configured here are different from the DNS for the original Wi-Fi or cellular connection. If needed, the administrator should set the appropriate routes to ensure that DNS routes the requests to the appropriate destination. Configuration conditions: • DNSResolverIP Configured, AllowedAppList Not Configured: DNS resolved by IP in DNSResolverIP. • Both DNSResolverIP and AllowedAppList Configured: ◦ Browsing with AllowedAppList app: DNS resolved by DNSResolverIP. ◦ Browsing with non-AllowedAppList app: DNS resolved by device DNS.
SplitUdpPortList	
SplitDomainsList	Enter a list of domains suffixes separated by a semicolon (;). Example: mobileiron.com;google.com DNS requests with domains matching the values are sent to the VPN's DNS. DNS requests with non-matching domains are sent to the device's DNS. Example: All DNS queries that match *.company.com are handled by the VPN DNS server, but all other queries are handled by the device network DNS i.e. not the VPN DNS server. The Ivanti Tunnel plugin's DNS handler decides which DNS request will be sent to which DNS server, based on the configured domains: • All sub domains are matched. Example: mobileiron.com matches mobileiron.com, taf.mobileiron.com, and jira.mobileiron.com • The configured domain is considered completed with top domains. Anything to the right of the top domain is omitted. Example: mobileiron.com does not match mobileiron.com.akamai.com • Only complete domains are matched. Example: mobileiron.com does not match mymobiliron.com • '*' and '?' are not valid characters for the configuration. The filtering is done on an IP packet level, therefore DNS resolver functionality is not provided. The default behavior sends all DNS requests to the VPN's DNS Server.
IsSplitDomainsListisPriority	Domains (DNS IP) are resolved based on this configuration if SplitDomainsList and DNSResolverIP are configured. If 'Yes', SplitDomainsList takes priority and DNS is resolved based on SplitDomainsList configuration. If 'No', DnsResolvedIP takes priority and DNS is resolved based on DnsResolvedIP configuration. Default for "IsSplitDomainsListisPriority" is Yes. Configuration conditions: • If SplitDomainsList Configured: DNS resolved by SplitDomainsList. • If SplitDomainsList Not Configured but DNSResolverIP Configured: DNS resolved by DNSResolverIP. • If Neither Configured: DNS resolved by Sentry DNS.
SearchDomain	Enter a list of search domains for DNS resolver separated by a semicolon (;).

Restriction	Description
SentryService (Ivanti Neurons for MDM only)	Name of the IP Tunnel service defined on Sentry.
SentryPort (Ivanti EPMM only)	Sentry Tunnel port. Use port 443, typically.
ClientCertAlias	**Ivanti EPMM** This is the certificate alias set up in Ivanti EPMM. The value is $CERT_ALIAS:<name-of-SCEP>$ where <name-of-SCEP> is the Certificate Enrollment setting configured in Ivanti EPMM UI. Example: $CERT_ALIAS:scepIdentityCert$ where scepIdentityCert is the name of the SCEP configured in Ivanti EPMM. **Ivanti Neurons for MDM** Select the Identity certificate setting you created.
SentryCertificate (Ivanti EPMM only)	Copy and paste the Sentry certificate from the sentry-server-cert-chain.pem file. This is required if DisablePinning is not selected. For information on how to retrieve the sentry-server-cert-chain.pem file see KB article. For an example of which section of the sentry-server-cert-chain.pem file to copy, see Example showing the Sentry certificate in the certificate chain.
ReservedSentryCertificate	Once the primary certificate expires, Sentry automatically switches to the reserved certificate without admin action. Tunnel stores both certificates and can match either, ensuring seamless connectivity. If the original Sentry certificate fails trust checks, the reserved certificate is used for verification. It supports both OpenSSL and NativeSSL, and its CN and IssuerCN can be viewed in the Configuration screen.
DisablePinning	Disabling certificate pinning is not recommended for security reasons. If selected, the SentryCertificate is not required.
EnableUserControl	Select the check box to enable. Enabled: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is available to the device user. Disabled: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is not available to the device user.
enableOpenssl	Enable this parameter to reduce the reading and writing time to Sentry. It also enhances performance for low-end devices where the performance is limited by device or application speed.
UINotificationLevel	Choose one of the following levels of user notifications that the Tunnel app will provide: • Never show notifications: Notifications or errors are not displayed, except if an error occurs upon establishing Tunnel. • Error notifications only: Only errors notifications are displayed. • All notifications: Error notifications and connect/disconnect confirmations are displayed. The user will see error notifications or all Ivanti Tunnel related notifications, based on the level of notifications you choose. There are no notifications to indicate that an app is blocked or allowed.

Restriction	Description
DebugLog	Controls the amount of logging. The client app can override the VPN profile. • Default setting if the key-value pair is not configured. Minimal level of logs are collected. • ERROR level • INFO level. • DEBUG level • VERBOSE level
TrafficVerboseLog	Captures traffic logs. • Off: Default setting. No logs are collected. • Minimal: Minimal logs are collected. • All: Detailed logs are collected.
Allow traffic capture	Allows users to capture traffic in a PCAP file. The PCAP file may contain sensitive information.
TcpIdleTmoMs	The Ivanti Tunnel TCP session idle timeout, on Standalone Sentry, in milliseconds. Tunnel sends this value to Standalone Sentry during the initial handshake in header X-App-TcpIdleTimeoutMs. If this key-value pair is not configured, the default value is 3600000 milliseconds (one hour). Frequently, in production environments, there are firewalls and load balancers between the device and Standalone Sentry. Each network element may have a different idle timeout, shorter than the timeout for Standalone Sentry. Ivanti recommends that the value for TcpIdleTmoMs is less than the idle timeout for all the other network elements. As an alternative, consider configuring TCP keep-alive.
UdpIdleTmoMs	
MTU	Enter an integer for Tunnel MTU. The default value is 1400.
DebugInfoRecipient	Provide an email address. The device debug logs are sent to the configured email address. When users tap Email Debug Info, the To field is autofilled with the value configured for debugInfoRecipient.
quickRetryMaxAttempts	Number of attempts to reconnect to VPN. The default is 3.
quickRetryIntervalSec	Time between attempts to reconnect to VPN in seconds. The default is 1.
slowRetryIntervalSec	Time between attempts to reconnect to VPN in seconds. The default is 60.
appRunningCheckIntervalSec	Time between app status checks in seconds. By default this key is enabled with an interval of 60 seconds. To disable this key, enter 0.
TcpKeepIdleSec	Enables or disables TCP keep-alive and specifies the interval between the last data packet sent and the first keep-alive probe in seconds. ACKs are not considered as data. A value of 0 means TCP keep-alive is disabled. The default value, if the key-value pair is not configured, is 0.

Restriction	Description
	TCP keep-alive helps detect a dead tunnel connection and prevents most network load balancers and firewalls from idle-out the connection. The Standalone Sentry TcpIdleTmoMs is not impacted by TCP keep-alive. The key is part of the Android operating system specifications.
TcpKeepCount	The value configured specifies the number of unacknowledged probes for TCP keep-alive to send before the connection is considered as dead. The default value, if the key-value pair is not configured, is 20. The key is part of the Android operating system specifications.
TcpKeepIntervalSec	The value configured specifies the TCP keep-alive interval between subsequent failed keep-alive probes in seconds. The default value, if the key-value pair is not configured, is 2 seconds. The key is part of the Android operating system specifications.
AtpProbeIdleSec	Sets the minimum idle time, in seconds, after which probe packets are sent out with outbound Tunnel traffic. If Tunnel does not receive a response for at least one of the probes sent, the existing connection is dropped and a new connection is established with the server. The minimum idle time is based on the last inbound response received by Tunnel. For example, if the value is 60 seconds, if Tunnel does not receive any inbound traffic for 60 seconds, probe packets are sent with the next outbound Tunnel traffic. Default value if the key-value pair is not configured: 60 seconds
AtpProbeIntervalSec	Sets the interval, in seconds, between probe packets sent after the minimum idle time specified in AtpProbeIdleSec. Default value if the key-value pair is not configured: 1 second
AtpProbeCount	Sets the total count of the probe packets sent after the minimum idle time specified in AtpProbeIdleSec. Default value if the key-value pair is not configured: 5
AtpProbeIdleLimit	
InternalDebugOption1	Use only if instructed by Support for troubleshooting purposes.
TunIP	Use only if instructed by Support for troubleshooting purposes.
MaxNumLogs	Specify the maximum number of log files. The default is 8.
MaxNumPcaps	Specify the maximum number for pcap files. The default is 10.
AnalyticsEnabled	Check to enable collection of analytics data for Mixpanel. The box is checked by default.
SaveAfwConfiguration	Enable this configuration only if requested by Support.
AutoBackgroundLaunch	Check to enable the Tunnel app to automatically launch. The app is automatically launched without user interaction when a user tries to connect to a backend resource. For the feature to work, ensure that always-on is also enabled. The feature is available on Android N, O, and P.

Restriction	Description
AllowPerAppTunnel	For internal use only. Do not use this setting.
ClientCertsNumInChain	The value designates the number of certificates in the certificate chain that are passed to Sentry or Access. By default, only the leaf certificate is used. Ivanti recommends not changing the default setting unless additional certificates need to be passed to Sentry or Access.
SendDeviceID	
slowRetryMaxAttempts	Allows restarting the tunnel session. When **slowRetryMaxAttempts** are reached, the session stops, and a new connection begins. If the key-value pair is not configured, the default value is 0 seconds, which disables auto restart.
FqdnAllowedList	Allows FQDN-based traffic management by letting admins define traffic rules using domain names rather than IP addresses. • Routing behavior based on configuration combinations: ◦ Only **FqdnAllowedList** configured: All FQDNs are resolved and routed via Sentry. ◦ **FqdnAllowedList** + **AddedRoutes**: FQDNs in the **FqdnAllowedList** and IPs in the **AddedRoutes** are routed via Sentry. ◦ **FqdnAllowedList** + **AddedRoutes** + **ExcludeRoutes**: All traffic goes through Sentry except for traffic to IPs specified in the **ExcludeRoutes**. ◦ **FqdnAllowedList** + **AllowedAppList**: Only apps in the **AllowedAppList** can send traffic from FQDNs in the **FqdnAllowedList** via Sentry. All other traffic is routed through non-Sentry.

