ClearPass 6.5

Tech Note: ClearPass

Integration with 3rd Party Enforcement Points
ClearPass & Fortinet utilizing RESTful API and RADIUS Accounting

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modified By</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1 / 0.2</td>
<td>Jan /Feb 2015</td>
<td>Danny Jump</td>
<td>Early Draft Versions</td>
</tr>
<tr>
<td>1.0</td>
<td>March 2015</td>
<td>Danny Jump</td>
<td>First Published Version</td>
</tr>
<tr>
<td>1.1</td>
<td>April 2015</td>
<td>Danny Jump</td>
<td>Calculating dynamically the user-role that is sent from TIPS role.</td>
</tr>
</tbody>
</table>
Figure 31 - Enabling RADIUS Accounting listener on FortiGate ................................................. 25
Figure 32 - Configuring RADIUS SSO on FortiGate ................................................................. 25
Figure 33 - FortiGate RADIUS default configuration................................................................. 26
Figure 34 - Setting the FortiGate to use User-Name as the authenticator ................................. 26
Figure 35 - FortiGate RADIUS configuration after changes ........................................................ 26
Figure 36 - FortiGate using RADIUS accounting 'class' to parse 'user-role' .............................. 27
Figure 37 - Changing FortiGate to use a different AVP than class for 'user-role’ ......................... 27
Figure 38 - FortiGate RADIUS configuration ............................................................................ 28
Figure 39 – Defining a GROUP within FortiGate to match CPPM ‘role’ for user ....................... 28
Figure 40 - FortiGate User groups ............................................................................................. 29
Figure 41 - Adding a rule referencing the group 'geek-group' .................................................... 29
Figure 42 - FortiGate showing differentiated users User-Group ................................................. 30
Introduction

This series of TechNotes describes the integration with 3rd Party firewall vendors. Within this document we have captured the integration with Fortinet. There are two methods of integration between ClearPass Policy Manager and Fortinet. One uses HTTP JSON encoded RESTful API’s the other uses RADIUS Accounting. In the table below we have captured the features and restrictions of different vendors and the capabilities they support.

Note that the framework we have developed is **not** limited to only the below vendors.

Similar to the integration that exists between ClearPass and other vendors, Fortinet supports at a basic level the ability to pass username and source IP address attributes. But other attributes shown below can also be passed. As a summary this is a list of the attributes we pass from CPPM 6.5 to the vendors we have tested.

<table>
<thead>
<tr>
<th>Feature/Firewall</th>
<th>CheckPoint</th>
<th>Fortinet</th>
<th>SonicWall</th>
<th>Palo Alto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Username</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>User Role</td>
<td>✓</td>
<td>✓ [a]</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Domain</td>
<td>✓</td>
<td>✗ [a]</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Device Type</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Machine OS</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Machine Name</td>
<td>✓ [b]</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Health/Posture</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

*Figure 1 - ClearPass Attributes sent to Firewalls*

[a] = Available from RADIUS Accounting, not from HTTP REST API calls  
[b] = Available from HTTP REST API calls not from RADIUS Accounting.

The intent is to document the integration between CPPM and Fortinet. This document focuses on testing and integration specifically Fortinet Authenticator 3.x and FortiGate 5.x or later and covers the integration using RESTful API and RADIUS Accounting.

Where it is practical, best practices will be documented, although not every conceivable use case or deployment can or will be covered here in this document.
**Note:** Within this document where you see a red-chili this is used to signify a ‘hot’ important point and highlights that this point is to be taken as a best-practice recommendation.

**Audience**

The reader is assumed to be familiar with the ClearPass family of products. Basic knowledge of IP networks and wide-area networking is also assumed. A general understanding and previous experience in the deployment/configuration of Fortinet products is also assumed. We also make the assumption that the firewall is already deployed. We will not cover the firewall deployment or configuration beyond the steps to integrate ClearPass.
CPPM 6.5 Post Auth Framework Changes

Historically the Post Auth framework has provided multiple functions such as Palo Alto Networks Integration, RADIUS Accounting, Session Restrictions such as Bandwidth Usage or Multiple Device Enforcement, etc. Starting in CPPM 6.5 we released an enhancement to the Post Auth framework. This will allow ClearPass to provide an enhanced level of integration and possibly allow the enduser/customer the ability to add additional 3rd party systems without Aruba having to specifically add support for the vendor directly in to ClearPass. This follows a similar approach to ClearPass Exchange where the flexibility of ClearPass Exchange encourages and allows for the integration into 3rd party system supporting HTTP RESTful frameworks.

