Trellix

The Ghost in the Machine: Unmasking CrazyHunter's Stealth Tactics

By Aswath A · January 6, 2026

CrazyHunter ransomware has emerged as a significant and concerning threat, highlighting the increasing sophistication of cybercriminal tactics. Trellix has been actively tracking this ransomware since its initial appearance, noting its rapid development and growing prevalence. The ransomware executable is a fork of the Prince ransomware, which surfaced in mid-2024. It has introduced notable advancements, particularly in network compromise techniques and anti-malware evasion.

This blog provides an in-depth analysis of CrazyHunter ransomware and its attack flow.

Overview

CrazyHunter, a Go-developed ransomware, employs advanced encryption and delivery methods targeted against Windows-based machines. It uses a data leak site to

publicize victim information.

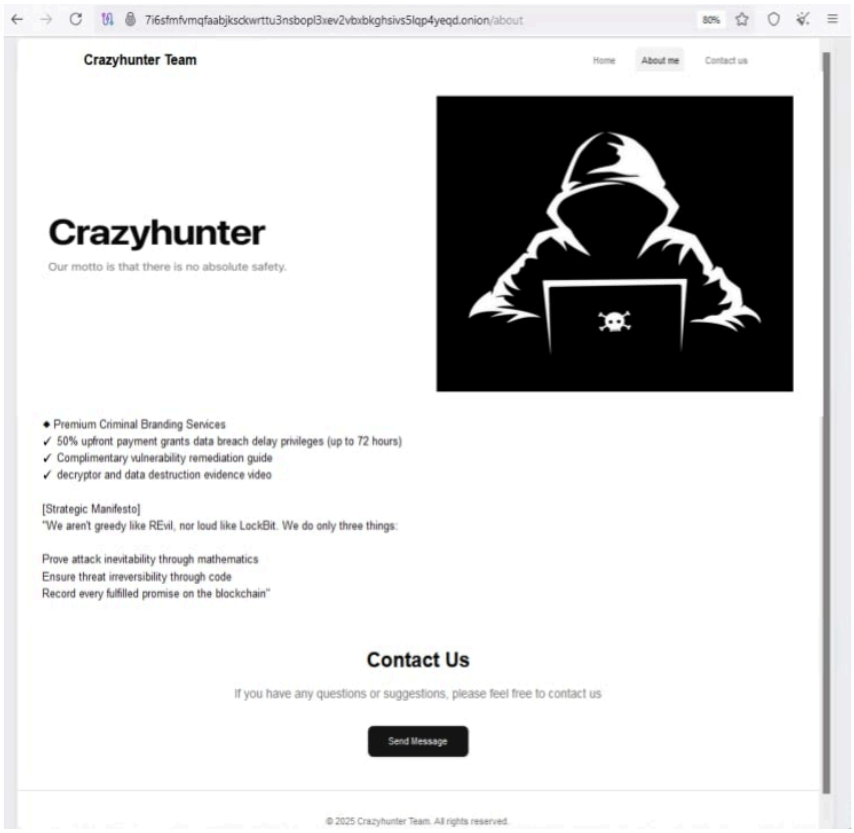

Figure 1: Attacker's data leak site

CrazyHunter ransomware has primarily affected Taiwan, comprising six targeted companies.

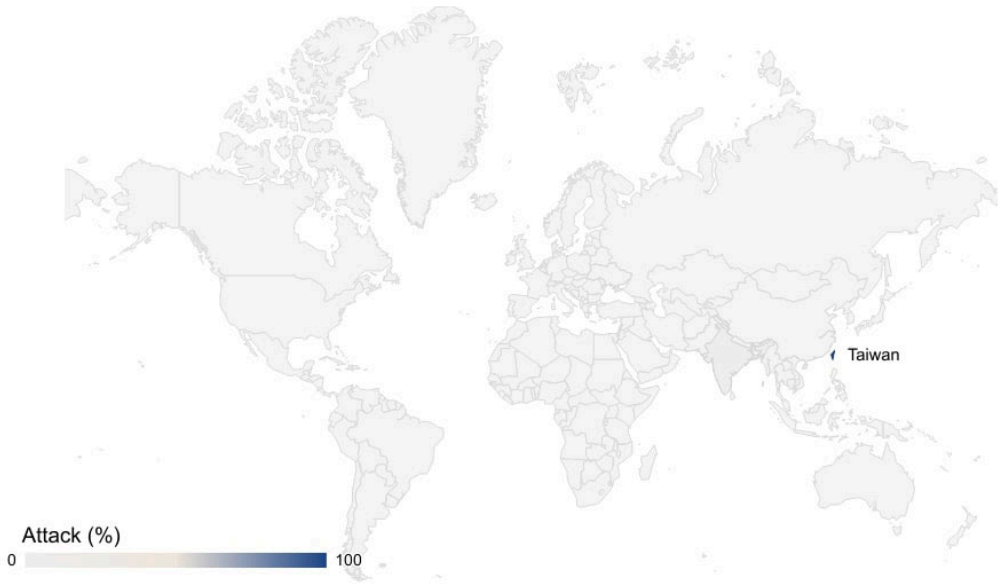

Figure 2: CrazyHunter ransomware global activity rate

According to available information, the primary industry targeted by CrazyHunter ransomware is the healthcare sector, with repeated attacks on hospitals in Taiwan. This preference is likely due to the critical nature of healthcare services, where vast amounts of sensitive patient data are held by these organizations and downtime can have severe consequences.

Victimology

The primary targets of the CrazyHunter ransomware have been companies in Taiwan, with six organizations known to be compromised. The attackers maintain a data leak site where they publicize information about their victims, particularly those who do not cooperate.

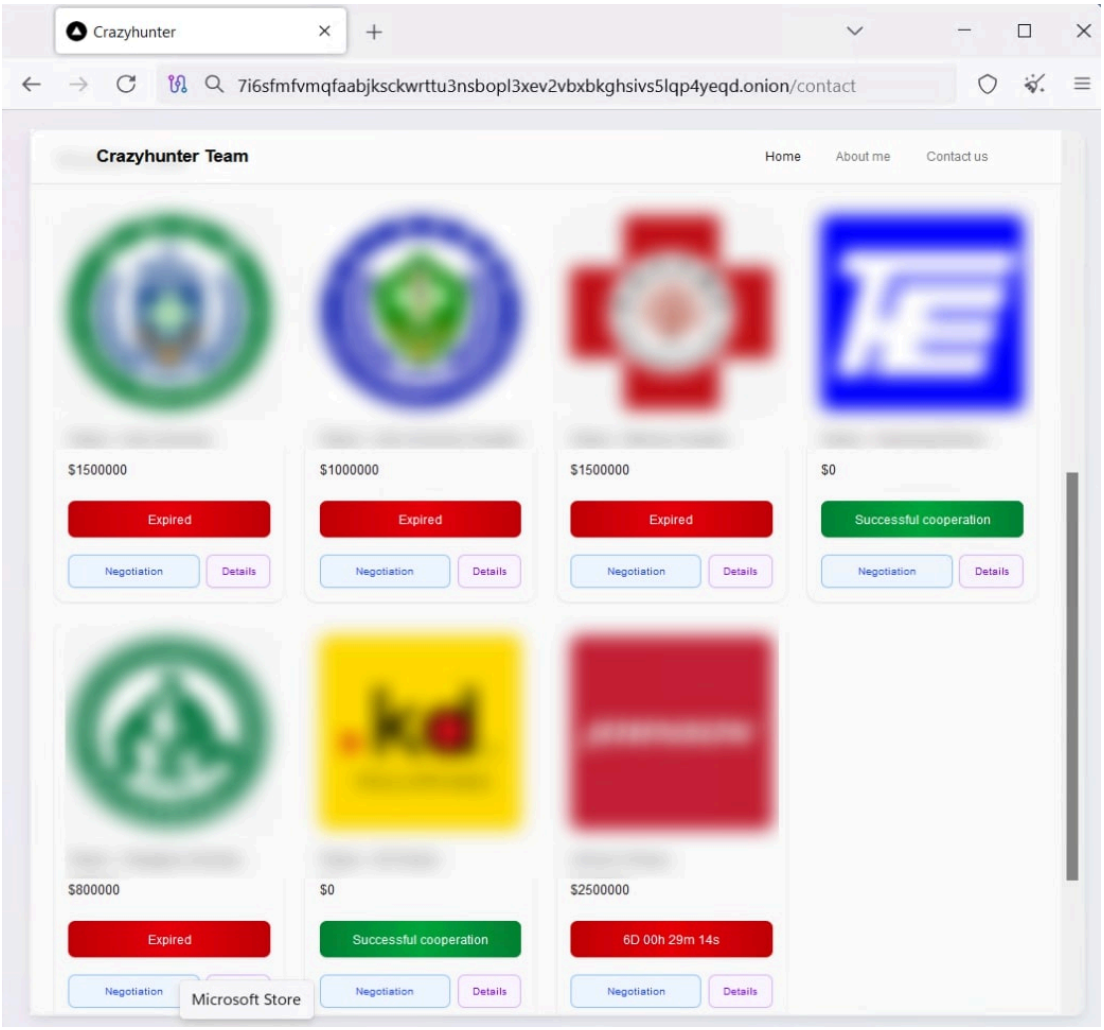

Figure 3: The victimology page from the CrazyHunter site

Attack lifecycle: A step-by-step descent into chaos

CrazyHunter's attack methodology is ruthlessly efficient, demonstrating a deep understanding of enterprise network vulnerabilities. The typical attack progresses through the following stages:

1. **Initial compromise: Exploiting the weakest link**
   The initial compromise often involves exploiting weaknesses in an organization's Active Directory (AD) infrastructure, frequently by leveraging weak passwords on domain accounts.
2. **Lateral movement and propagation: Rapid network domination**
   Once initial access is gained, attackers employ techniques for lateral movement and propagation. A key method observed in CrazyHunter attacks is the use of SharpGPOAbuse to distribute the ransomware payload through Group Policy

Objects (GPOs). This allows the malware to spread rapidly across the network to multiple systems. Attackers also leverage compromised AD credentials to facilitate this propagation.

3. **Privilege escalation: Bypassing security defenses**
   To establish control and dismantle security defenses, CrazyHunter leverages advanced privilege escalation tactics. A standout method is the bring-your-own-vulnerable-driver (BYOVD) approach. By weaponizing a modified Zemana anti-malware driver (zam64.sys), the attackers elevate their privileges, effectively bypassing security controls that would otherwise prevent them from succeeding.
4. **Encryption and ransom: Data hostage and financial extortion**
   The culmination of these stages is the encryption process, where files across the targeted network are encrypted, rendering them inaccessible. Following encryption, a ransom demand is issued, requiring payment for the decryption keys.

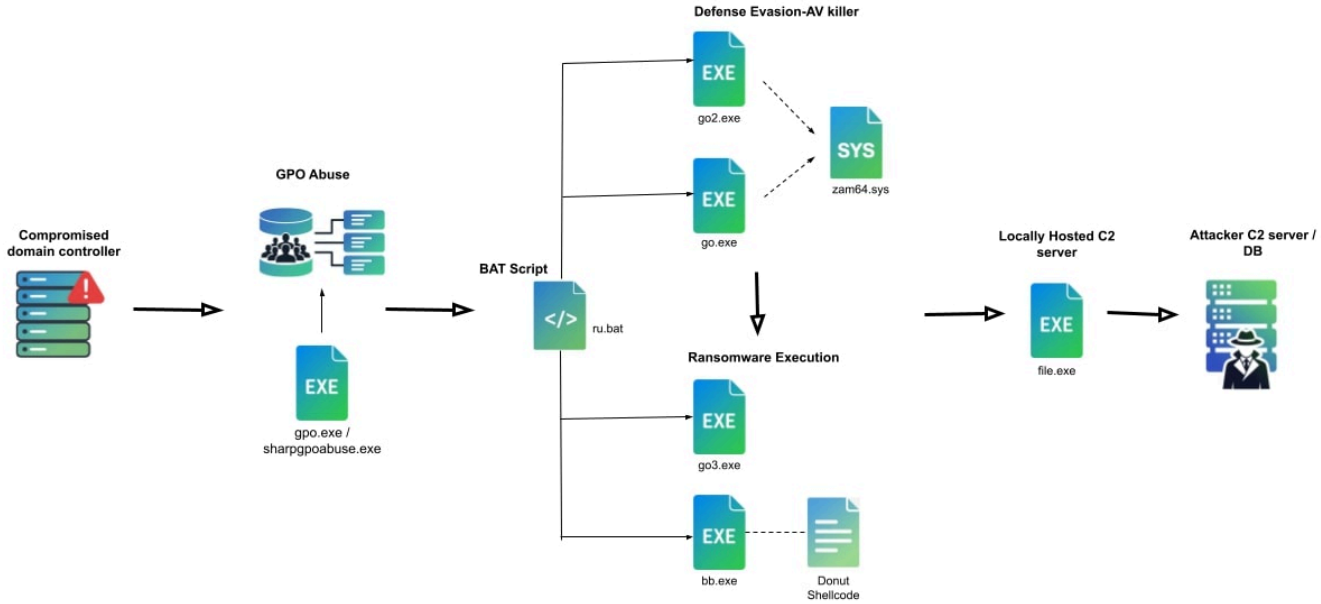

Figure 4: Attack flow overview

Technical analysis

Initial access

A concerning trend in modern cyberattacks involves multistage operations where initial actions are strategically designed to weaken or eliminate security measures before the primary malicious payload is executed. Examining a specific attack flow, as depicted in a particular batch script, reveals a deliberate and sophisticated approach to compromising systems. This analysis will dissect the script's actions, the underlying technical mechanisms employed to disable anti-malware software, and the context surrounding the final deployment of the CrazyHunter ransomware.

```
@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-lm.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 5: BAT script

Decoding the attack tree

Let's break down the CrazyHunter ransomware process tree shown in Figure 6.

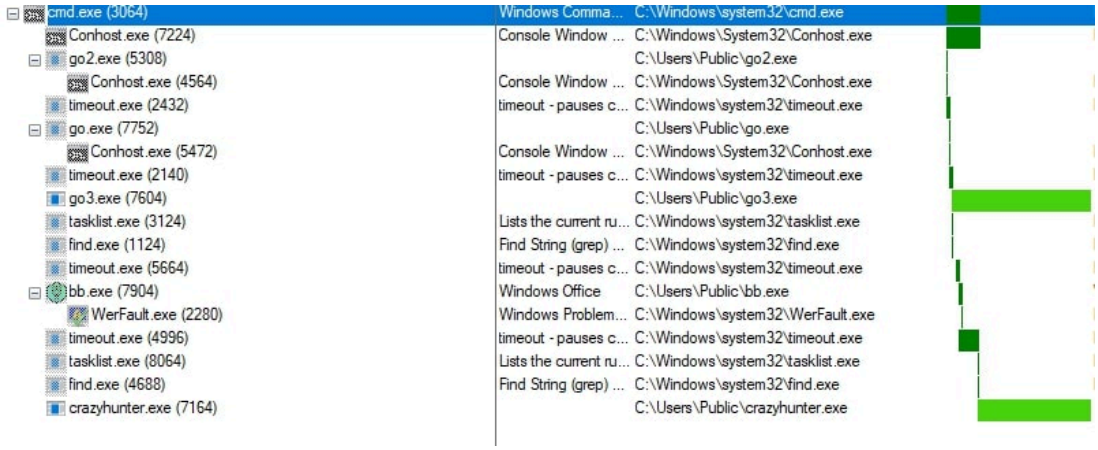

Figure 6: Process Tree

1. **Ignition point: ru.bat script execution**
   This batch script acts as the orchestrator for the CrazyHunter ransomware deployment and chain-launches all of the ransomware components.
2. **Silent takedown: Disabling security first**
   Almost immediately, the ru.bat script launches go2.exe, which is quickly followed by go.exe. These are the initial "AV Killers," designed to neutralize security software *before* the main payload runs. Notice the timeout.exe processes; these are deliberate pauses, likely used to evade detection or allow previous steps time to work.

