Wireless WPA-Enterprise Authentication Assignment

LT-2021-052 - Diwanga Amasith

Tasks done

- Configuring the password authentication with multi factor authentication in radius server.
- Use OPENLDAP as directory server
- Use google authenticator for MFA (TOTP)
- Use radtest tool in freeradius-utils package to test the connection with radius server.

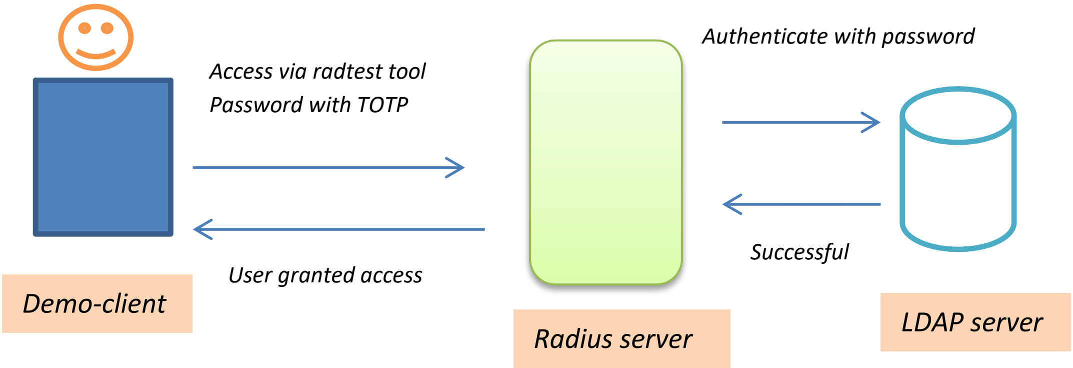


Fig 01: Simple network diagram

- Solution

Using only password based authentication is not secure. So using multifactor authentication is one of the solution for hardening the security with come with password.

To implement this on radius server I used Google authenticator. Which is implements two-step verification services using the Time-based One-time Password Algorithm and HMAC-based One-time Password algorithm for authenticating users of mobile applications by Google. Authenticator generates a six-digit one-time password which is valid for limited time and this must enter by user in addition to user credentials to login to system.

In here all servers are running on Ubuntu 20.04 LTS version, Further

- Google Authenticator PAM moduel
- FreeRADIUS
- OpenLDAP   are used

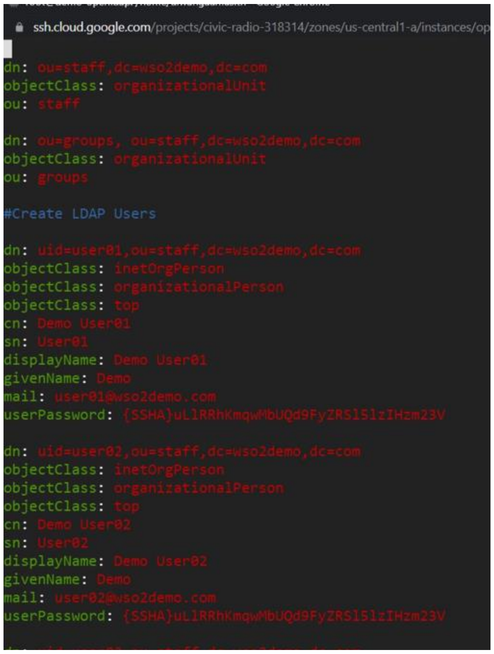
Steps :

1: **Configuring Open LDAP server**.

- Installing packages:

  apt-get update
  apt-get upgrade
  apt install slapd ldap-utils

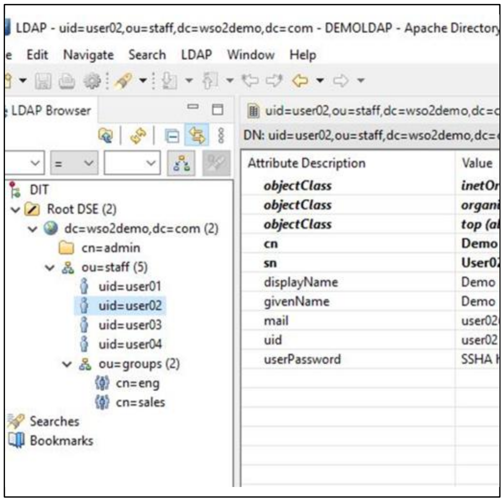
- Create Ldif file to create directory structure



