Winos 4.0 Spreads via Impersonation of Official Email to Target Users in Taiwan

fortinet.com/blog/threat-research/winos-spreads-via-impersonation-of-official-email-to-target-users-in-taiwan

February 27, 2025

Article Contents

By Pei Han Liao | February 27, 2025
Affected Platforms: Microsoft Windows
Impacted Users: Microsoft Windows
Impact: The stolen information can be used for future attack
Severity Level: High

In January 2025, FortiGuard Labs observed an attack that used Winos4.0, an advanced malware framework actively used in recent threat campaigns, to target companies in Taiwan. Figure 1 shows an example of the attack chain. Usually, there is a loader that is only used to load the malicious DLL file, and the Winos4.0 module is extracted from the shellcode downloaded from its C2 server.

Figure 1: Attack flow
FortiGuard Labs Outbreak Alerts

Subscribe today to have threat alerts delivered to your inbox

Phishing

According to a report released in November 2024, Winos4.0 was distributed through gaming-related applications, however, it spread via an email masquerading as from Taiwan's National Taxation Bureau in the campaign in January 2025. The sender claimed that the malicious file attached was a list of enterprises scheduled for tax inspection and asked the receiver to forward the information to their company's treasurer.

Figure 2: Phishing mail
The attachment also masquerades as an official document from the Ministry of Finance. It asks the victim to download the attached list of enterprises slated for tax inspection. However, the list is a ZIP file containing malicious DLL for the next attack stage.

Figure 3: PDF file in the phishing email

lastbld2Base.dll and its shellcode

The files in the ZIP file are executed in the following sequence: 20250109.exe, ApowerREC.exe, and lastbld2Base.dll. 20250109.exe is a launcher originally used to execute the actual APowerREC.exe in ./app/ProgramFiles. The attacker created the same folder structure in the ZIP file and used a loader to replace ApowerREC.exe. The fake ApowerREC.exe does nothing but call a function imported from lastbld2Base.dll.

When an executable file is run, it loads all necessary DLL files and executes their entry functions. As a result, the DLLMain function of lastbld2Base.dll, where the malicious code is located, is loaded when the fake ApowerREC.exe is executed.

Figure 4: The entry point of the fake ApowerREC.exe
Lastbld2Base.dll decrypts its data to get the shellcode for the next stage. At the bottom of the shellcode are configurations, including the IP address of the C2 server, the name of the base registry key for the next stage, and flags for features for the current stage.

Figure 5: The configuration at the bottom of the shellcode
The optional features include permission evaluation, hiding the window of the current process, and anti-sandbox functions. If higher permission is needed in this attack, it tests the current permissions by opening the registry key HKEY_LOCAL_MACHINE\SOFTWARE and executing ApowerREC.exe as an administrator.

For the anti-sandbox function, it takes two screenshots within a two-second interval. If there are more than 20,000 different pixels in the second screenshot, which means a user is active on the computer, it performs its remaining tasks. Otherwise, it continues taking screenshot and compares it with the first one for at most one hour. After the optional features are run, it downloads the encrypted shellcode data and the Winos4.0 module from its C2 server. The encrypted data is written to HKEY_CURRENT_USER\B118D5E900008F7A, the base registry for configurations in the next stage, with a value name of “0”. After this, it decrypts the data to get the shellcode, followed by partially decrypted data of the module.

Shellcode from server

The new shellcode decrypts the data with another algorithm to get a DLL file and parses its export table to get the address of the only export function.

Figure 6: Data for the Winos4.0 module follows the shellcode

登录模块.dll(login module)

In this attack, the module from the C2 server creates eight threads to perform different tasks: MainThread, CloseWindow, Screenshot, Keylog, Clipboard, USB, ReadReg, and Anti-AV.

MainThread

Mutex: Global\MainThreadB118D5E900008F7A
The MainThread creates the remaining seven threads. In addition, it performs the following actions:

- Persistence
  If the parent process is service.exe, it drops its copy as C:\ProgramData\BITTS2.exe
- Deactivate screen saver
  It calls the following APIs with specific constants to ensure the computer stays active

API	Constant	Description
SystemParametersInfoW	SPI_SETSCREENSAVEACTIVE	Deactivates the screen saver
SetThreadExecutionState	ES_CONTINUOUS ES_AWAYMODE_REQUIRED ES_SYSTEM_REQUIRED	Enables the Away mode so the program keeps working while the computer appears to be sleeping
PowerSetRequest	PowerRequestDisplayRequired	The display remains on even if the computer is idle

Bypass UAC
Bypasses the UAC (User Account Control) prompt by changing the following registry key values into specific values:

Registry key: SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System **Value name**: ConsentPromptBehaviorAdmin **Value**: 0 **Description**: Allows the Consent Admin to perform an operation that requires elevation without consent or credentials.
Registry key: SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System **Value name**: PromptOnSecureDesktop **Value**: 0 **Description**: Disables secure desktop prompting

Execute DLL
It decrypts data stored in values of HKEY_CURRENT_USER\B118D5E900008F7A\PLUG\0\{key name}. The result can be written to a file named {key name}.dll or loaded in memory.

