

# Unleashing the Kraken ransomware group

[blog.talosintelligence.com/kraken-ransomware-group](https://blog.talosintelligence.com/kraken-ransomware-group)

Chetan Raghuprasad

November 13, 2025



## [Threat Spotlight ransomware Threats](#)

- In August 2025, Cisco Talos observed big-game hunting and double extortion attacks carried out by Kraken, a Russian-speaking group that has emerged from the remnants of the HelloKitty ransomware cartel.
- Talos observed in one intrusion that the Kraken actor exploited Server Message Block (SMB) vulnerabilities for initial access, then used tools like Cloudflared for persistence and SSH Filesystem (SSHFS) for data exfiltration before encryption.
- Kraken is a cross-platform ransomware with distinct encryptors for Windows, Linux, and VMware ESXi, targeting a wide range of enterprise environments.
- Kraken ransomware benchmarks a victim machine before starting the encryption process, a feature rarely seen in ransomware.

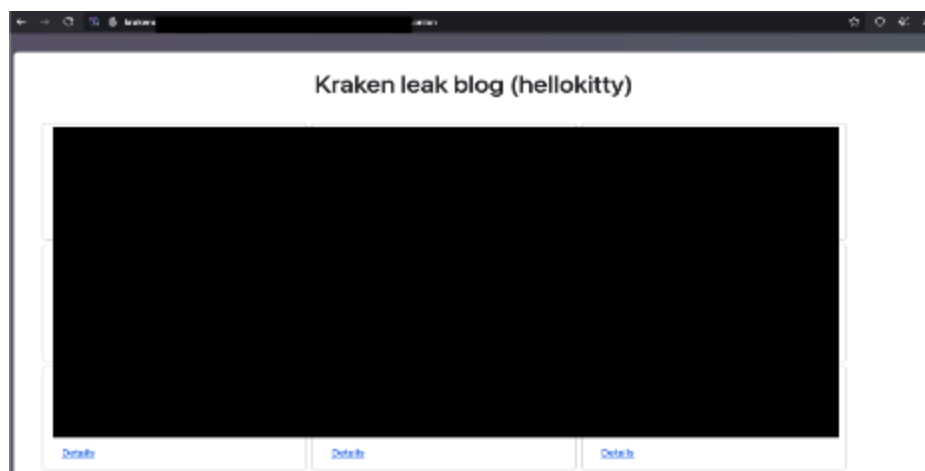
- Talos also observed the announcement of a new underground forum, “The Last Haven Board,” on Kraken’s data leak blog, aimed at creating an anonymous and secure communication channel for the cybercrime underground.

## Who is Kraken?

---

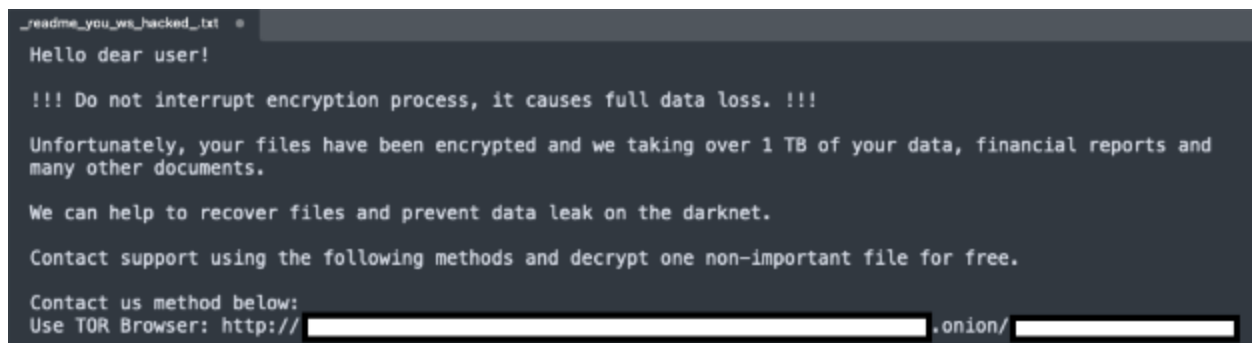
The Kraken ransomware group, which emerged in February 2025, employs a double extortion technique and appears to be opportunistic, as it has not concentrated on any specific business verticals. According to Kraken’s leak site, victims span various geographies, including the United States, the United Kingdom, Canada, Denmark, Panama, and Kuwait.

