ivanti

Guide

ivanti

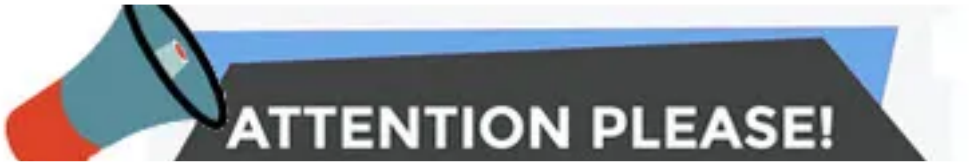
CONTENTS

1. **Ivanti Tunnel 4.14.0 - 4.15.0 for iOS Guide** **3**
2. **New features summary** **4**
3. **About Ivanti Tunnel for iOS** **5**
   - 3.1. Overview of Ivanti Tunnel for iOS 6
   - 3.2. About Ivanti Tunnel configuration 7
   - 3.3. Deployment use cases with Ivanti Tunnel for iOS 8
4. **Setting up Ivanti Tunnel** **9**
   - 4.1. Before you set up Ivanti Tunnel 10
   - 4.2. Main tasks for configuring Ivanti Tunnel for iOS (Ivanti EPMM) 12
   - 4.3. Main tasks for configuring Ivanti Tunnel for iOS (Ivanti Neurons for MDM) 14
   - 4.4. Ivanti Tunnel for iOS configuration field description 17
5. **Setting up single sign-on with Kerberos** **24**
   - 5.1. About the setup for single sign-on with Kerberos 25
   - 5.2. Authentication workflow for single sign-on with Kerberos 26
   - 5.3. Main tasks for configuring single sign-on with Kerberos (Ivanti EPMM or Ivanti Neurons for MDM) 27
   - 5.4. SSO with Kerberos configuration field description 30
6. **Additional configurations using key-value pairs for Ivanti Tunnel** **34**
7. **What users see in Ivanti Tunnel for iOS** **39**
   - 7.1. Ivanti Tunnel installation 40
   - 7.2. Emailing debug log information 42
   - 7.3. Clearning logs 43
   - 7.4. Sharing debug logs 46

1. Ivanti Tunnel 4.14.0 - 4.15.0 for iOS Guide

July 2026

This is the cumulative guide for Tunnel releases.



New features summary

2. New features summary

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

Ivanti Tunnel 4.15.0 - New features summary

- **Support Deprecation for iOS 16.x**: Starting with this release, Ivanti Tunnel for iOS requires iOS 17 or later. Devices running iOS 16.x are no longer supported.
- **Enhanced User Experience**: Ivanti Tunnel now features an updated user interface with a refreshed look and feel.

Ivanti Tunnel 4.14.0 - New features summary

- **Support for Liquid Glass**: Starting with iOS 26, Ivanti Tunnel uses Apple's Liquid Glass look for standard controls.

3. About Ivanti Tunnel for iOS

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

- Overview of Ivanti Tunnel for iOS
- About Ivanti Tunnel configuration
- Deployment use cases with Ivanti Tunnel for iOS

3.1. Overview of Ivanti Tunnel for iOS

Ivanti Tunnel enables VPN capability on iOS devices. Ivanti Tunnel interacts with the Unified Endpoint Management (UEM) platform, Standalone Sentry, and Access to secure access to enterprise resources from outside the enterprise network. The enterprise resource can be on premise or in the cloud. The UEM platforms are: Ivanti EPMM and Ivanti Neurons for MDM.

3.2. About Ivanti Tunnel configuration

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

3.3. Deployment use cases with Ivanti Tunnel for iOS

Ivanti Tunnel enables native per-app and device level VPN on iOS devices. Ivanti Tunnel is part of the following deployments for securing access to enterprise resources:

- UEM and Standalone Sentry.
- UEM and Access.

The following use cases are enabled with these deployments:

- access to internal corporate URLs from the Safari browser.
- per-app VPN for managed apps (managed apps do not need AppConnect wrapping or SDK).
- device-level VPN.
- single sign-on.

App proxy provider and packet tunnel provider

Ivanti Tunnel for iOS supports app proxy provider and packet tunnel provider VPN tunnels.

For apps that use a TCP connection, such as Office or GSuite apps, create an app proxy Ivanti Tunnel VPN configuration. An app proxy Ivanti Tunnel VPN configuration is applicable per app only. Previously, this was the only option available with Ivanti Tunnel.

For apps that use an IP connection (such as Skype for Business an Microsoft Teams), create a packet tunnel provider Ivanti Tunnel VPN configuration. A packet tunnel provider Ivanti Tunnel VPN configuration can be configured to be either per-app or device-level.

For additional information about use cases with specific types of apps, see the knowledge base article in the Support Community: **iOS and macOS - What are VPN Provider Types Packet-Tunnel and App-Proxy?**

- Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic. See **UDP traffic**.
- Split-tunneling for IP routes is supported only for device-level VPN. Configure the routes to through Ivanti Tunnel in the Included Routes (Added Routes) field in the Ivanti Tunnel VPN configuration. See **Ivanti Tunnel for iOS configuration field description**.
- Multiple per-app VPN configurations are supported on a device. However, only one device-level VPN configuration is supported on a device.

4. Setting up Ivanti Tunnel

The following addresses the setup required for Tunnel for iOS and contains the following:

- Before you set up Ivanti Tunnel
- Main tasks for configuring Ivanti Tunnel for iOS (Ivanti EPMM)
- Main tasks for configuring Ivanti Tunnel for iOS (Ivanti Neurons for MDM)
- Ivanti Tunnel for iOS configuration field description

In Ivanti Tunnel for iOS deployment, AppConnect wrapping or SDK is not required and the client certificate is directly authenticated with the backend resource.

4.1. Before you set up Ivanti Tunnel

Before you set up Ivanti Tunnel for iOS devices, see the following:

- **Required components for deploying Ivanti Tunnel for iOS**
- **Requirements for configuring Ivanti Tunnel for iOS**
- **Recommendations for setting up Ivanti Tunnel for iOS**

Required components for deploying Ivanti Tunnel for iOS

The following components are required for an Ivanti Tunnel deployment:

- Standalone Sentry with AppTunnel enabled or Access.
- Unified Endpoint Management (UEM) platform:
  - Ivanti EPMM

OR

  - Ivanti Neurons for MDM
- iOS devices registered with a UEM.
- Client for iOS:
  - Mobile@Work for Ivanti EPMM deployments
    OR
  - Go for Ivanti Neurons for MDM deployments
    OR
  - AppStation for Ivanti Neurons for MDM MAM-only deployments
    For information about deploying AppStation for MAM-only, see *AppStation for iOS Guide*.

For supported versions see the *Ivanti Tunnel for iOS Release Notes*.

