

朝鲜APT针对macOS用户的恶意软件攻击活动分析 | CTF导航

 ctfiot.com/291611.html

疑似朝鲜黑客组织正利用一种似曾相识但依然危险的恶意软件攻击活动，瞄准 macOS 用户。该攻击活动最早于 2025 年 4 月被发现，是 Moonlock Lab 此前记录的“传染性面试”（Contagious Interview）攻击的巧妙演变。此次攻击的诱饵不再是虚假的招聘信息，而是诱骗用户更新 Realtek 驱动程序的钓鱼通知。

此次更新后的攻击手段本质上是一种信息窃取程序，这种恶意软件旨在窃取 Mac 登录凭证、Apple 密码数据和浏览器 cookie。这正是朝鲜黑客通常用来定位、访问和盗取企业和个人加密货币钱包的目标。

过去两年，Moonlock Lab 追踪到针对 Mac 电脑的窃取恶意软件数量增加了 3.4 倍。正如这起恶意软件回收利用事件所表明的那样，黑客无需创建全新的恶意软件，只需对旧恶意软件进行微调即可逃避检测。

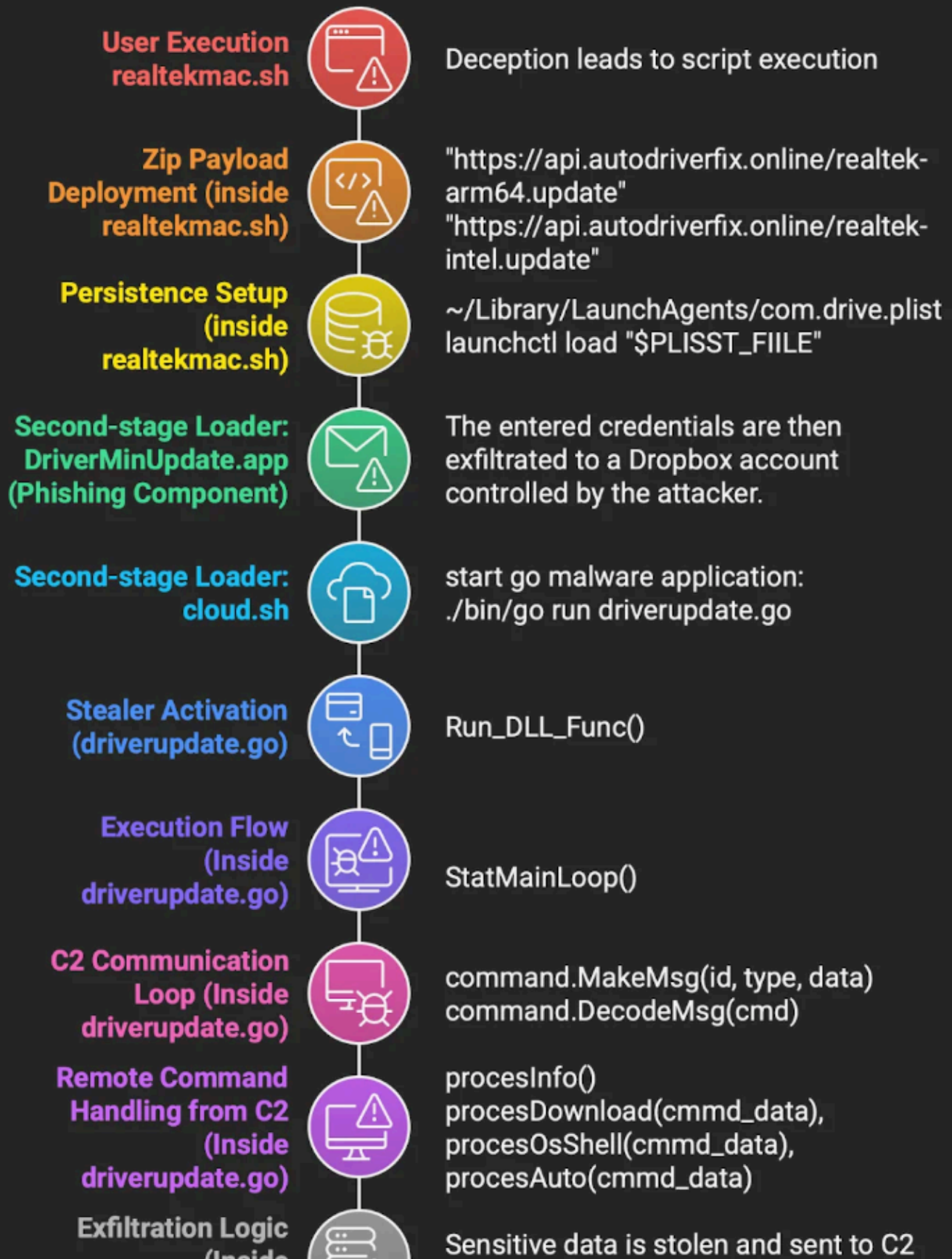
熟悉的敌人，换了张新面孔

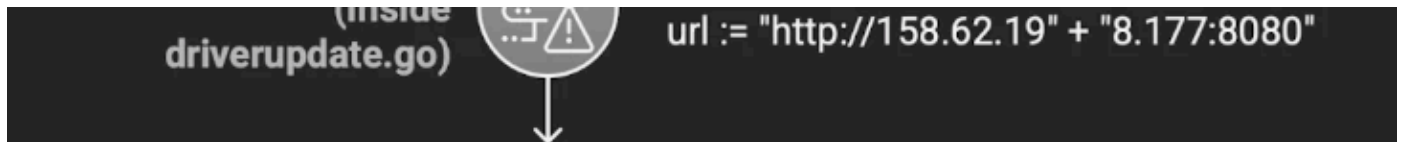
MalwareHunterTeam 在 X 上发布的一篇文章指出，当时 VirusTotal 上 64 家安全厂商中只有 2 家识别出一个可疑文件。经过深入调查，研究人员发现此次恶意软件活动与 Moonlock Lab 此前在 HackerNoon 上记录的行动如出一辙——只是现在增加了一些不易察觉的规避手段。

初始感染载体是一个名为 `realtekmac.sh` 的 Bash 脚本（SHA256：
`c192b93d30e482a20a958ceb329a53733debc4962d77cb273b019ca589dede6a`），攻击者通过社会工程学手段诱骗用户执行该脚本。脚本一旦运行，便会启动多阶段攻击，旨在实现持久化管理、窃取凭证和长期系统监控。

执行链：逐步分解

Malware Execution Flow: From Deception to Data Theft





初始感染

realtekmac.sh 脚本会检查 macOS 系统架构 (arm64 或 x86_64)，并从

