Newly discovered PamStealer isn’t your typical macOS malware

Researchers have found a never-before-seen piece of macOS malware that combines a series of clever tradecraft to infect Macs with stealthy, custom-developed credential-stealing code.

The malware is delivered in two stages. The first is distributed in a disk image that masquerades as Maccy, a clipboard manager for Macs. It’s compiled as AppleScript that is notable for the way it delivers the second stage. The malware is named PamStealer because the Rust-written infostealer uses the Pluggable Authentication Modules interface built into macOS to validate the target’s login password before sending it to an attacker-controlled server.

A quieter execution chain

The use of both disk image and AppleScript is common in malware for Macs. More unusual is the way PamStealer combines them to gain stealth. When the AppleScript is double-clicked, it’s opened in the macOS Script Editor, where the malicious functionality is buried deep within the file.

“Rather than relying on shell commands such as curl or zsh, the AppleScript executes a self-contained JavaScript for Automation (JXA) downloader that retrieves and stages the payload using native Objective-C APIs,” researchers from Jamf, a security firm for macOS users, wrote. “Combined with a Rust-based second stage and a password capture workflow that validates credentials locally through PAM, the result is a

quieter execution chain than we typically observe in commodity macOS stealers.”

When a user, expecting to install a trustworthy clipboard manager, encounters the disk image, they’re prompted to press Command-R immediately after double-clicking it. This command executes malicious code inside the AppleScript directly. It also allows the execution to bypass com.apple.quarantine, a macOS attribute that provides warnings and restrictions when executable files have been downloaded from the Internet.

As Jamf explained:

> *PamStealer combines a recently emerging delivery surface with a less familiar payload. While the clickable .scpt and Script Editor lure build on tradecraft that is already gaining adoption across the macOS threat landscape, the malware distinguishes itself through a self-contained JXA dropper, a Rust-based second stage, and a password capture workflow that validates credentials locally through PAM before harvesting them. That second stage puts considerable effort into staying hidden, masquerading as Finder, encrypting its command-and-control traffic, and holding back prompts like the Full Disk Access request for as long as forty minutes so its activity does not line up with launch. Together, these behaviors illustrate how commodity macOS stealers continue to evolve, adopting quieter execution chains and native implementations that reduce traditional detection opportunities while remaining compatible with standard macOS features.*

The first stage puts its payload inside an app bundle that impersonates real components built into macOS. The component changes from sample to sample of the malware. Finder.app under com.apple.finder.core or com.apple.finder.monitor, and a Software Update.app under com.apple.security.daemon, are two examples. In either case, they run hidden. They also display macOS’s genuine Finder.icns as its icon.

https://arstechnica.com/security/2026/07/new-pamstealer-macos-malware-uses-clever-tradecraft-to-remain-stealthy/