



Eva0x00

Ransomware

CONTENTS

Eva0x00 Ransomware and What You Need to Know	3
What is Eva0x00 Ransomware?	3
Infection Chain	4
Eva0x00 Ransomware Overview	5
Static Analysis	7
eva.exe Analysis	7
IOCs	14
HASHs:	14
YARA RULE	15
MITRE ATT&CK TABLE	16
MITIGATIONS	17

Eva0x00 Ransomware and What You Need to Know

What is Eva0x00 Ransomware?

Eva0x00 Ransomware is a data encryptor known for its low virus detection rate. By default, it encrypts file types such as ".pdf," ".png," ".jpg," ".jpeg," ".mp3," ".mp4," ".txt," and ".exe," but it also allows users to optionally add different file types.

Eva0x00 Ransomware encrypts files from within, which means that even if the file extension is restored to normal, the content remains encrypted, rendering the victim's data inaccessible.

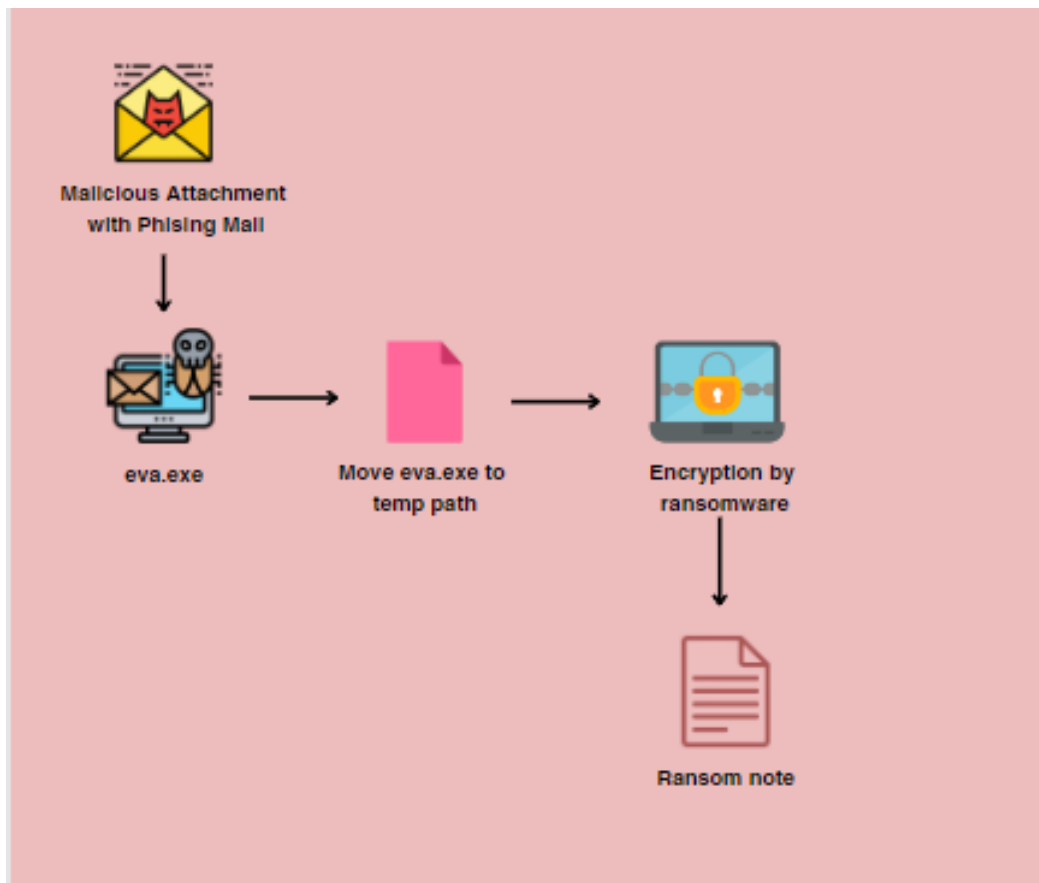
The ransomware encrypts data with the default format ".encrypt_by_eva0x00" but users have the option to change this extension according to their preference. Since the malware encrypts data in parallel, shortly after the malware is executed, all data for the specified file types is encrypted.

After encrypting all data, **Eva0x00 Ransomware** creates a readable file on the victim's machine. This file can be customized by the ransomware user to serve their purpose.

It's important to note that **Eva0x00 Ransomware** does not run as an autorun process, is not present in the Registry Editor (Regedit) section, and does not restart itself. Therefore, if the infected system is shut down and restarted after the malware has executed, the encryption process does not continue.

Eva0x00 Ransomware has a notably **low detection rate by antivirus software**, making it challenging to detect and mitigate.