Like other operators in the double extortion space, Kraken also operates a data leak site to disclose the stolen data of victims who do not meet their ransom demands.



*Figure 1. Kraken data leak blog.*

Kraken encrypts the victim’s environment, uses the .zpssc file extension for the encrypted files, and drops a ransom note titled “readme\_you\_ws\_hacked.txt.” In the ransom note, the actor threatens the victims by stating that they have stolen and encrypted their confidential data. They instruct the victim to contact them using an onion URL to prevent posting to their leak site.



*Figure 2. Kraken ransom note.*

Talos observed in one of the instances that the actor demanded a ransom of around 1 million USD to be paid in Bitcoin to the actor's wallet address. Kraken assures victims that after the successful payment, they will decrypt the environment and guarantee the non-disclosure of stolen data.

## Ties to HelloKitty

---

Kraken, a Russian-speaking gang, is suspected to have emerged from the ashes of the HelloKitty ransomware cartel or to have been established by some of its former members, according to [external reports](#). The title of the Kraken data leak site explicitly mentions the HelloKitty ransomware group name. Additionally, Talos has observed that Kraken and HelloKitty use the same ransom note filename, indicating a possible link between the two groups.

In September 2025, the Kraken group announced a new underground forum called "The Last Haven Board" in their data leak blog. According to its description, Last Haven's primary objective is to create an anonymous and secure environment for communication within the cybercrime underground. Talos observed that the Last Haven forum administrator announced support and collaboration from the HelloKitty team and WeaCorp, an exploit buyer organization, suggesting the possible involvement of HelloKitty operators with the Kraken group.

