GROUP-IB

← Blog

Vito Alfano

Head of DFIR Practice, EU

RansomHub Never Sleeps Episode 1: The evolution of modern ransomware

Discover how ransomware has evolved into a sophisticated cyber threat, with groups like RansomHub leading the charge. Learn more about their adaptability, TTPs, and the rise of Ransomware-as-a-service in this first-of-three-part trilogy.

February 12, 2025 · min to read · Ransomware

Affiliates DFIR RaaS RansomHub Ransomware

Introduction

The cybersecurity threat landscape is a constant arms race between attackers and defenders. As organizations strengthen their defenses, adversaries evolve their tactics, techniques and procedures (TTPs) to exploit emerging vulnerabilities. Among these threats, ransomware operations have become increasingly sophisticated and prominent.

In its early days, ransomware targeted individuals with relatively small demands. However, with growing digital interconnectivity and exposed system vulnerabilities, attackers have shifted to larger targets including healthcare, finance, critical infrastructure, and government sectors. The advent of Ransomware-as-a-Service (RaaS) platforms has further lowered barriers for aspiring cybercriminals, enabling them to access advanced tools in exchange for a share of the profits.

A key driver of ransomware's growth is its adaptability. Modern groups exploit unpatched vulnerabilities, use advanced reconnaissance techniques, and leverage automation to scale their operations. In this first part of a trilogy of Group-IB blogs on ransomware, we'll deep-dive into RansomHub, which emerged in early 2024, and how it exemplifies this evolution. Through innovation and rapid adaptation, this RaaS group has solidified its position as a significant threat in today's cybersecurity landscape.

Key discoveries in the blog

- RansomHub's operators strategically advertised the group's partnership program on RAMP forum on February 2, 2024.
- RansomHub's operators took advantage of the impact of law enforcement operations on LockBit and ALPHV to release a partnership program and recruit affiliates of these groups.
- The threat actors likely acquired the ransomware and web application source code from the Knight (aka Cyclops) group.
- The ransomware works on different operating systems and architectures including x86, x64 and ARM as well as Windows, ESXi, Linux and FreeBSD.
- The group started to use PCHunter to stop and bypass endpoint security solutions.
- RansomHub used Filezilla as an exfiltration tool.
- RansomHub's affiliates have disclosed around 44 healthcare companies including hospitals and clinics.
- Affiliates may eventually threaten and report cyber incidents to regulators such as PDPL (Personal Data Protection Law).

Who may find this article interesting

- Cybersecurity analysts and corporate security teams
- Malware analysts
- Threat Intelligence specialists
- Cyber investigators
- Computer Emergency Response Teams
- Law enforcement investigators
- Cyber Police Forces

Who is RansomHub?

RansomHub emerged in early February 2024 as a Ransomware-as-a-Service (RaaS) coinciding with the closure of ALPHV's operations. ALPHV shut down its infrastructure following the significant fallout from a disruptive attack on Change Healthcare.

During ongoing law enforcement actions targeting the ALPHV and LockBit ransomware groups, RansomHub strategically launched its partnership program. This effort was analyzed by Group-IB in August 2024, as noted earlier in this blog.

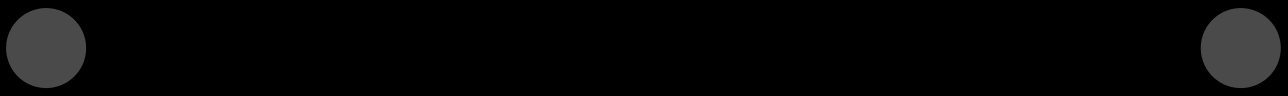


Figure 1. RansomHub's partnership program advertisement on RAMP forum

Group-IB's Threat Intelligence and Digital Forensics and Incident Response (DFIR) teams found that RansomHub capitalized on the void left by its disrupted competitors, focusing on recruiting affiliates from the now-defunct LockBit and ALPHV groups. The group actively sought new members through direct messaging and posts on underground forums like RAMP, XSS, and Exploit.in.

To expedite its operations, RansomHub acquired the source code and web application sold on the RAMP forum by the disbanded ransomware group Knight (formerly Cyclops), according to information obtained by Group-IB's Threat Intelligence team from RansomHub's affiliates.

Evidence suggests that RansomHub purchased and rebranded Knight's resources. The similarities between the affiliate panel used by RansomHub and Knight, the overlapping ransomware features, and the shared code corroborate this theory. The source code had reportedly been offered for sale on RAMP on February 18, 2024.

Initially, it appeared that neither the ransomware nor the affiliate panel offered any novel features, as compared to those observed in other RaaS groups that have been analyzed by our threat intelligence team. However, on July 18, 2024, **koley**, a RansomHub operator, advertised on RAMP forum a new strain of the ransomware, which was able to remotely encrypt data via SFTP protocol. This new version was announced on RAMP forum a few weeks after security companies published reports on the group.

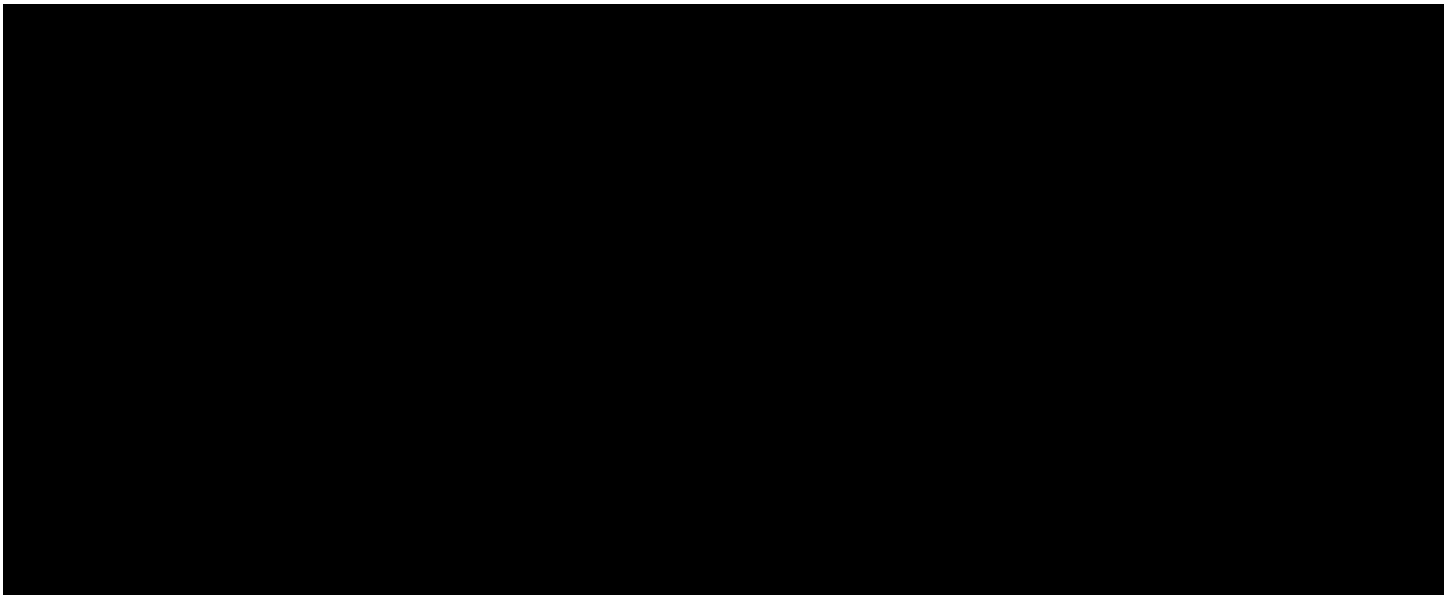


Figure 2. Comments from RansomHub's operator on rule changes after security companies accessed the affiliate panel

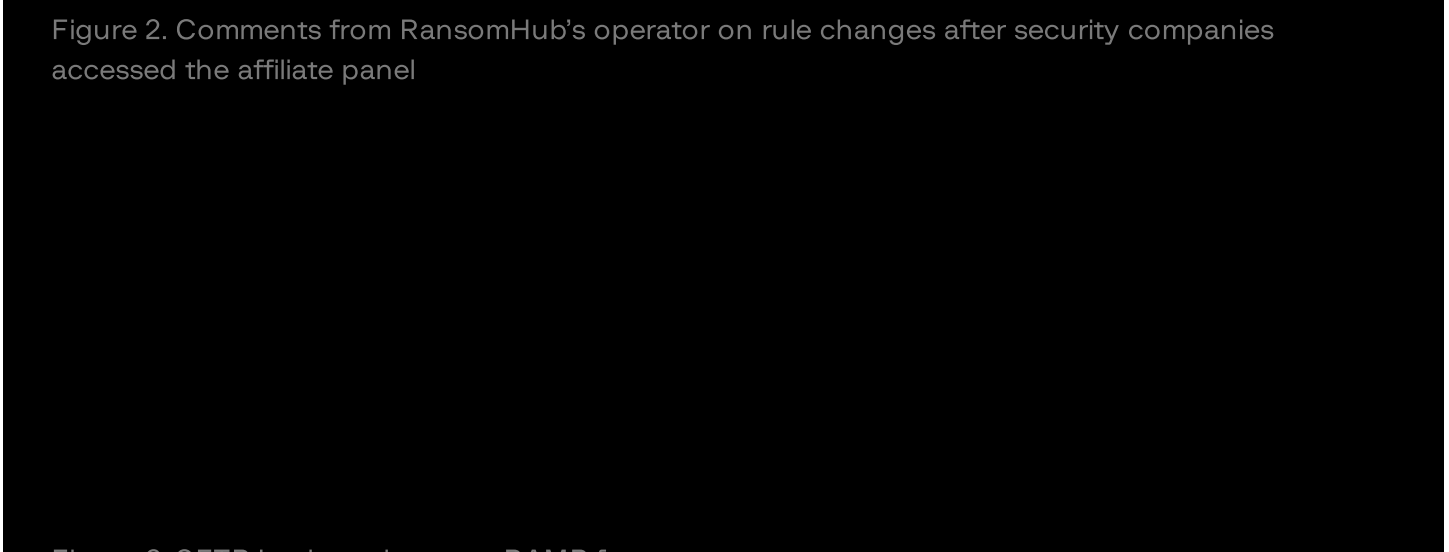


Figure 3. SFTP Locker release on RAMP forum

The release of this updated ransomware strain appears to have been a strategic move. It not only introduced new resources for affiliates but also aimed to mitigate potential reputational damage after security firms gained access to the affiliate panel.

At the time of writing, RansomHub has targeted over 600 organizations globally, spanning sectors such as healthcare, finance, government, and critical infrastructure, firmly establishing it as the most active ransomware group in 2024.

This article shares the details of an incident response case handled by Group-IB's DFIR team, including new insights into RansomHub's TTPs and a technical analysis of its ransomware uncovered by Group-IB's malware analysts.

Dissecting a ransomware operation

Picture a strategic operation unfolding, an adversary initiated a covert reconnaissance mission, systematically probing publicly exposed services and resources of a targeted organization. The objective: to identify weaknesses in the perimeter defenses.

Their intelligence revealed a critical vulnerability—**CVE-2024-3400**—impacting Palo Alto Networks firewall appliances running an outdated PAN-OS software. This exploit allowed attackers to execute arbitrary code with root privileges, bypassing authentication and gaining a foothold inside the network.

Figure 4. Extract of security advisory released by Palo Alto

The attack demonstrated both the sophistication and adaptability of RansomHub affiliates, who moved swiftly to weaponize this vulnerability. Notably, the exploit had only recently been used by a China-based threat actor targeting critical infrastructures, months before any proof-of-concept code was publicly available. This rapid deployment showcased the group's tactical acumen and readiness to capitalize on cutting-edge vulnerabilities before defenders could respond.