The new Enforcement Profile is a ‘Session Notification Enforcement’. Through the next few sections we will cover this in detail. This new Enforcement Profile enhances the functionality we previously provided within the ‘Session Restriction Enforcement Profile’.

![New Session Notification Enforcement Profile](image)

**Figure 2 - New Session Notification Enforcement Profile**

**Note:** When you migrate a system from a previous CPPM 6.x software level (6.4.x and lower), if previously you had configured integration with a Palo Alto Networks firewall, this would have been defined using a Session Restriction Enforcement Profile [Type – Session Check IP-Address-Change-Notify]. These Enforcement Profiles are **NOT** supported under CPPM 6.5 and any configuration using these profiles is migrated as you upgrade to CPPM 6.5.
Configuring Fortinet & ClearPass

Configuration is required on both the ClearPass and Forti-Authenticator to accomplish the integration. We DO NOT document the configuration between Forti-Authenticator and the FortiGate Firewalls. The Forti-Authenticator unit can integrate with external NAD such as RADIUS (CPPM) and LDAP to gather user logon information and send it to the FortiGate units. The integration discussed below uses a Fortinet exposed RESTful API within the Fori-Authenticator product.

Within ClearPass, the integration with Fortinet falls under the ClearPass Exchange Framework, this framework allows for an extremely flexible and vendor agnostic solution architecture. At the heart of ClearPass Exchange are Context-Servers [endpoints we interoperate with] and Context-Server-Actions [functions performed against the Context Servers]. Starting in ClearPass 6.5 released in February 2015 we have supplied two Fortinet context server actions templates, Login and Logout. We recommend that these default templates are copied and then modified as appropriate when configuring the context server actions for your own installation.

The following two sections walk you through the ClearPass and Fortinet configuration.

ClearPass Configuration

HTTP Context Server: First you need to add a HTTP Context Server (this is the Fortinet Authenticator endpoint). Follow the below directions to add the Context Server:- Administration -> External Servers -> Endpoint Context Servers [Add]

Add the appropriate Server Name (IP Address), this will be translated in to the Server Base URL. Add the Username/Password credentials used to communicate with the Forti-Authenticator firewall. The process of obtaining the username/password are discussed below in section ‘Fortinet Configuration/ Configuring HTTP API Access’.
**Context Server Actions:** Now that we have defined the Forti-Authenticator endpoint, we need to amend the Fortinet context server actions ‘Login & Logout’ to use this Generic HTTP endpoint just configured.

It’s very important that you modify **both** the Login and Logout Server Actions. These are what update the Firewall of a users session going active/de-active. ClearPass will then update the Forti-Authenticator, this in turn will permit/deny this user. We don’t want to update the firewall of a session starting and never clear this session down.

**Note:** Its recommended that you copy the supplied/original Fortinet Login/Logout context server actions templates and modify the copied items.

Within the ‘Action’ tab the ‘Server Name’ must be changed to that of the HTTP server you added in the previous section. The default will be localhost, select the HTTP context server added previously in the drop-down.

![Figure 4 – Adding the Endpoint Server to the Server Action's](image)

**Note:** Nothing needs to be changed on the Fortinet Login/Logout Context Server Actions beyond ensuring the correct Server has been selected as shown above by selecting the Context Server defined previously on the Action Tab. The above shows the Login Context Server Action, the process is exactly the same for the Logout Context server action.
Fortinet Configuration for RESTful Integration

Configuration needs to be completed within the Forti-Authenticator. Several items need to be configured. Follow the steps below to complete the configuration.

