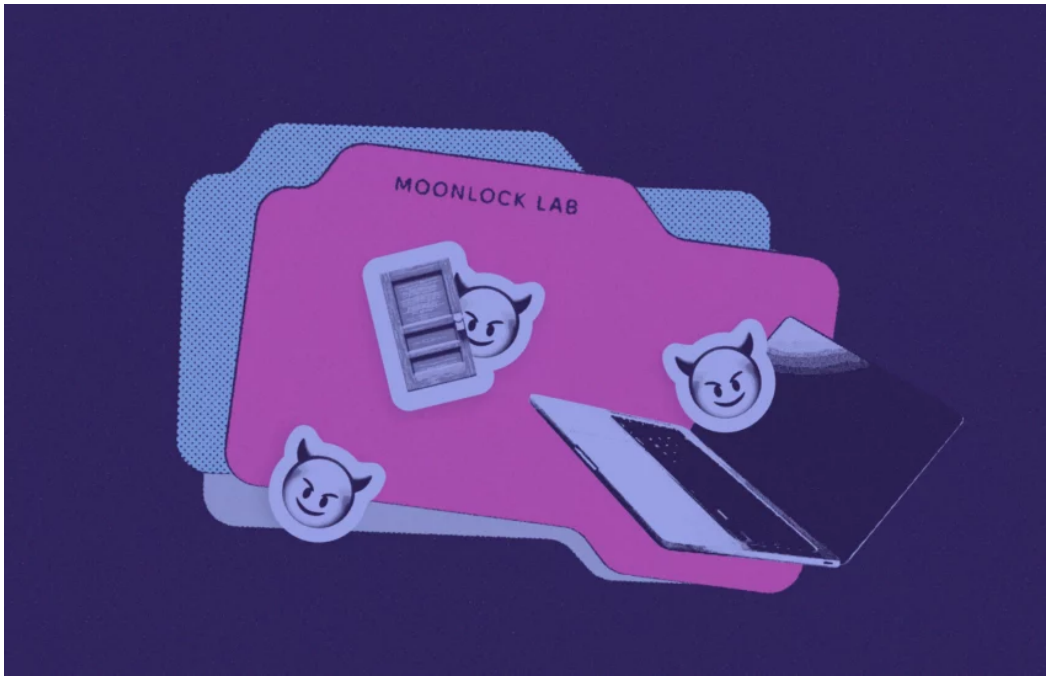
Mac.c stealer evolves into MacSync: Now with a backdoor

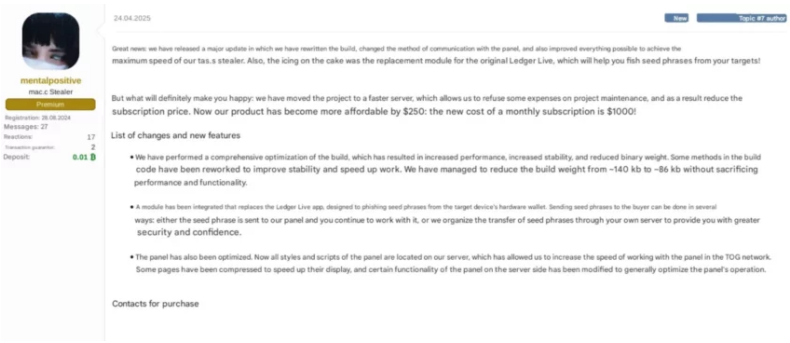
⋮ 9/11/2025



In April 2025, a new macOS stealer developer emerged under the alias “mentalpositive.” Their stealer, mac.c, wasn’t sophisticated. It wasn’t particularly stealthy or feature-rich at launch, either. However, it did have one important thing going for it. It was cheap.

The low price point gave mac.c a unique edge among traffer teams — groups that drive victims to malicious sites via phishing or malvertising — that were looking for low-cost, easily deployable macOS infostealers. We covered the initial rise of this malware in our article, Mac.c Stealer Takes on AMOS, at the time.

