





LockBit 3 Decryptors analysis

In this analysis we take in consideration the decryptor tools of LockBit 3.0 ransomware (Windows version) and LockBit (ESXi variant).

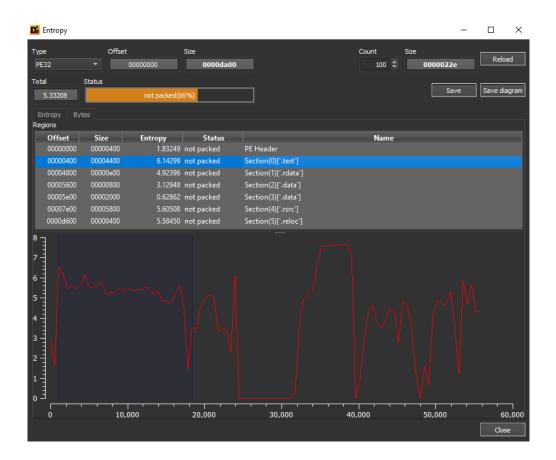
Generally, a decryptor tool permits to decrypt the files encrypted by a ransomware threat; this type of tool could be delivered directly by the ransomware gang, who developed the threat (after the payment of the requested ransom) or is developed ad-hoc from the global cybersecurity community to be distributed to help the victims to decrypt their files without the need to actually pay the ransom. Unfortunately, this is not always possible, however the cybersecurity community constantly work to develop new decryptors for the ransomware threats and the related variants. For this reason, it is not recommended to remove the encrypted files, because it is not excluded that, in a future, they can effectively be decrypted. Usually, the decryptors developed by the cybersecurity community take in consideration the ransomware threats (and the possible variants) which use an offline encryption key. In this case the analyzed decryptors have been downloaded from the Web.

The executable **LBB_Decryptor.exe** doesn't have a high level of entropy, which would indicate a "packed" state, in the imports we can see that the mpr.dll library is used to execute network checking functions, SHLWAPI.dll to manage filesystem items, search the paths of files to decrypt, check the existence of the files, if some directories are empty, etc.

AdvAPI32.dll library is used to call functions to manage the **MD5** contained in the **README** file and manipulate (restore) registry keys related to the infection (Create, Delete, SetValue operations). The library SHELL32.dll, instead, is imported to execute the function **SHGetSpecialFolderPathW** to access to the settings of the file **NTUSER.dat**; there are also evidences associated to the creation and management of threads to execute decryption functions in a more efficient way.



Detect It Easy v3.04	[Windows 10 Version 1809](x86_6-	4)		_	
File name					
C:/Users/IEUser/Deskto	p/Lockbit3-Decryptor/DECRYPTOR_A	ALL_PC/LBB_Decryp	tor.exe		
File type	Entry point		Base address		File info
PE32 -	00404c72 >	Disasm	00400000	Memory map	MIME
PE	Export Import	Resources	.NET TLS	Overlay	Hash
Sections	Time date stamp S	ize of image	Resources		Strings
0006 >	2022-07-14 03:29:16	00012000	Manife	st Version	Entropy
Scan	Endianness	Mode	Architecture	Туре	Hex
Automatic	▼ LE	32-bit	I386	GUI	Signatures
✓ PE32					Demangle
Linker: Microso	oft Linker(14.12)[GUI32,admin]				
					Shortcuts
Circultures	_	D			Options
Signatures		Deep scan 📕 Rec	cursive scan 📃 All types	Scan	About
Directory	100%	> Log	97 msec		Exit





property	value
md5	EEE0D4094CD4E7CF318DA3E7E29E90E2
sha1	34ED29F3CD23B7662ADAF4D8B8AD6D68C5298AA6
sha256	E73083AC86F25E8DA3AA28547ED386F5336C57CFD191117DEA3451498FEB36F1
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 0F FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00
first-bytes-text	M Z @
file-size	55808 (bytes)
entropy	5.332
imphash	3E00A7DD9BB741EE6A4F22F50F51341A
signature	n/a
entry-point	8B FF 55 8B EC 83 C4 DC 33 C0 89 45 FC 89 45 E8 89 45 E4 89 45 E0 89 45 EC 89 45 F0 89 45 DC C7 05
file-version	n/a
description	n/a
file-type	executable
сри	32-bit
subsystem	GUI
compiler-stamp	0x62CFEFFC (Thu Jul 14 03:29:16 2022)
debugger-stamp	0x62CFEFFC (Thu Jul 14 03:29:16 2022)
resources-stamp	0x00000000 (empty)
import-stamp	0x00000000 (empty)
exports-stamp	n/a
version-stamp	n/a
certificate-stamp	n/a

👔 Stud_PE ed	Stud_PE editing : "LBB_Decryptor.exe" - [32bit app] — 🗌 🗙								
File Edit To	File Edit Tools Help								
c:\users\ieuse	c:\users\ieuser\desktop\lockbit3-decryptor\decryptor_all_pc\lbb_decryptor.exe								
🎙 Headers 🐌 Dos 🛯 🖻 Sections 🛛 🕫 Functions 🛛 🕫 Resources 🛛 획 Signature 📔 🖲 F 🗨 🕨									
HEADERS	(Coff+Optional) DATA DIRECTORY							
00004072	EntryPoint (rva)	RVA Size Raw							
	EntryPoint (raw)	Import Table 000062F8 000000C8 00004AF8							
00400000	ImageBase	Export Table 00000000 00000000 00000000							
	Size of Image	Data Dir : IMAGE_DIR_ENTRY_RESOURCE 💌							
00001000	Sections Alignment								
00000200	File Alignment	GoHex ++ 0000B000 00005780 00007E00							
0006	Number of sections								
0102	Characteristics +	Basic HEADERS tree view in hexeditor SAVE to file							
Visit Stud PE F	Forum <-NewsHere	Test'it Rva<=>Raw File Compare OK							



Di die

_		
Address	Hex Symbols	•
0000:4072	8b ff 55 8b ec 83 c4 dc 33 c0 89 45 fc 89 45 e8U3EE.	i .
0000:4082	89 45 e4 89 45 e0 89 45 ec 89 45 f0 89 45 dc c7 .EEEE	i .
0000:4092	05 Oc 77 40 00 ff ff ff ff e8 40 c6 ff ff 85 c0	i) 🔰 .
0000:40a2	75 05 e8 97 ff ff ff e8 42 d6 ff ff 85 c0 0f 85 uB	ij 📕 .
0000:40b2	d4 01 00 00 e8 51 d8 ff ff e8 20 c6 ff ff 85 c0Q	i .
0000:40c2	75 05 e8 83 de ff ff 6a 00 6a 00 6a 11 6a fe e8 uj.j.j.j.	i .
0000:40d2	04 05 00 00 68 a0 76 40 00 e8 04 04 00 00 68 b4h.v@h.	i .
0000:40e2	76 40 00 50 e8 ff 03 00 00 89 45 dc 83 7d dc 00 v@.PE}	i .
0000:40f2	74 Oc 8d 45 f4 50 68 01 00 00 80 ff 55 dc e8 7b tE.PhU {	
0000:4102	da ff ff e8 42 d6 ff ff a3 84 80 40 00 e8 a0 03B@	
0000:4112	00 00 8b c8 8d 45 f8 50 51 e8 4e 04 00 00 89 45E.PQ.NE	
0000:4122	fc 83 7d fc 00 75 05 e9 3e 01 00 00 83 7d f8 01}.u>}.	🧧 .
Hex 4072	4072 00	
		Close

_

Here are the details of the import of mpr.dll for the functions WNetAddConnection2W and WNetGetUniversalNameW:

Di PE							_		×
Reload		Hex	Disasm	Strings	Memory map	Entropy	Heuristic scan	📕 Re	
Info Hex Disasm	Hash 64 00000034c2ec	Hash cab7b	aab1a433						
Hash Strings Signatures Memory map Entropy	ginalFirstT 000003e0 2 000064fc 3 00006488	hi neDateStar 00000000 00000000 00000000	orwarderCha 00000000 00000000 00000000	Name 0000090e 00006992 000069f4	FirstThunk 0000613c 000060c8	Hash 80849938 4b2fc056 b0944d95	KEKIVEL32.ali comctl32.dll SHELL32.dll		•
Heuristic scan IMAGE_DOS_HEADER ▼ IMAGE_NT_HEADERS IMAGE_FILE_HEADER	4 00006510 5 000063c0 6 00006540	00000000 00000000 00000000	00000000 00000000 00000000	00006a72 00006aee 00006c9a	00006150 00006000 00006180	2f46ccfe d2d829f4 4fa074d1	msvcrt.dll AdvAPI32.dll ntdll.dll		
IMAGE_OPTIONAL_HEADER IMAGE_DIRECTORY_ENTRIES Sections Import Resources	7 0000649c 8 00006504	00000000 00000000 Ordinal	00000000 00000000 Hint	00006d46 00006d80	000060dc 00006144	7ff1d223 2ba60aa4 Name	SHLWAPI.dll mpr.dll		
Resources Manifest Relocs Debug	0 00006d52 1 00006d68		0003 WNet/	AddConnect GetUniversal		Hame	_		
	••								_ 0



Di PE									_		×
Reload		н	ex	Disasm		Strings	Memory map	Entropy	Heuristic scan	📕 Re	
Info Hex	Hash 6	4	н	ash 32							
Disasm	00	000034c2ecal	57b	aab1	a433						
Hash Strings		ginalFirstThu				Name	FirstThunk				•
Signatures	1		00000000	00000		опполовое	00000020	80349938	KEKINEL32.0II		
Memory map	2		00000000	000000		00006992	0000613c	4b2fc056	comctl32.dll		
Entropy Heuristic scan	3		00000000	000000		000069f4	000060c8	b0944d95	SHELL32.dll		
IMAGE DOS HEADER	4		00000000	000000		00006a72	00006150	2f46ccfe	msvcrt.dll		
IMAGE_NT_HEADERS	5		00000000	000000		00006aee	00006000	d2d829f4	AdvAPI32.dll		
IMAGE_FILE_HEADER	6		00000000	000000		00006c9a	00006180	4fa074d1	ntdll.dll		
 IMAGE_OPTIONAL_HEADER IMAGE DIRECTORY ENTRIES 	7		00000000	000000		00006d46	000060dc	7ff1d223	SHLWAPI.dll		
Sections	8	00006504	00000000	000000	000	00006d80	00006144	2ba60aa4	mpr.dll	_	
Import Resources		Thunk	Ordina	l Hint				Name			
Manifest	U	00006cda	ordina		PathFir	ndFileName	W				
Relocs	1	00006cee		005a	PathlsD	DirectoryEm	nptyW				
Debug	2	00006cc4		0047	PathFir	ndExtensior	νW				
	3	00006cb2		0045	PathFil	eExistsW					
	4	00006d1a		0061	Pathls	VetworkPat	hW				
	5	00006d06		005b	Pathls	DirectoryW					
	6	00006d30		008b	PathRe	moveFileSp	becW				
	7	00006ca4		0034	PathAp	pendW					

.

De PE

Reload		F	lex	Disasm		Strings	Memory map	Entropy	Heuristic scan	E Readonly
Info	Hash 64		Has	h 32						
Hex Disasm	000	00034c2eca	b7b	aab1a	a433					
Hash Strings		ginalFirstThu	neDateSta			Name	FirstThunk	Hash 80849938	KEKINEL32.011	•
Signatures Memory map		000064fc	00000000	000000		00006992	0000613c	4b2fc056	comctl32.dll	
Entropy Heuristic scan	-	00006488	00000000	000000		000069f4	000060c8	b0944d95	SHELL32.dll	
IMAGE_DOS_HEADER		00006510 000063c0	00000000	000000		00006a72 00006aee	00006150 00006000	2f46ccfe d2d829f4	msvcrt.dll AdvAPI32.dll	
 IMAGE_NT_HEADERS IMAGE_FILE_HEADER 		00006540	00000000	000000		00006c9a	00006180	4fa074d1	ntdll.dll	
 IMAGE_OPTIONAL_HEADER IMAGE_DIRECTORY_ENTRIES 		0000649c 00006504	00000000	000000		00006d46 00006d80	000060dc 00006144	7ff1d223 2ba60aa4	SHLWAPI.dll mpr.dll	
Sections Import	-	00000304	0000000	000000	,00 1			2040044	mpron	⁰
 Resources 		Thunk	Ordinal	Hint				Name		
Manifest Relocs	0	00006aae		0000	MD5Up	odate				
Debug	1	00006aa4		0000	MD5Ini					
	2	00006a98		0000	MD5Fir	nal				
	3	00006a7e		0000	Conver	tSidToString	gSidW			
	4	00006acc		0000	RegDel	eteKeyW				
		00006adc		0000	RegSet	ValueExW				
	6	00006aba		0000	RegCre	ateKeyExW				
										•

_

 \times



Di PE									-		×
Reload		н	ex	Disasm		Strings	Memory map	Entropy	Heuristic scan	📕 🗖 Re	
Info Hex Disasm	Hash 64 0000	00034c2ecal	Hash b7b	aab1	a433						
Hash		ginalFirstThu	neDateStar	rwar	derCha	Name	FirstThunk	Hash			•
Strings Signatures	0 0	000064c0	00000000	00000	000	00006668	00006100	6653d947	USER32.dll		
Memory map		000063e0	00000000	00000	000	0000696e	00006020	8ba49938	KERNEL32.dll		
Entropy Heuristic scan)00064fc	0000000	00000	000	00006992	0000613c	4b2fc056	comctl32.dll		
IMAGE DOS HEADER		00006488	0000000	00000	000	000069f4	000060c8	b0944d95	SHELL32.dll		
IMAGE_NT_HEADERS		00006510	0000000	00000	000	00006a72	00006150	2f46ccfe	msvcrt.dll		
IMAGE_FILE_HEADER VIMAGE_OPTIONAL_HEADER		000063c0	0000000	00000	000	00006aee	00006000	d2d829f4	AdvAPI32.dll		
IMAGE_DIRECTORY_ENTRIES	6 (00006540	0000000	00000	000	00006c9a	00006180	4fa074d1	ntdll.dll		
Sections	7 (1000649c	0000000	00000	000	00006d46	000060dc	7ff1d223	IIIN IQAW IH2		•
Import Resources		Thunk	Ordinal	Hint				Name			
Manifest	0 0	000069da		0175	SHGet	SpecialFolde	rPathW				
Relocs	1 ()00069a0		0007	Comm	andLineToA	rgvW				
Debug		000069c8		008b	SHCha	ngeNotify					
		000069b6		0028		ueryFileW					

De PE

Reload Info Hex Disasm Hash Strings Signatures Memory map Entropy Heuristic scan IMAGE_DOS_HEADER • IMAGE_DOS_HEADER • IMAGE_FILE_HEADER • IMAGE_DIRECTORY_ENTRIES rections • rort * Cres * ifest Hex Disasm Strings Memory map Entropy Heuristic scan 📕 Reload Hash 64 Hash 32 00000034c2ecab7b aab1a433 rstThi neDateStan irwarderCha Name FirstThunk Hash 20 0000000 0000000 0000000 0000000 0000020 80249958 KERNELSZAGII 00000000 00000000 00006992 0000613c 4b2fc056 comctl32.dll fc 00000000 b0944d95 msvcrt.dll AdvAPI32.dll 00000000 00000000 d2d829f4 00006aee 00006000 4fa074d1 ntdll.dll SHLWAPI.dll 00000000 00006d80 2ba60aa4 Name Thunk Ordinal Hint 0006922 05ba WaitForMultipleObjects 006906 0544 SetThreadPriority 00068f2 0006674 0006682 007c CloseHandle 00c0 CreateFileW 006690 00c5 CreateloCompletionPort)0066aa 00e7 CreateThread

_

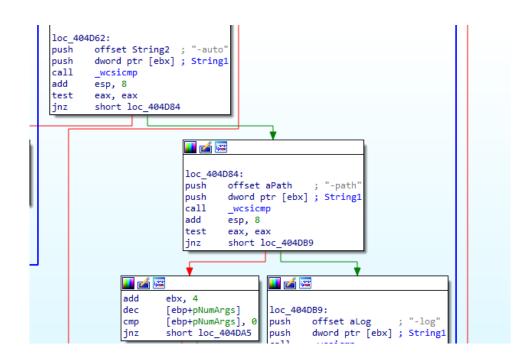


lea	eax, [ebp+pszPath]
push	eax ; pszPath
push	0 ; hwnd
call	SHGetSpecialFolderPathW
push	offset pszMore ; "NTUSER.DAT
lea	eax, [ebp+pszPath]
push	eax ; pszPath
call	PathAppendW
lea	eax, [ebp+pszPath]
push	2 I
call	GetFileAttributesW
cmp	eax, 0FFFFFFFh
jnz	short loc_40132B
	Y
	mov [ebp+var_4], 1
	loc_40132B:
	mov eax, [ebp+var_4]
	mov esp, ebp
	pop ebp
	retn
	sub 4012E0 endp

Contextually to the access to the file **NTUSER.dat** with the function **sub_4012E0** is possible to see how execution parameters have been set, included **"-log"** option:

CompletionKey= dword ptr -0Ch NumberOfBytesTransferred= dword ptr -8 Overlapped= dword ptr -4 lpThreadParameter= dword ptr 8 push ebp mov ebp, esp add esp, 0FFFFFECh push 2 ; nPriority	
add esp, 0FFFFFECh	
push @FFFFFFEh ; hThread call SetThreadPriority	
<pre>loc_4030FB: ; dwMilliseconds push 0FFFFFFFh lea eax, [ebp+Overlapped] push eax ; lpOverlapped lea eax, [ebp+CompletionKey] push eax ; lpCompletionKey lea eax, [ebp+NumberOfBytesTransferred] push eax ; lpNumberOfBytesTransferred</pre>	





By analyzing the extractable strings from the executable is possible to highlight the peculiarity to stop a timer (through the KillTimer function), the enumeration of the encrypted files through cycles and the use of the functions FindFirstFileExW, FindNextFileW, GetLogicalDrivesW and GetDriveTypeW which are called to manage the drives of the machine on which the decryptor is executed. Furthermore, some modifications of the files attribute are performed through the execution of the function SetFileAttributesW and, for efficiency, to point to the files that are managed, it is executed the function SetFilePointerEx.



