# BYOD Policy for Microsoft Intune Devices - Deployment Guide


Published 15 May 2020

Document Version 1.0.1

Pulse Secure, LLC
2700 Zanker Road,
Suite 200 San Jose
CA 95134

www.pulsesecure.net



*BYOD Policy for Microsoft Intune Devices - Deployment Guide*

The information in this document is current as of the date on the title page.

**END USER LICENSE AGREEMENT**

The Pulse Secure product that is the subject of this technical documentation consists of (or is intended for use with) Pulse Secure software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.pulsesecure.net/support/eula/. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

## Revision History

The following table lists changes made to this document:

| Document Revision | Release | Date | Feature | Changes |
|---|---|---|---|---|
| 1.0.1 | PCS 9.1R5 | May 2020 | Update | Updated the "Supported Platforms" section.<br>Updated the "Configuring PCS with Microsoft Intune" section.<br>Updated logging screens. |
| 1.0 | PCS 9.1R5 | April 2020 | Initial release | Initial release |

# Contents

# Introduction



## Understanding the Device Access Management Framework

The device access management framework leverages mobile device management (MDM) services so that you can use familiar Pulse Connect Secure client policies to enforce security objectives based on your device classification scheme: device status is MDM enrollment complete or incomplete; device status is MDM-policy compliant or non-compliant; device is employee owned or company owned; device platform is iOS, Android, or neither; and so forth.

In this framework, the MDM is a device authorization server, and MDM record attributes are the basis for granular access policy determinations. For example, you can implement policies that allow devices that have a clean MDM posture assessment and are compliant with MDM policies to access the network, but deny access to servers when you want to prevent downloads to employee-owned devices or to a particular platform that might be vulnerable. To do this, you use the device attributes and status maintained by the MDM in Pulse Secure client role-mapping rules, and specify the device-attribute-based roles in familiar Pulse Secure client policies.

The framework simply extends the user access management framework realm configuration to include use of device attributes as a factor in role mapping rules. Figure 1 illustrates the similarities.

Figure 1 User Access Management Framework and Device Access Management Framework

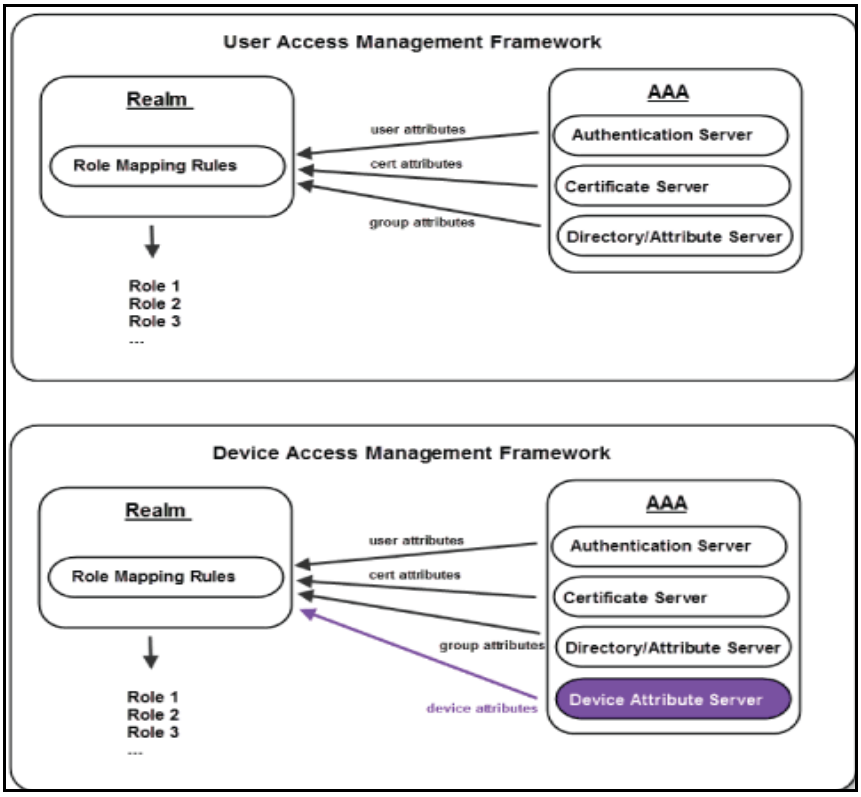



Table 1 summarizes vendor support for this release.

Table 1 MDM Vendors

| Product | Vendor |
|---|---|
| Pulse Connect Secure | • Pulse Workspace (PWS)<br>• AirWatch MDM<br>• MobileIron MDM<br>• Microsoft Intune |

Table 2 summarizes supported methods for determining the device identifiers.

Table 2 Device Identifiers

| Product | Policies |
|---|---|
| Pulse Connect Secure | Device certificate (required) |

Table 3 summarizes policy reevaluation features.

Table 3 Policy Reevaluation

| Product | Policy Reevaluation |
|---|---|
| Pulse Connect Secure | The MDM is query and policies evaluated only during sign-in. If desired, you can use the user role session timeout setting to force users to sign in periodically. If you use a certificate server for user authentication, the users are not prompted to sign in again; however, if you have enabled user role notifications, users do receive a notification each time sign-in occurs. |

**Note:** The dynamic policy evaluation feature is not used in the device access management framework.

Table 4 summarizes the policies in which you can specify device-attribute-based roles.

Table 4 Policies

| Product | Policies |
|---|---|
| Pulse Connect Secure | Resource policies or resource profiles |

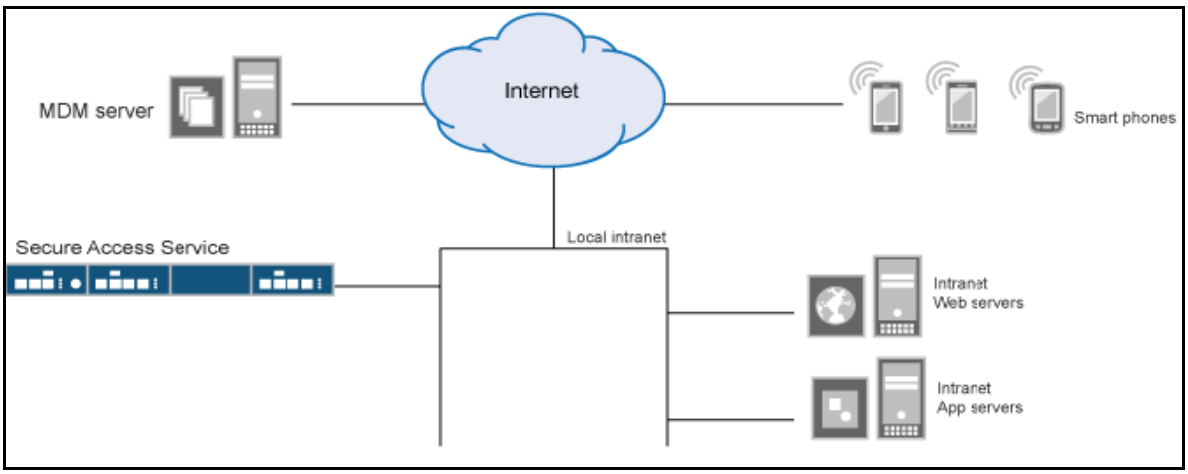
## Solution Overview

In the past, to ensure security and manageability of the corporate network, enterprise information technology (IT) departments had restricted network access to company-issued equipment. For mobile phones, the classic example was the company-issued BlackBerry handset. As powerful mobile smart phones and tablets have become commonly held personal possessions, the trend in enterprise IT has been to stop issuing mobile equipment and instead allow employees to use their personal smart phones and tablets to conduct business activities. This has lowered equipment costs, but BYOD environments pose capacity planning and security challenges: how can an enterprise track network access by non-company-issued devices? Can an enterprise implement policies that can restrict the mobile devices that can access the network and protected resources in the same way that SSL VPN solutions restrict user access?

MDM vendors have emerged to address the first issue. MDMs such as AirWatch, MobileIron, Microsoft Intune provide enrollment and posture assessment services that prompt employees to enter data about their mobile devices. The MDM data records include device attributes and posture assessment status that can be used in the access management framework to enforce security policies.

Figure 2 shows a deployment with Pulse Connect Secure and the MDM cloud service.

Figure 2 Solution Topology




The solution shown in this example leverages the Pulse Secure access management framework to support attribute-based network access control for mobile devices. In the device access management framework, the MDM is a device authorization server and MDM record attributes are the basis for access policy determinations. For example, suppose your enterprise wants to enforce a policy that allows access only to mobile devices that have enrolled with the MDM or are compliant with the MDM posture assessment policies. You can use the attributes and status maintained by the MDM in role-mapping rules to implement the policy.

In this framework, a native supplicant is used to authenticate the user of the device. The device itself is identified using a client certificate that contains device identity. The client certificate can be used to identify the device against the MDM records and authenticate the user against a certificate server.

The Pulse Secure solution supports granular, attribute-based resource access policies. For example, you can implement policies that allow devices that have a clean MDM posture assessment and are compliant with MDM policies to access the network, but deny access to servers when you want to prevent downloads to employee-owned devices or to a particular platform that might be vulnerable.

# Deploying a BYOD Policy for Microsoft Intune Managed Devices

This example shows how to use policies to enable security based on device identity, device posture, or user identity in a bring your own device (BYOD) environment for an enterprise that uses Microsoft Intune® for mobile device management (MDM). It includes the following information:



## Requirements

Table 5 lists version information for the solution components shown in this example.

Table 5 Component Version Information

| Component | Version |
|---|---|
| Connect Secure | Release 9.1R5 or later is required. |
| Microsoft Intune MDM | Release version 2002 is used in this example. Any version that supports the device ID and device attributes you plan to query is compatible. |

## Supported Devices

- Google Android 8.0 and later
- Apple iOS 11.0 and later

## Configuring the Microsoft Intune MDM Service

When the user installs the MDM application on the device and completes enrollment, the MDM pushes the device certificate to the device. After enrollment, the MDM maintains a database record that includes information about the enrollee—attributes related to device identity, user identity, and posture assessment against MDM policies.

Table 6 describes these attributes. In this solution, these attributes are used in PCS role mapping that is the basis for network access and resource access policies. When you configure role-mapping rules, you specify the normalized attribute name.

Table 6 Microsoft Intune Device Attributes

| Intune Attribute | Normalized Name | Description | Data Type |
|---|---|---|---|
| complianceState | isCompliant | True or false (string) based on whether device is compliant or non-compliant. | Boolean |
| isManaged | isEnrolled | True or false (indicating whether the client is managed by Intune or not). | Boolean |
| macAddress | macAddress | MAC address of the device. | String |
| serialNumber | serialNumber | Serial number of the device. Applies to iOS Devices only. | String |
| imei | IMEI | The device unique identifier. IMEI (15 decimal digits: 14 digits plus a check digit) or IMEISV (16 digits) includes information on the origin, model, and serial number of the device. | String |
| udid | UDID | The device unique identifier.<br>Unique Device Identifier (UDID), which is a sequence of 40 letters and numbers that is specific to iOS devices. | String |
| meid | MEID | MEID is 56 bits<br>long (14 hex digits). It consists of three fields, including an 8-bit regional code (RR), a 24-bit manufacturer code, and a 24-bit manufacturer-assigned serial number. | String |
| osVersion | osVersion | OS Version of the device. | String |
| model | Model | Model of the device. | String |
| manufacturer | manufacturer | Device Manufacturer. | String |
| azureDeviceId | deviceId | The device Id of the device after it has work place joined with Azure Active Directory. | String |
| lastContactTimeUtc | lastSeen | The date time when the device last checked in with the Intune management service endpoint. | String<br>The format is<br>MM/DD/YYYY HH:MM:SS |

Refer to third-party documentation for complete information and configuration details.

To configure the MDM:

1. Enroll devices in the MDM using the methods supported by the MDM.
2. Create a profile. The profile determines many MDM management options. The following configurations are key to this solution:

1. Create trusted certificate profiles in Intune. For detailed steps, refer to the procedure in the Microsoft Intune document.