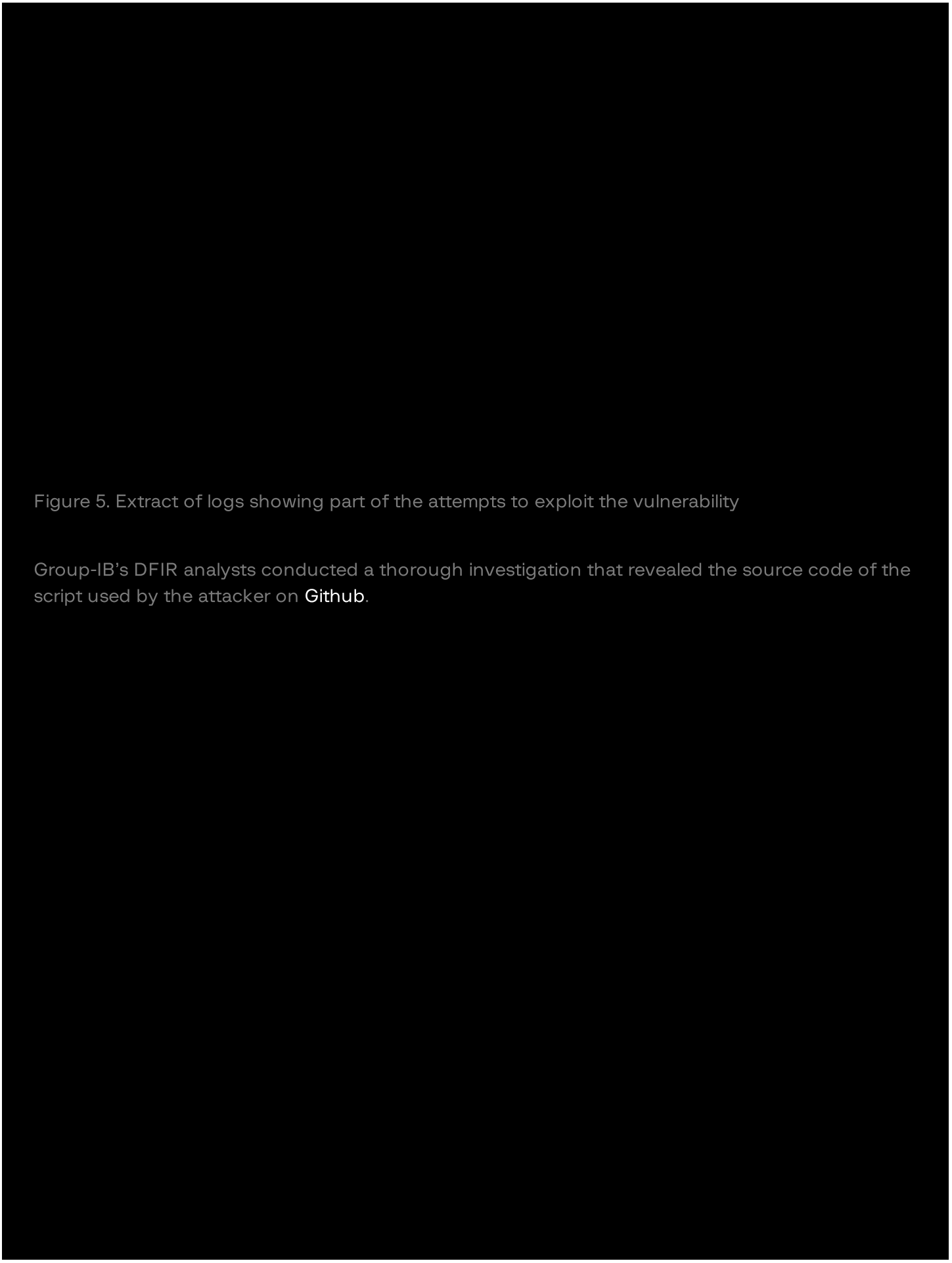


Figure 5. Extract of logs showing part of the attempts to exploit the vulnerability

Group-IB's DFIR analysts conducted a thorough investigation that revealed the source code of the script used by the attacker on Github.

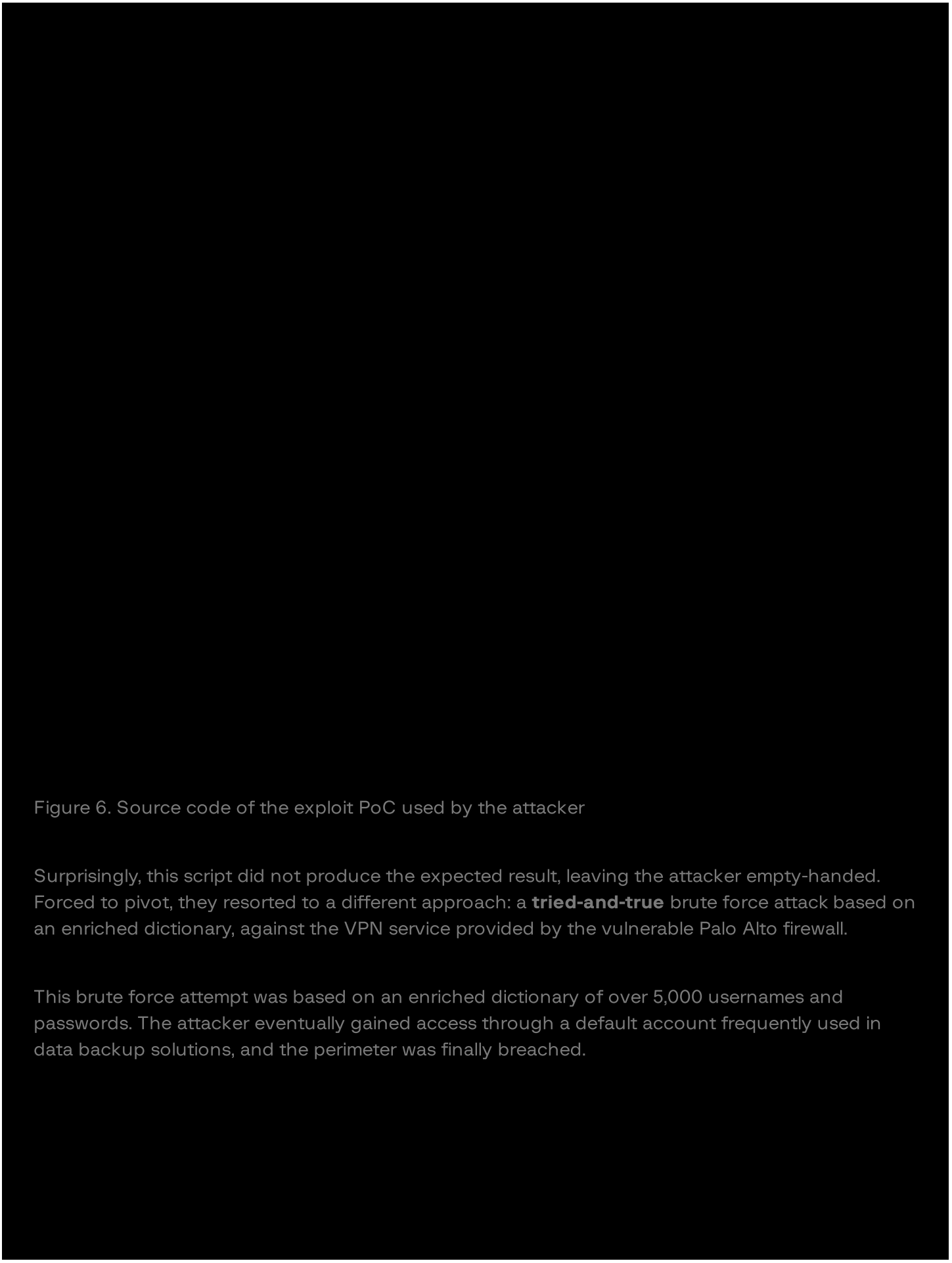


Figure 6. Source code of the exploit PoC used by the attacker

Surprisingly, this script did not produce the expected result, leaving the attacker empty-handed. Forced to pivot, they resorted to a different approach: a **tried-and-true** brute force attack based on an enriched dictionary, against the VPN service provided by the vulnerable Palo Alto firewall.

This brute force attempt was based on an enriched dictionary of over 5,000 usernames and passwords. The attacker eventually gained access through a default account frequently used in data backup solutions, and the perimeter was finally breached.

Figure 7. Extract logs showing the first malicious access

The unauthorized access rapidly escalated, with the victim experiencing both data encryption and exfiltration within 24 hours.

Similar to military tactics which have transcended into cyberspace, the attacker—having gained initial access—performed internal reconnaissance using tools like Angry IP Scanner, Nmap, and PowerShell scripts. By running these tools, the attacker aimed to gather detailed information about the perimeter, looking for vulnerable assets and aiming to get the access into the domain controller, which is always considered the most important element of an IT infrastructure based on MS Windows and the primary goal for any threat actor.

This technique allowed the attacker to exploit two vulnerabilities in the domain controller: CVE-2021-42278 (sAMAccount Spoofing) and CVE-2020-1472.

The vulnerability labelled as CVE-2021-42278 enables an attacker with limited domain user credentials to obtain a Kerberos service ticket for the domain controller, ultimately allowing them to compromise the domain controller.

- A new computer is added to the domain.
- The new computer is renamed with the name of a domain controller, but without the trailing “$”.
- A new Kerberos TGT is requested using the newly created computer name.
- The new computer account is then renamed to any other name.
- A Kerberos service ticket is requested using the S4U2self extensions and once obtained can be used to access any service on the domain controller

Figure 8. Screen shot of the extract logs sAMAccount spoofing

The second one, labelled with CVE-2020-1472 and known also as **ZeroLogon**, affects Microsoft's Active Directory NetLogon remote Protocol (MS-NRPC), and it allows a malicious actor without user credentials to gain the highest privileges in the domain, and take the control of a vulnerable domain controller via NT Lan Manager (NTLM).

Figure 9. Screen shot of the log ZeroLogon attempt

The exploitation of the above-mentioned vulnerabilities enabled the attacker to gain full privileged access to the domain controller, which is the nerve center of a Microsoft Windows-based infrastructure.

Once the attacker had gained full control of the domain, they were able to begin their lateral movements across the entire perimeter with any preferred user.

First, they accessed one of the main network-attached storage servers and created a new folder, configuring it as a shared resource. They then uploaded the tools that would be used during the next phases of the attack to establish a point of persistence and facilitate the movement of resources across the compromised perimeter.

Figure 10. Toolkit upload schema

This action marks a critical step in the attack lifecycle. The attacker successfully completed the initial data gathering and prepared the stage.

At this point, the attackers are ready to initiate the advanced phase of the attack, which involves moving laterally, identifying and targeting critical assets such as NAS and shared folders, along with backup systems. The scope is primarily the extraction of the data and subsequently exfiltrate it through external command and control servers.

Figure 11. Critical asset access schema

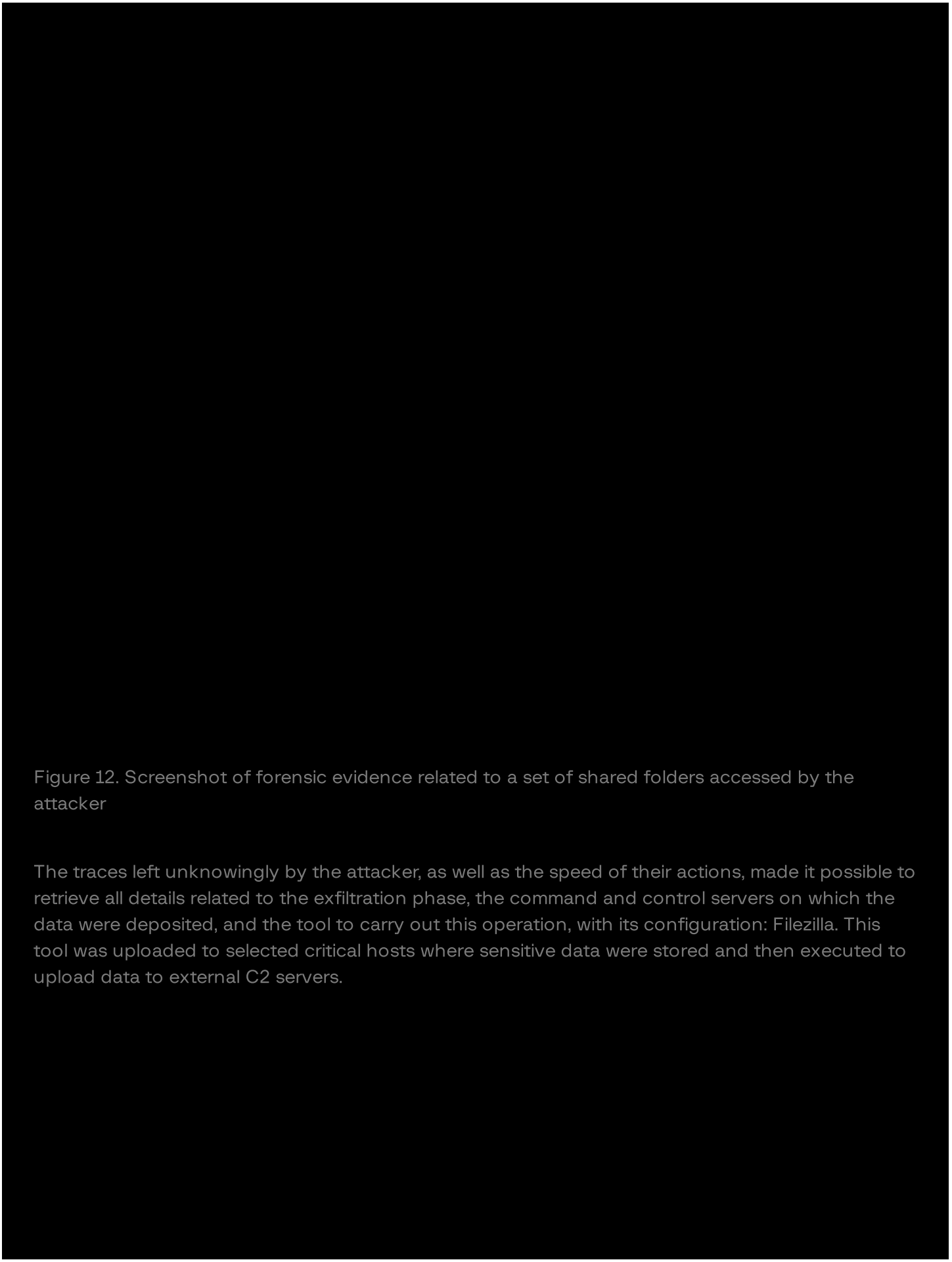


Figure 12. Screenshot of forensic evidence related to a set of shared folders accessed by the attacker

The traces left unknowingly by the attacker, as well as the speed of their actions, made it possible to retrieve all details related to the exfiltration phase, the command and control servers on which the data were deposited, and the tool to carry out this operation, with its configuration: Filezilla. This tool was uploaded to selected critical hosts where sensitive data were stored and then executed to upload data to external C2 servers.

