Microsoft Intune

ClearPass

a Hewlett Packard Enterprise company

Integration Guide
ClearPass and Microsoft Intune - Integration Guide

Change Log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modified By</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1 &amp; 0.2 &amp; 0.3</td>
<td>June 2016</td>
<td>Danny Jump</td>
<td>Draft checked by D Wilson, M Adjali and Microsoft</td>
</tr>
<tr>
<td>1.0</td>
<td>Oct 2016</td>
<td>Danny Jump</td>
<td>Initial Restricted-Access Published Version</td>
</tr>
<tr>
<td>1.1</td>
<td>Dec 2016</td>
<td>Danny Jump</td>
<td>Initial GA Published Version</td>
</tr>
<tr>
<td>1.2</td>
<td>May 2017</td>
<td>Josh Santomieri</td>
<td>Updates for new extension version (3.0.0)</td>
</tr>
<tr>
<td>2.0</td>
<td>May 2017</td>
<td>Danny Jump</td>
<td>Minor updates from TAC/ERT and new TechNote Template</td>
</tr>
<tr>
<td>3.0</td>
<td>Aug 2018</td>
<td>Arpit Bhatt</td>
<td>Updates for using Extension GUI and Intune extension v4</td>
</tr>
</tbody>
</table>

Copyright

© Copyright 2018 Hewlett Packard Enterprise Development LP.

Open Source Code

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett-Packard Company. To obtain such source code, send a check or money order in the amount of US $10.00 to:

Hewlett-Packard Company
Attn: General Counsel
3000 Hanover Street
Palo Alto, CA 94304
USA

Please specify the product and version for which you are requesting source code. You may also request a copy of this source code free of charge at HPE-Aruba-gplquery@hpe.com.
# Contents

Introduction ................................................................................................................................. 5
What's new in this ClearPass Extension v4 .................................................................................. 5
Software Requirements .................................................................................................................. 6
Installation and Deployment Guide ............................................................................................... 7
Pictorial view of the Integration ................................................................................................... 7
New Extension Support in ClearPass 6.7+ .................................................................................... 8
  Extensions and IP address configuration support .......................................................................... 8
  Extensions and web proxy support ............................................................................................... 8
Configurations Steps ..................................................................................................................... 10
  Step I: Collecting Information from Microsoft to Configure Intune extension ......................... 10
  Step II: Installation and Configuration of Intune Extension using GUI in ClearPass 6.7.X ........ 18
  Step III: Configuring ClearPass Policy Manager .......................................................................... 24
Using data from Intune in a ClearPass Enforcement Policy .......................................................... 26
Appendix A – Additional diagnostics / support ........................................................................... 28
  Extensions Service ..................................................................................................................... 28
  Extension Logs/Debugging .......................................................................................................... 28
  Accessing extension logs using 'Collect Logs' ........................................................................... 29
  Monitoring authorization performance ....................................................................................... 30
  ClearPass authorization throughput guidelines .......................................................................... 30
Appendix B – Lab Example for Authorization with caching ......................................................... 31
Figures

Figure 1: Pictorial view of ClearPass integration with Microsoft Intune and Azure AD .................................................. 7
Figure 2: Extension Framework GUI ......................................................................................................................... 8
Figure 3: Defining the base IP SUBNET and LOCALHOST for the Extensions Framework ........................................ 9
Figure 4: Azure Application registrations ................................................................................................................ 11
Figure 5: Capturing the OAuth 2.0 token endpoint value ....................................................................................... 12
Figure 6: Creating a new application in Azure ......................................................................................................... 13
Figure 7: Creating a new application registration in Azure ...................................................................................... 14
Figure 8: Capturing important data from your Azure application .......................................................................... 14
Figure 9: Setting application permissions – part1 ................................................................................................... 15
Figure 10: Setting application permissions – part2 ................................................................................................. 15
Figure 11: Setting application permissions – part3 ................................................................................................ 16
Figure 12: Creating application clientSecret keys .................................................................................................. 16
Figure 13: Copying the application clientSecret keys ............................................................................................ 17
Figure 14: Extensions Framework GUI ................................................................................................................. 18
Figure 15: GUI Extension Installation ..................................................................................................................... 18
Figure 16: GUI extension search ............................................................................................................................ 19
Figure 17: GUI extension configuration at Install time .......................................................................................... 19
Figure 18: Creating an API Admin user .................................................................................................................. 22
Figure 19: Setting the extension configuration ..................................................................................................... 22
Figure 20: Log validation ......................................................................................................................................... 23
Figure 21: Adding an HTTP authorization source ............................................................................................... 24
Figure 22: Adding HTTP authorization source credentials .................................................................................... 25
Figure 23: Adding HTTP authorization source query string and returned field definitions ................................ 25
Figure 24: Example of an Enforcement Policy utilizing attributes returned from Intune ...................................... 26
Figure 25: Checking on extension service and how to start/stop the service ............................................................ 28
Figure 26: Turning on Debug logging on an extension using GUI ........................................................................ 28
Figure 27: Extension logs location in 'Collect Logs' diagnostic GZ file .................................................................... 29
Figure 28: Monitoring the performance of the authorization process .................................................................. 30
Figure 29: Authorization attributes in Access Tracker ........................................................................................... 31
Figure 30: Role assignment in Access Tracker ....................................................................................................... 31
Figure 31: Debug Extension logs 1 ........................................................................................................................ 32
Figure 32: Debug Extension logs 2 ........................................................................................................................ 33
Introduction