Figure 3 Create Trusted Certificate - Android

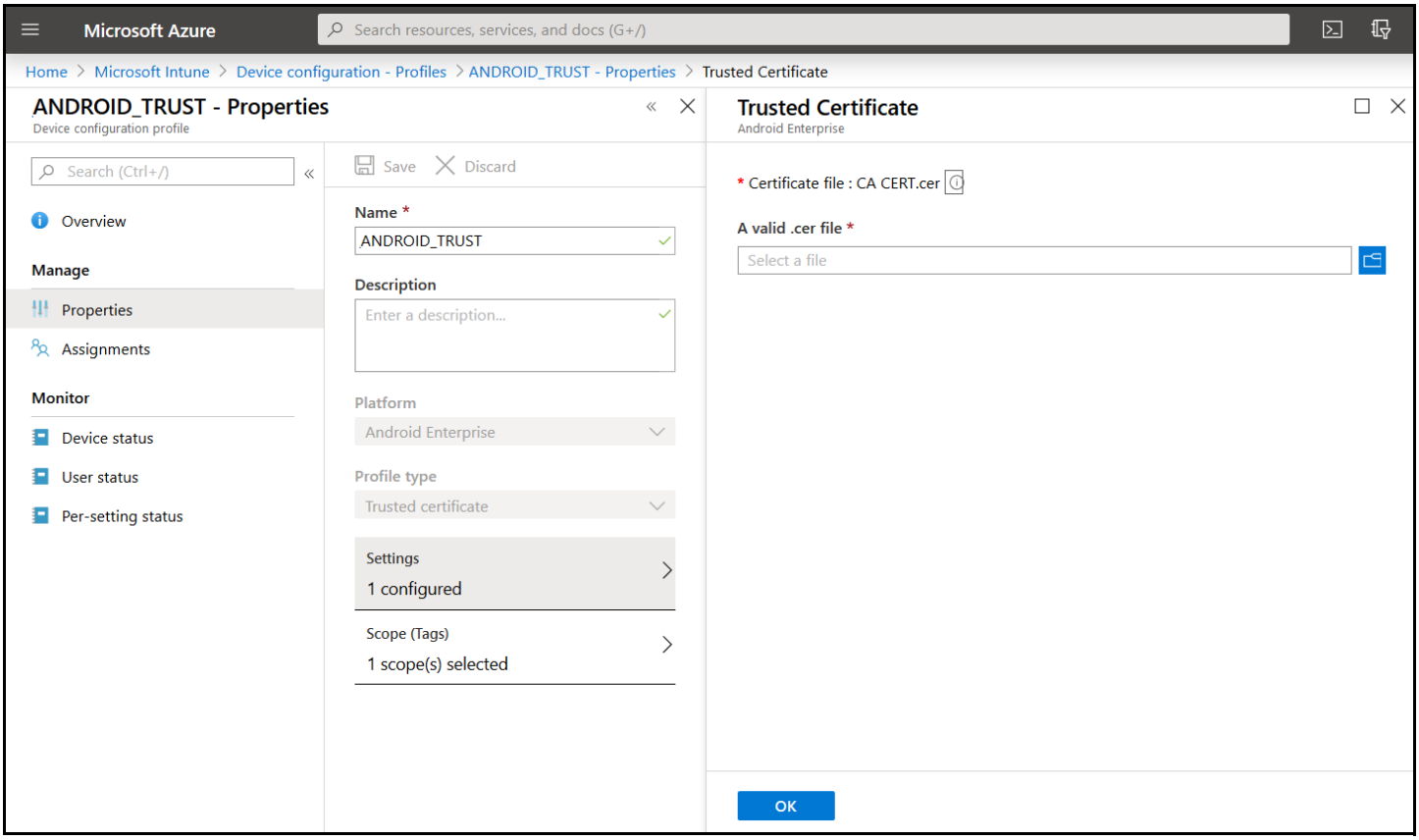
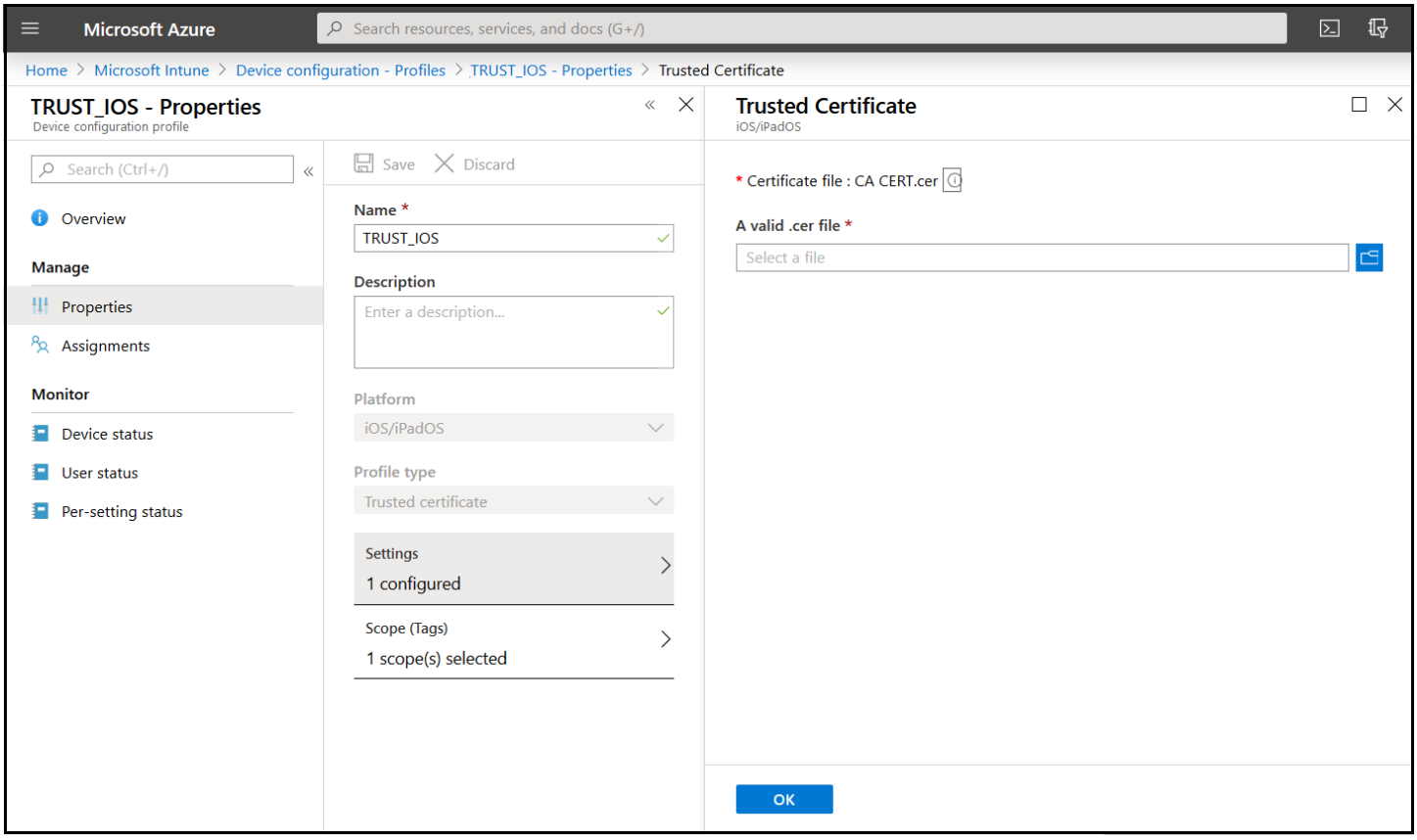


Figure 4 Create Trusted Certificate - iOS




2. Create and assign SCEP certificate profiles in Intune. For detailed steps, refer to the procedure in Microsoft Intune document.

Figure 5 Create SCEP Certificate Profile - Android

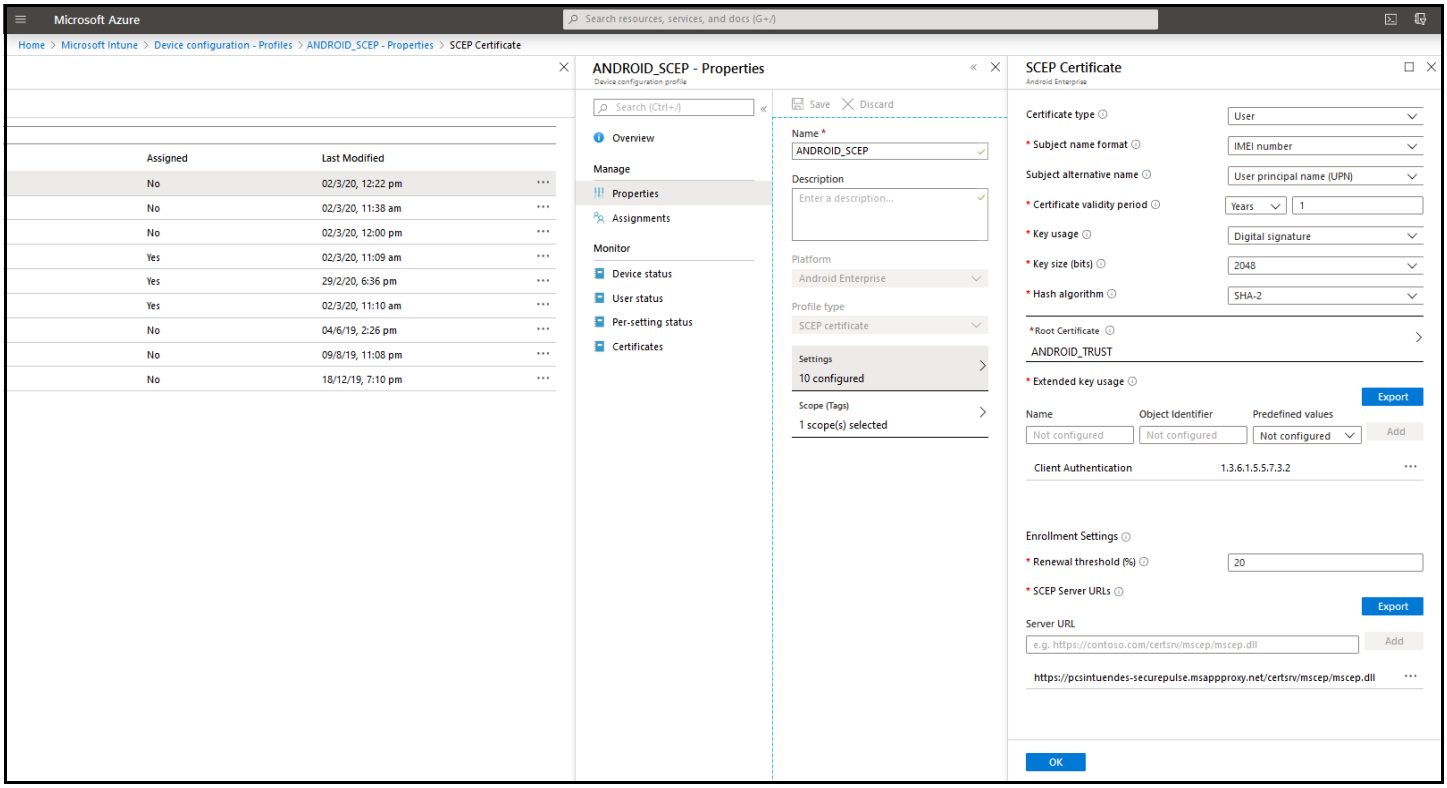
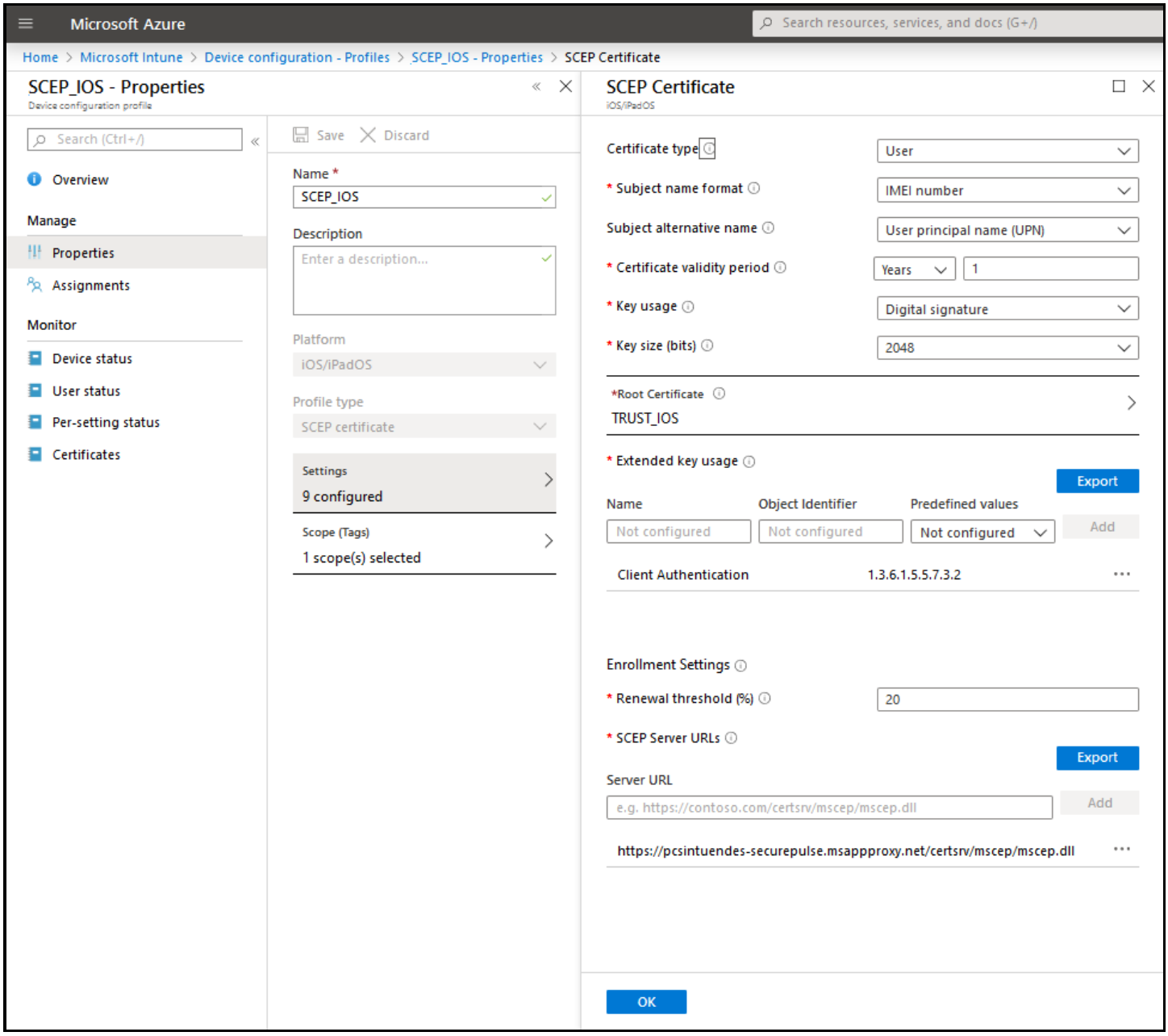