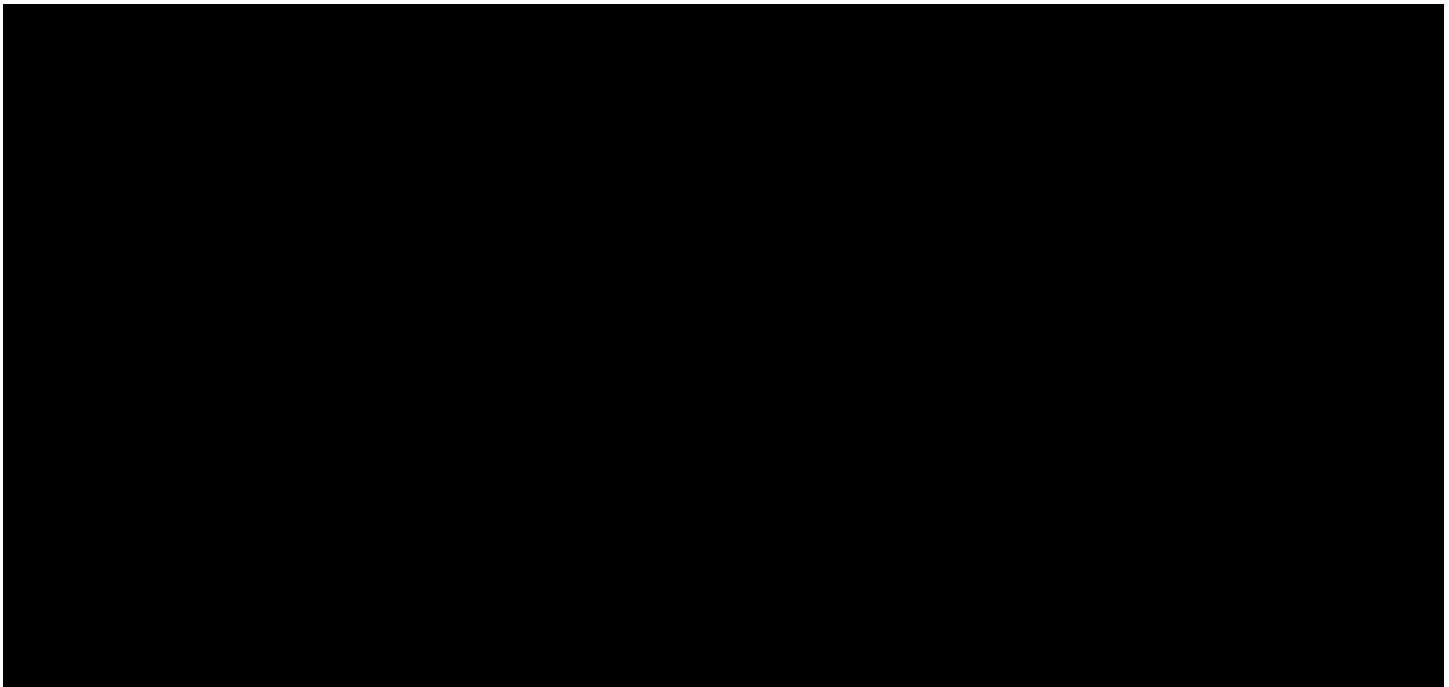


Figure 13. An illustration of the Filezilla upload schema

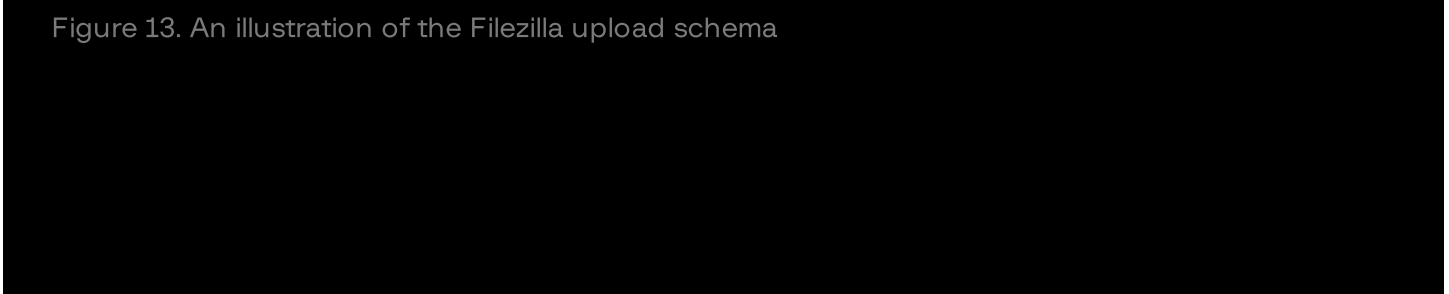


Figure 14. Extract of a $Jrnl Table record showing the upload of Filezilla

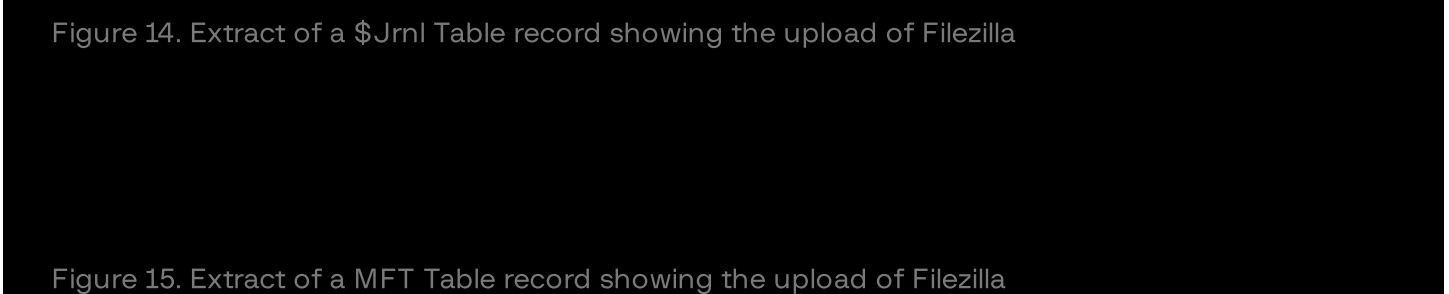


Figure 15. Extract of a MFT Table record showing the upload of Filezilla

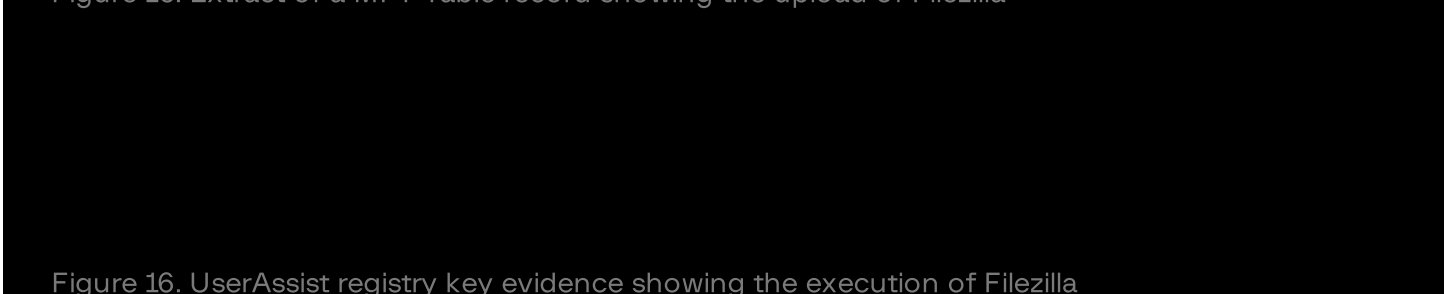


Figure 16. UserAssist registry key evidence showing the execution of Filezilla

The retrieved Filezilla configuration was of significant value due to the information it contained, including command and control server IP addresses, usernames, destination ports and listening service configurations. The in-depth, intelligence-driven analysis of these IoCs resulted in a lengthy research activity that produced interesting results. These will be shared in the next episodes of this blog.

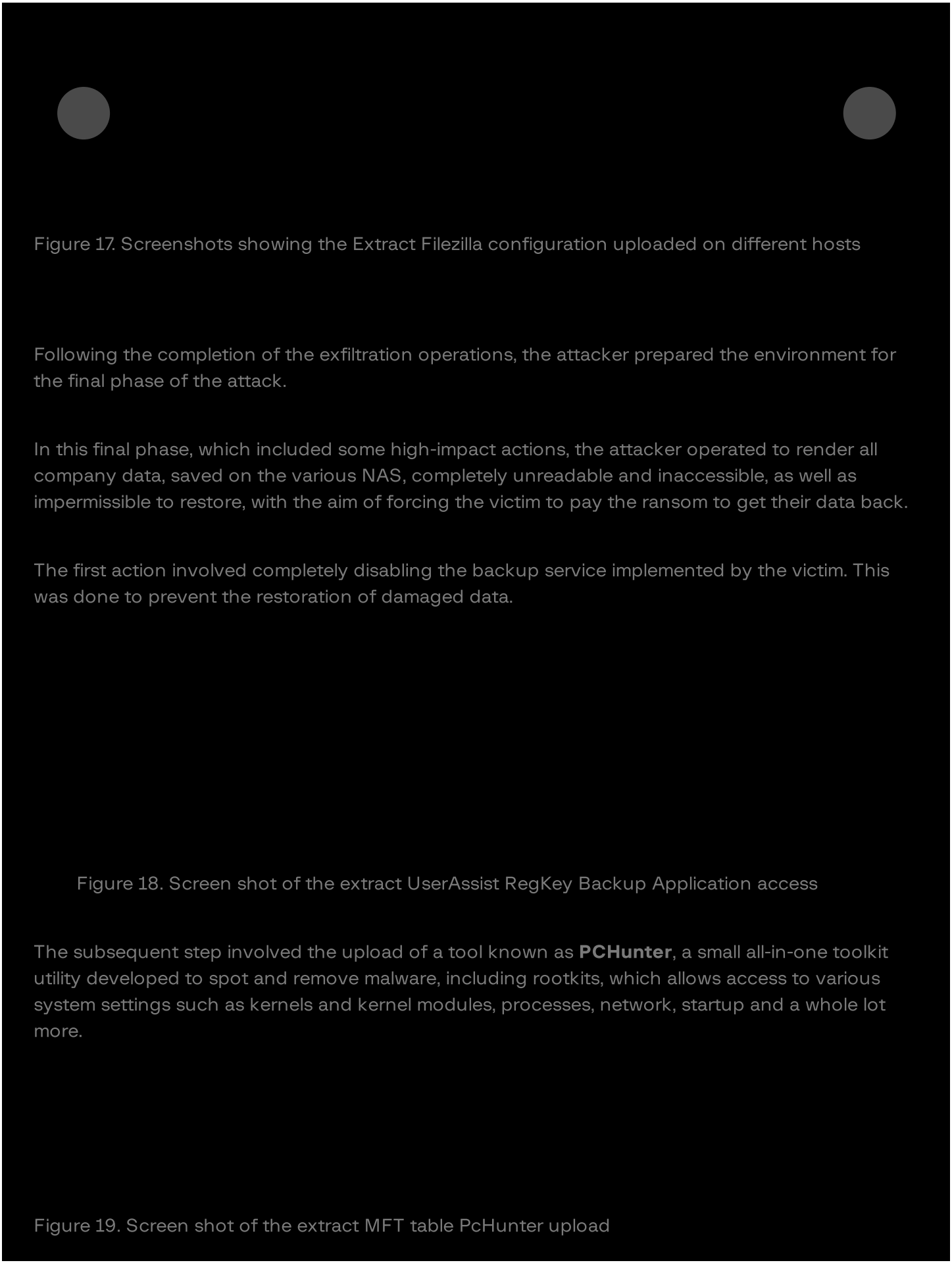


Figure 17. Screenshots showing the Extract Filezilla configuration uploaded on different hosts

Following the completion of the exfiltration operations, the attacker prepared the environment for the final phase of the attack.