7.4. Example showing the Sentry certificate in the certificate chain

The Sentry certificate is in bold. Copy and paste the section in bold for pinning.

Certificate(s) for host: app1416.auto.mobileiron.com

Certificate: C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support, CN=app1416.auto.mobileiron.com

Serial Number: 3173868363

Signature Algorithm: SHA256withRSA

Issuer: C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support, CN=ProxyCA

Validity: Sat Aug 07 16:22:47 UTC 2021

PEM:

-----BEGIN CERTIFICATE-----

MIIFKjCCAxKgAwIBAgIFAL0tY0swDQYJKoZIhvcNAQELBQAwczEQMA4GA1UEAwwH

UHJveHIDQTEQMA4GA1UECwwHU3VwcG9ydDETMBEGA1UECgwKTW9iaWxlSXJvbjEW

MBQGA1UEBwwNTW91bnRhaW4gVmlldzETMBEGA1UECAwKQ2FsaWZvcm5pYTELMAkG

A1UEBhMCVVMwHhcNMTYwODA4MTYyMjQ3WhcNMjEwODA3MTYyMjQ3WjCBhzEkMCIG

A1UEAwwbYXBwMTQxNi5hdXRvLm1vYmlsZWlyb24uY29tMRAwDgYDVQQLDAdTdXBw

b3J0MRMwEQYDVQQKDApNb2JpbGVJcm9uMRYwFAYDVQQHDA1Nb3VudGFpbiBWaWV3

MRMwEQYDVQQIDApDYWxpZm9ybmlhMQswCQYDVQQGEwJVUzCCASIwDQYJKoZIhvcN

AQEBBQADggEPADCCAQoCggEBAIydxpmUGJy6Z3BJ21MxBs5w3kKVddANQmo1cVJC

InhJLrm4IK3Mazs09/2bF3t+ND8xTkI2YjrRiZaz94B2dkrI7fcX0r7tjbStcXUP

yM6+49ipuBxjUKJNs20ZFJdRC0VK8ecbBS1DFOnlIW+fGUEqtWVA/k3nrwoeMfeP

zKg4hHBzB4+Bs369nzyIkxXxy9gUKfRLEs/kxWAexJB8eopxf6Zdf9W8tUp15hIC

ar6m3TY07pL3KU03U0K7mJXx0IsYqES8DHkTHfj2jYqnEqhxNurwTARYYmuV4iFU

BuztHaKzE0OSco4SMtBqa3FdU0J9EHI1tiuzPqPfA4HTxA0CAwEAAaOBrzCBrDAJ

BgNVHRMEAjAAMBMGA1UdDgQMBApTZW50cnktS2V5MIGJBgNVHSMEgYEwf6F3pHUw

czEQMA4GA1UEAwwHUHJveHIDQTEQMA4GA1UECwwHU3VwcG9ydDETMBEGA1UECgwK

TW9iaWxlSXJvbjEWMBQGA1UEBwwNTW91bnRhaW4gVmlldzETMBEGA1UECAwKQ2Fs

aWZvcm5pYTELMAkGA1UEBhMCVVOCBACs7+swDQYJKoZIhvcNAQELBQADgglBAEV/

sdXPHxUHZSBuKBbPj2h8oXQa4NIz1FawqNzbdBTktmUgqkcyu2hxi1uD6MgliqKc

Uz6IfLI3zMw4QULhwlaqleAlqkpQ9x45wySx+BufzpiC00qkC28wJdKnBOdM3Jig

CpvJCelvS3jTYuyjgJRNbaM0HGGlrHu4NBGrHljevHawOHTkvr9QmYhHhT2XYnug

FFXM5giC1ot9vGLA+UrZpVGrDg2Kcql5Eb+K99kjekTQ+0x7oFNj1wb8v1ZMpm/b

zzssOliltccZPVodJ0ksrmBFhH1m1L8VwcE5nqAwMrJ2vumplOlUXLxZHcWYYpYX

nZlDcvxZqzM78AaULQV7UnUCr4ldbu16X3/06LrCVBYq9zTiQwo/ZgWx5NFsXJVH

DpLhR30sBQ4kiorsXUlHxmnqA31snp3KDpt2WnJvFw9uCYNd0fg65zuSP/5Dw9Th

8zFv+ksSVVOXFEMJ7jL6j4LfCSB2weE1wdkqG2ze+fC259Gbq9pP4PAHpG9UtB2D

0d15saQKhBAwUhPNMlCHVxDi/1zpnbvtRNW3C5G/1uKpZ1V6rDeoaDsV/l+S9SZx

LvDsyje6vP0OVMOcv9wlY/iHvs/P2SBh2+ZhfYVt1/5v44cwRDYm1kdWDnsqHMUr

ESUMZjP96tJL1vK6GaM6AraGZSbtvhhjd5rZHZjd

-----END CERTIFICATE-----

Certificate: C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support, CN=ProxyCA

Serial Number: 11333611

Signature Algorithm: SHA256withRSA

Issuer: C=US, ST=California, L=Mountain View, O=MobileIron, OU=Support, CN=ProxyCA

Validity: Wed Aug 01 16:22:37 UTC 2046

PEM:

-----BEGIN CERTIFICATE-----

MIIFxDCCA6ygAwIBAgIEAKzv6zANBgkqhkiG9w0BAQsFADBzMRAwDgYDVQQDDAdQ

cm94eUNBMRAwDgYDVQQLDAdTdXBwb3J0MRMwEQYDVQQKDApNb2JpbGVJcm9uMRYw

FAYDVQQHDA1Nb3VudGFpbiBWaWV3MRMwEQYDVQQIDApDYWxpZm9ybmlhMQswCQYD

VQQGEwJVUzAeFw0xNjA4MDgxNjIyMzdaFw00NjA4MDExNjIyMzdaMHMxEDAOBgNV

BAMMB1Byb3h5Q0ExEDAOBgNVBAsMB1N1cHBvcnQxEzARBgNVBAoMCk1vYmlsZUly

b24xFjAUBgNVBAcMDU1vdW50YWluIFZpZXcxEzARBgNVBAgMCkNhbGlmb3JuaWEx

CzAJBgNVBAYTAlVTMIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAgkD6

rKQ5ASnilV3zvhnTtGjDhQ3dGvtUAll6S8jwCXbb7Ed4dK2zZ5Olf16z6LDNGPsE

0q3EdLVXuzeUA0nVkiThkd9hmXLrp91lj1TmuibMh6oF34YJAuMzpIMLVuRcQjhK

9k1b3mZsjSGeFGKWxgKJ+iAwz+Hp3OPptUreQYgCF1cPQQMbh2U5gGJm6zgN2sBt

2tfLtB6cDnurkriLfegm+o7ppvmULMof9OJF7Lr8vl8ozVArF60kaw0ywKex/1ZP

exruAEey/6QPWW6LWYuakmEGIq10R5WCkJ9p5Wv2+Y+pFTgcWmr0GmeHTFrfljsM

FJzLzMSbq2gAJBU1tEGzaY449y+yuSzGUqxTltoeWDQY/KDcvM9gny90Seo8h+8R

fn9/B0a//wo4H/fnJX7OZMXI1kPtEg/6roVvsexzHkI1H9FPrDW0g45Aps1/6g5n

QxE3wCCipHN2hmXgI33kREkXSL4pRTlbbdUW1+fkX2BYpxzYILS6ZCXf+RvNFenH

iWoogp6gUC1RzDKmwYMSpduWjireOP0MRJ8cckys8Bon+3/i7SPpAj7gfEYRb0BD

1yx/T0EIufP/wuLQSwdk2sqrwVYjdqrfqFSIAGyI8dkzTZue/tqzPTYMk0xyKCVX

Wly4v3PDRnm2G83qsC3r6ndK2ct52aInyOimBzcCAwEAAaNgMF4wDwYDVR0TAQH/

BAUwAwEB/zALBgNVHQ8EBAMCAaYwHwYDVR0jBBgwFoAUCqgSnfU4jXXhXPrrDVEr

UIzpckowHQYDVR0OBBYEFAqoEp31OI114Vz66w1RK1CM6XJKMA0GCSqGSIb3DQEB

CwUAA4ICAQBBfw9G+5U5wHDpX8ZCN8Jy5fqPHXjtDjcXJi4wgq75om2EAp4nWRb8

WmASZ8pz3ZA6JVM5QG1wS616bU5dxNMnu+snzueDdTDUZXV7aPtkn6QkNJeZqvMG

UWLr3TYHXQIeViOrsxkh2oW+9j4XBIN9Kb6R8H02S2JDu8tdX4GQUi6xt9gA9IfM

QakHGPDh0PsIHauT2Gz7KUYzRqscqF2NIKQS6Z/VhTm3CEexOZNHrBIZM/INMs2i

VMjWUI+d0ouUiagcenPtM26hR9uuCkvwzNSVrhPljNIVIc1juKGo3K9VTbISXkmG

hnWIB1QdPhpat1uDB49Lb7gZnIMDCcfezRIZhwgqJgFyTOeekJpMqtfJpU8s6Rbc

B7EY6i3AGFc8dbtEZbZEL8HHKBObL0EUjHeWJtadGKaakT0Rh6Qgc0boDx3mwwBG

lSl1J8/OqkQz4LYJuBWywYJu+BHcUfuKdduqDEfzgU83wwv1izRB5kZeWvSuu/+1

dz4p7yAB7mWC/IlafqR7WRmURWWhVhzMQFx3mr8wJZpL2MgbnF/zI2cGgytNgw6L

6/zj4l2DbSxouc6TrPtUtSK86Z+v4Ryi6waJGh/FglQQy8Ro+PMxT/gBvT7v3bwe

P3NBgqk3ncF8RMsQhWjlcuPWZX0cgLIlJ/hs2e5+HUrxzKInsQjI8Q==

-----END CERTIFICATE-----

8. Ivanti Tunnel for Android device user experience

The following provide some information on the device user experience for Ivanti Tunnel for Android:

- Ivanti Tunnel installation on devices
- Controlling VPN traffic
- Troubleshooting

8.1. Ivanti Tunnel installation on devices

The Ivanti Tunnel app is installed automatically on Android devices if the following conditions are true:

- The app configuration has the Silently Install option selected.
- The app is applied to a label that includes the Android devices.

