XWorm Attack Chain: Leveraging Steganography from Phishing Email to Keylogging via C2 Communication

By Sarviya

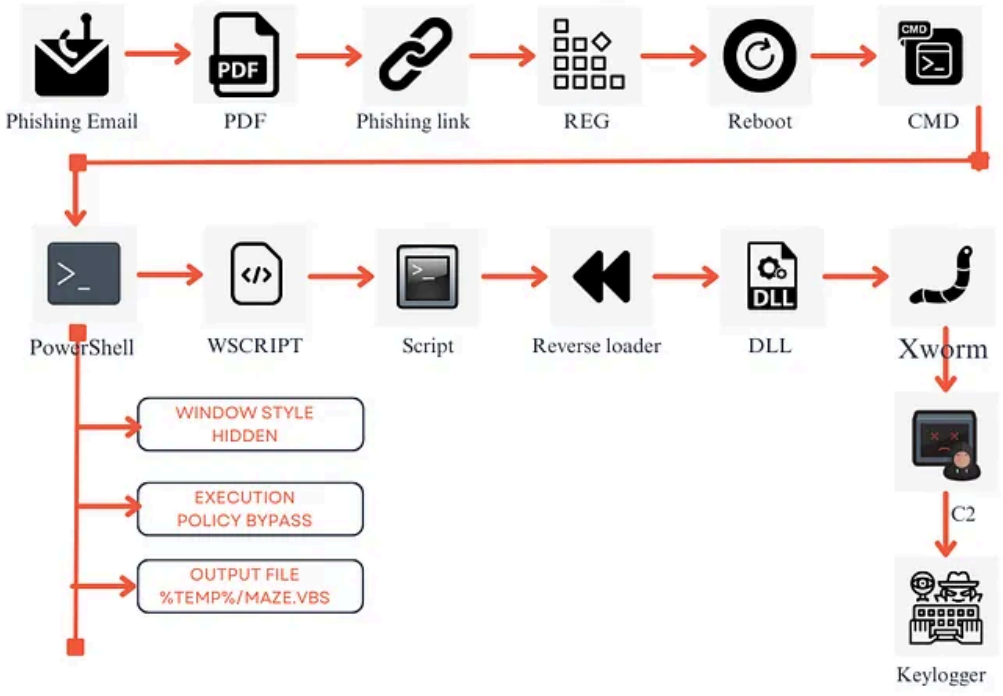
Published: 2025-09-12 · Archived: 2026-04-05 17:44:22 UTC

t

stegocampaign is a cyberattack using steganography to hide malware in images, making detection difficult. It delivers malware like AgentTesla, FormBook, Remcos, and LokiBot, using hidden payloads in images. Victims face data theft, remote control attacks, and credential harvesting.

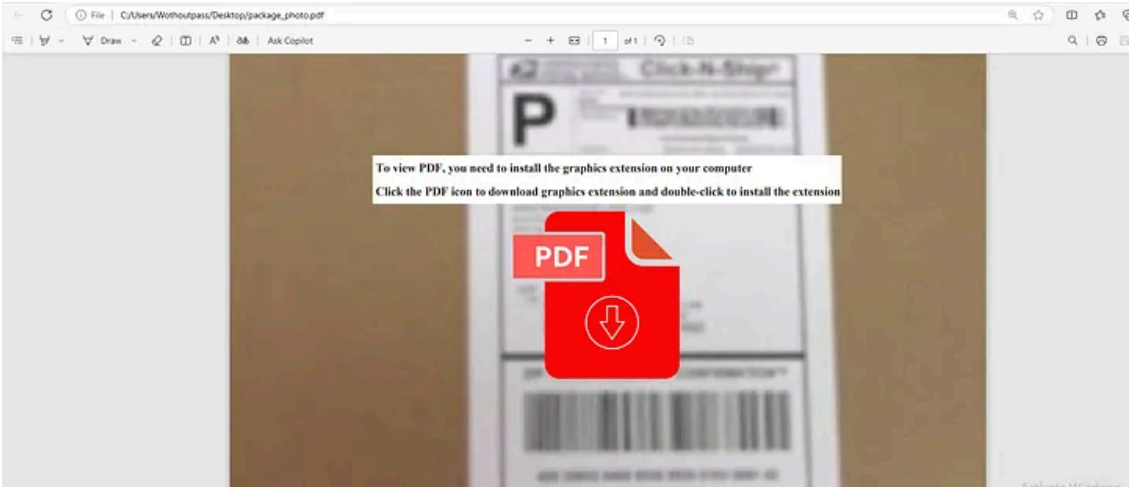
Steganography is the practice of concealing information within other media, such as images, audio files, and GIFs. Recently, we identified an active StegoCampaign and decided to investigate it further. In this blog, we will dive deep into the detailed kill chain of this campaign. Let’s get started!