In this final phase, which included some high-impact actions, the attacker operated to render all company data, saved on the various NAS, completely unreadable and inaccessible, as well as impermissible to restore, with the aim of forcing the victim to pay the ransom to get their data back.

The first action involved completely disabling the backup service implemented by the victim. This was done to prevent the restoration of damaged data.

Figure 18. Screen shot of the extract UserAssist RegKey Backup Application access

The subsequent step involved the upload of a tool known as **PCHunter**, a small all-in-one toolkit utility developed to spot and remove malware, including rootkits, which allows access to various system settings such as kernels and kernel modules, processes, network, startup and a whole lot more.

Figure 19. Screen shot of the extract MFT table PcHunter upload

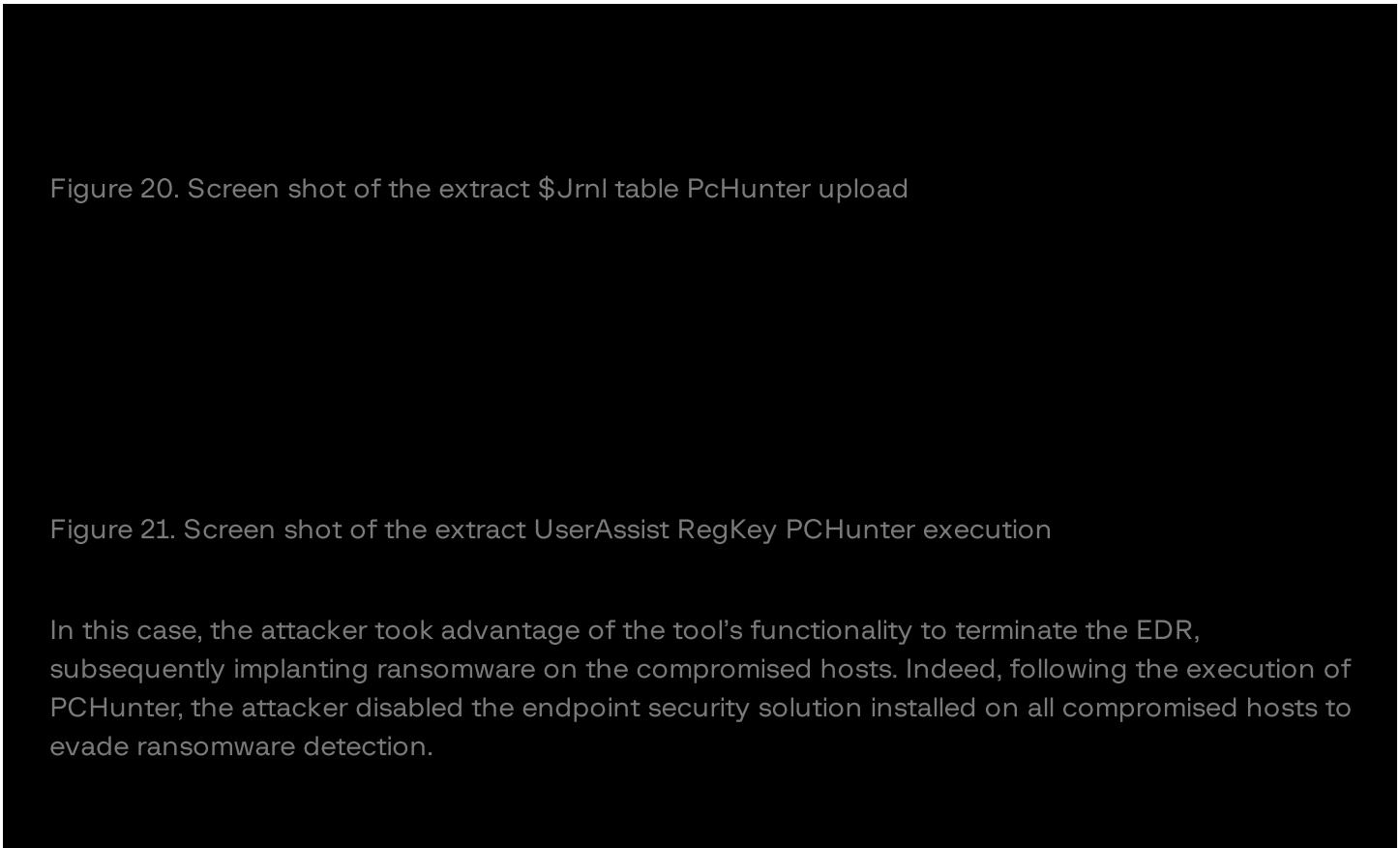


Figure 20. Screen shot of the extract $Jrnl table PcHunter upload

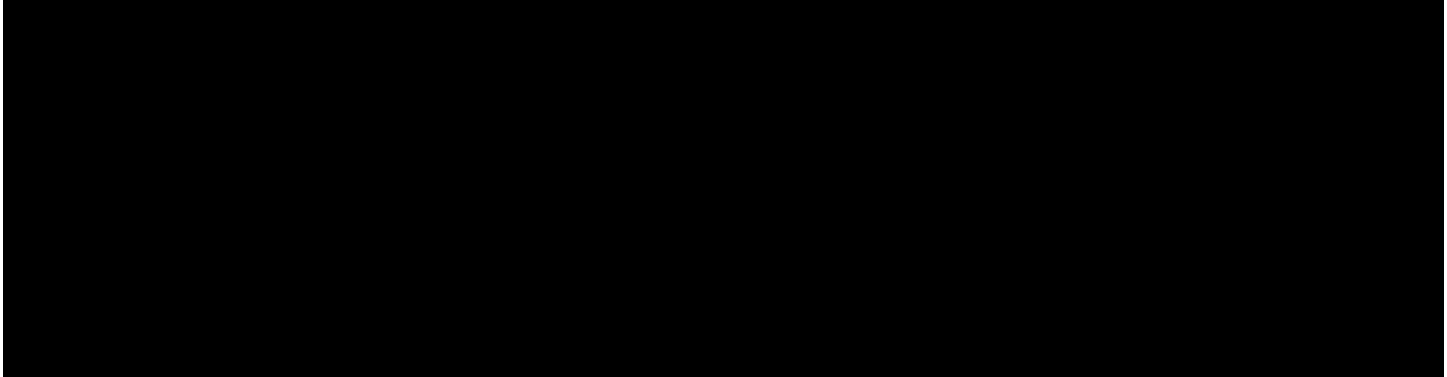


Figure 21. Screen shot of the extract UserAssist RegKey PCHunter execution

In this case, the attacker took advantage of the tool's functionality to terminate the EDR, subsequently implanting ransomware on the compromised hosts. Indeed, following the execution of PCHunter, the attacker disabled the endpoint security solution installed on all compromised hosts to evade ransomware detection.

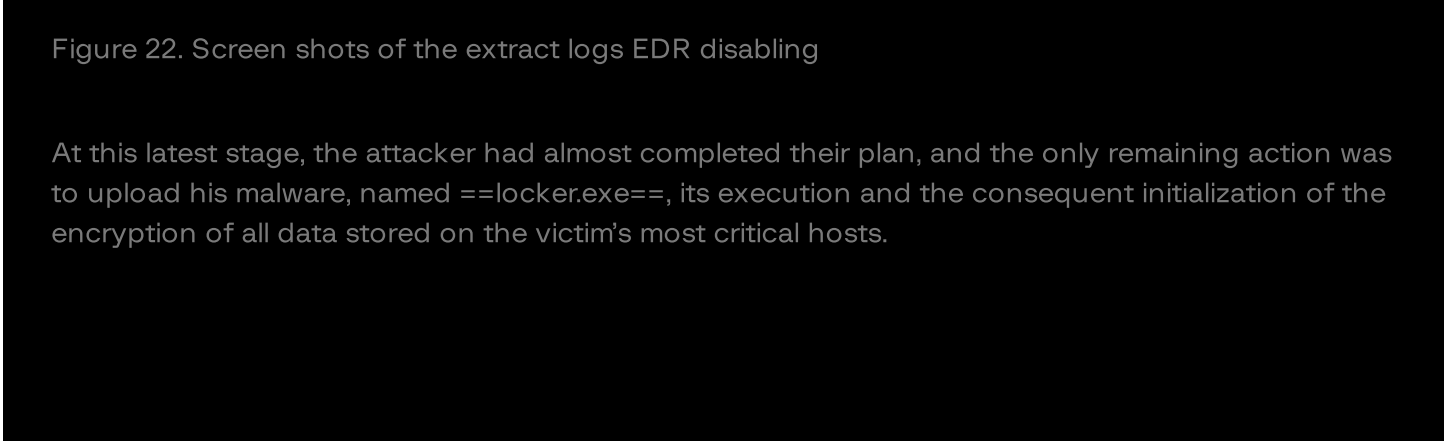


Figure 22. Screen shots of the extract logs EDR disabling

At this latest stage, the attacker had almost completed their plan, and the only remaining action was to upload his malware, named ==locker.exe==, its execution and the consequent initialization of the encryption of all data stored on the victim's most critical hosts.

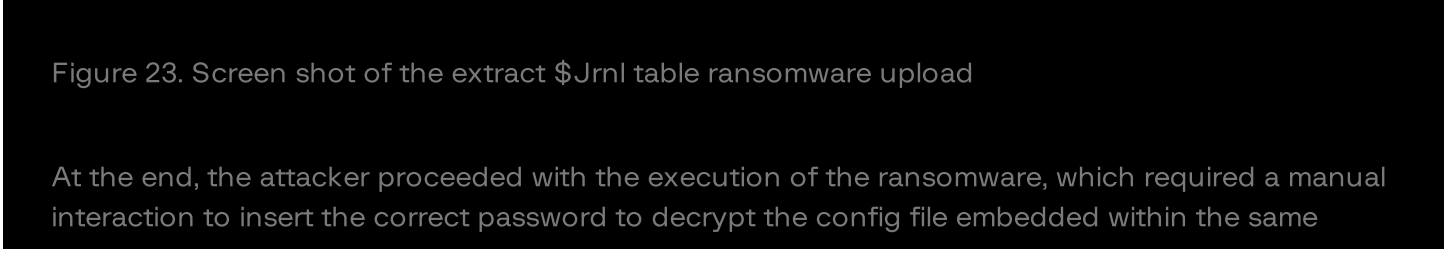


Figure 23. Screen shot of the extract $Jrnl table ransomware upload

At the end, the attacker proceeded with the execution of the ransomware, which required a manual interaction to insert the correct password to decrypt the config file embedded within the same

executable. The correct parameters were then loaded to encrypt all data, inhibit the system recovery and to remove any trace of the TA's actions.