If the Silently Install option is not selected, users can choose to install the app from the app catalog on the device.

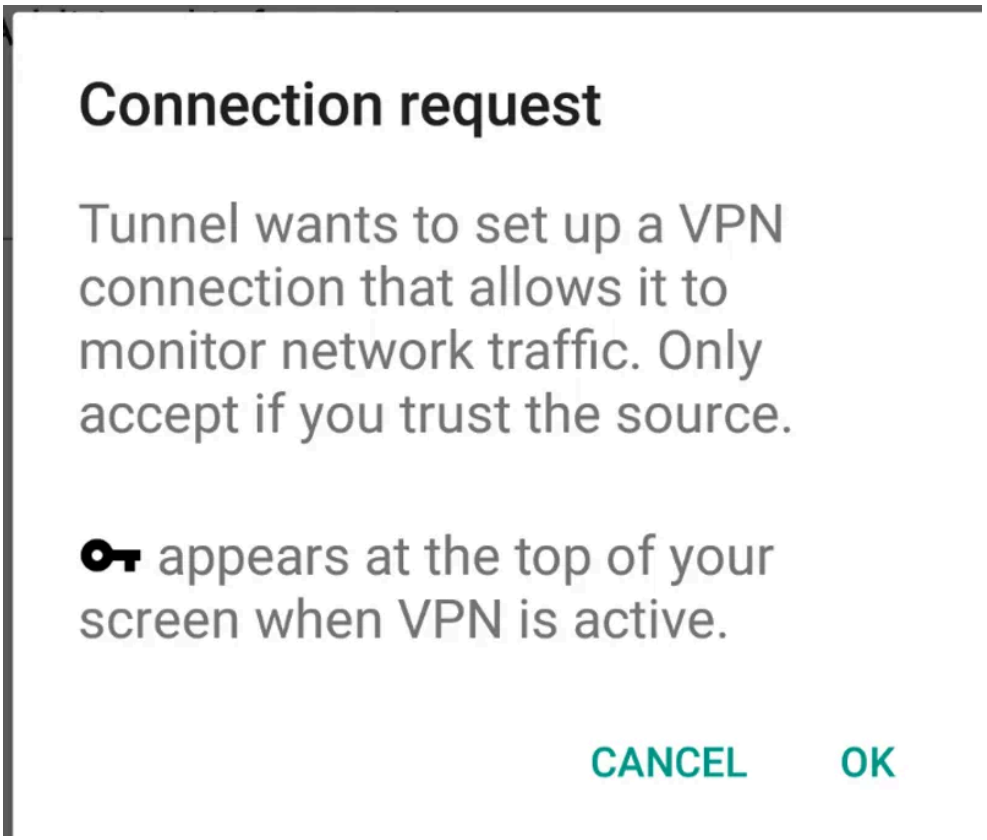
The following section proved some screen captures of what users see when they install Ivanti Tunnel:

- **Accept Ivanti Tunnel connection (Android native and enterprise only)**
- **Allow certificate (Android native and enterprise only)**
- **Ivanti Tunnel VPN connection**

Accept Ivanti Tunnel connection (Android native and enterprise only)

The first time that Ivanti Tunnel attempts to set up a VPN connection, device users are prompted to accept the Tunnel VPN connection. Device users must tap OK to continue using Ivanti Tunnel.

Figure 1. Accept Tunnel connection

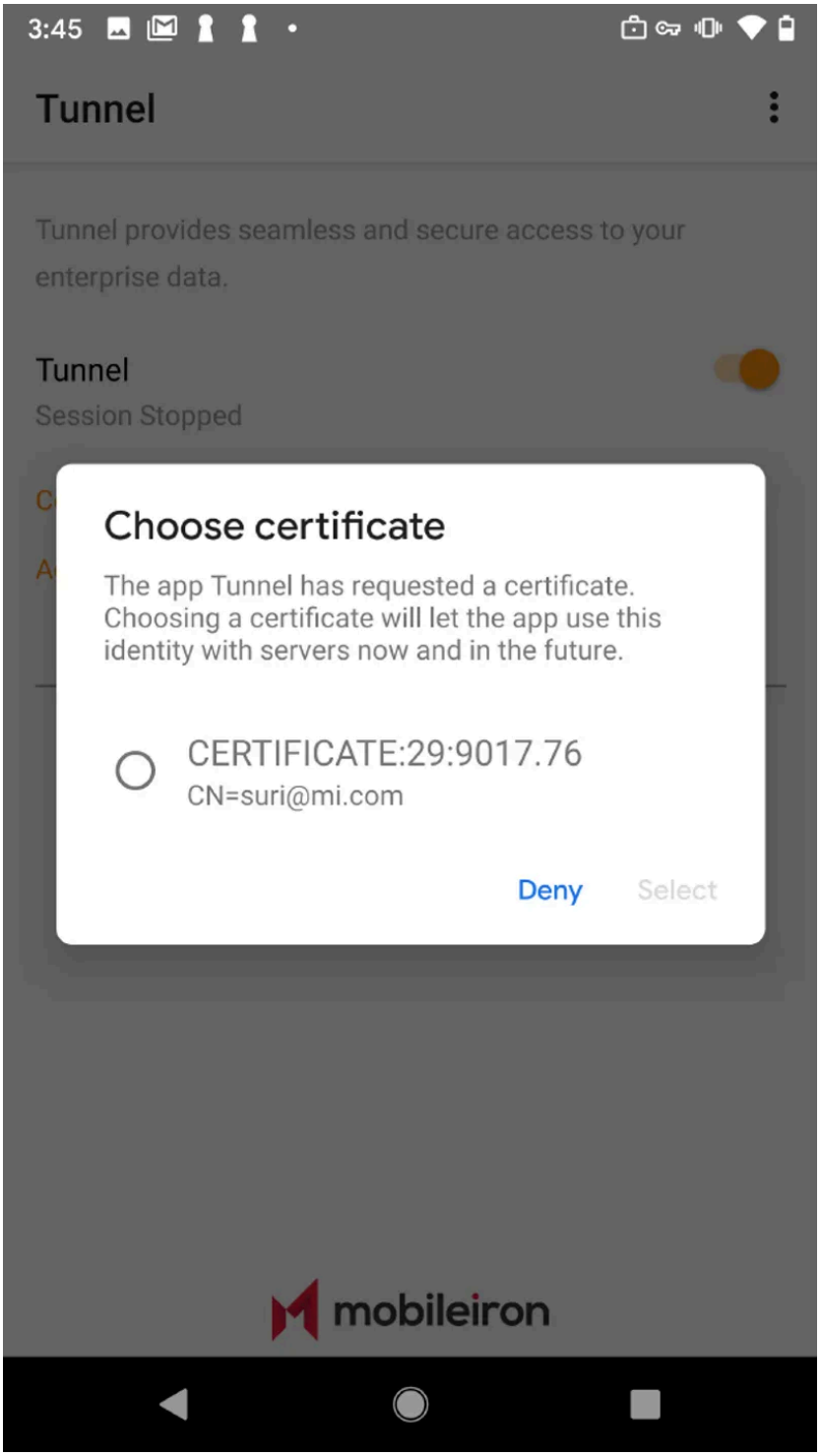


Allow certificate (Android native and enterprise only)

During the installation, users are prompted to accept a certificate. The certificate is preselected.

Do not change the certificate selection.

Figure 2. Allow certificate



Tap Select to install the certificate on the device and continue with the installation. Ivanti Tunnel uses this certificate to authenticate the device to Standalone Sentry.

On Android Enterprise devices, user interaction is not needed to accept the Ivanti Tunnel certificate for the following cases,

- Always On VPN (in Android Enterprise configuration on the UEM) and AutoBackgroundLaunch (Ivanti Tunnel for Android Enterprise configuration on the UEM) are enabled.

