Protecting SSH via OTP/2FA Authentication using the SecureAuth IdP Appliance

Summary / Overview

Using the SecureAuth appliance as a Radius server, allow OTP authentication for SSH clients into a Linux/Unix estate. The OTP replaces the local password.

Additional configuration is covered to show a 2FA experience with OTP and local password.

SecureAuth Version

Configured on 7.4.3

Description / Definitions

With the SecureAuth appliance acting as a Radius server, we can authenticate requests from any Radius Client. This allows us to provide OTP mechanisms for environments that provide Radius clients. In this example Linux is used with a simple pam_radius configuration to achieve OTP authentication, while the username is checked against the on premise data store by the SecureAuth appliance.

Pre-Requisites / Requirements

- SecureAuth configured as Radius Server (Follow the instructions within radius config to configure the SecureAuth appliance as a Radius Server)
- An enterprise data store user registered to use OTP
- Linux user with an equivalent username in the enterprise data store
- A Linux server – this documentation expects a Red Hat variant (OpenSuse was used here).
- Configured SecureAuth realm for the radius endpoint – as documented in the Radius instructions.
- SSHD running and connectivity tested from a suitable SSH client

Configuration Steps

- Configure Linux for PAM Radius
- Configure SecureAuth IDP to act as a Radius Server
- Configure PAM Radius Client

Configure Linux (OpenSuse instructions below)

From a terminal, as root - enter yast
a. Select software
b. Select software management
c. In the search phrase enter – “pam”
d. Select as a minimum :-
   • Pam
   • Pam-config
   • Pam-doc (for light reading)
   • Pam-radius
   • Pam-modules
   • Pam-ssh

   • Select Accept and let the software packages install
   • Quit yast

   Additionally if you want to build out the integration further, pam offers modules for LDAP and Kerberos

Once installed the remaining Linux configuration consists of two main files:

   • /etc/pam.d/sshd
   • /etc/raddb/server

Configure /etc/pam.d/sshd

Configure sshd to utilise pam_radius as its first authentication module. Example settings shown below:
Importantly we include the auth module config at the top of the file, set the requirement to sufficient and point the module to the correct library (/lib/security/pam_radius_auth.so).

If unsure of the path to the module execute: find . -name "radius_auth.so" from a terminal as root to find it.

*The setting of “sufficient” does not force the OTP to be required. Meaning you can fall back to local passwords during testing.*

Configure /etc/raddb/server

Enter your Radius server ip address, shared secret and timeout.

Example shown below:

```
# pam_radius_auth configuration file. Copy to: /etc/raddb/server
#
# For proper security, this file SHOULD have permissions 0600, that is readable by root, and NO ONE else. If anyone other than root can read this file, then they can spoof responses from the server!
#
# There are 3 fields per line in this file. There may be multiple lines. Blank lines or lines beginning with "#" are treated as comments, and are ignored. The fields are:
#  server[:port] secret [timeout]
#  the port number or number is optional. The default port name is "radius", and is looked up from /etc/services. The timeout field is optional. The default timeout is 3 seconds.
#  If multiple RADIUS server lines exist, they are tried in order. The first server to return success or failure causes the module to return success or failure. Only if a server fails to respond is it skipped, and the next server in turn is used.
#  The timeout field controls how many seconds the module waits before deciding that the server has failed to respond.
#
server[::port] shared_secret  timeout (s)
127.0.0.1  secret 1
192.168.93.134 testing123  50

having localhost in your radius configuration is a Good Thing.
#
# See the INSTALL file for pam.conf hints.
```

Make sure that no firewalls on the Linux server or SecureAuth server are blocking port communication...

- Radius server listens on ports 1812 and 1813 by default.
- SSH daemon listens on port 22 by default.
- OpenSuse allows firewall configuration via yast.

This completes the configuration on the Linux server, restart the ssh daemon.

Execute:

- rcsshd restart

to (re-)start the daemon from a terminal window.
SecureAuth Radius Config

Complete the Radius Server configuration to allow the Linux server to connect as a radius client. Example configuration is below:

- The endpoint is your SecureAuth realm
- Shared secret must match the radius client config file
- The ip addresses of the allowed Radius clients and supported Radius actions are listed.

Example Workflow:

- User name in the Enterprise Directory Store (AD in this case) is: tchi
- User name on Linux server is: tchi
- Linux password has never been communicated to the Linux user
- Linux user attempts to access Linux server via SSH client (Putty in this example)
- Enters username as normal
- Password now accepts the OTP
- Successful entry of the OTP allows SSH access

The current configuration will allow ssh access to users who already know their local Linux password. Radius will be attempted and fail over to local authentication. New users need not be informed of their local Linux password only the username.

*Configuration can be applied as required to ensure all users need an OTP. See next section.*

2FA Authentication – OTP and local password

To enable the use of the local password and OTP a small change is required to the /etc/pam.d/sshd configuration file.

We simply change the order of the authentication module and change it to be required as shown below:

*(Note the use of the debug line to add extra info to /var/log/messages. Remove in production)*
To restart sshd, use the command `rcsshd restart`.

We can now run through the workflow using an OTP and local Linux password.

*Important to note that ssh configuration must accept challenge response authentication which is default behaviour. If keys are required the configuration needs to change to reflect this.  

Example Workflow:

- User name in the Enterprise Directory Store (AD in this case) is: tchi
- User name on Linux server is: tchi
- Linux password is known by the end user
- Linux user attempts to access Linux server via SSH client (Putty in this example)
- Enters username as normal
- OTP is entered first
- Local password is entered
- Access is granted
Note the two password prompts in the screenshot above. The first one is for the OTP, the second is for the local password.

Username check is still performed centrally against the enterprise data store, configured on the SecureAuth realm.

The flow can be seen in var/log/messages as the check is performed against Radius and then locally.

Radius authentication below using OTP:

Then local password below:
Troubleshooting / Common Issues

Check the stdout log files from the SecureAuth Radius server for connection information and failure reasons.

Example of a user not existing in the enterprise data store but existing on the Linux server:

```
$ tail -f /var/log/messages
```

SSH logging in OpenSuse is configured by default to use syslogd which points the SSH entries to /var/log/messages. It is easy to setup logs just for SSH and point syslogd to use those instead to aid troubleshooting.