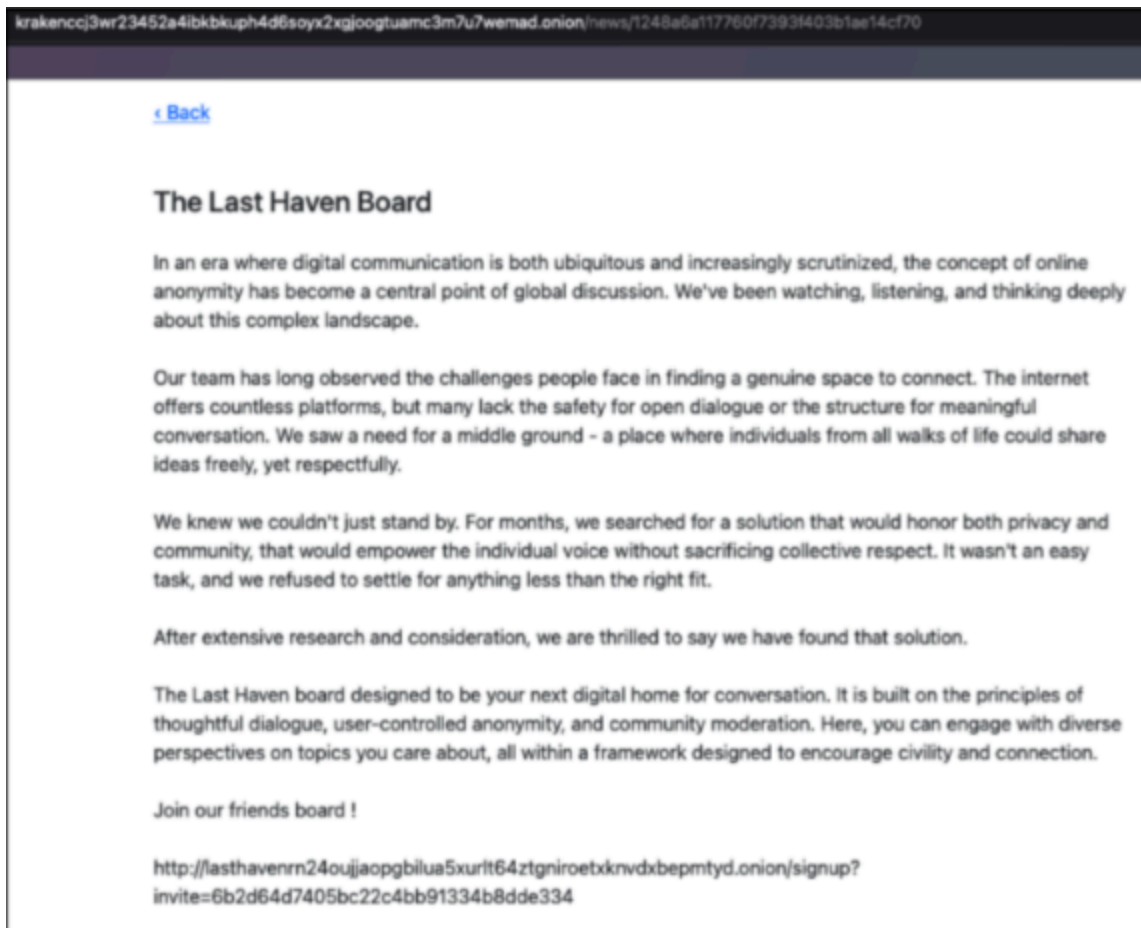


Figure 3. Last Haven underground forum announcement on Kraken data leak blog.