De Stri	ings			– 🗆 X
Filter		ANSI		▼ UTF8 Unicode C Strings 5 \$ Save Search
	Offset 👻	Size	Туре	String
22	4aec	80000000	A	.rsrc\$02
23	4d92	0000000f	A	DialogBoxParamW
24	4da4	0000000c	A	EnableWindow
25	4db4	00000009	А	EndDialog
26	4dc0	0000000a		GetDlgltem
27	4dce	00000009	A	KillTimer
28	4dda	00000009	A	LoadiconW
29	4de6	0000000Ь	A	MessageBoxW
30	4df4	0000000c	Α	SendMessageW
31	4e04	b0000000	A	SetDigitemint
32	4e14	0000000c	A	SetSysColors
33	4e24	80000008	A	SetTimer
34	4e30	0000000c	A	SetWindowPos
35	4e40	0000000e	A	SetWindowTextW
36	4e52	00000015	A	SystemParametersInfoW
37	4e68	0000000a	А	USER32.dll
38	4e76	0000000ь	А	CloseHandle
39	4e84	0000000ь	A	CreateFileW
40	4e92	00000016	A	CreateloCompletionPort
41	4eac	0000000c	А	CreateThread
42	4ebc	0000000ь	A	DeleteFileW
43	4eca	000000ь	A	ExitProcess
44	4ed8	00000009	A	FindClose
				Close

👔 Stud_PE editing : "LBB_Decryptor.exe" - [32bit app] File Edit Tools Help	- 🗆 X							
c:\users\ieuser\desktop\lockbit3-decryptor\decryptor_all_pc\lbb_decryptor.exe Dos Sections fx Functions Rs Resources Signature R Procs Imported Functions Imported Functions Exported Functions Exported Functions Exported Functions								
 ▲ -f(※) CloseHandle ord:124 rva2iat: 0000603C ▲ -f(※) CreateFileW ord:192 rva2iat: 00006040 ▲ -f(※) CreateIoCompletionPort ord:197 rva2iat: 00006048 ▲ Image Thunk Data raw: 00004848, rva: 01 ▲ Import by Name raw: 00004EAA, rva: 000 ▲ Import by Name raw: 00004EAA, rva: 000 ▲ Import by Name raw: 00004EAA, rva: 000 ▲ Import by Name raw: 0000604C ▲ Import by Name raw: 00006050 ▲ FindClose ord:360 rva2iat: 00006054 ▲ ImpFindFirstFileExW ord:366 rva2iat: 00006058 	Found No Exports !							
Show Imp C OriginalFirstThunk © FirstThunk	Show Exp 🔽							



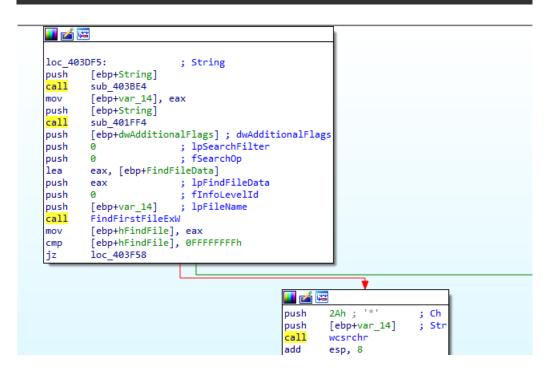


Here the extraction of the string related to the function GetLogicalDriveStringsW, which fills a strings buffer with the detected drives on the host:

🔓 Strings					- 🗆 ×
Filter			ANSI		▼ UTF8 Unicode C Strings 5 \$ Save Search
	Offset	•	Size	Туре	String
43		4eca	0000000Ь		ExitProcess
44		4ed8	00000009		FindClose
45		4ee4	00000010		FindFirstFileExW
46		4ef8	b0000000		FindNextFileW
47		4f08	00000017		FlushConsoleInputBuffer
48		4f22	0000000f		GetCommandLineW
49		4f34	00000010		GetConsoleWindow
50		4f48	b0000000		GetDriveTypeW
51		4f58	00000011		GetExitCodeThread
52		4fбc	00000012		GetFileAttributesW
53		4f82	000000ь		GetFileSize
54		4f90	b0000000		GetFileSizeEx
55		4fa0	00000017	A	GetLogicalDriveStringsW
56		4fba	00000010		GetModuleHandleW
57		4fce	0000000e		GetProcAddress
58		4fe0	00000019		GetQueuedCompletionStatus
59		4ffc	000000c		GetStdHandle
60		500c	0000000a		GlobalFree
61		501a	00000012		HeapSetInformation
62		5030	00000014		InterlockedIncrement
63		5048	000000c		lsBadReadPtr
64		5058	0000000ь		MoveFileExW
65		5066	0000001a	А	PostOueuedCompletionStatus
					Close



🔓 Strings					– – ×
Filter			ANSI		▼ UTF8 Unicode C Strings 5 \$ Save Search
	Offset	-	Size	Туре	String
64		5058	000000b	А	MoveFileExW
65		5066	0000001a	А	PostQueuedCompletionStatus
66		5084	00000008	А	ReadFile
67		5090	0000000c	А	ResumeThread
68		50a0	00000017	А	SetConsoleTextAttribute
69		50ba	00000010	А	SetConsoleTitleW
70		50ce	0000000c	А	SetEndOfFile
71		50de	00000012	А	SetFileAttributesW
72		50f4	00000010	А	SetFilePointerEx
73		5108	00000011		SetThreadPriority
74		511c	00000005	А	Sleep
75		5124	00000016	А	WaitForMultipleObjects
76		513e	00000013	А	WaitForSingleObject
77		5154	000000d	А	WriteConsoleW
78		5164	00000009	А	WriteFile
79		516e	0000000c	А	KERNEL32.dll
80		517e	00000012	А	InitCommonControls
81		5192	000000c	А	comctl32.dll
82		51a2	00000012		CommandLineToArgvW
83		51b8	0000000e	А	DragQueryFileW
84		51ca	0000000e	А	SHChangeNotify
85		51dc	00000017	А	SHGetSpecialFolderPathW
86		51f4	0000000b	А	SHELL32.dll



Close





🗾 🚄 🔛									
; Attrib	outes: bp-based frame								
<pre>; DWORDstdcall sub_402CE4(LPVOID lpThreadParameter) sub_402CE4 proc near</pre>									
lpThreadParameter= dword ptr 8									
push	ebp								
mov	ebp, esp								
	eax, [ebp+lpThreadParameter]								
	dword ptr [eax+4] ; lpBuffer								
	dword ptr [eax] ; nBufferLength								
call	GetLogicalDriveStringsW								
рор	ebp								
retn	4								
sub_4020	E4 endp								



mov push <mark>call</mark> push push	<pre>ebp, esp esp, 0FFFFFF8h [ebp+var_4], 0 [ebp+pszPath] ; pszPath PathFindFileNameW 0FFFFFFFFh eax sub_40103C</pre>	
_		
	<pre>push 80h ; dwFileAttributes push [ebp+pszPath] ; lpFileName call SetFileAttributesW push [ebp+pszPath] ; lpFileName call DeleteFileW mov [ebp+var_4], eax</pre>	
	↓ ↓ □ c_40284D:	

Following, the evidence of use of the library **AdvAPI32.dll** for the reasons mentioned before. It is fundamental to highlight the presence of the three main functions of the MD5 digest phase for the README file:

MD5Init -> Initialize the MD5 digesting context

MD5Update -> Partially generate the digest by taking in consideration some bytes pointed from the input and it updates the hashing digest context

MD5Final -> Concludes the MD5 digesting generation



Strir	ngs				- 🗆 X
Filter			ANSI		TITES Unicode C Strings 5 Save Search
	Offset	•		Туре	String
97		526a	00000007		wesrchr
98		5272	0000000a		msvcrt.dll
99		5280	00000016		ConvertSidToStringSidW
100		529a	80000008	Α	MD5Final
101		52a6	00000007		MD5Init
102		52b0	00000009	A	MD5Update
103		52bc	000000f	А	RegCreateKeyExW
104		52ce	0000000d		RegDeleteKeyW
105		52de	0000000e		RegSetValueExW
106		52ee	000000c		AdvAPI32.dll
107		52fe	00000007		NtClose
108		5308	00000010		NtDuplicateToken
109		531c	000000d		NtOpenProcess
110		532c	00000012		NtOpenProcessToken
111		5342	00000017		NtQueryInformationToken
112		535c	00000018		NtQuerySystemInformation
113		5378	00000017		NtSetInformationProcess
114		5392	00000016		NtSetInformationThread
115		53ac	00000011		NtTerminateThread
116		53c0	00000012		RtlAdjustPrivilege
117		53d6	0000000f		RtIAllocateHeap
118		53e8	b0000000		RtlCreateHeap
119		53f8	00000018	А	RtlDeleteCriticalSection
					Close

🚺 🚄 🚺	¥	🚺 🚄 [
oush Lea oush call call oush Lea oush call call	0 100h eax, [ebp+pszPath] eax 80000000h ds:dword_407F10 0 100h eax, [ebx+2] eax 80000000h ds:dword_407F10 0	loc_40 lea push	ath] ; lpSubKey ; hKey ; lpSubKey ; hKey ; lpSubKey ; hKey
push <mark>call</mark> jmp	100h ebx 8000000h ds:dword_407F10 short loc_402A59		



```
; int __stdcall sub_402600(wchar_t *Buffer)
sub_402600 proc near
var_68= byte ptr -68h
var_10= byte ptr -10h
Buffer= dword ptr 8
push
           ebp
          ebp, esp
esp, ØFFFFFF98h
mov
add
push
           ebx
           eax, [ebp+var_68]
lea
push
<mark>call</mark>
           eax
          MD5Init
mov
           ecx, ds:dword_407704
push
lea
          80h
          eax, [ecx+80h]
push
          eax
           eax, [ebp+var_68]
lea
push
           eax
call
          MD5Update
lea
           eax, [ebp+var_68]
push
           eax
<mark>call</mark>
lea
          MD5Final
          ebx, [ebp+var_10]
eax, byte ptr [ebx+0Fh]
movzx
```

Desktop wallpaper and the icons are restored by manipulating the registry keys associated to them:

```
call
        DeleteFileW
lea
        eax, [ebp+pszPath]
push
                        ; pszPath
        eax
call
        PathFindExtensionW
        offset aIco ; ".ico"
push
push
        eax
                        ; Destination
call
        wcscpy
add
        esp, 8
lea
        eax, [ebp+pszPath]
                        ; lpFileName
push
        eax
call
        DeleteFileW
lea
        eax, [ebx+2]
push
        eax
                        ; Source
        eax, [ebp+pszPath]
lea
push
        eax
                        ; Destination
call
        wcscpy
add
        esp, 8
        offset aDefaulticon ; "\\DefaultIcon"
push
lea
        eax, [ebp+pszPath]
                        ; Destination
push
        eax
call
        wcscat
add
        esp, 8
        offset ModuleName ; "advapi32.dll"
push
call
        GetModuleHandleW
        offset ProcName ; "RegDeleteKeyExW"
push
                        ; hModule
push
        eax
call
        GetProcAddress
        ds:dword 407F10, eax
mov
```



🚺 🚄 🖼

loc_40		h7								
	eax, [ebp+pszPath] eax : Destination									
push		,								
· · · · · · · · · · · · · · · · · · ·	ds:dword_408084	; int								
	sub_402858	1.1								
	eax, [ebp+pszPat	nj								
push										
	sub_401574									
		anelDe ; "Control Panel\\Desktop"								
	eax, [ebp+pszPat	-								
push		; Destination								
	wcscat									
	esp, 8									
mov	[ebp+phkResult],	0								
push	0	; lpdwDisposition								
lea	eax, [ebp+phkRes	ult]								
push	eax	; phkResult								
		; lpSecurityAttributes								
push	20106h	; samDesired								
push	0	; dwOptions								
push	0	; lpClass								
push	0	; Reserved								
lea	eax, [ebp+pszPat	h]								
push	eax	; lpSubKey								
push	8000003h	; hKey								

h





🗾 🚄 [S	
push	0 ; fCreate	
push	23h ; '#' ; csidl	
lea	eax, [ebp+pszPath]	
	eax ; pszPath	
push	0 ; hwnd	
call	SHGetSpecialFolderPathW	
	eax, [ebp+pszPath]	
push	eax	
call	sub 401574	
mov	ebx, [ebp+lpSubKey]	
lea	eax, [ebx+2]	
push	eax ; Source	
lea	eax, [ebp+pszPath]	
push	eax ; Destination	
call	wcscat	
add	esp, 8	
	offset Source ; ".bmp"	
lea	eax, [ebp+pszPath]	
push	eax ; Destination	
call	wcscat	
add	esp, 8	
	eax, [ebp+pszPath]	
push	eax ; lpFileName	
call	DeleteFileW	

A loop on 4 elements is performed for the RGB settings and the function SetSysColors is called.

This function can be executed to do the colour change of a graphic element:

loc_40	2ADD:	
nov	[ebp+aElements], 1	
xor	eax, eax	
mov	ah, 0A5h	
mov	al, 6Eh ; 'n'	
	eax, 8	
mov	al, 3Ah ; ':'	
mov	[ebp+aRgbValues], eax	
lea	eax, [ebp+aRgbValues]	
push	eax ; lpaRgbValues	
lea	<pre>eax, [ebp+aElements]</pre>	
push	eax ; lpaElements	
push	1 ; cElements	
call	SetSysColors	
mov	ebx, 2	
call	sub_401488	
cmp	eax, 34h ; '4'	
jz	short loc 402B13	



Strings				–
ilter		ANSI		🔻 🔲 UTF8 📕 Unicode 📕 C Strings 5 🗘 Save Search
	Offset 👻	Size T	ype	String
112		00000018 A		NtQuerySystemInformation
113	5378	00000017 A		NtSetInformationProcess
114	5392	00000016 A		NtSetInformationThread
115	53ac	00000011 A		NtTerminateThread
116	53c0	00000012 A		RtlAdjustPrivilege
117	53d6	0000000f A		RtIAllocateHeap
118	53e8	A b0000000		RtlCreateHeap
119	53f8	00000018 A		RtIDeleteCriticalSection
120	5414	0000000e A		RtIDestroyHeap
121	5426	00000017 A		RtlEnterCriticalSection
122	5440	0000000ь А		RtlFreeHeap
123	544e	0000001c A		RtlInitializeCriticalSection
124	546e	00000017 A		RtlLeaveCriticalSection
125	5488	00000011 A		RtIReAllocateHeap
126	549a	00000009 A		ntdll.dll
127	54a6	0000000ь А		PathAppendW
128	54b4	0000000f A		PathFileExistsW
129	54c6	00000012 A		PathFindExtensionW
130	54dc	00000011 A		PathFindFileNameW
131	54f0	00000015 A		PathlsDirectoryEmptyW
132	5508	00000010 A		PathlsDirectoryW
133		00000012 A		PathlsNetworkPathW
134	5532	00000013 A		PathRemoveFileSpecW
				Close

The analyzed decryptor performs, after creating the thread for efficiency scope, the subroutine of decryption checking if the RSA key is invalid or corrupted. The debugging string *"file rsa key not valid"* is indeed present in this case.

De Stri	ngs				_		\times
Filter			ANSI		▼ UTF8 Unicode C Strings 5 \$ Save	Searc	h
	Offset	•		Туре	String		•
131		54f0	00000015		PathlsDirectoryEmptyW		
132		5508	00000010		PathlsDirectoryW		
133			00000012		PathlsNetworkPathW		
134			00000013		PathRemoveFileSpecW		
135		5546	000000b		SHLWAPI.dll		
136		5554	00000013		WNetAddConnection2W		
137		556a	00000015		WNetGetUniversalNameW		
138		5580	00000007		mpr.dll		
139		5600	0000000a		NTUSER.DAT		
140		5630	0000000a		file empty		
141		5648	0000000e		file not found		
142		5668	0000001d		file not encrypted or damaged		
143		56a4	00000016		file rsa key not valid		
144		5728	00000032		{%08X-%04X-%04X-%02X%02X-%02X%02X%02X%02X%02X%02X}		
145		5790	b0000000		%s.README.txt		
146		57c4	000000c		\DefaultIcon		
147		57e0	000000c		advapi32.dll		
148		57fc	000000f		RegDeleteKeyExW		
149		580a	00000016		WControl Panel\Desktop		
150		5838	0000009		WallPaper		
151		584c	00000017		LockBit Black Decryptor		
152		59d6	00000013		Decrypting Finished		
153		5a06	00000018	U	Press Anv Kev To Exit		•
						Close	



```
push
        eax
movzx
        eax, byte ptr [ebx+0Bh]
push
        eax
        eax, byte ptr [ebx+0Ah]
movzx
push
        eax
        eax, byte ptr [ebx+9]
movzx
push
        eax
movzx
        eax, byte ptr [ebx+8]
push
        eax
        small 0
push
        small word ptr [ebx+6]
push
push
        small Ø
push
        small word ptr [ebx+4]
        dword ptr [ebx] ; Format
offset a08x04x04x02x02 ; "{%08X-%04X-%04X-%02X%02X%02X%02X%02X%0"...
push
push
push
        [ebp+Buffer]
                       ; Buffer
call
        swprintf
        esp, 34h
add
        ebx
рор
        esp, ebp
mov
рор
        ebp
retn
        4
sub_402600 endp
```

When the MD5Init function is executed, an empty buffer is used as parameter to be filled with the hash digest when it is actually executed.