ssh.cloud.google.com/projects/civic-radio-318314/zones/us-central1-a/instances/op

```
dn: ou=staff,dc=wso2demo,dc=com
objectClass: organizationalUnit
ou: staff

dn: ou=groups, ou=staff,dc=wso2demo,dc=com
objectClass: organizationalUnit
ou: groups

#Create LDAP Users

dn: uid=user01,ou=staff,dc=wso2demo,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: top
cn: Demo User01
sn: User01
displayName: Demo User01
givenName: Demo
mail: user01@wso2demo.com
userPassword: {SSHA}uLlRRhKmqwMbUQd9FyZRSl5lzIHzm23V

dn: uid=user02,ou=staff,dc=wso2demo,dc=com
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: top
cn: Demo User02
sn: User02
displayName: Demo User02
givenName: Demo
mail: user02@wso2demo.com
userPassword: {SSHA}uLlRRhKmqwMbUQd9FyZRSl5lzIHzm23V
```

- Creating Structure
  ldapadd -x -D cd=admin,dc=wso2demo,dc=com -W -f ldif-structure.ldif
- Use **slapcat** command to verify structure
- Connect Apache Directory Studio for visualization of server



LDAP - uid=user02,ou=staff,dc=wso2demo,dc=com - DEMOLDAP - Apache Directory

uid=user02,ou=staff,dc=wso2demo,dc=c

DN: uid=user02,ou=staff,dc=wso2demo,dc=

```
DIT
  Root DSE (2)
    dc=wso2demo,dc=com (2)
      cn=admin
      ou=staff (5)
        uid=user01
        uid=user02
        uid=user03
        uid=user04
        ou=groups (2)
          cn=eng
          cn=sales
Searches
Bookmarks
```

Attribute Description	Value
objectClass	***inetOr***
objectClass	***organi***
objectClass	***top (a***
cn	**Demo**
sn	**User0**
displayName	Demo
givenName	Demo
mail	user02
uid	user02
userPassword	SSHA I

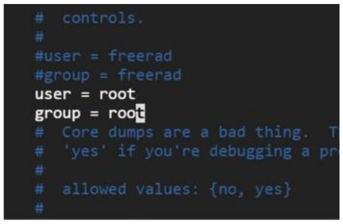
- For this I have to create firewall rule in gcp vpc. (port 389 tcp&udp)

2. **Configuring radius server**

- Get another ubuntu 20.04 LTS version installed vm from gcp.
- Installing packages
  apt-get update
  apt-get upgrade
  apt-get install freeradius freeradius-common
  apt-get install freeradius-utils freeradius-ldap
  apt-get install libpam-google-authenticator
  apt-get install libpam-google-authenticator

- **Configuring server files**

  1. `/etc/freeradius/3.0/radiusd.conf file`



```
#  controls.
#
#user = freerad
#group = freerad
user = root
group = root
#  Core dumps are a bad thing.  T
#  'yes' if you're debugging a pr
#
#  allowed values: {no, yes}
#
```

Comment existing freerad user and group and add root.
FreeRADIUS must run as root for this to work. The reason for this is so that FreeRADIUS can access the .google_authenticator token in each home directory. Otherwise FreeRADIUS does not have access.
You can find user and group inside security block.

  2. `/etc/freeradius/3.0/sites-enabled/default`

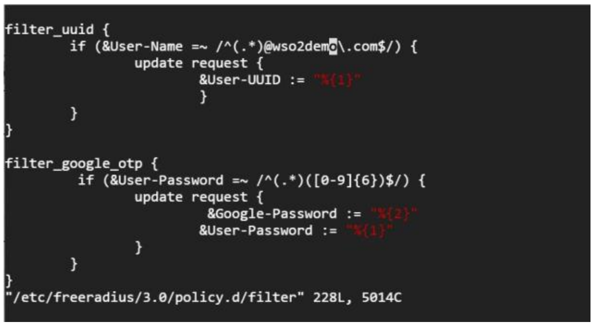
     In here setting authorization and authentication on FreeRadius server with PAM.

