

Chinese APT Campaign Targets Entities with Updated FDMTP Backdoor

 darktrace.com/blog/chinese-apt-campaign-targets-entities-with-updated-fdmtp-backdoor

Tara Gould

May 14, 2026



May 14, 2026

Darktrace researchers identified a Twill Typhoon–linked China-nexus campaign targeting APJ customers. The activity observed includes CDN impersonation, legitimate binaries, and DLL sideloading to deploy a modular .NET RAT.

Darktrace have identified activity consistent with Chinese-nexus operations, a Twill Typhoon-linked campaign targeting customer environments, primarily within the Asia-Pacific & Japan (APJ) region

Beginning in late September 2025, multiple affected hosts were observed making requests to domains impersonating content delivery networks (CDNs), including infrastructure masquerading as Yahoo- and Apple-affiliated services. Across these cases, Darktrace identified a consistent behavioral execution pattern: the retrieval of legitimate binaries

alongside malicious Dynamic Link Libraries (DLLs), enabling sideloading and execution of a modular .NET-based Remote Access Trojan (RAT) framework.

The activity aligns with patterns described in Darktrace's previous Chinese-nexus operations report, Crimson Echo. In this case, observed modular intrusion chains built on legitimate software, and staged payload delivery. Threat actors retrieve legitimate binaries alongside configuration files and malicious DLLs to enable sideloading of a .NET-based RAT.

Observed Campaign

Across cases, the same ordered sequence appears: retrieval of a legitimate executable, (2) retrieval of a matching .config file, (3) retrieval of the malicious

DLL, (4) repeated DLL downloads over time, and (5) command-and-control (C2) communication. The .config file retrieves a malicious binary, while the legitimate binary provides a legitimate process to run it in.

Darktrace assesses with moderate confidence that this activity aligns with publicly reported Twill Typhoon tradecraft. The observed use of FDMTP, DLL sideloading, and overlapping infrastructure is consistent with previously observed operations, though not unique to a single actor. While initial access was not directly observed, previous Twill Typhoon campaigns have typically involved spear-phishing.

What Darktrace Observed

Since late September 2025, Darktrace has observed multiple customer environments making HTTP GET requests to infrastructure presenting as "CDN" endpoints for well-known platforms (including Yahoo and Apple lookalikes). Across cases, the affected hosts retrieved legitimate executables, then matching .config files (same base filename), then DLLs intended for sideloading. The sequencing of a legitimate binary + configuration + DLL has been previously observed in campaigns linked to China-nexus threat actors.

In several cases, affected hosts also issued outbound requests to a /GetCluster endpoint, including the protocol=Dotnet-Tcpdmtip parameter. This activity was repeatedly followed by retrieval of DLL content that was subsequently used for search-order hijacking within legitimate processes.

In the September–October 2025 cases, Darktrace alerting commonly surfaced early-stage registration and C2 setup behaviors, followed by retrieval of a DLL (e.g., Client.dll) from the same external host, sometimes repeatedly over multiple days, consistent with establishing and maintaining the execution chain.

In April 2026, a finance-sector endpoint initiated a series of GET requests to yahoo-cdn[.]it[.]com, first fetching legitimate binaries (including vshost.exe and dfsvc.exe), then repeatedly retrieving associated configuration and DLL components (including dfsvc.exe.config and dnscfg.dll) over an 11-day window. The use of both Visual Studio hosting and OneClick (dfsvc.exe) paths are used to ensure the malware can run in the targeted environment.

Technical Analysis

Initial staging and execution

While the initial access method is unknown, Darktrace security researchers identified multiple archives containing the malware.

A representative example includes a ZIP archive (“test.zip”) containing:

- A legitimate executable: biz_render.exe (Sogou Pinyin IME)
- A malicious DLL: browser_host.dll

Contained within the zip archive named “test.zip” is the legitimate binary “biz_render.exe”, a popular Chinese Input Method Editor (IME) Sogou Pinyin.

Alongside the legitimate binary is a malicious DLL named “browser_host.dll”. As the legitimate binary loads a legitimate DLL named “browser_host.dll” via LoadLibraryExW, the malicious DLL has been named the same to sideload the malicious DLL into biz_render.exe. By supplying a malicious DLL with an identical name, the actor hijacks execution flow, enabling the payload to execute within a trusted process.

Figure 1: Biz_render.exe loading browser_host.dll.

The legitimate binary invokes the function GetBrowserManagerInstance from the sideloaded “browser_host.dll”, which then performs XOR-based decryption of embedded strings (key 0x90) to resolve and dynamically load mscoree.dll.