 003F2 	003F260A 50 003F260B E8 822B0000 003F2610 8B0D 04773F00 003F2616 68 80000000 003F2618 8D81 80000000 003F2622 8D45 98 003F2625 50 003F2626 E8 6D2B0000 003F2627 S0 003F2628 S045 98 003F2629 S045 98 003F2626 E8 6D2B0000 003F2627 S0 003F2628 S045 98 003F2629 S0								n pi pi poi po	push ebx lea eax,dword ptr ss:[ebp-68] push eax call <jmp.&mdsinit> mov ecx,dword ptr ds:[3F7704] push 80 lea eax,dword ptr ds:[ecx+80] push eax lea eax,dword ptr ss:[ebp-68] push eax call <jmp.&mdsupdate> lea eax,dword ptr ss:[ebp-68] push eax call <jmp.&mdsfinal> lea ebx,dword ptr ss:[ebp-10]</jmp.&mdsfinal></jmp.&mdsupdate></jmp.&mdsinit>						
Address 003F7704 003F7714	Hex 00 00		00		00	00	00	00	00	00	00	00	00	00	00	ASCII
003F7724 003F7734 003F7744	00 00	00	00 00 00	00	00	00	00	00	00 00 00	00	00	00	00	00 00 00	00	
003F7754 003F7764 003F7774	00 00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
003F7784 003F7794 003F7794 003F7784	00 00 00 00 00 00	00 00 00	00 00 00 00 00	00 00 00	00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	

Subsequently, the general register ECX is used for the execution content and also the EAX register to perform push instructions for the output data of the digest operation:



	003F260B	E8 822B0000	call <jmp.&md5init></jmp.&md5init>
	003F2610	8B0D 04773F00	mov ecx, dword ptr ds: [3F7704]
	003F2616	68 80000000	push 80
	003F261B	SD81 80000000	lea eax,dword ptr ds:[ecx+80]
	003F2621	50	push eax
	003F2622	8D45 98	lea eax,dword ptr ss:[ebp-68]
	003F2625	50	push eax
	003F2626	E8 6D2B0000	call <jmp.&md5update></jmp.&md5update>
	003F262B	8D45 98	lea eax,dword ptr ss:[ebp-68]
	003F262E	50	push eax
	003F262F	E8 582B0000	call <jmp.&md5final></jmp.&md5final>
	003F2634	8D5D F0	lea ebx,dword ptr ss:[ebp-10]
	003F2637	0FB643 0F	movzx eax, byte ptr ds:[ebx+F]
	003F263B	50	push eax
	003F263C	0FB643 0E	movzx eax, byte ptr ds:[ebx+E]
	003F2640	50	push eax
	003F2641	0FB643 0D	movzx eax, byte ptr ds:[ebx+D]
	003F2645	50	push eax
	003F2646	0FB643 0C	movzx eax, byte ptr ds:[ebx+C]
	003F264A	50	push eax
•	003F264B	0FB643 0B	movzx eax,byte ptr ds:[ebx+B]
•	003F264F	50	push eax
•	003F2650	0FB643 0A	movzx eax,byte ptr ds:[ebx+A]
•	003F2654	50	push eax
•	003F2655	0FB643 09	movzx eax,byte ptr ds:[ebx+9]
•	003F2659	50	push eax
•	003F265A	0FB643 08	movzx eax,byte ptr ds:[ebx+8]
•	003F265E	50	push eax
•	003F265F	66:6A 00	push 0
•	003F2662	66:FF73 06	push word ptr ds:[ebx+6]

Then, the MD5Final function is used to fill the general ID and the output of the MD5 hash digest:

•	push 0 push word ptr ds:[ebx+6]	
	push 0	
	push word ptr ds:[ebx+4]	
•	push dword ptr ds:[ebx]	
	push lbb_decryptor.3F7128	3F7128:L"{%08X-%04X-%04X-%02X%02X-%02X%02X%02X
•	push dword ptr ss:[ebp+8]	
	call <jmp.&swprintf></jmp.&swprintf>	
•	add esp,34	
•	pop ebx	
•	mov esp,ebp	
•	pop ebp	
•	ret 4	

So, it is observable that during the decryption phase, the decryptor tool reads the %s.README.txt file: our tests in lab show that also in absence of the README.txt file, the decryptor manages anyway to decrypt encrypted files. This indicates how in the decryptor tool there are all the information needed to decrypt encrypted files. Each decryptor in fact is generated contextually to its ransomware, starting from a couple of public and private key: the ransomware uses the public key to encrypt, meanwhile the decryptor tool will use the related private key for the decryption.

Continuing the decryptor analysis, it is noteable that the execution of the decryption tool is traced also with a log file: **trial_dec.log:**



Di Strings

 \Box \times

_

Filter					🔻 📃 UTF8 📕 Unicode 📕 C Strings 5 💠 🛛 Save Search
	Offset	•	Size	Туре	String
146		57c4	000000c l	J	\DefaultIcon
147		57e0	0000000c l	J	advapi32.dll
148		57fc	000000f 4	A	RegDeleteKeyExW
149		580a	00000016 ^l	J	WControl Panel\Desktop
150		5838	00000009 ^l	J	WallPaper
151		584c	00000017 ^l	J	LockBit Black Decryptor
152		59d6	00000013 ^l	J	Decrypting Finished
153		5a06	00000018 ^l	J	Press Any Key To Exit
154		5a38	b0000000 l	J	trial_dec.log
155		5a54	00000017 ^l	J	LockBit Black Decryptor
156		5a84	0000001b ^l	J	Decrypt All Encrypted Files
157		5abc	00000006 ^l	J	user32
158		5acc	00000019	A	ChangeWindowMessageFilter
159		5ae8	0000001e l	J	Network Connection Unavailable
160		5b28	00000007 ^l	J	[ERROR]
161		5b38	0000001c l	J	Finding Files In Progress
162		5b74	0000001e l	J	Decrypting File In Progress
163		5bb4	0000001f ^l	J	Decrypting Files In Progress
164		5bf4	0000001b ^l	J	Decrypt All Encrypted Files
165		5c2c	0000001b ^l	J	Decrypt All Encrypted Files
166		5c64	0000001c ^l	J	Finding Files In Progress
167		5ca0	00000009 l	J	ntdll.dll
168		5cb4	00000019	A	NtSetThreadExecutionState
					Close

173	86ac	00000006 U	Tahoma
174	86d8	00000007 U	IDC_BTN
175	8748	00000013 U	All Encrypted Files
176	8790	00000013 U	All Decrypted Files
177	d901	00000015 A	20202\$2*20262<2B2H2N2



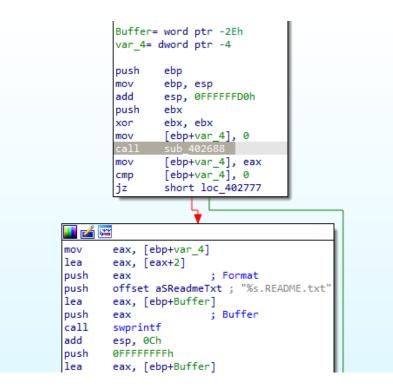
loc_4022A3: cmp [ebp+arg_4], 3 jnz short loc_4022AE	
t aFileNotEncrypt ; "file not encrypted or damaged" eax, offset aFileRsaKeyNotV ; "file rsa key not va 4022AE	lid"
4022AE	

It seems that during the execution the verdicts are written on files for the various operations, with *"True"* or *"False"* depending on the specific cases.

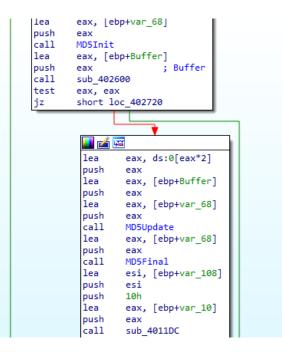
	*	* * * * *
🗾 🚄 🚺		
push push push call add mov jmp	<pre>[ebp+arg_C] [ebp+Format] ; Format offset aTrueSS ; "true %s %s\n" offset Buffer ; Buffer swprintf esp, 10h ecx, eax short loc_4022C6</pre>	<pre>loc_4022AE: push eax push [ebp+Format] ; Format push offset aFalseSS ; "false %s %s\n" push offset Buffer ; Buffer call swprintf add esp, 10h mov ecx, eax</pre>
	<pre> loc_4022C6: push offset Buffer push offset Buffer call sub_401094 </pre>	
	push 0 lea eax, [ebp+Numb push eax push ecx push offset Buffer push [ebp+hFile] call WriteFile	

Following are the details of the execution flow which is related to the function sub_402688 for the phase of README.TXT ID hashing.



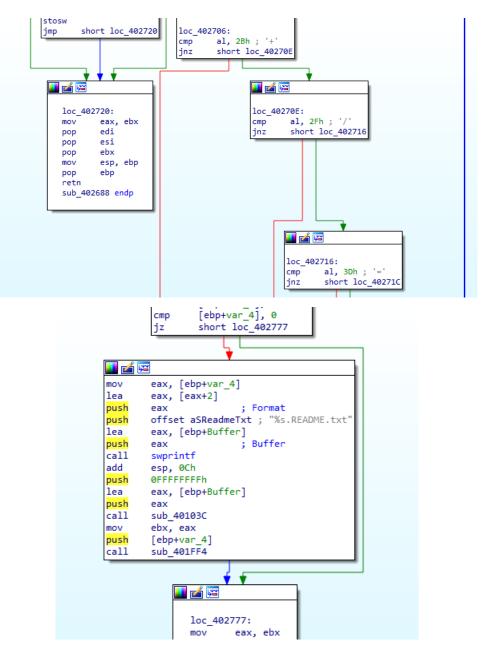


Here are the instances of the creation of the contexts to proceed with the MD5 hashing digest in the context of disassemblying:



Subsequently it seems that some compare, checking and validation execution are performed:

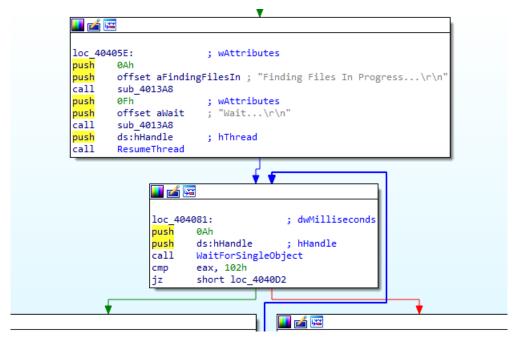




Following, instead, the initial juncture of the execution in which the phase of files search and decryption of them begins (please note that in the files enumeration loop the wildcard * is used):



<pre>mov eop, esp sub esp, 21Ch push ebx push edx push edx push edi push offset ConsoleTitle ; "LockBit Black Decryptor" call SetConsoleTitleW push OFFFFFFF6h ; nStdHandle call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h ; 'c' ; uFlags push 0 ; cx push 0 ; cx push 0 ; cx push 0 ; X push 0 ; StdHandle call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F4C, 0 mov ds:dword_407F4C, 0 push 0 ; lpThreadId push 0 ; lpThreadId push 4 ; dwCreationFlags</pre>		
<pre>push ebx push ecx push ecx push edx push edi push offset ConsoleTitle ; "LockBit Black Decryptor" call SetConsoleTitleW push 0FFFFFF6h ; nStdHandle call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h ; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; cx push 0 ; X push 0 ; StwindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>		A REAL PROPERTY OF A REA
<pre>push ecx push edx push edi push edi push offset ConsoleTitle ; "LockBit Black Decryptor" call SetConsoleTitleW push 0FFFFFF6h ; nStdHandle call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h ; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; cx push 0 ; X push 0 ; Stdword_407FFFFFF ; hWndInsertAfter push eax ; hWnd call SetWindowPos mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>		1.4
<pre>push edx push esi push edi push offset ConsoleTitle ; "LockBit Black Decryptor" call SetConsoleTitleW push 0FFFFFF6h ; nStdHandle call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h ; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; cx push 0 ; X push 0 ; Stdword_407FFFFFF ; hWndInsertAfter push eax ; hWnd call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>	push	ebx
<pre>push esi push edi push offset ConsoleTitle ; "LockBit Black Decryptor" call SetConsoleTitleW push 0FFFFFF6h ; nStdHandle call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h ; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; cx push 0 ; X push 0 ; SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>		
<pre>push edi push offset ConsoleTitle ; "LockBit Black Decryptor" call SetConsoleTitleW push 0FFFFFF6h ; nStdHandle call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h ; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; tx push 0 ; X push 0 ; SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>	push	edx
pushoffset ConsoleTitle ; "LockBit Black Decryptor"callSetConsoleTitleWpush0FFFFFF6h ; nStdHandlecallGetStdHandlemov[ebp+hConsoleInput], eaxcallGetConsoleWindowpush63h ; 'c' ; uFlagspush0 ; cypush0 ; cxpush0 ; Xpush0 ; Xpush0 ; Xpush0 ; StimtortAfterpushSetWindowPosmovds:dword_407F4C, 0movds:dword_407F50, 0movds:dword_0push0 ; lpThreadId	push	esi
<pre>call SetConsoleTitleW push 0FFFFFF6h ; nStdHandle call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h ; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; X push 0 ; X push 0 ; X push 0FFFFFFFh ; hWndInsertAfter push eax ; hWnd call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>	push	edi
pushOFFFFFFF6h; nStdHandlecallGetStdHandlemov[ebp+hConsoleInput], eaxcallGetConsoleWindowpush63h ; 'c'; uFlagspush0; cypush0; cxpush0; Xpush0; Xpush0; Xpush0; Kpush0; MundInsertAfterpusheax; hWndcallSetWindowPosmovds:dword_407F4C, 0movds:dword_407F50, 0movds:uValue, 0push0; lpThreadId	push	offset ConsoleTitle ; "LockBit Black Decryptor"
<pre>call GetStdHandle mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; X push 0 ; X push 0FFFFFFFh ; hWndInsertAfter push eax ; hWnd call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>	call	SetConsoleTitleW
<pre>mov [ebp+hConsoleInput], eax call GetConsoleWindow push 63h; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; X push 0 ; X push 0 ; X push 0FFFFFFFh ; hWndInsertAfter push eax ; hWnd call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId</pre>	push	0FFFFFF6h ; nStdHandle
call GetConsoleWindow push 63h; 'c' ; uFlags push 0 ; cy push 0 ; cx push 0 ; X push 0 ; X push 0FFFFFFFh ; hWndInsertAfter push eax ; hWnd call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId	call	GetStdHandle
push63h ; 'c'; uFlagspush0; cypush0; cxpush0; Ypush0; Xpush0FFFFFFFh; hWndInsertAfterpusheax; hWndcallSetWindowPosmovds:dword_407F4C, 0movds:dword_407F50, 0movds:uValue, 0push0; lpThreadId	mov	[ebp+hConsoleInput], eax
push0; cypush0; cxpush0; Ypush0; Xpush0FFFFFFF; hWndInsertAfterpusheax; hWndcallSetWindowPosmovds:dword_407F4C, 0movds:dword_407F50, 0movds:uValue, 0push0; lpThreadId	call	GetConsoleWindow
push0; cxpush0; Ypush0; Xpush0FFFFFFFh; hWndInsertAfterpusheax; hWndcallSetWindowPosmovds:dword_407F4C, 0movds:dword_407F50, 0movds:dword_407F50, 0movds:uValue, 0push0; lpThreadId	push	63h ; 'c' ; uFlags
push 0 ; Y push 0 ; X push 0FFFFFFFh ; hWndInsertAfter push eax ; hWnd call SetWindowPos	push	0 ; cy
push 0 ; X push 0FFFFFFFh ; hWndInsertAfter push eax ; hWnd call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId	push	0 ; cx
push0FFFFFFFh; hWndInsertAfterpusheax; hWndcallSetWindowPosmovds:dword_407F4C, 0movds:dword_407F50, 0movds:uValue, 0push0; lpThreadId	push	0 ; Y
pusheax; hWndcallSetWindowPosmovds:dword_407F4C, 0movds:dword_407F50, 0movds:uValue, 0push0; lpThreadId	push	0 ; X
call SetWindowPos mov ds:dword_407F4C, 0 mov ds:dword_407F50, 0 mov ds:uValue, 0 push 0 ; lpThreadId	push	0FFFFFFFh ; hWndInsertAfter
movds:dword_407F4C, 0movds:dword_407F50, 0movds:uValue, 0push0; lpThreadId	push	eax ; hWnd
mov ds:dword_407F50,0 mov ds:uValue,0 push 0 ; lpThreadId	call	SetWindowPos
mov ds:uValue, 0 push 0 ; lpThreadId	mov	ds:dword_407F4C, 0
push 0 ; lpThreadId	mov	ds:dword_407F50, 0
, , , , , , , , , , , , , , , , , , , ,	mov	ds:uValue, 0
push 4 ; dwCreationFlags	push	0 ; lpThreadId
	push	4 ; dwCreationFlags
push 0 ; lpParameter	push	0 ; 1pParameter
push offset sub 403FA8 : lpStartAddress	bush	offset sub 403FA8 : lpStartAddress





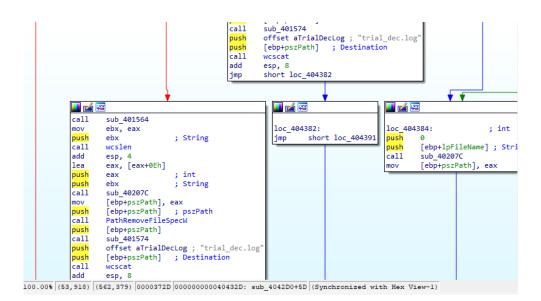
🗾 🚄 🔛	3				
loc 403	DF5:	; String			
	[ebp+String]				
call	sub 403BE4				
mov	[ebp+var_14], ea	x			
push	[ebp+String]				
call	sub 401FF4				
push	[ebp+dwAdditiona	lFlags] ; dwAddit	ionalFlags		
push		; lpSearchFilter	-		
push	0	; fSearchOp			
lea	eax, [ebp+FindFi	leData]			
push	eax	; lpFindFileData			
push		; fInfoLevelId			
push	[ebp+var_14]	; lpFileName			
call	FindFirstFileExW				
mov	[ebp+hFindFile],				
cmp	[ebp+hFindFile],	ØFFFFFFFh			
jz	loc_403F58				
				-	
				•	
			🚺 🚄 🔛		
				2Ah ; '*'	; Ch
				[ebp+var 14]	· ·
				wcsrchr	,
				esp, 8	

Based on the verdict of the searching of the files encrypted by LockBit Black is executed the push of the offset "aFoundUFileS_0" or "aFoundUFileS":

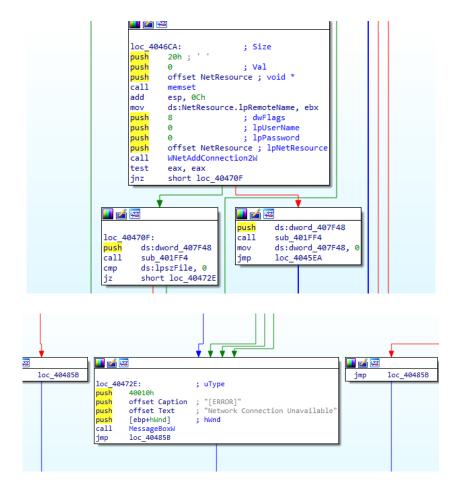
jz short loc	_4040D2
•	
🔜 🚄 🖼	
<pre>loc_4040D2: ; Format push ds:uValue push offset aFoundUFileS_0 ; "Found %u File(s)\r lea eax, [ebp+Buffer] push eax ; Buffer call swprintf add esp, 0Ch push 0Fh ; wAttributes lea eax, [ebp+Buffer]</pre>	push ds:hHandle ; hObject call CloseHandle push ds:uValue ; Format
push eax ; String call sub_4013A8 push [ebp+hConsoleInput]; hConsoleInput call FlushConsoleInputBuffer jmp loc_404081	<pre>push eax ; String call sub_4013A8 push [ebp+hConsoleInput]; hConsoleInput call FlushConsoleInputBuffer jmp short loc_404107</pre>
□	uValue, 0

Here are the details of the logging in the trial_dec.log file:





In case the executable doesn't detect network connectivity the debugging tag "[ERROR]" is written:





Following, the details of the most suspicious indicators of the executable in question:

indicator (39)	detail
The file references string(s)	type: blacklist, count: 30
The file execution privilege has been found	level: administrator
The file imports symbol(s)	type: blacklist, count: 29
The file references a string with a suspicious size	size: 1090 bytes
The time-stamp of a directory is suspicious	directory: debug, stamp: Thu Jul 14
The file references blacklist library(ies)	count: 1
The file references a group of API	type: windowing, count: 10
The file references a group of API	type: resource, count: 2
The file references a group of API	type: file, count: 44
The file references a group of API	type: execution, count: 31
The file references a group of API	type: console, count: 16
The file references a group of API	type: storage, count: 4
The file references a group of API	type: dynamic-library, count: 4
The file references a group of API	type: memory, count: 15
The file references a group of API	type: synchronization, count: 15
The file references a group of API	type: shell, count: 3
The file references a group of API	type: reckoning, count: 4
The file references a group of API	type: security, count: 8
The file references a group of API	type: cryptography, count: 8
The file references a group of API	type: registry, count: 8
The file references a group of API	type: network, count: 6
The file references a group of hint	type: file, count: 17
The file references a group of hint	type: function, count: 69
The file references a group of hint	type: size, count: 2
The file references a group of hint	type: format-string, count: 6
The file references a group of hint	type: utility, count: 3

Here's the Portable Executable sections, where it is highlight how the entropy of the .text section is **6.143**: this coefficient is quite low, so the analyzed decryptor doesn't present particular occultation techniques of the executable .text section.