Requirements for configuring Ivanti Tunnel for iOS

Ensure the following before configuring Ivanti Tunnel for iOS:

- If your deployment uses Standalone Sentry:
  - You have installed Standalone Sentry. See the *Standalone Sentry Installation Guide*.
  - Standalone Sentry is set up for AppTunnel using identity certificates for device authentication.
    For information about setting up a Standalone Sentry for AppTunnel, see *Tunnel Guide* for your Unified Endpoint Management (UEM) platform.
  - The Standalone Sentry IP address is publicly accessible.
  - The Standalone Sentry name is registered in DNS.
  - To tunnel IP traffic, ensure that you have created an IP_ANY service.
  - For documentation, see **Standalone Sentry product documentation**.
- Standalone Sentry is required for packet tunnel provider with per-app VPN.

- If your deployment uses Access, ensure that Access is set up. See the *Access Guide* for information on how to set up Access. For documentation, see Access product documentation.
- The appropriate ports are open.
  See the *Ivanti Tunnel for iOS Release Notes*.

Recommendations for setting up Ivanti Tunnel for iOS

Review the following recommendations for setting up Ivanti Tunnel for iOS.

- Standalone Sentry
- UDP traffic

Standalone Sentry

Ivanti recommends that Standalone Sentry use a trusted CA certificate. If Standalone Sentry uses a self-signed certificate, you must do the following additional setup in Ivanti EPMM:

- In the Services > Sentry page, for the Standalone Sentry, click the View Certificate link. This makes the Standalone Sentry's certificate known to Ivanti EPMM.
- Follow the instructions in the *Using a Self-signed certificate with Standalone Sentry and Ivanti Tunnel* knowledge base article in the Support and Knowledge Base portal at
  Using a self signed certificate with Standalone Sentry and Ivanti Tunnel
  If the self-signed certificate is changed at any time, you must push the changed certificate to the device, otherwise there may be a disruption in service. Therefore, Ivanti recommends using a certificate from a trusted certificate authority for the Standalone Sentry.

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.

To limit the UDP traffic through Standalone Sentry, gather a list of destination UDP ports that should be tunneled through Ivanti Tunnel VPN. All other UDP traffic is, therefore, not tunneled. Configure the SplitUDPPortList key-value pair to limit the UDP traffic through Ivanti Tunnel.

4.2. Main tasks for configuring Ivanti Tunnel for iOS (Ivanti EPMM)

Following are the main steps for configuring Ivanti Tunnel for iOS. These configuration tasks are performed in the Ivanti EPMM Admin Portal.

1. **Main tasks for configuring Ivanti Tunnel for iOS (Ivanti EPMM)**
2. **Applying the Ivanti Tunnel VPN setting to managed apps in Ivanti EPMM**

Configuring Ivanti Tunnel VPN in Ivanti EPMM

Ivanti Tunnel supports per-app and device-level VPN. Choose the appropriate configuration depending on whether you are creating a per-app VPN or a device-level VPN.

You can create multiple Ivanti Tunnel configurations to push to a device. The VPN profiles pushed to a device are listed in Settings > General > VPN, and in Settings > General > Device Management. Depending on the app in use, iOS automatically switches to use the VPN profile applied to the app.

You can apply both per-app VPN and device-level VPN to a device. However, per-app VPN takes priority over device-level VPN. The device-level VPN is used for apps that are not associated with a per-app VPN.

Before you begin

- If you are configuring app proxy VPN, ensure that you have created a TCP AppTunnel service in Standalone Sentry.
- If you are configuring packet tunnel provider type, ensure that you have created an IP AppTunnel service in Standalone Sentry.
- For information on setting up a TCP or IP AppTunnel service see “Working with Standalone Sentry for AppTunnel” in the *Standalone Sentry Guide* for Ivanti EPMM.
- If you are configuring Ivanti Tunnel for securing authentication traffic with Access see the *Access Guide*.

Ivanti strongly recommends creating separate Ivanti Tunnel VPN configurations for iOS and macOS. Using the same Ivanti Tunnel VPN configuration for iOS and macOS may cause issues with how Ivanti Tunnel operates and how traffic through Ivanti Tunnel is handled.

Procedure

1. In the Admin Portal, go to Policies & Configs > Configurations.
2. Click Add New > VPN.
3. For Connection Type, select Ivanti Tunnel.
4. Add the necessary configurations.
5. Click Save.
6. If you created a device-level VPN configuration, apply the configuration to a label that contains iOS devices.
   The configuration is distributed to the devices in the label.

Next steps

Go to **Applying the Ivanti Tunnel VPN setting to managed apps in Ivanti EPMM**.

Related topics

- For a description of the configuration fields for Ivanti Tunnel (iOS) VPN, see **Ivanti Tunnel for iOS configuration field description**.
- For a description of the key-value pairs, see **Additional configurations using key-value pairs for Ivanti Tunnel**.

Applying the Ivanti Tunnel VPN setting to managed apps in Ivanti EPMM

When you Add or Edit an app in the App Catalog, you have the option to select the per-app VPN setting to apply to the app. Select the per-app Ivanti Tunnel (iOS) VPN setting you created. This procedure is not needed for a device-level VPN configuration.

Procedure

1. In the Admin Portal, go to Apps > App Catalog.
2. Select iOS from the Platform list.
3. Click the Add+, or select an app and click the edit icon next to the app.
4. In the form, for Per App VPN Settings, select the per-app Ivanti Tunnel (iOS) VPN configuration you created.

Related topics

For more information about adding and editing apps for distribution, see the following sections in the *Ivanti EPMM Apps@Work Guide*:

- "Using the wizard to import iOS apps from the Apple App Store."
- "Using the wizard to add an in-house iOS or macOS app to the App Catalog."

Ivanti Tunnel for iOS distribution

Adding Ivanti Tunnel to the App Catalog makes the app available in the app storefront.

Ivanti Tunnel is also available in the Apple App Store. The device user can download the app directly from the Apple App Store. Device users can download the app directly from the Apple AppStore at **itunes**.

If you are a using a self-signed or an untrusted certificate for the Standalone Sentry, the certificate must be pushed to the device. The Standalone Sentry certificate is required on the device for Tunnel to authenticate the Standalone Sentry and establish a per-app VPN session. If the certificate is changed at any time, you must push the changed certificate to the device, otherwise there may be a disruption in service. Therefore, we **recommend using a certificate from a trusted certificate authority for the Standalone Sentry**.

If the certificate is changed at any time, you must push the changed certificate to the device, otherwise there may be a disruption in service. To push the Standalone Sentry certificate to the device, follow the instructions in the ***Using a Self-signed certificate with Standalone Sentry and Tunnel*** knowledge base article.

4.3. Main tasks for configuring Ivanti Tunnel for iOS (Ivanti Neurons for MDM)

You configure Ivanti Tunnel in Ivanti Neurons for MDM.

Before you begin

- If you are configuring app proxy VPN, ensure that you have created an Ivanti Tunnel service for iOS / mac with Service Type TCP_ANY in the Standalone Sentry profile.
- If you are configuring packet tunnel provider type VPN, ensure that you have created a Tunnel service for iOS / mac with Service Type IP_ANY in the Standalone Sentry profile.
- For information on setting up Standalone Sentry with Ivanti Tunnel service, see "Working with Standalone Sentry for AppTunnel" in the *Standalone Sentry Guide* for Ivanti Neurons for MDM.
- If you are configuring Ivanti Tunnel for securing authentication traffic with Access see the *Access Guide*.

