Earth Preta Updated Stealthy Strategies

By By: Vickie Su, Nick Dai, Sunny Lu Mar 23, 2023 Read time: 20 min (5505 words)

Published: 2023-03-23 · Archived: 2026-04-05 18:18:51 UTC

In our previous research, we disclosed and analyzed a new campaign initiated by the threat actor group Earth Preta (aka Mustang Panda). In a more recent campaign we've been tracking, we discovered Earth Preta delivering lure archives via spear-phishing emails and Google Drive links. After months of investigation, we found that several undisclosed malware and interesting tools used for exfiltration purposes were used in this campaign. We also observed that the threat actors were actively changing their tools, tactics, and procedures (TTPs) to bypass security solutions. In this blog entry, we will introduce and analyze the other tools and malware used by Earth Preta.

Infection chain

As we previously mentioned in our past blog entry, the entire attack begins with a spear-phishing email. After a long-term investigation into the attack routine, we've determined that the full infection chain works as follows:

We categorize the different TTPs into six stages: arrival vectors, discovery, privilege escalation, lateral movement, command and control (C&C) and exfiltration, respectively. In our previous research, we covered most of the new TTPs and malware during the first stage, arrival vectors. However, we observed that some of TTPs have been changed. In the following sections, we focus on the updated arrival vectors and their succeeding stages.

Arrival vectors

We previously summarized the arrival vectors used by Earth Preta by categorizing them into three types (DLL sideloading, shortcut links, and fake file extensions). Starting in October and November 2022, we observed that the threat actors began changing their TTPs to deploy the TONEINS, TONESHELL, and PUBLOAD malware. We believe that the threat actors are employing these new techniques to avoid detection.

Based on our earlier observation, the TONEINS and TONESHELL malware were downloaded from the Google Drive link embedded in the body of an email. To bypass email-scanning services and email gateway solutions, the Google Drive link has now been embedded in a lure document. The document lures users into downloading a malicious password-protected archive with the embedded link. The files can then be extracted inside via the password provided in the document. By using this technique, the malicious actor behind the attack can successfully bypass scanning services.

open on a new tab

Figure 2. A lure document (allegedly concerning the government-related Myanmar Sustainable Development Plan) embedded with a Google Drive link and a password

For the new arrival vector, the whole infection flow has been changed to the procedure shown in Figure 3.

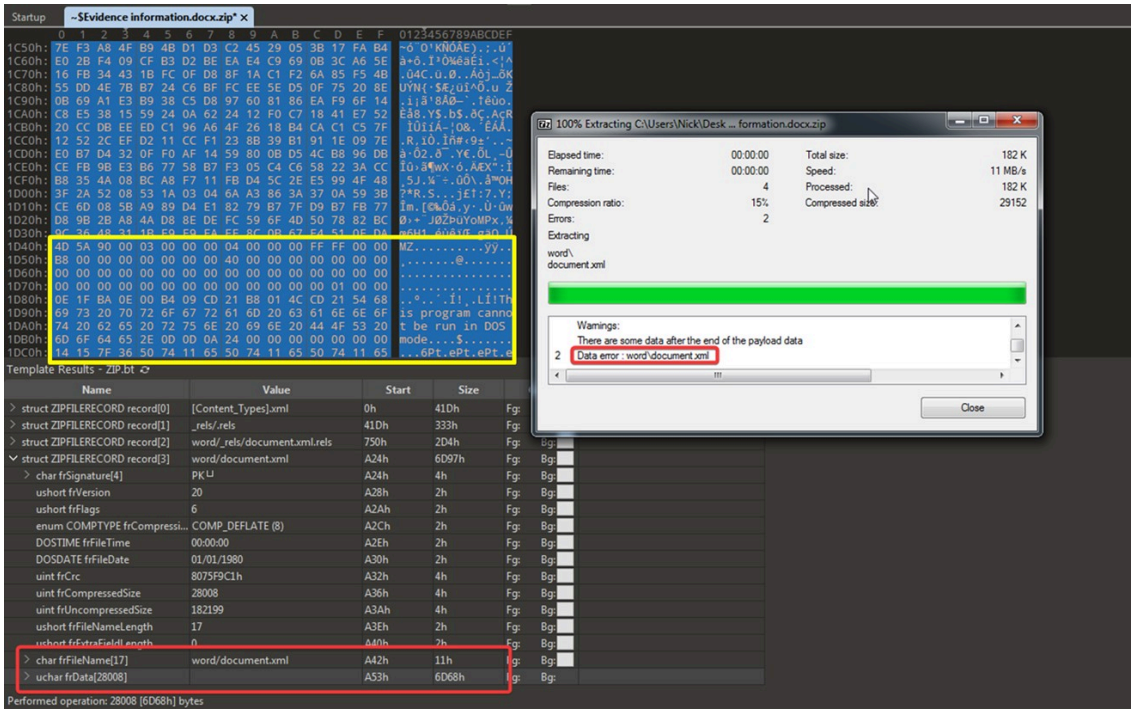
File name	Detection name	Description
Letter Head.docx		Decoy document with Google Drive link
List of terrorist personnel at the border.rar (all entries below are part of this archive)		
List of terrorist personnel at the border.exe		First-stage legitimate executable for DLL sideloading
libcef.dll	Trojan.Win32.TONEINS	First-stage malware
~$Evidence information.docx		Second-stage legitimate executable for DLL sideloading
~$List of terrorist personnel at the border.docx	Backdoor.Win32.TONESHELL	Second-stage malware

Internal Letter.docx		Decoy document
Letter Head.docx		Decoy document

Table 1. Files in the new arrival vector

After analyzing the downloaded archive, we discovered it to be a malicious RAR file with the TONEINS malware *libcef.dll* and the TONESHELL malware *~List of terrorist personnel at the border.docx*. The infection flow for these is similar to the arrival vector type C in our previous report, with the only difference being that the fake .docx files have XOR-encrypted content to prevent detection. For example, *~$Evidence information.docx* is a file disguising itself as an Office Open XML document. As such, it seems harmless and can even be opened by using decompression software such as 7-Zip.

We found that the threat actors have hidden a PE file in one of the archive's ZIPFILERECORD structures. The TONEINS malware, *libcef.dll*, will decrypt this file with a single byte in XOR operations, find the PE header, and drop the payload to the specified path.

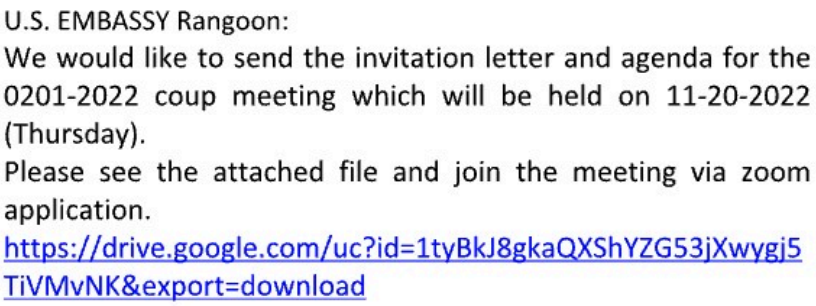


open on a new tab

Figure 4. A PE file is revealed after decrypting the frData member in the last ZIPFILECECORD structure.

