



MobileIron Tunnel 4.4.0 for Android Guide for Administrators

for Android native, Android enterprise, and Samsung Knox for MobileIron Core and MobileIron Cloud

July 13, 2020

For complete product documentation see:

[Tunnel for Android Product Documentation Home Page](#)

Copyright © 2015 - 2020 MobileIron, Inc. All Rights Reserved.

Any reproduction or redistribution of part or all of these materials is strictly prohibited. Information in this publication is subject to change without notice. MobileIron, Inc. does not warrant the use of this publication. For some phone images, a third-party database and image library, Copyright © 2007-2009 Aeleeeta's Art and Design Studio, is used. This database and image library cannot be distributed separate from the MobileIron product.

“MobileIron,” the MobileIron logos and other trade names, trademarks or service marks of MobileIron, Inc. appearing in this documentation are the property of MobileIron, Inc. This documentation contains additional trade names, trademarks and service marks of others, which are the property of their respective owners. We do not intend our use or display of other companies' trade names, trademarks or service marks to imply a relationship with, or endorsement or sponsorship of us by, these other companies.



Contents

Contents	3
New Features and Enhancements	6
About MobileIron Tunnel	7
MobileIron Tunnel overview	7
MobileIron Tunnel for Android support on MobileIron UEM	7
MobileIron Tunnel configuration	8
Tunnel creation with MobileIron Tunnel for Android	8
App traffic allowed through Tunnel VPN (Android native and Android enterprise)	9
Tunnel routes and MobileIron Tunnel for Android	10
DNS servers and MobileIron Tunnel for Android	10
Always-on Tunnel VPN and MobileIron Tunnel for Android	10
Connection recovery for MobileIron Tunnel for Android	10
Send and receive IP packets with MobileIron Tunnel	11
Tunnel connectivity probe	11
Setting up MobileIron Tunnel for Android native	12
Before you configure MobileIron Tunnel for Android native (Core and Cloud)	12
Required components for Tunnel for Android native	12
Requirements for Tunnel for Android native	12
Recommendations for Tunnel for Android native	13
Limitations for Tunnel for Android native	13
Configuration tasks overview for MobileIron Tunnel for Android native (Core)	13
Creating a MobileIron Tunnel VPN configuration (Core)	14
Distributing through the MobileIron app storefront	14
Configuration tasks overview for MobileIron Tunnel for Android native (Cloud)	15
Creating a MobileIron Tunnel VPN configuration for Android native (Cloud)	15
Adding and configuring the MobileIron Tunnel app (Cloud)	16



Setting up MobileIron Tunnel for Samsung Knox in MobileIron Core	18
Before you configure MobileIron Tunnel for Samsung Knox	18
Required components for Tunnel for Samsung Knox	18
Requirements for Tunnel for Samsung Knox	19
Recommendations for Tunnel for Samsung Knox	19
Limitations for Tunnel for Samsung Knox	19
Configuration overview for MobileIron Tunnel for the Samsung Knox container (Core)	20
Error messages for a per-container and on-demand Tunnel VPN setup	21
Configuring an IP_ANY AppTunnel service on a Standalone Sentry	22
Creating a MobileIron Tunnel VPN configuration for Samsung Knox Workspace (Core)	23
Distributing Tunnel through Apps@Work	24
Configuring app VPN in the Samsung Knox container	25
Configuring per-app VPN	26
Configuring per-container VPN	26
Configuring VPN chaining	27
Setting up MobileIron Tunnel for Android enterprise	29
Before you configure MobileIron Tunnel for Android enterprise (Core and Cloud)	29
Required components for deploying Tunnel for Android enterprise	29
Requirements for deploying Tunnel for Android enterprise	30
Recommendations for deploying Tunnel for Android enterprise	30
Limitations for Tunnel for Android enterprise	31
Configuration tasks overview for Android enterprise (Core)	31
Adding and configuring the MobileIron Tunnel for Android enterprise app (Core)	31
Creating an Always-On VPN configuration (Core, optional)	32
Configuration tasks overview for Android enterprise (Cloud)	33
Adding and configuring the MobileIron Tunnel app (Cloud)	33
Creating an Always-On VPN configuration (Cloud, optional)	34
Tunnel Configuration Fields and Custom Data	36
Tunnel for Android native configuration field description	36



Custom data key-value pairs for Tunnel for Android native and Samsung Knox Workspace	39
Tunnel configuration field description for Android enterprise	44
Example showing the Sentry certificate in the certificate chain	50
MobileIron Tunnel for Android device user experience	53
MobileIron Tunnel installation on devices	53
Accept Tunnel connection (Android native and enterprise only)	53
Allow certificate (Android native and enterprise only)	54
Tunnel VPN connection	54
Tunnel notifications icon	55
Controlling VPN traffic	56
Troubleshooting	57
Collecting log and PCAP files	57
Viewing Tunnel configuration	58



New Features and Enhancements

This guide documents the following new features and enhancements:

- **Rebranding:** MobileIron has updated the Tunnel for Android icon and user interface color scheme. For more information see the Knowledge Base article [Coming Soon - MobileIron UX changes MobileIron Tunnel Android and iOS App](#). For updated screen captures, see [MobileIron Tunnel for Android device user experience](#).
- **Report device ID to MobileIron Access:** For Access deployments, Tunnel reports the device ID to MobileIron Access if the key SendDeviceID is configured in the Tunnel VPN configuration with the value true. The device ID is reported on MobileIron Access in **Reports > Errors**. The key-value pair is useful in identifying devices that encounter connection errors when authenticating through Access. See [Custom data key-value pairs for Tunnel for Android native and Samsung Knox Workspace](#).



About MobileIron Tunnel

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

- [MobileIron Tunnel overview](#)
- [MobileIron Tunnel for Android support on MobileIron UEM](#)
- [MobileIron Tunnel configuration](#)
- [Tunnel creation with MobileIron Tunnel for Android](#)
- [App traffic allowed through Tunnel VPN \(Android native and Android enterprise\)](#)
- [Send and receive IP packets with MobileIron Tunnel](#)

MobileIron Tunnel overview

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

MobileIron Tunnel interacts with the MobileIron Unified Endpoint Management (UEM) platform, MobileIron Standalone Sentry, and MobileIron 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. MobileIron provides the following UEM platforms: MobileIron Core and MobileIron Cloud.

MobileIron Tunnel for Android support on MobileIron UEM

The following table describes Tunnel for Android support on the MobileIron unified endpoint management (UEM) platform.



TABLE 1. MOBILEIRON UEM SUPPORT

	MobileIron Core	MobileIron Cloud
Android native	Supported	Supported
Android enterprise	Supported	Supported NOTE: Only work profile and work managed device modes are supported.
Samsung Knox	Supported	Not supported

MobileIron Tunnel configuration

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

IMPORTANT: MobileIron recommends that you do not configure Tunnel for Android native and for Android enterprise on the same Core.

Tunnel creation with MobileIron Tunnel for Android

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

1. Tunnel validates the configuration syntactically.
2. Tunnel establishes a TCP connection with Standalone Sentry on port 443.
3. 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 Tunnel configuration. This step occurs if certificate pinning is enabled.
5. Tunnel initiates the MobileIron AppTunnel protocol handshake:
 - a. POST with device ID, user ID, and service ID are sent to Standalone Sentry.
 - b. Standalone Sentry validates the parameters. For example, Standalone Sentry checks if the user or device is blocked.
 - c. Standalone Sentry provides additional configuration parameters: interface IP and DNS server IP.
 - d. The TCP connection is switched to the MobileIron Tunnel protocol.
6. A VPN session is created using Android API VpnService.Builder.
 - a. VPN specific configuration is set in the VPN session based on the Tunnel configuration created in MobileIron UEM.



- b. 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 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.

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

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

App traffic allowed through Tunnel VPN (Android native and Android enterprise)

When a Tunnel VPN session is created, the Tunnel configuration is provided to the Android operating system. The Tunnel configuration includes information such as allowed and disallowed apps, routes, and domain name servers. Android enforces access to Tunnel, based on the provided configuration. The apps that use MobileIron 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 Tunnel. Traffic from all other apps is not allowed to go through Tunnel and goes through the device network.
- Disallowed: All apps have access to 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 app list. If an allowed list is not configured, MobileIron strongly suggests adding at least the following to a disallowed list to avoid OS traffic going through Tunnel VPN:

- Mobile@Work if your UEM is MobileIron Core (com.mobileiron)
- MobileIron Go if your UEM is MobileIron Cloud (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)
- All apps that have been enabled by default in the “Allow App while Data Saver On” or “Unrestricted data access” list under the data saver settings. These are mandatory apps required for the Android system.

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



- [Tunnel routes and MobileIron Tunnel for Android](#)
- [DNS servers and MobileIron Tunnel for Android](#)
- [Always-on Tunnel VPN and MobileIron Tunnel for Android](#)
- [Connection recovery for MobileIron Tunnel for Android](#)

Tunnel routes and MobileIron 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 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 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 Tunnel.