Procedure: Overview of steps

1. **Adding Ivanti Tunnel for iOS to the app catalog in Ivanti Neurons for MDM**
2. **Adding an Ivanti Tunnel configuration in Ivanti Neurons for MDM**

Adding Ivanti Tunnel for iOS to the app catalog in Ivanti Neurons for MDM

Ivanti Tunnel for iOS is available in the app catalog in Ivanti Neurons for MDM.

Procedure

1. In Ivanti Neurons for MDM, go to Apps > App Catalog > +Add.
2. In Business Apps, click Ivanti Tunnel (iOS 9+).
3. Make any updates as necessary and click Next.
   You can change the category and add a description.
4. Select an option for app delegation and click Next.
5. Choose a distribution option for the app and click Next.
6. Update the default App Configurations settings as necessary.
7. Click Done.

Next steps

Go to **Adding an Ivanti Tunnel configuration in Ivanti Neurons for MDM**.

Related topics

For more information about topics such as app delegation, see the *Ivanti Neurons for MDM Guide*.

Adding an Ivanti Tunnel configuration in Ivanti Neurons for MDM

You create the configuration for Ivanti Tunnel in Configurations. You can create multiple Ivanti Tunnel configurations to push to a device. The VPN profiles pushed to a device are listed in Settings > General > VPN, and in Settings > General > Device Management. Depending on the app in use, iOS automatically switches to use the VPN profile applied to the app.

Ivanti Tunnel supports per-app as well as device-level VPN. Choose the appropriate Tunnel configuration depending on whether you are creating a per-app VPN or a device-level VPN.

You can apply both per-app VPN and device-level VPN to a device. However, per-app VPN takes priority over device-level VPN. The device-level VPN is used for apps that are not associated with a per-app VPN.

Procedure

1. In Ivanti Neurons for MDM, go to Configurations > +Add.
2. Search for Ivanti Tunnel.
3. Click one of the following:
   - Ivanti Tunnel: Use this configuration to create a per-app VPN configuration for Ivanti Tunnel.
   - Ivanti Tunnel (On Demand): Use this configuration to create a device-level VPN configuration for Ivanti Tunnel.

The Ivanti Tunnel configuration page displays.

4. If you selected theIvanti Tunnel configuration, click iOS/macOS.
   The configuration for Ivanti Tunnel for iOS displays.
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 Ivanti Tunnel for iOS app.
7. Select one of the following distribution options:
   1. **All Devices**: Select one of the following options:
      - **Do not apply to other spaces**.
      - **Apply to devices in other Spaces**.
   2. **No Devices** (default)
   3. **Custom**: Select one of the following options:
      - **User/User Groups**
      - **Device/Device Groups**
        In the Distribution Summary, select one of the following options to enable or disable configurations across spaces:
      - **Do not apply to other spaces.**
      - **Apply to devices in other Spaces.**

      > The checkbox **Allow Space Admin to Edit the Distribution** appears if you select the **Apply to devices in other Spaces** option, and it allows the delegated space administrators to edit the distribution for the specific space.

      > Irrespective of spaces, you can configure the certificate for all spaces, distribute it to all devices, and apply it to all devices in the other device's spaces.
8. Click **Done**.

Next steps

Go to Applying the Ivanti Tunnel VPN setting to managed apps in Ivanti Neurons for MDM.

Related topics

- For a description of the configuration fields for Ivanti Tunnel (iOS) VPN, see **Ivanti Tunnel for iOS configuration field description**.
- For a description of the key-value pairs, see **Additional configurations using key-value pairs for Ivanti Tunnel**.

Applying the Ivanti Tunnel VPN setting to managed apps in Ivanti Neurons for MDM

When you Add or Edit an app in the App Catalog, you have the option to select the per-app VPN setting to apply to the app. For this workflow, select the Tunnel (iOS) VPN setting you created. This procedure is not needed if you configured device-level VPN using Tunnel (On Demand).

Procedure

1. In Apps > App Catalog, add or edit an app.
2. In App Configurations, add the Per App VPN configuration.
3. Enter a name for the configuration.
4. Check Enable Per-App VPN for this app.
5. Select the Tunnel configuration to apply to the app.
6. Select a distribution option and click Next.
7. Click Done.

Related topics

For more information about adding and editing apps for distribution, see the following sections in the *Ivanti Neurons for MDM Guide*:

- “Adding an app from a public store.”
- “Adding an In-house app.”

4.4. Ivanti Tunnel for iOS configuration field description

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

Item	Description
Name	Enter a name for the Ivanti Tunnel VPN profile.
Description	Enter a description for the profile.
Connection Type (Ivanti EPMM)	Select Ivanti Tunnel. Only fields relevant to Tunnel are displayed.
Choose OS to create Ivanti Tunnel Configuration (Ivanti Neurons for MDM. Per-app VPN)	Select iOS/macOS.
Profile selection mode to use for this configuration (Ivanti Neurons for MDM)	Select one of the following: • Sentry Profile Only: Select if Tunnel traffic goes only through Standalone Sentry only. ◦ Access Profile Only: Select if Tunnel traffic goes to Access only. Only authentication traffic is tunneled to Access. This option is available only if a Access deployment is set up. If Access Profile Only is configured with per-app VPN packet tunnel provider type, only authentication traffic is tunneled to Access. All other traffic is dropped. If Access Profile Only is configured with device-level VPN packet tunnel provider type, only authentication traffic is tunneled to Access. All other traffic goes directly to the destination. • 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 a Access as a service deployment is set up.
Legacy App Support (iOS only)	Select one of the following: • Enabled: Select to enable per-app VPN with the Tunnel Legacy app (versions of Tunnel prior to 2.0) on all versions of iOS. • Enabled for iOS 7 and 8: Select to enable per-app VPN using the Tunnel Legacy app for devices running iOS 7 and 8 only. This option enables the per-app VPN feature with Tunnel 2.0 on devices running iOS 9 through the most recently released version as supported by Ivanti. The per-app VPN feature with Tunnel requires a separate license and Sentry 5.0 through the most recently released version as supported by Ivanti. Ensure your organization has purchased the necessary license before enabling this feature. Tunnel 2.0 through the most recently released version as supported by Ivanti is required for devices running iOS 9 through the most recently released version as supported by Ivanti.
VPN Sub Type (Ivanti Neurons for MDM)	(Optional) Overrides the bundle identifier for a customized Tunnel app.

