a Hewlett Packard Enterprise company

ClearPass 6.x

Tech Note: ClearPass

Enterprise Mobility Management Integration

November 2015 V5

Version	Date	Modified By	Comments
1.0	April 2013	Danny Jump	Initial Published Version
2.0	Oct 2013	Danny Jump	Add XenMobile support and minor Updates
3.0	Oct 2014	Danny Jump	Added SAP Afaria, a new section on SCEP Enrollment for MobileIron/Airwatch and 'other' minor updates
4.0	Sept 2015	Danny Jump	Added details of BlackBerry BES10 server integration and minor updates for CPPM 6.5
5.0	Nov 2015	Danny Jump	Added integration with Globo "Go! Enterprise MDM"

Table of Contents

Table of Figures

Introduction

With the release of ClearPass Policy Manager 6.4.0 in August 2014 we have continued to build on our previous industry leading functionality, now with the CPPM 6.4.0 release we maintain and extend our technology lead with the major Enterprise Mobility Management (EMM) platforms, allowing Aruba ClearPass customers to extend the knowledge of managed device state (device type, policy compliance etc.) down to the business rules that govern their corporate network admission policies. With the release of CPPM 6.4.0 we added support for SAP's Afaria.

For example, if the EMM platform detects that a device is jailbroken, the EMM platform only has the option to attempt to enforce the business policy at the device level. By extending this policy state to ClearPass as the network policy definition point, the jailbreak status of a device can be used to deny access or quarantine this device the next time it attempts to connect to the secure network.

This walkthrough explains the details of the current integration, the configuration steps to establish the API relationship between ClearPass Policy Manager and the customer's chosen EMM platform and finally, the expected device inventory and policy compliance data expected from each EMM vendor.

Some familiarity with Mobile Device Management concepts and the use of ClearPass Policy Manager's network enforcement methodology are assumed throughout this Tech Note.

EMM Integration

ClearPass Policy Manager has an extensible database for tracking devices attempting to connect to secure corporate networks. The devices are stored within the Endpoints table that is indexed on a unique identifier for each device, its MAC Address.

Typically the Endpoints table stores only basic information about the device collected from RADIUS authentication transactions associated with its usage on the wired or wireless access networks as shown below on the right-hand-side. In-depth information relating to the device posture is not available as no Attribute data is shown and only basic information can be extracted such as the OUI (first 3-bytes of the mac) to identify the manufacturer.

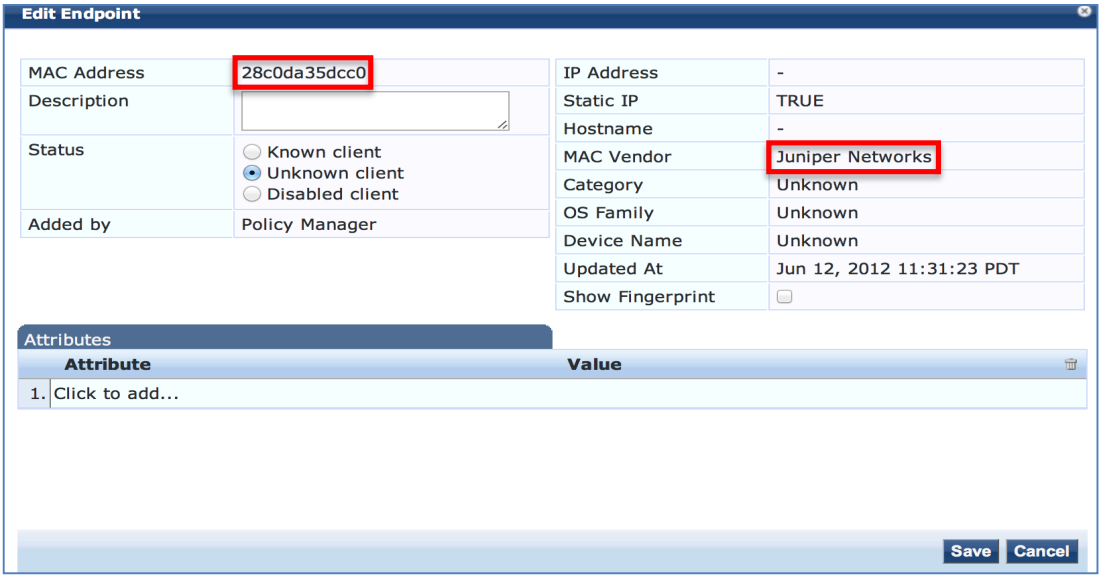


Figure 1 - Basic endpoint smart-device information

The knowledge about this device is increased through the use of ClearPass Policy Manager's built-in device profiling capabilities by monitoring traffic patterns from the device as it attempts to connect to the network. Information extracted from DHCP, HTTP packets and other sources of context can help provide additional details about the manufacturer and class of device.

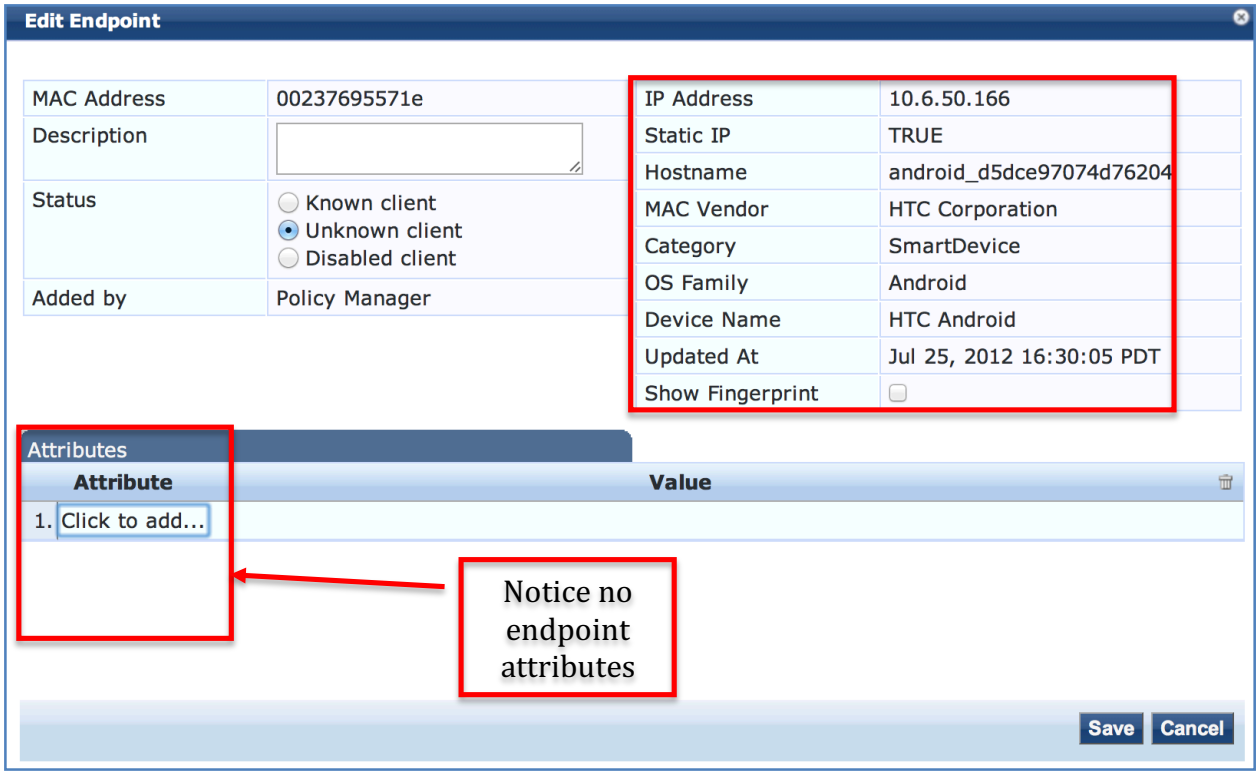


Figure 2 - Additional endpoint information retrieved thru DHCP fingerprinting

However, many customers have invested in EMM platforms to help them manage large rollouts of corporate issued smartphones or tablets. These EMM deployments can hold additional information about the device policy state that cannot be retrieved by passively monitoring network traffic as it enters the corporate network.

The EMM integration as part of ClearPass Policy Manager 6.0.2 and above leverages the extensive dataset that each entry in the Endpoints table can hold, by adding a set of EMM normalized data tags. These additional data tags can then be referenced in enforcement policies to implement various business rules based on the device state information received from the EMM platforms.

Below we show an example of the additional attributes that can be integrated into the ClearPass Endpoint profiler database that could be received from an EMM vendor. Not all EMM vendors expose the same level of data, we normalize the information received and present it in a standard attribute template.

Edit Endpoint

Edit Endpoint

MAC Address	00263795c3bb	IP Address	-
Description		Static IP	TRUE
Status	(•) Known client ( ) Unknown client ( ) Disabled client	Hostname	gvernot:Android 4.0.3:PDA
		MAC Vendor	Samsung Electro-Mechanics
		Category	SmartDevice
Added by	mobileironadmin	OS Family	Android
		Device Name	Samsung-GT-I9000
		Updated At	Dec 05, 2012 23:49:31 PST
		Show Fingerprint	☐

Attributes

	Attribute		Value
1.	Phone Number	=	PDA
2.	Source	=	MI
3.	MDM Identifier	=	776fccc4-de51-414f-a54f-8e45cac20b7c
4.	Display Name	=	Gabriel Vernot
5.	IMEI	=	351751041424147

Save Cancel

Figure 3 - Endpoint plus EMM attributes

Attributes

6.	Model	=	GT-I9000
7.	MDM Enabled	=	false
8.	Owner	=	gvernot
9.	OS Version	=	Android 4.0
10.	Last Check In	=	2012-04-10 08:33:36.0
11.	Carrier	=	PDA

Save Cancel

Figure 4 - More EMM attributes

Attributes

10.	Last Check In	=	2012-04-10 08:33:36.0
11.	Carrier	=	PDA
12.	Compromised	=	False
13.	Ownership	=	Employee
14.	Manufacturer	=	Samsung
15.	Click to add...		

Save Cancel

Figure 5 - Even more EMM attributes

Additionally, the ClearPass EMM integration updates the internal device Profile database with knowledge of the device type learned from the configured EMM platform. This valuable inventory data about the device manufacturer, its hardware platform type and software version are all recorded in the ClearPass Endpoint profiler database and provide the definitive knowledge of the device type that could not otherwise be collected from passive network monitoring. This level of detail is equivalent to the device information recovered from ClearPass' own Onboard device provisioning and OnGuard device posture assessment technologies.

Configuration of EMM Vendors

From the Administration menu of ClearPass Policy Manager, a new menu option has been added under External Servers called Endpoint Context Servers, under 6.0.2 (the initial release when we added EMM support) the menu looks like the below screen shot.

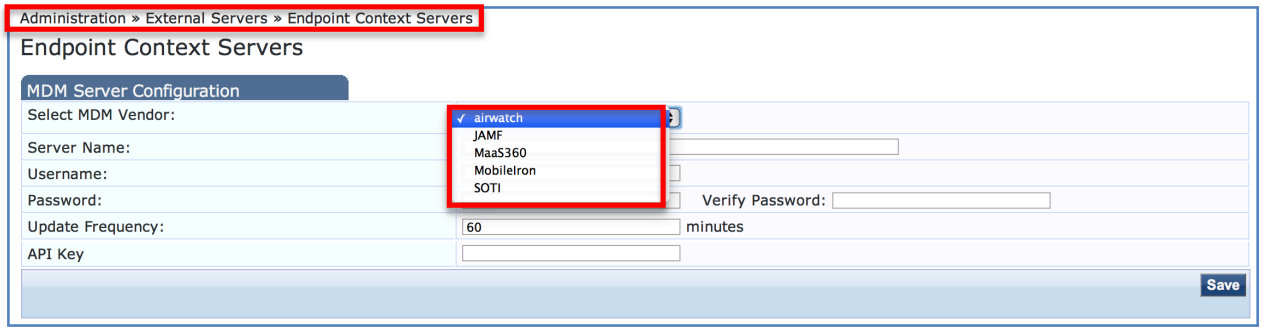


Figure 6 - Endpoint Context Server configuration for CPPM v6.0.2

Since CPPM 6.1 the configuration varies slightly, in that under the Endpoint Context Servers, you can now define **multiple** servers and these will now operate concurrently, i.e. multiple EMM vendors can be configured and CPPM will ingest data from more than one EMM vendor. As previously stated under 6.2 we added Citrix XenMobile, in 6.4.0 we added to the list of supported EMM vendors, SAP Afaria. The configuration under 6.1 & 6.2 requires that you use the menu option 'Add Context Server', under **Administration-> External Servers->Endpoint Context Servers** the pop-up box below show the current full list of supported vendors.

Figure 7 - Endpoint Context Server configuration for CPPM v6.4.0

Server Configuration varies slightly by vendor. But for all EMM partners some baseline parameter are required such as, Server Name, Server Base URL, User Name, and Password. Authentication is typically HTTPS authentication.

The differences that exist in addition to the baseline options discussed above are:

- AirWatch makes use of an API Key
- MaaS360 makes use of an Application Access Key, Application ID, Application Version, Platform ID and a Billing ID.
- SOTI (sometimes) makes use of a Group ID attribute

The details behind the above options are explained further in the section titled Supported EMM Vendors.

The Update Frequency configuration defines how often ClearPass Policy Manager will check in with the configured EMM platform to retrieve any new managed device details or update the status of an existing managed devices. Policy Manager is tracking the changes within the EMM device records and will only update an Endpoint record in the event of a change in device inventory or policy state.

Note: In v6.1.0, the Update Frequency option has been replaced by the cluster-wide service parameter "Endpoint Context Servers polling interval". Go to **Administration > Server Manager > Server Configuration** and click on **Cluster-Wide Parameters**.

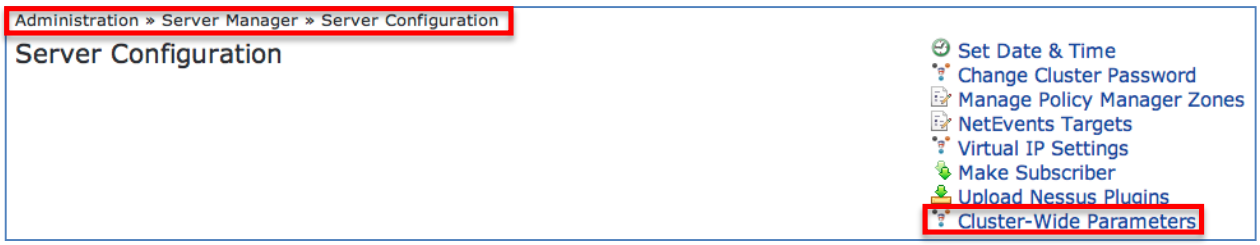


Figure 8 - Cluster-Wide Parameters

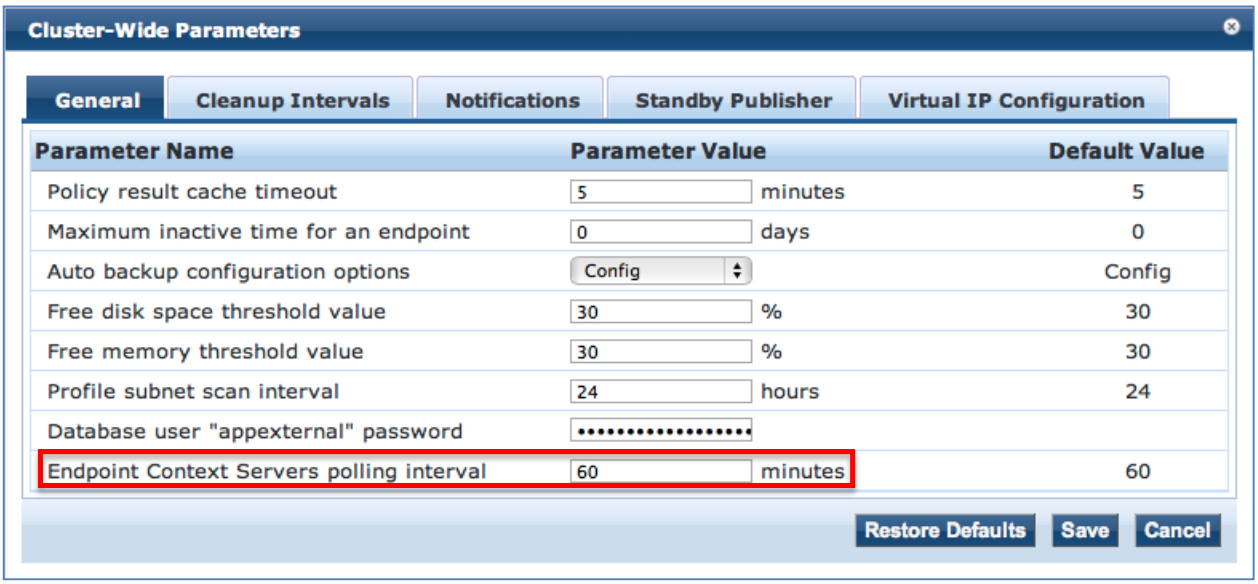


Figure 9 - Endpoint Context Servers polling interval – default 60 minutes

Typically the Update Frequency (Polling Interval) should be set relative to the device check-in interval configured on the EMM platform. Each EMM vendor will have a different default check-in interval used to get updated status information from the devices it has under management. This check-in interval is how often the EMM agent on the device itself connects back to the EMM server. For many platforms this could be 4 hours or longer, so there is no benefit to having an aggressive polling interval on ClearPass as the data returned will largely be the same.

Note: Customers are recommended to consult their chosen EMM vendor on the best update frequency based on their deployment configuration.

Normalized Dataset

ClearPass Policy Manager communicates with the configured EMM platform via their published API interface. Typically these are HTTP Based API's - Typically these API communications are defined using RESTful API calls returning XML or JSON output.. The ClearPass integration consumes these XML or JSON outputs, which are very specific to each EMM platform, and normalizes their output to a common set of Endpoint tags that can be added to the ClearPass database.

By normalizing the output, common and easy to understand enforcement policies can be created within ClearPass without the need for the administrator to understand the semantics of the EMM API interface.

The following table shows the currently available normalized data set implemented by the ClearPass EMM Integration. Not all of these attributes will be available consistently from each EMM platform or for each device type within a chosen EMM platform. For example, the Carrier attribute will not be available for a WiFi only tablet as it does not have a cellular chipset.

In the event that an attribute is not available from the configured EMM platform or not supported on the returned device type, the ClearPass Endpoints table will not contain a value for that normalized attribute.

Endpoint Tag	Tag Type	Comments
Manufacturer	Inventory	Manufacture name such as Apple, Samsung, etc. For Activate will always be “Aruba Networks”
Model	Inventory	Model name such as iPad, DROID X, etc. with extraneous sub-model info removed.
OS Version	Inventory	Version number such as iOS 6.1, Android 4.0, etc. Minor version numbers are removed so that 6.1.1 becomes 6.1.
UDID	Inventory	Device unique identifier
Serial Number	Inventory	Device serial number
IMEI	Inventory	Cellular only devices
Phone Number	Inventory	Cellular only devices
Carrier	Inventory	Cellular only devices
Owner	Inventory	Registered enterprise username
Display Name	Inventory	Full name of registered owner
Description	Inventory	Display a description of the device.
Source	Inventory	Display which EMM vendor supplied the device details
Ownership	Inventory	“Corporate” or “Employee”
EMM Identifier	Inventory	Internal identifier used by EMM API interface. This varies between EMM vendors.
Compromised	Policy	“True” or “False”. Jail broken device or Root-kit detected.
Encryption Enabled	Policy	Device level encryption status
Blacklisted App	Policy	“True” or “False”. A blacklisted app is installed on the device.
Required App	Policy	“True” or “False”. A required corporate app is missing from the device.
EMM Enabled	Policy	“True” or “False”. The device is under EMM management.
Last Check In	Policy	Last time the device last checked in to EMM server

Figure 10 - List of all possible normalized attributes

Using EMM Data for Network Enforcement

Once the EMM integration is configured and device data is being populated to the Endpoints and Profile databases within ClearPass, this information can be used to enforce various business rules on how these corporate managed devices are admitted on to the network.

Given EMM platforms are largely focused on smartphone and tablet devices, the network of interest is typically limited to WiFi connectivity. The following examples provide some guidance on how to leverage the EMM data to change the way these mobile devices are admitted onto a corporate WiFi network.

Endpoint Data

The data retrieved from the EMM platform and stored in the Endpoints table as additional tags contains both inventory data and policy state information. Therefore, an incredibly rich set of business rules can be enforced on the corporate network as it relates to the device type, ownership, compromised status, and the impact of Apps that are installed or missing, just to name a few. The following sample business rules included below illustrate how the EMM data included in the Endpoints table can be used to enforce network policy decisions and control the way these devices are admitted onto the network.

Jail broken or Rooted-Device Detected

A common used case of EMM platforms is to leverage the presence of the EMM agent (App) to attempt to detect if a device has been jail broken (Apple iOS devices) or a root-kit installed (Android). This status of the device being compromised is reported by the EMM agent back to the EMM server either during a regular check-in interval or as an alert message and will then be reflected in the ClearPass Endpoints table via the API integration.

A device being compromised will often result in the IT administrator being less trusting of the device and depending on the local security policy may result in a reduced level of network access or complete quarantining of the device. ClearPass' rich policy enforcement allows the administrator to chose how these compromised devices should be handled the next time the user attempts to connect to the enterprise network. The following enforcement policy example shows how the Endpoint *Compromised* data tag is being referenced whenever a device attempts to connect to the enterprise network.

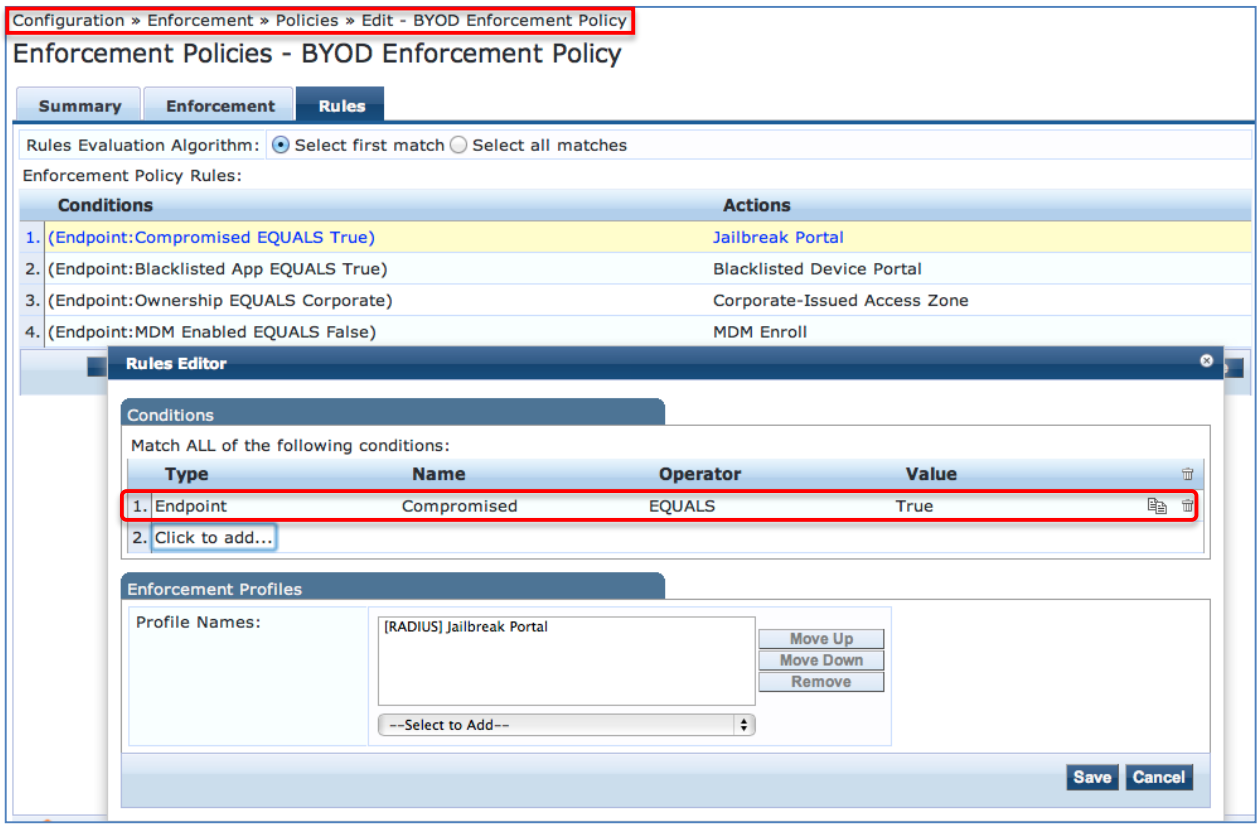


Figure 11 - Enforcement Policy – Endpoint compromised

In the event that this flag is set to True by the configured EMM platform, then the network enforcement profile applied will result in the device being placed in a quarantine state. This is achieved by ClearPass informing the Aruba controller to redirect the access attempt to a captive portal page informing the user of their breach of network access policy.

