Threat Group Targets Companies in Taiwan | FortiGuard Labs

By Pei Han Liao

Published: 2025-06-17 · Archived: 2026-04-05 20:29:47 UTC

Affected Platforms: Microsoft Windows
Impacted Users: Microsoft Windows
Impact: The stolen information can be used for future attacks
Severity Level: High

In January 2025, FortiGuard Labs observed an attack targeting users in Taiwan. The threat actor is spreading the malware known as winos 4.0 via an email masquerading as being from Taiwan's National Taxation Bureau. Through continued monitoring, we identified further malware samples associated with this campaign. Among the new samples, a phishing email was sent in March 2025 with an attachment that contained a link used in another attack campaign.

Figure 1: The HTML file in the phishing email

The first link belongs to the domain twszz[.]xin, which follows a similar naming pattern to the campaign targeting users in Taiwan. The second link directs to an image file about tax inspection, while the HTML filename claims to include account statement details.

This link enabled us to trace the attack and identify additional malware samples, along with further links. Figure 2 provides a simplified threat map. The files on the left side of Figure 2 are XLS files used in campaigns that took place in June 2024.

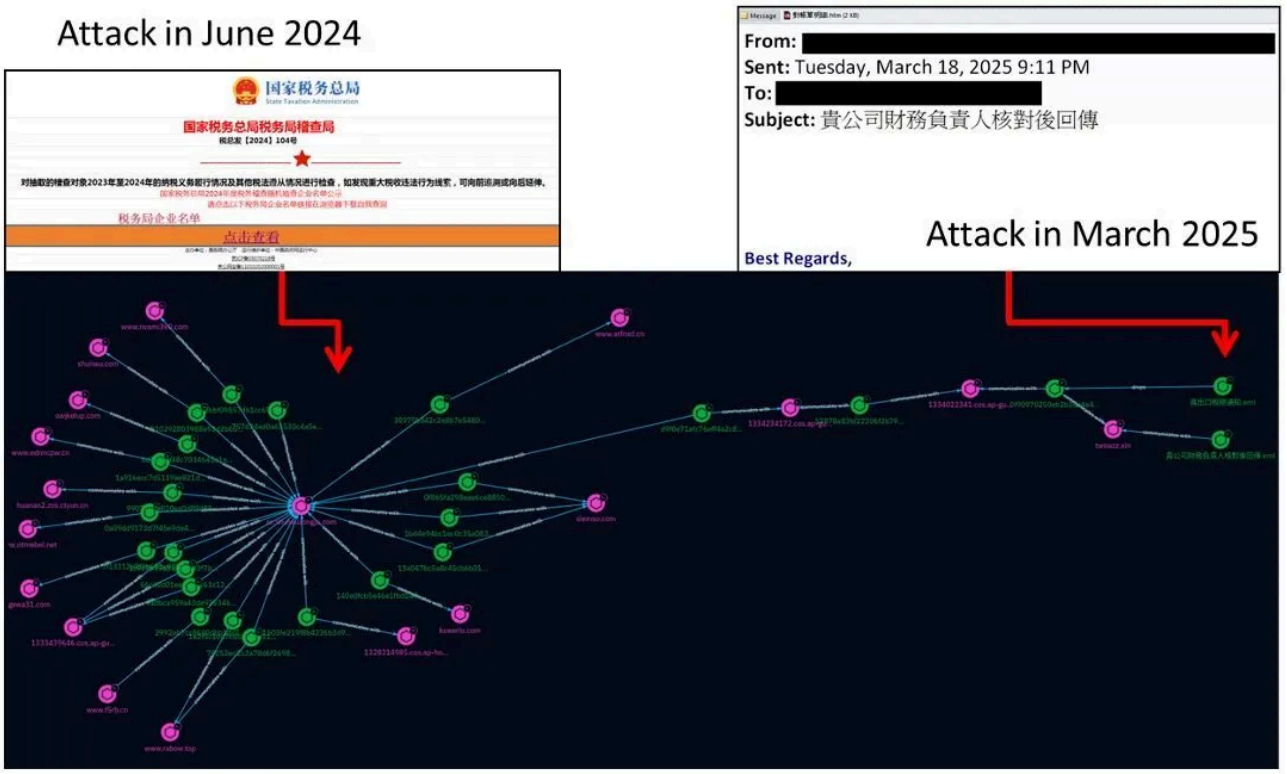

Figure 2: Threat map

Over the past few months, this threat group has deployed malware based on the HoldingHands RAT (Remote Access Trojan), also known as Gh0stBins, to compromise users in Taiwan. The malware typically comprises multiple files embedded within a ZIP file and is distributed via phishing emails.