Figure 6 Create SCEP Certificate Profile - iOS




3. Create VPN profile in Intune. For detailed steps, refer to the procedures in Microsoft Intune document.

Figure 7 Create VPN Profile - Android

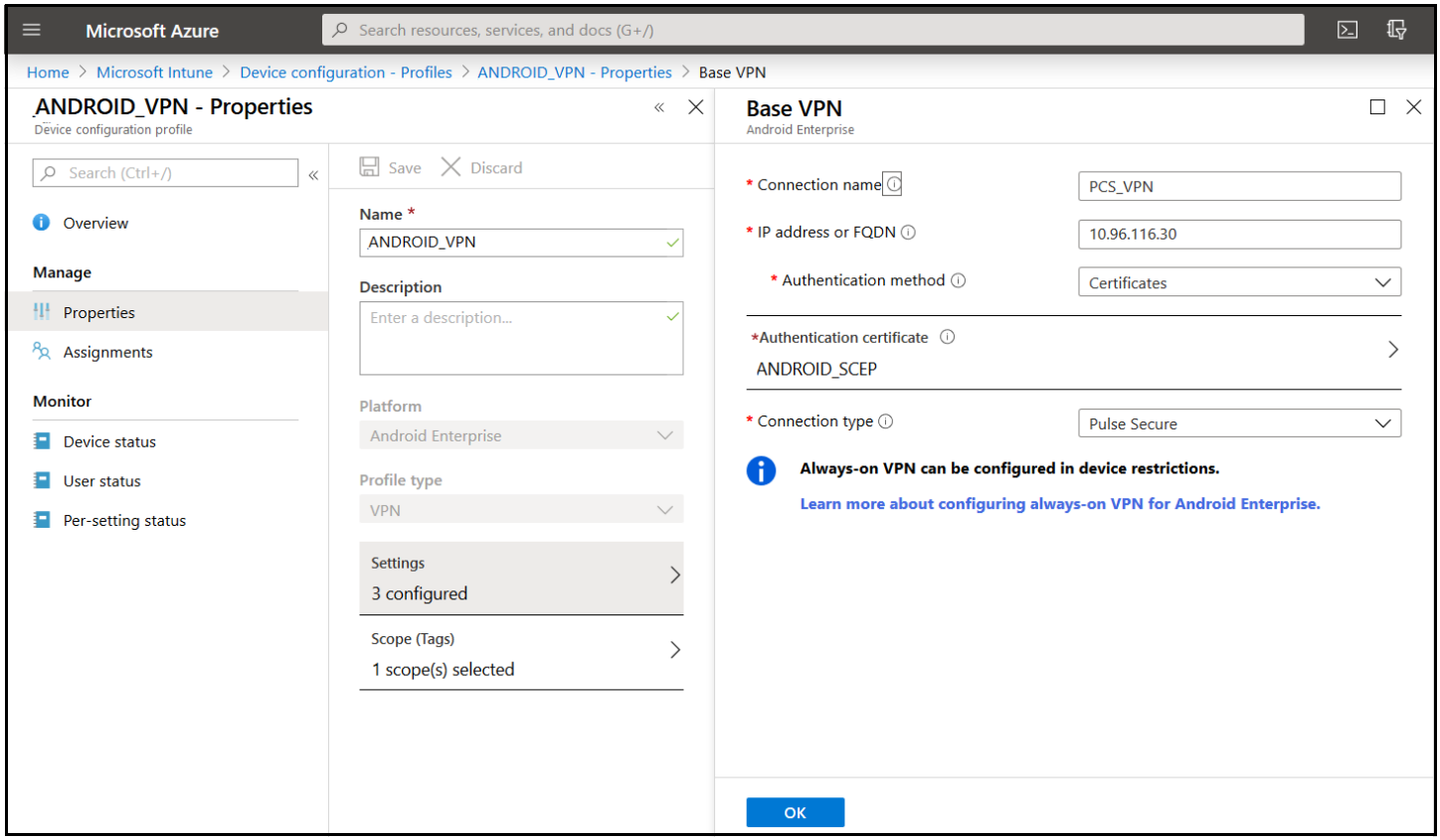
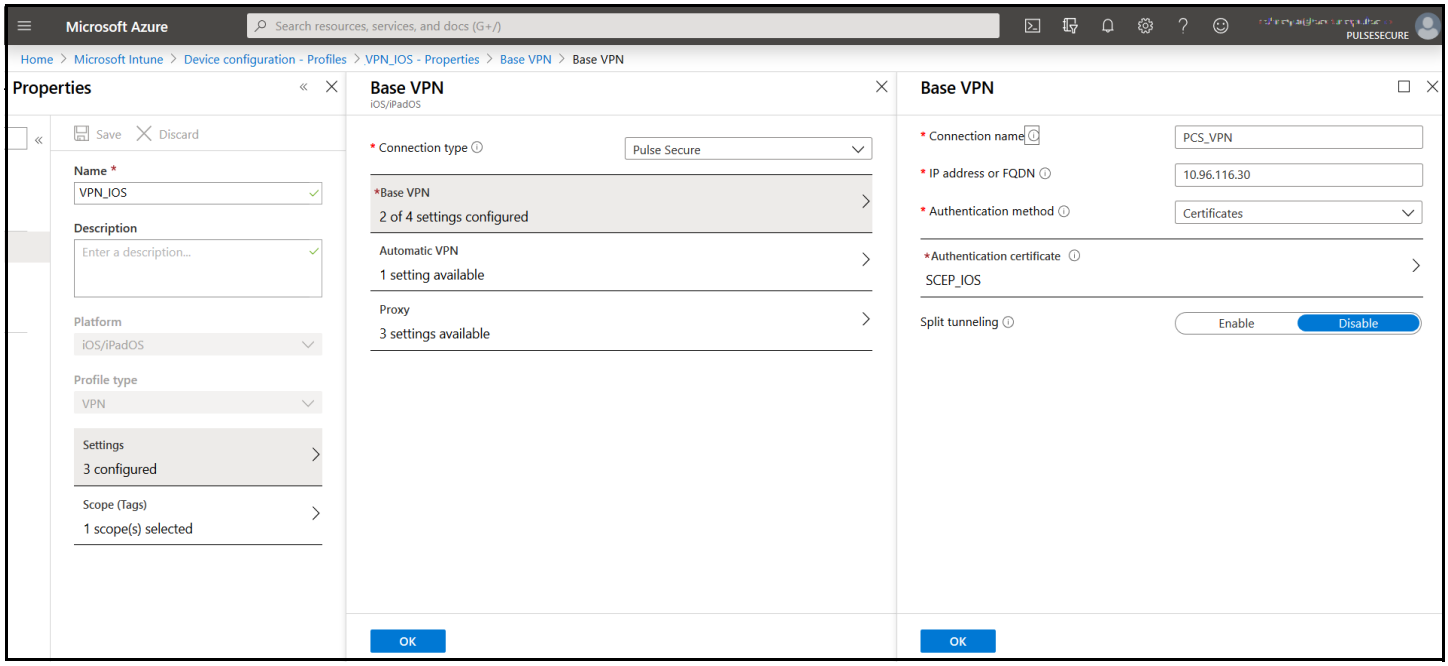



Figure 8 Create VPN Profile - iOS



## Configuring PCS with Microsoft Intune

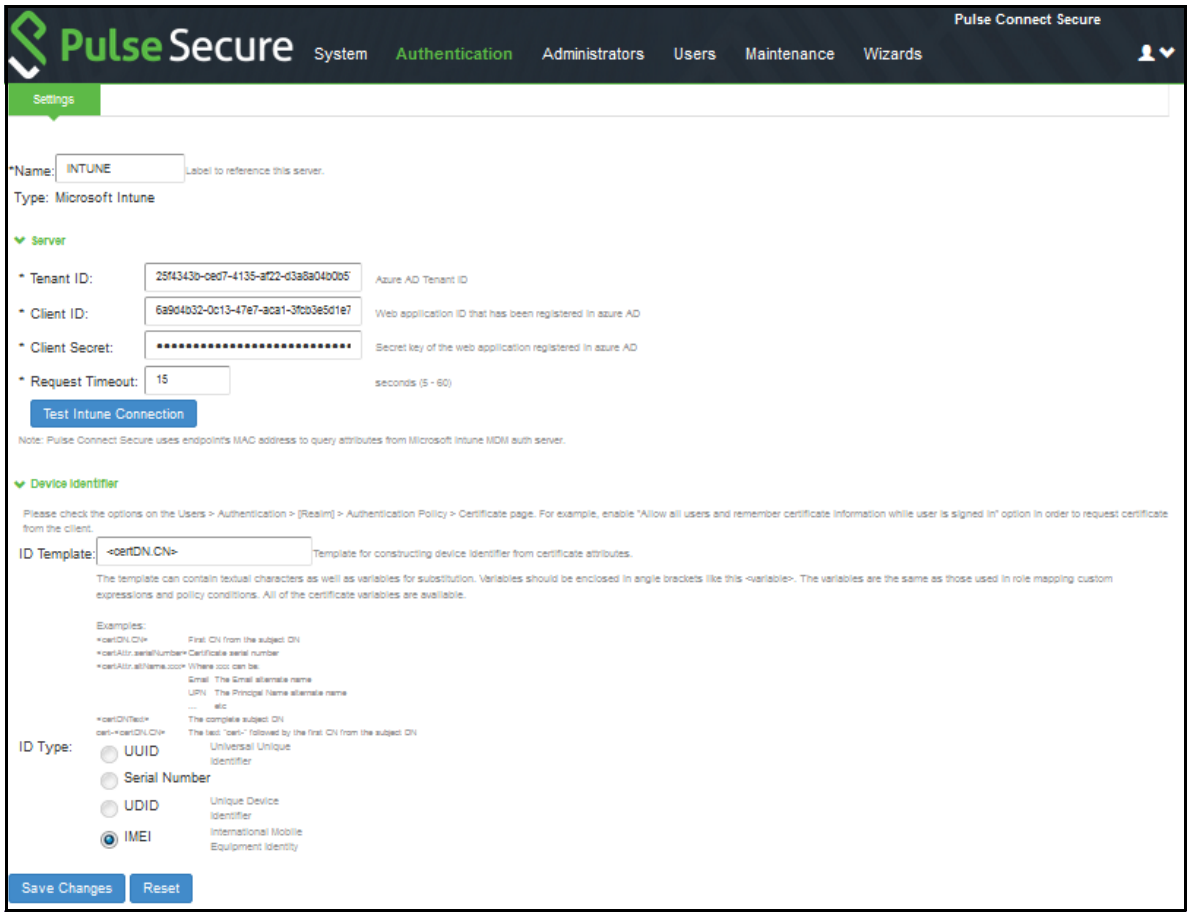
Microsoft Intune is an MDM server which provides the device compliance status for the mobile devices. PCS retrieves the device attributes from Microsoft Intune and uses it for compliance assessments and role assignment. This feature integrates Microsoft Intune and PCS for providing compliance check and onboarding of devices. This feature works only with certificate authentication.

To configure Microsoft Intune MDM server:

1. Select **Authentication > Auth. Servers > New MDM Server.**
2. Enter the server name, select Microsoft Intune as MDM.
   - Enter the Azure AD Tenant ID.
   - Enter the Web application ID or Client ID that is registered in Azure AD.
   - Enter the Client Secret key registered in the Azure AD.
   - Enter the Timeout duration in seconds. Default is 15 seconds.