Configuring HTTP API Access: To allow CPPM to send username info into Forti-Authenticator requires that we use a username/password that is enabled for this function. Firstly you need to configure SMTP as the password that will be used in conjunction with the username must be emailed to a user configured as shown below e.g. a GMAIL account.

First setup your SMTP server... System -> Messaging -> SMTP Servers... then click on ‘Create now’ .... below is an example of using GMAIL server as a mail-relay .... Your SMTP configuration will likely differ.

![FortiAuthenticator SMTP Configuration](image)

Figure 5 - Configuring SMTP on Forti-Authenticator

Next you need to create a local on box account that you will later configure within CPPM to be used by the RESTful integration. Go to Authentication -> User Management -> Local Users .... then click on ‘Create now’ ....

**Note:** The password specified below is **not** the password configured in CPPM for the HTTP Context Server admin-user definition. That Password is sent out via the SMTP configured above as we previously mentioned.
Following the creation of the account, it needs to be modified to enable the API service. The account will likely have been created as a 'User', change the account to ‘Administrator’ and enable the 'Web service access'. Ensure that you enter a VALID email address in the User Information. Forti-Authenticator will email the PSK to this email address. This is the PSK configured as the password in CPPM that will be used to communicate using the RESTful API.

Having made the above changes, the below confirmation shows the account has been changed, more importantly the confirmation of the email with the password has been sent.
Below is an example of the email sent by Forti-Authenticator with the PSK password.

Adding Active-Directory Server: You may have a need to add an Active-Directory server to the Forti-Authenticator. Complete this step from Authentication -> Remote Auth. Servers -> LDAP. Add the AD server as required. An example is below. Set your IP address, etc.

Adding the AD server allows Forti-Authenticator to validate the username sent by ClearPass. Within the Forti-Authenticator you can monitor the usernames received from in the Logging -> Log Access -> Logs. In this Log-file you can see the usernames.
Now, the username sessions here do not necessarily translate to an SSO valid session that can be seen in **Monitor -> SSO -> SSO Sessions**. The above log file shows a user ‘danny_jump’.... But there is no valid SSO session for this user as danny_jump is **not** in the Active Directory that Forti-Authenticator is joined to. See below, ‘0 SSO sessions’ shown.

Again we see the user ‘danny’ in the Log file below,... but this time we also see the user in the SSO Sessions log as this user ‘danny’, is also in the Active Directory that Forti-Authenticator is joined to and thus can be verified by the Forti-Authenticator.
User ‘danny’ logged as a valid SSO session in the Monitor -> SSO -> SSO Sessions.

Figure 14 - Checking SSO sessions in Forti-Authenticator for AD user
Manually Creating Users on Forti-Authenticator for integration testing

Forti-Authenticator has the ability via curl/wget to manually create test users within the Forti-Authenticator system. This is a great feature to allow us to test the configuration.

**Note:** You need to have the PSK password that was discussed earlier as this is used to authenticate the curl/wget HTTP session.

Below are examples of a Login. Logout is very similar and explained below. I’m not going to provide a detailed description of the curl cmd beyond what relates to the Login/Logout.

**Curl Forti-Authenticator Login Example**

```
curl -k -v -u "admin:pKAHp8kJXPQt2fQXmKrYeIxmAgfOXLGGgqbx" -d '{"event":"1","username":"danny","user_ip":"10.1.73.175"}' -H "Content-Type: application/json" https://IP_of_Forti-Authenticator/api/v1/ssoauth/
```

What’s important from the above example...

"admin:followed_by_the_PSK" .... this is the PSK we discussed that must be emailed out.

"event":1 .... this makes this command effectively a LOGIN command. Change the 1 to a 0 to make this command function as a LOGOUT.

"username":"danny" .... Pretty obvious, this is the username sent from CPPM..!!

"user_ip":"10.1.73.175" .... Pretty obvious, this is the IP addresses of the endpoint..!!

**https://IP_of_Forti-Authenticator /api/v1/ssoauth/** .... This is the IP address of the Forti-Authenticator, you should also recognize the path after the IP address as that matching within the Context Server Action URL for Fortinet Login.