DNS servers and MobileIron 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 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 MobileIron Tunnel for Android

On Android 5 and 6 devices, always-on is a MobileIron 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 MobileIron, 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 MobileIron Core and in the Always-on configuration in MobileIron Cloud.

Connection recovery for MobileIron Tunnel for Android

If a connection fails, Tunnel tries to reconnect periodically, by default. 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.



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

Send and receive IP packets with MobileIron 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 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.

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

Tunnel connectivity probe

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 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: `AtpProbeldieSec`, `AtpProbeIntervalSec`, `AtpProbeCount`. For information about the key-value pairs, see [Custom data key-value pairs for Tunnel for Android native and Samsung Knox Workspace](#).



Setting up MobileIron Tunnel for Android native

The following address the setup required for app VPN using MobileIron Tunnel for Android native in MobileIron Core and MobileIron Cloud:

- [Before you configure MobileIron Tunnel for Android native \(Core and Cloud\)](#)
- [Configuration tasks overview for MobileIron Tunnel for Android native \(Core\)](#)
- [Configuration tasks overview for MobileIron Tunnel for Android native \(Cloud\)](#)

Before you configure MobileIron Tunnel for Android native (Core and Cloud)

Before you configure 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 Tunnel for Android native](#)
- [Requirements for Tunnel for Android native](#)
- [Recommendations for Tunnel for Android native](#)
- [Limitations for Tunnel for Android native](#)

Required components for Tunnel for Android native

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

- Standalone Sentry with AppTunnel enabled or MobileIron Access.
- MobileIron Unified Endpoint Management (UEM) platform: MobileIron Core or MobileIron Cloud.
- MobileIron client for Android:
 - MobileIron Core: Mobile@Work
 - MobileIron Cloud: MobileIron Go

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

Requirements for Tunnel for Android native

The following are requirements for deploying 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 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
Core: MobileIron Sentry Guide for Core.
Cloud: MobileIron Sentry Guide for Cloud.
- If your deployment uses MobileIron Access, ensure that MobileIron Access is set up. See the *MobileIron Access Guide* for information on how to set up MobileIron Access.
- Ensure that the appropriate ports are open. See the *MobileIron Tunnel for Android Release Notes*.

Recommendations for Tunnel for Android native

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

- MobileIron strongly recommends that Standalone Sentry use a publicly trusted CA certificate. Android version 7 through the latest versions as supported by MobileIron 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 Tunnel for Android native

The following are limitations of MobileIron 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 as needed.
- 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, MobileIron recommends configuring SplitUDPPortList to manage UDP traffic.

Configuration tasks overview for MobileIron Tunnel for Android native (Core)

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



1. [Creating a MobileIron Tunnel VPN configuration \(Core\)](#).
2. [Distributing through the MobileIron app storefront](#).

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 MobileIron Core.
- Ensure that you have created a certificates enrollment setting in MobileIron Core. The identity certificate generated must be trusted by the certificate chain in the certificate you uploaded to Standalone Sentry for device authentication.

Creating a MobileIron Tunnel VPN configuration (Core)

Create a Tunnel (Android) VPN configuration in the MobileIron Core Admin Portal.

Procedure

1. In the MobileIron Core 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 **MobileIron 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 MobileIron app storefront, go to [Distributing through the MobileIron app storefront](#).

Related topics

- [Tunnel for Android native configuration field description](#).
- [Custom data key-value pairs for Tunnel for Android native and Samsung Knox Workspace](#).

Distributing through the MobileIron app storefront

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

Procedure

1. In the MobileIron Core Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+**.
3. Click **Google Play**.
4. Enter “MobileIron Tunnel” for **Application Name**, and click **Search**.



5. Select the line for MobileIron 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*.

Configuration tasks overview for MobileIron Tunnel for Android native (Cloud)

The following configuration tasks are required to set up MobileIron Tunnel. These configuration tasks are performed in MobileIron Cloud.

1. [Creating a MobileIron Tunnel VPN configuration for Android native \(Cloud\)](#)
2. [Adding and configuring the MobileIron Tunnel app \(Cloud\)](#)

Before you begin

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

Creating a MobileIron Tunnel VPN configuration for Android native (Cloud)

Create a MobileIron Tunnel VPN configuration in **Configurations**.



Procedure

1. In MobileIron Cloud, go to **Configurations > +Add**.
2. Search for MobileIron Tunnel.
3. Click the **MobileIron Tunnel** configuration.
The **Create MobileIron 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 MobileIron Tunnel for Android app.

Next steps

[Adding and configuring the MobileIron Tunnel app \(Cloud\)](#).

Related topics

- For a description of the configuration fields, see [Tunnel for Android native configuration field description](#).
- For a description of the key-value pairs, see [Custom data key-value pairs for Tunnel for Android native and Samsung Knox Workspace](#).

Adding and configuring the MobileIron Tunnel app (Cloud)

Upload the MobileIron Tunnel app to MobileIron Cloud 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 MobileIron Cloud 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 MobileIron Tunnel app in the Google Play store.
5. Select the MobileIron 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.
 - a. Edit the **Name** and **Description** of the settings if necessary.
 - b. **Install on Device**: Enable Install on devices, if you want to require that the app is installed on devices.
 - c. **Silently install on Samsung KNOX and Zebra devices**: This option is not applicable to Android native apps.
 - d. **Do not show app in end user App Catalog**: Select if you do not want the app displayed in the MobileIron 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 *MobileIron Cloud Guide* or help for more information on adding apps to the MobileIron Cloud app catalog.



Setting up MobileIron Tunnel for Samsung Knox in MobileIron Core

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

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

Before you configure MobileIron 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 Tunnel for Samsung Knox](#)
- [Requirements for Tunnel for Samsung Knox](#)
- [Recommendations for Tunnel for Samsung Knox](#)
- [Limitations for Tunnel for Samsung Knox](#)

Required components for Tunnel for Samsung Knox

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

- Standalone Sentry with AppTunnel enabled.
- MobileIron Core 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.
- MobileIron Tunnel for Android.
- MobileIron client for Android: Mobile@Work.



NOTE: MobileIron Tunnel and Mobile@Work for Android are available from the Google Play store.

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

Requirements for Tunnel for Samsung Knox

The following are required for deploying Tunnel for Samsung Knox:

- Set up MobileIron Core for Samsung Knox. For more information, see the “Samsung Knox support” section in the *MobileIron Core 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 *MobileIron Sentry Guide for MobileIron Core*.
- Add the apps that will use the Tunnel VPN to the app catalog on MobileIron Core and to the Samsung Knox container. For information about adding apps to the MobileIron Core app catalog see the “Adding Google Play apps for Android” and “Apps on Samsung Knox devices” sections in the *MobileIron Core Apps@Work Guide*.

Recommendations for Tunnel for Samsung Knox

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

Limitations for Tunnel for Samsung Knox

The following are limitations of MobileIron 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 MobileIron Core are ignored by Samsung Knox Workspace. Route lists are not supported in the Knox Workspace. All traffic from an app that uses 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, MobileIron recommends configuring SplitUDPPortList to manage UDP traffic.



Configuration overview for MobileIron Tunnel for the Samsung Knox container (Core)

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

1. [Configuring an IP_ANY AppTunnel service on a Standalone Sentry.](#)
2. [Creating a MobileIron Tunnel VPN configuration for Samsung Knox Workspace \(Core\).](#)
3. [Distributing Tunnel through Apps@Work.](#)
4. [Configuring app VPN in the Samsung Knox container.](#)

The VPN configuration for MobileIron Tunnel is done in two separate configurations in the MobileIron Core Admin Portal: the VPN configuration for **MobileIron Tunnel (Samsung Knox Workspace)** and the **Samsung KNOX Container** configuration. The MobileIron 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, 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 MobileIron Tunnel VPN. With on-demand VPN, a Tunnel VPN connection is started when an app that uses MobileIron Tunnel is launched, and the connection stays on till the last app that can use the Tunnel VPN is killed.

NOTE: MobileIron 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 MobileIron Tunnel.

The following table describes MobileIron 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.



TABLE 2. MOBILEIRON TUNNEL BEHAVIOR

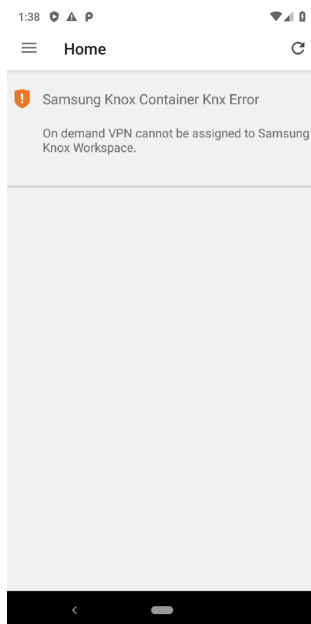
	On-demand	Always-on
Per-container	<p>Not a valid configuration for Samsung Knox.</p> <p>This combination is configurable in MobileIron Core, however MobileIron Tunnel will not work. See Error messages for a per-container and on-demand Tunnel VPN setup.</p>	<p>Tunnel starts when the Samsung Knox Workspace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

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 MobileIron Core Admin Portal. After the device syncs with MobileIron Core, error messages are seen in Mobile@Work on the device and in the device profile in the Admin Portal.