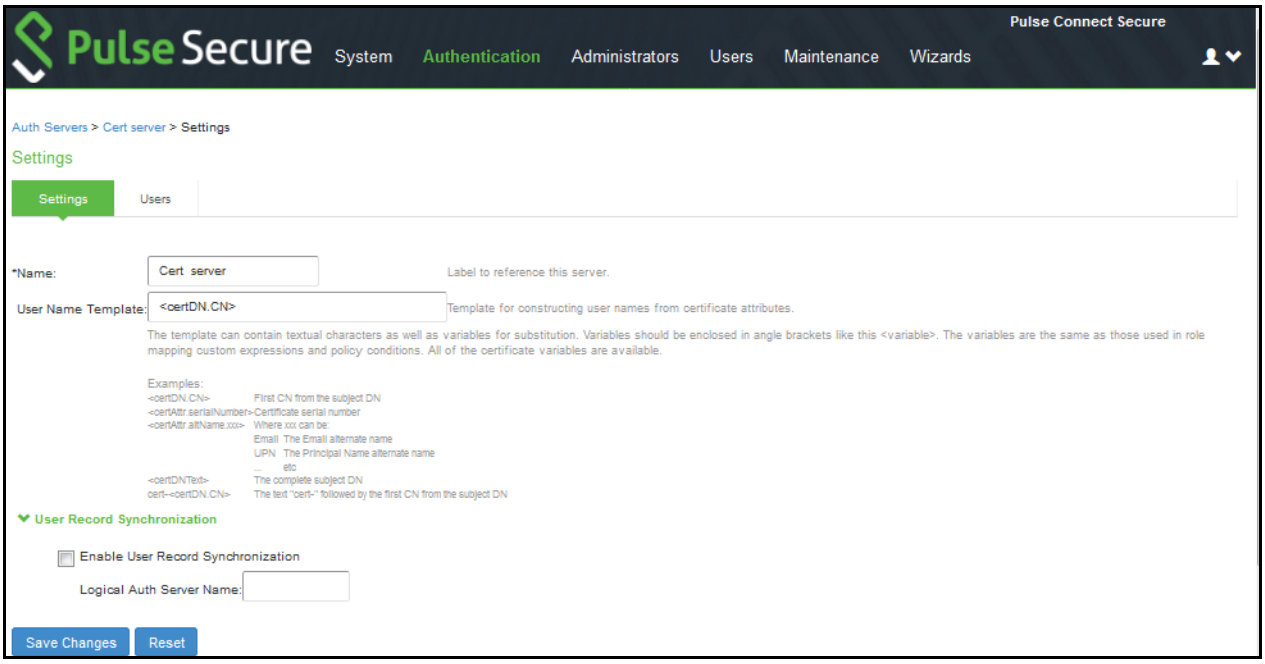
   To obtain Tenant ID, Client ID, Client Secret Key, see "Viewing Client ID, Tenant ID, and Client Secret" on page 24.
3. Click **Save changes**.

Figure 9 Intune MDM Server



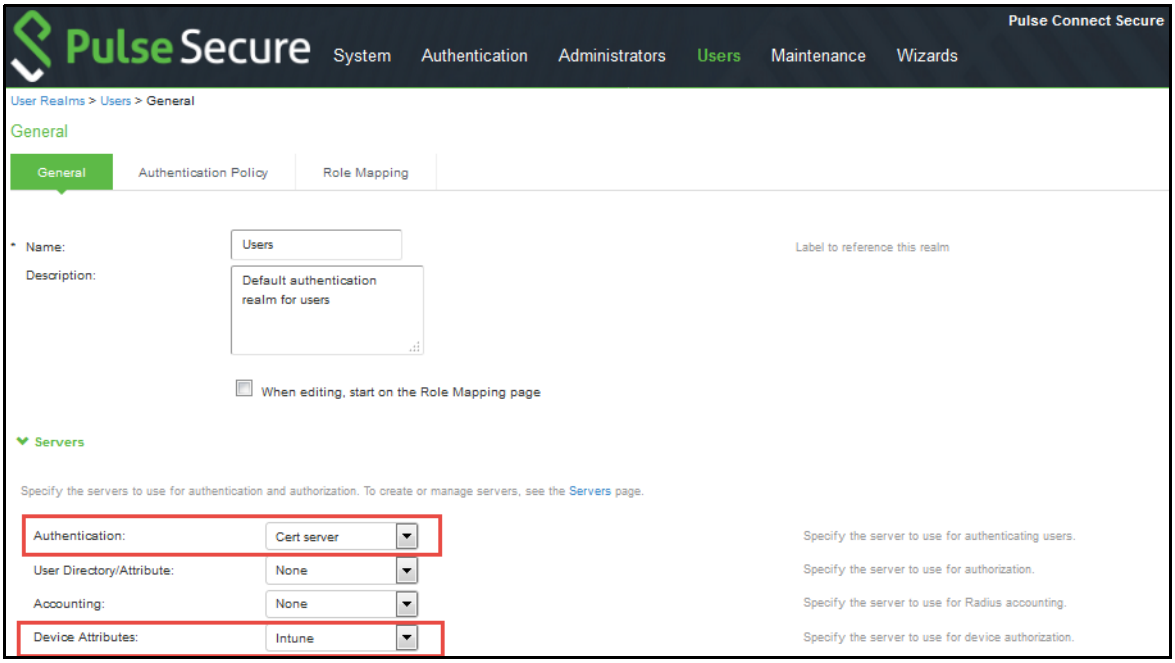
4. Select **Authentication > Auth. Servers > Cert Server** and create certificate server.

Figure 10 Cerificate Server




5. Select **Users > User Realms** and select the **Authentication server** and **Device Attribute server** for Microsoft Intune.

Figure 11 Realm




6. Select **Role Mapping** tab of the user realm to create role mapping rules. Configure the role mapping rules based on the Microsoft Intune supported device attributes.

Figure 12 Role Mapping Configuration Page

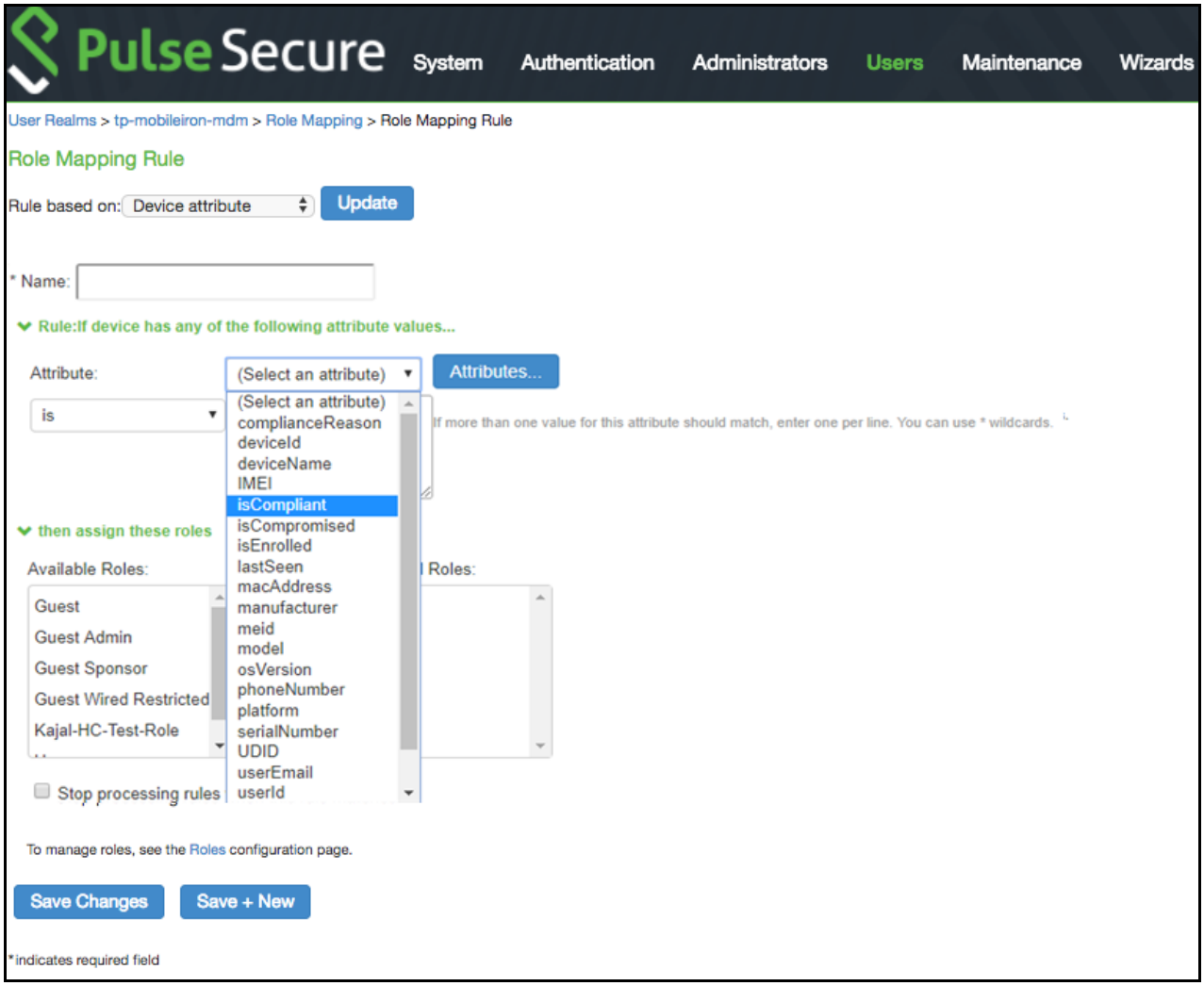


Table 7 Microsoft Intune Role Mapping Attributes

| Role Mapping Attribute Name | Microsoft Intune Attribute Name | Description | Data Type |
|---|---|---|---|
| deviceid | azureDeviceId | The device Id of the device after it has work place joined with Azure Active Directory. | String |
| IMEI | imei | The device unique identifier. IMEI (15 decimal digits: 14 digits plus a check digit) or IMEISV (16 digits) includes information on the origin, model, and serial number of the device. | String |
| isCompliant | complianceState | True or false (string) based on whether device is compliant or non-compliant. | Boolean |
| isEnrolled | isManaged | True or false (indicating whether the client is managed by Intune or not). | Boolean |
| lastSeen | lastContactTimeutc | The date time when the device last checked in with the Intune management service endpoint. | String<br>The format is<br>MM/DD/YYYY HH:MM:SS |
| macAddress | macAddress | MAC address of the device. | String |
| manufacturer | manufacturer | Device Manufacturer. | String |
| meid | meid | MEID is 56 bits<br>long (14 hex digits). It consists of three fields, including an 8-bit regional code (RR), a 24-bit manufacturer code, and a 24-bit manufacturer-assigned serial number. | String |
| model | model | Model of the device. | String |
| osVersion | osVersion | OS Version of the device. | String |
| serialNumber | serialNumber | Serial number of the device. Applies to iOS Devices only. | String |

| Role Mapping Attribute Name | Microsoft Intune Attribute Name | Description | Data Type |
|---|---|---|---|
| UDID | udid | The device unique identifier.<br>Unique Device Identifier (UDID), which is a sequence of 40 letters and numbers that is specific to iOS devices. | String |
| UUID | uuid | Universal unique device identifier. | String |

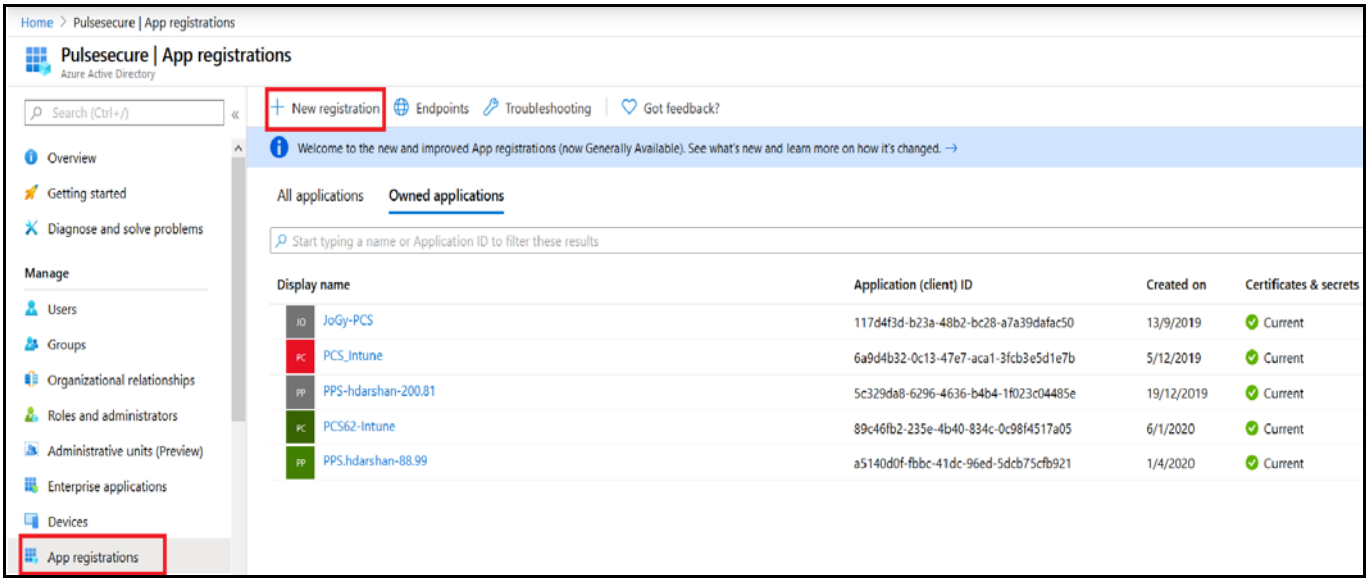
## Configuring the Microsoft Intune MDM Server

Microsoft Intune acts as the Mobile Device Management (MDM) Server for PCS solution. PCS users have to register their mobile devices with Microsoft Intune. As part of registration, the relevant Profiles get automatically provisioned to mobile device.