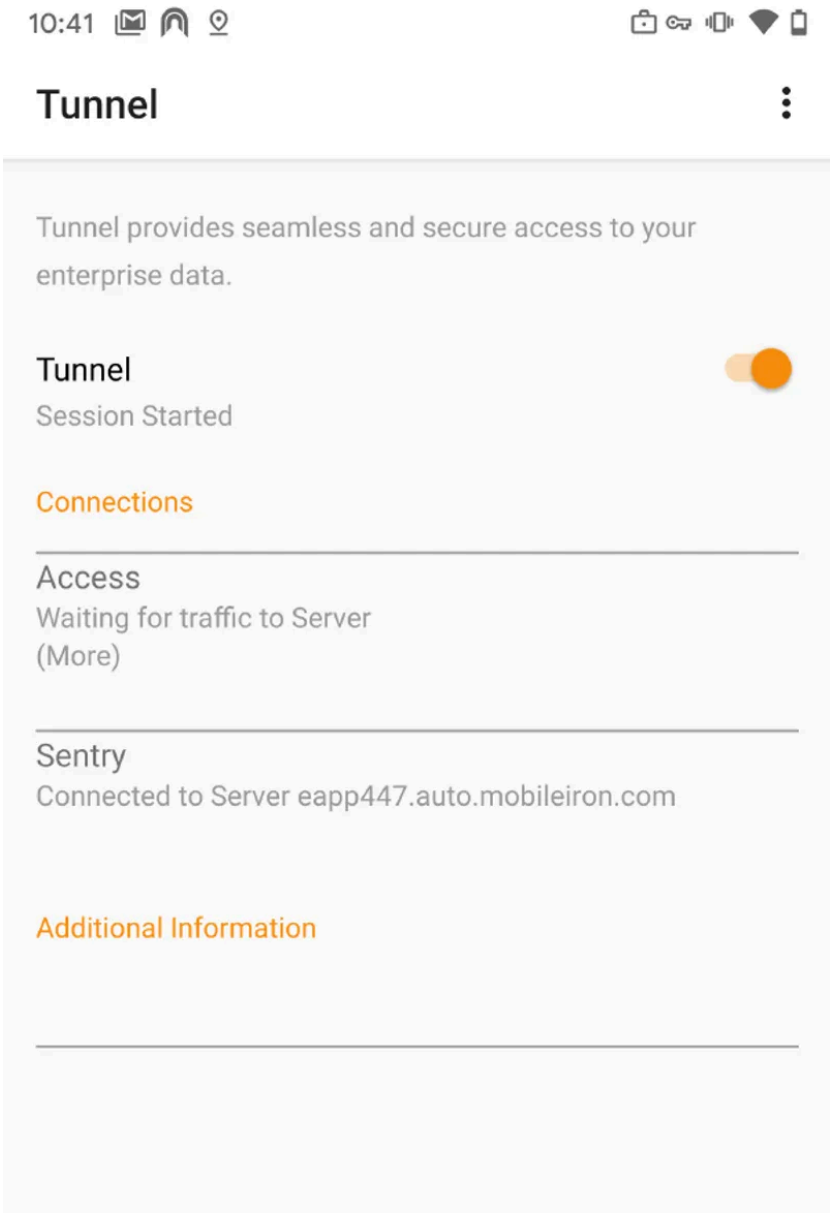
- The client version on the device is Go 72 for Android or Mobile@Work 10.8.0.0 for Android through the most recently released version as supported by Ivanti. In this case, the Choose Certificate dialog is not seen. The UEM client silently accepts the certificate. This change improves the user experience when the device is in shared-kiosk mode.

Ivanti Tunnel VPN connection

The key icon on Android native and Android Enterprise devices, or the lock icon on Samsung Knox devices, indicates that the Ivanti Tunnel VPN configuration has been pushed and verified without any errors, and the VPN session has been established. This does not indicate if Tunnel is connected or not. The location and the icon can vary depending on the device and Android version.

The state of the Tunnel session after it is initiated remains as Started. With an Access deployment, the connection status changes from Connected to Waiting periodically if there is no Access traffic going through Ivanti Tunnel.

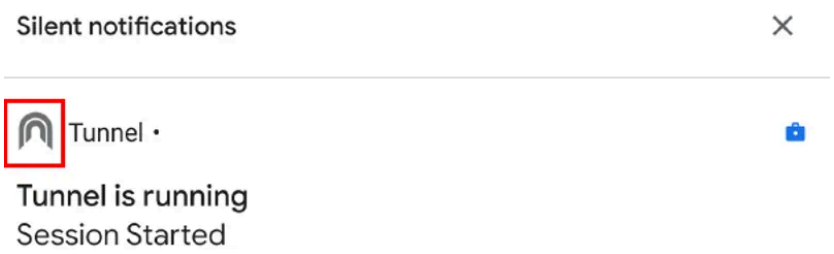
Figure 3. Tunnel VPN connection state and icons



Ivanti Tunnel notifications icon

If Ivanti Tunnel notifications is enabled, users see the following icon if there are any notifications. On Android O through the latest version as supported by Ivanti, the ability to configure notifications in the Ivanti Tunnel app is not available. On these devices, configure notification in Android Settings.

Figure 4. Tunnel notifications icon



The icon is not visible if there are no notifications.

8.2. Troubleshooting

The Ivanti Tunnel app collects device and traffic logs to help with troubleshooting. The Ivanti Tunnel configurations from the UEM are also available to view in the app.

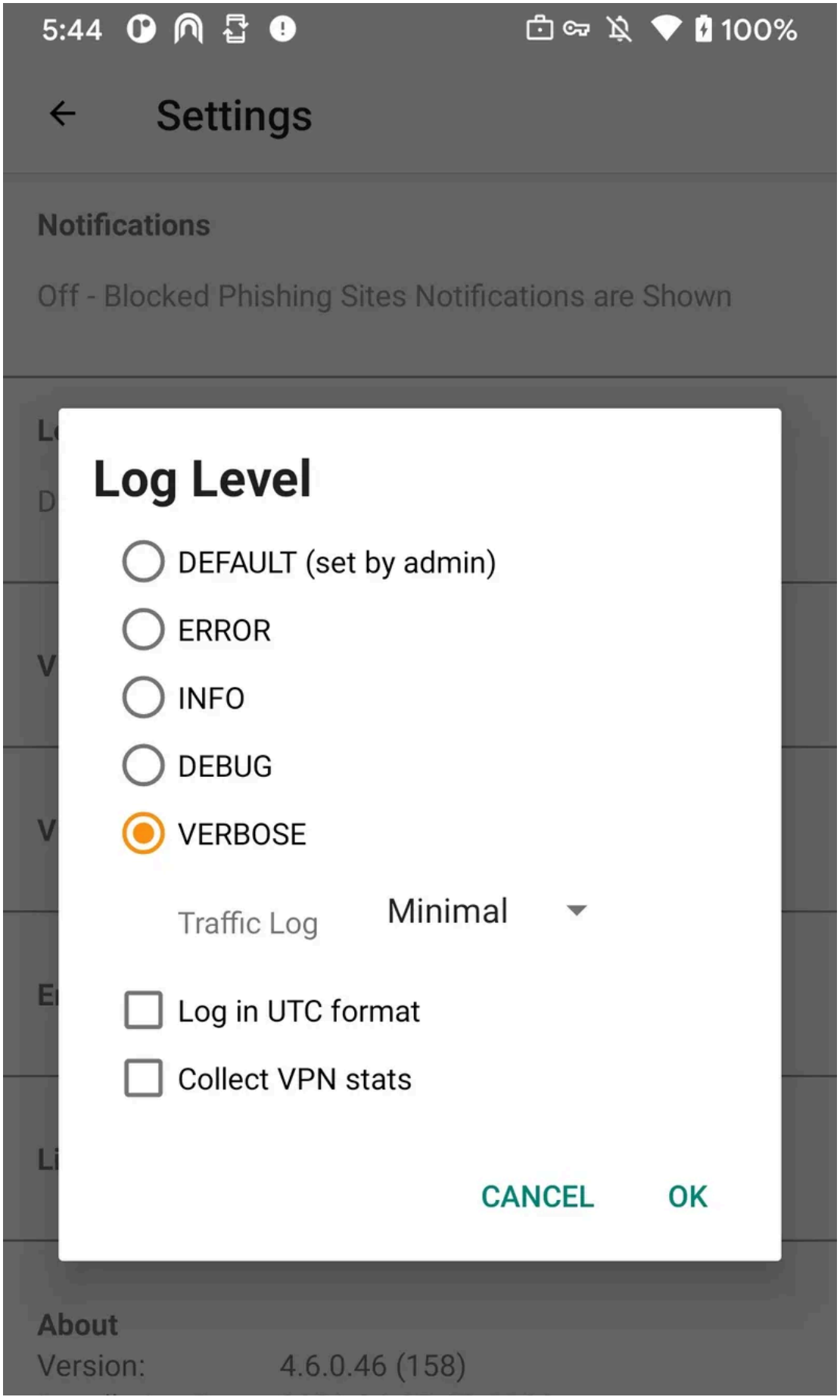
- Collecting log and PCAP files
- Viewing logs
- Clearing logs
- Viewing Ivanti Tunnel configuration

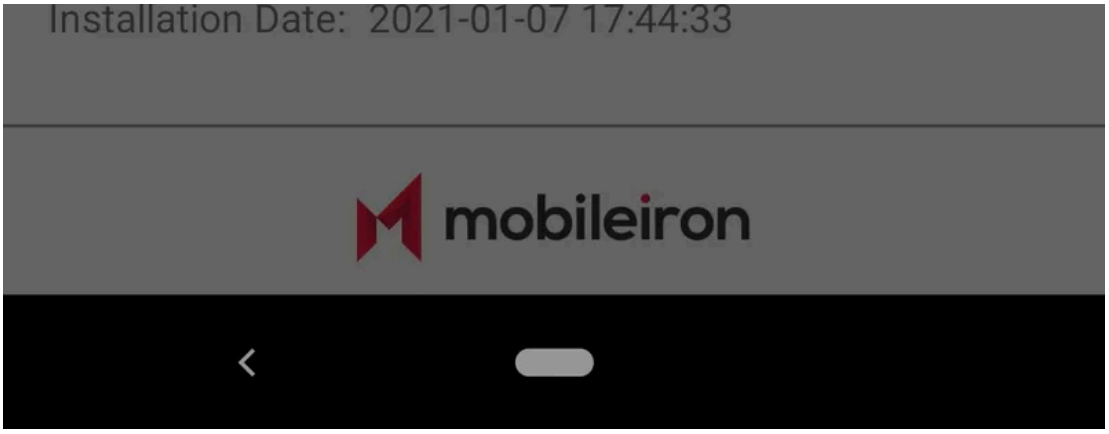
Collecting log and PCAP files

You can collect tunnel log and PCAP files to help with diagnostics and troubleshooting.

Procedure

1. In the Ivanti Tunnel configuration (VPN configuration for Android native or app configuration for Android Enterprise) on your UEM (Ivanti EPMM or Ivanti Neurons for MDM), set the following:
   - AllowCapture to **true**
   - UINotificationLevel
   - DebugLog
   - TrafficVerboseLog
   - debugInfoRecipient
2. Force the device to check in.
3. In the Tunnel app go to **Settings** and select a log level.
   If you select Verbose, you also have the option to select the Traffic Log level as Off, Minimal, or ALL, The traffic log level is disabled (Off) by default.
   Due to the amount of data that is collected when you select Verbose, you will notice a decline in Tunnel performance.
   Figure 1. Tunnel log level


5:44
100%
Settings
Notifications
Off - Blocked Phishing Sites Notifications are Shown
Log Level
DEFAULT (set by admin)
ERROR
INFO
DEBUG
VERBOSE
Traffic Log
Minimal
Log in UTC format
Collect VPN stats
CANCEL
OK
About
Version:
4.6.0.46 (158)



1. To collect PCAP files, under Capture Traffic, check **Enable**.
2. Tap device **Settings > Email Logs**.
   The default email client for the device will be opened. The log files and PCAP files will be compressed and attached to an email.

