Hunting For TamperedChef Infostealer

By Ameer Mane

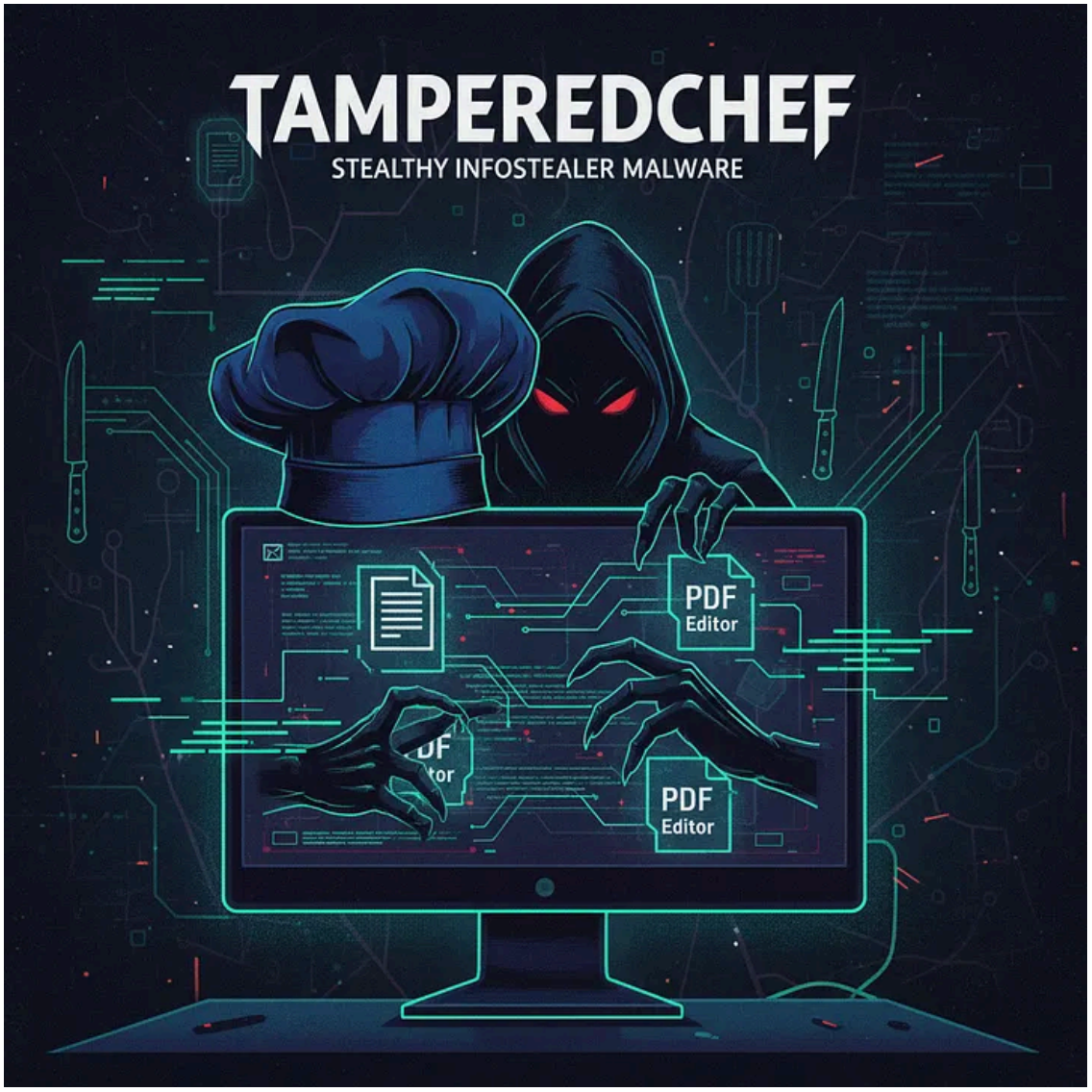
Published: 2025-09-21 · Archived: 2026-04-06 01:03:40 UTC

A

4 min read

Sep 21, 2025

Press enter or click to view image in full size



What if that innocent-looking “free PDF editor” you downloaded wasn’t really what it claimed to be? What if, weeks later, it quietly reached back to the attacker, stole your passwords, and set up camp on your system?