To configure the Microsoft Intune MDM:

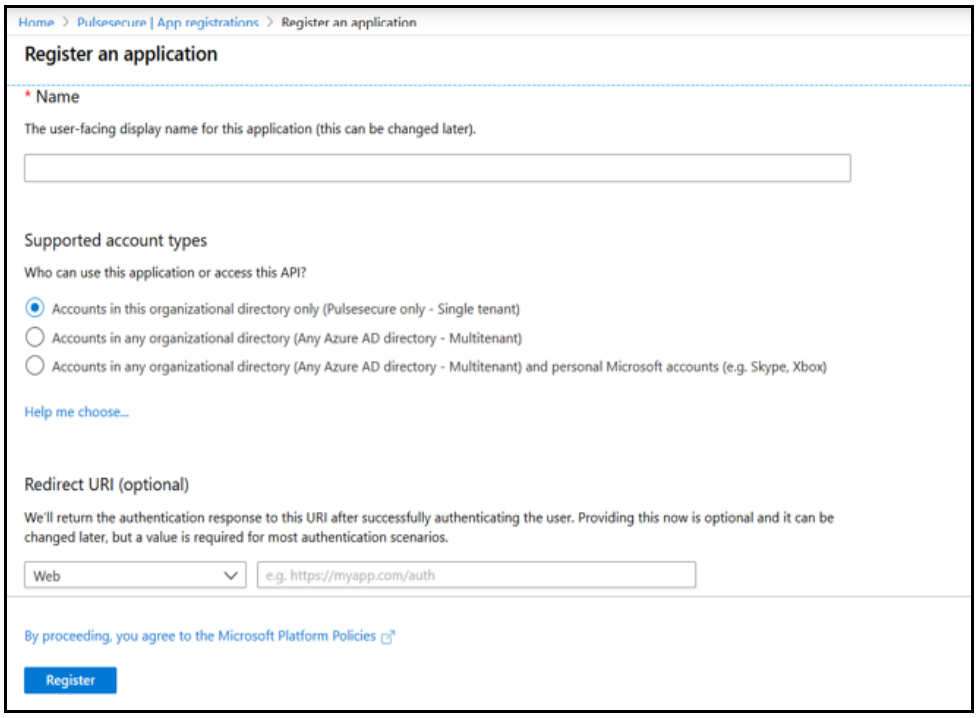
1. Enroll the devices with the MDM server.
2. Create Azure Active Directory (AAD) web application.
3. Go to **portal.azure.com**, click on the **Azure Active Directory** on the left of the screen, click on the **App registrations** and then click on **New registration**.

   Figure 13 Creating New Application



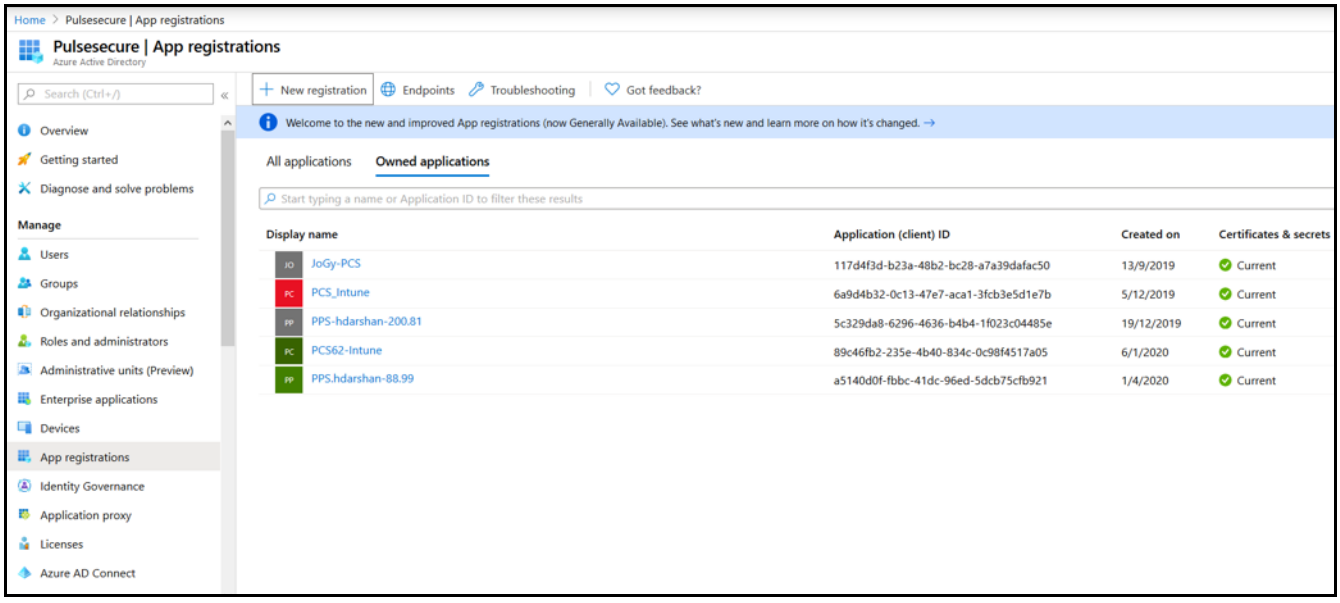

4. Enter the application name, select **Web** app as application type, and enter the IP address/FQDN for redirect-URL and click **Register**.

Figure 14 Registering a New Application



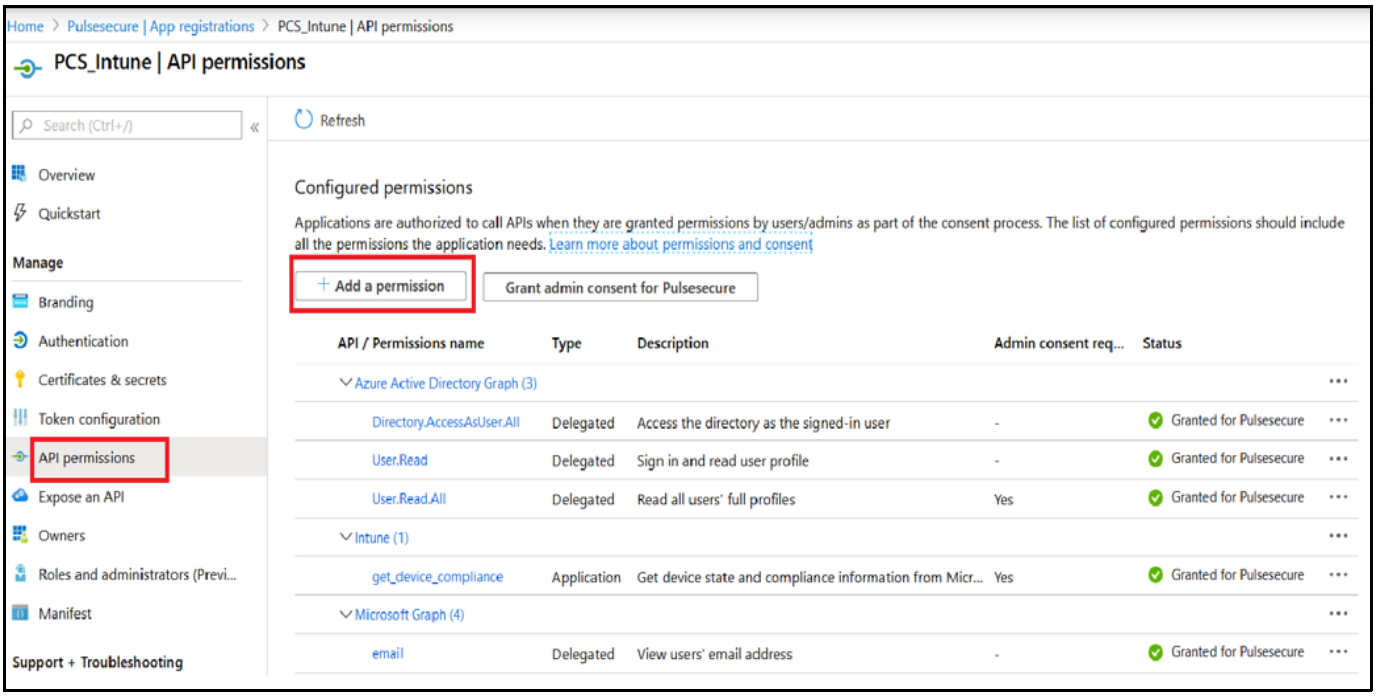

The Application Registration page appears if the registration is successful.