Related topics

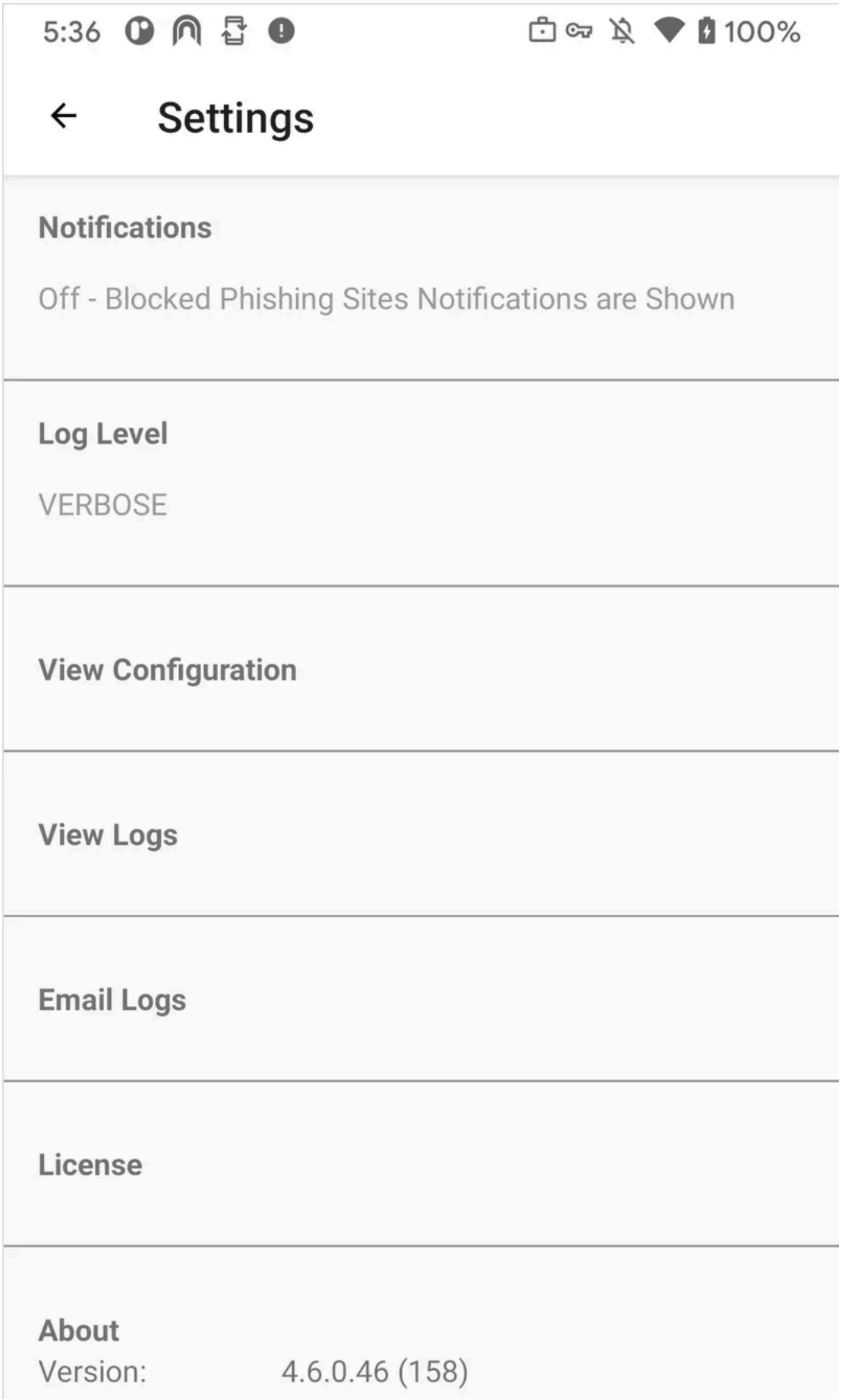
For a description of the custom data, see **Ivanti Tunnel Configuration Fields and Custom Data**.

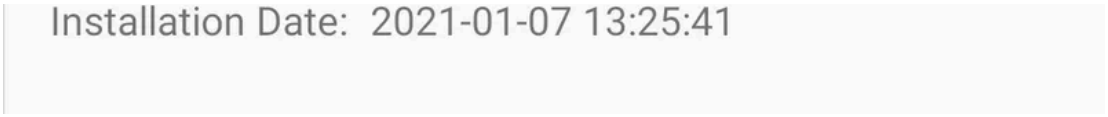
Viewing logs

You can view logs on the Ivanti Tunnel app.

Procedure

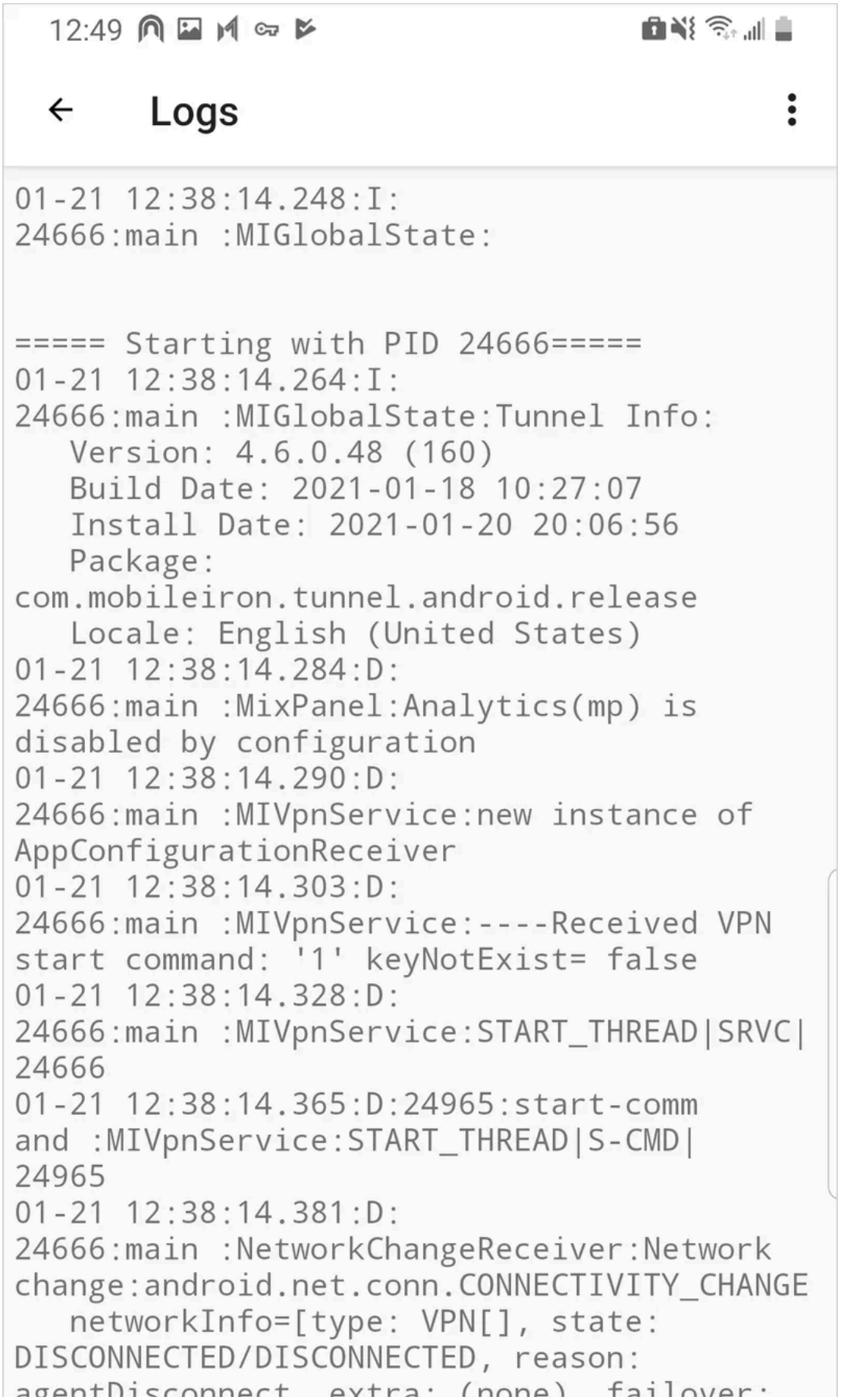
1. In the Ivanti Tunnel app, tap on the three vertical dots to expand the menu.
   Figure 2. Tunnel settings

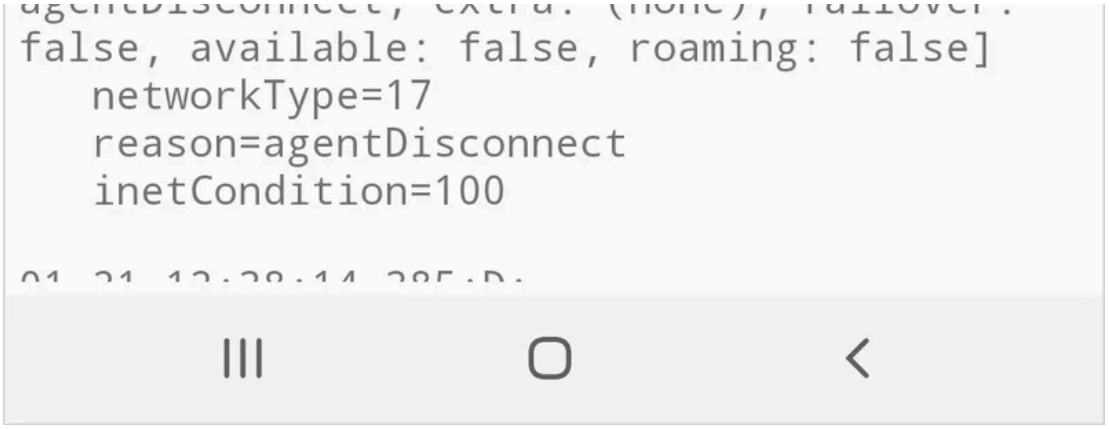

5:36
100%
Settings
Notifications
Off - Blocked Phishing Sites Notifications are Shown
Log Level
VERBOSE
View Configuration
View Logs
Email Logs
License
About
Version: 4.6.0.46 (158)