Below is a verbose example of running the Login command using the curl command. The -v option shows all the dialogue. You can see the curl command connecting to Forti-Authenticator, then the server authenticating the admin and then a successful POST of the user-data with the HTTP 200 OK status-code.

```
danny-jump:~ djump$ curl -k -v -u "admin:pKAHp8kJXPQt2fQXmKrYeIxmAgfOXLGGgqbx" -d '{"event":"1","username":"danny","user_ip":"10.1.73.175"}' -H "Content-Type: application/json" https://IP_of_Forti-Authenticator/api/v1/ssoauth/
danny-jump:~ djump$ curl -k -v -u "admin:pKAHp8kJXPQt2fQXmKrYeIxmAgfOXLGGgqbx" -d '{"event":"1","username":"danny","user_ip":"10.1.73.175"}' -H "Content-Type: application/json" https://IP_of_Forti-Authenticator/api/v1/ssoauth/
* Hostname was NOT found in DNS cache
* Trying IP_of_Forti-Authenticator...
* Connected to IP_of_Forti-Authenticator port 443 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
* Server certificate: FAC-VM0A14000265
* Server certificate: support
```
* Server auth using Basic with user 'admin'
> POST /api/v1/ssoauth/ HTTP/1.1
> Authorization: Basic
YWRtaW46cEtBSHA4a0xKWFBRdDJmUVhtS3J2WWVJeElkQWdnZm1PWEwHR3FieA==
> Host: IP_of_Forti-Authentocator
> Accept: */*
> Content-Type: application/json
> Content-Length: 56
>
* upload completely sent off: 56 out of 56 bytes
< HTTP/1.1 200 OK
< Date: Sat, 24 Jan 2015 02:29:23 GMT
* Server Apache is not blacklisted
< Server: Apache
< Vary: Accept,Accept-Language,Cookie
< X-Frame-Options: SAMEORIGIN
< Content-Language: en
< Content-Length: 0
< Content-Type: text/html; charset=utf-8

Figure 15 - Running a Login with a curl command in verbose mode

As an example of the same command run without the verbose [-v] option........

danny-jump:~ djump$ curl -k -u "admin:pKAHp8kJXPQt2fQxMkrvYeIxEAggfmOLGqbx" -d '{"event":"1","username":"danny","user_ip":"10.1.73.176"}' -H "Content-Type: application/json" https://IP_of_Forti-Authentocator/api/v1/ssoauth/
danny-jump:~ djump$

Figure 16 - Running a Login with a curl command in normal mode

Two users 'danny' from different IP address's 10.1.73.175 & 176 from the above two curl cmds.

Figure 17 - SSO Logged in users
Configuring Radius Accounting Proxy

An additional integration method to support 3rd Party vendors was also added to the CPPM 6.5 release. We now support the ability to configure a RADIUS Accounting Proxy. This allows CPPM to proxy the RADIUS accounting data that is received to an external system. When CPPM processes an authentication, as part of the session configuration on CPPM a RADIUS Accounting Proxy target can also be configured. This then allows CPPM to forward the interim accounting updates it receives from the NAS to this external target.

Configure Accounting Proxy on CPPM as shown below: First configure your targets, under Configuration-> Network -> Proxy Targets just like you would previously if you were configuring RADIUS authentication proxy-ing.

![Figure 18 - Adding a PROXY Target](image)

Figure 18 - Adding a PROXY Target
By default the Account Proxy Tab is not shown, you must enable it in the Service Definition as highlighted below.

![No Proxy Tab](image)

**Figure 19 - Enabling Accounting Proxy Configuration**

Now that the target is defined and the Accounting Proxy is enabled the remaining configuration can be completed.

Below is an example of adding the Accounting Proxy to the above service definition. We added a proxy target from one of the targets configured in the previous step.