Figure 15 Application Created



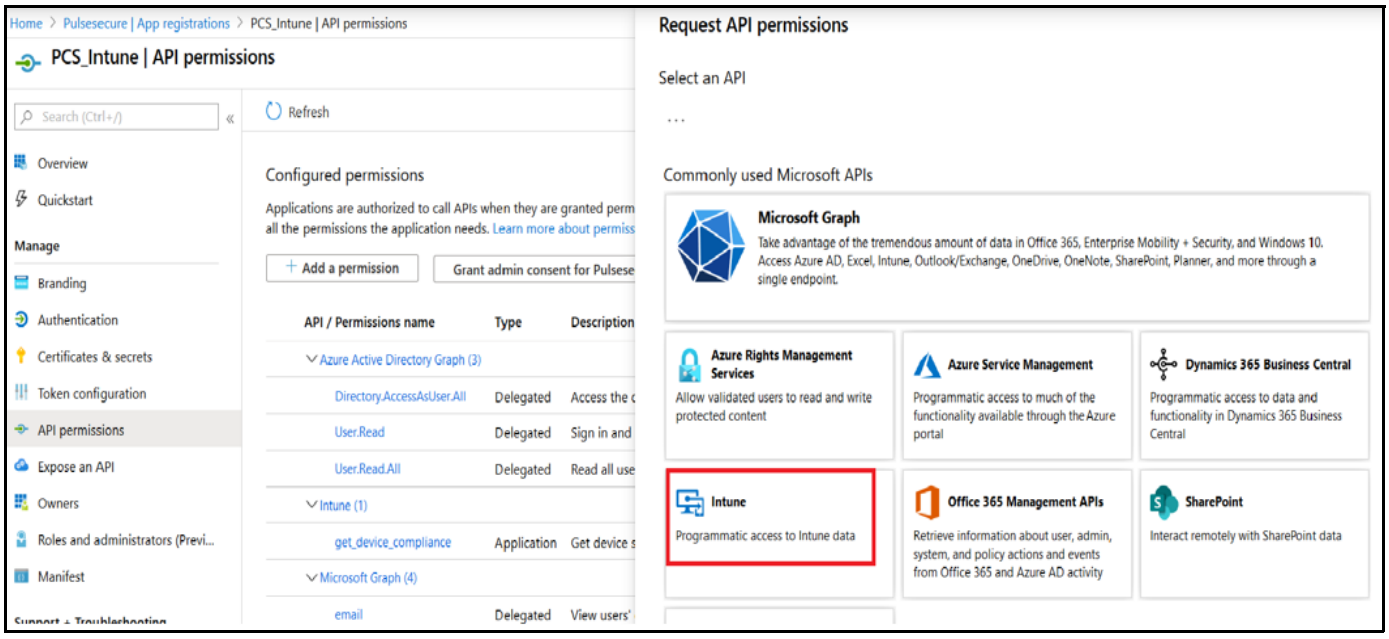

5. Click the application, then select **API permissions** and click **Add a permission**.

Figure 16 Adding Permissions



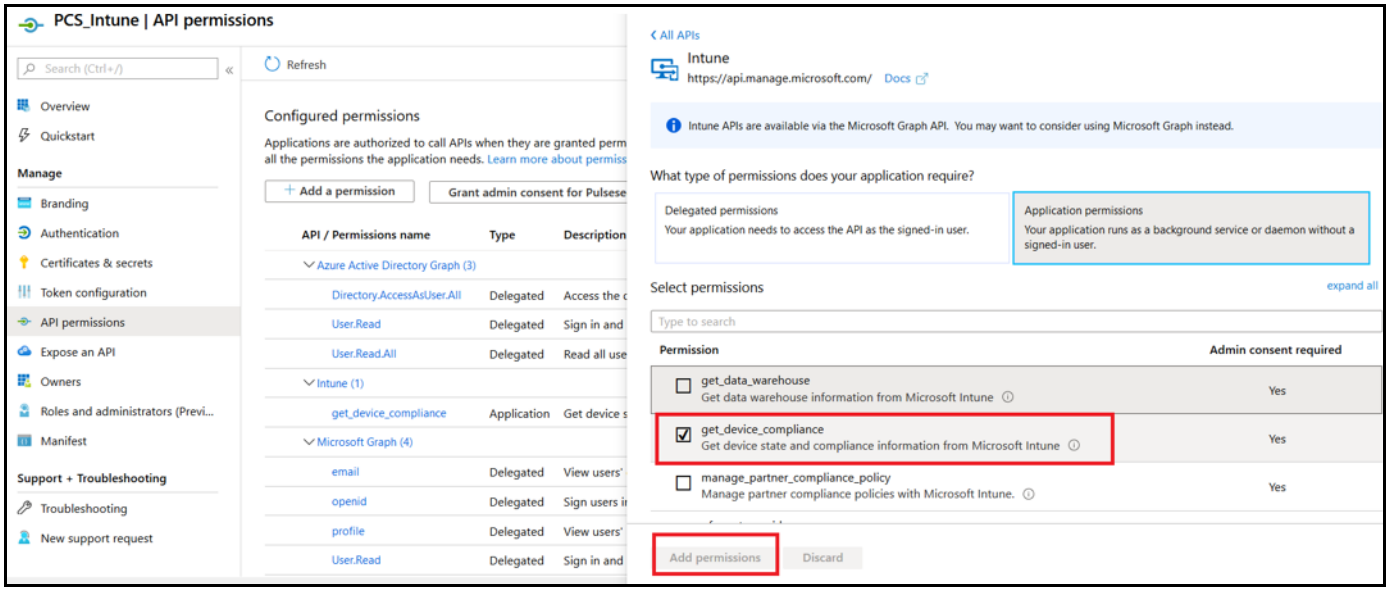

6. Select **Microsoft Intune API.**

Figure 17 Setting Intune Permissions



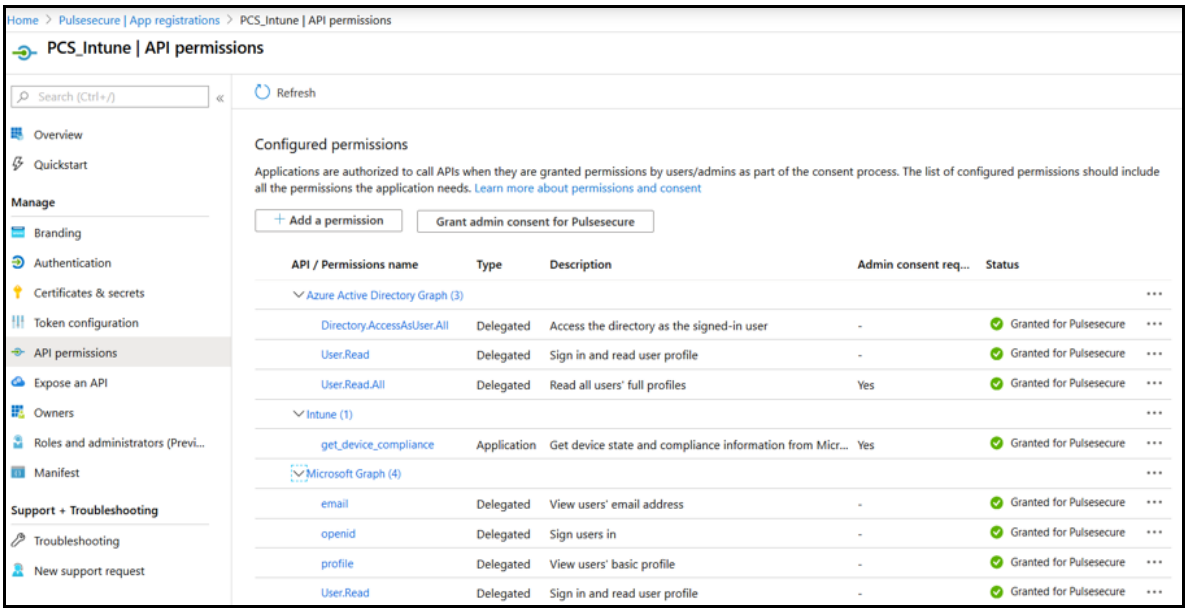

7. Under Application Permissions, select **Get device compliance** information from Microsoft Intune and click **Add permissions**.

Figure 18 Setting Intune Permissions




8. (Optional) You must add the following delegated permissions for Microsoft Graph API.
   - Sign in and read user profile
   - Sign Users in
   - View users' email address
   - View users' basic profile
9. (Optional) Add the following delegated permissions for Azure Active Directory.
   - Sign in and read user profile
   - Read all users' basic profiles
   - Access the directory as the signed-in user.

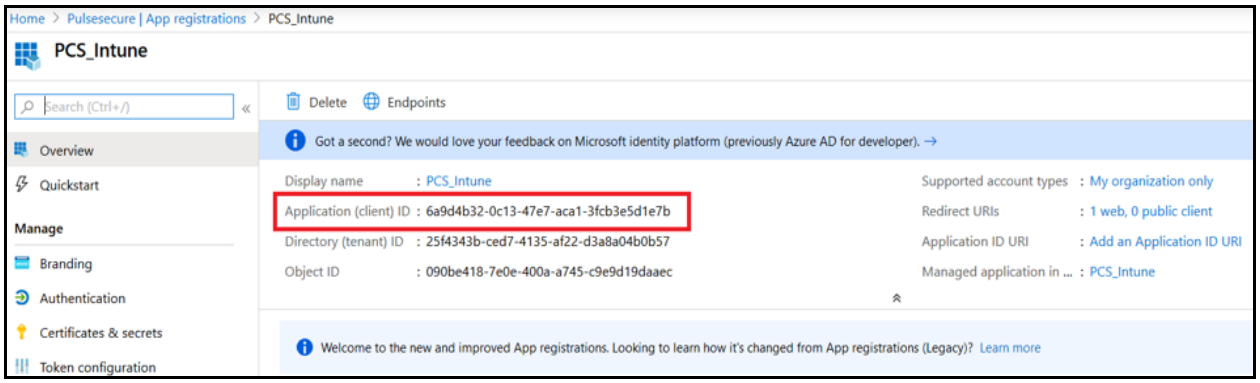
Figure 19 Permissions




## Viewing Client ID, Tenant ID, and Client Secret

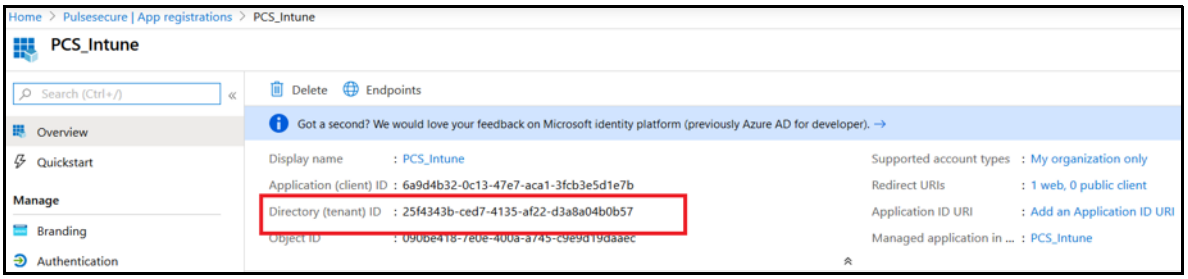
The Client ID/Application ID is created automatically once the AAD web application/API is created. You can view the client ID/application ID from the application properties page.

Figure 20 Client ID/Application ID



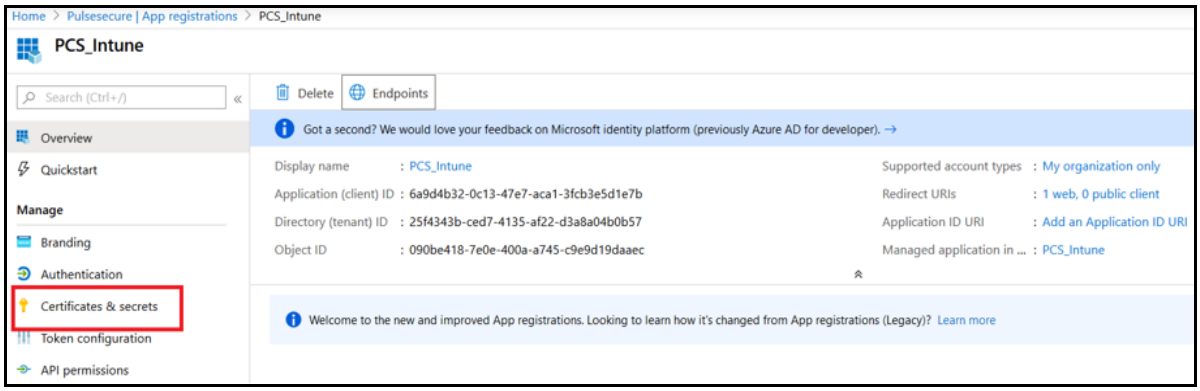

Every organization in Microsoft cloud is called tenant and it is organization specific. Each Tenant will be having a unique Tenant ID. Select the web application/API and then you can copy the tenant ID.

Figure 21 Tenant ID

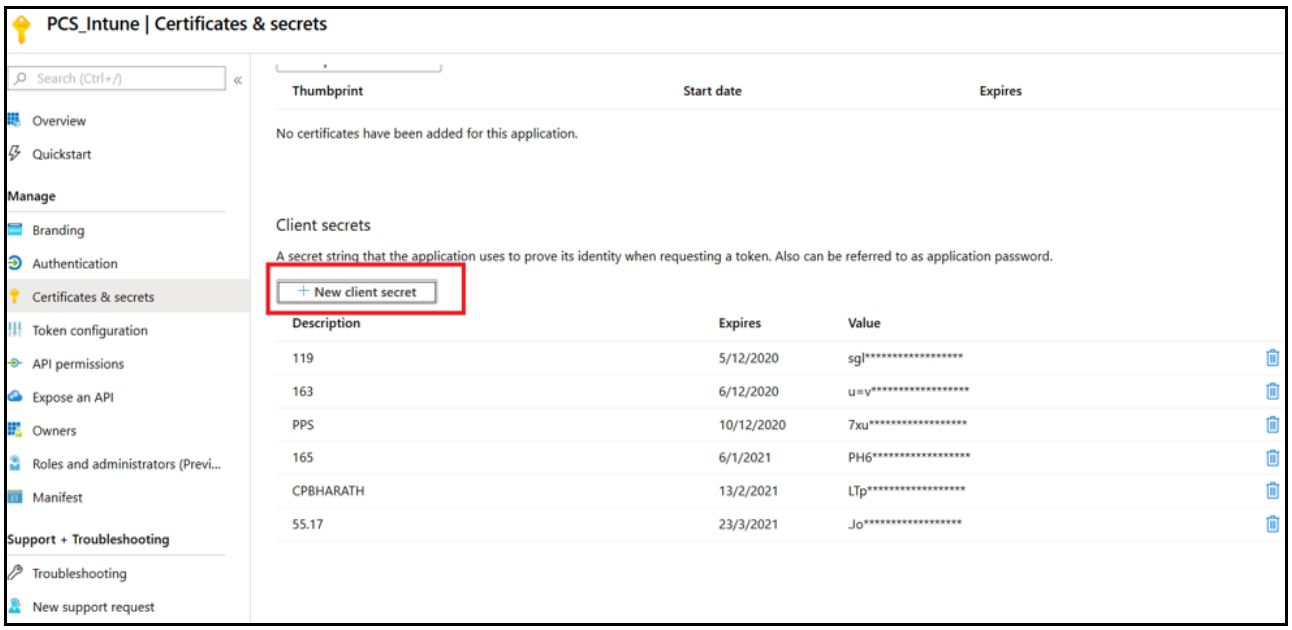


To create the secret key:

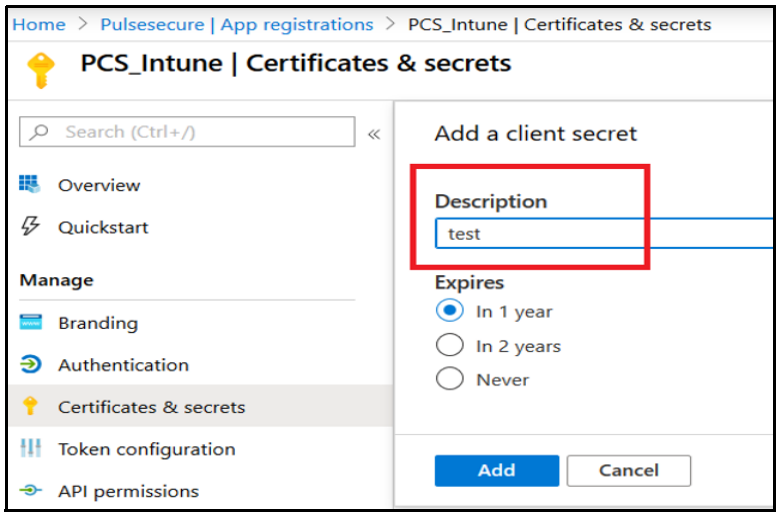
1. Click the **Web Application/API** and then click **Certificates & Secrets**.