Configuration » Enforcement » Profiles » Edit Enforcement Profile - Jailbreak Portal

Enforcement Profiles - Jailbreak Portal

Summary | Profile | Attributes

Profile:

Name:	Jailbreak Portal
Description:	
Type:	RADIUS
Action:	Accept
Device Group List:	-

Attributes:

	Type	Name		Value
1.	Radius:Aruba	Aruba-User-Role	=	jailbreak-portal

Figure 12 - Enforcement Profile – redirect for Jailbreak/rooted devices

Figure 13 - Captive portal Jailbreak detection warning

Blacklisted App Detected

Several EMM platforms have the ability to build compliance policies around the Apps that have been installed on a smartphone or tablet. This is possible because the EMM platform will harvest the entire list of Apps that have been installed on the device and track them on an ongoing basis. This is a key reason why EMM is more appropriate for corporate issued devices where there are no privacy concerns about personally installed Apps that is often the case in a BYOD environment.

As part of the EMM compliance policy, a list of Blacklisted Apps can be defined and during a device check-in, if one of these Apps is installed on the device, the compliance state can be triggered. The EMM integration with ClearPass does not recover the details of the Apps installed on the device, but instead recognizes that the EMM has detected the presence of a Blacklisted App by the EMM internal compliance policy. This allows ClearPass to maintain its BYOD friendly approach, which is central to its Onboard provisioning solution, by avoiding any potential violation of the end user privacy through personal App visibility.

The following enforcement policy example shows how the Endpoint *Blacklist App* data tag is being referenced whenever a device attempts to connect to the enterprise network.

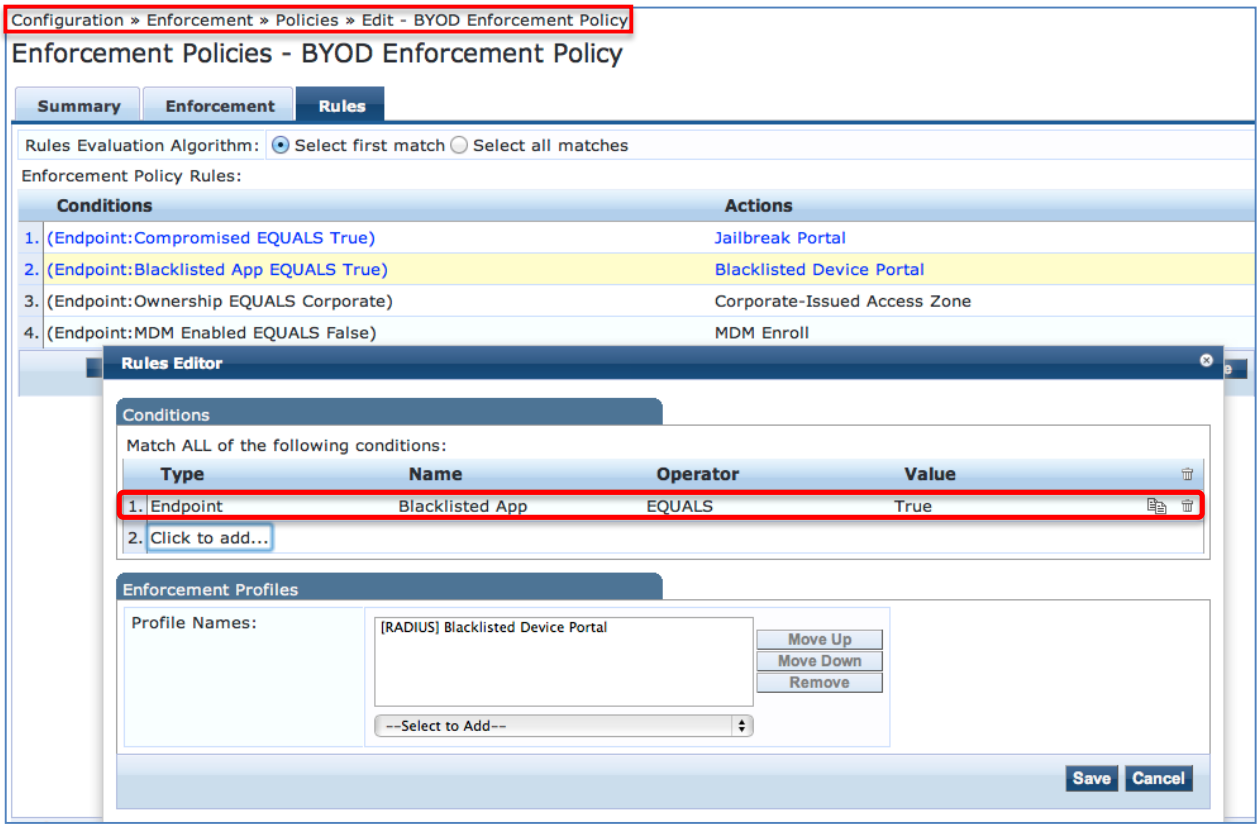


Figure 14 - Enforcement Policy – Blacklisted App

In the event that this flag is set to True by the configured EMM platform, then the network enforcement profile applied will result in the device being redirected to a Blacklisted App portal informing the user of their breach of network access policy. Optionally the device could be restricted network access, such as only to the Internet.

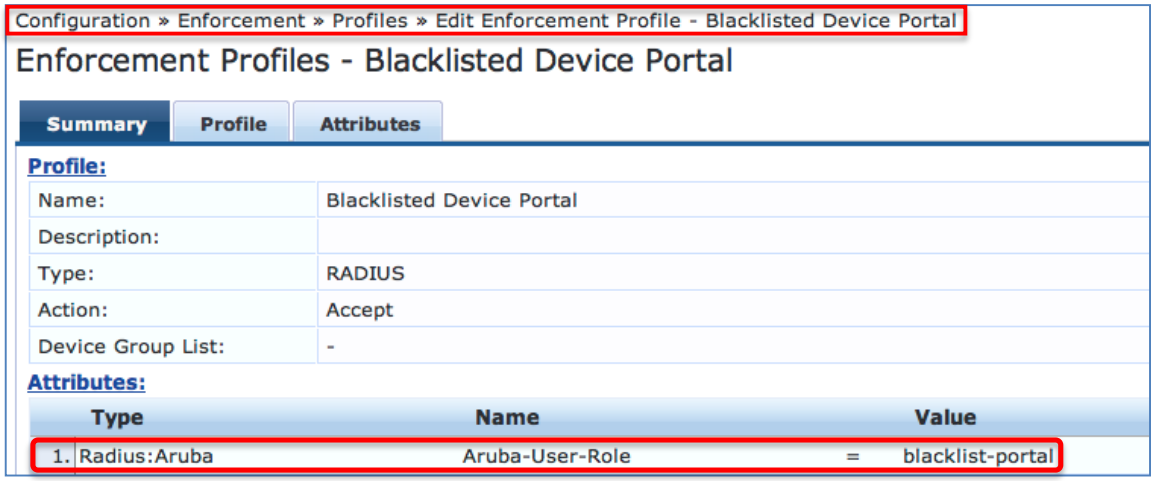


Figure 15 - Enforcement Profile – redirect for Blacklisted App

Corporate Issued vs. Employee Liable Device

Many EMM platforms allow the administrator to define the ownership type of each device taken under management. Corporate-owned or Employee-owned device types are tracked and can be reported to ClearPass via the API integration.

Some customers may wish to leverage this knowledge of corporate issued devices and an associated rollout of a corporate application to change the way that device accesses the network.

The following enforcement policy example shows how the Endpoint *Ownership* data tag is being referenced whenever a device attempts to connect to the enterprise network.

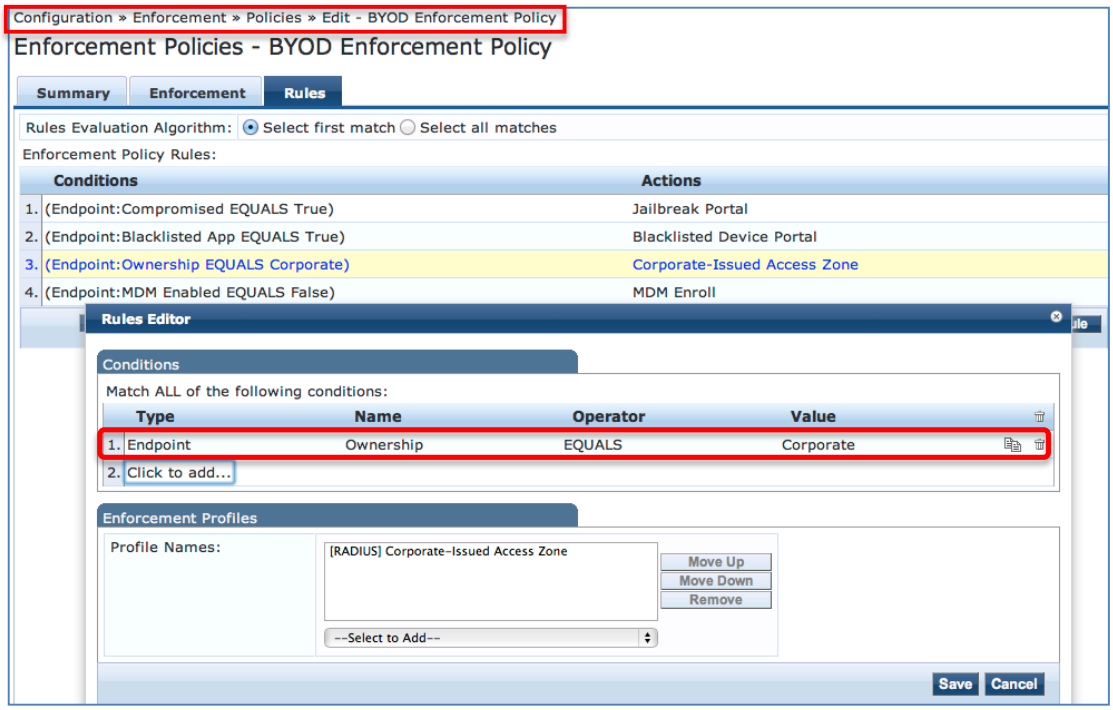


Figure 16 - Enforcement Policy – Corporate device

In the event that this flag is set to Corporate by the configured EMM platform, then the network enforcement profile applied will result in the device being placed in the corporate access role which grants access to specific application servers and also enables a high level of Quality of Service (QoS) for these applications. Alternatively, if the flag is set to Employee, the network enforcement profile applied will restrict access to only essential internal resources and apply a best effort QoS profile for the user.

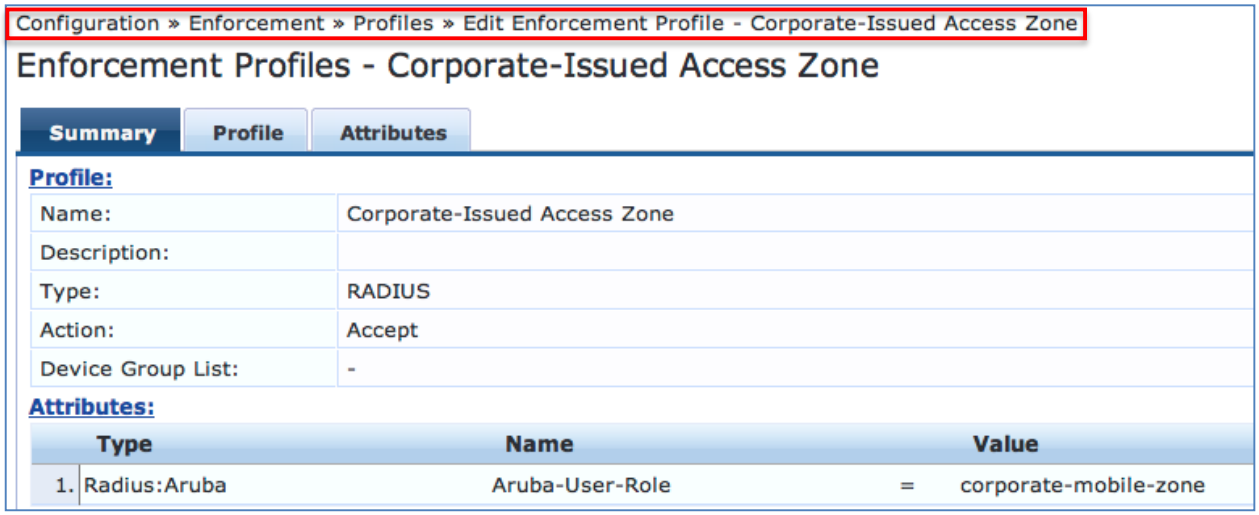


Figure 17 - Enforcement Profile – Corporate device

EMM Agent Removed

A common scenario in many EMM deployments occurs when a user either purposely (to avoid corporate monitoring) or by accident removes the EMM agent or profile from their device. This results in the device management communication channels being severed and the ability for the EMM platform to enforce policy to be become marginalized.

The ability for ClearPass to learn via the API integration that the device is no longer under management allows the administrator to differentiate this device the next time it attempts to access the enterprise network and redirect it back to the device management portal for re-provisioning.

The following enforcement policy example shows how the Endpoint *EMM Enabled* data tag is being referenced whenever a device attempts to connect to the enterprise network.

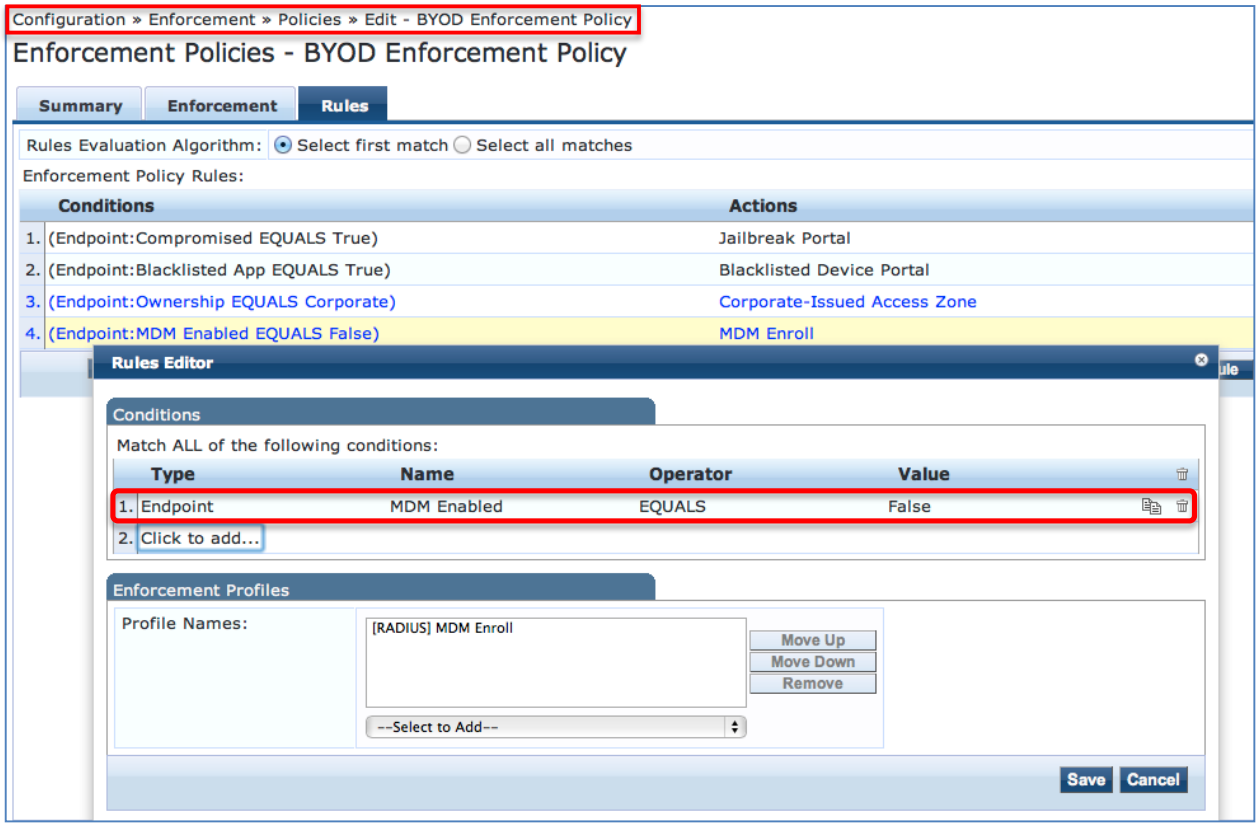


Figure 18 - BYOD enforcement – endpoint EMM managed

In the event that this flag is set to False by the configured EMM platform, then the network enforcement profile applied will result in the device being placed in a quarantine state and redirected to the device management provisioning page. This results in the user being forced to comply with the corporate policy and place the device back under management if they wish to access any corporate resources, without any manual intervention from the IT helpdesk staff.

Profile Data

The data retrieved from the EMM platform and stored in the Profile database table consists of a dataset made up of *Device Category*, *OS Family* and *Device Name* as shown in the screenshot below.

Hostname	AndroidSamsung 00086
MAC Vendor	Samsung Electronics
Category	SmartDevice
OS Family	Android
Device Name	Samsung-SGH-T679
Updated At	Dec 05, 2012 23:54:58 PST

Figure 19 - Profile database info

The inventory data available from the EMM platforms allows for explicit device type knowledge such as the Samsung device listed above. Alternatively, relying solely on passively collected network data would result in the device only being seen as a generic Android device manufactured by Samsung.

The following sample business rules included below illustrate how the EMM data included in the device Profile database can be used to enforce network policy decisions and control the way these devices are admitted onto the network.

iPad vs iPhone/iPod Network Access

Small screen devices are not always appropriate for the roll out of some corporate applications. For example, if a customer had deployed a corporate application that is designed to be accessed via a VDI solution such as Citrix Receiver, the administrator may wish to restrict use to iPad devices to take advantage of the larger screen. Having knowledge of the class of device as it connects to the network and being able to differentiate iPads allows the administrator to open up access to the Citrix server farm and potentially provide differentiated QoS for the Citrix ICA traffic.

The following enforcement policy example shows how the Profiler *Model* attribute is being referenced whenever a device attempts to connect to the enterprise network.

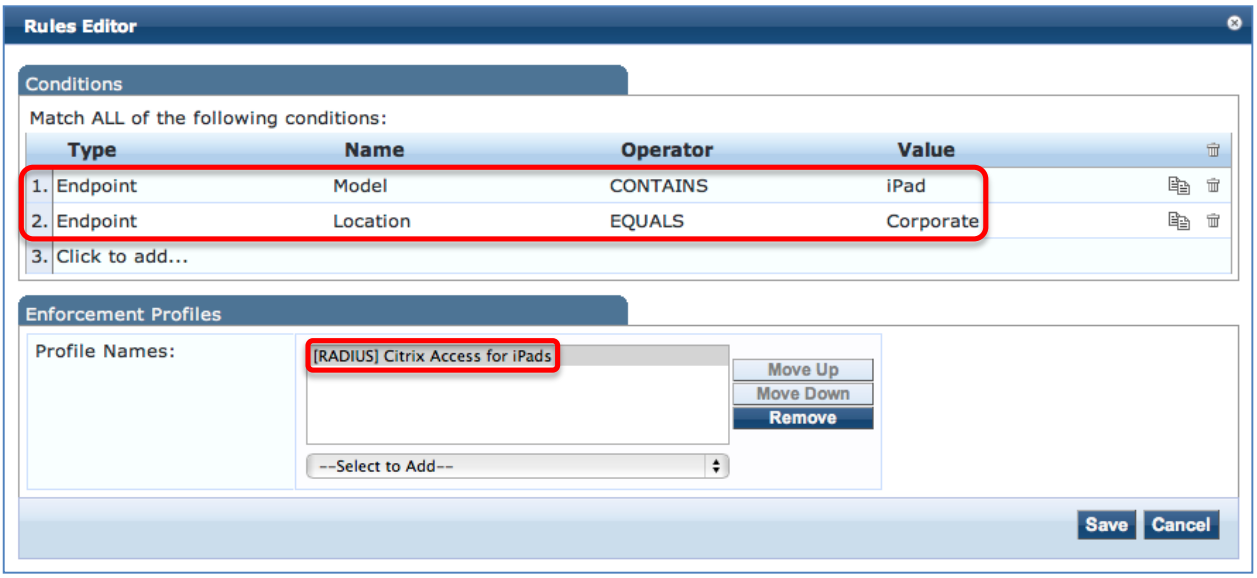


Figure 20 - Network Enforcement – Device Model type

In the event that this attribute contains a reference to iPad as configured by the EMM platform, then the network enforcement profile applied will result in the device being placed in the corporate access role which grants access to the Citrix application servers and also enables a high level of Quality of Service (QoS) for these applications. Alternatively, if this attribute does not include a reference to iPad as the device name, the network enforcement profile applied will restrict access to only essential internal resources and apply a best effort QoS profile.

Quarantine Device Type

It has become a regular occurrence that vulnerabilities are being discovered on smart phones and tablets. The open source nature of the Android operating system has provided a rich environment for potential vulnerabilities to be exposed and being able to classify devices at a granular level allows for administrators to quickly put in place quarantine rules in the event of a targeted exposure being discovered.

For example in early 2012, a vulnerability in a range of HTC smartphones was discovered where enterprise 802.1x credentials could be recovered from the operating system from a rouge application and potentially published remotely via standard Internet access.

Leveraging the EMM inventory data, ClearPass can clearly differentiate between Android devices from different manufacturers such as Samsung, HTC and Motorola and moreover leverage knowledge of individual model types of devices if such a granular policy is needed.

The following enforcement policy example shows how the Profiler *Device Name* attribute is being referenced whenever a device attempts to connect to the enterprise network.

