Analysis of Apple Unified Logs: Quarantine Edition [Entry 6] – Working From Home? Remote Logins — mac4n6.com

Published: 2020-04-30 · Archived: 2026-04-05 20:19:02 UTC

I'm sure many of us are working remote right now possibly using some of these remote capabilities. Remote Logins can include a few different services; SSH and Screen Sharing are two that I'll show here. These services are disabled by default and would need to be turned on in the user's Sharing preferences.

When Remote Login is turned on in the Sharing preferences, the system will have an SSH server enabled. Let's take a look at what an incoming SSH connection might look like first for a user account on the system that does not have this option turned on (janedoe). We are looking for the entries for the process 'sshd'.

```
log show --predicate 'process = "sshd"'
```

One entry to key in on is the "user account has expired". A user attempted to use SSH to login to this system using the 'janedoe' account coming from IP 192.168.1.170, however the connection failed.

Now on a system that does have remote login turned on. This first example shows an incorrect password attempt.

And a correct password attempt and login.

Connections can of course be incoming or outgoing. If the user were trying to access another system it might look like this. Not a whole lot unfortunately.

```
log show --info --predicate 'process = "ssh" or eventMessage contains "ssh"'
```

…and when the connection closes.

Screen Sharing is another service that needs to be explicitly enabled in the Sharing preferences. Incoming connections will show the user who logged in and where they came from. The example below shows an incorrect password that failed, and another that was correct. I've only queried for messages that contain the text 'Authentication:'. Looking for all messages associated with the 'screensharingd' process will be quite verbose with some metadata about the session.

```
log show --predicate 'process = "screensharingd" and eventMessage contains "Authentication:"'
```

Outgoing connections, like incoming connections, can be verbose. The process is ‘Screen Sharing’ like the application name.

```
log show --info --predicate 'process = "Screen Sharing"'
```

I might do a specific filter for ‘connect’ and ‘disconnect’ in the messages to see multiple sessions over time.