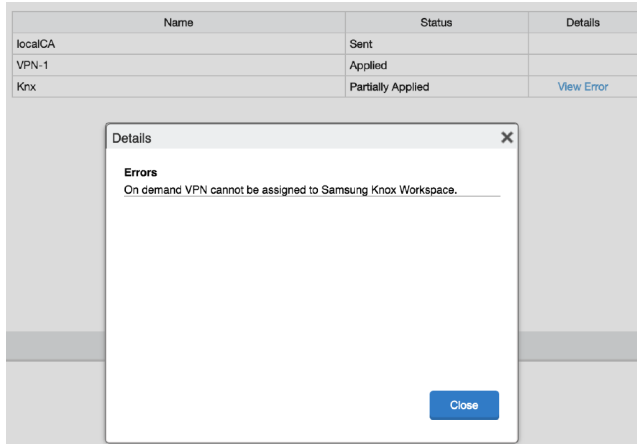
Mobile@Work displays the error as seen in the following figure.

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



Configuring an IP_ANY AppTunnel service on a Standalone Sentry

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

NOTE: 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 MobileIron Core 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 a MobileIron Tunnel VPN configuration for Samsung Knox Workspace \(Core\)](#).

Related topics

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



Creating a MobileIron Tunnel VPN configuration for Samsung Knox Workspace (Core)

The MobileIron 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 MobileIron Core. 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 MobileIron Core 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 MobileIron 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 MobileIron app storefront, go to [Distributing Tunnel through Apps@Work](#).

Related topics

- See [Custom data key-value pairs for 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 MobileIron Tunnel for the Samsung Knox container \(Core\)](#).

Distributing Tunnel through Apps@Work

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

Procedure

1. In the MobileIron Core Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+**.
3. Click **Google Play**.
The app wizard appears.
4. Type "MobileIron Tunnel" for **Application Name**, and click **Search**.
5. Select the listing for MobileIron 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 MobileIron 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*.



Configuring app VPN in the Samsung Knox container

Update the Samsung Knox container configuration:

- Add MobileIron 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 MobileIron Tunnel VPN.

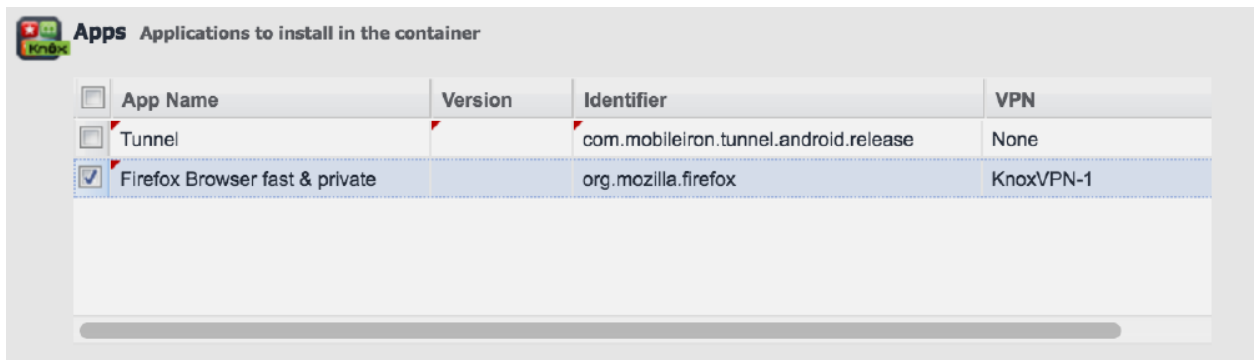
The Samsung Knox container configuration determines which VPN configuration is used and whether the VPN configuration is applied per app or per container.

IMPORTANT: 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 MobileIron Core Admin Portal, go to **Policies & Configs > Configurations**.
2. Select the Samsung Knox container configuration and click **Edit**.

FIGURE 3. APPS CONFIGURATION



<input type="checkbox"/>	App Name	Version	Identifier	VPN
<input type="checkbox"/>	Tunnel		com.mobileiron.tunnel.android.release	None
<input checked="" type="checkbox"/>	Firefox Browser fast & private		org.mozilla.firefox	KnoxVPN-1

3. In the **Apps** section do the following:
 - a. Click **+** to add MobileIron Tunnel.
 - b. For **App Name**, select the MobileIron 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.
 - c. Similarly, if needed, add other apps to make the apps available in the Samsung Knox container.
4. Configure the apps to use Tunnel VPN. Do one of the following:
 - [Configuring per-app VPN](#).
 - [Configuring per-container VPN](#).
5. Click **Save**.

Related topics

See also, [Configuration overview for MobileIron Tunnel for the Samsung Knox container \(Core\)](#).



Configuring per-app VPN

If you configure per-app VPN, only apps to which the 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 Tunnel VPN, in the **VPN** field, select the MobileIron Tunnel (Samsung Knox Workspace) VPN configuration from the drop down list.
Only the specified apps can use MobileIron Tunnel VPN.

NOTE: 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](#).

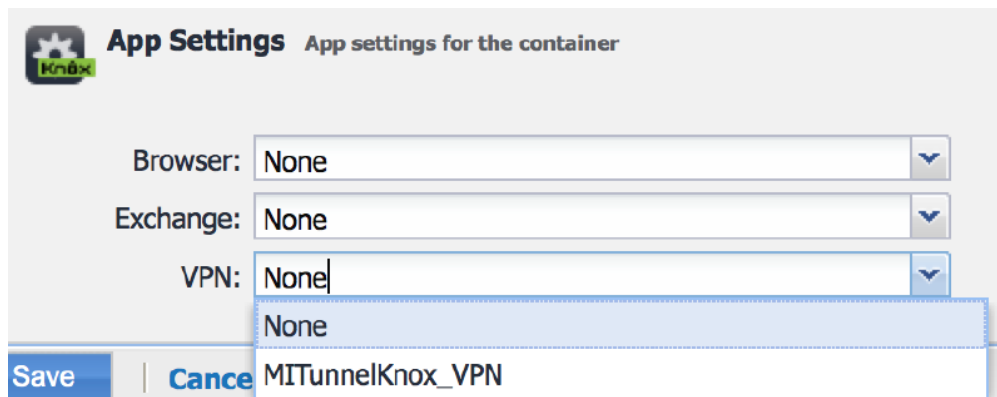
Configuring per-container VPN

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

Procedure

1. In the Samsung Knox container configuration, scroll down to the **App Settings** section.

FIGURE 4. APPS SETTINGS CONFIGURATION



2. For **VPN** in **App Settings**, select the MobileIron 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.

NOTE: 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**.

3. Click **Save**



Related topics

See also, [Configuring app VPN in the Samsung Knox container](#).

Configuring VPN chaining

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

Before you begin

- Configure MobileIron Tunnel for Samsung Knox Workspace as described in [Configuration overview for MobileIron Tunnel for the Samsung Knox container \(Core\)](#).
- Configure an OpenVPN VPN setting in the MobileIron Core Admin Portal. For more information, see the “Configuring new VPN settings” and the “OpenVPN” sections in the *MobileIron Core Device Management Guide* for Android.

NOTE: Use the OpenVPN setting on MobileIron Core 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 MobileIron Core Admin Portal, go to **Policies & Configs > Configurations**.
2. Select and **Edit** the Tunnel VPN configuration for Samsung Knox Workspace.
 - a. In the Tunnel VPN configuration for Samsung Knox Workspace, for **VPN Chaining**, select **Inner**.
 - b. Click **Save**.
3. Select and **Edit** the OpenVPN configuration.
 - a. In the OpenVPN configuration, for **VPN Chaining**, select **Outer**.
 - b. Click **Save**.
4. Select and **Edit** the Samsung Knox container configuration.

FIGURE 5. APPS CONFIGURATION



App Name	Version	Identifier	VPN
Firefox Browser fast & private		org.mozilla.firefox	None
Google Chrome: Fast & Secure		com.android.chrome	Android Knox Config
Tunnel		com.mobileiron.tunnel.andro...	Samsung Open VPN
Facebook		com.facebook.katana	None

5. In the **Apps** section of the Samsung Knox container configuration, do the following:
 - a. For VPN for Tunnel, select the OpenVPN configuration with outer VPN chaining (Configured in [Configuring VPN chaining](#)).
 - b. For apps that will use VPN chaining, select the Tunnel VPN configuration with inner VPN chaining (Configured in step 2).
6. Ensure that the configurations are applied to a label that contains the devices for which you want to allow VPN chaining with MobileIron Tunnel.