Infection Chain



Eva0x00 Ransomware Overview

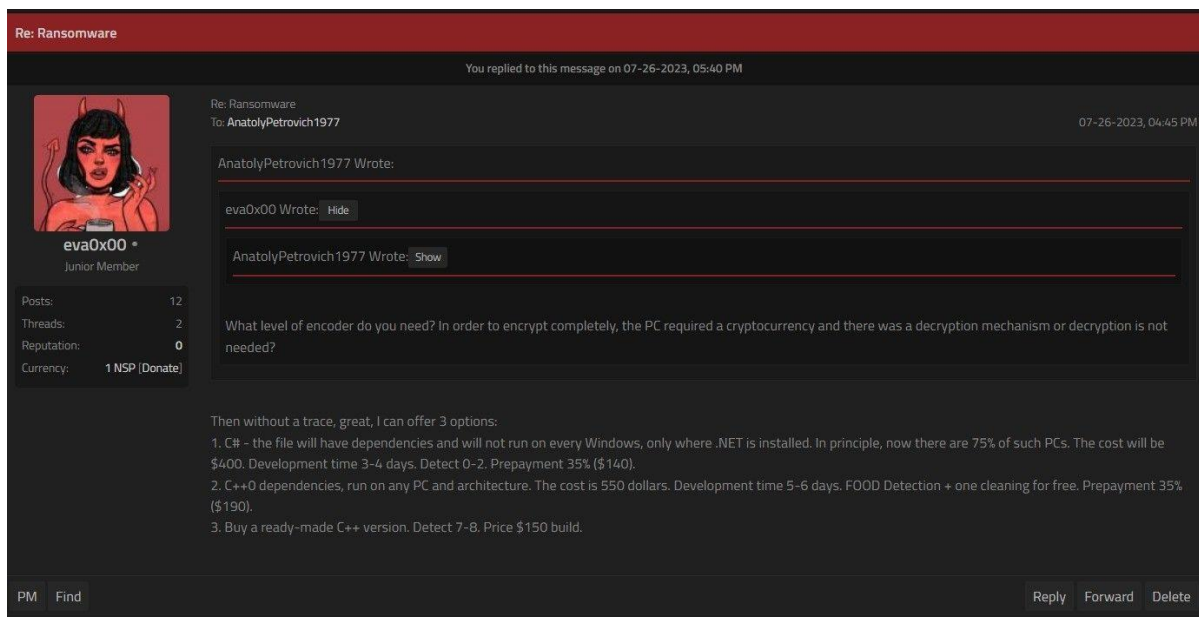


Figure 1- Dark Web forum info

Eva0x00 is a user on a dark web forum. He is selling custom made ransomware virus to the forum users. The ransomware does not require an internet connection. It is an offline ransomware and does not need any network connectivity. It's able to bypass many antivirus software used by computer users.

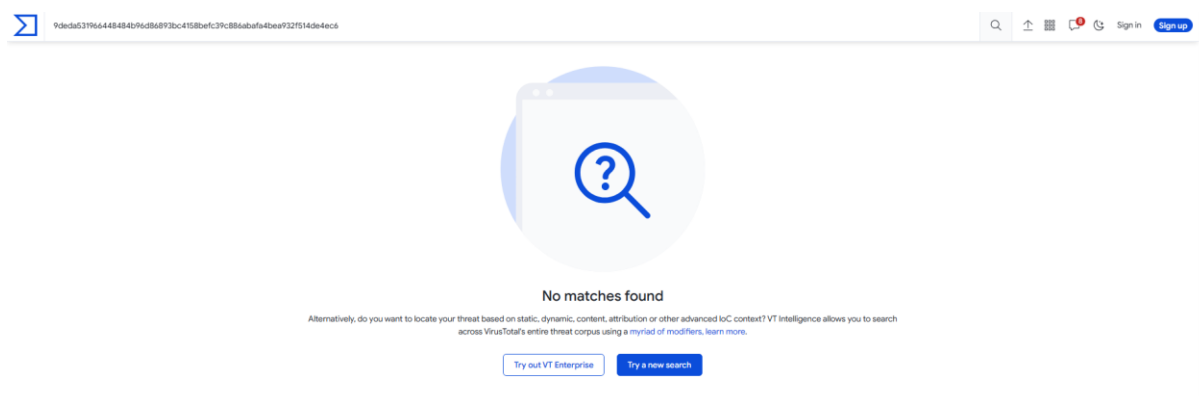


Figure 2- Virus Total info

The hash information of the Eva0x00 Ransomware **does not appear in the VirusTotal results**. This situation indicates that the software hasn't been scanned by anyone. For viruses, this is often observed in newly released products.

Eva0x00 RANSOMWARE

Scan result:	This file was detected by [4 / 40] engine(s)
File name:	rransom.exe
File size:	216064 bytes
Analysis date:	2023-09-14 20:51:20
CRC32:	a59afdb1
MD5:	45a357276700238ab1ea952f8ae8ab1b
SHA-1:	c5afe7a1585f00829d2793b649447b98aa857844
SHA-2:	9deda531966448484b96d86893bc4158befc39c886abafa4bea932f514de4ec6
SSDEEP:	3072:bhtYy2U9Hj+lieT131k7M1R71v7X3uzrVi1ELE:lrCTT13ee51v7X+s1E

Figure 3 - Scan Result

Eva0x00 Ransomware has **4/40 detection rate** on Kleenscan. It bypasses mostly used antivirus programs.

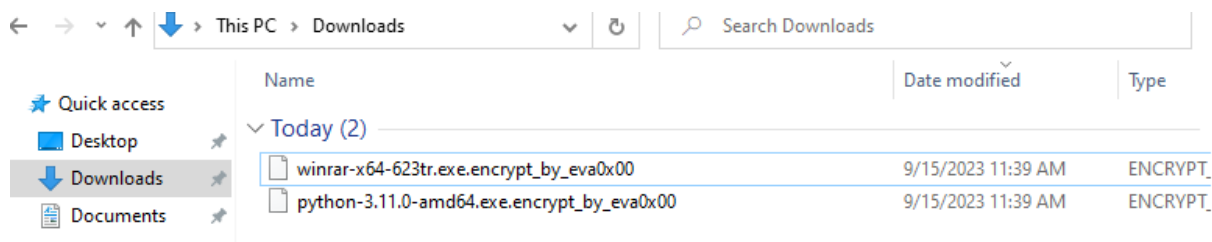


Figure 4- Encrypted file extension

Eva0x00 Ransomware encrypts the victim data with the extension of **“encrypt_by_eva0x00”**. This extension can be changed by the user.

```
READ_me - Notepad
File Edit Format View Help
Hi it's Eva0x00, you just got hacked .
The harddisks of your computer have been encrypted with an Military grade encryption algorithm.
There is no way to restore your data without a special key.
Only we can decrypt your files!
To purchase your key and restore your data, please follow these three easy steps:
1. Email the file called TheKey.txt at C:/Users/Your_name/EMAIL_ME.txt to eva0x00@protonmail.com
2. You will recieve your personal BTC address for payment.
Once payment has been completed, send another email to eva0x00@protonmail.com stating 'PAID'.
We will check to see if payment has been paid.
3. You will receive a text file with your KEY that will unlock all your files.
IMPORTANT: To decrypt your files, place text file on desktop and wait. Shortly after it will begin to decrypt all files.
WARNING:
Do NOT attempt to decrypt your files with any software as it is obselete and will not work, and may cost you more to unlckok your files.
Do NOT change file names, mess with the files, or run decryption software as it will cost you more to unlock your files-
-and there is a high chance you will lose your files forever.
Do NOT send 'PAID' button without paying, price WILL go up for disobedience.
Do NOT think that we wont delete your files altogether and throw away the key if you refuse to pay. WE WILL.
To finish, when the countdown will be done all of the keys will be destroy so all of your files will be lost !
```