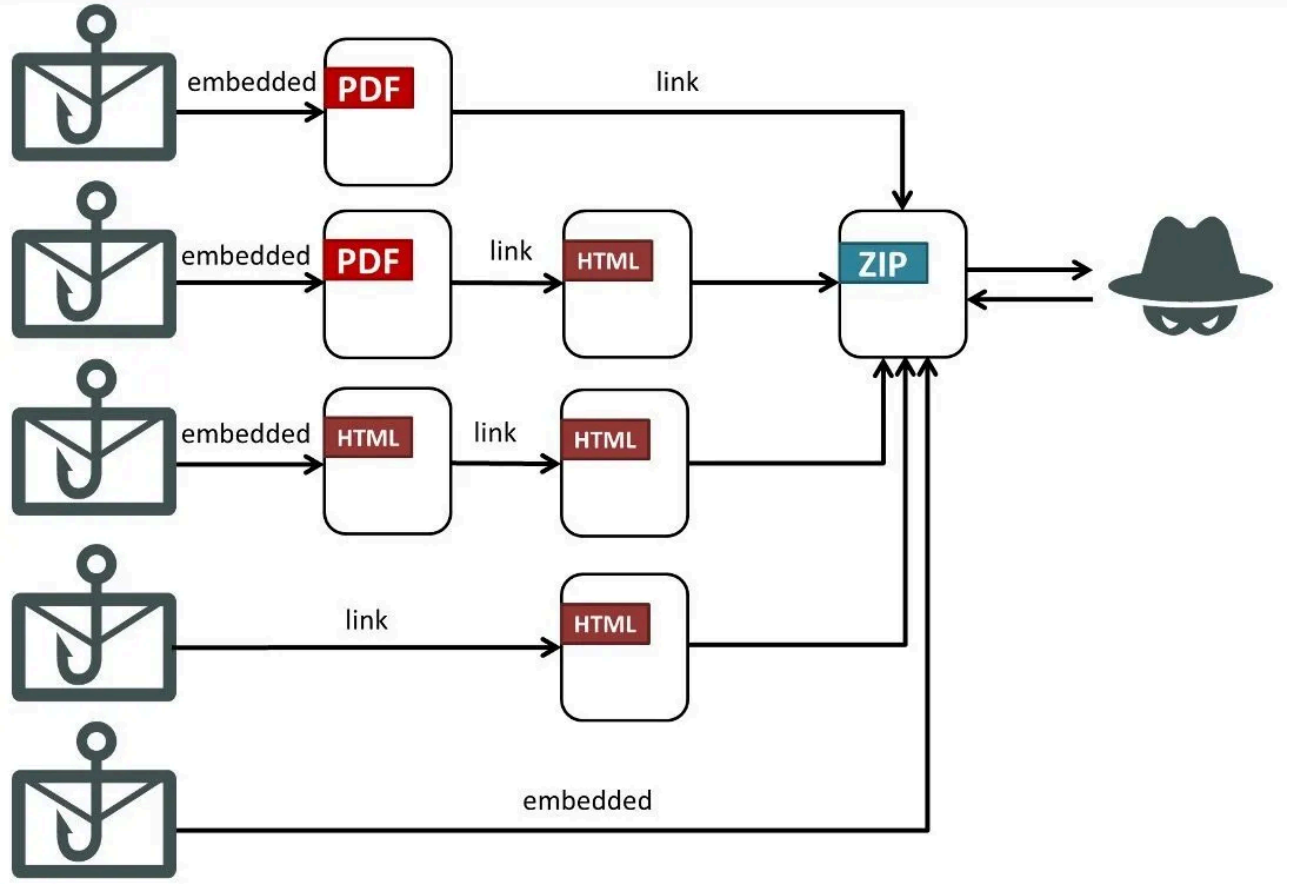

Figure 3: Attack flow

Phishing

Phishing emails typically masquerade as messages from the government or business partners, using topics such as taxes, pensions, invoices, and other subjects that prompt the recipient to immediately click on or open an attachment. Sometimes, the email content can be a picture with a hyperlink that asks the recipient to click on it, inadvertently downloading the malware.

Figure 4: An example of an email containing a picture with a hyperlink

The attached PDF file uses content related to the phishing email to trick the recipient into opening the link. In newer attack chains, the link leads to a download page.

Figure 5: An example of a phishing email

Figure 6: The PDF file attached to the email in Figure 3

The malware download page looks much simpler than the PDF file and email. It only contains text and a download button. In some attack chains, the malware is embedded in a password-protected ZIP file, and the

password is on the download page. This prevents analysts who get the ZIP file but don't have access to the download page from opening it.