This integration guide covers the setup, configuration, and monitoring of the Microsoft Intune ClearPass Extension within ClearPass. ClearPass Extensions are micro-services running on top of the base ClearPass platform. These micro-services enable Aruba to deliver new features outside of the main software release cycle and facilitate a faster time to market for specific features.

What’s new in this ClearPass Extension v4

In v4, we have added the capability to cache Intune attributes for a configurable time. This would get device attributes from Intune and write it in the Endpoint Repository of ClearPass when the device authenticates for the first time or after the cache period has expired. If a device authenticates again within the cache period, ClearPass would not send the request to Intune Authorization source, rather use the attributes cached in the Endpoint Repository. This helps us reduce the number of authentications that should traverse to Intune residing in the cloud hence reducing the time to authenticate an endpoint and also helps reducing the load on Intune.

<table>
<thead>
<tr>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>v2</td>
<td>Internal release only</td>
</tr>
<tr>
<td>v3</td>
<td>Added support for Ownership as an endpoint attribute</td>
</tr>
<tr>
<td>v4</td>
<td>Cache attributes from Intune for a defined time-frame</td>
</tr>
</tbody>
</table>
Software Requirements

The minimum software version required for ClearPass is 6.7.2. At the time of writing, ClearPass 6.7.5 is the latest available and recommended release. ClearPass runs on a hardware appliance with pre-installed software or as a Virtual Machine under the following hypervisors.

- VMware ESXi 5.0, 5.1, 5.5, 6.0, 6.5, 6.7 or higher
- Microsoft Hyper-V Server 2012 R2
- Hyper-V on Microsoft Windows Server 2012 R2

Hypervisors that run on a client computer such as VMware Player are not supported.

Microsoft Intune can manage the following device platforms:

- Apple iOS 9.0 and later
- Google Android 4.4 and later (including Samsung KNOX SDK 4.0 and higher)
- Google Android for Work (requirements)
- Windows Phone 8.1 and later
- Windows 8.1 RT
- PCs running Windows 8.1
- PCs running Windows 10 (Home, Pro, Education, and Enterprise versions)
- Devices running Windows 10 IoT Enterprise (x86, x64)
- Devices running Windows 10 IoT Mobile Enterprise
- Windows Holographic Business
- Mac OS X 10.11 and later

Microsoft maintains an up to date version of this list located here:

Installation and Deployment Guide

The document assumes your ClearPass environment is already configured and operational. If you require assistance with basic deployment refer to the following deployment guide located here:


Pictorial view of the Integration

The diagram below shows an overview of the components and how they interact together.

*Figure 1: Pictorial view of ClearPass integration with Microsoft Intune and Azure AD*
New Extension Support in ClearPass 6.7+

With the release of 6.7, several new features have been added to enhance the functionality of the extension framework. Previously, all extension installation and operation tasks required use of the API Explorer to interoperate with the Extension and the underlying framework. This functionality has been exposed with a new GUI in the Guest and is shown below, Administration -> Extensions.

Extensions and IP address configuration support

The other major additions in the 6.7 release are the ability to define the extension framework base IP network and statically define the IP address of the individual extensions. The latter being useful when deploying extensions in a cluster and the requirement for a fixed IP address for the same extension across a cluster regardless of which ClearPass node or nodes it is installed on.

