==========

Modified Steps to Run MUD Controller and FreeRadius

Megan Massey
Lalka Rieger

==========

Setting up FreeRadius Server on Ubuntu Virtual Machine

1. In your home directory, download FreeRADIUS 3.0x Series-Stable using wget

   $ wget https://github.com/FreeRADIUS/freeradius-server/archive/release_3_0_11.tar.gz

2. Certain packages may be missing. Ensure that libtalloc and libcrypto are installed. If not, run the following commands:

   $ sudo apt-get install litalloc-dev
   $ sudo apt-get install libssl-dev

3. Unzip the resulting .tar.gz file by running the following command:

   $ tar -xvzf release_3_0_11.tar.gz

4. Enter the freeradius-server-release_3_0_11 directory and configure, make, and install by running the following commands:

   $ ./configure
   $ make
   $ sudo make install

5. All the associated FreeRADIUS folders should be in the appropriate places now. Create a new vendor dictionary file called dictionary.mudserver in /usr/local/share/freeradius/ and copy and paste the below text:

```
##########################################################################
########
VENDOR              CISCO-IOT           16122

BEGIN-VENDOR        CISCO-IOT

ATTRIBUTE           Cisco-MUD-URI       1       string

END-VENDOR CISCO-IOT

##########################################################################
########
```

6. Open usr/local/share/freeradius/dictionary file and locate the lines:

   $INCLUDE dictionary.motorola
   $INCLUDE dictinary.motrola.wimax

   and add the following on the next line:
   $INCLUDE dictionary.mudserver

7. To create a user, add a User-Name called <username> and Cleartext-Password <password> in /usr/local/etc/raddb/users file under the following lines:

   #bob Cleartext-Password := “hello”
   # Reply-Message := “Hello, %{User-Name}”

   The added line should look like the following example:
   megan Cleartext-Password := “testing123”

8. You may leave /usr/local/etc/raddb/clients.conf as it is if you are only working with localhost IP

9. Change the exec configuration file in /usr/local/etc/raddb/mods-enabled/exec from wait “no” to “yes”.

10. In /usr/local/etc/raddb/sites-enabled/default, add the following code in the “authorize” section after “filter_username”:

```
if (User-Name == "%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py ‘null’ 'U1'}.in") {
        update control {
                Auth-Type := Accept
        }
}

if (User-Name == "%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py 'null' 'U2'}.out") {
        update control {
                Auth-Type := Accept
        }
}
```

    Be sure that no newline character are inserted in the middle of the long commands

11. In the same file at the “post-auth” section add the below code after “exec”

```
        if (Cisco-MUD-URI) {
                if (User-Name == "<username>") {
                update reply {
                        Exec-Program = "%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py %{Cisco- MUD-
URI} 'W'}"
                        Cisco-AVPair := "ACS:CiscoSecure-Defined-ACL=%{exec:/usr/bin/python /usr/local/etc/controller/
mud_controller.py 'null' 'U1'}.in",
                        Cisco-AVPair += "ACS:CiscoSecure-Defined-ACL=%{exec:/usr/bin/python /usr/local/etc/controller/
mud_controller.py 'null' 'U2'}.out"
                        }
                }

        if (User-Name == "%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py 'null' 'U1'}.in") {
        update reply {
                User-Name = "%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py 'null' 'U1'}",
                Cisco-AVPair := "ip:inacl#1=%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py ‘null’ ‘R1’}”,
                Cisco-AVPair += "ip:inacl#2=permit udp any any eq 67",
                Cisco-AVPair += "ip:inacl#3=permit udp any any eq 68",
                Cisco-AVPair += "ip:inacl#4=%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py ‘null’ ‘R3’}”,
                Reply-Message += "DACL Ingress Downloaded Succesfully.”,
                        }
                }

        if (User-Name == "%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py 'null' 'U2'}.out") {
        update reply {
                User-Name = "%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py 'null'
'U2'}. out",
                Cisco-AVPair := "ip:outacl#1=%{exec:/usr/bin/python /usr/local/etc/controller/mud_controller.py 'null' 'R2'}",
                Cisco-AVPair += "ip:outacl#2=permit udp any any eq 67",
                Cisco-AVPair += "ip:outacl#3=permit udp any any eq 68", Cisco-AVPair += "ip:outacl#4=%{exec:/usr/bin/python
/usr/local/etc/controller/mud_controller.py 'null' 'R3'}",
                Reply-Message += "DACL Egress Downloaded Succesfully."
                        }
                }
        }
```

In this part, make sure that in the line that says “if (User-Name == “<username>)”, change <username> to the username you have assigned in the users file.

