CrazyHunter: The Rising Threat of Open-Source Ransomware

Archived: 2026-04-05 17:20:15 UTC

Background

A ransomware attack on the Mackay Memorial hospital in Taiwan is the latest example of a growing number of incidents revolving around publicly available, offensive tools and code that threat actors are utilizing. The ransomware encryptor used in this incident, dubbed “CrazyHunter”, was built using a ransomware builder called “Prince Ransomware” which was publicly available on GitHub. WithSecure has observed a growing number of actors employing this specific ransomware builder in ransomware attacks. There are a number of ‘lone wolf’ ransomware events that do not seem to use ransomware-as-a -service, affiliate models [read about that here]. As such, these can often be under-reported as we – as an industry – tend to focus on ‘big game’, more productive and attributable ransomware “franchises”. The purpose of this blog is to provide some technical analysis into the Prince Ransomware builder, and the tactics, techniques and procedures (TTP) behind the Mackay Memorial Hospital – and likely other Taiwanese – incidents.

Summary of the Incident

As noted in the report by CMMedia, the incident at Mackay Memorial Hospital Taiwan began on 9th February 2025. The threat actor began by infecting a small number of computers, probably to gauage the hospital's network defence. Upon seeing no or limited security, the threat actor continued their attack, laterally moving across the entire hospital network before detonating the ransomware encryptor. This resulted in the encryption of over 600 devices across two district branches, Taipei and Tamsui. The encryption of files caused key systems to crash and prevented staff access to patient data.

The initial point of entry was reported as a USB device inserted into a computer within the network (reportedly by a staff member). A physical initial access vector (IAV) is relatively rare in ransomware incidents, however, there is some precedent of pre-infected USB devices propagating malware. Reports of this incident did not state what the first stage malware dropped by the USB device was, and WithSecure has been unable to ascertain it.

WithSecure was able to detect a portion of the malware artifacts on VirusTotal that were likely used to conduct the attack, allowing for further analysis on the tools deployed in this incident. The artifacts were bundled in a file called “bb2.zip” which was uploaded to the platform multiple times, twice from Taiwan.

The file called “bb2.zip” which was dropped in the “C:\Users\Public” directory, contained the following files:

File Name	Description
bb.exe	Shellcode loader which loads “crazyhunter.sys”

crazyhunter.exe	A ransomware encryptor built with “Prince Ransomware” builder
crazyhunter.sys	A shellcode binary file based on “crazyhunter.exe”
file.exe	A custom exfiltration tool
go.exe	A defence evasion tool
go2.exe	A defence evasion tool
go3.exe	A ransomware encryptor built with “Prince Ransomware” builder, same as “crazyhunter.exe”
gpo.exe	SharpGPOAbuse tool used for lateral movement
ru.bat	A batch script file used to start the encryption process
zam64.sys	A vulnerable Zemana Anti-Logger kernel driver

Artifact Analysis

Overview

A batch script, “ru.bat”, found in the malware artifacts, was almost certainly used by the threat actor to automate the execution of several malicious actions. The script was not obfuscated in any way, and seeks to perform the following actions:

- Run “go2.exe”
- Run “go.exe”
- Run “go3.exe”
- Run “av-1m.exe” if “go.exe” is not running
- Run “bb.exe” and pass the driver file “crazyhunter.sys” as an argument
- Run “crazyhunter.exe” if “bb.exe” is not running

```
@echo off

start C:\Users\Public\go2.exe

timeout /t 10 /nobreak > nul

start C:\Users\Public\go.exe

timeout /t 10 /nobreak > nul

start C:\Users\Public\go3.exe


tasklist /FI "IMAGENAME eq go.exe" 2>NUL | find /I "go.exe" > NUL
if %errorlevel% equ 0 (
    echo go.exe is running.
) else (
    start C:\Users\Public\av-1m.exe
)

timeout /t 10 /nobreak > nul


start C:\Users\Public\bb.exe -f C:\Users\Public\crazyhunter.sys

timeout /t 60 /nobreak > nul

tasklist /FI "IMAGENAME eq bb.exe" 2>NUL | find /I "bb.exe" > NUL
if %errorlevel% equ 0 (
    echo bb.exe is running.
) else (
    start C:\Users\Public\crazyhunter.exe
)
```