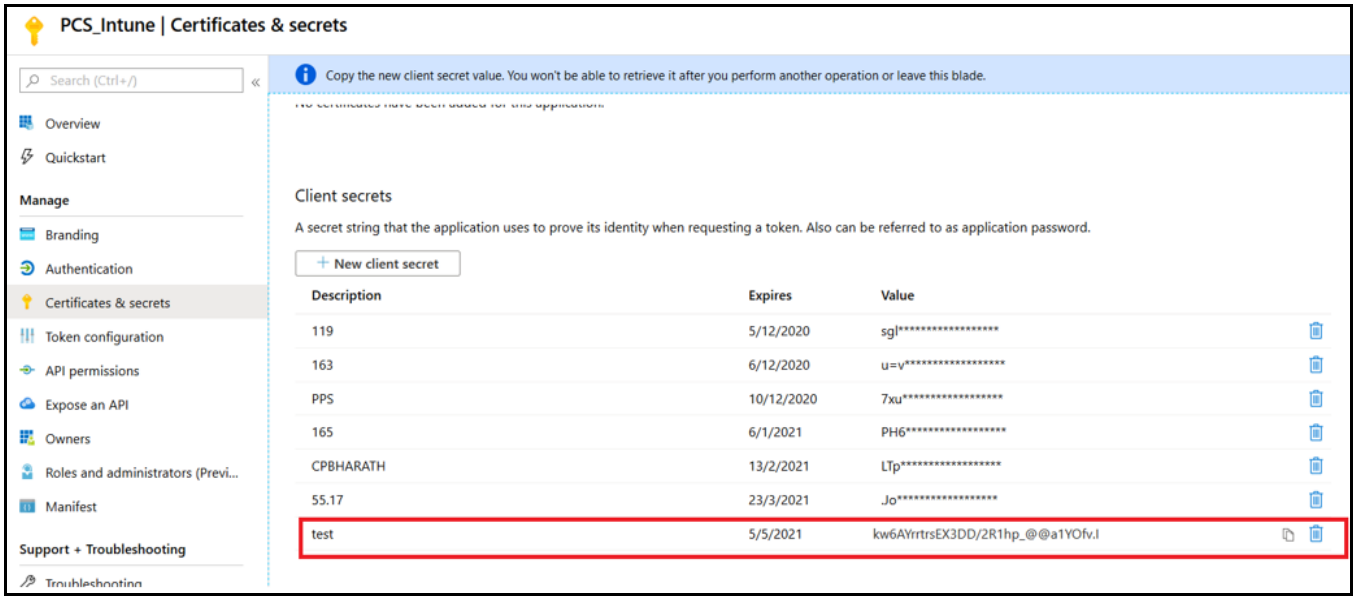

2. Click on **New client secret**.




3. Enter appropriate description and click **Add**.




The client secret is created.



# Using Logs to Verify Proper Configuration

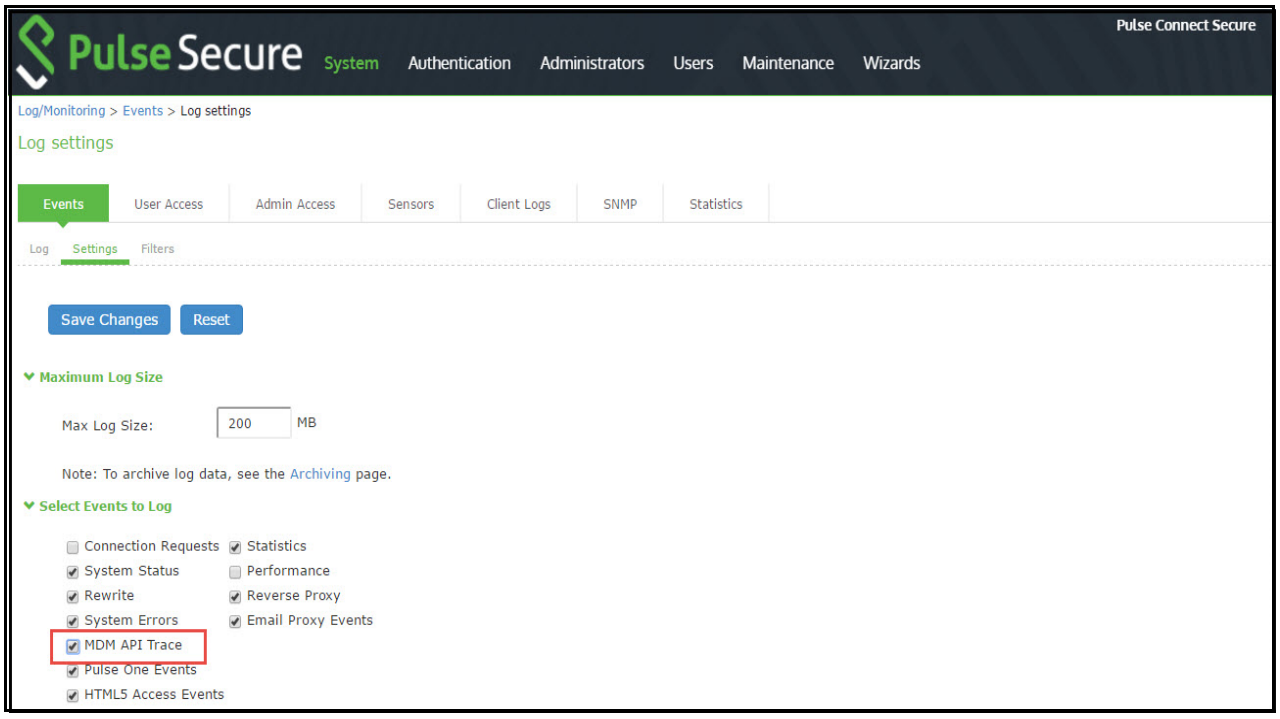
During initial configuration, enable event logs for MDM API calls. You can use these logs to verify proper configuration. After you have verified proper configuration, you can disable logging for these events. Then, enable only for troubleshooting.

To enable logging for MDM API calls:

1. Select **System Log/Monitoring**.
2. Click the **Events** tab.
3. Click the **Settings** tab to display the configuration page.

   Figure 22 shows the configuration page for Pulse Connect Secure.
4. Enable logging for MDM API events and save the configuration.

Figure 22 Events Log Settings Configuration Page - Pulse Connect Secure



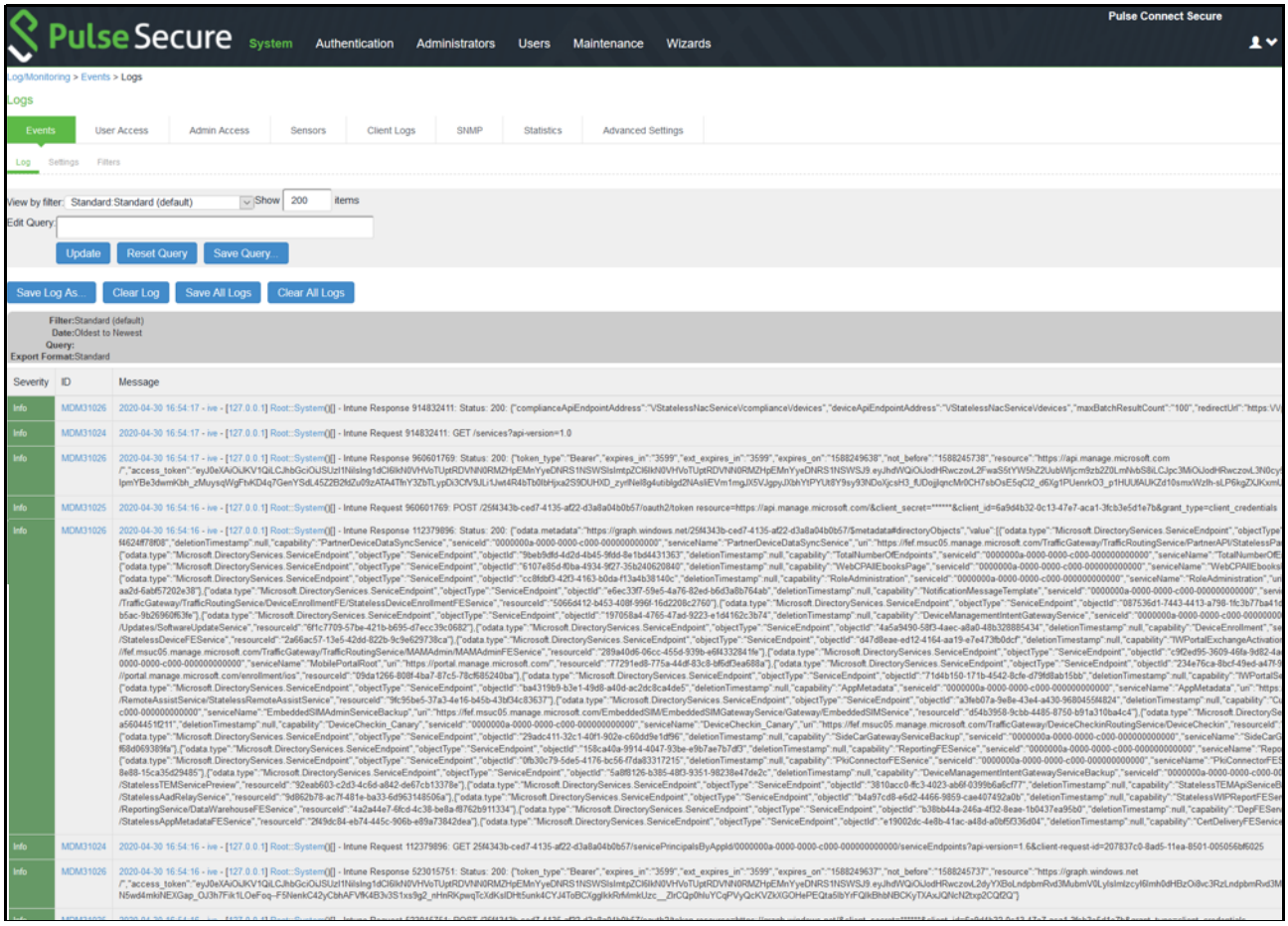

After you have completed the MDM server configuration, you can view system event logs to verify that the polling is occurring.

To display the Events log:

1. Select **System Log/Monitoring**.
2. Click the **Events** tab.
3. Click the **Log tab**.

Figure 24 shows the Events log for Pulse Connect Secure.

Figure 23 Events Log - Pulse Connect Secure



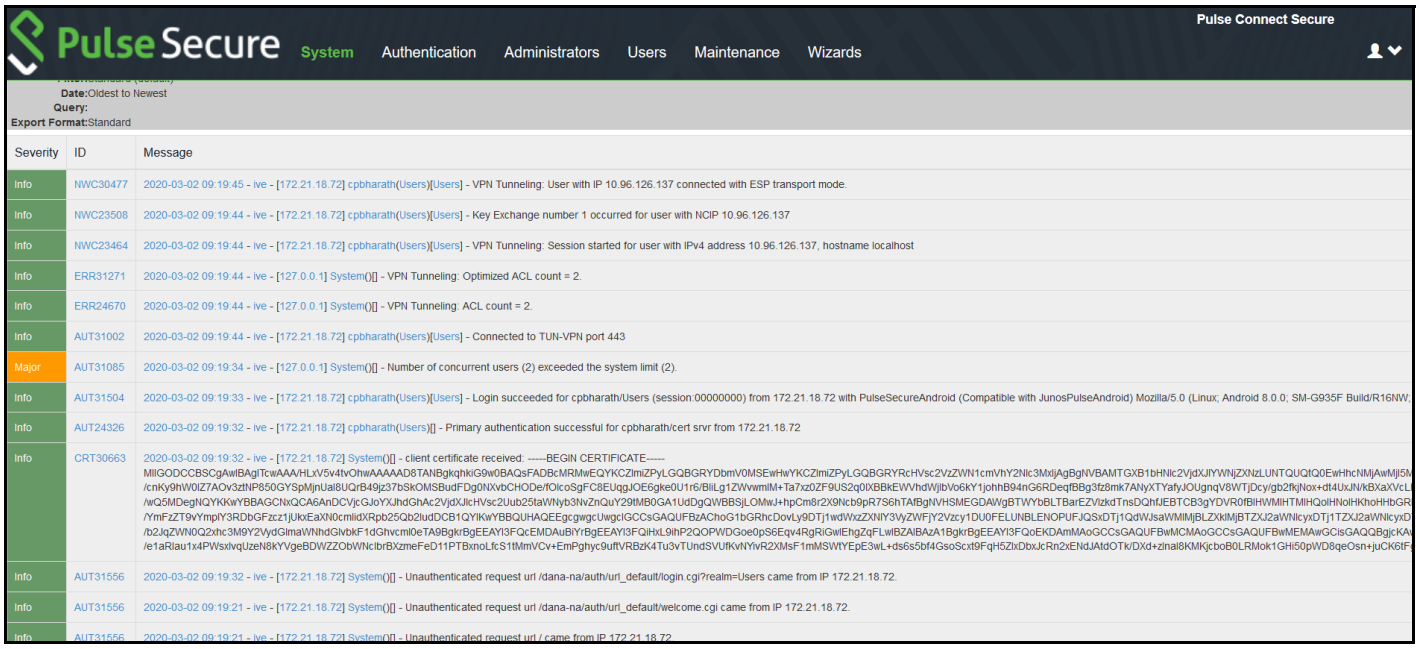

Next, to verify user access, you can attempt to connect to a wireless access point with your smart phone, and then view the user access logs.

To display the User Access log:

1. Select **System Log/Monitoring.**
2. Click the **User Access** tab.
3. Click the **Log** tab.

Figure 24 shows the User Access log for Pulse Connect Secure.