Few could have predicted what would happen next.


mentalpositive
mac.c Stealer
Developer
Registration: 28.08.2024
Messages: 27
Reactions: 17
2
Deposit: 0.01 ₿

24.04.2025

New Topic #7 author

Great news: we have released a major update in which we have rewritten the build, changed the method of communication with the panel, and also improved everything possible to achieve the maximum speed of our tas.s stealer. Also, the icing on the cake was the replacement module for the original Ledger Live, which will help you fish seed phrases from your targets!

But what will definitely make you happy: we have moved the project to a faster server, which allows us to refuse some expenses on project maintenance, and as a result reduce the subscription price. Now our product has become more affordable by $250: the new cost of a monthly subscription is $1000!

List of changes and new features

- We have performed a comprehensive optimization of the build, which has resulted in increased performance, increased stability, and reduced binary weight. Some methods in the build code have been reworked to improve stability and speed up work. We have managed to reduce the build weight from ~140 kb to ~86 kb without sacrificing performance and functionality.
- A module has been integrated that replaces the Ledger Live app, designed to phishing seed phrases from the target device's hardware wallet. Sending seed phrases to the buyer can be done in several ways: either the seed phrase is sent to our panel and you continue to work with it, or we organize the transfer of seed phrases through your own server to provide you with greater security and confidence.
- The panel has also been optimized. Now all styles and scripts of the panel are located on our server, which has allowed us to increase the speed of working with the panel in the TOG network. Some pages have been compressed to speed up their display, and certain functionality of the panel on the server side has been modified to generally optimize the panel's operation.

Contacts for purchase

One month later, the stealer has undergone a surprising transformation. Rebranded as MacSync, the tool now includes a fully-featured Go-based agent acting as a backdoor, expanding its functionality far beyond basic data exfiltration.

This makes MacSync one of the first known cases of a macOS stealer with modular, remote command and control capabilities.

The rebrand: Same stealer, new ambitions

No one anticipated how swiftly mac.c would expand beyond pure info-stealing. In a recent interview with security researcher @g0njxa, published just a week before our sample analysis, the MacSync team revealed the following key insights:

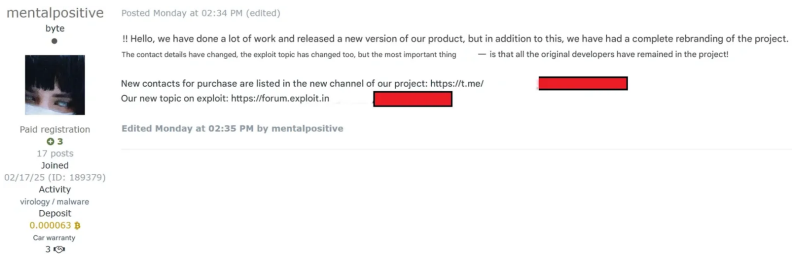
- **Userbase**: The project may be young, but its userbase has already grown to nearly 3 dozen.
- **Rebranding rationale**: The project was at risk of dying, so it was sold to ensure further development.

> *MacSync is a just a rebrand project where a new administration took management of what we knew previosuly as "Mac.c Stealer by mentalpositive". It still has the same MaaS business model and same product. If this is true, why this rebrand happened now?*

> The old project risked dying from lack of time and funding, so it was purchased and will be developed further without dwelling on past difficulties.

The rebrand of mac.c into MacSync is discussed in an interview with cybersecurity researcher @g0njxa.

- **Team continuity**: Despite the change in management, the development team remained intact.