<https://api.autodriverfix.online> 下载相应的 ZIP 压缩包。该压缩包名为 WebCam.zip (SHA256 : 6f6a9e8134e22c5b604b34256173848530136bec0d82ce78eaea61866fa2066a)，会被解压到 /var/tmp/WebCam/ 目录下。其中包含一个第二阶段脚本 (cloud.sh) 和一个钓鱼应用程序 (DriverMinUpdate.app)。

通过 LaunchAgents 实现持久化

为了确保长期访问，该脚本会在 ~/Library/LaunchAgents/ 目录下创建一个 launchd plist 文件 (com.drive.plist)。这确保 cloud.sh 脚本在系统启动时自动执行，并使用 launchctl 加载配置。

利用虚假应用程序进行网络钓鱼

DriverMinUpdate.app (SHA256 :

34b6fc9024591e7a107ffbd7b77e788ec2dd83943ab4ba4889b9996ca08260b8) 是一款欺骗性的 macOS 应用程序，它会通过图形用户界面 (GUI) 提示用户输入系统密码。如果用户照做，窃取的凭据将被泄露到攻击者控制的 Dropbox 帐户中，这种手法与之前分析过的一个样本类似。

第二阶段加载器

cloud.sh 脚本 (SHA256 :

b98e6ea26eba32001a3e414b8648db9c74690e3bc2e8d649beb3336216c9949f) 使用捆绑的 Go 运行时执行主要有效载荷，即名为 driverupdate.go (SHA256 : f748e7f01f2daf6cbdb72fddd7a8e1415bceea0c42004d034f60faa5331c44e4) 的基于 Go 的窃取程序。

窃取程序激活和 C2 通信

Go 组件建立了一个持久的命令与控制 (C2) 循环，其硬编码端点为 <http://158.62.198.177:8080>，托管于美国的 Shock Hosting LLC。该组件收集系统信息 (用户名、主机名、操作系统、架构)，窃取 Chrome Cookie、登录数据和 Keychain 凭据，并支持远程命令执行。所有通信均使用 RC4 加密，通过 MD5 校验和进行身份验证，并使用 Base64 编码以逃避检测。