Item	Description
Enable Access (Ivanti EPMM)	Select to enable authentication traffic through Access. The option is available only if Access as a service is set up with Ivanti EPMM. For information about how to set up Access as a service with Ivanti EPMM, see the *Access Guide*.
Provider Type (In Ivanti Neurons for MDM, this field is available only in the Tunnel configuration for per-app VPN.	app-proxy: This is the default setting. Use this setting for TCP tunneling only. packet-tunnel: Select to allow Tunnel to also handle IP traffic. Device-level VPN automatically uses the packet tunnel provider type.
Per-app VPN (Ivanti EPMM)	The options are available if Provider Type is packet-tunnel. Otherwise, the options are grayed out. Device-level VPN is not available for app proxy tunnel. Yes: This is the default setting. Connectivity is established for an app, rather than the device. No: Select to establish connectivity for the device, rather than just an app.
Sentry (Profile)	**Ivanti EPMM:** Select the Standalone Sentry on which you created the tunnel service. **Ivanti Neurons for MDM:** Select the Standalone Sentry profile on which you created the Tunnel for iOS service. The field is not available if the profile mode is Access Profile Only.
Sentry Service	**Ivanti EPMM:** Select the TCP or IP service that the Safari domain or managed app will use. If you are configuring packet tunnel provider type, select the IP service you created for Tunnel. If you are configuring app proxy, select the TCP service you created for Tunnel. **Ivanti Neurons for MDM:** Select the Tunnel for iOS service. The field is not available if the profile mode is Access Profile Only. Only TCP services are available for selection if the provider type is app proxy. Only IP services are available for selection if the provider type is packet tunnel.
SCEP Identity (Ivanti Neurons for MDM)	Select the Identity Certificate configuration you created for Tunnel. The Identity Certificate is automatically selected if Sentry Profile Only or Sentry + Access Profile is enabled.
Debug Info Recipient (Ivanti Neurons for MDM)	Enter an email address to forward the debug information.
Identity Certificate (Ivanti EPMM)	Select the certificate setting you created. If you are using user-provided certificates, select the user provided certificate you created for Tunnel.
On Demand Rules (iOS 9 and later; macOS 10.13 and later) VPN on-demand rules are applied when the device's primary network interface changes, for example, when the device switches to a different Wi-Fi network. Devices	

Item	Description
will drop the Tunnel VPN connection if an enterprise Wi-Fi is detected. If the network is not a Wi-Fi network or if its SSID does not appear in the list, the device will continue to use Ivanti Tunnel VPN. • A matching rule is not required. The Default Rule is applied if a matching rule is not defined. • If you select Evaluate Connection, a matching rule is not required. • You can create up to 10 On Demand matching rules. • For each matching rule you can create up to 50 Type and Value pairs. An Ethernet on-demand rule is only applicable to macOS devices. If the rule is pushed to iOS device, the rule may cause issues with Tunnel behavior and how traffic through Tunnel is handled. Therefore, Ivanti strongly recommends using separate Tunnel VPN configurations for iOS and macOS.	
Add +	Click to add a new On Demand matching rule.
On Demand Action	Select one of the following actions to apply to the matching rule: • Connect • Disconnect
Matching Rules For each On Demand matching rule to which the action is applied enter the type and value pair.	
Add +	Click to add a new On Demand matching rule. A dialog box appears.
Type	Select the following key type: • SSID
Value	Enter a list of SSIDs to match the enterprise Wi-Fi. If the network is not a Wi-Fi network or if its SSID does not appear in the list, the match will fail. To add multiple SSIDs, create a separate SSID Type-Value pair for each SSID.
Description	Enter additional information about this matching rule.
OK	Click to add the On Demand Action and the associated Matching Rules.
Default Rule The default rule (action) is applied to a connection that does not match any of the matching rules.	

Item	Description
On Demand Action	From the drop down list, select Connect.
Safari Domains The device user can access servers ending with these domains in Safari. A Tunnel configuration is only applied to a managed app. Therefore, a managed app with the Tunnel configuration must be installed on the device for the device user to access the domains using per-app VPN. • If the device resolves the destination domain, then Tunnel is not launched. • If the Safari domains use Kerberos authentication, you must also do the setup described in **Setting up single sign-on with Kerberos**.	
Safari Domain	Enter a domain name. Only alphanumeric characters and periods (.) are supported.
Description	Enter a description for the domain.
Add New	Click to add a domain.
Calendar Domains (iOS 13 and later; macOS 10.15 and later) A Tunnel VPN connection is automatically established for these domains. Only available for per-app VPN.	
Calendar Domain	Enter a domain name. Only alphanumeric characters and periods (.) are supported.
Description	Enter a description for the domain.
Add New	Click to add a domain.
Contact Domains (iOS 13 and later; macOS 10.15 and later) A Tunnel VPN connection is automatically established for these domains. Only available for per-app VPN.	
Contact Domain	Enter a domain name. Only alphanumeric characters and periods (.) are supported.
Description	Enter a description for the domain.
Add New	Click to add a domain.
Mail Domains (iOS 13 and later; macOS 10.15 and later) A Tunnel VPN connection is automatically established for these domains. Only available for per-app VPN.	

Item	Description
Mail Domain	Enter a domain name. Only alphanumeric characters and periods (.) are supported.
Description	Enter a description for the domain.
Add New	Click to add a domain.
Included Routes (Added Routes) Only available for device-level VPN. Configured routes are set to the TUN interface. If routes are not configured, Tunnel uses 0.0.0.0/0. Enter list of IPv4 ranges in CIDR format. For multiple values, enter a semicolon separated list.	
DNS Resolver IPs Only for packet tunnel provider type. Enter a domain name server (DNS) to resolve the IP address. IPv4 only. For multiple values, enter a semicolon separated list. Ensure that the DNS is routable if the default route in not used. If DNS is not configured, the Sentry DNS is used.	
DNS Search Domain List Only for packet tunnel provider type. Enter DNS search domains for resolving the domain names. For multiple values, enter a semicolon separated list.	
Match Domain List Only for packet tunnel provider type. Enter domains for the VPN DNS to resolve. For multiple values, enter a semicolon separated list.	
Custom Data Enter Key Value pair to configure the Tunnel VPN disconnect, debug, and timeout behavior. See **Additional configurations using key-value pairs for Ivanti Tunnel**.	
iOS 14.0+ and macOS 11.0+	
Associated Domains	Specify one or more associated domains. Connections to servers within one of these domains are associated with the Ivanti Tunnel.
Excluded Domains	Specify one or more excluded domains. Connections to servers within one of these domains are excluded from the Ivanti Tunnel.
Disconnection Timeout	Enter the disconnection timeout duration (in seconds). You can set any value between 0 and 86,400. Default value is set to 60 seconds.

Item	Description
+ Add Network Rules	Enter the disconnection timeout duration (in seconds). You can set any value between 0 and 86,400. Default value is set to 60 seconds. • DNS Domain Match • DNS Server Address Match • SSID Match • URL String Probe • URL String Probe
+ Add Connection Rules	Connection rules allow when needed, or never allow connections to the networks that evaluate as true. For connection rules, you can specify the following types of parameters: • DNS Domain Match • DNS Server Address Match • SSID Match • URL String Probe • Interface Type Match
Action	Select one of the following options to apply the Rule Type and Value for the configuration: • Connect • Disconnect • Ignore
Enforce Routes	Select this check box to enforce routes through Ivanti Tunnel.
Exclude Local Networks	Select this check box to exclude all local networks.
Include All Networks	Select this check box to include all network types.