- In authorize block
  - delete "-" before ldap to read passward from LDAP database.
  - Add custom filters (filter_uuid, filter_google_otp) to get uuid and google otp.
- In authenticate block add this

```
Auth-Type PAP {
          pap
     if (&Google-Password) {
                update request {
          &User-Name := "%{&User-UUID}"
               &User-Password := "%{&Google-Password}"
               }
               pam
          } else {
               update reply {
                    Reply-Message := "Login incorrect: TOTP Fail"
          }
               reject
               }
     }
```

  3. `/etc/freeradius/3.0/policy.d/filter`

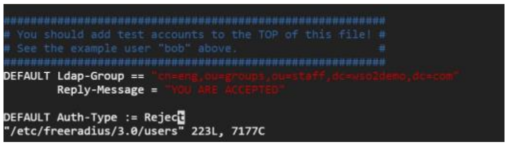
     Implement custom filters.



```
filter_uuid {
        if (&User-Name =~ /^(.*)@wso2demo\.com$/) {
                update request {
                        &User-UUID := "%{1}"
                        }
        }
}

filter_google_otp {
         if (&User-Password =~ /^(.*)([0-9]{6})$/) {
                update request {
                         &Google-Password := "%{2}"
                        &User-Password := "%{1}"
                }
        }
}
"/etc/freeradius/3.0/policy.d/filter" 228L, 5014C
```

4. /etc/freeradius/3.0/users

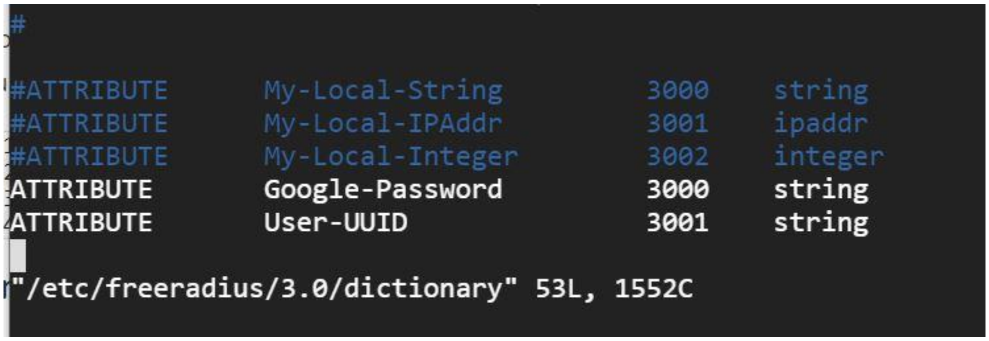
In here we specify only one group users in lDAp is authenticate for service. We get eng and sales group. We set only eng group members can authenticate from this. Add this lines to end of file.



```
############################################################
# You should add test accounts to the TOP of this file!  #
# See the example user "bob" above.                       #
############################################################
DEFAULT Ldap-Group == "cn=eng,ou=groups,ou=staff,dc=wso2demo,dc=com"
        Reply-Message = "YOU ARE ACCEPTED"

DEFAULT Auth-Type := Reject
"/etc/freeradius/3.0/users" 223L, 7177C
```

4. /etc/freeradius/3.0/dictionary

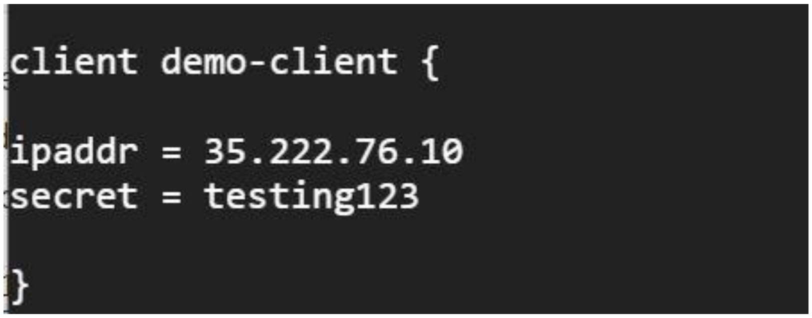
Adding new attributes for usage of server processes.



```
#

#ATTRIBUTE      My-Local-String         3000    string
#ATTRIBUTE      My-Local-IPAddr         3001    ipaddr
#ATTRIBUTE      My-Local-Integer        3002    integer
ATTRIBUTE       Google-Password         3000    string
ATTRIBUTE       User-UUID               3001    string

"/etc/freeradius/3.0/dictionary" 53L, 1552C
```

5. /etc/freeradius/3.0/clients.conf

Add demo-client details after the client localhost to connect radius server from demo-client.