威胁的演变

研究人员在 HackerNoon 文章中分析的该恶意软件早期变种相比，其核心功能未发生变化。不过，攻击者已实施部分细微调整，以此规避安全检测。

具体来看，文件名与攻击基础设施的调整如下：文件名已完成更新，例如以 realtekmac.sh 取代原有的 ffmpeg.sh，以 WebCam.zip 替换此前的 VCam_intel.zip；攻击所使用的下载 URL 已变更为 api.autodriverfix.online，控制服务器 (C2 服务器) 的 IP 地址则更新为 158.62.198.177。

```

1 #!/bin/bash
2
3 # Define variables for URLs
4 ZIP_URL_ARM64="https://api.nvidia-
cloud.online/VCam1.update"
5 ZIP_URL_INTEL="https://api.nvidia-
cloud.online/VCam2.update"
6 ZIP_FILE="/var/tmp/VCam.zip"
# Path to save the downloaded ZIP file
7 WORK_DIR="/var/tmp/VCam"
# Temporary directory for extracted files
8 EXECUTABLE="vcamservice.sh"
# Replace with the name of the executable file
inside the ZIP
9 APP="ChromeUpdateAlert.app"
# Replace with the name of the app to open
10 PLIST_FILE=~/.Library/LaunchAgents/com.vcam.plist
# Path to the plist file
11
12 # Determine CPU architecture
13 case $(uname -m) in
14     arm64) ZIP_URL=$ZIP_URL_ARM64 ;;
15     x86_64) ZIP_URL=$ZIP_URL_INTEL ;;
16     *) exit 1 ;; # Exit for unsupported
architectures
17 esac

```

```

1 #!/bin/bash
2
3 ZIP_AARRMM64_URL="https://api.autodriverfix.online/r
ealtek-arm64.update"
4 ZIP_INNTEL_URL="https://api.autodriverfix.online/rea
ltek-intel.update"
5 ZIPP_FILE="/var/tmp/WebCam.zip"
6 WOORRK_DIR="/var/tmp/WebCam"
7 EXE_CTUT_ABLE="cloud.sh"
8 SAAP="DriverMinUpdate.app"
9 PLISST_FIILE=~/.Library/LaunchAgents/com.drive.plist
10
11
12 case $(uname -m) in
13     arm64) ZIP_URL=$ZIP_AARRMM64_URL ;;
14     x86_64) ZIP_URL=$ZIP_INNTEL_URL ;;
15     *) exit 1 ;;
16 esac

```

网络钓鱼应用程序存储DriverMinUpdate.app 现在将窃取的凭据存储在 /pwd.txt 而不是 /password.txt 中。

功能上没有变化执行流程、持久化机制和窃取功能（针对 Chrome 和 Keychain 数据）保持不变。

```

23 func StartMainLoop(id string, url string) {
24
25     var (
26         msg_type string
27         msg_data [][]byte
28         msg      string
29         cmd      string
30         cmd_type string
31         cmd_data [][]byte
32         alive    bool
33     )
34
35     // initialize
36     cmd_type = config.COMMAND_INFO
37     alive = true
38     for alive {

```

```

23 func StatMainLoop(id string, url string) {
24
25     var (
26         mssg_type string
27         mssg_data [][]byte
28         msg      string
29         cmd      string
30         cmmd_type string
31         cmmd_data [][]byte
32         isLive    bool
33     )
34
35     // initialize
36     cmmd_type = config.COMMAND_INFO
37     isLive = true
38     for isLive {

```

技术深度解析：深入剖析基于围棋的窃取器

driverupdate.go payload 是一个基于 Go 语言的窃取程序，它负责协调命令与控制通信、系统性能分析和凭证窃取。让我们逐个模块地剖析它的内部运作机制，并追踪其执行流程，以揭示它是如何保持隐蔽性和控制力的。

核心功能

窃取者的主要任务是建立持久的C2通道，分析受感染系统，并窃取敏感数据。它通过系统侦察、凭证窃取和远程命令执行等手段来实现这一目标。

代码库被组织成多个不同的模块，每个模块处理攻击的特定方面：

- `instance/`管理进程锁定和唯一标识符生成，以确保单实例执行和持久跟踪。
- `core/`实现主要的 C2 循环并处理命令执行，作为窃取者的运行骨干。
- `auto/`执行自动凭据窃取，目标是Chrome浏览器数据和macOS钥匙串凭据。
- `传输/`促进使用 RC4 和 MD5 进行混淆和认证的加密网络通信。
- `command/`定义了 C2 交互的消息格式化和解析逻辑。
- `util/`提供实用函数，例如用于数据泄露的文件压缩。