5. Setting up single sign-on with Kerberos

The following addresses how to set up single sign-on with Kerberos on iOS devices and contains the following:

- About the setup for single sign-on with Kerberos
- Authentication workflow for single sign-on with Kerberos
- Main tasks for configuring single sign-on with Kerberos (Ivanti EPMM or Ivanti Neurons for MDM)
- SSO with Kerberos configuration field description

5.1. About the setup for single sign-on with Kerberos

The setup described allows Safari and managed apps that support Kerberos to securely access an internal resource using SSO when the device is outside the corporate network. The Key Distribution Center (KDC) sits inside the corporate network. A major architectural change was introduced in Ivanti Tunnel 2.0 allowing Tunnel to use Network Extension framework introduced in iOS 9.0. Due to the support for Network Extension framework, use of Standalone Sentry as a KKDCP is no longer required for SSO with Kerberos.

Related topics

- **Main tasks for configuring single sign-on with Kerberos (Ivanti EPMM or Ivanti Neurons for MDM)**
- **SSO with Kerberos configuration field description**

5.2. Authentication workflow for single sign-on with Kerberos

The following describes the authentication flow for single sign-on with Kerberos:

1. The managed app or Safari domain initiates a connection with the backend resource through the TCP tunnel configured on the Standalone Sentry. The managed app must support Kerberos.
2. The backend resource, via the Standalone Sentry, returns a request to authenticate and the KDC realm information to the device.
3. The device sends an SRV Kerberos DNS query to Ivanti Tunnel. Tunnel matches the requested domains to the domains configured in the SRV key-value pair. The kerberos DNS query is resolved to the host name (target) configured in the SRV key-value pair.
   SRV configuration is not required if packet tunnel provider is configured.
4. The device communicates with the KDC server (target) through Tunnel and a ticket is returned to the device. The ticket is stored on the device.
5. The device presents the ticket to the backend resource for authentication.
6. The device uses the ticket to authenticate to backend resources configured in the single sign-on setting.

5.3. Main tasks for configuring single sign-on with Kerberos (Ivanti EPMM or Ivanti Neurons for MDM)

Following are the main steps for configuring single sign-on with Kerberos:

1. **Configuring SRV (Ivanti EPMM or Ivanti Neurons for MDM)**
2. **Configuring single sign-on (Ivanti EPMM)**
   OR
   Configuring single sign-on (Ivanti Neurons for MDM)

Before you begin

- Set up per-app VPN with Tunnel as described in **Setting up Ivanti Tunnel**. Apply the Tunnel VPN setting to the managed apps that will use single sign-on with Kerberos authentication. The managed app must support Kerberos.
- If you want an app to use a certificate to authenticate the device user to a backend resource when the Kerberos ticket has expired, create a certificate enrollment setting. You will reference the certificate in the single sign-on setting.
- If you do not provide an identity certificate, the device user is prompted to enter a user ID and password when the Kerberos ticket has expired.
- Ensure that devices have access to a Kerberos Domain Controller (KDC) and the backend resources that you specify in the single sign-on setting.

Configuring SRV (Ivanti EPMM or Ivanti Neurons for MDM)

Configuring SRV is not required if you configure packet tunnel provider type in the Ivanti Tunnel VPN configuration.

The SRV feature resolves Kerberos DNS requests from devices in environments with different internal and public Kerberos domain controller (KDC) DNS domains. In order to resolve the SRV query and determine the KDC that handles the authentication requests for the backend server, you configure key-value pairs in the Tunnel VPN configuration for iOS. This feature replaces the need to create an SRV record in your DNS.

> If you are configuring split tunneling in Access, ensure that domain name in the split tunneling configuration matches exactly the SRV record in the Tunnel for iOS configuration.

Procedure

1. In your Unified Endpoint Management (UEM) platform, select the Ivanti Tunnel configuration for iOS to edit.
   - In the Ivanti EPMM Admin Portal, go to Policies & Configs > Configurations.
   - In Ivanti Neurons for MDM, go to Configurations.
2. In Custom Data, add the following key-value pair:
   - Key: SRV_kerberos._tcp.DnsDomainName
   - Value: SRV Priority Weight Port Target

To configure multiple values for the same domain name, add a new row. Enter the same key with a trailing #n, where n is an integer, to the key. Add a trailing #n to the first record as well. Ensure that there are no spaces between the key and #n. If there are multiple

entries, a KDC is contacted based on priority and weight.

Example

Key	Value
SRV_kerberos._tcp.example.com#1	SRV 0 100 88 kdc.example.com
SRV_kerberos._tcp.example.com#2	SRV 0 100 88 kdc2.example.com

3. In Safari Domains, add the root domain.

Configuring the root domain allows all traffic, including Kerberos traffic, to go through Tunnel.

Example : example.com

Alternately, if you do not want to configure the root domain, add the following to Safari Domains:

- the backend resource being accessed.
- _kerberos._tcp.DnsDomainName: configured in Custom Data.
  The realm name in the Kerberos DNS query is case sensitive. Therefore, the DnsDomainName must be in upper case.
- Target: configured in Custom Data.

Configuring the domains ensures that traffic required to resolve the Kerberos DNS request goes through Tunnel.

Example

- sharepoint.example.com
- _kerberos._tcp.EXAMPLE.COM
- kdc.example.com

4. Click Save.

Next steps

- **Configuring single sign-on (Ivanti EPMM)**
  OR
- **Configuring single sign-on (Ivanti Neurons for MDM)**

Related topics

- For more information about the values for the SRV key, see **https://www.ietf.org/rfc/rfc2782.txt.**
- For more information about the SRV key-value pair, see **Additional configurations using key-value pairs for Ivanti Tunnel**.

Configuring single sign-on (Ivanti EPMM)

Specify the URLs or resources that the device user can access using single sign-on (SSO).

- For Realm, enter $REALM$.
- Create a separate Single sign-on configuration for each realm.

Procedure

1. In the Admin Portal, go to Policies & Configs > Configurations.

2. From the Add New drop-down menu, go to iOS and OS X > Single Sign-On Account.
   The New Single Sign-On (SSO) Configuration screen displays.
3. Complete the form.
4. Click Save.
5. In the Configurations page, select the configuration.
6. Click More Actions > Apply To Label.
7. Select a label to apply, and click Apply.

Related topics

- For a description of the fields in New Single Sign-On (SSO) Configuration, see **SSO with Kerberos configuration field description**.

Configuring single sign-on (Ivanti Neurons for MDM)

Specify the URLs or resources that the device user can access using SSO.

> Create a separate Single sign-on configuration for each realm.

Procedure

1. In Ivanti Neurons for MDM, go to Configurations > +Add.
2. Search for single sign-on.
3. Click the Single Sign-On Account configuration
   The Create Single Sign-On Account Configuration page displays.
4. Add the necessary configurations and click Next.
5. Choose a distribution option for the configuration and click Done.
   The configuration is distributed to the devices in distribution option. Select the same distribution option that you selected for the Ivanti Tunnel for iOS app.