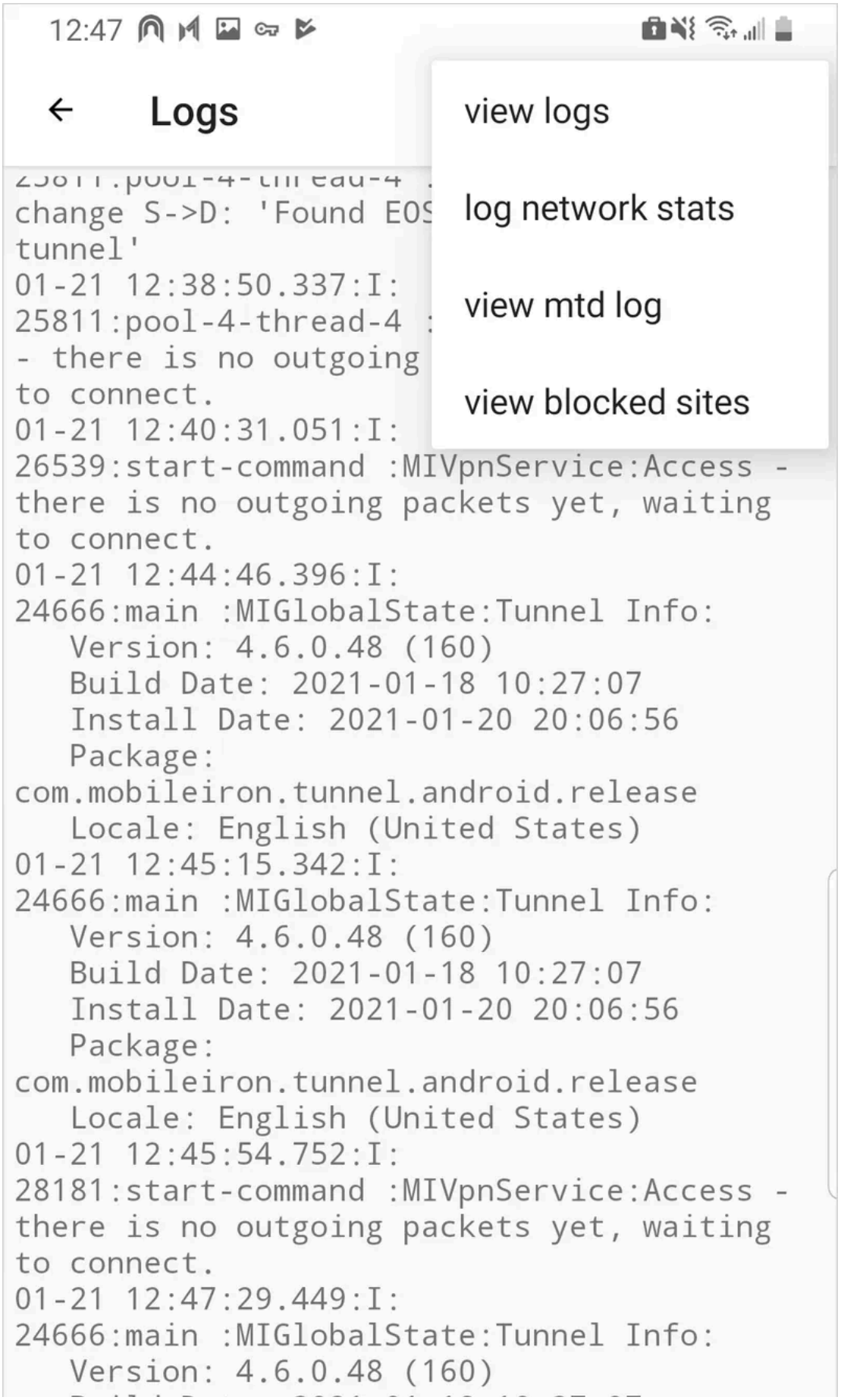
1. Tap Settings > Logs.

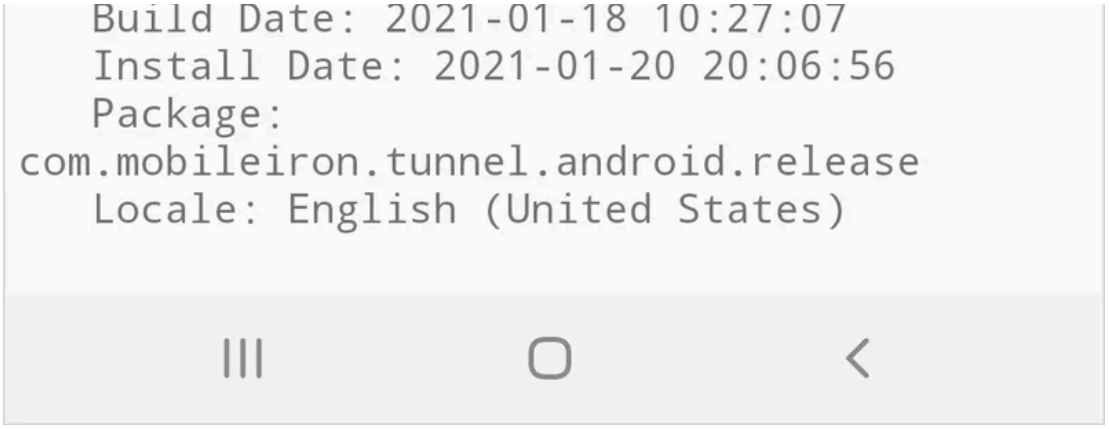
Figure 3. Tunnel configuration


12:49
Logs
01-21 12:38:14.248:I:
24666:main :MIGlobalState:
===== Starting with PID 24666=====
01-21 12:38:14.264:I:
24666:main :MIGlobalState:Tunnel Info:
Version: 4.6.0.48 (160)
Build Date: 2021-01-18 10:27:07
Install Date: 2021-01-20 20:06:56
Package:
com.mobileiron.tunnel.android.release
Locale: English (United States)
01-21 12:38:14.284:D:
24666:main :MixPanel:Analytics(mp) is
disabled by configuration
01-21 12:38:14.290:D:
24666:main :MIVpnService:new instance of
AppConfigurationReceiver
01-21 12:38:14.303:D:
24666:main :MIVpnService:----Received VPN
start command: '1' keyNotExist= false
01-21 12:38:14.328:D:
24666:main :MIVpnService:START_THREAD|SRVC|
24666
01-21 12:38:14.365:D:24965:start-comm
and :MIVpnService:START_THREAD|S-CMD|
24965
01-21 12:38:14.381:D:
24666:main :NetworkChangeReceiver:Network
change:android.net.conn.CONNECTIVITY_CHANGE
networkInfo=[type: VPN[], state:
DISCONNECTED/DISCONNECTED, reason:



1. Tap the vertical three dots on the top right corner to expand the menu.

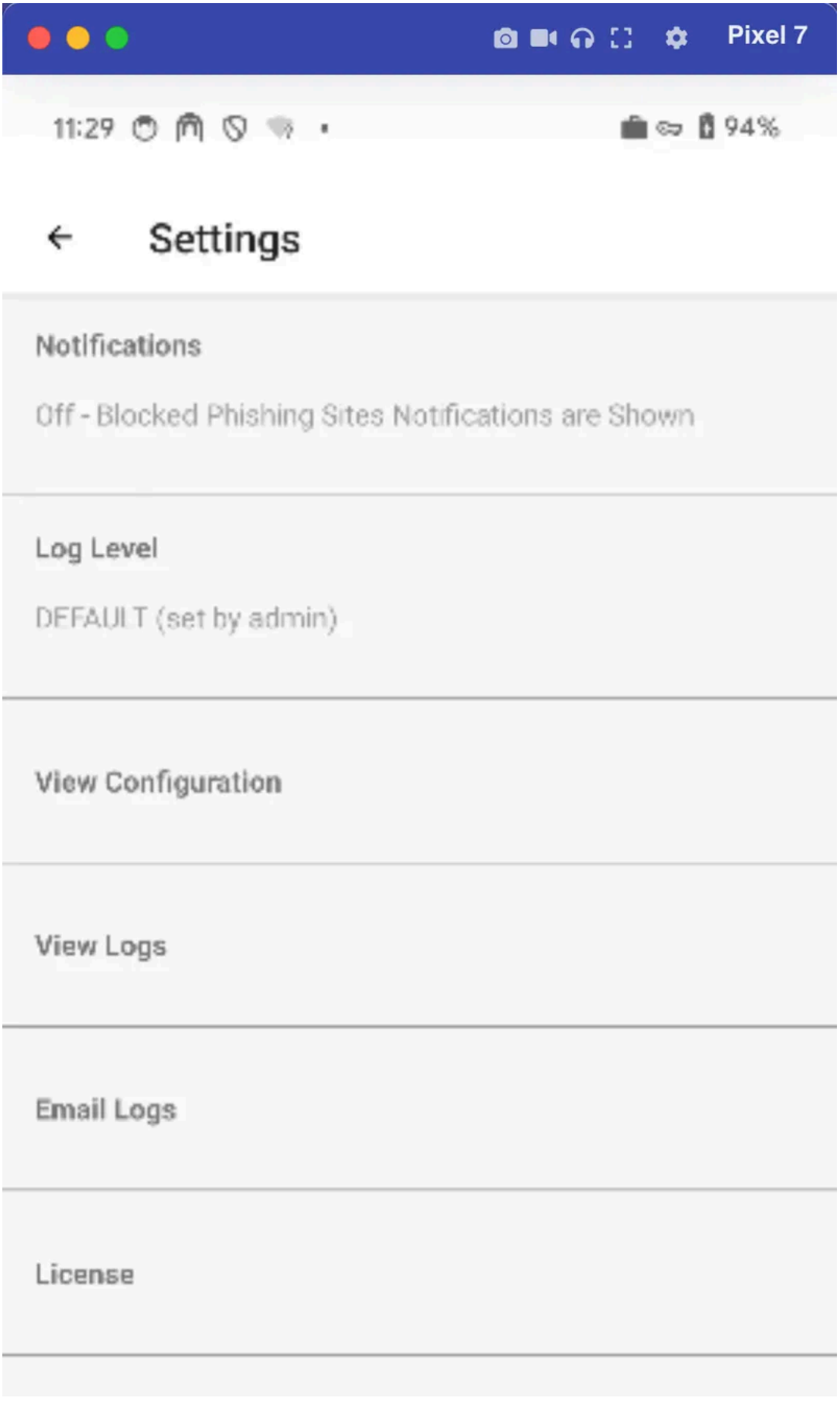

12:47
Logs
view logs
log network stats
view mtd log
view blocked sites
change S->D: 'Found EOS
tunnel'
01-21 12:38:50.337:I:
25811:pool-4-thread-4 :
- there is no outgoing
to connect.
01-21 12:40:31.051:I:
26539:start-command :MIVpnService:Access -
there is no outgoing packets yet, waiting
to connect.
01-21 12:44:46.396:I:
24666:main :MIGlobalState:Tunnel Info:
Version: 4.6.0.48 (160)
Build Date: 2021-01-18 10:27:07
Install Date: 2021-01-20 20:06:56
Package:
com.mobileiron.tunnel.android.release
Locale: English (United States)
01-21 12:45:15.342:I:
24666:main :MIGlobalState:Tunnel Info:
Version: 4.6.0.48 (160)
Build Date: 2021-01-18 10:27:07
Install Date: 2021-01-20 20:06:56
Package:
com.mobileiron.tunnel.android.release
Locale: English (United States)
01-21 12:45:54.752:I:
28181:start-command :MIVpnService:Access -
there is no outgoing packets yet, waiting
to connect.
01-21 12:47:29.449:I:
24666:main :MIGlobalState:Tunnel Info:
Version: 4.6.0.48 (160)

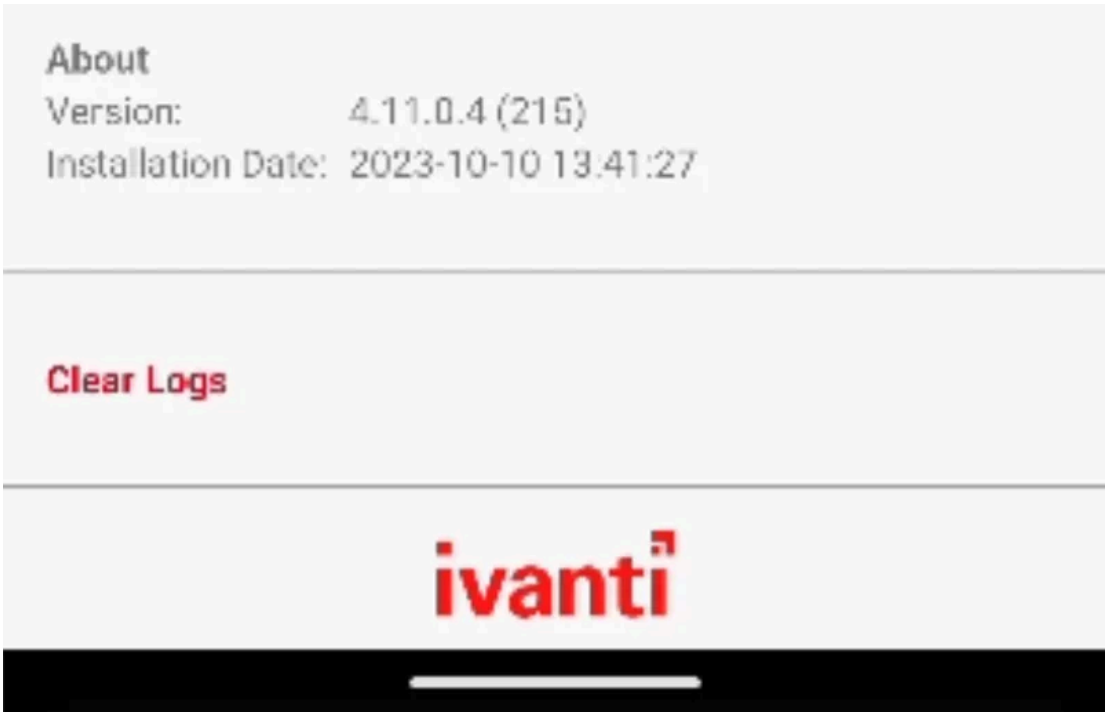


1. Tap an option from the menu to view a filtered set.

Clearing logs

The **Clear Logs** option is now added to Tunnel for Android apps **Settings** section.


Pixel 7
11:29
94%
Settings
Notifications
Off - Blocked Phishing Sites Notifications are Shown
Log Level
DEFAULT (set by admin)
View Configuration
View Logs
Email Logs
License



Using this option logs will be cleared for the following files:

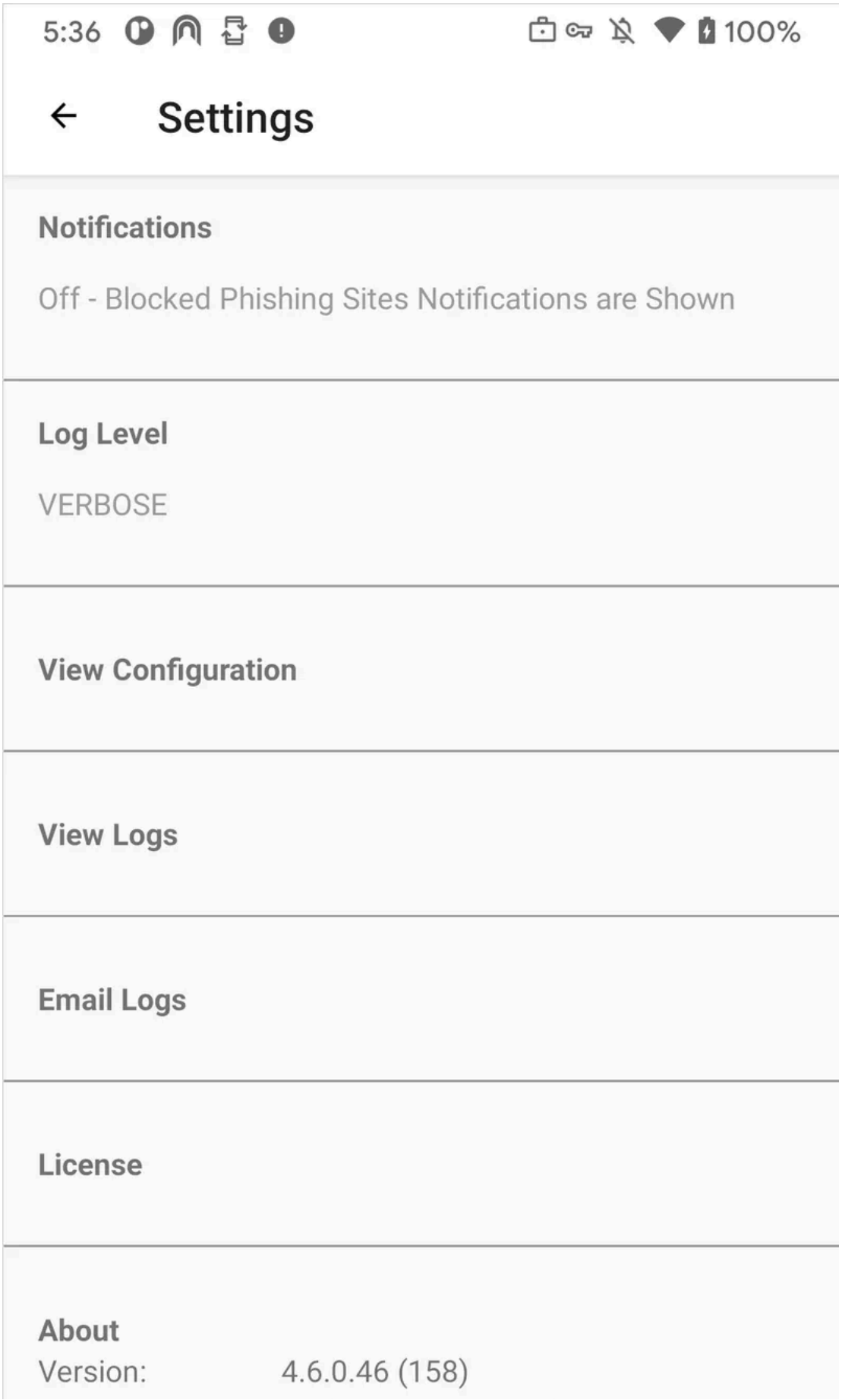
- Tunnel Log file
- MTD Log File
- PCAP Log file

Viewing Ivanti Tunnel configuration

You can view Ivanti Tunnel configuration in the Tunnel app.