Figure 5- Readme.txt file

Once the ransomware is being executed, it encrypts all data under in **3 minutes** and creates the **“READ_me.txt”** file. The file's content stipulates the necessity of a Bitcoin payment for decryption.

Static Analysis

eva.exe Analysis

File Name	eva.exe
MD5	45a357276700238ab1ea952f8ae8ab1b
SHA256	9deda531966448484b96d86893bc4158befc39c886abafa4bea932f514de4ec6
File Type	PE/32

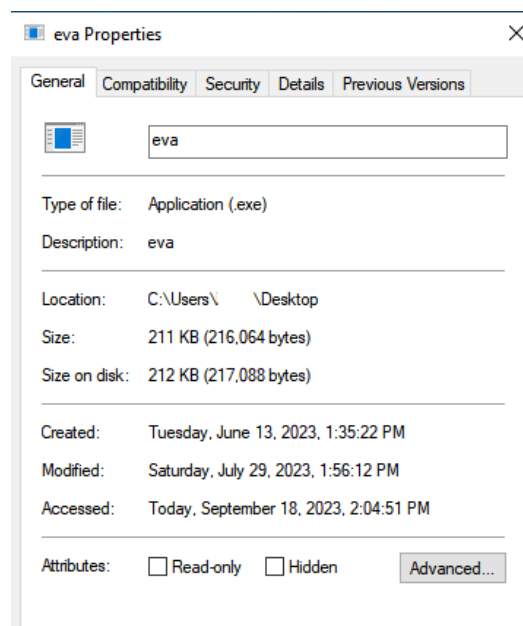


Figure 1- Information about malicious file

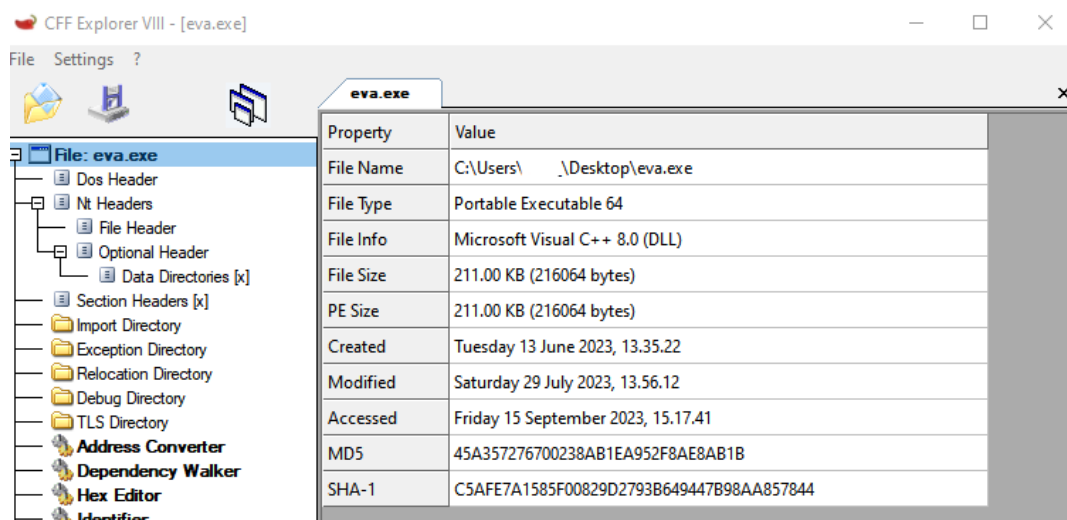


Figure 2- General Information

Eva0x00 RANSOMWARE

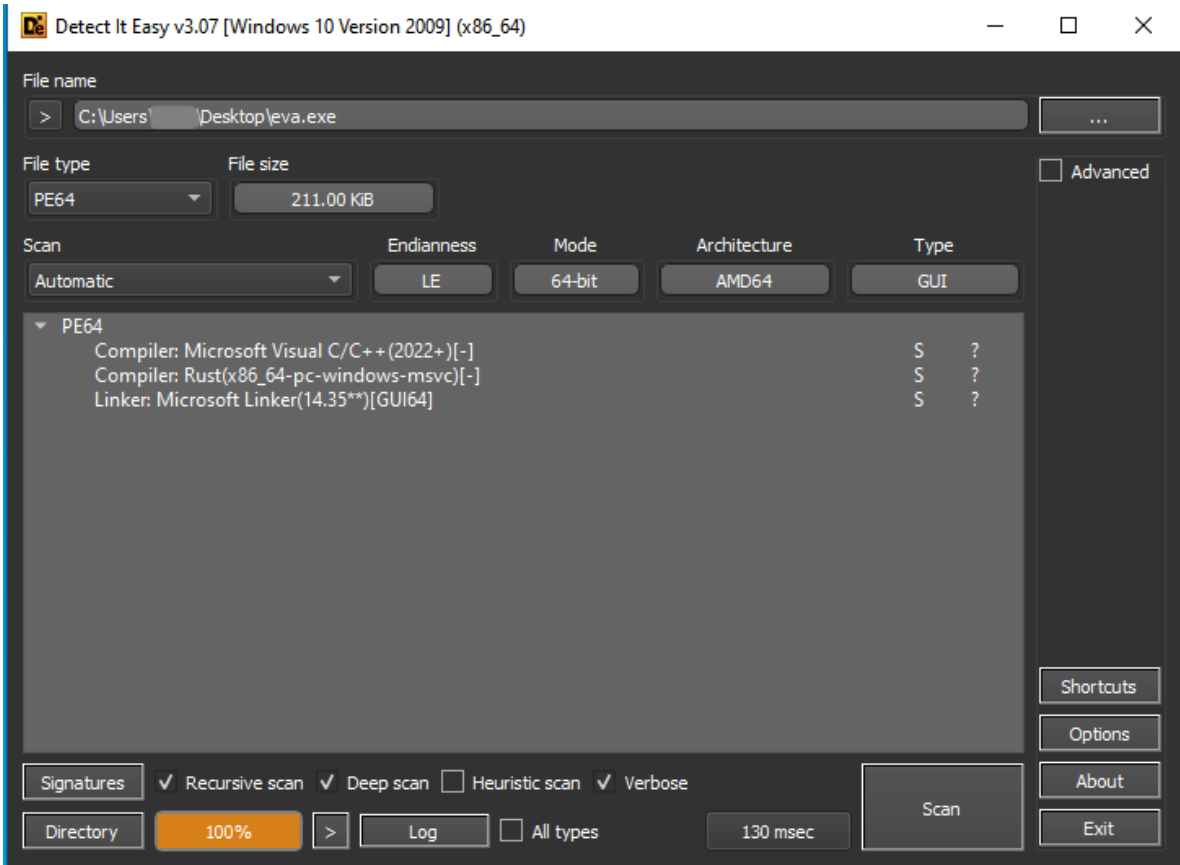


Figure 3- General Information

The ransomware was compiled using Microsoft Visual **C/C++** compiler, specifically a version from 2022 or later. The ransomware was also compiled using the **Rust programming language**, targeting the x86_64 architecture on the Windows platform with the MSVC (Microsoft Visual C++) toolchain.

11:38:...	eve_ransom.exe	3052	CreateFile	C:\Users\vbouser\Desktop\eve_ransom.exe	SUCCESS
11:38:...	eve_ransom.exe	3052	QueryAttributeTagFile	C:\Users\vbouser\Desktop\eve_ransom.exe	SUCCESS