Press enter or click to view image in full size



stegocampaign- XWorm Attack Chain

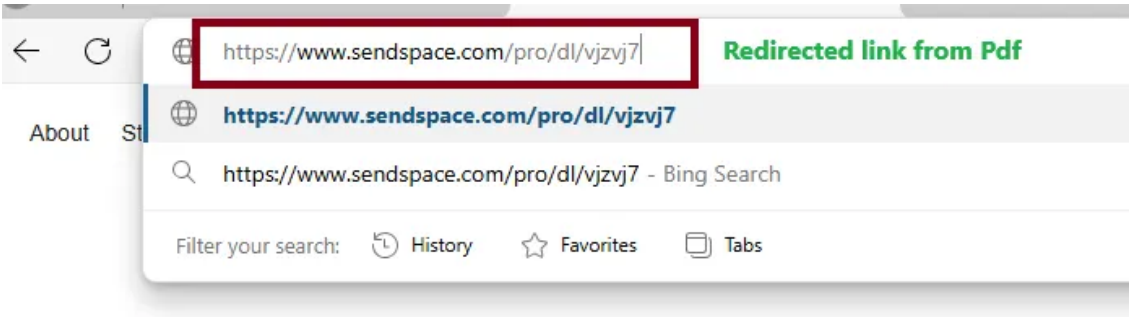
Press enter or click to view image in full size



Attached PDF from the phishing mail

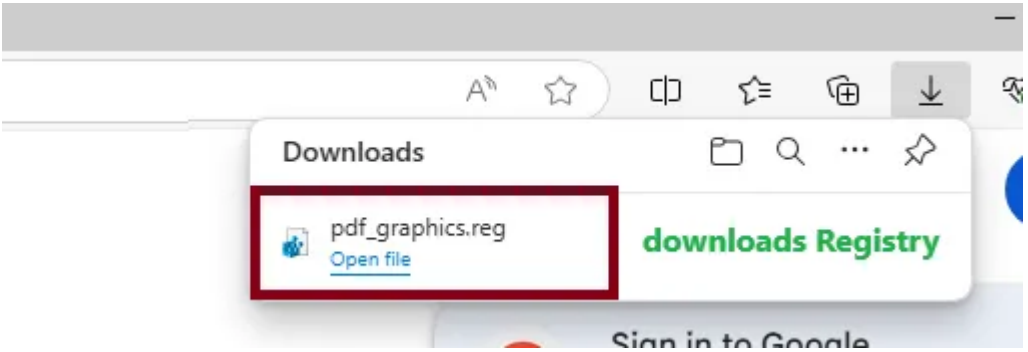
So the initial vector of this campaign is a phishing mail that comes with a attached Pdf and on viewing that it shows “Download Graphics extension” to view the PDF.

Press enter or click to view image in full size



URL redirected from the PDF

The above URL is the one that the PDF file redirected to download the graphics extension but instead it downloads a registry entry.