Figure 24. Sample ransomware execution options

The updated version of the ransomware used by the attacker, included various features, which will be dissected in the next section, such as:

1. It retrieved information about virtual machines (VMs) and forcefully stops them through the following embedded base64-encoded command:

Figure 25. Extract Powershell encoded command

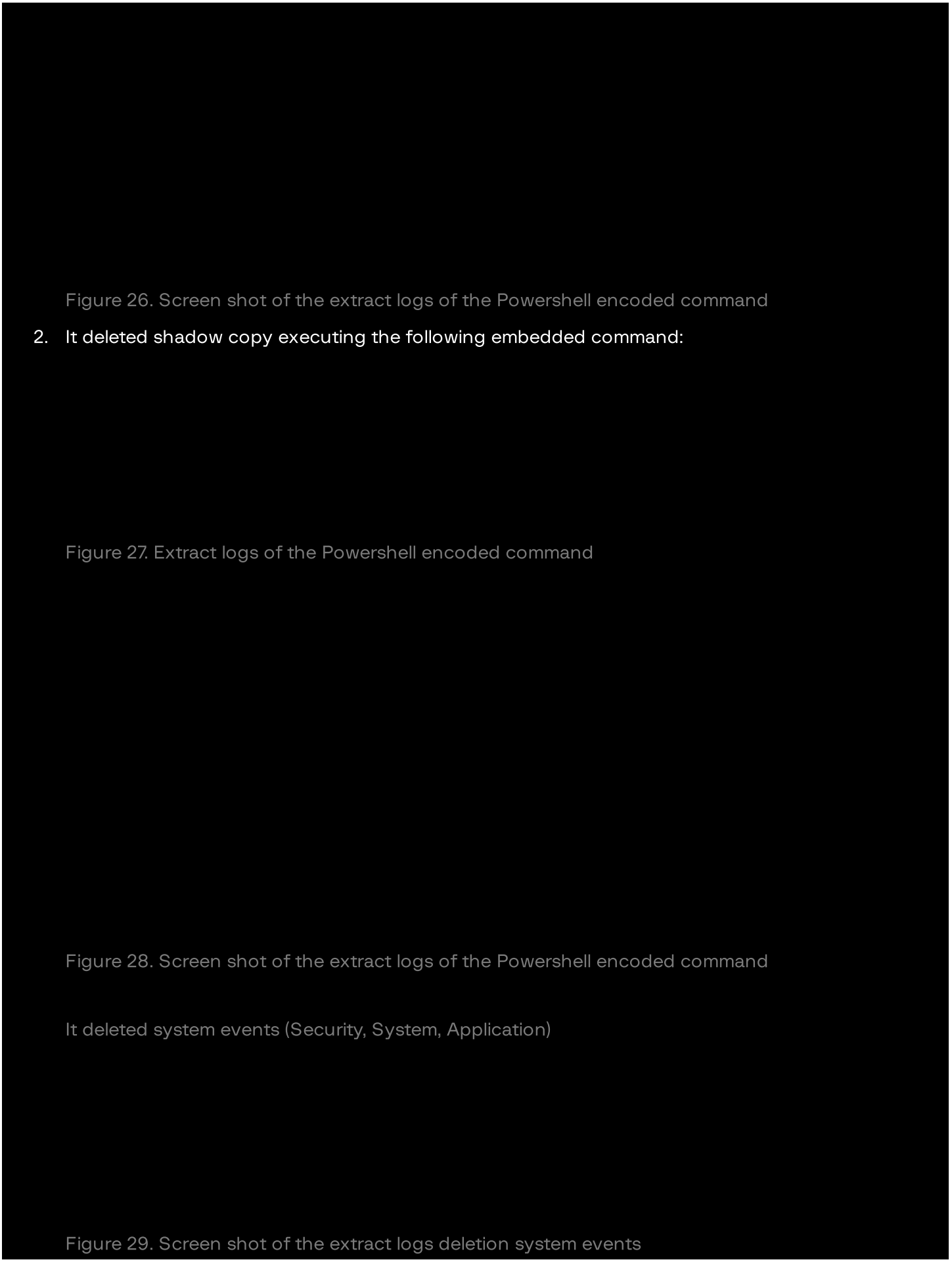


Figure 26. Screen shot of the extract logs of the Powershell encoded command

2. It deleted shadow copy executing the following embedded command:

Figure 27. Extract logs of the Powershell encoded command

Figure 28. Screen shot of the extract logs of the Powershell encoded command

It deleted system events (Security, System, Application)

Figure 29. Screen shot of the extract logs deletion system events

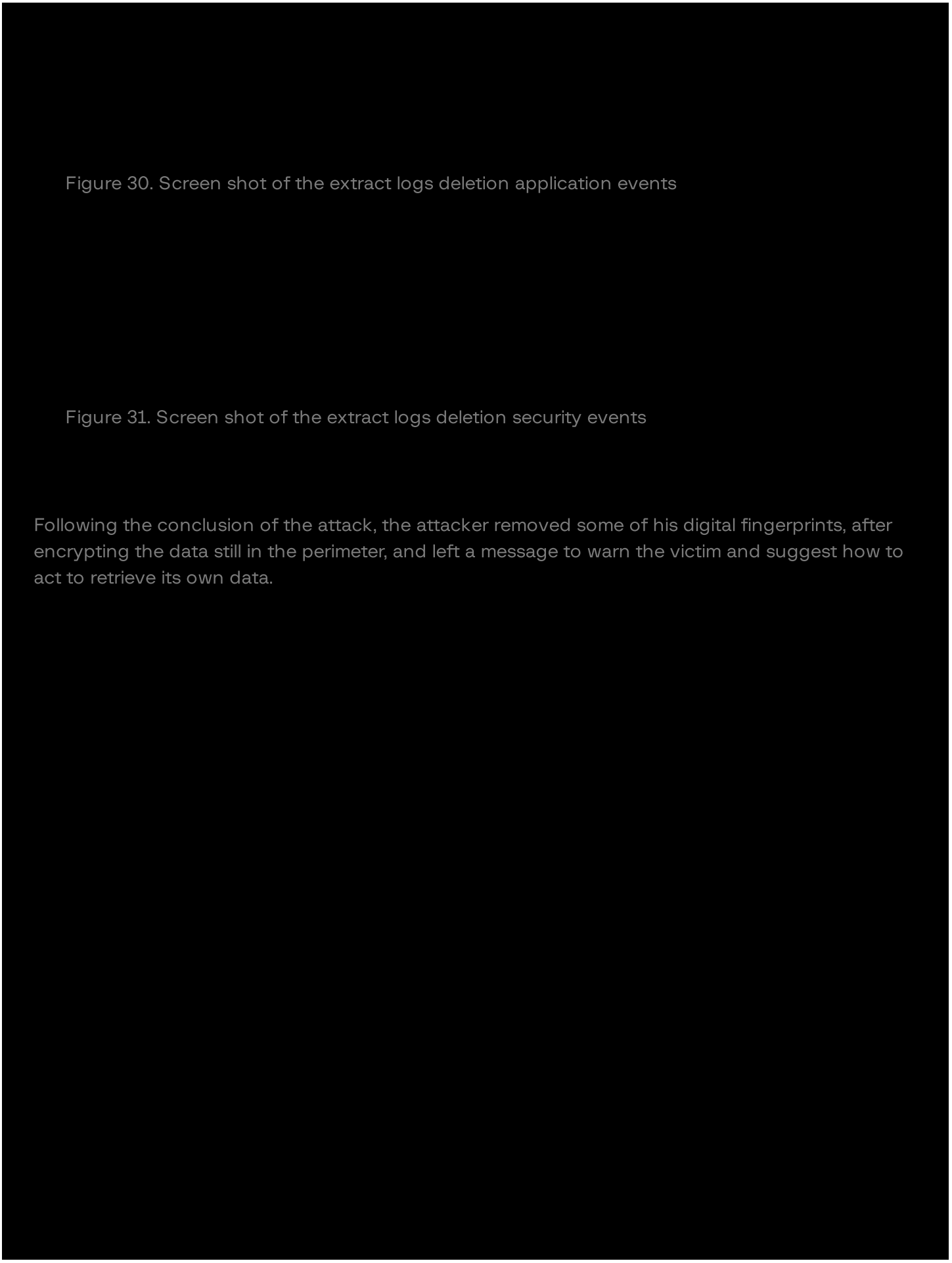


Figure 30. Screen shot of the extract logs deletion application events

Figure 31. Screen shot of the extract logs deletion security events

Following the conclusion of the attack, the attacker removed some of his digital fingerprints, after encrypting the data still in the perimeter, and left a message to warn the victim and suggest how to act to retrieve its own data.

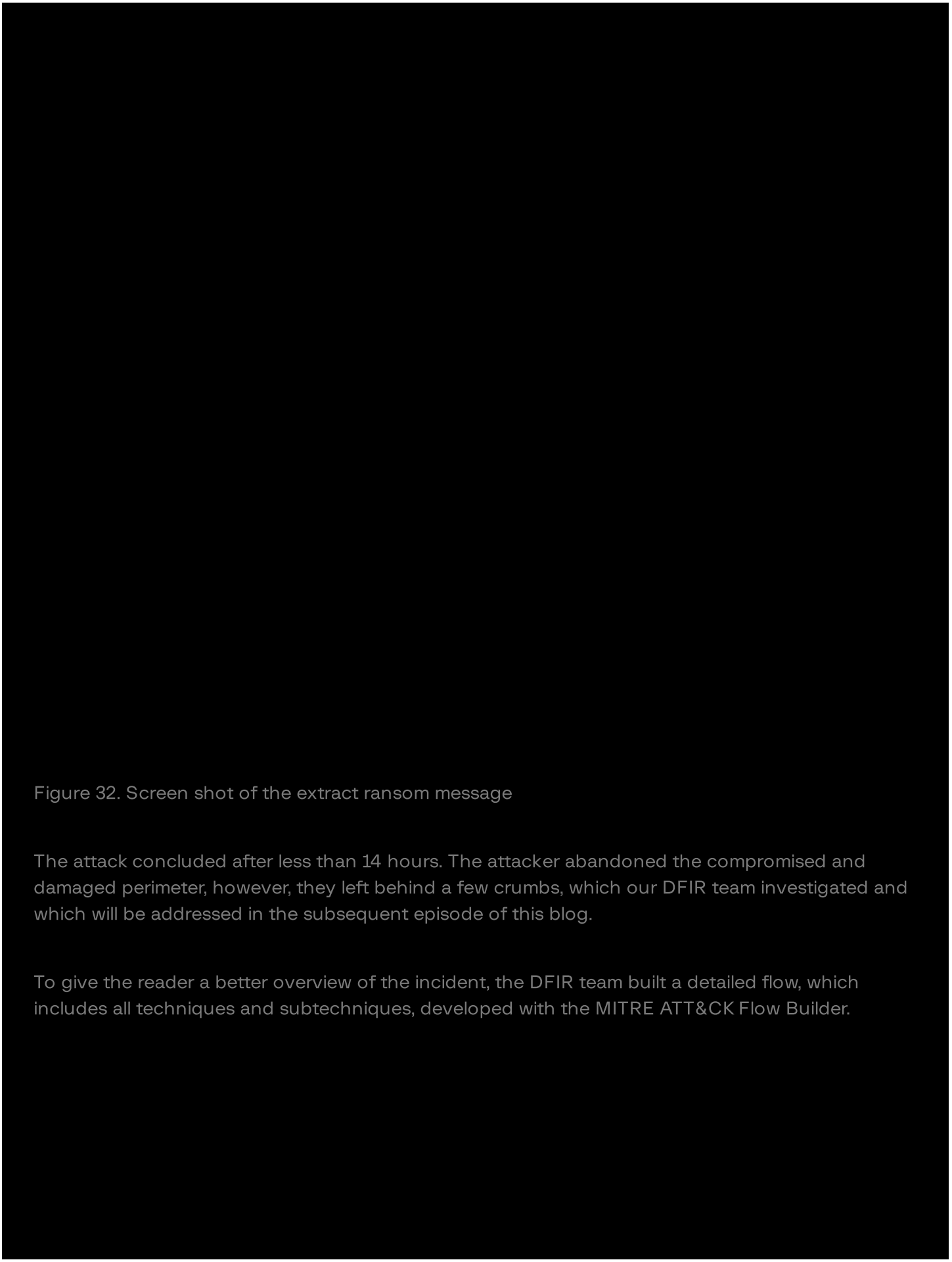
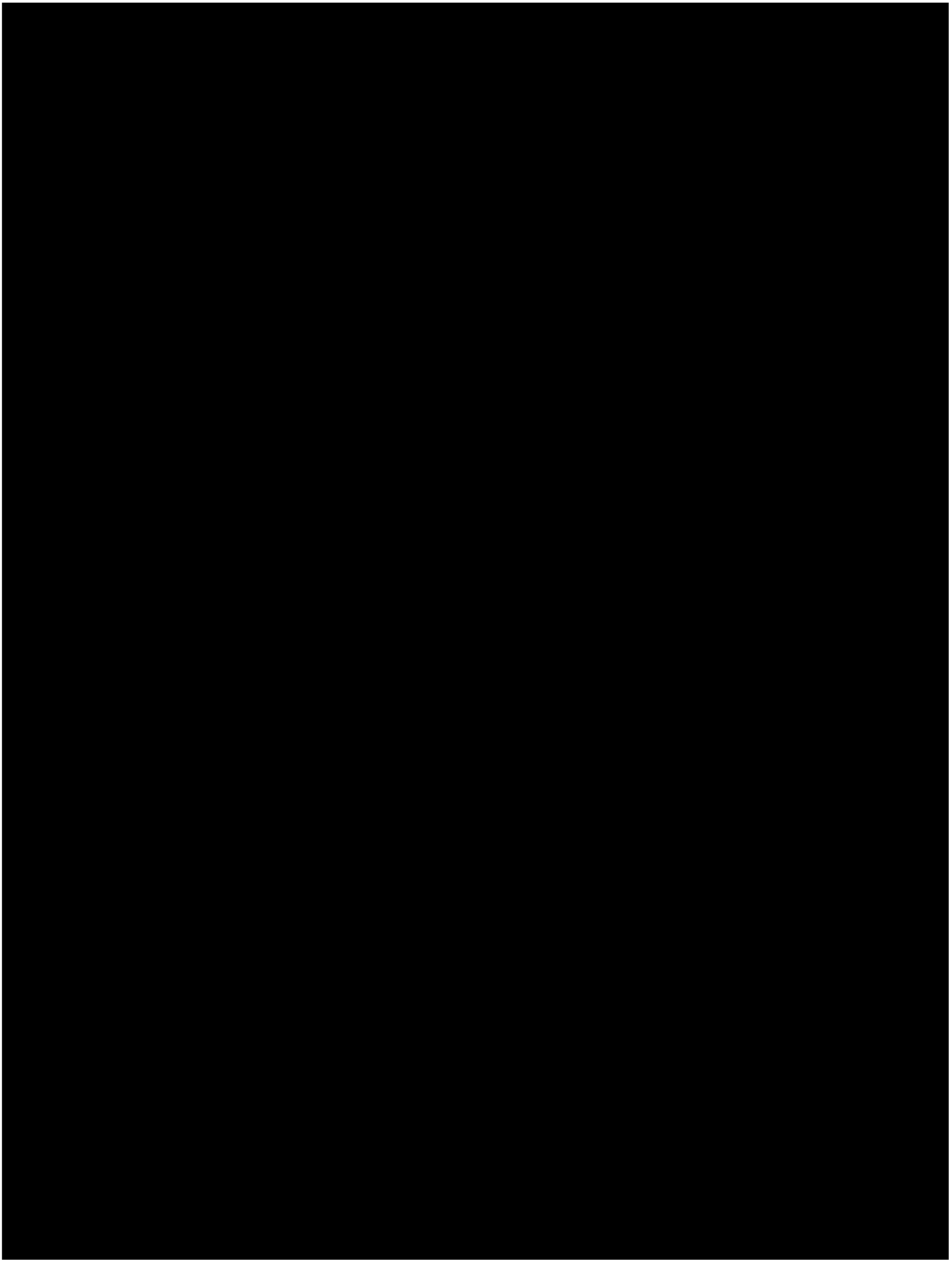