The DLL uses the Windows Common Language Runtime (CLR) to execute managed .NET code inside the process rather than relying solely on native binaries. During execution, the loader loads a payload directly into memory as .NET assemblies, enabling an in-memory execution.

C2 Registration

A GET request is made to:

```
GET /GetCluster?protocol=DotNet-TcpDmtp&tag={0}&uid={1}
```

with the custom header:

```
Verify-Token: Dmtp
```

This returns Base64-encoded and gzip-compressed IP addresses used for subsequent communication.

Figure 2: Decoded IPs.

Staged payload retrieval

Subsequent activity includes retrieval of multiple components from yahoo-cdn.it[.]com. The following GET requests are made:

```
/dfsvc.exe
```

```
/dnscfg.dll
```

```
/dfsvc.exe.config
```

```
/vhost.exe
```

```
/Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll
```

```
/config.etl
```

ClickOnce and AppDomain hijacking

Dfsvc.exe is the legitimate Windows ClickOnce Engine, part of the .NET framework used for updating ClickOnce Applications. Accompanying dfsvc.exe is a legitimate dfsvc.exe.config file that is used to store configuration data for the application. However, in this instance the malware has replaced the legitimate dfsvc.exe.config with the one retrieved from the server in: C:\Windows\Microsoft.NET\Framework64\v4.0.30319.

Additionally, vhost.exe the legitimate Visual Studio hosting process is retrieved from the server, along with “Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll” and “config.etl”. The DLL is used to decrypt the AES encrypted payload in config.etl and load it. The encrypted payload is dnscfg.dll, which can be loaded into vshost instead of dfsvc, and may be used if the environment does not support .NET.

Figure 3: ClickOnce configuration.

The malicious configuration disables logging, forces the application to load dnscfg.dll from the remote server, and uses a custom AppDomainManager to ensure the DLL is executed during initialization of dfsvc.exe. To ensure persistence, a scheduled task is added for %APPDATA%\Local\Microsoft\WindowsApps\dfsvc.exe.

Core payload

The DLL dnscfg.dll is a .NET binary named Client.TcpDmtp.dll. The payload is a heavily obfuscated backdoor that generates its logic at runtime and communicates with the command and control (C2) over custom TCP, DMTP (Duplex Message Transport Protocol) and appears to be an updated version of FDMTP to version 3.2.5.1

Figure 4: InitializeNewDomain.

The payload:

- Uses cluster-based resolution (GetHostFromCluster)
- Implements token validation
- Enters a persistent execution loop (LoopMessage)
- Supports structured remote tasking over DMTP

Once connected, the malware enters a persistent loop (LoopMessage), enabling it to receive commands from the remote server.

Figure 5: DMTP Connect function.

Rather than referencing values directly, they are retrieved through containers that are resolved at runtime. String values are stored in an encrypted byte array (_0) and decrypted by a custom XOR-based string decryption routine (dcsoft). The lower 16 bits of the provided key are XORed with 0xA61D (42525) to derive the initial XOR key, while subsequent bits define the string length and offset into the encrypted byte array. Each character is reconstructed from two encrypted bytes and XORed with the incrementing key value, producing the plaintext string used by the payload.

Figure 6: Decrypted strings.

Embedded in the resources section are multiple compressed binaries, the majority of which are library files. The only exceptions are client.core.dll and client.dmtppframe.dll.

Figure 7: Resources.

Modular framework and plugins

The payload embeds multiple compressed libraries, notably:

- `client.core.dll`
- `client.dmtppframe.dll`

`Client.core.dll` is a core library used for system profiling, C2 communication and plugin execution. The implant has the functionality to retrieve information including antivirus products, domain name, HWID, CLR version, administrator status, hardware details, network details, operating system, and user.

Figure 8: `Client.Core.Info` functions.

Additionally, the component is responsible for loading plugins, with support for both binary and JSON-based plugin execution. This allows plugins to receive commands and parameters in different formats depending on the task being performed.

The framework handles details such as plugin hashes, method names, task identifiers, caller tracking, and argument processing, allowing plugins to be executed consistently within the environment. In addition to execution management, the library also provides plugins with access to common runtime functionality such as logging, communication, and process handling.

Figure 9: `Client.core` functions.

`client.dmtppframe.dll` handles:

- DMTP communication
- Heartbeats and reconnection
- Plugin persistence via registry:

`HKCU\Software\Microsoft\IME\{id}`

`Client.dmtppframe.dll` is built on the `TouchSocket` DMTP networking library and continues to manage the remote plugins. The DLL implements remote communication features including heartbeat maintenance, reconnection handling, RPC-style messaging, SSL support, and token-based verification. The DLL also has the ability to add plugins to the registry under `HKCU/Software/Microsoft/IME/{id}` for persistence.