![Accounting Proxy](image)

**Figure 20 - Configuring Accounting Proxy on an example service**

**Note:** Whatever RADIUS accounting data we receive from the NAS we will forward to the Accounting Proxy target. We can also add VSA attributes to the data we forward. However, to add a VSA we must have the Dictionary’s for the vendor’s product installed/enabled within CPPM, or we may be able to use an IETF VSA. Some vendors have multiple RADIUS Dictionaries across their product ranges, so just because we have one for company X does not mean it will encompass all their products and the VSA’s they support.
Configuring RADIUS Accounting on Fortinet (Authenticator & FortiGate)

Utilizing the RADIUS Accounting Proxy in ClearPass allows us to proxy the accounting information to Forti-Authenticator and FortiGate firewalls. Which can use the RADIUS Accounting records to simplify the SSO process for users. Forti-Authenticator can also manipulated the data before it forwards it to the subscribing FortiGate firewalls.

**Note:** If a customer does not have a Forti-Authenticator then RADIUS Accounting data could be sent directly to the FortiGate firewall, this is explained below in a later section. If a customer has multiple FortiGate firewalls by using the Forti-Authenticator the forwarding of data from ClearPass is simplified as ClearPass now only has to send to a single target. Forti-Authenticator will then forward the data to the subscribing FortiGate firewalls. However, if there is no Forti-Authenticator in the customers network, ClearPass can fulfill the RADIUS accounting ‘broadcast’ role.

As discussed below in more depth and the reasoning, we will be using the Filter-Id field to pass the ‘user-role’ from ClearPass to the Forti-Authenticator and FortiGate endpoints.

Here is how you add the RADIUS Accounting attribute Filter-Id [attribute 11] in to the ClearPass Accounting Proxy policy so that when the RADIUS Accounting data is forwarded to the Fortinet product we set the ‘user-role’ as appropriate. In the below example we are setting the Filter-Id to be ‘geek-group’, in other ClearPass services we configured for our testing we set a different Filter-Id as appropriate, e.g. **geek-group**, **PLM** or **TME**. This is then parsed and matched on the Forti-Authenticator/FortiGate to tie the user to the relevant User Group to enforce policy.

In the service enable, the Accounting Proxy, add a RADIUS attribute, **Radius: IETF**, under Name, find **Filter-Id** (attribute 11) and set the value as required.

![Configuration > Services > Edit - Fortinet-Checkpoint testing](image)

Figure 21 - Setting RADIUS accounting attribute Filter-Id = ‘geek-group’
**Note:** The **Value** field above can also be a substitutional attribute as shown below.... Just to call this out you could choose from any of the allowable values we support.... Whether they make sense to be used for a 'user-role' will be appropriate to individual deployments.!!

Using AD TIPS Role as user-role The above method is a little bit inflexible and maybe not suitable or representative of how you want to identify a users-role. In the below example we are using the AD **memberOf** attribute to assign a Tips ROLE to the user, using the** CONTAINS** calculation.... In our example we look for PLM or TME and assign a similar TIPS role.

![Figure 22 - Calculating the TIPS role from AD memberOf](image-url)
Then within the Accounting Proxy we use the value {Tips:Role} to pass the assigned role to the Fortinet enforcement point.

**Note:** You have to manually complete and type the text {Tips:Role}, as we do not auto-complete this parameter.

![Figure 23 - Using TIPS role as the bases for User-Role in Fortinet](image)
**Configure Forti-Authenticator RADIUS Accounting** To start, ensure that the Forti-Authenticator is able to receive RADIUS Accounting messages. Check under **System -> Network -> Interfaces [data interface]** and then check the interfaces in use to ensure the Services are enabled as shown below.

![FortiAuthenticator Interface Configuration](image)

*Figure 24* - Checking RADIUS Accounting listen service is enabled on the interface
Next enable RADIUS Accounting SSO clients – **Fortinet SSO Methods -> General.**

![Figure 25 - Enable RADIUS SSO client on Forti-Authenticator](image)

Next we need to configure the RADIUS Accounting and define the accounting source, **Fortinet SSO Methods -> SSO -> RADIUS Accounting.** Pay special attention to the RADIUS attributes below. We have mapped **Username** to **User-Name**, **Client IP** to **Framed-IP-Address** and **User group** attribute to **Filter-ID.**

![Figure 26 - Setting RADIUS Accounting source and remapping attributes](image)
Now we need to navigate to **Fortinet SSO -> Methods -> Accounting Proxy -> Sources** and click ‘Create New’ to add ClearPass as the source for the RADIUS Accounting messages.