这种模块化设计使得恶意软件能够高效运行，同时保持适应未来更新的能力。

执行流程

窃取程序的执行始于 `driverupdate.go` 文件，其中 `main()` 函数调用 `Run_DLL_Func()` 来初始化恶意逻辑。流程如下。

实例初始化

- `instance.genrateId()`生成或检索存储在 `/tmp/.host` 中的唯一机器 ID。此 ID 确保 C2 服务器能够跨会话跟踪受感染的设备，从而为受害者的机器创建持久的指纹。
- `instance.CheckDup_Instance()`函数会检查 `/tmp/.store` 目录是否存在，以防止恶意软件的多个实例同时运行。这可以降低被安全工具检测到的风险。
- ``instance.Delay()``方法会引入 10 秒的延迟，以绕过沙箱环境。沙箱环境通常会终止那些没有立即活动的进程。这种反分析技巧可以帮助恶意软件绕过自动化防御系统。
- `instance.RegisterIn_stance()`将进程 ID 存储在临时文件中，从而加强单实例锁定。

```
1 package main
2
```

C2通信

`core.StatMainLoop(id, url)` 函数定义在 `core/loop.go` 文件中，它会与位于 `http://158.62.198.177:8080` 的 C2 服务器建立持久连接，该服务器由位于美国的 Shock Hosting LLC 托管。该通信协议旨在实现隐蔽性：

- 出站消息：`command.MakeMsg(id, type, data)` 函数会构造包含唯一 ID、消息类型（例如 MSG_PING、MSG_FILE）和数据负载的消息。这些消息会进行 Base64 编码，使用 RC4 加密，并通过 HTTP POST 发送。
- 初始数据：`procesInfo()` 函数收集系统详细信息（用户名、主机名、操作系统版本和架构），并将其作为第一条出站消息发送，使攻击者能够了解受害者的环境快照。

```
22 func StatMsgLoop(id string, url string) {
```

远程命令处理

C2 服务器以 [command_type, arg1, arg2, ...] 的格式，以 RC4 加密、MD5 认证和 base64 编码的命令进行响应。

该窃取程序支持多种命令，每条命令都与特定的恶意操作相关联：

- COMMAND_INFO请求更新的系统信息，重新运行 processInfo() 以收集用户、主机名、操作系统和架构数据。
- COMMAND_UPLOAD指示恶意软件从 C2 服务器接收一个压缩文件，并使用 processUpload(data) 函数将其写入磁盘。这使得部署其他有效载荷成为可能。
- COMMAND_DOWNLOAD通过 processDownload(path) 提取本地文件或目录，该函数读取目标，使用 util.Compress() 对其进行压缩，并将其作为 MSG_FILE 消息发送。
- COMMAND_OSSHELL通过 procesOsShell() 使用 exec.Command() 执行 shell 命令。命令可以静默运行（分离式），也可以将输出作为 MSG_LOG 消息返回，从而允许攻击者远程操纵系统。
- COMMAND_AUTO通过 processAuto(mode) 触发自动凭据窃取，调用 auto.AutoModeC_chromeCookie() 提取 Chrome cookie，调用 auto.AutoModeMa_cChromeLoginData() 窃取登录凭据和钥匙串数据。
- COMMAND_WAIT使用 processWait(hex_duration) 将恶意软件置于休眠状态指定时长。因此，窃取程序在发送周期性心跳信号 (MSG_PING) 的同时，保持低调运行。
- COMMAND_EXIT终止 C2 循环并通过 processExit() 停止执行，允许攻击者在需要时进行清理。

数据泄露逻辑

- 文件泄露：procesDownload(path) 函数读取文件或目录，使用 util.Compress() 将它们压缩成 tar+gzip 归档，并将结果作为 MSG_FILE 消息发送。
- Shell 命令输出：procesOsShell() 的结果被捕获并作为 MSG_LOG 消息发送，为攻击者提供已执行命令的实时反馈。
- 凭证窃取：该自动模块的目标是 Chrome 和钥匙串数据，包括：
 - auto.AutoModeC_chromeCookie() 函数会解密并提取 Chrome cookie，从而实现会话劫持。
 - auto.AutoModeMa_cChromeLoginData() 函数会检索登录凭据并访问存储在钥匙串中的密码，从而暴露敏感的帐户信息。