3. **Executing the ransomware: go3.exe**
   Next up is go3.exe, identified as the primary CrazyHunter ransomware executable. This is the core component responsible for the destructive file encryption routine.
4. **Conditional anti-virus evasion**
   If go.exe fails to run, the script tries to deploy av-1m.exe, suggesting an attempt to disable or hinder anti-malware software. Av-1m.exe is not available in the public domain and may be a future development component included. The attacker may be attempting to weaken the system's defenses if an earlier stage component (go.exe) fails to maintain its presence. Av-1m.exe also serves as a fallback mechanism to ensure the AV software is terminated properly.
5. **Memory morph: The donut loader (bb.exe)**
   The script launches bb.exe, a "Donut Loader." This isn't the encryptor itself, but a tool designed to load shellcode directly into memory. This "fileless" technique helps evade detection that relies on scanning malicious files on disk.
6. **Plan B: The backup encryptor**
   CrazyHunter.exe is the *backup* ransomware executable. It's a fail-safe if the primary payload (go3.exe) or the shellcode injection via bb.exe fails, this ensures the encryption still has a chance to succeed.

Executable	Purpose
go2.exe	AV killer (Initial stage component)
go.exe	AV killer (Initial stage component)
go3.exe	Primary CrazyHunter ransomware executable (File encryption)
av-1m.exe	Malicious executable for disabling or interfering with anti-virus software (Likely)
bb.exe	Donut Loader
crazyhunter.sys	CrazyHunter ransomware donut shellcode
crazyhunter.exe	Backup CrazyHunter ransomware executable (backup execution method)

Table 1: Executables Summary

Tooling: SharpGPOAbuse

SharpGPOAbuse is a publicly available .NET application written in C# designed to exploit Group Policy Objects' (GPOs) inherent management capabilities within an

Active Directory environment. It allows attackers to manipulate GPOs to achieve various malicious objectives, such as deploying malware, creating rogue user accounts, or modifying security settings.

The CrazyHunter team used the tool to deploy ransomware and malware payloads via Group Policy Objects, enabling them to distribute the malware to a large number of computers across the victim's network.

Defense evasion

AV killer - Go.exe and go2.exe

The core of the anti-malware disablement mechanism lies in the functionality of go.exe and go2.exe. These executables exploit a vulnerable driver, zam64.sys, to achieve their objective.

The process involves two key steps:

1. Registering the driver.
2. Enumerating and terminating processes associated with antivirus software.

This technique falls under **Bring-Your-Own-Vulnerable-Driver (BYOVD)** attacks, where attackers exploit a legitimate but vulnerable driver, zam64.sys version 2.18.371.0, signed by trusted vendors like Zemana, to perform malicious actions.

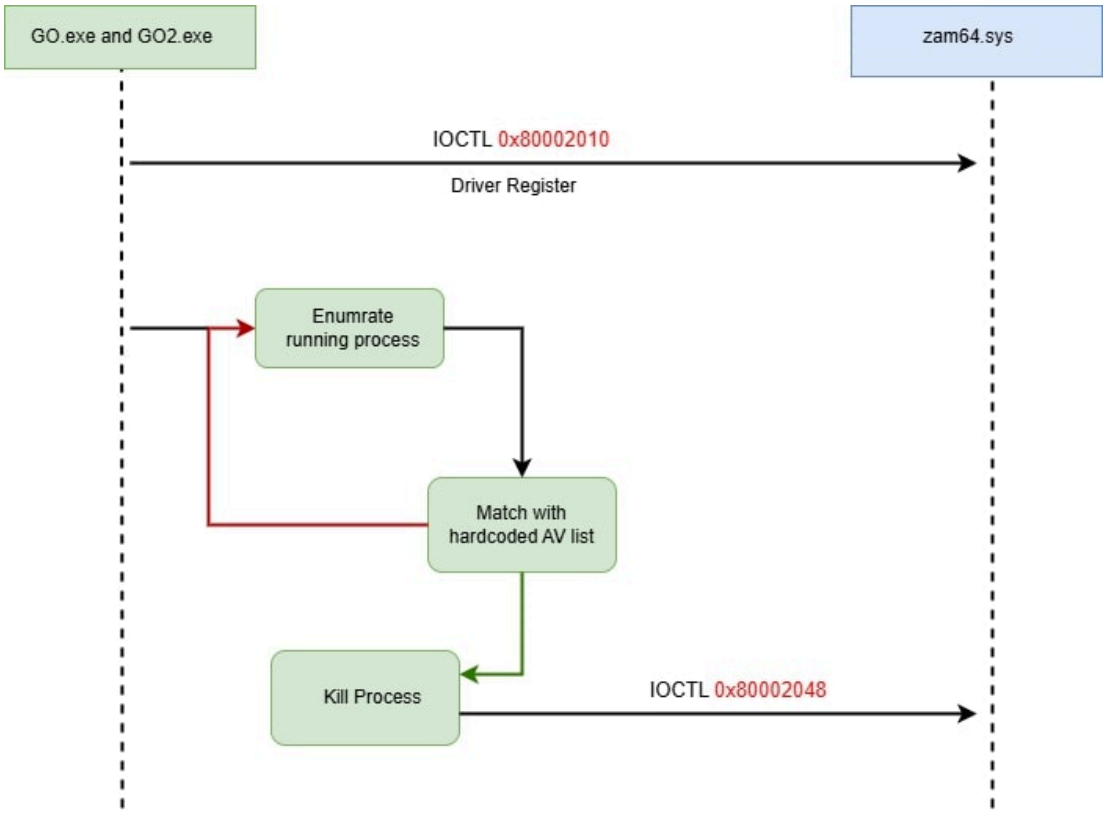

Figure 7: IOCTL code communication