The succeeding behaviors of the infection flow are generally the same as those in our previous analysis, where we provide more details.

In more recent cases, the malware PUBLOAD was also being delivered through Google Drive links embedded in decoy documents.



open on a new tab

Figure 6. The lure document Invitation letter from the US embassy.docx

Since October 2022, we have been observing a new variant of PUBLOAD, which uses the spoofed HTTP header to transfer the data, as LAC's report also discusses. In contrast to the previous PUBLOAD variant, it prepends an HTTP header with a legitimate-looking host name to the packets. We believe that the threat actors are trying to conceal malicious data among normal traffic. The data in the HTTP body is the same as the past variant, which has the same magic bytes *17 03 03* and the encrypted victim information. We were able to successfully retrieve the payload from a live C&C server and were therefore able to continue our analysis.

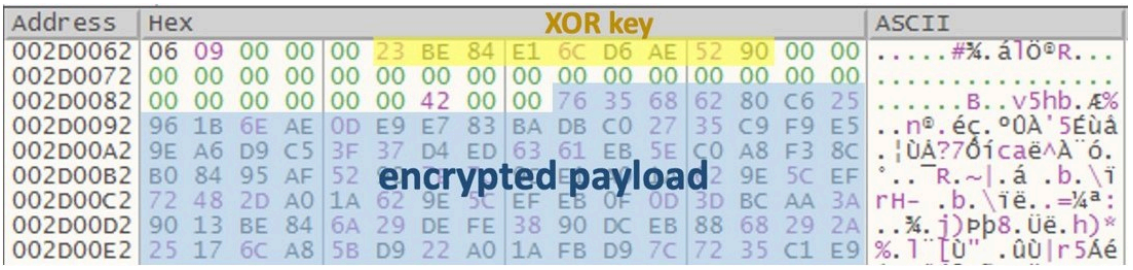
Once the payload is received, it will check if the first three magic bytes are *17 03 03* and if the following two bytes are the size of payload. It will then decrypt the encrypted payload with the predefined RC4 key *78 5A 12 4D 75 14 14 11 6C 02 71 15 5A 73 05 08 70 14 65 3B 64 42 22 23 20 00 00 00 00 00 00 00*, which is the same as the one used in the PUBLOAD loader.

Address	Hex	ASCII
00300000	17 03 03 42 29 2C D8 BC 35 0F A1 A6 5D 55 F8 2C	...B),Ø¼5.¡¦]Uø,
00300010	EA DE 16 28 27 CA A5 49 F0 D2 74 10 C1 A3 42 B5	êÞ.('Ê¥IðÒt.Á£Bµ
00300020	00 4A 6C 7C 83 0D 7D 60 A9 AC 65 5B 44 48 12 7C	.Jl|..}`©¬e[DH.|
00300030	26 16 B5 20 CC BC A0 2C D3 F4 2A ED 5E 4E 73 A2	&.µ Ì¼ ,Óô*í^Ns¢
00300040	E7 A8 7C 39 1D 64 9D 1C 6A CE C2 B5 8C 61 BB 59	ç¨|9.d..jÎÂµ.a»Y
00300050	95 A5 CD 5A 4F 9C DA 96 58 DB 52 1F A8 DC 78 8E	.¥ÍZO.Ú.XÛR.¨Üx.
00300060	2F 64 E4 1C FE 87 3C 1E 44 9E 83 CD 70 E0 3C F5	/dä.þ.<.D..Ípà<õ
00300070	5A A4 8A 2F 43 10 07 22 1A 76 36 07 F4 D8 65 44	Z¤./C..".v6.ôØeD
00300080	DE 26 B6 A3 80 [illegible] 71 26 2D 55	Þ&¶£.öÿ#..Åxq&-U
00300090	98 10 FB 5C 17 E3 5C 90 35 A0 57 4A 35 69 A9 52	..û\.ã\.5 WJ5i©R
003000A0	97 37 BF BB 18 4D BB F4 38 0B 7E 36 AE 28 62 C9	.7¿».M»ô8.~6®(bÉ
003000B0	04 13 DB 83 D6 9F 26 D0 96 A3 73 30 37 53 3C CE	..Û.Ö.&Ð.£s07S<Î
003000C0	40 B1 77 7B C8 81 AB 57 FB 6C BA 36 74 1A D6 E8	@±w{È.«WûlÞ6t.Öè
003000D0	0D 3B A4 14 61 0E 6C 84 41 AC 57 6E F6 39 7B A0	.;¤.a.l.A¬Wnö9{
003000E0	4A A5 12 2B 23 C6 7A 05 AF D6 1C 29 7E 50 94 5B	J¥.+#Æz.¯Ö.)~P.[
003000F0	D3 F4 DE DB 55 E8 67 4E FE 16 16 81 11 6C 79 56	ÓôÞÛUègNb....lyV

open on a new tab

Figure 8. The first payload retrieved from the PUBLOAD HTTP variant

After decryption, it then checks if the first byte of the decrypted payload is *0x06*. The decrypted payload contains another payload that is XOR-encrypted with the bytes *23 BE 84 E1 6C D6 AE 52 90*.



open on a new tab

Figure 9. The second payload retrieved from the PUBLOAD HTTP variant

After this is decrypted, there is yet another final backdoor payload that supports data upload and command execution.

Command	Internal string
0x03	-
0x01	-
0x1B	*UploadBegin error : %d!*
0x1D	*UploadData error : %d!*
0x1A	-
0x1E	*CmdStart error : %d!*
0x1F	*CmdWrite error : %d!*
0x30	*CmdWrite error : %d*
0x20	-

Table 2. Command codes in the PUBLOAD HTTP variant

In addition, we found some interesting debug strings and event names among the PUBLOAD samples.

In summary, we think that the new TONESHELL and PUBLOAD archives have been evolving and now have something in common. For example, both of them are now being placed in decoy documents (such as Google Drive links) in order to bypass antivirus scanning.

Discovery

Once the threat actors obtain access to the victim's environment, they can start inspecting the environment via the following commands:

net user

net user <username>

net user <username> /DOMAIN

Privilege escalation

In this campaign, we discovered several tools used for UAC bypass in Windows 10. We will go into detail for each of them.

