

Beyond the good ol' LaunchAgents - 4 - cron jobs

theevilbit.github.io/beyond/beyond_0004

March 18, 2021

This is part 4 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](#).

[cron](#) is probably one of the most well known persistence mechanisms for macOS and basically any *nix operating system. It was originally developed for Unix back in 1975, and made its way to most platforms, which has Unix origins, like Linux, FreeBSD and thus macOS.

We can use the `crontab` utility to manage our cron jobs. The `-l` option will list our currently scheduled jobs.

```
csaby@mac ~ % crontab -l
* * * * * /bin/bash -c "touch /tmp/cron2"
```

The output above show that we have a bash script to run every minute.

The option `-r` allows us to delete the current crontab file.

Finally we can use the `-e` option to edit the scheduled jobs. By default it will drop us inside a `vim` editor, and if we don't like that we can specify our editor of choice in the `EDITOR` environment variable.

```
csaby@mac ~ % EDITOR=nano crontab -e
```

The above command will drop us inside the `nano` editor.

When we edit crontab files, they will be created inside the `/tmp/` folder first, with a random name like `crontab.TSwrU7e0Xy`. Once we finished editing, they will be moved to their final location. On macOS this can be found at `/private/var/at/tabs/` and it's only readable and writeable for the root user.

```
csaby@mac ~ % sudo ls -l /private/var/at/tabs
total 8
-rw----- 1 root  wheel  267 Mar 18 06:43 csaby
```

We can also use a one liner to edit the cron tab.

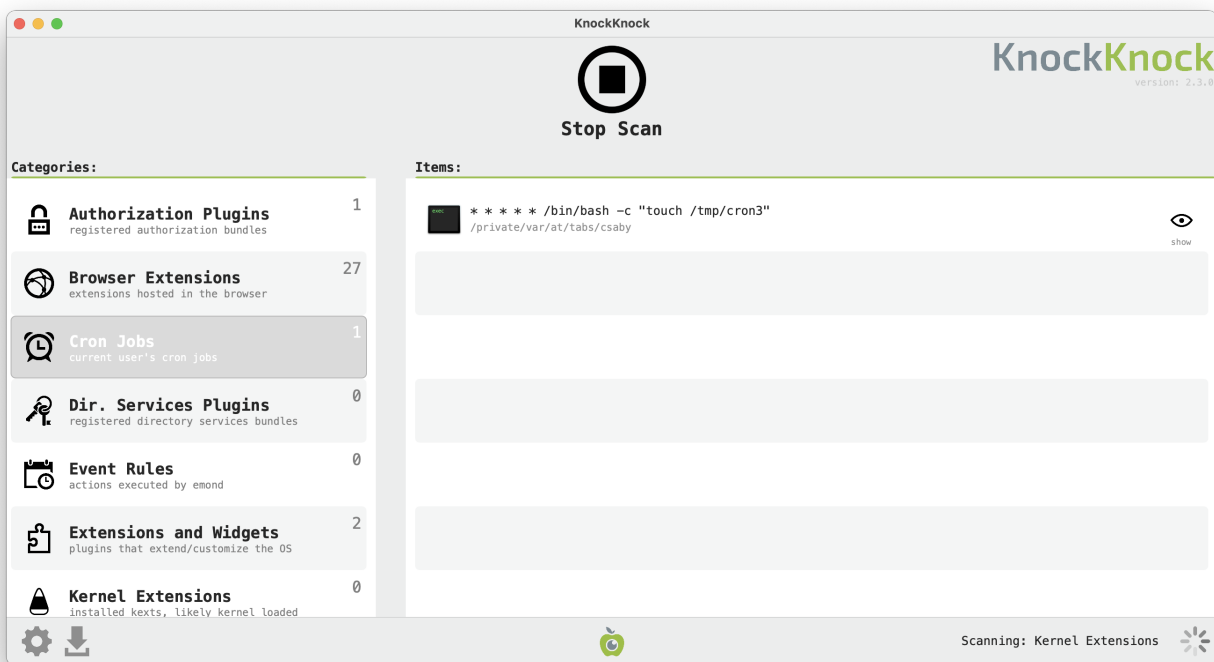
```
csaby@mac ~ % echo "* * * * * /bin/bash -c \"touch /tmp/cron3\"" | crontab -
```

```
csaby@mac ~ % crontab -l
* * * * * /bin/bash -c "touch /tmp/cron3"
```

Objective-See's [BlockBlock](#) will alert us upon cron job creation.



The other Objective-See tool, [KnockKnock](#) can also detect it.



cron's scheduling format can be hard to learn, we can tune every minute, hour, day, month and weekday. A few years ago I made an app that can help with that. It's called **Crontab Creator** and available [on the Mac App Store](#).

It allows us to create a cron job with the right scheduling syntax.

Crontab Creator

Create Examples

Minutes	Hours	Days	Months
<input checked="" type="radio"/> Every minute <input type="radio"/> Odd minutes <input type="radio"/> Even minutes <input type="radio"/> Every 5 minutes <input type="radio"/> Every 10 minutes <input type="radio"/> Every 15 minutes <input type="radio"/> Every 30 minutes <input type="radio"/> Custom (select from table)	<input checked="" type="radio"/> Every hour <input type="radio"/> Odd hours <input type="radio"/> Even hours <input type="radio"/> Every 2 hours <input type="radio"/> Every 3 hours <input type="radio"/> Every 6 hours <input type="radio"/> Every 12 hours <input type="radio"/> Custom (select from table)	<input checked="" type="radio"/> Every day <input type="radio"/> Odd days <input type="radio"/> Even days <input type="radio"/> Every 2 days <input type="radio"/> Every 5 days <input type="radio"/> Every 7 days <input type="radio"/> Every 15 days <input type="radio"/> Custom (select from table)	<input checked="" type="radio"/> Every month <input type="radio"/> Odd months <input type="radio"/> Even months <input type="radio"/> Every 2 months <input type="radio"/> Every 3 months <input type="radio"/> Every 4 months <input type="radio"/> Every 6 months <input type="radio"/> Custom (select from table)

Weekdays	Weekdays
<input checked="" type="radio"/> Every weekday <input type="radio"/> Monday-Friday <input type="radio"/> Weekend <input type="radio"/> Mon, Wed, Fri <input type="radio"/> Tue, Thu <input type="radio"/> Custom (select from table)	<input type="radio"/> Mon <input type="radio"/> Tue <input type="radio"/> Wed <input type="radio"/> Thu <input type="radio"/> Fri <input type="radio"/> Sat <input type="radio"/> Sun

<p>Application to run</p> <input style="width: 100%;" type="text"/> <p>Select File Clear selection</p> <p>Command arguments or custom command to run</p> <input style="width: 100%;" type="text" value="/bin/bash -c 'touch /tmp/cron3'"/> <p>Clear</p>	<p>Result</p> <pre>***** /bin/bash -c "touch /tmp/cron3"</pre> <p style="text-align: right;">Save to File</p> <p style="text-align: right;">Copy to clipboard</p>
--	---