Figure 7: Encrypted data from the C2 server.

Collect user information
It collects the computer name, architecture, version, anti-virus software, video capture device, and timestamp.

CloseWindow

Mutex: Global\ClosewWindow
It calls the **EnumWindows** function to enumerate all visible windows to find the windows of kxecenter(Kingsoft Security) and HipsTray(Huorong). It checks the window's width to ensure it is the security prompt window. When the target window is found, it clicks the "Permit" button on the prompt window.

Figure 8: Huorong prompt window

Screenshot

Mutex: Global\ScreenShotB118D5E900008F7A
It takes screenshots of applications that contain the keywords stored in the value **picshotdata** of HKEY_CURRENT_USER\B118D5E900008F7A, and the screenshots are

saved to C:\ProgramData\B118D5E900008F7A\{keyword}\{Date}.
If **picshotdata** doesn't exist, this thread will not be executed.

Figure 9: An example of the folder structure

Keylog

Mutex: Global\KeylogB118D5E900008F7A
It keeps checking the value of the **KEYLOG** of HKEY_CURRENT_USER\B118D5E900008F7A. If the value is 1, it creates a mutex C:\ProgramData\B118D5E900008F7A\Regedit.log and starts recording the user's keystrokes and the contents in the clipboard. The data is written to C:\ProgramData\B118D5E900008F7A\Regedit.log.

Figure 10: An example of the Regedit.log.

Clipboard

Mutex: Global\ClipboardB118D5E900008F7A
It replaces keywords in the clipboard with the text stored in the registry value clipboarddata of HKEY_CURRENT_USER\B118D5E900008F7A. The value contains three properties: Mode, Expression, and Replace. When Mode is "Modify," Expression specifies the pattern to look for in the clipboard, and Replace specifies the replacement.
If clipboarddata doesn't exist, this thread will not be executed.

USB

Mutex: Global\UsbB118D5E900008F7A
It collects the names of connected USB devices every three seconds except for those that include the following keywords: mouse, keyboard, wlan, lenovo, and sanmsung (misspelling of samsung). If a new USB device is inserted or removed, it updates the device list with a Chinese annotation that means "USB device inserted" or "USB device removed."

Figure 11: Recording changes of the connected device list in Chinese.

ReadReg

It reads the value B118D5E900008F7A0 from HKCU\Console to get the shellcode and execute it every five seconds.

Anti-AV

First, it bypasses the UAC prompt by modifying the registry key values mentioned in MainThread. Then, it calls **GetTcpTable2** to obtain active TCP connections. If a TCP connection is owned by 360Safe, Kingsoft, or Huorong processes, it disables it.

Other Attack Chain

There are other attack chains used in this campaign.

Figure 12: Another attack flow
The查看10.exe(view10) is compiled from a Python script by Nuitka, and it loads Python311.dll, which is the malicious file. The shellcode from Python311.dll decrypts its data to get a DLL file that writes another shellcode to the registry value of **hrqnmlb{XXXXXX}** of the HKCU\Console\, and the shellcode is also saved as bb.jpg in C:\Users\Public\Download. The shellcode plays the same role as the shellcode from lastbld2Base.dll we mentioned above. However, its marker string is used by a version preceding the one described in a report released in November 2024.

Figure 13: The shellcode in the registry key and bb.jpg
Another point worth mentioning is that the DLL contains multiple snippets of shellcode that are identical to Figure 13 except for the C2 domain. While only 9010[.]360sdgg[.]com is used in this attack, other domains have been observed in different campaigns.

Figure 14: Multiple snippets of shellcode are found in the DLL file.
The 上线模块.dll(online module) is used to take screenshots of WeChat and the online bank, and the akagi.exe is a module of UACMe.

Conclusion

Winos4.0 makes good use of registry keys. The C2 server writes most configurations for optional features and encrypts data to the values of the base registry key and its subkeys. This provides the flexibility of optional features. However, it's also a good hint for forensic analysis. We can rebuild files from the data and perform further analysis. FortiGuard will continue monitoring these attack campaigns and providing 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
W32/Agent.7BBA!tr
W64/UACMe.O!tr
W64/ValleyRat.A!tr.spy

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 these products with up-to-date protections are protected.

The FortiGuard CDR (content disarm and reconstruction) service, which runs on both FortiGate and FortiMail, can disarm the malicious macros in the document.