Figure 1. File content of “ru.bat” batch script

Defense Evasion

The threat actor employed a frequently used “Bring Your Own Vulnerable Driver” (BYOVD) technique to disable security products on the systems. This is becoming increasingly common in ransomware attacks. This method allows the execution of malicious code with kernel-level privilege by exploiting signed and legitimate drivers with known vulnerabilities. In this instance, “go2.exe” and “go.exe” are malware written in Go programming language designed to load a vulnerable version of Zemana Anti-Logger kernel driver, “zam64.sys”. This allows the termination of security products, with “go2.exe” targeting Windows Defender and “go.exe” targeting both Windows Defender and Trend Micro products. The usage of two executables for this purpose may suggest an attempt by the threat actor to ensure the termination of the security products. However, it is also realistically possible that the threat actor is low skilled and unsure of what they are doing, relying on multiple tools to achieve their goal. The exploitation of Zemana vulnerable drivers is similar to the Terminator EDR tool sold by a Russian threat actor, which loads the same vulnerable version of Zemana Anti-Logger kernel driver to disable security products.

go2.exe

```
.data:00000000005672A0 off_5672A0      dq offset aSecurityhealth
.data:00000000005672A0                                          ; DATA XREF: .data:main_avlist↑o
.data:00000000005672A0                                          ; "SecurityHealthService.exe"
.data:00000000005672A8 ; .data:00000000005672A8
.data:00000000005672B0                 dq offset aSmartscreenExe ; "smartscreen.exe"
.data:00000000005672B8 ; .data:00000000005672B8
.data:00000000005672C0                 dq offset aMssenseExe   ; "MsSense.exe"
.data:00000000005672C8 ; .data:00000000005672C8
.data:00000000005672D0                 dq offset aMsmpengExe   ; "MsMpEng.exe"
```

go.exe

```
.data:00000000005674A0 off_5674A0      dq offset aSecurityhealth
.data:00000000005674A0                                          ; DATA XREF: .data:main_avlist↑o
.data:00000000005674A0                                          ; "SecurityHealthService.exe"
.data:00000000005674A8 ; .data:00000000005674A8
.data:00000000005674B0                 dq offset aMsmpengExe   ; "MsMpEng.exe"
.data:00000000005674B8 ; .data:00000000005674B8
.data:00000000005674C0                 dq offset aEndpointbaseca ; "EndpointBasecamp.exe"
.data:00000000005674C8 ; .data:00000000005674C8
.data:00000000005674D0                 dq offset aPccntmonExe  ; "PccNTMon.exe"
.data:00000000005674D8 ; .data:00000000005674D8
.data:00000000005674E0                 dq offset aPccntExe     ; "PccNt.exe"
.data:00000000005674E8 ; .data:00000000005674E8
.data:00000000005674F0                 dq offset aNtrtscanExe_0 ; "Ntrtscan.exe"
.data:00000000005674F8 ; .data:00000000005674F8
.data:0000000000567500                 dq offset aNtrtscanExe  ; "NTRTScan.exe"
.data:0000000000567508 ; .data:0000000000567508
.data:0000000000567510                 dq offset aMssenseExe   ; "MsSense.exe"
.data:0000000000567518 ; .data:0000000000567518
.data:0000000000567520                 dq offset aNissrvExe    ; "NisSrv.exe"
.data:0000000000567528 ; .data:0000000000567528
.data:0000000000567530                 dq offset aTmccsfExe    ; "TmCCSF.exe"
.data:0000000000567538 ; .data:0000000000567538
.data:0000000000567540                 dq offset aTmbmsrvExe   ; "TMBMSRV.exe"
.data:0000000000567548 ; .data:0000000000567548
.data:0000000000567550                 dq offset aTmlistenExe  ; "TmListen.exe"
.data:0000000000567558 ; .data:0000000000567558
.data:0000000000567560                 dq offset aCntaosmgrExe ; "CNTAoSMgr.exe"
.data:0000000000567568 ; .data:0000000000567568
.data:0000000000567570                 dq offset aSensetvmExe  ; "SenseTVM.exe"
.data:0000000000567578 ; .data:0000000000567578
.data:0000000000567580                 dq offset aCetasvcExe   ; "CETASvc.exe"
.data:0000000000567588 ; .data:0000000000567588
.data:0000000000567590                 dq offset aWscommunicator ; "WSCommunicator.exe"
.data:0000000000567598 ; .data:0000000000567598
.data:00000000005675A0                 dq offset aDsagentExe   ; "dsagent.exe"
.data:00000000005675A8 ; .data:00000000005675A8
.data:00000000005675B0                 dq offset aSupportconnect ; "SupportConnector.exe"
.data:00000000005675B8 ; .data:00000000005675B8
.data:00000000005675C0                 dq offset aAvguardExe   ; "avguard.exe"
.data:00000000005675C8 ; .data:00000000005675C8
.data:00000000005675D0                 dq offset aAvshadowExe  ; "avshadow.exe"
.data:00000000005675D8 ; .data:00000000005675D8
.data:00000000005675E0                 dq offset aAvgntExe     ; "avgnt.exe"
.data:00000000005675E8 ; .data:00000000005675E8
.data:00000000005675F0                 dq offset aAviraSystrayEx ; "Avira.Systray.exe"
.data:00000000005675F8 ; .data:00000000005675F8
.data:0000000000567600                 dq offset aTrtscanExe   ; "trtscan.exe"
.data:0000000000567608 ; .data:0000000000567608
.data:0000000000567610                 dq offset aMpcmdrunExe  ; "MpCmdRun.exe"
```