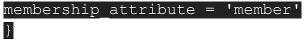


```
client demo-client {

ipaddr = 35.222.76.10
secret = testing123

}
```

6. /etc/freeradius/3.0/mods-available/ldap

In here we configure OpenLdap details with FreeRaduis server.

```
ldap {
server = 'demo-openldap.wso2demo.com'
identity = 'cn=admin,dc=wso2demo,dc=com'
password = diwanga
base_dn = 'ou=staff,dc=wso2demoe,dc=com'
user {
  base_dn = 'ou=staff,dc=wso2demo,dc=com'
filter = "(mail=%{%{Stripped-User-Name}:-%{User-Name}})"
}
group {
base_dn = "ou=groups,ou=staff,dc=wso2demo,dc=com"
filter = '(objectClass=GroupOfNames)'
membership_filter = "(|(&(objectClass=GroupOfNames)(member=%{control:Ldap-UserDn}))(&(objectClass=GroupOfNames)(member=%{control:Ldap-UserDn})))"
```



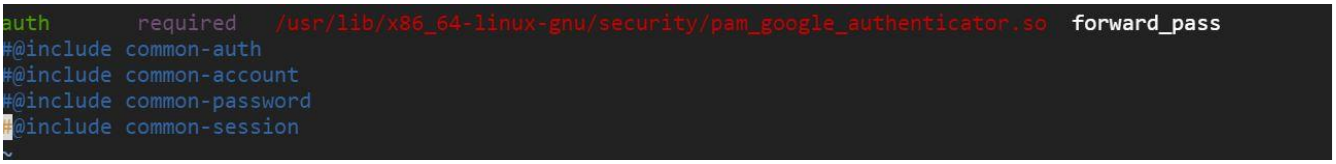
```
membership_attribute = 'member'
}
```

In here, we give ldap server credentials tomake connection for free radius server
And also we use User mail attribute for identyfiing users uniquely. Further inside group we user "member" as membership attribute.

For enabling this files we have to make symlink from mod-available to
mode-enable like nginx configuration.
ln -s /etc/ freeradius/3.0/mods-available/ldap /etc/freeradius/3.0/mod-enabled

7. **Confuguring Google Authenticator PAM with FreeRADIUS**

   In `/etc/pam.d/radius` file comment existing active lines and add these lines,



```
auth       required   /usr/lib/x86_64-linux-gnu/security/pam_google_authenticator.so  forward_pass
#@include common-auth
#@include common-account
#@include common-password
#@include common-session
~
```

Like previously we have to enable this module.