## Infection chain

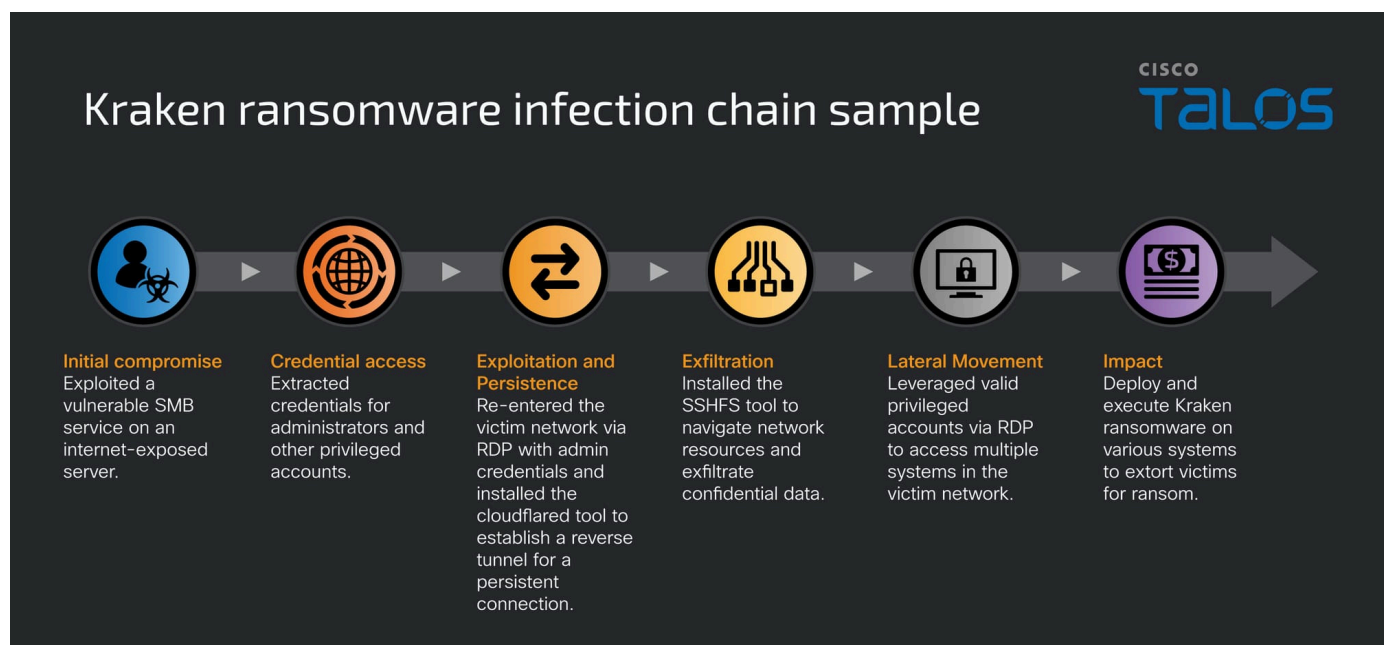


Figure 4. Kraken infection chain.

In August 2025, Cisco Talos Incident Response (Talos IR) observed in one instance that the Kraken ransomware actor gained initial access to the victim's machine by exploiting an existing vulnerability in the SMB service on servers exposed to the internet. Once they established their foothold on the victim's machine, they extracted valid administrators' and other privileged accounts' credentials. Subsequently, they re-entered the victim environment through a Remote Desktop connection using the exfiltrated privileged account credentials.

After re-entering the victim machine, the attacker established a persistent connection by installing the Cloudflared tool and configuring a reverse tunnel on the victim's machine. Additionally, the attacker installed the SSHFS tool on the victim machine, utilizing it to navigate the victim's environment and exfiltrate sensitive data. The attacker then deployed the Kraken ransomware binary and moved laterally to other machines connected to the infected machine through Remote Desktop Protocol (RDP) connections, using the stolen privileged user accounts to deploy the ransomware binaries. Through this persistent remote connection, the attacker executed commands to run the ransomware on multiple systems within the victim's environment.

## Kraken ransomware analysis

---

Kraken ransomware is a sophisticated ransomware family with variants that target Windows, Linux, and ESXi systems. This ransomware offers extensive command-line options, providing operational flexibility for the actors who utilize Kraken ransomware in their attacks. It has the capability for either full or partial encryption of targeted files, along with features that allow for the encryption of specific files, including SQL databases and network shares.

To encrypt targeted files, Kraken ransomware employs RSA encryption algorithms with a key length of 4096 bits and ChaCha20 symmetric encryption. Additionally, the ransomware features encryption benchmarking capabilities to assess how quickly it can operate on the victim's machine without causing system overload, ensuring maximum damage in minimal time while evading detection through resource exhaustion.

Talos observed that the attacker executed the commands on Windows and ESXi environments to run the encryptor program. The Kraken encryptor is engineered with various command line arguments that the attacker could leverage depending on the victim's environment.

### Commands for Windows machine:

---

```
Encryptor[.]exe -key <32-byte key> -path <\\targeted path for encryption> -timeout -d
```

Command-line options	Description
-path	Targeted drive or file's location in the victim machine

-timeout N	Delays the execution of the encryptor for N seconds
-solid	Full file encryption without blocks
-step N	Numbers of blocks of a file to encrypt
-limit N	Limit encryption to first N megabytes
-d	For the execution through remote SSH connection
-noteonly	Drops ransom note only without performing the encryption
-tests	Run encryption performance tests
-tempfile	Temporary test file path
-tempsize	Test file size in megabytes

## Commands for Linux/ESXi:

---

```
chmod +x ./encryptor[.]elf && ./encryptor[.]elf -path -d -timeout
```

Command-line options	Description
-path	Targeted encryption path
-timeout N	Delays the execution of the encryptor for N seconds
-solid	Full file encryption without blocks
-step N	Numbers of blocks of a file to encrypt
-limit N	Limit encryption to first N megabytes

-d	Runs as daemon and execution through remote SSH connection
-noteonly	Drops ransom note only without performing the encryption
-tests	Run encryption performance tests
-tempfile	Temporary test file path
-tempsize	Test file size in megabytes
-all	Encrypt all files
-nolsof	Disable lsof checking
-nokillallvms	Skip VM termination

## Kraken Windows encryptor

---

The Windows version of Kraken ransomware is a 32-bit executable written in C++ and possibly obfuscated using a Golang-based packer. The ransomware exhibits features such as anti-reinfection checks, anti-analysis, and anti-recovery, and it encrypts the targeted files, appending the .zpsc file extension to the encrypted files.

