RADIUS Configuration Note

WINS™: Wireless Interoperability & Network Solutions
1. OVERVIEW ................................................................. 3
2. AUTHENTICATION AND ACCOUNTING .......................... 4
3. 802.1X, CAPTIVE PORTAL AND MAC-FILTERING .................. 6
   3.1 802.1x ................................................................. 6
   3.1 Captive Portal Authentication with RADIUS ..................... 8
   3.2 MAC-filtering ..................................................... 8
4. COMMONLY USED RADIUS FEATURES .............................. 10
   4.1 Dynamic VLAN assignments: .................................... 10
   4.2 Personal Firewall: ................................................ 12
   4.3 Restricted SSID .................................................... 14
5 PEAP, EAP-TLS TERMINATION -STARNET RADIUS SUPPORT ........ 16
   5.1 Overview ............................................................. 16
6. RADIUS FAIL-OVER AND HEALTH CHECK ....................... 19
   6.1 Health Check: ...................................................... 21
7. MISCELLANEOUS: ....................................................... 22


1. **OVERVIEW**

RADIUS is a protocol that provides centralized authentication, accounting & authorization (AAA) management for user laptops/computers to gain access to a network. Meru Controller as a Radius client sends user credentials and connection parameter information in the form of a RADIUS messages to a RADIUS server. The RADIUS server authenticates and authorizes the RADIUS client requests, and sends back a RADIUS message response. Meru supports various EAP types such as EAP-TLS, TTLS, PEAP etc associated to 802.1 x using RADIUS. While not getting in to the low level details, the real intention of this document is to make the reader understand basic and optional configuration on Meru controllers/RADIUS Servers, Failover mechanism defined and some other Meru specific aspects.

**EAP exchanges during 802.1 x authentication**
2. AUTHENTICATION AND ACCOUNTING

If there are any specific software requirements in the controller to support a particular client model as suggested in Meru documents, the same should be installed in the controller accordingly. The client devices used in the test bed for this documentation are generic and supported by standard software releases. Below procedure explains how to create RADIUS authentication and accounting profiles and map it to the ESS profiles in a Meru Controller.

Global RADIUS Authentication and Accounting Profile

Mapping the RADIUS authentication profile to a Security profile
Mapping the RADIUS accounting profile to ESS profile
3. **802.1X, CAPTIVE PORTAL AND MAC-FILTERING**

3.1 **802.1X**

802.1X is an IEEE standard for authenticated network access to wired Ethernet networks and wireless 802.11 networks. IEEE 802.1X enhances security and deployment by providing support for centralized user identification, authentication, dynamic key management, and accounting. The support that 802.1X provides for Extensible Authentication Protocol (EAP) types such as, EAP, EAP-TLS, EAP-MS-CHAP v2, and PEAP allows you to choose several authentication methods for wireless clients and servers. Meru controllers and APs are transparent to the EAP types as no special configuration is required to enforce any of it. However, the wireless client supplicants and RADIUS Server should be appropriately configured to support a specific EAP type. Below screen shots explains the configuration requirement in Intel client supplicant and Network Policy Server running in windows 2008 Server to support PEAP.

**Sample configuration with Intel Utility to support PEAP**
Windows NPS configuration. Meru controller already added to the RADIUS client list
3.1 Captive Portal Authentication with RADIUS

Captive Portal is a feature designed to isolate temporary users on a network, for example guests in a company or students using a library. If Captive Portal is enabled, the HTTP protocol over Secure Socket Layer (SSL, also known as HTTPS) provides an encrypted login interchange with the RADIUS server until the user is authenticated and authorized. (Captive Portal feature also supports local authentication). During this interchange, all traffic with the Client station except DHCP, ARP, and DNS packets are dropped until access is granted. If access is not granted, the user will not be able to leave the Captive Portal login page. If access is granted, the user is released from the Captive Portal page and is redirected to the originally requested URL/Website as the user now gain full access to WLAN. This section provides instructions to implement Captive Portal in Meru controllers by using RADIUS Server to authenticate the users.

The back-end authentication between RADIUS server and controller uses PAP, in other words the user credentials are sent in clear text. Also if an external captive portal server is used, the authentication happens between External Captive portal server and RADIUS server directly (the external captive portal server is the RADIUS client).