Figure 2. Targeted antivirus services

Although the file “av-1m.exe” was not included in the malware artifacts, based on the file name and the check for whether “go.exe” is running, it can be assumed that it was used to bypass AV as well.

Encryption

The threat actor used an open-source ransomware builder (a tool to automate the creation of ransomware) written in the Go programming language called “Prince Ransomware”, which was freely available on GitHub. This is no longer available on Github, however it can be retrieved from a snapshot of the builder repository. The builder utilizes both ChaCha20 and ECIES (Elliptic Curve Integrated Encryption Scheme) cryptography to encrypt files securely, making it more difficult to recover the encrypted files. This works by generating a unique ChaCha20 key and nonce for each file. The file is then encrypted using a pattern where 1 byte is encrypted, followed by 2 bytes left unencrypted. The ChaCha20 key and nonce are then encrypted using an ECIES public key and added to the start of file. The encrypter loops through all drives and directories on the system, ignoring blocklisted files, directories and extensions, to perform the encryption and drop the ransom note. The “CrazyHunter” encrypter was found in the malware artifacts as “go3.exe” and “crazyhunter.exe”, which had the same file hashes.

Since the builder was freely accessible and effective, other similar ransomware samples utilizing this builder have been found on VirusTotal. Other variants includes, Black (Prince), Wenda, UwU, and many others – in our opinion, also under-reported. The only difference between these variants lies in the file extension and the ransom note dropped, which can be customized within the configuration file of the builder to fit the needs of the threat actors. The ransom note dropped by “CrazyHunter” is only slightly modified, which gives an indication as to how ready ‘out of the box’ this ransomware code is. The threat actor simply needs to edit a single configuration file to essentially deploy a “fresh” ransomware brand.

```
---------- Prince Ransomware ----------
Your files have been encrypted using Prince Ransomware!
They can only be decrypted by paying us a ransom in cryptocurrency.

Encrypted files have the .prince extension.
IMPORTANT: Do not modify or rename encrypted files, as they may become unrecoverable.

Contact us at the following email address to discuss payment.
example@airmail.cc
---------- Prince Ransomware ----------
```