Related topics

- For a description of the fields in New Single Sign-On (SSO) Configuration, see **SSO with Kerberos configuration field description**.

5.4. SSO with Kerberos configuration field description

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

Field	Description
Name	Enter a name for this configuration.
Description	Enter additional information that describes this configuration.
User Name	(Required) Enter the Kerberos user name. Ivanti EPMM: You can also specify the variable $USERID$. Ivanti Neurons for MDM: You can also specify the variable $
Realm	(Required) Ivanti EPMM: The default is $Realm$. This is the only valid variable. $Realm$ is supported for LDAP users only. The realm is calculated by extracting the base DN (e.g. DC=auto, DC=MyCompany, DC=com) and converting to a domain. Example: AUTO.MYCOMPANY.COM. **Ivanti Neurons for MDM:** Enter a domain name. Example: AUTO.MYCOMPANY.COM.
Identity Certificate (Ivanti EPMM)	(Optional) Select a certificate enrollment setting from the drop-down list to specify an identity certificate. An app uses this identity certificate to authenticate the device user to the KDC server. After the user is authenticated, the KDC server issues a ticket to the user. If the Kerberos ticket has expired, it is silently renewed after the user is authenticated. If you do not provide an identity certificate, the device user is prompted to enter a user ID and password when the Kerberos ticket has expired.
Certificate (Ivanti Neurons for MDM)	(Optional) Select the certificate to use. An app uses this identity certificate to authenticate the device user to the KDC server. After the user is authenticated, the KDC server issues a ticket to the user. If the Kerberos ticket has expired, it is silently renewed after the user is authenticated. If you do not provide an identity certificate, the device user is prompted to enter a user ID and password when the Kerberos ticket has expired.
URL Prefix Matches (Required) Add the URLs or resources that the device user can access using SSO. At least one URL is required. If a bundle ID (application ID) is configured, SSO is enabled for the specified apps only when the apps access the URLs that match the configured URL prefixes. If a bundle ID (application ID) is not	

Field	Description
configured, SSO is applicable to all apps that support SSO when they access the URLs that match the configured URL prefixes.	
+	Click to add an URL.
URL	Enter the URL that the user can access using SSO. • The website or resource must support Kerberos based authentication. • Entries must begin with the URL scheme: HTTP:// or HTTPS:// • A simple string match is performed. For example, http://www.example.com/ does not match http://www.example.com:80/ • If an entry does not end with the character /, a / is appended to the entry. • For devices running iOS 9 through the most recently released version as supported by Ivanti, you can use a single wildcard * to specify all matching values. For example, http://*.example.com matches both http://store.example.com/ and http://www.example.com/ However, a wildcard at the end of the URL will not work. Example of incorrect url: http://www.example.com/* • The entries http://.com and https://.com match all HTTP and HTTPS URLs, respectively.
Description	Enter additional information describing this resource.
-	Click to delete the URL.
Application Identifier Matches (Optional) Add the apps that the device user can use to access the URLs or resources listed in URL Prefix Matches without having to enter their enterprise credentials. You can add up to twenty bundle IDs (application IDs) per configuration. If no apps are entered, the device user can access the URLs or resources from any app without having to enter their enterprise credentials.	
+	Click to add an app.
BundleID	Enter an exact or partial bundle ID (application ID) for the app. Use the following rules for formatting an entry:

Field	Description
	• The string you specify can be an exact match with a bundle ID. Example: com.mycompany.myapp • Partial matches are supported. • The string you specify can match a prefix of a bundle ID by using exactly one * wildcard character. The * appears after a period character, and at the end of the string. Example: com.mycompany.* matches any app for which the bundle ID begins with com.mycompany.
Description	Enter additional information describing the app.
–	Click to delete the entry.

6. Additional configurations using key-value pairs for Ivanti Tunnel

Key-value pairs are used to customize Ivanti Tunnel for iOS app behavior. These key-value pairs define app behavior such as idle timeout, email address for sending debug information, and level of log detail that is collected.

The following table provides the key-value pairs for customizing Tunnel for iOS.

Key	Value
Manage Tunnel timeout	
disconnectTimeoutInSeconds (Ivanti EPMM)	Enter 0 or a number between 5 - 18000. If the value is 0, then Tunnel VPN never disconnects itself. You have to manually disconnect the VPN in the Tunnel. If the value is > 0, the Tunnel VPN is disconnected after number entered. If this key-value pair is not configured, the default is 60 seconds.
TcpIdleTmoMs	Enter any integer between 5000 - 18000000. The timeout is measured in milliseconds. Configuring idle timeout allows you to control the idle session timeout for the TCP connection between the app and the backend server. You may want to configure idle timeout if the backend server takes more than 60 seconds to respond to a request. The default idle timeout with Standalone Sentry for per-app VPN if the key-value pair is not configured: 60 seconds. For packet tunnel, Ivanti recommends setting the idle timeout equal to or larger than the idle timeout for the enterprise server being accessed. If you do not know the idle timeout for the server, set the value to 3600000.
Troubleshooting	
debugInfoRecipient (Available as field value in Ivanti Neurons for MDM)	Enter an email address to forward the debug information.
LogLevel	Enter debug <Log Level> Use one of the following log level options. The options are listed from the least to the most verbose level. • error: Captures error logs if the Tunnel app errors out while performing an action. • warning: Captures warning messages logged if there is missing or incorrect information that might cause an error. This log level is rarely used. • info: Captures informational level details such as, log prints inputs, metadata, parameter values. • debug: Captures debug level information such as, actions, operations, values of critical data, and information that is helpful in debugging. • session: Captures everything that occurs during a tunnel session. • packet: Captures packet level information, such as, length in bytes. Used for troubleshooting DNS queries and responses to and from Tunnel. Default if the key-value pair is not configured: info
UseSecureEMail	Enter true. Tunnel uses Email+ to send debug logs. If the key-value pair is not configured, Tunnel uses the native iOS email client to send debug logs.
SendDeviceID	Enter true. Tunnel provides the device ID to Access. The device ID is reported on Access in Reports > Errors.

Key	Value
	The key-value pair is useful in identifying devices that encounter connection errors when authenticating through Access. Default if the key-value pair is not configured: false.
MaxLogFolderSize	When this KVP is set to true, then support for setting the log folder size in MB is enabled. If the KVP is not set, it defaults to 10MB.
EnableConsoleLogging	When this KVP is set to true, then Tunnel app logs messages to console
AnalyticsEnabled	**true**: Enables collection of analytics data in network extension. **false**: Disables collection of analytics data in network extension. Default value if the key-value pair is not configured: **true**.
DNS and network	
PublicDNS	Enter a space-separated list of DNS servers that are accessible from the device. Each DNS entry is - separated by a space. IPv4 and IPv6 addresses are supported. Since (managed) apps have access to the DNS servers configured on the device, this KVP is needed only in rare cases. Example 8.8.8.8 8.8.8.1
IPv6NetworkPrefix	IPv6 ULA network prefix to use for internal NAT table.
DNS query for SRV record (for SSO with Kerberos)	
SRV_kerberos._tcp.DnsDomainName Where DnsDomainName is the internal domain name of the KDC server. Example: SRV_kerberos._tcp.example.com	Enter SRV Priority Weight Port Target Where: • Priority is the priority of the server. • Weight is the load-balancing mechanism that is used when selecting a target • Port is the port number the server is listening. • Target is the fully qualified domain name (FQDN) of the KDC server. Example SRV 0 100 88 kdc.example.com SRV record derived from the key-value pair: _kerberos._tcp.example.com. SRV 0 100 88 kdc.example.com. Ensure that the domain configured for DnsDomainName and for Target is also configured in Safari Domains in the Tunnel VPN configuration. Configuring the domains in Safari Domains ensures that the traffic goes through Tunnel.
Certificates	
DisablePinning	false: Default, if the key-value pair is not configured. Certificate pinning is enabled. true: Certificate pinning is disabled. Disabling certificate pinning is not recommended for security reasons. The Standalone Sentry server certificate is automatically pushed to the device.