That's the story of **TamperedChef** — a recently observed infostealer that dresses up as helpful software, only to reveal its true colors later.

Security researchers found TamperedChef bundled in malicious installers (like fake PDF editors), spread through **malvertising campaigns** and **poisoned download sites**. The clever twist? It often **sleeps for weeks** before going live, letting it spread silently and then activate when no one's watching.

In further investigation, I observed that it's **not limited to PDF tools**, but extended to applications such as:

`totalusermanuals` , `manualreaderpro` , `allmanualsreader` , `justaskjacky` , `AllManualsReader` , etc.

How Does It Work?

TamperedChef is delivered via **user-initiated downloads**. It comes bundled with seemingly legitimate software like PDF editors or manual readers.

Once executed, it silently triggers **Node.js (node.exe)** via the Windows command line. Node.js executes a JavaScript payload dropped in the user's temporary folder.

```
"C:\Windows\System32\cmd.exe" /C start "" /min "C:\Users\<user_name>\AppData\Local\Programs\AllManualsReader\node\node.exe" "C:\Users\<user_name>\AppData\Local\Programs\AllManualsReader\2d4d7602-8032-4207-a03f-be08e68d1094.js"
```

Breaking it Down

- `cmd.exe /c start "" /min` – Runs a minimized command prompt silently.
- `node.exe` – Node.js runtime executes the JavaScript payload.
- **Payload Location** — The `[GUID]of.js` file in `%TEMP%` contains routines to harvest data and communicate with the attacker.

Payload Naming Convention

- Typically named with a **GUID suffix** ending with **"or"**, **"ro"**, or **"of"**.
- Consistently observed in internal investigations and public malware samples, making it a **useful hunting indicator**.

Persistence

TamperedChef creates a **scheduled task** to remain active across reboots:

```
C:\windows\system32\cmd.exe /d /s /c "schtasks /Create /TN "sys_component_health_{UID}" /TR "\"C:\Windows\system32\cmd.exe\" /c start \"\" /min \"%^LOCALAPPDATA^%\Programs\AllManualsReader\2d4d7602-8032-4207-a03f-be08e68d1094.js\"" /SC DAILY /ST 10:51 /RI 240 /DU 24:00 /F"
```

Breaking it down:

- `/TN "sys_component_health_{UID}"` – Uses a legitimate-sounding task name.
- `/TR` – Runs Node.js to execute the payload in minimized mode.
- `/SC DAILY /ST 10:51 /RI 240 /DU 24:00` – Executes daily with a repetition interval of 240 minutes.
- `/F` – Overwrites existing tasks with the same name.

This ensures TamperedChef **maintains stealthy persistence**, enabling long-term data exfiltration.

Reconnaissance

TamperedChef actively surveys the system to identify browsers and security software.

1. WMI for Process Enumeration

The malware uses **Windows Management Instrumentation (WMI)** to check if browsers like Chrome or Edge are running:

```
C:\windows\system32\cmd.exe /d /s /c “powershell.exe “Get-WmiObject Win32_Process | Where-Object { $_.Name -eq ‘chrome.exe’ }””
```

```
C:\windows\system32\cmd.exe /d /s /c “powershell.exe “Get-WmiObject Win32_Process | Where-Object { $_.Name -eq ‘msedge.exe’ }””
```

This helps target credentials and active sessions.

