The Rust Revolution: New Embargo Ransomware Steps In - Cyble

By cybleinc

Published: 2024-05-24 · Archived: 2026-04-05 22:37:41 UTC

Cyble analyzes the Rust-based Embargo ransomware, investigating its operations and possible variants.

Key Takeaways

- Cyble Research & Intelligence Labs (CRIL) identified a sample of Embargo ransomware, developed in Rust.
- The Threat Actors behind this ransomware are using double extortion tactics.
- We observed an instance where the ransomware group Initially demanded a $1 million ransom payment, threatening data leak and notifications to various parties upon non-payment.
- The leak site User Interfaces of Embargo and ALPHV ransomware resemble each other. Additionally, the leak site of ALPHV ransomware was taken down by law enforcement in March 2024.
- The log generation structure of both the ransomware looks similar.
- Embargo, to date, has disclosed details of four victims globally.
- This ransomware Utilizes ChaCha20 and Curve25519 for file encryption and appends “.564ba1” extension to encrypted files.

Overview

CRIL found a sample of Embargo ransomware, which is developed in Rust programming language. TAs behind this ransomware are using double extortion to target its victims. In double extortion, the TAs exfiltrate sensitive information from the victim’s systems before encrypting the data.

They then threaten to publicly release or sell this stolen data if the ransom is not paid. This adds additional pressure on the victim, as the potential data breach can lead to severe reputational damage, legal consequences, and loss of customer trust. The figure below shows the leak site of Embargo ransomware.

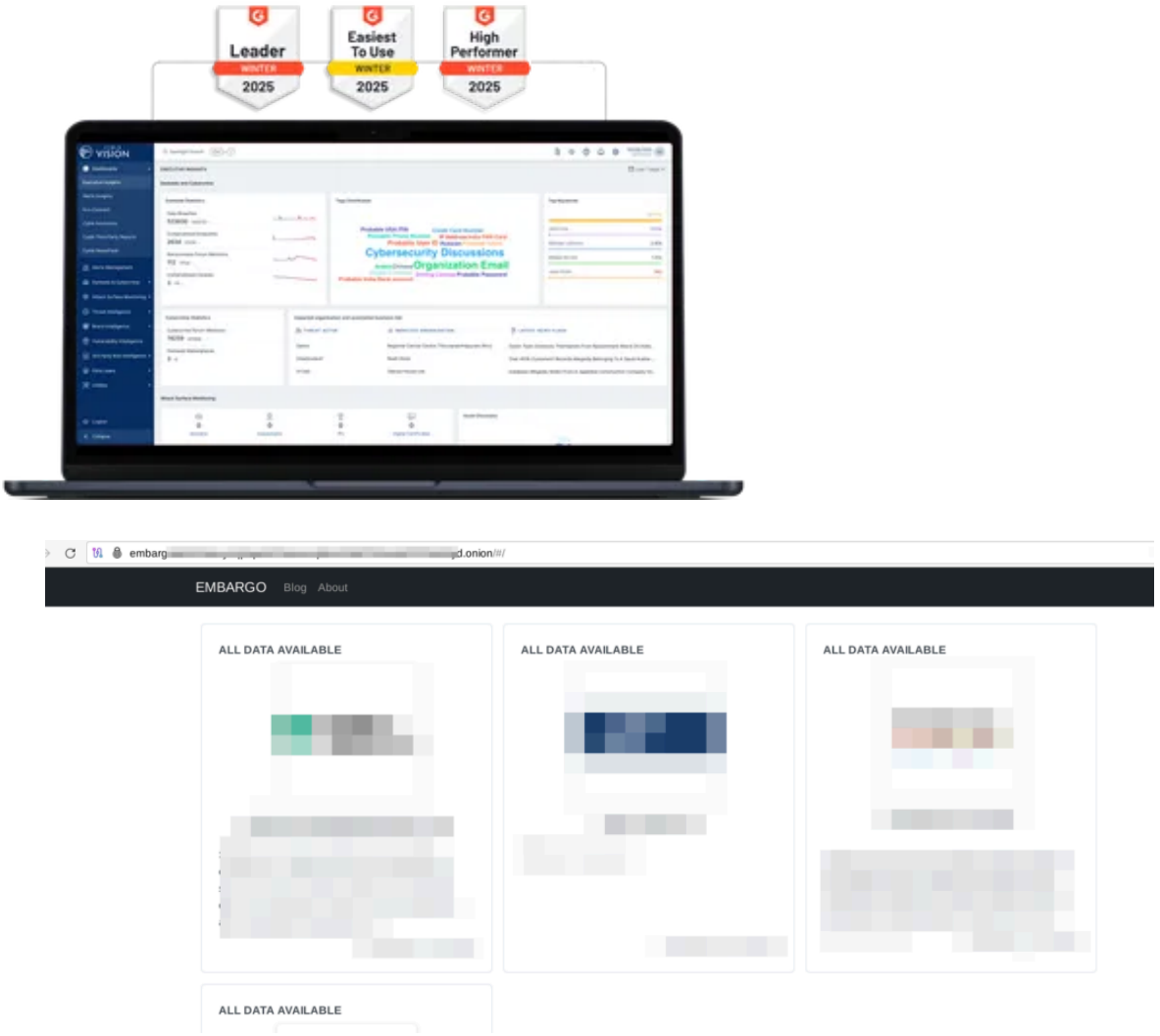

Figure 1 – Embargo Ransomware Leak Site

During our investigation, we discovered that this ransomware group initially demanded a $1 million ransom, as indicated on their chat site. The Threat Actors (TAs) also claim on this site that if the victim fails to pay the ransom, they will not only leak the data but also notify the victim's clients, employees, partners, investors, stakeholders, and government authorities about the attack. The figure below illustrates the ransom amount demanded by the group.