HackTool.Win32.ABPASS is a tool used to bypass UAC in Windows 10. Based on our analysis, it reuses codes from the function ucmShellRegModMethod3, which is from a famous open-source project called UACME. A report from Sophosnews article introduces this tool. news article

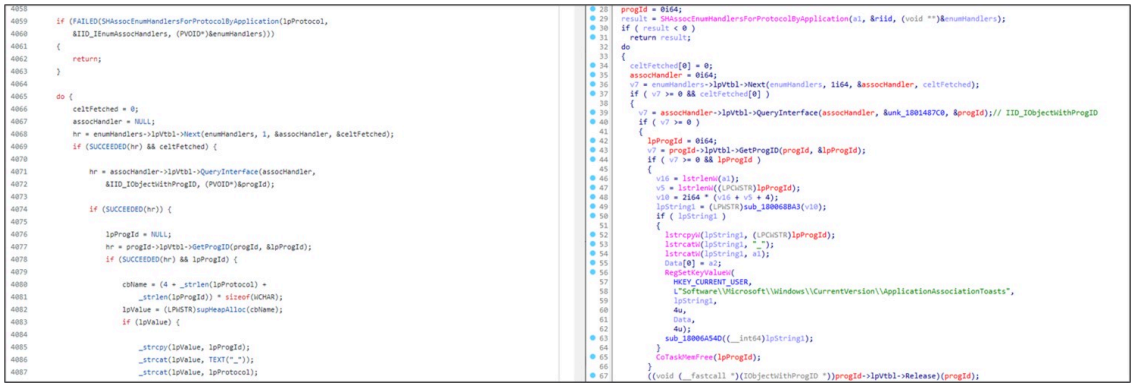
This tool accepts an argument, and the following data is written into registry:

Registry Key	Name	Value
HKEY_USERS\<SID>-1001_Classes\aaabbb32\shell\open\command	*(Default)*	*argv[1]*
HKEY_USERS\<SID>-1001_Classes\ms-settings\CurVer	*(Default)*	*aaabbb32*

Table 3. Registry keys changed by ABPASS

It also changes how Windows handles the *ms-settings* protocol — in this case, the string *ms-settings* is a Programmatic Identifier (ProgID). If the CurVer key is set under a ProgID, it will be used for versioning and mapping the current ProgID (*ms-settings*) to the one specified in the CurVer's default value. In turn, the behavior of *ms-settings* is redirected to the custom defined ProgID *aaabbb32*. It also sets up a new ProgID *aaabbb32* and its shell open command. Finally, *fodhelper.exe* or *computerDefaults.exe* will be executed to trigger the *ms-settings* protocol.

HackTool.Win32.CCPASS is another tool that is also used for Windows 10 UAC bypass and similarly reuses codes from the function ucmMsStoreProtocolMethod in the project UACME.



open on a new tab

Figure 14. Code similarities in CCPASS and ucmMsStoreProtocolMethod

It works in a similar way to ABPASS. However, unlike ABPASS, it hijacks the *ms-windows-store* protocol. The hack tool CCPASS works as follows:

1. It disables the application association toasts for the protocol *ms-windows-store*.
2. It creates a new Shell in the registry.
3. It invokes the undocumented API UserAssocSet to update the file association.
4. It executes *WSReset.exe* to trigger this protocol.

In Windows 10 and above, the system shows a new toast dialog for selecting the open application for the selected file type. To hide this window, the tool explicitly adds new entries to *HKCU\Software\Microsoft\Windows\CurrentVersion\ApplicationAssociationToasts* to disable all toasts related to the protocol *ms-windows-store*.

Once this is done, the tool starts to alter the shell command of *ms-windows-store* and finally triggers it using *WSReset.exe*.

In Windows 10, there is a native Windows service called “SilentCleanup.” This service has the highest privileges that can be abused for Windows 10 UAC bypass. Normally, this service is intended for running *%windir%\system32\cleanmgr.exe*. However, the environment variable *%windir%* can be hijacked and changed to any path to achieve privilege escalation.

We observed that the threat actors used this technique to execute *c:\users\public\1.exe*.

Lateral movement

In this stage, we observed certain malware such as HIUPAN and ACNSHELL (initially introduced and analyzed by Mandiant and Sophosnews article) being used to install themselves to removable disks and create a reverse shell.

We found a pair of malware comprised of a USB worm and a reverse shell —includin g a USB worm and a reverse shell (detected as Worm.Win32.HIUPAN and Backdoor.Win32.ACNSHELL, respectively,) — being used to spread themselves over removable drives.

Figure 18 shows the infection chain for both.

The *USB Driver.exe* program first sideloads *u2ec.dll*, which then loads the payload file *usb.ini*. They have the following PDB strings, respectively:

- *G:\project\APT\U盘劫持\new\u2ec\Release\u2ec.pdb*
- *G:\project\APT\U盘劫持\new\shellcode\Release\shellcode.pdb*

The string *U盘劫持* means "U disk hijacking," where "U disk" refers to removable drives.

USB Driver.exe then starts checking whether it is properly installed. If it is installed, it will start to infect more removable disks and copy files to a folder named *autorun.inf*. If it is not installed, it installs itself to *%programdata%* and then sets the registry run key for persistence.

Finally, the ACNSHELL malware *rzlog4cpp.dll* is sideloaded. It will then create a reverse shell via *ncat.exe* to the server *closed[.]theworkpc[.]com*.

Command and Control (C&C) stage

Earth Preta employed several tools and commands for the C&C stage. For example, the group used *certutil.exe* to download the legitimate WinRAR binary as *rar1.exe* from the server 103[.]159[.]132[.]91.

We also observed that the threat actors used PowerShell to download multiple malware and archives from the server 103[.]159[.]132[.]181 for future use.

In certain instances, they even leveraged the WinRAR binary installed on the victim hosts to decompress all the malware.

Although we found several logs involving multiple pieces of dropped malware, we only managed to retrieve a few of them. Among all our collected samples, we will introduce the most noteworthy ones.

The file name of the backdoor CLEXEC is *SensorAware.dll*. This is a simple backdoor that is capable of executing commands and clearing event logs.

The backdoor COOLCLIENT was first introduced in a report from Sophosnews article; the sample mentioned in the report was compiled in 2021. In our case, the COOLCLIENT sample we analyzed had a more recent compilation time in 2022, and while it provides the same functionalities, it has the added capability to open a decoy document (*work.pdf*) when the current process name has ".pdf" or ".jpg" file extensions. It contains less OutputDebugStrings calls. Meanwhile, *loader.ja* is used under two processes: One is under *googleupdate.exe*, which is used for the first sideloading. The second is under *winver.exe*, which is injected to conduct backdoor behaviors. Furthermore, COOLCLIENT applies obfuscation techniques that we discuss in later sections.

Figure 24 shows the whole execution flow of COOLCLIENT.

The arguments of COOLCLIENT provide the following capabilities:

install. There are several ways to install COOLCLIENT, detailed here:

1. It installs itself by creating an InstallSvc service called InstallSvc which will trigger "*googleupdate.exe work*"..
2. It sets up a run key for via the command *C:\ProgramData\GoogleUpdate\googleupdate.exe work* for persistence.

work. The malware will continue to read and decrypt *goopdate.ja* and inject it into *winver.exe* for the next-stage payload (COOLCLIENT), which contains malicious behaviors.

passuac. The malware will check if the process *avp.exe* exists. If *avp.exe* doesn't exist, UAC bypass will be executed via the CMSTPLUA COM interface. If *avp.exe* exists, UAC bypass will be executed via the AppInfo RPC service.

According to our analysis, it reads the encrypted configuration file *time.sig*. It is also able to communicate through different network protocols such as UDP (User Datagram Protocol) and TCP (Transmission Control Protocol). Based on some internal strings and the APIs used by Earth Preta, the functionalities of this backdoor can be inferred as follows:

- Send portmap
- Build connection
- Read file
- Delete file
- Keystrokes and windows monitoring

The backdoor TROCLIENT, which was also first disclosed in Sophos's report, is similar to COOLCLIENT. However, this backdoor has an anti-debugging technique, which will check if the running processes have the strings *dbg.exe* or *olly*.