Plugins observed

While the full set of plugins remains unknown, researchers were able to identify four plugins, including:

- **`Persist.WpTask.dll`** - used to create, remove and trigger scheduled Windows tasks remotely.

- **Persist.registry.dll** - used to manage registry persistence with the ability to create, and delete registry values, along with hidden persistence keys.
- **Persist.extra.dll** - used to load and persist the main framework.
- **Assist.dll** - used to remotely retrieve files or commands, as well as manipulate system processes.

Figure 10: Plugins stored in IME registry.

Figure 11: Obfuscated script in plugin resources.

Persist.extra.dll is a module that is used to load a script “setup.log” to load and persist the main framework. Stored within the resources section of the binary is an obfuscated script that creates a .NET COM object that is added to the registry key HKCU\Software\Classes\TypeLib\ {9E175B61-F52A-11D8-B9A5-505054503030} \1.0\1\Win64 for persistence. After deobfuscating this script, another DLL is revealed named “WindowsBase.dll”.

Figure 12: Registry entry for script.

The binary checks in with icloud-cdn[.]net every five minutes, retrieves a version string, downloads an encrypted payload named checksum.bin, saves it locally as C:\ProgramData\USOShared\Logs\checksum.etl, decrypts it with AES using the hardcoded key P0t_L[Bsh0=+@0a., and loads the decrypted assembly directly from memory via Assembly.Load(byte[]). The version.txt file acts as an update marker so it only re-downloads when the remote version changes, while the mutex prevents duplicate instances.

Figure 13: USOShared/Logs.

Checksum.etl is decrypted with AES and loaded into memory, loading another .NET DLL named “Client.dll”. This binary is the same as “dnscfg.dll” mentioned at the start and allows the threat actors to update the main framework based on the version.

Conclusion

Across cases, Darktrace consistently observed the following sequence:

- Retrieval of legitimate executables
- Retrieval of DLLs for sideloading
- C2 registration via /GetCluster

This approach is consistent with broader China-nexus tradecraft. As outlined in Darktrace’s *Crimson Echo* report, the stable feature of this activity is behavioral. Infrastructure rotates and payloads can change, but the execution model persists. For defenders, the implication is

straightforward: detection anchored to individual indicators will degrade quickly. Detection anchored to a behavioral sequence offer a far more durable approach.

Credit to Tara Gould (Malware Research Lead), Adam Potter (Senior Cyber Analyst), Emma Foulger (Global Threat Research Operations Lead), Nathaniel Jones (VP, Security & AI Strategy)

Edited by Ryan Traill (Content Manager)

Appendices

A detailed list of detection models and triggered indicators is provided alongside IoCs.

Indicators of Compromise (IoCs)

Test.zip - fc3959ebd35286a82c662dc81ca658cb

Dnscfg.dll - b2c8f1402d336963478f4c5bc36c961a

Client.TcpDmtp.dll - c52b4a16d93a44376f0407f1c06e0b

Browser_host.dll - c17f39d25def01d5c87615388925f45a

Client.DmtpFrame.dll - 482cc72e01dfa54f30efe4fefde5422d

Persist.Extra - 162F69FE29EB7DE12B684E979A446131

Persist.Registry - 067FBAD4D6905D6E13FDC19964C1EA52

Assist - 2CD781AB63A00CE5302ED844CFBECC27

Persist.WpTask - DF3437C88866C060B00468055E6FA146

Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll -
c650a624455c5222906b60aac7e57d48

www.icloud-cdn[.]net

www.yahoo-cdn.it[.]com

154.223.58[.]142[AP8] [EF9]

MITRE ATT&CK Techniques

T1106 – Native API

T1053.005 - Scheduled Task

T1546.16 - Component Object Model Hijacking

T1547.001 - Registry Run Keys

T1511.001 - Dynamic Link Library Injection

T1622 – Debugger Evasion

T1140 – Deobfuscate/Decode Files or Information

T1574.001 - Hijack Execution Flow: DLL

T1620 – Reflective Code Loading

T1082 – System Information Discovery

T1007 – System Service Discovery

T1030 – System Owner/User Discovery

T1071.001 - Web Protocols

T1027.007 - Dynamic API Resolution

T1095 – Non-Application Layer Protocol

Darktrace Model Alerts

- Compromise / Beacons Activity To External Rare
- Compromise / HTTP Beacons to Rare Destination
- Anomalous File / Script from Rare External Location
- Compromise / Sustained SSL or HTTP Increase
- Compromise / Agent Beacon to New Endpoint
- Anomalous File / EXE from Rare External Location