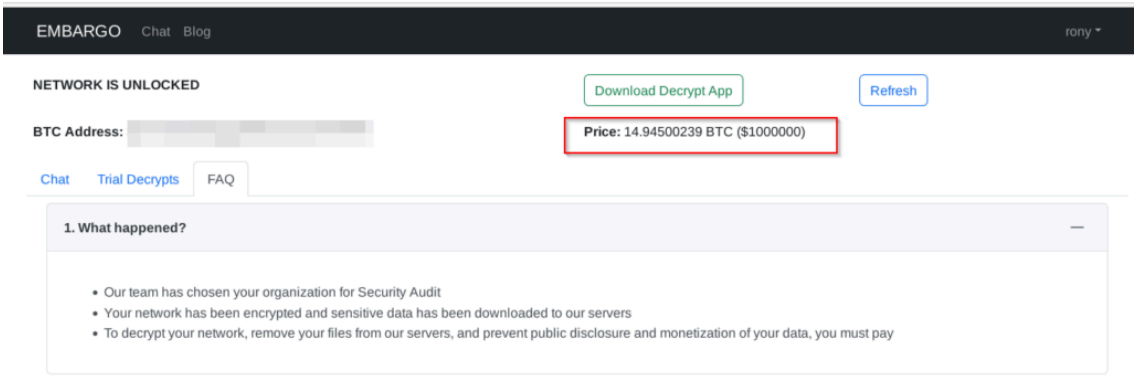

Figure 2 – Ransom Demand

We also observed similarities in the user interface of ALPHV (Blackcat) ransomware and the Embargo ransomware leak site. The leak site of ALPHV ransomware was taken down by law enforcement in March 2024. The figure below shows the comparison between the two leak sites.

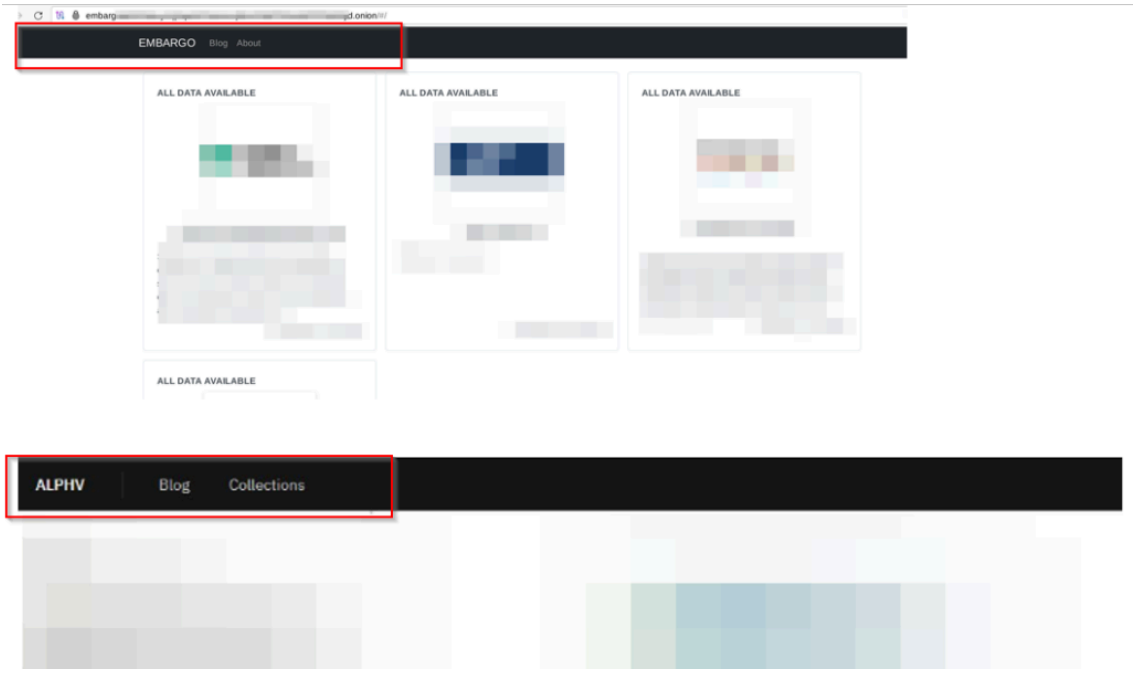

Figure 3 – Comparison of ALPHV and Embargo Ransomware Leak Site

Upon comparing the Rust binaries of both ransomware samples, we observed an overlap in the structure and syntax used for generating log files. Although the ALPHV ransomware binary has more capabilities, we suspect that Embargo might be a rewritten version of ALPHV. The Rust binary of ALPHV ransomware surfaced in 2022. The figure below shows both ransomware binaries executed in verbose mode.