Figure 28 shows the whole execution flow of TROCLIENT.

The arguments of TROCLIENT provide the following capabilities:

install. There are two waysto determine the method of installation for TROCLIENT, detailed here:

1. It installs itself by creating aservice called InstallSvc which will trigger "*C:\programdata\netsky\netsky.exe online*".
2. It sets up a run key for the command *C:\programdata\netsky\netsky.exe onlin*e for persistence.

online: It will read the next stage payloads, *free.plg* and *main.plg*, and inject them into *dllhost.exe.*

passuac: The malware will check if the process *avp.exe* exists. If it does not, UAC bypass is executed via the CMSTPLUA COM interface. If *avp.exe* exists, UAC bypass is executed via token manipulation.

This backdoor provides the following capabilities:

- Read file
- Delete file
- Monitor keystrokes and windows

There are several similarities and differences between COOLCLIENT and TROCLIENT, as Table 3 shows.

Argument/Behaviors	**COOLCLIENT**	**TROCLIENT**
install		
Creates a service named InstallSvc	✓	✓
Executes itself with passuac	✓	✓
Sets Run Key with “work/online”	✓	✓
passuac		
AppInfo RPC	✓	
CMSTPLUA COM	✓	✓
Token manipulation		✓
work/online		
Send portmap	✓	
Connect to C&C	✓	✓
File operations	✓	✓
Keylogging	✓	✓

Table 3. Comparison of COOLCLIENT and TROCLIENT

In addition to the aforementioned malware, we also found several shellcode loaders for PlugX. Since it is a known malware family, we will not expand on its details in this blog entry.

Exfiltration

Based on our telemetry, we found that Earth Preta used multiple approaches to exfiltrate sensitive data from the victims. For example, in some cases, we observed that WinRAR and curl (or cURL) were leveraged to collect and transfer data to the threat actor’s server. After further investigation, we even found some previously unseen pieces of malware that were used to collect data in a custom-made file format. In the following sections, we share the details of the unique exfiltration toolsets developed by Earth Preta.

According to some of our monitoring logs, the threat actors abused the installed WinRAR binary and the uploaded curl executable to exfiltrate the files (Figure 30 shows the executed command). Note that the executable *log.log* is a legitimate curl binary. All the exfiltrated data was collected and sent back to the threat actor-controlled FTP (File Transfer Protocol) server.

In some cases, we accidentally stumbled on the account and password of the FTP server. Upon checking the FTP server, we learned that the threat actors focused on sensitive and confidential documents, most of which were compressed and protected with a password. Based on our observations, the documents were organized via the categorization of the victim's host name and disk drive.

Apart from well-known legitimate tools, the threat actors also crafted highly customized tools used for exfiltration. We named this malware "NUPAKAGE," a name derived from its unique PDB string, *D:\Project\NEW_PACKAGE_FILE\Release\NEW_PACKAGE_FILE.pdb*.

The NUPAKAGE malware needs a unique passcode to be executed, with the exfiltrated data being wrapped in a custom file format. It seems that the threat actors are continuously updating this tool to provide more flexibility and lower the possibility of detection, including adding more command-line arguments and obfuscation mechanisms. By default, it only collects documents, including the files with the following extensions:

- .doc
- .docx
- .xls
- .xlsx
- .ppt
- .pptx
- .pdf

It avoids collecting documents with file names starting with "$" or "~" since these types of documents are usually either temporary files generated by the system or PE files pretending to be decoy documents (as we discussed in the arrival vectors section).

The usage of this tool is as follows:

malware.exe ***passcode start end chunk -s extension_A extension_B …***

Argument Name	**Format**	**Example Value**	**Description**
passcode	String	comeon	A unique code to execute it
start	String	2022-01-01	The start range of the exfiltrated file's modification timestamp
end	String	2022-12-31	The end range of the exfiltrated file's modification timestamp
chunk	Integer	4096	Splits the generated data in chunks by the specified size (MB)
-s	String		File extensions to be collected; optional

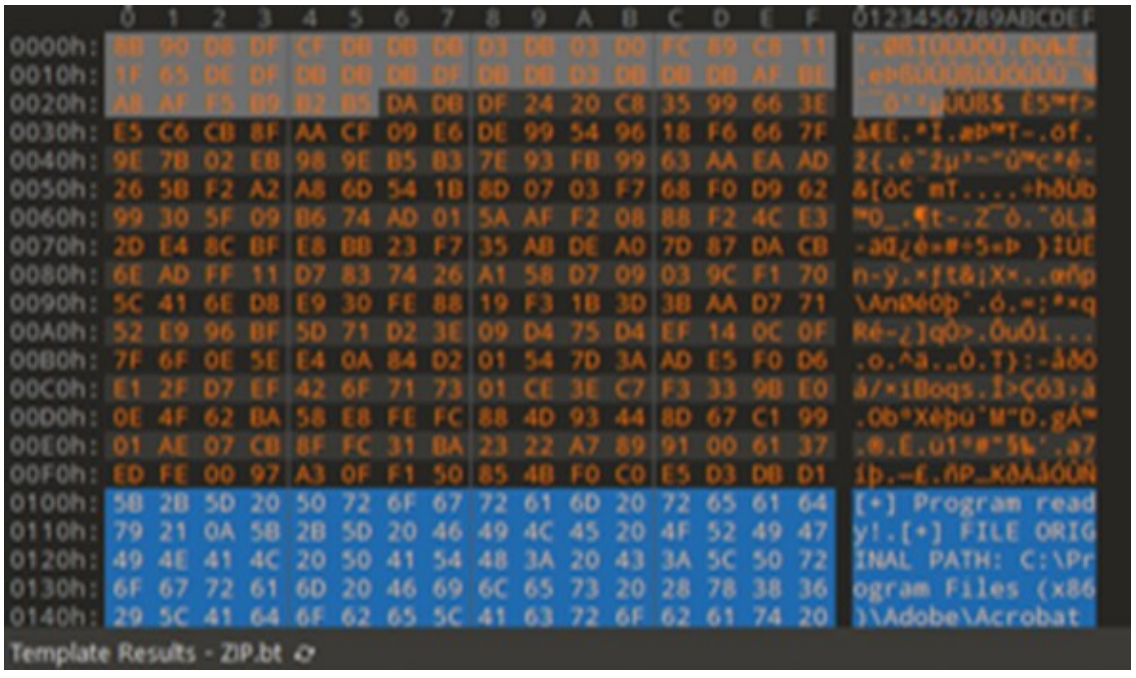
Table 5. Arguments of the NUPAKAGE malware

Every NUPAKAGE malware needs a unique passcode as its first argument to continue execution. As Figure 32 shows, it first checks if the passcode exists. If not, the malware execution procedure will terminate. In our collection, we observed different passcodes in each malware.