Registry File getting downloaded

Analysis Reg File

On examining the Reg file, we can see that ‘Run’ entry being added to the Values. The value corresponds to calling powershell via Cmd with window hidden and execution policy bypassed to download a vbs file from the

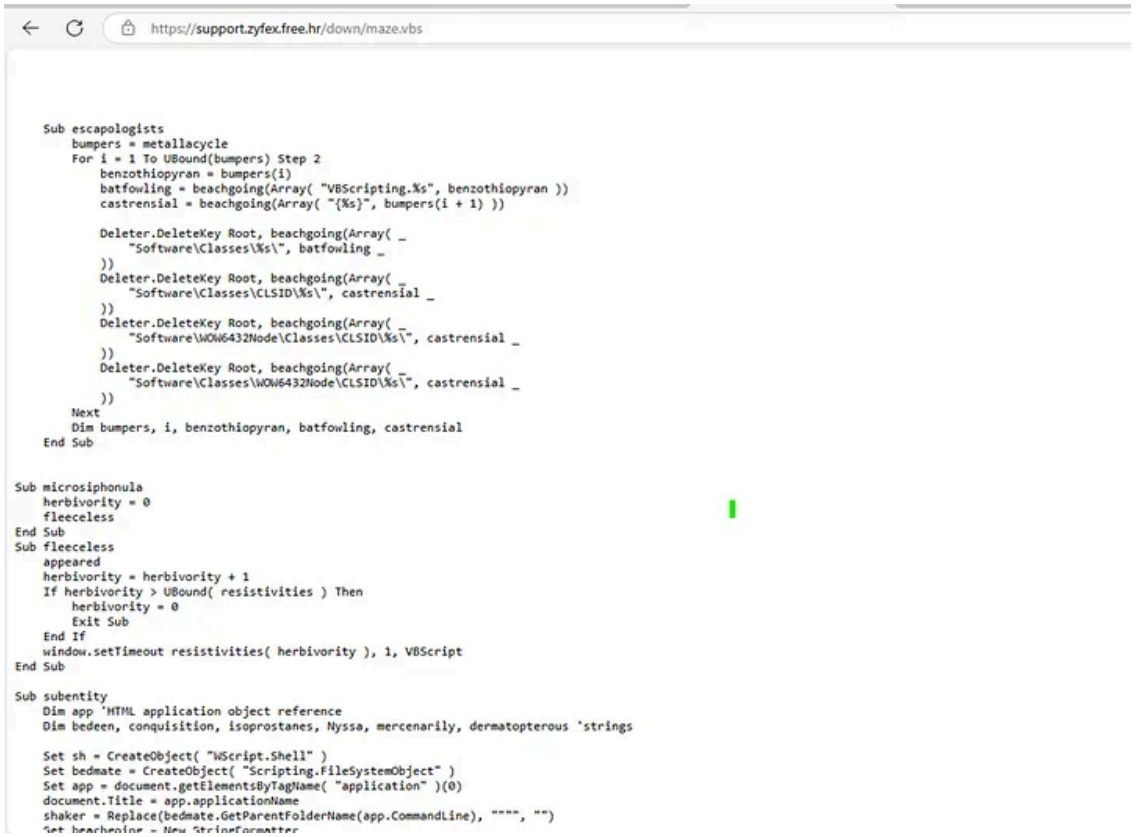
URL and saving that as “maze.vbs” in Temp location and it got executed right away(As per the script). since it’s a run entry, only when the system reboots it runs, in the mean time it stays till the user reboots.

Press enter or click to view image in full size

Persistence presented in reg file

Below image is the vbs script obtained from the mentioned URL.

Press enter or click to view image in full size

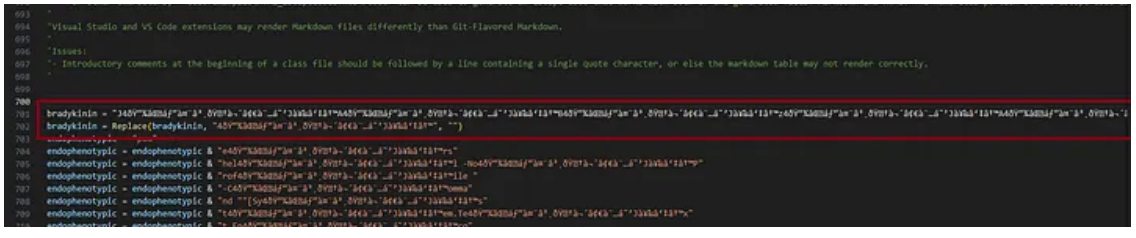


maze.vbs

Open Visual studio code

In the script we can see that there’s some obfuscated content and also found a replace