The initial step performed by go.exe and go2.exe is to load and register the calling process with the zam64.sys driver using the IOCTL code 0x80002010.

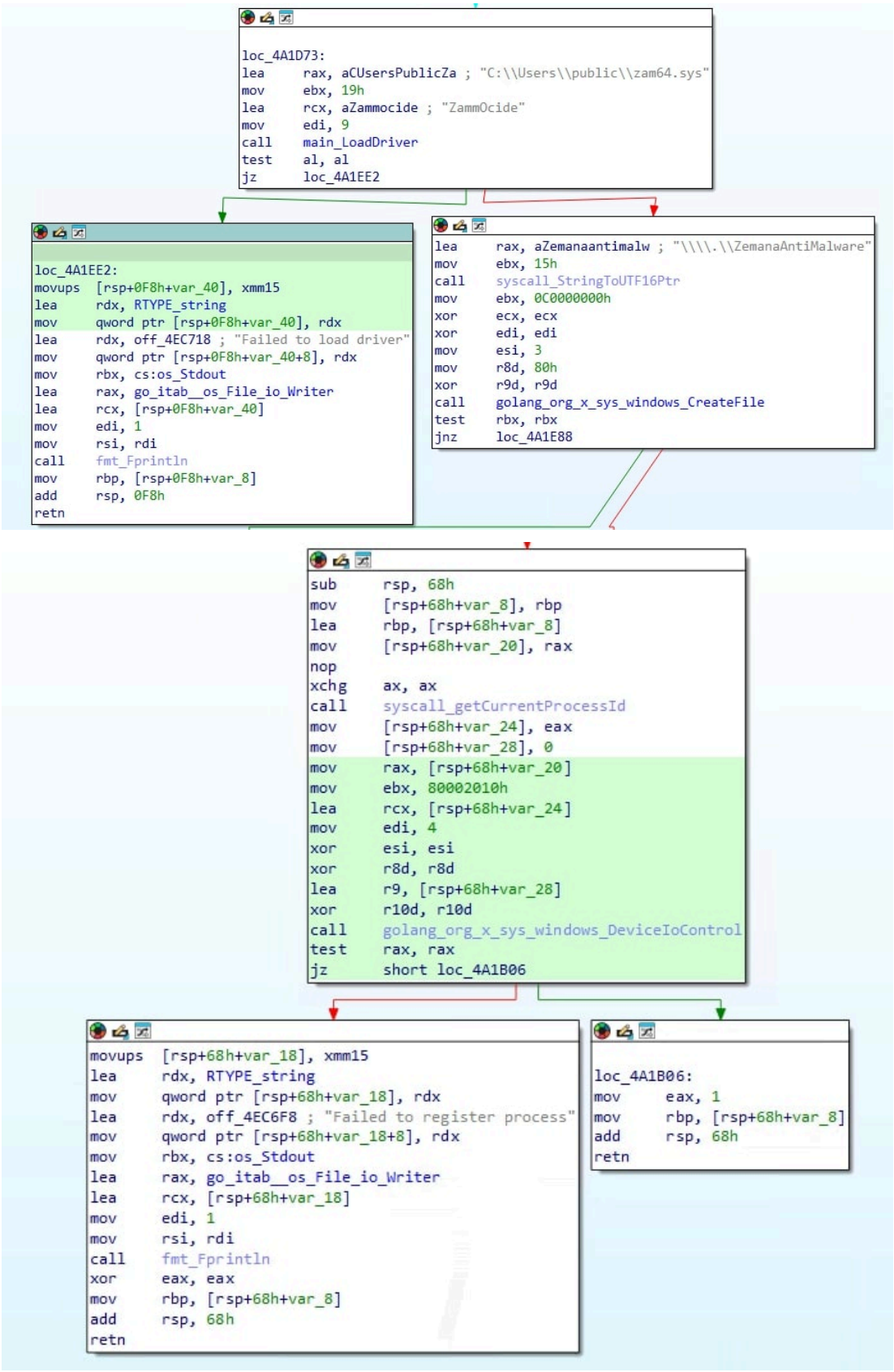

Figure 8: Zam64.sys Driver Registration (GO.exe and GO2.exe)

After driver registration, the AV killers enumerate the running processes on the system. They then compare these processes against a predefined list of known anti-malware products. This hardcoded list suggests the attackers have specific security solutions in mind as targets, indicating prior reconnaissance or a focus on commonly deployed security software.

Figure 9: Hardcoded AV list (GO2.exe)

If a running process matches an entry in the hardcoded list of target processes, the AV killer initiates its termination. This is achieved by sending another IOCTL code, 0x80002048, to the zam64.sys driver.

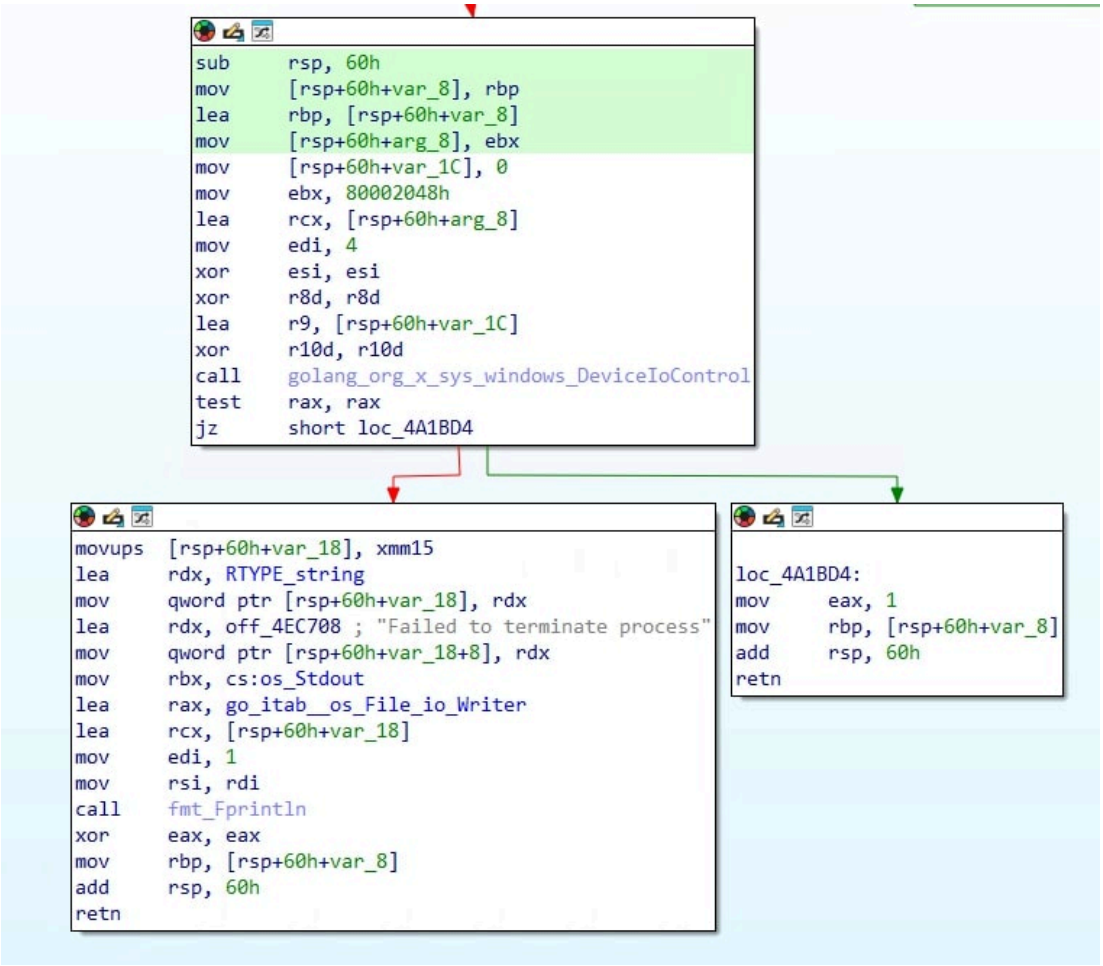

Figure 10: Process Termination IOCTL request (GO.exe and GO2.exe)

The successful execution of this IOCTL directly leads to the termination of anti-malware processes, effectively disabling the system's primary defense mechanism against malware.

IOCTL Code	Description	Vulnerability Associated	Function in the Attack
0x80002010	Registers the calling process ID as an authorized IOCTL process caller.	Denial of Service (DoS)	Allows go.exe and go2.exe to interact with the zam64.sys driver.
x80002048	Initiates the termination of a specified process.	Arbitrary Process Termination	Used by go.exe and go2.exe to terminate processes associated with antivirus software.

Table 2: zam64.sys IOCTL Usage

Unmasking crazyHunter ransomware: A look inside the encryptor

The CrazyHunter ransomware core functionality is outlined below.

1. **Drive enumeration**: The program begins by identifying all available drives on the system using the getDrives() function. This function iterates through the alphabet (A-Z) and checks if a drive exists for each letter.
2. **Directory encryption**: For each identified drive, the program calls the EncryptDirectory function from the filewalker package. This suggests the filewalker package contains the logic to recursively traverse directories and encrypt files within them.
3. **Wallpaper setting**: After attempting to encrypt all drives, the program executes the setWallpaper() function. This function is designed to change the victim's desktop wallpaper.

Drive enumeration: main_getDrives() function

This function iterates through the letters 'A' to 'Z' and for each letter, it attempts to open the root directory of the corresponding drive. If the drive exists, its drive letter is added to the list of available drives. The function then returns the list of all detected drives.

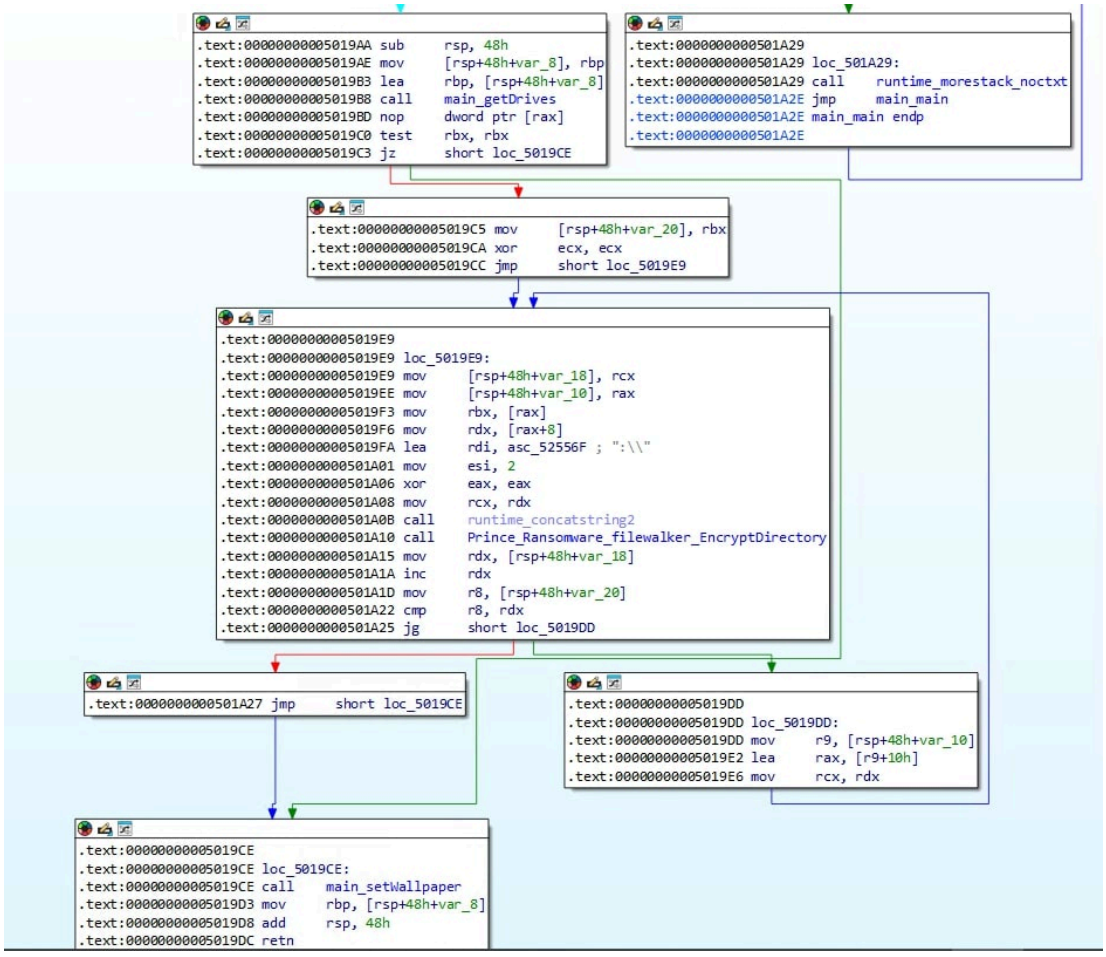

Figure 11: main_getDrives() Function

Defining the scope: Exclusion criteria

After looping through all directories for each drive letter, the directory exclusion check will occur, excluding extensions shown in Table 3.

.sys	.exe	.dll	.com	.scr