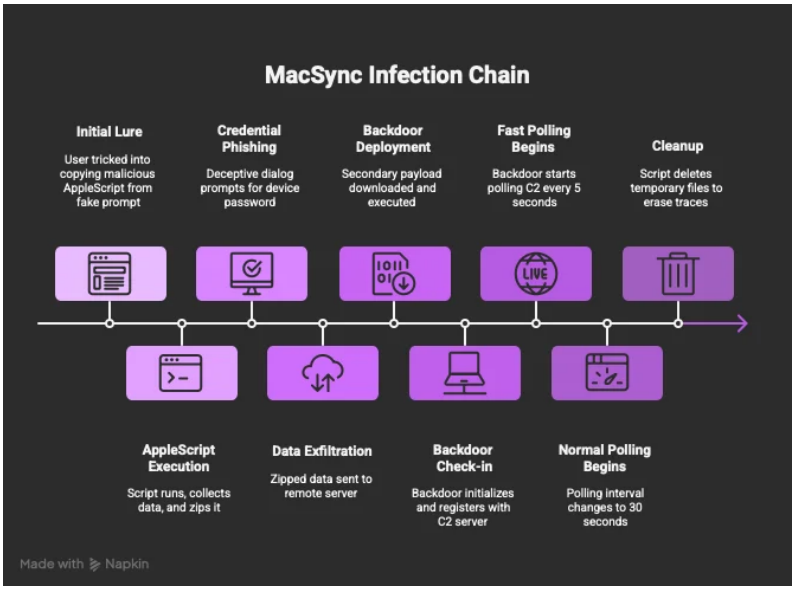
Bad actor mentalpositive posted an announcement about the rebranding of mac.c to MacSync.

- **Future threats**: There will be an emphasis on phishing, combined with app-specific vulnerabilities. This acknowledges AI-enhanced defenses, but one can bet on the human factor in "8 out of 10 cases."
- **Feature expansion**: When asked about competitors blending stealers with backdoor capabilities, the response was telling: "Expansion of functionality is positive; we're not an exception. Work is underway, and a new release will be published soon."

This teased "expansion" materialized more quickly than expected. On the same day the interview dropped, we encountered an interesting sample (SHA256: a42eece43aad2e2a2f98d41b1b48025c252452318a864d32d0f9156adaabe65b) tied to MacSync.

According to a Reddit post, it spread through a known "ClickFix" campaign: a fake Cloudflare Turnstile prompt urging users to copy a command, which instead pasted a Base64-obfuscated AppleScript. This script was executed in the background, stealing data and dropping the new backdoor component.

Consequently, the infection chain for MacSync follows a structured progression, leveraging social engineering, multi-stage execution, and data exfiltration.

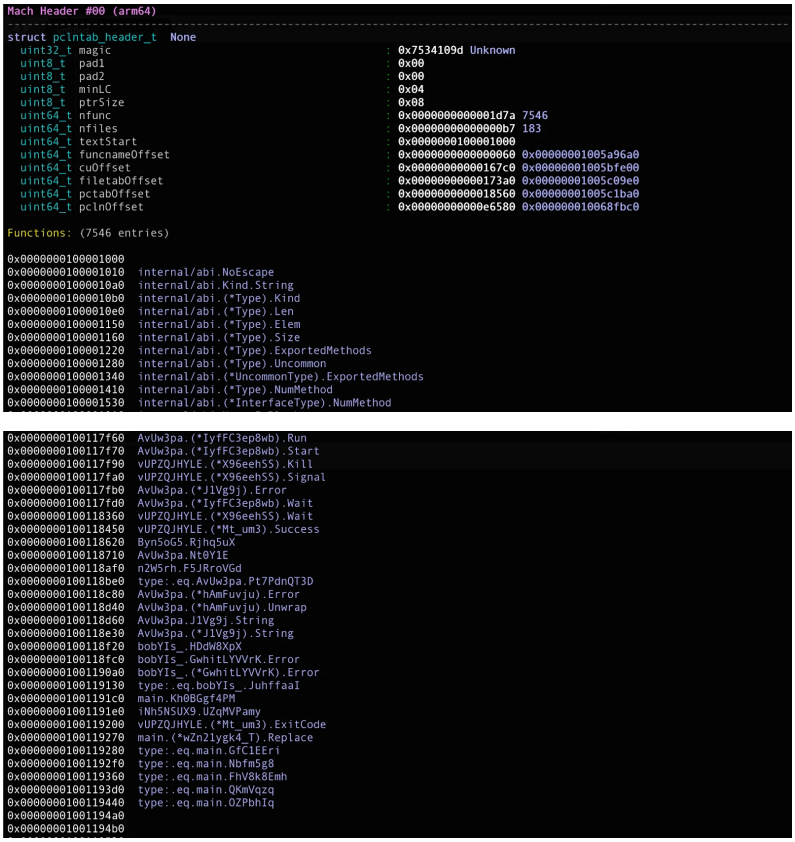


Technical breakdown: The new Go-based backdoor

The core of this stealer remains an AppleScript payload, unchanged from earlier versions. It collects sensitive data (e.g., credentials, wallets), zips it as /tmp/salmonela.zip (a fun nod to the bacteria Salmonella), and exfils via a POST to https://meshsorterio[.]com/api/data/receive with a custom X-Bid header (e.g., f48fbe39836779cadbf148b5952919fd).