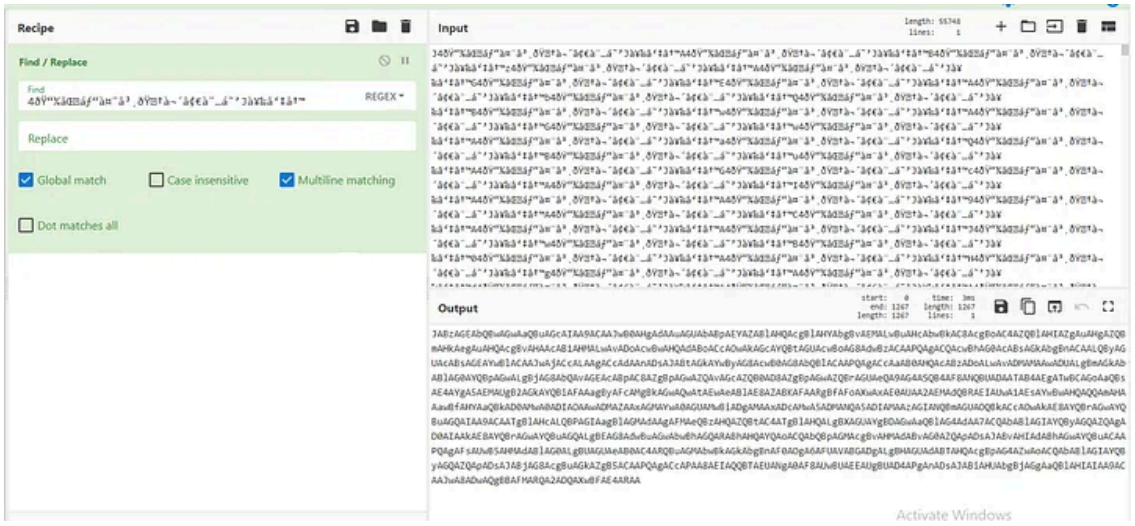
Press enter or click to view image in full size



Obfuscated String

Open CyberChef and paste the obfuscated “bradykinin” data into the Input field. In the Recipe section, add the **Find/Replace** operation, set the second parameter as **Find**, and the third parameter as **Null** in **Replace**. This will generate a Base64-encoded string.

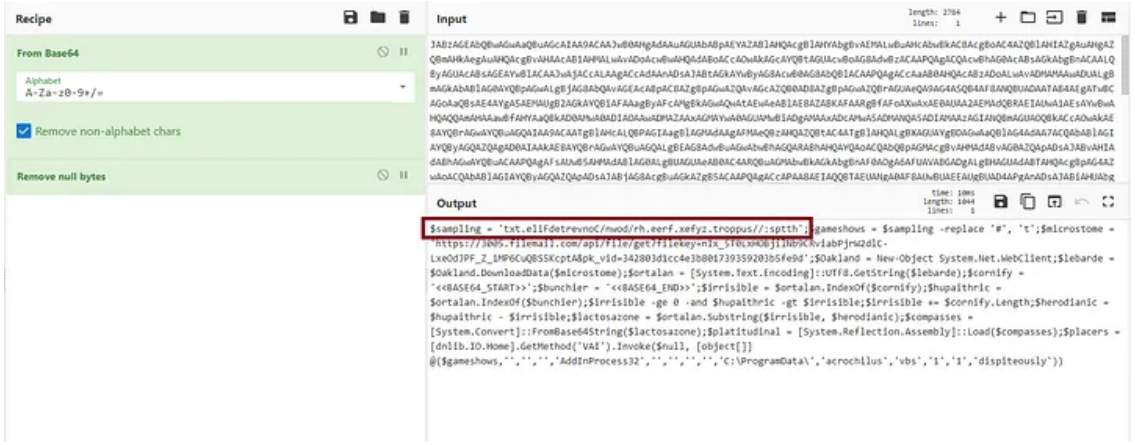
Press enter or click to view image in full size



Decode Obfuscated String

Take the Base64-encoded output from the previous step and enter it into the Input field. Then, apply the following recipe: **From Base64** and **Remove null bytes**. This will decrypt the string, revealing a **Reverse URL Construction**, which points to the URL: **“https[.]//support.zyfex[.]free[.]hr/down/ConvertedFile[.]txt”**.

