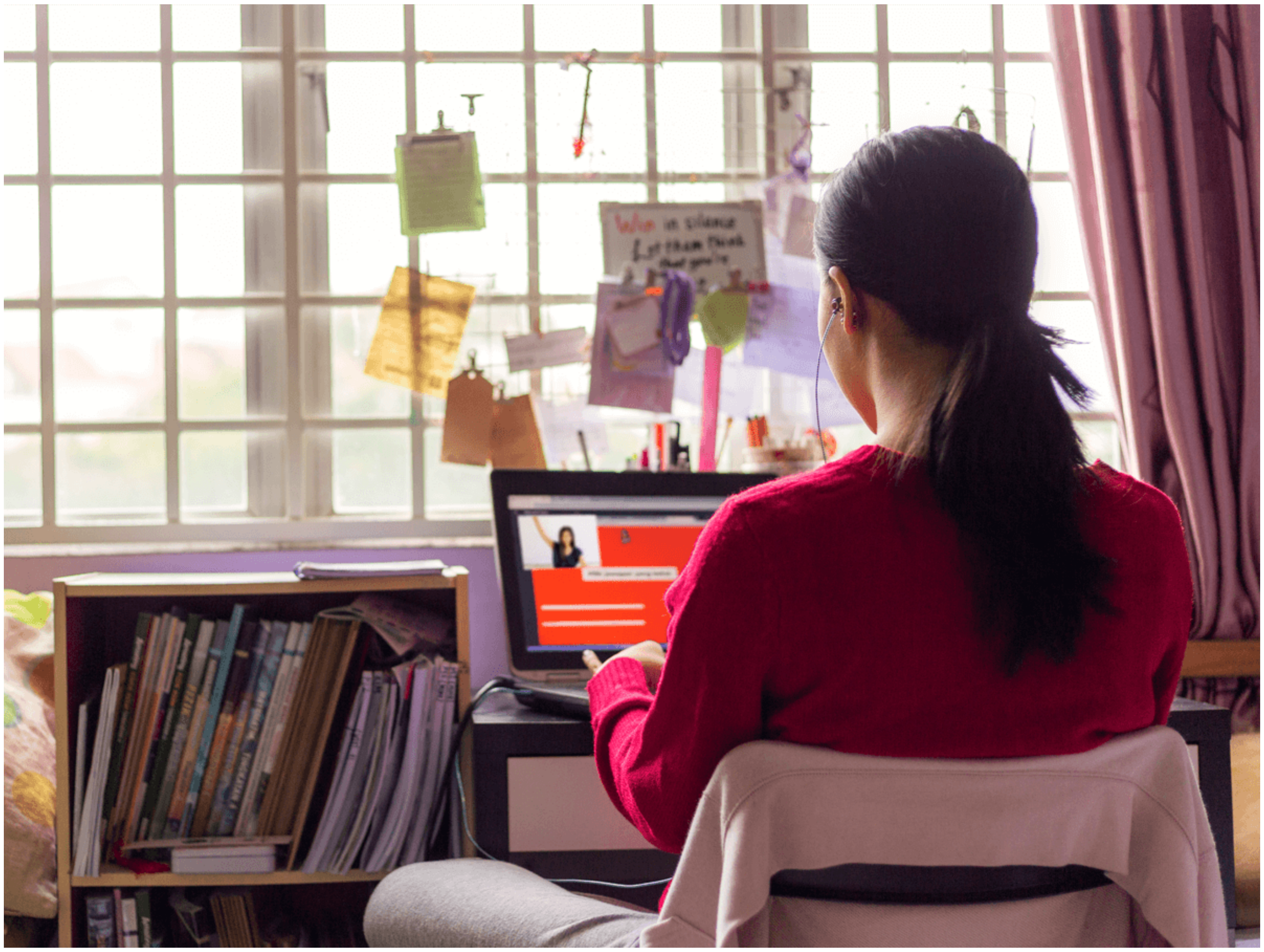
Deceptive Cracked Software Spreads Lumma Variant on YouTube

January 8, 2024



Article Contents

By Cara Lin | January 08, 2024

Affected Platforms: Microsoft Windows
Impacted Users: Microsoft Windows
Impact: The information collected can be used for future attacks
Severity Level: High

FortiGuard Labs recently discovered a threat group using YouTube channels to distribute a Lumma Stealer variant. We found and reported on a similar attack method via YouTube in March 2023. These YouTube videos typically feature content related to cracked applications, presenting users with similar installation guides and incorporating malicious URLs often shortened using services like TinyURL and Cuttly. To circumvent straightforward web filter blacklists, the attackers exploit open-source platforms like GitHub and MediaFire instead of deploying their malicious servers. In this case, the shared links lead to the direct download of a new private .NET loader responsible for fetching the final malware, Lumma Stealer.