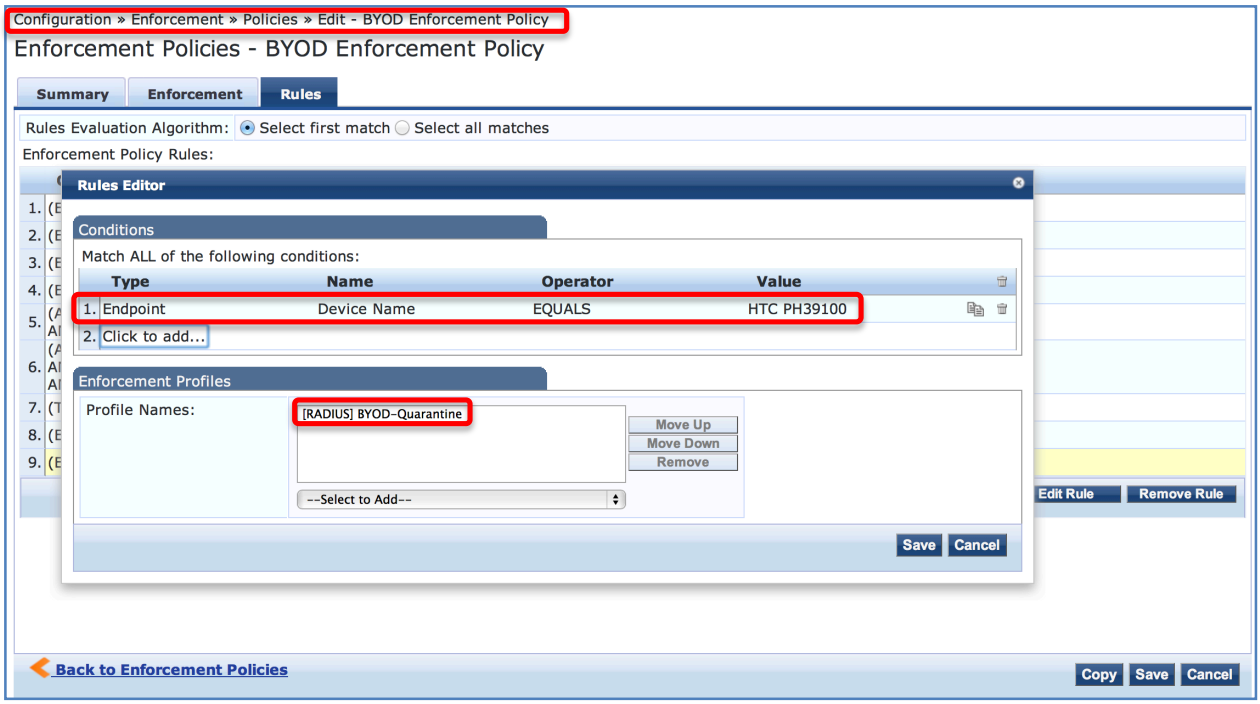


Figure 21 - Network Enforcement – device name

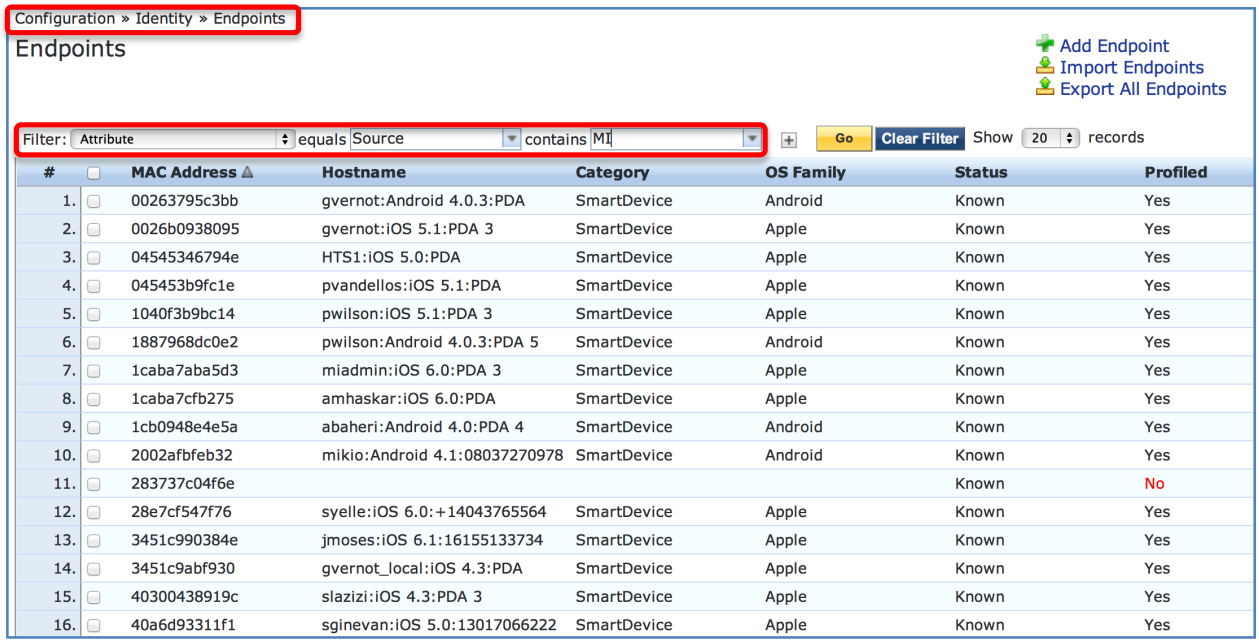
In the event that this attribute contains a reference to HTC as configured by EMM platform, then the network enforcement profile applied will result in the device being placed in a quarantine state and redirected to a captive portal page informing the user of the potential vulnerability on their device and advise on remediation steps via software upgrade. For more information on this vulnerability, please refer to the following article. http://www.kb.cert.org/vuls/id/763355

Managing Endpoint Data

The data received from EMM vendors is normalized and stored into the Endpoint database can be accessed from the **ClearPass Configuration > Identity > Endpoints** menu option.

Using the Endpoint information in the Endpoint Database you can query the ingested endpoint information using the following options.

Note: A filter can be created within the Endpoint database to restrict the view of endpoints to only those populated via the selected/preferred EMM platforms.



#	MAC Address	Hostname	Category	OS Family	Status	Profiled
1.	00263795c3bb	gvernot:Android 4.0.3:PDA	SmartDevice	Android	Known	Yes
2.	0026b0938095	gvernot:iOS 5.1:PDA 3	SmartDevice	Apple	Known	Yes
3.	04545346794e	HTS1:iOS 5.0:PDA	SmartDevice	Apple	Known	Yes
4.	045453b9fc1e	pvandellos:iOS 5.1:PDA	SmartDevice	Apple	Known	Yes
5.	1040f3b9bc14	pwilson:iOS 5.1:PDA 3	SmartDevice	Apple	Known	Yes
6.	1887968dc0e2	pwilson:Android 4.0.3:PDA 5	SmartDevice	Android	Known	Yes
7.	1caba7aba5d3	miadmin:iOS 6.0:PDA 3	SmartDevice	Apple	Known	Yes
8.	1caba7cfb275	amhaskar:iOS 6.0:PDA	SmartDevice	Apple	Known	Yes
9.	1cb0948e4e5a	abaheri:Android 4.0:PDA 4	SmartDevice	Android	Known	Yes
10.	2002afbfeb32	mikio:Android 4.1:08037270978	SmartDevice	Android	Known	Yes
11.	283737c04f6e				Known	No
12.	28e7cf547f76	syelle:iOS 6.0:+14043765564	SmartDevice	Apple	Known	Yes
13.	3451c990384e	jmoses:iOS 6.1:16155133734	SmartDevice	Apple	Known	Yes
14.	3451c9abf930	gvernot_local:iOS 4.3:PDA	SmartDevice	Apple	Known	Yes
15.	40300438919c	slazizi:iOS 4.3:PDA 3	SmartDevice	Apple	Known	Yes
16.	40a6d93311f1	sginevan:iOS 5.0:13017066222	SmartDevice	Apple	Known	Yes

Figure 22 – Example of Endpoint device list

It's important you configure the Filter in the following fashion.

- **Filter** = 'Attribute'
- **Equals** = 'Source'
- **Contains** = 'as shown in the table below'

Vendor	MobileIron	AirWatch	SOTI	JAMF	MaaS360	XenMobile	SAP
Use this value in the **Contains** field	MI	AIRWATCH	SOTI	JAMF	M360	XenMobile	SAP

Supported EMM Vendors s/w Versions

Below are the current EMM vendors supported and the version of ClearPass in which their support first appeared. Other vendors may be added depending on the market dynamics and the demand from the field. In addition, though not an EMM solution, Aruba Activate support is provided to obtain device information about registered access points.

EMM Vendors	Minimum Software Release	Latest Software Tested	ClearPass Version
AirWatch	6.2	7.3.0.400	6.0.2 or later
FiberLink MaaS360	4.0	Latest Cloud	6.0.2 or later
JAMF Casper Suite	8.5	9.4.28064.se	6.0.2 or later
MobilieIron VSP	4.5.3	6.0	6.0.2 or later
SOTI MobiControl	9.03	11.01.14221	6.0.2 or later
Citrix XenMobile	8.5	8.6.0 (9.0)	6.2.0 or later
SAP Afaria	7.0 SP4	7.0 SP4	6.4.0 or later
BlackBerry	BES10	BES10	6.4.0 or later

Figure 23 – Supported EMM Vendor software levels

Enable / Disable Individual MDM Context Servers

In CPPM 6.4, we added the ability to effectively disable or enable a Context Server. When adding context-severs they are enabled,it may be pertinent to disable the server for testing.

Note: Only one server for a particular EMM vendor can be active/enabled at the same time. Using this feature allow you to define multiple say MobileIron context servers, with one active and one disabled as required.

Note: When adding new Context servers in CPPM 6.4+ you need to specifically 'enable' the context server as highlighted below.

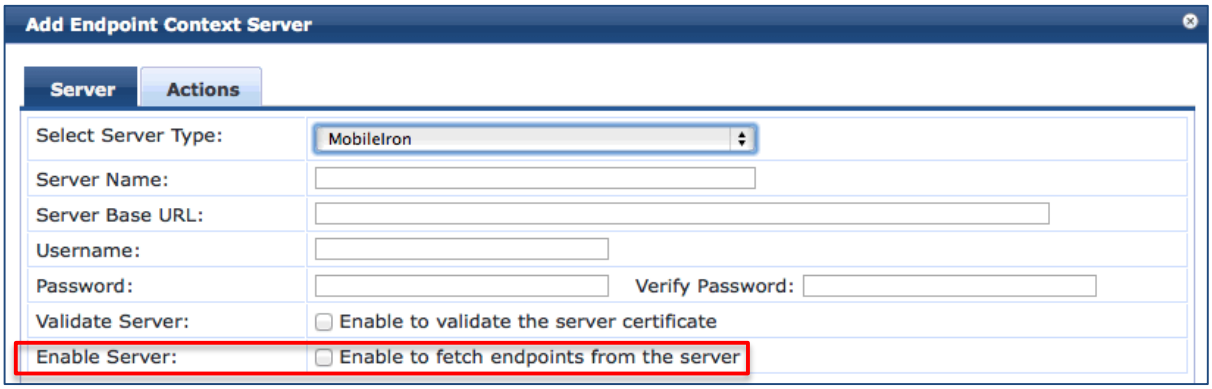


Figure 24 - Enabling/Disabling Context Servers

AirWatch (Acquired by VMware in January 2014)

To configure the AirWatch connector, enter a hostname into the "Server Name" field. This hostname can be derived as shown in the next paragraph. You typically do not need to alter the "Server Base URL". The API Key can be found in the location described later in this section.

Figure 25 - AirWatch Context Server configuration screen

Note: We have seen that AirWatch instances can be referenced proceeding with either **'as'** or **'cn'** characters, i.e. **asXXX.awEMM.com** or **cnXXX.awEMM.com.**
Note: The EMM instances that begin with **cnXXX** typically do not support the API interface required for CPPM to extract information from the EMM instance.

A foolproof way to determine the appropriate value for "Server Name" is to look in the AirWatch portal configuration under **Menu > System Configuration > System > Advanced > Site URLs**. The value required for "Server Name" is the *hostname* portion of the value in the field "REST API URL".

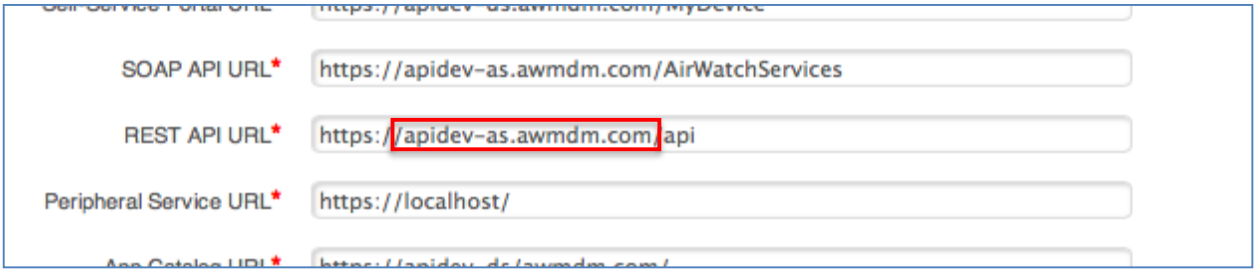


Figure 26 - AirWatch server name

AirWatch utilizes a variable called the API Key. This is configured within the AirWatch portal, and must be enabled for CPPM to authenticate itself with AirWatch.

In the AirWatch portal, go to **Menu > System > Advanced > API > REST API** as shown below and click the "Enable API Access" checkbox. This API key is leveraged by the EMM Integration API calls between CPPM and AirWatch to provide an additional level of authentication over and above the basic HTTP authentication of Username and Password.

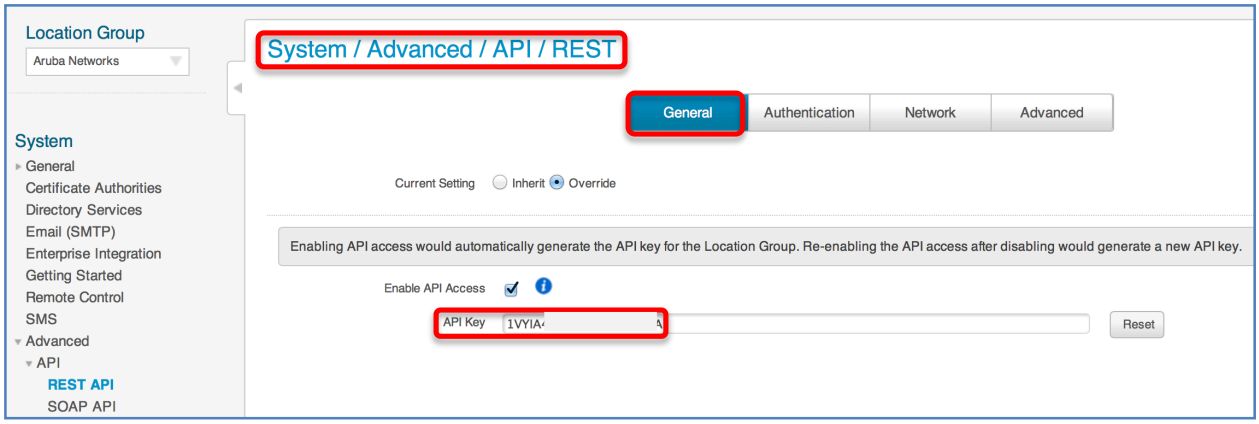


Figure 27 - AirWatch portal configuration

The account you use for API access must have either a role of **System Administrator** or **API Full Access**. The role can be changed by creating an administrator-type account (under **Menu > Accounts > Administrators**) and setting the role on the Roles tab.

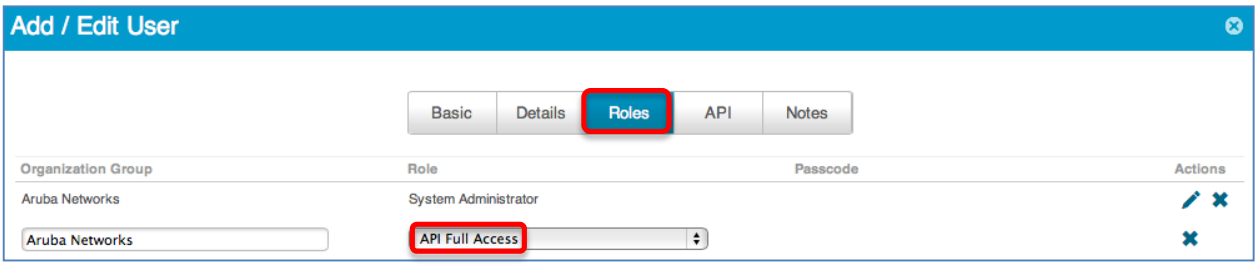


Figure 28 - Enable AirWatch admin account for API access

Finally to allow CPPM to 'communicate' with AirWatch, the admin account you're using must be enabled to support HTTP authentication for API access to the Air-Watch EMM platform.

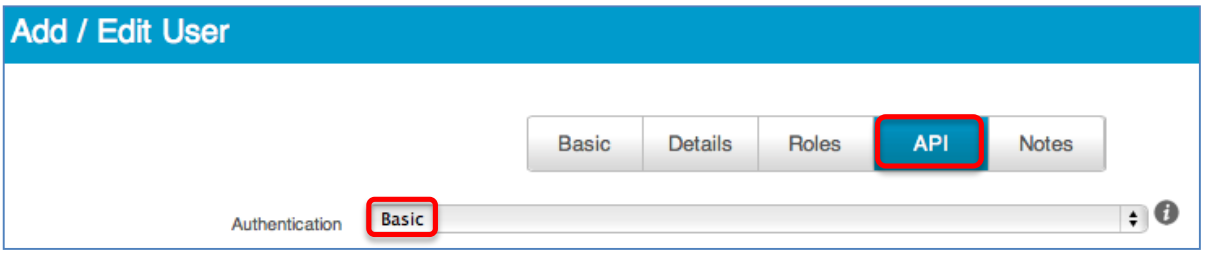


Figure 29 - Enable AirWatch admin account for Basic Authentication

AirWatch Endpoint Attributes

The CPPM EMM service will normalize data received from AirWatch in the Endpoint identity database. The table below shows the normalized data attributes that are available from AirWatch. If there are specific normalized values, those are also shown in the table.

Endpoint Tag	Tag Type	Specific Values
Manufacturer	Inventory	
Model	Inventory	
OS Version	Inventory	
UDID	Inventory	
Serial Number	Inventory	
Phone Number	Inventory	
Description	Inventory	
Source	Inventory	AirWatch
Ownership	Inventory	Corporate/Employee/Shared
EMM Identifier	Inventory	
Compromised	Policy	True or False
Encryption Enabled	Policy	True or False
EMM Enabled	Policy	True or False
Last Check In	Policy	
Required Application	Policy	False/Installed/Missing
Blacklisted Application	Policy	True or False

Figure 30 - AirWatch Endpoints Attributes

JAMF Configuration

To configure the JAMF connector, enter a hostname into the "Server Name" field. This is typically **jss.jamfcloud.com** for cloud-hosted deployments. On premises installations will differ and will most likely be the local server's hostname. For the "Server Base URL", you will most likely have to add on your customer name or other identifier to form the complete URL. The value in the 'Server Base URL' field should be the same URL used to access your JAMF console.

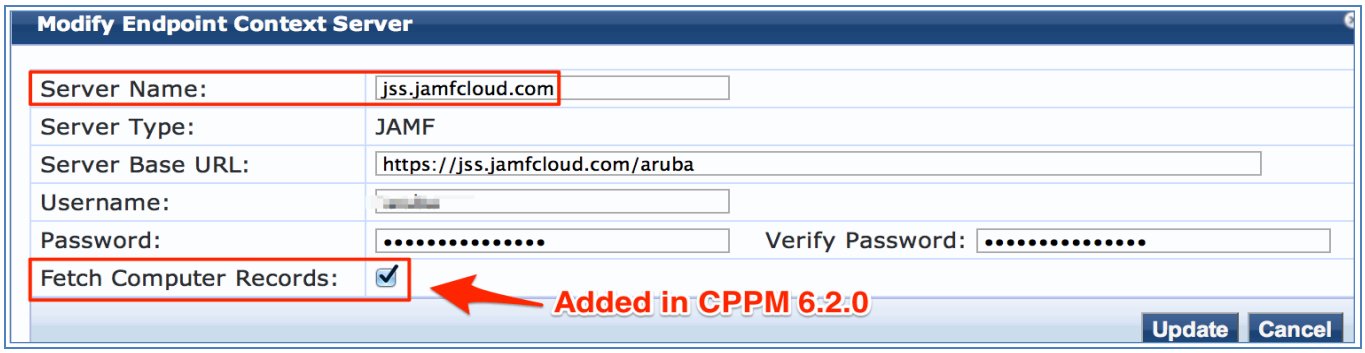


Figure 31 - JAMF Context Server configuration screen

Note: Starting in CPPM 6.2.0 we added the ability to ingest Computer in addition to the existing smart-devices that are under JAMF management. To enable this feature select the 'Fetch Computer Records' option as shown above in the JAMF endpoint definition.

JAMF Endpoint Attributes

The CPPM EMM service will normalize data received from JAMF into the Endpoint database. The table below shows the normalized data attributes that are available from JAMF, any specific attributes are noted below.

Endpoint Tag	Tag Type	Specific Values
Model	Inventory	
OS Version	Inventory	
UDID	Inventory	
Serial Number	Inventory	
Ownership	Inventory	
Source	Inventory	JAMF
IMEI	Inventory	Field only shown for Smartphones
Phone Number	Inventory	Shown even for Computers
Display Name	Inventory	
EMM Identifier	Inventory	
EMM Enabled	Policy	True or False
Compromised	Policy	True or False
Encryption Enabled	Policy	True or False
Blacklisted App	Policy	True or False
Required Application	Policy	False/Installed/Missing
Last Check In	Policy	

Figure 32 - JAMF Endpoints Attributes

Note: While the EMM vendor does not report the Manufacturer, DHCP fingerprinting can be used to provide this.

MaaS360 Configuration (Acquired by IBM in December 2013)

To configure the MaaS360 connector, you will need a considerable amount of information. To start, enter a hostname into the "Server Name" field. This is typically **services.fiberlink.com**. You should not need to alter the "Server Base URL". See the following paragraph for an explanation of the additional values required.

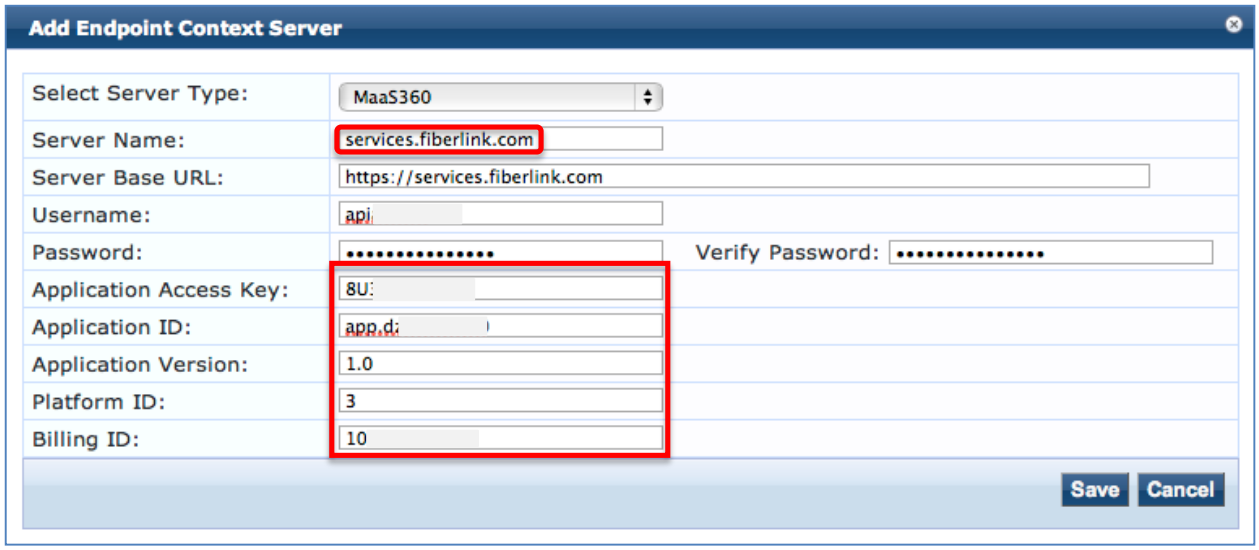


Figure 33 - MaaS360 Context Server configuration screen

MaaS360 utilizes multiple attributes over and above basic HTTP authentication as shown above. The following inputs will need to be configured inside of ClearPass.

- Application Access Key: `<Obtained from MaaS360>`
- App ID (for App authorized to use MaaS360 services): `<your-network-domain.com>`
- App Version: `1.0`
- Platform ID: `3`
- Billing ID: `<Your MaaS360 ID>`

Most of the above details will be supplied by Fiberlink, however your Billing ID is visible in the footer of your MaaS360 portal page (labeled Account #) as shown below.

MaaS360 Endpoint Attributes

The CPPM EMM service will normalize data received from MaaS360 in the Endpoint identity database. The table below shows the normalized data attributes that are available from MaaS360. If there are specific normalized values, those are also shown in the table.

Endpoint Tag	Tag Type	Specific Values
Manufacturer	Inventory	
Model	Inventory	
OS Version	Inventory	
Phone Number	Inventory	
Source	Inventory	MaaS360
Owner	Inventory	
UDID	Inventory	
IMEI	Inventory	
Display Name	Inventory	
Ownership	Inventory	
EMM Identifier	Inventory	
Last Check In	Policy	
Compromised	Policy	True or False
Blacklisted App	Policy	True or False
Required Apps	Policy	False/Installed/Missing
Encryption Enabled	Policy	True or False

Figure 34 - Maas360 Endpoints Attributes

Note: Not all endpoint attributes are available for all OS types.

MobileIron Configuration

To configure the MobileIron connector, enter a hostname into the "Server Name" field. This is typically **m.mobileiron.net** for cloud-based deployments. On premises installations will differ and will most likely be the local server's FQDN. For the "Server Base URL", you will most likely have to append your customer name or other identifier to form the complete Server Base URL.

Note: Use the following URL https://trust.mobileiron.com to check the service status of the MobileIron global system operations. You can also subscribe to updates from this portal.

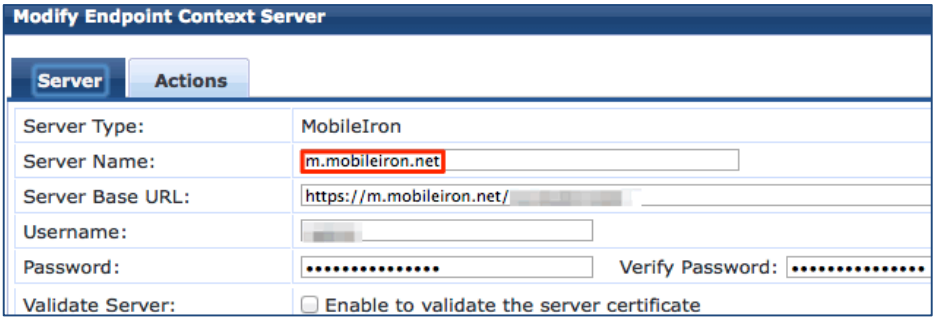


Figure 35 - MobileIron Context Server configuration screen.

