

MBSD Blog

Analyzing “Ragnar Locker” ransomware that threatens a company by its name

Takashi Yoshikawa

Senior Malware Analyst, Cyber Intelligence Group
Mitsu Bussan Secure Directions, Inc.

Kei Sugawara

Senior Malware Analyst, Cyber Intelligence Group
Mitsu Bussan Secure Directions, Inc.

November 2020

Table of Contents

1.	Introduction.....	3
2.	Specimen	3
3.	A number of anti-analysis features	6
4.	Checking language of infected PC	10
5.	Interfering with system recovery	11
6.	Stopping processes and services	13
7.	Encryption of files.....	17
8.	Creating and displaying threatening letter	22
9.	Supplemental information - Checking behavior of execution arguments.....	26
10.	Ragnar Locker leak site	27
11.	Summary.....	29
12.	About us	30

1. Introduction

In November 2020, CAPCOM's cyber-attack news widely spread out primarily in foreign media. According to the news, the cyber-attack involved ransomware called Ragnar Locker. After the incident happened, Ragnar Locker attack group actually released a criminal statement on November 9, 2020.

Based on the publicly available data, we investigated and found a suspected specimen on VirusTotal.

This article describes the results of the analysis of the relevant specimen.

Please note that after we discovered the specimen on VirusTotal, we contacted a person who supplied information to publications such as Bleeping Computer. The person has confirmed the specimen is identical to the one available on the VirusTotal. At the same time however, we didn't verify that the specimen was actually used in CAPCOM cyber-attack. Therefore, the content of this article only describes the result of my analysis of the same sample as Ragnar Locker mentioned in the publication.

Hash value: 6f559fd57304197443b71d8bf553cce3c9de8d53

2. Specimen

The relevant sample was submitted from Japan to VirusTotal on November 4, 2020, and only one sample was submitted at the time of this analysis.

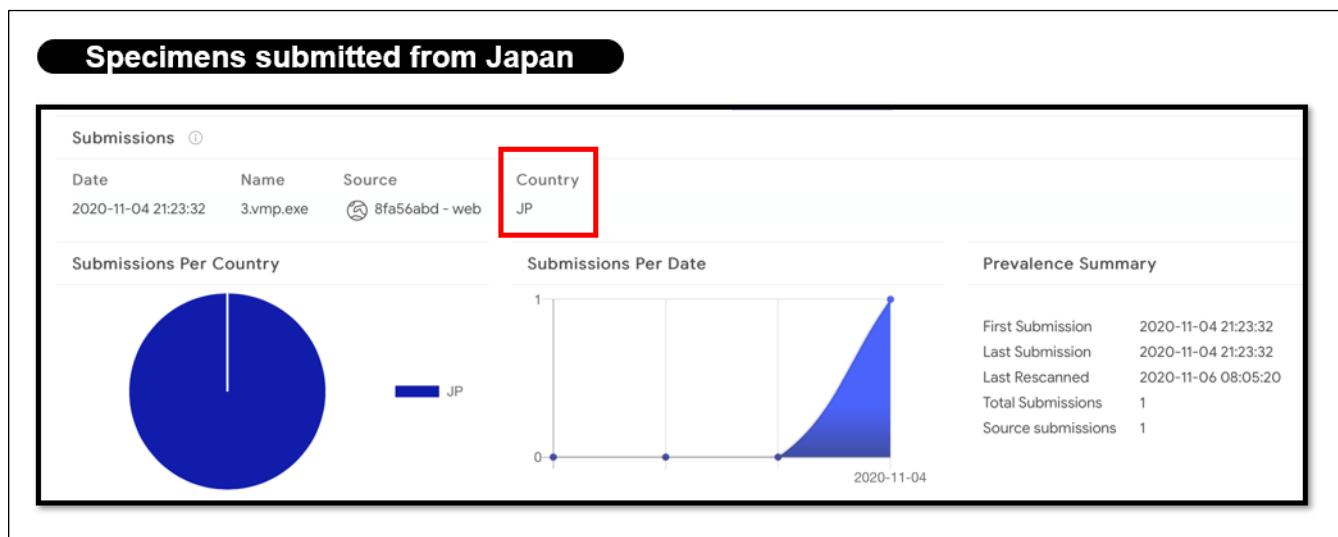


Fig. 1 - Ragnar Locker specimen uploaded to VirusTotal from Japan

The original file name of the file is uploaded with the name "3.vmp.exe", and the file icon does not exist as shown in the following figure.

Sample executable file		
The executable file has no file icon.		
File Name	Type	Size
9416e5a57e6de00c685560fa9fee761126569d123f62060792bf2049ebba4151.exe	Application	5,234 KB

Fig. 2 - Executable file of Ragnar Locker does not have an icon

Ragnar Locker executable file for this specimen has a valid digital signature, as shown in the following figure, and the time stamp of the digital signature is November 1, 2020. The official announcement of CAPCOM stated that the incident occurred from November 2. If this specimen was actually used for the corresponding cyber-attack, then it may have been signed immediately before the attack.

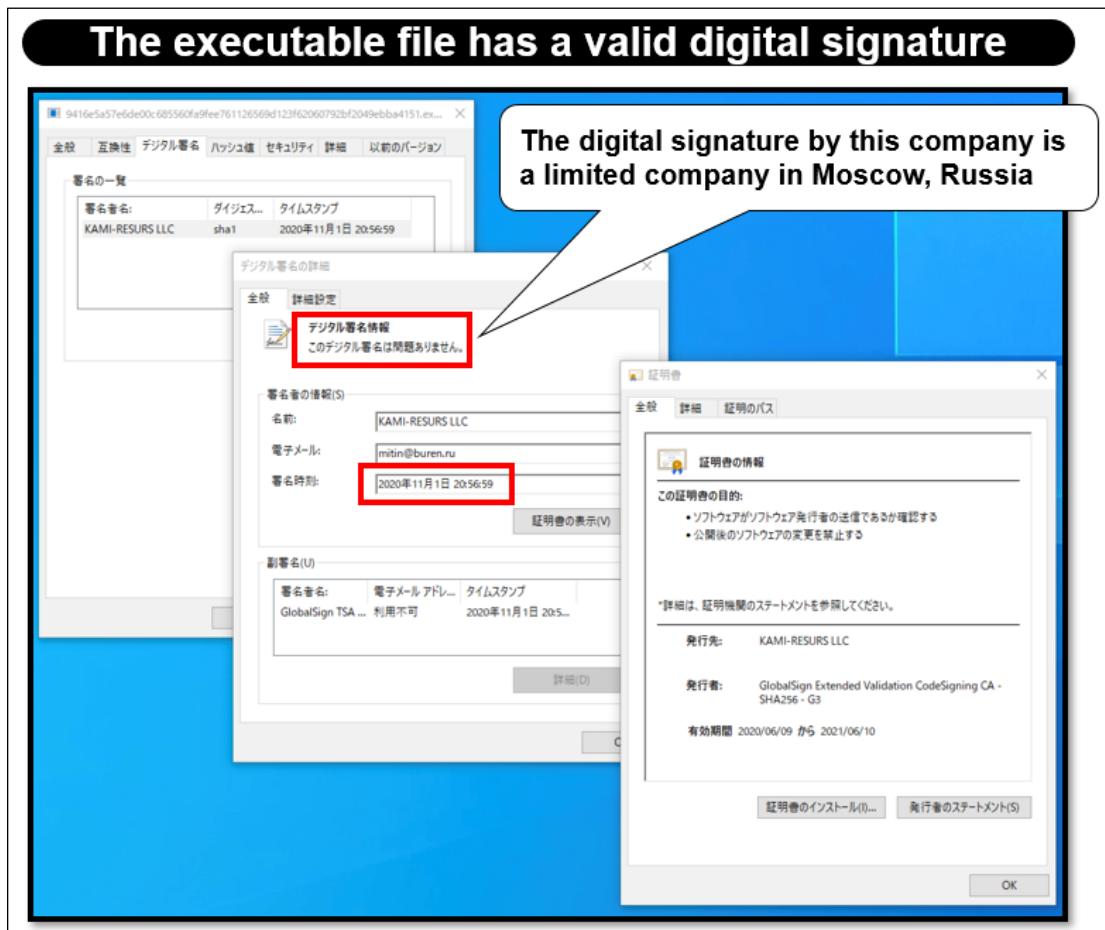


Fig. 3 - A valid digital signature found in the Ragnar Locker executable file

In addition, Time Date Stamp in the PE header, which indicates the compilation date and time of the executable file, is October 20, 2020. This indicates, same as digital signature, the creation of the ransomware itself may have created just before the attack.

Compilation date and time information for the executable file

Time Date Stamp for the date and time when the EXE file was compiled is October 20, 2020.

Offset	Name	Value	Meaning
84	Machine	14c	Intel 386
86	Sections Count	7	7
88	Time Date Stamp	5f8eff91	火曜日, 20.10.2020 15:17:37 UTC
8C	Ptr to Symbol Table	0	0
90	Num. of Symbols	0	0
94	Size of OptionalHeader	e0	224
96	Characteristics	103	
		1	Relocation info stripped from file.
		2	File is executable (i.e. no unresolved external references).
		100	32 bit word machine.

Fig. 4 - Ragnar Locker executable compilation date and time

3. A number of anti-analysis features

Ragnar Locker EXE-file has many analysis prevention features, so it is not easy to analyze.

First, it is packaged by VMProtect, a very robust commercial protector, as follows:

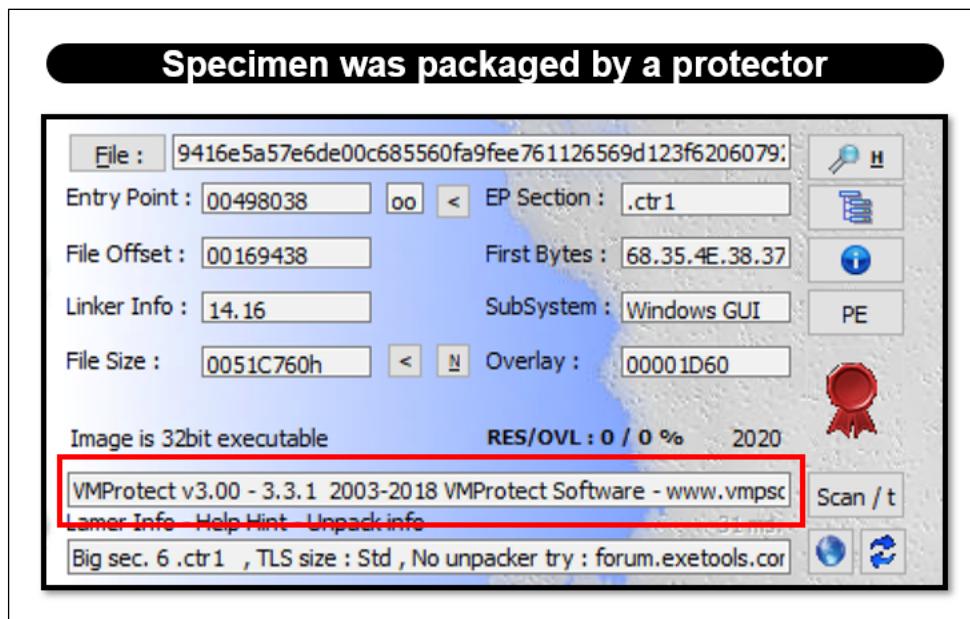


Fig. 5 Analyzed by a surface analysis tool

If you simply try to analyze using a debugger, the packer shows an error message as shown below, and Ragnar Locker executable file is forcibly terminated, and the analysis cannot continue.