11:38:...	eve_ransom.exe	3052	QueryBasicInformationFile	C:\Users\vbouser\Desktop\eve_ransom.exe	SUCCESS
11:38:...	eve_ransom.exe	3052	CreateFile	C:\Users\vbouser\AppData\Local\Temp	SUCCESS
11:38:...	eve_ransom.exe	3052	SetRenameInformationFile	C:\Users\vbouser\Desktop\eve_ransom.exe	SUCCESS
11:38:...	eve_ransom.exe	3052	CloseFile	C:\Users\vbouser\AppData\Local\Temp	SUCCESS
11:38:...	eve_ransom.exe	3052	CloseFile	C:\Users\vbouser\AppData\Local\Temp\eve_ransom.exe	SUCCESS
11:38:...	eve_ransom.exe	3052	CreateFile	C:\Windows\System32\windowsupdate\...	SUCCESS

Figure 4- Move ransomware to Temp path

After the software is executed, it moves itself from the directory it was run in, to the **\$HOME\AppData\Local\Temp** directory.

Dynamic Analysis

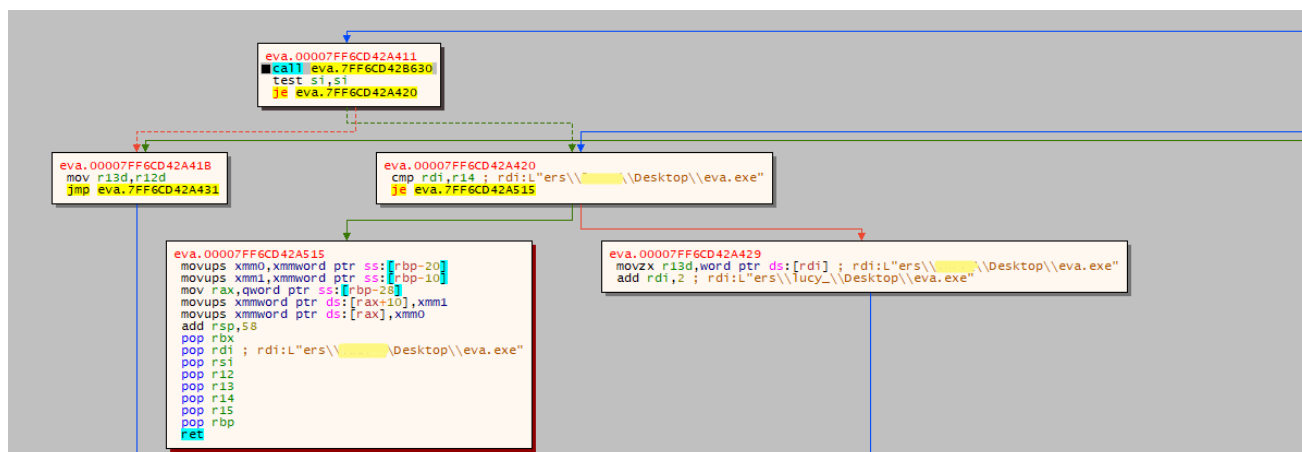


Figure 1- Gets location information

The malicious file gets the information in which location it is running.



Figure 2- Gets temp path information

When the ransomware runs, it uses the **GetTempPathw** API to move itself to the temp directory from its current location to get the path to the temp directory.

Eva0x00 RANSOMWARE

Address	Disassembly	Comment
FF6CD41615	4A 58	pop r8
FF6CD41615	4C:8BA5 78050000	mov r12,qword ptr ss:[rbp+578]
FF6CD41615	4C:8B9F 00000000	mov rcx,r12
FF6CD41615D	48:89DA	mov rdx,rbx
FF6CD416160	FF15 DA3E0200	call qword ptr ds:[<MoveFileExW>]
FF6CD416166	85C0	test eax,eax
FF6CD416168	74 27	jz eva.7FF6CD416191
FF6CD41616A	48:85F6	test rs1,rs1
FF6CD41616D	74 08	jz eva.7FF6CD41617A
FF6CD41616F	6A 02	push 2
FF6CD416171	5A	pop rdx
FF6CD416172	48:89D9	mov rcx,rbx
FF6CD416175	E8 46120000	call eva.7FF6CD4173C0
FF6CD41617A	48:83BD 38050000 00	cmp qword ptr ss:[rbp+538],0
FF6CD416182	74 5A	jz eva.7FF6CD4161DE
FF6CD416184	6A 02	push 2
FF6CD416186	5A	pop rdx
FF6CD416187	4C:89E1	mov rcx,r12
FF6CD41618A	E8 31120000	call eva.7FF6CD4173C0
FF6CD41618F	E8 4D	jmp eva.7FF6CD4161DE
FF6CD416191	FF15 893E0200	call qword ptr ds:[<GetLastError>]
FF6CD416197	41:89C4	mov r12d,eax
FF6CD41619A	49:C1E4 20	shl r12,20
FF6CD41619E	49:83CC 02	or r12,2
FF6CD4161A2	48:85F6	test rs1,rs1
FF6CD4161A5	74 08	jz eva.7FF6CD416182

r12:L"C:\\Users\\[redacted]\\Desktop\\eaa.exe"

r12:L"C:\\Users\\[redacted]\\Desktop\\eaa.exe"
r12:L"C:\\Users\\[redacted]\\Desktop\\eaa.exe"

```
<eva.&MoveFileExW>=<kernel32.MoveFileExW>
```

\$6160 #5560

Dump 4 | Dump 5 | Watch 1 | [x]=Locals | Struct

Offset	Hex	ASCII
0	65 72 73	C:\Users\[redacted]\A
1	61 5C 4C	ppdata\Local Tem
2	F0 AD BA	p\ea.e,exe,b...b..
3	F0 AD BA	b...b...b...b...
4	AB AB AB	.o....»««««««««««««
5	FE EE FE	««««ipib...ö...
6	01 00 00	ÖA.xb...ßq.xb...
7	00 00 00	

Figure 3- Move ransomware to Temp path

After the malware detects its location and the location of the temp directory, it moves the malicious file to the temp directory using the **MoveFileEx API**.

C:\Users\Admin\AppData\Local\Temp\eva.exe

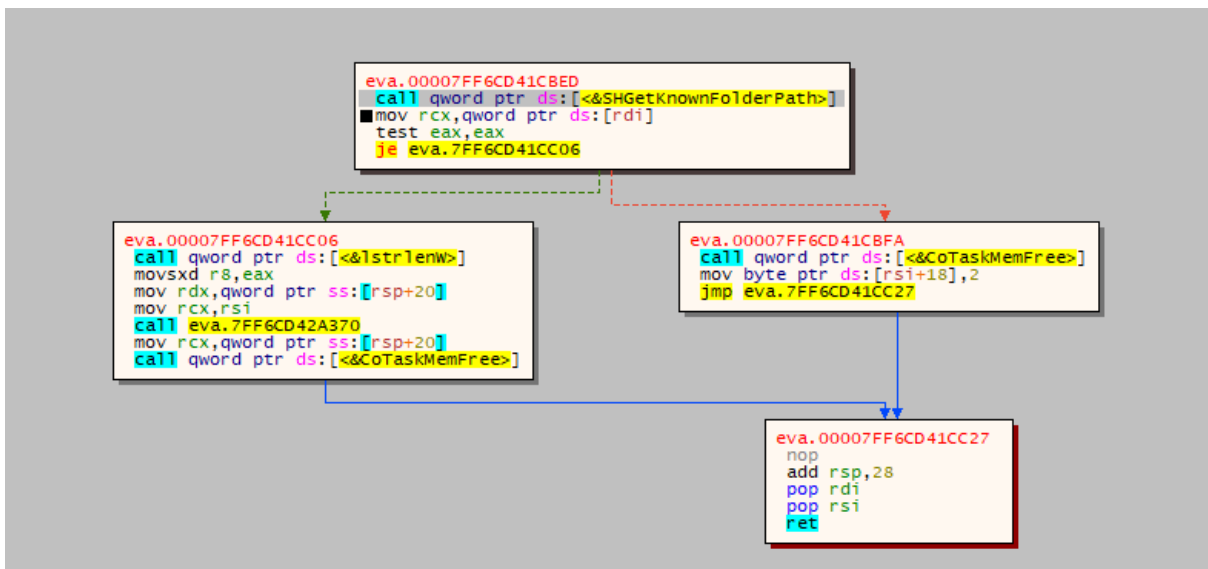


Figure 4- Gets the paths to be encrypted

The program utilizes the **SHGetKnownFolderPath API** to systematically retrieve individual directory paths for encryption. The directories being scanned are specified within a table. The process involves iterating through the provided table, invoking the **SHGetKnownFolderPath API** for each directory entry, and subsequently processing the retrieved paths for encryption purposes.