Press enter or click to view image in full size



Reverse Loader

The downloaded payload is stored in **gameshow**, while **AddInProcess32** runs the add-in in an isolated 32-bit process, separate from the main application. The payload is saved in **C:\ProgramData\acrohilus**, with **vbs** indicating a **VBScript execution**. The values **‘1’, ‘1’** are unclear but may serve as execution flags or commands.

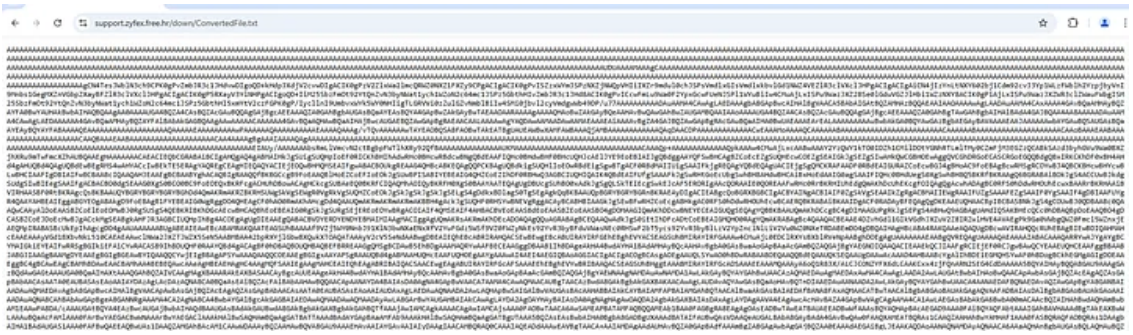
Get Sarviya’s stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

The website mentioned in the Ps script redirects to the below page where the ConvertedFile.Txt is present.

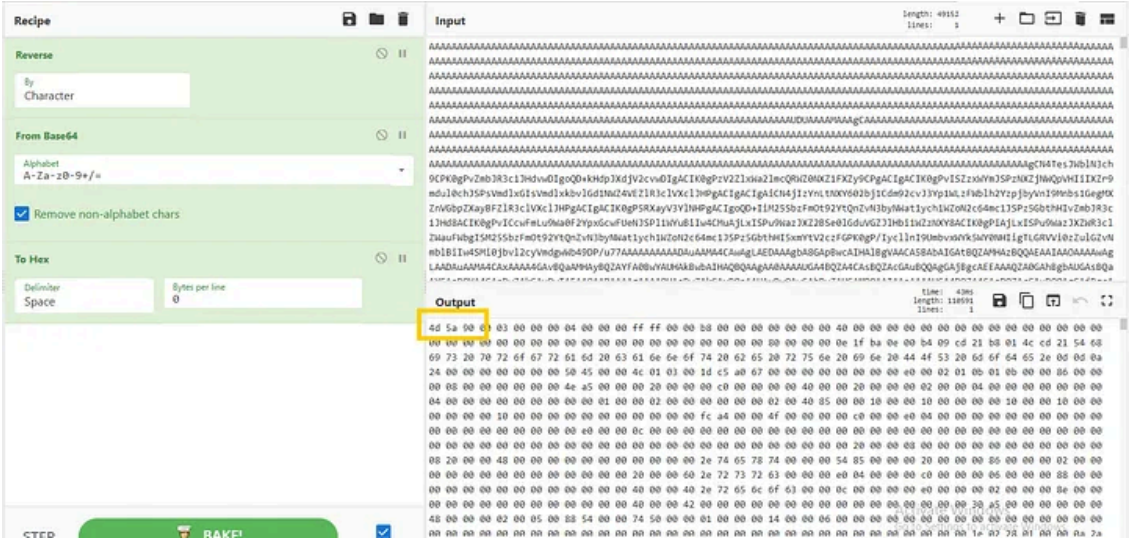
Press enter or click to view image in full size



Payload from Reverse Link

On looking at it , we can see that it's obfuscated. So we gonna try several options in Cyberchef. Since the URL is in reverse, this probably in reverse (Strikes in my mind) and I tried FromBase64 too. AND BANG!