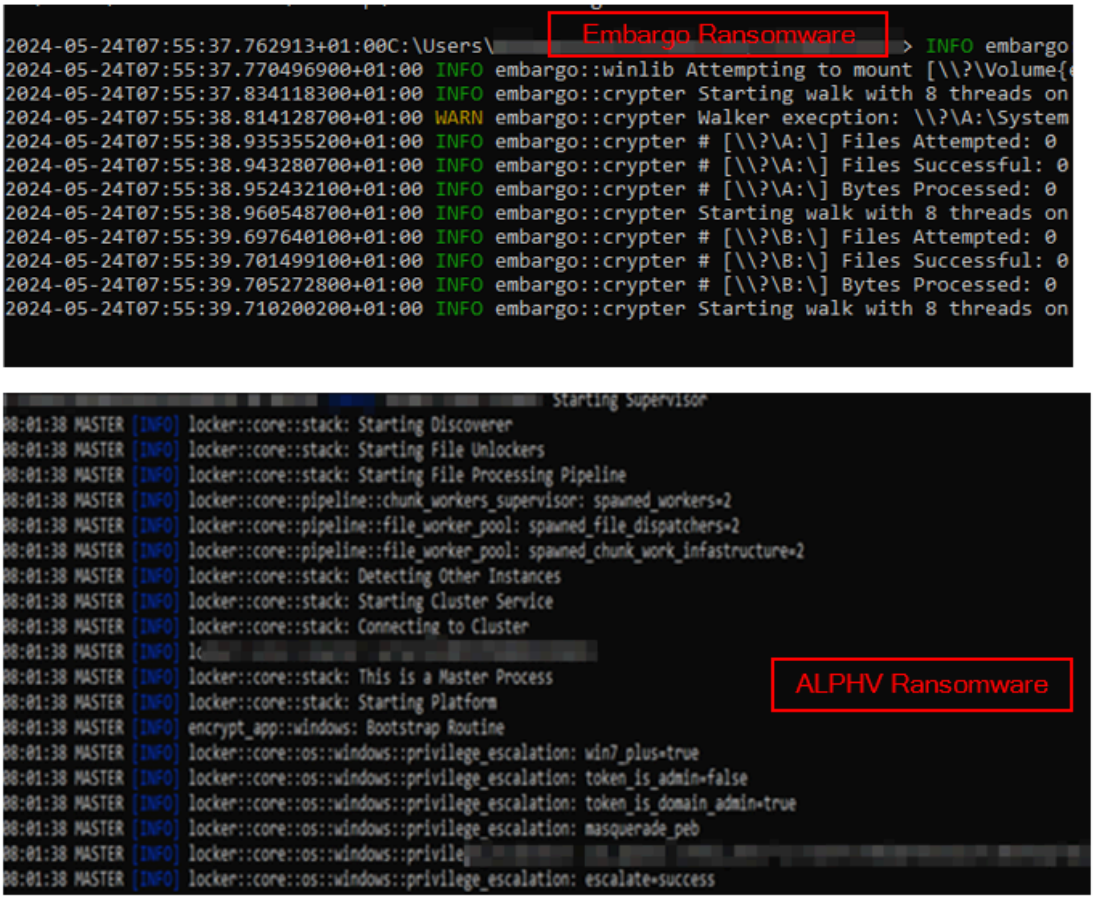

Figure 4 – Comparison of Logs

Embargo ransomware, to date, has mentioned four victims globally. The figure below shows the distribution of Embargo's victims.

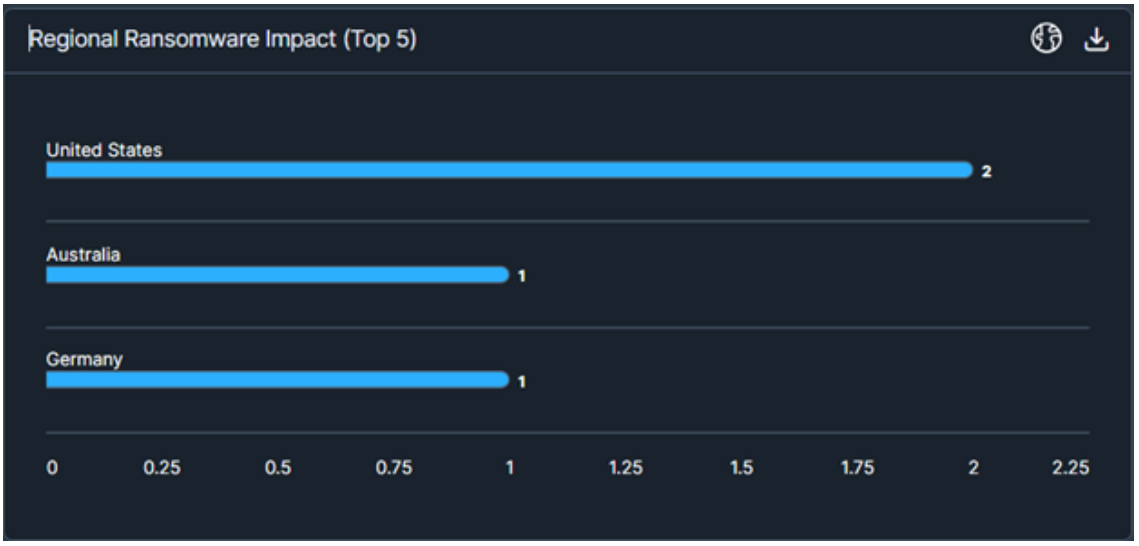

Figure 5 – Geographic distribution of victims

Technical Analysis

The behavior of this ransomware can be controlled using command-line arguments. Upon execution, it calls *GetCommandLineW()* to check for the arguments as shown in the figure below.