.bat	.vbs	.ps1	.lnk	.inf
.reg	.msi	.ini		

Table 3: Excluded file extension

The following file names in Table 4 are also excluded from file encryption.

boot.ini	bootmgr	bcd	desktop.ini	config.sys
autoexec.bat	decryption instructions.txt			

Table 4: Excluded files

The following directory names in Table 5 are also excluded from file encryption.

windows	system32	programdata	program files	Program Files (x86)
public	System Volume Information	\\system volume information	efi	boot
perflogs	microsoft	intel	appdata	.dotnet
.gradle	.nuget	.vscode	msys64	

Table 5: Excluded directories

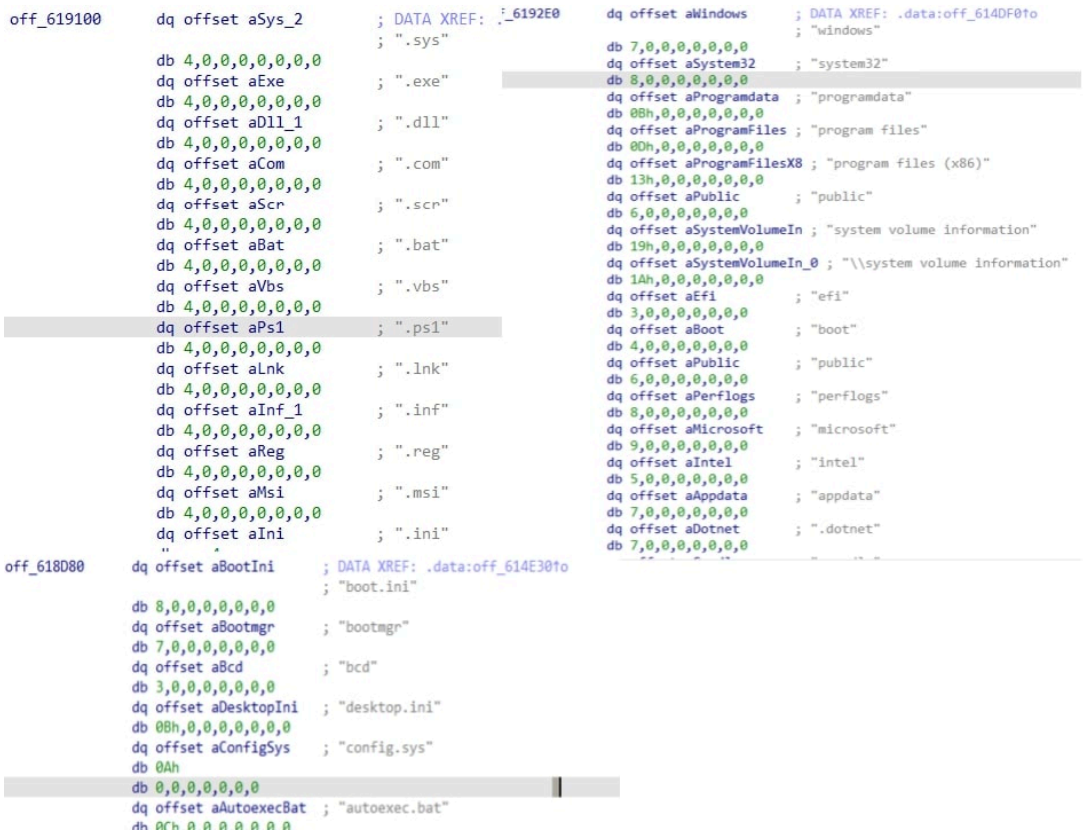

Figure 12: CrazyHunter ransomware - hardcoded exclusion lists.

Delving into the cryptographic core:

At its core, CrazyHunter ransomware employs a hybrid encryption strategy that combines symmetric and asymmetric algorithms to effectively secure files. This dual-layered approach is inherited from its foundation, the "Prince Ransomware" builder — an open-source tool written in Go.

ChaCha20: The workhorse for data encryption

For the primary task of encrypting file content, CrazyHunter utilizes the ChaCha20 stream cipher. A distinctive feature of this ransomware is its partial encryption. Instead of encrypting the entire file, it encrypts one byte of data and then skips the next two, leaving them in their original, unencrypted state. This 1:2 encryption ratio is a deliberate design choice from the underlying Prince builder. The likely rationale for this technique is to significantly increase the speed of the encryption process, allowing the ransomware to compromise a larger number of files in less time and potentially evade security solutions that monitor for heavy, sustained disk I/O operations.

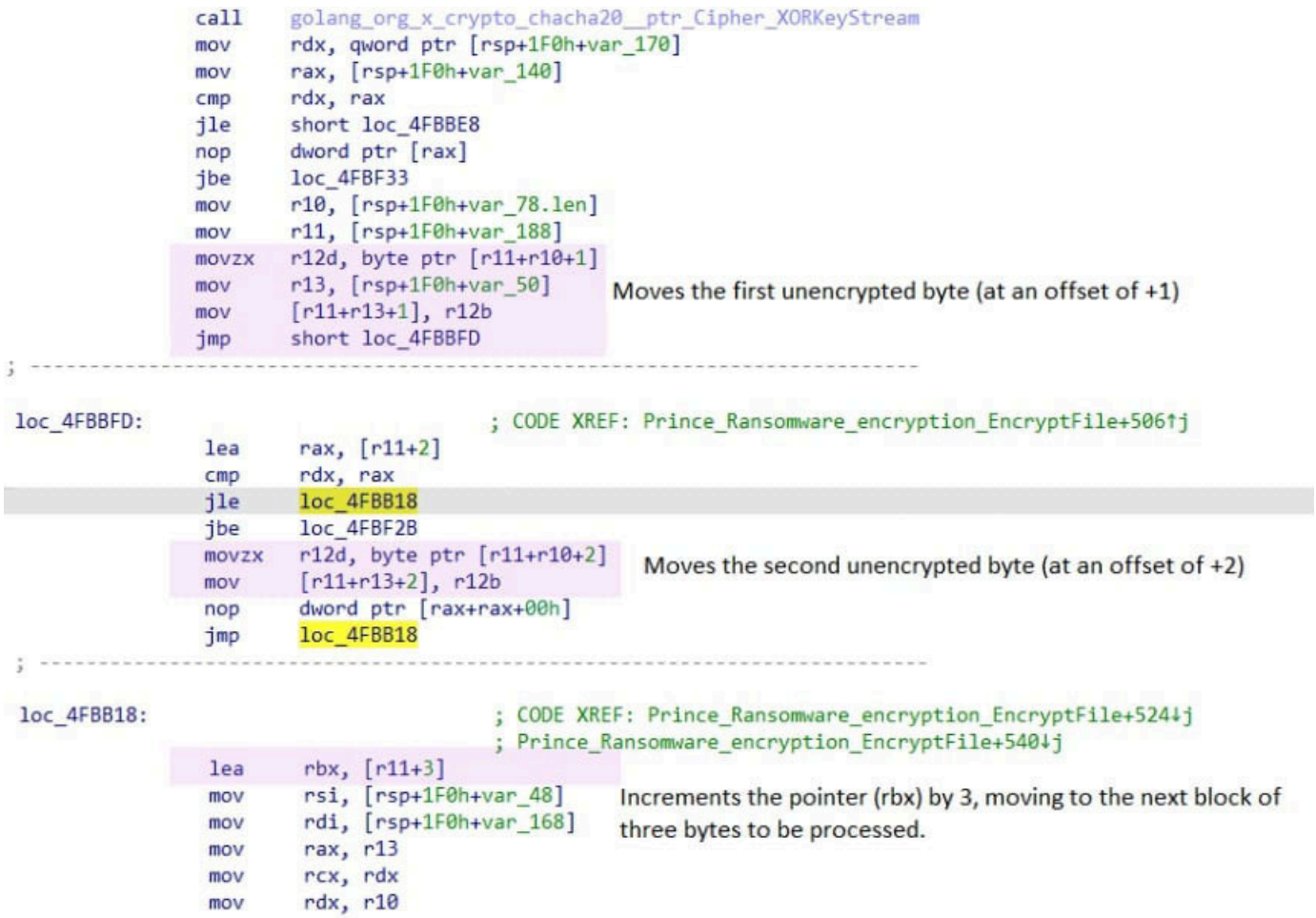

Figure 13: Offset calculation for byte encryption.

ECIES: Safeguarding the encryption keys

While ChaCha20 encrypts the data, the security of the entire operation depends on protecting the unique key and nonce generated for each file. To achieve this, CrazyHunter employs the Elliptic Curve Integrated Encryption Scheme (ECIES). ECIES is an efficient and secure asymmetric encryption method that provides robust security with shorter key lengths than other algorithms, such as RSA. This method ensures that decryption is impossible without the corresponding ECIES private key, which remains exclusively in the attacker's possession. Encrypted files are typically renamed with a .Hunter extension.

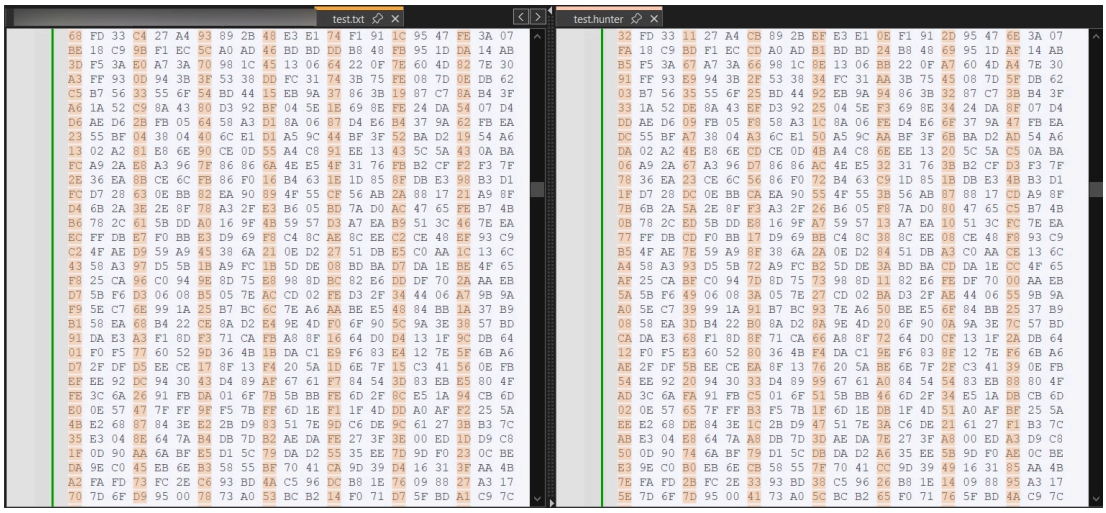

Figure 14: Encryption ratio 1:2 - test.text (right) and encrypted test.hunter (left)

Feature	Description
Encryption Algorithm (Data)	ChaCha20 stream cipher
Encryption Pattern	1 byte encrypted, 2 bytes unencrypted
Encryption Algorithm (Key)	ECIES (Elliptic Curve Integrated Encryption Scheme)
Key Protection	ChaCha20 key and nonce encrypted with ECIES public key and prepended to the file
Key Generation	Unique ChaCha20 key and nonce generated per file