Extensions and web proxy support

Prior to 6.7 support for web proxy was limited to the installation of the extensions. Starting in ClearPass 6.7, extensions now support communications with 3rd parties via a web proxy. This adds incremental web proxy functionality. If a web proxy is defined in ClearPass Policy Manager, then an extension will use that configuration.

Note that the Policy Manager web proxy configuration is ONLY read by the extension at installation time. If the web proxy configuration is changed in Policy Manager, then the extension must be re-installed so the new settings are re-read and bonded to the extension.

Figure 2: Extension Framework GUI
Configuring the base Extension IP subnet, is defined within Policy Manager as shown below under Administration->Server Manager->Server Configuration [chose your node] Service Parameters [ClearPass system service]. The default is 172.17.0.1/16, this address is the non-routed address of the ClearPass node itself. The IP addresses range for the extensions are based upon the network prefix used.

Note that the subnet defined here for the extension framework must be one of the following 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16.

Figure 3: Defining the base IP SUBNET and LOCALHOST for the Extensions Framework

Note that changing the extension base IP address will require the extension service to be restarted.

Changing the “Extensions Network Address” range is necessary if either the MGMT or DATA interface are also using an address in the extension default range of 172.17.x.x/12. Set the new network address range as needed and restart the extension service for this to take effect.
Configurations Steps

There are primarily 3 steps involved in getting this Integration configured.

- Collecting Information from Microsoft to Configure Intune extension
- Installation and Configuration of Intune Extension using GUI in ClearPass 6.7.X
- Configuration of ClearPass Policy Manager to use Intune as an Authorization source

**Step I: Collecting Information from Microsoft to Configure Intune extension**

Below we will cover the process of adding a ‘ClearPass App’ into Azure as an application and enabling the necessary application level permissions. Think of this as the gateway between the ClearPass on-premises environment and Microsoft Intune.

It is assumed you have your Intune/Azure environment already setup and configured. The setup of these environments is beyond the scope of this TechNote.

In order to complete the integration, you need to collect multiple pieces of information from Intune and the Azure platform that are required to allow us to complete the extension configuration. The goal is to collect information to complete the highlighted attributes below in red:

```json
{
    "tokenEndpoint": "<tokenEndpoint>",
    "tenantId": "<tenantId>",
    "clientId": "<clientId>",
    "clientSecret": "<clientSecret>",
    "resourceUri": "https://api.manage.Microsoft.com/",
    "apiVersion": "1.1",
    "verifySSLCerts": true,
    "enableEndpointCache": {introduced in v4},
    "cacheExpirationMinutes": {introduced in v4},
    "cppmUserName": "{CP user}",
    "cppmPassword": "{CP Password}",
    "logLevel": "INFO"
}
```

To start, open up your favorite text editor, and copy and paste the above text block into it. You'll be editing several lines for this JSON payload.

The first piece of information you need to update is the “tokenEndpoint”. This is the URL that ClearPass uses to create OAuth 2.0 Tokens that provide access to Azure Active Directory and Graph services.

To get the “tokenEndpoint” value, first log into the Azure Portal. Point your browser to https://portal.azure.com. Log in using your Intune Tenant Admin account. We assume here you have already identified and configured at least one of your Intune accounts with Administrator rights. You can see below where we've logged in with a “onmicrosoft.com” account.
You may have to accept permissions for the account to use the API Explorer features.

Once logged in, open “Azure Active Directory” and select “App Registrations”.

In “App Registrations”, click on the “Endpoints” menu option to view your Azure endpoints. The relevant links are highlighted below.

**Figure 4: Azure Application registrations**
From the list of endpoints, copy the OAUTH 2.0 TOKEN ENDPOINT value. This is the value you will use as the "tokenEndpoint" in the configuration.

*Figure 5: Capturing the OAuth 2.0 token endpoint value*
Paste the copied endpoint URL in your ID string into the `tokenEndpoint` configuration item.

Next, we need the “`tenantId`” value. To get this, simply copy out the ID portion of the OAuth 2.0 Endpoint. For example, our token endpoint is, https://login.microsoftonline.com/6a02bb69-c703-4cac-8db3-20414baabbcc/oauth2/token

From this URL, the highlighted portion is your Tenant ID. Copy this value into the `tenantId` setting of your configuration.

If you already have an Intune Application Registration in Azure Active Directory, you may use that for the rest of the configuration. If you do not have an Application registered in Azure Active Directory, follow the following steps to create one.