MobileIron Endpoint Attributes

The CPPM EMM service will normalize data received from MobileIron in the Endpoint identity database. The table below captures the normalized data attributes from MI.

Endpoint Tag	Tag Type	Specific Values
Manufacturer	Inventory	
Model	Inventory	
OS Version	Inventory	
UDID	Inventory	See Note
Serial Number	Inventory	See Note
IMEI	Inventory	
Phone Number	Inventory	
Carrier	Inventory	
Source	Inventory	MobileIron
Owner	Inventory	
Display Name	Inventory	
Ownership	Inventory	
EMM Identifier	Inventory	
Compromised	Policy	True or False
Encryption	Policy	True or False
Blacklisted App	Policy	True or False
Required App	Policy	Installed/False/Missing
EMM Enabled	Policy	True or False
Last Check In	Policy	

Figure 36 - MobileIron Endpoints Attributes

Note: UDID and Serial Number are only available for IOS devices.

SOTI Configuration

To configure the SOTI connector, enter a hostname into the "Server Name" field. This is typically **XXXX.mobicontrolcloud.com** for cloud-based deployments. The XXXX portion will be your specific customer name or other identifier. On premises installations will differ and will most likely be the local server's hostname. The "Server Name" field should be the same hostname you use to access your MobiControl console. No changes are required to the "Server Base URL" which will be populated from you adding the Server Name.

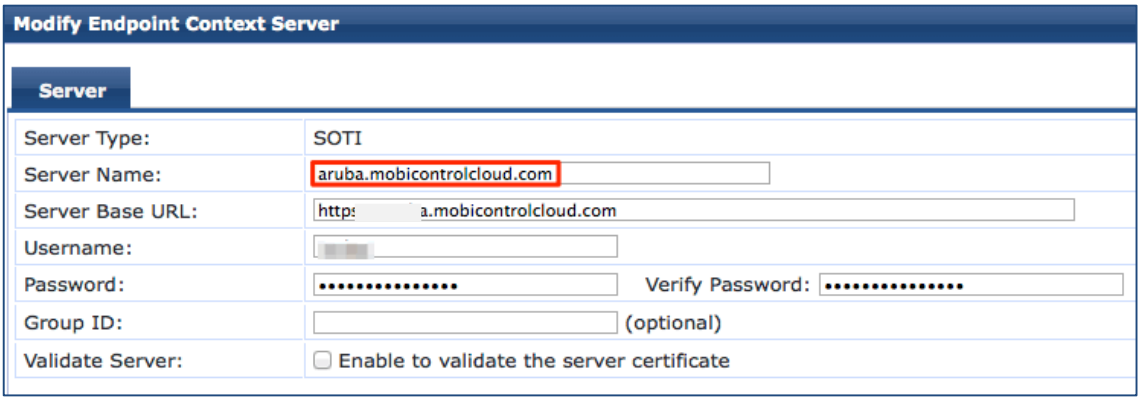


Figure 37 - SOTI Context Server configuration screen

You may have been provided with a Group ID from SOTI. If you don't have a Group ID, leave the field blank. However, on CPPM v6.0.2 the Group ID field is mandatory, just enter "0000000000" (that's ten zeros).

SOTI Endpoint Attributes

The CPPM EMM service will normalize data received from SOTI in the Endpoint identity database. The table below shows the normalized data attributes that are available from SOTI. If there are specific normalized values, those are also shown in the table.

Endpoint Tag	Tag Type	Specific Values
Manufacturer	Inventory	
Model	Inventory	
OS Version	Inventory	
Serial Number	Inventory	
Phone Number	Inventory	
Source	Inventory	SOTI
Ownership	Inventory	
Display Name	Inventory	
EMM Identifier	Inventory	
Encryption	Policy	True or False
Compromised	Policy	True or False
EMM Enabled	Policy	True or False
Last Check In	Policy	
Required Apps	Policy	Installed / Missing
Blacklisted Apps	Policy	True or False

Figure 38 - SOTI Endpoint Attributes

XenMobile Configuration

To configure the XenMobile EMM integration the Server Base URL is made up of several components. For cloud deployments there is a concept of a customer instance and this instance name is referenced twice in the Base URL as shown below:

https://<instance-name>.zc.zenprise.com/<instance-name>/nac

for example. **https://aruba.zc.zenprise.com/aruba/nac**

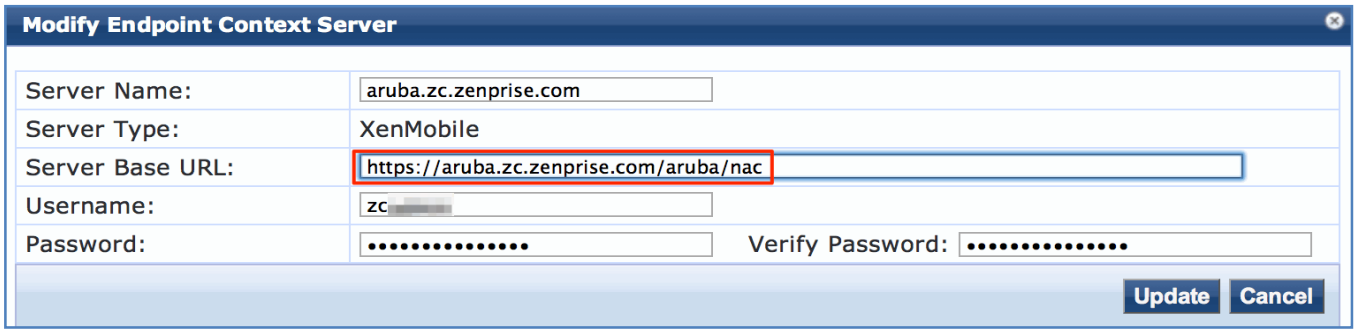


Figure 39 - XenMobile Context Server configuration screen

Note: For on-prem deployment of XenMobile, the concept of a customer instance is no longer required and the base URL reverts back to default format that includes the /zdm/nac suffix as shown below:

https://<customer-on-prem-deployment-fqdn>/zdm/nac

Xenprise Endpoint Attributes

The CPPM EMM service will normalize data received from Xenprise in the Endpoint identity database. The table below shows the normalized data attributes that are available from Xenprise. If there are specific normalized values, those are also shown in the table.

Endpoint Tag	Tag Type	Specific Values
Manufacturer	Inventory	
Model	Inventory	
OS Version	Inventory	
Source	Inventory	XenMobile
IMEI	Inventory	
Ownership	Inventory	
Serial Number	Inventory	
Phone Number	Inventory	
EMM Identifier	Inventory	
Encryption	Policy	True or False
Compromised	Policy	True or False
EMM Enabled	Policy	True or False
Blacklisted Apps	Policy	True or False
Required Apps	Policy	True or False

Figure 40 - Xenprise Endpoint Attributes

SAP Afaria Configuration

Included in the CPPM 6.4.0 release is support for SAP Afaria as an EMM Vendor, complementing the existing vendors we support. To configure the Afaria EMM integration, enter a hostname into the "Server Name" field. There is no specific name formatting for the HOST FQDN URL. SAP also supports an on-prem VM version of their application, this is deployed as a VM and there is no SAP appliance based hardware.

To use the Afaria NAC API, the credentials used in the API call need to be mapped to the specific tenant and they must have the Access Control Role configured.

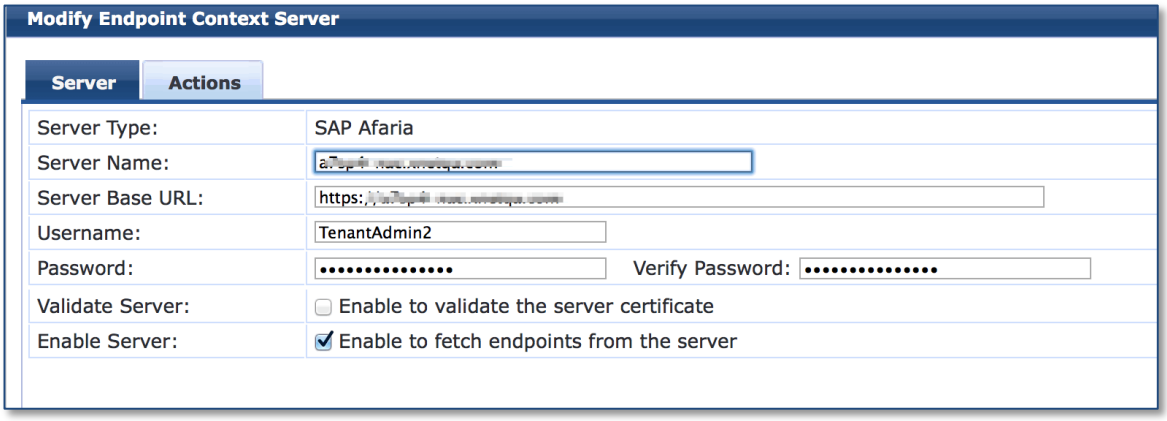


Figure 41 - SAP Afaria Context Server configuration screen

Afaria Endpoint Attributes

The CPPM EMM service will normalize data received from Afaria in to the Endpoint identity database. The table below shows the normalized data attributes that are available from Afaria. If there are specific normalized values, those are also shown in the table.

Endpoint Tag	Tag Type	Specific Values
Manufacturer	Inventory	
Model	Inventory	
OS Version	Inventory	
Serial Number	Inventory	
IMEI	Inventory	
Phone Number	Inventory	
Source	Inventory	SAP Afaria
Ownership	Inventory	
EMM Identifier	Inventory	
Compromised	Policy	True or False
Encryption	Policy	True or False
Blacklisted App	Policy	True or False
Required App	Policy	
EMM Enabled	Policy	True or False

Figure 42 - SAP Afaria Endpoint Attributes

Blackberry Enterprise Server v10

The integration we developed for BlackBerry BES10 server is slightly different than the other EMM integrations. It's important to know this was co-developed. The BlackBerry server runs on top of a Microsoft SQL DB, and the integration involves CPPM making real-time SQL calls to the BES10 MS-SQL tables. BlackBerry co-operation with Aruba developed a special MS-SQL view of their underlying tables to allow us to check on a number of attributes related to enrolled devices. In the example below we show how to make an Authz call to the BES MS-SQL DB to extract the Ownership of an endpoint and then use this to drive the role-mapping for a user. A list of the other exposed SQL attributes is shown later in this section.

We are assuming that you have the BES Server installed. We do not cover the setup or installation of BES or the enrollment of Devices in BES. Once this has been completed we recommend the use of the the Microsoft SQL Management Studio Express for SQL-Admin activities. It can be downloaded from https://www.microsoft.com/en-ca/download/details.aspx?id=7593

Before you begin the below configuration, we recommend you take a backup of the BES Database.

BES10 Configuration

Once the SQL Mgmt tool is installed, we need to create the SQL view. At the top of SQL Mgmt studio you should see a spot where you can select the database. It's a drop down box. It's probably set to Master. Open up a new query window, Select the drop down box and select BDSMgmt_UDS, load the script into a new query window and click "Execute", see below for how to access this script.

The script will create a view called **vw_wifi_mac_device**. To test this is working you just need to run **"SELECT * FROM vw_wifi_mac_device"** to obtain the data from it. Be careful as this will select all endpoint records in the DB. If you know the MAC address of a single device, a better test would be to just query for that device. Use the following SQL to test for a single device **"SELECT * FROM vw_wifi_mac_device WHERE WiFiMac ='00:11:22:33:44:55"** [change **00:11:22:33:44:55** to equal your MAC address]. It's a good test to ensure the above SELECT works as this is a final acceptance test to be certain the SQL View is installed correctly.

To obtain the SQL source to create the table view contact danny@arubanetworks.com. We are not generally publishing the SQL at this time, as we want to make sure the CPPM and BES10 integration are managed successfully.

To check the view has been created successfully, review the following example below on the left to check the view has been created. Also below on the RHS of the page is a copy of all of the available fields exposed under the SQL view we have just created. In theory you

could use any of these fields in processing on Authz check within CPPM. For our documented example we will only be using the **'Ownership'** field, not the Owner field.

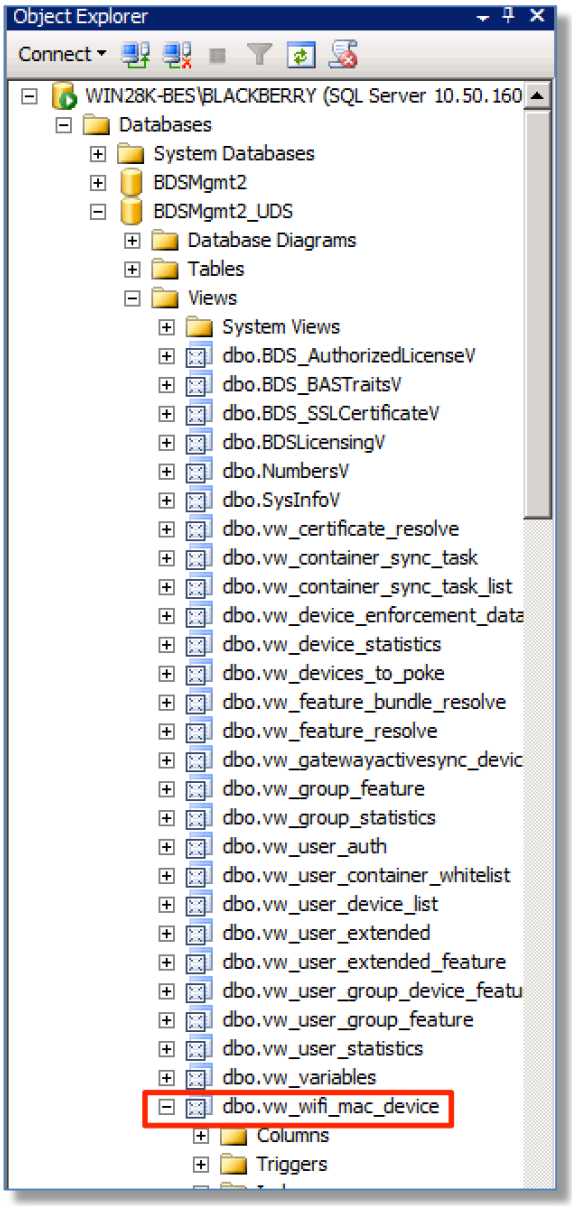


Figure 43 - Checking new SQL is created

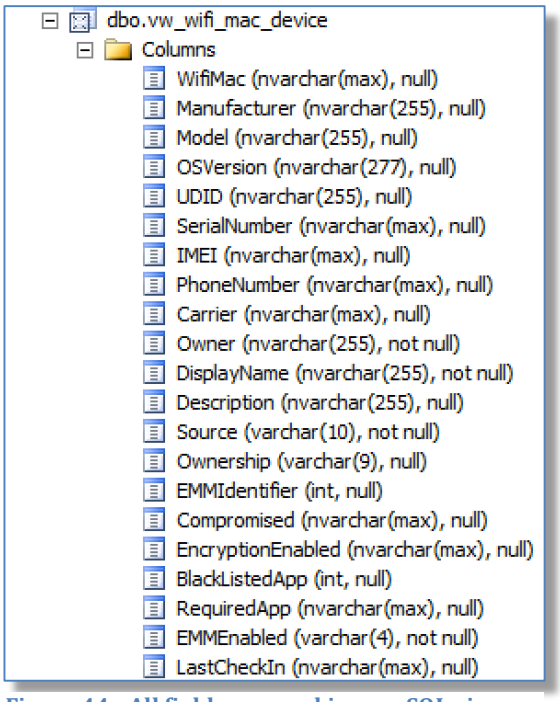


Figure 44 - All fields exposed in new SQL view

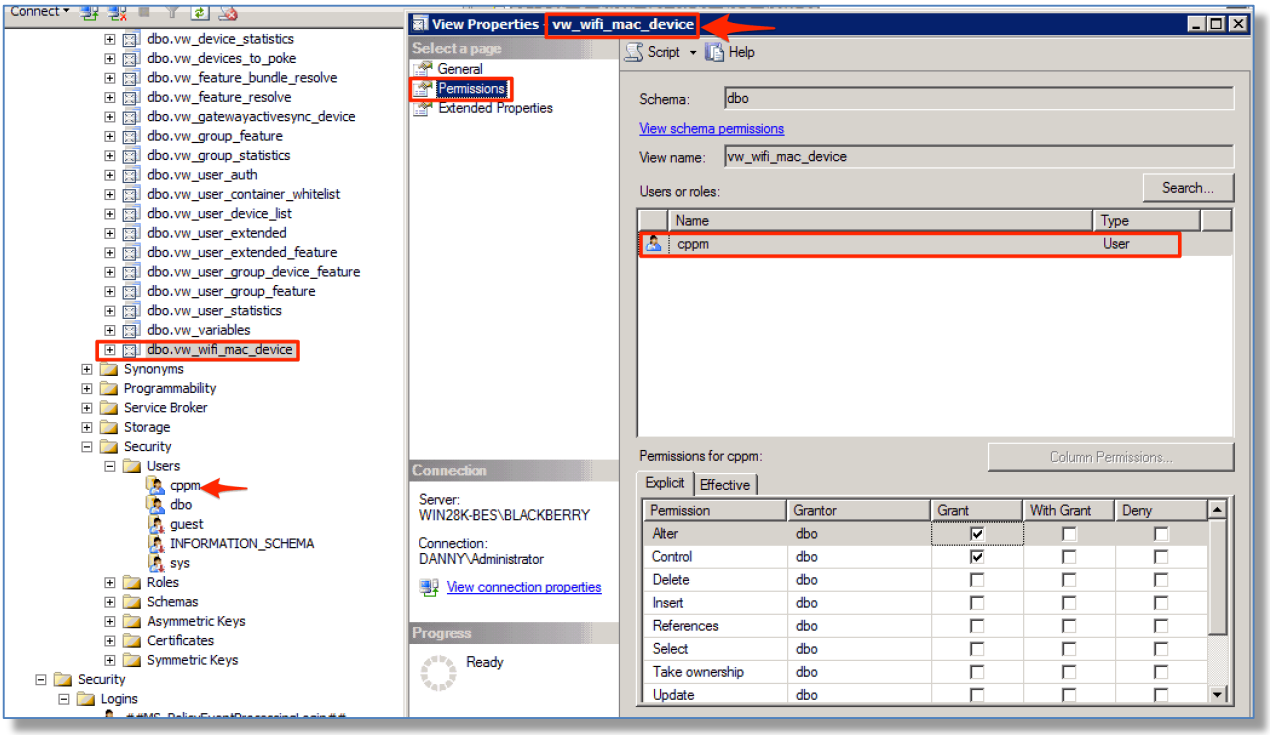


Figure 45 - Setting Table View security

Above we see some of the SQL-Admin changes we had to make. For our testing we created a new user that was authorized to the table-view. It's recommended that you consult with the enduser/customer to ensure that any security/access to the SQL-DB is secured as required. Our above example is more for the simplicity of our LAB testing.

We created a user 'cppm' as can be see above, then we edited the Permissions for the SQL view and granted all access to cppm. As can be seen below we then used this user 'cppm' configured within ClearPass as the Login user to the MS-SQL DB.

ClearPass Configuration for BESv10

ClearPass configuration requires several steps. The first is to configure and connect CPPM to the MS-SQL DB we want to access. Configure this under Configuration -> **Authentication -> Sources -> [Add a new source]**

As you can see below, add a new SQL source (ensure you select MSSQL on the later), provide the required IP address, Login Name/Password and leave the Password type as 'Cleartext'.

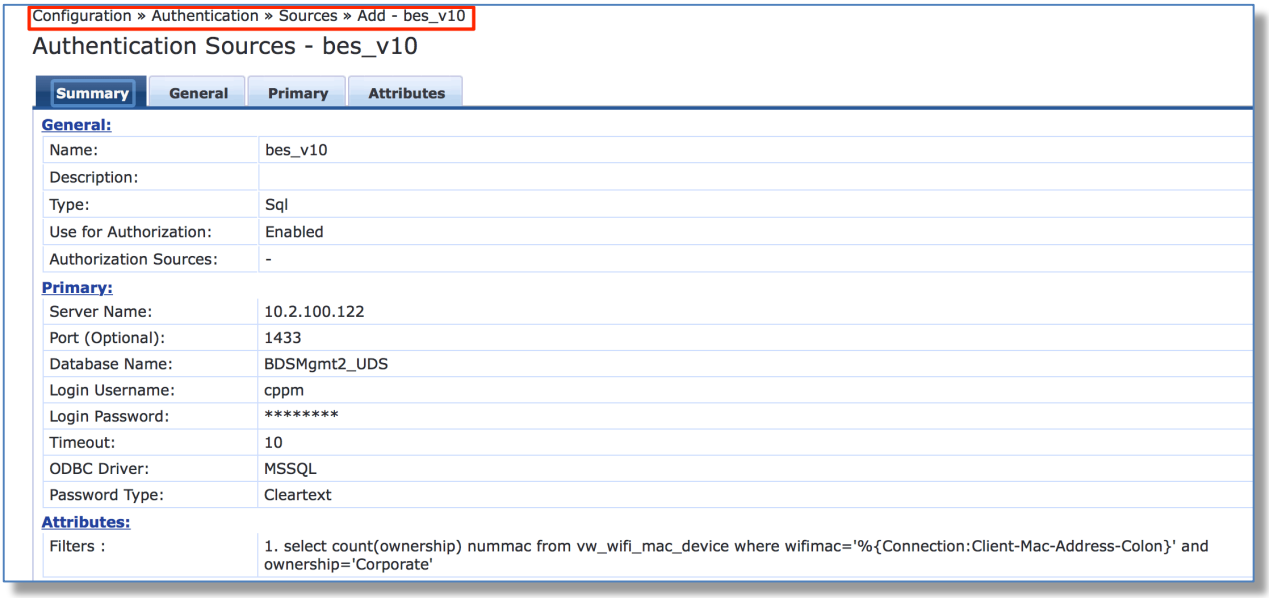


Figure 46 – Adding a the BES10 SQL database into ClearPass

Within the Attributes Tab, you need to take particular care of the configuration. Create a filter and add a field-name. I suggest you follow my guidance below but you may want multiple fields or want to call your fields by another name. Be aware as some of the fields you create are referenced in other places so making changes can cause additional issues.

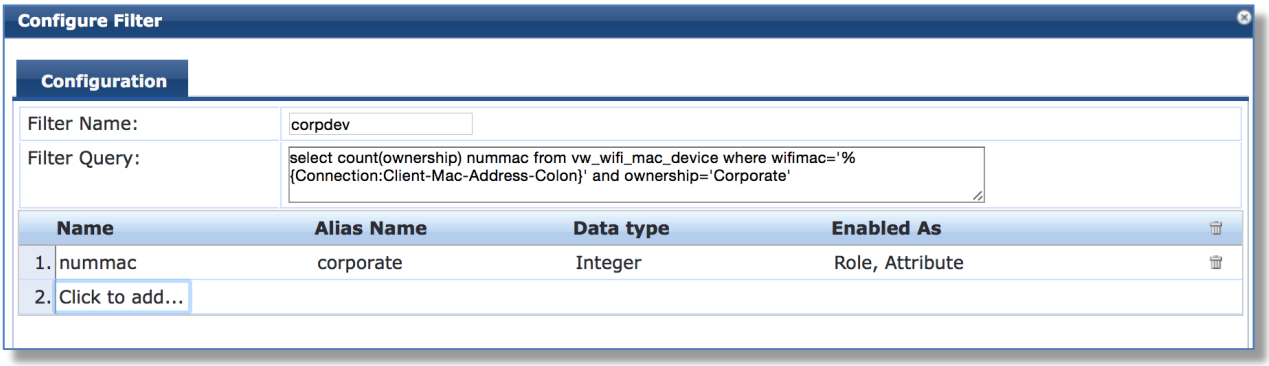


Figure 47 - Creating the SQL filter to 'grab' data from the BES10 MS-SQL view

Below I have added the actual SQL Query that you need to paste when creating the Filter above. Again note the field nummac, which I also used in the filter. The key to the below is the use of the Connection namespace. We grab the MAC address in colon format. All of the MAC addresses in the BES10 DB are stored in colon format, so the below is taking the MAC from the incoming RADIUS auth and using this to lookup the endpoint in the BES10 DB. The other important thing to be aware of from the below is we are also grabbing the ownership of the endpoint. As we discussed there could be other attributes you want to check in the BES10 DB but hopefully the example here provides enough information for you to expand/modify as required.

```
select count(ownership) nummac from vw_wifi_mac_device where
wifimac='%{Connection:Client-Mac-Address-Colon}' and
ownership='Corporate'
```

Figure 48 – CPPM SQL Query to check on MAC address and grab Ownership attribute

Now that we have the BES10 SQL-DB defined and the SQL Query setup in CPPM we can now use these attributes/fields within our Service Policy. The process to use the BES10 context in our example is to examine the Ownership of the endpoint. By this we want to make a role mapping decision based upon if the endpoint is enrolled within BES and the ownership is Corporate. By this we know the user is using a Company provided Smart-Device and as such we can assign network privileges accordingly.