ln –s /etc/freeradius/3.0/mod-available/pam.d   /etc/freeradius/3.0/mod-enabled

8. **Restart the radius server**

```
systemctl restart freeradius.service
```

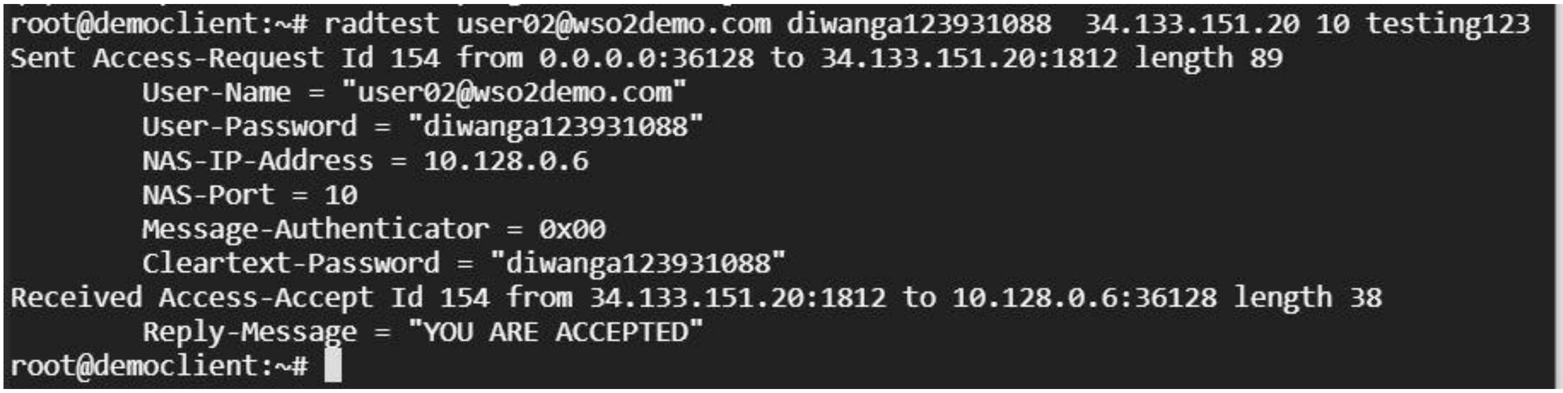
`for debugging we start server in debug mode`

```
freeradius -XXX
```

3. Testing with Demo-Client

For this we have to open new ssh window from radius server and add the users of same as ldif usernames. After that we have to configure Google Authenticator on that user. By doing that, keep save secret key, verification code and emergency scratch cod.
After that you have TOTP with your android app. You can Test your connection with radius server by using radtest tool come with freeradius-utils. You have to install this in to demo-client.

When You give user password and TOTP with radius server access password for client, you can see this output.



```
root@democlient:~# radtest user02@wso2demo.com diwanga123931088  34.133.151.20 10 testing123
Sent Access-Request Id 154 from 0.0.0.0:36128 to 34.133.151.20:1812 length 89
        User-Name = "user02@wso2demo.com"
        User-Password = "diwanga123931088"
        NAS-IP-Address = 10.128.0.6
        NAS-Port = 10
        Message-Authenticator = 0x00
        Cleartext-Password = "diwanga123931088"
Received Access-Accept Id 154 from 34.133.151.20:1812 to 10.128.0.6:36128 length 38
        Reply-Message = "YOU ARE ACCEPTED"
root@democlient:~#
```

Challenges

- Get to familiarize about new technologies. Radius server,Google Authenticator for the I read Documentation of the FreeRADIUS and watch some youtube videos.
- After all configuration are done, server was run with out giving error. But every time debug console given ERROR Authentic failure. I see /var/log/auth.log but no information

could found. To over come this first I read Debug messages and clarify these is have to done some modification in filter file. After that All are working. This get a Half a day.

What I learned

- I got a lot of experience with working on this Assignment. Specially this radius server is new to me. And also I got much more Understanding about OPENLDAP. Google Authenticator is new thing to me and I willing to apply this to other protocols near future.
- And also I get much experience with GCP with working 3vm instance at once in first time. Making firewall rules also give me border experience.

- radius configurations

1. /etc/freeradius/3.0/radiusd.conf file
   https://github.com/Diwanga/Wireless-Authentication/blob/master/radiusd.conf
2. /etc/freeradius/3.0/sites-enabled/default
   https://github.com/Diwanga/Wireless-Authentication/blob/master/sites-available/default
3. /etc/freeradius/3.0/policy.d/filter
   https://github.com/Diwanga/Wireless-Authentication/blob/master/policy.d/filter
4. /etc/freeradius/3.0/users
   https://github.com/Diwanga/Wireless-Authentication/blob/master/mods-config/files/authorize
5. /etc/freeradius/3.0/dictionary
   https://github.com/Diwanga/Wireless-Authentication/blob/master/dictionary
6. /etc/freeradius/3.0/clients.conf
   https://github.com/Diwanga/Wireless-Authentication/blob/master/clients.conf
7. /etc/freeradius/3.0/mods-available/ldap
   https://github.com/Diwanga/Wireless-Authentication/blob/master/mods-available/ldap

8. Ldif structure
   https://github.com/Diwanga/Wireless-Authentication/blob/master/ldifstructure.ldif