### Initial execution phase

---

In the initial phase of execution, Kraken processes the command line parameters and performs the anti-reinfection checks on the victim machine to avoid double-encryption. The actor has employed anti-reinfection checks to effectively manage the decryption keys.

Kraken ransomware disables the WoW64 filesystem redirection on the victim machine by using the function `Wow64EnableWow64FsRedirection` with the argument “\0 (False)” to enable the 32-bit binary to access the 64-bit files on Windows machine.

WoW64 is a compatibility layer on a 64-bit Windows operating system that allows 32-bit applications to run seamlessly. The key feature of WoW64 is file system redirection, which ensures that when a 32-bit application attempts to access the “C:\Windows\System32” folder, WoW64 redirects it to “C:\Windows\SysWow64”, allowing the 32-bit application to load the correct 32-bit version of system DLLs.



```

LAB_00426039                                     XREF[1... 00425ff7(j)]
00426039      PUSH      0x8003
0042603e      CALL     dword ptr [->KERNEL32.DLL::SetErrorMode]
00426044      PUSH     lpTopLevelExceptionFilter_0041e060
00426049      CALL     dword ptr [->KERNEL32.DLL::SetUnhandledExceptionFilter]
0042604f      PUSH     0x0
00426051      CALL     dword ptr [->KERNEL32.DLL::Wow64EnableWow64FsRedirection]
00426057      CALL     enable_seDebugPrivilege

```

Figure 5. Function snippet disabling the WoW64 redirection.

Kraken ransomware, after disabling the WoW64 redirection, modifies its process token privilege, enabling the debugging rights. This privilege is essential for ransomware to access and encrypt files belonging to other processes. Further, the ransomware encrypts the local drives, network shares, and SQL database files and disables the backup services on the 64-bit Windows operating system. All these operations of the 32-bit ransomware binary would require access to the folder “C:\Windows\System32”. Disabling the redirection in Wow64 will enable the 32-bit ransomware binary to access the “C:\Windows\System32” folder on the 64-bit Windows operating system.

## Anti-analysis and anti-recovery techniques

Kraken ransomware utilizes anti-analysis techniques to evade detection, complicate analysis, and prevent execution in sandbox environments.

The ransomware employs extensive control flow obfuscation with multiple conditional loops throughout the code, concealing the actual control flow paths and increasing complexity for static analysis and pattern matching for signature generation.

It also manipulates system exception handlers to prevent Windows error dialogs from appearing by executing `SetErrorMode` function with the value 0x8003 which is a bitwise OR combination of three Windows error mode flags:

- SEM\_FAILCRITICALERRORS (0x0001) - no critical error handler message box
- SEM\_NOGPFAULTERRORBOX (0x0002) - no general protection fault error box
- SEM\_NOOPENFILEERRORBOX (0x8000) - no open file error box

```

LAB_00426039                                     XREF[1... 00425ff7(j)]
00426039      PUSH      0x8003
0042603e      CALL     dword ptr [->KERNEL32.DLL::SetErrorMode]
00426044      PUSH     lpTopLevelExceptionFilter_0041e060
00426049      CALL     dword ptr [->KERNEL32.DLL::SetUnhandledExceptionFilter]

```

Figure 6. Function snippet sets the error mode flags.