![Adding a RADIUS Accounting source](image1)

**Figure 27 - Adding a RADIUS Accounting source [ClearPass]**

Ensure that the PSK password matches the proxy-target you configured above in the ClearPass section.

![Adding ClearPass as a RADIUS Accounting source](image2)

**Figure 28 - Adding ClearPass as a RADIUS Accounting source**

Next, you’ll have to configure the Accounting Proxy Targets, were not going to cover this in detail but this is effectively the FortiGate firewalls.
Also not covered in this document is Forti-Authenticator ability to add or modify RADIUS IETF standard or VSA Accounting attributes to the Accounting data before it forwards to the FortiGate firewalls.

But once the above configuration is complete, you can check on and monitor the SSO session that are known in Forti-Authenticator under Monitor -> SSO Sessions

Figure 29 - Configuring the Forti-Authenticator Accounting Target/Destinations

Figure 30 - Monitor RADIUS SSO Sessions in Forti-Authenticator
Configure FortiGate RADIUS Accounting

As discussed previously for customers that do not have Forti-Authenticator ClearPass can send RADIUS Accounting data directly to the FortiGate firewalls, this is discussed below.

To start, ensure that the FortiGate is able to receive RADIUS Accounting messages. Check on the FortiGate firewall to allow the interfaces to listen for RADIUS Accounting Messages under *System -> Network -> Interfaces [data interface]* ensure that the *Listen for RADIUS Accounting Messages* is enabled.

![Figure 31 - Enabling RADIUS Accounting listener on FortiGate](image)

Next enable the RADIUS SSO agent (RSSO), *Users & Devices -> Authentication -> Single Sign-On* and set the shared secret to match CPPM.

![Figure 32 - Configuring RADIUS SSO on FortiGate](image)
Now, the FortiGate will be receiving and consuming the RADIUS Accounting messages being sent from ClearPass. By default the FortiGate firewall will use the MAC address to authenticate the user. We need to modify the mapping so the FortiGate uses the username.

We do this from the CLI... SSH to the firewall with the appropriate credentials.

Below is the configuration for the user ‘radius’ before we modify it.

```
fortigate_1 # show user radius
cfg user radius
  edit "RSO Agent"
    set rsso enable
    set rsso-radius-response enable
    set rsso-validate-request-secret enable
    set rsso-secret ENC MDPY2AMlzbzBxUy7CxeE6BPNTNQPZ71wC8xapuzEzm9HzVBrI3JU6JLQul9dJHLuABymovkgXKsPLJtvPsaFVrk0QR0mTjrdQ
    set rsso-log-flags protocol-error profile-missing accounting-stop-missed accounting-event endpoint-block radiusd-other
next
end
```

Figure 33 - FortiGate RADIUS default configuration

Change the radius configuration attribute **rsso-endpoint-attribute** to use ‘User-Name’ as the authenticator. **Note:** Other attributes could be used.

```
fortigate_1 # config user radius
fortigate_1 (radius) # edit CPPM_650
fortigate_1 (CPPM_650) # set rsso-endpoint-attribute User-Name
fortigate_1 (CPPM_650) # end
```

Figure 34 - Setting the FortiGate to use User-Name as the authenticator

Below is the configuration following the change, notice the new line in the configuration **set rsso-endpoint-attribute User-Name**.

```
fortigate_1 # show user radius
cfg user radius
  edit "RSO Agent"
    set rsso enable
    set rsso-radius-response enable
    set rsso-validate-request-secret enable
    set rsso-secret ENC MDPY2AMlzbzBxUy7CxeE6BPNTNQPZ71wC8xapuzEzm9HzVBrI3JU6JLQul9dJHLuABymovkgXKsPLJtvPsaFVrk0QR0mTjrdQ
    set rsso-endpoint-attribute User-Name
    set rsso-log-flags protocol-error profile-missing accounting-stop-missed accounting-event endpoint-block radiusd-other
next
end
```