Lumma Stealer targets sensitive information, including user credentials, system details, browser data, and extensions. It has been advertised on the dark web and a Telegram channel since 2022, with over a dozen observed command-and-control (C2) servers in the wild and multiple updates. Figure 1 shows Lumma Stealer's C2 server telemetry, illustrating a global presence with a peak observed in December.

In this article, we will elaborate on each stage's behaviors that facilitated the stealer's distribution.

Figure 1: Telemetry of Lumma Stealer’s C2

Figure 2: Attack flow

Initial Infection Vector

The hacker initially breaches a YouTuber's account and uploads videos masquerading as sharing cracked software. Figure 3 shows the video descriptions in which a malicious URL is embedded, enticing users to download a ZIP file that harbors malicious content for the next stage of the attack. The videos were uploaded earlier this year, but the files on the file-sharing site receive regular updates (Figure 4), and the number of downloads keeps growing. This indicates that the ZIP file is always new and that this method effectively spreads malware.

Figure 3: The hacked YouTube Channel with a similar fake software installation guide

Figure 4: The malicious files updated on MediaFire
The ZIP file, “installer_Full_Version_V.1f2.zip,” contains an LNK file that calls PowerShell to download a .NET execution file via the GitHub repository “New” owned by John1323456 (Figure 6). The abbreviated URL, “hxxp://cutt[.]ly/lwD7B7lp,” connects to “hxxps://github[.]com/John1323456/New/raw/main/Installer-Install-2023_v0y.6.6[.]exe.” The other two repositories, “LNK” and “LNK-Ex,” also include NET loaders and spread InfoStealer as the final payload.

Figure 5: Malicious LNK file content

Figure 6: .NET executable on GitHub

.NET Executable – Installer-Install-2023_v0y.6.6.exe

The private .NET loader is obfuscated with SmartAssembly. It first gets the system's environment value, shown in Figure 7. Once the number of the data is correct, it proceeds to load the PowerShell script. Otherwise, it exits the program.

Figure 7: Getting the system's information by GetEnvironmentVariables()

Figure 8 shows the construction of a dictionary that defines the following properties of the ProcessStartInfo object that the malicious code uses to execute discreetly and avoid raising suspicion from its victims:

- RedirectStandardInput: This property set to true enables the redirection of the standard input stream of the process.
- CreateNoWindow: This property set to true indicates that the process should not create a visible window when it starts. It allows the executed command or script to run without displaying a command prompt window.
- UseShellExecute: This property set to false specifies that the process should not use the user's default shell for execution. Instead, the process is executed directly.

Figure 8: The ProcessStartInfo dictionary

Next, the ProcessStartInfo object is employed to launch the PowerShell process, wherein the PowerShell script is directed to the process's standard input. Figure 9 illustrates a portion of the code and the newly generated process.

Figure 9: Creates process and loads the PowerShell script

Figure 10 shows the partial PowerShell code from the private .NET loader. The script encodes the server IP address in Base64 and encompasses four servers. It assesses the system date, choosing the appropriate IP to retrieve encrypted binary data. To obfuscate analysis, the script incorporates a substantial amount of extraneous code. Figure 11 shows captured traffic downloaded from the first server, 176[.]113[.]115[.]224:29983.

Figure 10: Downloaded encrypted data from a server

Figure 11: The encrypted data from a remote server

After receiving the data, the script decrypts it using AES CBC, followed by GZip decompression to obtain the DLL file for the next stage. It then invokes the DLL file with a specific method and type via "[System.Reflection.Assembly]::Load()," as shown in Figure 12.

Figure 12: Loads decrypted data for the next stage

DLL File – Agacantwhitey.dll

Figure 13 shows the targeted function "PerkyRiggal," which is pivotal in inspecting the system and environment. It employs several PNG files in the Resources section to decipher the ultimate payload of Lumma Stealer. To avoid detection, the file encodes all its strings using the "BygoLarchen" method. Figure 14 demonstrates the function of decoding the target text with a predefined key string.

Figure 13: Targeted method and Resources in Agacantwhitey.dll

Figure 14: The method for decoding strings
It checks the following items to achieve Anti-VM and Anti-Debug:

- Verifies the user's active window by invoking “GetForegroundWindow” and assesses whether it contains any of the specified remote debugger strings, “x32dbg,” “x64dbg,” “windbg,” “ollydbg,” “dnspy,” “immunity debugger,” “hyperdbg,” “debug,” “debugger,” “cheat engine,” “cheatengine,” “ida.”
- Checks for the following modules about security appliances or sandboxes, “SbieDll.dll” (Sandboxie), “cmdvrt64.dll” (Comodo Antivirus), “cuckoomon.dll” (Cuckoo Sandbox), and “SxIn.dll” (360 Total Security). It then attempts to locate wine_get_unix_file_name to determine if Wine is being used in an analysis environment.
- Examines the presence of the following sandbox usernames: “Johnson,” “Miller,” “malware,” “maltest,” “CurrentUser,” “Sandbox,” “virus,” “John Doe,” “test user,” “sand box,” and “WDAGUtilityAccount.”
- Detects the presence of popular virtualization platforms via WMI queries “Select * from Win32_ComputerSystem” to retrieve information about the computer system, including the manufacturer and model. It then examines the manufacturer names, such as “innotek gmbh” (associated with VirtualBox) and “microsoft corporation” (often linked to Hyper-V), along with the model names “VirtualBox” and “vmware.” It also checks for directory “C:\Program Files\VMware” and “C:\Program Files\oracle\virtualbox guest additions.”
- Checks if the following files exist in folder “C:\Windows\system32\,” which are indicative of a virtualized environment presence: “balloon.sys,” “netkvm.sys,” “vioinput,” “viofs.sys,” “vioser.sys,” “VBoxMouse.sys,” “VBoxGuest.sys,” “VBoxSF.sys,” “VBoxVideo.sys,” “vmmouse.sys,” and “vboxogl.dll.”
- Checks the following system services: "vmbus," "VMBusHID," and "hyperkbd."
- Inspects the following process names: "vboxservice," "VGAuthService," "vmusrvc," and "qemu-ga."

After completing all environment checks, the program decrypts the resource data and invokes the "SuspendThread" function. This function is employed to transition the thread into a "suspended" state, a crucial step in the process of payload injection (see Figure 15).

Figure 15: Inject the final payload

Lumma Stealer Variant

Lumma stealer is a type of malware that can steal sensitive information from a user’s computer. It can target the system data, the browsers, crypto wallets, and browser extensions. It is written in C language and sold on underground forums. To elude detection and analysis, it employs diverse obfuscation techniques. The malware establishes communication with a command and control server, facilitating the exchange of instructions and transmitting pilfered data.

Figure 16 shows the method to contact a command and control (C2) server. Once it gets the first server that can set up a connection, it then sends out a POST message with hardcoded User-Agent “Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Safari/537.36“ and parameter “act=life” to check-in. The corresponding code is shown in Figure 17.

Figure 16: Resolve the C2 server list

Figure 17: Sending check-in message to C2 server
Next, it sends a POST request with the Lumma ID and “act=receive-message,” shown in Figure 18. Then, the compressed stolen data is uploaded to the C2 server with URI “/api.” Although the version is still “4.0,” Lumma Stealer has recently updated its exfiltration to leverage HTTPS to better evade detection.

Figure 18: POST request to C2 server with Lumma ID “JVryU4--LNK”

Figure 19: The zip file

Conclusion

In this attack, the malicious actor targets YouTube channels to disseminate Lumma Stealer. The crafted installation ZIP file serves as an effective bait to deliver the payload, exploiting the user's intention to install the application and prompting them to click the installation file without hesitation. URLs from open-source websites throughout the scheme aim to diminish user awareness. The attackers further deploy a private .NET loader with environment checks, various anti-virtual machine (Anti-VM), and anti-debugging functions. Users must exercise caution regarding unclear application sources and ensure legitimate applications from reputable and secure origins are used.

Fortinet Protections

The malware described in this report are detected and blocked by FortiGuard Antivirus as:

W32/Stealer.QLD!tr
MSIL/Agent.WML!tr
MSIL/Kryptik.BJF!tr
LNK/Agent.WML!tr

FortiGate, FortiMail, FortiClient, and FortiEDR support the FortiGuard AntiVirus service. The FortiGuard AntiVirus engine is a part of each of those solutions. As a result, customers who have these products with up-to-date protections are protected.

The URLs are rated as “Malicious Websites” by the FortiGuard Web Filtering service.

We also suggest that organizations go through the free Fortinet Certified Fundamentals (FCF) in Cybersecurity training. The training is designed to help end users learn about today's threat landscape and will introduce basic cybersecurity concepts and technology.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you believe this or any other cybersecurity threat has impacted your organization, please contact our Global FortiGuard Incident Response Team.

IOCs

IP Addresses

176[.]113[.]115[.]224
176[.]113[.]115[.]226
176[.]113[.]115[.]227
176[.]113[.]115[.]229
176[.]113[.]115[.]232

Hostnames

Netovrema[.]pw
opposesicknessopw[.]pw
politefrightenpowoa[.]pw
chincenterblandwka[.]pw

Files

48cbeb1b1ca0a7b3a9f6ac56273fbaf85e78c534e26fb2bca1152ecd7542af54
483672a00ea676236ea423c91d576542dc572be864a4162df031faf35897a532
01a23f8f59455eb97f55086c21be934e6e5db07e64acb6e63c8d358b763dab4f
7603c6dd9edca615d6dc3599970c203555b57e2cab208d87545188b57aa2c6b1