A health check hits /api/health. The script then fetches the backdoor from https://gamma[.]meshsorterio[.]com/trovo/index.php, unzipping and executing it as ./shell after a 30-second delay. Finally, it cleans up the temporary files.

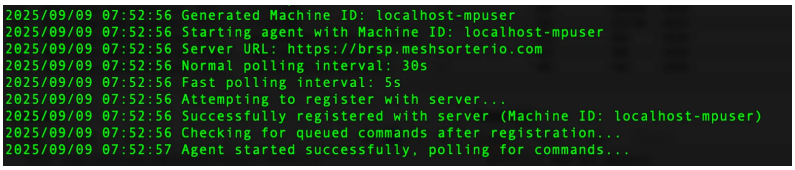
Here's where the new version of the stealer diverges from the original. The backdoor is now an obfuscated Go Mach-O binary, exhibiting agent behavior. This obfuscation (Go garbling) complicates static reversing.



In spite of the obfuscation, dynamic analysis and network captures were sufficient to map its communication flow and capabilities. So, let's break it down, step by step.

1. Background execution and agent startup

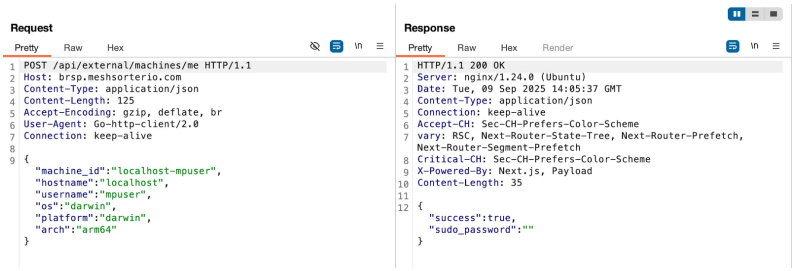
The backdoor runs as a background process, initializing with log messages such as the one seen in the following image.



The log message shown above confirms a dual polling cadence: fast polling (5s) immediately after launch for rapid task acquisition, transitioning to steady-state polling (30s).

2. Initial check-in to C2

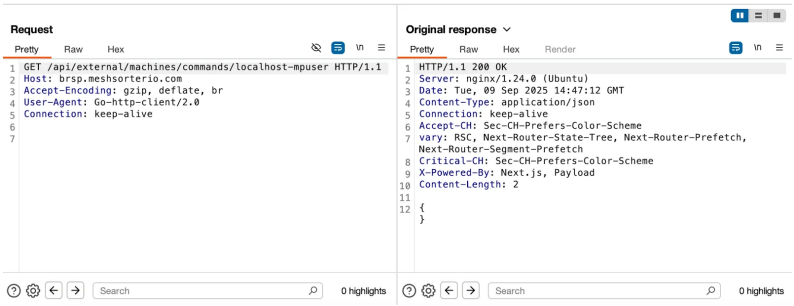
The agent then registers to the C2 via a POST request to /api/external/machines/me.



This suggests the AppleScript phase phishes credentials, and the Go backdoor is designed to receive and use them for privileged actions.

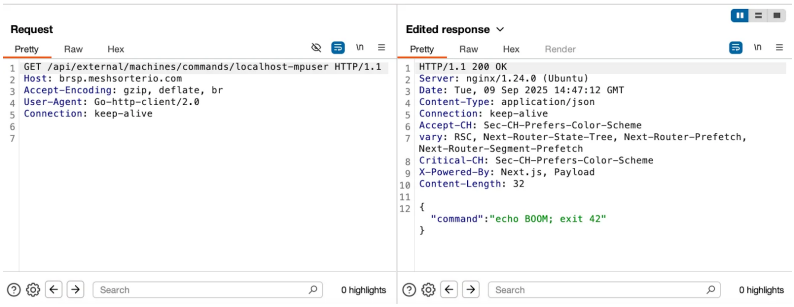
3. Polling for commands

At this point, the agent polls its task queue with a GET request to /api/external/machines/commands/<machine_id>. An empty response ({}) indicates no current tasking.