SHA256	Passcode
634977a24e8fb2e3e82a0cddfe8d007375d387415eb131cce74ca03e0e93565f	notebook
c835577f1ddf66a957dd0f92599f45cb67e7f3ea4e073a98df962fc3d9a3fbe0	comeon
2937580b16e70f82e27cfbc3524c2661340b8814794cc15cb0d534f5312db0e0	update
c2f5a12ebaeb39d4861e4c3b35253e68e6d5dc78f8598d74bc85db21aeb504e8	comeon

Table 4. Passcodes in NUPAKAGE

After execution, NUPAKAGE will drop two files, *xxx.zip* and *xxx.z*. The file *xxx.zip* is a logging file with a fake ZIP header prepended at offset 0x0 and taking up the first 0x100 bytes. Starting from the offset 0x100, the logging strings are encrypted with a single byte in XOR operations as shown Figure 33.



open on a new tab

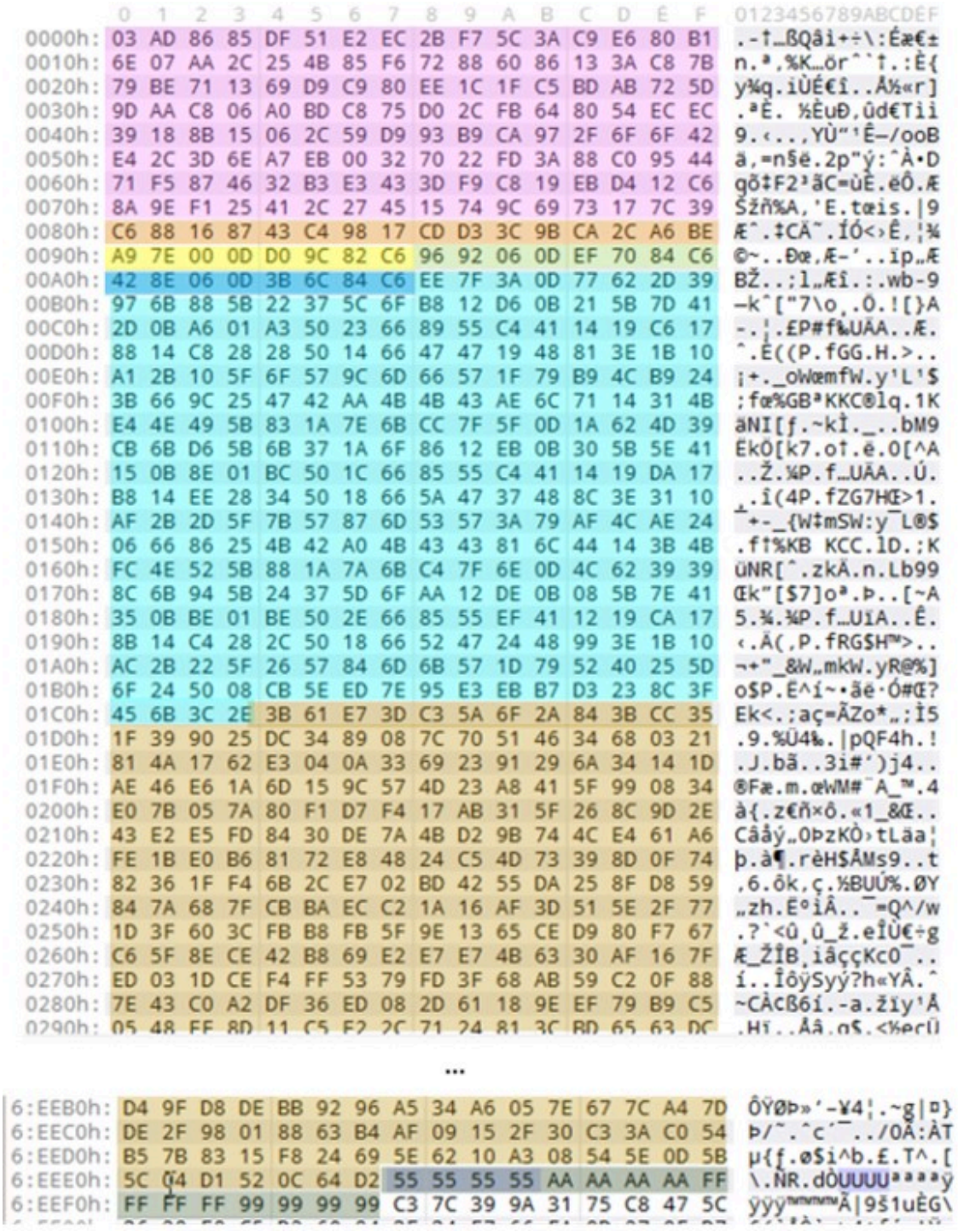
Figure 33. The original logging file (top), with plain text revealed in the decrypted logging file (bottom)

Taking one of the execution results as an example, much of the information of the exfiltrated data is saved, including the original file path, the original file size, and the compressed file size. We believe that the threat actors use it to further track which files have been processed. For security researchers, this logging file also helps reveal how much data is exfiltrated and provides information on the impact scope.

[+] Program ready!
[+] FILE ORIGINAL PATH: C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\1494870C-9912-C184-4CC9-B401-A53F4D8DE290.pdf
[+] FILE PATH SIZE: 198
[+] FILE ORIGINAL SIZE: 186837
[+] FILE COMPRESSED SIZE: 183734
[+] FILE ORIGINAL PATH: C:\Program Files (x86)\Adobe\Acrobat Reader DC\Reader\Click on ‘Change’ to select default PDF handler.pdf
[+] FILE PATH SIZE: 210
[+] FILE ORIGINAL SIZE: 186837
[+] FILE COMPRESSED SIZE: 183734
...
<omitted>
...
[*] File or folder access denied!
[*] File or folder access denied!
[+] All completed!

The file with a .z extension is a blob of exfiltrated data within a self-defined file format. The NUPAKAGE malware first generates a key blob randomly, with the key being encrypted in a custom algorithm. After, it stores the encrypted key blob into the first 0x80 bytes of the file with the .z extension. Starting from the offset 0x80, there exists a long array of all the exfiltrated data.

Much of the information from the exfiltrated files are saved, such as the MD5 hash, the length of the file name, the compressed file size, the original file size, the file name, and the file’s content. To separate the file blobs, it puts a unique byte sequence at the end of each, *55 55 55 55 AA AA AA AA FF FF FF FF 99 99 99 99*.



[open on a new tab]

Figure 34. Self-defined format in the file with the .z extension generated by NUPAKAGE

Offset	Field Name	Size	Description
0x0	key	0x80	Encrypted Key
0x80	md5	0x10	MD5 (XORed with Decrypted Key)
0x90	len	0x8	The length of file name (XORed with Decrypted Key)
0x98	size2	0x8	Comepressed file size (XORed with Decrypted Key)
0xA0	size1	0x8	Original file size (XORed with Decrypted Key)
0xA8	file_name	len	File name (XORed with Decrypted KEy)
0xA8 + len	file_content	size2	File content (XORed with Decrypted Key)
...	...	...	...
0xA8 + len + size2	delimiter	0x10	The mark to tell the end of one file object 55 55 55 55 AA AA AA AA FF FF FF FF 99 99 99 99

open on a new tab

Table 5. Self-defined format description in the file with the .z extension generated by NUPAKAGE

It's also worth mentioning that in the more recent versions of NUPAKAGE, an increasing number of obfuscations are being adopted to thwart static analysis.