Here are further details:



property	value	value	value
name	.text	.rdata	.data
md5	1BE6DF46D84967F7EA0AFD6	1897CB665DDE48474BE633D	05FD5450D3417D244B19AB2
entropy	6.143	4.924	3.129
file-ratio (98.17%)	31.19 %	6.42 %	3.67 %
raw-address	0x00000400	0x00004800	0x00005600
raw-size (54784 bytes)	0x00004400 (17408 bytes)	0x00000E00 (3584 bytes)	0x00000800 (2048 bytes)
virtual-address	0x00401000	0x00406000	0x00407000
virtual-size (56058 bytes)	0x00004252 (16978 bytes)	0x00000D88 (3464 bytes)	0x00001088 (4232 bytes)
entry-point	0x00004C72	-	-
characteristics	0x60000020	0x40000040	0xC0000040
writable	-	-	x
executable	x	-	-
shareable	-	-	-
discardable	-	-	-
initialized-data	-	x	x
uninitialized-data	-	-	-
unreadable	-	-	-
self-modifying	-	-	-
virtualized	-	-	-
file	n/a	n/a	n/a

🛍 Stud_PE editing : "LBB_Decryptor.exe" - [32bit app] — 🗌 🗙

c:\users\ieuser\desktop\lockbit3-decryptor\decryptor_all_pc\lbb_decryptor.exe							
 Dos P Sections fx Functions Rs Resources P Signature P Procs P Opti < . 							
		Luci Ioc					
No	Name	VirtualSize	VirtualOffset	RawSize	RawOffset	Characteri	
Ep 01	.text	00004252	00001000	00004400	00000400	60000020	
02	.rdata	00000D88	00006000	00000E00	00004800	40000040	
03	.data	00001088	00007000	00000800	00005600	C0000040	
04	.data	00002000	00009000	00002000	00005E00	C0000040	
05	.rsrc	00005780	0000B000	00005800	00007E00	40000040	
06 📃	.reloc	00000318	00011000	00000400	0000D600	42000040	

Here further details related to the imports performed and the number of functions associated to them:



library (9)	blacklist (1)	type (1)	functions (107)	description
user32.dll	- 1	implicit	14	Multi-User Windows USER API Client DLI
kernel32.dll	-	implicit	41	Windows NT BASE API Client DLL
comctl32.dll	-	implicit	1	Common Controls Library
shell32.dll	-	implicit	4	Windows Shell Common DII
msvcrt.dll	-	implicit	11	Windows NT CRT DLL
advapi32.dll	-	implicit	7	Advanced Windows 32 Base API
ntdll.dll	-	implicit	19	NT Layer DLL
shlwapi.dll	-	implicit	8	Shell Light-weight Utility Library
mpr.dll	x	implicit	2	Multiple Provider Router DLL

functions (107)	blacklist (29)	type (1)	ordinal (0)	library (9)
EnableWindow	-	implicit	-	user32.dll
DialogBoxParamW	-	implicit	-	user32.dll
SetDigitemInt	-	implicit	-	user32.dll
SetSysColors	-	implicit	-	user32.dll
SetTimer	-	implicit	-	user32.dll
<u>SetWindowPos</u>	-	implicit	-	user32.dll
SetWindowTextW	-	implicit	-	user32.dll
<u>SystemParametersInfoW</u>	x	implicit	-	user32.dll
EndDialog	-	implicit	-	user32.dll
<u>SendMessageW</u>	-	implicit	-	user32.dll
MessageBoxW	-	implicit	-	user32.dll
<u>LoadlconW</u>	-	implicit	-	user32.dll
KillTimer	-	implicit	-	user32.dll
GetDigitem	-	implicit	-	user32.dll
WriteFile	x	implicit	-	kernel32.dll
WriteConsoleW	-	implicit	-	kernel32.dll
WaitForSingleObject	-	implicit	-	kernel32.dll
WaitForMultipleObjects	-	implicit	-	kernel32.dll
Sleep	-	implicit	-	kernel32.dll
SetThreadPriority	-	implicit	-	kernel32.dll
<u>SetFilePointerEx</u>	-	implicit	-	kernel32.dll
<u>CloseHandle</u>	-	implicit	-	kernel32.dll
<u>CreateFileW</u>	-	implicit	-	kernel32.dll
<u>CreateloCompletionPort</u>	-	implicit	-	kernel32.dll
CreateThread	-	implicit	-	kernel32.dll
<u>DeleteFileW</u>	x	implicit	-	kernel32.dll
ExitProcess	-	implicit	-	kernel32.dll
FindClose	-	implicit	-	kernel32.dll
FindFirstFileExW	x	implicit	-	kernel32.dll
FindNextFileW	x	implicit	-	kernel32.dll
FlushConsoleInputBuffer	x	implicit	-	kernel32.dll



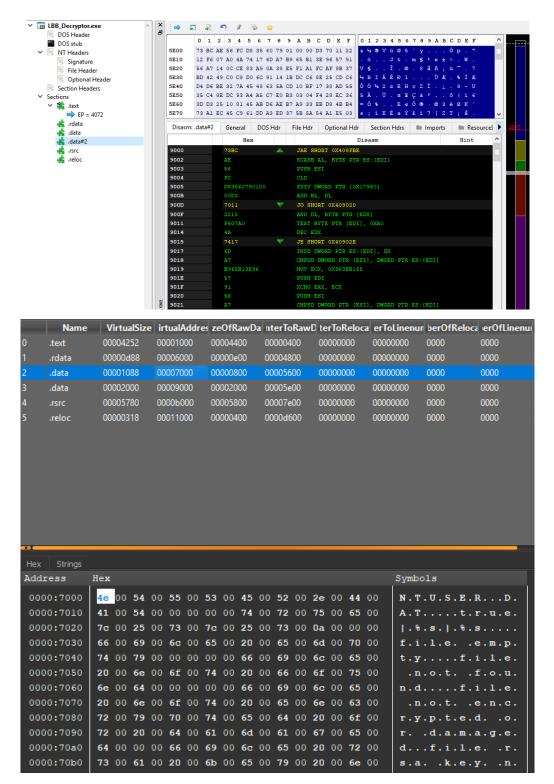
functions (107)	blacklist (29)	type (1)	ordinal (0)	library (9)	
wcscat	-	implicit	-	msvcrt.dll	
wcscpy	-	implicit	-	msvcrt.dll	
MD5Update	x	implicit	-	advapi32.dll	
MD5Init	x	implicit	-	advapi32.dll	
MD5Final	x	implicit	-	advapi32.dll	
<u>ConvertSidToStringSidW</u>	-	implicit	-	advapi32.dll	
RegDeleteKeyW	×	implicit	-	advapi32.dll	
RegSetValueExW	x	implicit	-	advapi32.dll	
RegCreateKeyExW	-	implicit	-	advapi32.dll	
RtIDeleteCriticalSection	-	implicit	-	ntdll.dll	
RtIDestroyHeap	-	implicit	-	ntdll.dll	
<u>RtlCreateHeap</u>	-	implicit	-	ntdll.dll	
<u>RtlFreeHeap</u>	-	implicit	-	ntdll.dll	
RtllnitializeCriticalSection	-	implicit	-	ntdll.dll	
RtlLeaveCriticalSection	-	implicit	-	ntdll.dll	
RtIReAllocateHeap	-	implicit	-	ntdll.dll	
NtClose	-	implicit	-	ntdll.dll	
<u>RtIAllocateHeap</u>	-	implicit	-	ntdll.dll	
RtlAdjustPrivilege	x	implicit	-	ntdll.dll	
NtTerminateThread	-	implicit	-	ntdll.dll	
<u>NtSetInformationThread</u>	x	implicit	-	ntdll.dll	
NtSetInformationProcess	x	implicit	-	ntdll.dll	
NtQuerySystemInformation	x	implicit	-	ntdll.dll	
NtQueryInformationToken	-	implicit	-	ntdll.dll	
<u>NtOpenProcessToken</u>	x	implicit	-	ntdll.dll	
NtOpenProcess	x	implicit	-	ntdll.dll	
NtDuplicateToken	-	implicit	-	ntdll.dll	
RtlEnterCriticalSection	-	implicit	-	ntdll.dll	
PathFindFileNameW	x	implicit	-	shlwapi.dll	
PathlsDirectoryEmptyW	x	implicit	-	shlwapi.dll	
PathFindExtensionW	x	implicit	-	shlwapi.dll	

Here's how the decryptor is in a debugging instance:

CPU - main thread, module LBB_Decryptor					
000744521 • 60 08 PUSH 8 000744523 • 60 00 PUSH 0 000744525 • 60 00 PUSH 0 000744527 • 68 04275000 000744527 • 68 04275000 000744527 • 58 00070000 CALL XUMP.%mor.WhetAddConnec 00074541 • 65C0	tion2W>	Ar94 = 8 Ar93 = 0 Ar92 = 0 Ar91 = L Mpr.WNet		64	▲ Registers (FPU) EAX 010FFF50 ECX 004H4C72 LBB_Decryptor. EDX 004H4C72 LBB_Decryptor. EDX 00EE0000
B06PA4573	I		yptor.00AA1FF4		ESP 816FF98 EBP 816FF14 ESI 00A44C72 LBE_Decryptor.(ModuleEntryPoint) EDI 00A44C72 LBE_Decryptor.(ModuleEntryPoint) EIP 00A44C72 LBE_Decryptor.(ModuleEntryPoint) C 0 ES 0028 32bt 0(FFFFFFF)
		Arg1 = 0 LBB_Decr Type = M Caption Text = "		B_DEFBUTTON1 Unavailable	P 1 CS 0023 25bi 0(FFFFFFF) 0 SS 0025 82bi 0(FFFFFFFF) 2 1 DS 0028 82bi 0(FFFFFFFF) 5 0 FS 0028 82bi 0(FFFFFFFF) 1 0 GS 0028 82bi 0(FFFFFFFF) 0 G ustrict 000002 EROR_SUCCESS 0 0 Luster: 00000246 (NO,NS,E,BE,NS,PE,GE,LE)
00644745 E5 11010800 Jrm 0864458 00644745 E5 11010800 Jrm 05:000 Prm 05:00477401 00644755 E5 44725000 00644755 E5 44725000 00644755 E5 452500 Jrm 08607 Prm 05:0047741,50 00644756 E5 05 00644756 E5 05 00644756 E5 12000 Jrm 0864758 00644756 E5 12000 Jrm 0864758	×	CArg1 = 0 LBB_Decr	yptor.00AA2134		ST0 enpty 0.0 ST1 enpty 0.0 ST3 enpty 0.0 ST3 enpty 0.0 ST4 enpty 0.0 ST5 enpty 0.0 ST5 enpty 0.0 ST6 enpty 0.0 ST7 enpty 0.0
Bitherarch Fras Bitch Pick		_	yptor.00AA20D4		ST7 empty 0.8 3210 ESPU0ZDI FST 0000 Cond 0.0 0 Err 0.0 0.0 0.0 0.0 (GT FCW 027F Prec NERR,S3 Mask 1.1.1.1.1 Last cmnd 0000:00000000
EDI=000A4C72 (LBB_Decryptor. <moduleentrypoint>)</moduleentrypoint>	All Encry	t Black Dec	ryptor 0		× 8 0000000 0000000 0 0000000 0000000 0 0000000 0000000 0 00000000
	All Decry	pted Files	0	Decrypt	t All Encrypted Files
Roddress Hext dump 006na7000 Hext dump 006na7000 Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp Hext dupp Hext dupp Hext dupp Hext dupp 006na7000 Hext dupp	1000000000000000000000000000000000000	the second - XXXXX	010FFF20 010FFF20 010FFF20 010FFF20 010FFF50 010FFF50 010FFF50 010FFF40 010FFF40 010FFF40 010FFF40 010FFF50 010FFF50 010FFF50	00EE0000 € F6E5F6SH + 00000000 00E00000 00E00000 000000000 000000000 000000	0 00 Pointer to next SEH record



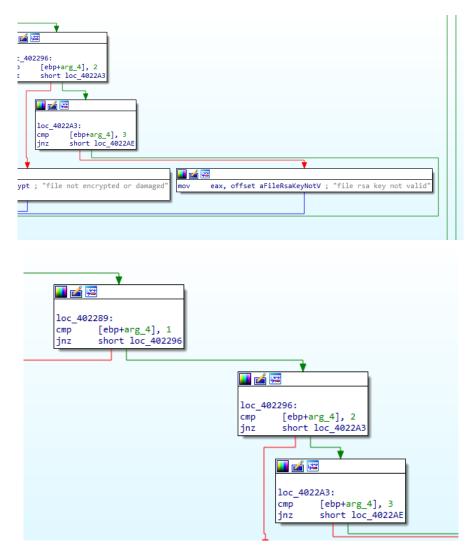
Curiously, the executable has two .data sections. In the second section there are details related to strings used as output during the "compare" instructions as verdicts of the decryption threads:





	Address	-	Size	Туре	String				
1		7000	0000000a	U	NTUSER.DAT				
2		7030	0000000a	U	file empty				
3		7048	0000000e	U	file not found				
4		7068	0000001d	U	file not encrypted or damaged				
5		70a4	00000016	U	file rsa key not valid				
6		7128	0000032	U	{%08X-%04X-%04X-%02X%02X-%02X%02X%02X%02X%02X%02X}				
7		7190	0000000d	U	%s.README.txt				
8		71c4	000000c	U	\DefaultIcon				
9		71e0	000000c	U	advapi32.dll				
.data:	: 004070A : 004070A : 004070D	4	RsaKeyNot	tex	; DATA XREF: sub_402240+69↑o t "UTF-16LE", 'file rsa key not valid',0 gn 4				

After a series of checks ("compare") and conceputal "if", we arrive at the result "file rsa key not found":



Here's the details of an extract of the hexadecimal code of the analyzed executable:

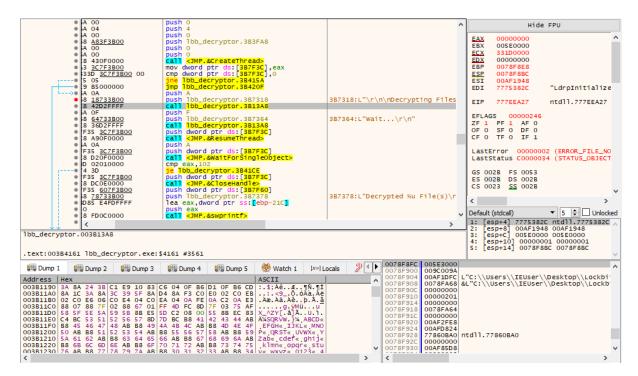


J 🖏	LBB_Decry	ptor.exe				
S 101	VA					
: LBB_Decryptor.exe	RVA					
Dos Header		_	-			
Nt Headers	File Offset					
File Header						
 Optional Header Data Directories [x] 	h 🖬 💼		P 🖬			
Section Headers [x]		·	· ·			
mport Directory	Offset	0 1 2 3	456	789A	BCDEF	Ascii
Resource Directory	00005510	72 65 63 74		7 00 00 61	00 50 61 74 68	rectoryW.a.Path
Relocation Directory	00005520	49 73 4E 65 8B 00 50 61	74 77 6F 7 74 68 52 6	2 6B 50 61 5 6D 6F 76	74 68 57 00 00 65 46 69 6C 65	IsNetworkPathW .PathRemoveFile
Debug Directory	00005540	53 70 65 63	57 00 53 4	8 4C 57 41	50 49 2E 64 6C	SpecW.SHLWAPI.dl
Address Converter	00005550	6C 00 03 00 63 74 69 6F		4 41 64 64 0 22 00 57	43 6F 6E 6E 65 4E 65 74 47 65	1.0.WNetAddConne ction2W.".WNetGe
Dependency Walker	00005570	74 55 6E 69		3 61 6C 4E		tUniversalNameW.
Hex Editor	00005580	6D 70 72 2E 00 00 00 00	64 6C 6C 0 00 00 00 0	IO OO OO OO IO OO OO OO		mpr.dll
Identifier	000055A0	00 00 00 00 00 00 00 00			00 00 00 00 00 00 00 00 00 00	
Import Adder	000055B0	00 00 00 00	00 00 00 0		00 00 00 00 00	
Quick Disassembler Rebuilder	000055C0 000055D0	00 00 00 00 00 00 00 00 00	00 00 00 0	10 00 00 00 10 00 00 00	00 00 00 00 00 00 00 00 00 00	
Resource Editor	000055E0	00 00 00 00	00 00 00 0	0 00 00 00	00 00 00 00 00	
UPX Utility	000055F0 00005600	00 00 00 00 4E 00 54 00	00 00 00 0 0 0 55 00 53 0	IO OO OO OO IO 45 OO 52	00 00 00 00 00 00 2E 00 44 00	N.T.U.S.E.RD.
	00005610	41 00 54 00		0 74 00 72	00 75 00 65 00	A.Tt.r.u.e.
	00005620	7C 00 25 00 66 00 69 00	73 00 7C 0 6C 00 65 0		00 0A 00 00 00 00 6D 00 70 00	.%.s. .%.s f.i.l.ee.m.p.
	00005640	74 00 79 00	00 00 00 0		00 6C 00 65 00	t.yf.i.l.e.
	00005650	20 00 6E 00 6E 00 64 00	6F 00 74 0 00 00 00 0		00 6F 00 75 00 00 6C 00 65 00	n.o.tf.o.u.
	00005660	20 00 6E 00	6F 00 74 0		00 6C 00 65 00 00 6E 00 63 00	n.df.i.l.e.
	00005680	72 00 79 00		0 65 00 64	00 20 00 6F 00	r.y.p.t.e.do.
	00005690 000056A0	72 00 20 00 64 00 00 00	64 00 61 0 66 00 69 0	IO 6D 00 61	00 67 00 65 00 00 20 00 72 00	rd.a.m.a.g.e. df.i.l.er.
	000056B0	73 00 61 00	20 00 6B 0	0 65 00 79	00 20 00 6E 00	s.ak.e.yn.
	000056C0 000056D0	6F 00 74 00 00 00 00 00		10 61 00 6C 10 6C 00 73	00 69 00 64 00 00 65 00 7C 00	o.tv.a.l.i.d. f.a.l.s.e. .
	000056E0	25 00 73 00	7C 00 25 0			%.s. .%.s
	000056F0	11 00 00 00		0 24 00 00	00 1D 00 00 00	0\$
	00005700	OE OO OO OO OD OO OO OO	05 00 00 0		00 1C 00 00 00 00 0B 00 00 00	00
	00005720	09 00 00 00	00 00 00 0	0 7B 00 25		

LBB_Decryptor.exe					
Member	Offset	Size	Value	Meaning	
Magic	0000098	Word	010B	PE32	
MajorLinkerVersion	0000009A	Byte	0E		
MinorLinkerVersion	0000009B	Byte	0C		
SizeOfCode	000009C	Dword	00004400		
SizeOfInitializedData	000000A0	Dword	00009C00		
SizeOfUninitializedData	000000A4	Dword	00000000		
AddressOfEntryPoint	000000A8	Dword	00004C72	.text	
BaseOfCode	000000AC	Dword	00001000		
BaseOfData	000000B0	Dword	00006000		
ImageBase	000000B4	Dword	00400000		
SectionAlignment	000000B8	Dword	00001000		
FileAlignment	000000BC	Dword	00000200		
MajorOperatingSystemVers	000000C0	Word	0005		
MinorOperatingSystemVers	000000C2	Word	0001		
MajorImageVersion	000000C4	Word	0000		
MinorlmageVersion	00000C6	Word	0000		
MajorSubsystemVersion	000000C8	Word	0005		
MinorSubsystemVersion	000000CA	Word	0001		
Win32VersionValue	000000CC	Dword	00000000		
SizeOfImage	000000D0	Dword	00012000		
SizeOfHeaders	000000D4	Dword	00000400		
CheckSum	000000D8	Dword	0000FFC3		



Following, the details related to the phase of the execution of the decryption thread where is possible to notice references to the management of registry keys for their disinfection:



int32_t SetFileAttributesW = 0x68dc;

```
void fun_40513e(struct s0* a1, struct s0* a2, struct s0* a3) {
goto SetFileAttributesW;
}
```

```
int32_t DeleteFileW = 0x66ba;
```

struct s0* fun_405090(struct s0* a1, struct s0* a2, struct s0* a3, struct s0* a4, struct s0* a5, struct s0* a6, struct s0* a7) {
 goto DeleteFileW;

int32_t SHGetSpecialFolderPathW = 0x69da;

```
void fun_405180(struct s0* a1, struct s0* a2, struct s0* a3, struct s0* a4) {
  goto SHGetSpecialFolderPathW;
}
void fun_401574(struct s0* ecx, struct s0* a2, struct s0* a3, struct s0* a4, struct s0* a5, struct s0* a6, struct s0* a7, struct s0* a8, struct
  int32_t eax19;
  struct s0* ebx20;

  if (!a2) {
    addr_40159d_2:
    return;
  } else {
    eax19 = 0;
  }
}
```

```
ebx20 = a2;
```

 $while (*reinterpret_cast < struct s0^{**} > (reinterpret_cast < unsigned char> (ebx20) + eax19 * 2)) \ (a) = 10^{-10} (a) + (a) +$



int32_t PathFindExtensionW = 0x6cc4;

```
goto PathFindExtensionW;
 \label{eq:structs0*a} structs0*a0, structs0*a3, structs0*a4, structs0*a5, structs0*a6) \\ \{ structs0*a0, str
 int32_t GetModuleHandleW = 0x67b8;
 struct s0* fun_4050e4(struct s0* a1, struct s0* a2, struct s0* a3, struct s0* a4, struct s0* a5, struct s0* a6, struct s0* a7, struct s0* a8, str
     goto GetModuleHandleW;
 3
 int32 t GetProcAddress = 0x67cc;
 struct s0* fun_4050ea(struct s0* a1, struct s0* a2, struct s0* a3, struct s0* a4, struct s0* a5, struct s0* a6, struct s0* a7, struct s0* a8, str
      goto GetProcAddress;
 int32_t RegSetValueExW = 0x6adc;
 void fun_4051aa(int32_t a1, int32_t a2, int32_t a3, int32_t a4, int32_t a5, int32_t a6, struct s0* a7, struct s0* a8, struct s0* a9, struct s0
      goto RegSetValueExW;
 int32_t SystemParametersInfoW = 0x6650;
 void fun_405072(int32_t a1, int32_t a2, int32_t a3, uint32_t a4, struct s0* a5, struct s0* a6, struct s0* a7, struct s0* a8, struct s0* a9, str
      goto SystemParametersInfoW;
 }
 int32_t SHChangeNotify = 0x69c8;
 struct s0* fun_40517a(int32_ta1, int32_ta2, int32_ta3, int32_ta4, int32_ta5, int32_ta6, int32_ta7, uint32_ta8, struct s0* a9, struct
      goto SHChangeNotify;
  int32_t NtQuerySystemInformation = 0x6b5a;
  int32_t fun_4051ce() {
  goto NtQuerySystemInformation;
}
  int32_t RtlReAllocateHeap = 0x6c86;
  void fun_40521c() {
 goto RtlReAllocateHeap;
}
  int32_t RtlFreeHeap = 0x6c3e;
```

goto RtlFreeHeap;
} int32_t RtlAdjustPrivilege = 0x6bbe;

void fun_40520a() {

int32_t fun_4051e6(int32_t a1, int32_t a2, int32_t a3, void* a4) { goto RtlAdjustPrivilege; 3

int32_t RegCreateKeyExW = 0x6aba;

int32_t fun_40519e(struct s0*a1, struct s0*a2, struct s0*a3, struct s0*a4, struct s0*a5, struct s0*a6, struct s0*a7, struct s0*a8, struct s0*a goto RegCreateKeyExW;

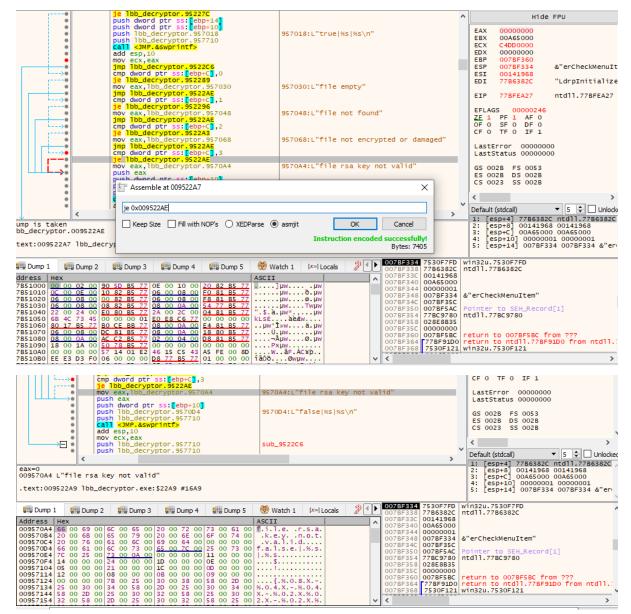


```
if (v4) {
     v5 = reinterpret_cast<struct s0*>(reinterpret_cast<int32_t>(ebp2) + 0xffffde8);
      fun_405180(0, v5, 35, 0)
     fun_401574(ecx, reinterpret_cast<int32_t>(ebp2) + 0xfffffde8, 0, v5, 35, 0, edi6, esi7, edx8, ecx, ebx9, v10, v11, v12, v13, v14, v15, v16)
fun_404eb6(reinterpret_cast<int32_t>(ebp2) + 0xfffffde8, &v4>f2, 0, v5, 35, 0, edi6, esi7, edx8, ecx, ebx9, v17, v18, v19, v20, v21, v22, fun_404eb6(reinterpret_cast<int32_t>(ebp2) + 0xffffde8, ".", 0, v5, 35, 0, edi6, esi7, edx8, ecx, ebx9, v24, v25, v26, v27, v28, v29, v30);
     v31 = reinterpret_cast<struct s0*>(reinterpret_cast<int32_t>(ebp2) + 0xffffde8);
     fun_405090(v31, 0, v5, 35, 0, edi6, esi7);
     v32 = reinterpret_cast<struct s0*>(reinterpret_cast<int32_t>(ebp2) + 0xfffffde8);
eax33 = fun_40522e(v32, v31, 0, v5, 35, 0);
fun_404ebc(eax33, ".", v32, v31, 0, v5, 35, 0, edi6);
     v34 = reinterpret_cast<struct s0*>(reinterpret_cast<int32_t>(ebp2) + 0xffffde8);
      fun_405090(v34, v32, v31, 0, v5, 35, 0)
    g407f10 = eax46;
     zf47 = g407f10 == 0;
if (zf47) {
v48 = reinterpret_cast<struct s0*>(reinterpret_cast<int32_t>(ebp2) + 0xfffffde8);
           v49 = reinterpret_cast<struct s0*>(0x8000000);
fun_4051a4(0x80000000, v48, eax42, "RegDeleteKeyExW", "a", v34, v32, v31, 0, v5, 35, 0, edi6, esi7, edx8, ecx);
v50 = reinterpret_cast<struct s0*>(&v4>f2);
           fun_4051a4(0x80000000, v50, 0x80000000, v48, eax42, "RegDeleteKeyExW", "a", v34, v32, v31, 0, v5, 35, 0, edi6, esi7);
           v51 = v4
           v52 = reinterpret_cast<struct s0*>(0x8000000))
           fun_4051a4(0x80000000, v51, 0x80000000, v50, 0x80000000, v48, eax42, "RegDeleteKeyExW", "a", v34, v32, v31, 0, v5, 35, 0);
     } else {
           v48 = reinterpret_cast<structs0*>(0);
           \label{eq:construct_solution} v49 = reinterpret_cast<structs0*>(0x100); v50 = reinterpret_cast<structs0*>(reinterpret_cast<int32_t>(ebp2) + 0xfffffde8); v50 = reinterpret_cast<structs0*>(reinterpret_cast<int32_t>(ebp2) + 0xfffffde8); v50 = v50 
           g407f10(0x80000000, v50, 0x100, 0, eax42, "RegDeleteKeyExW", "a", v34, v32, v31, 0, v5, 35, 0);
v51 = reinterpret_cast<struct s0*>(0);
           v52 = reinterpret_cast<struct s0*>(0x100);
           v53 = &v4->f2;
          9407f10(0x80000000, v53, 0x100, 0, 0x80000000, v50, 0x100, 0, eax42, "RegDeleteKeyExW", "a", v34, v32, v31, 0, v5, 35, 0);
g407f10(0x80000000, v4, 0x100, 0, 0x80000000, v53, 0x100, 0, 0x80000000, v50, 0x100, 0, eax42, "RegDeleteKeyExW", "a", v34, v32, v31,
```

After the execution of the subroutines of the management of the files attributes, it is executed a series of compare instructions.

In the following case it doesn't succeed the phase of the decryption of files (obtained through enumeration), in fact the verdict is "false". In case the results of the operations are true (so succeded) a positive verdict of the execution is written in the log file trial_dec.log (then called from the filepath trailing removing function PathRemoveFileSpecW):





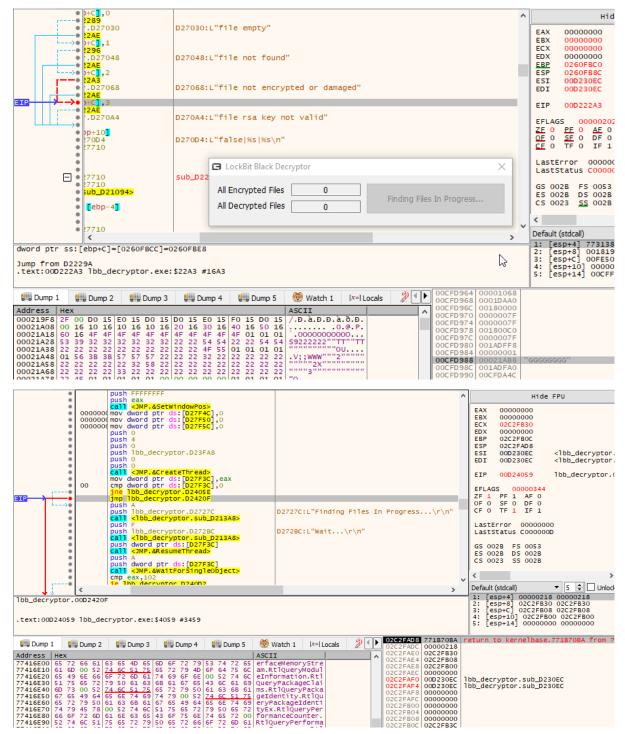
Following some details of the dumping of the EBP register, which is used to perform the compare instructions and execute the operation of switching through the various output options:



00952263 00952266	FF75 10 68 1870950		ish dword ptr s ish lbb_decryp		[ebp+10]:L"OneN 57018:L"true %	01 A	Hide
 0095226B 00952270 	68 1077950 E8 3B2C000	<u>00</u> pt	all <jmp.&swpr< td=""><td>tor.957710</td><td></td><td></td><td>EAX</td><td></td></jmp.&swpr<>	tor.957710			EAX	
 00952275 00952278 	83C4 10 8BC8	ac	d esp,10 ov ecx,eax		S	ub_952278	EB) EC)	X 00000000
0095227A v 0095227C v	EB 4A 837D 0C 00	jn	np <1bb_decrypt np dword ptr s	tor.sub_9522C6			ED) EBF	
• 00952280 × • 00952282	75 07	jr		or.952289		57030:L"file e	ES	P 043BFD90
-• 00952287 ·	B8 <u>3070950</u> EB 25	jn	np 1bb_decrypt	or.9522AE		57050.E THE E	m; ES: ED:	
	75 07	jr		or.952296		70494 "file m	EI	P 778C320C
	B8 <u>4870950</u> EB 18	jn	ov eax,1bb_decm np 1bb_decrypt	or.9522AE	9	57048:L"file n		LAGS 00000216
	75 07	jr	np dword ptr s ne lbb_decrypt	or.9522A3			ZF	0 PF 1 AF 1
0095229C 009522A1	B8 <u>6870950</u> EB 0B	<u>00</u> mc jn	ov eax, lbb_decr np_lbb_decrypt	ryptor.957068 or.95224E	9	57068:L"file n	OT CF	
009522A3 > 009522A7 ~	837D OC 03 75 05	3 cn jr	🖪 LockBit Black	Decryptor				× ror 000000
● 009522A9 →● 009522AE >	B8 <u>A470950</u> 50	00 mo pu		iles 0				atus C000000
 009522AF 009522B2 	FF75 10 68 D470950	00 pt			Fi	nding Files In Prog	ress	B FS 0053 B DS 002B
 009522B7 009522BC 	68 1077950 E8 EF2B000	<u>00</u> pt	All Decrypted r	-iles 0				3 SS 002B
 009522C1 009522C4 	83C4 10 8BC8	ac					<	-
- 009522C6 r>	68 1077950		ish 1bb decrypt	tor.957710	s	ub 9522C6	Y	
•	lecryptor							
•		s:[ebp+10] pr.957018		957018:L"tr	uel%s %s\n			
•	decrypto	or.957710						
>	2.&swprin							
•	eax decrypto	or.sub_95220	6>	sub_952278				
	i ptr ss:	[ebp+C],0						
•	bb_decry	.952289 yptor.957030)	957030:L"fi	le empty"			
L	lecryptor							
0	ecryptor.	. 332230						
			tor	0570 (D.) 153	1 E	×		
•		Bit Black Decryp	tor		1	×		
	C LockB		tor 0		· · · · · ·		maged"	
	C LockB	Bit Black Decryp	0	Decry	ot All Encrypt		maged"	
	C LockB	Bit Black Decryp		Decry	ot All Encrypt		maged"	
	All Encry All Decry	Bit Black Decryp ypted Files	0	Decry	ot All Encrypt		maged"	
	All Encry All Decry	Vit Black Decryp ypted Files ypted Files s: [ebp+10]	0			ed Files	maged"	
	All Encry All Decry d ptr se decrypto decrypto	Vit Black Decryp ypted Files ypted Files s:[ebp+10] pr.9570D4 pr.957710	0	Decry 9570D4:L"fa		ed Files	maged"	
	All Encry All Decry	Vit Black Decryp ypted Files ypted Files s:[ebp+10] pr.9570D4 pr.957710	0			ed Files	maged"	
	All Encry All Decry decrypto decrypto cessprin o sax	Vit Black Decryp ypted Files ypted Files s:[ebp+10] pr.9570D4 pr.957710	0			ed Files	maged"	
	All Encry All Decry decrypto .&swprin 0 eax <	Bit Black Decryp ypted Files ypted Files s:[ebp+10] or.9570D4 or.957710 ntf>	0	9570D4:L"fa	1se %s %s∖	ed Files		
dword ptr ss:[All Encry All Decry All Decry decrypto decrypto decrypto cepp+C]=[1	Bit Black Decryp ypted Files ypted Files s:[ebp+10] or.9570D4 or.957710 ntf>	0	9570D4:L"fa	1se %s %s∖	ed Files		
	All Encry All Decry decrypto decrypto cosax ebp+C]=[0 255	Bit Black Decryp ypted Files ypted Files s: [ebp+10] or.957004 or.957710 ntf> 0445FE64	0 0 BaseThreadIn	9570D4:L"fa	1se %s %s∖	ed Files		
dword ptr ss:[Jump from 9522	All Encry All Decry decrypto decrypto cosax ebp+C]=[0 255	Bit Black Decryp ypted Files ypted Files s: [ebp+10] or.957004 or.957710 ntf> 0445FE64	0 0 BaseThreadIn	9570D4:L"fa	1se %s %s∖	ed Files		
dword ptr ss:[Jump from 9522 .text:00952270	All Encry All Decry decrypto decrypto cosax ebp+C]=[0 255	Bit Black Decryp ypted Files ypted Files s: [ebp+10] or.957004 or.957710 ntf> 0445FE64	0 0 BaseThreadIn	9570D4:L"fa	1se %s %s∖	ed Files		004FFC20 7
dword ptr ss:[Jump from 9522 .text:00952270	C LockB All Encry All Decry decrypto decrypto decrypto decrypto cax < cebp+C]=[1 SE 1bb_decr	Wit Black Decryp ypted Files ypted Files s: [ebp+10] or. 957014 or. 957710 htf> 0445FE64 <&# ryptor.exe: 1</td><td>0 0 BaseThreadIn \$227C #167C</td><td>9570D4:L"fa</td><td>lse %s %s∖ kernel32.Ba</td><td>ed Files</td><td>:Thunk></td><td>004FFC20 7 004FFC24 0 004FFC28 7</td></tr><tr><td>dword ptr ss:[Jump from 9522 .text:00952270</td><td>C LockB All Encry All Decry decryptc decryptc decryptc decryptc cebp+C]=[1 SE Dbb_decr Dump 2 0 10 SE F</td><td>Bit Black Decryp ypted Files ypted Files ypted Files s: [ebp+10] or.957004 or.957710 ntf> 0445FE64 ryptor.exe: Imp Dump 3 E FF E FF FF FF</td><td>0 0 BaseThreadIn \$227C #167C Ump 4 28 FB 4F 00</td><td>9570D4:L"fa itThunk>]=< Ump 5 Dump 5 DC 59 D4 75</td><td>lse %s %s\ <ernel32.Ba</td><td>ed Files</td><td>:Thunk></td><td>004FFC20 7 004FFC24 C 004FFC28 7 004FFC28 7 004FFC20 7</td></tr><tr><td>dword ptr ss:[Jump from 9522 .text:00952270 Address Hex 004FFAD4 70 60 004FFAD4 16 3/</td><td>LockB All Encry All Decry decrypto .&swprir .0 ax < ebp+C]=[Dump 2 0 10 5E F <u>FF 72</u> 0 DA DA 00 8</td><td>Bit Black Decryp ypted Files ypted Files ypted Files s: (ebp+10) pr.95704 pr.957710 ntf> 0445FE64 www.pytor.exe:3 www.pytor.exe:3 </</td><td>0 0 BaseThreadIn \$227C #167C Ump 4 28 FB 4F 00 13 01 00 00 00 00 00 00</td><td>9570D4:L"fa itThunk>]=<k Dump 5 DC 59 D4 75 27 76 00 00 00 00 00 00</td><td>lse %s %s\ kernel32.Ba</td><td>ed Files</td><td>:Thunk></td><td>004FFC20 7 004FFC24 C 004FFC28 7 004FFC20 C 004FFC30 7 004FFC30 7</td></tr><tr><td>dword ptr ss:[Jump from 9522 .text:00952270 Address Hex 004FFAD4 70 60 004FFAD4 16 30 004FFAD4 16 30 004FFAD4 16 30</td><td>LockB All Encry All Decry All Decry decryptc decryptc decryptc decryptc c decryptc d</td><td>Bit Black Decryp ypted Files ypted Files <</td><td>0 0 3aseThreadIn \$227C #167C U Dump 4 28 FB 4F 00 13 01 00 00 00 00 00 00 35 FB 4F 00</td><td>9570D4:L"fa itThunk>]=<k Dump 5 DC 59 D4 75 27 76 00 00 00 00 00 00 6E F7 0D 00</td><td>lse %s %s\ <pre></td><td>ed Files</td><td>:Thunk></td><td>004FFC20 7 004FFC24 C 004FFC28 7 004FFC28 7 004FFC30 7 004FFC34 C 004FFC38 3 004FFC38 3</td></tr><tr><td>dword ptr ss:[Jump from 9522 .text:00952270 Address Hex 004FFAD4 10 3/ 004FFAD4 10 3/ 004FFAD4 10 3/ 004FFAD4 00 00</td><td>C LockB All Encry All Decry All Decry Al</td><td>Bit Black Decryp ypted Files ypted Files ypted Files s: ebp+10] pr.95704 pr.957710 ntf> 0445FE64 wptor.exe: ump 3 E FF FF 028 028 084 F6</td><td>0 0 3aseThreadIn \$227C #167C 0 5 FB 4F 00 74 05 35 00 FF 3D 00 00</td><td>9570D4:L"fa itThunk>]=<</td><td>lse %s %s\ kernel32.Ba</td><td>ed Files</td><td>:Thunk></td><td>▶ 004FFC20 7 004FFC24 7 004FFC28 7 004FFC26 0 004FFC30 7 004FFC38 3</td></tr><tr><td>dword ptr ss:[Jump from 9522 .text:00952270 Address Hex 004FFAB4 70 60 004FFAB4 16 3/ 004FFAC4 16 3/ 004FFAF4 00 00</td><td>C LockB All Encry All Decry All Decry All Decry All Decry action of the second of the second action of the second of the</td><td>Bit Black Decryp ypted Files ypted Files <</td><td>0 0 3aseThreadIn \$227C #167C</td><td>9570D4:L"fa itThunk>]=< Dump 5 <u>PC 59 D4 75</u> 27 76 00 00 00 00 00 00 <u>SE F7 OD 00</u> 00 00 00 00</td><td>lse %s %s\ kernel32.Ba</td><td>ed Files</td><td>:Thunk></td><td>004FFC20 7 004FFC28 7 004FFC28 7 004FFC28 7 004FFC30 7 004FFC34 0 004FFC38 3 004FFC34 0 004FFC44 0 004FFC44 0</td></tr><tr><td>dword ptr ss:[Jump from 9522 .text:00952270 Address Hex 004FFAB4 70 60 004FFAC4 50 AG 004FFAC4 50 AG 004FFAC4 74 05 004FFAF4 00 00 004FFB4 00 00 004FFB4 00 00 004FFB4 00 00</td><td>C LockB All Encry All Decry All Decry decryptc *.&swprir 0 *.&swprir</td><td>Bit Black Decryp ypted Files ypted Files <</td><td>0 0 3aseThreadIn \$227C #167C \$227C #167C \$200 00 00 \$25 FB 4F 00 \$25 FB</td><td>9570D4:L"fa itThunk>]=< Dump 5 DC 59 D4 75 27 76 00 00 00 00 00 00 6E F7 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td><td>Ise %s %s (ernel32.Ba (ernel32.Ba) (ernel32.Ba)</t</td><td>ed Files</td><td>:Thunk></td><td> ▶ 004FFC20 7 > 004FFC24 C > 004FFC28 7 > 004FFC30 7 > 004FFC34 C > 004FFC38 3 > 004FFC38 3 > 004FFC48 C > 004FFC40 C </td></tr><tr><td>dword ptr ss:[Jump from 9522 .text:00952270 Address Hex 004FFAB4 70 60 004FFAC4 16 3/ 004FFAC4 16 3/ 004FFAC4 16 3/ 004FFAC4 00 00 004FFAC4 74 00 004FFAC4 74 00 004FFAC4 74 00</td><td>LockB All Encry All Decry All Decry All Decry All Decry All Decry Comparison (acrypton decrypton decrypton decrypton (comparison (comparison)</td><td>Bit Black Decryp ypted Files ypted Files <</td><td>0 0 3aseThreadIn \$227C #167C Ump 4 28 FB 4F 00 13 01 00 00 00 00 00 00 35 FB 4F 00 74 05 35 00 FF 3D 00 00 74 05 35 00 FF 3D 00 00 75 FB 4F 00 74 05 35 00 FF 3D 00 00 75 FB 4F 00 74 05 35 00 75 75 75 75 00 00 00 00 76 87 75 75 75 75 00 00 00 00 76 3A DA 00 00 00 00 00</td><td>9570D4:L"fa itThunk>]=< Dump5 Dump5 DC 59 D4 75 27 76 00 00 00 00 00 00 6E F7 0D 00 00 00 00 00 10 00 10</td><td><pre>lse %s %s\ cernel32.Ba</td><td>ed Files</td><td>:Thunk></td><td>▶ 004FFC20 7 004FFC24 C 004FFC28 7 004FFC28 7 004FFC28 7 004FFC34 C 004FFC34 C 004FFC34 C 004FFC44 C 004FFC44 C 004FFC48 C</td></tr></tbody></table>						

Here's a detail of debugging with the EIP pointed before the verdict "RSA key not valid", so it is simulated the read of the content of the registers (so the data related to the execution) during the decryption and logging phase:





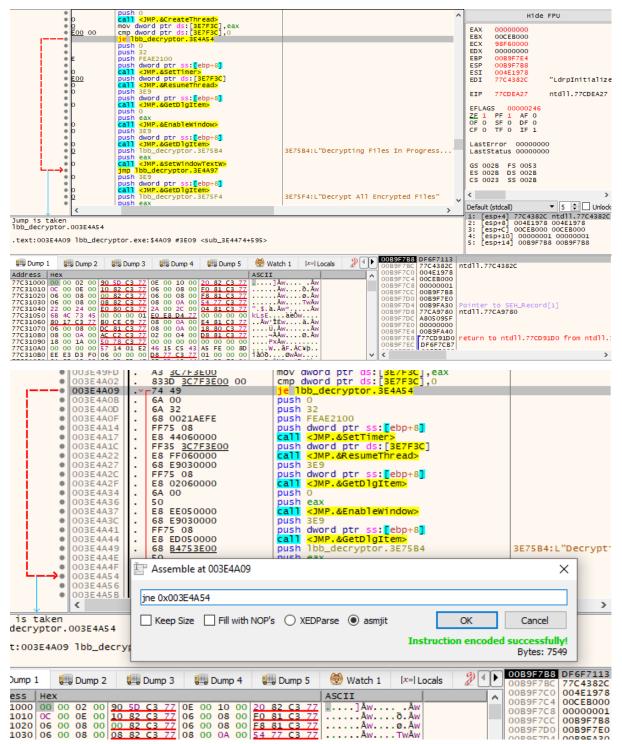
Following the evidence of breakpoint hitting where you can see that during the phase of debugging it was intercepted the breakpoint previously set:



Software	00D222A3	1bb_decryptor.exe		Enabled	cmp dword ptr ss	:[ebp+C].3			<u> </u>			
	00D24059	1bb_decryptor.exe			jmp 1bb_decrypto	r.D2420F			1			
	• :C	ryptor.D27018 ryptor.D27710	D27	7018:L"tr	rue %s %s\n"					^	Hide	FPU
343	• • •	wprintf> ryptor.sub_D222C6> r ss:[ebp+C],0	sut	_D22278						EAX EBX ECX EDX	00000000	
		yptor.D22289 decryptor.D27030 yptor.D222AE r ss:[ebp+C],1 yptor.D22296			ile empty"					ESP ESI EDI	0334F988 00D230EC	<1bb_decry <1bb_decry
		decryptor.D27048 yptor.D222AE r ss:[ebp+C],2	D27	7048:L"f	ile not found"					EIP	00D2227C	1bb_decryp
) it	<pre>yptor.D222A3 decryptor.D27068 yptor.D222AE r ss:[ebp+C],3 yptor.D222AE</pre>	D27	B Lock	Bit Black Decrypto	r r	cod"				0 <u>PE 1 AE 0</u> 0 <u>SE 0 DF 0</u> 0 TF 0 IF 1	
		decryptor.D270A4	D27	All Enc	rypted Files	0	Decry	pt All Encr	voted Files		tError 0000000 tStatus <mark>C000000</mark>	
	• :C	tr ss:[ebp+10] ryptor.D270D4 ryptor.D27710 wprintf>	D27	All Dec	rypted Files	0]		,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	5	002B FS 0053 002B DS 002B 0023 <u>SS</u> 002B	
		ryptor.D27710	sut	_D222C6					>	< ✓ Defau	ult (stdcall)	▼ 5 🜩 🗆
Jump fro	m D2225E	<pre>pp+C]=[0334F9C8]=0: bb_decryptor.exe::</pre>								2: 3: 4:	[esp+4] 77313820 [esp+8] 0148196 [esp+C] 0100600 [esp+10] 000000 [esp+14] 012FF2	8 014B1968 0 01006000 01 00000001
🚛 Dump	1	Dump 2 🛛 💷 Dump 3	💷 Dump 4	Uump	5 🛞 Watch 1	[x=] Locals	2 • •	D12FF2A8 D12FF2AC	50000063 03682680			
012FF75C 012FF76C 012FF77C 012FF78C 012FF78C 012FF78C 012FF78C 012FF78C 012FF77C 012FF77C	7C F7 2 00 00 E 74 44 E 8A 50 C 74 44 E 28 71 E 8 03 0 41 40 E 00 00 0 01 00 0	2: 0:<	7A 7C 5F 14 7 22 36 C0 76 0 01 00 00 0 8 7S BS D2 00 0 0 00 00 00 00 0 7 37 4C D2 00 0 0 00 00 00 00 0 0 00 00 00 00 0 0 00 00 00 00 0 0 00 00 00 00 0 0 00 00 00 00 0 0	Image: Non-Stress Image: No-Stress Image: No-Stres	00 İ÷/z . 00 0÷/ 01 tDô 00 °PÄvò.x.(00 (qô.ô÷/.7L0) 00 ètD(00 AMÒ. /7W 80	x.ò. W ÷+/.) rLò.)ò.) /£t		012FF2D4 012FF2D8	03682680 0000001 03682788 01680000 012FF288 0000000 012FF420 77379780 4C362932		77379780 1 to ntd]].773E	B10E from nt

Following, the details of an attempt to modify a **JE** assembly instruction in the context of the creation of the bulk decryption thread to modify the execution flow after the creation of it:







003E48C7 .* 0F84 CA010000 je lbb_decryptor.3E4A97 003E48D3 . FF35 3C7F3E00 push dword ptr ds:[3E7F3C] 003E48D3 . C705 3C7F3E00 0000000 003E48E2 . 68 E26FAE2B push dword ptr ds:[3E7F3C],0 003E48E4 . E8 4D070000 call call SMP.&ZwClose> 003E48E7 . . E8 4D070000 call Moord ptr ds:[3E7F5C],1 003E48E6 . . 833D 5C7F3E00 01 cmp dword ptr ds:[3E7F5C],1 003E48F6 . 833D 5C7F3E00 01 cmp dword ptr ds:[3E7F54],0 jme lbb_decryptor.3E4944 003E48F6 . 833D 5C7F3E00 00 cmp dword ptr ds:[3E7F54],0 je lbb_decryptor.3E490F 003E4903 . 74 0A jme lbb_decryptor.3E490F lea ebx,dword ptr ds:[3E3FA8] 003E4905 . 8D1D 2C423E00 mov ecx,dword ptr ds:[3E7F41],0 003E4905 . 8D1D 2C47E3E00 0000000 mov dword ptr ds:[3E7F42],1 003E4915 . 8D00 417F3E00 mov dword ptr ds:[3E7F41],0 je ebx,dword ptr ds:[3E7F42],1 003E4925	₩22C>]
•••••••••••••••••••••••••••••	Hide FPU EAX 0000000 EBX 0000000 EBX 0000000 EBX 0000000 EDX 0000000 EDX 0000000 EDX 0000000 EBP 0285FE40 ESI 003E30EC Clbb_decryptor. EIP D03E30EC <1bb_decryptor.
Trc31040 00 <	return to gdi32.778F6E05 from ??? L"nei Dollar" return to gdi32full.778FE4E3 from ??'
<pre>push lbb_decryptor.3E7318 call <lbb_decryptor.sub_3e13a8> push F push lbb_decryptor.3E7364 call <lbb_decryptor.sub_%e13a8> push dword ptr ds:[3E7F343]</lbb_decryptor.sub_%e13a8></lbb_decryptor.sub_3e13a8></pre>	iles In Progress\r\n"
<pre>call <jmp.&resumethread> push A push A push dword ptr ds:[3E7F3C] mov ebp,esp call <jmp.&waitforsingleob <jmp.&closehandle="" call="" cmp="" ds:[3e7f3c]="" dword="" eax,102="" ie="" lbb_decryptor.3e41ce="" ptr="" push=""> push dword ptr ds:[3E7F60] push lbb_decryptor.3E7378 lea eax,dword ptr ss:[ebp-4],eax cmp dword ptr ss:[ebp+C],0 </jmp.&waitforsingleob></jmp.&resumethread></pre>	:)\r" >
<pre>isoperation to construct the second sec</pre>	001EF934 F06658BD 001EF938 77C4382C 001EF93C 006C1978

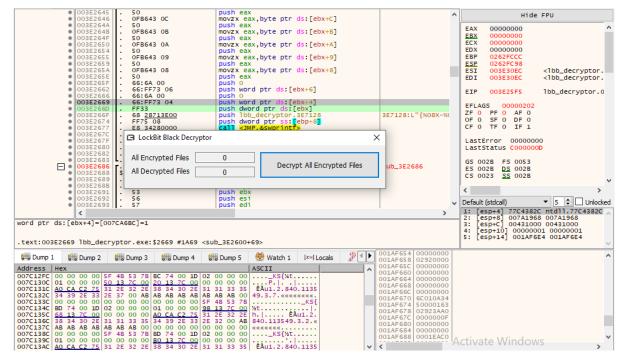


Here are, instead, details related to a session of breakpoint analysis when the buffer is filled with the ID during the MD5 digest:

003E2666 - 66:64 003E2669 - 66:FF 003E2660 - FF33 003E2667 - 68 22 003E2674 - FF5 003E2677 - E8 34 003E2677 - E8 34 003E2677 - 83C4 003E2677 - 83C4 003E2687 - 58 003E2683 - C2 04 003E2688 - 88E5 003E2688 - 88E5 003E2688 - 88E5 003E2688 - 88E5 003E2688 - 88E5 003E2689 - 88E5 003E2691 - 53 003E2691 - 55 003E2691 - 55 003E2691 - 55 003E2691 - 55 003E2691 - 55 005 003E2691 - 55 005 005 005 00	73 06 push word ptr ds 00 push o push o 73 04 push word ptr ds 90 push dword ptr ds push dword ptr ds 08 push dword ptr s push dword ptr s 1280000 call sJMP.&swpri 34 add esp,34 pop ebx mov esp,ebp pop ebp pop	:[ebx+4] ds:[ebx] cor.3E7128 ss:[ebp+8]	3E7128:L"{%08X-%(
O03E2645 O03E2645 O03E2644 O03E2644 O03E2644 O03E2644 O03E2644 O03E2644 O03E2647 O03E2654 O03E2654 O03E2654 O03E2655 O03E2655 O03E2655 O03E2665	0 0 0 0 0 0 0 0 0 0 0 0 0 0	Berring String	0 DF 0 0 IF 1 00000000 FS 0053 DS 0028 SS 0028
.text:0322669 lbb_decryptor.exe:\$2669 #1A69 ## Dump 1 ## Dump 2 ## Dump 3 ## Dump 4 Address Hex 007863FC 00 00 00 00 00 00 00 00 00 00 00 00 00	Dump 5 Watch 1 Ix= Locals Ix= 00 <td>001AF664 00000000 001AF664 77C6AE50 return to n 001AF666 0000000 001AF676 0000000 001AF677 02920000 001AF677 02920000 001AF677 02920000 001AF677 02920000 001AF685 001AF688</td> <td>tdll.77C6AE71 from ntdll.: tdll.77C6AE50 from ntdll.: tdll.77D1BD67 from ntdll.: Windows</td>	001AF664 00000000 001AF664 77C6AE50 return to n 001AF666 0000000 001AF676 0000000 001AF677 02920000 001AF677 02920000 001AF677 02920000 001AF677 02920000 001AF685 001AF688	tdll.77C6AE71 from ntdll.: tdll.77C6AE50 from ntdll.: tdll.77D1BD67 from ntdll.: Windows

By performing a dump of the EBX register it is possible to see some embedded numeric patterns which, with probability, are related to the digesting operation:





It is possible to highlight the presence of a security identifier. The decryptor tool performs infact also OS settings and information gathering. Those evidences show, however, the trend to customize those decryptor tools for the victims.



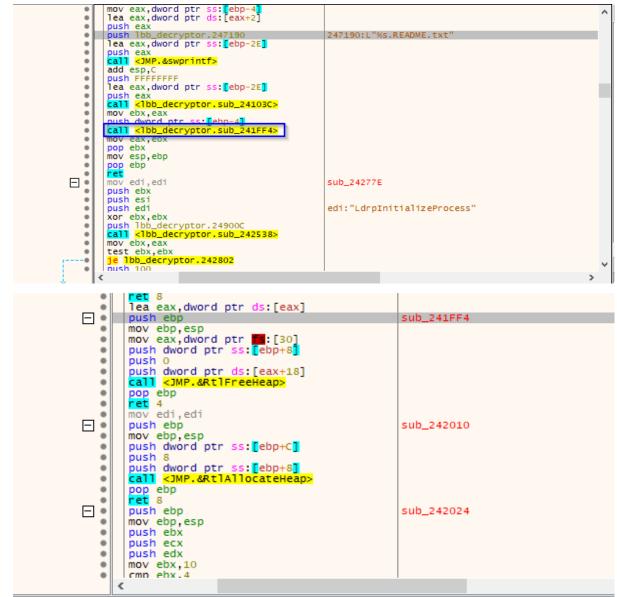
003E2645 003E2646 003E2646 0FE	3643 0C push eax movzx eax	<pre>c,byte ptr ds:[ebx+C]</pre>	^	Hide FPU	
003E264A 003E264B 003E264B 0FE	push eax	(,byte ptr ds:[ebx+B]		EAX 0000000	^
003E264F 50 003E2650 0FE	push eax	(,byte ptr ds:[ebx+A]		EBX 0000000 ECX 0000000	
	8643 09 push eax movzx eax	(,byte ptr ds:[ebx+9]		EDX 0000000 EBP 0262FCCC	
003E2659 003E265A 005E265A	push eax	,byte ptr ds:[ebx+8]		ESP 0262FC98 ESI 003E30EC <1bb_decryptor	
003E265E 50 003E265F 66	6A 00 push eax push 0			EDI 003E30EC <1bb_decryptor	
003E2666 . 66:	64 00 nush 0	i ptr ds:[ebx+6]		EIP 003E25F5 lbb_decryptor.)
003E266D . EE3	FF73 04 push word 33 push dwor	i ptr ds:[ebx+4] d ptr ds:[ebx]		EFLAGS 00000202	
003E266F 003E2674 FF7	28713E00 75 08 push 1bb_ push dwor	d ptr ds:[ebx+4] d ptr ds:[ebx] decryptor.3E7128 d ptr ss:[ebp+8]	3E7128:L"{%08X-%0	ZF 0 PF 0 AF 0 OF 0 SF 0 DF 0	
003E2677 E8 003E267C	34280000 call < JMF ockBit Black Decryptor	2.&swprintf>		CF 0 TF 0 IF 1	
• 003E2680 .	оскыт ыаск рестуртог	/		LastError 00000000 LastStatus C000000D	
003E2682 003E2683 All E	Encrypted Files 0			G5 0028 E5 0053	
O03E2686 003E2688 \$ All [Decrypted Files 0	Decrypt All Encrypted Files	ub_3E2686	ES 002B DS 002B CS 0023 SS 002B	
• 003E2689 • 003E2688				<	۷
003E2691 . 53 003E2692 . 56 003E2693 . 57	push ebx push esi			Default (stdcall)	od
• 003E2693 . 57	push edi		×	1: [esp+4] 77C4382C ntd]].77C4382C	
word ptr ds:[ebx+4]=[007CA6BC]=	=1			2: [esp+8] 007A1968 007A1968 3: [esp+C] 00431000 00431000	
text:00252550 lbb despurter ex	(a) \$2550 #1150			4: [esp+10] 00000001 00000001 5: [esp+14] 001AF6E4 001AF6E4	
.text:003E2669 lbb_decryptor.ex			001AF654 00000000		~
Ump 1 Dump 2 Dump		👹 Watch 1 🛛 [x=] Locals 🖉 🕪	001AF658 02920000 001AF65C 00000000		^
Address Hex 0078005C AB AB AB AB AB 00 00 00 0 0078006C BE 74 00 18 53 00 20 0		ASCII ASCII	001AF660 00000000 001AF664 00000000		
007B006C BE 74 00 18 53 00 2D (007B007C 2D 00 33 00 2D 00 37 0	00 <u>31 00 20 00</u> 31 00 35 00 5 00 38 00 37 00 34 00 34 00 -	««««GKSC ¥tS11.5. 37.8.7.4.4.	001AF668 00000000 001AF66C 00000000		
007B008C 38 00 32 00 35 00 34 0 007B009C 37 00 39 00 37 00 32 0	0 38 00 35 00 <u>38 00 20 00</u> 7	7.9.7.2.8.5.8	001AF670 97010766 001AF674 50000163		
		3.5.5.8.6.3.3.6.	001AF678 02923AA0 001AF67C 0000000F		
007B00CC 38 00 36 00 39 00 36 (00 34 00 00 00 00 00 00 00 00 8	3.6.9.6.4	001AF680 00000000		
007B00EC 00 00 00 00 5F 4B 53 007B00FC 76 00 69 00 63 00 65 0	AB AB AB AB AB AB 00<	KS{¥td.e. /.i.c.e.c.a.p.a.	001AF684 00000000 001AF688 0001EAC0	ctivate Windows	¥
007B010C 62 00 69 00 6C 00 69 0	00 <u>74 00 79 00</u> 63 00 61 00 k	0.1.1.1.t.y.c.a. ♥	<	>	
007B010C 62 00 69 00 6C 00 69 0			<	\$	
0078010C 62 00 69 00 6C 00 69 0 003E2669 003E266D	. 66:FF73 04 . FF33	push word ptr ds push dword ptr d	<pre>{ [ebx+4] [ebx] </pre>	>	
0078010C 62 00 69 00 6C 00 69 0 003E2669 003E266D 003E266F	. 66:FF73 04 . FF33 . 68 <u>28713E00</u>	push word ptr ds push dword ptr d push lbb_decrypto	<pre>(ebx+4) (ebx) /pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 60 00 60 00 003E2669 003E2669 003E2667 003E267 003E267	. 66:FF73 04 . FF33 . 68 <u>28713E00</u> . FF75 08 . E8 34280000	push word ptr ds push dword ptr d	<pre> (ebx+4] (ebx) (ebx</pre>	>	
0078010C 62 00 69 00 6C 00 69 0 003E2669 003E2669 003E2667 003E2674 003E2677 003E2677	. 66: FF73 04 . FF33 . 68 <u>28713E00</u> . FF75 08 . E8 34280000	push word ptr ds push dword ptr d push lbb_decrypt push dword ptr s call <jmp.&swpri< td=""><td><pre> (ebx+4] (ebx) (ebx</pre></td><td>></td><td></td></jmp.&swpri<>	<pre> (ebx+4] (ebx) (ebx</pre>	>	
0078010C 62 00 69 00 60 00 60 00 003E2660 003E2660 003E267 003E267 003E267 003E2677 003E2677 003E2677 003E2676	66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> C3 LockBit Black Dec	push word ptr ds push dword ptr d push lbb_decrypt push dword ptr s call <jmp.&swpri< td=""><td><pre> (ebx+4] (ebx) (ebx</pre></td><td>3E7128:L"{%08X-%(</td><td></td></jmp.&swpri<>	<pre> (ebx+4] (ebx) (ebx</pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 60 00 60 00 60 00 60 00 60 00 00 82 666 003 82 666 003 82 2667 003 82 2677 003 82 2677 003 82 2677 003 82 2677 003 82 2677 003 82 2672 003 82 2672	66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> C3 LockBit Black Dec	push word ptr ds push dword ptr d push lbb_decrypt push dword ptr s call ryptor	<pre>c [ebx+4] c:[ebx] cr.3E7128 c:[ebp+8] cf></pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 60 00 0	66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> E3 LockBit Black Dec All Encrypted Files	push word ptr ds push dword ptr d push lbb_decrypto call <jmp.&swprid ryptor</jmp.&swprid 	<pre> (ebx+4] (ebx) (ebx</pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 60 00 0	66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> GI LockBit Black Dec All Encrypted Files All Decrypted Files	push word ptr ds push dword ptr d push lbb_decrypt push dword ptr s call ryptor	<pre>c [ebx+4] c:[ebx] cr.3E7128 c:[ebp+8] cf></pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 6C 00 69 0 003E266P 003E266P 003E267 003E267 003E267 003E267 003E267 003E267 003E2682 003E2682 003E2688 003E2688 003E2688 003E2688 003E2689 003E2	G6: FF73 04 FF33 G8 <u>28713E00</u> FF75 08 E8 34280000 G3 LockBit Black Dec All Encrypted Files All Decrypted Files	push word ptr ds push dword ptr d push 1bb_decrypto call <jmp.&swpri ryptor</jmp.&swpri 	<pre>c [ebx+4] c:[ebx] cr.3E7128 c:[ebp+8] cf></pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 60 00 80 00 00382669 00382667 0038267 0038267 0038267 0038267 0038267 00382683 00086868 00086868 00086868 00086868 0086868	 66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 34280000 C3 LockBit Black Dec All Encrypted Files \$ All Decrypted Files 53 	push word ptr ds push dword ptr d push lbb_decrypto call <jmp.&swpri ryptor 0 De 0 De</jmp.&swpri 	<pre>c [ebx+4] c:[ebx] cr.3E7128 c:[ebp+8] cf></pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 60 00 60 00 60 00 60 00 60 00 60 00 0	66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56	push word ptr ds push dword ptr d push 1bb_decrypto call <jmp.&swpri ryptor</jmp.&swpri 	<pre>c [ebx+4] c:[ebx] cr.3E7128 c:[ebp+8] cf></pre>	3E7128: L" {%08X-%(X Files ub_3E2686	
0078010C 62 00 69 00 60 00 0	 66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> C3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr d push lbb_decrypt call ryptor	<pre>c [ebx+4] c:[ebx] cr.3E7128 c:[ebp+8] cf></pre>	3E7128:L"{%08X-%(
0078010C 62 00 69 00 60 00 60 00 60 00 60 00 60 00 60 00 0	 66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> C3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr d push lbb_decrypt call ryptor	<pre>c [ebx+4] c:[ebx] cr.3E7128 c:[ebp+8] cf></pre>	3E7128: L" {%08X-%(X Files ub_3E2686	
007B010C 62 00 69 00 6C 00 69 0 003E266P 003E266P 003E2674 003E2677 003E2677 003E2677 003E2670 003E2680 003E2691 003E2691 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2690 003E2690 003E2690 003E2680 003E2690 004000000000000000000000000000000000	. 66: FF73 04 . FF33 . 68 <u>28713E00</u> . FF75 08 . E8 <u>34280000</u> . E3 LockBit Black Dec . All Encrypted Files . S3 . S6 . S7 . S7	push word ptr ds push dword ptr d push 1bb_decrypte push dword ptr s: call <jmp.&swprij ryptor 0 De 0 De push ebx push esi push edi</jmp.&swprij 	<pre>c [ebx+4] c:[ebx] c.327128 c:[ebp+8] cf></pre>	3E7128: L" {%08X-%(X Files ub_3E2686	
0078010C 62 00 69 00 60 00 0	. 66: FF73 04 . FF33 . 68 <u>28713E00</u> . FF75 08 . E8 <u>34280000</u> . E3 LockBit Black Dec . All Encrypted Files . S3 . S6 . S7 . S7	push word ptr ds push dword ptr d push 1bb_decrypte push dword ptr s: call <jmp.&swprij ryptor 0 De 0 De push ebx push esi push edi</jmp.&swprij 	<pre>c [ebx+4] c:[ebx] c.327128 c:[ebp+8] cf></pre>	3E7128: L" {%08X-%(X Files ub_3E2686	
007B010C 62 00 69 00 6C 00 69 0 003E266P 003E266P 003E2674 003E2677 003E2677 003E2677 003E2670 003E2680 003E2691 003E2691 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2690 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2690 003E2690 003E2690 003E2680 003E2690 004000000000000000000000000000000000	. 66: FF73 04 . FF33 . 68 <u>28713E00</u> . FF75 08 . E8 <u>34280000</u> . E3 LockBit Black Dec . All Encrypted Files . S3 . S6 . S7 . S7	push word ptr ds push dword ptr d push lbb_decrypt push dword ptr s call ryptor 0 De 0 De 0 De 0 Push ebx push esi push edi	< [ebx+4] s; [ebx] or.3E7128 s; [ebp+8] ortf> crypt All Encrypted	3E7128: L" {%08X-%(X Files ub_3E2686	
007B010C 62 00 69 00 6C 00 69 0 003E2669 003E2667 003E267 003E267 003E267 003E267 003E267 003E267 003E267 003E2683 003E2683 003E2683 003E2688 003E2688 003E2689 003E2689 003E2689 003E2692 003E2692 003E2692 003E2691 003E2691 003E2692 003E2691	 66: FF73 04 FF33 68 28713E00 FF75 08 E8 34280000 E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr ds push lbb_decrypt push dword ptr s call ryptor 0 De 0 De 0 De 0 Push ebx push esi push edi	< [ebx+4] s; [ebx] or.3E7128 s; [ebp+8] ortf> crypt All Encrypted	3E7128: L" {%08X-%(× Files ub_3E2686 001AF654 0000000 001AF655 002920000 001AF655 002920000 001AF655 002920000 001AF655 002920000	
007B010C 62 00 69 00 6C 00 69 0 003E2669 003E2667 003E267 003E267 003E267 003E267 003E267 003E267 003E267 003E2688		<pre>push word ptr ds push dword ptr ds push 1bb_decrypt; push dword ptr s: call </pre> push conductory of the push dword ptr s: call <pre>push esi push esi push esi push edi </pre> <pre> 9 <sub_3e2600+69> 4</sub_3e2600+69></pre>	< [ebx+4] [ebx] r.3E7128 [ebp+8] ntf> [v= Locals	3E7128:L"{%08X-%(× Files ub_3E2686 001AF654 0000000 001AF655 002920000 001AF655 002920000 001AF656 00000000 001AF656 00000000 001AF656 00000000	
007B010C 62 00 69 00 6C 00 69 0 003E2669 003E2669 003E267 003E267 003E267 003E267 003E267 003E267 003E2683 003E2683 003E2683 003E2688 0003E2688 00000000000000000000000000000000000	 66: FF73 04 FF33 68 28713E00 FF75 08 E8 34280000 E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr dipush lbb_decrypt push dword ptr si call call push dword ptr si call push dword ptr si push esi push edi 9 <sub_3e2600+69> 4 JDump 5 Watch ASCII 10 00 00 00 00 00 «««««««««» 8 64 00 65 00 si 64 00 65 00</sub_3e2600+69>	< :: [ebx+4] :: [ebx] :. [ebx] or. 3E7128 :: [ebp+8] ttf> crypt All Encrypted 1	3E7128: L"{%08X-%(× Files ub_3E2686 001AF654 00000000 001AF655 02920000 001AF660 00000000 001AF664 00000000 001AF665 00000000 001AF668 00000000	
007B010C 62 00 69 00 6C 00 69 0 003E2669 003E2667 003E267 003E267 003E267 003E267 003E267 003E267 003E267 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2689 003E2688 003E2688 003E2689 003E2688 003E2688 003E2688 003E2689 003E2688 003E2688 003E2689 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2689 003E2688	 66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 57 58 59 57 56 57 57 57 56 57 57 57 56 57 57 57 58 57 57 58 57 58 57 58 57 59 50 /ul>	push word ptr ds push dword ptr ds push lbb_decrypt push dword ptr s: call 0 De 0 Secondary 0 O 9 <sub_3e2600+69> 4 Dump 5 Secondary 0 O O 0 O O 0 O O 0 O O 0 O O 0 O O 0 O O 0 O O 0 O O 0 O O 0 O O 0 O O</sub_3e2600+69>	<pre> [ebx+4] [ebx] [ebx] or.3E7128 [ebp+8] ntf> [rypt All Encrypted [x=]Locals [x=</pre>	BE7128:L"{%08X-%(X Files ub_3E2686 001AF654 0000000 001AF658 02920000 001AF65C 0000000 001AF664 0000000 001AF664 0000000 001AF664 0000000 001AF664 0000000	
0078010C 62 00 69 00 6C 00 69 0 003E2669 003E2667 003E267 003E267 003E267 003E267 003E267 003E267 003E267 003E2682 003E2688 003E2688 003E2688 003E2689 003E2689 003E2689 003E2689 003E2693 < 003E2693 <br word ptr ds:[ebx+4]=[00 .text:003E2669 lbb_decr .text:003E2669 lbb_decr .text:003E2688 lbb_decr .text:003E2669 lbb_decr .text:003E2669 lbb_decr .text:003E2688 lbb_decr .text:003E269 lbb_decr .text:003E2669 lbb_decr .text:003E2669 lbb_decr .text:003E2688 lbb_decr .text:003E2669 lbb_decr .text:003E2688 lbb_decr .text:003E2688 lbb_decr	 66: FF73 04 FF33 68 28713E00 FF75 08 E8 34280000 E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr dipush lbb_decrypt push dword ptr sick call push dword ptr sick call push dword ptr sick call push dword ptr sick push esi push edi 9 <sub_3e2600+69> 4 ● Dump 5 ● Watch 10 00 00 00 00 ≪<<<<<<<<<<<<<<<<<<>>Watch 10 40 065 00 p.e.n.d. 10 74 00 6D 00 p.e.n.d. 10 72 00 69 00 n.t.i.n.</sub_3e2600+69>	< <pre> [ebx+4] :[ebx] :[ebx] :[ebp+8] :[ebp+</pre>	3E7128: L" {%08X-%(× Files ub_3E2686 001AF654 0000000 001AF658 02920000 001AF656 0000000 001AF656 0000000 001AF666 0000000 001AF666 0000000 001AF666 0000000 001AF674 5000163	
007B010C 62 00 69 00 6C 00 69 0 003E2669 003E2667 003E267 003E267 003E267 003E267 003E267 003E267 003E267 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2689 003E2688 003E2689 003E2688 003E2689 003E2688 003E2689 003E2688 003E2688 003E2689 003E2688 003E2689 003E2688 003E2688 003E2688 003E2689 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2688 003E2689 003E2688	 66: FF73 04 FF33 68 28713E00 FF75 08 E8 34280000 E3 LockBit Black Dec All Encrypted Files All Decrypted Files \$3 \$6 \$7 	push word ptr ds push dword ptr ds push dword ptr d push dword ptr s call 0 Decrypt push dword ptr s push dword ptr s dword ptr s push dword ptr	< <pre> [ebx+4] [ebx] [ebx] or.3E7128 [ebp+8] or.f> [v=lLocals </pre>	3E7128:L"{%08X-%(X Files ub_3E2686 001AF654 0000000 001AF658 02920000 001AF656 0000000 001AF664 0000000 001AF664 0000000 001AF664 0000000 001AF664 0000000 001AF670 02010988 001AF674 5000163 001AF678 02923A00	
007B010C 62 00 69 00 6C 00 69 0 003E266P 003E266P 003E267 003E267 003E267 003E267 003E267 003E267 003E2682 003E2682 003E2682 003E2682 003E2682 003E2682 003E2682 003E2682 003E2682 003E2693 < word ptr ds:[ebx+4]=[00 .text:003E2669 1bb_decr ↓ 007AFC6C AB AB AB AB 007AFC7C 00 00 00 05 007AFC8C 70 00 65 00 61 007AFC8C 74 00 74 00 65 007AFC8C 72 00 66 00 74 007AFC8C 72 00 06 00 74 007AFC8C 74 00 74 00 65 007AFC8C 74 00 74 00 65 007AFC8C 74 00 74 00 65 007AFC8C 72 00 06 00 74 007AFC8C 72 00 06 00 74 007AFC8C 74 00 74 00 65 007AFC8C 74 00 74 00 65 007AFC8C 72 00 06 00 74 007AFC8C 72 00 06 00 74 007AFC8C 72 00 06 00 74 007AFC8C 74 00 74 00 65 007AFC8C 74 0	 66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr d push lbb_decrypt push dword ptr s call 0 Decomposition 0 Second	< <pre> [ebx+4] :[ebx] :[ebx] or.3E7128 :[ebp+8] otf> [ebp+8] otf> [x=]Locals [x=]Locals [x=].cocals [</pre>	3E7128: L" {%08X-%(× Files ub_3E2686 ♥ ● 001AF654 0000000 001AF658 02920000 001AF658 002920000 001AF656 0000000 001AF664 0000000 001AF664 0000000 001AF664 0000000 001AF670 0201988 001AF677 0201988 001AF677 0201038 001AF678 02923A0 001AF678 00920006	
0078010C 62 00 69 00 6C 00 69 0 003E2669 003E2667 003E267 003E267 003E267 003E267 003E267 003E267 003E2680 0000000 000000 00000000 000000000	 66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr d push dword ptr d push dword ptr s: call call yush dword ptr s: call push dword ptr s: call push dword ptr s: call push esi push esi push edi 9 <sub_3e2600+69> 4 ■ Dump 5 ● Watch 00 00 00 00 00 00 ««««««««««<</sub_3e2600+69>	<pre> [ebx+4] [ebx] [ebx] or.3E7128 [ebp+8] ntf> (rypt All Encrypted</pre>	3E7128:L"{%08X-%(> Files ub_3E2686 001AF654 001AF658 02920000 001AF658 001AF658 0000000 001AF660 001AF660 001AF668 0000000 001AF668 0000000 001AF668 0000000 001AF670 001AF67 0000000 001AF678 0000000 001AF678 0000000 001AF678 0000000 001AF678 0000000 001AF678 0000000 001AF678 0000000 001AF678 0000000 001AF678 0000000 001AF678 00000000 001AF678 0000000 001AF678 0000000 001AF678 000000000000 001AF678 000000000000000000000000000000000000	
007B010C 62 00 69 00 6C 00 69 0 003E266P 003E266P 003E267 003E267 003E267 003E267 003E267 003E267 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2680 003E2693 C 003E2693 C 003E2693 C 003E2680 003E2693 C 003E2680 003E2693 C 003E2680 003E2693 C 003E2693 C 003E2693 C 003E2680 003E2693 C 003E2680 003E2693 C 003E2680 003E2693 C 003E2680 003E2693 C 003E2693 C 003E2680 003E2680 003E2680 003E2680 003E2680 003E2693 C 003E2693 C 003E2693 C 003E2680 000000 00000 00000 00000 00000 000000	 66: FF73 04 FF33 68 <u>28713E00</u> FF75 08 E8 <u>34280000</u> E3 LockBit Black Dec All Encrypted Files All Decrypted Files 53 56 57 	push word ptr ds push dword ptr d push dword ptr d push dword ptr s: call call yush dword ptr s: call push dword ptr s: call push dword ptr s: call push esi push esi push edi 9 <sub_3e2600+69> 4 ■ Dump 5 ● Watch 00 00 00 00 00 00 ««««««««««<</sub_3e2600+69>	<pre> [ebx+4] [ebx] [ebx] or.3E7128 [ebp+8] ntf> [v= Locals [v= Locals</pre>	3E7128: L" {%08X-%(× Files ub_3E2686 ♥ ● 001AF654 0000000 001AF658 02920000 001AF658 002920000 001AF656 0000000 001AF664 0000000 001AF664 0000000 001AF664 0000000 001AF670 0201988 001AF677 0201988 001AF677 0201038 001AF678 02923A0 001AF678 00920006	•