Procedure

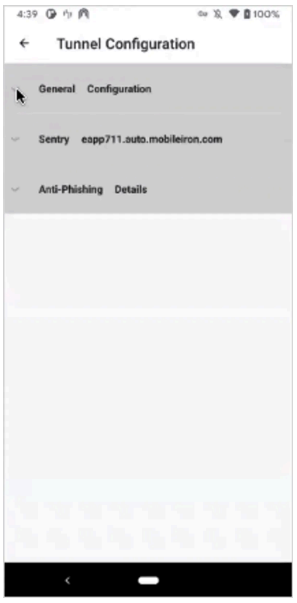
1. In the Ivanti Tunnel app, tap on the three vertical dots to expand the menu.
   Figure 4. Tunnel settings


5:36
100%
Settings
Notifications
Off - Blocked Phishing Sites Notifications are Shown
Log Level
VERBOSE
View Configuration
View Logs
Email Logs
License
About
Version: 4.6.0.46 (158)

Installation Date: 2021-01-07 13:25:41

1. Tap Settings > View Tunnel Configuration.

   Figure 5. Tunnel configuration



The Ivanti Tunnel Configuration information is grouped under General configuration, Access, Standalone Sentry, and Anti-phishing. Expand the section to view configuration details.

8.3. Anti-phishing enabled user experience

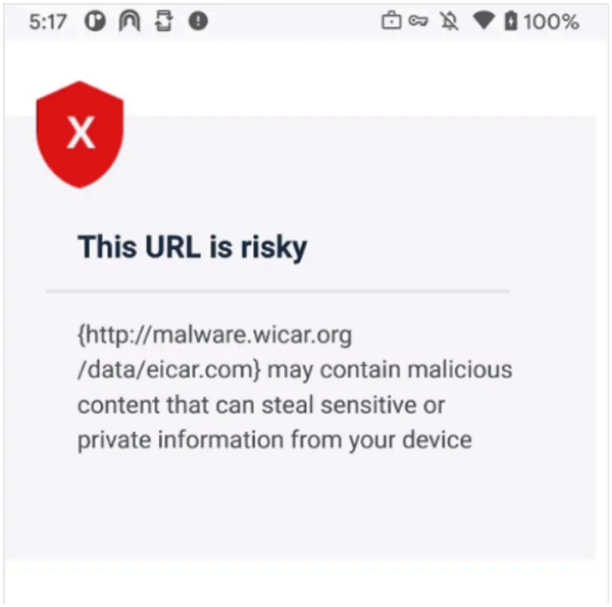
Anti-phising support with Threat Defense provides an additional layer of protection to users. For more information see **Support for anti-phishing with Mobile Threat Defense**.

If the anti-phishing feature is configured, the user experience differs depending on whether the link URL is an HTTP URL or an HTPPS URL.

HTTP phishing URL

When users tap on an unsafe HTTP link, instead of the link URL, users see a browser page notifying users that they are accessing an unsafe page.

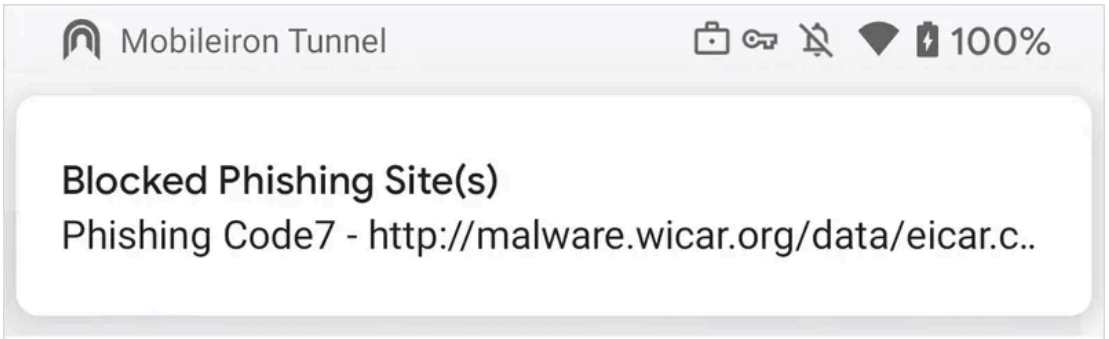
Figure 1. HTTP phishing URL

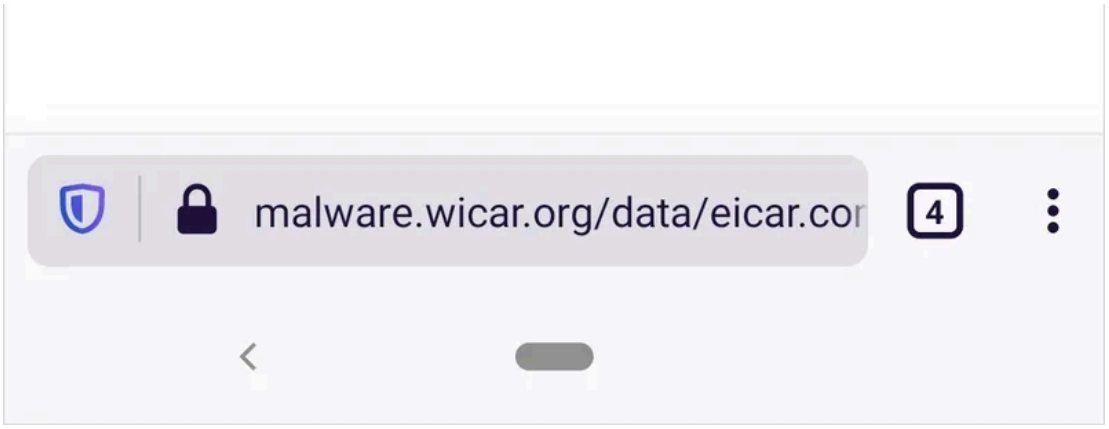


HTTPS phishing URL

When users tap on an unsafe HTTPS link, users see a heads up notification that they are accessing an unsafe page and the link URL is blocked. For subsequent requests, by default, the page continues to be blocked but the heads up notification is not seen.

Figure 2. HTTPS phishing URL

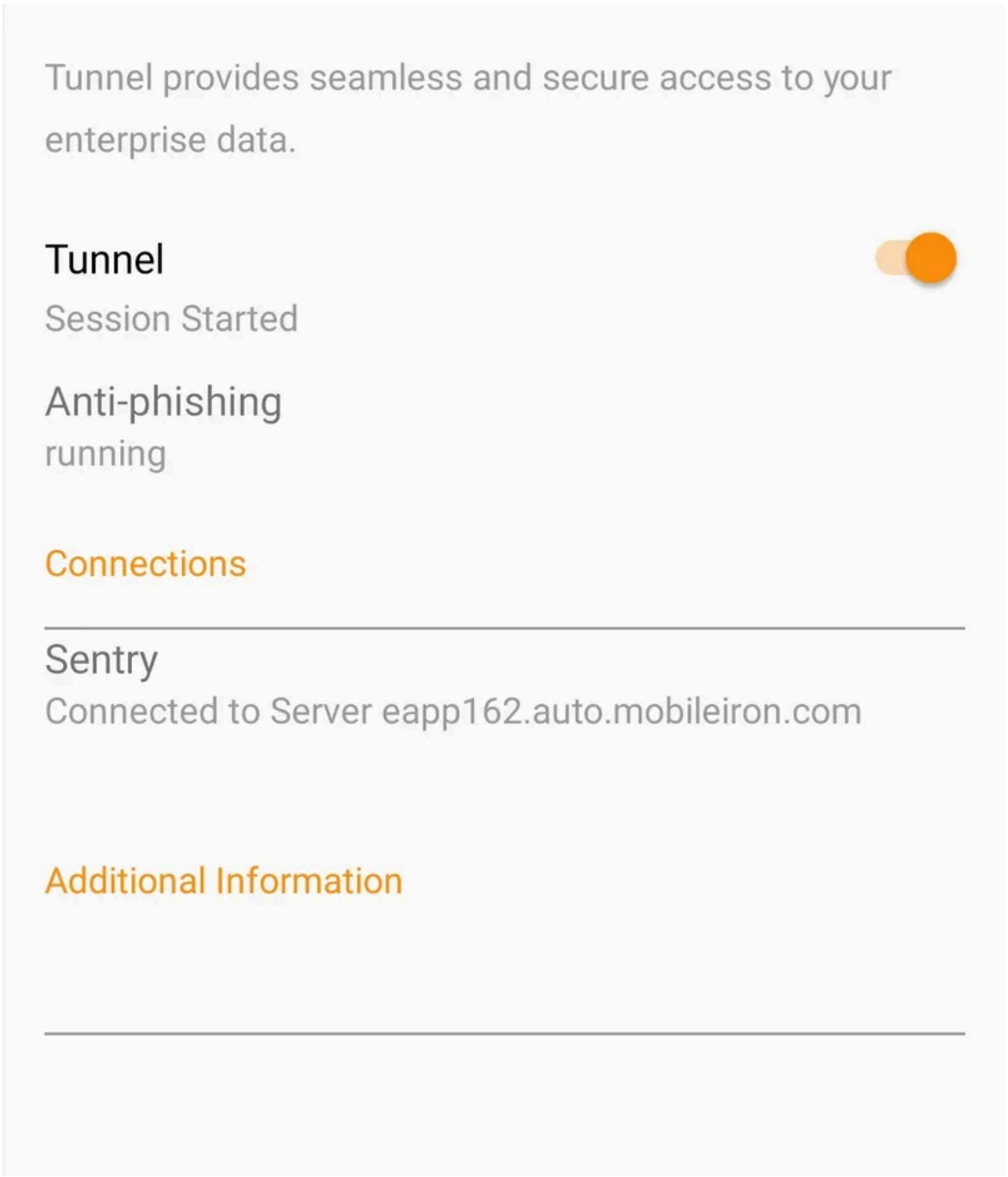

Mobileiron Tunnel
100%
Blocked Phishing Site(s)
Phishing Code7 - http://malware.wicar.org/data/eicar.c..



Anti-phishing status

You can check whether anti-phishing is enabled or running by launching the Ivanti Tunnel app. If anti-phising support for Android devices is not configured, the status for Anti-phising displays as Disabled. If anti-phishing is configured and running, the status displays as Running. In some cases you may see the status as Not running. This may be due to a time lag for the configuration to propagate or there may be a misconfiguration.

Figure 3. Anti-phising status


Tunnel provides seamless and secure access to your enterprise data.
Tunnel
Session Started
Anti-phishing
running
Connections
Sentry
Connected to Server eapp162.auto.mobileiron.com
Additional Information