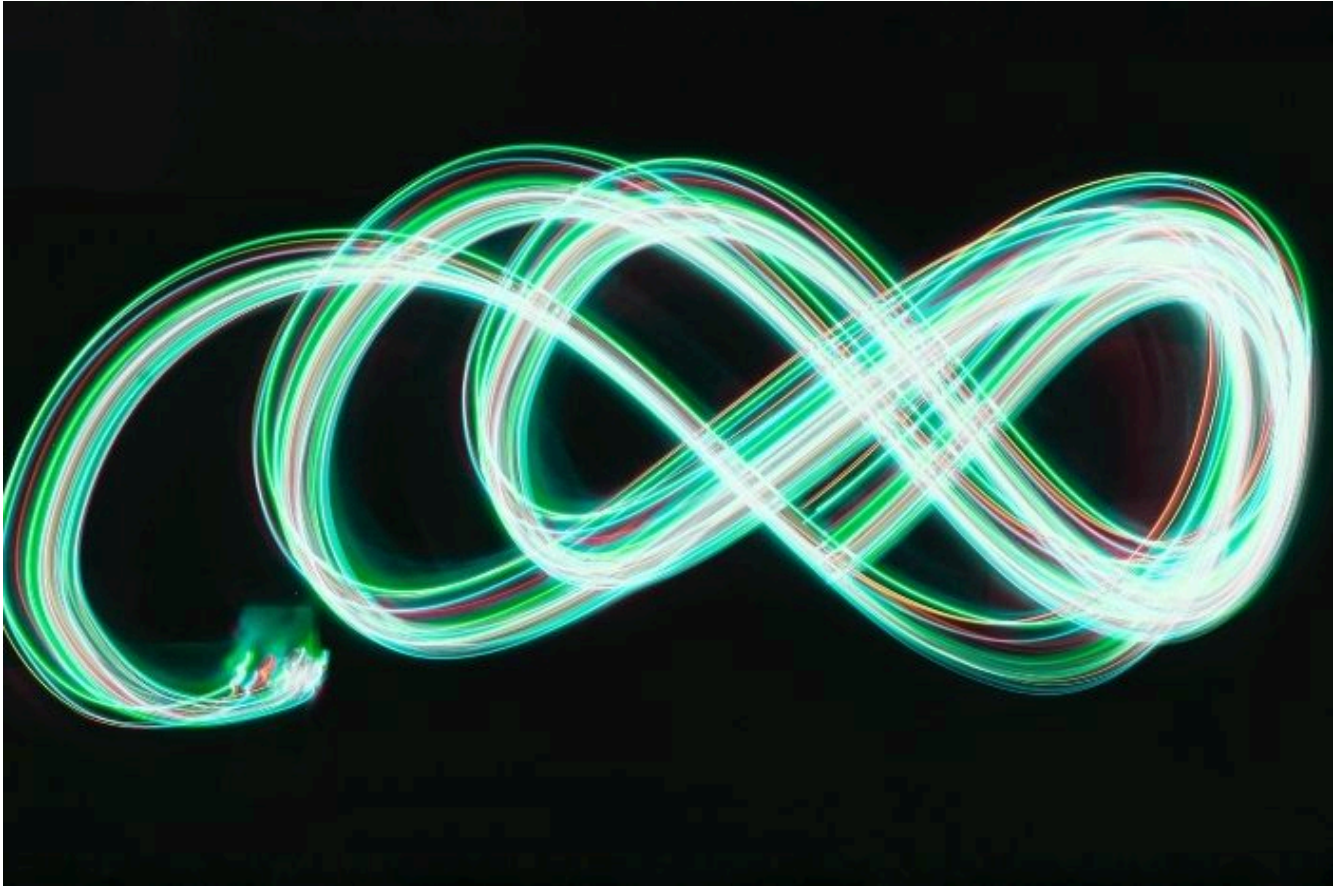


C2Looper: A New Backdoor Likely Tied To Ransomware With GitHub C2

zscaler.com/blogs/security-research/c2looper-new-backdoor-likely-tied-ransomware-github-c2

ThreatLabz

August 17, 2026



[Security Research](#)

Introduction

In July 2026, Zscaler ThreatLabz identified a new Rust-based malware family that we track as *C2Looper*, which is likely leveraged by a ransomware-related threat actor. Furthermore, ThreatLabz assesses with low to medium confidence that C2Looper has been delivered to victims through a multi-stage ClickFix infection chain. C2Looper supports backdoor commands including executing arbitrary commands, performing reconnaissance, and deploying second-stage payloads.

In this blog post, ThreatLabz provides a technical analysis of the identified C2Looper variants, including their network communication protocols and capabilities.