These next steps will be used to collect the `clientId` and `clientSecret` settings.

The next step is to create a new App Registration in Azure Active Directory. This is done from https://portal.azure.com. You must login with an account that has Administrative access to Azure Active Directory and Intune.

Once logged into the Azure Portal, navigate to Azure Active Directory, select “App Registrations” and then click on “New application registration”, as shown below.

*Figure 6: Creating a new application in Azure*
The next step is to create a new application registration. We suggest using the name ClearPass, or something that will clearly identify what the application registration is for. The application type should be set to “Web app / API” and Sign-on URL should be set to a valid URL. After entering your settings, click on Create.

*Figure 7: Creating a new application registration in Azure*

Copy the Application ID: The **Application ID** is the value required for the **clientId** configuration in the extension. You can copy and paste that value to your extension configuration now.

*Figure 8: Capturing important data from your Azure application*
Next, set the required permissions for the App Registration. To do this click on the “**Required permissions**” option available under “**Settings**”. Next select “**Add**”, then “**Select an API**” finally followed by “**Microsoft Intune API**”. Once you have completed that, click on **select** to create the permissions.

**Figure 9: Setting application permissions – part1**

![Figure 9](image)

After clicking **select**, you must enable access to **Get device state and compliance information from Microsoft Intune** then click **Select** followed by **Done**. Your permissions will now be added.

**Figure 10: Setting application permissions – part2**

![Figure 10](image)
The next step is to grant access for your created application to access the APIs. For this click on “Grant permissions” and select “Yes”.

Please Note this is an important step which was missing from the previous version of this document. Kindly ensure you follow this step. If not, ClearPass will not have the necessary privileges to access Intune APIs.

**Figure 11: Setting application permissions – part3**

The next and final step is to capture the “clientSecret”, this currently is a fixed value and maps to the registered Microsoft Intune ClearPass Extension.

When you register the Azure AD (AAD) App, the “*“ will be displayed, you must capture it at this time as it can't be displayed in the future, this is covered below in the following Azure configuration. Follow these steps carefully.

After setting permissions, navigate back to the Application settings and select “Keys”. In the Keys settings, enter a key description. Use something appropriate to identify the keys for Intune. Then select the duration, we recommend “Never Expires” else you will be forced to update the extension configuration when the key expires.

**Figure 12: Creating application clientSecret keys**
After entering your desired information, click “Save”. This will save your settings and generate the `clientSecret`. Copy the “value” to the `clientSecret` setting in the Intune Extension configuration.

**Figure 13: Copying the application clientSecret keys**

Remember to save these keys, as the warning above shows, once you exit this screen you are unable to see the keys again.

Finally, you can easily build the string for “resourceURI” line (if needed). It should simply be `https://api.manage.microsoft.com`.

These three remaining lines are unchanged and should only be modified if directed by Aruba TAC.

- "verifySSLCerts": true
- "apiVersion": "1.1"
- "logLevel": "INFO"

The `apiVersion` above refers to the Microsoft Intune API version, **not** the ClearPass Extension version.
Step II: Installation and Configuration of Intune Extension using GUI in ClearPass 6.7.X

Starting in ClearPass 6.7, a Graphical User Interface (GUI) was introduced to make the process of interacting with the extension framework easier. To access the extension GUI, from the Guest System, under Administration find the Extension User Interface as shown below.

**Figure 14: Extensions Framework GUI**

![Extensions Framework GUI](image)

From here, click on “Install Extension”, and the search box below appears.

**Figure 15: GUI Extension Installation**

![GUI Extension Installation](image)

Starting 6.7, in a cluster environment an extension can be installed on the subscriber nodes directly.
Enter the Store-ID **b163dcd1-227c-4282-b671-4fbea8ab545d** and click on 'Search'. See the example below:

**Figure 16: GUI extension search**

![GUI extension search](image)

Click on the extension and then the **Install** option, and if necessary, set the IP address. Note it can be set later if required. Do not select the option to start the extension yet.

**Figure 17: GUI extension configuration at Install time**

![GUI extension configuration at Install time](image)
After the extension has been installed, edit the extension configuration as necessary. Notice the options to **Start**, **Delete**, **Reinstall** or **Show Logs** and the option to review and set the extension configuration.