ZPAKAGE is another example of custom malware used for packing files; it also works similarly to NUPAKAGE. It also needs a passcode to ensure that it is being used as intended. In the example shown in Figure 36, the passcode is "start".

ZPAKAGE also supports command-line arguments, but it possesses less functions than NUPAKAGE. The usage of this tool is shown as follows:

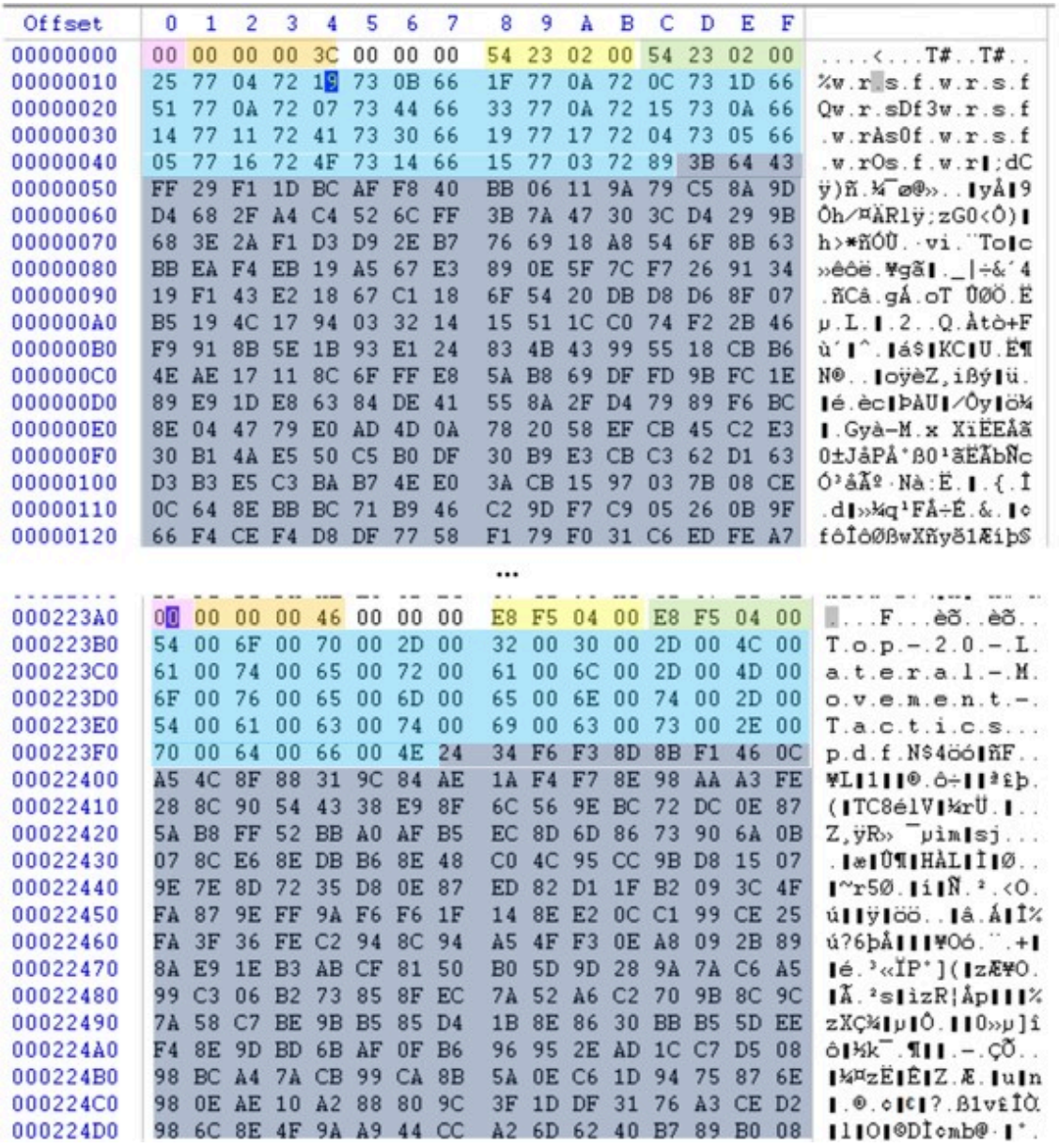
malware.exe ***passcode time***

Argument Name	Format	Example Value	Description
Passcode	String	start	A unique code in order to execute it
Time	String	20221221	The start date

Table 6. Arguments supported by ZPAKAGE

ZPAKAGE also shows similar behaviors to NUPAKAGE. For instance, it also avoids files with names starting with “$” or “~”. In addition, it generates two files, one with a .z extension and another with a .zip extension. The file with a .z extension is the exfiltrated data blob and the file with a .zip extension is the logging file.

In the generated file with a .z extension, the exfiltrated files will be compressed by the zlib algorithm to minimize the file size. It also defines a Boolean field “type” for storage, whether a file is compressed or not. If a file is compressed and its file size is less than the original one, the type will be 1. Otherwise, the type will be set to 0, and the original file content will be chosen instead of the compressed one. Regardless of whether the file content is compressed or not, it will be encrypted in XOR operations with a specific string, *qwerasdf*.



[open on a new tab]

Figure 37. Self-defined format in the file with .z extension generated by ZPAKAGE

Offset	Field Name	Size	Description
0x0	type	1	Compression type, 0x0 or 0x1
0x1	len	4	Length of filename
0x5	reserved	3	
0x8	size1	4	Original file size
0xC	size2	4	Compressed file size
0x10	file_name	len	Encoded filename (XOR with “qwerasdf”)
0x10 + len	file_content	size2	Encoded file content (zlib + XOR with “qwerasdf”)
…		…	…

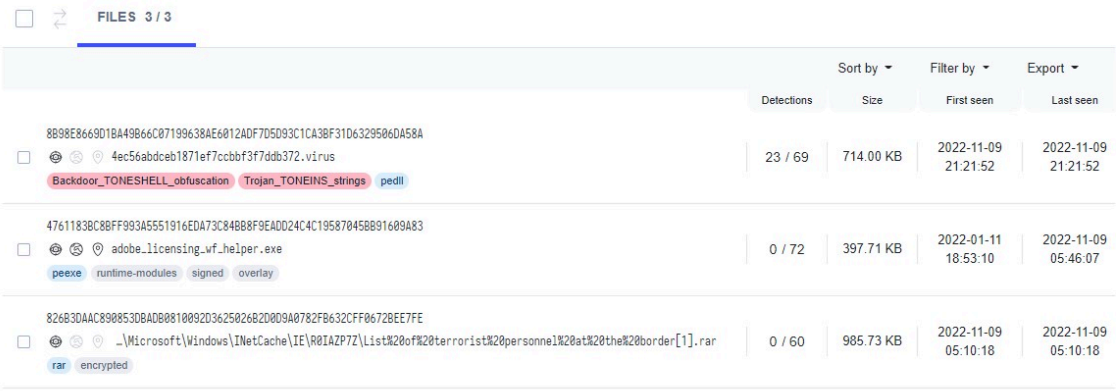
open on a new tab

Table 7. Self-defined format description in the file with the .z extension generated by ZPAKAGE

Threat hunting

Since October 2022, the threat actors have changed their TTPs and have started using password-protected archives. For example, we found a TONEINS sample (SHA256: *8b98e8669d1ba49b66c07199638ae6012adf7d5d93c1ca3bf31d6329506da58a*) on VirusTotal that can’t be linked to any other file in the “Relations” tab. However, we observed two files that have been opened in the “Behaviors” tab with the file names *~$Evidence information.docx* and *~$List of terrorist personnel at the border.docx*. As mentioned in the arrival vectors section, the next stage payloads are normally embedded in the fake document files.