It employs a sleep-based execution delay to evade sandbox analysis, stops the backup services, and executes the embedded command to remove all restore points on the victim machine.



```
vssadmin delete shadows /all /quite
```

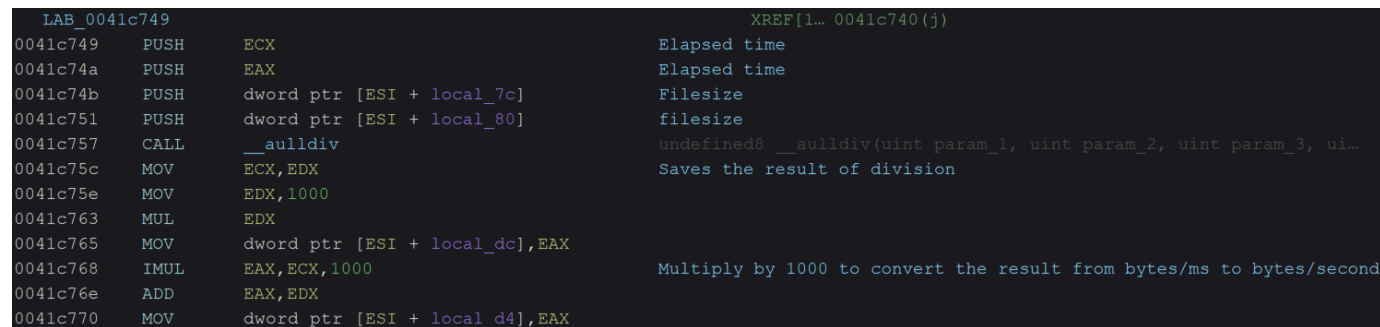
It also deletes the recycle bin using the Windows function `SHEmptyRecycleBinA`.

## Encryption performance testing and benchmarking

Kraken ransomware has the ability to conduct performance testing on the victim's machine before initiating the actual encryption. An actor can use this feature through command line options such as “-tests,” “-tempfile,” and “-tempsize” to assess the victim machine’s performance and optimize the ransomware encryption process.

Kraken does this by first creating a temporary test file, using the path and filename specified via the “-tempfile” parameter. It then populates this file with random data, writing in 1MB chunks until the total size defined by the “-tempsize” parameter is reached. To time the core operation, the module records the start time with the `clock_gettime` function, performs the actual encryption on the test file, and then records the end time. Finally, it calculates the elapsed time and computes the encryption speed for the victim machine, expressed in MB/s, using the formula:

$$\text{Speed} = ((\text{total bytes} / \text{elapsed time}) * 1000) / 1048576.$$



```
LAB_0041c749                                     XREF[1... 0041c740(j)
0041c749      PUSH      ECX                        Elapsed time
0041c74a      PUSH      EAX                        Elapsed time
0041c74b      PUSH      dword ptr [ESI + local_7c] Filesize
0041c751      PUSH      dword ptr [ESI + local_80] filesize
0041c757      CALL      __aulldiv                    undefined8 __aulldiv(uint param_1, uint param_2, uint param_3, ui...
0041c75c      MOV       ECX,EDX                        Saves the result of division
0041c75e      MOV       EDX,1000
0041c763      MUL       EDX
0041c765      MOV       dword ptr [ESI + local_dc],EAX
0041c768      IMUL      EAX,ECX,1000                      Multiply by 1000 to convert the result from bytes/ms to bytes/second
0041c76e      ADD       EAX,EDX
0041c770      MOV       dword ptr [ESI + local_d4],EAX
```

Figure 7. Function snippet performs calculation.

Based on the throughput results, the function validates if the attacker should choose full encryption mode or partial encryption mode with the maximum file size chunks to encrypt. After the performance testing process, it removes the test file using the function `unlink()`.

## Parallel encryption operation

The Kraken Windows encryptor has four encryption modules including SQL database, Network share, Local drive, and Hyper-V encryption. Based on the command-line flags provided by the attacker, the encryptor determines which encryption module to execute.

The SQL database encryption module encrypts Microsoft SQL server databases. To target database files, the module accesses the Microsoft SQL Server registry keys on the victim machine, specifically querying “HKLM\SOFTWARE\Microsoft\Microsoft SQL Server” and its “Instance Names\SQL” subkey to search for the “MSSQLSERVER” and “SQLEXPRESS”

instances. Upon locating an instance, it retrieves the "SQLDataRoot" registry value to determine the path to the database files. The module then validates that these paths exist using the PathFileExistsW Windows API before proceeding to encrypt the database files.