Key Takeaways

- *C2Looper*, a new malware family likely used in ransomware attacks to establish a foothold for lateral movement.
- C2Looper supports typical backdoor commands including remote shell execution, reconnaissance, and deploying additional malware tooling.
- C2Looper dynamically resolves Windows APIs and encrypts strings.
- C2Looper appears to be under active development. ThreatLabz identified a variant with additional features and capabilities, including the use of GitHub for command-and-control (C2) communications.
- ThreatLabz assesses with low to medium confidence that C2Looper is distributed through ClickFix campaigns.

Technical Analysis

In the following sections, ThreatLabz examines the technical details of the identified C2Looper variants, including their features, network communication protocols and commands.

String encryption and Windows API resolution

All identified C2Looper variants use string encryption and dynamically resolve Windows APIs.

- C2Looper uses the Windows API functions `LoadLibrary` and `GetProcAddress` to load and resolve any necessary functions at runtime.
- C2Looper decrypts strings at runtime using a bitwise XOR operation with an 8-byte key. Despite not having a single decryption function for every encrypted string, C2Looper uses the same XOR key throughout the code.

Network communication

C2Looper uses plaintext HTTP to communicate with its C2 server through a simple network protocol. First, C2Looper collects host information and sends it to the C2 server as a JSON object in an HTTP POST request. The JSON object contains the following information:

- Username.

- DNS hostname of the compromised host.
- The process identifier (PID) of C2Looper that is currently running on the compromised host.
- Bot ID (id key), which is a combination of the username and hostname.

This information is formatted as follows:

```
{
  "id": "DNSHOSTNAME_Username",
  "user": "username",
  "host": "DNSHOSTNAME",
  "pid": C2LOOPER_PID_NUMBER
}
```

C2Looper sends this information to the endpoint `/api/beacon` every second to request a command. Each command that is received from the C2 server is formatted as a JSON object containing the keys described in the table below.

JSON

Key	Description
arg	The command parameters. For example, this key contains the payload URL for the upload command.
cmd	Specifies the command name.
ok	A Boolean value indicating whether the command was parsed successfully.
seq	Appears to serve as a task ID. C2Looper includes this value when reporting command results to the endpoint <code>/api/result/</code> . In this version, the value received from the C2 server is an integer but C2Looper expects a string. This is a parsing bug that always causes C2Looper to set the task ID value to 0 when reporting results.

Table 1: C2Looper command message format.

The table below describes the C2 endpoints used by C2Looper.

Endpoint	Description
/api/beacon	Retrieves commands from the C2 server.
/api/result/BOT_ID/task_ID	Reports command output to the C2 server.

Table 2: C2Looper C2 endpoints.

The table below describes the commands supported by C2Looper.

Command	Description
ping	Ping message
run	Executes a system command on the compromised host using <code>cmd.exe</code> . The output is ignored.
shell	Executes a system shell command and returns the output to the C2 server. C2Looper first writes the output to the file <code>c2_out.txt</code> . C2Looper then reads the file and sends its contents to the C2 server.
upload	Downloads a PE file from a specified URL to the compromised host and saves it as the filename <code>wtsapi32.dll</code> under the path <code>%LocalAppData%\Microsoft\OneDrive\</code> . C2Looper then terminates the OneDrive process and abuses a legitimate OneDrive executable to load the malicious DLL. We assess that this command functions as an update mechanism and uses DLL sideloading to evade detection.
download	Downloads a file from a specified location and saves it as the filename <code>p1d.exe</code> under <code>%LocalAppData%</code> . The command's intended purpose is currently unknown.

Table 3: Commands supported by C2Looper in older variants.

ANALYST NOTE: Interestingly, Oyster malware (which is likely related with the threat actor behind Lactrodectus) uses similar API endpoints for C2 communication.

C2Looper version 2

During our research, ThreatLabz identified a new version of C2Looper that supports additional commands, and most interestingly, uses GitHub for C2 communication. We also observed that the older C2Looper variant described previously was used to download the latest version of C2Looper through the `upload` command. This latest version appears to be internally tagged as version 2 as we will discuss later.

C2Looper v2 uses GitHub for all C2 operations, including storing exfiltrated data and reporting command output. In addition to the use of GitHub, C2Looper v2 introduces new commands and modifies the functionality of existing commands. The table below describes these additions and changes.

Command	Description
ls	Lists files in a specified directory.
recon	Collects additional host information using the following commands: • <code>ipconfig /all</code> • <code>whoami /all</code> • <code>nltest /dclist</code> • <code>net group /domain "domain computers"</code> • <code>net group /domain "domain admins"</code> • <code>wmic product get name, version</code>
drives	Returns a list of drives on the compromised host.
inject	Loads the legitimate <code>winspool.drv</code> and copies the downloaded shellcode data into its <code>text</code> section. Then, C2Looper executes this section by creating a new thread that points to this section's address.
upload	Unlike the previous version, this version downloads a specified file from a dedicated folder in the GitHub repository to the Windows temporary folder.

Command	Description
run	Unlike the previous version, C2Looper v2 executes a specified file on disk using <code>ShellExecuteW</code> and then deletes the file if the execution was successful.
shell	Instead of storing the command output in the file <code>c2_out.txt</code> , this variant captures the output using Windows pipes.