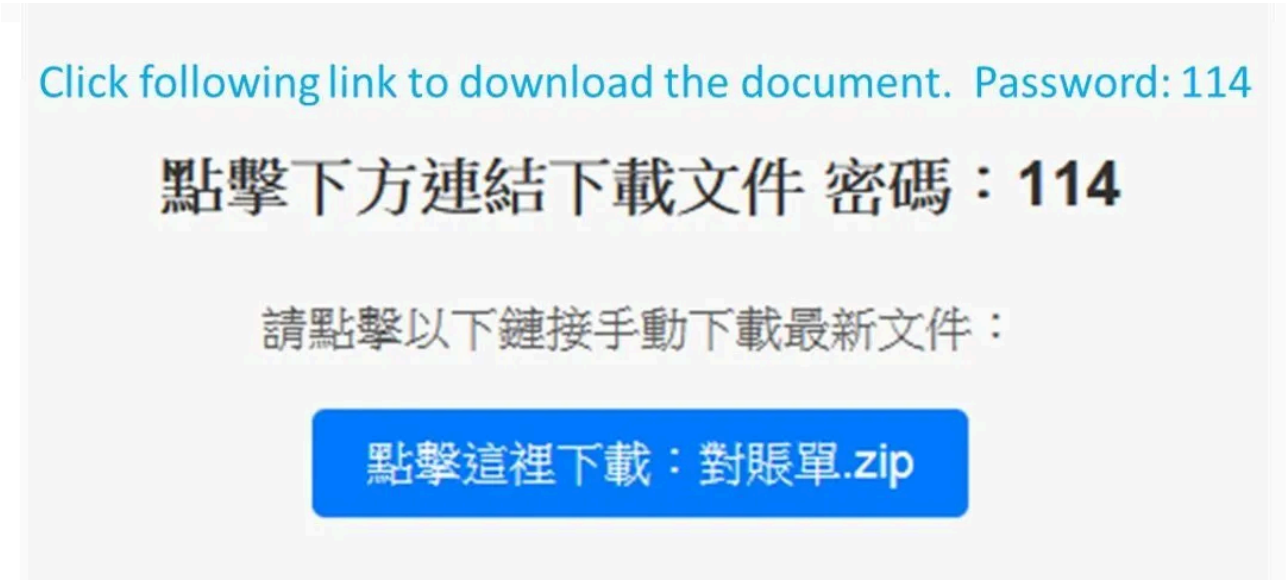

Figure 7: An example of the download page with a password

ZIP file

Multiple files are used during the attack, including legitimate executable files and necessary DLL files, encrypted shellcode, and shellcode loaders. The shellcode loaders, which decrypt and execute the encrypted shellcode, are DLL files loaded by a legitimate executable via side-loading.

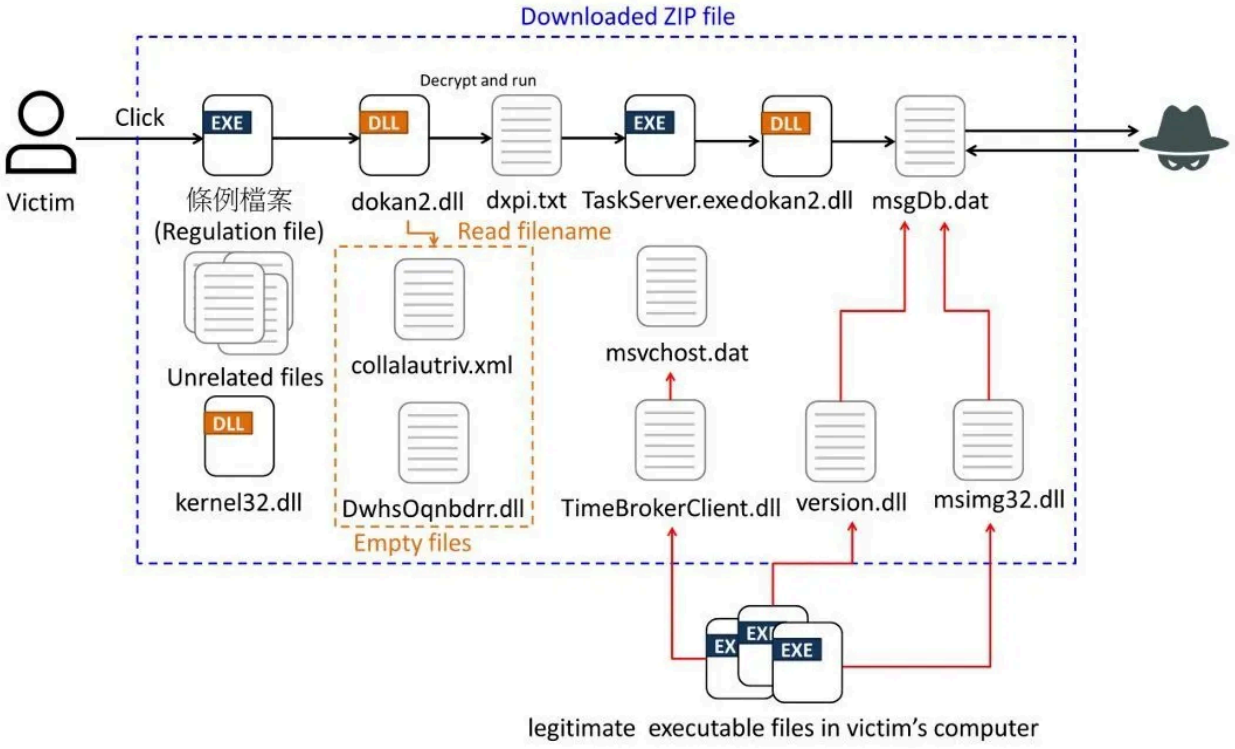

Figure 8: An example of the execution flow of the files in the ZIP file

Figure 8 shows an example of the files embedded in the ZIP file and the execution flow. **條例檔案**is the legitimate executable file used to load **dokan2.dll** via side-loading. In addition to the main execution flow, encrypted shellcodes support persistence, and empty files provide unique filenames. Although the ZIP files downloaded from different PDF files and webpages may have varying folder structures and files, their execution flows are similar to those shown in Figure 8. Sometimes, the ZIP file only contains an executable that drops the duplicate files observed in other chains. According to the image debug directory of the executable file, the malware is based on the HoldingHands Remote Access Trojan (RAT).