Key Pair Generation	ECIES key pair generated by the builder tool

Table 6: CrazyHunter Ransomware Encrypted Mechanism

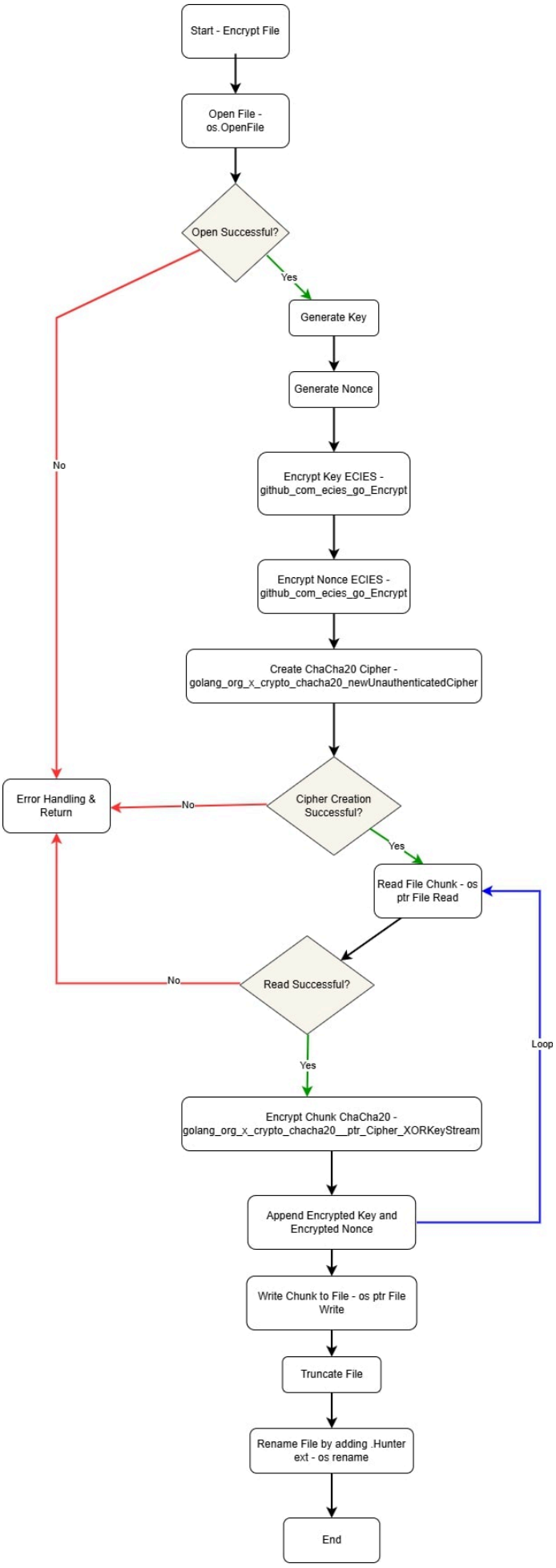

Figure 15: Encryption mechanism.

Encrypted file

The encrypted files with the .hunter extension are structured as,

[ECIES-encrypted ChaCha20 Key] || [ECIES-encrypted Nonce] || [Partially ChaCha20-encrypted File Content]

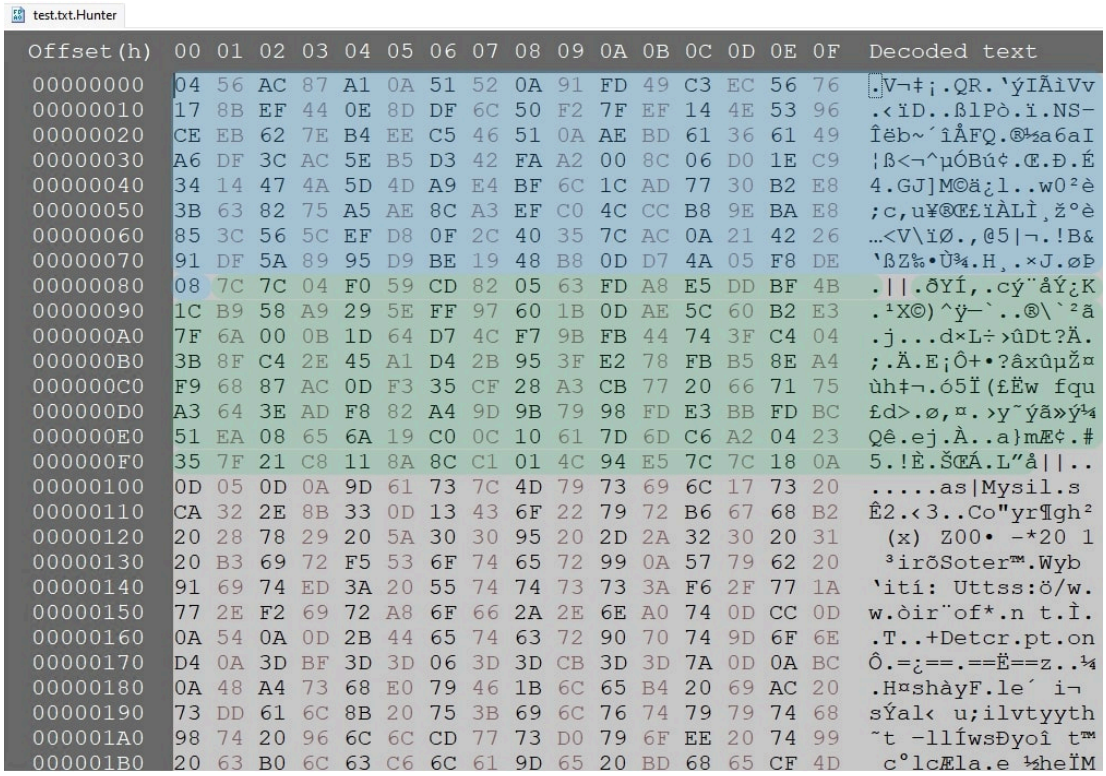

Figure 16: .hunter encrypted file structure.

File Offset (hex)	Data Description	Size (bytes)
0-81	ECIES-encrypted ChaCha20 key	129
83-FB	ECIES-encrypted Nonce	121
Variable	ChaCha20-encrypted file content (1 byte encrypted, 2 bytes unencrypted)	Remaining

Table 7: CrazyHunter Ransomware Encrypted File Structure

Setting the wallpaper

Executes a PowerShell script to download a file from a remote URL: hxxps[://]ncmep[.]org/files/2023/05/ransomeware-01-1280x640[.]png. This downloaded file is saved in the \temp directory as "Wallpaper.png". Another PowerShell command is executed to set it as wallpaper.

Figure 17: Wallpaper.png

Donut loader and shellcode

CrazyHunter.sys is an encrypted shellcode made with the donut framework, and the bb.exe file is a loader. The script executes bb.exe with the -f flag followed by the path to crazyhunter.sys. This command instructs bb.exe to decrypt the shellcode and execute it in memory without writing it to disk.

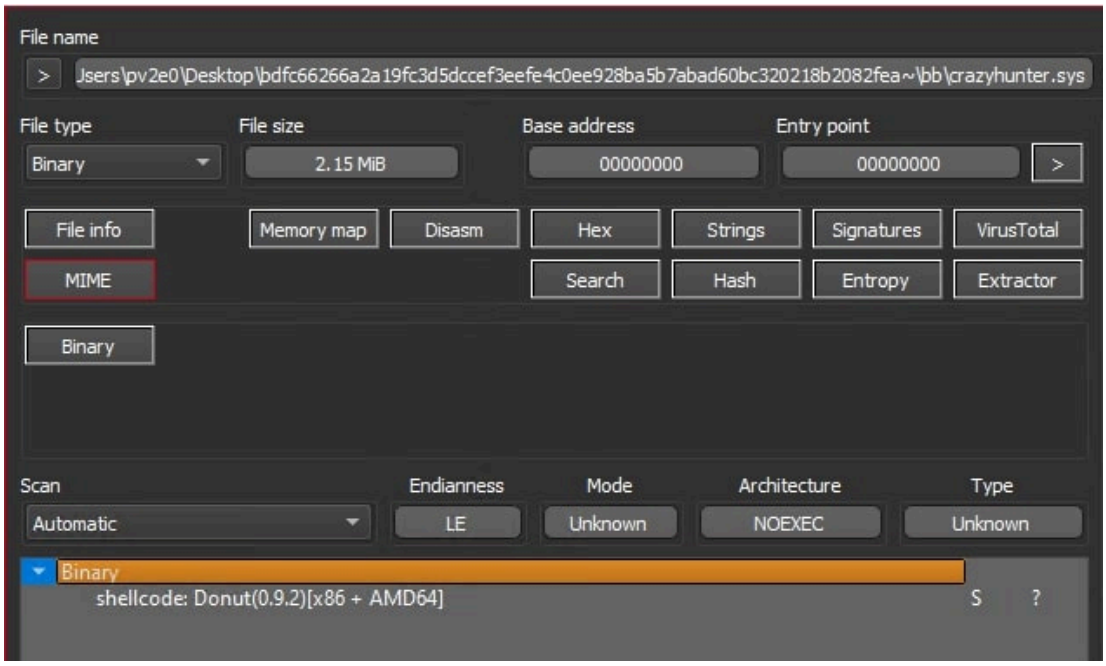

Figure 18: CrazyHunter.sys Donut shellcode file

We manually decoded the shellcode using the open-source tool "donut-decryptor" on GitHub and discovered that it was the same go.exe payload.

Ransom negotiation and payment methods: Extortion process

Communication during ransom negotiation for CrazyHunter ransomware occurs through various channels. One identified method is via email, with the address *attack-tw1337@proton.me* being associated with CrazyHunter Ransomware.

Additionally, a Telegram channel (*Telegram@Magic13377*) and a TOR address (*hxxp://7i6sfmfvmqfaabjksckwrttu3nsbopl3xev2vbxbkghsivs5lqp4yeqd.onion*) are also linked to the CrazyHunter Team. These platforms likely serve as avenues for communicating with victims and, potentially, for hosting leak sites or negotiation portals. Most of the contact details are mentioned in the ransom note.

Figure 19: Ransom note

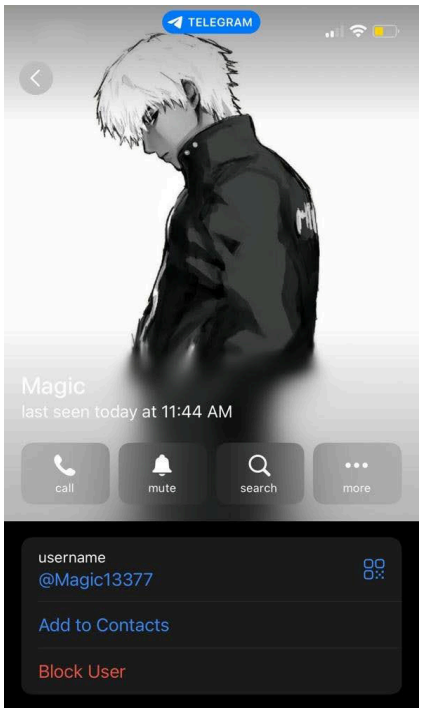

Figure 20: The attacker's Telegram page.

File.exe: Data exfiltration tooling

The file.exe executable, a component of the CrazyHunter ransomware attack, accepts several command-line arguments that provide insights into its functionalities. These arguments include: -d, -e, -f, -func, -port, -t, and -white. The -func parameter is particularly significant as it dictates the primary mode of operation for the executable.

Analysis revealed that the file.exe artifact possesses dual functionality: it can transform a compromised machine into a file server or act as a file-monitoring and deletion tool. When operating as a file server, it exposes the designated directory (defaulting to the