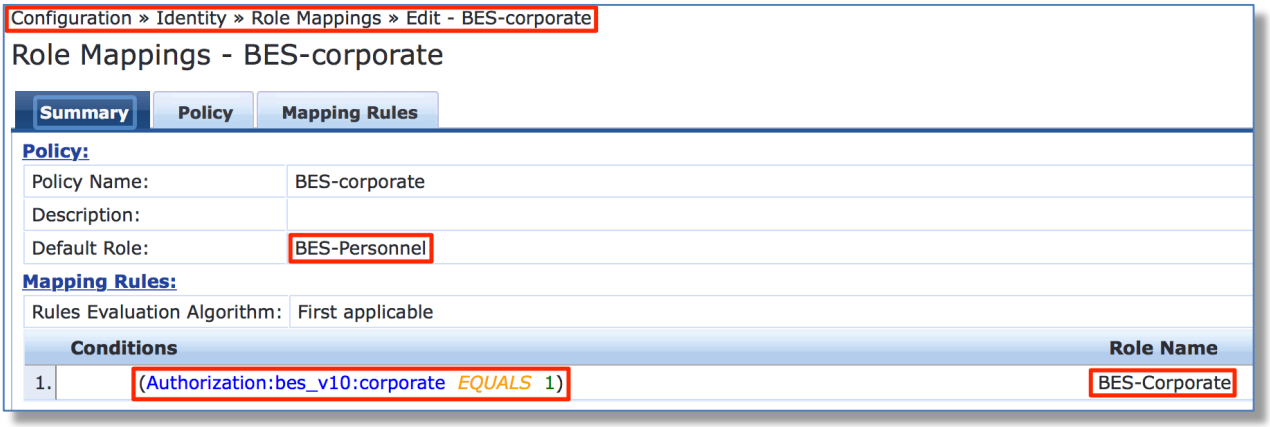


Figure 49 – Assigning a role to the session based upon the BES lookup

The above role-mapping is using the BES10 DB to perform an Authorization lookup. We check above in our SQL Query for the MAC being present **AND** the Ownership being set to Corporate. If both of these check are true then we will return a value of '1' else a value of '0'. Our role mapping above checks for a 1 being returned by the SQL Query and sets a Role of BES-Corporate.

Your logic can and will likely differ.

Note: Remember to add the SQL Auth source to the actual Service Policy you will process this under

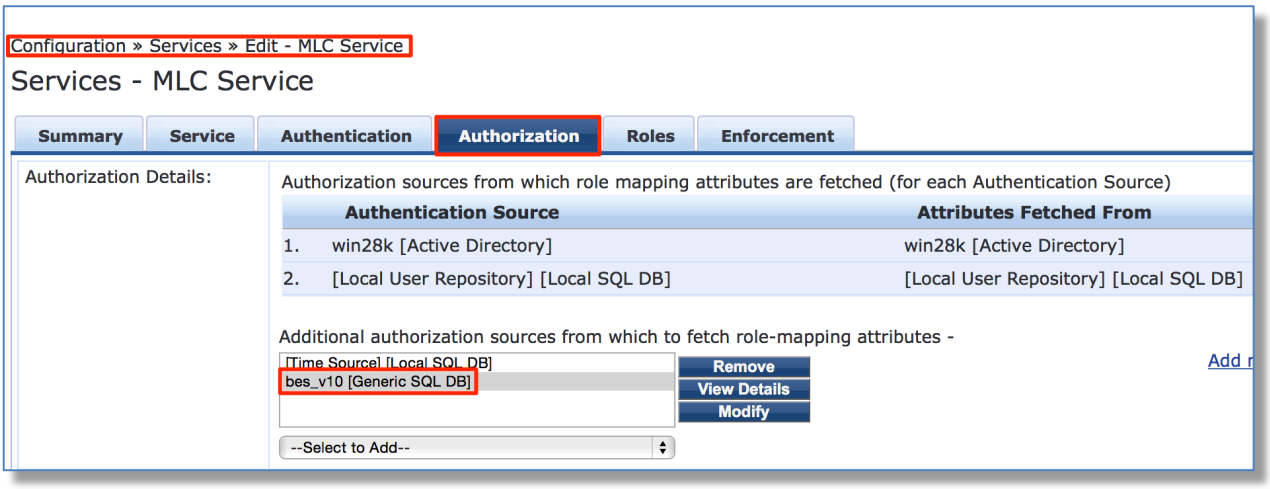


Figure 50 – Remember to add the BES10 as an Authorization source

Globo GO! Enterprise v3.9.2

The integration we developed for Globo GO! Enterprise is slightly different than the most of our other MDM/EMM integrations, it does follow a similar framework we recently completed for BlackBerry in that we are querying an underlying SQL DB and using this as an Authz source when processing the users network-access. It's important to know this solution was co-developed with Globo and has their approval.

The Globo GO! server runs on top of a Microsoft SQL DB, and this integration involves CPPM making real-time SQL calls to the MS-SQL tables. With the co-operation of Globo we have been able build/test and document a few common integration use cases. In the two examples that follows we show how to make an authz call to the MS-SQL DB to extract the Ownership of an endpoint and then use this to drive the role-mapping for a user, and separately how to check on the device being jailbroken. These being just two simple but common request to utilize the device context which Globlo Go! has as an authorization source for the users device.

We are assuming that you have the Globo GO! Server installed. We do not cover the setup or installation or the enrollment of Devices in Globo GO!. Once this has been completed we recommend the use of the the Microsoft SQL Management Studio Express for SQL-Admin activities. It can be downloaded from https://www.microsoft.com/en-ca/download/details.aspx?id=7593

Note: We make the assumption that the Globo GO! Enterprise software is deployed, we provide no installation/configuration guidelines for the deployment and configuration of the Globo product.

Note: Globo has not performance tested this integration and therefore any Customer should have their DBA closely baseline this process before and after integration to ensure general SQL performance is not negatively impacted.

Globo Configuration

Following on from the installation of the MSFT SQL Mgmt Studio we must complete some additional network specific configuration changes to allow CPPM to access the SQL DB.

Load the SQL Server Configuration Manager, and then there are several checks and changes we need to make to complete the configuration.

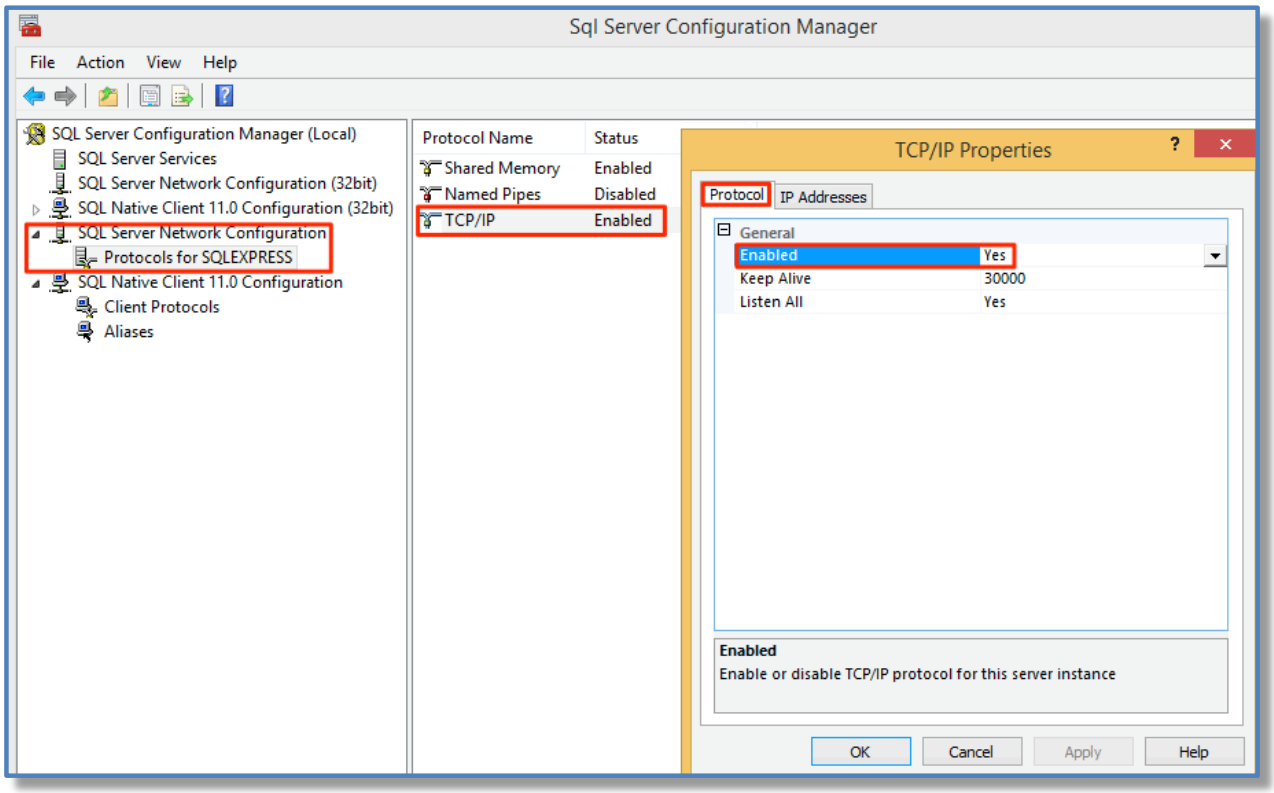


Figure 51 - SQL Server Configuration Manager

Select the **'SQL Server Network Configuration'**, expand that, select the **'Protocols for….'** And double-click on **'TCP/IP'**.

Under the **Protocol** Tab on **TCP/IP Properties – 'Enabled' & 'Listen All'** are set as **Yes**, if either/any are set to No, click on the box, select Yes and hit Apply at the bottom.

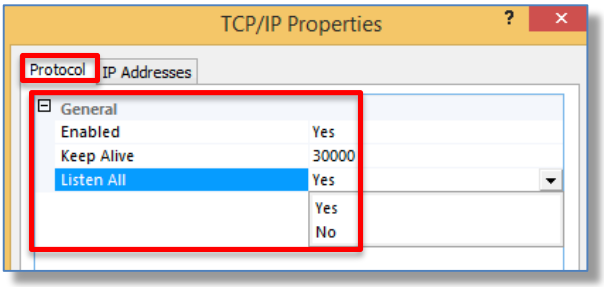


Figure 52 - Setting Listen All to 'Yes'

Next click on the IP Address Tab……

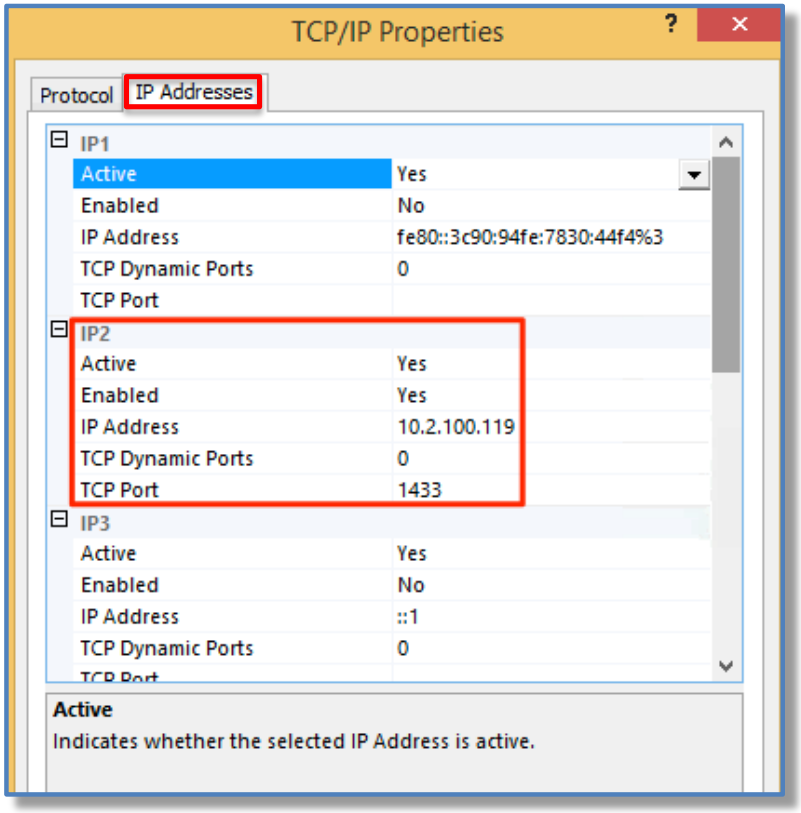


Figure 53 - Setting the TCP Port to '1433' and disable Dynamic Ports

Under the **IP Address** Tab there will be multiple entries based upon the number of physical interfaces on the server, and additionally the IPv4 and IPv6 interfaces. We are specifically interested in setting the configuration for the IP address that you want ClearPass to communicate with. In the above, we are interested in the highlighted interface with IP address 10.2.100.119. We have set the **TCP Dynamic Ports** to '0' that's a zero and hard-configured the **TCP Port** to listen on port 1433. Again ensure the interface in question is **Active** and **Enabled**.

Note: Consider that you may have to amend the Server firewall to allow SQL traffic through [TCP port 1433], whilst testing and to remove additional complication during this stage we recommend you consider disabling the server firewall to remove the potential of additional complications and then once the process of the CPPM integration is complete you re-visit the firewall configuration issue.

The final step we need to perform for the Globo configuration is to ensure we have a user-id configured in the MSFT SQL DB that can be used by CPPM to query the underlying tables.

Load the **MSFT SQL Serer Management Studio**, expand the **Databases** and under **Security**, expand Login…….. then right-click on Logins to create a New Logins……

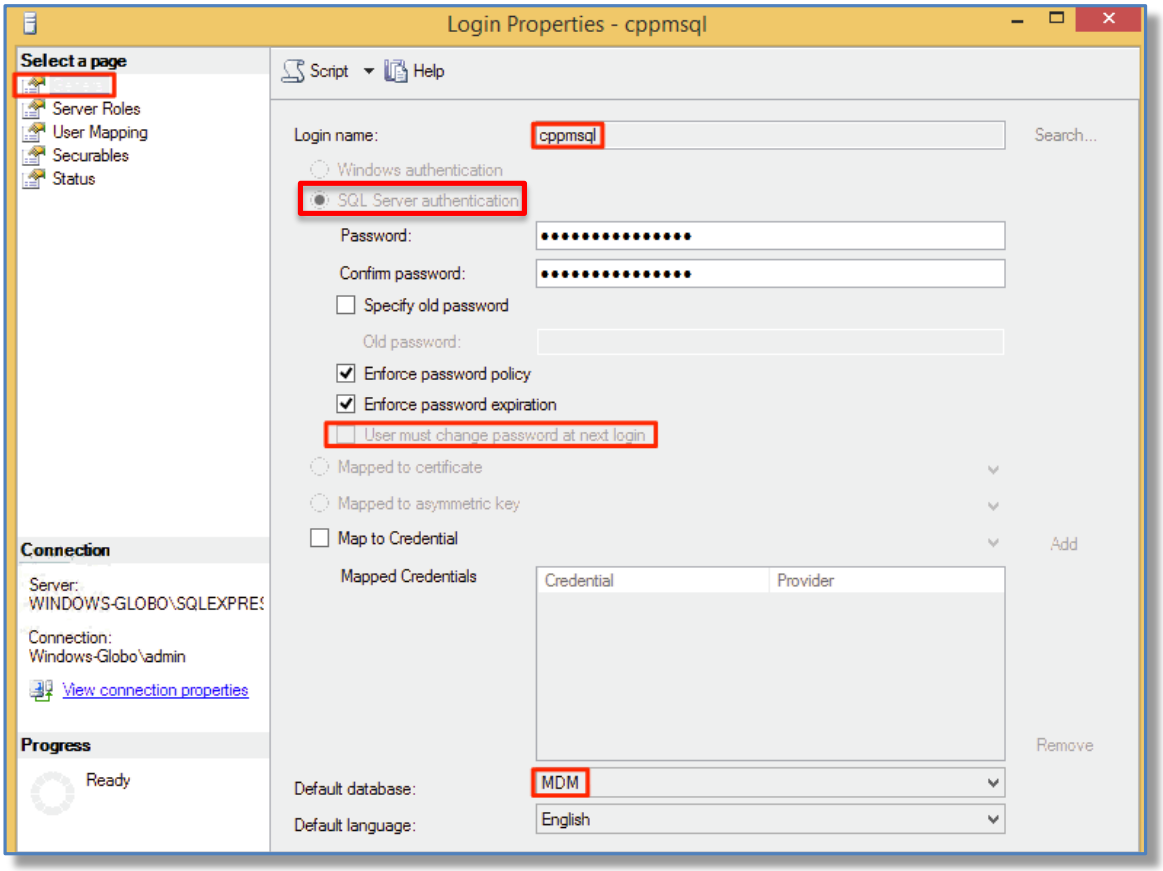


Figure 54 - Creating a user in the MSFT SQL DB

A few settings are required, I've documented how I set this up to work in my LAB, your SQL DB-Admins may have a different preferred approach.

Ensure you choose **'SQL Server authentication'**, de-select **'User must change password at next login'**, and set the Default database to **'MDM'**. Under **'Server Roles'** below check **'public'** is selected.

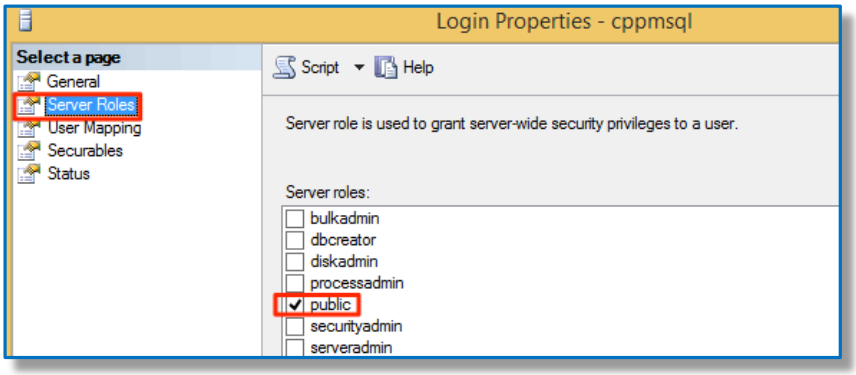


Figure 55 - Check SQL Server Roles

Under the **User Mapping**, ensure you set up the options for the **'MDM'** database set as shown below…….

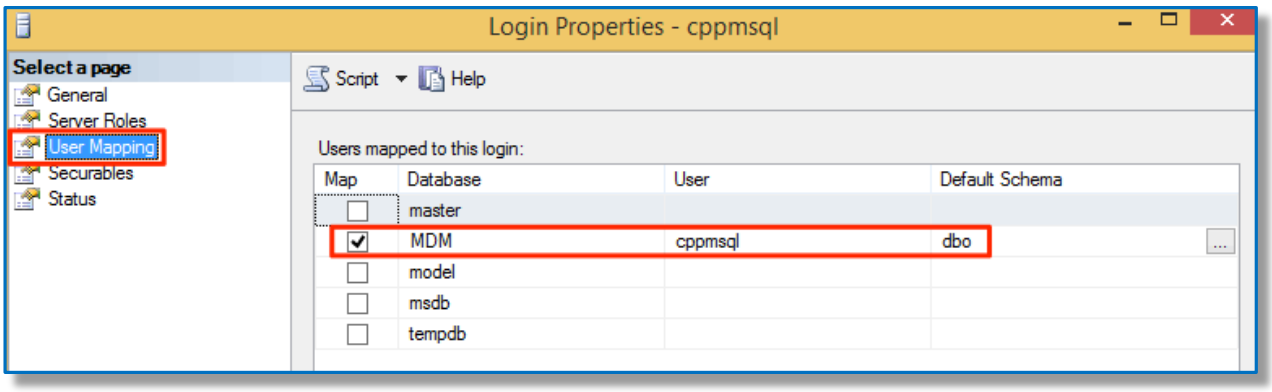


Figure 56 - Configure User Mapping

After creating the user, we nee d to make a couple of final changes in the **Security/Users** section for our user, ensure that for **Owned Schemas** & **Membership** that **db_owner** is selected.

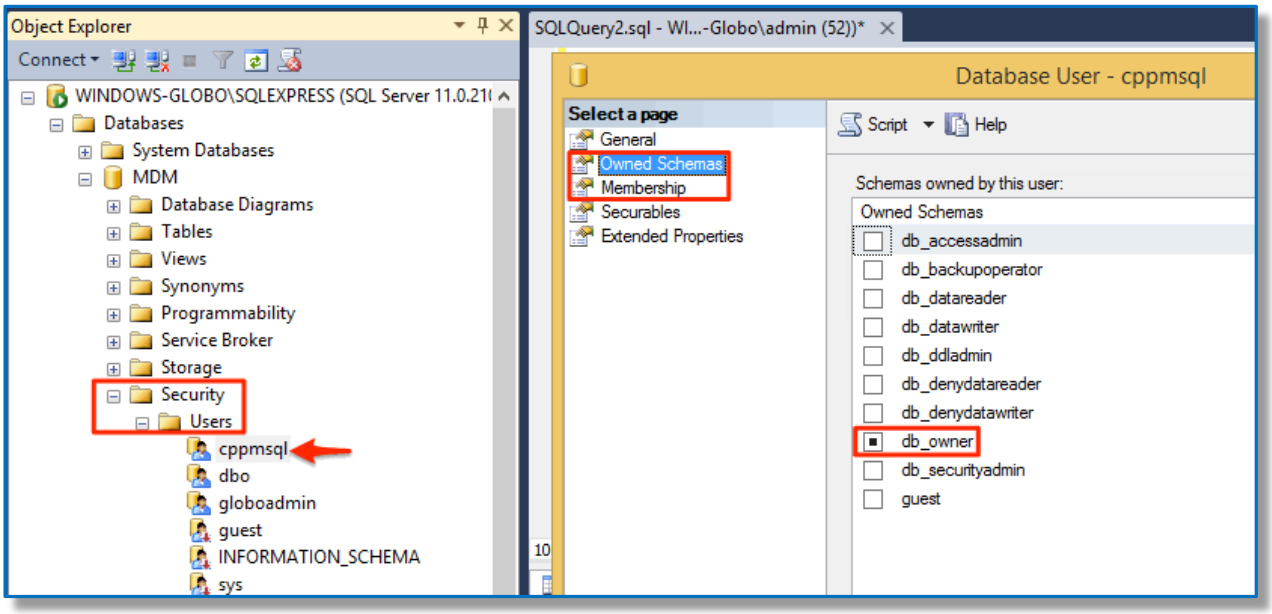


Figure 57 - Check Security setting for user

ClearPass Configuration for Globo GO!

The first thing we have to do on ClearPass is define Globo Go! as an authentication source, go to **Configuration->Authentication->Sources-> Add** provide a name and click Next.

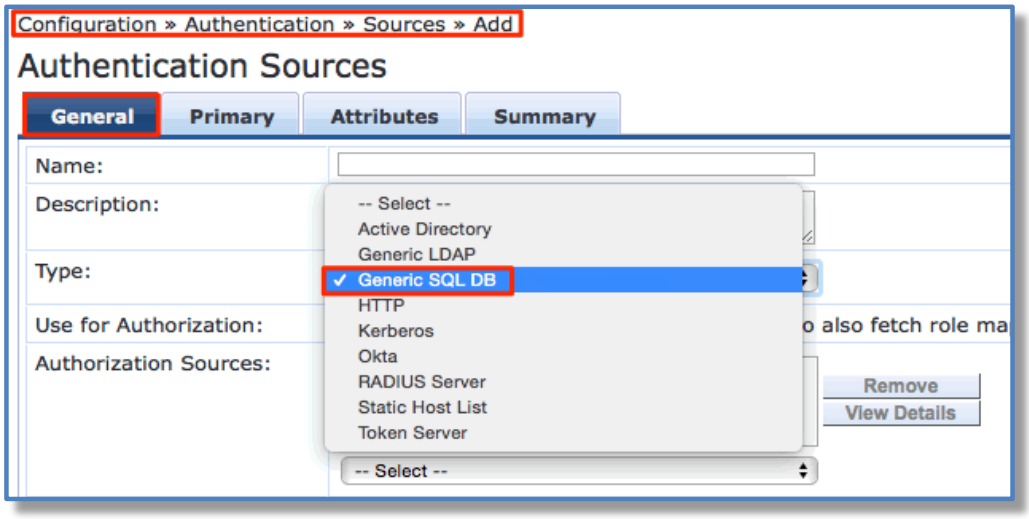


Figure 58 - Adding a Globo as an SQL Authentication source

Next you need to add the Globo GO! Server IP address, the port we will use that we fixed previously, the DB Name [always **MDM**], the **Username/Password** we set up inside of SQL previously and then ensure you select **MSSQL** as the **ODBC Driver**, finally ensure you leave the **Password** as **Cleartext**.