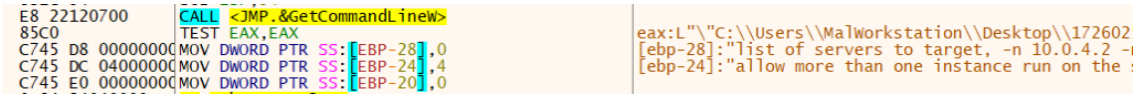

Figure 6 – Checking CommandLine Arguments

It then checks for the following arguments

Argument	Alternate Argument	Description
-t	–threads <THREADS>	number of threads, leave default to automatically assign
-p	–path <PATH>	path to directory
–no-delete		disable self-delete
–partial		enable searching for partially encrypted files and finish encrypting (if a previous run failed)
-l	–log	enable output to log
-v	–verbose	enable verbose output
-f	–follow-sym	enable follow symlinks
-m	–multi-run	allow more than one instance run on the same host
–no-net		do not search for network resources to encrypt
-n	–net-path <NETWORK_PATHS>	list of servers to target
-h	–help	Print help

The presence of hardcoded commands on how to use this ransomware binary reveals insights into the TA's mentality and tactics. These examples suggest the specific types of folders and directories the TAs typically target during their attacks. The TA has mentioned the following directories:

- R:\backups\
- \\files01\finance
- \\10.0.3.2\D$\Accounting

```
Options:
  -t, --threads <THREADS>          number of thread, leave default to automatic assign [default: 0]
  -p, --path <PATH>                path to directory, -p R:\backups\ -p \\files01\finance -p \\10.0.3.2\D$\Accounting
      --no-delete                  disable self-delete
      --partial                    enable searching for partially encrypted files and finish encrypting (if a previous run failed)
  -l, --log                        enable output to log
  -v, --verbose                    enable verbose output
  -f, --follow-sym                 enable follow symlinks
  -m, --multi-run                  allow more than one instance run on the same host
      --no-net                     do not search for network resources to encrypt
  -n, --net-path <NETWORK_PATHS>   list of servers to target, -n 10.0.4.2 -n FILES02.contoso.local -n FILES03
  -h, --help                       Print help
```

Figure 7 – Command line Arguments

After getting the command line arguments, the ransomware binary creates a mutex named "*LoadUpOnGunsBringYourFriends*" using the *CreateMutexW()* function. Unlike other ransomware variants, this one uses a hardcoded mutex name instead of generating a string at runtime. The figure below illustrates the mutex created by the ransomware binary.

```
68 E8E5C500     PUSH embargo.C5E5E8                  C5E5E8:L"LoadUpOnGunsBringYourFriends"
6A 01           PUSH 1
6A 00           PUSH 0
E8 B88D4400     CALL <JMP.&CreateMutexW>
```

Figure 8 – CreateMutex()

Following this, the ransomware proceeds to clear the recycle bin by invoking the *SHEmptyRecycleBinW()* function. Typically, this action is taken to hinder the victim's ability to restore any deleted files after encryption. The figure below shows the ransomware clearing the recycle bin.

```
00F455A7   0F84 A00700   JE embargo.F45D4D
00F455AD   E8 1ECB2600   CALL embargo.11B20D0
00F455B2   6A 07         PUSH 7
00F455B4   6A 00         PUSH 0
00F455B6   50            PUSH EAX
00F455B7   E8 FCE14600   CALL <JMP.&SHEmptyRecycleBinW>
```

Figure 9 – Clears RecycleBin

Next, it executes the following command to disable the Windows recovery:

- *C:\\Windows\\System32\\cmd.exe /q /c bcdedit /set {default} recoveryenabled no*

The ransomware then captures a snapshot of active running processes using *CreateToolhelp32Snapshot()* and iterates over them with *Process32First()* and *Process32Next()*. It checks if any of the processes listed below are running and terminates them if a match is found.

agntsvc.exe	sql.exe	QBIDPService.exe
dbeng50.exe	steam.exe	QBDBMgrN.exe
dbsnmp.exe	synctime.exe	QBCFMonitorService.exe
encsvc.exe	tbirdconfig.exe	SAP.exe
excel.exe	thebat.exe	TeamViewer_Service.exe

firefox.exe	thunderbird.exe	TeamViewer.exe
infopath.exe	visio.exe	tv_w32.exe
isqlplussvc.exe	winword.exe	tv_x64.exe
msaccess.exe	wordpad.exe	cvd.exe
mspub.exe	xfssvccon.exe	cvfwd.exe
mydesktopqos.exe	*sql*.exe	cvods.exe
mydesktopservice.exe	bedbh.exe	saphostexec.exe
notepad.exe	vxmon.exe	saposcol.exe
ocautoupds.exep	benetns.exe	sapstartsrv.exe
ocomm.exe	bengien.exe	avscc.exe
ocssd.exe	pvlsvr.exe	DellSystemDetect.exes
onenote.exe	beserver.exe	EnterpriseClient.exe
oracle.exe	raw_agent_svc.exe	veeam*.exe
outlook.exe	vsnapvss.exe	VeeamNFSSvc.exe
powerpoint.exe	CagService.exe	VeeamTransportSvc.exe
sqbcoreservice.exe	vsnapvss.exe	VeeamDeploymentSvc.exe

The figure below illustrates the ransomware iterating through the active processes.

