SCENARIO

As a measure to use a reliable OTP generator other than the current Forti-Token, after a we came up with a solution of using Google-Authenticator which works on time-based One-time Password Algorithm and HMAC-based One-time Password algorithm, for authenticating users of software applications. The concatenation of end user's AD's password and the OTP is used as the password to authenticate the SSL VPN connection and the username used for such connection is the actual domain user specified including the domain e.g. "abc.def@sanimabank.com".

For a scenario, if an Active Directory account is "abc.def@sanimabank.com", the AD account's password is gHi@123 and the code in Google Authenticator app is 666 666. The **username** for SSL VPN authentication will be "**abc.def@sanimabank.com**" and **password** will be **gHi@123666666**

WORKING MECHANISM

To achieve this system, a RADIUS Server has been deployed which handles the mechanism for creating as well as checking the two-factor-authentication logins. This RADIUS server has Google-Authenticator PAM (Pluggable Authentication Module) installed in it and is also made to communicate to our AD server for creating and authenticating AD user's username and password. RADIUS server's authentication module is configured in such a manner that any radius user's authentication request should pass the Google-Authentication verification process.

The working mechanism of the system can be understood through the diagram and the respective explanation for each logical steps:

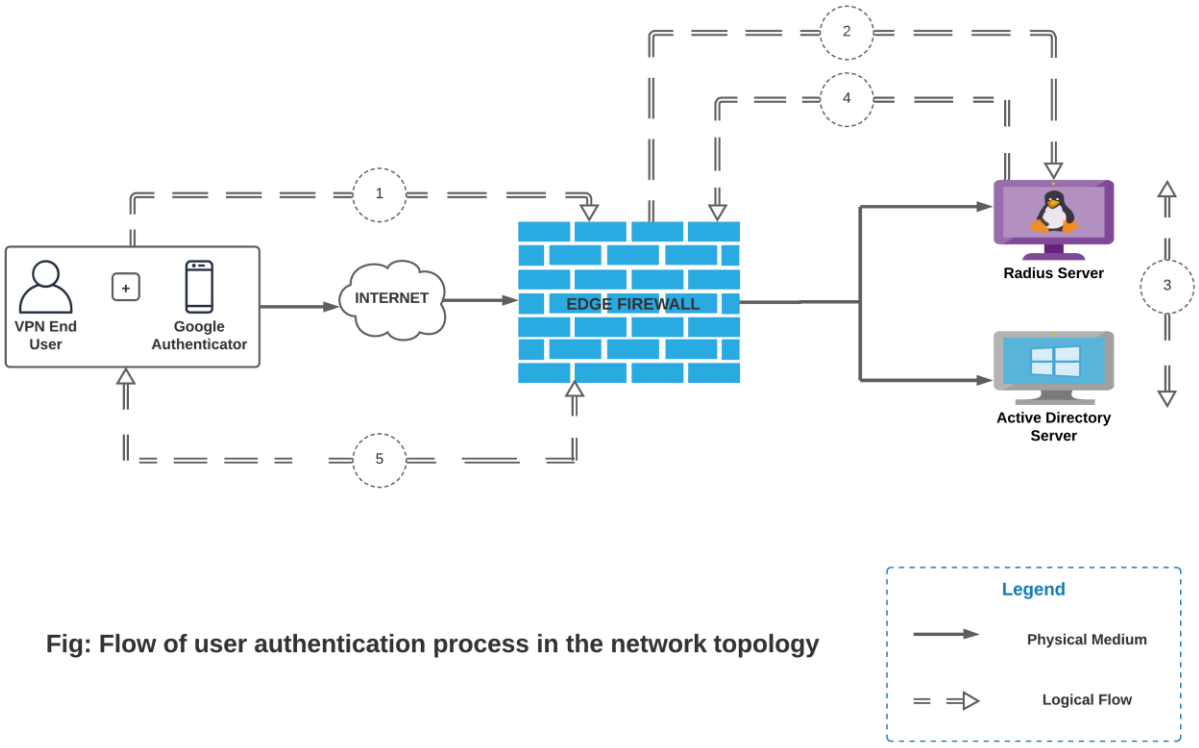


Fig: Flow of user authentication process in the network topology

1. The end user creates a set of username and password+code and requests the firewall for authentication of VPN. For Google-Authentication OTP, the user is pre-provided with a secret key generated for the same user on the radius server. This secret key once entered on the user's Authenticator mobile app, keeps on generating time based OTP tokens.
2. The firewall after receiving the request, forwards username and password+code to the RADIUS server.
3. The radius server then looks for the entry of the mentioned username into its own user list. Once matched, using Kerberos, the server queries the windows AD for password verification. In the meantime, the "code" part of the entered password is also checked by the RADIUS server. Upon verification of both i.e. AD's password + OTP, a success or failure message is generated by the PAM of RADIUS server.
4. The success or failure message is conveyed from RADIUS server to the firewall.
5. If success, VPN tunnel is established.

BACKEND

1. RADIUS INSTALLATION AND CONNECTIVITY WITH AD

The RADIUS server is the main part of this functioning system. A free radius server is installed on a Linux OS into a Virtual Machine (versions and other technical details in later section). The SSSD is then configured in order to let the RADIUS communicate to AD. The System Security Services Daemon (SSSD) is a linux system service which allows to access remote directories and authentication mechanisms. Through this, the RADIUS is now made to join with the AD for future authentications. It uses realm command for this operation. Realm is a command line tool that can be used to manage enrollment in Kerberos realms, like Active Directory domains.

```
[root@centos7radius ~]# realm join sanimabank.com
```

If domain available, by default, the execution of this command prompts for Administrator's password.

```
Password for Administrator:
```

Entering the Administrator's password, RADIUS is now set for AD user's authentication by communicating via Kerberos with the AD. **In this solution as we use SSSD to integrate with the AD, the account information is only stored on AD server, it won't be synchronized to RADIUS server or Firewall**.