Eva0x00 RANSOMWARE

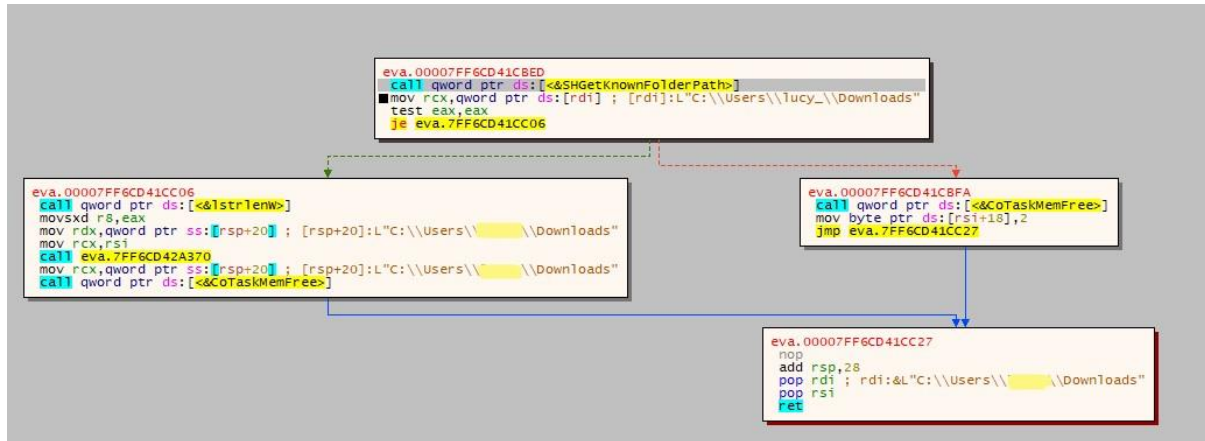


Figure 5- Gets the downloads path

C:\Users\Admin\Desktop	C:\Users\Admin\Downloads
C:\Users\Admin\Documents	C:\Users\Admin\desktop.ini
C:\Users\Admin\Videos	C:\Users\Admin\AppData\Roaming
C:\Users\Admin\Pictures	C:\Users\Admin\AppData\Default
C:\Users\Public	C:\Users\Admin\AppData\Local

Table 1 - Paths scanned

The directories being scanned are enumerated exactly as they are listed in the provided table.

```

50 push rsi
57 push rdi
48:81EC A8000000 sub rsp,A8
48:89D7 mov rdi,rdx
48:89CE mov rsi,rcx
0F1005 EA1D0100 movups xmm0,xmmword ptr ds:[7FF6CD43FA50]
0F294424 60 movaps xmmword ptr ss:[rsp+60],xmm0
B8 FFFFFFFF mov eax,FFFFFFFF
49:39C1 cmp r9,rcx
49:0F42C1 cmovb rax,r9
0F57C0 xorps xmm0,xmm0
0F114424 38 movups xmmword ptr ss:[rsp+38],xmm0
894424 30 mov dword ptr ss:[rsp+30],eax
4C:894424 28 mov qword ptr ss:[rsp+28],r8
48:8D4424 60 lea rax,qword ptr ss:[rsp+60]
48:894424 20 mov qword ptr ss:[rsp+20],rax
48:89D1 mov rcx,rdx
31D2 xor edx,edx
45:31C0 xor r8d,r8d
45:31C9 xor r9d,r9d
FF15 5DC60000 call qword ptr ds:[<&ZwReadFile>]
3D 03010000 cmp eax,103
75 12 jne eva.7FF6CD42DCBC
48:89F9 mov rcx,rdi
BA FFFFFFFF mov edx,FFFFFFFF
FF15 08C40000 call qword ptr ds:[<&WaitForSingleObject>]
8B4424 60 mov eax,dword ptr ss:[rsp+60]
3D 110000C0 cmp eax,C0000011
75 08 jne eva.7FF6CD42DCC8
0F57C0 xorps xmm0,xmm0
0F1106 movups xmmword ptr ds:[rsi],xmm0
EB 38 jmp eva.7FF6CD42DD03
3D 03010000 cmp eax,103
74 38 jbe eva.7FF6CD42DD00
85C0 test eax,eax
78 12 js eva.7FF6CD42DCE8
48:8B4424 68 mov rax,qword ptr ss:[rsp+68]
48:8946 08 mov qword ptr ds:[rsi+8],rax
48:C706 00000000 mov qword ptr ds:[rsi+0]
    
```

Figure 6- Using ZwReadFile

Using the **ZwReadFile API**, the program reads the file slated for encryption in a manner consistent with standard file read operations.

Eva0x00 RANSOMWARE

E8 5A2EFEFF	call eva.7FF6CD4173C0
48:8843 1C	mov rax,qword ptr ds:[rbx+1C]
0F1043 0C	movups xmm0,xmmword ptr ds:[rbx+C]
8848 24	mov ecx,dword ptr ds:[rbx+24]
48:C1E1 20	shl rcx,20
8853 28	mov edx,dword ptr ds:[rbx+28]
48:09CA	or rdx,rcx
8848 08	mov ecx,dword ptr ds:[rbx+8]
41:89C8	mov r8d,ecx
41:C1E0 15	shl r8d,15
41:C1F8 1F	sar r8d,1F
44:8848 2C	mov r9d,dword ptr ds:[rbx+2C]
0F284D C0	movaps xmm1,xmmword ptr ss:[rbp-40]
0F2855 D0	movaps xmm2,xmmword ptr ss:[rbp-30]
0F1156 58	movups xmmword ptr ds:[rsi+58],xmm2
0F114E 48	movups xmmword ptr ds:[rsi+48],xmm1
8366 10 00	and dword ptr ds:[rsi+10],0
45:21C8	and r8d,r9d
8366 18 00	and dword ptr ds:[rsi+18],0
0F1146 20	movups xmmword ptr ds:[rsi+20],xmm0
48:8946 30	mov qword ptr ds:[rsi+30],rax
48:8956 38	mov qword ptr ds:[rsi+38],rdx
894E 40	mov dword ptr ds:[rsi+40],ecx
44:8946 44	mov dword ptr ds:[rsi+44],r8d
894E 68	mov dword ptr ds:[rsi+68],ecx
44:894E 6C	mov dword ptr ds:[rsi+6C],r9d
48:8845 00	mov rax,qword ptr ss:[rbp]
48:8946 70	mov qword ptr ds:[rsi+70],rax
C646 78 00	mov byte ptr ds:[rsi+78],0
48:8326 00	and qword ptr ds:[rsi],0
48:81C4 98000000	add rsp,98
5B	pop rbx

Figure 7- Part of encryption

The eva0x00 ransomware initiates the encryption process on the read file after performing an extended encryption algorithm.

00007FF6CD417375	48:89CE	mov rsi,rcx	
00007FF6CD417378	48:880D 11DE0200	mov rcx,qword ptr ds:[7FF6CD445190]	
00007FF6CD41737F	48:85C9	test rcx,rcx	
00007FF6CD417382	75 15	jne rramsom.7FF6CD417399	
00007FF6CD417384	FF15 DE2C0200	call qword ptr ds:[<&GetProcessHeap>]	
00007FF6CD41738A	48:85C0	test rax,rax	
00007FF6CD41738D	74 18	je rramsom.7FF6CD4173AA	
00007FF6CD41738F	48:89C1	mov rcx,rax	
00007FF6CD417392	48:8905 F7DD0200	mov qword ptr ds:[7FF6CD445190],rax	
00007FF6CD417398	31D2	xor edx,edx	
00007FF6CD41739B	49:89F0	mov r8,rsi	
00007FF6CD41739E	48:83C4 20	add rsp,20	
00007FF6CD4173A2	5E	pop rsi	
00007FF6CD4173A3	48:FF25 C62C0200	jmp qword ptr ds:[<&RtlAllocateHeap>]	JMP.&RtlAllocateHeap
00007FF6CD4173AA	31C0	xor eax,eax	
00007FF6CD4173AC	48:83C4 20	add rsp,20	
00007FF6CD4173B0	5E	pop rsi	
00007FF6CD4173B1	C3	ret	
00007FF6CD4173B2	CC	int3	
00007FF6CD4173B3	CC	int3	
00007FF6CD4173B4	CC	int3	
00007FF6CD4173B5	CC	int3	

Figure 8- Using RtlAllocateHeap

The eva0x00 ransomware allocates memory space equal to the size of the encrypted file using the **RtlAllocateHeap** API in order to create the encrypted file.

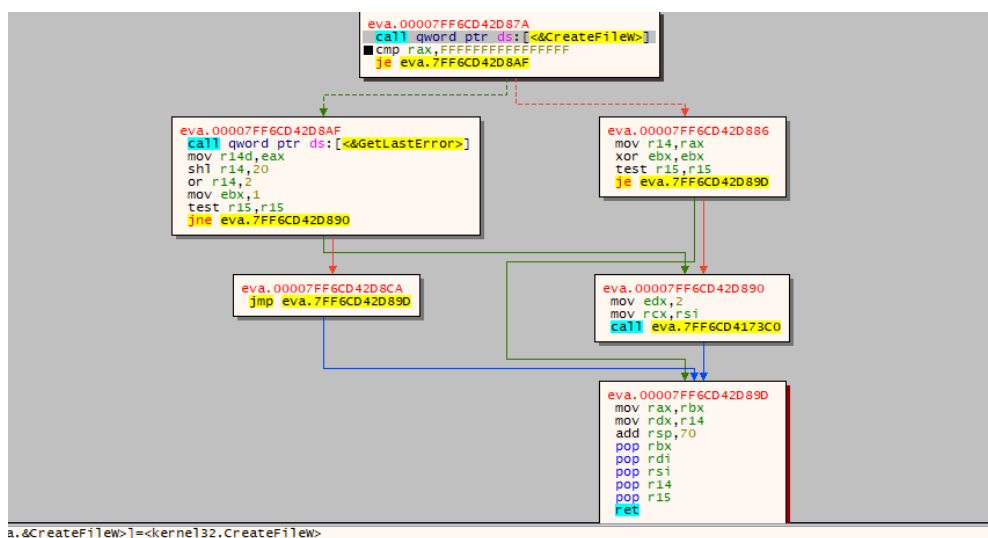


Figure 9- Creating the encrypted file

Using the **CreateFile** API, the eva0x00 ransomware generates the encrypted file in the directory where the original file was located, appending the extension **".encrypt_by_eva0x00"** to it.

Eva0x00 RANSOMWARE

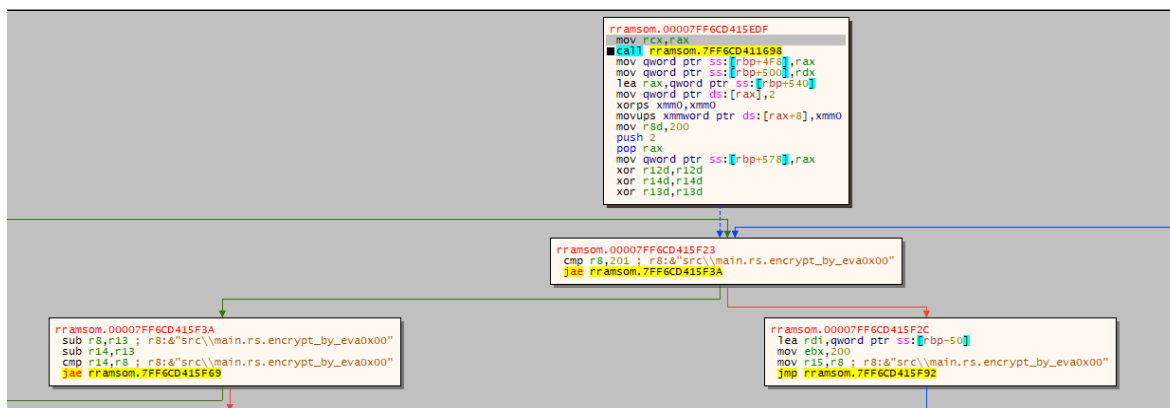


Figure 10- Adding an extension to the encrypted file

In the Rust programming language, the ransomware defines the **".encrypt_by_eva0x00"** extension, typically by using a variable or constant, to represent the file extension.