Figure 3. Default ransom note template by Prince Ransomware

```
---------- Hunter Ransomware ----------
Your files have been encrypted using Hunter Ransomware!
They can only be decrypted by paying us a ransom in cryptocurrency.

Encrypted files have the .hunter extension.
IMPORTANT: Do not modify or rename encrypted files, as they may become unrecoverable.

Contact us at the following email address to discuss payment.
attack-tw1337@proton.me
---------- Hunter Ransomware ----------
```

Figure 4. Ransom note left by CrazyHunter

Another file found in the malware artifacts was “bb.exe”, which loads a binary shellcode file called “crazyhunter.sys”. Analysis of the binary shellcode reveals the use of a tool called Donut, which generates shellcode from PE files. In this case, the standalone “CrazyHunter” encrypter (go3.exe and crazyhunter.exe) mentioned above was converted to shellcode and stored as “crazyhunter.sys”, which is then loaded into memory

using “bb.exe”. This technique was likely used to evade detection from security products in case the standalone encrypter was detected. The resulting encryption and ransom note would be the same as those produced by the standalone encrypter (“go3.exe” and “crazyhunter.exe”).

Lateral Movement

Based on the malware artifact, “gpo.exe”, which is SharpGPOAbuse, an open-source offensive tool available on GitHub, it can be said that the threat actor used it to spread the ransomware to other computers on the network. This is performed by exploiting the user’s edit rights on a Group Policy Object (GPO) to compromise the objects controlled by that GPO. The threat actor can then setup malicious scripts configured to run automatically during system startup, user logon, or at a scheduled time.

Additional Tooling

One of the artifacts called “file.exe” was particularly interesting. Further analysis revealed that it is a tool capable of hosting/setting up the victim’s machine as a file server or to monitor for files with specific extensions in the specified directory (default is current directory), including subdirectories. When set to function as a file server, it will open the specified port (default is 9999) at the specified directory (default is current directory). This can then be accessed on “localhost:<port>”. Additionally, when configured to monitor files, it will periodically scan the specified directory for files with the monitored extensions and delete any matching files. Based on its capabilities, it is almost certain that this tool is used for data exfiltration and to prevent any recovery actions by monitoring and deleting specific file extensions like .exe or .ps1.

```
PS C:\Users\Public > .\file.exe -help
Usage of C:\Users\Public\file.exe:
  -d string
        path (default "C:\\Users\\Public\\")
  -e string
        monitor suffix support .asp,.php,.jsp (default ".asp")
  -f string
        exclude (default "1.asp")
  -func string
        fileserver,monitor
  -port int
        fileserver listen port (default 9999)
  -t int
        1000=1S (default 1000)
  -white
        true is Whitelist mode false is Blacklist mode
```

Figure 5. Command line arguments of “file.exe”

Other Incidents

There is limited information available regarding CrazyHunter, but their attacks first started in early 2025. At the time of writing, they have been involved in multiple incidents, mostly targeting hospitals and some industrial sectors in Taiwan. This pattern suggests that the actor might be a local actor. The use of USB device as the initial

access vector (IAV) in Mackay Memorial Hospital incident further indicates that the threat actor is likely based in Taiwan. As a result, the likelihood of this group targeting other geographical regions appears to be relatively low at the time of writing.

Mitigation

To mitigate the risk of ransomware attacks similar to the one at Mackay Memorial Hospital, organizations should implement strong endpoint protection, regularly update antivirus software, and secure against untrusted USB devices by disabling ports where possible and scanning them for malware. Proper network segmentation and access controls can limit the spread of malware. Continuous monitoring and auditing of network traffic and system logs can help identify and address potential weaknesses early on.

Conclusion

The incident at Mackay Memorial Hospital in Taiwan showcased how accessible and effective publicly available tools and malwares can be, enabling a wide range of threat actors to perform cyberattacks. Notably, this includes multiple tools like SharpGPOAbuse and Donut, as well as the Prince Ransomware builder, used specifically in this incident. Such readily available resources greatly lower the barrier for ransomware actors, allowing even those with limited technical expertise to launch complex attacks.