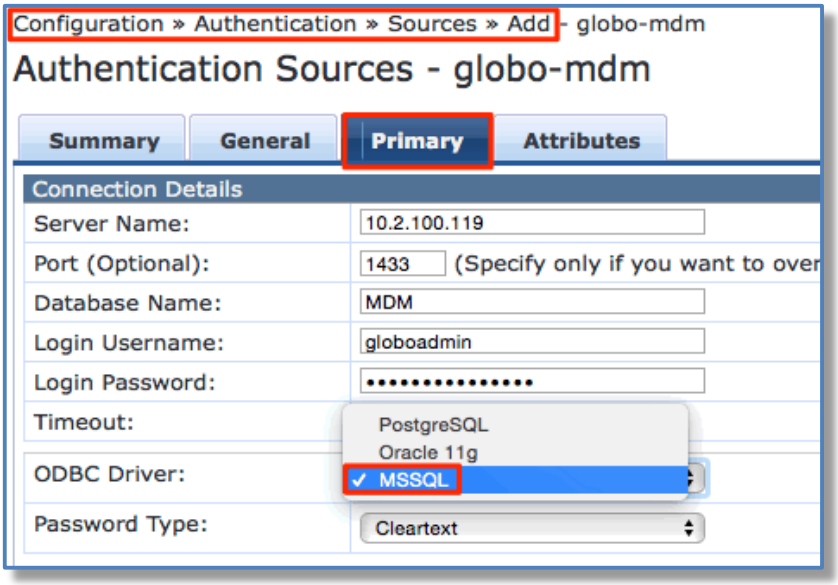


Figure 59 - Defining the Globo Go! SQL Database

The next section is the most critical and prone to errors, so take special care when creating the next few steps. We have provided below two SQL filters which we will use later to process an Authz against a users Authentication. This is a brief list of the two filters many other possible filters can be created once the correct tables structure is understood.

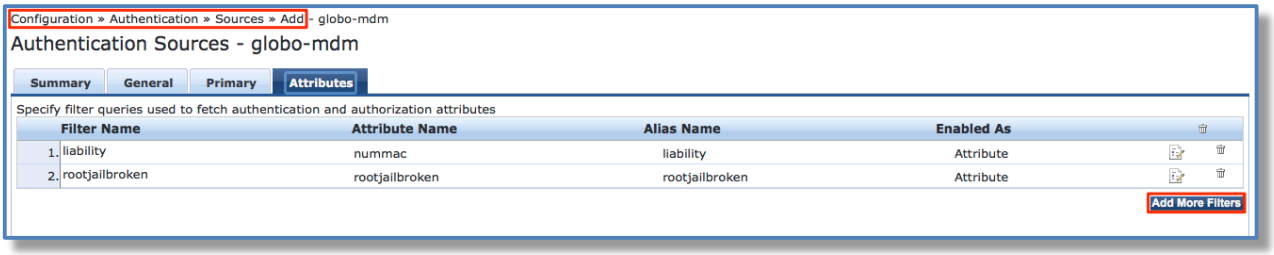


Figure 60 - Summary of the two SQL Filters

The first filter is used to check on whether a device that is authenticating on the network is actually enrolled as a Corporately Owned and enrolled device in the Globo Go! MDM. We use the Mac-Address from the in-coming RADIUS Request to go and make a SQL call. If we find a record i.e. COUNT>0 then the device exists and is Corporately Owned, this is flagged by Liability=2. Ensure you create the filter as defined below.

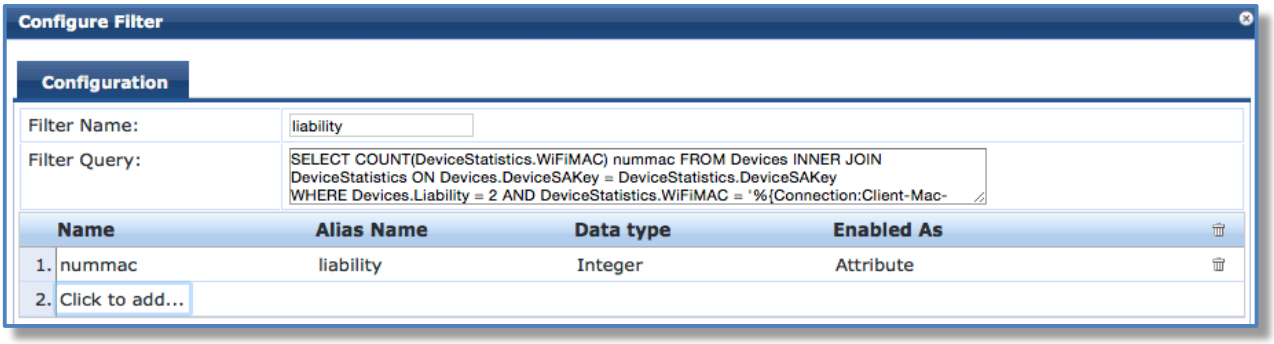


Figure 61 - Filter to check on Device Enrollment and being Corporately Owned

You can copy the below SQL into your Filter Query to minimize errors.

```
SELECT COUNT(DeviceStatistics.WiFiMAC) nummac FROM Devices INNER JOIN
DeviceStatistics ON Devices.DeviceSAKey = DeviceStatistics.DeviceSAKey
WHERE Devices.Liability = 2 AND DeviceStatistics.WiFiMAC =
'%{Connection:Client-Mac-Address-Colon}'
```

Figure 62 - SQL to track device enrollment and Corporate ownership

The second filter is used to check on whether a device that is authenticating on the network is jailbroken and an enrolled device in the Globo Go! MDM. We use the Mac-Address from the in-coming RADIUS Request to go and make a SQL call. If we find a record i.e. COUNT>0 then the device exists and is **NOT** Jailbroken, this is flagged by JailBrokenDevice =0. Ensure you create the filter as defined below.

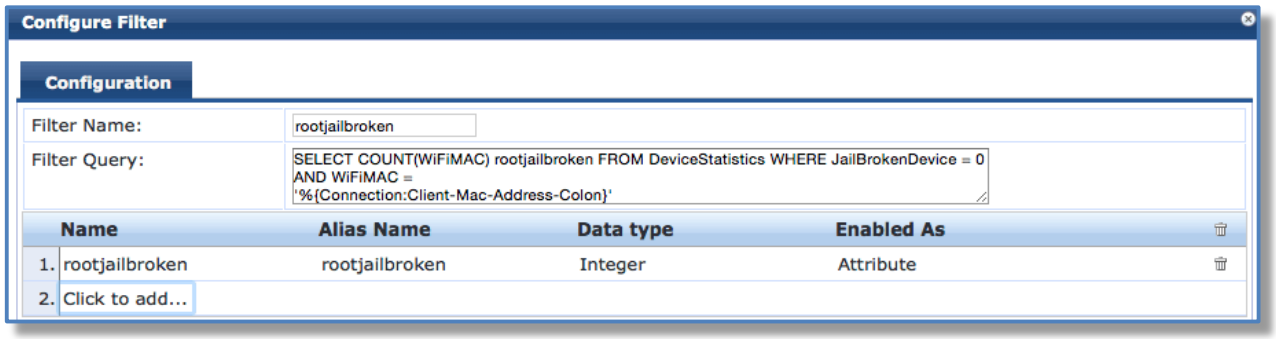


Figure 63 – Filter to check on a device enrollment and not being Jailbroken

You can copy the below SQL into your Filter Query to minimize errors.

```
SELECT COUNT(WiFiMAC) rootjailbroken FROM DeviceStatistics WHERE
JailBrokenDevice = 0 AND WiFiMAC = '%{Connection:Client-Mac-Address-
Colon}'
```

Figure 64 - SQL to track device enrollment and Jailbroken status

Now we have defined the above two example filters we can use and reference these in our role-mapping process shown below to set a role for the user which could then be enforced via a standard enforcement policy.

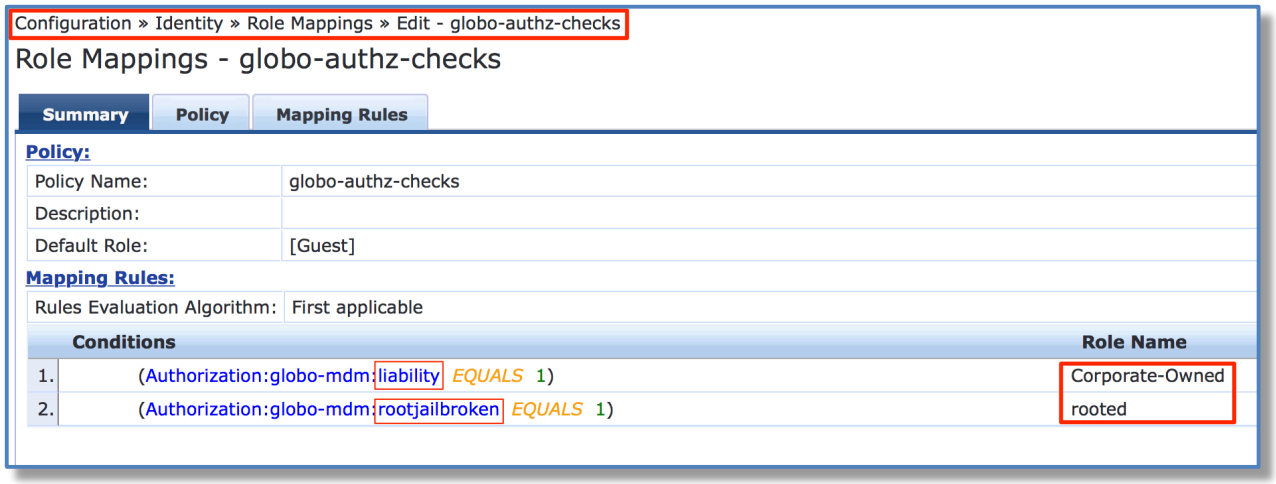


Figure 65 - Role-mapping using the authz results

This completes the section covering ClearPass and Globo Go! MDM integration.

CPPM & MDM/EMM SCEP Setup

This feature introduced in CPPM 6.3 provides for a 3rd party gateway to send Simple Certificate Enrollment Protocol (SCEP) requests to the ClearPass Onboard CA to automate the enrollment provisioning process and leverage certificates for advanced user authentication. Primarily we have tested with EMM vendors as the SCEP client (Proxy).

CPPM SCEP Configuration

Configuring the SCEP Server functionality on CPPM is very simple. We are assuming you already have configured a Certificate Authority (CA) for Onboard. Initially when we added the proxy-enrollment process we provided for just SCEP based enrollment. In CPPM 6.4 we added support for Enrollment over Secure Transport (EST), a new comprehensive and more secure method of obtaining certificates than previous approaches, such as SCEP.

For CPPM 6.4, enable this within the Onboard CA **Guest -> Onboard -> Certificate Authorities**. Take special notice of the SCEP/EST URL that will be used on the SCEP/EST proxy server. Set a strong-shared SCEP/EST password.

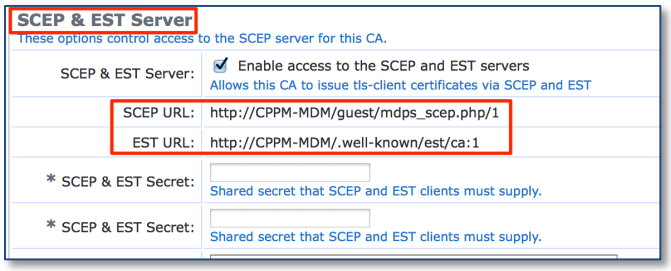


Figure 66 - Configuring SCEP & EST in CPPM 6.4

In releases prior to 6.4 the SCEP server was enabled in **Guest -> Onboard +Workspace -> Initial Setup -> Certificate Authorities.** Take special notice of the SCEP URL that will be used on the SCEP proxy server. Set a strong shared SCEP password.

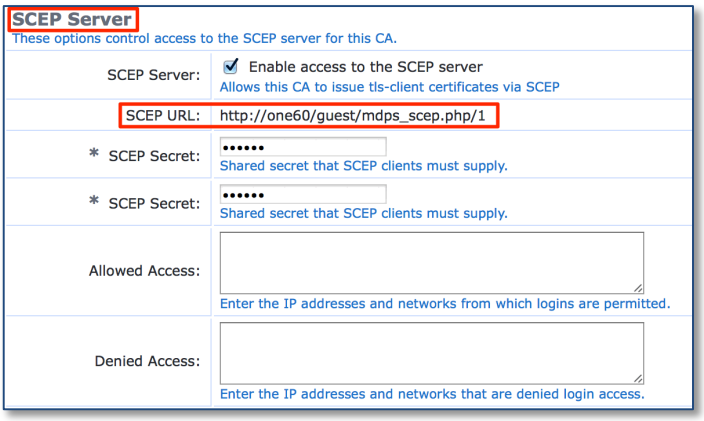


Figure 67 - Configuring SCEP Server in CPPM 6.3

EMM SCEP Configuration

Configuration within the EMM portals differs as vendors have differing frameworks and workflows. We will enhance this section as we document the workflows of other vendors.

AirWatch SCEP Configuration

Creating the SCEP server and template is the first step, this template has to be included in a profile that will be pushed to the managed device. AirWatch has an interesting model for scoping which devices should get which profile (employee owned vs corporate, device type, os version etc). To complete the configuration you must have Admin credentials sign on.

Configure SCEP in AirWatch

From the LHS nav bar, go **Groups & Settings, All Settings, System, Enterprise Integration, Certificate Authorities**…… then you need to add a new CA, this will be the CPPM OnBoard CA you configured previously.

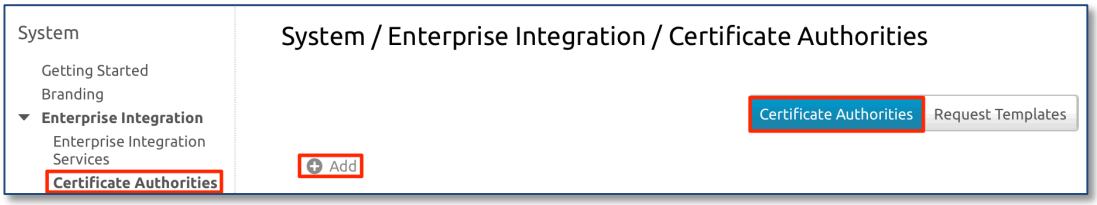


Figure 68 - Adding a NEW CA (SCEP) Server in AirWatch… part1

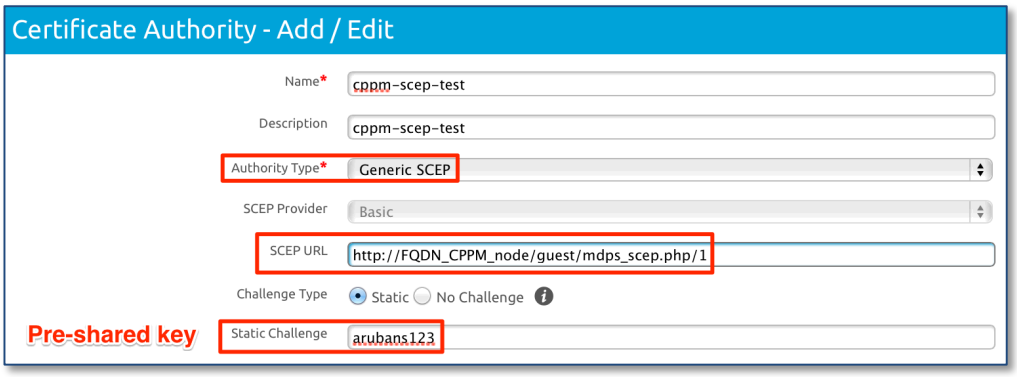
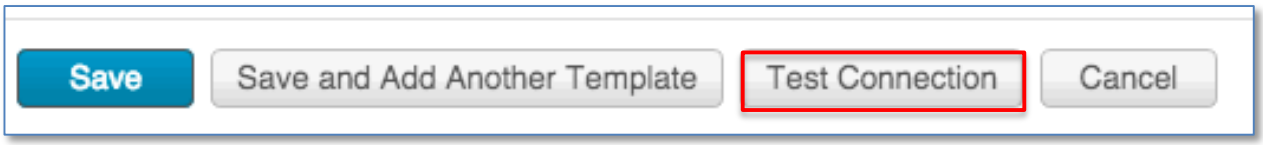


Figure 69 - Adding a NEW CA (SCEP Server) in AirWatch… part2

Note: At the bottom of the configuration screen is an the option to TEST CONNECTION… In our testing this option never worked. See a section later for generating a test scep request.



Next we create a Request Template, notice the highlighted box below on the RHS……

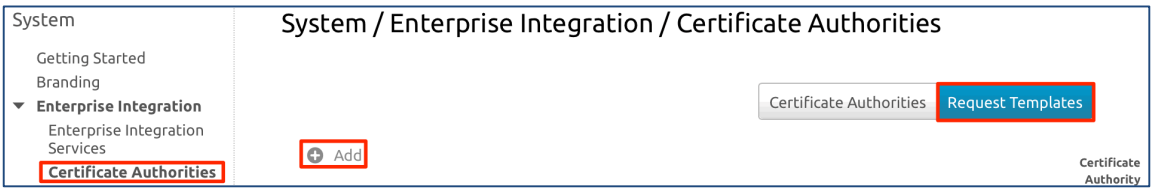


Figure 70 - Adding a NEW Request Template in AirWatch

When creating the Request template, ensure that for the Certificate Authority, you choose from the drop-down the CA you just configured in the previous step. The Common Name in our example is set to use '**EnrollmentUser**', this effectively creates the client-certificate with the CN equal to the user name as shown below.

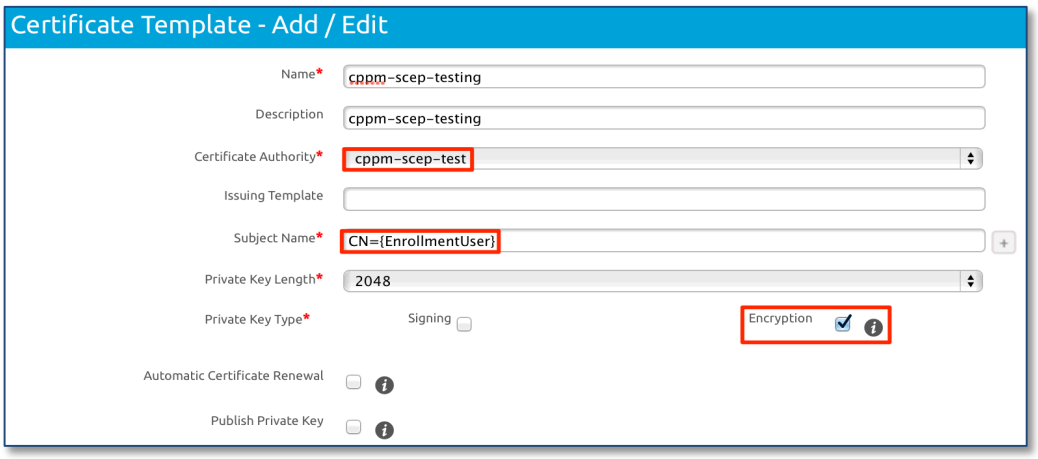


Figure 71 - Setting the Certificate Template to use the Onboard CA and CN=User

After creating the above, we need to create or change our platform profile that will be applied to the managed devices.

<u>Create:</u> If you need to create a new Profile follow these steps, go to **Devices, Profile, List View, Add [Choose relevant device platform]**

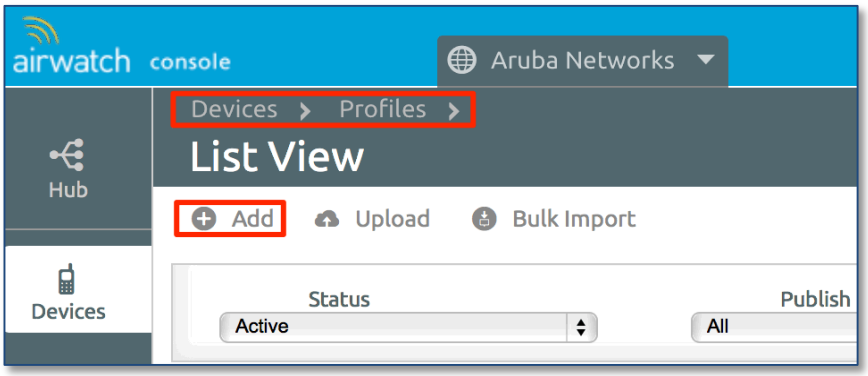


Figure 72 - Adding a platform profile

Choose the required template…..

Figure 73 - Many different Platform templates supported

Depending on the platform you are configuring different options exist based on the template previously choosen. Be aware that a lot of the supported platform templates do not support SCEP. Only iOS, MAC OSX and Windows 8 support SCEP on the AirWatch platform. When configuring SCEP you are able to choose the CA and Certificate templates configured previously. Be sure to select in the 'General' option which you can see in red below the required Smart Groups this SCEP profile will apply to. You may have to create a new Smart Group.

Note: Devices should be added to the correct SmartGroups after/during the enrollment process.

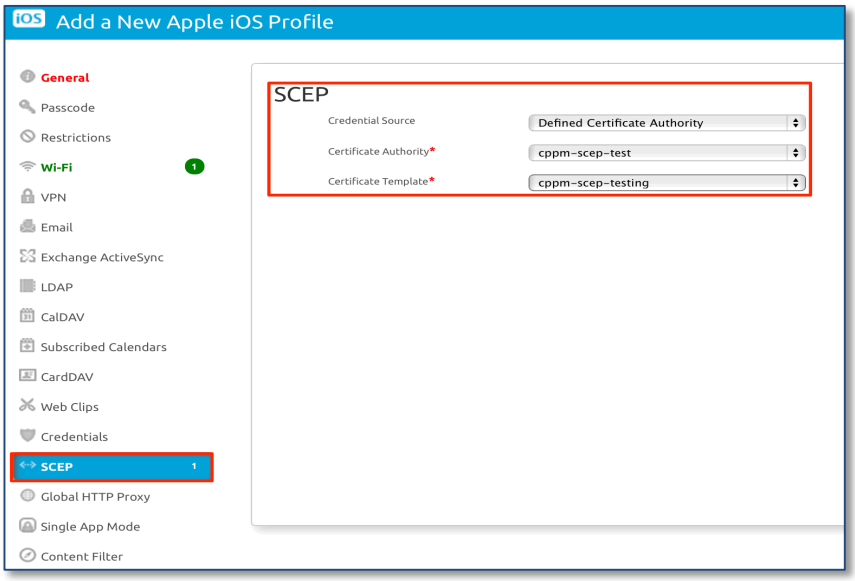


Figure 74 – Setting policy configuration, SSID, Passcode, SCEP Etc.

For the testing we created a Smart Group (ios-scep-test) and assigned a device to this group. This was the group we selected in the 'General' section above. Notice below that once the profile has been saved and applied its shows as pending whilst the SCEP request is serviced by CPPM.

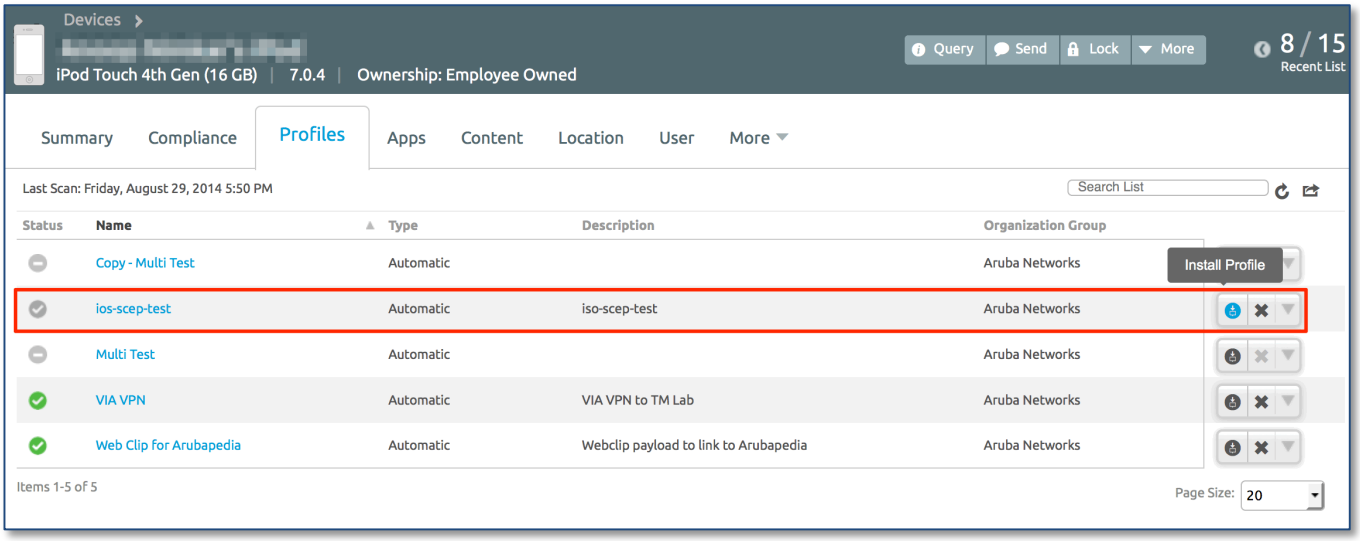


Figure 75 - SCEP request sent to CPPM for processing

On CPPM, you can see that a tls-client certificate was created upon receiving this request from AirWatch. For the testing we created a new CA (SCEP_TEST-CA) within CPPM to handle SCEP.