48:885424 78 E8 04E50100 48:89C1 4C:8D05 9A860200 E8 4D4FFFFF 48:89C1 E8 D01C0200 48:85C0 0F84 78370000 48:89C6 48:89D7 31D2 48:8D5A 10 48:83F8 20 0F84 A5000000 49:880C16 49:8B5416 08 49:89F0 49:89F9 E8 3C400100 48:89DA 84C0 74 D7 EB 59 48:888C24 F0010000 48:889424 F8010000 E8 69200200 48:888F34 10030000	mov rdx,qword ptr ss:[rsp+78] call eva.7FF6CD430740 mov rcx,rax lea r8,qword ptr ds:[7FF6CD43A8E0] call eva.7FF6CD411698 mov rcx,rax call eva.7FF6CD433F30 test rax,rax je eva.7FF6CD415904 mov rsi,rax mov rdi,rdx xor edx,edx lea rbx,qword ptr ds:[rdx+10] cmp rbx,20 je eva.7FF6CD412317 mov rcx,qword ptr ds:[r14+rdx] mov rdx,qword ptr ds:[r14+rdx+8] mov r8,rsi mov r9,rdi call eva.7FF6CD4262C2 mov rdx,rbx test al,al je eva.7FF6CD412264 jmp eva.7FF6CD4122E8 mov rcx,qword ptr ss:[rsp+1F0] mov rdx,qword ptr ss:[rsp+1F8] call eva.7FF6CD434300 mov rcx,qword ptr ss:[rsp+210]	rcx:"READ_me.txt" 00007FF6CD43A8E0:&"src\\main.rs.encrypt_by_eva0x00" rcx:"READ_me.txt" 20: ' ' rcx:"READ_me.txt"
--	--	---

Figure 11- Create READ_me.txt

After creating the encrypted file, the eva0x00 ransomware proceeds to append the **"READ_me.txt"** file to the same directory where the encrypted file is located.

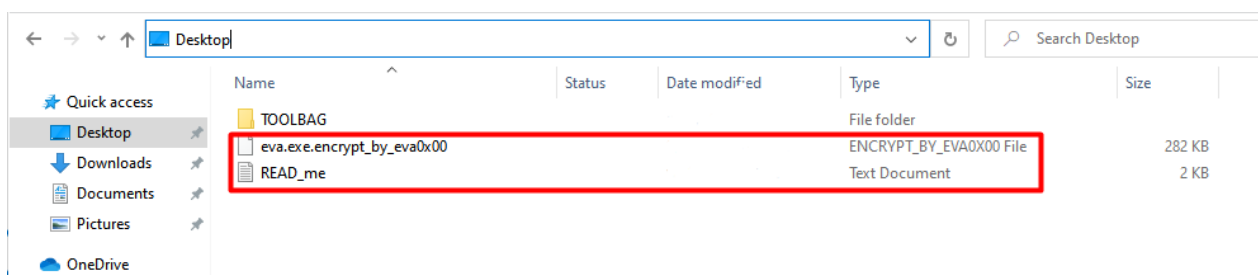


Figure 12- Encrypted files

The program systematically traverses all directories it searches, targeting files with specific extensions **".encrypt_by_eva0x00"**., including **".pdf," ".png," ".jpg," ".jpeg," ".mp3," ".mp4," ".txt," and ".exe."** It encrypts these files and alters their file extensions accordingly. Upon completion of the file traversal process, it generates a **"READ_me.txt"** file. Following the successful encryption of all scanned files, the program proceeds to terminate itself.

IOCs

HASHs:

IOC Type	IOC
MD5	45a357276700238ab1ea952f8ae8ab1b
SHA1	c5afe7a1585f00829d2793b649447b98aa857844
SHA256	9deda531966448484b96d86893bc4158befc39c886abafa4bea932f514de4ec6

YARA RULE

```
import "hash"
rule Eva0x00
{
  meta:
    author = "Kerime Gencay"
    description = "Eva0x00 Ransomware Rule"
    file_name = "eva.exe"
    hash = "45a357276700238ab1ea952f8ae8ab1b"

  strings:

    $text1 = "TheKey.txtC:\\Users\\abist\\.cargo\\registry\\src\\index.crates.io-6f17d22bba15001f\\aes-0.7.5\\src\\soft\\fixslice64.rs" //rust library
    $text2 = "library\\std\\src\\sys\\windows\\path.rs" //rust library
    $text3 = "library\\std\\src\\sys_common\\thread_info.rs" //rust library
    $text4 = "src\\main.rs.encrypt_by_eva0x00" // encrypted file extension

    $opc1 = {FF 15 DE 2C 02 00 48 85 C0 74 1B 48 89 C1 48 89 05 F7 DD 02 00 31 D2 49 89 F0 48 83 C4 20 5E 48 FF 25 C6 2C 02 00 31 C0 48 83 C4 20 5E}
    $opc2 = {72 22 48 8B 0E 48 01 D9 48 8D 54 24 2C 49 89 F8 E8 B9 C5 00 00 48 01 FB 48 ?? ?? 10 48 ?? ?? 30 5B 5F 5E}

  condition:

    uint16(0) == 0x5A4D and (any of ($text*, $opc*))
}
```

MITRE ATT&CK TABLE

Discovery	Impact	Defense Evasion	Persistence	Reconnaissance
T1082 Information Discovery	T1486 Data Encrypted for Impact	T1036 Masquerading	T1047 Create or Modify Systems	T1566 Phishing
T1543 File and Directory Discovery		T1564.001 Hidden Files and Directories		

MITIGATIONS

- Ransomware is often spread through phishing emails. Be careful when receiving a suspicious email and avoid opening attachments or clicking on links.
- Strong passwords provide better protection against ransomware. Change your passwords frequently and make them as complex as possible.
- Paying the ransom only encourages ransomware attacks. Before paying the ransom, try other methods to recover your files.
- It is recommended that organizations use security products to secure potential entry points such as endpoints, email, webs, and networks. Thus, they can protect themselves against ransomware attacks.