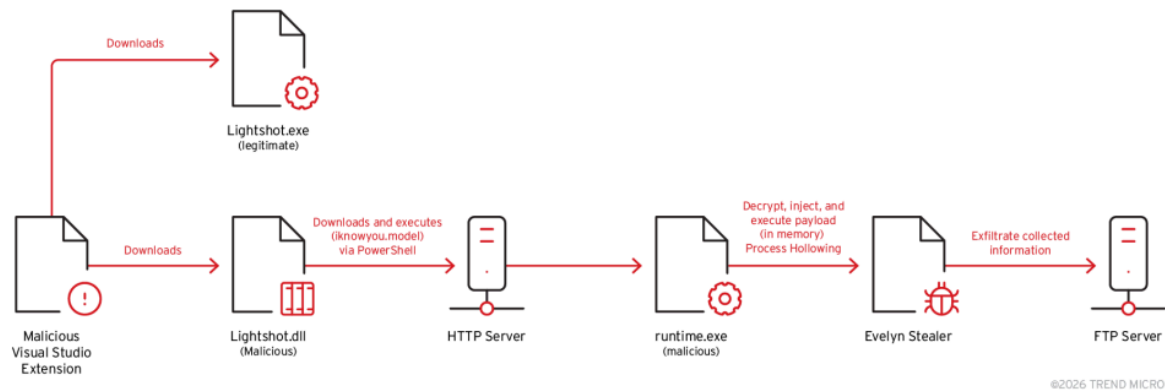


From Extension to Infection: An In-Depth Analysis of the Evelyn Stealer Campaign Targeting Software Developers



Malware

This blog entry provides an in-depth analysis of the multistage delivery of the Evelyn information stealer, which was used in a campaign targeting software developers.

By: Ahmed Mohamed Ibrahim Read time: 6 min (1670 words)

Key takeaways

- Analysis of the Evelyn Stealer campaign targeting software developers shows that threat actors are weaponizing the Visual Studio Code (VSC) extension ecosystem to deploy a multistage, information-stealing malware.
- The malware is designed to exfiltrate sensitive information, including developer credentials and cryptocurrency-related data. Compromised developer environments can also be abused as access points into broader organizational systems.
- This activity affects organizations with software development teams that rely on VSC and third-party extensions as well as those with access to production systems, cloud resources, or digital assets.
- TrendAI Vision One™ detects and blocks the indicators of compromise (IOCs) outlined in this blog, and provides customers with tailored threat hunting queries, threat insights, and intelligence reports.

On December 8, 2025, Koi.ai published their [findings](#) about a campaign specifically targeting software developers through weaponized Visual Studio Code extensions. Here, we'll provide a more in-depth analysis of the multistage delivery of the Evelyn information stealer.

Evelyn implements multiple anti-analysis techniques to evade detection in research and sandbox environments. It collects system information and harvests browser credentials through DLL injection as well as files and information such as clipboard and Wi-Fi credentials . It can also capture screenshots and steal cryptocurrency wallet. The malware communicates with its command-and-control (C&C) server over FTP.

Attack diagram

Figure 1. Attack chain of the Evelyn campaign
[download](#)

Technical analysis

First stage: Downloader

File name	SHA256	Type
Lightshot.dll	369479bd9a248c9448705c222d81ff1a0143343a138fc38fc0ea00f54fcc1598	DLL

Table 1. Details of the first-stage downloader

Upon successfully installing the malicious VSC extension, the downloader pretends to be a legitimate Lightshot DLL component, which is then executed by the genuine Lightshot.exe. The malware executes when the DLL is loaded into a process. Upon loading, the DLL immediately executes its payload. The malware implements a legitimate-looking Lightshot function exports to maintain its cover and implements a singleton pattern to ensure that it only executes once. It then launches a hidden PowerShell command to download and execute a second-stage payload and store it in the Local Temp directory as "runtime.exe".

Figure 2. Code snippet showing the logic of the first-stage downloader
[download](#)

Additionally, the downloader creates a mutual exclusion (mutex) object to ensure that only one instance of the malware can run at any given time, ensuring that multiple instances of the malware cannot be executed on a compromised host.

Figure 3. Code snippet showing how the mutex object is created and used by the first-stage downloader
[download](#)

The following is the download request sent by the downloader (shown in Figure 2) to download the second-stage injector:

Figure 4. Download request of the injector.
[download](#)

Second stage: Injector

File name	SHA256	Type
iknowyou.model	92af258d13494f208ccf76f53a36f288060543f02ed438531e0675b85da00430	EXE

Table 2: Details of the second-stage injector

The second-stage payload of this malware campaign is a process hollowing injector, designed to decrypt and inject a third-stage payload into the legitimate Windows process, "grpconv.exe". The malware uses AES-256-CBC encryption to decrypt the final payload, which is a copy of Evelyn Stealer.