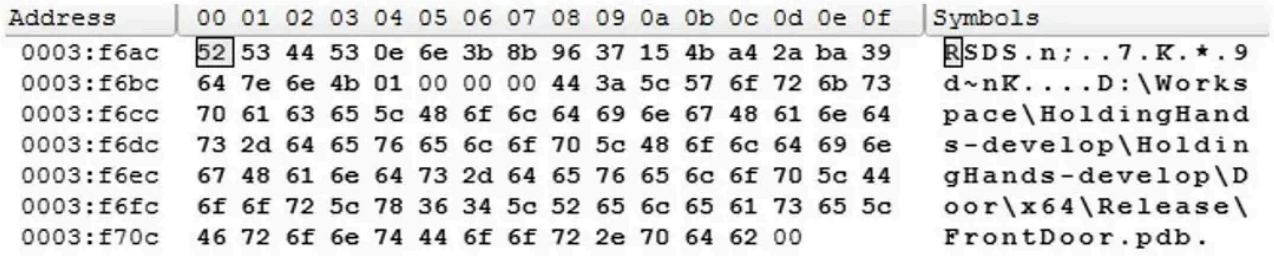

Figure 9: The image debug directory of the executable file in other attack chains.

Over the past two months, the ZIP file has included a text file containing the passwords for other files in the ZIP file, which makes detection more difficult.

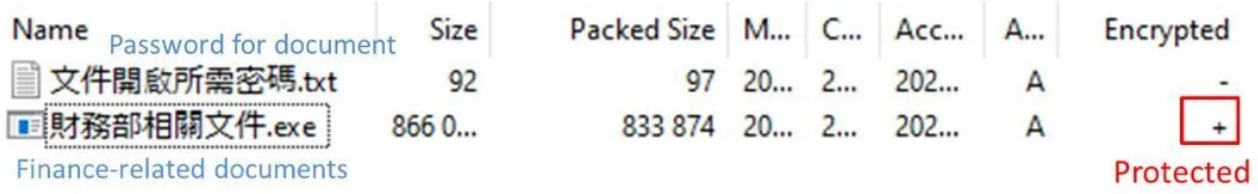

Figure 10: An example of the password-protected ZIP file

Dokan2.dll

Dokan2.dll creates a thread to decrypt data in **dxpi.txt** and execute it. Before this, it calls the **ShowWindow** function to hide the executable's window for side-loading. It then searches for **kernel32.dll** and **DwhsOqnbdrr.dll** by comparing the lengths of the filenames of the files extracted from the ZIP file.

DwhsOqnbdrr.dll is an empty file. By shifting each letter in the filename "**DwhsOqnbdrr"** forward one position in the alphabet, it becomes **ExitProcess** and loads the function from the **kernel32.dll** it just found. It replaces the address of the **ExitProcess** function in the import table with the address of a function that calls the **WaitForSingleObject** function to wait for a signal from the thread that decrypts **dxpi.txt**.

When the thread finishes, it calls the **ExitProcess** function that it just loaded. In the thread, it executes the **條例檔案** as an administrator if it doesn't have high enough privileges. Then it searches for **collalautriv.xml** and converts the filename to get **VirtualAlloc**, the API used in decryption.

dxpi.txt

dxpi.txt executes initial setups for the next stage, including anti-VM, privilege escalation, and installation.

- **Anti-VM**
  This function checks the amount of physically installed RAM because many sandboxes and virtual machines are assigned lower amounts of memory to reduce system load. If the amount of physically installed RAM is less than 8 GB, it exits.
- **Privilege escalation**
  First, it enables the **SeDebugPrivilege** privilege to bypass the access restriction of **WinLogon**. It then calls the **ImpersonateLoggedOnUser** function to impersonate the user (SYSTEM) of **WinLogon**. Finally, it impersonates the **TrustedInstaller** service's thread to obtain the highest privilege.
- **Installation**
  It creates a registry key as an infection marker:

  Subkey: SOFTWARE\MsUpTas

  Value name: State

  Value: 1

In addition, it drops other files extracted from the ZIP file to C:\Program Files (x86)\WindowsPowerShell\Update.

Original filename	**After Filename of dropped file**	**Description**
bkproc.dll	TaskServer.exe	The same file as the **條例檔案**.
code.dll	code.bin	It's copied as System32**msvchost.dat**.
Db.dll	msgDb.dat	The malicious payload. Shellcode based on HoldingHands.
Doport.dll	dokan2.dll	Shellcode loader for **msgDb.dat**.
EGLProtect.dll	libEGL.dll	The legitimate DLL file for **條例檔案**.

fig32.dll	config32.bin	Unused. It renames the legitimate **version.dll** as **confVersion.dll** and writes the decrypted data of **config32.bin** to SysWOW64**version.dll** if it's used.
fig64.dll	config64.bin	It renames the legitimate **TimeBrokerClient.dll** as **BrokerClientCallback.dll** and writes the decrypted data of **config64.bin** to **TimeBrokerClient.dll**.
simg32.dll	simg64.dll	Binary file that is used by **msvchost.dat**.

In addition, it terminates if **BrokerClientCallback.dll** and **Blend.dll** are present, indicating that the computer is infected. **Blend.dll** is the legitimate **msimg32.dll** that is later renamed by **msvchost.dat**. After installation, it executes **TaskServer.exe**, which loads **dokan2.dll** via side-loading. Dokan2.dll **then** decrypts and executes the shellcode in **msgDb.dat** for the next stage.