The default configuration used for extension is below:

```json
{
    "tokenEndpoint": "https://login.windows.net/{TENANT_ID}/oauth2/token",
    "tenantId": "{TENANT_ID}",
    "clientId": "{CLIENT_ID}",
    "clientSecret": "********",
    "resourceUri": "https://api.manage.microsoft.com/",
    "apiVersion": "1.1",
    "enableEndpointCache": false,
    "cacheExpirationMinutes": 30,
    "cppmUserName": "{ADMIN_USER}",
    "cppmPassword": "********",
    "verifySSLCerts": true,
    "logLevel": "INFO"
}
```

Configure the values for `tokenEndpoint`, `tenantId`, `clientId`, `clientSecret` obtained from **Step I**.

The `cppmUserName` and `cppmPassword` should be an API Administrator account.

The values for `enableEndpointCache` and `cacheExpirationMinutes` varies in a customer environment and needs to be determined based on the latency to query Intune.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Values/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>tokenEndpoint</td>
<td>URL used by ClearPass to create OAuth 2.0 Tokens to get access to Microsoft Intune services</td>
<td><code>https://login.microsoftonline.com/XXXXXXXXX-b3e5-12ab-34cd-ZZZZZZZZZZZ/oauth2/token</code></td>
</tr>
<tr>
<td>tenantId</td>
<td>Azure Tenant Id for the instance</td>
<td><code>XXXXXXXXX-b3e5-12ab-34cd-ZZZZZZZZZZZ</code></td>
</tr>
<tr>
<td>clientId</td>
<td>The API Client ID configured for the Intune instance</td>
<td><code>123abcde-45fg-67hi-jk89-1234567890a</code></td>
</tr>
<tr>
<td>clientSecret</td>
<td>The Client Secret that corresponds to the clientId</td>
<td><code>XyzYZ123456-aabbccddee123-abcd12ghe12=</code></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>resourceUri</td>
<td>The Intune Resource URI. This is generally the same for all configurations</td>
<td><a href="https://api.manage.microsoft.com/">https://api.manage.microsoft.com/</a></td>
</tr>
<tr>
<td>apiVersion</td>
<td>The version of the Intune API to use. This should not be changed unless you have a specific need.</td>
<td>1.1</td>
</tr>
<tr>
<td>enableEndpointCache</td>
<td>Allows ClearPass to use cached attribute values from Intune for the configured duration</td>
<td>true/false</td>
</tr>
<tr>
<td>cacheExpirationMinutes</td>
<td>Integer Value in Minutes. The field holds true if the enableEndpointCache is set to true</td>
<td>Default : 30 mins</td>
</tr>
<tr>
<td>cppmUserName</td>
<td>The attributes fetched are written into the Endpoint Repository leveraging the APIs. A user account with the privilege level “API Administrator”</td>
<td>API Administrator user</td>
</tr>
<tr>
<td>cppmPassword</td>
<td>Password for the account</td>
<td>API Administrator password</td>
</tr>
<tr>
<td>verifySSLCerts</td>
<td>Should the extension validate SSL certificates</td>
<td>true/false</td>
</tr>
<tr>
<td>logLevel</td>
<td>The logging level the extensions should use</td>
<td>DEBUG, INFO, WARN, ERROR</td>
</tr>
</tbody>
</table>
The fetched attributes from Intune need to be written into the endpoint repository leveraging the REST APIs. The ClearPass API Administrator account can be created from ClearPass Policy Manager under Administration > Users and Privileges > Admin Users. Click on “Add”. A user with the following privileges needs to be created.

**Figure 18: Creating an API Admin user**

This account should be used for the “cppmUserName” and “cppmPassword” for the extension configuration.

---

**NOTE**

Always use an account with the API Administrator privileges only. Do not use a Super Administrator account.

An example of Intune extension configuration is shown above. Include appropriate values for your environment based on the information gathered in Step I. Select “Restart” and click on “Save Changes” to restart the extension.

**Figure 19: Setting the extension configuration**
Following the restart, click on “Show Logs”. You should see the following:

**Figure 20: Log validation**

![Log validation](image)

Note the IP address used. This will be used in **Step III** when configuring Intune as an HTTP Authorization source within ClearPass Policy Manager using XML.
Step III: Configuring ClearPass Policy Manager

To complete the configuration, configure an authorization source within ClearPass. With Intune as an authorization source, ClearPass can check with Intune to see if the device is enrolled and managed by Intune before allowing it to connect. Other common use-cases are that ClearPass could any of the returned context such as the version of the installed operating system as the basis for applying specific access policy, or another popular use-case as supported in this latest version of the Intune Extension, is to use the ownership attribute to differentiate between a Corporate or Privately (BYOD) device. These and/or other contextual attributes can be used to evaluate an endpoint at the time of network authentication.