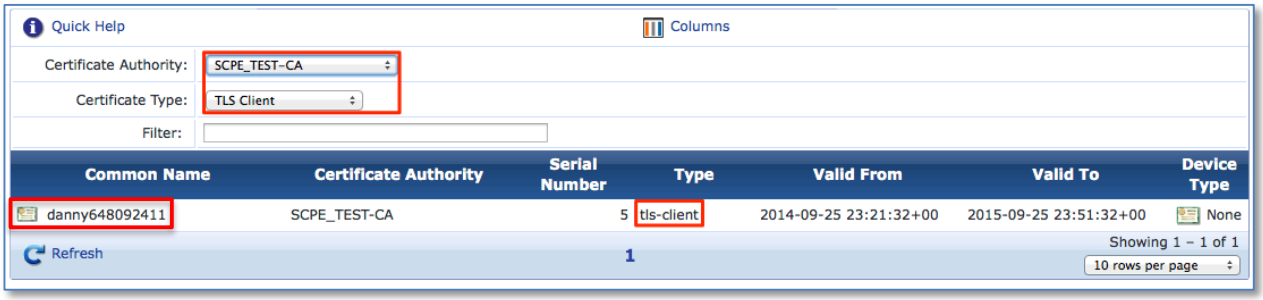


Figure 76 - SCEP request on CPPM, client TLS cert created

Note: Pay attention to the Common Name of the certificate created, "danny648092411" above by the SCEP request. This appears below in the iPAD Device Identity Certificate.

On the iPAD Client we see the device identity certificate installed succesfully…….

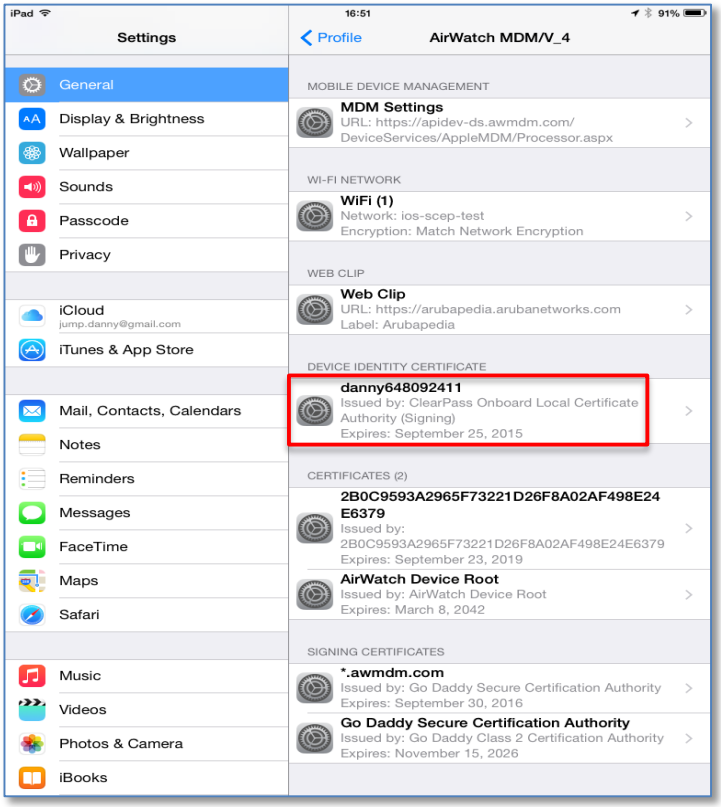


Figure 77 - Certificates etc. installed successfully on the iPad

Note: Take notice of the device identity certificate (danny648092411) above installed on the iPad. This matches the Onboard client cert that was created and shown previously.

Within the AirWatch console you can also see the confirmation messages of the above process go to **Accounts -> Users -> List View [click on your user]** and it will show the activity for this user. Below can be seen the Profile etc. being installed

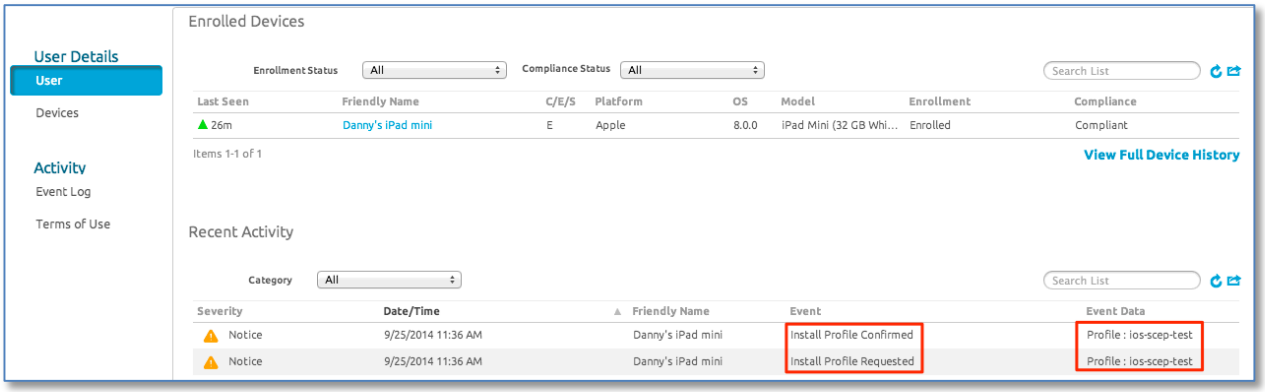


Figure 78 - AirWatch console messages

Airwatch/SCEP-Server/Endpoint Dataflow

The data flow shown below is important to understand. Once the Endpoint has had its SCEP configuration applied it makes a calls to the SCEP Server. The SCEP request comes directly from the Endpoint to ClearPass. The data is returned directly to the device as shown below.

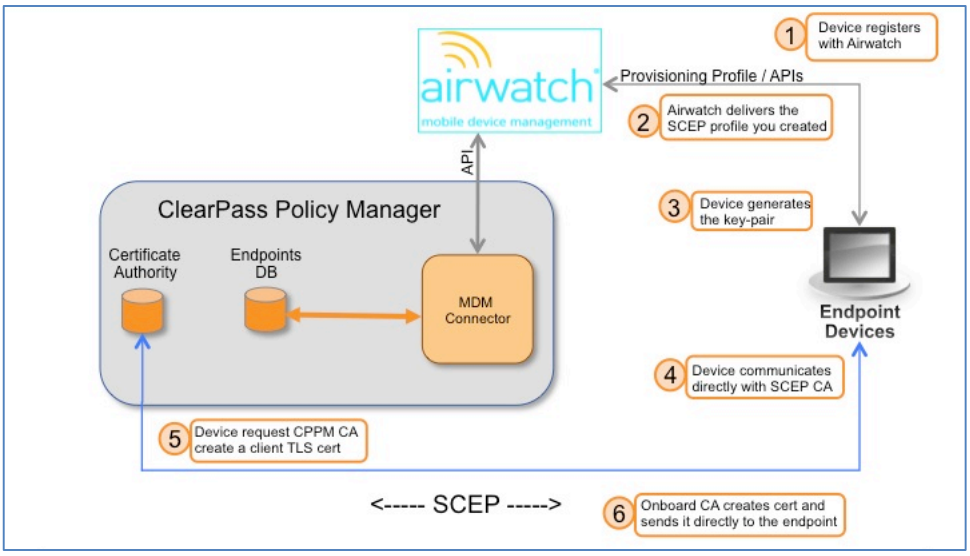


Figure 79 – Airwatch SCEP workflow enrollment with ClearPass CA

Generating a SCEP Test Request in Airwatch

Airwatch provides a very useful tool for generating SCEP request. From the LHS nav bar, go to **Groups & Settings, All Settings, Admin, Troubleshooting, SCEP Certificate Tool** as shown below in the SCEP Cert Tool UI. Ensure you select the correct CA and Certificate Template before you generate the SCEP request to the ClearPass Onboard CA by clicking on the 'Test Certificate Retrieval'. If this is successful then you will see a returned test certificate in at the top of the screen, this can be clearly seen below in **green text**.

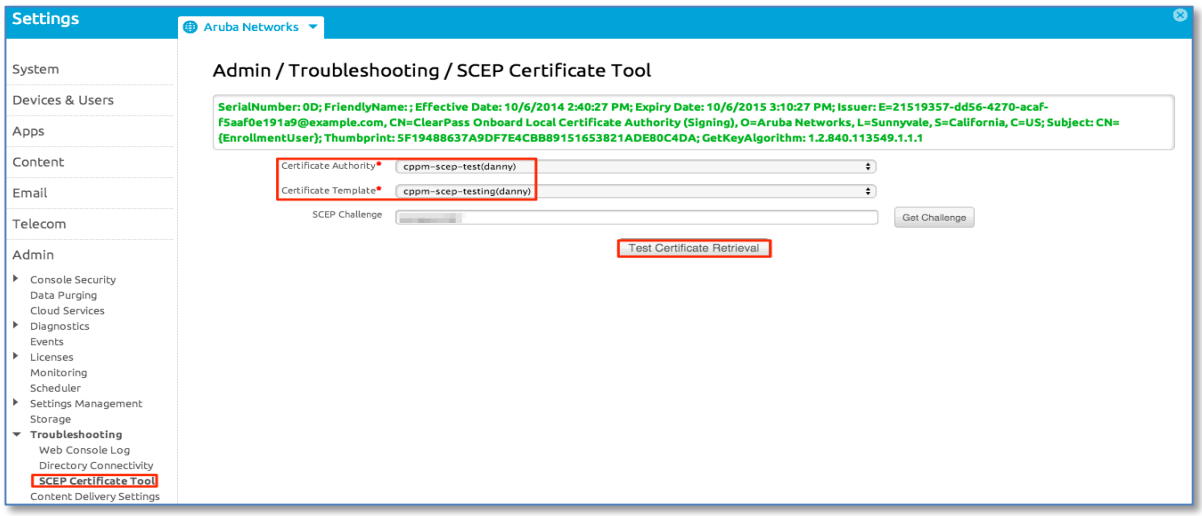


Figure 80 - Generating an Airwatch SCEP test request

MobileIron SCEP Configuration can provision certificates onto the many platforms via SCEP-PROXY. They support SCEP enrollment for iOS, Google Android, Windows Phone 8, and Windows 8.1 RT/Pro (though there are some additional steps for Windows devices).

Note: MobileIron currently does not support EST, we understand it is under investigation.

Different vendors support/implement differing workflows for SCEP enrollment, MobileIron support a couple. One allows the SCEP enabled mobile device to communicate and enroll directly with CPPM whilst the other 'forces' the enrollment through the MobileIron VSP platform. Whichever workflow you take is controlled by whether or not you select the "**Enable Proxy**" check box in the SCEP setting in VSP. If you select Enable Proxy, MobileIron Core will proxy the request to CPPM. If you uncheck that box, the device will attempt to access the server directly. Our advice is to always use the proxy, because you can allow enrollment from outside the network (e.g. 3G/4G/LTE) and not have to expose the SCEP server to the Internet (which is not a recommended design).

Configure SCEP in MobileIron

So to create the SCEP configuration from the VSP console….. Go to **Policies & Configs -> Configurations -> 'Add New' -> SCEP**… then fill in the template as in the next screen shot.

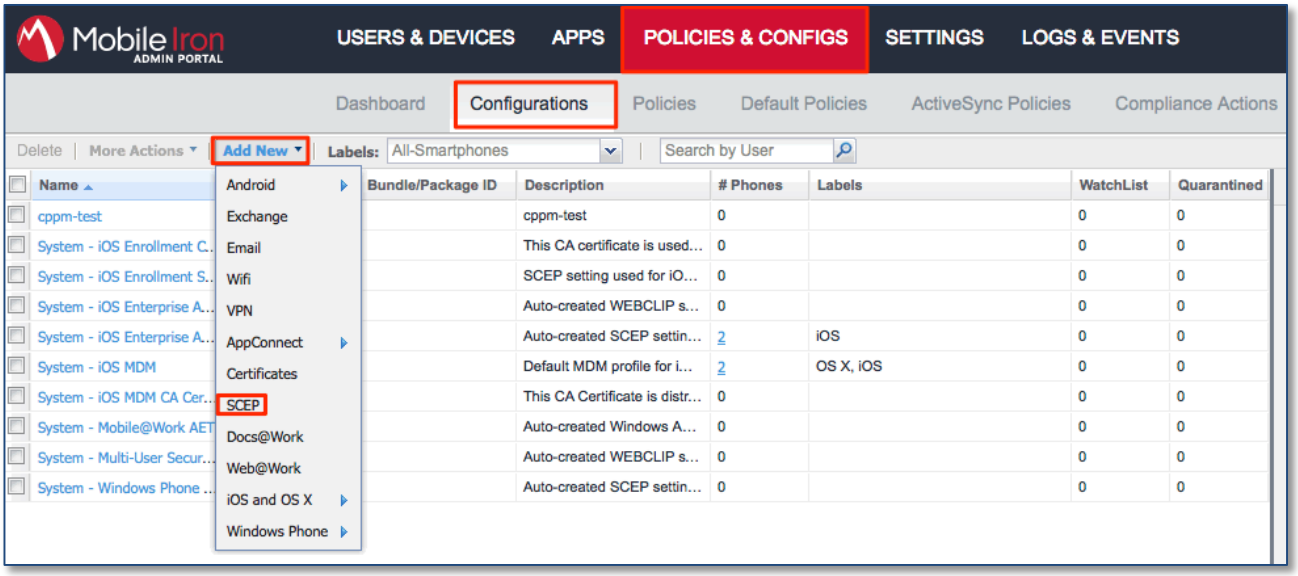


Figure 81 - Configuring SCEP on MobileIron … part1

We have set the below SCEP configuration such that the Subject for the certificate will be the UserID. We can also set a multitude of additional parameters that will be passed in the SCEP request sent to CPPM. Below is a list of the supported variables in MobileIron. In Figure 59 below we have chosen several values to be include in the SCEP request that is sent to CPPM for the creation of the certificate. We selected the Common name (CN) to be that of the User Name, additionally we selected some fields for the Subject Alternate Name (SAN). You can select the CN and SAN fields as required.

SCEP Subject Alternative Name Value
Enter value like $EMAIL$, $USERID$, $FIRST_NAME$, $LAST_NAME$, $DISPLAY_NAME$, $USER_DN$, $USER_UPN$, $USER_LOCALE$, $DEVICE_UUID$, $DEVICE_UDID$, $DEVICE_IMSI$, $DEVICE_IMEI$, $DEVICE_SN$, $DEVICE_MAC$, $USER_CUSTOM1$, $USER_CUSTOM2$, $USER_CUSTOM3$, $USER_CUSTOM4$, $NULL$ or any custom format like $USERID$:$EMAIL$ or $USERID$_$EMAIL$ etc...
At runtime these variables will get resolved in to user values.

Figure 82 - List of attributes available on MI for SCEP request

Save | Cancel

Name: cppm-test

Description: cppm-test

Enable Proxy: ☑
☑ Cache locally generated keys on the VSP
○ User Certificate ◉ Device Certificate

Note 'Device Certificate'

Setting Type: SCEP

URL: http://[redacted]/guest/mdps_scep.php/2

This URL comes from CPPM's CA SCEP configuration

Subject: CN=$USERID$

Subject Common Name Type: User Display Name

Subject Alternative Name Type: RFC 822 Name — Subject Alternative Name Value: $DEVICE_MAC$

RFC 822 Name — Subject Alternative Name Value: $DEVICE_IMEI$

RFC 822 Name — Subject Alternative Name Value: $EMAIL$

None

Key Size: 2048

CSR Signature Algorithm: SHA1

Key Usage: ☐ Signing ☑ Encryption

Finger Print:

Challenge Type: Manual

Challenge: ••••••••••

Confirm Challenge: ••••••••••

Shared secret set on CPPM CA when enabling SCEP server

Issue test certificate: ☑

Save | Cancel

Figure 83 - Configuring SCEP on MobileIron … part2

After clicking on 'Save' MobileIron will attempt to connect to CPPM's CA and create a test certificate, an example of this is as shown below in Figure 61.

If the step of creating the test certificate is successful the following is displayed on the screen….

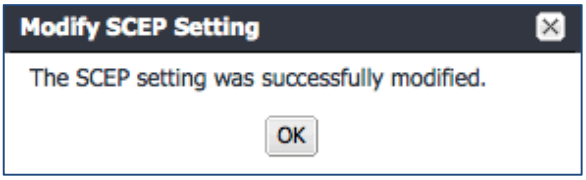


Figure 84 - Creation of SCEP certificate – successful on CPPM

Below is an example of the test certificates created within the Onboard CA from the previous step of generating test certificates. So as not to 'burn' Onboarding licenses with in your CPPM production environment it would be good practice to delete these Test certificate.

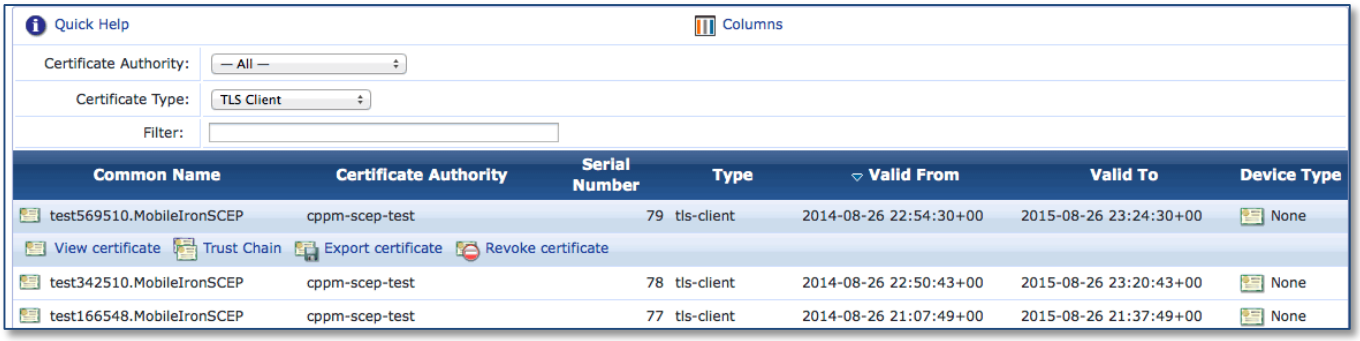


Figure 85 - SCEP test certificate

After you have created the SCEP configuration and proved connectivity between MobileIron and the CPPM SCEP service the remaining steps required relate to Policy and Configuration on the VSP portal for the endpoints.

Note: We are assuming that you have enrolled the device within the MobileIron VSP platform and the EMM profiles are installed on the endpoint.

Setting SCEP policy against EMM endpoint

Next we need to create configuration and policy, you need to utilize 'Labels' to drive this. First lets create a 'Label'… create this under **User & Devices -> Labels**

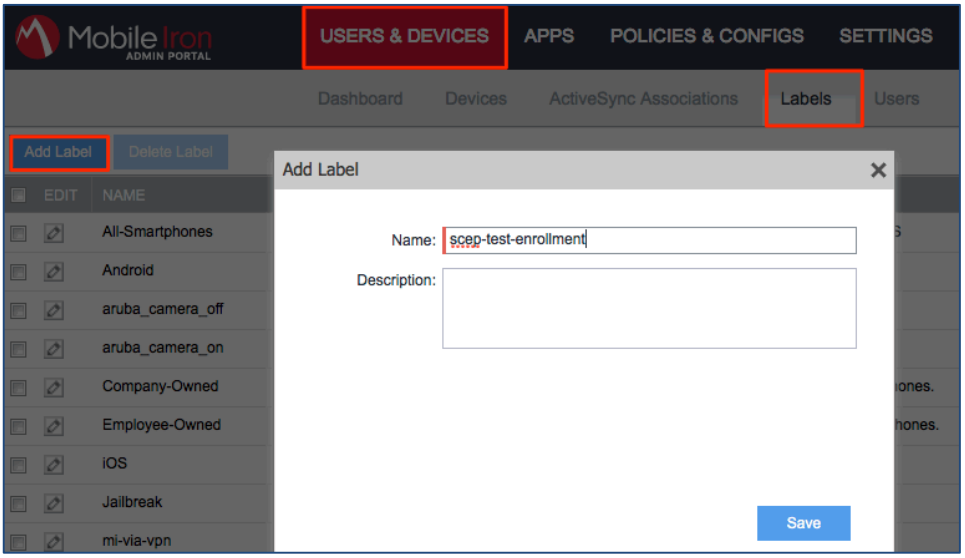


Figure 86 - Creating a MobileIron 'Label'

Next we have to assign the Label just created to the SCEP configuration we previously created. Do this in **Policy & Config -> Configuration -> [Choose the Policy] More Actions -> Apply To Label (not shown) -> [Select the Label]**

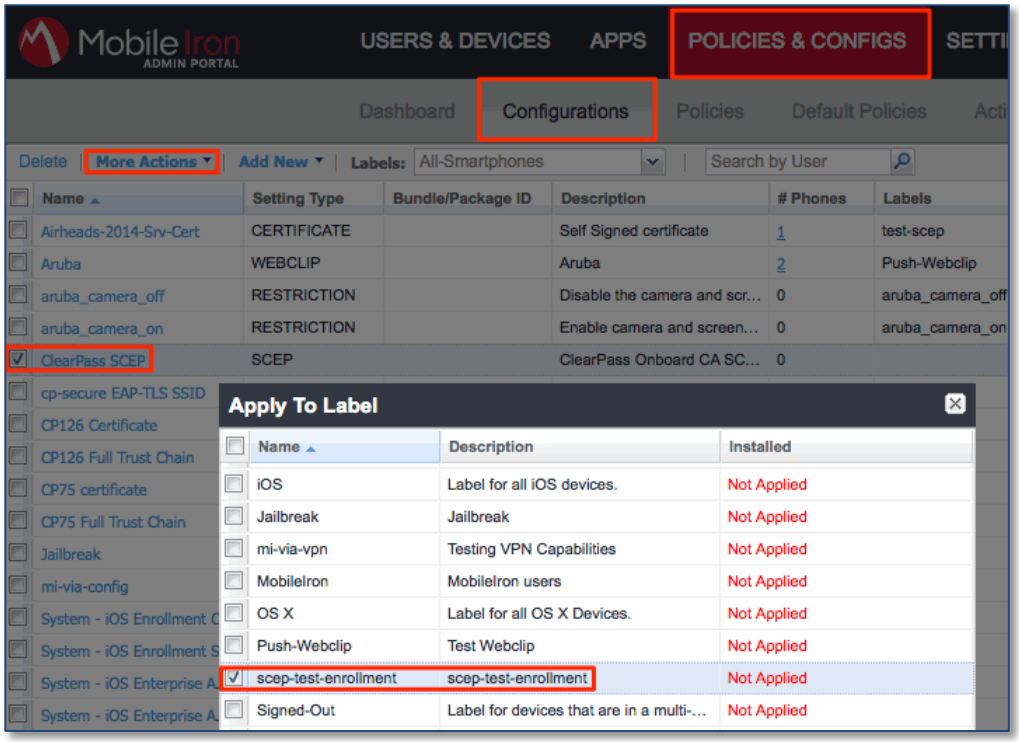


Figure 87 - Adding the Label to the SCEP Policy

Now we have created the SCEP Configuration and associated the new `scep-test-enrollment` label with this configuration we need to finally assign the label to one of the managed devices.

Above you notice on the 'ClearPass SCEP' line, there are no devices assigned to this Profile under '# Phones'. From **Users & Devices -> Devices -> Actions -> Apply to Label (not shown) -> [Select the Label] -> Apply** below you can see that the label we have created is showing currently as 'Not Applied'.