Setting up MobileIron Tunnel for Android enterprise

The following address the setup required for app VPN using MobileIron Tunnel for Android enterprise in MobileIron Core and MobileIron Cloud:

- [Before you configure MobileIron Tunnel for Android enterprise \(Core and Cloud\)](#)
- [Configuration tasks overview for Android enterprise \(Core\)](#)
- [Configuration tasks overview for Android enterprise \(Cloud\)](#)

Before you configure MobileIron Tunnel for Android enterprise (Core and Cloud)

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

- [Before you configure MobileIron Tunnel for Android enterprise \(Core and Cloud\)](#)
- [Requirements for deploying Tunnel for Android enterprise](#)
- [Recommendations for deploying Tunnel for Android enterprise](#)
- [Limitations for Tunnel for Android enterprise](#)

Required components for deploying Tunnel for Android enterprise

The following components are required for a MobileIron Tunnel deployment on Android enterprise devices:

- Standalone Sentry with AppTunnel enabled or MobileIron Access
- MobileIron UEM with the following:
 - MobileIron UEM enabled for Android enterprise
 - Users have Android enterprise-capable device.
MobileIron UEM is MobileIron Core or MobileIron Cloud.
- MobileIron client for Android enterprise:
 - MobileIron Core: Mobile@Work
 - MobileIron Cloud: MobileIron Go

NOTE: MobileIron Tunnel for Android enterprise and Mobile@Work for Android are available from the Google Play store.



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

Requirements for deploying Tunnel for Android enterprise

The following are required for deploying Tunnel for Android enterprise:

- Your MobileIron Cloud must be set up for Android enterprise. For more information, see:
 - MobileIron Core: *MobileIron Core Device Management Guide for Android and Android enterprise*.
 - MobileIron Cloud: *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:
 - MobileIron Cloud: *MobileIron Sentry Guide for Cloud*.
 - MobileIron Core: *MobileIron Sentry Guide for Core*.
- If your deployment uses MobileIron Access, ensure that MobileIron Access is set up. See the *MobileIron Access Guide* for information on how to set up MobileIron Access.
- Ensure that the appropriate ports are open. See the *MobileIron Tunnel for Android Release Notes*.

Recommendations for deploying Tunnel for Android enterprise

The following are recommendations for deploying MobileIron Tunnel for Android enterprise:

- MobileIron strongly recommends that Standalone Sentry use a publicly trusted CA certificate. Android version 7 through the latest versions as supported by MobileIron 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 MobileIron Tunnel* knowledge base article in the MobileIron Support and Knowledge Base portal at <https://community.mobileiron.com/docs/DOC-1713>. The configuration sections describe the use of MobileIron Core UI. However for MobileIron Cloud as well, create a certificate setting and upload the Sentry server certificate to MobileIron Cloud and distribute the certificate setting to devices.
- 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 Tunnel for Android enterprise

The following are limitations of MobileIron 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.
- 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.

Configuration tasks overview for Android enterprise (Core)

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

1. [Adding and configuring the MobileIron Tunnel app \(Cloud\)](#)
2. [Creating an Always-On VPN configuration \(Core, optional\)](#)

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 MobileIron Core.

Adding and configuring the MobileIron Tunnel for Android enterprise app (Core)

Upload the MobileIron Tunnel app to MobileIron Core from Google Play and configure it as follows to make it available to Android enterprise devices.

Procedure

1. In the MobileIron Core Admin Portal, go to **Apps > App Catalog**.
2. Click **Add+**.
3. Click **Google Play**.
4. Enter MobileIron Tunnel for **Application Name**, and click **Search**.
5. Select the line for MobileIron 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 MobileIron Access**, only if you have an Access as a service deployment.
Selecting this option enables authentication traffic through MobileIron Access. The option is available only if Access as a service is set up with MobileIron Core.
12. Configure the restrictions for the app.
13. Click **Finish**.

Next steps

Go to [Creating an Always-On VPN configuration \(Core, optional\)](#).

Related topics

- See [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 MobileIron Core, see the *MobileIron Access Guide*.
- For information about adding and configuring an Android enterprise app, see "App configuration for Android enterprise apps," in the *MobileIron Apps@Work Guide*.

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

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

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. Go to **Policies & Configs > Configurations** and click the **Add New** pull down menu.
2. Select **Android > Android enterprise** to display the **New Android enterprise Setting** screen.
3. Select the **Always-On VPN** check box to display the **App Identifier** pull down menu.
The pulldown 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**.



NOTE: 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.

Configuration tasks overview for Android enterprise (Cloud)

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

1. [Adding and configuring the MobileIron Tunnel app \(Cloud\)](#).
2. [Creating an Always-On VPN configuration \(Cloud, optional\)](#).

Before you begin

- If you are configuring app VPN,
 - You must have created a MobileIron Tunnel service for Android in Standalone Sentry. For information on setting up Standalone Sentry with a MobileIron Tunnel service, see “Working with Standalone Sentry for AppTunnel” in the *Standalone Sentry Guide* for MobileIron Cloud.
 - Standalone Sentry must be set up to use identity certificates for device authentication.
 - Ensure that you have created a Identity Certificate configuration in MobileIron Cloud. 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 MobileIron Tunnel for securing authentication traffic with MobileIron Access, you must have setup MobileIron Access. For information about setting up MobileIron Access see the *MobileIron Access Guide*. As part of the Access setup, you will have created a MobileIron Tunnel service.

Adding and configuring the MobileIron Tunnel app (Cloud)

Upload the MobileIron Tunnel app to MobileIron Cloud from Google Play and configure it to make it available to Android enterprise devices. You can download the app from Google Play.

Procedure

1. In the MobileIron Cloud 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 MobileIron Tunnel app in the Google Play store.
5. Select the MobileIron 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 a distribution option and click **Next**.
The configuration will be distributed to the devices in the group you selected.
8. Click **+** for **Android for Work** to configure settings for the app.



9. Enter a name and description for the configuration.
10. Select **Blocks the user for uninstalling the app** if you do not want device users to uninstall the app.
11. Configure the restrictions for the app and click **Next**.
12. Click **Install Application configuration settings** to configure the install options.
 - a. Edit the **Name** and **Description** of the settings if necessary.
 - b. **Install on Device**: Enable Install on devices, if you want to require that the app is installed on devices.
 - c. **Silently install on Samsung KNOX and Zebra devices**: This option is not applicable to Android enterprise apps.
 - d. **Do not show app in end user App Catalog**: Select if you do not want the app displayed in MobileIron's app catalog on users' devices.
13. Click **Next**.
14. 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.
15. Click **Next** and then click Done.

Next steps

Go to [Creating an Always-On VPN configuration \(Cloud, optional\)](#).

Related topics

See [Tunnel configuration field description for Android enterprise](#) for a description of the restrictions.

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

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

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.

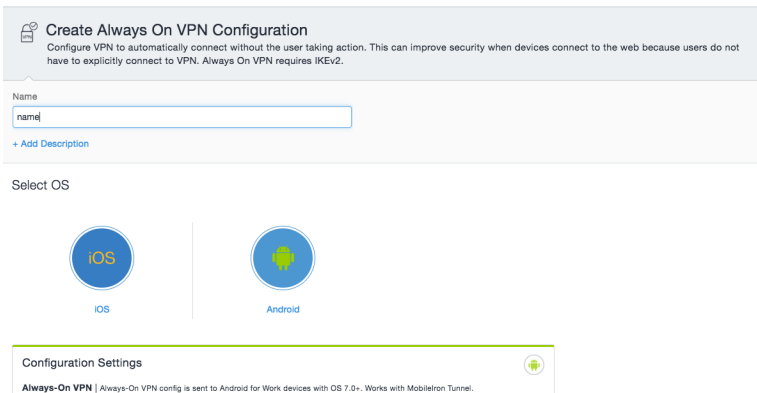
Procedure

1. In MobileIron Cloud, go to **Configuration** and click **+Add**.
2. Click **Always On VPN**.
3. Enter a name for the configuration.



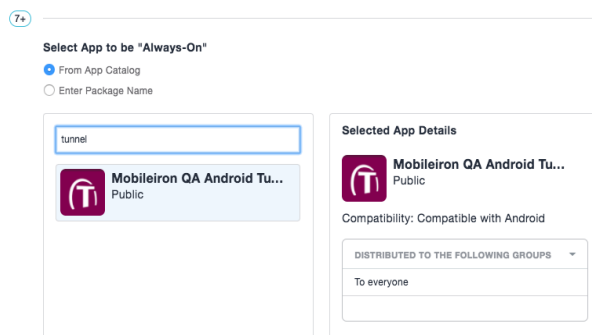
4. Select the **Android** operating system.

FIGURE 6. ALWAYS ON VPN CONFIGURATION



5. In Configuration settings, enter MobileIron Tunnel in the search box.
6. Select the MobileIron Tunnel app. Click **Next**.