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

43[.]137[.]42[.]254
206[.]238[.]221[.]60
206[.]238[.]221[.]240
124[.]156[.]100[.]172
206[.]238[.]221[.]244

Domain

1234[.]360sdgg[.]com
9001[.]360sdgg[.]com
9002[.]360sdgg[.]com
9003[.]360sdgg[.]com
9005[.]360sdgg[.]com
9006[.]360sdgg[.]com
9007[.]360sdgg[.]com
9009[.]360sdgg[.]com
9010[.]360sdgg[.]com
ffggssa-1329400280[.]cos[.]ap-guangzhou[.]myqcloud[.]com
fuued5-1329400280[.]cos[.]ap-guangzhou[.]myqcloud[.]com
0107-1333855056[.]cos[.]ap-guangzhou[.]myqcloud[.]com
rgghrt1140120-1336065333[.]cos[.]ap-guangzhou[.]myqcloud[.]com
hei-1333855056[.]cos[.]ap-guangzhou[.]myqcloud[.]com
chakan202501-1329400280[.]cos[.]ap-guangzhou[.]myqcloud[.]com
wrwyrdujtw114117-1336065333[.]cos[.]ap-guangzhou[.]myqcloud[.]com
fdsjg114-1336065333[.]cos[.]ap-guangzhou[.]myqcloud[.]com

sjujfde-1329400280[.]cos[.]ap-guangzhou[.]myqcloud[.]com
htrfe4-1329400280[.]cos[.]ap-guangzhou[.]myqcloud[.]com
0611-1333855056[.]cos[.]ap-guangzhou[.]myqcloud[.]com
twzfw[.]vip

Phishing mail

36afc6d5dfb0257b3b053373e91c9a0a726c7d269211bc937704349a6b4be9b9
0e3c9af7066ec72406eac25cca0b312894f02d6d08245a3ccef5c029bc297bd2
67395af91263f71cd600961a1fd33ddc222958e83094afdde916190a0dd5d79c
f4d3477a19ff468d234a5e39652157b2181c8b51c754b900bcfa13339f577e7c
c9a8db23d089aa71466b4bde51a51a8cfdcc28e8df33b4c63ce867bd381e5fe5

PDF

e2b75baeb7ed21fb8f27984f941286770d1c3c0b60fce8d7fa5b167bd24ba6dc
dffbeefc632b20d2ef867553684e9971ab76e1223e743604a5275713423b6168
20c34b5f0983021414b168913c3da267caf298d8f0f5e3ec0ce97db5f4f48316
6c33715a14fdc917b5b09b6e1b5dad07bb769493eafbf7ca1023830b4059e003
75a4d75c35724140149c9c5056c1bcbd328bbe1e5d1d1ef34205ed5442d2b348
fed394a3653b7c6fcc1b277eda6e18eb0983a7e024be5b51e5188b3cfb9512e8
a067d848f099e6d1e465f9761a5b85392d550303bfa75fac920d444fd980c949
c55757075259fa4be6941dd273c4a4a2fcc29e6ba427dec124b25b299b3505fe
64a876e6cb3cf3122febc84a00ec3e0740c054cff955164971c470e1b5e5f1bb
d4ac82de8dda9796579cd8ea0f84b43c7a980cdb0e9cdb8abe8981a2d215ed2f
(20c34b5f0983021414b168913c3da267caf298d8f0f5e3ec0ce97db5f4f48316 Corrupt)

DLL

268c72f5482374660a132d1b91cac0c04b4724a214db4f052eb421e36c282921
0a4bbb998bd3a3bcc72cf759689a5656dc74590b731d0affbfc317cf484ed28b
79c64d2e77acdbcdbd35cbb29497941335d7e3ab6ebb474064f095e745f0d643
7f22305679e46e1fd5043beb136108197c0921643ce0d680f990a3018ade485b
594d907855d35ee7689a568e4ac43e4e0ed90de047d91b0253ef79da71ecbc08
1f3b041eee1ece8cf6aa5c742aeb8c0ac2266cccecca7888772509227c4f8669
514933468ac1dd9f7db4e2693f1be7f84deb35c33f8f9934fad32caaae9ef611
7a5b26f6dd7b8e0d648e9804ec932603b7d7a5f76c7a8c537ab0c2be54f51fa9
8b1b9a789136ca3abe25938204845c351aaf0c97c0708ade8d4d8ba4ded95ba7
1ad1f2eec961bc7a35abeac486f843b7caece0929b13f1dab47fbdc0406ac4e3
4c1ea827713f1eb57cc0e8e9d171d4e21d116f846b174bc05114eef5674c9653
1a342426d59e7fdc4abfb74c2225f68382172e03b0f8d496a57ae647411f0fbd
2ce73cbfab0beb3663c0151ba7c310e4dbf69f295d8a18114435506483d774ac

0a4bbb998bd3a3bcc72cf759689a5656dc74590b731d0affbfc317cf484ed28b
514933468ac1dd9f7db4e2693f1be7f84deb35c33f8f9934fad32caaae9ef611
76ac08358f230bca3e8b8448b3c177094aeac25402b929f5f73869ec77173a44