current) via localhost on a specified port (default: 9999). In monitoring mode, it systematically scans and deletes files matching predefined extensions within the directory and its sub-directories

Based on our research, we predict that file.exe is used in the extortion process to control and monitor the victim's machine.

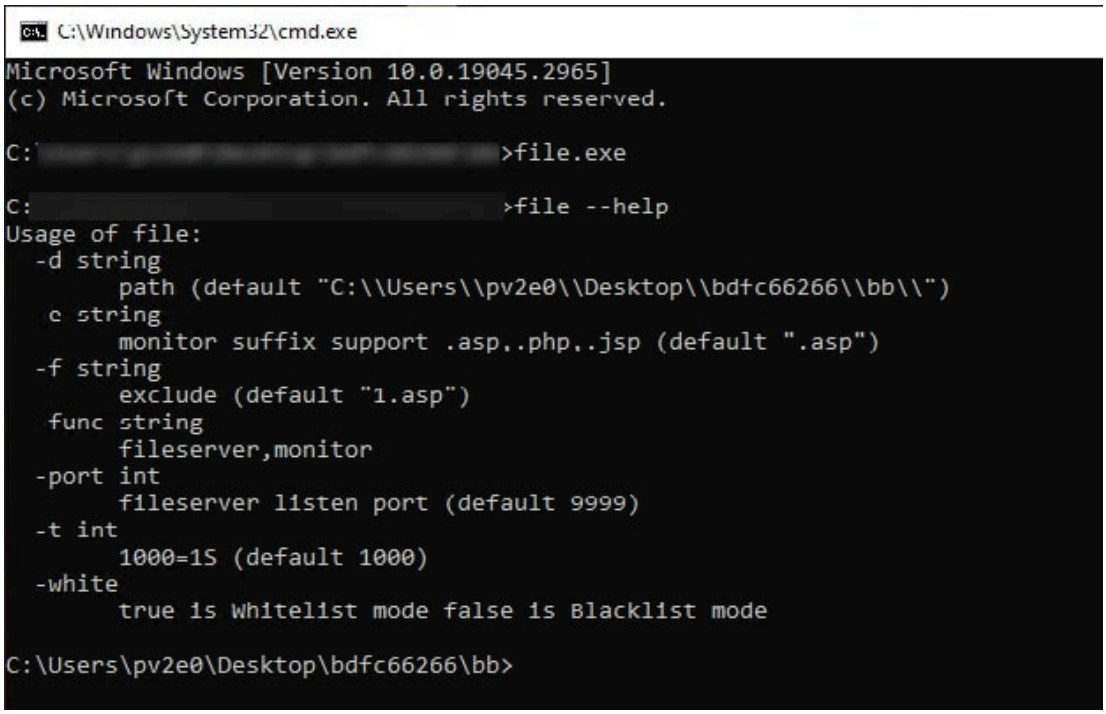

Figure 21: File.exe command-line utilities.

Fortifying your defenses: A CISO's guide to neutralizing CrazyHunter ransomware

Secure Active Directory (AD): Enforce MFA for all domain accounts and strictly control GPO modification rights to prevent credential theft and payload distribution via SharpGPOAbuse.

Neutralize Evasion Tactics: Utilize **Trellix Endpoint Detection** capabilities to counter AV killers and ransomware payloads, and block the execution of BYOVD attacks that exploit vulnerable drivers for privilege escalation and security termination.

Ensure Robust Recovery: Implement a proper backup strategy (offsite/offline) to ensure backups are immutable and inaccessible to the ransomware, and regularly test the incident response plan for effective post-attack recovery.

Restrict Lateral Movement: Use network segmentation and strict access controls to limit the ransomware's rapid propagation capability across the network, particularly by preventing widespread deployment through compromised AD credentials and GPOs.

Trellix protection and mitigation

Trellix has implemented comprehensive protection and mitigation measures against the CrazyHunter ransomware. Our security solutions now include coverage for all known CrazyHunter-related executables, ensuring robust defense against this threat.

This proactive approach aims to safeguard our customers by preventing infection and minimizing potential damage.

Appendix A - Indicators of compromise

SHA256 / Data	Descrip
f72c03d37db77e8c6959b293ce81d009bf1c85f7d3bdaa4f873d3241833c146b	go3.exe, CrazyHu ransom
754d5c0c494099b72c050e745dde45ee4f6195c1f559a0f3a0fddba353004db6	go.exe, /
983f5346756d61fec35df3e6e773ff43973eb96aabaa8094dcbfb5ca17821c81	go2.exe
512f785d3c2a787b30fa760a153723d02090c0812d01bb519b670ecfc9780d93	gpo.exe SharpG
2cc975fdb21f6dd20775aa52c7b3db6866c50761e22338b08ffc7f7748b2acaa	bb.exe, Shellcoc
d1081c77f37d080b4e8ecf6325d79e6666572d8ac96598fe65f9630dda6ec1ec	ru.bat, s
53160607452717 23c9934047155dae95a3920cb6343ca08c93531e1c235861ba	crazyhu Donut CrazyHu ransom shellcoc
Telegram@Magic13377	Telegra channe
attack-tw1337@proton.me	Attacke ID
7i6sfmfvmqfaabjksckwrttu3nsbopl3xev2vbxbkghsivs5lqp4yeqd[.]onion	Onion p

Appendix B - Trellix detection signatures

Product	Signature
Trellix Endpoint Security (ENS)	Ransomware-HWL Ransom-crazyhunter!mem

	BAT/CrazyHunter.a ShellCode/Donut.a Trojan-FZBH Trojan-FZBH Downloader-FCTN trojan.bkq
Trellix EDR	Win_ransomware_crazyhunter_1 win_file_possible_ransomware_infection