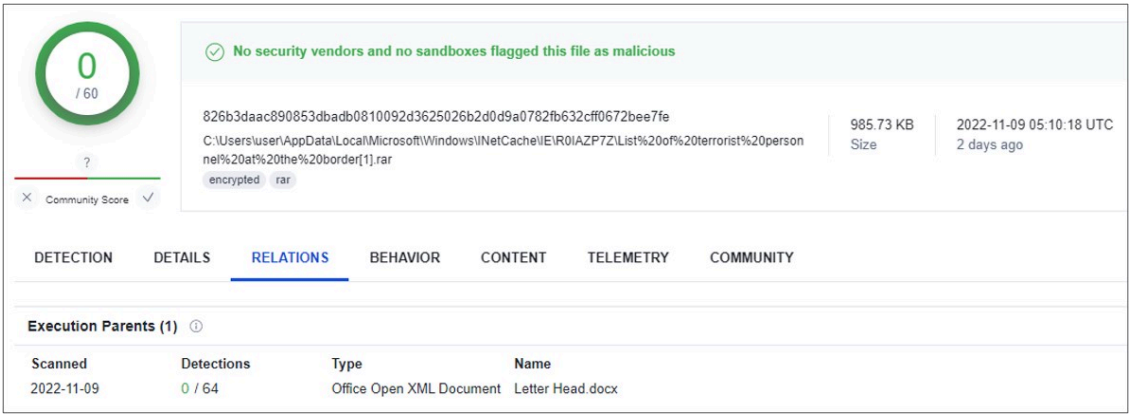
Figure 39 shows the search results for the query “List of terrorist personnel at the border” on VirusTotal. The first file is the TONEINS DLL sample that we mentioned earlier in this section, while the second file is a benign executable file originally named *adobe_licensing_wf_helper.exe*, which was apparently uploaded to VirusTotal with the file name *List of terrorist personnel at the border.exe*.



open on a new tab

Figure 39. Search result for the string List of terrorist personnel at the border on VirusTotal

The third file is a password-protected archive, which has the exact same file name, *List of terrorist personnel at the border[1].rar*. Unfortunately, we didn't have the password, so we were unable to decompress it. But it has an interesting execution parent in the "Relations" tab, which is a document file named *Letter Head.docx*.



open on a new tab

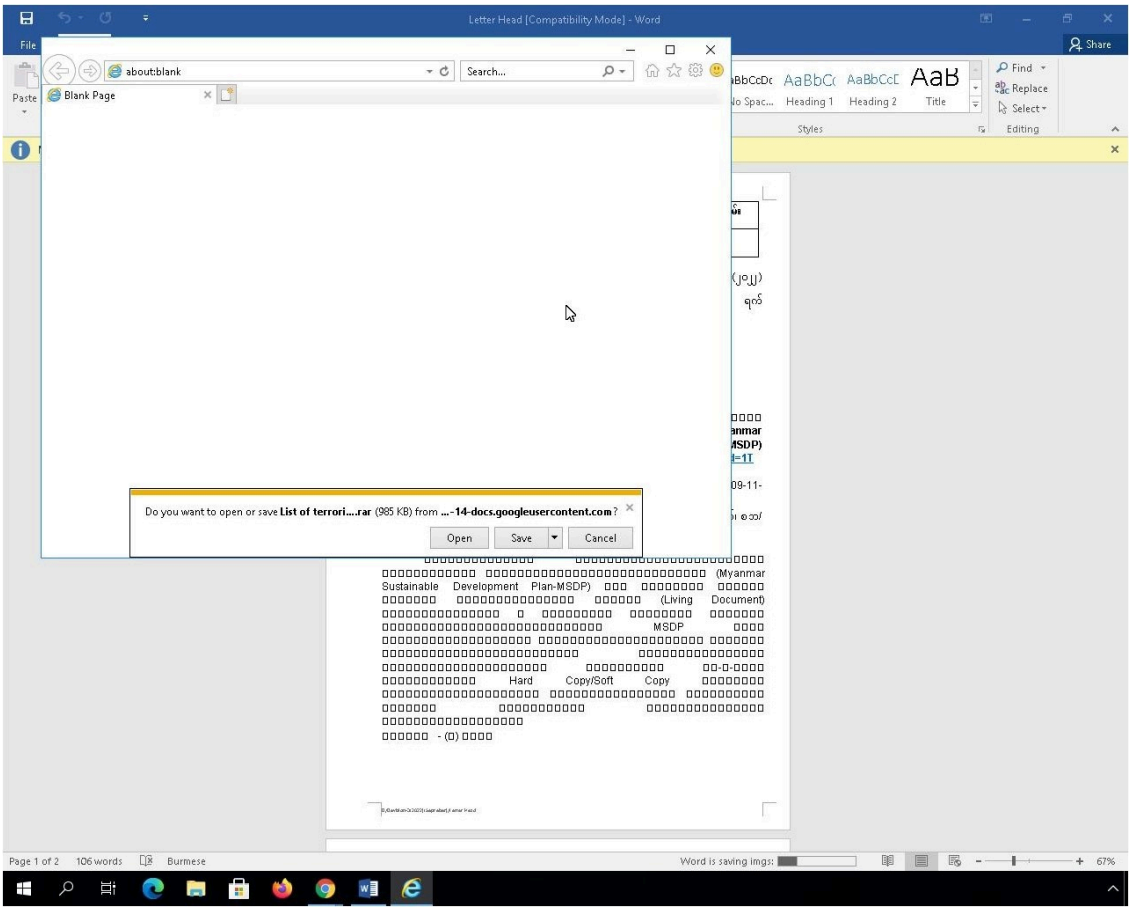
Figure 41. Execution parent of List of terrorist personnel at the border[1].rar

Inside the document *Letter Head.docx*, there is a Google Drive link and a password. The content itself is related to the Government of the Republic of the Union of Myanmar, and is written in Burmese.

Upon checking the download link, we discovered that it was the same password-protected archive file that we found on VirusTotal earlier.

The new arrival vector flow is similar to the one we introduced in the arrival vector section: Victims will receive and interact with a decoy document containing a Google Drive link and a corresponding password instead of an archive download link embedded in the email.

As for why the password-protected archive has the execution parent, upon checking the sandbox execution behaviors of *Letter Head.docx* on VirusTotal, we discovered that the VirusTotal sandbox will select any link embedded in the document. This leads to the opening of an Internet Explorer window with the file download prompt.



open on a new tab

Figure 44. Sandbox screenshot of the file Letter Head.docx on VirusTotal

When the download prompt is shown, Internet Explorer will silently download this file in the background even before the user selects the “Save” button.