2. Software Enumeration via Registry

It queries the registry to enumerate installed **security and antivirus software**, often searching for uninstall strings or configuration keys:


```
reg query “hkcu\software\microsoft\windows\currentversion\uninstall\episoftware epibrowser” /v “uninstallstring”
reg query “hkcu\software\zillya\zillya antivirus”
reg query “hkcu\software\kasperskylabsetup”
reg query “hklm\software\fortinet”
reg query “hkcu\software\checkpoint\zang”
reg query “hklm\software\classes\g data antivirus”
reg query “hklm\software\wow6432node\microsoft\windows\currentversion\uninstall\g data antivirus” /v “uninstallstring”
reg query “hklm\software\wow6432node\microsoft\windows\currentversion\uninstall\{4073cd02-7996-48d7-b68e-297676c27ca6}” /v “uninstallstring”
```

```
reg query “hklm\software\wow6432node\microsoft\windows\currentversion\uninstall\rec” /v “uninstallstring”
reg query “hklm\software\microsoft\windows\currentversion\uninstall\bitdefender” /v “uninstallstring”
reg query “hklm\software\microsoft\windows\currentversion\run\bitdefender” /v “uninstallstring”
reg query “hklm\software\microsoft\windows\currentversion\uninstall\{96a251bd-7532-4cf9-b87d-158fc685dbc4}” /v “uninstallstring”
```

By gathering this information, TamperedChef **maps the system’s security landscape**, allowing it to avoid or disable defenses.

Credential Harvesting

Targets include:

- **Browser SQLite databases** — `Login Data` and `Web Data` from Chrome and Edge.
- **Windows DPAPI secrets** — Local system-level encrypted data.
- **Local password managers** — Any stored credentials accessible to the user context.

To facilitate extraction, the malware may forcibly terminate browser processes:

```
taskkill /F /IM msedge.exe
taskkill /IM msedge.exe
taskkill /F /IM chrome.exe
taskkill /IM chrome.exe
```

File Duplication for Credential Extraction:

Copies of key browser files are created with `"Sync"` appended:

```
C:\Users\<user_name>\AppData\Local\Microsoft\Edge\User Data\Default\Web Data Sync
C:\Users\<user_name>\AppData\Local\Microsoft\Edge\User Data\Default\Preferences Sync
C:\Users\<user_name>\AppData\Local\Google\Chrome\User Data\Default\Preferences Sync
```

This allows attackers to extract credentials **without corrupting original files**.

Command & Control (C2)

TamperedChef communicates with attacker-controlled servers:

- **Domains:** `api.[random18].com`
- **Ports:** 8080, 443 (uncommon for typical apps)

This traffic delivers harvested credentials and may receive updated payloads or instructions.

MITRE ATT&CK Mapping

- **Execution:** T1059 — Command & Scripting Interpreter
- **Credential Access:** T1555 — Credentials from Password Stores, T1003 — OS Credential Dumping
- **Defense Evasion:** T1218 — Signed Binary Proxy Execution, T1036 — Masquerading
- **Persistence:** T1053 — Scheduled Task/Job
- **Discovery:** T1082 — System Information Discovery, T1012 — Query Registry

Hunting Queries

1. Execution of javascript through `node.exe` executed by `cmd.exe` `processCmd:node.exe AND eventId: 1 AND processCmd:cmd.exe AND ProcessCmd:start`
2. Registry Enumeration:
   `commandline:"*reg query*" AND commandline:("*kasperskylabsetup*" OR "*bitdefender*" OR "*cryptography*" OR "*antivirus*")`
3. Browser Termination:
   `commandline:"*taskkill*" AND commandline:("*msedge.exe*" OR "*chrome.exe*")`
4. File Access (Web Data / Login Data in temp folder)
5. Outbound Traffic:

```
logs
| where tostring(url) matches regex @"api\.[a-zA-Z0-9]{18}\.com"
| extend Domain = extract(@"api\.([a-zA-Z0-9]{18})\.com", 1, tostring(url))
| where Domain matches regex @".*\d.*"
| project tostring(url)
```

IoCs:

C2 Domains:

`api[.]cjby76nlcynrc4jvrb[.]com`
`api[.]j6vmldsufhwx8zn69z[.]com`
`api[.]k2ioeasm874fnacr9x[.]com`
`api[.]pyej17uw09d1bqlndg[.]com`
`api[.]vtqgo0729ilnmyxs9q[.]com`

References:
