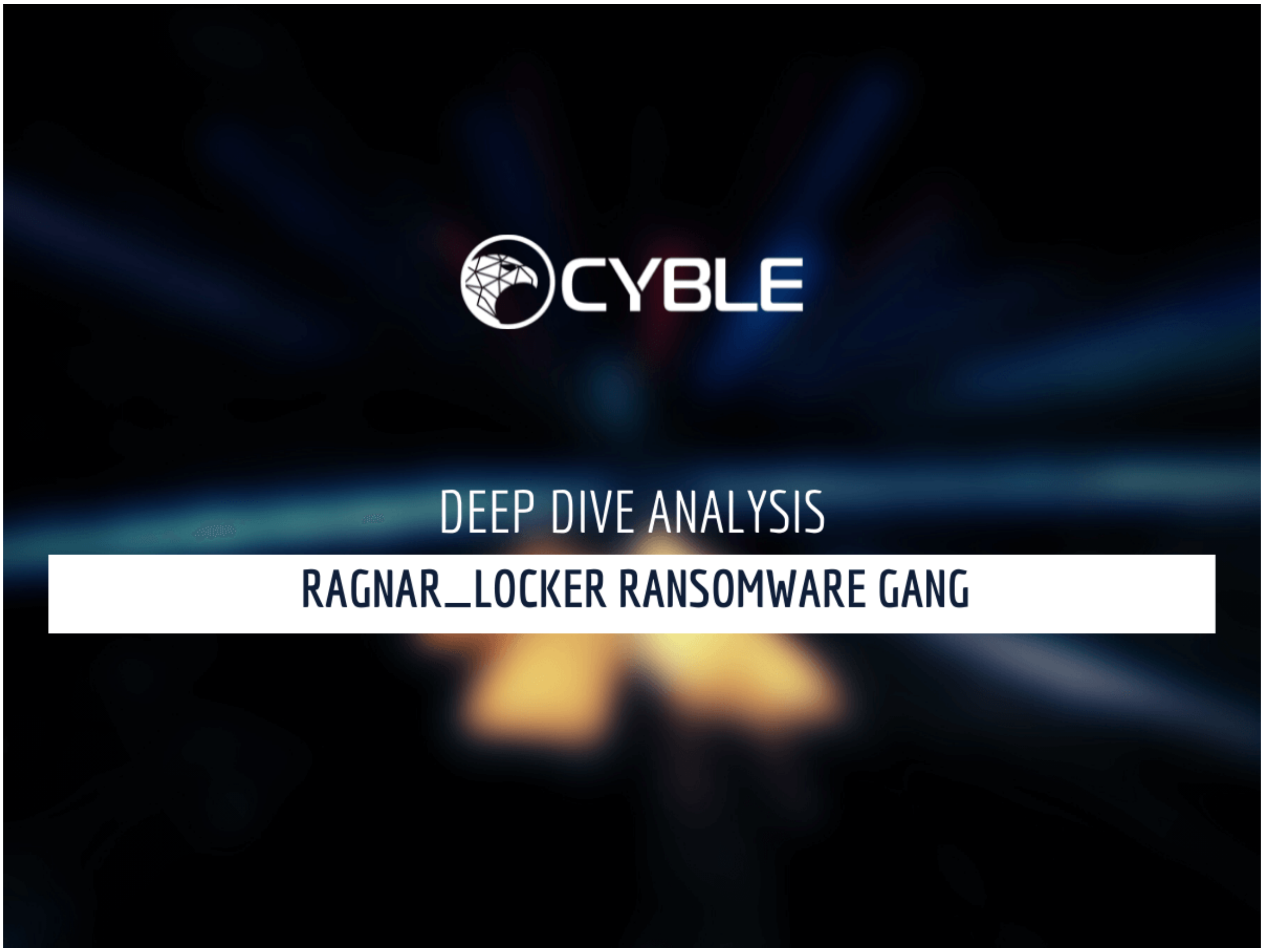
Deep dive into Ragnar_locker Ransomware Gang

blog.cyble.com/2022/01/20/deep-dive-into-ragnar-locker-ransomware-gang/

January 20, 2022

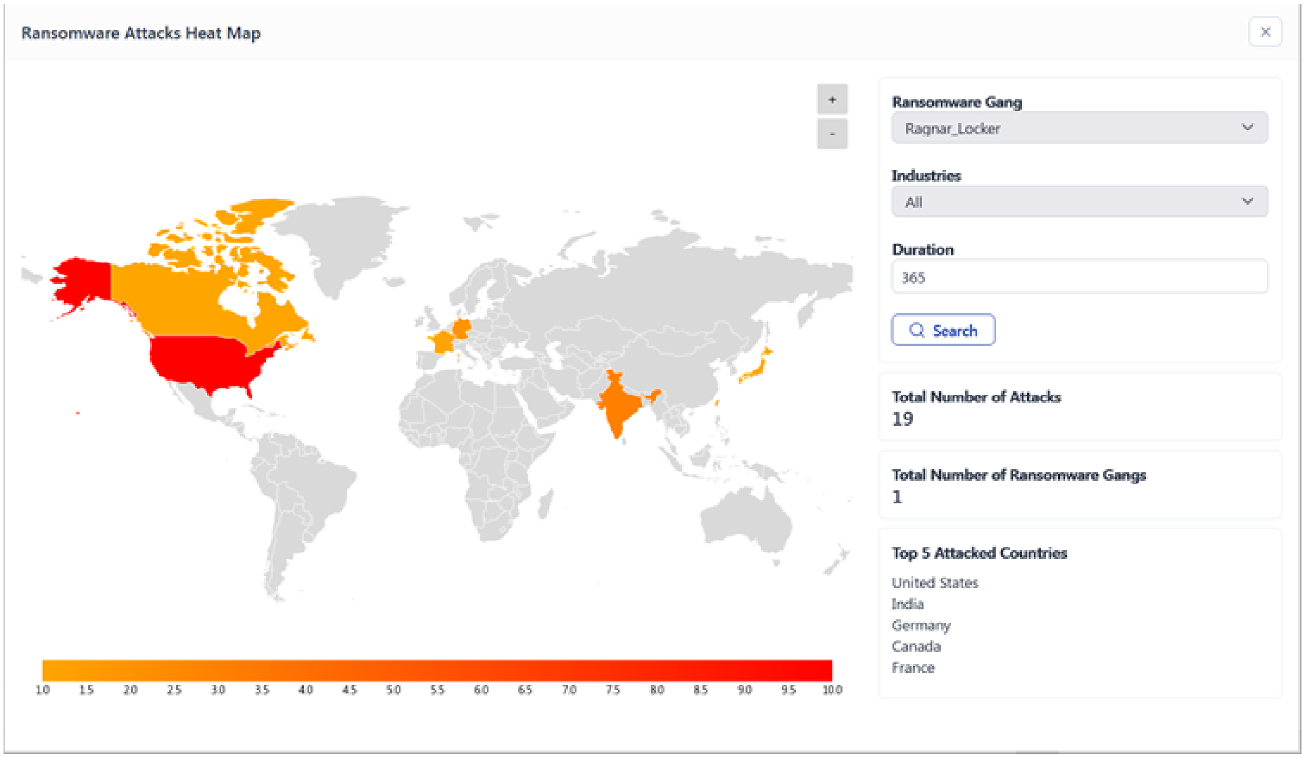


Organizations worldwide face a multi-pronged threat from Ransomware groups at a greater frequency than recorded before. As the organizations' primary danger remains losing access to their systems and data, the threat of Ransomware groups leaking the data if their ransom requests are not met or the victim reaches out to law enforcement authorities has been raising more concern.

Cyble Research Labs has analyzed and published information about the most prominent and active ransomware groups in the past and provided recommendations to prevent such incidents. This blog is a deep dive into one of the most active Ransomware groups, Ragnar_Locker, how they operate, their capabilities, and how to secure yourself/your organization from them.

Ragnar_locker ransomware was first observed in late 2019, targeting multiple high-profile targets on Windows platforms. Ragnar_locker also uses the double extortion technique for financial gain like most notorious ransomware gangs.

This group targets several countries worldwide, as shown in the figure below.