3.2 MAC-filtering

MAC-filtering is a global configuration that can be used in the system to control access by enabling a permit or deny list, based on the MAC-address of users. Similar to captive portal, the user database can reside in the controller locally also can be in an external database authenticated using a RADIUS server. As explained in the sequential screen shots below, MAC-filtering procedure involves 3 steps; a) to configure a RADIUS profile, b) to enable ACL globally, c) enabling the “MAC-filtering flag” in a security profile. Note the delimiter type defined in RADIUS profile will be the format used by Controller to send the MAC address (user name) to the RADIUS Server. Also the password can be same MAC-address or RADIUS secret/shared key by itself which needs to be configured in the directory server accordingly.
To configure MAC-filtering

Enable MAC-filtering in a specific Security Profile
4. COMMONLY USED RADIUS FEATURES

4.1 Dynamic VLAN assignments:

Each WLAN has a static network policy that applies to all clients associated with a Service Set Identifier (SSID). Clients are required to associate with different SSIDs in order to inherit different QoS and security policies depending on the subnet they belong. Dynamic VLAN assignment is a feature to help overcome such a situation which places a wireless user into a specific VLAN, based on the credentials supplied by the user.

All supported VLANs should be configured in the controller and mapped to the correct controller interfaces to allow segmentation of traffic once a VLAN-id is returned by the RADIUS server upon a successful user authentication. Also the uplink switch ports where controller is connected should be tagged with the same VLANs to forward traffic. Following snapshot explains how dynamic VLAN assignment works in Meru Infrastructure using windows 2008 NPS.

Add a VLAN profile in the controller

Select tunnel interface type in an ESS-configuration to “RADIUS VLAN only or RADIUS and configured VLAN”
Add a VLAN attribute in the connection request policy of NPS

Packet inspection made using a Sniffer. RADIUS-ACCEPT returning the VLAN tag

The returned VLAN tag
4.2 Personal Firewall:

RADIUS-configured filter-id provides a policy firewall after successful 802.1X authentication of the user. This feature requires the RADIUS server to return a firewall filter-id upon a successful user authentication and a matching QoS rule configured in controller. A PEF license also should be installed in the controller for the policy enforcement. The below example demonstrates a test case to deny the FTP usage for a set of users.

Create a Qos Rule to deny FTP traffic

**Turn ON RADIUS-configured firewall capability in the Security Profile**

![Image of RADIUS configuration settings]
Add the attribute to the RADIUS server policy

Verify the filter-id returned by RADIUS from a capture
4.3 Restricted SSID

RADIUS-Based ESS Profile Restriction is a feature that gives a controller the capability to restrict wireless clients attempting connection through multiple ESS profiles which uses same RADIUS profiles in the backend to authenticate users. The clients can connect only to certain SSIDs which will be mentioned in a RADIUS Accept message. In absence of the RSSID feature, all wireless clients provisioned in the RADIUS Server have access to all ESS profiles and hence all associated VLANS. With SSID restriction, the RADIUS server can be configured for each wireless client specifying the SSIDs they can connect with. You can use a RADIUS server to restrict SSID connection using VSA in the RADIUS Accept message.

There are three possible conditions for an SSID

<table>
<thead>
<tr>
<th>RADIUS Server is sending</th>
<th>Results in</th>
</tr>
</thead>
<tbody>
<tr>
<td>No list of acceptable SSIDs</td>
<td>Connection is accepted</td>
</tr>
<tr>
<td>A list of acceptable SSIDs that includes the ID</td>
<td>Connection is accepted</td>
</tr>
<tr>
<td>A list of acceptable SSIDs that does not include the ID</td>
<td>Connection is not accepted</td>
</tr>
</tbody>
</table>

The RADIUS server should return the allowed SSID(s) in a Vendor-specific attribute (VSA) with Vendor code 9 and attribute number 1 in the Access-Accept message. The attribute value should be string format. The string should say ssid=<ssid-string> where <ssid-string> is replaced by the actual SSID (also known as the ESSID). If a list of multiple allowed SSIDs is used, put each SSID in a separate instance of the attribute. The order of the attributes does not matter. If the SSID to which the station is trying to connect is not among the SSIDs returned by the RADIUS server, the station access will be denied. This feature has no CLI or Web UI commands associated with it. If the RADIUS responds with a list of allowed SSIDs, the list is used to process and limit the user.