Figure 24 User Access Log




# Using Policy Tracing and Debug Logs

This topic describes the troubleshooting tools available to diagnose issues. It includes the following information:


- "Using Policy Tracing to Troubleshoot Access Issues" on page 29
- "Using the Debug Log" on page 31


## Using Policy Tracing to Troubleshoot Access Issues

It is common to encounter a situation where the system denies a user access to the network or to resources, and the user logs a trouble ticket. You can use the policy tracing utility and log to determine whether the system is working as expected and properly restricting access, or whether the user configuration or policy configuration needs to be updated to enable access in the user's case.

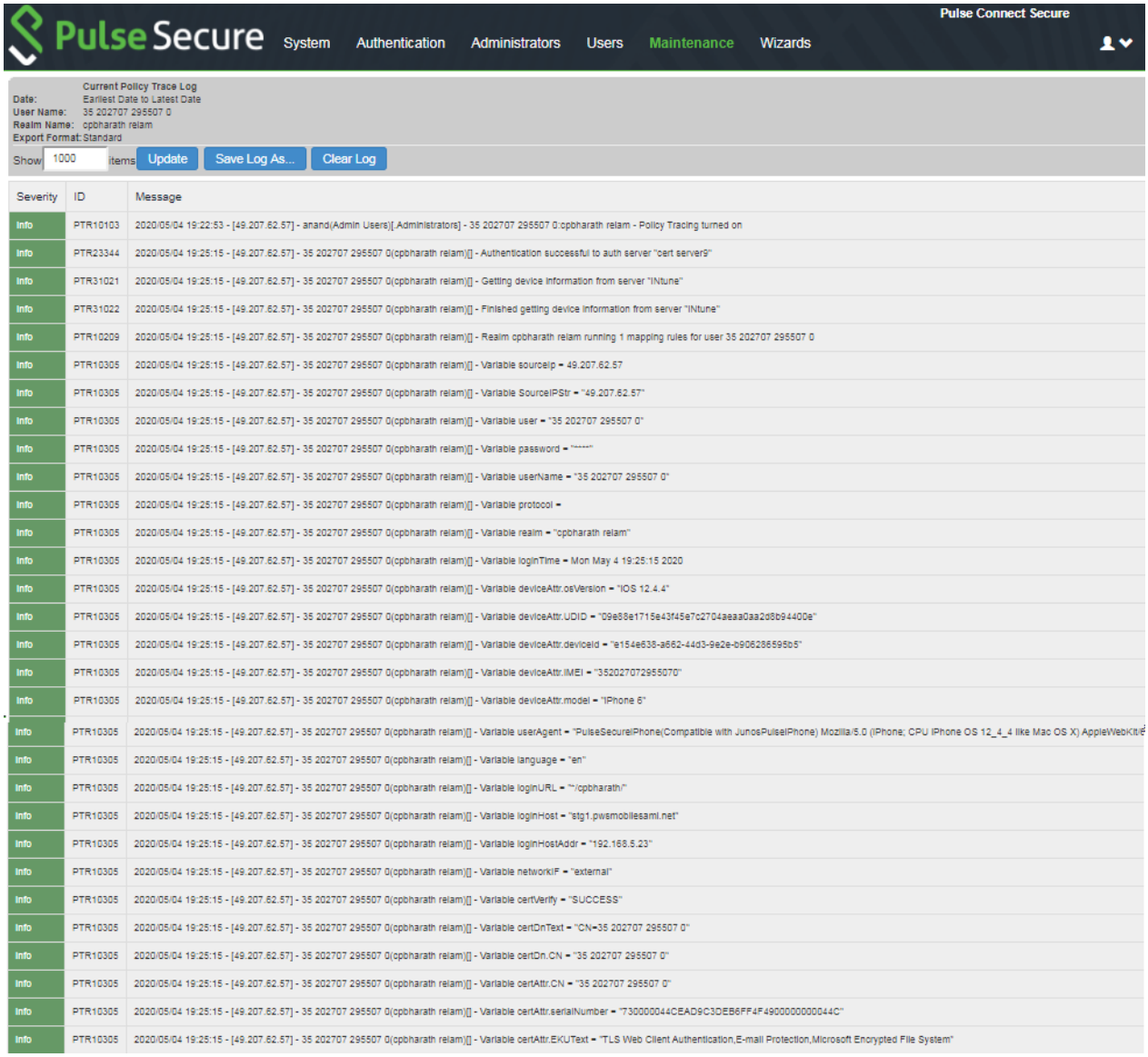
To create a policy trace log:

1. Select **Troubleshooting > User Sessions > Policy Tracing** to display the configuration page.
2. Specify the username, realm, and source IP address if you know it. If you provide the source IP address, the policy trace log can include events that occur before the user ID is entered into the system.
3. Select the events to trace, typically all but **Host Enforcer and IF-MAP**, unless you have enabled those features.
4. Click **Start Recording**.
5. Initiate the action you want to trace, such as a user sign in.
6. Click **View Log** to display the policy trace results log.

7. Click **Stop Recording** when you have enough information.

Figure 25 shows policy trace results.

Figure 25 Policy Tracing Results



| | | |
|---|---|---|
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certAttr.EKUOID = "1.3.6.1.5.5.7.3.2,1.3.6.1.5.5.7.3.4,1.3.6.1.4.1.311.10.3.4" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certAttr.altName.UPN = "cpbharath@securepulse.onmicrosoft.com" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certAttr.altName.UPNuid = "cpbharath" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certAttr.altName.UPNdomain = "securepulse.onmicrosoft.com" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certAttr.crlDist = " Full Name: URI:ldap:///CN=pulsesecureaccess-CSAD-CA,CN=csad,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN= |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certAttr.publicKey = "MIIBCgKCAQEAlSYTQbWl8OpVecxbo5ufT9UKQOl0qT/FFU5p87q5+CpDMbUV7Z+kensyvylZQvvGPTJoRoUgpl6l5PTX1AFvxXaGS9k9AlqOa7HFDjGd3y1U3PYjSjbjdlEDR0zg6hG2rIuY0dGVmma8SRG267B7PtspdU5rtnltrcPFF2h8lqrz |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certIssuerDnText = "CN=pulsesecureaccess-CSAD-CA, DC=pulsesecureaccess, DC=net" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certIssuerDn.CN = "pulsesecureaccess-CSAD-CA" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certIssuerDn.DC = "pulsesecureaccess" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable certIssuerDn.DC = "net" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable deviceAttr@INtune.UDID = "09e88e1715e43f45e7c2704aeaa0aa2d8b94400e" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable deviceAttr@INtune.deviceId = "e154e638-a662-44d3-9e2e-b906286595b5" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable deviceAttr@INtune.IMEI = "352027072955070" |
| Info | PTR10305 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Variable deviceAttr@INtune.model = "iPhone 6" |
| Info | PTR10212 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Mapped to roles cpbharath role by rule 'user = "*"' |
| Info | PTR10205 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Realm cpbharath relam mapped user 35 202707 295507 0 to roles cpbharath role |
| Info | PTR23353 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[] - Role restrictions successfully passed for roles: cpbharath role |
| Info | PTR23362 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[cpbharath role] - Sign-in successful, creating session |
| Info | PTR23363 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[cpbharath role] - Session created, redirecting user to start page. Sign-in done. |
| Info | PTR24559 | 2020/05/04 19:25:15 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[cpbharath role] - Automatically redirected from page "login" to the next start page "/dana/home/starter0.cgi?check=yes" before starting the session. |
| Info | PTR23471 | 2020/05/04 19:25:18 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[cpbharath role] - VPN Tunneling: IP Address Pools obtained for the current session are 10.96.18.16 |
| Info | PTR24639 | 2020/05/04 19:25:18 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[cpbharath role] - VPN Tunneling: ACL rule [1] resource = "*", action = ACCEPT |
| Info | PTR23468 | 2020/05/04 19:25:18 - [49.207.62.57] - 35 202707 295507 0(cpbharath relam)[cpbharath role] - VPN Tunneling: Session started with IP 10.96.18.13, hostname Bharaths-iPhone |
| Info | PTR10104 | 2020/05/04 19:25:38 - [49.207.62.57] - anand(Admin Users)[.Administrators] - 35 202707 295507 0:cpbharath relam - Policy Tracing turned off |

## Using the Debug Log

The Pulse Secure Global Support Center (PSGSC) might direct you to create a debug log to assist them in helping you debug an issue with the system. The debug log is used only by Pulse Secure Global Support Center.

In 9.1R3 release, the last-hit timestamp is included in each debug log statement. This timestamp helps the support in debugging and correlating timings of certain critical logs in some events.

To use debug logging:

1. Select **Troubleshooting > Monitoring > Debug Log** to display the configuration page.

   Figure 26 shows the configuration page for Pulse Connect Secure.

2. Complete the configuration as described in Table 8.

3. Click **Save Changes**. When you save changes with Debug Logging On selected, the system begins generating debug log entries.

4. Initiate the action you want to debug, such as a user sign in. You can reset the debug log file to restart debug logging if it takes you too long to initiate the action.

5. Click **Save Debug Log** to save the debug log to a file that you can send to Pulse Secure Global Support Center. You can clear the log after you have saved it to a file.

6. Clear the **Debug Logging On** check box and click **Save Changes** to turn off debug logging.

Figure 26 Debug Logging Configuration Page

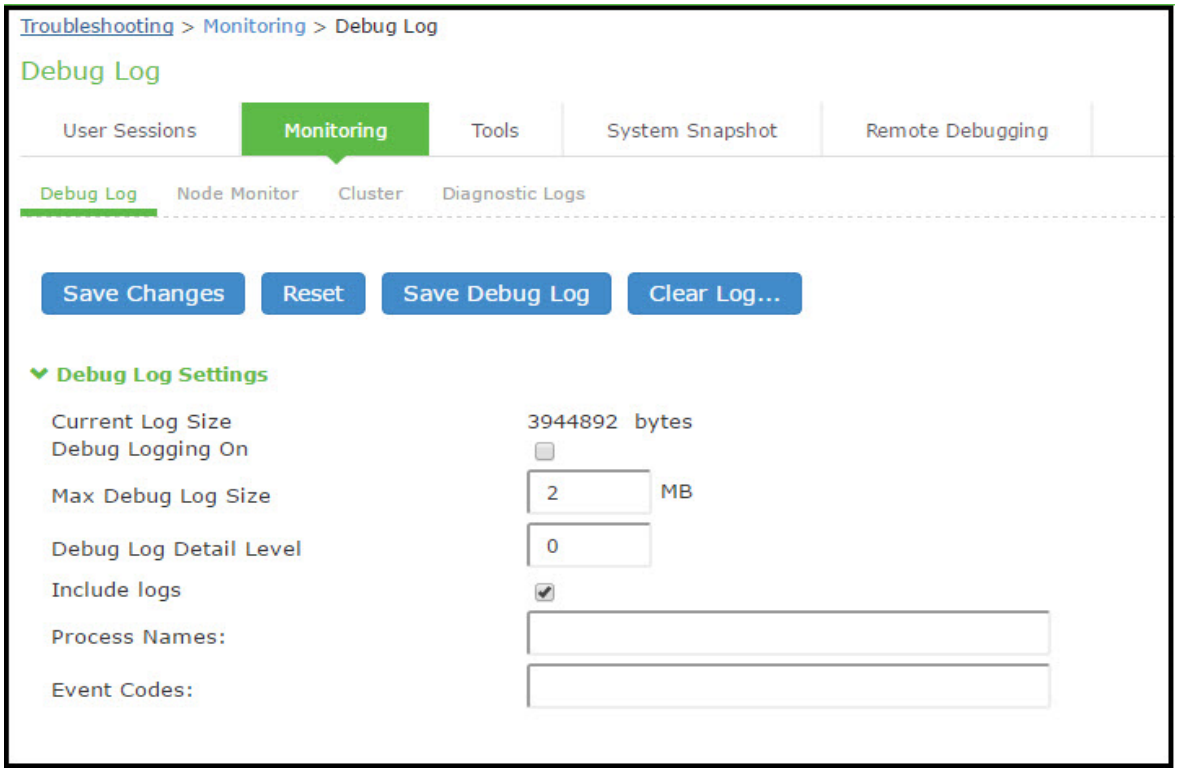



Table 8 Debug Log Configuration Guidelines

| Settings | Guidelines |
|---|---|
| Current Log Size | Displays the size of the current log file. If it is large, use the controls to save, reset, or clear the log file. |
| Debug Logging On | Select to turn on debug logging. |
| Debug Log Size | Specify a maximum debug logfile size. The default is 2 MB. The maximum is 250 MB. |
| Debug Log Detail Level | Specify the debug log detail level. Obtain this from Pulse Secure Global Support Center. |
| Include logs | Select this option to include system logs in the debug log file. Recommended. |
| Process Names | Specify the process name. Obtain this from Pulse Secure Global Support Center. |
| Event Codes | Specify the event code. Obtain this from Pulse Secure Global Support Center. For MDM integration issues, Pulse Secure Global Support Center typically likes to collect debugging information for codes MDM, Auth, agentman, and Realm. The text is not case sensitive. |