Press enter or click to view image in full size



Decrypted MZ File

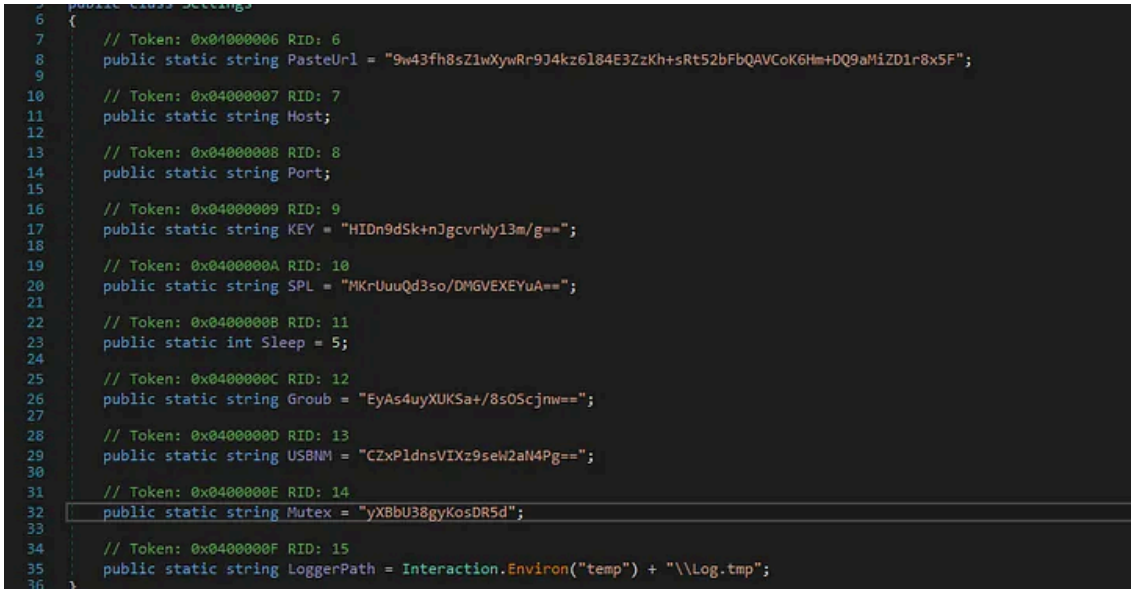
We got MZ header. So now its some executable. I'm gonna dump it and save it for further.

Xworm Execution:

Open the extracted PE file in **dnSpy**, then right-click and select **Go to Entry Point**. Initially, the **Main** function reveals an **AES decryption method**, suggesting the presence of a **hardcoded obfuscated string**. Set a **breakpoint** at **pasteurl**, then execute step by step. As you progress, the **decrypted string** will become visible in the **value section** of dnSpy.

Inside the key, an **encrypted string** is visible, along with two strings: **Host** and **Port**. These may indicate a **C2 (Command and Control) server**, which will be decrypted from the encoded string.