FIGURE 7. SELECT APP FOR ALWAYS-ON



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



Tunnel Configuration Fields and Custom Data

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

- [Tunnel for Android native configuration field description](#)
- [Custom data key-value pairs for Tunnel for Android native and Samsung Knox Workspace](#)
- [Tunnel configuration field description for Android enterprise](#)
- [Example showing the Sentry certificate in the certificate chain](#)

Tunnel for Android native configuration field description

The following table provides field descriptions for the Tunnel configuration. There are some variations in field names between MobileIron Core and MobileIron Cloud.

TABLE 3. TUNNEL CONFIGURATION FIELD DESCRIPTION

Item	Description
Name	Enter a name for the MobileIron Tunnel VPN profile.
Description	Enter a description for the profile.
Connection Type (MobileIron Core)	Select MobileIron Tunnel (Android) . Only fields relevant to MobileIron Tunnel for Android are displayed.
Choose OS to create Tunnel Configuration (MobileIron Cloud)	Click Android . Fields relevant to MobileIron Tunnel for Android are displayed.
Enable Access (MobileIron Core)	Select to enable authentication traffic through MobileIron Access. The option is available only if Access as a service is set up with MobileIron. For information about how to set up Access as a service with MobileIron Core, see the <i>MobileIron Access Guide</i> .
Profile selection mode to use for this configuration (MobileIron Cloud)	Select one of the following: <ul style="list-style-type: none"> • Sentry Profile Only: Select if Tunnel traffic goes only through Standalone Sentry. • MobileIron 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 MobileIron Cloud. • MobileIron Sentry + Access Profile: Select if Tunnel VPN supports



TABLE 3. TUNNEL CONFIGURATION FIELD DESCRIPTION (CONT.)

Item	Description
	<p>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 MobileIron Cloud.</p>
Sentry (Profile)	<p>Core: Select the Standalone Sentry on which you created the IP_ANY tunnel service.</p> <p>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 MobileIron Access Profile Only.</p>
Sentry Service (MobileIron Cloud)	<p>Select the MobileIron Tunnel service you created for Android. The option is not available if the profile mode is MobileIron Access Profile Only.</p>
Identity Certificate (MobileIron Core)	<p>Select the Certificate Enrollment setting you created for Sentry setup for AppTunnel.</p>
Client Cert. Alias (MobileIron Cloud)	<p>Select the Identity Certificate configuration you created for Standalone Sentry setup.</p> <p>If the profile mode is Access only or Sentry + Access, select the same certificate you select for SCEP Identity.</p>
SCEP Identity (MobileIron Cloud)	<p>Select the Identity Certificate configuration you created for Tunnel.</p> <p>This field is applicable if the profile mode is Access only or Sentry + Access.</p>
Debug Info Recipient (MobileIron Cloud) For MobileIron Core, the setting is configured using key-value pairs in Custom Data.	<p>Enter a valid email address. The device debug logs are sent to the configured email address.</p> <p>When users tap Email Debug Info, the To field is auto filled with the configured email address.</p>
UI Notification Level (MobileIron Cloud) For MobileIron Core, the setting is configured using key-value pairs in Custom Data.	<p>The user will see error notifications or all Tunnel related notifications, based on the level of notifications you configure.</p> <ul style="list-style-type: none"> • 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. <p>NOTE: There are no notifications to indicate that an app is blocked</p>



TABLE 3. TUNNEL CONFIGURATION FIELD DESCRIPTION (CONT.)

Item	Description
	or allowed.
Debug Log (MobileIron Cloud) For MobileIron Core, 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 (MobileIron Core)	Select one, either Add Allowed Apps or Add Disallowed Apps , to configure the apps that can use MobileIron Tunnel. If you select an app from the MobileIron 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 MobileIron App Catalog can be added to the app list. This setting creates a whitelist. For MobileIron Cloud, <ul style="list-style-type: none"> • 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 Tunnel VPN. Only apps that are not listed will use Tunnel VPN. This setting creates a blacklist. For MobileIron Cloud, <ul style="list-style-type: none"> • 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 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 Tunnel. Example: 10.0.0.0/8;101.210.48.9/32 NOTE: 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.



TABLE 3. TUNNEL CONFIGURATION FIELD DESCRIPTION (CONT.)

Item	Description
	MobileIron 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.
DNS Resolver IP	<p>Configure the list of DNS for Tunnel.</p> <p>Each entry is separated by ‘;’. IPv4 only.</p> <p>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.</p>
Search Domain	Enter a list of search domains for DNS resolver separated by a semicolon (;)
<p>Custom Data</p> <p>Add key-value pairs to configure the app. See Custom data key-value pairs for Tunnel for Android native and Samsung Knox Workspace for a description of the restrictions.</p>	

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

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



TABLE 4. TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION

Key	Value: Enter	Description
Manage Tunnel timeout		
TcplIdleTmoMs	<i>An integer</i>	<p>The Tunnel TCP session idle timeout, on Standalone Sentry, in milliseconds.</p> <p>Tunnel sends this value to Standalone Sentry during the initial handshake in header X-MobileIron-App-TcplIdleTimeoutMs. If this key-value pair is not configured, the default value is 3600000 milliseconds (one hour).</p> <p>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. MobileIron recommends that the value for TcplIdleTmoMs is less than the idle timeout for all the other network elements.</p> <p>As an alternative, consider configuring TCP keep-alive.</p>
VPN connection		
AllowBypass (Android native only)	<ul style="list-style-type: none"> • true • false 	<p>true: Allows all apps to bypass this VPN connection. Apps may use methods such as <code>setProcessDefaultNetwork(Network)</code> to send and receive directly over the underlying network or any other network for which they have permissions.</p> <p>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.</p>
SplitDomainsList	<i>List of domain suffixes separated a semicolon (;)</i>	<p>Example: acme.com; google.com</p> <p>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.</p> <p>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.</p> <p>The DNS handler for the Tunnel plugin decides which DNS request will be sent to which DNS server, based on the configured domains:</p> <ul style="list-style-type: none"> • 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



TABLE 4. TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
		<p>omitted. Example: example.com does not match example.com.akamai.com</p> <ul style="list-style-type: none"> Only complete domains are matched. Example: example.com does not match myexample.com '*' and '?' are not valid characters for the configuration. <p>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.</p>
SplitUDPPortList	List of UDP ports separated by a semicolon (;)	<p>List of UDP ports to send through Tunnel VPN. All other UDP packets are sent directly to destination.</p> <p>If the key-value pair is not configured, all UDP packets are sent through Tunnel VPN.</p> <p>Example 53;161-162;200-1024</p>
MTU	<i>An integer</i>	<p>Tunnel MTU.</p> <p>The default value if the key-value is not configured is 1400</p>
quickRetryMaxAttempts	<i>An integer</i>	<p>Number of attempts to reconnect to VPN.</p> <p>The default if the key-value pair is not configured is 3.</p>
quickRetryIntervalSec	<i>An integer</i>	<p>Time between attempts to reconnect to VPN in seconds.</p> <p>The default if the key-value pair is not configured is 1.</p>
slowRetryIntervalSec	<i>An integer</i>	<p>Time between attempts to reconnect to VPN in seconds.</p> <p>The default if the key-value pair is not configured is 60.</p>
TcpKeepCount	<i>An integer</i>	<p>The value configured specifies the number of unacknowledged probes for TCP keep-alive to send before the connection is considered as dead.</p> <p>The default value, if the key-value pair is not configured, is 20.</p> <p>The key is part of the Android operating system specifications.</p>
TcpKeepIntervalSec	<i>An integer</i>	<p>The value configured specifies the TCP keep-alive interval between subsequent failed keep-alive probes in seconds.</p> <p>The default value, if the key-value pair is not configured, is 2 seconds.</p> <p>The key is part of the Android operating system specifications.</p>

TABLE 4. TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
AtpProbeldleSec	<i>An integer</i>	<p>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.</p> <p>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.</p> <p>Default value if the key-value pair is not configured: 60 seconds</p>
AtpProbeIntervalSec	<i>An integer</i>	<p>Sets the interval, in seconds, between probe packets sent after the minimum idle time specified in AtpProbeldleSec.</p> <p>Default value if the key-value pair is not configured: 1 second</p>
AtpProbeCount	<i>An integer</i>	<p>Sets the total count of the probe packets sent after the minimum idle time specified in AtpProbeldleSec.</p> <p>Default value if the key-value pair is not configured: 5</p>
Certificates		
DisablePinning	<ul style="list-style-type: none"> • true • false 	<p>false: Default, if the key-value pair is not configured. Certificate pinning is enabled.</p> <p>true: Certificate pinning is disabled. Disabling certificate pinning is not recommended for security reasons.</p> <p>NOTE: The Standalone Sentry server certificate is automatically pushed to the device.</p>
Troubleshooting		
UINotificationLevel	<ul style="list-style-type: none"> • 0 • 1 • 2 	<p>The user will see error notifications or all Tunnel related notifications, based on the level of notifications you configure.</p> <p>Configure one of the following levels of user notifications that the Tunnel app will provide:</p> <ul style="list-style-type: none"> • 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. <p>NOTE: There are no notifications to indicate that an app is</p>