The network share encryption module enumerates and encrypts accessible network shares by using Windows WNet APIs to detect both mapped and unmapped network locations, specifying RESOURCETYPE\_DISK and RESOURCETYPE\_ANY. During enumeration, it iterates through the discovered network resources but explicitly skips the ADMIN\$ and IPC\$ shares. For each accessible network share it finds, the module creates dedicated encryption worker threads to handle the encryption process.

```
004150e0  PUSH     ESI
004150e1  PUSH     0x3                                RESOURCETYPE_ANY (enumerate all resources)
004150e3  PUSH     0x0
004150e5  CALL     FUN_00404470                       Enumerate network resource; uses WNetEnumResourceW
004150ea  ADD      ESP, 0xc
004150ed  LEA      EAX=>local_64, [EBP + -0x60]
004150f0  PUSH     EAX
004150f1  PUSH     0x1                                RESOURCETYPE_DISK (disk shares)
004150f3  PUSH     0x0
004150f5  CALL     FUN_00404470                       Enumerate network resources; uses WNetEnumResourceW
```

*Figure 8. Function snippet enumerates different network resource types.*

The local drive encryption module encrypts all locally attached drives by first using the GetLogicalDrives function to enumerate all available drive letters from A to Z. For each letter, it checks the drive type with the GetDriveTypeW function, targeting drives identified as DRIVE\_REMOVABLE, DRIVE\_FIXED, or DRIVE\_REMOTE while excluding CD-ROM and network-only drives. After constructing the drive path (e.g., "X:\\"), it creates a dedicated encryption worker thread for each validated drive path.

The Hyper-V virtual machine encryption module targets virtual machine files by executing a series of embedded PowerShell commands. First, it disables PowerShell restrictions on the victim machine to ensure its commands run. It then discovers the virtual machine files by listing all VMs and extracting their corresponding hard disk file paths. To unlock these files for encryption, the module forcefully stops all running virtual machines. After these prerequisite steps, it creates encryption worker threads to encrypt the located virtual machine files. The PowerShell commands executed by the module:

```
powershell -c "Set-ExecutionPolicy bypass"
powershell -c "get-vm | format-list"
powershell -c "get-vm | Get-VMHardDiskDrive | ForEach-Object {$_.Path}"
powershell -c "get-vm | stop-vm -force -turnoff"
```

The ransomware excludes the executables (.exe) and dynamic-link library (.dll) files along with the folders "Program Files", "Program Files (X86)", and "ProgramData" from the encryption processes on the victim machine, allowing the victims to still access the system to communicate with the threat actor.

## Kraken Linux/ESXi encryptor

---

The Linux or ESXi version of the Kraken ransomware is 64-bit executable written in C++ and compiled using the tool crosstool-NG version 1.26.0.

In the initial phase of the execution, the Linux executable file version of Kraken ransomware processes the command-line parameters specified by the attacker.

### Platform discovery

---

The ransomware runs the platform detection module to discover the type of victim machine by executing the commands mentioned below and adapting the behavior based on the detected platform.

System type	Command
ESXi	esxcli system version get
Nutanix	uname -a with “nutanix”
Ubuntu Linux	uname -a with “ubuntu”
Synology NAS devices	cat /etc.defaults/VERSION with “dsm”

While targeting the ESXi environments, the ransomware lists any running virtual machines and forcefully attempts to kill them by executing the following commands embedded in the ransomware binary:

```
esxcli vm process list
esxcli vm process kill --type=force --world-id=
```

### Encryption types

---

The ELF version of Kraken ransomware performs the multi-threaded encryption, supporting both “solid - Full encryption” and “setp - partial encryption”. It also employs the encryption performance benchmarking module that an attacker can leverage during the attack to calculate the encryption speed and decide if they want to perform full or partial encryption. The performance benchmarking algorithm is like the Windows version of Kraken ransomware described in the previous section.