The following steps to add Intune as an Authentication Source can be easily accomplished by importing it into ClearPass using the XML file available in our GitHub repository:

https://github.com/aruba/clearpass-exchange-snippets/tree/master/extensions/microsoft-intune

- Open the XML file using a simple editor and replace X.X.X.X with your EXTENSION IP ADDRESS (Refer Figure 20).
- You can also import the XML without making any edits and then change the “Base URL:” from “http://X.X.X.X” to your extension IP address using the ClearPass UI. This can be changed under the Primary tab of the imported Authentication Source (Refer Figure 22).
- The XML file can be imported into ClearPass by navigating to Configuration > Authentication > Sources. Click on Import and use the file downloaded.

Follow the manual steps below if not using the XML to import the Authentication Source.

Add Intune as an HTTP Authorization Source. Under Configuration > Authentication > Sources, click “Add”.

Figure 21: Adding an HTTP authorization source
Click on Next. This will advance to the Primary Tab provide the connection details.

The Base URL IP address is what was captured in Figure 20 above.

**Figure 22: Adding HTTP authorization source credentials**

![Authentication Sources - Microsoft Intune](image)

Click on Next. This will advance you to the Attributes Tab where you need to provide the authorization attributes. Click on “Add More Filters”. Provide a Name for the filter and then a Filter Query. It’s extremely important that the Filter Query is defined correctly. This is the query string that is sent to the Intune extension asking for context about the endpoint. The query is indexed off the mac-address of the authenticating endpoint. For completeness, the Filter Query is provided here, copy it carefully.

```
?macAddress=%{Connection:Client-Mac-Address-NoDelim}
```

Next build out the definitions of the attributes that will be returned from the Filter Query. These attributes will subsequently be used within our policy-evaluation and ultimately the enforcement policy applied.

**Figure 23: Adding HTTP authorization source query string and returned field definitions**

![Add Filter](image)
Once the HTTP authorization source is defined you can use the returned attributes in your policy processing. Below we cover options on how to use the results from the authorization query in an enforcement policy.

**Using data from Intune in a ClearPass Enforcement Policy**

Multiple use-cases exist for how the data that is returned from Intune can be used in your policy enforcement. In the example below, we are performing multiple checks:

1. Check the device is a Corporately issued and managed device. If true then update the Palo Alto and CheckPoint corporate firewall with context about this device.

2. Check that the device exists in Intune and that it's compliant. In addition to allowing access for these devices, we're also updating the endpoint with the authentication Date & Time so we can track the device's access to the network.

3. If the device is not in compliant then we will apply a Quarantine role.

4. If the device is running an OS that begins with 9.2 [assume iOS] then we flag it as an old-OS.

5. If the device is running an OS that begins with 9.3 [assume iOS] then we flag it as an approved-OS.

6. If the device is running Android OS then we attach a label of Android.

7. If the device is running Android OS then we attach a label of Apple.

**Figure 24: Example of an Enforcement Policy utilizing attributes returned from Intune**
Different companies will have different enforcement profiles and policies. The key take away here is that we are using the authorization attributes received from Intune to drive the policy engine into making and taking different enforcement actions for the device as they authenticate on the network.
Appendix A – Additional diagnostics / support

Extensions Service

ClearPass Extensions are supported by corresponding service under Administration > Server > Services Control called "Extensions service". This service should be running by default.

Restarting this service will affect all deployed and running extensions.

To check on the state and make changes to the service navigate to Administration > Server Manager > Server Configuration [select your ClearPass node] > Service Control. You can also start/stop the extension service from here. By default, this service is automatically started.

![Figure 25: Checking on extension service and how to start/stop the service](image)

Extension Logs/Debugging

If there is a need to access the logs from inside the extension, adjust the "logLevel" to "DEBUG". In the new 6.7 GUI, change the config and restart the extension as shown below. Logs can then be viewed from the 'Show Logs'.

![Figure 26: Turning on Debug logging on an extension using GUI](image)

Once the extension is configured to capture logs, you can access them using the "Collect Logs" function.
Accessing extension logs using ‘Collect Logs’

In addition to viewing the logging of messages as shown above, we can also configure the extension to log messages so that they can be collected and examined via the Policy Manager ‘Collect Logs’ system function, this is extremely useful for our support team.