Key	Value
Packet-tunnel	
IPRoutes	IP routes of the iOS or macOS device VPN. Enter list separated by semicolon. The default value if the key-value is not configured is 0.0.0.0/0 Example 10.0.0.0/8;172.16.0.0/16
ExcRoutes	IP routes that will be excluded from IPRoutes. Example 10.10.10.10/32.
SplitUDPPortList	Enter list of UDP ports to send through Tunnel VPN. All other UDP packets are sent directly to destination. If the KVP is not configured, all UDP packets are sent through Tunnel VPN. Example 53;161-162;200-1024 Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.
MTU	Tunnel MTU. The default value if the key-value is not configured is 1400.
TunIP	IP address of the VPN network interface. Configure only if customer network is in the same range. Example 192.168.13.10
AtpProbeIdleSec	Sets the minimum idle time, in seconds, after which probe packets are sent out with outbound Tunnel traffic. If Tunnel does not receive a response for at least one of the probes sent, the existing connection is dropped and a new connection is established with the server. The minimum idle time is based on the last inbound response received by Tunnel. For example, if the value is 60 seconds, if Tunnel does not receive any inbound traffic for 60 seconds, probe packets are sent with the next outbound Tunnel traffic. Default value if the key-value pair is not configured: 60 seconds
AtpProbeIntervalSec	Sets the interval, in seconds, between probe packets sent after the minimum idle time specified in AtpProbeIdleSec. Default value if the key-value pair is not configured: 1 second
AtpProbeCount	Sets the total count of the probe packets sent after the minimum idle time specified in AtpProbeIdleSec. Default value if the key-value pair is not configured: 5
App proxy	
DirectLocalhost	Enter true. Configure if using app proxy Tunnel. The key-value pair is required for Tunnel to handle app proxy localhost traffic from apps. true: If an app uses localhost, ::1, or 127.0.0.1, the localhost app proxy (TCP) traffic is redirected to the device itself.

Key	Value
SplitUDPPortList	Enter list of UDP ports to send through Tunnel VPN. All other UDP packets are sent directly to destination. If the KVP is not configured, all UDP packets are sent through Tunnel VPN. Example 53;161-162;200-1024 Standalone Sentry supports only limited types of UDP traffic, such as DNS traffic. Audio and video traffic through Standalone Sentry is not supported. Therefore, Ivanti recommends configuring SplitUDPPortList to manage UDP traffic.
EnableLegacyAppProxyDNSSetup	When this KVP is set to true, then Tunnel code will use the old logic. But if this KVP does not exist or if this KVP is set to false, then the tunnel code will not set the system level DNS servers.

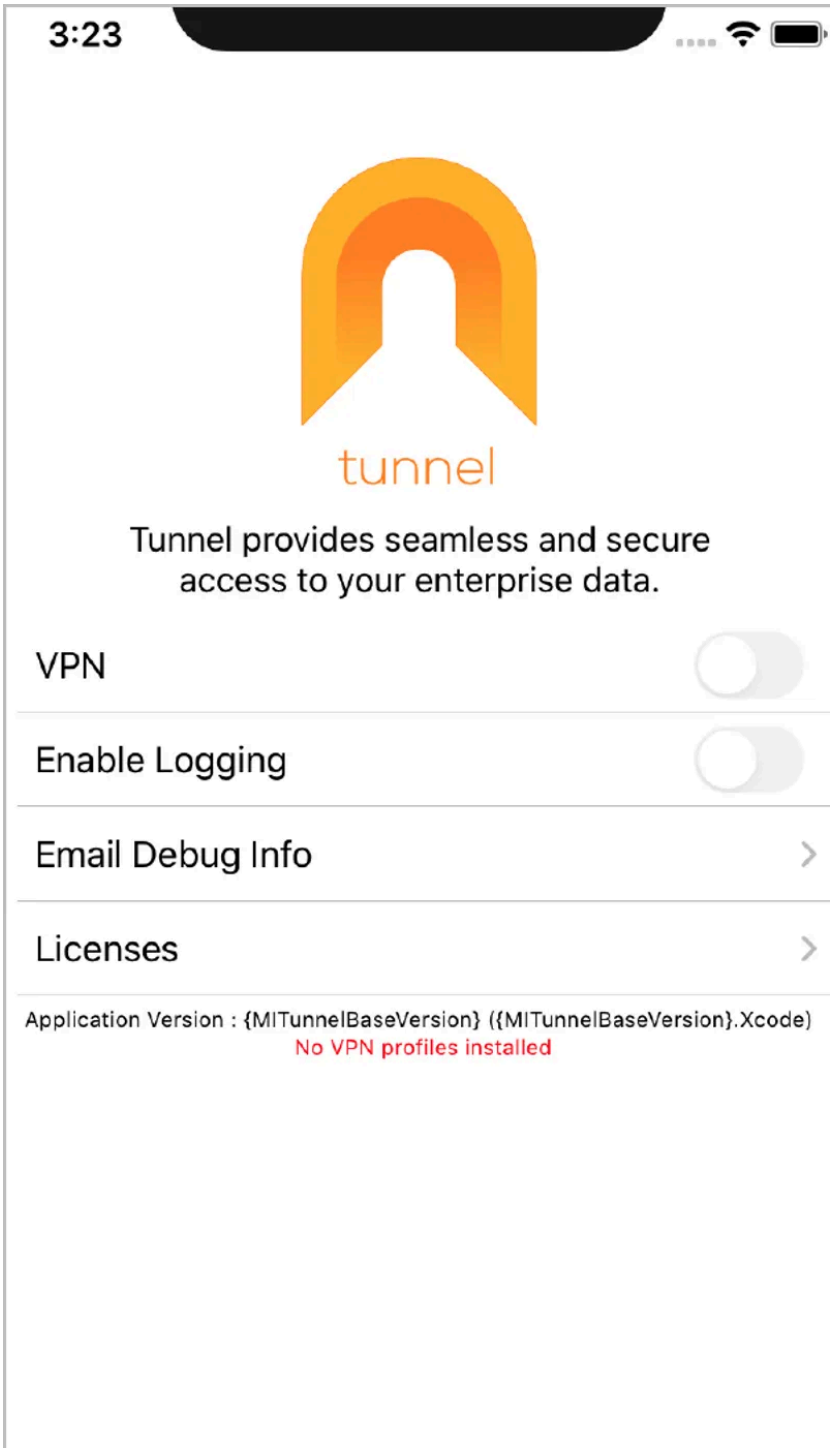
7. What users see in Ivanti Tunnel for iOS

- Ivanti Tunnel installation
- Emailing debug log information
- Clearning logs
- Sharing debug logs

7.1. Ivanti Tunnel installation

Ivanti Tunnel is available in the Apple AppStore. You can install the app directly from the Apple AppStore.

