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 <u>introduction</u>.

<u>cron</u> 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 **-1** option will list our currently scheduled jobs.

csaby@mac ~ % crontab -1
* * * * * /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 <u>-e</u> option to edit the scheduled jobs. By default it will drop us inside a <u>vim</u> editor, and if we don't like that we can specify our editor of choice in the <u>EDITOR</u> 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 <u>BlockBlock</u> will alert us upon cron job creation.

		BlockBlock Alert		
exec	ຼຼີ crontab created a cron job	,	virus total	ancestry
crontab (pid: 94	4081)			
process path:	/usr/bin/crontab			
process args:	-			
Cron Job				
startup file:	/private/var/at/tabs/cs	aby		
startup object:	* * * * * /bin/bash -c	"touch /tmp/cron3"		
		rule scope:		
		Process + File + Item ᅌ	Block	Allow
2021-03-18 05:49:4	8 +0000		temporarily	(pid: 9408

The other Objective-See tool, <u>KnockKnock</u> can also detect it.

•••		KnockKnock			
			KnockKnock		
		Stop Scan			
Categories:		Items:			
Authorization Plugins registered authorization bundles	1	<pre>** * * * /bin/bash -c "touch /tmp/cron3" /private/var/at/tabs/csaby</pre>	© shou		
Browser Extensions extensions hosted in the browser	27				
Cron Jobs current user's cron jobs	1				
Dir. Services Plugins registered directory services bundles	0				
Event Rules actions executed by emond	0				
Extensions and Widgets plugins that extend/customize the OS	2				
Kernel Extensions installed kexts, likely kernel loaded	0				
⇔ .≛		Č	Scanning: Kernel Extensions 🕂		

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 <u>on the Mac App Store</u>.

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

) 😑 🔘				Crontab Creato	or				
				Create Examp	les				
Minutes		Hours		Days		Months		Weekdays	
 Every minute Odd minutes Even minutes 	Minutes 0 1 2	Every hour Odd hours Even hours	Hours 0 1 2	 Every day Odd days Even days 	Days 1 2 3	Odd months	Months Jan Feb Mar	 Every weekday Monday-Friday Weekend 	Weekdays Mon Tue Wed
Every 5 minutes Every 10 minutes Every 15 minutes Every 30 minutes Custom (select from t	3 4 5 6 7 8 9 9 10 11 12 13	 Every 2 hours Every 3 hours Every 6 hours Every 12 hours Custom (select from table) 	3 4 5 6 7 8 9 10 11 12 13	Every 2 days Every 5 days Every 7 days Every 15 days Custom (select from table)	4 5 7 8 9 10 11 12 13 14	Every 2 months Every 3 months Every 4 months Every 6 months Custom (select from table)	Apr May Jun Jul Aug Sep Oct Nov Dec	Mon, Wed, Fri Tue, Thu Custom (select from table)	Thu Fri Sat Sun
Application to run Select File CI Command arguments or cu /bin/bash -c *touch /tm Clear		n		Result	in/bash -c "t		e to File e clipboard		