Trellix Network Security Trellix VX Trellix Cloud MVX Trellix File Protect Trellix Malware Analysis Trellix SmartVision Trellix Email Security Trellix Detection As A Service Trellix NX	FE_Loader_Win_Generic_180_FEBeta FE_Loader_MSIL_Generic_230_FEBeta FE_Loader_MSIL_Generic_231_FEBeta FE_HackTool_MSIL_SharpGPOAbuse_1_FEBeta FEC_Loader_BAT_Generic_15_FEBeta

Appendix C - MITRE ATT&CK

Tactical Goal	ATT&CK Technique (Technique ID)	Description
Initial Access	T1078.002 Valid Accounts - Domain Accounts	Exploited weak passwords to compromise AD accounts.
Execution	T1204.002 User Execution - Malicious File	Leveraged SharpGPOAbuse to deploy malware via Group Policy Objects (GPOs).
Persistence	T1484.001 Domain Policy Modification	Executed the ransomware payload after gaining initial access.
Privilege Escalation	T1068 Exploitation for Privilege Escalation	Utilised BYOVD with a modified Zemana driver to bypass security controls.
Defense Evasion	T1553.002 Code Signing T1036 Masquerading	Signed malicious drivers to avoid detection. Disguised ransomware as a legitimate process.

Credential Access	T1003 Credential Dumping	Credentials will likely be extracted to facilitate lateral movement within the network.
Discovery	T1018 Remote System Discovery	Identified accessible systems to expand the attack.
Lateral Movement	T1021 Remote Services	Propagated the ransomware using compromised AD credentials and GPOs.
Impact	T1486 Data Encrypted for Impact T1485 Data Destruction	Encrypts the target systems, severely disrupting operations. Possibly deleted backups or logs to complicate recovery efforts.

Discover the latest cybersecurity research from the Trellix Advanced Research Center.

This document and the information contained herein describes computer security research for educational purposes only and the convenience of Trellix customers.

Aug 13, 2026

Signed, sealed, injected: The mechanics of DCRat in 2026

Aug 12, 2026

Weaponized AI: The Commoditization of Cybercrime

Aug 10, 2026

Accelerating Trellix's Engineering Transformation with AI

Latest from our newsroom

Blogs | Research

Fileless Multi-Stage Remcos RAT: From Phishing to Memory-Resident Execution

By Madhini Muralidharan · March 11, 2026

This blog examines a Remcos campaign demonstrating the transition from phishing-based initial access to fully fileless execution.

Read the Blog →

Blogs | Platform

Why Trellix SecondSight Is Like Gaining a Team of Elite Threat Hunters

By Brian B Brown · March 3, 2026

Trellix SecondSight brings proactive Threat Hunting to organizations with no requirement for extra resources.

Read the Blog →

Blogs | Research

European Sovereignty-By-Design: Trellix's Foundation for Operational Autonomy and Local Control

By Chris Hutchins · February 19, 2026

Trellix solutions align with European values, providing the security, resilience, flexibility, and interoperability government and enterprises need.

Read the Blog →

Get the latest

Stay up to date with the latest cybersecurity trends, best practices, security vulnerabilities, and so much more.

Email

Submit

Zero spam. Unsubscribe at any time.