Beyond the good ol' LaunchAgents - 18 - X11 and XQuartz

theevilbit.github.io/beyond/beyond_0018

June 28, 2021

> This is part 18 in the series of "Beyond the good ol' LaunchAgents", where I try to collect various persistence techniques for macOS. For more background check the introduction.

I learned about XQuartz while reading Armin Briegel's macOS Terminal and shell book. It's one of the alternative third party terminals we can install on macOS. As most terminals, this one also offers unique options to persist on the system.

X11 used to be part of OS X, till 10.7, and it was open source, which we can find here.

> The XQuartz project is an open-source effort to develop a version of the X.Org X Window System that runs on macOS.

This X window environment supports various initialization script. They can be found under the `/opt/X11/etc/X11/xinit` directory.

```
csaby@bigsur ~ % ls -lR /opt/X11/etc/X11/xinit
total 8
drwxr-xr-x  5 root  wheel  160 Jun 26 12:50 privileged_startx.d
-rw-r--r--  1 root  wheel  957 Apr 25 15:08 xinitrc
drwxr-xr-x  5 root  wheel  160 Apr  7 23:31 xinitrc.d

/opt/X11/etc/X11/xinit/privileged_startx.d:
total 24
-rwxr-xr-x  1 root  wheel  2263 Apr 25 15:08 10-tmpdirs
-rwxr-xr-x  1 root  wheel  1561 Apr 25 15:08 20-font_cache
-rwxr-xr-x  1 root  wheel    32 Jun 26 12:52 x.sh

/opt/X11/etc/X11/xinit/xinitrc.d:
total 24
-rwxr-xr-x  1 root  wheel  638 Apr  7 23:31 10-fontdir.sh
-rwxr-xr-x  1 root  wheel  157 Jan  9 14:57 98-user.sh
-rwxr-xr-x  1 root  wheel  297 Jan  9 14:57 99-quartz-wm.sh
```

The `xinitrc` script is similar to the other shell startup files I discussed in my previous post. It's owned by root, but we can make a copy of it to our HOME directory, name it `.xinitrc` and it will be executed upon opening `xterm`.

`.xserverrc` is another similar file, that will be consumed by this environment.

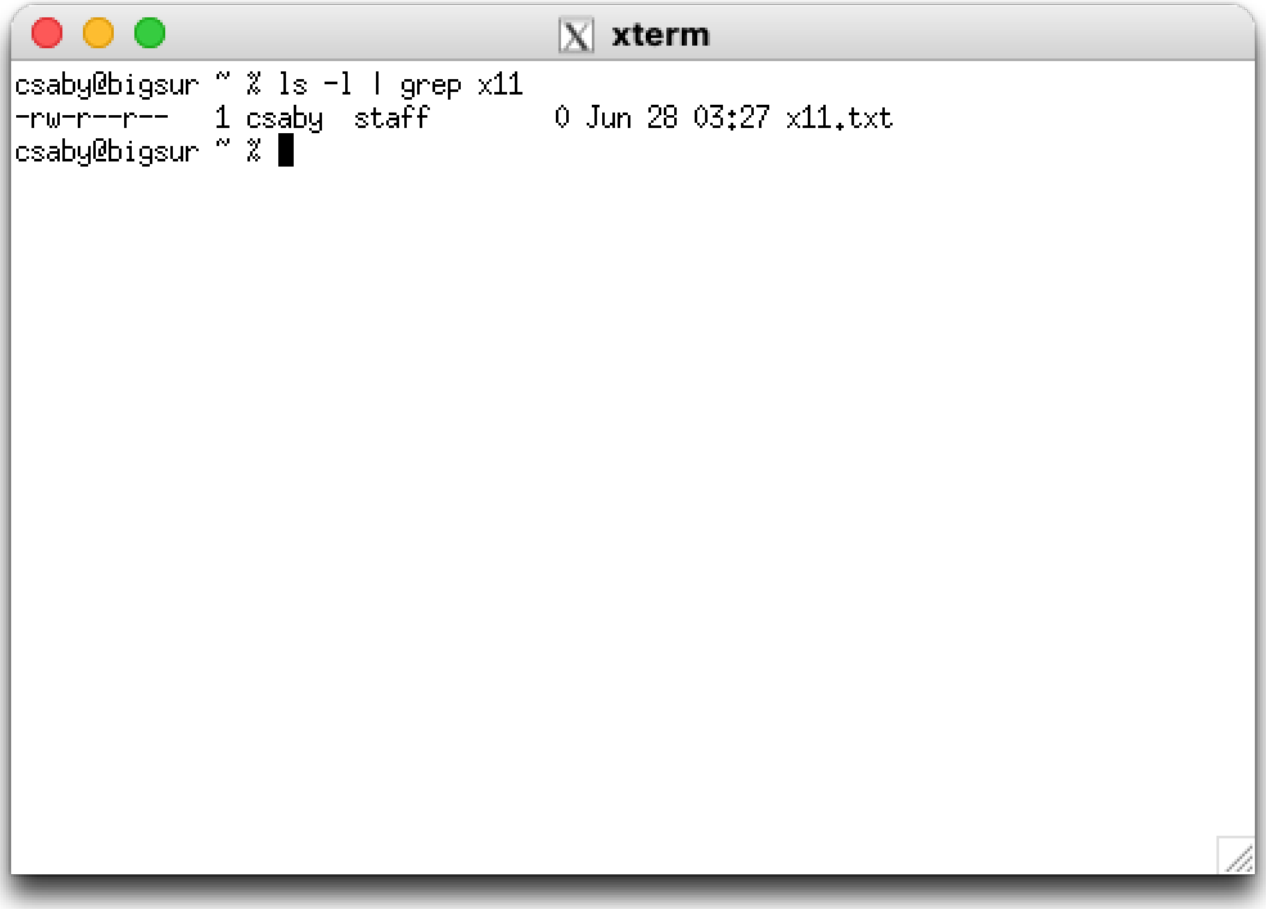
The `/opt/X11/etc/X11/xinit/xinitrc.d` directory contains other scripts.