Adding attribute in RADIUS Server
Verifying using a capture

The RADIUS Server authenticates the user, but the controller can drop the user if the RSSID string is not matching the SSID to which user connection was attempted. The reason for disconnect, as of today will be printed as a “back end authentication failure” in the station logs, but more detailed information can be gathered from security traces with flags 800009 enabled. Below is example of an extract of traces collected when a user trying to establish a connection and there is a mismatch in the SSID.

```
[08/28 23:38:03.982] SEC: Cisco attribute : attr id : 8
[08/28 23:38:03.982] SEC: rad_ssid : avalanche attribute is : noc
[08/28 23:38:03.982] SEC: Restrict SSID 1

[08/28 23:38:03.982] SEC: Attribute 11 (?)Unknown?) length=9
[08/28 23:38:03.982] SEC: Attribute 65 (Tunnel-Medium-Type) length=6
[08/28 23:38:03.982] SEC: Attribute 64 (Tunnel-Type) length=6
[08/28 23:38:03.982] SEC: Attribute 7 (?)Unknown?) length=6
[08/28 23:38:03.982] SEC: Attribute 6 (?)Unknown?) length=6
```

The RADIUS Server authenticates the user, but the controller can drop the user if the RSSID string is not matching the SSID to which user connection was attempted. The reason for disconnect, as of today will be printed as a “back end authentication failure” in the station logs, but more detailed information can be gathered from security traces with flags 800009 enabled. Below is example of an extract of traces collected when a user trying to establish a connection and there is a mismatch in the SSID.
5. PEAP, TTLS TUNNEL TERMINATION - STARNET RADIUS SUPPORT

5.1 Overview

PEAP uses Transport Layer Security (TLS) to create an encrypted channel between an authenticating PEAP client, such as a wireless computer, and a PEAP authenticator, such as an Internet Authentication Service (IAS) or Remote Authentication Dial-In User Service (RADIUS) server. PEAP does not specify an authentication method, but provides additional security for other EAP authentication protocols, such as EAP-MS-CHAP v2, that can operate through the TLS encrypted channel provided by PEAP. PEAP is used as an authentication method for 802.1X wireless client computers.

**PEAP authentication process**

There are two stages in the PEAP authentication process between PEAP client and authenticator. The first stage sets up a secure channel between the PEAP client and the authenticating server. The second stage provides EAP authentication between the EAP client and authenticator.

**PEAP stage one: TLS encrypted channel**

The wireless client associates with a wireless access point. An IEEE 802.11-based association provides an Open System or Shared Key authentication before a secure association is created between the client and access point. After the IEEE 802.11-based association is successfully established between the client and access point, the TLS session is negotiated with the access point. After authentication is successfully completed between the wireless client and the server (for example, an IAS server), the TLS session is negotiated between them. The key that is derived during this negotiation is used to encrypt all subsequent communication.

**PEAP stage two: EAP-authenticated communication**

Complete EAP communication, including EAP negotiation, occurs inside the TLS channel created by PEAP during the first stage of the PEAP authentication process. The IAS server authenticates the user or the client computer with the method that is determined by the EAP type and selected for use within PEAP. For deployments of WPS technology, EAP-MS-CHAP v2 is the authentication type used within PEAP. The controller only forwards messages between wireless client and RADIUS server—the controller (or a person monitoring it) cannot decrypt these messages because it is not the TLS end point.

The structure of TTLS and PEAP are quite similar. Both are two-stage protocols that establish security in stage one and then exchange authentication in stage two. Stage one of both protocols establishes a TLS tunnel and authenticates the authentication server to the client with a certificate. Once that secure channel has been established, client authentication credentials are exchanged in the second stage.

TTLS uses the TLS channel to exchange "attribute-value pairs" (AVPs). The general encoding of information allows a TTLS server to validate AVPs against any type of authentication mechanism. TTLS implementations today support all methods defined by EAP, as well as several older methods (CHAP, PAP, MS-CHAP and MS-CHAPv2).
PEAP Tunnel Termination

Starnet is a RADIUS server which does not understand PEAP messages and have limited support for EAP-MD5 and MS-CHAP-V2. As illustrated below, the PEAP tunnel is terminated in controller and only a supported authentication method (MS-CHAPv2 for example) is forwarded to the RADIUS server.