Ransomware Attacks Heat Map

Ransomware Gang: Ragnar_Locker

Industries: All

Duration: 365

Total Number of Attacks: 19

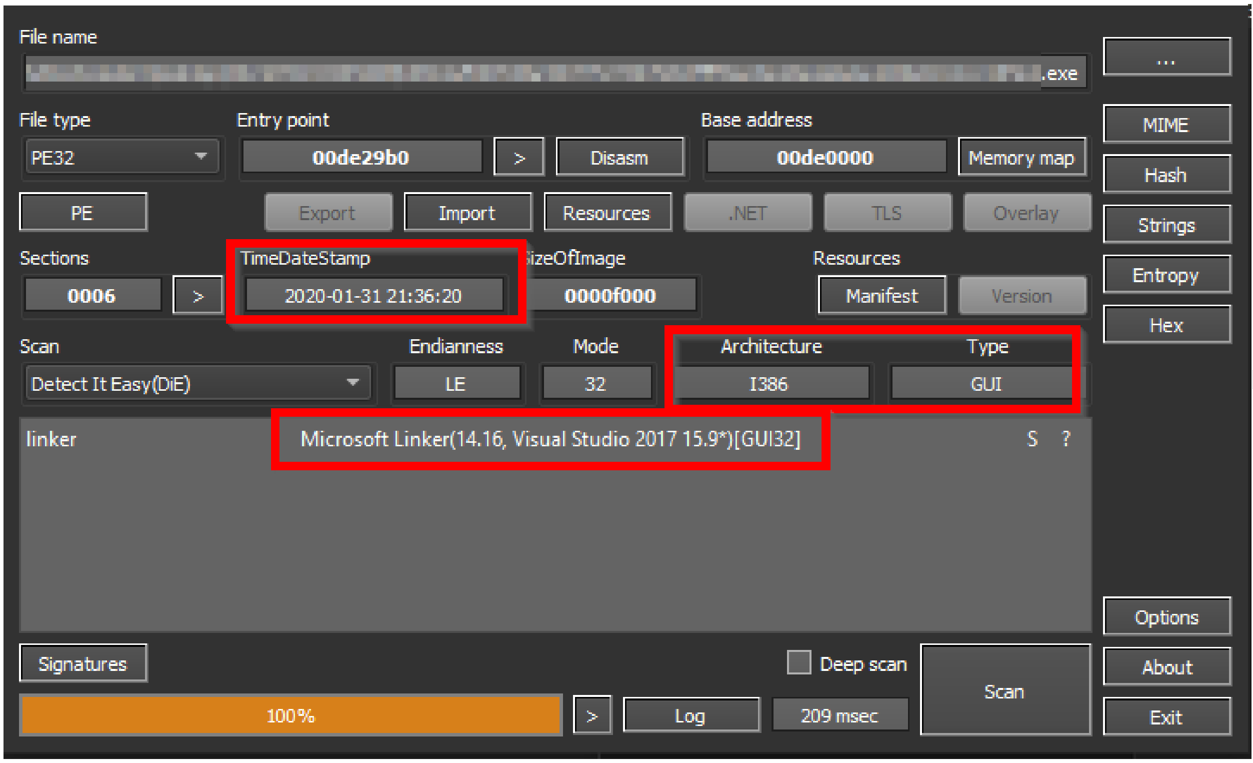
Total Number of Ransomware Gangs: 1

Top 5 Attacked Countries:
United States
India
Germany
Canada
France

Figure 1 Ragnar_locker Ransomware Victim Details

Technical Analysis

Based on static analysis, we found that the malicious file is a 32-bit Graphical User Interface (GUI) based binary, as shown in Figure 2.


File name: [redacted].exe

File type: PE32

Entry point: 00de29b0

Base address: 00de0000

Sections: 0006

TimeDateStamp: 2020-01-31 21:36:20

SizeOfImage: 0000f000

Scan: Detect It Easy(DiE)

Endianness: LE

Mode: 32

Architecture: I386

Type: GUI

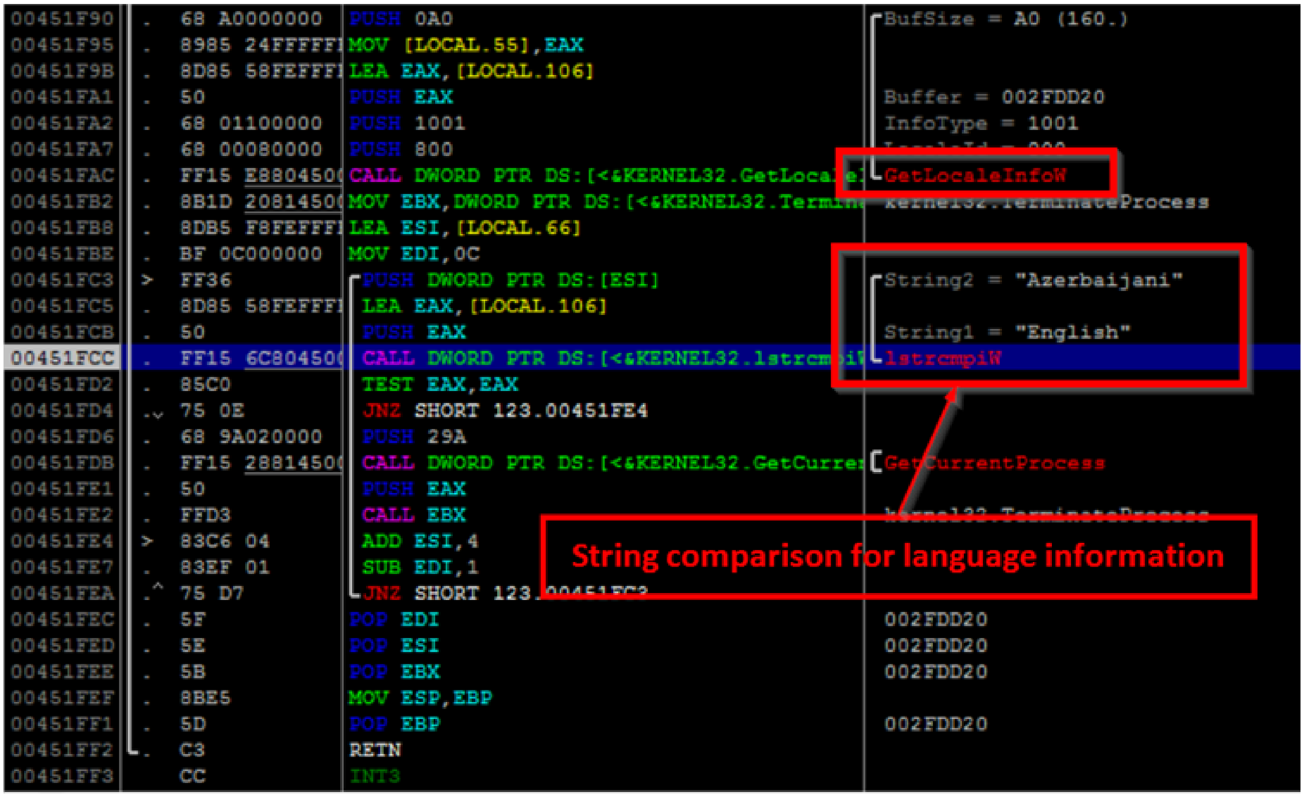
linker: Microsoft Linker(14.16, Visual Studio 2017 15.9*)[GUI32]

100%

209 msec

Figure 2 Static File Details of Ragnar_locker Ransomware

After execution, Ragnar Ransomware initially searches for system details using *GetLocalInfoW() API,* which extracts the system's default language. After identifying the system language, it compares this with a hardcoded list of languages present in the Ransomware binary, as shown in the figure below.



```
00451F90  .  68 A0000000   PUSH 0A0                                  ┌BufSize = A0 (160.)
00451F95  .  8985 24FFFFFF MOV [LOCAL.55],EAX                        │
00451F9B  .  8D85 58FEFFFF LEA EAX,[LOCAL.106]                       │
00451FA1  .  50            PUSH EAX                                  │Buffer = 002FDD20
00451FA2  .  68 01100000   PUSH 1001                                 │InfoType = 1001
00451FA7  .  68 00080000   PUSH 800                                  │LocaleId = 800
00451FAC  .  FF15 E8804500 CALL DWORD PTR DS:[<&KERNEL32.GetLocaleI  └GetLocaleInfoW
00451FB2  .  8B1D 20814500 MOV EBX,DWORD PTR DS:[<&KERNEL32.Termina  kernel32.TerminateProcess
00451FB8  .  8DB5 F8FEFFFF LEA ESI,[LOCAL.66]
00451FBE  .  BF 0C000000   MOV EDI,0C
00451FC3  >  FF36          ┌PUSH DWORD PTR DS:[ESI]                  ┌String2 = "Azerbaijani"
00451FC5  .  8D85 58FEFFFF │LEA EAX,[LOCAL.106]                      │
00451FCB  .  50            │PUSH EAX                                 │String1 = "English"
00451FCC  .  FF15 6C804500 │CALL DWORD PTR DS:[<&KERNEL32.lstrcmpiW  └lstrcmpiW
00451FD2  .  85C0          │TEST EAX,EAX
00451FD4  .v 75 0E         │JNZ SHORT 123.00451FE4
00451FD6  .  68 9A020000   │PUSH 29A
00451FDB  .  FF15 28814500 │CALL DWORD PTR DS:[<&KERNEL32.GetCurren  [GetCurrentProcess
00451FE1  .  50            │PUSH EAX
00451FE2  .  FFD3          │CALL EBX                                 kernel32.TerminateProcess
00451FE4  >  83C6 04       │ADD ESI,4
00451FE7  .  83EF 01       │SUB EDI,1
00451FEA  .^ 75 D7         └JNZ SHORT 123.00451FC3
00451FEC  .  5F            POP EDI                                   002FDD20
00451FED  .  5E            POP ESI                                   002FDD20
00451FEE  .  5B            POP EBX                                   002FDD20
00451FEF  .  8BE5          MOV ESP,EBP
00451FF1  .  5D            POP EBP                                   002FDD20
00451FF2  └. C3            RETN
00451FF3     CC            INT3
```

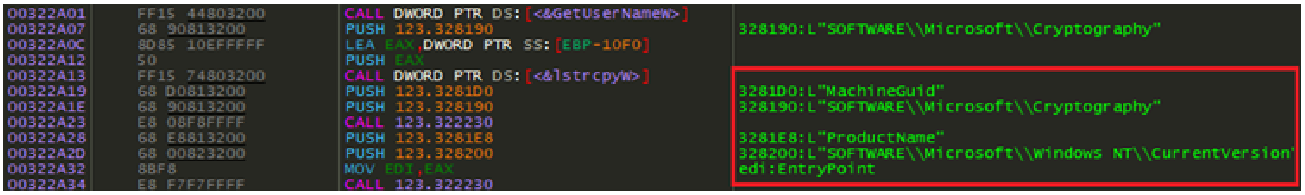
String comparison for language information

Figure 3 Ragnar_locker Ransomware Language Check

If the identified system language is present in the hardcoded list, the Ransomware terminates its execution using the*TerminateProcess()* API*.*

The languages hardcoded into the Ransomware are Belorussian, Azerbaijani, Ukrainian, and other languages commonly spoken in the former Soviet Union (USSR).

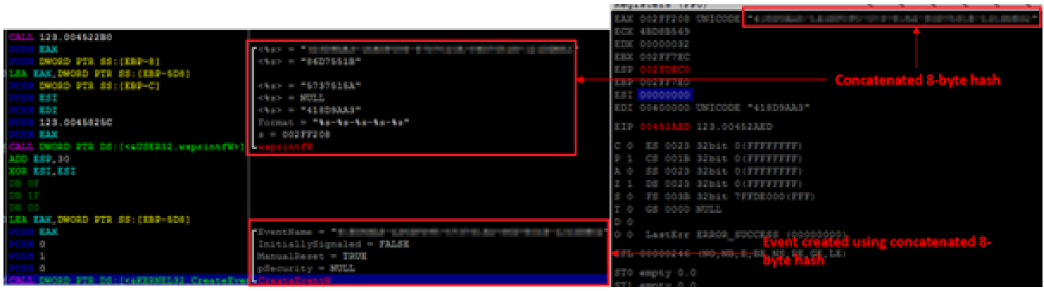
Ragar Ransomware then looks for other system information using APIs to retrieve the victim's system name, username, GUID, and product name.



```
00322A01    FF15 44803200    CALL DWORD PTR DS:[<&GetUserNameW>]
00322A07    68 90813200      PUSH 123.328190                          328190:L"SOFTWARE\\Microsoft\\Cryptography"
00322A0C    8D85 10EFFFFF    LEA EAX,DWORD PTR SS:[EBP-10F0]
00322A12    50               PUSH EAX
00322A13    FF15 74803200    CALL DWORD PTR DS:[<&lstrcpyW>]
00322A19    68 D0813200      PUSH 123.3281D0                          3281D0:L"MachineGuid"
00322A1E    68 90813200      PUSH 123.328190                          328190:L"SOFTWARE\\Microsoft\\Cryptography"
00322A23    E8 08F8FFFF      CALL 123.322230
00322A28    68 E8813200      PUSH 123.3281E8                          3281E8:L"ProductName"
00322A2D    68 00823200      PUSH 123.328200                          328200:L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion"
00322A32    8BF8             MOV EDI,EAX                              edi:EntryPoint
00322A34    E8 F7F7FFFF      CALL 123.322230
```

Figure 4 Ragnar_locker Ransomware Enumerating System Information

The Ransomware collects the above system information and calculates its size. This information and size are then fed to a custom logic to generate a unique hash to create an event in the system using *CreateEventW()* API, as shown in Figure 5.



```
CALL 123.004522B0
PUSH EAX
PUSH DWORD PTR SS:[EBP-8]
LEA EAX,DWORD PTR SS:[EBP-5D8]
PUSH DWORD PTR SS:[EBP-C]
PUSH ESI
PUSH EDI
PUSH 123.0045825C
PUSH EAX
CALL DWORD PTR DS:[<&USER32.wsprintfW>]
ADD ESP,30
XOR ESI,ESI
DB 0F
DB 1F
DB 00
LEA EAX,DWORD PTR SS:[EBP-5D8]
PUSH EAX
PUSH 0
PUSH 1
PUSH 0
CALL DWORD PTR DS:[<&KERNEL32.CreateEve
```

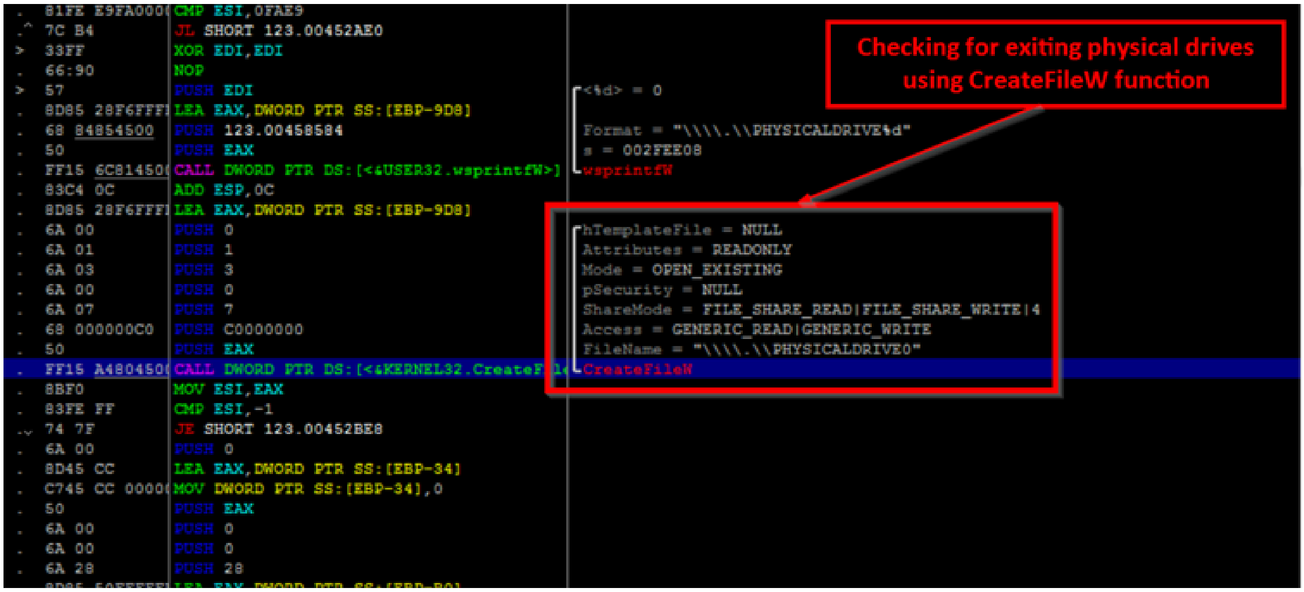
```
<%s> = "[redacted]"
<%s> = "86D7551B"

<%s> = "5737516A"
<%s> = NULL
<%s> = "418D9AA3"
Format = "%s-%s-%s-%s-%s"
s = 002FF208
wsprintfW

EventName = "[redacted]"
InitiallySignaled = FALSE
ManualReset = TRUE
pSecurity = NULL
CreateEventW
```

```
Registers (FPU)
EAX 002FF208 UNICODE "[redacted]"
ECX 48D8B569
EDX 00000032
EBX 002FF7EC
ESP 002FDEC0
EBP 002FF7E0
ESI 00000000
EDI 00400000 UNICODE "418D9AA3"

EIP 00452AED 123.00452AED

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDE000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 0.0
ST1 empty 0.0
```

Figure 5 Ragnar_locker Ransomware Creating Event

This malware then enumerates all the physical drives in the system. Ragnar Ransomware uses *CreateFileW()* API function to check which physical drives are accessible by the system. The malware then executes a loop that runs sixteen times to get all the accessible physical drives.

Figure 6 shows the enumeration of \\\\.\\PHYSICALDRIVE.



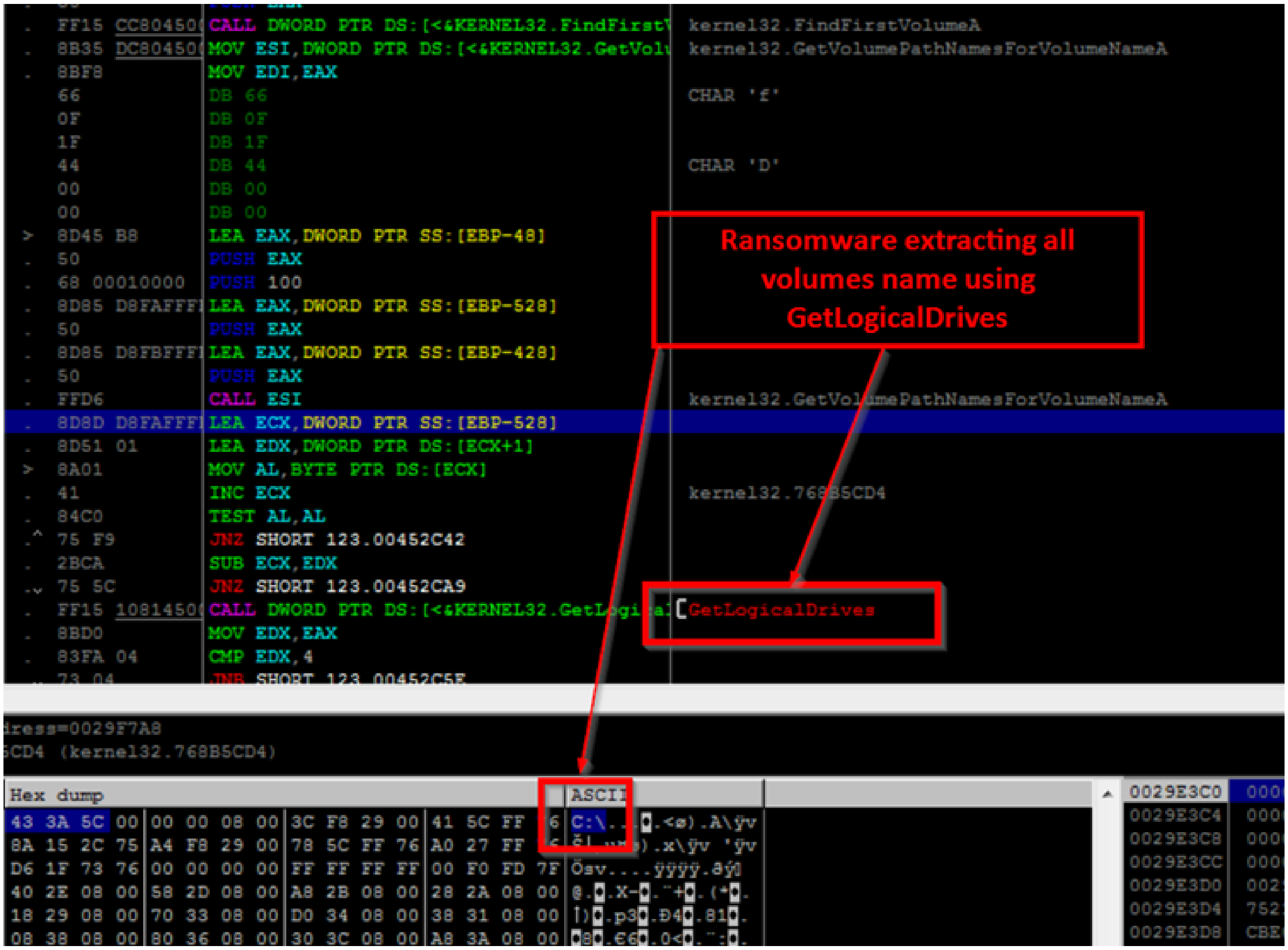
```
.    81FE E9FA000( CMP ESI,0FAE9
.^   7C B4         JL SHORT 123.00452AE0
>    33FF          XOR EDI,EDI
.    66:90         NOP
>    57            PUSH EDI
.    8D85 28F6FFF] LEA EAX,DWORD PTR SS:[EBP-9D8]
.    68 84854500   PUSH 123.00458584
.    50            PUSH EAX
.    FF15 6C81450( CALL DWORD PTR DS:[<&USER32.wsprintfW>]
.    83C4 0C       ADD ESP,0C
.    8D85 28F6FFF] LEA EAX,DWORD PTR SS:[EBP-9D8]
.    6A 00         PUSH 0
.    6A 01         PUSH 1
.    6A 03         PUSH 3
.    6A 00         PUSH 0
.    6A 07         PUSH 7
.    68 000000C0   PUSH C0000000
.    50            PUSH EAX
.    FF15 A480450( CALL DWORD PTR DS:[<&KERNEL32.CreateFile
.    8BF0          MOV ESI,EAX
.    83FE FF       CMP ESI,-1
.v   74 7F         JE SHORT 123.00452BE8
.    6A 00         PUSH 0
.    8D45 CC       LEA EAX,DWORD PTR SS:[EBP-34]
.    C745 CC 0000( MOV DWORD PTR SS:[EBP-34],0
.    50            PUSH EAX
.    6A 00         PUSH 0
.    6A 00         PUSH 0
.    6A 28         PUSH 28
```

```
<%d> = 0

Format = "\\\\.\\PHYSICALDRIVE%d"
s = 002FEE08
wsprintfW

hTemplateFile = NULL
Attributes = READONLY
Mode = OPEN_EXISTING
pSecurity = NULL
ShareMode = FILE_SHARE_READ|FILE_SHARE_WRITE|4
Access = GENERIC_READ|GENERIC_WRITE
FileName = "\\\\.\\PHYSICALDRIVE0"
CreateFileW
```

Checking for exiting physical drives using CreateFileW function

Figure 6 Ragnar_locker Ransomware Checking for Physical Drives

After checking the physical drives, the Ransomware extracts all the system volume names using *GetLogicalDrives()* API, as shown in the figure below.

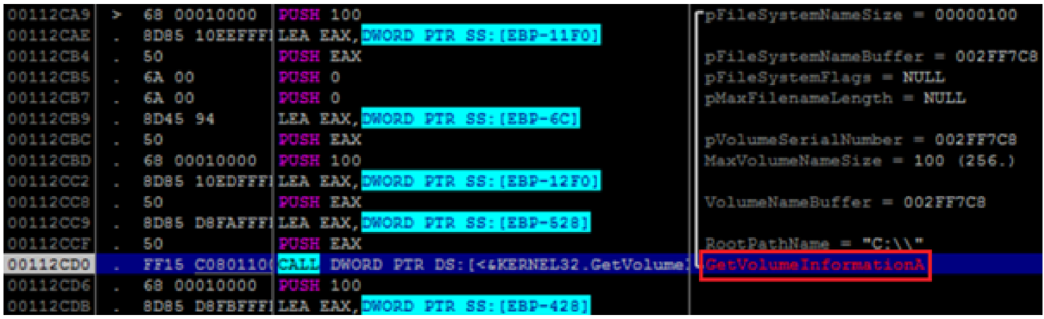


```
.    FF15 CC80450( CALL DWORD PTR DS:[<&KERNEL32.FindFirstV  kernel32.FindFirstVolumeA
.    8B35 DC80450( MOV ESI,DWORD PTR DS:[<&KERNEL32.GetVolu  kernel32.GetVolumePathNamesForVolumeNameA
.    8BF8          MOV EDI,EAX
     66            DB 66                                     CHAR 'f'
     0F            DB 0F
     1F            DB 1F
     44            DB 44                                     CHAR 'D'
     00            DB 00
     00            DB 00
>    8D45 B8       LEA EAX,DWORD PTR SS:[EBP-48]
.    50            PUSH EAX
.    68 00010000   PUSH 100
.    8D85 D8FAFFFI LEA EAX,DWORD PTR SS:[EBP-528]
.    50            PUSH EAX
.    8D85 D8FBFFFI LEA EAX,DWORD PTR SS:[EBP-428]
.    50            PUSH EAX
.    FFD6          CALL ESI                                  kernel32.GetVolumePathNamesForVolumeNameA
.    8D8D D8FAFFFI LEA ECX,DWORD PTR SS:[EBP-528]
.    8D51 01       LEA EDX,DWORD PTR DS:[ECX+1]
>    8A01          MOV AL,BYTE PTR DS:[ECX]
.    41            INC ECX                                   kernel32.768B5CD4
.    84C0          TEST AL,AL
.^   75 F9         JNZ SHORT 123.00452C42
.    2BCA          SUB ECX,EDX
.v   75 5C         JNZ SHORT 123.00452CA9
.    FF15 10814500 CALL DWORD PTR DS:[<&KERNEL32.GetLogical  GetLogicalDrives
.    8BD0          MOV EDX,EAX
.    83FA 04       CMP EDX,4
     73 04         JNB SHORT 123.00452C5E

dress=0029F7A8
5CD4 (kernel32.768B5CD4)

Hex dump                                            ASCII
43 3A 5C 00|00 00 08 00|3C F8 29 00|41 5C FF 76  C:\.....<ø).A\ÿv
8A 15 2C 75|A4 F8 29 00|78 5C FF 76|A0 27 FF 76  Š.,u¤ø).x\ÿv 'ÿv
D6 1F 73 76|00 00 00 00|FF FF FF FF|00 F0 FD 7F  Ösv....ÿÿÿÿ.ðý
40 2E 08 00|58 2D 08 00|A8 2B 08 00|28 2A 08 00  @...X-..¨+..(*..
18 29 08 00|70 33 08 00|D0 34 08 00|38 31 08 00  .)..p3..Ð4..81..
08 38 08 00|80 36 08 00|30 3C 08 00|A8 3A 08 00  .8..€6..0<..¨:..

0029E3C0  000
0029E3C4  000
0029E3C8  000
0029E3CC  000
0029E3D0  002
0029E3D4  752
0029E3D8  CBE
```

Ransomware extracting all volumes name using GetLogicalDrives

Figure 7 Ragnar_locker Ransomware Enumerating Hard Drive Volumes

After retrieving the volume names, the Ransomware then calls *GetVolumeInformationA()* API to get the details of the volume.



```
00112CA9  >  68 00010000   PUSH 100                                   pFileSystemNameSize = 00000100
00112CAE  .  8D85 10EEFFFI LEA EAX,DWORD PTR SS:[EBP-11F0]
00112CB4  .  50            PUSH EAX                                   pFileSystemNameBuffer = 002FF7C8
00112CB5  .  6A 00         PUSH 0                                     pFileSystemFlags = NULL
00112CB7  .  6A 00         PUSH 0                                     pMaxFilenameLength = NULL
00112CB9  .  8D45 94       LEA EAX,DWORD PTR SS:[EBP-6C]
00112CBC  .  50            PUSH EAX                                   pVolumeSerialNumber = 002FF7C8
00112CBD  .  68 00010000   PUSH 100                                   MaxVolumeNameSize = 100 (256.)
00112CC2  .  8D85 10EDFFFI LEA EAX,DWORD PTR SS:[EBP-12F0]
00112CC8  .  50            PUSH EAX                                   VolumeNameBuffer = 002FF7C8
00112CC9  .  8D85 D8FAFFFI LEA EAX,DWORD PTR SS:[EBP-528]
00112CCF  .  50            PUSH EAX                                   RootPathName = "C:\\"
00112CD0  .  FF15 C0801100 CALL DWORD PTR DS:[<&KERNEL32.GetVolume]   GetVolumeInformationA
00112CD6  .  68 00010000   PUSH 100
00112CDB  .  8D85 D8FBFFFI LEA EAX,DWORD PTR SS:[EBP-428]
```

Figure 8 Retrieves Volume Details

The malware now prepares the key required to encrypt the files in the latter part of its execution. The malware uses cryptographic APIs such as *CryptAcquireContextW()* , *CryptGenRandom()* and *CryptReleaseContext()* to generate random keys.

Then, the malware uses a custom decryption logic which decrypts the strings that have information about the name of the services. After identifying the names of the services, the Ransomware checks for their presence and terminates them if the services are actively running on the victim’s machine. Some of these services include VSS, SQL, Memtas, etc.

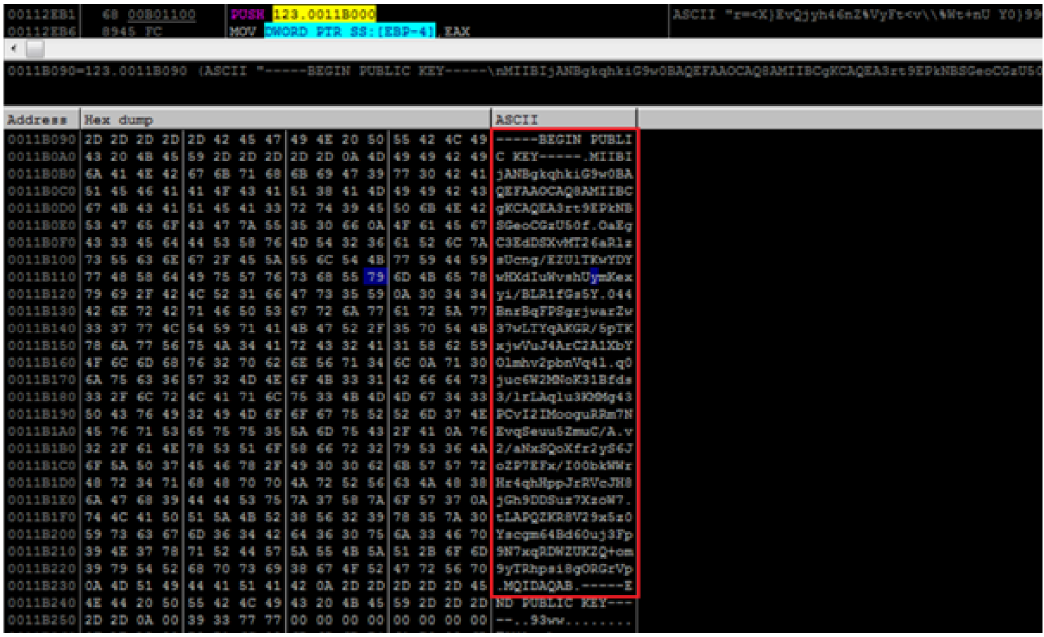
To identify the services running in the machine, the Ransomware first calls *OpenSCManagerA()* API, which establishes a connection to the service control manager that gives the TA access to the service control manager database.

Upon gaining access to this database, the following APIs() will be called:

- *OpenServiceA()* – Opens the specified service.
- *QueryServiceStatusEx()* – Gets the status of the service.
- *EnumDependentServiceA()* – Retrieves the dependent services.
- *ControlService()* – takes control of the service for stopping.

If *OpenSCManagerA()* API fails to get the handle to Service Control Manager (SCM), then the Ransomware skips calling the above service-related APIs.

The Ransomware then proceeds to execute *CreateProcessW()* API to call wmi/vssadmin to delete any shadow copies in the system. After this, the Ransomware decrypts the RSA public key, encrypting the randomly generated key, as shown in Figure 9.



```
00112EB1    68 00B01100     PUSH 123.0011B000                         ASCII "r=<X}EvQjyh46nZ%VyFt<v\\%Wt+nU Y0}99
00112EB6    8945 FC         MOV DWORD PTR SS:[EBP-4],EAX

0011B090=123.0011B090 (ASCII "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3rt9EPkNBSGeoCGzU50

Address  Hex dump                                         ASCII
0011B090 2D 2D 2D 2D 2D 42 45 47 49 4E 20 50 55 42 4C 49 -----BEGIN PUBLI
0011B0A0 43 20 4B 45 59 2D 2D 2D 2D 2D 0A 4D 49 49 42 49 C KEY-----.MIIBI
0011B0B0 6A 41 4E 42 67 6B 71 68 6B 69 47 39 77 30 42 41 jANBgkqhkiG9w0BA
0011B0C0 51 45 46 41 41 4F 43 41 51 38 41 4D 49 49 42 43 QEFAAOCAQ8AMIIBC
0011B0D0 67 4B 43 41 51 45 41 33 72 74 39 45 50 6B 4E 42 gKCAQEA3rt9EPkNB
0011B0E0 53 47 65 6F 43 47 7A 55 35 30 66 0A 4F 61 45 67 SGeoCGzU50f.OaEg
0011B0F0 43 33 45 64 44 53 58 76 4D 54 32 36 61 52 6C 7A C3EdDSXvMT26aRlz
0011B100 73 55 63 6E 67 2F 45 5A 55 6C 54 4B 77 59 44 59 sUcng/EZUlTKwYDY
0011B110 77 48 58 64 49 75 57 76 73 68 55 79 6D 4B 65 78 wHXdIuWvshUymKex
0011B120 79 69 2F 42 4C 52 31 66 47 73 35 59 0A 30 34 34 yi/BLR1fGs5Y.044
0011B130 42 6E 72 42 71 46 50 53 67 72 6A 77 61 72 5A 77 BnrBqFPSgrjwarZw
0011B140 33 37 77 4C 54 59 71 41 4B 47 52 2F 35 70 54 4B 37wLTYqAKGR/5pTK
0011B150 78 6A 77 56 75 4A 34 41 72 43 32 41 31 58 62 59 xjwVuJ4ArC2A1XbY
0011B160 4F 6C 6D 68 76 32 70 62 6E 56 71 34 6C 0A 71 30 Olmhv2pbnVq4l.q0
0011B170 6A 75 63 36 57 32 4D 4E 6F 4B 33 31 42 66 64 73 juc6W2MNoK31Bfds
0011B180 33 2F 6C 72 4C 41 71 6C 75 33 4B 4D 4D 67 34 33 3/lrLAqlu3KMMg43
0011B190 50 43 76 49 32 49 4D 6F 6F 67 75 52 52 6D 37 4E PCvI2IMooguRRm7N
0011B1A0 45 76 71 53 65 75 75 35 5A 6D 75 43 2F 41 0A 76 EvqSeuu5ZmuC/A.v
0011B1B0 32 2F 61 4E 78 53 51 6F 58 66 72 32 79 53 36 4A 2/aNxSQoXfr2yS6J
0011B1C0 6F 5A 50 37 45 46 78 2F 49 30 30 62 6B 57 57 72 oZP7EFx/I00bkWWr
0011B1D0 48 72 34 71 68 48 70 70 4A 72 52 56 63 4A 48 38 Hr4qhHppJrRVcJH8
0011B1E0 6A 47 68 39 44 44 53 75 7A 37 58 7A 6F 57 37 0A jGh9DDSuz7XzoW7.
0011B1F0 74 4C 41 50 51 5A 4B 52 38 56 32 39 78 35 7A 30 tLAPQZKR8V29x5z0
0011B200 59 73 63 67 6D 36 34 42 64 36 30 75 6A 33 46 70 Yscgm64Bd60uj3Fp
0011B210 39 4E 37 78 71 52 44 57 5A 55 4B 5A 51 2B 6F 6D 9N7xqRDWZUKZQ+om
0011B220 39 79 54 52 68 70 73 69 38 67 4F 52 47 72 56 70 9yTRhpsi8gORGrVp
0011B230 0A 4D 51 49 44 41 51 41 42 0A 2D 2D 2D 2D 2D 45 .MQIDAQAB.-----E
0011B240 4E 44 20 50 55 42 4C 49 43 20 4B 45 59 2D 2D 2D ND PUBLIC KEY---
0011B250 2D 2D 0A 00 39 33 77 77 00 00 00 00 00 00 00 00 --..93ww........
```

Figure 9 RSA Public Key

The Ransomware decrypts the ransom notes in the memory, shown to the victims after file encryption on their system. Then, it gets the device name and creates a unique hash used to generate the ransom note name in the below format.