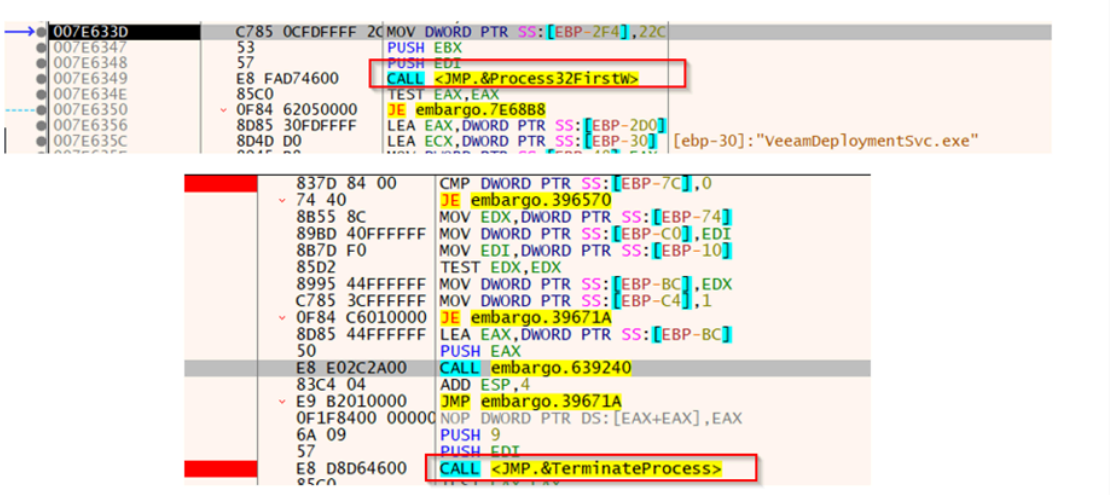

Figure 10 – Terminating Process

After this, the ransomware retrieves the active services running on the victim's system. It first calls the *OpenSCManagerW()* function to obtain a handle to the service control manager database. Then, it calls *EnumServicesStatusExW()* to enumerate the services in the service control manager database. This call retrieves the status of services, including their names and current states. The ransomware then checks if any of the running services match the following services:

GxCIMgr$	MSExchange\$.*$
MVArmor$	AcronisAgent$
VSNAPVSS$	VeeamTransportSvc$
VeeamNFSSvc$	BackupExecVSSProvider$
QBCFMonitorService$	BackupExecManagementService
GxVssHWProv$	SAPD\$$xecManagementSe
QBDBMgrN$	QBIDPService$
BackupExecRPCService$	AcrSch2Svc$
VeeamTransportSvc$	SAPService$
MVarmor64$	SAPHostControl$
SAPHostControl$	BackupExecJobEngine$ce
SAPHostExec$	BackupExecRPCService$
QBCFMonitorService$	GxClMgrS$

If a match is found, it closes the service by calling the *CloseServiceHandle()* function. The figure below illustrates the ransomware iterating through services.

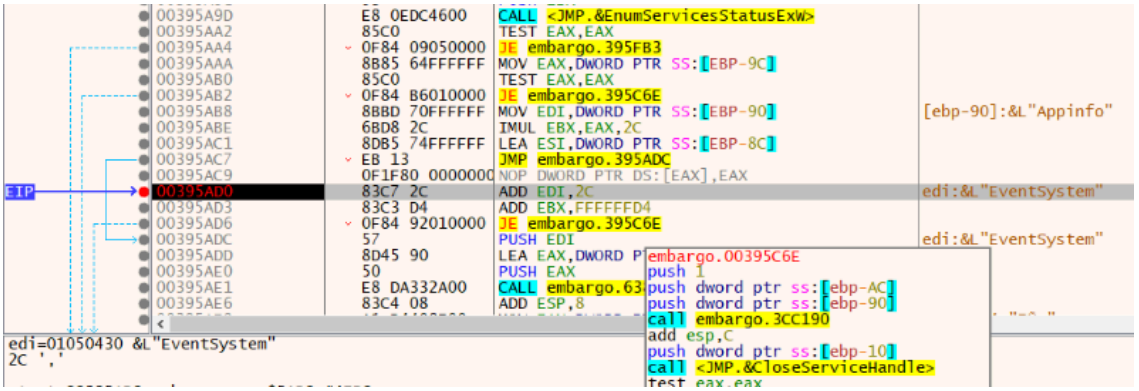

Figure 11 – Terminating Services

The ransomware now starts iterating through device volumes using the *FindFirstVolumeW()* and *FindNextVolumeW()* functions. It then calls the *GetVolumePathNamesForVolumeNameW()* function to retrieve a list

of drive letters and mounted folder paths for each specified volume.

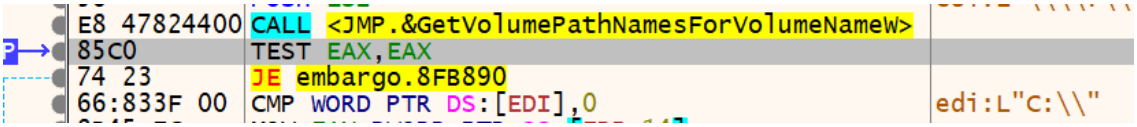

Figure 12 – Fetching Drives

After this, it uses the *WNetEnumResourceW()* function to enumerate the network resources. It then starts enumerating the files in the drives for encryption using *GetDriveTypeW()* and, *FindFirstFileW()* and *FindNextFileW()* functions, as shown in the figure below.