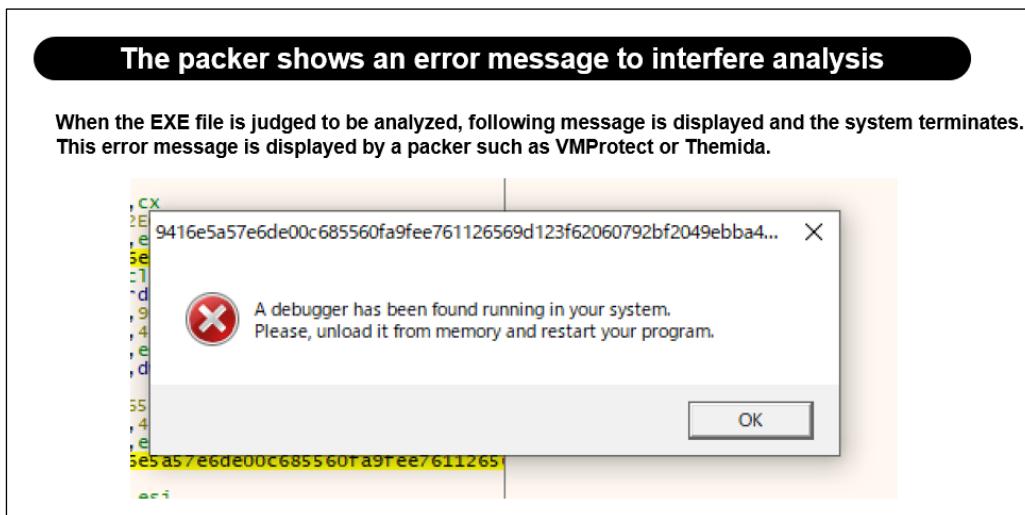


Fig. 6 - Debugger detection message that interfere analysis

In addition, even after the main code of Ragnar Locker is successfully unpacked onto memory, a mechanism that interferes API hook is implemented.

Specifically, the first 5 bytes of the DbgUiRemoteBreakin function of the ntdll.dll (which is a system DLL) is written in order to jump to its own original hook function, so that it is not able to be attached by the debugger (see below figure).

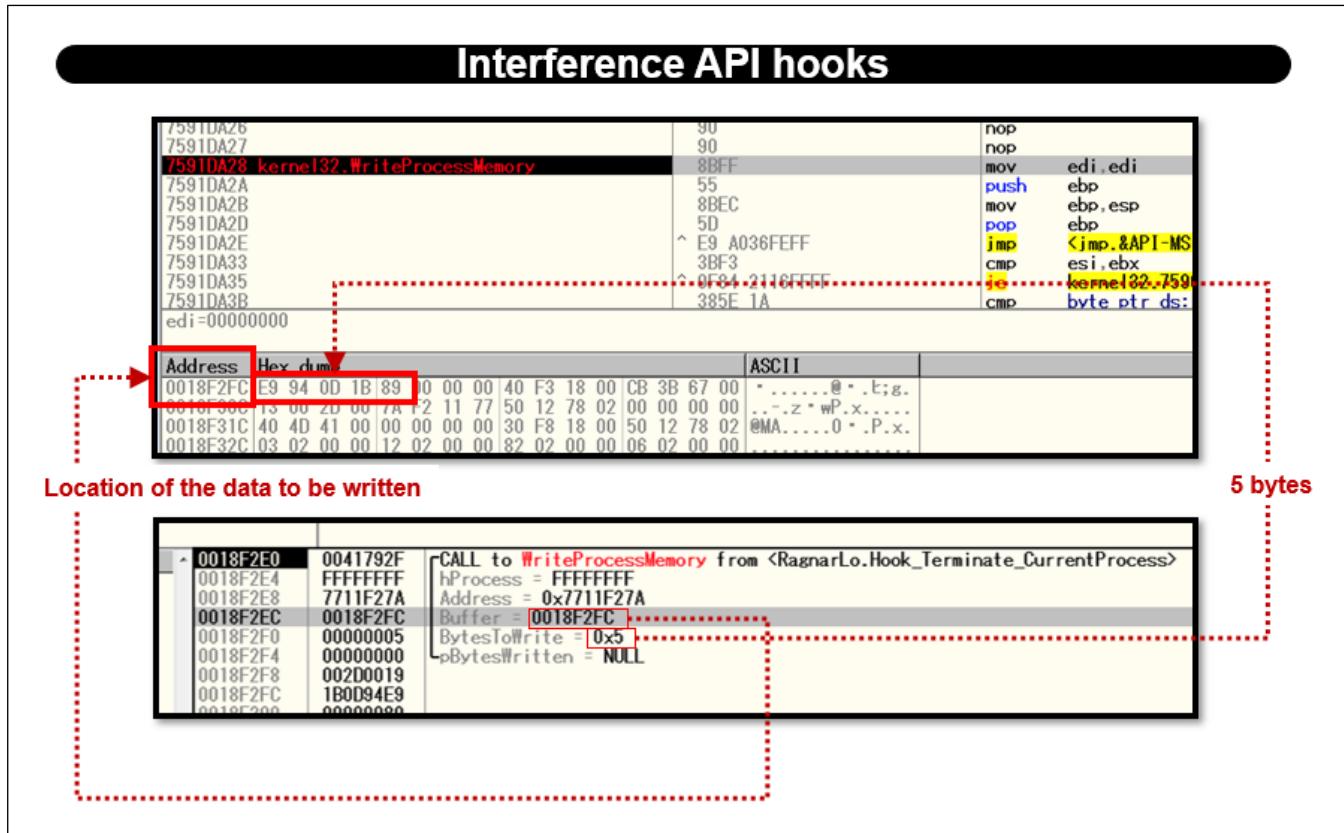


Fig. 7 - Ragnar Locker writes the first 5 bytes of the system DLL

The following illustration shows before-and-after comparison of DbgUiRemoteBreakin function in ntdll.dll. Before Ragnar Locker writes the first 5 bytes, it shows "6A 08 68 50 BB", which have been tampered to "E9 94 0D 1B 89". This is a jump instruction with destination address to Ragnar Locker.

The reason why Ragnar Locker tampers the first 5 bytes of the DbgUiRemoteBreakin function is that Ragnar Locker does not want to be debugged by DbgUiRemoteBreakin function. Usually debug function attaches to the object process and the DbgUiRemoteBreakin function is called. However, if the first 5 bytes of the DbgUiRemoteBreakin function is tampered, then it will automatically jump to Ragnar Locker's function.

Tampered the first 5 bytes of DbgUiRemoteBreakin, before-and-after

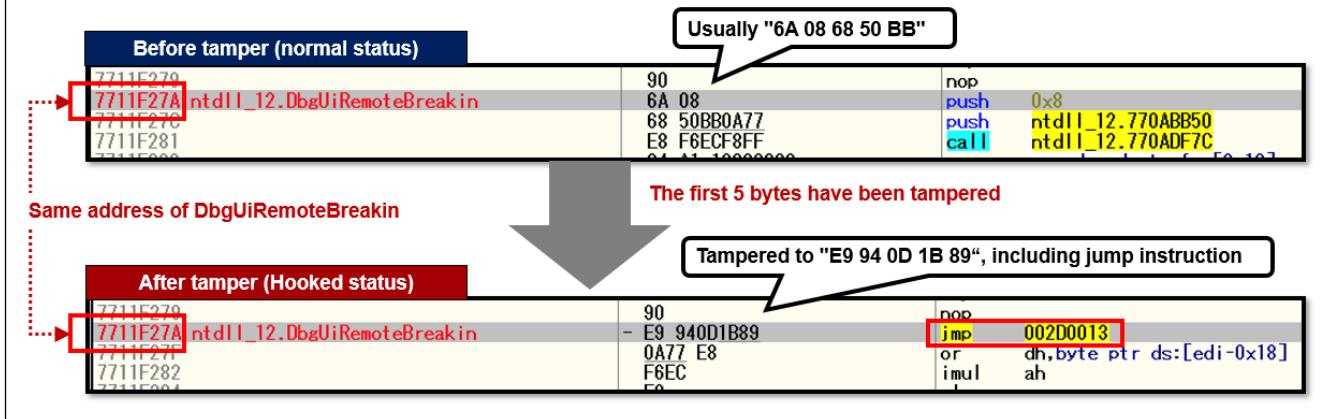


Fig. 8 - Comparing the first 5 bytes of the APIs

The jump destination address, tampered by the hook, instructs jump to another address again and again, then it finally jumps to an address that instructs force termination of its own process (see the figure below). This means that when the analyst attaches the debugger, the DbgUiRemoteBreakin function is called, and Ragnar Locker terminates, resulting that the analyst cannot continue analysis.

Process after DbgUiRemoteBreakin is called

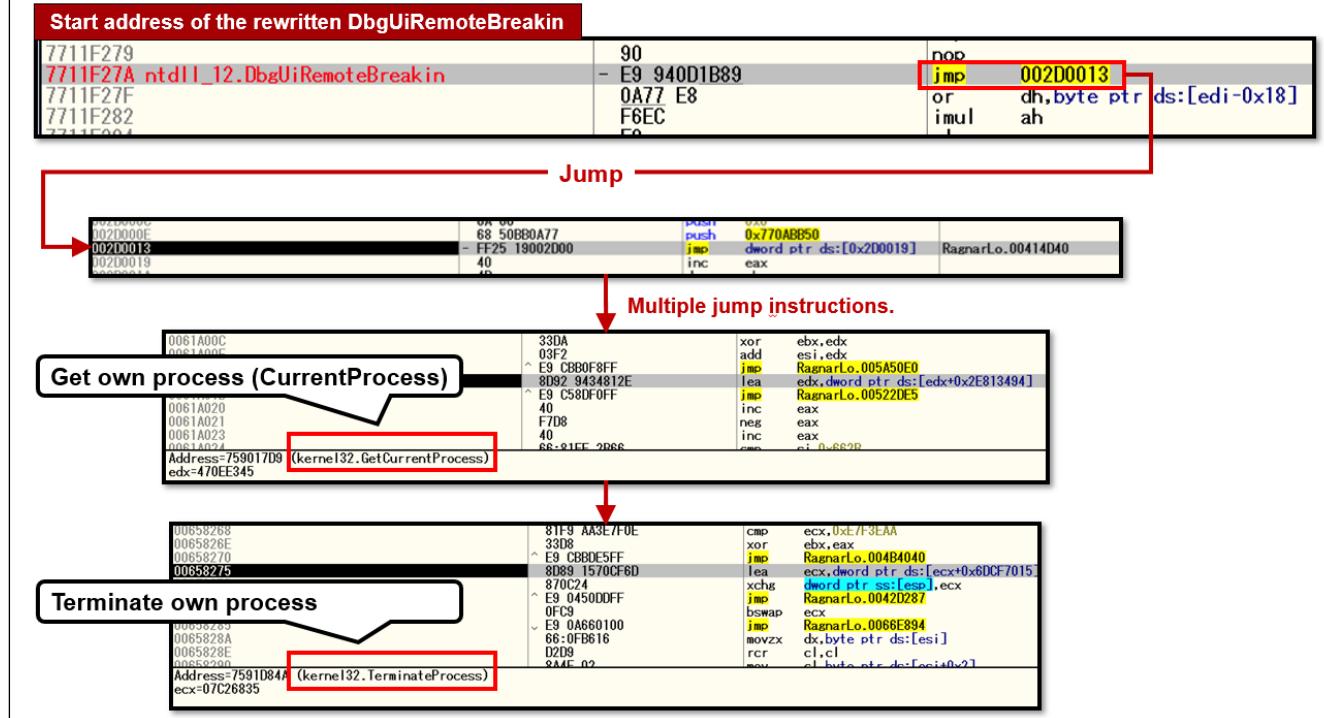


Fig. 9 - Process when DbgUiRemoteBreakin is called

Maze ransomware also focused on DbgUiRemoteBreakin function, but it only changes the first byte to the “return” instruction, so that DbgUiRemoteBreakin immediately returns to the main program, meaning that the DbgUiRemoteBreakin function eventually do nothing. On the other hand, Ragnar Locker tampers the first 5 bytes and ultimately force terminate own process, so I think Ragnar Locker technique is more complicated than Maze ransomware.

Likewise, NtProtectVirtualMemory function is hooked and it jumps to their own code.

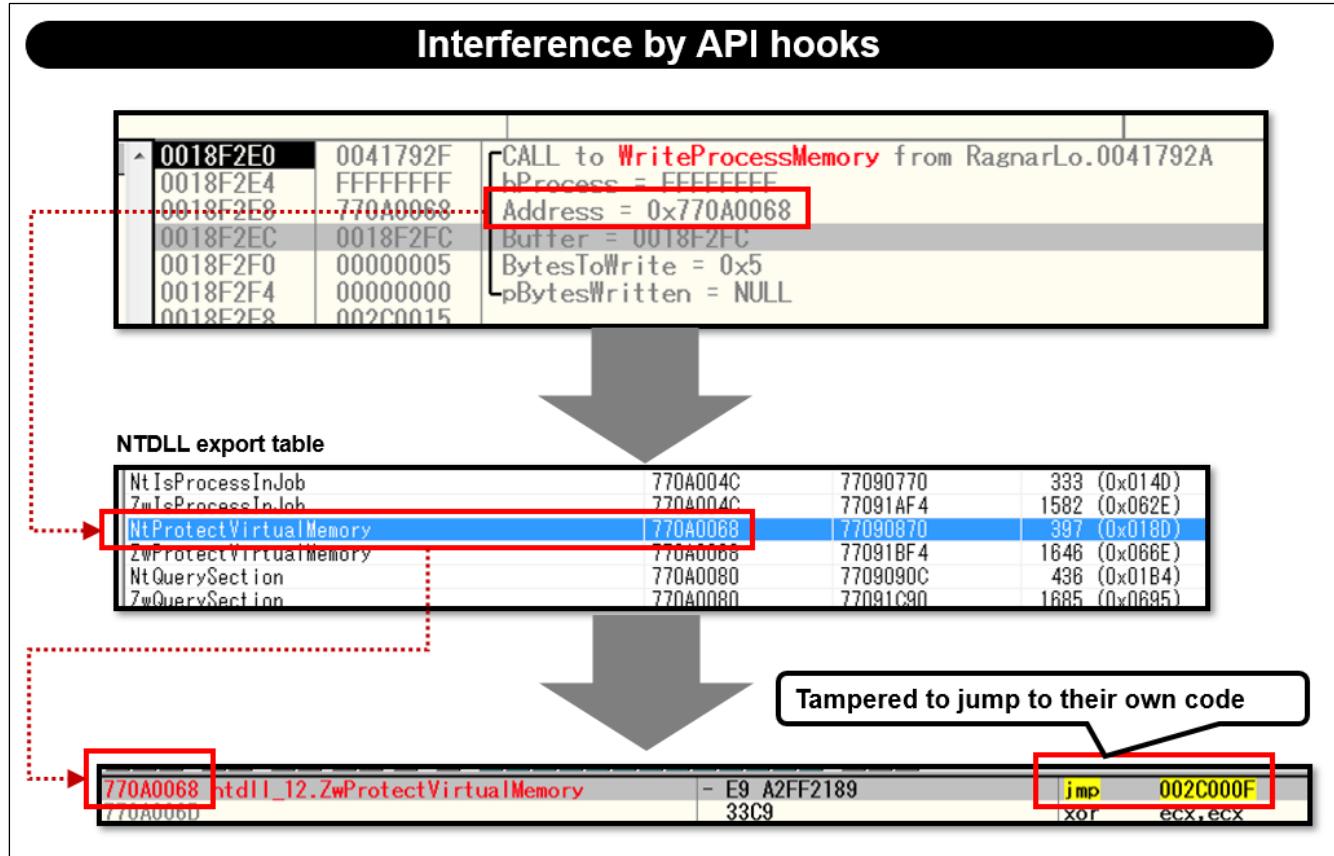


Fig. 10 - NtProtectVirtualMemory is also tampered

4. Checking language of infected PC

Ragnar Locker does not infect PCs that are identified Russian and other specific languages. Ragnar Locker obtains language information of the infected PC by using GetLocaleInfo, and if the result matches to one of the pre-defined languages, then it finally terminates its own process.

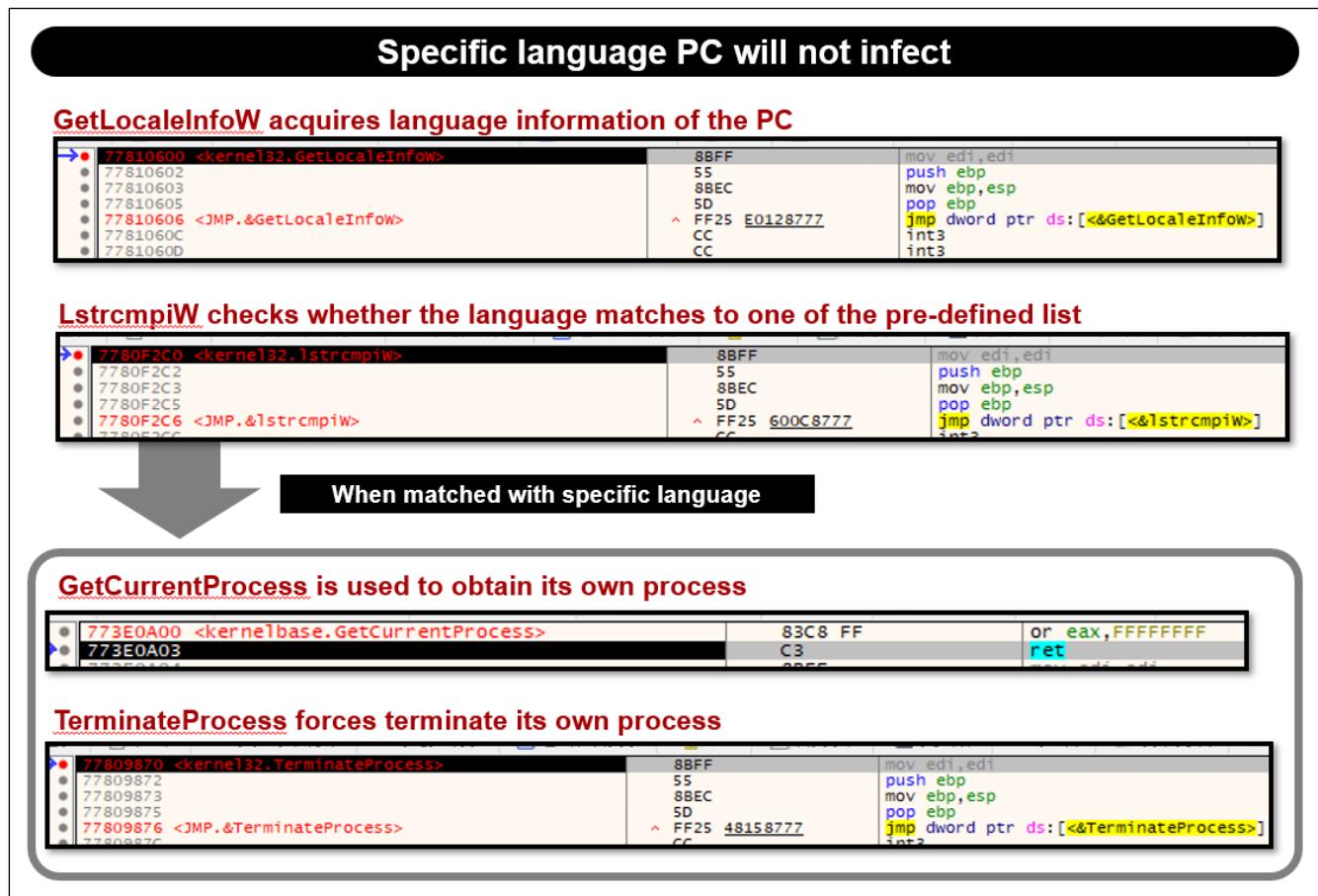


Fig. 11 - Process to check infected PC's language

Below is the list of languages that will NOT infect Ragnar Locker.

- | | |
|---|---|
| <ul style="list-style-type: none"> ● Azerbaijani ● Armenian ● Belorussian ● Kazakh ● Kyrgyz ● Moldavian | <ul style="list-style-type: none"> ● Tajik ● Russian ● Turkmen ● Uzbek ● Ukrainian ● Georgian |
|---|---|

5. Interfering with system recovery

Ragnar Locker will then take action that interferes with the recovery of the system.

First, disables “System Restore” by using Windows regular process, wmic, and runs a command to delete volume shadow copy.

Disabling System Restore by deleting shadow copy

0018CB50	004013B8	CALL to CreateProcessW ModuleFileName = NULL CommandLine = "wmic.exe shadowcopy delete" pProcessSecurity = NULL pThreadSecurity = NULL InheritHandles = FALSE CreationFlags = NORMAL_PRIORITY_CLASS pEnvironment = NULL CurrentDir = NULL pStartupInfo = 0018CBAC pProcessInfo = 0018CD58
0018CB54	00000000	
0018CB58	0018CD20	
0018CB5C	00000000	
0018CB60	00000000	
0018CB64	00000000	
0018CB68	00000020	
0018CB6C	00000000	
0018CB70	00000000	
0018CB74	0018CBAC	
0018CB78	0018CD58	
0018CB7C	00D192E8	

Fig. 12 - Deleting shadow copy

Then disables the ability to automatic repair when Windows fails startup, by passing the "recoveryenabled No" parameter to Bcdedit, which is a Windows' regular process.

Disable Windows auto repair function

0018CB50	004013FB	CALL to CreateProcessW ModuleFileName = NULL CommandLine = "bcdedit /set {default} recoveryenabled No" pProcessSecurity = NULL pThreadSecurity = NULL InheritHandles = FALSE CreationFlags = NORMAL_PRIORITY_CLASS pEnvironment = NULL CurrentDir = NULL pStartupInfo = 0018CBAC pProcessInfo = 0018CD58
0018CB54	00000000	
0018CB58	0018CCCC	
0018CB5C	00000000	
0018CB60	00000000	
0018CB64	00000000	
0018CB68	00000020	
0018CB6C	00000000	
0018CB70	00000000	
0018CB74	0018CBAC	
0018CB78	0018CD58	
0018CB7C	00D192E8	

Fig. 13 - Disabling Windows' automatic repair function (1)

Similarly, Bcdedit (Windows' regular process) to configure a normal startup without repairing Windows startup failures by passing a "bootstatuspolicy IgnoreAllFailures" parameter.

Disable Windows auto repair function			
0018CB50	0040143E	CALL to CreateProcessW from RagnarLo.00401439	
0018CB54	00000000	ModuleFileName = NULL	
0018CB58	0018CBF0	CommandLine = "bcdedit /set {default} bootstatuspolicy IgnoreAllFailures"	
0018CB5C	00000000	pProcessSecurity = NULL	
0018CB60	00000000	pThreadSecurity = NULL	
0018CB64	00000000	InheritHandles = FALSE	
0018CB68	00000020	CreationFlags = NORMAL_PRIORITY_CLASS	
0018CB6C	00000000	pEnvironment = NULL	
0018CB70	00000000	CurrentDir = NULL	
0018CB74	0018CBAC	pStartupInfo = 0018CBAC	
0018CB78	0018CD58	pProcessInfo = 0018CD58	
0018CB7C	00D192E8		
0018CB80	0018CB80		

Fig. 14 - Disabling Windows' automatic repair function (2)

In addition, by passing the "advancedoptions false" parameter to Bcdedit, the "Advanced Settings of Startup Options" menu is set to be inaccessible from the F8 key at Windows startup.

Disable Windows auto repair function			
0018CB50	00401481	CALL to CreateProcessW	
0018CB54	00000000	ModuleFileName = NULL	
0018CB58	0018CC64	CommandLine = "bcdedit /set {globalsettings} advancedoptions false"	
0018CB5C	00000000	pProcessSecurity = NULL	
0018CB60	00000000	pThreadSecurity = NULL	
0018CB64	00000000	InheritHandles = FALSE	
0018CB68	00000020	CreationFlags = NORMAL_PRIORITY_CLASS	
0018CB6C	00000000	pEnvironment = NULL	
0018CB70	00000000	CurrentDir = NULL	
0018CB74	0018CBAC	pStartupInfo = 0018CBAC	
0018CB78	0018CD58	pProcessInfo = 0018CD58	
0018CB7C	00D192E8		

Fig. 15 - Disabling Windows' automatic repair function (3)

As mentioned above, there are some activities that prevent the PC from being repaired when Windows start up. However, it seems Ragnar Locker does not destroy files that requires Windows start up. We guess that the attacker wanted to keep current Windows environment, rather than encrypt start up files and user recovers to unknown environment.