Table 4: C2Looper v2 commands.

Furthermore, we identified the following noteworthy changes:

- Unlike the previous variant, the beacon request includes only the host's username and timestamp (based on the host's local time).
- The binary includes debug messages including `!!! v2 !!! pongv2 from`. This is used as a response when the network command ping is used. This is an indicator that the malware developer refers to this variant as version 2 of C2Looper.
- The JSON key `ok` has been removed from beacon requests and responses.
- The parsing bug related to the JSON key `seq` has been fixed in this version and C2Looper uses the correct task ID with the appropriate command output.

Since this variant uses GitHub for C2 communication, the endpoints have been removed. Instead, C2Looper creates a directory for each bot in the GitHub repository and uses the JSON files in the table below:

Filename	Description
<code>cmd.json</code>	Stores command data for C2Looper to execute.
<code>result.json</code>	Stores command output.
<code>beacon.json</code>	Stores the bot ID and a timestamp that indicates the last successful request.

Table 5: JSON files used by C2Looper v2 with GitHub-based C2 communication.

Conclusion

Overall, C2Looper is a new malware family that provides common backdoor capabilities, including system command execution and deploying arbitrary second-stage binaries. This functionality is likely used by an initial access broker to steal sensitive data and deploy ransomware. Despite its code simplicity, the additional functionality in the latest variant indicates that its developers continue to expand and refine its capabilities.

Zscaler Coverage

Zscaler’s multilayered cloud security platform detects indicators related to C2Looper at various levels. The figure below depicts the Zscaler Cloud Sandbox, showing detection details for C2Looper.

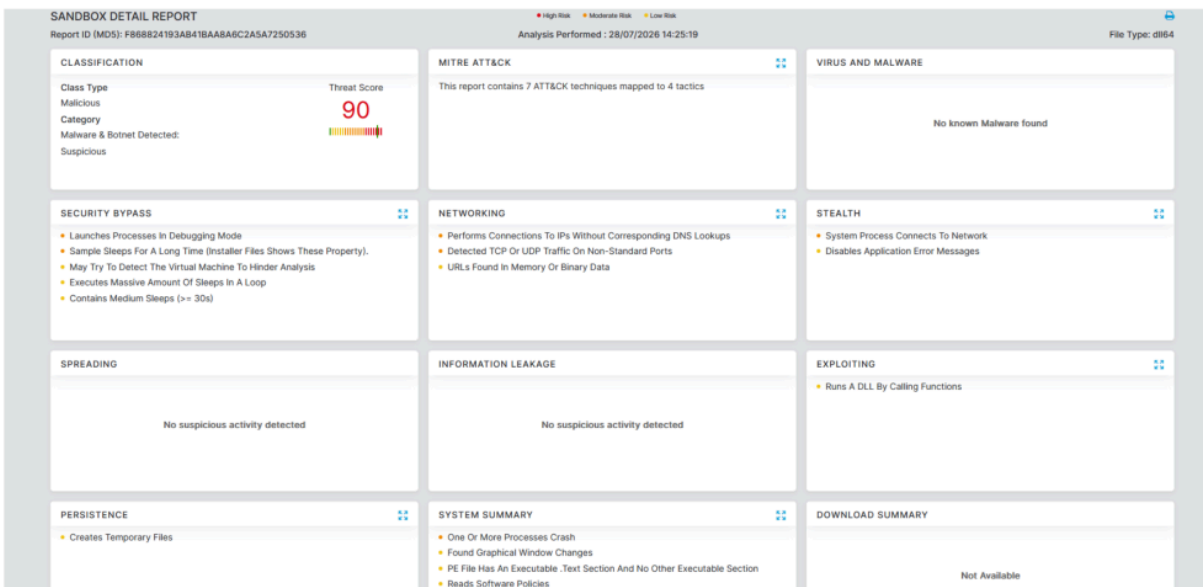


Figure 1: Zscaler Cloud Sandbox Report for C2Looper.

In addition to sandbox detections, Zscaler’s multilayered cloud security platform detects indicators related to C2Looper at various levels with the following threat names:

Indicators Of Compromise (IOCs)

IOC	Description
f96ff2f3abbbf7f382ace509b90e54853b4b61c402ecde27d82f1c17b414867b	Latest variant of C2Looper.
20675a659c338f7267fd09bacb431f4491f061d3acf42d07aca2dec3d25fa549	C2Looper debug build.
f59f32c9af4fa8a5dbd4668df8893593bc0c4324816cbf9b956acedcbfb8cdb6	Older variant of C2Looper.
45.158.196[.]23:8888	C2Looper C2.
45.158.196.184:8888	C2Looper C2.

Was this post useful?

Explore more Zscaler blogs

Get the latest Zscaler blog updates in your inbox

By submitting the form, you are agreeing to our [privacy policy](#).