It performs the recursive directory traversal and encrypts the file based on the type of encryption mode specified in the command line parameter by the attacker and appends the .zpsc file extension to the encrypted files.

## Anti-analysis and detection evasion

The ELF version of Kraken ransomware employs control flow obfuscation with the complex loop structure to hinder the analysis and operates in daemon mode by forking into background process through `fork_as_daemon()` function and continues to run, performing the encryption in background. It also ignores the signal handlers SIGCHLD (child process termination) and SIGHUP (Terminal hangup).

The ransomware employs a multi-stage self-deletion and cleanup process to erase traces of its execution, leaving a minimal forensic artefact, after completing the encryption operation. Kraken creates a bash script “\_bye\_bye\_.sh” in the same directory as the ransomware binary. It then builds the script with the commands to delete the log files, shell history, ransomware binary, and the script itself.

```
rm -f "/var/logs/*"
rm -f "/.ash_history"
rm -f "ransomware binary path"
rm -f "delete the script _bye_bye_.sh"
```

It executes the script using popen function `popen("sh \"<deletion_script_path>\"", "r")` which runs in a separate shell process, and the parent process can exit before the script finishes its execution which helps to delete itself before the completion of the execution.

## Coverage

Ways our customers can detect and block this threat are listed below.

Extended Detection and Response: Cisco XDR	Multi-Factor Authentication: Cisco Duo	Endpoint: Cisco Secure Endpoint
✓	N/A	✓
Email: Cisco Secure Email Threat Defense	Network security: Cisco Secure Firewall	Multi-Cloud Security: Cisco MultiCloud Defense
✓	✓	N/A
Secure Internet Gateway: Cisco Umbrella	Analytics: Cisco Secure Network Analytics	Security Service Edge (SSE): Cisco Secure Access
✓	N/A	✓

[Cisco Secure Endpoint](#) (formerly AMP for Endpoints) is ideally suited to prevent the execution of the malware detailed in this post. Try Secure Endpoint for free [here](#).

[Cisco Secure Email](#) (formerly Cisco Email Security) can block malicious emails sent by threat actors as part of their campaign. You can try Secure Email for free [here](#).

[Cisco Secure Firewall](#) (formerly Next-Generation Firewall and Firepower NGFW) appliances such as [Threat Defense Virtual](#), [Adaptive Security Appliance](#) and [Meraki MX](#) can detect malicious activity associated with this threat.

[Cisco Secure Network/Cloud Analytics](#) (Stealthwatch/Stealthwatch Cloud) analyzes network traffic automatically and alerts users of potentially unwanted activity on every connected device.

[Cisco Secure Malware Analytics](#) (Threat Grid) identifies malicious binaries and builds protection into all Cisco Secure products.

[Cisco Secure Access](#) is a modern cloud-delivered Security Service Edge (SSE) built on Zero Trust principles. Secure Access provides seamless transparent and secure access to the internet, cloud services or private application no matter where your users work. Please

contact your Cisco account representative or authorized partner if you are interested in a free trial of Cisco Secure Access.

[Umbrella](#), Cisco's secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs and URLs, whether users are on or off the corporate network.

[Cisco Secure Web Appliance](#) (formerly Web Security Appliance) automatically blocks potentially dangerous sites and tests suspicious sites before users access them.

Additional protections with context to your specific environment and threat data are available from the [Firewall Management Center](#).

[Cisco Duo](#) provides multi-factor authentication for users to ensure only those authorized are accessing your network.

Open-source Snort Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on [Snort.org](#).

Snort SIDs for the threats are: 65480 and 65479.

ClamAV detections are also available for this threat:

- Win.Ransomware.Kraken-10056931-0
- Unix.Ransomware.Kraken-10057031-0

## Indicators of compromise (IOCs)

---

The IOCs can also be found in our GitHub repository [here](#).