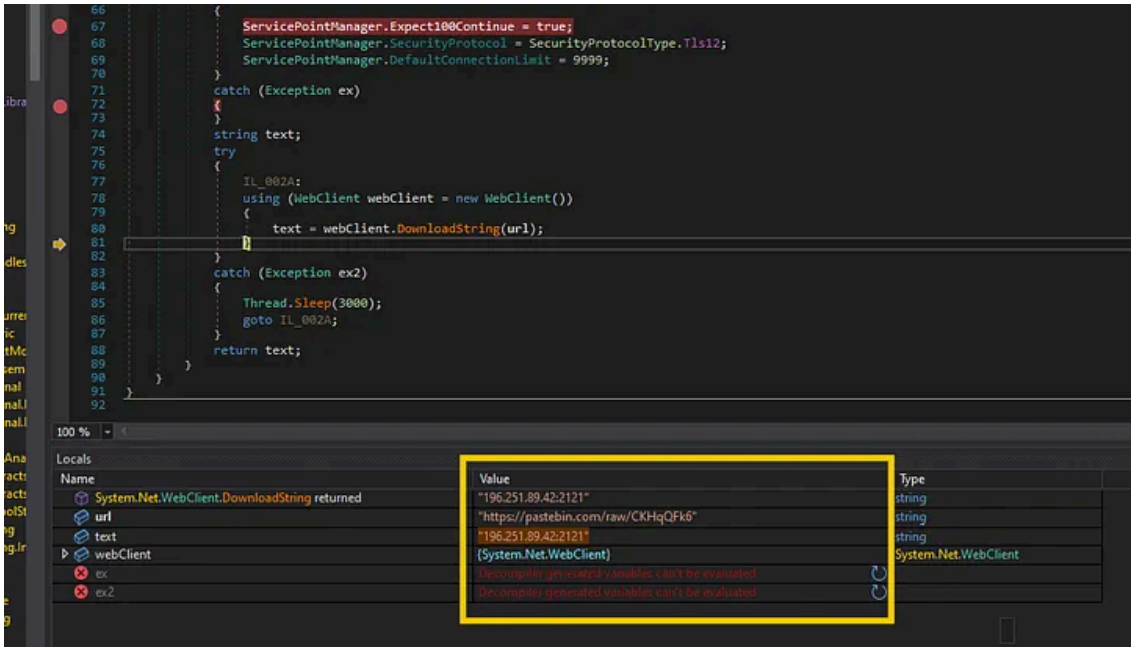
Press enter or click to view image in full size



Before decrypted String

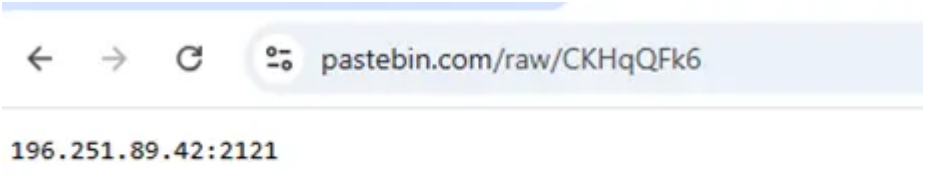
After execution of download string, we got the c2 Host and Port address.

Press enter or click to view image in full size



C2 Host and port

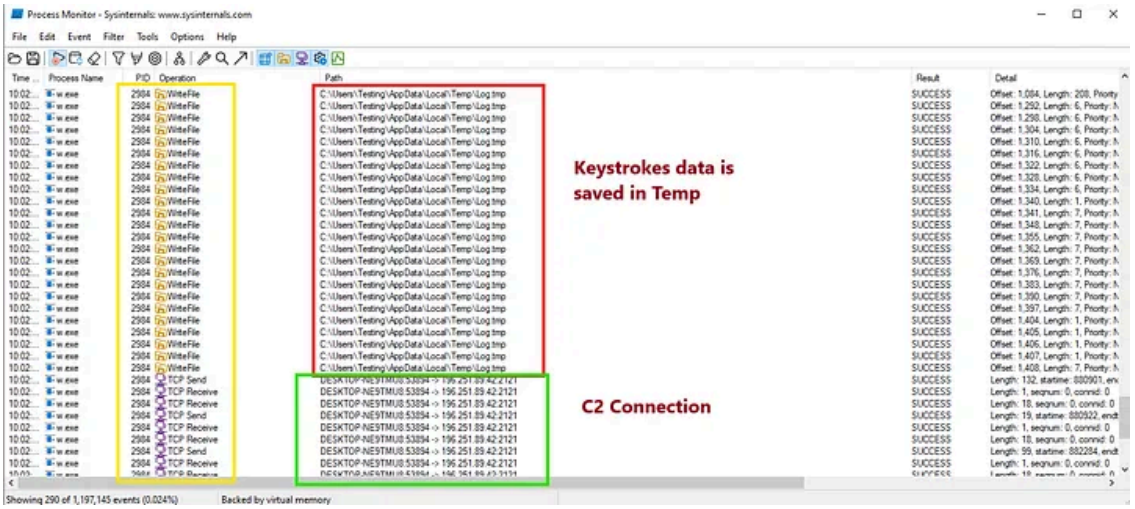
Enter the **decoded URL** into a browser, which will reveal the **IP address and port** of the **C2 (Command and Control) server**.



