Beyond the good ol' LaunchAgents - 14 - atrun

theevilbit.github.io/beyond/beyond_0014

April 27, 2021

> This is part 14 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.

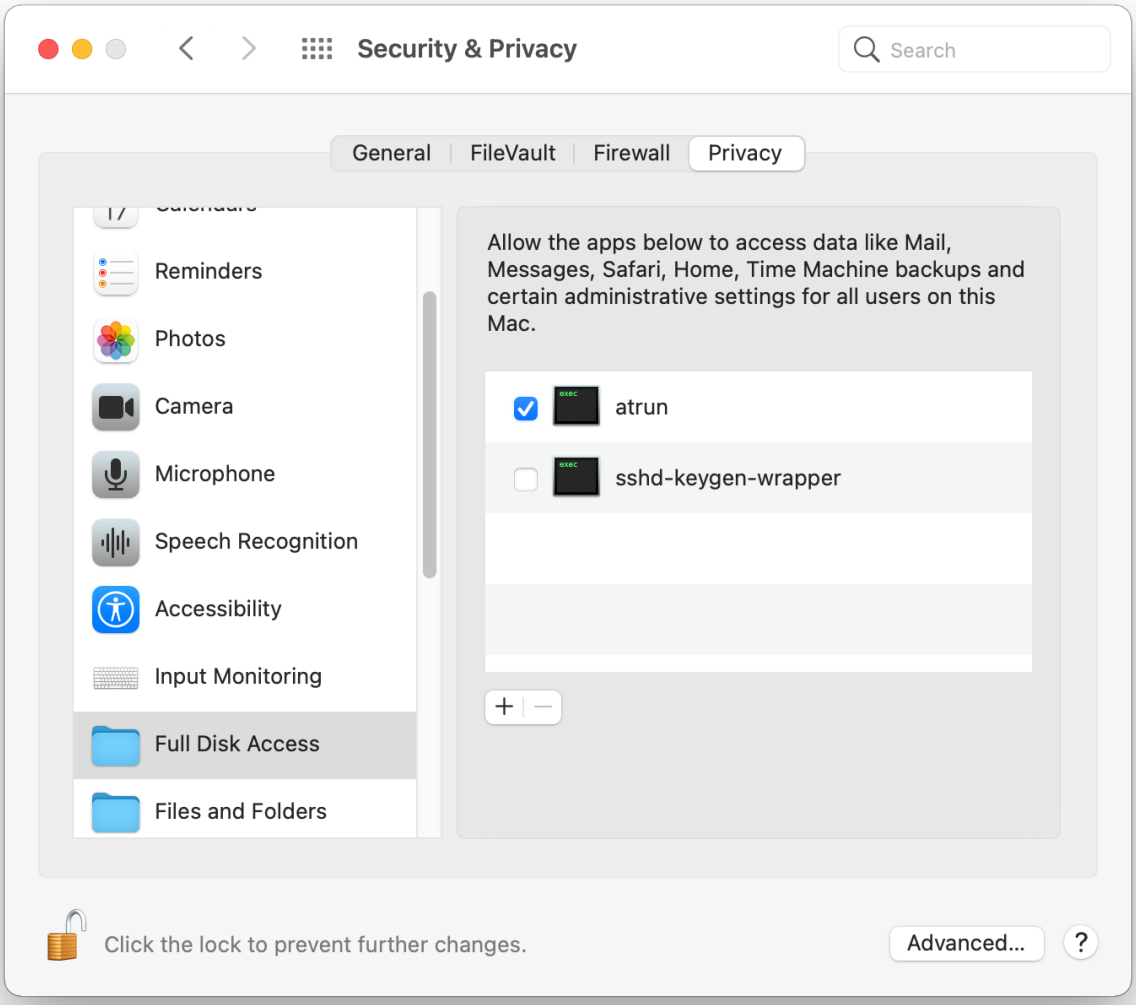
The `at` command set is a heritage *nix job scheduler on macOS. Although it's slowly being deprecated, it's still available on Big Sur, although disabled by default.

Enabling atrun

As described by `atrun`'s man page, the scheduler can be enabled using the following command:

```
sudo launchctl load -w /System/Library/LaunchDaemons/com.apple.atrun.plist
```

What is not mentioned, is that in order it to work, it needs Full Disk Access rights granted under privacy setting.



Because of all of these prerequisites (being root, and FDA access), this might not be a very attractive persistence method for attackers, however I think this is a commonly missed persistence, and could help staying under the radar.

If all of this was done, we can start using it.

Scheduling jobs

We can schedule jobs using the `at` command.

```
sh-3.2# echo "echo 11 > /tmp/at.txt" | at now+1
job 26 at Tue Apr 27 00:46:36 2021
```

The above will schedule a job in 1 minute, and will execute the command `echo 11 > /tmp/at.txt`. It got the job number 26.

We can check the job queue using `atq`.

```
sh-3.2# atq
26      Tue Apr 27 00:46:00 2021
22      Wed Apr 28 00:29:00 2021
```

Above we can see two jobs scheduled. We can print the details of the job using `at -c JOBNUMBER`

```
sh-3.2# at -c 26
#!/bin/sh
# atrun uid=0 gid=0
# mail csaby 0
umask 22
SHELL=/bin/sh; export SHELL
TERM=xterm-256color; export TERM
USER=root; export USER
SUDO_USER=csaby; export SUDO_USER
SUDO_UID=501; export SUDO_UID
SSH_AUTH_SOCK=/private/tmp/com.apple.launchd.co51iLHIjf/Listeners; export
SSH_AUTH_SOCK
__CF_USER_TEXT_ENCODING=0x0:0:0; export __CF_USER_TEXT_ENCODING
MAIL=/var/mail/root; export MAIL
PATH=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin; export PATH
PWD=/Users/csaby; export PWD
SHLVL=1; export SHLVL
SUDO_COMMAND=/usr/bin/su; export SUDO_COMMAND
HOME=/var/root; export HOME
LOGNAME=root; export LOGNAME
LC_CTYPE=UTF-8; export LC_CTYPE
SUDO_GID=20; export SUDO_GID
_=/usr/bin/at; export _
cd /Users/csaby || {
         echo 'Execution directory inaccessible' >&2
         exit 1
}
unset OLDPWD
echo 11 > /tmp/at.txt
```

At the bottom we will find the command that will be run, and at the top we will find under which user the command will run, in this case root (`atrun uid=0 gid=0`).

In order to schedule a job we need to have Full Disk Access, otherwise we will receive the following prompt.

Job files

The job files can be found at `/private/var/at/jobs/`

```
sh-3.2# ls -l /private/var/at/jobs/
total 32
-rw-r--r--  1 root  wheel    6 Apr 27 00:46 .SEQ
-rw-------  1 root  wheel    0 Apr 26 23:17 .lockfile
-r--------  1 root  wheel  803 Apr 27 00:46 a00019019bdcd2
-rwx------  1 root  wheel  803 Apr 27 00:46 a0001a019bdcd2
```

The filename contains the queue, the job number, and the time it's scheduled to run. For example let's take a loot at `a0001a019bdcd2`.

- `a` - this is the queue
- `0001a` - job number in hex, `0x1a = 26`
- `019bdcd2` - time in hex. It represents the minutes passed since epoch. `0x019bdcd2` is `26991826` in decimal. If we multiply it by 60 we get `1619509560`, which is `GMT: 2021. April 27., Tuesday 7:46:00`.

If we print the job file, we find that it contains the same information we got using `at -c`.