2. INSTALLING GOOGLE-AUTHENTICATION PAM

The Google-Authentication PAM is installed and the RADIUS's PAM is configured to use the Google-Authenticator's PAM routine to make authentication verification. The PAM configuration for RADIUS is located in the file location shown below:

```
[root@centos7radius ~]# vi /etc/pam.d/radiusd
```

The content of this file needs to be changed in order for the authentication method to use the 2 FA system. In normal condition the content of this file is:

```
#%PAM-1.0
#auth          include      password-auth
#account       required     pam_nologin.so
#account       include      password-auth
#password      include      password-auth
#session       include      password-auth
```

Now, we have updated this to use the Google-Authentication's PAM routines for 2 FA. The content of updated file looks as:

```
#%PAM-1.0
auth requisite /usr/local/lib/security/pam_google_authenticator.so forward_pass
auth required pam_sss.so use_first_pass
account     required      pam_nologin.so
account     include       password-auth
password    include       password-auth
session     include       password-auth
~
```

This ensures that every authentication request handled by the RADIUS now also looks for verification in Google-Authentication's PAM routines before making the final decision on the authentication. Also SSSD PAM is present in this file which handles the AD authentication process. Hence, now any authentication request must now pass a password comprising of AD password as well as Google-Authentication OTP.

3. CONFIGURING GOOGLE-AUTHENTICATION FOR AD USERS

Now that the RADIUS mandatorily looks for Google-Authentication OTP code for authentication verification, the user must be provided with this OTP. To do so, firstly, a home directory for the respective user should be created in the RADIUS server.

```
[root@centos7radius ~]# su - abhishek.poudel@sanimabank.com
Last login: Tue Dec  1 14:20:14 +0545 2020 on pts/1
[abhishek.poudel@sanimabank.com@centos7radius ~]$
```

For valid users, upon execution of above command, a directory for the user is created in location "/home/". We can switch to any valid user of AD. If, the user doesn't exists on the AD, an error is returned.

```
[root@centos7radius ~]# su - abhishekll.poudel@sanimabank.com
su: user abhishekll.poudel@sanimabank.com does not exist
[root@centos7radius ~]#
```

Having created a directory, RADIUS is ready to serve for this specific user. But, still it needs 2 FA via Google-Authentication as per the PAM settings. For Google-Authentication, a simple command "google-authenticator" inside the user's home directory is sufficient. This command creates a QR code as well as

a secret key for that particular user, any of which can be used by the end user for creating OTP tokens via Google's Authenticator mobile app.

```
[abhishek.poudel@sanimabank.com@centos7radius ~]$ google-authenticator
```

Having executed the above command, we get output somewhat like this:

Either of the QR or secret key can be now used by the end user in their Google's Authenticator mobile app.

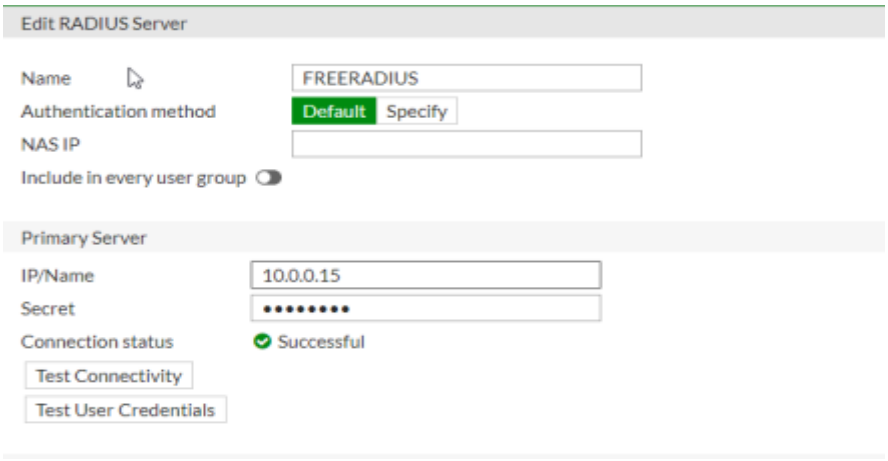
4. CONFIGURING FIREWALL

In the firewall part, in the RADIUS server section, an entry for this RADIUS server is made. The connection between firewall and RADIUS server starts after agreeing on a secret key which present on the RADIUS server for clients configuration. Inside the /etc/raddb/clients.conf file entries for the clients that can use this RADIUS server, is made. The entry is made in this file by supplying the following details in same code structure as below:

client 10.0.0.15{

ipaddr = 10.0.0.15
secret = abcdef
require_message_authenticator = no
nas_type = other
}

The IP address is of the Firewall and the secret key specified here is used to add the radius server on our Firewall.



5. REQUEST FOR VPN AUTHENTICATION

Having gone through the previous steps, the end user now needs to connect to SSL VPN. The username for such connection is now the actual domain user specified in full (including domain), which is actually present as a user in RADIUS server. And, for the password, the "AD password + OTP code" must be supplied.

OS AND VERSIONS DETAILS

- RADIUS SERVER: FreeRADIUS Version 3.0.13
- OS for RADIUS Server: CentOS 7. Kernel version：3.10.0-862.el7.x86_64
- RADIUS Server's IP Address: 192.168.3.20
- RADIUS Service port: 1812