Upon execution, the malware dynamically imports Windows APIs and creates a new instance of "grpconv.exe" using "CreateProcessA" with the CREATE_SUSPENDED flag. It then decrypts the final embedded payload (Evelyn Stealer), which is stored within the malware binary using AES-256-CBC encryption algorithm. The malware uses the following AES Key and IV to decrypt the embedded payload:

- **AES Key (32 bytes):** 2e649f6145f55988b920ff5a445e63aae29c80495b830e0d8bb4b3fff4b1f6f4
- **IV (16 bytes):** 5c507b22e9814428c5f2b1ef213c5c4a

Once the payload is decrypted, the malware injects it into the "grpconv.exe" and resumes the execution of the suspended process.

Figure 5. Code snippet showing how it creates the "grpconv.exe" process with a "suspend" flag to inject the final payload
[download](#)

Final payload: Evelyn Stealer

File name	SHA256	Type
EvelynStealer.exe	aba7133f975a0788dd2728b4bbb1d7d948e50571a033a1e8f47a2691e98600c5	EXE

Table 3: Details of Evelyn Stealer

Upon execution of Evelyn Stealer, the malware dynamically resolves all Windows APIs needed for malware operations, including process injection, file operations, registry access, network communication, and clipboard access. The malware employs multiple layers of evasion techniques specifically designed to thwart security researchers, automated analysis systems, and sandbox environments. The malware implements different virtual machine detection methods, debugger detection, and specialized checks for analysis environments like Remote Desktop Protocol (RDP) sessions and Hyper-V.

The following are the list of anti-VM and anti-sandbox techniques used by the malware:

- **GPU analysis:** Detects VMware, VirtualBox, Hyper-V, Parallels, QEMU, VirtIO, and basic display adapters
- **Hostname analysis:** Checks the computer name for VM indicators
- **Disk size analysis:** Flags systems with less than 60 GB of disk space, such as VMs
- **Process analysis:** Scans for VM-related processes (e.g., vmttoolsd.exe, vboxservice.exe)
- **Registry analysis:** Checks hardware registry keys for VM identifiers, a relatively sophisticated evasion technique to avoid analysis environments

Figure 6. Evelyn Stealer's API resolving and evasion logic
[download](#)

After successful initialization and environment validation, the malware establishes its operational workspace by creating a dedicated folder structure in the user's AppData directory for storing collected data.

Figure 7. Creation of Evelyn directory to store the collected data

[download](#)

The malware performs a critical preparatory phase that involves recovering any existing browser data and terminating active browser processes. This two-step approach ensures maximum data collection efficiency and eliminates potential conflicts during the injection process, suggesting that the threat actors have a deeper understanding of browser security mechanisms.

Figure 8. Code snippet showing how the malware recovers browser data and terminates browser processes

The malware then implements a sophisticated multi-tier strategy for acquiring its critical browser injection component "abe_decrypt.dll". This component is essential for the malware's capabilities to extract browser credentials and represents a key dependency for successful data theft operations. The malware first checks the TEMP directory, which suggests that it could have been previously deployed or cached from earlier infections. If not found in the TEMP directory, the malware tries to download it from the FTP server. Lastly, it tries to load it from the current directory.

Figure 9. The acquisition logic of abe_decrypt.dll, Evelyn Stealer's critical browser injection component

[download](#)

Figure 10. The FTP download logic of abe_decrypt.dll

[download](#)

Figure 11. FTP requests showing abe_decrypt.dll being downloaded

[download](#)

Once the malware acquires abe_decrypt.dll, the malware targets browsers by implementing a process creation and DLL injection technique specifically designed to compromise browser security mechanisms. It bypasses many standard protection mechanisms, including sandboxing, extension-based security tools, and user interface (UI) protections. The malware constructs an extensive command line with more than 15 browser flags, specifically designed to minimize detection and forensic traces:

- "--headless=new": Runs browser without GUI (invisible operation)
- "--disable-gpu": Prevents GPU acceleration (reduces system footprint)
- "--no-sandbox": Disables browser security sandbox (enables deeper access)
- "--disable-extensions": Prevents legitimate security extensions from interfering
- "--disable-logging": Eliminates browser log generation
- "--silent-launch": Suppresses startup notifications
- "--no-first-run": Bypasses initial setup dialogs
- "--disable-popup-blocking": Ensures malicious content can execute
- "--window-position=-10000,-10000": Positions window off-screen
- "--window-size=1,1": Minimizes window to 1x1 pixel

Figure 12. Code snippet showing the malware's browser DLL injection technique

[download](#)