Other files

- **fig64.dll → config64.dll → TimeBrokerClient.dll**
  The original **TimeBrokerClient.dll** is a legitimate DLL related to **TaskScheduler** loaded by **svchost.exe**. It terminates if the calling process is not **svchost.exe**. After a command-line check, it decrypts and runs the shellcode in **msvchost.dat.**

- **code.dll → code.bin → msvchost.dat**
  The fake TimeBrokerClient.dll executes this. It only continues when the command-line is C:\windows\system32\svchost.exe -k netsvcs -p -s Schedule and avp.exe (Kaspersky) is not running. After the check, it uses the same method as **dxpi.txt** to escalate privileges and then copies files from C:\Program Files (x86)\WindowsPowerShell\Update to C:\Windows\System32:

Original Filename	**Filename of dropped file**	**Description**
msgDb.dat	system.dat, mymsc.nls	The malicious payload.
dokan2.dll	dokan2.dll	Shellcode loader for **msgDb.dat**.
libEGL.dll	libEGL.dll	The legitimate DLL file for **條例檔案**.

TaskServer.exe	taskyhost.exe	The same file as the **條例檔案**.
simg64.dll	msimg32.dll	Shellcode loader for **system.dat**.

- **simg32.dll → simg64.dll → msimg32.dll**
  The original **msimg32.dll** is a legitimate DLL used by Microsoft Graphical Device in many applications, including LINE and WeChat. The fake **msimg32.dll** terminates if the calling process is not LINE.exe or WeChat.exe. It also sleeps if **TaskServer.exe** is running. After the check, it decrypts and runs the shellcode in **system.dat** (the malicious payload).

- **fig32.dll → config32.bin → SysWOW64\version.dll (if used)**
  The original **version.dll** is a legitimate DLL file about version information used by many applications. The fake **version.dll** is not used in this attack chain, and its code is incomplete. By comparing its code to the **version.dll** dropped in other attack chains, we assume it is a shellcode loader for the malicious payload, similar to **msimg32.dll**.

msgDb.dat

MsgDb.dat implements C2 tasks for setting registry keys, data collection, and module download from the HoldingHands RAT. It also sends heartbeat packets to ensure the connection is active.

Below is the packet's data structure, excluding the header. The packets from **msgDb.dat** and the C2 server follow this structure.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	Magic				Data size				Unused				Command			
1	Payload(optional)															
⋮	⋮															

Magic: 0xDEADBEEF
Data size: The size of the command and the payload

The first outgoing packet doesn't contain a payload. The **KNEL** command indicates that the packet is from a kernel module. As a response, the C2 server sends a data collection request. After sending the user information,

msgDb.dat sends heartbeat packets and waits for further instructions.

- **Heartbeat**
  Command: 0x12, 0x13, 0x14

  msgDb.dat sends heartbeat packets every three minutes, and the C2 server responds with command 0x12. In addition, **msgDb.dat** sends a packet with command 0x13 after the computer has been idle for 30 seconds and 0x14 when user activity resumes.

- **Data Collection**
  Command: 0x00, 0x01
  Payload: Delivers user information, including IP address, computer name, user name, operating system, architecture, install time, CPU frequency, number of processors, physical memory, registry values set by other commands, and the interval between pings to the C2 server.

The response command is 0x00. To get the install date, it reads the **InstallDate** value from the SOFTWARE\HHClient registry key. If this is the first time the C2 server queries for this information, it writes the current time to the value. The registry values set by other commands are **Comment** and **Group** from the SOFTWARE\HHClient registry key. If the **Comment** value is not set, it writes **default** to the packet.