6. Stopping processes and services

Ragnar Locker then checks active processes to see if particular process is running. After Ragnar Locker acquires the list of processes running on the PC, it compares with the list that is hard-coded in the Ragnar Locker. As shown in the figure below, the list includes databases, document software, and mail software that need to be terminated, in order to encrypt files.

Check running processes																																																																																																																																																					
Obtain the list of processes that are running (using Createtoolhelp32snapshot/Process32First/ Process32Next), Compares (StrStr) with a hard-coded abort-list in its own binary.																																																																																																																																																					
<table border="1"> <thead> <tr> <th>Address</th><th>Hex dump</th><th>ASCII</th><th>0359F4BC</th><th>00402CD0</th><th>CALL to StrStrA from RagnarLo.00402CD2</th></tr> </thead> <tbody> <tr> <td>0359F4C4</td><td>D4 F4 59 03 00 00 00 00 00 00 00 00 DC FE 18 00</td><td>sal.mysql.veeam.</td><td>0359F4C0</td><td>0359FF00</td><td>String = smss.exe</td></tr> <tr> <td>0359F4D4</td><td>77 6E 69 6E 77 6F 72 64 2C 6E 65 74 6E 65 74 6E</td><td>oracle,ocssd,db\$</td><td>0359F4D4</td><td>0359F4D4</td><td>Pattern = sql</td></tr> <tr> <td>0359F4F4</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>nmp,syntime,agn</td><td>0359F4C8</td><td>00000000</td><td></td></tr> <tr> <td>0359F504</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>tsvc,isa1plusvc</td><td>0359F4CC</td><td>00000000</td><td></td></tr> <tr> <td>0359F514</td><td>2C 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F 70</td><td>.xssvcon,mydes</td><td>0359F4D0</td><td>0018FEDC</td><td></td></tr> <tr> <td>0359F524</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>ktopservice,oacau</td><td>0359F4D4</td><td>006C7173</td><td></td></tr> <tr> <td>0359F534</td><td>74 6F 75 70 64 73 2C 65 6E 63 73 6F 70 6E 65 73</td><td>toupds,encsvc,fi</td><td>0359F4D8</td><td>717379B3</td><td></td></tr> <tr> <td>0359F544</td><td>72 65 66 6F 78 2C 74 62 69 72 64 63 66 6F 78 2C</td><td>refox,tbirdconfi</td><td>0359F4DC</td><td>0006C7173</td><td></td></tr> <tr> <td>0359F554</td><td>67 2C 6D 79 64 65 73 6B 74 6F 70 71 6F 73 6E 67</td><td>g,mydesktopqos,o</td><td>0359F4E0</td><td>0006C7173</td><td></td></tr> <tr> <td>0359F564</td><td>63 6E 6D 6D 2C 64 62 65 6E 67 35 30 2C 72 71 6F</td><td>comm,dbeng50,sab</td><td>0359F4E4</td><td>0006C7173</td><td></td></tr> <tr> <td>0359F574</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>coreservice,exce</td><td>0359F4E8</td><td>0006C7173</td><td></td></tr> <tr> <td>0359F584</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>l,infopath,msacc</td><td>0359F4EC</td><td>64737363</td><td></td></tr> <tr> <td>0359F594</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>ess,msppub,onenot</td><td>0359F4F0</td><td>7362642C</td><td></td></tr> <tr> <td>0359F5A4</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>e,out look,powerp</td><td>0359F4F4</td><td>2C706D6E</td><td></td></tr> <tr> <td>0359F5B4</td><td>6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F</td><td>nt,steam,thebat,</td><td>0359F4F8</td><td>636E7973</td><td></td></tr> <tr> <td>0359F5C4</td><td>74 6F 75 70 64 73 2C 65 6E 63 73 6F 70 6E 65 73</td><td>thunderbird,visi</td><td>0359F4FC</td><td>856D6974</td><td></td></tr> <tr> <td>0359F5D4</td><td>6F 2C 77 69 6E 77 6F 72 64 2C 6E 65 74 6E 65 74</td><td>o,winword,wordpa</td><td>0359F500</td><td>6E67612C</td><td></td></tr> <tr> <td>0359F5E4</td><td>64 2C 45 64 75 4C 69 6E 6B 32 53 43 64 70 6E 65</td><td>d,EduLink2SIMS,b</td><td>0359F504</td><td>63767374</td><td></td></tr> <tr> <td>0359F5F4</td><td>65 6E 67 69 6E 65 2C 62 65 6E 65 74 6E 73 6C 67</td><td>engine,benetn,b</td><td>0359F508</td><td>7173692C</td><td></td></tr> <tr> <td>0359F604</td><td>65 73 65 72 76 65 72 2C 70 76 6C 73 76 72 2C 67</td><td>eserver,pvlsrv,b</td><td>0359F50C</td><td>756C706C</td><td></td></tr> <tr> <td>0359F614</td><td>65 72 65 6D 6F 74 65 2C 56 78 4C 6F 63 6B 64 67</td><td>eremote,VxLockdo</td><td>0359F510</td><td>63767373</td><td></td></tr> <tr> <td>0359F624</td><td>77 6E 53 65 72 76 65 72 2C 70 6F 73 74 67 72 65</td><td>wmServer,postgre</td><td>0359F514</td><td>7366782C</td><td></td></tr> <tr> <td>0359F634</td><td>73 2C 66 64 68 6F 73 74 2C 57 53 53 41 44 4D 4F</td><td>s,fdhost,WSSADMIN</td><td>0359F518</td><td>63637673</td><td></td></tr> </tbody> </table>						Address	Hex dump	ASCII	0359F4BC	00402CD0	CALL to StrStrA from RagnarLo.00402CD2	0359F4C4	D4 F4 59 03 00 00 00 00 00 00 00 00 DC FE 18 00	sal.mysql.veeam.	0359F4C0	0359FF00	String = smss.exe	0359F4D4	77 6E 69 6E 77 6F 72 64 2C 6E 65 74 6E 65 74 6E	oracle,ocssd,db\$	0359F4D4	0359F4D4	Pattern = sql	0359F4F4	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	nmp,syntime,agn	0359F4C8	00000000		0359F504	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	tsvc,isa1plusvc	0359F4CC	00000000		0359F514	2C 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F 70	.xssvcon,mydes	0359F4D0	0018FEDC		0359F524	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	ktopservice,oacau	0359F4D4	006C7173		0359F534	74 6F 75 70 64 73 2C 65 6E 63 73 6F 70 6E 65 73	toupds,encsvc,fi	0359F4D8	717379B3		0359F544	72 65 66 6F 78 2C 74 62 69 72 64 63 66 6F 78 2C	refox,tbirdconfi	0359F4DC	0006C7173		0359F554	67 2C 6D 79 64 65 73 6B 74 6F 70 71 6F 73 6E 67	g,mydesktopqos,o	0359F4E0	0006C7173		0359F564	63 6E 6D 6D 2C 64 62 65 6E 67 35 30 2C 72 71 6F	comm,dbeng50,sab	0359F4E4	0006C7173		0359F574	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	coreservice,exce	0359F4E8	0006C7173		0359F584	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	l,infopath,msacc	0359F4EC	64737363		0359F594	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	ess,msppub,onenot	0359F4F0	7362642C		0359F5A4	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	e,out look,powerp	0359F4F4	2C706D6E		0359F5B4	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	nt,steam,thebat,	0359F4F8	636E7973		0359F5C4	74 6F 75 70 64 73 2C 65 6E 63 73 6F 70 6E 65 73	thunderbird,visi	0359F4FC	856D6974		0359F5D4	6F 2C 77 69 6E 77 6F 72 64 2C 6E 65 74 6E 65 74	o,winword,wordpa	0359F500	6E67612C		0359F5E4	64 2C 45 64 75 4C 69 6E 6B 32 53 43 64 70 6E 65	d,EduLink2SIMS,b	0359F504	63767374		0359F5F4	65 6E 67 69 6E 65 2C 62 65 6E 65 74 6E 73 6C 67	engine,benetn,b	0359F508	7173692C		0359F604	65 73 65 72 76 65 72 2C 70 76 6C 73 76 72 2C 67	eserver,pvlsrv,b	0359F50C	756C706C		0359F614	65 72 65 6D 6F 74 65 2C 56 78 4C 6F 63 6B 64 67	eremote,VxLockdo	0359F510	63767373		0359F624	77 6E 53 65 72 76 65 72 2C 70 6F 73 74 67 72 65	wmServer,postgre	0359F514	7366782C		0359F634	73 2C 66 64 68 6F 73 74 2C 57 53 53 41 44 4D 4F	s,fdhost,WSSADMIN	0359F518	63637673	
Address	Hex dump	ASCII	0359F4BC	00402CD0	CALL to StrStrA from RagnarLo.00402CD2																																																																																																																																																
0359F4C4	D4 F4 59 03 00 00 00 00 00 00 00 00 DC FE 18 00	sal.mysql.veeam.	0359F4C0	0359FF00	String = smss.exe																																																																																																																																																
0359F4D4	77 6E 69 6E 77 6F 72 64 2C 6E 65 74 6E 65 74 6E	oracle,ocssd,db\$	0359F4D4	0359F4D4	Pattern = sql																																																																																																																																																
0359F4F4	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	nmp,syntime,agn	0359F4C8	00000000																																																																																																																																																	
0359F504	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	tsvc,isa1plusvc	0359F4CC	00000000																																																																																																																																																	
0359F514	2C 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F 70	.xssvcon,mydes	0359F4D0	0018FEDC																																																																																																																																																	
0359F524	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	ktopservice,oacau	0359F4D4	006C7173																																																																																																																																																	
0359F534	74 6F 75 70 64 73 2C 65 6E 63 73 6F 70 6E 65 73	toupds,encsvc,fi	0359F4D8	717379B3																																																																																																																																																	
0359F544	72 65 66 6F 78 2C 74 62 69 72 64 63 66 6F 78 2C	refox,tbirdconfi	0359F4DC	0006C7173																																																																																																																																																	
0359F554	67 2C 6D 79 64 65 73 6B 74 6F 70 71 6F 73 6E 67	g,mydesktopqos,o	0359F4E0	0006C7173																																																																																																																																																	
0359F564	63 6E 6D 6D 2C 64 62 65 6E 67 35 30 2C 72 71 6F	comm,dbeng50,sab	0359F4E4	0006C7173																																																																																																																																																	
0359F574	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	coreservice,exce	0359F4E8	0006C7173																																																																																																																																																	
0359F584	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	l,infopath,msacc	0359F4EC	64737363																																																																																																																																																	
0359F594	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	ess,msppub,onenot	0359F4F0	7362642C																																																																																																																																																	
0359F5A4	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	e,out look,powerp	0359F4F4	2C706D6E																																																																																																																																																	
0359F5B4	6F 70 6E 65 73 6F 70 6E 65 73 6F 70 6E 65 73 6F	nt,steam,thebat,	0359F4F8	636E7973																																																																																																																																																	
0359F5C4	74 6F 75 70 64 73 2C 65 6E 63 73 6F 70 6E 65 73	thunderbird,visi	0359F4FC	856D6974																																																																																																																																																	
0359F5D4	6F 2C 77 69 6E 77 6F 72 64 2C 6E 65 74 6E 65 74	o,winword,wordpa	0359F500	6E67612C																																																																																																																																																	
0359F5E4	64 2C 45 64 75 4C 69 6E 6B 32 53 43 64 70 6E 65	d,EduLink2SIMS,b	0359F504	63767374																																																																																																																																																	
0359F5F4	65 6E 67 69 6E 65 2C 62 65 6E 65 74 6E 73 6C 67	engine,benetn,b	0359F508	7173692C																																																																																																																																																	
0359F604	65 73 65 72 76 65 72 2C 70 76 6C 73 76 72 2C 67	eserver,pvlsrv,b	0359F50C	756C706C																																																																																																																																																	
0359F614	65 72 65 6D 6F 74 65 2C 56 78 4C 6F 63 6B 64 67	eremote,VxLockdo	0359F510	63767373																																																																																																																																																	
0359F624	77 6E 53 65 72 76 65 72 2C 70 6F 73 74 67 72 65	wmServer,postgre	0359F514	7366782C																																																																																																																																																	
0359F634	73 2C 66 64 68 6F 73 74 2C 57 53 53 41 44 4D 4F	s,fdhost,WSSADMIN	0359F518	63637673																																																																																																																																																	

Fig. 16 - Checking if a specific process is running

If it is determined that a matching process is running, the corresponding process is terminated forcibly (see the figure below). If a process keeps running, then the corresponding data files are locked by the process and Ragnar Locker will NOT be able to encrypt those data files, so they need to be terminated beforehand.