If we create a script here, it will be executed upon starting xterm.

```
csaby@bigsur xinitrc.d % pwd
/opt/X11/etc/X11/xinit/xinitrc.d

csaby@bigsur xinitrc.d % cat 11-touch.sh
touch ~/x11.txt
```

Above we have a script which creates a file `x11.txt` in the user's HOME folder upon starting `xterm`. This is shown below.



The last directory to explore is `/opt/X11/etc/X11/xinit/privileged_startx.d`. It also contains scripts, and whatever we put here, will be executed by the `privileged_startx` process as root. This process is defined in `/Library/LaunchDaemons/org.xquartz.privileged_startx.plist` as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Label</key>
                <string>org.xquartz.privileged_startx</string>
        <key>ProgramArguments</key>
                <array>
                        <string>/opt/X11/libexec/privileged_startx</string>
                        <string>-d</string>
                        <string>/opt/X11/etc/X11/xinit/privileged_startx.d</string>
                </array>
        <key>MachServices</key>
                <dict>
                        <key>org.xquartz.privileged_startx</key>
                                <true/>
                </dict>
        <key>TimeOut</key>
                <integer>120</integer>
        <key>EnableTransactions</key>
                <true/>
</dict>
</plist>
```

If we check Apple's source code server.c we can find the related code that runs these files.

```
kern_return_t do_privileged_startx(mach_port_t test_port __attribute__((unused))) {
...
    /* Itterate over these files in alphabetical order */
    for(; ftsent; ftsent = ftsent->fts_link) {
        /* We only source regular files that are executable */
        /* Note: This assumes we own them, which should always be the case */
        if((ftsent->fts_statp->st_mode & S_IFREG) &&
           (ftsent->fts_statp->st_mode & S_IXUSR)) {

            /* Complete the full path filename in fn_buf */
            strcpy(s, ftsent->fts_name);

            /* Run it */
            error_code = system(fn_buf);
            if(error_code != 0) {
                asl_log(NULL, NULL, ASL_LEVEL_ERR,
                        "do_privileged_startx: %s: exited with status %d\n",
                        fn_buf, error_code);
                retval = KERN_FAILURE;
            }
        }
    }
```

This is a nice place to persist as root. Since Xquartz is not sandboxed (as shown below) it's even better.

```
csaby@bigsur ~ % codesign -dv --entitlements :- /Applications/Utilities/XQuartz.app
Executable=/Applications/Utilities/XQuartz.app/Contents/MacOS/X11
Identifier=org.xquartz.X11
Format=app bundle with Mach-O universal (x86_64 arm64)
CodeDirectory v=20500 size=507 flags=0x10000(runtime) hashes=9+3 location=embedded
Signature size=9072
Timestamp=2021. Apr 25. 15:38:28
Info.plist entries=20
TeamIdentifier=NA574AWV7E
Runtime Version=11.4.0
Sealed Resources version=2 rules=13 files=96
Internal requirements count=1 size=176
```