> RGNR_[Unique-hash].txt

It calls *SHGetSpecialFolderPathW ()* API, gets the path of the Public folder (c:\user\public\Documents), and creates ransom notes in it. The ransom note content is then written using *WriteFile() API*.

The Ransomware then searches for files in the Windows directory for encryption using the *FindFirstFileW()* and *FindNextFileW()* APIs.

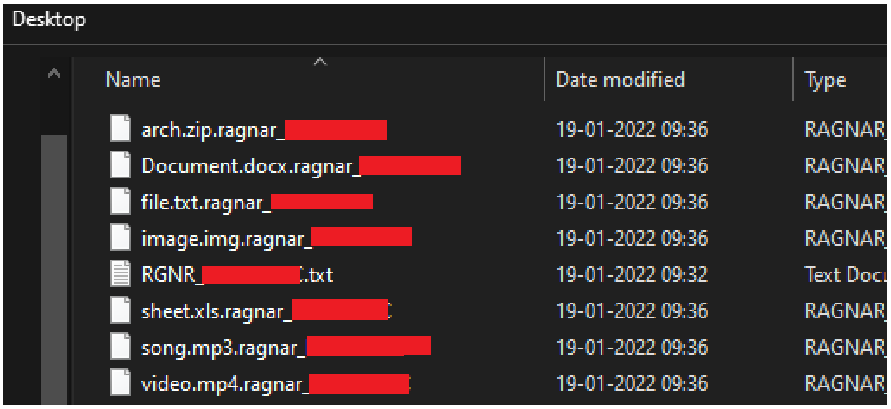
Before initiating encryption, the ransomware checks and excludes specific folders from encryption – such as Windows, Tor Browser, Google, Opera.

The Ransomware also excludes certain files from encryption such as RGNR_[unique_hash].txt, autorun.inf, boot.ini, amongst others.

Specific extensions are also exempted from encryption – such as .db, .sys, .dll.

The Ransomware specifically excludes these files, folders, and extensions to ensure that TAs are not damaging any system-critical files. Victims will thus have access to the affected device to pay the ransom after successful encryption.

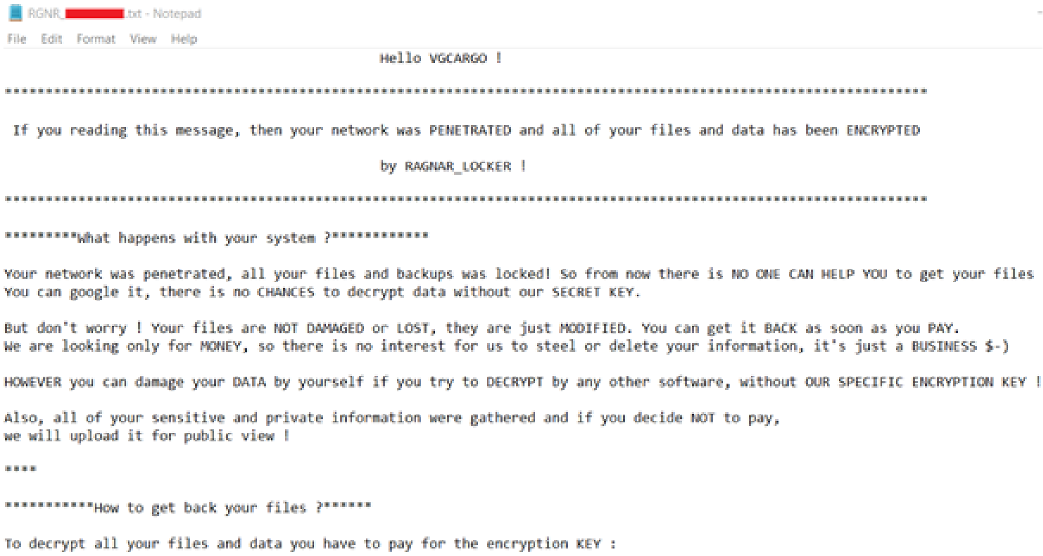
Finally, the Ransomware encrypts the file using the salsa20 algorithm and displays a ransom note on the victims' machine. As shown in the figure below, the encrypted files will have appended extension ragnar_[unique_hash] in the victims' device.



Name	Date modified	Type
arch.zip.ragnar_[redacted]	19-01-2022 09:36	RAGNAR_
Document.docx.ragnar_[redacted]	19-01-2022 09:36	RAGNAR_
file.txt.ragnar_[redacted]	19-01-2022 09:36	RAGNAR_
image.img.ragnar_[redacted]	19-01-2022 09:36	RAGNAR_
RGNR_[redacted].txt	19-01-2022 09:32	Text Doc
sheet.xls.ragnar_[redacted]	19-01-2022 09:36	RAGNAR_
song.mp3.ragnar_[redacted]	19-01-2022 09:36	RAGNAR_
video.mp4.ragnar_[redacted]	19-01-2022 09:36	RAGNAR_

Figure 10 Encrypted Files on the Machine

In their ransom note below, the TAs have instructed victims to contact them via qTox and have also given an Email ID: *cargowelcome@protonmail[.]com* in case the victim cannot contact them through qTox to pay the ransom of 25 Bitcoin (BTC) for the decryption key.

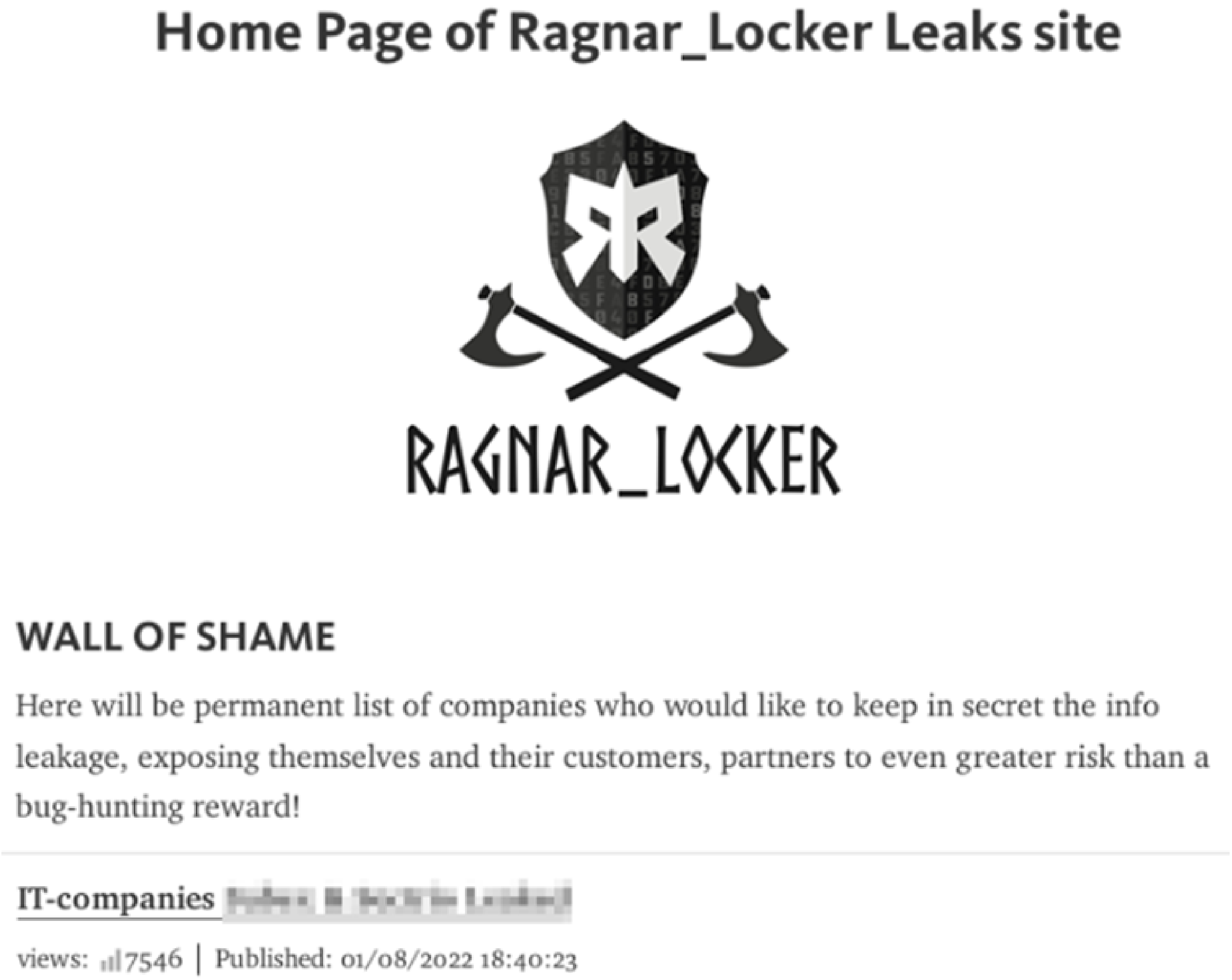


```
                                             Hello VGCARGO !

*****************************************************************************************************************

 If you reading this message, then your network was PENETRATED and all of your files and data has been ENCRYPTED

                                             by RAGNAR_LOCKER !

*****************************************************************************************************************

*********What happens with your system ?************

Your network was penetrated, all your files and backups was locked! So from now there is NO ONE CAN HELP YOU to get your files
You can google it, there is no CHANCES to decrypt data without our SECRET KEY.

But don't worry ! Your files are NOT DAMAGED or LOST, they are just MODIFIED. You can get it BACK as soon as you PAY.
We are looking only for MONEY, so there is no interest for us to steel or delete your information, it's just a BUSINESS $-)

HOWEVER you can damage your DATA by yourself if you try to DECRYPT by any other software, without OUR SPECIFIC ENCRYPTION KEY !

Also, all of your sensitive and private information were gathered and if you decide NOT to pay,
we will upload it for public view !

****

***********How to get back your files ?******

To decrypt all your files and data you have to pay for the encryption KEY :
```