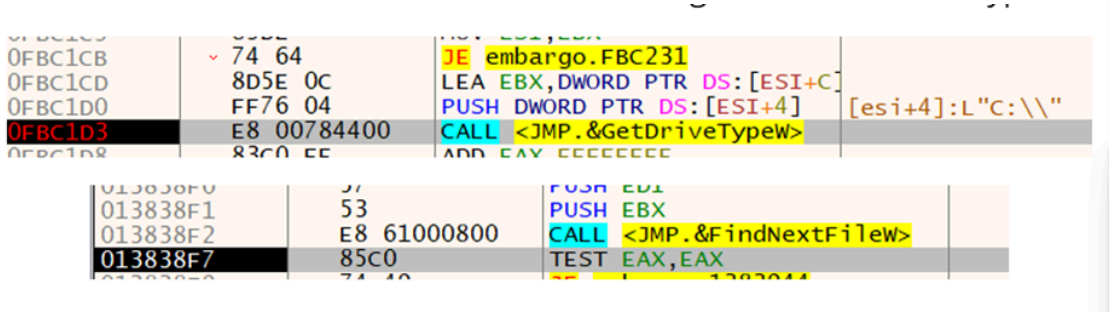

Figure 13 – Enumerating Drives

The ransomware does not encrypt files present in the following directories on an infected system. The ransomware binary contains regular expressions for these directory names.

ProgramData/Microsoft/DeviceSync[^/]*$	ProgramData/USOShared[^/]*$
ProgramData/Microsoft/Diagnosis[^/]*$	Program Files/WindowsApps[^/]*$
ProgramData/ssh/[^/]*$	ProgramData/Microsoft/UEV[^/]*$
Program Files/Windows Portable Devices[^/]*$	ProgramData/Microsoft/Device Stage[^/]*$
Program Files/Uninstall Information[^/]*$	ProgramData/Packages/USOShared[^/]*$
ProgramData/regid\.[^/]*\.com\.microsoft$vic	ProgramData/Microsoft/Event Viewer[^/]*$
ProgramData/USOPrivate[^/]*$	ProgramData/Microsoft/Provisioning[^/]*$
Program Files/Windows Defender[^/]*$	ProgramData/Microsoft/IdentityCRL$
Program Files/Windows Media Player[^/]*$	ProgramData/Microsoft/NetFramework[^/]*$
Program Files/Windows Security[^/]*$	ProgramData/Microsoft/Spectrum[^/]*$
Program Files/Windows Photo Viewer[^/]*$	ProgramData/Microsoft/Windows Defender[^/]*$

Program Files/ModifiableWindowsApps[^/]*$	ProgramData/Microsoft/MapData[^/]*$
Program Files/Internet Explorer[^/]*$	ProgramData/Microsoft/WDF[^/]*$
Program Files/Windows NT[^/]*$	ProgramData/Microsoft/Storage Health[^/]*$
Program Files/Windows Sidebar[^/]*$	ProgramData/Microsoft/Windows[^/]*$
Program Files/WindowsPowerShell[^/]*$	ProgramData/Microsoft/Search[^/]*$
Program Files \(x86\)/Microsoft\.NET[^/]*$	ProgramData/Microsoft/Vault[^/]*$
Program Files \(x86\)/Windows Defender[^/]*$	ProgramData/Microsoft/SmsRouter[^/]*$
Program Files/Invisible Things Lab/[^/]*$	ProgramData/Microsoft/Speech_OneCore[^/]*$
Program Files \ (x86\)/Microsoft/Temp[^/]*$	ProgramData/Microsoft/Windows NT[^/]*$
Program Files \(x86\)/Windows NT[^/]*$	ProgramData/Microsoft/MF[^/]*$
Program Files \(x86\)/Windows Security[^/]*$s	ProgramData/Microsoft/Network[^/]*$
Program Files \(x86\)/Windows Mail[^/]*$	ProgramData/Microsoft/WinMSIPC[^/]*$
Program Files \(x86\)/Windows Sidebar[^/]*$am	ProgramData/Microsoft/WPD[^/]*$
Program Files \(x86\)/Common Files[^/]*$	ProgramData/Microsoft/EdgeUpdate[^/]*$
Program Files/Common Files/System[^/]*$	ProgramData/Packages/USOPrivate[^/]*$
Program Files/Windows Mail[^/]*$o	ProgramData/Microsoft/DRM[^/]*$
ProgramData/ntuser\.pol$X	ProgramData/USOShared[^/]*$

Also, this ransomware does not encrypt files with the following extensions. The list includes the ".564ba1" extension, which is appended to files after encryption. This ensures that the ransomware will not encrypt the same file twice.