Figure 13. Code snippet showing how the malware creates a new browser process with injected DLL

[download](#)

Additionally, the malware captures desktop screenshots and collects various information from the infected machine, including the following:

- System information such as username, computer name, OS version, installed software, running processes, sensitive files, VPN configuration, and more
- Cryptocurrency wallets
- Clipboard data
- Wi-Fi passwords

Figure 14. How Evelyn Stealer gathers the target's information

[download](#)

Once the malware collects all the necessary information, it archives the data into a ZIP file and sends it to the attacker's C&C server over FTP. The malware employs a detailed filename format containing 15 components to ensure comprehensive data tracking and organization. The filename format takes the form of:

```
| {COUNTRY_CODE}-{IP_ADDRESS}-{USERNAME}-{OS_VERSION}-{CRYPTO_FOUND}-{PAYPAL_FOUND}-{CRYPTO_WEBSITES}-{RAM_INFO}-{GPU_INFO}-{METAMASK}-{PHANTOM}-{TRUSTWALLET}-{OTHER_WALLETS}-{TIMESTAMP}.zip.
```

Figure 15. Code snippet showing how the stolen data is archived and uploaded

[download](#)

Figure 16. Code snippet showing the malware's FTP exfiltration logic

Figure 17. FTP request showing the exfiltration of the stolen information

[download](#)

Conclusion

The Evelyn Stealer campaign reflects the operationalization of attacks against developer communities, which are seen as high-value targets given their important role in the software development ecosystem. By embedding itself in VSC extensions and staging its execution through loaders and process hollowing, the campaign treats the developer environment itself as the delivery mechanism. Reinforcing its attack chain with advanced capabilities such as AES-256-CBC encryption, multilayered anti-analysis techniques, and an apparently disciplined operational security, the campaign underscores a level of maturity designed to evade detection while exploiting the implicit trust developers place in their tools.

As developers increasingly become prime targets due to their privileged access and cryptocurrency holdings, organizations must implement comprehensive security measures including extension vetting, behavioral monitoring, and zero-trust architectures specifically designed for development workflows.

The technical and operational maturity demonstrated by this campaign suggests that we're likely to see more targeted attacks against developer communities in the future, especially as more teams and companies adopting AI-powered tools and extensions, which further expands the attack surface. Security teams must adapt their strategies to protect these high-value users who serve as gateways to critical systems and intellectual property.

Proactive security with TrendAI Vision One™

[TrendAI Vision One™one-platform](#) is the industry-leading AI cybersecurity platform that centralizes cyber risk exposure management, security operations, and robust layered protection.

TrendAI Vision One™ Network Security

| TCP: Trojan.Win64.EvelynStealer.A Runtime Detection

TrendAI Vision One™ Threat Intelligence Hub

[TrendAI Vision One™ Threat Intelligence Hubproducts](#) provides the latest insights on emerging threats and threat actors, exclusive strategic reports from TrendAI™ Research, and TrendAI Vision One™ Threat Intelligence Feed in the TrendAI Vision One™ platform.

Emerging Threats: [Dissecting Evelyn Stealer: A Comprehensive Analysis of a Multi-Stage Data Theft Campaign](#)

TrendAI Vision One™ Intelligence Reports (IOC Sweeping)

[Dissecting Evelyn Stealer: A Comprehensive Analysis of a Multi-Stage Data Theft Campaign](#)

Hunting Queries

TrendAI Vision One™ Search App

TrendAI Vision One™ customers can use the Search App to match or hunt the malicious indicators mentioned in this blog post with data in their environment.

Hunting query for Evelyn Stealer C2

| "eventSubId:204 AND request:\"server09.mentality.cloud\""

More hunting queries are available for TrendAI Vision One™ with Threat Intelligence Hub entitlement enabled.

Indicators of Compromise (IoCs)

File	SHA256	Detection name
Lightshot.dll	369479bd9a248c9448705c222d81ff1a0143343a138fc38fc0ea00f54fcc1598	Trojan.Win32.DOWNLOAD
iknowyou.model	92af258d13494f208ccf76f53a36f288060543f02ed438531e0675b85da00430	Trojan.Win64.MALINJECT
EvelynStealer.exe	aba7133f975a0788dd2728b4bbb1d7d948e50571a033a1e8f47a2691e98600c5	Trojan.Win64.EVELYNSTEAL
abe_decrypt[1].dll	74e43a0175179a0a04361faaaaf05eb1e6b84adca69e4f446ef82c0a5d1923d5	Trojan.Win64.BADBROWSE
server09.mentality.cloud	FTP Server	
syn1112223334445556667778889990.org	HTTP Server	

Tags

[Malware](#) | [Research](#) | [Articles, News, Reports](#) | [Cyber Threats](#)