Figure 32. Screen shot of the extract ransom message

The attack concluded after less than 14 hours. The attacker abandoned the compromised and damaged perimeter, however, they left behind a few crumbs, which our DFIR team investigated and which will be addressed in the subsequent episode of this blog.

To give the reader a better overview of the incident, the DFIR team built a detailed flow, which includes all techniques and subtechniques, developed with the MITRE ATT&CK Flow Builder.



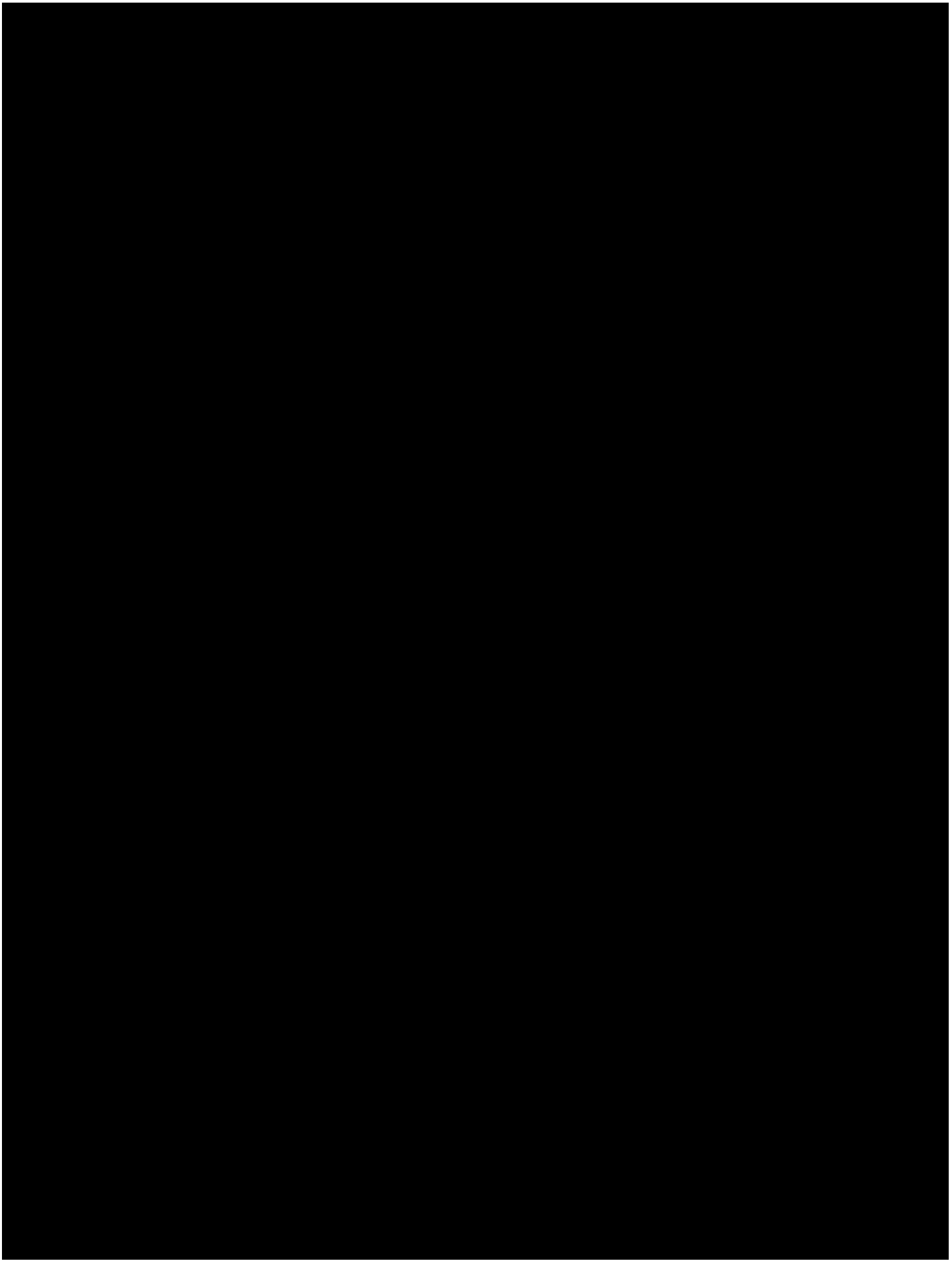


Figure 33. A depiction of the enriched Intelligence driven incident

Dissecting RansomHub Ransomware

The ransomware **Master Public Key** is used to encrypt generated keys for file encryption. By default, it encrypts **1 MB** of file content at regular intervals to optimize performance and speed. Default interval size: **3 MB** (encrypt 1 MB, skip 3 MB), enabling faster encryption for larger files. Additionally, the *Fast Encryption Mode* configured via command-line arguments available in all versions of the ransomware may be used by affiliates to increase the encryption interval size to **9MB** for even faster encryption of very large files.

Additionally, each encrypted file is appended with a custom extension (e.g., .6706c3). At the end of the encryption process, there will be a ransom note (README_<random>.txt) in each encrypted directory.

Ransomware variants and cli options

The **RansomHub ransomware** comes in multiple variants, each tailored for specific platforms and architectures, offering distinct sets of features and command-line options to optimize its functionality for various environments. Below are the command line switches of each version:

Microsoft Windows variant

The Windows variant of the ransomware is distributed as **EXE files** and includes a comprehensive set of command-line options for full control over its execution. This version supports advanced targeting, Safe Mode execution, and encryption of local and networked files (SMB shares).

Command Switch	Description
-cmd	Execute a specific command before encryption.
-disable-net	Disable network interfaces before starting encryption.
-fast	Enable fast encryption mode for quicker processing.

-file	Encrypt specific files only. Example: -file C://1.txt -file D://2.txt.
-host	Target only specific network shares. Example: -host 10.10.10.10 -host 10.10.10.11
-no-folder-filter	Disable folder filtering, allowing all folders to be targeted.
-only-local	Restrict encryption to local disks only.
-pass	Specify a passphrase for execution.

Linux and FreeBSD variant

This variant targets **Linux** and **FreeBSD** systems. It provides fewer options than the Windows version, focusing on encrypting files in specified directories. Without a specified path, the ransomware does not encrypt any files.

Command Switch	Description
-background	Run the ransomware in the background.
-fast	Enable fast encryption mode.
-pass	Specify a passphrase for execution.
-path	Encrypt files in specific directories. Example: -path /var/www -path /var/sqldata.
-verbose	Log actions to the console.
-sleep	Introduce a delay (in minutes) before execution.

Vmware variant

This version targets **VMware ESXi** servers, encrypting files in the **/vmfs/volumes** directory by default. It includes a feature to ignore specific running virtual machines from the encryption process.

Command Switch	Description
-pass	Specify a passphrase for execution.
-path	Specify the directory to encrypt (default is /vmfs/volumes). Example: -path /vmfs/other.
-sleep	Introduce a delay (in minutes) before execution.
-skip_vms	Skip stopping and encrypting VMs listed in a file. Example: -skip_vms skip.txt.
-fast	Enable fast encryption mode.
-verbose	Output encryption logs to the console.

Sftp variant

This variant encrypts files on a **remote SFTP server**. It can connect directly or through a proxy server. It supports multi-threaded encryption for faster processing.

Command Switch	Description
-cmd	Execute a specific command before encryption.
-fast	Enable fast encryption mode.
-host	Specify the target SFTP host. Default: 10.10.10.10:22.
-pass	Specify the passphrase for execution.
-path	Encrypt files in specific directories. Example: -path /volume1 -path /volume2.
-proxy	Use a proxy for connecting to the SFTP server. Example: -proxy socks5://127.0.0.1:1090.
-skip_vm	Skip encrypting specific virtual machine files. Example: -skip_vm “VM1”.
-thread	Set the number of encryption threads (default is 8).

Windows Ransomware Internals

When the malware is executed, it parses the command-line arguments to locate the **-pass** parameter, which is critical for its operation. The provided passphrase is used to decrypt the configuration file, enabling the malware to access its essential parameters. Without the correct **-pass**, the malware will terminate and print “bad config” to the console.