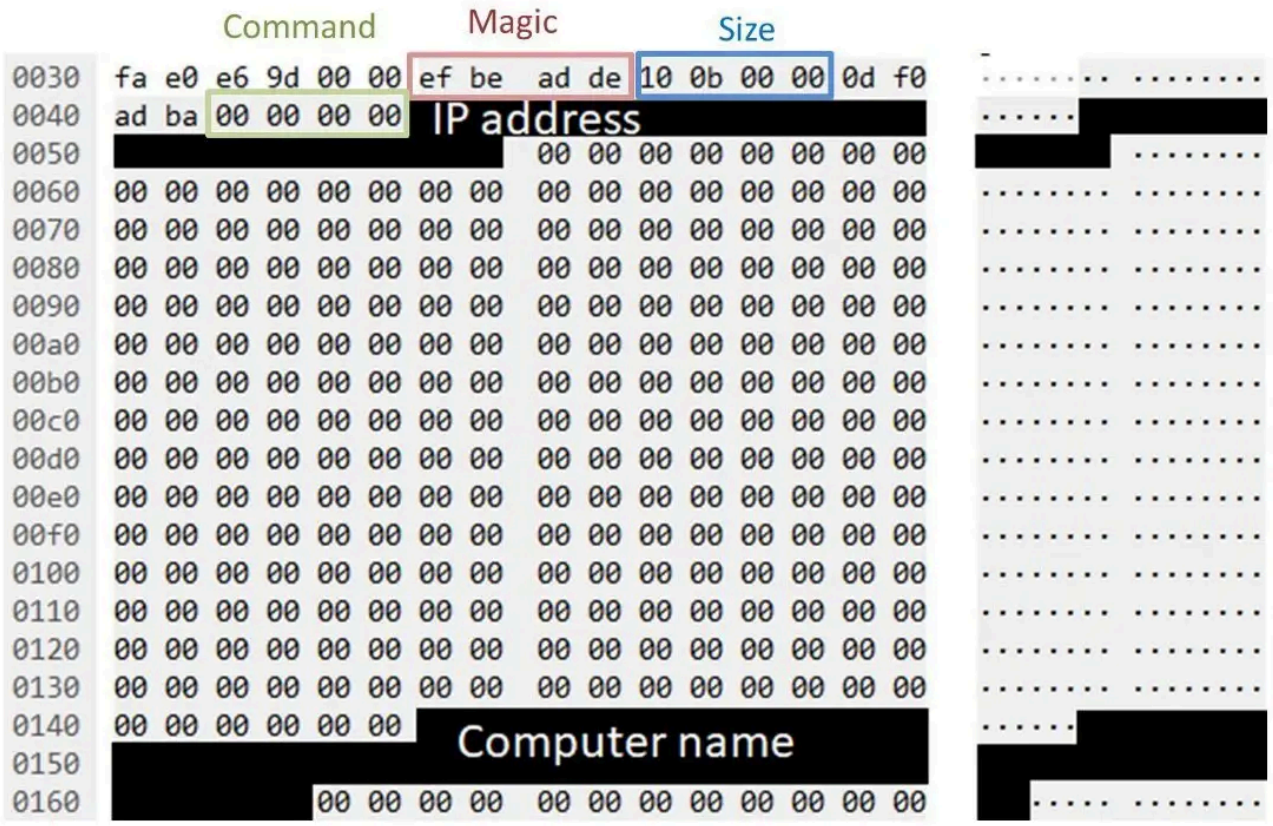

Figure 11: The packet containing victim information

- **Edit Comment**
  Command: 0x04, 0x5

Payload: Value of **Comment**It writes data from the server to the **Comment** value in the SOFTWARE\HHClient registry key. The result is sent to the C2 server with 0x05.

- **Edit Group**
  Command: 0x06
  Payload: Value of **Group**It writes data from the server to the **Group** value in the SOFTWARE\HHClient registry key, and the result is sent to the C2 server with 0x07.

- **Module Info**
  Command: 0x0A, 0x0B
  Payload: Module size and module name

  This is the module name and size to be executed. This is sent when the current module is not the module specified by the server. **msgDb.dat** requests module data from the C2 server using the information from the server and command 0x0B.

- **Add module**
  Command: 0x0B, 0x0C
  Payload: module size, data size in this packet, module data

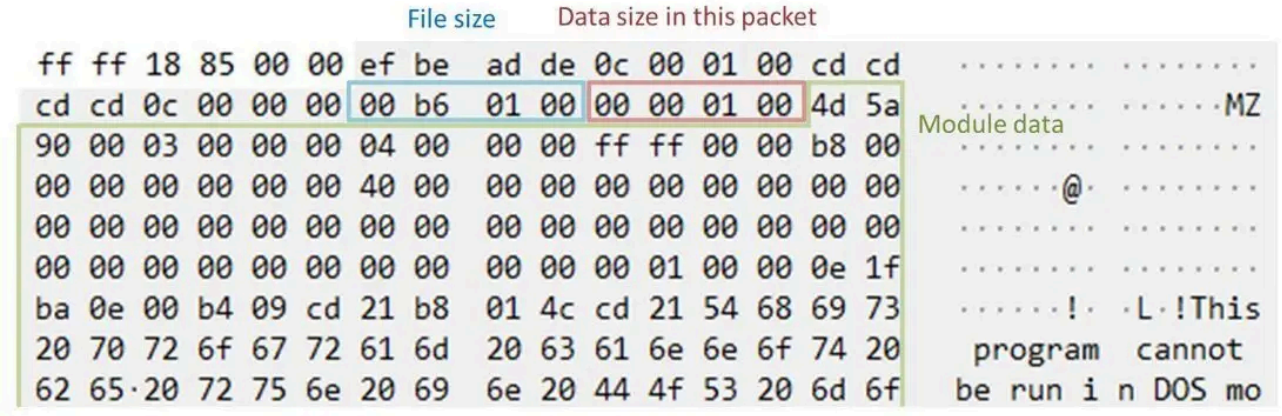

Figure 12: Packet from server

Once all data is downloaded, **msgDb.dat** executes the module. Otherwise, it sends 0x0B to ask for more data.

During our analysis, we identified three modules delivered by the C2 server, including two remote desktop modules and a file manager. **msgDb.dat** calls the only export function, ModuleEntry, to proceed to the next stage of the attack. Below are the commands in the initial packet:

Module name	**Command**	**Description**
rd	RDTP	Remote desktop

rd_dxgi	RDTP	Remote desktop
filemgr	FMGR	File manager

The packets follow the same structure as **msgDb.dat**.