.cpl	.sys	.drv
d3d9caps.dat	*/NTUSER.DAT	.lnk
thumbs.db	.msi	*.search-ms
.ico	.dll	desktop.ini

.bat	.lock	.deskthemepack
iconcache.db	.msc	.theme
ntldr	.themeckpack	autorun.inf
d3d9caps.dat	.lock	boot.ini
.spl	.exe	.msstyles
.cab	.msu	.themepack
.564ba1		

This ransomware uses ChaCha20 and Curve25519 to encrypt files, as shown in the figure below. ChaCha20 and Curve25519 are often used together in file encryption to provide secure key exchange and encryption. Curve25519 establishes a shared secret key, which is then used by ChaCha20 to encrypt and decrypt the file contents.

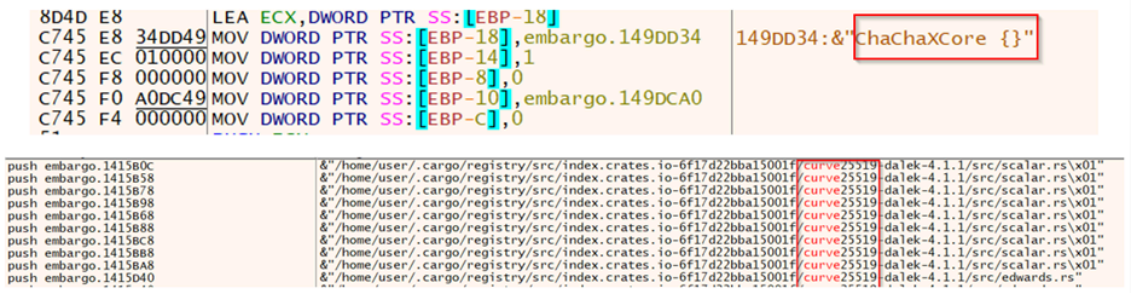

Figure 14 – Cryptographic Algorithms

Next, the ransomware drops a ransom note named “HOW_TO_RECOVER_FILES.txt” in every directory it iterates through. This note appears to be created specific to a victim, as date and time values are hardcoded rather than dynamically loading the current date and time.

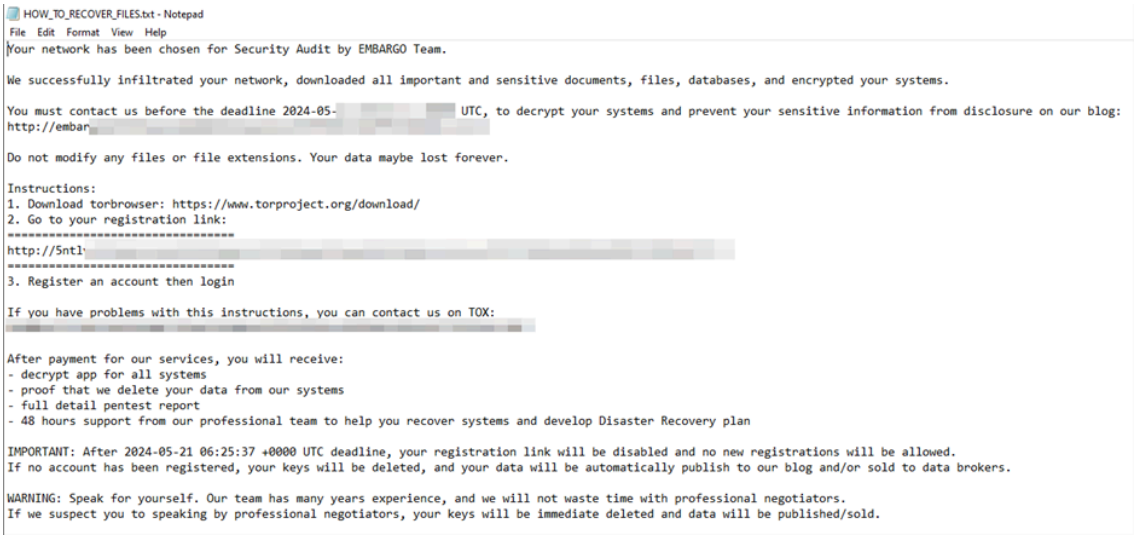

Figure 15 – Ransom Note

All the encrypted files consist of ".564ba1" as a file extension, as shown in the figure below.

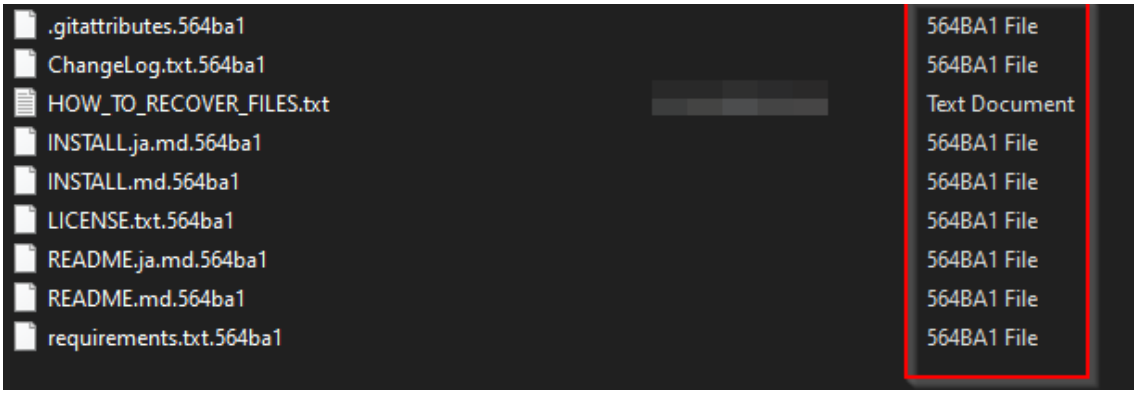

Figure 16 – Encrypted Files

LINUX and ESXI Variants

Additional files obtained from their onion site are suspected to be Linux and ESXi variants of the Embargo ransomware, but their true origin is uncertain. We suspect these to be test files as they lack configuration data and are not able to encrypt the files. These alleged variants of Embargo ransomware are 64-bit executables. Although they use the same encryption algorithm across all variants, the Linux binary offers fewer options than the Windows binary. By default, these variants utilize 4 threads, whereas the Windows variant employs 8 threads for execution. The figure below depicts the command-line arguments available in Linux variants.

