Beyond the good ol' LaunchAgents - 19 - Periodic Scripts

theevilbit.github.io/beyond/beyond_0019

August 6, 2021

> This is part 19 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.

This post has been long due, as it's one of my favorite persistence tricks. Up until Big Sur 11.5 you could also exploit it for privilege escalation if Homebrew was installed on the system.

Periodic scripts have a FreeBSD origin. These scripts are doing some maintenance tasks on the system, and scheduled to be run on a daily, weekly and monthly basis.

Scripts Execution

On macOS these tasks are scheduled by `launchd`, and as they are located inside `LaunchDaemons` they will run as root.

```
csaby@mac ~ % ls -l /System/Library/LaunchDaemons/*periodic-*
-rw-r--r--  1 root  wheel  887 Jan  1  2020
/System/Library/LaunchDaemons/com.apple.periodic-daily.plist
-rw-r--r--  1 root  wheel  895 Jan  1  2020
/System/Library/LaunchDaemons/com.apple.periodic-monthly.plist
-rw-r--r--  1 root  wheel  891 Jan  1  2020
/System/Library/LaunchDaemons/com.apple.periodic-weekly.plist
```

These PLIST files will run the program `/usr/libexec/periodic-wrapper`. This utility is open sourced by Apple as part of the `crontab` library. The latest (at the time of this writing) can be found here: crontabs-54

`periodic-wrapper` will eventually call `/usr/sbin/periodic` which is really just a bash script: periodic.sh

Script Location

The scripts can be found in `/etc/periodic/` and there is a directory for each schedule, `daily`, `weekly` and `monthly`. This location is not protected by SIP, thus if we put our own script here, it will be executed by the bash script as it will run each script defined here.

As the periodic script are scheduled to be run as root, our script will be run as root. There were some changes at Big Sur 11.5, but more about that later.

The Vulnerability & The Fix

Periodic scripts have a secondary location defined in the configuration file `/etc/defaults/periodic.conf`.

```
# periodic script dirs
local_periodic="/usr/local/etc/periodic"
```

Anyone who has Homebrew installed knows that the owner of the `/usr/local/` directory is changed from root to the admin user. This means that an admin user can write to this location without further privilege elevation. Thus if we create the above folder, and drop our script inside we gain code execution as root.

I reported this to Apple, and although it's not a default configuration they still fixed it. To my surprise they fixed it the way I recommended it. Here is a shortened version of the `diff` of the old and patched periodic bash script.

```
csaby@mac ~ % diff periodic_11.4 periodic_11.5
...
result=(`/usr/bin/stat -f '%Su %Ul' $file`)
user=${result[0]}
hardlinks=${result[1]}
if [ $hardlinks -ne 1 ] ; then
    skippedlist+=("$file")
    continue
fi
/usr/bin/su $user -c $file </dev/null >$tmp_output 2>&1
...
```

What happens here is that they check the owner of the script file, and will execute the script as the owner. This means that if you create a script with your user it will be executed as your user instead of root.

Moreover this also kills an exploit weaponization technique where you could move a file with an exploit. If you dropped a script here, you could turn a "file move" exploit into a full privilege escalation. This path is closed now. Sorry.

It was slightly different but I also utilized this weaponization trick in CVE-2021-1815.

But Not All Is Lost

There is a `999.local` script for each period. The daily can be read here: 999.local What it does is execution the script defined by the variable `$daily_local`. This is defined in `periodic.conf`.

```
csaby@mac /etc % grep .local /etc/defaults/periodic.conf
...
# 999.local
daily_local="/etc/daily.local"                          # Local scripts
# 999.local
weekly_local="/etc/weekly.local"                        # Local scripts
# 999.local
monthly_local="/etc/monthly.local"                      # Local scripts
                local i sourced_files
```

As we can find there is one for each schedule. As there are no further checks in the `999.local` scripts for user ownership, if we can move and create the file `/etc/daily.local`, that will be executed as root. Although we can't use it as privilege escalation, we can still use it for weaponization.

You are welcome :)

Further notes

The periodic script uses the previously mentioned configuration file to locate some of the scripts directories. This means that if we as an attacker overwrite it, we can place our scripts anywhere, so someone has to monitor for any file changes. Beyond that the config file defines other config files.

```
# What files override these defaults ?
periodic_conf_files="/etc/periodic.conf /etc/periodic.conf.local"
```