- 诸如 `auto.AutoModeCh_romeGather()` 和 `auto.AutoModeCh_romeChangeProfile()` 之类的附加函数会收集扩展程序数据并修改 Chrome 首选项，这可能会注入恶意设置。

一个以盈利为目的的持续威胁

这场持续到2025年4月的攻击活动表明，网络威胁行为者采取的是长期策略。他们的目标是获取经济利益，为此，他们专注于长期访问和数据窃取。通过不断升级其基础设施，同时保持恶意软件的核心功能，他们很可能会继续攻击毫无防备的macOS用户。

根据我们之前对“传染性访谈”活动的分析，我们知道这些参与者适应能力很强。他们依靠社会工程学，并使用不同的借口来广撒网，最大限度地提高有人上钩的几率。

对于 macOS 用户来说，要点很简单：保持谨慎，除非您完全了解其作用，否则不要执行任何命令或安装任何文件。此外，请遵循以下最佳实践。

- 避免点击陌生人发来的信息中的链接。承诺巨额遗产或梦寐以求的工作机会几乎都是陷阱。
- 切勿通过弹出窗口安装驱动程序或更新。如果某个程序提示需要更新，请直接打开该程序并查看其设置。正常的更新通知通常会显示在那里。
- 使用密码管理器，而不是依赖 Apple Passwords 或将登录信息保存在浏览器中。窃取信息的恶意软件可以读取这些信息，并帮助黑客入侵您的帐户。
- 定期清除浏览器 Cookie。虽然 Cookie 本身无害，但有些 Cookie 会存储敏感数据，帮助攻击者入侵您的帐户。特别是第三方 Cookie，其中包含追踪器，会收集您的信息，并可能被用于制作极具迷惑性的钓鱼邮件。
- 在您的 Mac 上安装反恶意软件。虽然苹果的安全更新很可靠，但通常落后于第三方厂商。这意味着当出现新的恶意软件时，您的 Mac 可能无法得到全面保护。额外的安全防护可以提高您及早发现入侵者的几率，避免他们对您的电脑、账户和加密货币钱包造成损害。

对于安全团队而言，此次攻击活动提醒我们，攻击者只需稍作改变就能绕过我们的安全防护。应对此类威胁需要我们快速更新检测系统，以便捕获混淆的多阶段恶意软件。为了支持这项工作，我们将在下方分享此恶意软件的入侵指标 (IoC)。

MITRE ATT&CK 地图

IoC

文件哈希值 (SHA256)

```
realtekmac.sh: c192b93d30e482a20a958ceb329a53733debc4962d77cb273b019ca589dede6aWebCam.zip  
(Intel): 6f6a9e8134e22c5b604b34256173848530136bec0d82ce78eaea61866fa2066aWebCam.zip (ARM):  
d08b8734aa15819f51052e3ebfb95733fcb09bddef5de29d5c6cce43866ea462DriverMinUpdate.app:  
34b6fc9024591e7a107ffbfdbf77e788ec2dd83943ab4ba4889b9996ca08260b8cloud.sh:  
b98e6ea26eba32001a3e414b8648db9c74690e3bc2e8d649beb3336216c9949fdriverupdate.go:  
f748e7f01f2daf6cbdb72fddd7a8e1415bceea0c42004d034f60faa5331c44e4
```

网络

https://api.autodriverfix.online/realtek-arm64.update
https://api.autodriverfix.online/realtek-intel.update
C2服务器：
http://158.62.198.177:8080

文件路径

提取的有效载荷目录：/var/tmp/WebCam/持久化 plist 文件：~/Library/LaunchAgents/com.drive.plist
网络钓鱼凭证存储：/pwd.txt 机器 ID 存储：/tmp/.host
实例锁定文件：/tmp/.store
相关样本（旧版本）
ffmpeg.sh：3697852e593cec371245f6a7aaa388176e514b3e63813fdb136a0301969291ea
VCam_intel.zip：60ec2dbe8cfacdff1d4eb093032b0307e52cc68feb1f67487d9f401017c3edd7

| 本篇文章来源于微信公众号: 威胁情报Z分析