Please find below the ransomware settings:

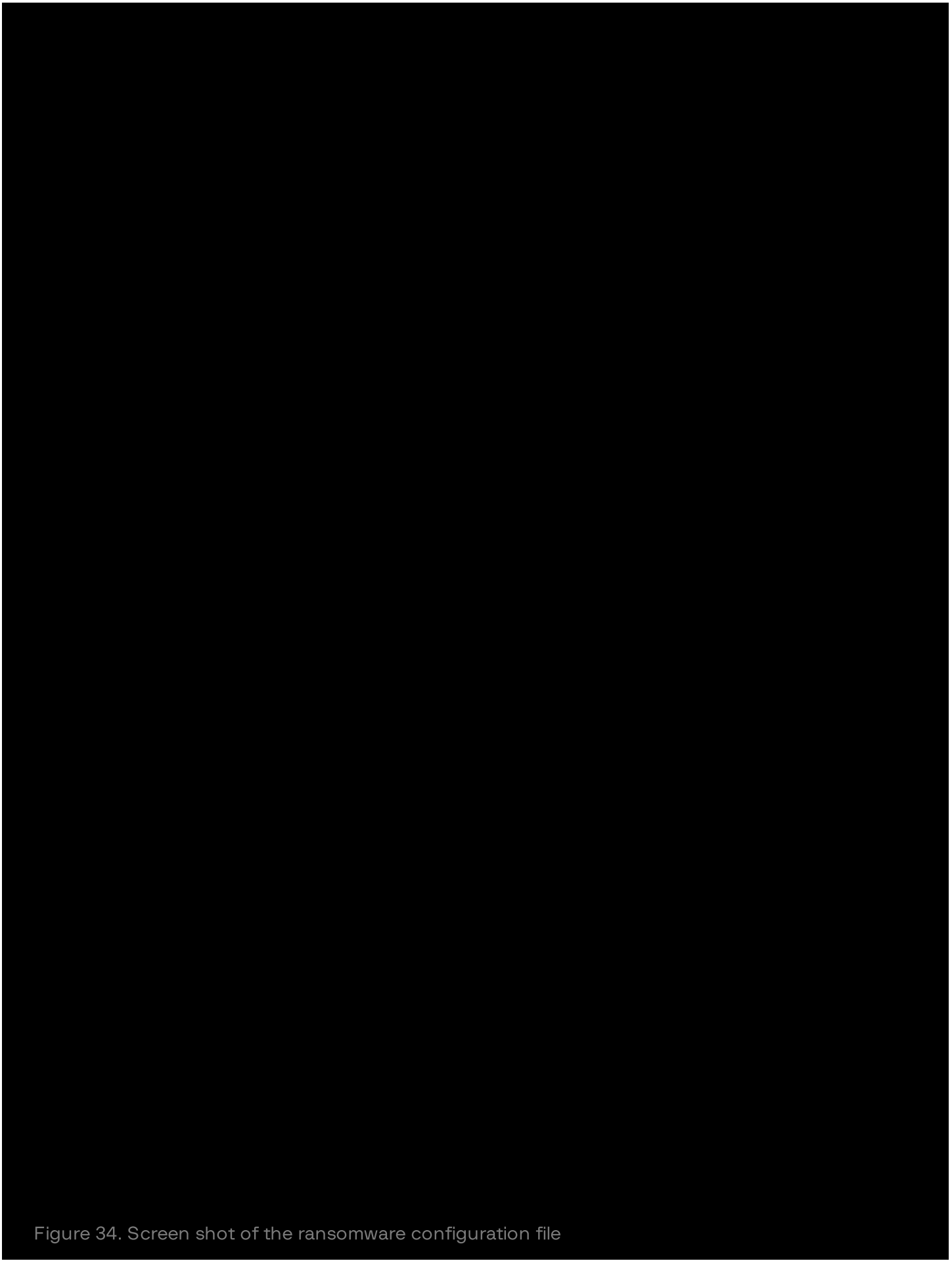


Figure 34. Screen shot of the ransomware configuration file

After the ransomware starts, it checks whether the current machine is included in the **whitelisted machines** previously specified in its configuration in the affiliate panel. If the machine is whitelisted, the ransomware will terminate without proceeding with encryption. Otherwise, it continues its operation.

Next, based on the **configuration settings**, the ransomware determines whether to self-delete. If the *self_delete* flag is set to *true*, the ransomware will delete itself using the following steps:

1. Open a handle to its executable file.
2. Rename the file to “amd64.exe:dea”.
3. Close the handle.
4. Reopen the file handle and set the **delete-on-close** flag.
5. Close the handle again.

This sequence ensures the file is marked for deletion and will be removed. This technique allows the ransomware to erase traces of its presence while still running.

Figure 35. Screen shot of the ransomware self-delete procedures

After completing its initial checks, the ransomware executes any command specified in the -cmd command-line argument, if provided.

Following this, its default behavior is to stop all running virtual machines (VMs). However, VMs listed in the *-skip-vm* command-line argument are excluded from this operation, allowing specific VMs to continue running while others are forcibly stopped.

Figure 36. The command to shut down virtual machines on Hyper-V

Figure 37. White listed machines

Next, the ransomware proceeds to delete all Shadow Copies, modify the **SymLink Evaluation** behavior to Enables remote to local symbolic links and remote to remote symbolic links, and deletes security, system and **Application** Event Logs by executing the following commands:

powershell.exe -Command PowerShell -Command "|"Get-CimInstance Win32_ShadowCopy | Remove-CimInstance|""

cmd.exe /c "|"vssadmin.exe Delete Shadows /all /quiet|""

cmd.exe /c "|"fsutil behavior set SymlinkEvaluation R2L:1|""

cmd.exe /c "|"fsutil behavior set SymlinkEvaluation R2R:1|""

cmd.exe /c wevtutil cl security

cmd.exe /c wevtutil cl system

cmd.exe /c wevtutil cl application

Please note that the commands related to modification of symLink evaluation behavior are also present in the BlackCat and Cicada3301 ransomware.

Afterward, it terminates or stops the services and processes specified in its configuration, ensuring minimal interference during encryption.

Executing in Safe Mode:

When the ***–safeboot*** parameter is provided in the command line argument the ransomware will change system configuration using *bcdedit* to enable safe mode (***bcdedit /set {default} safeboot network***) and also enable autologin to the system with the credential from the configuration. Additionally, it creates autorun registry key for the ransomware with the following command line argument *-safeboot-instance -pass*

Figure 38. Screen shot that shows the capability of the ransomware to enable the autologon changing a system registry key

Figure 39. A screenshot of setting the registry key for the username and password and autorun registry

Figure 40. User credential for autologin

Figure 41. Enabling safe mode

Network Spreading (encrypting other machines):

If the ***net_spread*** flag is set to true, the ransomware initiates network propagation. It enumerates all accessible machines from the currently infected system and uses the credentials provided in its configuration to establish connections to those machines via **SMBv2**.

Once connected, the ransomware:

1. Enumerates all accessible files on the remote machines.
2. Encrypts the files using its encryption algorithm.

3. Writes the ransom note to inform the victim about the attack and payment instructions.

This network spreading capability allows the ransomware to extend its impact across the environment, targeting multiple systems simultaneously.

Figure 42. Getting accessible files on the remote host

Figure 43. Writing the ransome note to the remote host

Figure 44. Requesting access to files and directory

Files Encryption:

After parsing all command-line switches and reading its configuration, the ransomware determines its mode of operation: whether to encrypt **local files only**, **remote shares**, or both, and whether it will perform network propagation.

It then enumerates all directories, dropping the ransomware note in each one before beginning the encryption process. For each file, the ransomware:

Generates a random key:

- This random key is used to encrypt the file.
- The key itself is then encrypted using the master public key from the configuration.

Encrypts and stores metadata:

- The ransomware calculates the number of blocks in the file based on its size and the interval size (default: encrypt 1 MB, skip 3 MB).
- The metadata—comprising the encrypted key, block count, and the master public key—is written to the end of the file.

Encrypts file content:

- Reads the file content (including the metadata already written).
- Encrypts the content using **AES-CBC mode** encryption.

Finalizes the file:

- Rewrites the metadata at the end of the file after completing the encryption process.

Figure 45. A screenshot showing the ransom note

SFTP Ransomware Internals

This version of the ransomware is specifically designed to target **SFTP servers** and does not encrypt any local files. It connects to the target server either directly, which is the default operation, or through a proxy server which is passed in the command line argument **-proxy** socks5://IP:PORT, ensuring the attacker's IP address remains hidden.

Once connected, the ransomware:

1. **Authenticates** with the SFTP server using the username and password it requests or retrieves from its configuration.
2. **Enumerates files** within the root directory of the SFTP server.
3. **Encrypts the files**, rendering them inaccessible to the server's users.

Figure 46. Requesting credentials to access the SFTP server

Ransomware killer

This killer tool acts as a terminator that abuses a vulnerable driver that has exposed interface to kill security producers. It has two stages: the loader and the final payload.

Loader:

It first checks the "**-pass**" command line switch and make sure it is 64 characters then get the shellcode from the resources section and write it to disk under the name **config.bin**

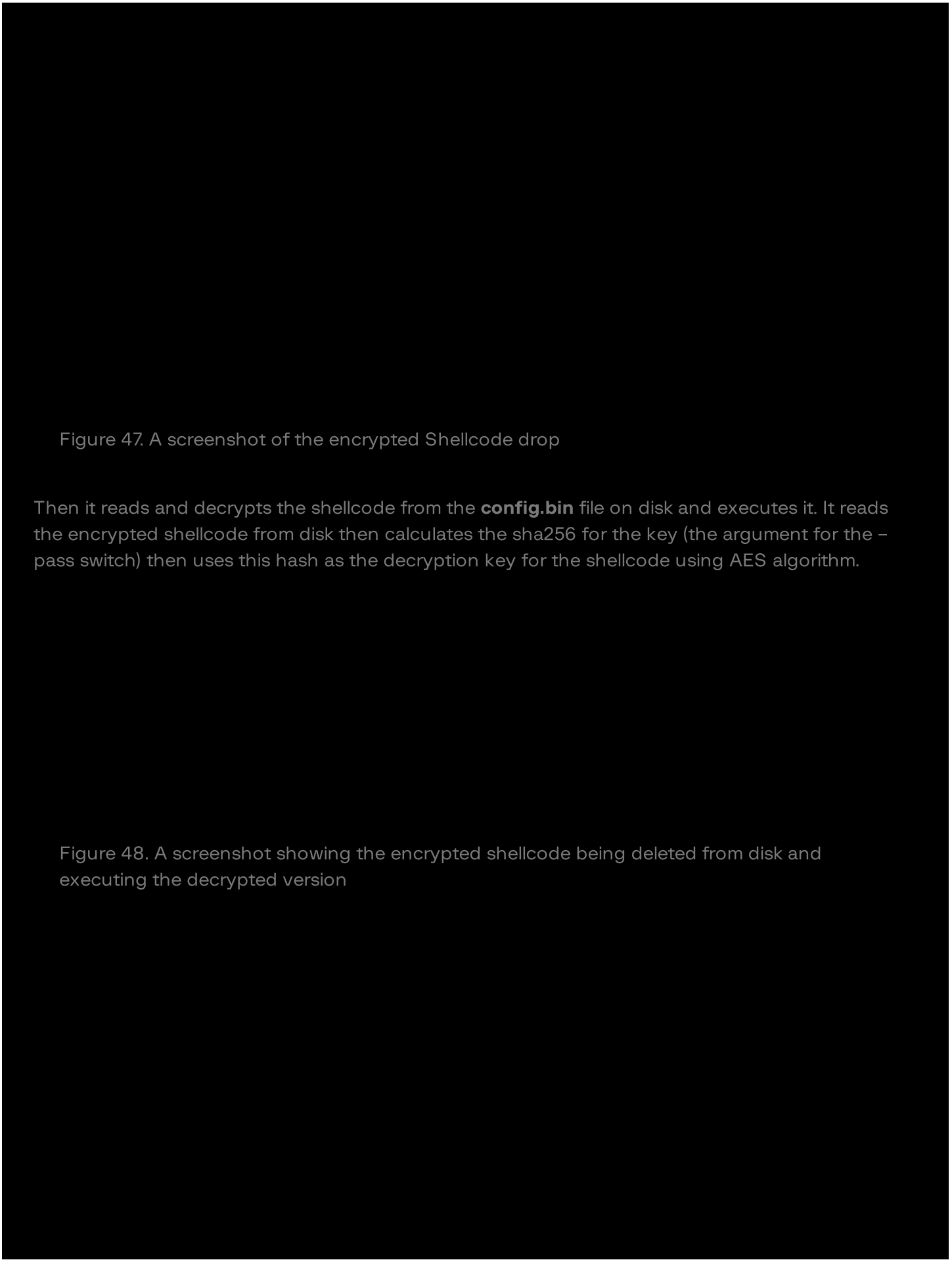


Figure 47. A screenshot of the encrypted Shellcode drop

Then it reads and decrypts the shellcode from the **config.bin** file on disk and executes it. It reads the encrypted shellcode from disk then calculates the sha256 for the key (the argument for the –pass switch) then uses this hash as the decryption key for the shellcode using AES algorithm.