- Anomalous File / Multiple EXE from Rare External Locations
- Compromise / Quick and Regular Windows HTTP Beacons
- Compromise / High Volume of Connections with Beacon Score
- Anomalous File / Anomalous Octet Stream (No User Agent)
- Compromise / Repeating Connections Over 4 Days
- Device / Large Number of Model Alerts
- Anomalous Connection / Multiple Connections to New External TCP Port
- Compromise / Large Number of Suspicious Failed Connections
- Anomalous Connection / Multiple Failed Connections to Rare Endpoint
- Device / Increased External Connectivity

Inside the SOC

Darktrace cyber analysts are world-class experts in threat intelligence, threat hunting and incident response, and provide 24/7 SOC support to thousands of Darktrace customers around the globe. *Inside the SOC* is exclusively authored by these experts, providing analysis of cyber incidents and threat trends, based on real-world experience in the field.

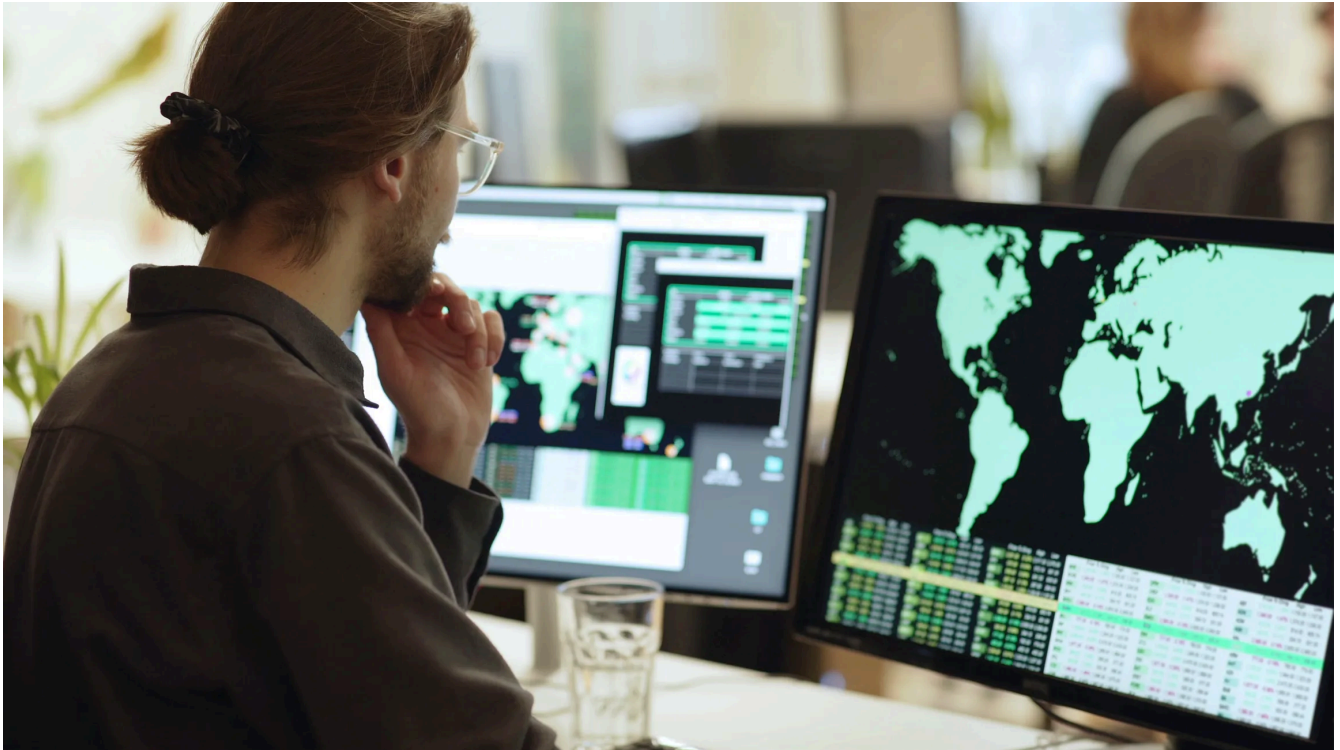
Written by

Adam Potter

Senior Cyber Analyst

Share this post

Latest blogs



When Trust Becomes the Attack Surface: Supply-Chain Attacks in an Era of Automation and Implicit Trust

Network

•

May 5, 2026

Nathaniel Jones

VP, Security & AI Strategy, Field CISO



Prompt Security in Enterprise AI: Strengths, Weaknesses, and Common Approaches

AI

-

May 20, 2026

Jamie Bali

Technical Author (AI) Developer