12. Replace the tls.c file under freeradius-server-release_3_0_11/src/main/ with the tls.c file from our github

13. Under freeradius-server-release_3_0_11/share/ open dictionary.freeradius.internal:

    Locate the line:
    ATTRIBUTE TLS-PSK-Identity 1933 string

    Add below it:
    ATTRIUBTE TLS-Client-Cert-Subject-Alt-Name-URI 1934 string

14. Change directories back to the FreeRADIUS release. Reconfigure, make, and install.

    ```
    $ cd ..
    $ ./configure
    $ make
    $ sudo make install
    ```

15. Attempt to start FreeRADIUS in debugging mode by running the following command:

    ```
    $ sudo radiusd -x
    ```

If this does not work, make sure that FreeRADIUS is not already running as a process. To check, run the following commands:

```
$ ps aux
find the pid of FreeRADIUS

$ kill -9 <pid of FreeRADIUS>
```

After this, you should be able to run FreeRADIUS

At enterprise side where MUD files are stored

1. In your home directory, make a project folder entitled “elear_mud”

2. Underneath this project directory, create a new directory named “mud’ and one called “private”

   ```
   $ mkdir mud private
   ```

   Follow this link to generate MUD file
   https://www.ofcourseimright.com/mudmaker/

   Once you have specified your options and details, click on submit button to generate the MUD file

   Copy and paste the mud string and paste it on new mud file named with .json extension and save it in the mud directory

3. In the mud folder, generate key and self-signed certificate using the following commands:

   ```
   $ openssl genrsa -out key.pem 2048
   $ openssl req -new -x509 -key key.pem -out ck.pem
   ```

4. Sign and verify the MUD file using the following commands:

   ```
   $ openssl cms -sign -in <name of MUD file>.json -signer ck.pem -inkey key.pem -binary
         -outform DER -out <name of MUD file>.p7s
   $ openssl cms -verify -in  <name of MUD file>.p7s -out mud.json -CAfile ck.pem -inform
         DER -content -<name of MUD file>.json
   ```

The output of the second command should be “Verification successful”

5. Move key.pem to the private folder

   ```
   $ mv key.pem ~/elear_mud/private
   ```

6. Change permissions on the private folder as well as the key

   ```
   $ chmod 700 ~/elear_mud/private
   $ chmod 700 ~/elear_mud/private/key.pem
   ```

7. Create a directory called “controller” under /usr/local/etc

8. Download and copy mud_controller.py file into /usr/local/etc/controller directory

9. Open the mud_controller.py file and comment out the last line that should be “read_json()”

10. Copy ck.pem to the controller directory by running the following command from the elear_mud/mud directory

    $ cp ck.pem /usr/local/etc/controller

11. Run SimpleHTTPServer in the elear_mud directory:

    $ python -m SimpleHTTPServer

This command will by default run the HTTP server on port 8000. If you wish, you can specify a port with your IP address by running the following command:

    $ python -m SimpleHTTPServer 127.0.0.1:8080

Testing MUD Controller

1. With SimpleHTTPServer running, run the following command in the controller directory:

    $ python mud_controller.py http://localhost:8000/mud/<name of MUD file>.json W

On success, the output will be “Verification successful” and the following three files should appear in the controller directory:

    <name of MUD fille>.json
    <name of MUD file>.p7s
    mud.json

The “W” option tells the controller to get the MUD file and signature file from the MUD URI and verify the signature and store the MUD file.

Testing the Radius Client

1. Start the radius server in debugging mode.

2. With SimpleHTTPServer still running, run the following command:

    $ echo “User-Name=<username>, User-Password=<password>, Cisco-MUD-URI=http://127.0.0.1:8000/mud/<name of MUD file>.json” | /usr/local/bin/radclient -x localhost:1812 auth testing123

Change <username> and <password> to username and password specified in the users file

3. On success, the output will indicate that an Access-Request Id was sent and an Access-Accept Id was received.