Figure 35 - FortiGate RADIUS configuration after changes

Finally for this section we need to get the FortiGate RSSO User Group to match the Role set within ClearPass. This section requires several changes from the default.
By default the Fortigate firewall expects to parse the ‘user-role’ from the **RADIUS Attribute ‘Class’** sent by ClearPass in the RADIUS Accounting-start messages. However this attribute [Class] is typically present in the RADIUS accounting messages ClearPass receives from some NAD’s and would forward this to the FortiGate with an incorrect value. When ClearPass adds the [Class] AVP it adds it as a second index of [Class] but the Fortigate firewall only reads the first index value thus missing the AVP we have added used to set the ‘user-role’. To overcome this issue we suggest using an RADIUS accounting attribute that is not being used. For our testing we have used Filer-ID.

Configuring this requires several. Initially when setting up the FortiGate User-Groups and when we add the ‘user-role’ tag if you float your cursor over the ? as below you see the default that it will look to match on in the Accounting-Start messages is ‘Class’.

![Figure 36 - FortiGate using RADIUS accounting 'class' to parse 'user-role'](image)

So we need to tell ClearPass to send the user-role in a different attribute and have FortiGate look for this AVP in the attribute we define.

Firstly we need to modify FortiGate to look elsewhere in the accounting messages. This is accomplished from within the CLI. Log into the CLI and follow the below commands.

Remember we opted to use the RADIUS attribute ‘Filter-Id’, you may select a different attribute as long as the ClearPass and FortiGate systems match.

If you look in the default FortiGate configuration the value ‘sso-attribute’ is not visible, the default is set to use the RADIUS accounting attribute ‘class’ though.

```
#config user radius
#edit CPPM_650       [this is the name of my CPPM node in FortiGate]
#set sso-attribute Filter-Id
#end
```

![Figure 37 - Changing FortiGate to use a different AVP than class for 'user-role'](image)

After this change my radius configuration on the FortiGate looks like this. Notice towards the end of the configuration the sso-attribute set to Filter-Id.
C
learPass 6.x
CPPM Integration with 3rd Party Enforcement Points (Fortinet)

fortigate_1 # show user radius
config user radius
  edit "CPPM_650"
    set rsso enable
    set rsso-radius-response enable
    set rsso-validate-request-secret enable
    set rsso-secret ENC
au6KRSjplp6R1DJyT4Bp39rDalokGw1e85zr0xcY5NSqHZQt0R+CQyTRswMcq3UOsQYOlvKH92k7cL LJsz/vMNbKyLGTyhMw5bdem8UfAtHNgM+Ag022hSTvXFDi0MsoM9pKh7j12liEcOU+H/1L+7Spr0I j4KBncRzn5mXdpBjRG/Sj/Db2hFPtYG2AblPBPA==
    set rsso-endpoint-attribute User-Name
    set sso-attribute Filter-Id
next
end

Figure 38 - FortiGate RADIUS configuration

Now we need to create the User Groups on the FortiGate appliance and the appropriate label to associate an incoming username to a Fortinet User Group.

Create the User Groups as required, the Type must be set as RADIUS Single Sign-on (RSSO).

Under User & Device -> User -> User Groups [Create New]

The name is what FortiGate will use when you are defining policy.... See below....

![Figure 39 - Defining a GROUP within FortiGate to match CPPM 'role' for user](image)

We created as shown below three User Groups, and gave each Group a separate Label, aka the RADIUS Attribute Value. The Groups for our testing are PLM, TME and geek-group. The RADIUS attribute value is the same as the User Groups name, just for simplicity.
Now the Groups are created, we can create policy to apply enforcement to users based upon the Fortigate Group membership. This can be thought of as ClearPass is passing the role of the user within the RADIUS Accounting and FortiGate is grouping the users of this role under the User Group for the same level of enforcement.

Finally we defined a simple policy to show how we can reference the User Groups adding a policy rule referencing the group **geek-group**, also notice the other User Groups of **PLM** and **TME** could be selected.

Now that ClearPass is passing the username with a 'user-role' we can see below the FortiGate authenticated users are matched to the relevant User-Group.
Figure 42 - FortiGate showing differentiated users User-Group