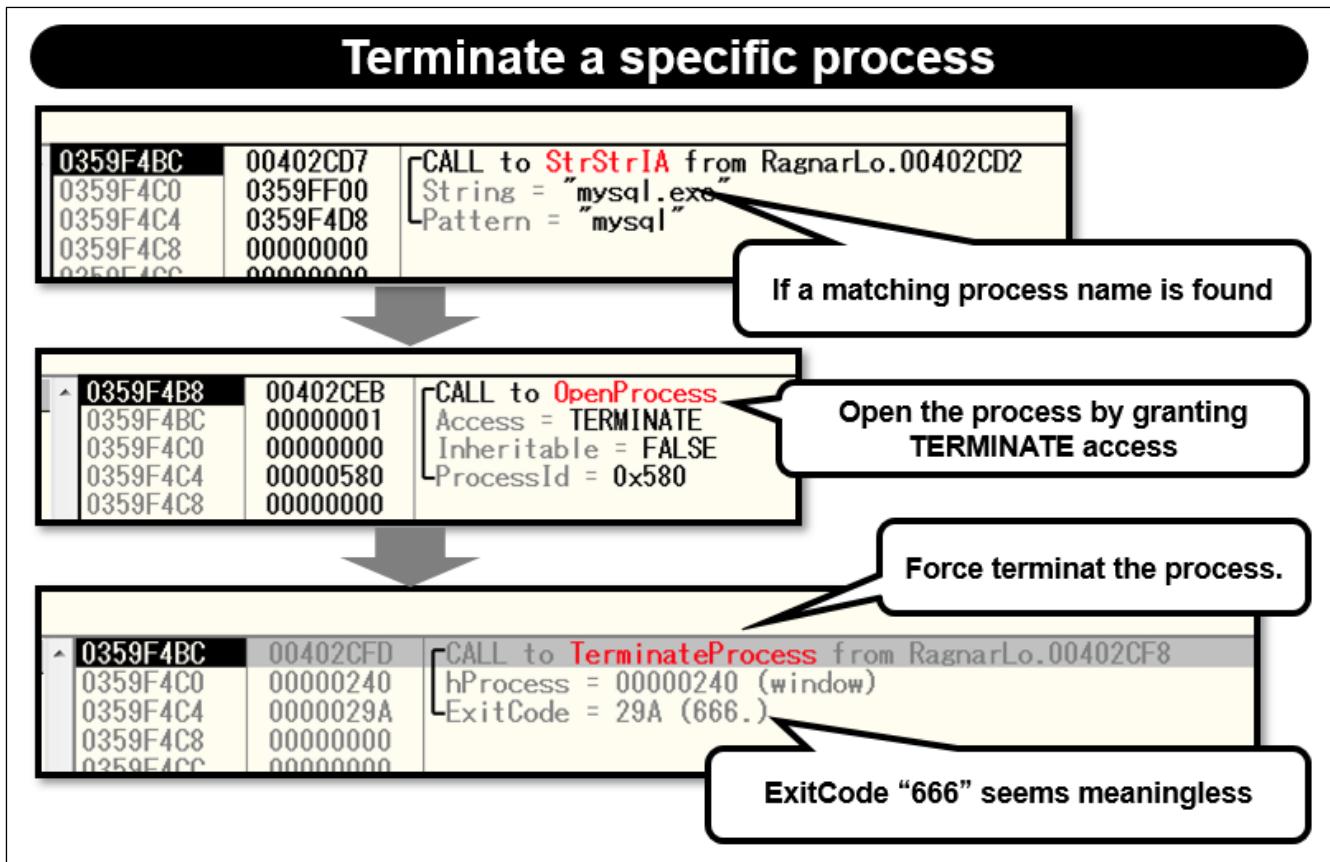


Fig.17 - A process is forcibly terminated if it is running

Following is the list of processes that Ragnar Locker terminates:

sql	msaccess	wsstracing
mysql	mspub	OWSTIMER
veeam	onenote	dfssvc.exe
oracle	outlook	dfsrs.exe
ocssd	powerpnt	swc_service.exe
dbsnmp	steam	sophos
synctime	thebat	SAVAdminService
agntsvc	thunderbird	SavService.exe
isqlplusussvc	visio	Hyper-v

xfssvccon	winword	TeamViewer
mydesktopservice	wordpad	Teamviwer
ocautoupds	EduLink2SIMS	Vmware
encsvc	bengine	vss
firefox	benetns	memtas
tbirdconfig	beserver	mepocs
mydesktopqos	pvlsvr	backup
ocomm	beremote	pulseway
dbeng50	VxLockdownServer	logme
sqbcoreservice	postgres	logmein
excel	fdhost	connectwise
infopath	WSSADMIN	splashtop
		wuauserv

It also forcibly stops specific services. If it is determined that specific services are running, the corresponding services are requested to stop (see the figure below). ControlService function is used and SERVICE_CONTROL_STOP is passed to request that the service to stop working.

Force stop specific services

The screenshot shows a debugger interface with several windows:

- Registers:** Shows CPU registers with values like 764D2910, 55, 8BEC, SD, FF25, 84C65076, and CC.
- Stack:** A dump of memory starting at address 0588FA80, showing values like 004030A4, 02E585E8, 00000001, 0588FF08, 00402D60, 00402D60, and 0019FEB.
- Callout:** A red box highlights the value 00000001 in the stack dump, which corresponds to the **SERVICE_CONTROL_STOP** parameter in the documentation below.
- Documentation Callout:**
 - SERVICE_CONTROL_STOP**: *0x00000001*
 - Notifies a service that it should stop. The *hService* handle must have the **SERVICE_STOP** access right.
 - After sending the stop request to a service, you should not send other controls to the service.
- Reference:** Microsoft document: <https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-controlservice>

Fig. 18 - Force stop when specific services are running

The actual list of services to stop is as follows:

vss	pulseway
sql	logme
memtas	logmein
mepocs	connectwise
sophos	splashtop
veeam	wuauserv
backup	

7. Encryption of files

Next, Ragnar Locker enters into the file encryption process, but one of the distinctive behaviors of Ragnar Locker is the following drive-mapping behavior:

Instead of simply enumerating the drives like other ransomware, Ragnar Locker forces to map volumes that are not mapped as local drives before encrypting them. This means that hidden volumes that are not mapped as drives are forced to the mapping state and encrypted. Following shows how drive mapping is performed.

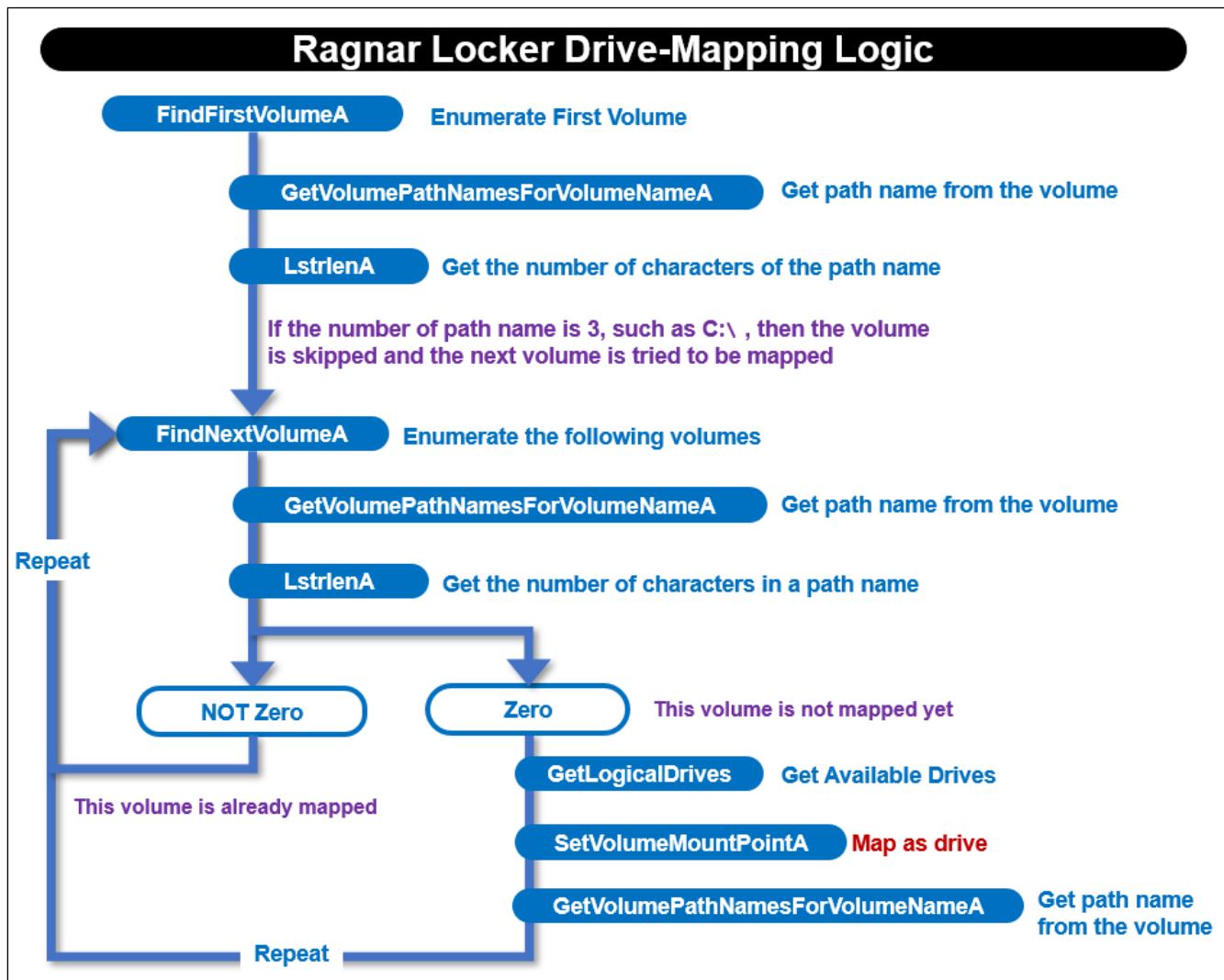


Fig. 19 - Drive-mapping logic of Ragnar Locker

Ragnar Locker encrypts all files that were found on the drives including newly mapped drives. However, following specific file names, extensions, and files in folders are excluded from the encryption.

Files and folders excluded from encryption

004022A8	MOV	DWORD	PTR	SS:[EBP-0x74],9416e5a5.0040A460	UNICODE	"autorun.inf"
004022AF	MOV	DWORD	PTR	SS:[EBP-0x70],9416e5a5.0040A478	UNICODE	"boot.ini"
004022B6	MOV	DWORD	PTR	SS:[EBP-0x6C],9416e5a5.0040A48C	UNICODE	"bootfont.bin"
004022BD	MOV	DWORD	PTR	SS:[EBP-0x68],9416e5a5.0040A4A8	UNICODE	"bootsect.bak"
004022C4	MOV	DWORD	PTR	SS:[EBP-0x64],9416e5a5.0040A4C4	UNICODE	"bootmgr"
004022CB	MOV	DWORD	PTR	SS:[EBP-0x60],9416e5a5.0040A4D4	UNICODE	"bootmgr.efi"
004022D2	MOV	DWORD	PTR	SS:[EBP-0x5C],9416e5a5.0040A4EC	UNICODE	"bootmgfw.efi"
004022D9	MOV	DWORD	PTR	SS:[EBP-0x58],9416e5a5.0040A508	UNICODE	"desktop.ini"
004022E0	MOV	DWORD	PTR	SS:[EBP-0x54],9416e5a5.0040A520	UNICODE	"iconcache.db"
004022E7	MOV	DWORD	PTR	SS:[EBP-0x50],9416e5a5.0040A53C	UNICODE	"ntldr"
004022EE	MOV	DWORD	PTR	SS:[EBP-0x4C],9416e5a5.0040A548	UNICODE	"ntuser.dat"
004022F5	MOV	DWORD	PTR	SS:[EBP-0x48],9416e5a5.0040A560	UNICODE	"ntuser.dat.log"
004022FC	MOV	DWORD	PTR	SS:[EBP-0x44],9416e5a5.0040A580	UNICODE	"ntuser.ini"
00402303	MOV	DWORD	PTR	SS:[EBP-0x40],9416e5a5.0040A598	UNICODE	"thumbs.db"
0040233C	MOV	DWORD	PTR	SS:[EBP-0x3C],9416e5a5.0040A5AC	UNICODE	".db"
00402343	MOV	DWORD	PTR	SS:[EBP-0x38],9416e5a5.0040A5B4	UNICODE	".sys"
0040234A	MOV	DWORD	PTR	SS:[EBP-0x34],9416e5a5.0040A5C0	UNICODE	".dll"
00402351	MOV	DWORD	PTR	SS:[EBP-0x30],9416e5a5.0040A5CC	UNICODE	".lnk"
00402358	MOV	DWORD	PTR	SS:[EBP-0x2C],9416e5a5.0040A5D8	UNICODE	".msi"
0040235F	MOV	DWORD	PTR	SS:[EBP-0x28],9416e5a5.0040A5E4	UNICODE	".drv"
00402366	MOV	DWORD	PTR	SS:[EBP-0x24],9416e5a5.0040A5F0	UNICODE	".exe"
00402A03	MOV	DWORD	PTR	SS:[EBP-0x38],9416e5a5.0040A328	UNICODE	"Windows"
00402A0A	MOV	DWORD	PTR	SS:[EBP-0x34],9416e5a5.0040A338	UNICODE	"Windows.old"
00402A11	MOV	DWORD	PTR	SS:[EBP-0x30],9416e5a5.0040A350	UNICODE	"Tor browser"
00402A18	MOV	DWORD	PTR	SS:[EBP-0x2C],9416e5a5.0040A368	UNICODE	"Internet Explorer"
00402A1F	MOV	DWORD	PTR	SS:[EBP-0x28],9416e5a5.0040A38C	UNICODE	"Google"
00402A26	MOV	DWORD	PTR	SS:[EBP-0x24],9416e5a5.0040A39C	UNICODE	"Opera"
00402A2D	MOV	DWORD	PTR	SS:[EBP-0x20],9416e5a5.0040A3A8	UNICODE	"Opera Software"
00402A34	MOV	DWORD	PTR	SS:[EBP-0x1C],9416e5a5.0040A3C8	UNICODE	"Mozilla"
00402A3B	MOV	DWORD	PTR	SS:[EBP-0x18],9416e5a5.0040A3D8	UNICODE	"Mozilla Firefox"
00402A42	MOV	DWORD	PTR	SS:[EBP-0x14],9416e5a5.0040A3F8	UNICODE	"\$Recycle.Bin"
00402A49	MOV	DWORD	PTR	SS:[EBP-0x10],9416e5a5.0040A414	UNICODE	"ProgramData"
00402A50	MOV	DWORD	PTR	SS:[EBP-0xC],9416e5a5.0040A42C	UNICODE	"All Users"
00402A57	MOV	DWORD	PTR	SS:[EBP-0x8],9416e5a5.0040A440	UNICODE	"Sysvol"
00402A5E	MOV	DWORD	PTR	SS:[EBP-0x4],9416e5a5.0040A450	UNICODE	"McAfee"

Fig. 20 - List of Ragnar Locker excluded from encryption

Ragnar Locker file encryption process is shown below. As you can see, it uses ReadFile and WriteFile, which is the same as another ransomware.

File manipulation logic for Ragnar Locker file encryption

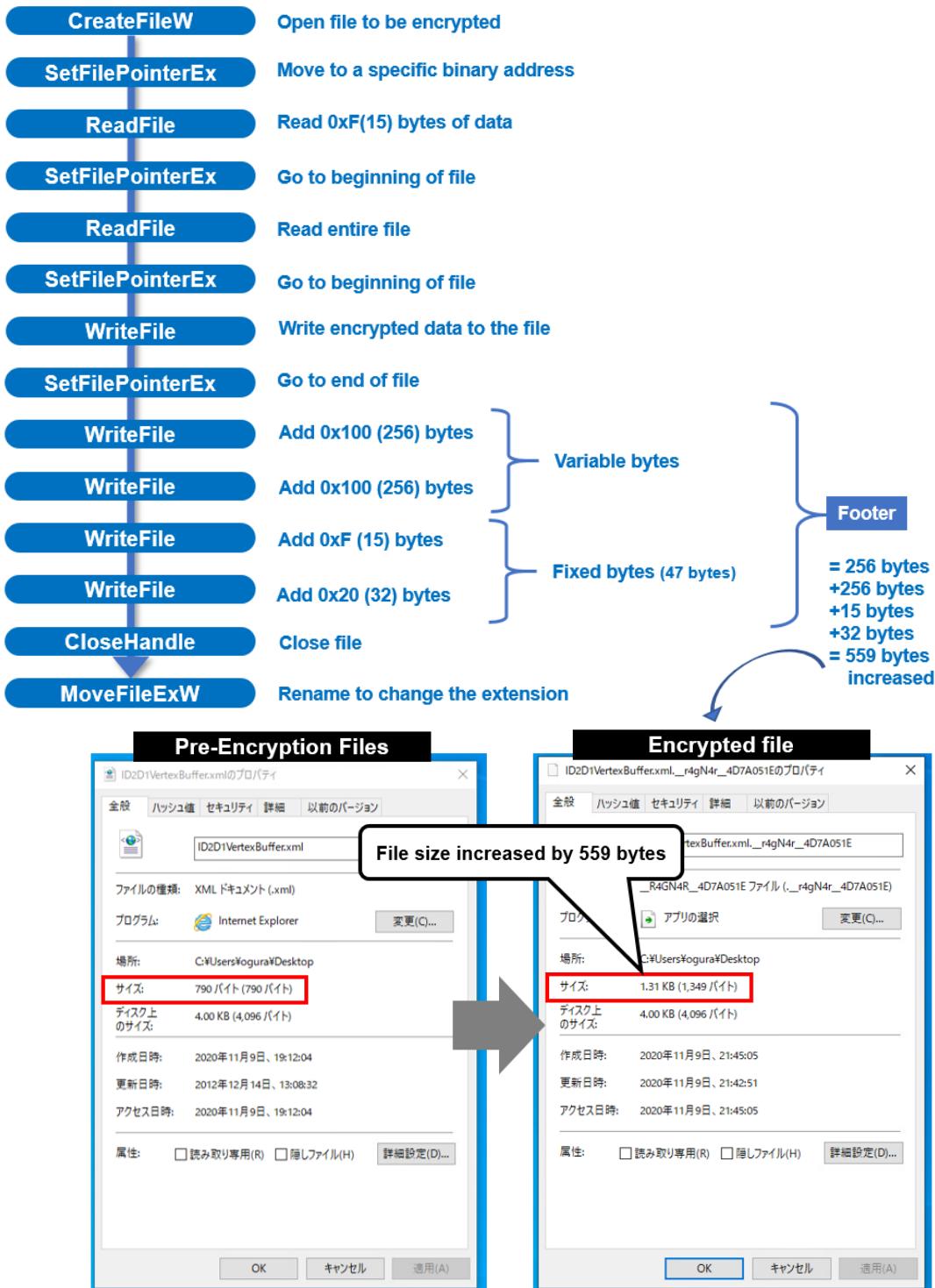


Fig. 21 - File manipulation logic during encryption

The last 47 bytes of the data is shown below. Since all files encrypted by Ragnar Locker have the same footer at the end of the 47 bytes, checking this area allows you to determine if they are Ragnar Locker encrypted files.

The last 47 bytes of the footer of the encrypted file	
000940 9D F6 7E 4A 54 14 0C E2-A1 06 4F 8D 9A 64 AB 75 000950 F6 D2 A9 FA 12 E3 53 B3-4F 75 86 21 40 23 5F AE 000960 61 67 6E 61 AE 5F 23 40-21 5F 46 42 35 64 44 41 000970 45 43 36 46 36 33 61 41-36 63 64 35 44 35 32 42 000980 38 38 32 32 65 32 45 62-30 32 8822eEb02	.. JT.....0..d.uS.Ou.!#H.. agn...#0_LFB5dDA EC6F63aA6cd5052B 8822eEb02
Encrypted file A	Encrypted file B

Fig. 22 - Common Footer, 47 bytes of an encrypted file

Encrypted file name is changed by moving the file using MoveFileEx as follows:

Changing encrypted file extension

Fig. 23 - Changing encrypted file extension

This gives the encrypted file the following extensions:

Extension given to the encrypted file

Fig. 24 - Extensions given to the encrypted files

The following RSA public key is used as part of the encryption process:

Public key used for encryption

```
'-----BEGIN PUBLIC KEY-----',0Ah
; DATA XREF: .data:off_40C018t0
'MIIBIjANBggkqhkiG9w0BAQEFAOCAQ8AMIIIBCgKCAQEA1BCMqlGD1Mlq/Qtya4h0'
'+SN1BvJS06/9ubWM7/S8rofNFGsj/ZGTovnLRDXKn7A2qwIEtPTpM8tih09HuZT8'
'wsu/nenL1rjvOPuMtDGoguNc8vpbeQEUDYneKUEdLq1fke6rD0Eq545eUDpgUQ7m'
'Ts5v1AK5a/ySGCe1r2k2hH10kYdEb7Rg5vGP1uC5tRLPPE/MOFm/d+xs1Sed5FWc'
'BJSVIXiYlzdifBDc58nBDFXARmJX2xeqohJwoSW3j0asRKYuuuznyQW/zTzyAh3GR'
'ZUpnGBVc33U06AgysBmj3nMGdbBTG12mBfMxXjEotrjSMjV0g97i9Q9hI/TZdM90'
'uQIDAQAB',0Ah
'-----END PUBLIC KEY-----',0Ah,0
```

Fig. 25 - Public key used for encryption

8. Creating and displaying threatening letter

Ragnar Locker creates a threatening letter in all folders as a text file. The file names always start from "!!!_READ_ME" (see illustration below).

Create threatening sentences in all folders			
000001F8 Timer 3. 00100002	00000204 File 2. 0012019F	Size 5125. (00001405) bytes	c:\Users\Public\Documents\!!!_READ_ME_EAF8DD6B_!!!.txt

Fig. 26 - Creating a text file with threats in all folders

Some sentences are hard-coded in Ragnar Locker binary file to be used for the threatening letter.