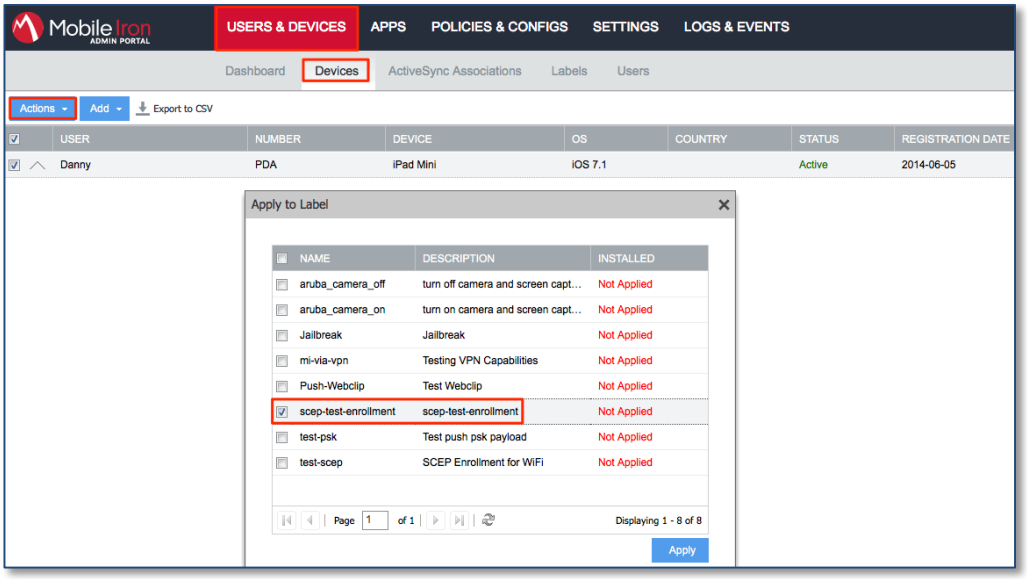


Figure 88 - Applying the Label to an endpoint

A confirmation message should be received on the GUI.

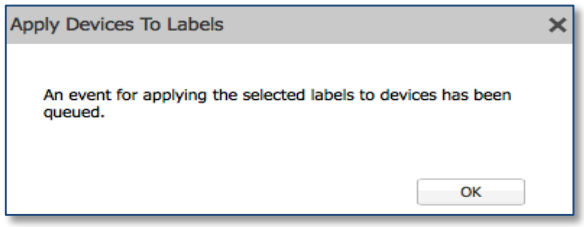


Figure 89 - Label applied and queued for action

At this point, MobileIron will make a request to CPPM's CA via SCEP for the certificate. You can see below the Label '*scep-test-enrollment*' assigned to the Device. This can be reviewed from **Users & Devices -> Devices**

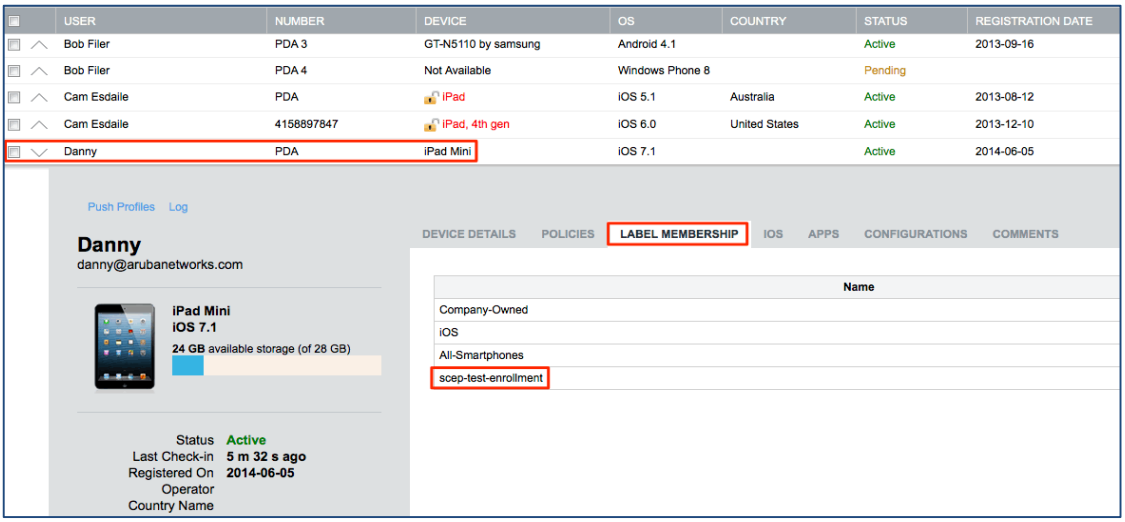


Figure 90 - Labels assigned to the endpoint

You can see the labels above that have been assigned to this endpoint, including the *scep-test-enrollment* label.

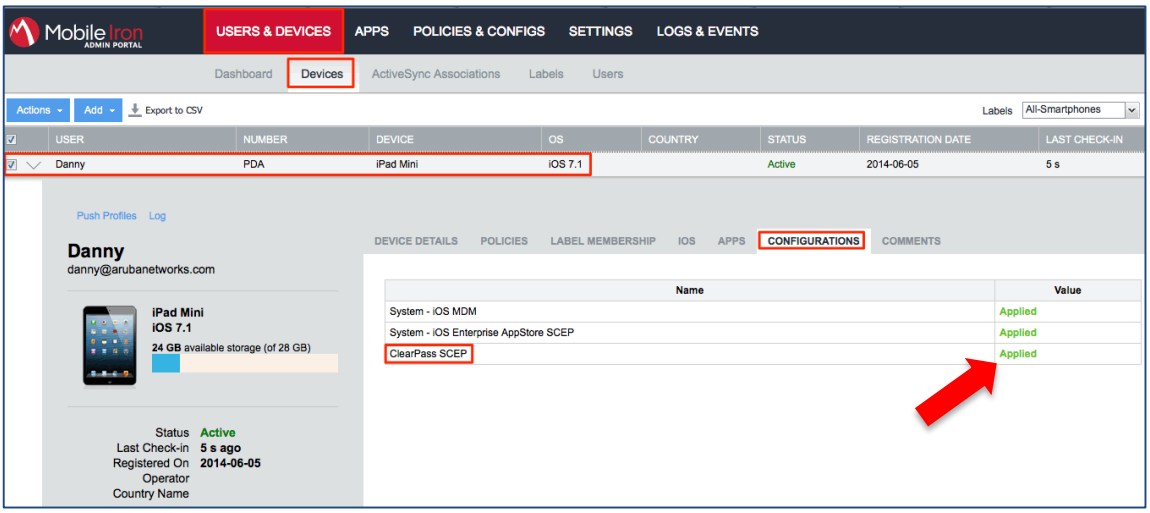


Figure 91 - Configuration applied to the endpoint

Above the endpoint shows the SCEP configuration is 'Applied'.

MobileIron/SCEP-Server/Endpoint Dataflow

Below shows the SCEP workflow for the MobileIron framework. This differs from Airwatch in that the MobileIron solution proxies all request to/from the ClearPass CA.

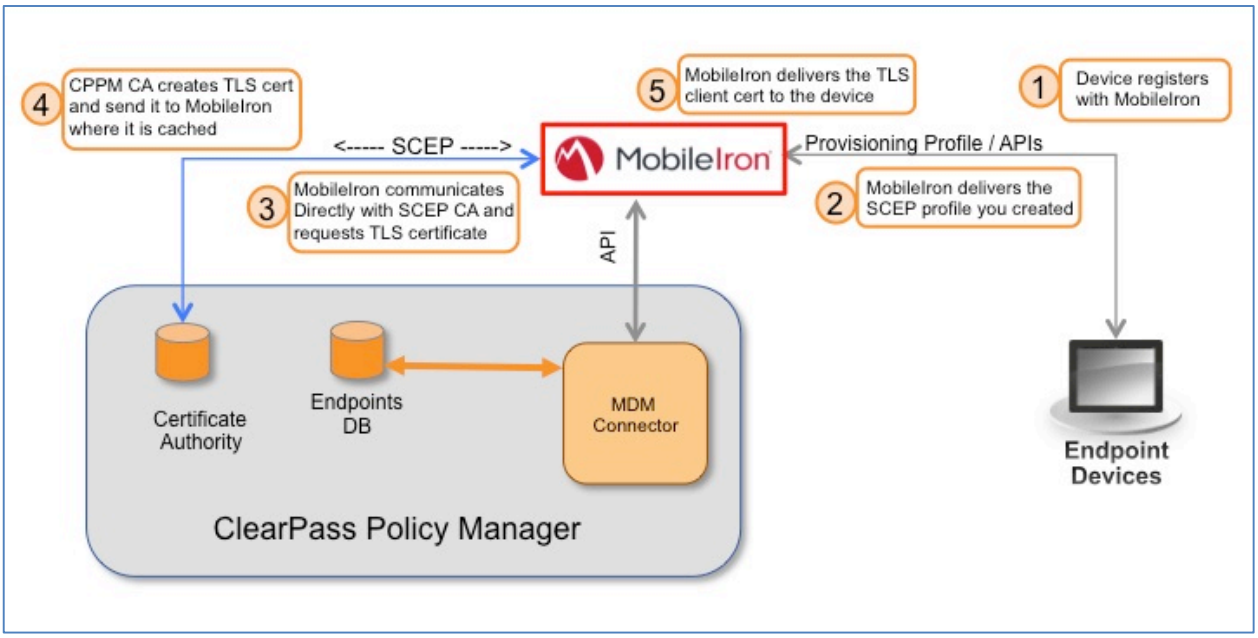


Figure 92 - MobileIron SCEP workflow enrollment with ClearPass CA

Deleting Client TLS Certificates on MobileIron

The certificates that are created and download to the device exist within our eco-system in several places. A copy exist in the CPPM CA, a cached copy also exist in the MobileIron VSP.

We previously discussed deleting the certs in CPPM (https://cppm_fqdn/guest/) then go to **Onboard->Management and Control->View by Certificate.**

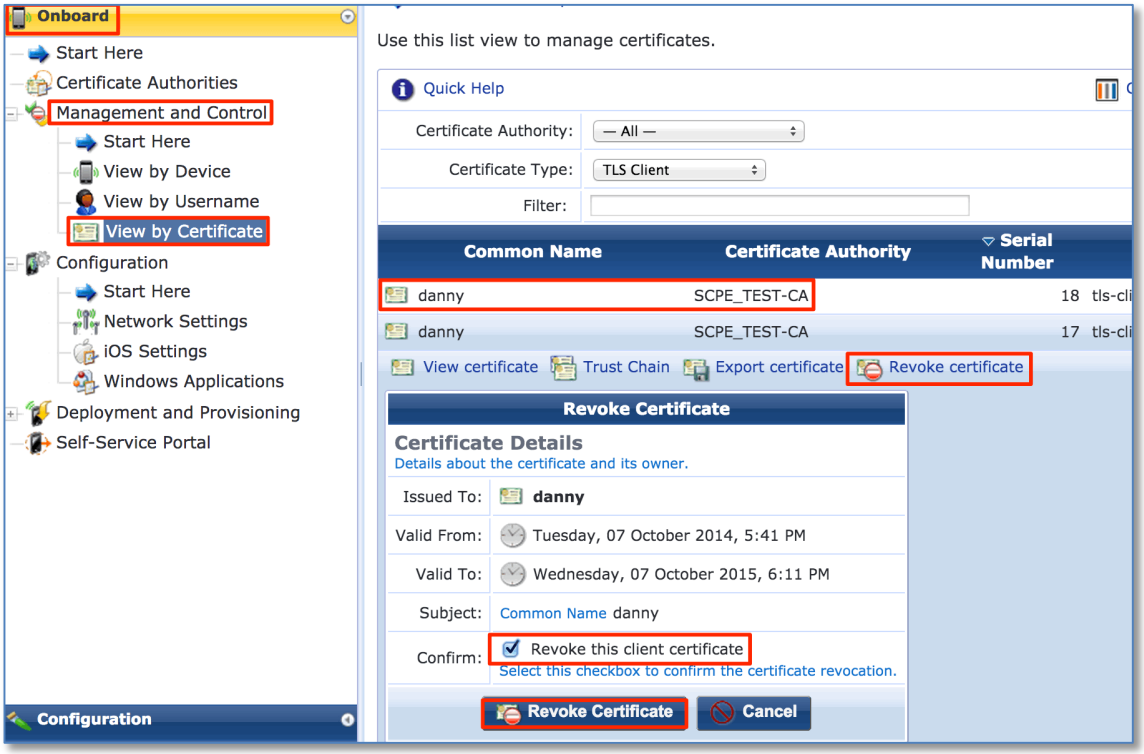


Figure 93 - Deleting client certs in CPPM CA

If you must delete the cached client certificates within MobileIron, follow these steps **Logs and Events->Certificate Logs-> [Select the Cert]** then select '**Revoke**' as shown below.

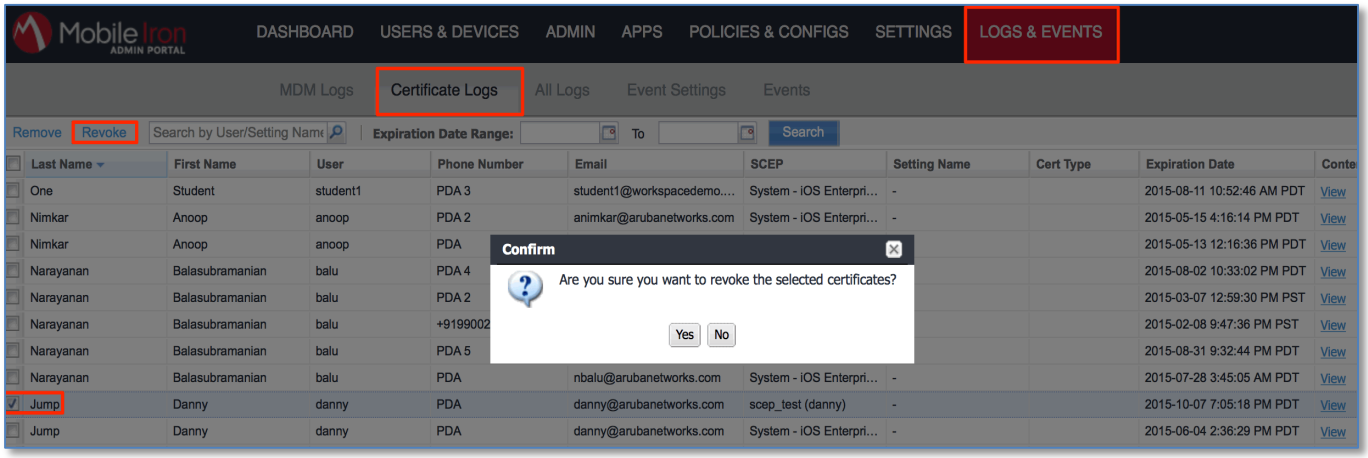


Figure 94 - Deleting Certs in MobileIron

Troubleshooting

Logging information regarding the Endpoint Context servers is available in the Event Viewer.

Go to **Monitoring > Event Viewer**. You should see various messages relating to your configured EMM connectors.

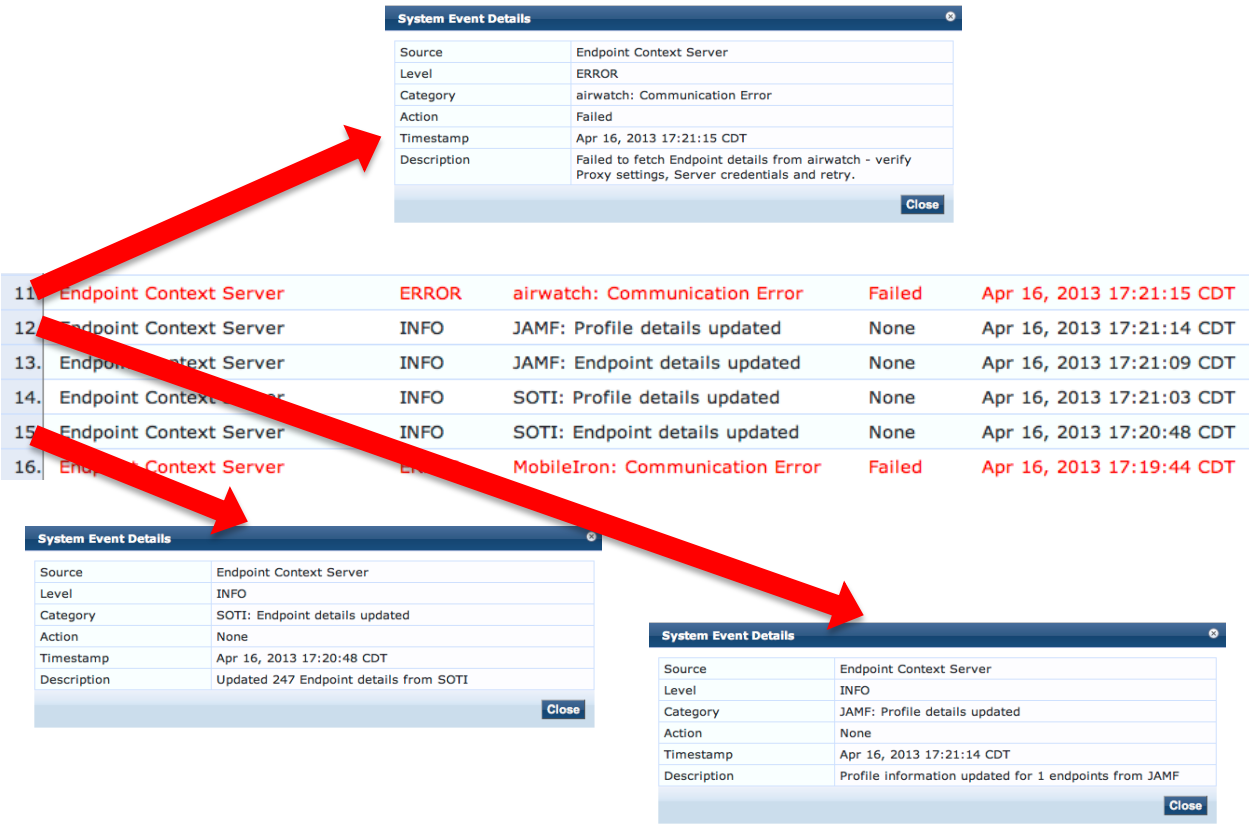


Figure 95 - Event Viewer

Checking Logs files in CPPM

CPPM collects multiple log files that can assist the user in debugging CPPM's EMM integration problem. The most useful of these logs is the **mdm.log** file.

To collect and access this log file is slightly complicated and lengthy, follow these steps….

Under Administration -> Server Manager -> Server Configuration, select your system then '**Collect Logs'**. Once this process has completed you need to download this tar file and open with an appropriate application. For OS-X, **finder** will allow you to extract the file to a folder for analysis. For MSFT Windows multiple applications exist, but a really good free one is **7-Zip** http://www.7-zip.org.

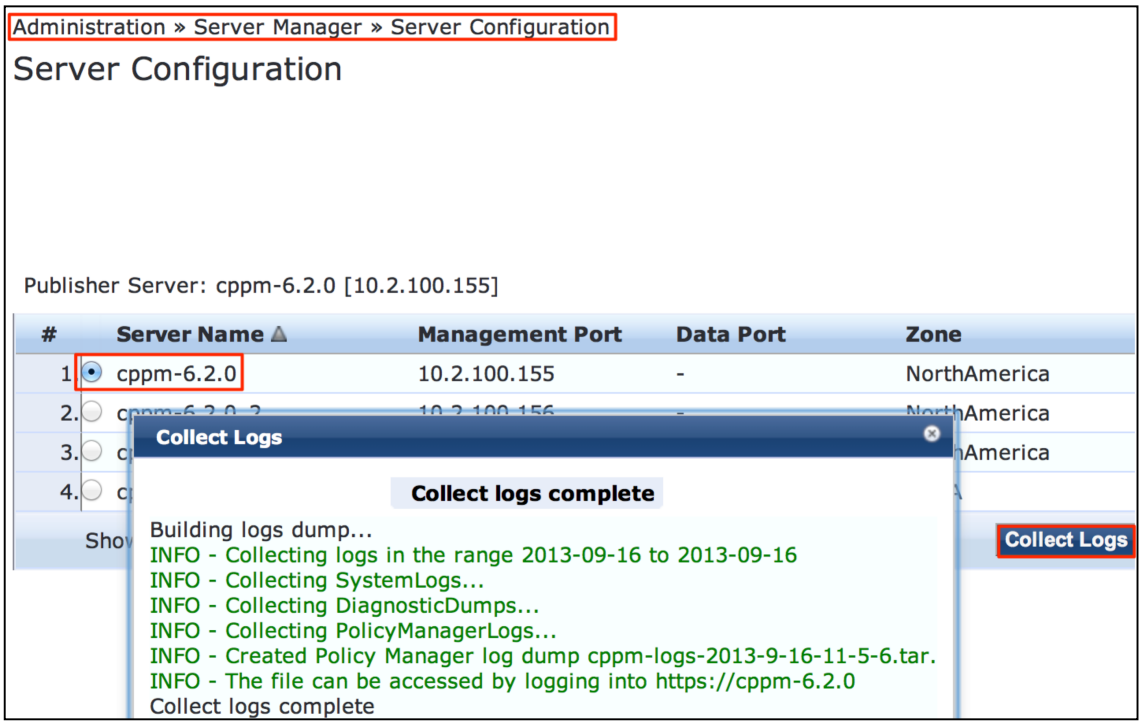


Figure 96 - How to collecting CPPM Logs

After you have opened the archive, the mdm.log file can be found in the following path…

PolicyManagerLogs/async-netd/mdm.log as shown below.

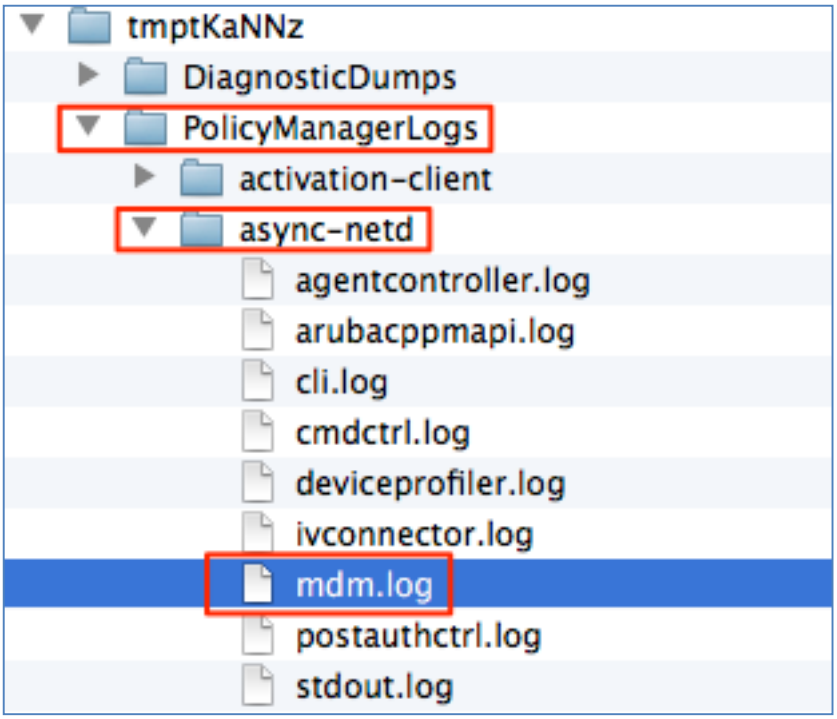


Figure 97 - Where to locate mdm.log file

www.arubanetworks.com

1344 Crossman Avenue
Sunnyvale, CA 94089
Phone: 1-800-WIFI-LAN (+800-943-4526)

General SCEP/EST – Licensing – Q&A….

Anything you configure within the 'Onboard' menu that interact with a device for provisioning, will consume an 'onboard' license. That includes MSFT Active Directory Certificate Services and SCEP/EST server. Therefore EVERY issued certificate will consume and require a license in Onboard.

Caveats/Queries for CPPM SCEP/EST

SCEP/EST is **only** for TLS client certificates (and device identity certificates used for configuration profiles, an internal detail of iOS/OS X over-the-air provisioning).

Q) Is it programmable via API, i.e, Can we revoke certificates via API calls?

A) No, today we do not provide an API interface into the CPPM CA to revoke/disable certificate.

Q) If Onboard CA is being used only to issue certificates via SCEP/EST then how is Onboard expected to know the "device/user attributes"?

A) SCEP signs the certificate request and sends back the result as a certificate - Whatever is in the CSR should be part of the certificate. Onboard will honor the attributes presented in the CSR of a SCEP / EST request so it is critical to ensure that the EMM configured CSR meets your deployment requirements.