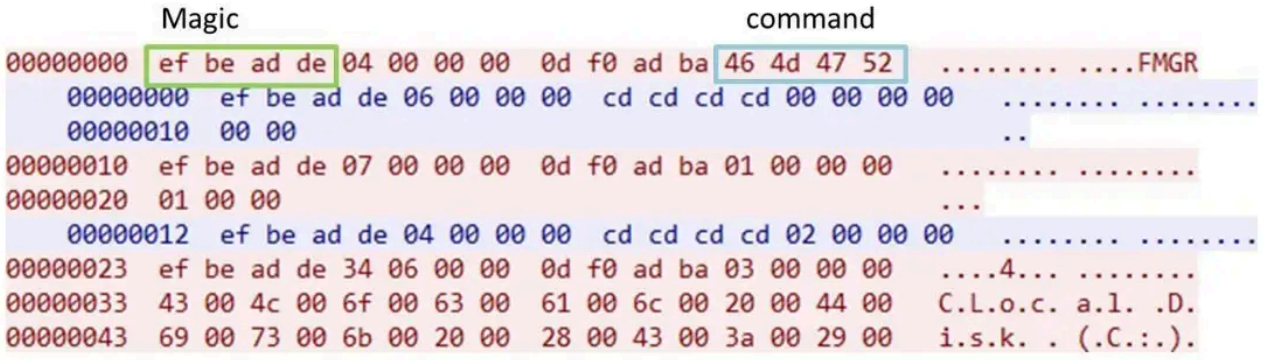

Figure 13: The communication between the C2 server and the filemgr module.

The modules' image debug directories indicate that they also belong to the HoldingHands RAT. Some modules appear to be simplified versions, as indicated by the term 'jingjianban' (meaning 'lite version' in Chinese) in the Image Debug Directory.

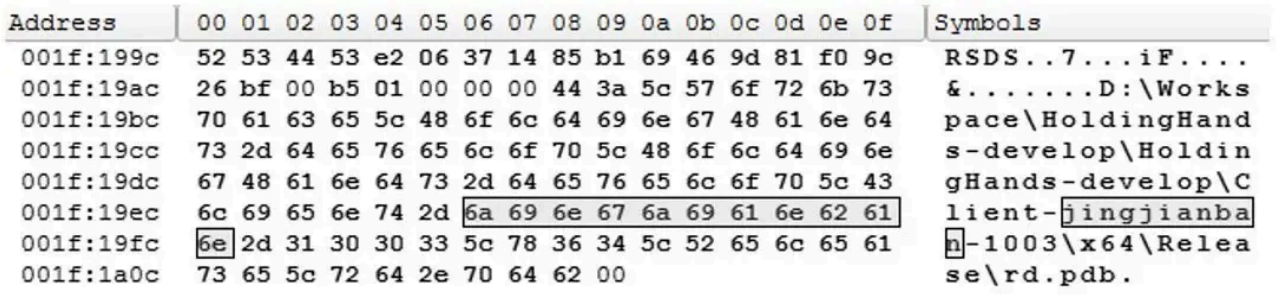

Figure 14: The image debug directory of the rd module.

- **Run Module**
  Command: 0x09, 0x11
  Payload: Module name and function name

  This command asks **msgDb.dat** to run the module specified by the payload. If the module is not found, **msgDb.dat** sends command 0x09 along with the module name to request module information.

- **Exit**
  Command: 0x15

  Terminates.

Other Attack Chains

In addition to winos, which we covered in February 2025, and HoldingHands, discussed in this article, this threat group frequently employs Gh0stCringe. Figures 5 through 7 include screenshots of files in this attack chain.

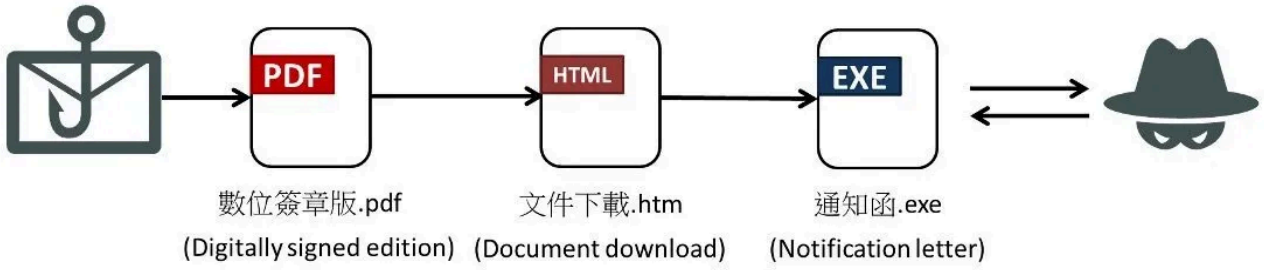

Figure 15: Attack chain of Gh0stCringe

Conclusion

This analysis revealed further malware samples associated with the attack that began targeting Taiwan in January 2025. The attack chain comprises numerous snippets of shellcode and loaders, making the attack flow complex. However, the purpose of these samples is to execute a malicious payload that accesses a C2 server to receive further instructions. Across winos, HoldingHands, and Gh0stCring, this threat group continuously evolves its malware and distribution strategies.

FortiGuard will continue to monitor these attack campaigns and provide appropriate protections as required.

Fortinet Protections

The malware described in this report is detected and blocked by FortiGuard Antivirus as:

PDF/Agent.A6DC!tr.dldr
W64/ShellcodeRunner.ARG!tr
W64/Agent.FIN!tr
W64/HHAgent.BEE8!tr

FortiGate, FortiMail, FortiClient, and FortiEDR support the FortiGuard AntiVirus service. The FortiGuard AntiVirus engine is part of each of these solutions. As a result, customers who have installed the latest updates for these products are protected.