If there is a requirement for Aruba support to investigate a system issue, one of the items they regularly ask for is the system logs to aid with their diagnostic investigation. By default the “logLevel” is set to INFO but TRACE, DEBUG, INFO, WARN, ERROR, FATAL can also be set. Any of the levels will display the information for the selected state and lower... so if INFO is selected, it will show messages for INFO, WARN, ERROR, FATAL.

After the logs have been collected and expanded, you can locate the extension logs in the following location ‘PolicyManagerLogs->extension’ as shown below.

**Figure 27: Extension logs location in ‘Collect Logs’ diagnostic GZ file**
Monitoring authorization performance

Since we are authorizing against an external system, it is important to monitor the performance of these transactions as you setup and deploy. If you suspect there is a performance issue, ClearPass provides a way to monitor the authorization processing time. The graph below shows an example of this data, navigate to Monitoring -> Live Monitor -> System Monitor [click on ClearPass Tab, then select [Authorization]....

*Figure 28: Monitoring the performance of the authorization process*

ClearPass authorization throughput guidelines

Based upon scale & performance testing completed under ideal test conditions we have concluded that a ClearPass 25K Appliance is capable of sustaining 200 network authentications/second and ClearPass 5K Appliance is capable of sustaining 100 network authentications/second. The test conditions included a service categorization with an authorization check to the Microsoft Cloud based Intune MDM service, EAP-PEAP MS-CHAPv2 authentication between client and ClearPass and local user accounts in ClearPass.
Appendix B – Lab Example for Authorization with caching

A device registered with Intune tries connecting to the Corporate SSID. The corresponding authentication request can be seen under **Monitoring > Access Tracker**.

Following is the list of attributes obtained from Intune as an HTTP Authorization source.

**Figure 29: Authorization attributes in Access Tracker**

![Authorization attributes in Access Tracker](image)

The enforcement Policy defined here dynamically assigns Personal role based on the Ownership attribute returned as shown below:

**Figure 30: Role assignment in Access Tracker**

![Role assignment in Access Tracker](image)
Following is the extension log in DEBUG for the first request received for an endpoint or when the cache has expired for an endpoint.

**Figure 31: Debug Extension logs**

```plaintext
[2018-07-03T03:58:04.008] [DEBUG] intune - Request received. /?macAddress=xxxxxxxxxxxx

[2018-07-03T03:58:04.272] [DEBUG] intune - Cache age for mac xxxxxxx is 296614272ms. (Last Update: Fri Jun 29 2018 17:34:30 GMT+0530 (IST))

[2018-07-03T03:58:04.272] [DEBUG] intune - The token is invalid. Getting a new one.

[2018-07-03T03:58:04.272] [DEBUG] intune - The NAC endpoint is invalid. Updating.

[2018-07-03T03:58:04.273] [INFO] intune - Performing NAC endpoint update.

[2018-07-03T03:58:04.273] [DEBUG] intune - Requesting token for resource "00000002-0000-0000-c000-000000000000".

[2018-07-03T03:58:06.195] [INFO] intune - Performing token update.

[2018-07-03T03:58:06.195] [DEBUG] intune - Requesting token for resource "https://api.manage.microsoft.com/".

[2018-07-03T03:58:06.803] [DEBUG] intune - Querying Intune at https://fef.msua01.manage.microsoft.com/StatelessNACService/devices

[2018-07-03T03:58:08.307] [DEBUG] intune - Adding endpoint with mac address xxxxxxxxxx to the endpoint database...

[2018-07-03T03:58:08.308] [DEBUG] intune - Attempting to update endpoint...

```

In the Access Tracker, if you click on Show Logs you will see the time it takes for the Policy engine to evaluate the conditions. In the first case, when the cache had expired the time it took was 4325ms. This is due to the time it takes to get a response from Intune which resides in the cloud and depends on the latency in your environment.

```
```

However, the subsequent authentication request which was within next 30 minutes was significantly faster.

```
2018-07-03 04:01:26,203  [Th 40 Req 163 SessId R0000009f-01-5b3aa7bd] INFO RadiusServer.Radius - Policy Evaluation time = 469 ms
```
Hence, we can conclude that ClearPass Integration with Microsoft Intune using the extension version Intune v4 significantly speeds up authentications relying on the cached attributes for the endpoint.