C2 Response

Now, execute the **extracted PE file**. In **ProcMon**, you can observe that it writes data to the **Temp folder** under the name **log.temp**. Additionally, network activity reveals that the file is **connecting to the C2 server**, establishing a **send and receive** communication channel.

Press enter or click to view image in full size



ProcMon- Xworm connecting to c2

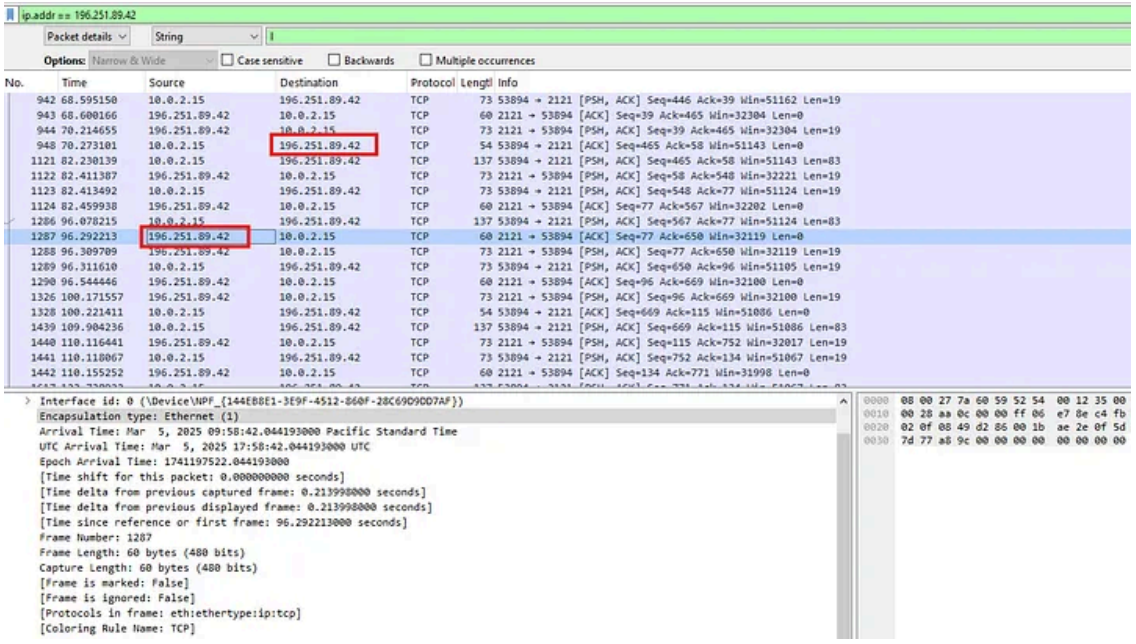
In the **Temp** folder, the file **log.tmp** stores recorded data, revealing that it is capturing keystrokes. This confirms that the malware functions as a **keylogger**, recording user input and potentially exfiltrating sensitive information.

Press enter or click to view image in full size

Keystrokes Stored in Temp Folder

In **Wireshark**, the captured network traffic shows data being **transferred to the C2 server**, with corresponding **ACK (Acknowledgment) packets**, confirming successful communication between the infected system and the attacker's server.

Press enter or click to view image in full size



Wireshark — C2

IoC: sha1

Maze.vbs :64F19C6E30548BC3880DD6B1B4D21D174D5C8EFF

Xworm:99C5F8B888CD29574173AE0F03F6AEEBAC3AB2E1

AnyRun:Analysis xworm (MD5: B3A89B0BF85BDA317F428C807637F9D5) Malicious activity — Interactive analysis ANY.RUN
