Beyond the good ol' LaunchAgents - 2 - iTerm2 startup

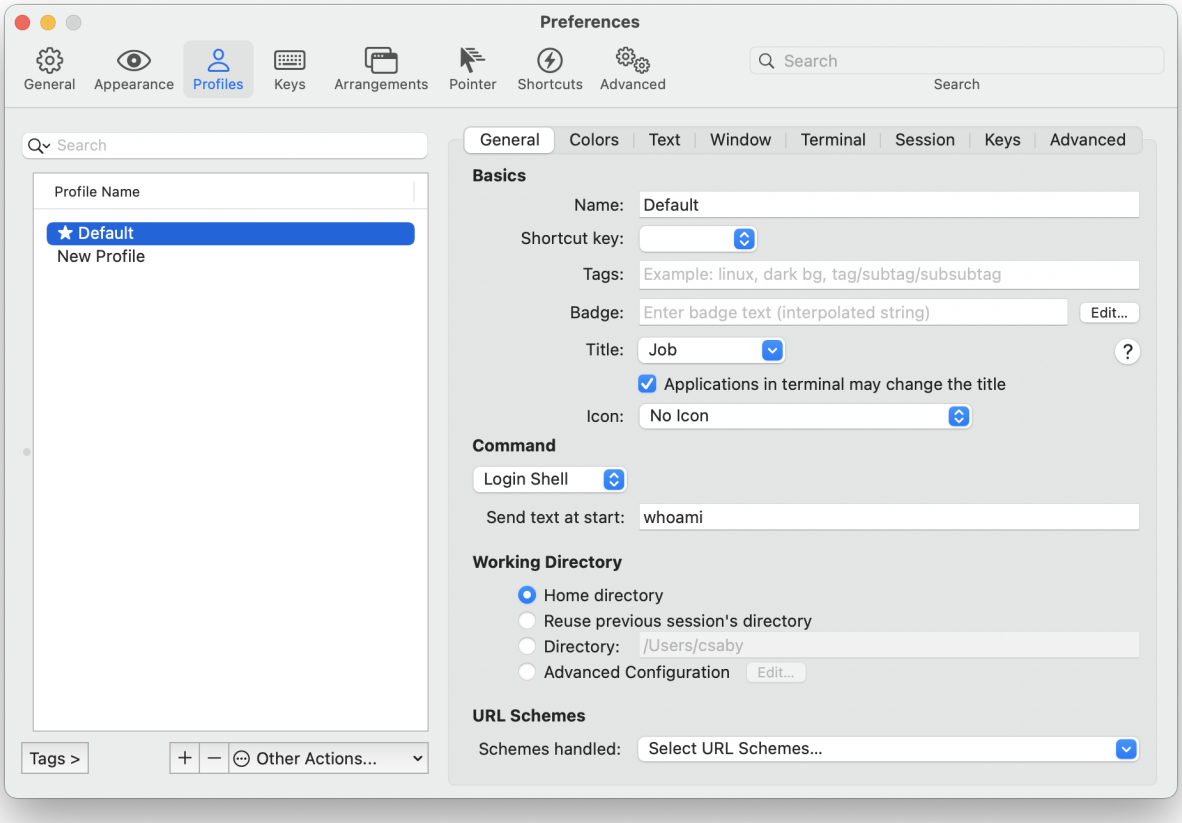
theevilbit.github.io/beyond/beyond_0002

March 16, 2021

> This is part 2 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.

If the first part was about `Terminal` and shell profiles, it's worth to mention iTerm2, which is a popular Terminal alternative on macOS. It's being used by many people, especially power users.

When we start `iTerm2` it starts the same shell environment as `Terminal`, and thus the same startup files apply here as well. However this application has an additional way to execute code.



In the Profile settings, there is an option, which is **Send text at start**. This is essentially a command we can run when someone starts up iTerm2. This is saved in iTerm2's preferences, which can be found at `~/Library/Preferences/com.googlecode.iterm2.plist`.

This is a binary PLIST, and we can use `plutil` to convert it to XML format.

```
csaby@mac ~ % plutil -convert xml1 ~/Library/Preferences/com.googlecode.iterm2.plist
-o - | grep -A 1 "Initial Text"
                        <key>Initial Text</key>
                        <string>whoami</string>
--
                        <key>Initial Text</key>
                        <string>uname</string>
```

The command is saved under the **Initail Text** key, and there can be multiple for different profiles. We can also use the `defaults` command to get this data.

```
csaby@mac ~ % defaults read com.googlecode.iterm2 | grep Initial
            "Initial Text" = whoami;
            "Initial Text" = uname;
```

Beyond persisting, `iTerm2` can have plenty of TCC exception if it receives user consent.

```
Executable=/Applications/iTerm.app/Contents/MacOS/iTerm2
Identifier=com.googlecode.iterm2
Format=app bundle with Mach-O universal (x86_64 arm64)
CodeDirectory v=20500 size=140993 flags=0x10000(runtime) hashes=4395+7
location=embedded
Signature size=8978
Timestamp=2021. Feb 8. 7:27:16
Info.plist entries=51
TeamIdentifier=H7V7XYVQ7D
Runtime Version=11.1.0
Sealed Resources version=2 rules=13 files=318
Internal requirements count=1 size=216
<?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>com.apple.security.automation.apple-events</key>
        <true/>
        <key>com.apple.security.cs.allow-jit</key>
        <true/>
        <key>com.apple.security.device.audio-input</key>
        <true/>
        <key>com.apple.security.device.camera</key>
        <true/>
        <key>com.apple.security.personal-information.addressbook</key>
        <true/>
        <key>com.apple.security.personal-information.calendars</key>
        <true/>
        <key>com.apple.security.personal-information.location</key>
        <true/>
        <key>com.apple.security.personal-information.photos-library</key>
        <true/>
</dict>
</plist>
```

As we can find in its entitlements, it can have access to camera, microphone, and several sensitive folders. Thus if access was granted by the user, we could access the same information with our persisted command.

Beyond this iTerm2 supports various auto launch scripts.

Scripts placed inside `~/Library/Application Support/iTerm2/Scripts/AutoLaunch` will be automatically launched upon starting iTerm. More information: Daemons — iTerm2 Python API 0.26 documentation

The file `~/Library/Application Support/iTerm2/Scripts/AutoLaunch.scpt` is also checked at startup, and it will be executed. This should be an Apple Script. More information: Scripting - Documentation - iTerm2 - macOS Terminal Replacement