4. Server trust and command execution

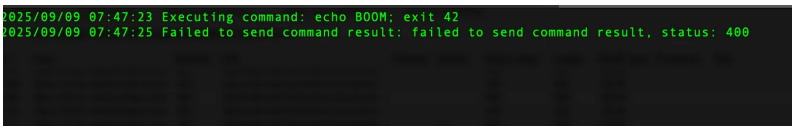
We tested the agent behavior by injecting a fake command, “echo BOOM; exit 42,” via BurpSuite.



As seen in the image above, the agent accepted and executed the command.

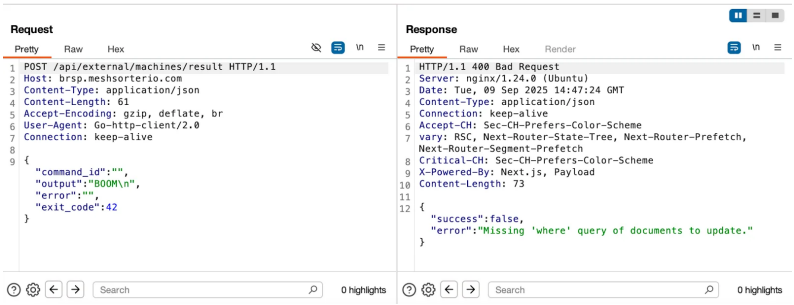
5. Logging and RCE

The agent now logs the local execution output.



6. Result reporting and protocol enforcement

The agent attempts to send a POST to /api/external/machines/result.



7. Internal JSON Schema from binary

A strings dump from the binary reveals all keys used in its C2 protocol: “os”, “arch”, “error”, “hostname”, “success”, “output”, “username”, “command”, “platform”, “command_id”, “exit_code”, “machine_id”, “sudo_password”.

A summing up of our analysis reveals a classic backdoor lifecycle:

1. **Registration**: Generates a machine ID (e.g., localhost-mpuser) and POSTs inventory to brsp[.]meshsorterio.com/api/external/machines/meon. Payload includes os: darwin, arch: arm64, hostname, username, and platform. Response schema includes a sudo_password field, hinting at integration with the stealer’s phishing for elevated creds
2. **Polling**: Dual cadence: fast (5s) for initial checks, normal (30s) thereafter, and GETs /api/external/machines/commands/<machine_id> for tasks
3. **Execution and reporting**: Executes received commands (e.g., our test echo BOOM; exit 42) and POSTs results to /api/external/machines/result with output, error, exit_code, and command_id (a tampered response confirmed remote code execution capability)

How MacSync differs from AMOS

In short, this isn’t just about more features. It’s a total architectural shift.

AMOS, updated in July 2025 with its own backdoor, relies on C-based components and shells out to curl for C2 communication. This creates noisy artifacts (process chains like osascript -> bash -> curl) that EDRs and IDS have hunted for years.

MacSync’s approach is stealthier:

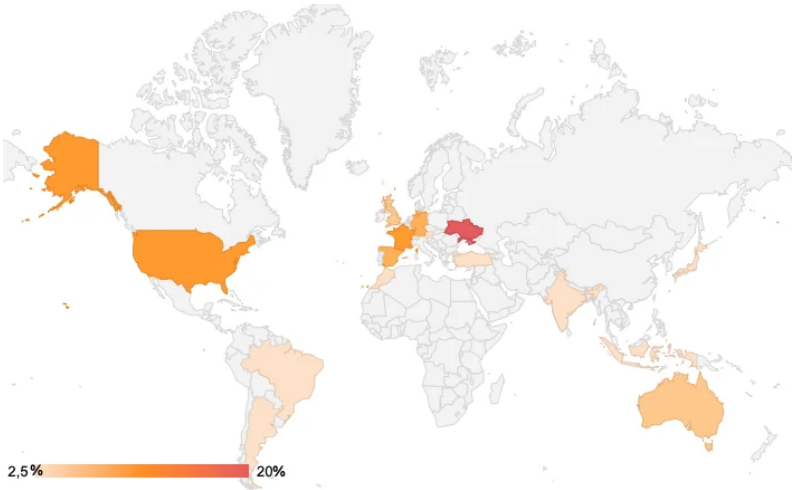
- **Language and networking**: Written in Go, MacSync uses the native net/http library for HTTPS requests. There are no external curlcalls, reducing host-level noise. Curl-hunting is mature, but Go’s embedded client evades those rules, pushing detection toward network heuristics.
- **Modularity**: The stealer (AppleScript) and backdoor (Go agent) are separate modules. The stealer phishes creds, while the backdoor reuses them (via sudo_password) for privileged RCE.
- **Protocol**: RESTful queue semantics enable scalable botnet ops.

MacSync in the wild: A glimpse at distribution

MacPaw's CleanMyMac telemetry confirms that MacSync has already reached users in multiple countries, with detections concentrated in Europe and North America.

The heatmap below visualizes the geographic distribution of observed stealer activity, with the highest share of detections coming from Ukraine, followed by the United States, Germany, and the UK.

While the infection volume remains relatively low overall (hundreds of detections compared to AMOS's tens of thousands), the spread across diverse regions signals growing adoption among traffer teams targeting macOS users globally.



The current MacSync stealer heatmap should be treated as directional, as coverage reflects both attacker reach and where we have visibility.

Wrapping up: A quiet evolution with loud implications

The story of MacSync is a reminder that in the world of malware, price and accessibility can beat elegance — at least at first.

What began as a budget-friendly stealer is now evolving into a modular remote access tool for macOS, complete with credential reuse support and REST-style tasking. This combination puts individual users at real risk of account takeovers and asset theft. Plus, it puts companies at risk of source code exposure, credential compromise, and follow-up attacks from Macs that often hold disproportionate access.

By ditching noisy shell commands in favor of native Go-based HTTP clients, MacSync is more difficult to catch. For defenders, this shift indicates a need to move detection from the command line to the network layer, while watching out for small, quiet agents doing very real damage.

Stay vigilant. The macOS threat space is heating up. If you spot more samples or variants, share them with @moonlock_lab. Let's collaborate to track this evolution together.

Indicators of compromise

type	indicator	notes
sha256	a42eece43aad2e2a2f98d41b1b48025c252452318a864d32d0f9156adaabe65b	Mach-O (Go). Registers, polls, executes, posts results.
sha256	cfd338c16249e9bcae69b3c3a334e6deafd5a22a84935a76b390a9d02ed2d032	Stealer/first stage; exfil + staging of Go agent.
domain	meshsorterio[.]com	Shared infrastructure
domain	www[.]meshsorterio[.]com	Front host
domain	brsp[.]meshsorterio[.]com	Registration, commands, results API
domain	gamma[.]meshsorterio[.]com	Delivers SHS.zip →

	./shell
domain d[.]meshsorterio[.]com	Related subdomain
domain plsp[.]meshsorterio[.]com	Related subdomain
domain cnct[.]meshsorterio[.]com	Related subdomain
domain con[.]meshsorterio[.]com	Related subdomain
domain dev[.]meshsorterio[.]com	Dev environment
domain staging[.]meshsorterio[.]com	Staging environment
domain testing[.]meshsorterio[.]com	Testing environment
domain rxkbnwuc[.]meshsorterio[.]com	Related subdomain
domain sphnugamma[.]meshsorterio[.]com	Related subdomain
domain b3e34878-5a7d-458b-8a35-3ea1dae23fdd[.]meshsorterio[.]com	Related subdomain
domain _msdcs[.]meshsorterio[.]com	Related subdomain
domain _tcp[.]meshsorterio[.]com	Related subdomain

Kseniia Yamburh

Kseniia is a malware research engineer at Moonlock, the cybersecurity division of MacPaw. She specializes in OSINT intelligence gathering and analysis. Her passion lies in writing about new investigations and findings in the field of cybersecurity.