Figure 11 Ransom note

Other Observations

Cyble Research Labs had found that the TAs leaked their victim's details on their leak website when victims did not pay the ransom. The following figure showcases the Ragnar_locker's leak website with recent victims.



Home Page of Ragnar_Locker Leaks site

RAGNAR_LOCKER

WALL OF SHAME

Here will be permanent list of companies who would like to keep in secret the info leakage, exposing themselves and their customers, partners to even greater risk than a bug-hunting reward!

IT-companies [redacted]

views: 7546 | Published: 01/08/2022 18:40:23

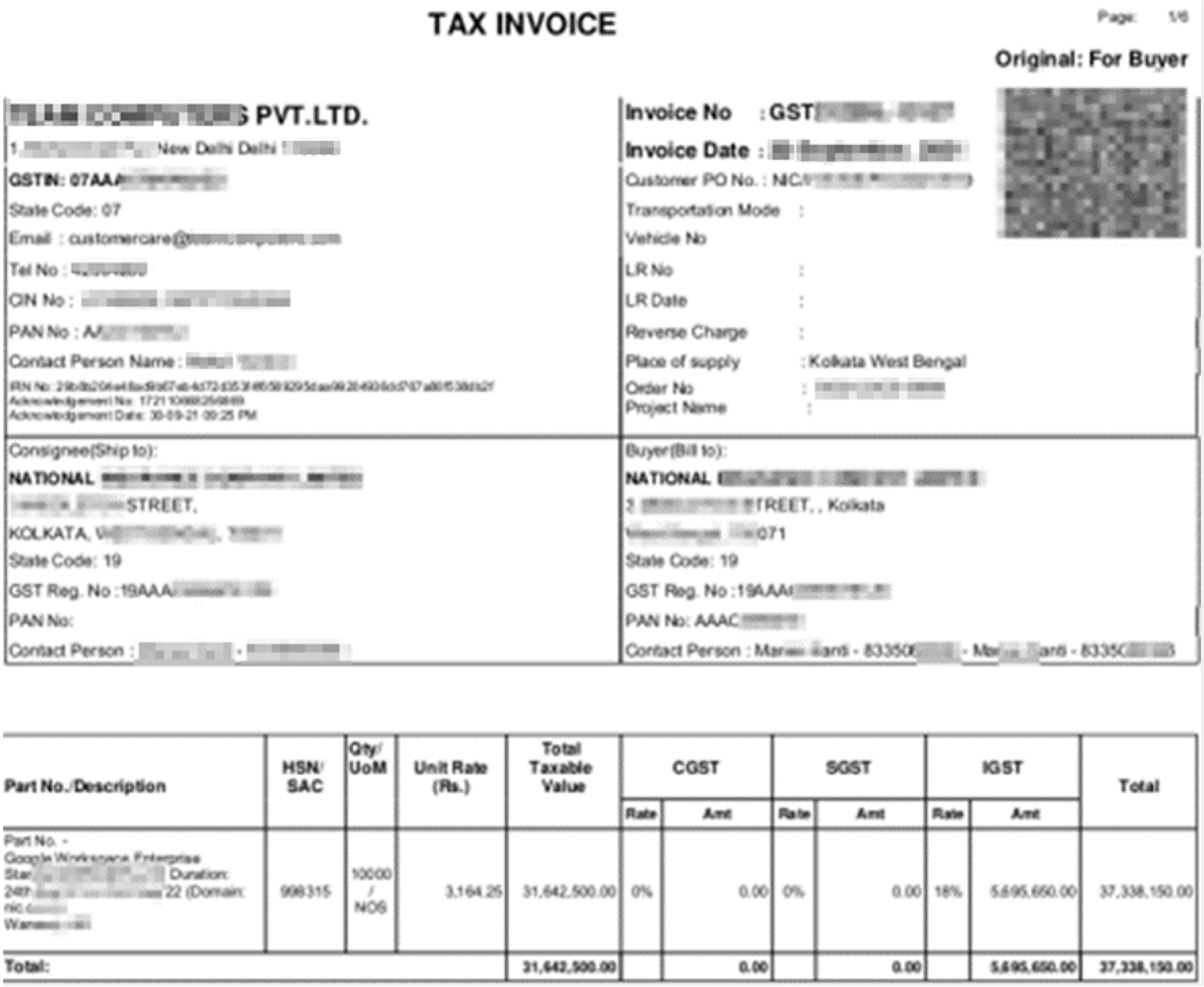
Figure 12 Victims Mentioned on Leak Site

As per their leak site, the Ragnar_locker ransomware group claims to be a team of cyber security enthusiasts working to make a profit.

The group alleges that their primary motivation to attack organizations is to help them improve their security measures. In addition, they want companies to take responsibility for securely storing the personal data of their clients and partners.

In one case, it was observed that the TAs had stolen the data of a victim's machine and shared the same on their leak site. The stolen data claimed by the TAs include name, PAN Number, mobile numbers, GST numbers, etc.

The victim's data posted on the TAs leak site is shown in the figure below.



TAX INVOICE

Page: 1/6

Original: For Buyer

[redacted] PVT.LTD.
1.[redacted] New Delhi Delhi [redacted]
GSTIN: 07AA[redacted]
State Code: 07
Email : customercare@[redacted]
Tel No : [redacted]
CIN No : [redacted]
PAN No : A[redacted]
Contact Person Name : [redacted]
IRN No: [illegible]
Acknowledgement No: 1721 10889258869
Acknowledgement Date: 30-09-21 09:25 PM

Invoice No : GST[redacted]
Invoice Date : [redacted]
Customer PO No. : NIC/[redacted]
Transportation Mode :
Vehicle No
LR No :
LR Date :
Reverse Charge :
Place of supply : Kolkata West Bengal
Order No : [redacted]
Project Name :