Figure 1. Tunnel



- If your device does not have the Tunnel VPN profile, you will see the 'No VPN profiles installed' message. After the Tunnel VPN profile is pushed to the device, the message goes away.

- If the Tunnel VPN profile is installed on your device, when you tap a supported managed app and the app attempts to connect to a backend resource, the VPN connection is automatically turned on, and the app can securely connect to the enterprise resource. In some 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. You have to turn on VPN only once for the device.

7.2. Emailing debug log information

IT administrators may require Tunnel debug and log data for troubleshooting purposes. Ivanti Tunnel provides device users the option to email the Tunnel debug and log file to the IT administrator. Depending on the Tunnel setup, users can either send debug logs using the native iOS email app or using Email+.

By default Ivanti Tunnel uses the native iOS email app.

Before you begin

Do one of the following:

- Ensure you have an email account set up in the native iOS email app your device.
  OR
- If your administrator has configured Ivanti Tunnel to use Email+, ensure that Email+ is deployed on your device.
  See the description for UseSecureEMail in the table in **Additional configurations using key-value pairs for Ivanti Tunnel**

Procedure

1. In the Ivanti Tunnel app, turn on Enable Logging.
2. Tap Email Debug Info.
3. Log information is included only if there has been activity using the Tunnel app.
4. Enter the email address provided by your administrator.
5. Tap Send.

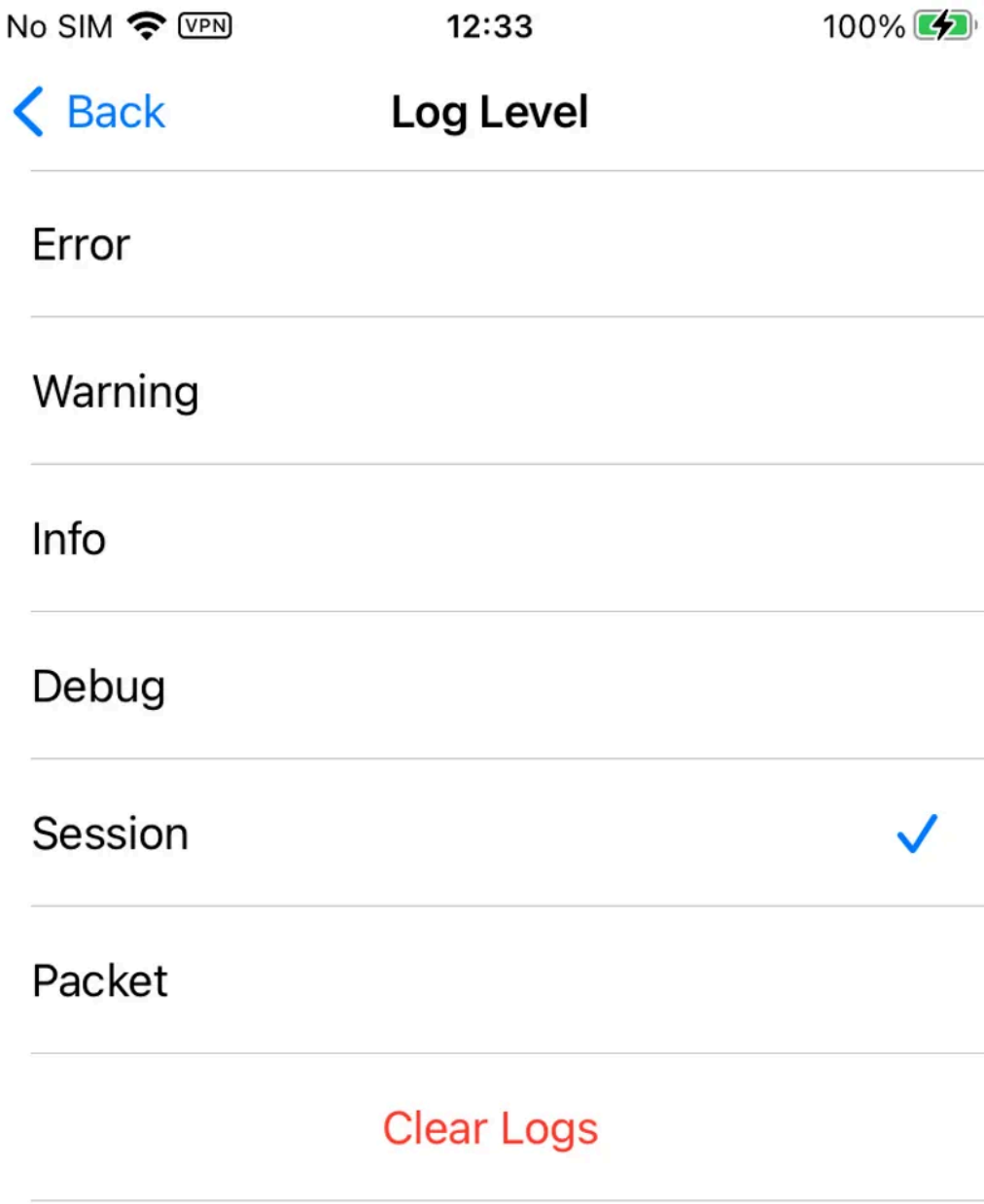
Log information continues to be collected till Enable Logging is turned off. Therefore, after collecting the logs for debugging, turn off Enable Logging to stop collecting app logs. If the UseSecureEmail key is configured as true in the Tunnel VPN configuration, then Email+ must be set up on the device. If UseSecureEmail is not configured, then the iOS native email app must be set up. Otherwise, users see an error message and logs cannot be emailed.

If the device has multiple Ivanti Tunnel VPN profiles,

- the email body includes the number of profiles and lists each profile with its app list and Safari domains.
- if Ivanti Tunnel is set up to use Email+, and any one of the profiles is set up to use Email+, then all profiles automatically use Email+.

7.3. Clearning logs

The **Clear Logs** option in the **Settings** section of the Tunnel iOS app allows the user to clear logs from the application. This helps the user to clear the logs cache and share the latest logs related to the issues faced with administrators.


No SIM
VPN
12:33
100%
Back
Log Level
Error
Warning
Info
Debug
Session
Packet
Clear Logs

7.4. Sharing debug logs

Tunnel for iOS logs can now be shared locally or through other managed app such as OneDrive, Outlook or Teams and not dependent on Email+ or native email client.

