Table of Contents

OpenSSH Hardening

Introduction

This document describes a possible way of achieving PFS over SSH while using 3 factor authentcation with Yubico OTP, Publickey and password authentication protected by a ECDH over Curve25519 with SHA2 key exchange. A pretty secure, yet compatible fallback (diffie-hellman-group-exchange-sha256 key exchange) is implemented, too. Minimum OpenSSH version required is v6.5p1 (6.2p1 has the AuthenticationMethods option while the used key exchange algorithms got added in 6.5p1). Additional details can be obtained by reading the websites linked at the end of this document.

We only cover the configuration aspects of the whole process involved. For example, these topics are not included at all:

- Building and installing OpenSSH v6.5p1+
- Customizing your YubiKey
- Building and/or installing the Yubico PAM module

You'll have to figure out these for yourself. Also, the instructions below might be kinda Debian-ish. For Raspbian (and maybe others), you'll have to manually compile the latest version of the Yubico tools, see this document for details.

If you're on a Raspberry Pi, please follow these steps before (re-)creating the SSH keys below. This should greatly enhance entropy.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design, and manufacturing. The author shall have no liability for any error or damages of any kind resulting from the use of this document. There is no warranty; not even for merchantability or fitness for a particular purpose.

Server side

Renegerate moduli file

Regenerate your `/etc/ssh/moduli` file by using these commands:

```
mv /etc/ssh/moduli /etc/ssh/moduli.old
ssh-keygen -G "${HOME}/moduli" -b 4096
ssh-keygen -T /etc/ssh/moduli -f "${HOME}/moduli"
rm "${HOME}/moduli"
```

This took 1hr35min on a 2.9Ghz QuadCore i7 and around 3 days on a Raspberry Pi B+. If you have a

`/etc/moduli` file and you don't want to do this, at least edit your current version and remove all lines where the 5th column is less than 2000.

Yubico PAM

Get a fresh Yubico API key.

Edit `/etc/pam.d/common-auth` and add the 'try_first_pass' param to the 'pam_unix.so' entry so it looks like this:

```
auth    [success=1 default=ignore]  pam_unix.so try_first_pass nullok_secure
```

Edit `/etc/pam.d/sshd` and add this to the top:

```
auth       required     pam_yubico.so authfile=/etc/yubikeys
id=<api_user_id> key=<api_key>
```

In case you use your own authentication server, refer to this document for the additionally required PAM parameters.

You may monitor your `/var/log/auth.log` file and/or add 'debug' as an additional parameter to each of the above mentioned pam lines and create a file called `/var/run/pam-debug.log` (chmod 666) to obtain additional debug information.

Create a file called `/etc/yubikeys` (chmod 644) and fill it with your YubiKey ID to username mappings (one per line) like this:

```
root:cccccccabcd:cccccccbcde
flo:cccccccabcd
```

Generally, the pattern is:

```
<username>:<yubikey 1 identity>[:<yubikey 2 identity>...]
```

Host keys

Generate new host keys like this:

```
cd /etc/ssh
rm ssh_host_*key*
ssh-keygen -t ed25519 -f ssh_host_ed25519_key < /dev/null
ssh-keygen -t rsa -b 4096 -f ssh_host_rsa_key < /dev/null
```

Config changes

Edit `/etc/ssh/sshd_config` to include these settings:

```
Protocol 2
HostKey /etc/ssh/ssh_host_ed25519_key
HostKey /etc/ssh/ssh_host_rsa_key
KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-
sha256
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-
gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-
ripemd160-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-
sha2-256,hmac-ripemd160,umac-128@openssh.com
PasswordAuthentication yes
PubkeyAuthentication yes
AuthenticationMethods publickey,password
```

If you want to login as root like this, you'll also have to enable this:

```
PermitRootLogin yes
```

Other stuff

In case your sshd init script contains code to automatically regenerate deleted host keys, you might want to comment out these parts.

Client side

YubiKey

Make sure you have your YubiKey ready The cheapest model comes in at about 23EUR, which is pretty affordable.

Key pairs

Generate the key pairs using the following commands:

```
ssh-keygen -t ed25519 -o -a 100
ssh-keygen -t rsa -b 4096 -o -a 100
```

Provide strong passphrases, ideally not the same for both keys. Obviously, you'll have to copy the public keys over to the server on a secure channel.

Config changes

Edit either `/etc/ssh/ssh_config` or your `~/.ssh/config` file to include these settings:

```
Host *
  PasswordAuthentication yes
  PubkeyAuthentication yes
  KexAlgorithms curve25519-sha256@libssh.org,diffie-hellman-group-exchange-sha256
  Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr
  MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-ripemd160-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-ripemd160,umac-128@openssh.com
  RekeyLimit 1024K 5m
```

Conclusion

This setup enables PFS over SSH while incooperating multi-factor authentication using PKI, YubiKey OTP and unix password. This means for every login, a valid private key has to be provided, after which the user password followed by the YubiKey OTP is requested by the server.

Under some circumstances it might be more practical to have only 2FA using OTP and PKI. To achieve this, slight modifications to the steps above have to be made:

- `/etc/pam.d/common-auth` is not modified at all.
- `/etc/pam.d/sshd` is modified to remove the inclusion of `/etc/pam.d/common-auth`.
- `/etc/ssh/sshd_config` is modified to include `ChallengeResponseAuthentication yes`, `PasswordAuthentication no` and `AuthenticationMethods publickey,keyboard-interactive`

Establishing these modifications will require a valid private key and ask for the YubiKey OTP afterwards.

Sources

https://stribika.github.io/2015/01/04/secure-secure-shell.html
https://developers.yubico.com/yubico-pam/
http://ejohansson.se/archives/2014/01/22/ssh-login-with-yubikey-password-on-debian/
http://lwn.net/Articles/544640/
https://www.digitalocean.com/community/tutorials/how-to-protect-ssh-with-two-factor-authentication
https://security.stackexchange.com/questions/41941/consequences-of-tampered-etc-ssh-moduli
http://chrisdrake.com/safe-with-google-authenticator-think-again/

crypto, openssl, openssh, pam, yubikey, pfs