As a result, the file will be saved to the cache folder named “INetCache,” after which we see a dropped RAR file:

- *C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\IE\R0IAZP7Z\List%20of%20terrorist%20personnel%20at%20the%20border[1].rar*.

Since the RAR file is downloaded automatically by Internet Explorer, *Letter Head.docx* will be treated as its execution parent. This sample can then be used for hunting this campaign.

To find additional password-protected archives and documents embedded with a Google Drive link, we tried to use the following query:

tag:rar tag:encrypted name:INetCache size:500kb+

The query finds any encrypted RAR archive with a large enough file size containing the folder name “INetCache” in its path. Fortunately, we found another RAR file with the document execution parent “*Notic(20221010)(final).docx”* that turned out to be a TONESHELL archive.

It's interesting to note that the threat actors use date and time strings written in the same format (DD-MM-YYYY) as the extracting passwords in all the cases we've collected so far.

Connecting the dots

During our investigation, we observed some data points that connect to the same personnel. For example, we found a specific name "TaoZongjie" among the different malware samples we collected. In addition, the GitHub repository named "YanNaingOo0072022," mentioned in Avast's December 2022 report, hosted multiple pieces of malware, including TONESHELL. We also observed that the obfuscation methods have similarities among the different malwares.

We found some samples sharing the same special string/name "TaoZongjie," including the Cobalt Strike malware, a Windows user on a TONESHELL C&C server, and the displayed message in the pop-up dialog box of TONESHELL.

Our investigation started with the TONESHELL C&C server 38[.]54[.]33[.]228 that had the remote desktop service enabled. Here, we found that one of the Windows users was called "TaoZongjie."

While hunting samples related to this campaign, we came across a tweet about Cobalt Strike posted in April 2021. At first glance, Cobalt Strike was used in a manner similar to this campaign, including the use of DLL sideloading, the use of a Google Drive link for delivery, and the creation of a schedule task.

The infection flow is as follows: The archive file is delivered through a Google Drive link, which contains a legitimate EXE file, a malicious DLL file, and a decoy document written in Burmese. Once the malicious DLL is sideloaded, it will drop the legitimate EXE file and the malicious DLL file, which are embedded in the resource section of the DLL file. In this sample, the string *By:Taozongjie* is being used as the event name.

In one TONEINS sample (SHA256: *7436f75911561434153d899100916d3888500b1737ca6036e41e0f65a8a68707*), we also observed the string *taozongjie*, which was being used for an event name.

In another TONESHELL sample (SHA256: *d950d7d9402dcf014d6e77d30ddd81f994b70f7b0c6931ff1e705abe122a481a*), there are some insignificant export functions, which will appear via message boxes, with the strings *Tao* or *zhang!*. Even though the names of these two strings are not spelled exactly same way as *taozongjie*, their spellings are still similar.

Based on what we found among the different samples, we assume that *taozongjie* could be one of the flags used by the threat actors.

The GitHub user "YanNaingOo0072022" was mentioned in both an Avast and an ESET report. The user's repositories host various malware, including the latest versions of TONEINS, TONESHELL, and a new tool, QMAGENT, which is ESET named MQsTTang". At the time of writing, this GitHub space was still accessible, with five repositories: "View2015," "View2016," "1226," "ee," and "14." Among these, "View2015" and "View2016" were empty.

The archive files in this repository are all the same but have different file names. We believe that these files were meant for different victims.

Upon unarchiving the compressed file, we found two files with the fake extension ".doc" containing one-byte XOR encrypted sections. Both share the same file structure (a PE payload hidden in a DOCX file) as the one we referred to in the arrival vectors section. These files ended up being the TONEINS and TONESHELL malware.

The file *Documents members of delegation diplomatic from Germany.Exe*, found in the *Documents.rar* archive, is a novel malware that communicates over the MQTT protocol. In March 2023, ESET published a detailed technical report on this backdoor, which it named "MQsTTang."

Beginning in January, we discovered that MQsTTang was being used as the new arrival vector in some of incidents we encountered, specifically in campaigns targeting individuals involved with government entities. This backdoor is unique because it communicates to its C&C servers over the MQTT protocolnews article, which is commonly used in internet-of-things (IoT) devices. Malicious actors using this technique can effectively hide the real C&C server behind the protocol.

The file *CVs Amb Office PASSPORT Ministry Of Foreign Affairs.exe*, which is the malware QMAGENT, can be found in the *CVs Amb.rar* archive.

Conclusion

Over the past year, security researchers have been discovering and analyzing Earth Preta's campaigns and toolsets. We were able to attribute some of these to Earth Preta based on similarities among the TTPs, the malware being used, and the timeline of the campaigns. Starting October 2022, the threat actors changed the arrival vector of the TONEINS, TONESHELL, and PUBLOAD malware. Instead of attaching malicious archives or Google Drive links to an email, they now embed the download link in another decoy document and add a password to the archive.

Based on our observations, Earth Preta tends to hide malicious payloads in fake files, disguising them as legitimate ones — a technique that has been proven effective for avoiding detection. As for privilege escalation, the threat actors tend to reuse codes copied from open-source repositories. Meanwhile, they developed customized toolsets designed to collect confidential documents in the exfiltration stage.

Overall, we believe that Earth Preta is a capable and organized threat actor that is continuously honing its TTPs, strengthening its development capabilities, and building a versatile arsenal of tools and malware.

To help prevent potential threats such as the one posed by advanced persistent threat (APT) groups, we suggest that organizations conduct phishing awareness training for their employees and partners to stress the importance of caution when opening emails, particularly those messages from unfamiliar senders or with unknown subjects.

To assist organizations in protecting themselves against sophisticated threats, we recommend adopting a comprehensive security strategy that employs advanced technologies capable of identifying and halting such threats across multiple channels, including endpointsproducts, serversproducts, networksproducts, and email communicationsproducts.

Indicators of Compromise (IOCs)

The full list of IOCs can be found here.

MITRE ATT&CK

Tactic	ID	Name
Resource Development	T1583.004	Acquire Infrastructure: Server
	T1587.001	Develop Capabilities: Malware
	T1585.002	Establish Accounts: Email Accounts
	T1588.002	Obtain Capabilities: Tool
	T1608.001	Stage Capabilities: Upload Malware
Initial Access	T1566.002	Phishing: Spearphishing Link
Execution	T1204.001	User Execution: Malicious Link
	T1204.002	User Execution: Malicious File
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder
	T1574.002	Hijack Execution Flow: DLL Side-Loading
	T1053.005	Scheduled Task/Job: Scheduled Task

Privilege Escalation	T1068	Exploitation for Privilege Escalation
	T1134	Access Token Manipulation
Defense Evasion	T1140	Deobfuscate/Decode Files or Information
	T1036.005	Masquerading: Match Legitimate Name or Location
Lateral Movement	T1091	Replication Through Removable Media
Command and Control	T1071.001	Application Layer Protocol: Web Protocols
	T1573.001	Encrypted Channel: Symmetric Cryptography
	T1104	Multi-Stage Channels
	T1095	Non-Application Layer Protocol
Exfiltration	T1048	Exfiltration Over Alternative Protocol