Threatening letter is hard-coded			
J040DFB0 ASCII " - "	J040E100 ASCII "ible.DU NUI Us"	J040E1E0 ASCII "e any third-part"	
J040DFC0 ASCII "by R A G N "	J040E1F0 ASCII "y or public Decr"	J040E200 ASCII "option software,	
J040DFD0 ASCII "A R L O C"	J040E200 ASCII "it also may DAM"	J040E220 ASCII "AGE files.DO N"	
J040DFE0 ASCII " - K E R !	J040E230 ASCII "OT Shutdown or R"	J040E240 ASCII "eset your system"	
J040DFF0 ASCII "*****	J040E250 ASCII " , it can DAMAGE"	J040E260 ASCII "files-----"	
J040E000 ASCII "*****	J040E270 ASCII "-----"	J040E280 ASCII "-----"	
J040E010 ASCII "*****	J040E290 ASCII "There is ONLY ON"	J040E2A0 ASCII "E possible way t"	
J040E020 ASCII "*****	J040E2B0 ASCII "o get back your"	J040E2C0 ASCII "files - contact"	
J040E030 ASCII "*****	J040E2D0 ASCII "us via LIVE CHAT"	J040E2E0 ASCII "and pay for the"	
J040E040 ASCII "*****	J040E2F0 ASCII "special DECRYPT"	J040E300 ASCII "ION KEY !For y"	
J040E050 ASCII "*****	J040E310 ASCII "our GUARANTEE we"	J040E320 ASCII "will decrypt 2"	
J040E060 ASCII " - "	J040E330 ASCII "of your files FO"	J040E340 ASCII "R FREE, to show"	
J040E070 ASCII " *YOU HAVE TO"	J040E350 ASCII "that it Works."	J040E360 ASCII "Don't waste yo"	
J040E080 ASCII " CONTACT US via	J040E370 ASCII "ur TIME, the lin"	J040E380 ASCII "k for contact us"	
J040E090 ASCII "LIVE CHAT IMMEDI"	J040E390 ASCII " will be deleted"	J040E3A0 ASCII " if there is no"	
J040E0A0 ASCII "ATELY TO RESOLVE	J040E3B0 ASCII "contact made in"	J040E3C0 ASCII "closest time and"	
J040E0B0 ASCII " THIS CASE AND M"	J040E3D0 ASCII " you will NEVER"	J040E3E0 ASCII " restore your DAT"	
J040E0C0 ASCII "AKE A DEAL*	J040E3F0 ASCII "A!!!! HOMEVER"		
J040E0D0 ASCII " - "			
J040E0E0 ASCII " (contact i"			
J040E0F0 ASCII "nformation you w"			
J040E100 ASCII "ill find at the"			
J040E110 ASCII "bottom of this n"			
J040E120 ASCII "otes)			
J040E130 ASCII " - "			
J040E140 ASCII " - "			
J040E150 ASCII "!!!! WAR"			
J040E160 ASCII "NING !!!!!DO"			
J040E170 ASCII "NOT Modify, ren"			
J040E180 ASCII "ame, copy or mov"			
J040E190 ASCII "e any files or y"			
J040E1A0 ASCII "ou can DAMAGE th"			
J040E1B0 ASCII "em and decryptio"			
J040E1C0 ASCII "n will be imposs"			

Fig. 27 - Threatening letter strings visible in memory

Ragnar Locker writes threatening letters from memory to text files by using WriteFile (see illustration below).

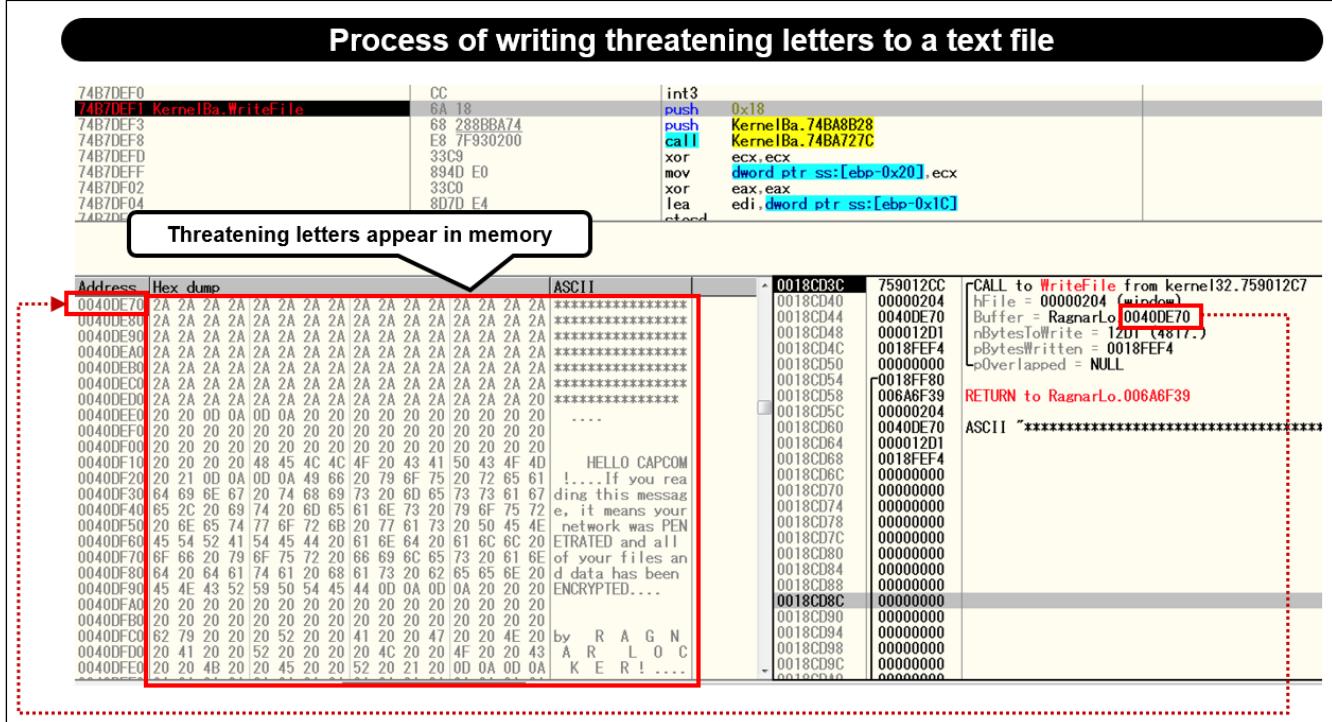


Fig. 28 - Writing Threatening letters from memory to text files

Threatening letter consists from 2 parts: fixed sentences and dynamically generated sentences, each written in two sessions. The following is the second write session that writes encrypted sentences encoded with a Base64 that was dynamically generated.

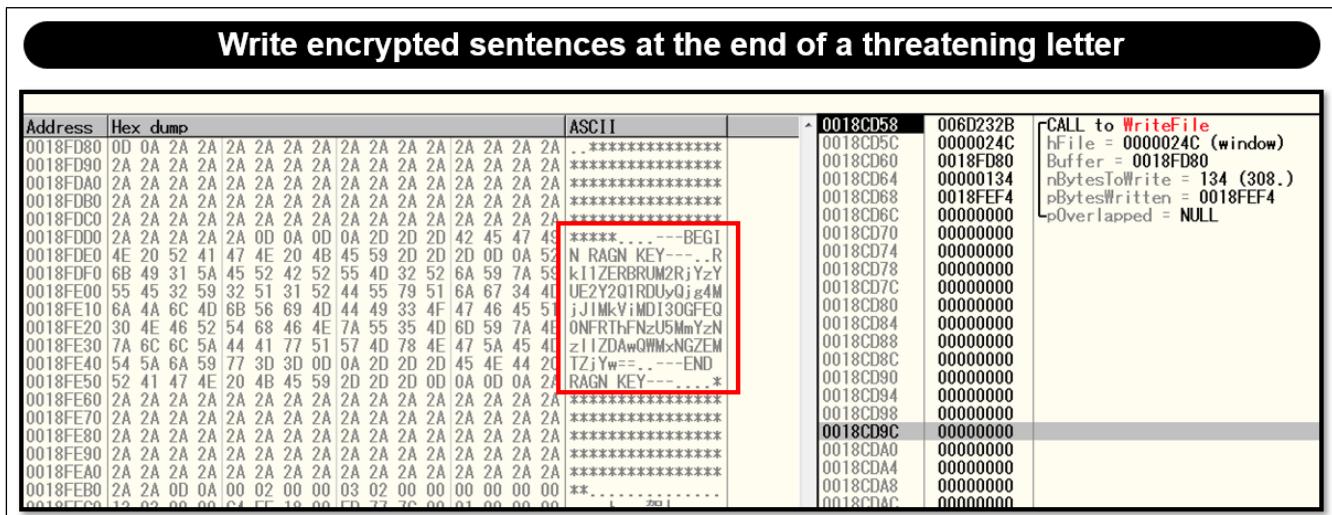


Fig. 29 - Writing encrypted sentences at the end of a threatening letter

Note that Ragnar Locker provides temporary and private leak page that is restricted in access separately from the leak sites to be disclosed to the public, and private leak page is presented only to the target (if the target did not pay the ransom money, the stolen information will be released to the public leak sites). The URL and password of the private leak page is only presented to the affected PC, and those URL and password is also hard-coded in Ragnar Locker binaries (see figure below).

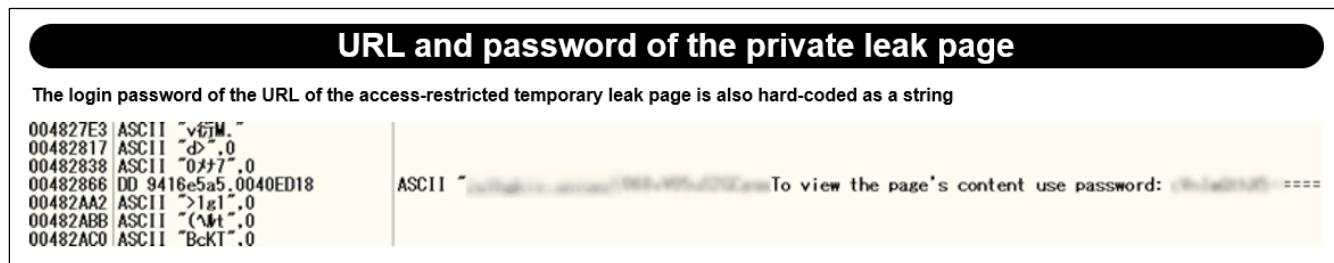


Fig. 30 - Temporary leak page URL and password

When all encryption is completed, Ragnar Locker uses log-in user's privilege and start notepad.exe, and then displays the threatening letter created in the Document folder, by using CreateProcessAsUserW function in KernelBase.dll.

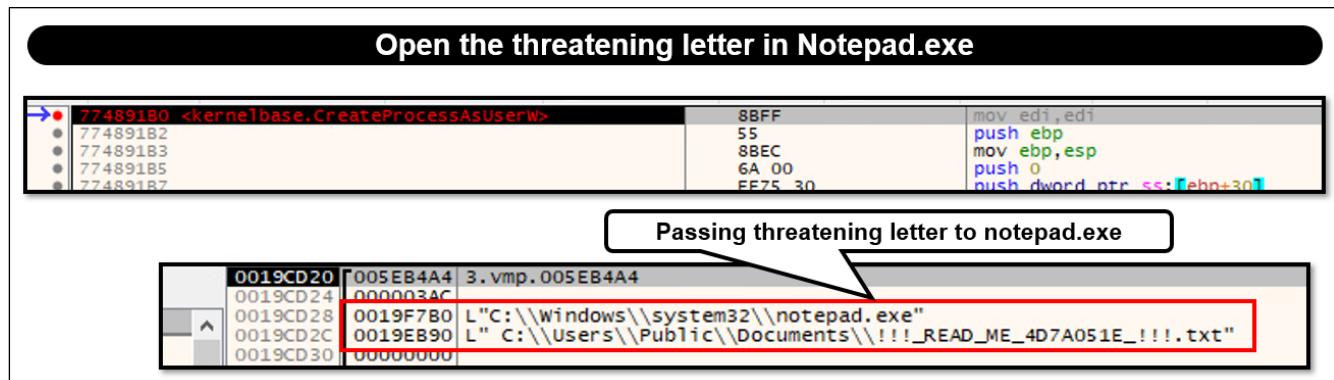


Fig. 31 - Opening the threatening letter in notepad.exe

The following shows the threatening letter on the infected PC. As you can see from the text beginning with Hello <company name>, the attacker clearly targeted the company and sent the ransomware binaries individually to the target company