Configuration check box enabled in 802.1x security profile.
Custom certificates installed in the controller. Choose the option “Security”.

Server certificate installed in a client
Note: Note sure of any use case scenarios from the real world that can be accomplished with this specific feature. Majority of today's enterprise RADIUS servers support almost all EAP types. For documentation purpose, it was tested in a limited Lab environment using NPS and by terminating PEAP in the controller.

6. RADIUS FAILOVER AND HEALTH CHECK

There are 2 internal modules or services in a Meru controller which mandates the backend RADIUS authentication feature. Since its uses Meru proprietary engineering names or terms, we are calling the modules as category A and B. The failover method defined is different in each module as it depends on the type of user authentication. For example, standard 802.1x or enterprise WPA/WPA2 authentication/accounting is managed by category A and RADIUS-based MAC filtering, CP authentication/accounting, RADIUS-based access Management for WEBGUI falls in category B.

Category A: Authentication failover (802.1x)
Category A: Accounting failover (802.1x)

Client

Original request
1st retry
2nd retry
3rd retry
4th retry
5th retry

Primary Server

16 seconds
32 seconds
Sixth and fail over

Secondary Server

15 minutes
Request after fail-over window

The sniffer capture

<table>
<thead>
<tr>
<th>No.</th>
<th>Time (UTC)</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>927.897839</td>
<td>172.16.0.4</td>
<td>172.18.10.13</td>
<td>RADIUS</td>
<td>Access-challenge (11) (14d-11, 1-127)</td>
<td>0.000273</td>
</tr>
<tr>
<td>21</td>
<td>917.896414</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-request (1) (1d=3, 1-263)</td>
<td>0.018853</td>
</tr>
<tr>
<td>22</td>
<td>917.890375</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-challenge (11) (14d-7, 1-128)</td>
<td>0.000934</td>
</tr>
<tr>
<td>23</td>
<td>917.990427</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-request (1) (1d=26, 1-263)</td>
<td>0.027874</td>
</tr>
<tr>
<td>24</td>
<td>917.990299</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-challenge (11) (14d-26, 1-129)</td>
<td>0.000886</td>
</tr>
<tr>
<td>25</td>
<td>918.002674</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-request (1) (1d=26, 1-263)</td>
<td>0.013631</td>
</tr>
<tr>
<td>26</td>
<td>918.002672</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-challenge (11) (14d-26, 1-130)</td>
<td>0.005408</td>
</tr>
<tr>
<td>27</td>
<td>918.005846</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-request (1) (1d=26, 1-263)</td>
<td>0.034604</td>
</tr>
<tr>
<td>28</td>
<td>918.006079</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-challenge (11) (14d-28, 1-131)</td>
<td>0.002443</td>
</tr>
<tr>
<td>29</td>
<td>918.004948</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-request (1) (1d=26, 1-263)</td>
<td>0.034179</td>
</tr>
<tr>
<td>30</td>
<td>918.005526</td>
<td>172.18.10.13</td>
<td>172.19.0.4</td>
<td>RADIUS</td>
<td>Access-challenge (11) (14d-29, 1-132)</td>
<td>0.001556</td>
</tr>
</tbody>
</table>

The Fail Over to Secondary
Category B: Authentication and accounting failover (Captive Portal, MAC-filtering etc)

5.1 Health Check:

There is no mechanism used by the controller to check the primary and secondary server availability or status besides sending standard RADIUS requests in predefined intervals described in above authentication and accounting scenarios. However if inference logs are enabled, controllers will send ICMP messages to configured RADIUS servers to report back the availability. The default interval for RADIUS health check is 60 seconds. The following command is used to enable logging in the CLI.
7. MISCELLANEOUS:

Presently the failover algorithm is defined on per-ESS basis. It means a failover scenario is not updated globally which makes users in other ESS profiles to send requests to an already failed primary server before switching to secondary.

Run state RADIUS failover information is not carried forward during Nplus1 failover scenarios.

The inference logs are classified as events. The system also generates syslog messages during RADIUS fail-over.

Different failover algorithms are used for authentication and accounting in 802.1x, as per the system design.