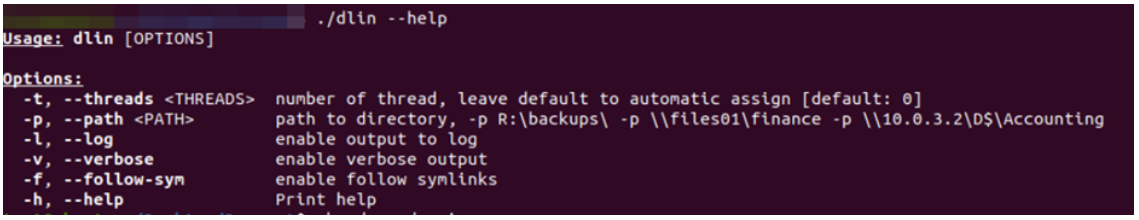

Figure 17 – Embargo Linux executable

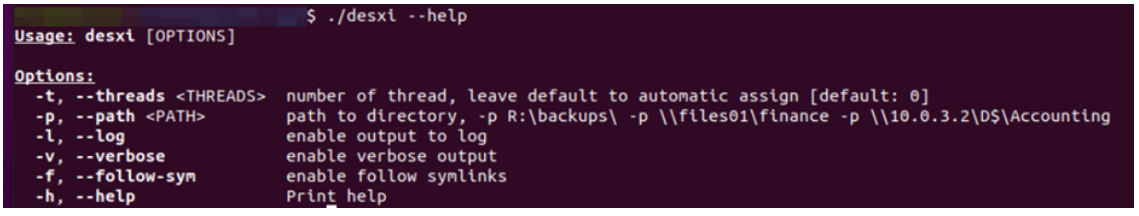

Figure 18 – Embargo ESXi executable

Conclusion

Embargo ransomware exemplifies the growing trend of using programming languages like Rust to create sophisticated, cross-platform ransomware. The choice of Rust provides the attackers with advantages such as cross-platform compatibility, speed, and memory safety, making the ransomware more robust and difficult to analyze or reverse-engineer. The double extortion technique used by ransomware not only pressurizes victims to pay quickly to avoid data breaches but also exposes them to potential legal and reputational damage. The TA's threat to notify

clients, employees, partners, investors, stakeholders, and government authorities further amplifies the urgency and severity of the situation.

Our Recommendations

We have listed some essential cybersecurity best practices that create the first line of control against attackers. We recommend that our readers follow the best practices given below:

Safety Measures to Prevent Ransomware Attacks

- Do not open untrusted links and email attachments without first verifying their authenticity.
- Conduct regular backup practices and keep those backups offline or in a separate network.
- Turn on the automatic software update feature on your computer, mobile, and other connected devices wherever possible and pragmatic.
- Use a reputed anti-virus and Internet security software package on your connected devices, including PC, laptop, and mobile.

MITRE ATT&CK® Techniques

Tactic	Technique	Procedure
Execution	T1204.002 (User Execution)	Malicious file.
Defense Evasion	T1070.004 (Indicator Removal: File Deletion)	Ransomware deletes itself after execution.
Defense Evasion	T1140 (Deobfuscate/Decode Files or Information)	Contains encrypted strings.
Discovery	T1083 (File and Directory Discovery)	Ransomware enumerates folders for file encryption and file deletion.
Discovery	T1135 (Network Share Discovery)	Target Network Shares
Impact	T1486 (Data Encrypted for Impact)	Ransomware encrypts the data for extortion.
Impact	T1490 (Inhibit System Recovery)	Disable automatic Windows recovery

Indicators of Compromise (IOCs)

Indicators	Indicator Type	Description
98cc01dcd4c36c47fc13e4853777ca170c734613564a5a764e4d2541a6924d39	SHA256	Embargo Ransomware (Windows)

7bfb789f5825f17a01cccd2fbd62635ce20f6ed7e488fded20549a806371aeb6	SHA256	Embargo Ransomware (ESXi)
e6b6503217b0cf50e262a6a843624068f8f6a96441d241695893e6cab3c60a2c	SHA256	Embargo Ranosmware (Linux)

Yara Rule

```
 rule Embargo{

meta:

     author = "Cyble Research and Intelligence Labs"

     description = "Detects Embargo Ransomware"

     date = "2024-05-24"

     os = "Windows"

strings:

     $a1  = "LoadUpOnGunsBringYourFriends" fullword ascii wide

     $a2  = "embargo" nocase ascii wide

     $a3  = "files01" nocase ascii wide

condition:

     all of them

}
```

Source: https://cyble.com/blog/the-rust-revolution-new-embargo-ransomware-steps-in/