Furthermore, attributing such attacks to a specific ransomware affiliate or collective is particularly challenging due to the widespread availability and use of these open-source tools enabling lone-wolf attackers. Throughout 2024, WithSecure could not attribute 38% of its ransomware incidents to an identifiable Ransomware-as-a-service franchises an(other) indication of the increase in lone-wolf ransomware events enabled by readily available offensive tooling. Moreover, there are numerous other cases of leaked ransomware enablers being deployed, notably leaked builders like Lockbit and Babuk, which WithSecure often see deployed by ransomware actors not affiliated to any particular RaaS.

The initial access vector (IAV) for this incident was reportedly a USB device, which is uncommon in ransomware incidents. The physical nature of this IAV, combined with the use of open-source tools and ransomware, and the absence of links to other known attacks, suggests that this might be the work of a local “lone wolf” ransomware threat actor targeting businesses and organization exclusively in Taiwan. However, this remains inconclusive due to the limited data available at the time of investigation and writing. Whether the incident was accidental or involved a staff member, it highlights the importance of implementing physical security measures for networked devices and data ports in public buildings like hospitals.

TTP

Tactic	Techniques	Description
Execution	T1059.003 – Command and Scripting Interpreter: Windows	The threat actor used a batch script to automate the execution of malicious actions.

	Command Shell	
Persistence	T1547 – Boot or Logon AutoStart Execution	The threat actor used SharpGPOAbuse to setup malicious script configured to run during startup or user logon.
Privilege Escalation	T1068 – Exploitation for Privilege Escalation	The threat actor used two executables that load a vulnerable driver to exploit permission to run malicious code in kernel mode.
	T1484.001 – Domain or Tenant Policy Modification: Group Policy Modification	The threat actor used SharpGPOAbuse to modify the GPO and setup malicious script configured to run during startup or user logon on the computers within the network.
Defense Evasion	T1562.001 – Impair Defense: Disable or Modify Tools	The threat actor used two executables that loads a vulnerable driver to disable EDR and AV tools.
	T1211 – Exploitation for Defense Evasion	The tool used to disable EDR and AV tools loads a vulnerable driver to execute malicious code in kernel mode.
Discovery	T1083 – File and Directory Discovery	The threat actor uses “file.exe” to perform file and directory discovery to identify files to exfiltrate.
Lateral Movement	T1570 – Lateral Tool Transfer	The threat actor used “file.exe” fileserver to transfer the malicious tools and executable within the network.
Collection	T1005 – Data from Local System	The threat actor used “file.exe” host/setup a fileserver for accessing outside the network.
Exfiltration	T1048 – Exfiltration Over Alternative Protocol	The threat actor used “file.exe” host/setup a fileserver that can be used to exfiltrate data

Impact	T1486 – Data Encrypted for Impact	The ransomware encrypts file using ChaCha20 and ECIES cryptography which makes it difficult to recover the files

IOC

File Name	**Sha256**
bb.exe	2cc975fdb21f6dd20775aa52c7b3db6866c50761e22338b08ffc7f7748b2acaa
crazyhunter.exe	f72c03d37db77e8c6959b293ce81d009bf1c85f7d3bdaa4f873d3241833c146b
crazyhunter.sys	5316060745271723c9934047155dae95a3920cb6343ca08c93531e1c235861ba
file.exe	14359f54d49799c713c2a8cc0c19a88392a0c6ad2c383494023008326cd0ba15
go.exe	754d5c0c494099b72c050e745dde45ee4f6195c1f559a0f3a0fddba353004db6
go2.exe	983f5346756d61fec35df3e6e773ff43973eb96aabaa8094dcbfb5ca17821c81
go3.exe	f72c03d37db77e8c6959b293ce81d009bf1c85f7d3bdaa4f873d3241833c146b
gpo.exe	512f785d3c2a787b30fa760a153723d02090c0812d01bb519b670ecfc9780d93
ru.bat	d1081c77f37d080b4e8ecf6325d79e6666572d8ac96598fe65f9630dda6ec1ec
zam64.sys	2bbc6b9dd5e6d0327250b32305be20c89b19b56d33a096522ee33f22d8c82ff1
bb2.zip	bdfc66266a2a19fc3d5dccef3eefe4c0ee928ba5b7abad60bc320218b2082fea

---