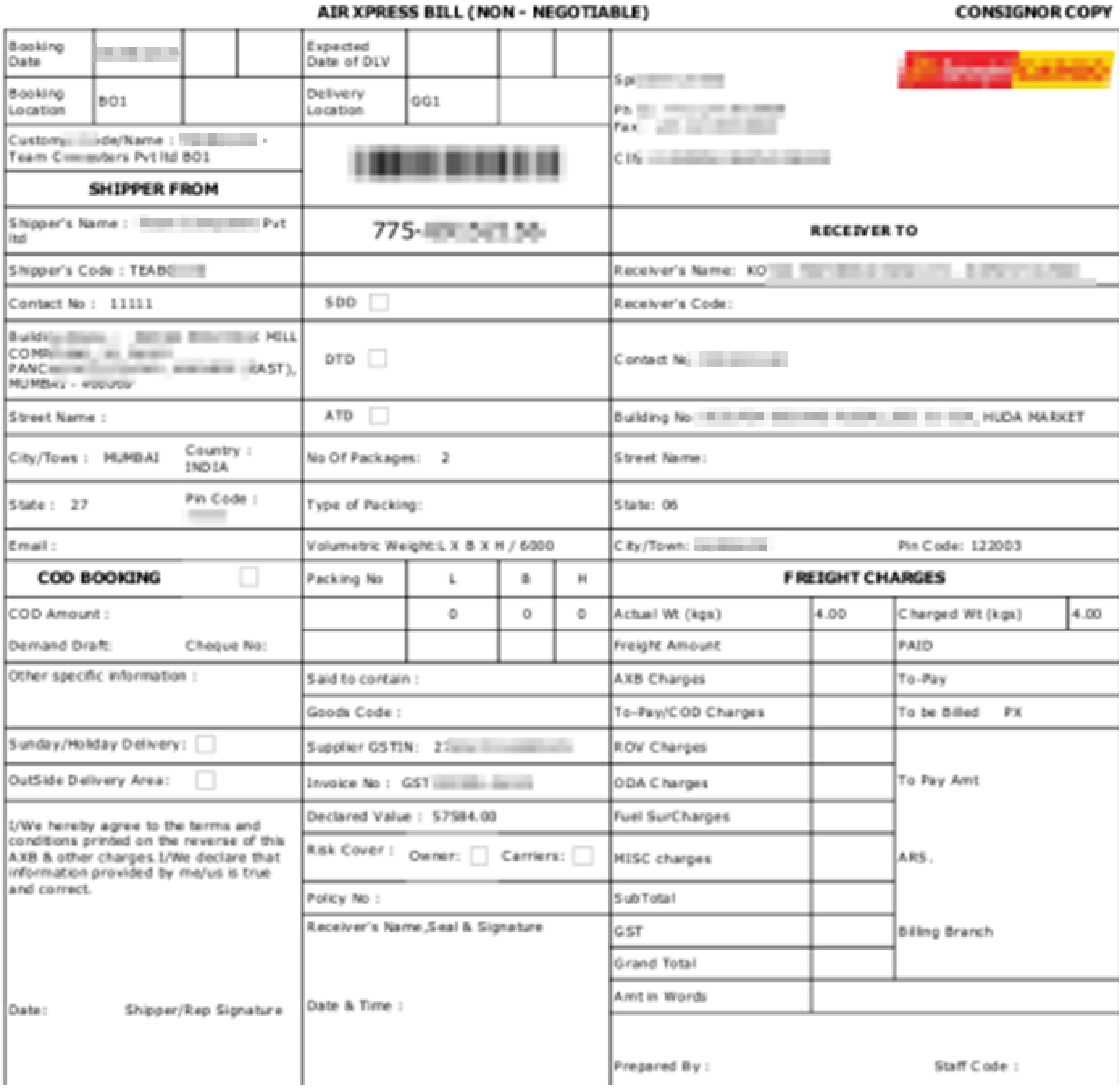
[redacted]

Consignee(Ship to):
NATIONAL [redacted]
[redacted] STREET,
KOLKATA, [redacted]
State Code: 19
GST Reg. No :19AAA[redacted]
PAN No:
Contact Person : [redacted] - [redacted]

Buyer(Bill to):
NATIONAL [redacted]
2 [redacted] TREET, , Kolkata
[redacted] 071
State Code: 19
GST Reg. No :19AA[redacted]
PAN No: AAAC[redacted]
Contact Person : Mar[redacted]anti - 83350[redacted] - Ma[redacted]anti - 8335[redacted]

Part No./Description	HSN/ SAC	Qty/ UoM	Unit Rate (Rs.)	Total Taxable Value	CGST		SGST		IGST		Total
					Rate	Amt	Rate	Amt	Rate	Amt	
Part No. - Google Workspace Enterprise Star[redacted] Duration: 24th [redacted] 22 (Domain: nic.[redacted] Wan[redacted]	998315	10000 / NOS	3,164.25	31,642,500.00	0%	0.00	0%	0.00	18%	5,695,650.00	37,338,150.00
Total:				31,642,500.00		0.00		0.00		5,695,650.00	37,338,150.00

Figure 13 Tax Invoice



AIR XPRESS BILL (NON - NEGOTIABLE) CONSIGNOR COPY

Booking Date [redacted]	Expected Date of DLV	Sp[redacted] Ph [redacted] Fax [redacted] CIN [redacted]
Booking Location BO1	Delivery Location GG1	
Custom[redacted]de/Name : [redacted] - Team C[redacted] Pvt ltd BO1	[redacted]	
SHIPPER FROM		
Shipper's Name : [redacted] Pvt ltd	775-[redacted]	RECEIVER TO
Shipper's Code : TEAB[redacted]		Receiver's Name: KO[redacted]
Contact No : 11111	SDD □	Receiver's Code:
Buildi[redacted] MILL COM[redacted] PANC[redacted]EAST), MUMBA[redacted]	DTD □	Contact No[redacted]
Street Name :	ATD □	Building No[redacted] HUDA MARKET
City/Tows : MUMBAI Country : INDIA	No Of Packages: 2	Street Name:
State : 27 Pin Code : [redacted]	Type of Packing:	State: 06
Email :	Volumetric Weight:L X B X H / 6000	City/Town: [redacted] Pin Code: 122003

COD BOOKING □	Packing No	L	B	H	FREIGHT CHARGES			
COD Amount :		0	0	0	Actual Wt (kgs)	4.00	Charged Wt (kgs)	4.00
Demand Draft: Cheque No:					Freight Amount		PAID	
Other specific information :	Said to contain :				AXB Charges		To-Pay	
	Goods Code :				To-Pay/COD Charges		To be Billed PX	
Sunday/Holiday Delivery: □	Supplier GSTIN: 2[redacted]				ROV Charges		To Pay Amt	
OutSide Delivery Area: □	Invoice No : GST[redacted]				ODA Charges			
I/We hereby agree to the terms and conditions printed on the reverse of this AXB & other charges.I/We declare that information provided by me/us is true and correct.	Declared Value : 57584.00				Fuel SurCharges			
	Risk Cover : Owner: □ Carriers: □				MISC charges		ARS.	
	Policy No :				SubTotal			
	Receiver's Name,Seal & Signature				GST		Billing Branch	
					Grand Total			
Date: Shipper/Rep Signature	Date & Time :				Amt in Words			
					Prepared By :		Staff Code :	

Figure 14

Airways Bill

Conclusion

There are likely multiple variants of Ragnar_locker ransomware active in the wild. In addition, TAs keep improving their code with new features to evolve their Ransomware-as-a-Service (RaaS) business model with new Tactics, Techniques, and Procedures (TTPs) to target devices. Based on these observations, we can safely assume that there may be further enhancements in upcoming variants of Ragnar_locker.

We continuously monitor Ragnar_locker's extortion campaigns and update our readers with the latest information.

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 needed to prevent ransomware attacks

- 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.
- Refrain from opening untrusted links and email attachments without verifying their authenticity.

Users should take the following steps after the ransomware attack

- Detach infected devices on the same network.
- Disconnect external storage devices if connected.
- Inspect system logs for suspicious events.

Impacts and cruciality Of Ragnar_locker Ransomware

- Loss of Valuable data.
- Loss of organization’s reliability or integrity.
- Loss of organization’s businesses information.
- Disruption in organization operation.
- Economic loss.

MITRE ATT&CK® Techniques

Tactic	Technique ID	Technique Name
Initial Access	T1078	– Valid Accounts
Execution	T1059	– Command and Scripting Interpreter
Privilege Escalation	T1548 T1134	– Abuse Elevation Control Mechanism – Access Token Manipulation
Defense Evasion	T1112 T1027 T1562.001	– Modify Registry – Obfuscated Files or Information – Impair Defenses: Disable or Modify Tools
Discovery	T1082 T1083 T1135	– System Information Discovery – File and Directory Discovery – Network Share Discovery
Impact	T1490 T1489 T1486	– Inhibit System Recovery – Service Stop – Data Encrypted for Impact

Indicators of Compromise (IOCs)

Indicators	Indicator type	Description
b6663af099538a396775273d79cb6fff99a18e2de2a8a2a106de8212cc44f3e2	SHA256	Ragnar_locker Executable
ac16f3e23516cf6b22830c399b4aba9706d37adceb5eb8ea9960f71f1425df79	SHA256	Ragnar_locker Executable

68eb2d2d7866775d6bf106a914281491d23769a9eda88fc078328150b8432bb3	SHA256	Ragnar_locker Executable
b670441066ff868d06c682e5167b9dbc85b5323f3acfbbc044cabc0e5a594186	SHA256	Ragnar_locker Executable
9bdd7f965d1c67396afb0a84c78b4d12118ff377db7efdca4a1340933120f376	SHA256	Ragnar_locker Executable
dd5d4cf9422b6e4514d49a3ec542cffb682be8a24079010cda689afbb44ac0f4	SHA256	Ragnar_locker Executable
63096f288f49b25d50f4aea52dc1fc00871b3927fa2a81fa0b0d752b261a3059	SHA256	Ragnar_locker Executable
a8ee0fafbd7b84417c0fb31709b2d9c25b2b8a16381b36756ca94609e2a6fcf6	SHA256	Ragnar_locker Executable
5fc6f4cfb0d11e99c439a13b6c247ec3202a9a343df63576ce9f31cffcdbaf76	SHA256	Ragnar_locker Executable
1472f5f559f90988f886d515f6d6c52e5d30283141ee2f13f92f7e1f7e6b8e9e	SHA256	Ragnar_locker Executable
ec35c76ad2c8192f09c02eca1f263b406163470ca8438d054db7adcf5bfc0597	SHA256	Ragnar_locker Executable
68eb2d2d7866775d6bf106a914281491d23769a9eda88fc078328150b8432bb3	SHA256	Ragnar_locker Executable