TABLE 4. TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
		blocked or allowed.
DebugLog	<ul style="list-style-type: none"> • 0 • 6 • 4 • 3 • 2 	<p>Controls the amount of logging. The client app can override the VPN profile.</p> <ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • false • true 	<p>Allows users to capture traffic in a PCAP file.</p> <p>false: Device users are not allowed to trigger inner traffic capture.</p> <p>true: Device users are allowed to trigger inner traffic capture and email the PCAP file.</p> <p>The default, if the key-value pair is not configured, is false.</p> <p>NOTE: The PCAP file may contain sensitive information.</p>
debugInfoRecipient	<i>Email address</i>	<p>The device debug logs are sent to the configured email address.</p> <p>When users tap Email Debug Info, the To field is auto filled with the value configured for debugInfoRecipient.</p>
EnableUserControl	<ul style="list-style-type: none"> • true • false 	<p>true: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is available to the device user.</p> <p>false: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is not available to the device user.</p> <p>Default value if the key-value pair is not configured: true</p> <p>The key-value pair is not applicable to MobileIron 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.</p>
DefaultMaxNumLogs	<i>An integer</i>	<p>Sets the maximum number of log files.</p> <p>The default if the key-value pair is not configured is 8.</p>
DefaultMaxPcapSize	<i>An integer</i>	<p>Sets the maximum pcap file size in bytes.</p> <p>The default if the key-value pair is not configured is 2097152.</p>

TABLE 4. TUNNEL CONFIGURATION KEY-VALUE PAIRS DESCRIPTION (CONT.)

Key	Value: Enter	Description
DefaultMaxNumPcaps	<i>An integer</i>	Sets the maximum number of pcap files. The default if the key-value pair is not configured is 10.
AnalyticsEnabled	<ul style="list-style-type: none"> • true • false 	<p>true: Enables collection of analytics data for Mixpanel.</p> <p>false: Collection of analytics data is disabled.</p> <p>Default value if the key-value pair is not configured: true.</p>
SendDeviceID	<ul style="list-style-type: none"> • true • false 	<p>true: Tunnel provides the device ID to MobileIron Access.</p> <p>The device ID is reported on MobileIron Access in Reports > Errors.</p> <p>false: Tunnel does not provide the device ID to MobileIron Access.</p> <p>The key-value pair is useful in identifying devices that encounter connection errors when authenticating through Access.</p> <p>Default value if the key-value pair is not configured: false</p>
Tethering		
ExcludeTethering	<ul style="list-style-type: none"> • true • false 	<p>true: Tunnel VPN continues to work on the tethered host device without impacting the tethering client connection.</p> <p>false: Tunnel VPN may impact the tethering client connection.</p> <p>Default value if the key-value pair is not configured: false</p> <p>This key-value pair may be required for Tunnel for Android native only.</p> <p>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)</p> <p>NOTE: Tethering traffic from client devices does not go through the VPN of the host device.</p>

Tunnel configuration field description for Android enterprise

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



TABLE 5. CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE

Restriction	Description
Tunnel profile mode (MobileIron Cloud)	<p>Select one of the following:</p> <ul style="list-style-type: none"> • Sentry Profile Only: Select if Tunnel traffic goes only through Standalone Sentry. • MobileIron 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 MobileIron Cloud. • MobileIron Sentry + Access Profile: Select if 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 MobileIron Cloud.
Sentry Server	<p>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 Tunnel profile modes:</p> <ul style="list-style-type: none"> • Sentry Profile Only • MobileIron Sentry + Access Profile
AllowedAppList	<p>Optional. Use only if DisallowedAppList is empty. Applies only to apps in the Android enterprise work profile.</p> <p>Provide a list of apps in the Android enterprise profile that are allowed to use the Tunnel VPN connection by supplying the app package names, separated by ‘;’.</p> <p>Example</p> <p>Example: com.salesforce.chatter;com.appexample.two</p> <p>If AllowedAppsList has one or more entries, only the apps in the list are allowed to use VPN.</p> <p>This is a whitelist.</p>
DisallowedAppList	<p>Optional. Use only if AllowedAppList is empty. Applies only to apps in the enterprise work profile.</p> <p>Provide a list of applications in the Android enterprise profile to be prevented from using Tunnel by supplying the app package names separated by ‘;’.</p> <p>Example: com.salesforce.chatter;com.appexample.two</p> <p>If AllowedAppList is empty, then all apps can use VPN except the apps in the DisallowedAppList.</p> <p>This is a blacklist.</p>
AllowBypass	Select to allow all apps to bypass this Tunnel VPN.



TABLE 5. CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
AddedRoutes	<p>Enter the network routes that are allowed through Tunnel.</p> <p>Use CIDR format. Each entry in the list is separated by a semicolon (;). IPv4 only.</p> <p>This enables split tunneling where only specific traffic can be taken through Tunnel. The routes configured only impact apps that use Tunnel.</p> <p>Example: 10.0.0.0/8;101.210.48.9/32</p> <p>NOTE: 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. MobileIron 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.</p>
DNSResolverIP	<p>Enter the list of DNS for Tunnel. Each entry is separated by a semicolon (;). IPv4 only.</p> <p>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.</p>
SplitDomainsList	<p>Enter a list of domains suffixes separated by a semicolon (;).</p> <p>Example: mobileiron.com; google.com</p> <p>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.</p> <p>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.</p> <p>The Tunnel plugin's DNS handler decides which DNS request will be sent to which DNS server, based on the configured domains:</p> <ul style="list-style-type: none"> • 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: mobilerion.com does not match mobilerion.com.akamai.com • Only complete domains are matched. Example: mobileiron.com does not match mymobiliron.com • '*' and '?' are not valid characters for the configuration.



TABLE 5. CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
	<p>The filtering is done on an IP packet level, therefore DNS resolver functionality is not provided.</p> <p>The default behavior sends all DNS requests to the VPN's DNS Server.</p>
SearchDomain	Enter a list of search domains for DNS resolver separated by a semicolon (;).
SentryService (Cloud only)	Name of the IP Tunnel service defined on Sentry.
SentryPort (Core only)	Sentry Tunnel port. Use port 443, typically.
ClientCertAlias	<p>Core</p> <p>This is the certificate alias set up in Core. The value is <code>\$CERT_ALIAS:<name-of-SCEP>\$</code> where <code><name-of-SCEP></code> is the Certificate Enrollment setting configured in Core UI.</p> <p>Example: <code>\$CERT_ALIAS:scepIdentityCert\$</code> where <code>scepIdentityCert</code> is the name of the SCEP configured in Core.</p> <p>Cloud</p> <p>Select the Identity certificate setting you created.</p>
SentryCertificate (Core only)	<p>Copy and paste the Sentry certificate from the sentry-server-cert-chain.pem file.</p> <p>This is required if DisablePinning is not selected.</p> <p>For information on how to retrieve the sentry-server-cert-chain.pem file see https://community.mobileiron.com/docs/DOC-1713.</p> <p>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.</p>
DisablePinning	Disabling certificate pinning is not recommended for security reasons. If selected, the SentryCertificate is not required.
EnableUserControl	<p>Select the check box to enable.</p> <p>Enabled: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is available to the device user.</p> <p>Disabled: Tunnel VPN is enabled. The option to enable or disable Tunnel VPN is not available to the device user.</p>
UINotificationLevel	<p>Choose one of the following levels of user notifications that the Tunnel app will provide:</p> <ul style="list-style-type: none"> • 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.