The FortiGuard CDR (Content Disarm and Reconstruction) service, which runs on both FortiGate and FortiMail, can disarm malicious macros in documents.

We also suggest that organizations go through Fortinet's free NSE training module: FCF Fortinet Certified Fundamentals. This module is designed to help end users learn how to identify and protect themselves from phishing attacks.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you believe this or any other cybersecurity threat has impacted your organization, please contact our Global FortiGuard Incident Response Team.

IOCs

IP

154[.]91[.]85[.]204
154[.]86[.]22[.]47
156[.]251[.]17[.]17
206[.]238[.]179[.]173
206[.]238[.]220[.]60
206[.]238[.]199[.]22
154[.]91[.]85[.]201
206[.]238[.]221[.]182
206[.]238[.]196[.]32
154[.]91[.]64[.]45
206[.]238[.]115[.]207
156[.]251[.]17[.]12
107[.]149[.]253[.]183

Domain

00-1321729461[.]cos[.]ap-guangzhou[.]myqcloud[.]com
6-1321729461[.]cos[.]ap-guangzhou[.]myqcloud[.]com
twzfte-1340224852[.]cos[.]ap-guangzhou[.]myqcloud[.]com
cq1tw[.]top
twcz[.]pro
twczb[.]com
twnc[.]ink
twnic[.]icu
twnic[.]ink
twnic[.]ltd
twnic[.]xin
twsa[.]top
twsw[.]cc
twsw[.]club
twsw[.]info

twsw[.]ink
twsw[.]ltd
twsw[.]pro
twsww[.]vip
twsww[.]xin
twswz[.]top
twswzz[.]xin
twtgtw[.]net
twzfw[.]vip

Phishing mail

6558dfb070421c674b377a0a6090593fa0c44d5b0dec5325a648583f92175ce2
d3a270d782e62574983b28bd35076b569a0b65236e7f841a63b0558f2e3a231c
a8430ce490d5c5fab1521f3297e2d277ee7e7c49e7357c208878f7fd5f763931
7d3f352ded285118e916336da6e6182778a54dc88d4fb7353136f028ac9b81e0
143f434e3a2cac478fb672b77d6c04cdf25287d234a52ee157f4f1a2b06f8022
c25e80cd10e7741b5f3e0b246822e0af5237026d5227842f6cf4907daa039848
7263550339c2a35f356bb874fb3a619b76f2d602064beada75049e7c2927a6dc

PDF

a8b6c06daeede6199e69f4cafd79299219def5bf913a31829dede98a8ad2aaa9
6fcd6aef0678d3c6d5f8c2cb660356b25f68c73e7ee24fbb721216a547d17ffa
ed72721837c991621639b4e86ffe0c2693ef1a545741b5513d204a1e3e008d8c
65edd9e1a38fd3da79c8a556eb2c7c595125ffec9f7483e2e6e189a08cc5d412
0a0375648bc9368bccfd3d657d26976d5b1f975381d1858d001404d807334058
e809582faccdd27337aa46b4a11dd11f5d0c7d7428ebdc8c895ea80777e4da5f
59d2433264d8ec9e9797918be3aa7132dbeb71e141f6e5c64c0d6f1cb4452934

ZIP

ac957ba4796f06c4bf0c0afb8674bbeb30eb95cef85bc68ced3ee1aa30e3acff
9296adb71bc98140a59b19f68476d45dbb38cc60b9e263d07d14e7178f195989
636c2ccffce7d4591b0d5708469070b839f221400b38189c734004641929ae05
31ffa4e3638c9e094275051629cc3ac0a8c7d6ae8415bbfcacc4c605c7f0df39
da3deea591b59b1a0f7e11db2f729a263439a05f3e8b0de97bbac99154297cea

Executable

e2269b38655a4d75078362856c16594e195cd647c56b8c55883b8e1286baa658
52632d9e24f42c4651cf8db3abc37845e693818d64ab0b11c235eddf8e011b2f
7200155f3e30dbbd4c4c26ce2c7bd4878ab992b619d80b43c0bd9e17390082fc
e516b102a2a6001eafb055e42feb9000691e2353c7e87e34ddaa99d7d8af16fd

a9ddd4e4d54336ce110fdc769ff7c4940f8d89b45ee8dc24f56fc3ea00c18873
a12d17cca038cdbf79b72356e5d20b17722c7b20bd2ee308601bac901890f3f4
b1ac2178c90c8eafd8121d21acbae7a0eb0cbc156d4a5f692f44b28856a23481
a6c1629b4450f713b02d24f088c4f26b0416c6a7924dcf0477425f3a67a2e3ff
3ce81c163ddedb132116cdf92aae197ced0b94f3fc3d1036f5c41b084a256a03
a19fdfc131e8fbe063289c83a3cdefb9fb9fb6f1f92c83b892d3519a381623db
db15f45f69f863510986fb2198a8a6b3d55d8ccc8a2ed4bb30bc27bdd1bf151c
bf1a7938f61a9905e1b151c7a5f925a2ce3870b7c3e80f6e0fc07715bdc258b7
f42c6949c6d8ecf648bacca08cde568f11ec2663221a97dae5fbf01218e8775a

---