Figure 48. A screenshot showing the encrypted shellcode being deleted from disk and executing the decrypted version

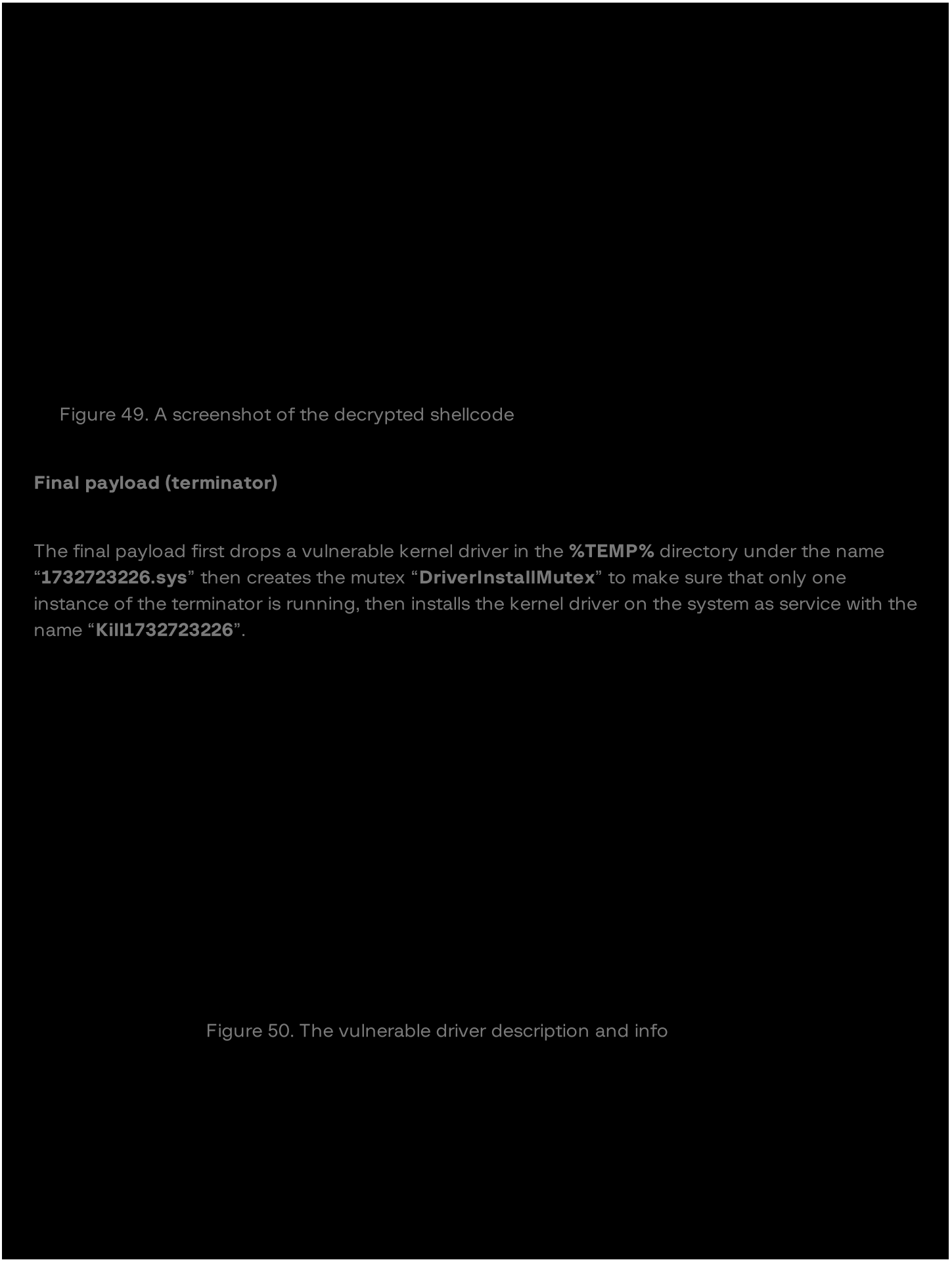


Figure 49. A screenshot of the decrypted shellcode

Final payload (terminator)

The final payload first drops a vulnerable kernel driver in the **%TEMP%** directory under the name "**1732723226.sys**" then creates the mutex "**DriverInstallMutex**" to make sure that only one instance of the terminator is running, then installs the kernel driver on the system as service with the name "**Kill1732723226**".

Figure 50. The vulnerable driver description and info

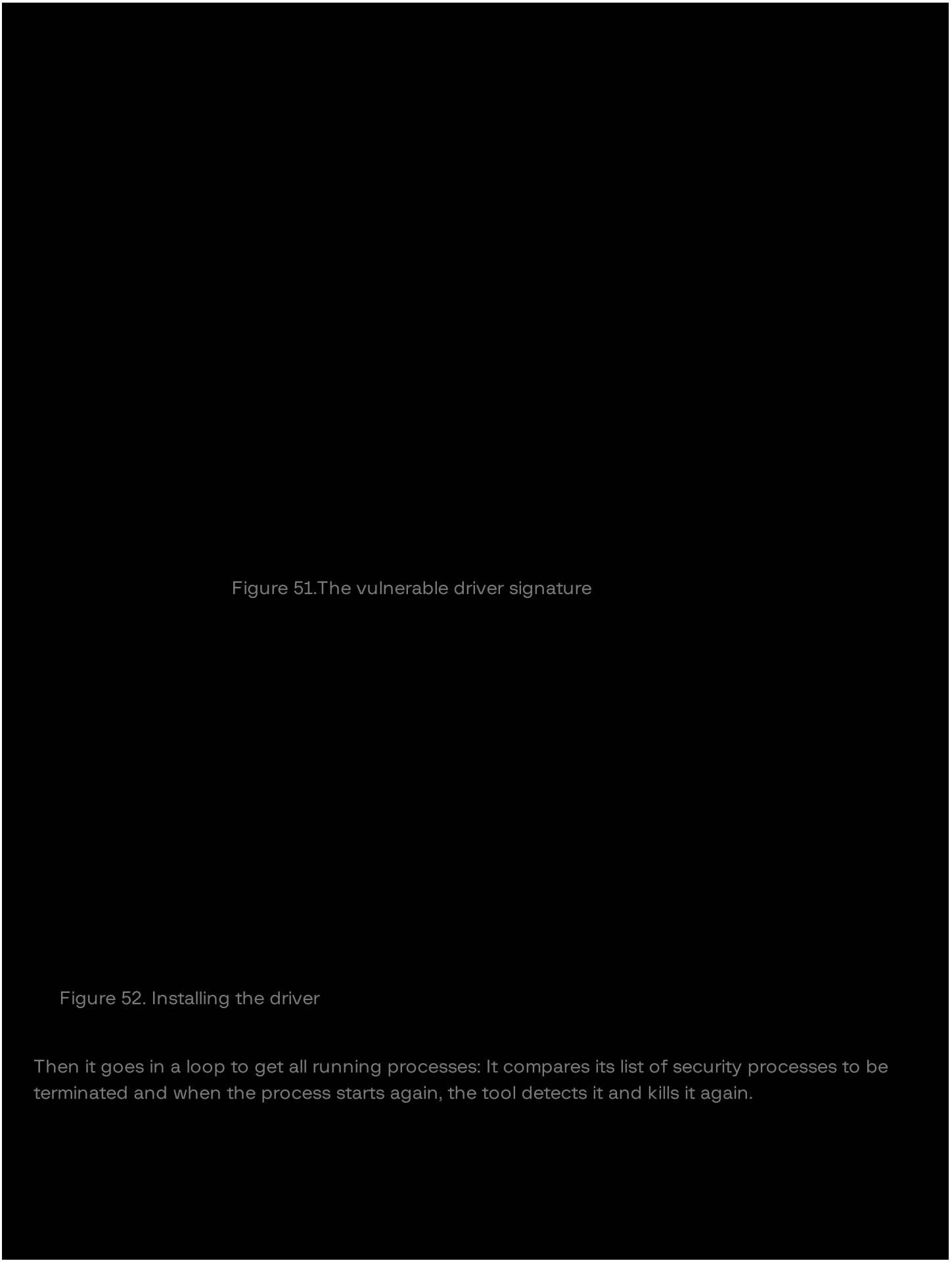


Figure 51.The vulnerable driver signature

Figure 52. Installing the driver

Then it goes in a loop to get all running processes: It compares its list of security processes to be terminated and when the process starts again, the tool detects it and kills it again.

Figure 53. A screenshot showing communication with the kernel driver to kill the AV process

Figure 54. A list of AV/EDR process to kill

Figure 55. A screenshot of the console Message

List of AV/EDR to kill:

Process Name			
MsMpEng.exe	AmSvc.exe	TaniumCX.exe	Ntrtscan.exe
MsSense.exe	CrAmTray.exe	Traps.exe	TmWSCSvc.exe

Process Name			
SenseIR.exe	CrsSvc.exe	cyserver.exe	PccNTMon.exe
SenseNdr.exe	CybereasonAV.exe	CyvrFsFlt.exe	TMBMSRV.exe
winlogbeat.exe	RepMgr.exe	fortiedr.exe	CNTAoSMgr.exe
elastic-agent.exe	RepUtils.exe	EIConnector.exe	TmCCSF.exe
filebeat.exe	RepUx.exe	hurukai.exe	SophosClean.exe

Conclusions

The insights shared by Group-IB's Cyber Threat Intelligence group, combined with the findings from the Incident Response case handled by the DFIR team highlight the dynamic nature of the ransomware landscape.

The origins of the RansomHub group, its offensive operations, and its overlapping characteristics with other groups confirm the existence of a vivid cybercrime ecosystem. This environment thrives on the sharing, reusing, and rebranding of tools and source codes, fueling a robust underground market where high-profile victims, infamous groups, and substantial sums of money play central roles.

Within this dynamic context, Ransomhub has quickly become a point of reference among researchers, responders and affiliates-along with its unfortunate victims. The group has demonstrated the ability to rapidly adapt and evolve its TTPs, tools, and capabilities, often surprising even seasoned professionals in the field.

It is evident that this landscape will continue to evolve, particularly with the growing influence of generative AI platforms. This evolution will undoubtedly present a significant challenge for security researchers for years to come.

Group-IB remains committed to its research and knowledge-sharing efforts and will release further details in the upcoming episodes of this RansomHub-focused trilogy.

Mitre Att&ck Mapping

Yara Rules for Ransomhub

To be able to detect in real-time any sample related to the ransomware developed by RansomHub, Group-IB's analysts built an ad-hoc yara.

```
rule RansomHub_AVKiller
{
meta:
        company = "Group-IB"
        author = "Mahmoud Zohdy"
        date = "2024-09-26"
        description = "Detection for RansomeHub AV Killer"
```

```
        hash0 = "c618c943840269eb753cb389029d331c"
    strings:
        $Argument_1 = "-pass" nocase
        $Argument_2 = "-key" nocase
        $PDB_1 = "Loader.pdb" nocase
        $PDB_2 = "C:\\Users\\Private\\Source\\repos\\Loader\\" nocase
        $InternalName_1 = "Loader.exe" wide nocase
        $InternalName_2 = "Config.exe" wide nocase
        $ProductName = "-Game" wide nocase
        $EncryptedShellCode_1 = "Config.bin" wide nocase
        $EncryptedShellCode_2 = "Data.bin" wide nocase
        $FileDescription = "Loader Config" wide nocase
    condition:
        6 of them
    }

    rule ransomehub_ransome
    {
    meta:
        author = "M.Zohdy Group-ib"
        date = "2025-01-29"
        description = "Detect RansomeHub Ransomware"
        hash0 = "2b7a13837039f4f5ff6aeaa0b135e712"
        hash1 = "35353c1c33c6e8a9c5944ae1b1541512"
        hash2 = "7ea71f9c62e5067da16df949542148da"
        hash3 = "271c4158f9a807fd92bfe65bbd4744cf"
        hash4 = "4b194e9b87c14d1c24aa0603b5bae00f"
        hash5 = "53987a86915d63db7c70998957d5a58d"
        hash6 = "4c6616c79ef2904b238dd9ed45ac6054"
        hash7 = "389c64831dd5d409153eaf352f5537e1"
    strings:
        $string0 = "extension"
        $string1 = "settings"
        $string2 = "master_public_key"
        $string3 = "remove"
        $string4 = "note_full_text"
        $string5 = "note_file_name"
    condition:
        5 of them
    }
```

Share this article

Found it interesting? Don't hesitate to share it to wow your friends or colleagues

GROUP-IB

Products

- Threat Intelligence
- Fraud Protection
- Managed XDR
- Attack Surface Management
- Digital Risk Protection
- Business Email Protection
- Cyber Fraud Intelligence Platform
- Unified Risk Platform
- Integrations

Resources

- Research Hub
- Success Stories
- Knowledge Hub
- Certificates
- Webinars
- Podcasts
- TOP Investigations
- Ransomware Notes
- AI Cybersecurity Hub

Partners

- Partner Program
- MSSP and MDR Partner Program
- Technology Partners
- Partner Locator

Company

- About Group-IB
- Team
- CERT-GIB
- Careers

Internship
Academic Aliance
Sustainability
Media Center
Contact

Subscription plans
Services
Resource Center

Contact

APAC: +65 3159 3798
EU & NA: +31 20 226 90 90
MEA: +971 4 568 1785

info@group-ib.com

Subscribe to stay up to date with the latest cyber threat trends