TABLE 5. CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
	<ul style="list-style-type: none"> • All notifications: Error notifications and connect/disconnect confirmations are displayed. <p>The user will see error notifications or all Tunnel related notifications, based on the level of notifications you choose.</p> <p>NOTE: There are no notifications to indicate that an app is blocked or allowed.</p>
DebugLog	<p>Controls the amount of logging. The client app can override the VPN profile.</p> <ul style="list-style-type: none"> • 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	<p>Captures traffic logs.</p> <ul style="list-style-type: none"> • Off: Default setting. No logs are collected. • Minimal: Minimal logs are collected. • All: Detailed logs are collected.
Allow traffic capture	<p>Allows users to capture traffic in a PCAP file.</p> <p>NOTE: The PCAP file may contain sensitive information.</p>
TcpIdleTmoMs	<p>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-MobileIron-App-TcpIdleTimeoutMs. If this key-value pair is not configured, the default value is 3600000 milliseconds (one hour).</p> <p>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. MobileIron recommends that the value for TcpIdleTmoMs is less than the idle timeout for all the other network elements.</p> <p>As an alternative, consider configuring TCP keep-alive.</p>
MTU	<p>Enter an integer for Tunnel MTU.</p> <p>The default value is 1400.</p>
DebugInfoRecipient	<p>Provide an email address.</p>



TABLE 5. CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
	<p>The device debug logs are sent to the configured email address.</p> <p>When users tap Email Debug Info, the To field is autofilled with the value configured for debugInfoRecipient.</p>
quickRetryMaxAttempts	<p>Number of attempts to reconnect to VPN.</p> <p>The default is 3.</p>
quickRetryIntervalSec	<p>Time between attempts to reconnect to VPN in seconds.</p> <p>The default is 1.</p>
slowRetryIntervalSec	<p>Time between attempts to reconnect to VPN in seconds.</p> <p>The default is 60.</p>
appRunningCheckIntervalSec	<p>Time between app status checks in seconds.</p> <p>By default this key is enabled with an interval of 60 seconds.</p> <p>To disable this key, enter 0.</p>
TcpKeepIdleSec	<p>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.</p> <p>A value of 0 means TCP keep-alive is disabled.</p> <p>The default value, if the key-value pair is not configured, is 0.</p> <p>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.</p> <p>The key is part of the Android operating system specifications.</p>
TcpKeepCount	<p>The value configured specifies the number of unacknowledged probes for TCP keep-alive to send before the connection is considered as dead.</p> <p>The default value, if the key-value pair is not configured, is 20.</p> <p>The key is part of the Android operating system specifications.</p>
TcpKeepIntervalSec	<p>The value configured specifies the TCP keep-alive interval between subsequent failed keep-alive probes in seconds.</p> <p>The default value, if the key-value pair is not configured, is 2 seconds.</p> <p>The key is part of the Android operating system specifications.</p>
AtpProbIdleSec	<p>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.</p>

TABLE 5. CONFIGURATION FIELD DESCRIPTION FOR TUNNEL ENTERPRISE (CONT.)

Restriction	Description
	<p>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.</p> <p>Default value if the key-value pair is not configured: 60 seconds</p>
AtpProbeIntervalSec	<p>Sets the interval, in seconds, between probe packets sent after the minimum idle time specified in AtpProbeldleSec.</p> <p>Default value if the key-value pair is not configured: 1 second</p>
AtpProbeCount	<p>Sets the total count of the probe packets sent after the minimum idle time specified in AtpProbeldleSec.</p> <p>Default value if the key-value pair is not configured: 5</p>
InternalDebugOption1	Use only if instructed by MobileIron Support for troubleshooting purposes.
TunIP	Use only if instructed by MobileIron Support for troubleshooting purposes.
MaxNumLogs	<p>Specify the maximum number of log files.</p> <p>The default is 8.</p>
MaxNumPcaps	<p>Specify the maximum number for pcap files.</p> <p>The default is 10.</p>
AnalyticsEnabled	Check to enable collection of analytics data for Mixpanel. The box is checked by default.
SaveAfwConfiguration	Enable this configuration only if requested by MobileIron Support.
AutoBackgroundLaunch	<p>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.</p> <p>For the feature to work, ensure that always-on is also enabled.</p> <p>The feature is available on Android N, O, and P.</p>
AllowPerAppTunnel	For internal MobileIron use only. Do not use this setting.
ClientCertsNumInChain	<p>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. MobileIron recommends not changing the default setting unless additional certificates need to be passed to Sentry or Access.</p>

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
UHJveH1DQTEQMA4GA1UECwwHU3VwcG9ydDETMBEQA1UECgwKTW9iaWx1SXJvbjEw
MBQGA1UEBwwNTW91bnRhaW4gVm1ldzETMBEGA1UECAwKQ2FsaWZvcms5pYTELMAkG
A1UEBHMCMVVMwHhcNMTYwODA4MjYyMjQ3WhcNMjYyMjQ3WjCBhZEkMCIQ
A1UEAwWbYXBMWQxNi5hdXRvLm1vYm1sZW1yY24uY29tMRADgYDVQQLDAdTDXBw
b3J0MRMwEQYDVQKDApNb2JpbGVJcm9uMRYwFAVDVQQA1Nb3VudGFpbjBwWV3
MRMwEQYDVQIDApDYWxpZm9ybm1hMQswCQYDVQGEwJVUzCCASiWdQYJKoZIhvcN
AQEBBQADggEPADCCAQoCggEBAIYdxpmUGJy6Z3BJ21MxBS5w3kKVdANQmo1cVJC
InhJLrm41K3Mazs09/2bf3t+ND8xTkI2YjRiZaz94B2dkrI7fcX0r7tjbStcXUP
yM6+49ipuBxjUKJNs20ZFJdRC0VK8ecbBS1DFOnIIW+fgUEqtWVA/k3nrwoemfep
zKg4hHBzB4+B5369nzyIxxXy9gUKFRLes/kxWAexJB8eopxf6Zdf9W8tUp15h1C
ar6m3TY07pL3KU03U0K7mJXx0lsYqES8DHkTHfj2jYqnEqhxNurwTARYYmuV4iFU
BuztHaKzE00Sco4SmtBqa3FdU0J9EH11tiuzPqPfa4HTxA0CAwEAaA0BrzCbrDAJ
BgNVHRMEAIAAMBGA1UddgQMBAPtZw50cnktS2V5MIGjBgNVHSMegYEw6F3pHUw
czEQMA4GA1UEAwwHUHJveH1DQTEQMA4GA1UECwwHU3VwcG9ydDETMBEQA1UECgwK
TW9iaWx1SXJvbjEwMBQGA1UEBwwNTW91bnRhaW4gVm1ldzETMBEGA1UECAwKQ2Fsa
WZvcms5pYTELMAkGA1UEBHMCMVVMCMV3+swDQYJKoZIhvcNAQELBQADggIBAEBV/
sdXPHxUHZSBuKBbPj2h8oXQa4N1z1FawqNzbdBTktmUgqkyu2hxi1uD6Mg1iqKc
Uz6IFLI3zMw4QULhwlq1eAlqkPQ9x45wySx+BufzpiC00qkC28wJdKnB0dM3Jig
CpvJcelvS3jTYuyjgJRNbaM0HGGrHu4NBGrH1jevHawOHTkvr9QmYhHhT2XYnug
FFXM5giC1ot9VGLA+UrZpVGrDg2Kcql5Eb+K99kjekTQ+0x7oFNj1wb8v1ZMpm/b
zzssOIiltccZPVodJ0ksrmbFH1m1L8VwcE5nqAwMrJ2vump10IUXLxZHCwYpYX
nZ1DcvxZqz7M8AaULQV7UnUCr4Idbu16X3/06LrCVBYq9zTiQwo/ZgWx5NFsXJVH
DpLhR30sBQ4kiorsXULHxmnqA31snp3KDpt2WnJvFw9uCYNd0fg65zuSP/5Dw9Th
8zFv+ksSVVOXFEMJ7jL6j4LFCsB2weE1wdkqG2ze+fc259Gbgq9pP4PAHgp9UtB2D
0d15saQkHBAwUhpNM1CHVxDi/1zpnbvtRNW3C5G/1uKpZ1V6rDeoasV/I+S9SZx
LvDsyje6vP00VM0cv9w1Y/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-----
MIIFxDCCA6ygAwIBAgIEAKzv6zANBgkqhkiG9w0BAQsFAADBzMRADgYDVQDDAdQ
cm94eUNBMRADgYDVQQLDAdTDXBw3J0MRMwEQYDVQKDApNb2JpbGVJcm9uMRYw
FAVDVQQA1Nb3VudGFpbjBwWV3MRMwEQYDVQIDApDYWxpZm9ybm1hMQswCQYD
VQQGEwJVUzAeFw0xNjA4MDgxNjIyMzdaFw00NjA4MDExNjIyMzdaMHMxEDA0BgNV
BAMMB1Byb3h5Q0ExEDA0BgNVBAsMB1N1cHBvcnQxEzARBGNVBAoMCK1vYm1sZUly
b24xYjEjAUBGNVBAcMDU1vdW50YWluIFZpZXcxZzARBGNVBAgMCKNhbG1mb3JuaWEx
CzAJBgNVBAYTA1VTMIIICjANBgkqhkiG9w0BAQEFAAOCAg8AMIICGKCAgEAgkD6
rKQ5ASn1lV3zvhnTtGjDhQ3dGvtUAlI6S8jwCXbb7Ed4dK2zZ50If16z6LDNGPsE
0q3EdLVXuZeUA0nVkiThkd9hmXLRp91lj1TmuibMh6of34YJAuMzpIMLVuRcQjHk
9k1b3mZsjSgeFGKwXgKJ+iAwz+Hp30PptUreQYgCF1cPQQMbh2U5gGJm6zgn2sBt
2trfLtb6cDnurkriLfeqm+o7ppvmULMof90JF7Lr8v18ozVArF60kaw0ywKex/1ZP
extuAEy/6QPW6LWYuakmEGiQ10R5WCKJ9p5Wv2+Y+pFTgcWmr0GmeHTFrfljsm
FJzLzMSbq2gAJBU1tEGzaY449y+yuSzGUqxTitoewDQY/KDcvM9gny90Seo8h+8R
fn9/B0a//wo4H/fnJX70ZMXI1kPtEg/6roVvsexzHkI1H9FPrDw0g45Aps1/6g5n
QxE3wCCipHN2hmXgI33kREKXSL4pRT1bbdUW1+fkX2BYpxzY1LS6ZCXf+RvNFenH
iWoogp6gUc1RzDKmwYMSpduWjireOP0MRJ8cckys8Bon+3/i7SPpAj7gfEYRb0BD
1yx/T0EiUfP/wuLQSwdk2sqrwYjddrfqFSIAGyI8dkzTzue/tqzPTYMk0xyKCVX
Wly4v3PDRnm2G83qsC3r6ndK2ct52aIny0imBzcCAwEAaAaNgMF4wDwYDVR0TAQH/
BAUwAwEB/zALBgNVHQ8EBAMCAaYwHwYDVR0jBBgwFoAUCqgSnfU4jXXhXPrdVEr
UIzpcKowHQYDVR00BBYEFaQoEp31OI114Vz66w1Rk1CM6XJKMA0GCSqGSIb3DQEB
CwUAA4ICAQBBfw9G+5U5WHDpX8ZCN8Jy5fqPHXjtdJcXJi4wgq75om2EAp4nWRb8



```
WmASZ8pz3ZA6JVM5QG1wS616bU5dxNMnu+snzueDdTDUZ XV7aPtKn6QkNJeZqvMG
UWLr3TYHXQIEvi0rsxkh2ow+9j4XB1N9Kb6R8H02S2JDu8tdX4GQUI6xt9gA9IfM
QakHGPDh0PslHauT2Gz7KUYzRqscqF2N1KQS6Z/VhTm3CEex0ZNHrBIZM/INMs2i
VMjwUI+d0ouUiagcenPtM26hR9uuCkvwzNSVrhPljN1V1c1juKGo3K9VTbISXkmG
hnW1B1QdPhpat1uDB49Lb7gZnIMDCcfzR1ZhwgqJgFyT0eekJpMqtFJpU8s6Rbc
B7EY6i3AGFc8dbtEZbZEL8HHKB0bL0EUjHeWJtadGKaakT0Rh6Qgc0boDx3mwwBG
lSI1J8/OqkQz4LYJuBWywYJu+BHcUfuKdduqDEfzgU83wwv1izRB5kZeWvSuu/+1
dz4p7yAB7mWC/IlafqR7WRmURWWhVhzMQFx3mr8wJZpL2MgbnF/z12cGgytNgw6L
6/zj4l2DbSxouc6TrPtUtSK86Z+v4Ryi6waJGh/FglQQy8Ro+PMxT/gBvT7v3bwe
P3NBgqk3ncF8RMsQhwjlCuPWZX0cgL1lJ/hs2e5+HURxzKInsQj18Q==
-----END CERTIFICATE-----
```



