APPLICATION NOTE

No. 330181203-03

Date: **April 2003**
Author: **Eli Shmulenson, ITC Tier IV Product Support Engineering**
Product: **Avaya P330, P330-ML, C460**
General: **FreeRadius configuration for 802.1x support**

Configuring FreeRadius 0.8.1 for IEEE 802.1x support

The FreeRadius Server Project is an attempt to create a high-performance and highly configurable GPL'd-free RADIUS server. The server is similar to Livingston's 2.0 server. FreeRADIUS is a variant of the Cistron RADIUS server, but they don't have a lot in common any more.

FreeRadius server supports EAP IEEE 802.1x based authentication. This document contains examples the FreeRadius server to work with Avaya P330, P330-ML and C460 switches.

This document relates to FreeRadius server version 0.8.1 installed on a RedHat 7.3 Linux platform.

Before enabling the 802.1x authentication support, make sure the Avaya switch has IP connectivity to the RADIUS server. If the RADIUS server is connected directly to the switch, make sure its dot1x port-control is in "Force-Authorize" status.

1. User and user related parameters definition:

User related parameters are defined in /usr/local/etc/raddb/users file. We will define the following user-related parameters in our example:

User name: joe
Password: test1
The VLAN the user should be assigned to: 5
Static VLAN binding for the user: 2
User port priority: 0

Important notice:

The feature applying network parameters to a user port by using Radius server is not supported in the Avaya P330 family switches (not including the P330-ML family). Other Avaya products support this feature on copper 10/100 ports only.

Below is an example for user definition in the /usr/local/etc/raddb/users file according to the above example:

../raddb/users

```
joe    Auth-Type := EAP, User-Password == "test1"
       Tunnel-Private-Group-Id:1 = 5,
       Avaya-StaticVlan-Type = 2,
       Avaya-PortPriority-Type = 0,
       Tunnel-Type:1 = VLAN
```

2. Client (NAS) definition:

You should define RADIUS clients (Network Access Servers) in the /usr/local/etc/raddb/clients file. In the following example file, the Radius client IP address is 149.49.138.126 and its shared secret is “test123”.

../raddb/clients

```
# Client Name           Key
#---------------- ----------
149.49.138.126          test123
```

3. Avaya specific and standard attributes definition

Edit the following files in the /usr/local/etc/raddb directory and copy into them the corresponding lines:

../raddb/dictionary

```
$INCLUDE dictionary.tunnel
$INCLUDE dictionary.avaya
```

../raddb/dictionary.avaya

```
# Avaya P330 dictionary file

VENDOR        Cajun_p330   2167

ATTRIBUTE     Cajun-Service-Type  1      integer      Cajun_p330
VALUE         Cajun-Service-Type  Cajun-Read-Only-User       1
VALUE         Cajun-Service-Type  Cajun-Read-Write-User      2
VALUE         Cajun-Service-Type  Cajun-Admin-User           3

ATTRIBUTE     Avaya-StaticVlan-Type      12     string Cajun_p330

ATTRIBUTE     Avaya-PortPriority-Type    13     integer      Cajun_p330
VALUE  Avaya-PortPriority-Type    0      0
VALUE  Avaya-PortPriority-Type    1      1
VALUE  Avaya-PortPriority-Type    2      2
VALUE  Avaya-PortPriority-Type    3      3
VALUE  Avaya-PortPriority-Type    4      4
VALUE  Avaya-PortPriority-Type    5     5
VALUE  Avaya-PortPriority-Type    6     6
VALUE  Avaya-PortPriority-Type    7     7
```

../raddb/dictionary.tunnel

```
ATTRIBUTE    Tunnel-Private-Group-Id          81     string has_tag

ATTRIBUTE    Tunnel-Type          64     integer       has_tag
VALUE        Tunnel-Type   PPTP   1
VALUE        Tunnel-Type   L2F    2
VALUE        Tunnel-Type   L2TP   3
VALUE        Tunnel-Type   ATMP   4
VALUE        Tunnel-Type   VTP    5
VALUE        Tunnel-Type   AH     6
VALUE        Tunnel-Type   IP     7
VALUE        Tunnel-Type   MIN-IP 8
VALUE        Tunnel-Type   ESP    9
VALUE        Tunnel-Type   GRE    10
VALUE        Tunnel-Type   DVS    11
VALUE        Tunnel-Type   VLAN   13
```

4. Setting the EAP authentication type

The file /usr/local/etc/raddb/radiusd.conf contains the EAP related definitions. The following example file defines the MD5-Challenge authentication type.

../raddb/radiusd.conf

```
# Extensible Authentication Protocol
       #
       #  For all EAP related authentications
       eap {
              # Invoke the default supported EAP type when
              # EAP-Identity response is received
              #      default_eap_type = md5

              # Default expiry time to clean the EAP list,
              # It is maintained to co-relate the
              # EAP-response for each EAP-request sent.
              #      timer_expire     = 60

              # Supported EAP-types
              md5 {
              }

              ## EAP-TLS is highly experimental EAP-Type at the moment.
              #      Please give feedback on the mailing list.
              #tls {
              #      private_key_password = password
              #      private_key_file = /path/filename

              #      If Private key & Certificate are located in the
              #      same file, then private_key_file & certificate_file
              #      must contain the same file name.
              #      certificate_file = /path/filename

              #      Trusted Root CA list
```

```
                #CA_file = /path/filename

        #       dh_file = /path/filename
                #random_file = /path/filename
        #
        #       This can never exceed MAX_RADIUS_LEN (4096)
        #       preferably half the MAX_RADIUS_LEN, to
        #       accomodate other attributes in RADIUS packet.
        #       On most APs the MAX packet length is configured
        #       between 1500 - 1600. In these cases, fragment
        #       size should be <= 1024.
        #
        #               fragment_size = 1024

        #       include_length is a flag which is by default set to yes
        #       If set to yes, Total Length of the message is included
        #       in EVERY packet we send.
        #       If set to no, Total Length of the message is included
        #       ONLY in the First packet of a fragment series.
        #
        #               include_length = yes
        #}
}
```

5. Troubleshooting

- Make sure the switch is properly configured for 802.1x authentication.
- Validate that the switch has IP connectivity to the Radius server (use “ping” command). If the RADIUS server is connected directly to the switch, make sure its dot1x port-control is in “Force-Authorize” status.
- Make sure the FreeRadius is up and running. Use *ps –ef | grep radius* command to see if the radius process is running.
- Take a look at the FreeRadius log files. Usually they are located in the /usr/local/var/log/radius/ directory.

For more information about FreeRadius please refer to http://www.freeradius.org.

Important notice:

The feature of applying network parameters to a user port by using Radius server is not supported in the Avaya P330 family switches, except for the P334T-ML. This feature is supported on copper 10/100 ports only.