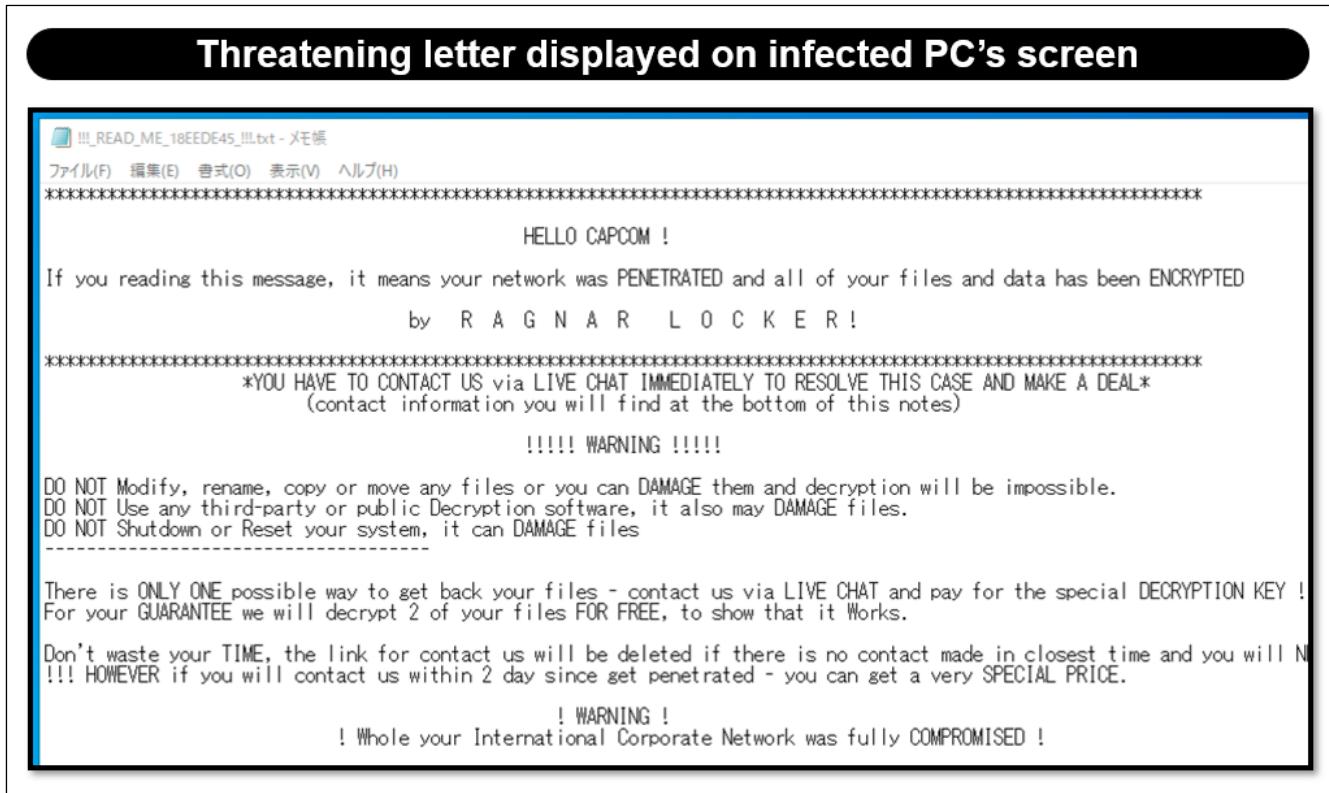


Fig. 32 - Threatening letter displayed after all encryption is completed

As shown above in Fig. 32, the threatening letter begins with the word "HELLO CAPCOM", the attacker customized the Ragnar Locker before they send. In addition to this specimen, we confirmed other Ragnar Locker also customized individually to each target company, in the format HELLO <company name>.

9. Supplemental information - Checking behavior of execution arguments

This article analyzed Ragnar Locker without adding arguments when executed. However, Ragnar Locker executable file has been designed to recognize the following executable arguments.

- Backup
- List
- Force
- Vm
- Vmback
- Share_network

In addition, when these execution arguments are given, there are minor differences such as the behavior of only process kill without file encryption and threatening statement creation, and the same behavior as when no execution argument is given. However, the major portion of behavior is the same as described above.

Other Ragnar Lockers discovered so far have been distributed together with the VM image at the time of deployment after intrusion, and attack methods that exploit the virtual environment have also been confirmed. Arguments such as -vm and -vmback have been confirmed in those methods. However, in this analysis, the attack method of the invasion route is unknown because I focused specimen analysis, so it is not clear how it was actually executed.

10. Ragnar Locker leak site

Like other targeted ransomware, Ragnar Locker has the following public leak sites, and the leak sites contain a large number of targeted companies.

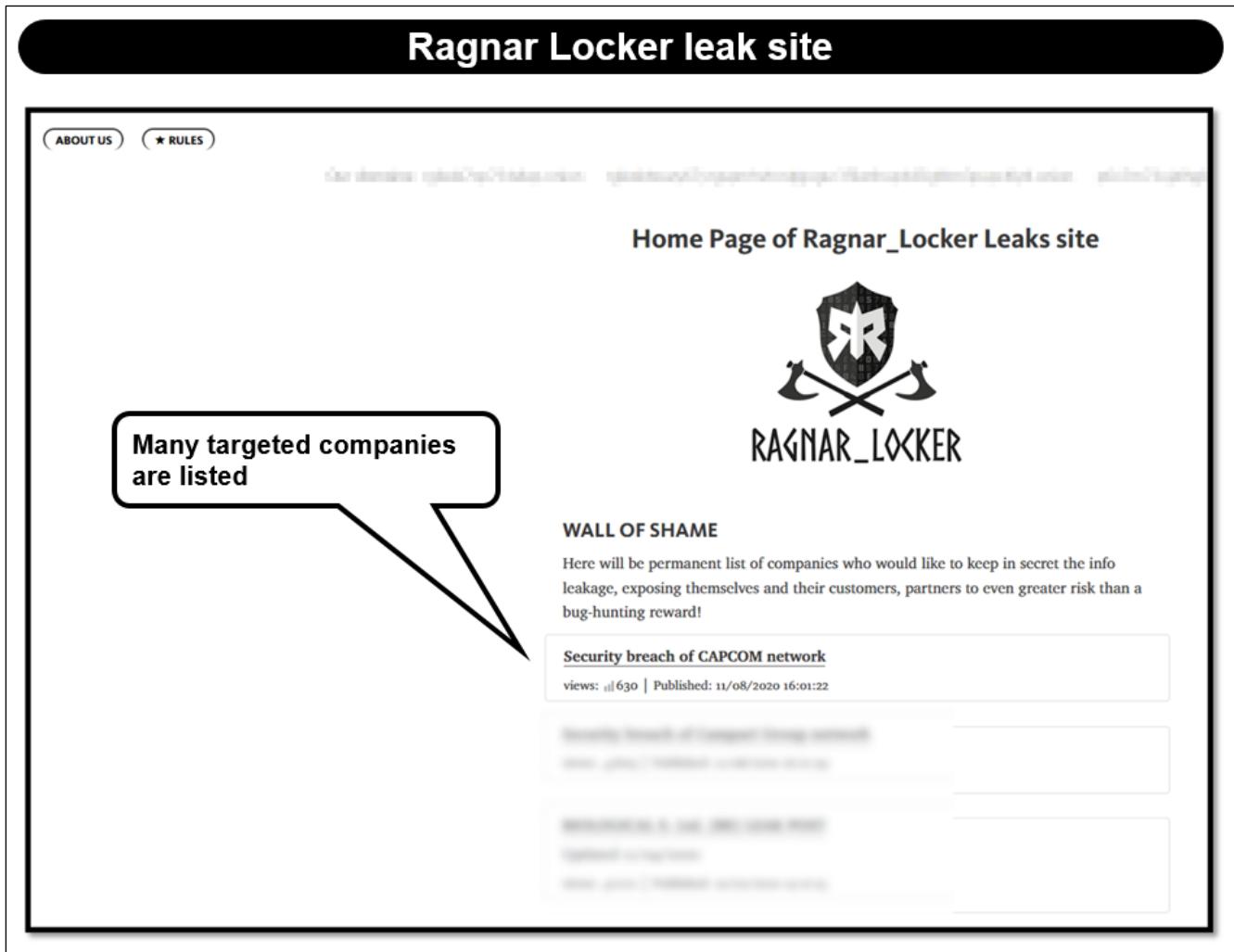


Fig. 33 - Ragnar Locker leakage site

Large amount of data that were stolen by the Ragnar Locker are available to download.

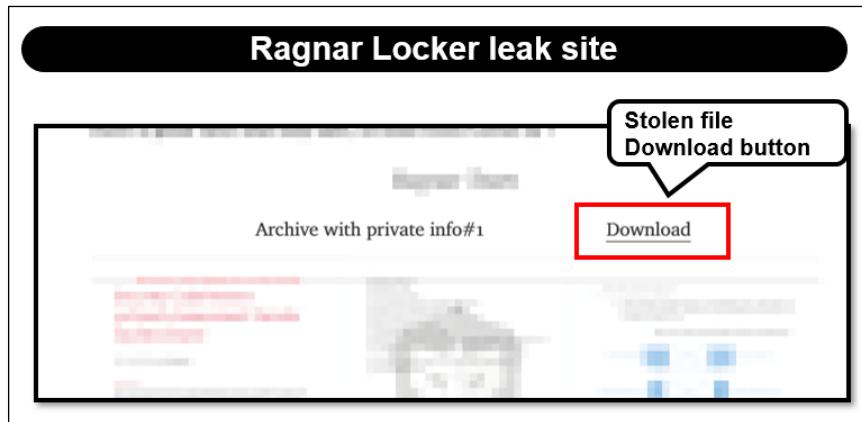


Fig. 34 - Stolen data from the targeted companies can be downloaded

The leak site contains an attack statement on CAPCOM, but at the time of this analysis, it is only a warning message, and no data has become public on the leak site. However, a screenshot is presented on the leak site as an evidence.

* Updated on Nov. 11, 2020: A portion of the stolen data was confirmed disclosed on the site.

Ragnar Locker leak site

Security breach of CAPCOM network

Ragnar_Locker Team Press Release

Dear CAPCOM executives, you can consider this message as a final warning.

As it already known there was a successful attack on CAPCOM's network. You can find official press release [here](#). Before the servers was encrypted we did download a high volume of sensitive data, approximately 1TB. Stolen data includes: personal information of clients and employee's, business information, a lot of confidential files and Non-Disclosure Agreements.

If the executives of CAPCOM is really cares about their confidentiality and security of partners and clients, they should contact us to negotiate conditions of the deal with us.

We gave the CAPCOM time until Tuesday 10th of November 18:00 EST time. If they will not contact us, than everyone will knows about such an negligence and irresponsibility from their side.

CAPCOM still have a chance to resole this issue beneficially and avoid the data leakage.



Fig. 35 - CAPCOM related message on Ragnar Locker leak site

11. Summary

As mentioned in the analysis article of SNAKE (EKANS) ransomware, there is a trend that targeted ransomware in recent years is sent after narrowing down the target to the attack target and customize it in advance. Therefore, even with the same type of ransomware, there is a high possibility that the behavior will be different depending on the attacked target. Therefore, it is going to be difficult to clearly and uniformly describe the behavior of the specimen. Detailed analysis of each specimen is essential to understand the behavior of the specimen itself.

12. About us

MBSD (Mitsui Bussan Secure Directions, Inc.) is the Japanese leading security company in managed security services, vulnerability assessment and testing, GRC (Governance, Risk, Compliance) consulting, incident response and handling, digital forensics, and secure programming training services. The MBSD services are provided by its personnel including the leading security experts in the field of secure programming, application security, penetration testing and threat analysis who have in-depth knowledge and understanding of attackers' methodologies. MBSD is working for the Internet infrastructure companies, cyber commerce and media giants, financial institutes, global enterprise, and government agencies in Japan to support their strategies against rapidly increasing threats from cyber space.

Company Contact Information:

Mitsui Bussan Secure Directions, Inc.

Yusen Suitengumae Building 6F, 1-14-8, Nihonbashi Ningyo Cho, Chuo-ku, Tokyo, 103-0013,
Japan

+81 3 5649 1961 | <http://www.mbsd.jp/>