MobileIron Tunnel for Android device user experience

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

- [MobileIron Tunnel installation on devices](#)
- [Controlling VPN traffic](#)
- [Troubleshooting](#)

MobileIron Tunnel installation on devices

The MobileIron 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 MobileIron app catalog on the device.

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

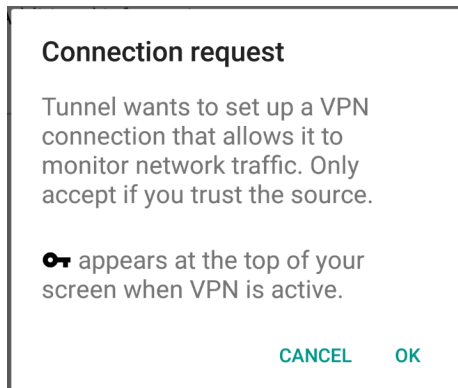
- [Accept Tunnel connection \(Android native and enterprise only\)](#)
- [Allow certificate \(Android native and enterprise only\)](#)
- [Tunnel VPN connection](#)

Accept Tunnel connection (Android native and enterprise only)

The first time that 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 Tunnel.



FIGURE 8. ACCEPT TUNNEL CONNECTION

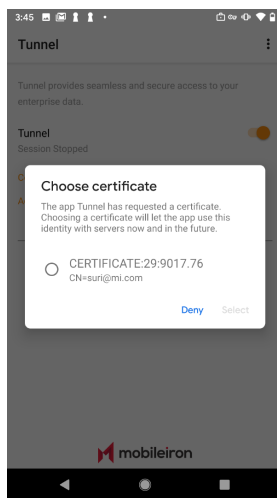


Allow certificate (Android native and enterprise only)

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

IMPORTANT: Do not change the certificate selection.

FIGURE 9. ALLOW CERTIFICATE



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

For Android enterprise devices, if Always On VPN (in Android enterprise configuration on the UEM) and AutoBackgroundLaunch (Tunnel for Android enterprise configuration on the UEM) are enabled, user interaction is not needed to accept the Tunnel certificate.

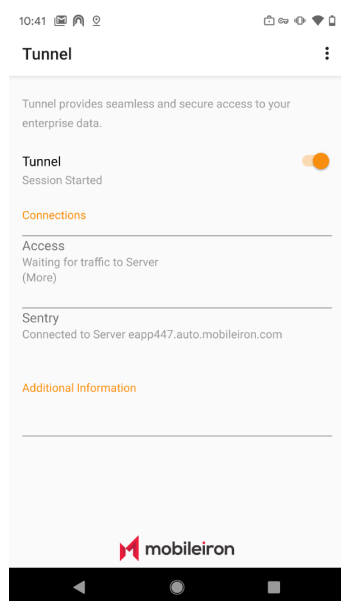
Tunnel VPN connection

The key icon on Android native and Android enterprise devices, or the lock icon on Samsung Knox devices, indicates that the 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 a MobileIron Access deployment, the connection status changes from **Connected** to **Waiting** periodically if there is no Access traffic going through Tunnel.

FIGURE 10. TUNNEL VPN CONNECTION STATE AND ICONS



Tunnel notifications icon

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

FIGURE 11. TUNNEL NOTIFICATIONS ICON



The icon is not visible if there are no notifications.

Controlling VPN traffic

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 Tunnel VPN configuration.

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

TABLE 6. COMPARISON BETWEEN TUNNEL FOR ANDROID AND IOS

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



TABLE 6. COMPARISON BETWEEN TUNNEL FOR ANDROID AND IOS (CONT.)

Function	Behavior on Android	Behavior on iOS
	users for various events (connect/disconnect, allow/block).	per-app 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, MobileIron 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, MobileIron 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.

Troubleshooting

- [Collecting log and PCAP files](#)
- [Viewing Tunnel configuration](#)

Collecting log and PCAP files

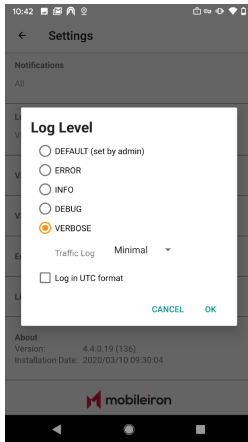
You can collect tunnel log and PCAP files to help with diagnostics and troubleshooting.

Procedure

1. In the Tunnel configuration (VPN configuration for Android native or app configuration for Android enterprise) on your UEM (MobileIron Core or MobileIron Cloud), 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.

FIGURE 12. TUNNEL LOG LEVEL





4. To collect PCAP files, under **Capture Traffic**, check **Enable**.
5. 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 [Tunnel Configuration Fields and Custom Data](#).

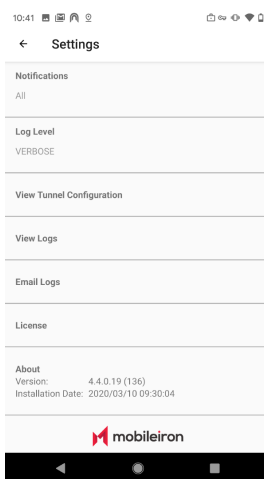
Viewing Tunnel configuration

Tunnel configuration can be viewed in the Tunnel app.

Procedure

1. In the Tunnel app, tap on the three vertical dots to expand the menu.

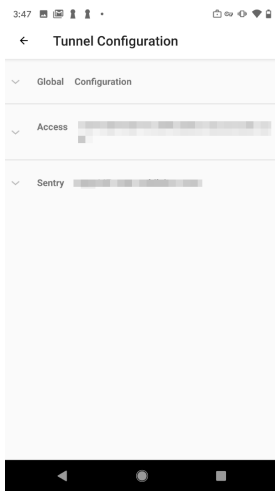
FIGURE 13. TUNNEL SETTINGS



2. Tap **Settings > View Tunnel Configuration**.



FIGURE 14. TUNNEL CONFIGURATION



The Tunnel Configuration information is grouped under Global configuration and under Access specific or Standalone Sentry specific configurations.