Hex	c i														1	ASCII
00	00	00	00	00	00	00	00	00	00	00	00	17	00	00	00	
17	00	00	00	11	00	00	00	AB	AB	AB	AB	AB	AB	AB	AB	
00	00	00	00	00	00	00	00	00	00	00	00	51	4B	53	75	QKSu
BC	74	00	18	70	00	72	00	65	00	66	00	65	00	72	00	%tp.r.e.f.e.r.
73	00	79	00	73	00	74	00	65	00	6D	00	33	00			s.y.s.t.e.m.3.2.
00	00	00	00		AB				AB	AB		00	00	_		
00	00	00				_			74	00						^KSz*tPnz.
								_	_	00			00			xýz. ðz
	_	_			_	_			00	00			00	_		
		AB							00	00			00			«««««««««
		00							74	00		_	00			QKSu¼tr.e.
74	00	75	00	72	00	6E	00	66	00	6C	00	6F	00	11	00	t.u.r.n.f.l.o.w.
Hex	t –															ASCII
BD	74	00	1C	F8	EB	7A	00	50	E7	-7A	00	18	EE	3 7/	A 00) %tøëz.Pçzëz.
00	00	00	00	00	00	00	00	00	00	00	00	117	00	0 00	0 00	<u>.</u>
17												1 1 2)
1/	00	00	00	09	00	00	00	AB	AB	AB	_					
00	00	00 00	00 00	09 00	00 00	00 00	00 00	AB 00	AB 00	AB 00	AB	AB	A	A AE	3 AB	3
_	-	00 00 00			00 00 00		_				AB	AB 5D	4E	A AE	3 AB	3«««««««««« 9]KSy 9 ¼td.i.s.a.b.l.
00	-	00 00 00 77	00	00	00 00 00 00	00	00	00		00	AB 00	AB 5D 62	4E 00	8 AE	B AB 3 79	3«««««««««« 9]KSy 9 %td.i.s.a.b.l.
00 BC	00 74	00 00 00 77 73	00 18	00 64	00 00 00 00	00 69	00	00 73	00	00 61 32	AB 00 00	AB 5D 62 68	4E 00	8 AE 3 53 0 60 0 73	AE 3 79 5 00 8 00	3
00 BC 65	00 74 00	00 00 77	00 18 00	00 64 69	00 00 00 00 00 00	00 69 6E	00 00 00	00 73 33	00	00 61 32	AB 00 00	AB 5D 62 68 61	AE 4E 00 00	8 AE 3 53 0 60 0 73	AB 3 79 00 3 00 00 00	3
00 BC 65 79	00 74 00 00	00 00 77 73	00 18 00 00	00 64 69 74	00 00 00 00 00 AB	00 69 6E 65	00 00 00 00	00 73 33 6D	00 00 00 00	00 61 32 63 00	AB 00 00 00 00	AB 5D 62 68 61 00	AE 4E 00 00 00	AE 3 53 0 60 0 73 0 60	AE 3 79 3 00 3 00 5 00 0 00	3
00 BC 65 79 6C	00 74 00 00 00	00 00 77 73	00 18 00 00 00	00 64 69 74 00	00 00 00 00 00 AB 4B	00 69 6E 65 00	00 00 00 00 00	00 73 33 6D 00	00 00 00 00 00	00 61 32 63 00 AB	AE 00 00 00 00 00 AE	AB 5D 62 68 61 00 60 60	AE 4E 00 00 00 00 00	AE 3 53 5 60 5 73 5 60 5 73 5 60 5 73 5 60 5 73 5	AE 3 79 3 00 3 00 5 00 0 00	3]KSy 9]KSy 9 ¼td.i.s.a.b.l. 9 e.w.i.n.3.2.k.s. 9 y.s.t.e.m.c.a.l. 9 l.s
00 BC 65 79 6C 00 00	00 74 00 00 00 00 00	00 00 77 73 73 00	00 18 00 00 00 00 00 00	00 64 69 74 00 AB 5E		00 69 6E 65 00 AB 53	00 00 00 00 00 AB 7A	00 73 33 6D 00 AB	00 00 00 00 00 AB	00 61 32 63 00 AB 00	AB 00 00 00 00 00 00 AB 10	AB 5D 62 62 68 61 00 61 00 80 61 00 80	AE 4E 00 00 00 00 00 00 00 00	AE 3 5 3 0 60 0 7 3 0 60 0 00 0 00 0 00 0 00	AE 3 79 3 00 3 00 3 00 0 00 0 00 0 00 0 00	3]KSy 9]KSy 9]KSy 9]KSy 9]KSy 9 9 9
	00 17 00 8C 73 00 78 00 78 00 78 00 78 00 78 00 74 Hex BD 00	17 00 00 00 BC 74 73 00 00 00 00 00 78 FD 00 00 AB AB 00 00 74 00 Hex BD 74	00 00 00 17 00 00 00 00 00 BC 74 00 73 00 79 00 00 00 00 00 00 78 FD 7A 00 00 00 AB AB AB 00 00 00 74 00 75 Hex BD 74 00	00 00 00 00 17 00 00 00 00 00 00 00 BC 74 00 18 73 00 79 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 AB AB AB AB 00 00 00 00 74 00 75 00 Hex BD 74 00 1C	00 00 00 00 00 17 00 00 00 11 00 00 00 00 00 BC 74 00 18 70 73 00 79 00 73 00 00 00 00 AB 00 00 00 00 5E 78 FD 7A 00 AB 00 00 00 00 0E AB AB AB AB AB 00 00 00 00 51 74 00 75 00 72 Hex BD 74 00 1C F8	00 00 00 00 00 00 17 00 00 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 BC 74 00 18 70 00 73 00 00 00 00 00 00 AB AB 00 00 00 00 05 E 4B 78 FD 7A 00 AC FO 00 00 00 00 0E 00 AB AB AB AB AB AB 00 00 00 00 51 48 74 00 75 00 72 00	00 00 00 00 00 00 17 00 00 00 11 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 BC 74 00 18 70 00 72 73 00 79 00 73 00 74 00 00 00 00 55 48 53 78 FD 7A 00 AD FO 7A 00 00 00 00 00 00 00 00 AB AB AB AB AB AB AB AB 00 00 00 00 00 51 48 53 74 00 75 00 72 00 6E Hex	00 00 00 00 00 00 00 17 00 00 00 11 00 00 00 17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 18 70 00 72 00 73 00 79 00 73 00 74 00 00 00 00 00 00 AB AB AB AB 00 00 00 00 5E 48 53 7A 00 00 00 00 0E 00 00 00 AB AB AB AB AB AB AB AB 00 00 00 00 51 48 53 75 74 00 75	00 00 00 00 00 00 00 00 17 00 00 00 11 00 00 00 AB 00 00 00 00 00 00 00 00 AB 00 00 00 00 00 00 00 00 00 BC 74 00 18 70 00 72 00 65 73 00 74 00 65 73 00 74 00 65 00 00 00 00 AB AB AB AB AB 00 00 00 00 SE 4B 53 7A B3 78 FD 7A 00 AO FO 7A 00 00 00 00 00 00 SI 4B AB AB AB AB AB AB	00 00<	00 00<	00 00<	00 00 00 00 00 00 00 00 00 00 00 00 17 17 00 00 00 11 00 </td <td>00 00 00 00 00 00 00 00 00 00 00 00 17 00 17 00 00 00 11 00<!--</td--><td>00 00<</td><td>00 00<</td></td>	00 00 00 00 00 00 00 00 00 00 00 00 17 00 17 00 00 00 11 00 </td <td>00 00<</td> <td>00 00<</td>	00 00<	00 00<

In the context of the analysis of the file %s.README.txt (where %s stands for string variable which is related to a pattern contained the filename) by the decryptor, is called the function RtIFreeHeap to free a memory block allocated from an heap.





ebp=0135F4B4

Call from sub_24205E+F, 2424B9, 2424FA, sub_24256C+76, 242772, 242807, 2428DF, sub_24290.text:00241FF4 lbb_decryptor.exe:\$1FF4 #13F4 <sub_241FF4>

🚛 Dump 1	🚚 Dump 2	🚛 Dump 3	🚛 Dump 4	🚛 Dump 5	👹 Watch 1 🛛 [x=] L	.ocals 🛛 🖉 া 🕨
Address He	ex .				ASCII	
			F0 2C 40 7C			
			9A 00 9C 00			
0135F4D4 34		00 00 00 00				
					005(044.	
			<u>90 33 34 00</u>			
			00 10 00 00			
	00 00 00	F8 62 24 00	02 00 00 00	00 00 00 00	øb\$	
0135F524 00	00 00 00	00 00 00 00	00 00 00 00	00 00 00 00		
0135F534 00		00 00 00 00	00 00 00 00			
0135F544 00	00 00 00	00 00 00 00	00 00 00 00	00 00 00 00		
0135F554 00		00 00 00 00	00 00 00 00	00 00 00 00		
0135F564 00	00 00 00	00 00 00 00	28 00 00 00	00 60 17 01		× 1

It is important to underline that, in the context of files loop gathering to obtain the details of the files to decrypt on the disks of the compromised machine, it is created an execution thread



which points to the function **sub_403FA8**, which calls the function **sub_402EA4**. After the execution of the functions of the thread is present the instruction **jz short loc_4041CE** and the "switch" is managed through the offsets **aDecryptedUFile** and **aDecryptedUFile_0** based on the result get from the thread execution:

	• • • • • • • • • • • • • • • • • • •
🚺 🛃 🖉	
loc_404 mov mov push push push push call mov cmp jnz	H115: ds:dword_407F4C, 1 ds:dword_407F50, 0 ds:dword_407F60, 0 0 ; lpThreadId 4 ; dwCreationFlags 0 ; lpParameter offset sub_403FA8 ; lpStartAddress 0 ; dwStackSize 0 ; lpThreadAttributes CreateThread ds:hHandle, eax ds:hHandle, 0 short loc_40415A
	loc_40415A: ; wAttributes push 0Ah push offset aDecryptingFile ; "\r\n\nDecrypting Files In Progress\r" call sub_4013A8 push 0Fh ; wAttributes push offset aWait 0 : "Wait\r\n"

call sub_4013A8 push 0Fh push offset aWai: call sub_4013A8	<pre>ryptingFile ; "\r\n\nDecrypting Files In Progress\r" ; wAttributes t_0 ; "Wait\r\n" ; hThread d</pre>
	<pre>loc_40417D: ; dwMilliseconds push 0Ah push ds:hHandle ; hHandle call WaitForSingleObject cmp eax. 102h jz short loc_4041CE</pre>
•	
<mark>sh</mark> ds:hHandle ; hObject 11 CloseHandle	loc 4041CE: ; Format
ll CloseHandle sh ds:dword 407F60 ; Format	push ds:dword 407F60
<pre>sh offset aDecryptedUFile ; "Decry</pre>	
a eax, [ebp+Butter] sh eax : Buffer	lea eax, [ebp+Butter] bush eax : Buffer



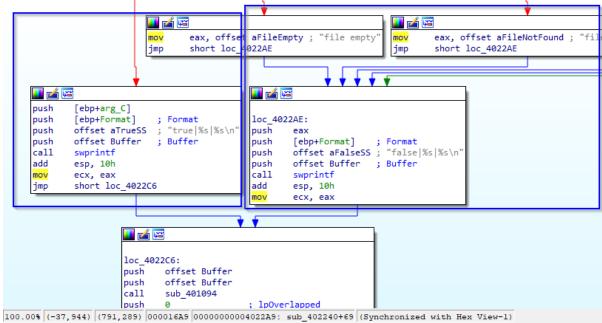
sub_403FA8 proc near	
lpThreadParameter= dword ptr 4	
<pre>push 2 ; nPriority push 0FFFFFFEh ; hThread call SetThreadPriority call sub_402EA4 cmp ds:dword_407F4C, 0 jz short locret_403FDD</pre>	
call sub_402900 mov ds:dword_407F4C, 0 mov ds:dword_407F50, 1 call sub_402EA4	
100.00% (-152,75) (621,411) 000033B1 000000000403FB1: sub 403FA8+9 (Synchronized with Hex View-1)	

The function **sub_402EA4** is fundamental because a check operation is performed with a sort of "execution key", infact it is present a chain of cmp and jnz instructions that point to others labels (for example loc_402F01).

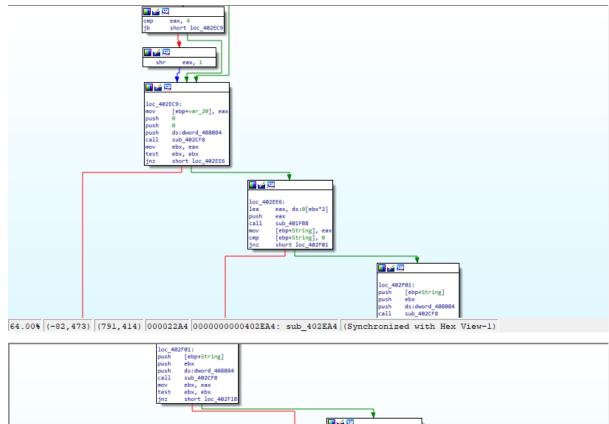


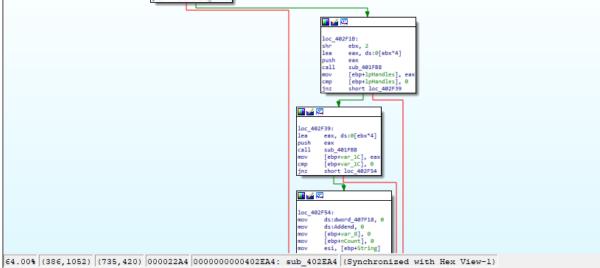
. Attributor, by barod form
; Attributes: bp-based frame
sub_402EA4 proc near
var_20= dword ptr -20h
var_1C= dword ptr -1Ch
lpHandles= dword ptr -18h
String= dword ptr -14h
lpParameter= dword ptr -10h
nCount= dword ptr -0Ch
var_8= dword ptr -8
Handle= dword ptr -4
push ebp
mov ebp, esp
add esp, 0FFFFFE0h
push ebx
push ecx
push edx
push esi
push edi
xor ebx, ebx
mov [ebp+String], ebx
call sub_401428
cmp ds:dword_407F4C, 0
jz short loc_402EC9

100.00% (-104,27) (621,418) 000022A4 00000000002EA4: sub_402EA4 (Synchronized with Hex View-1)



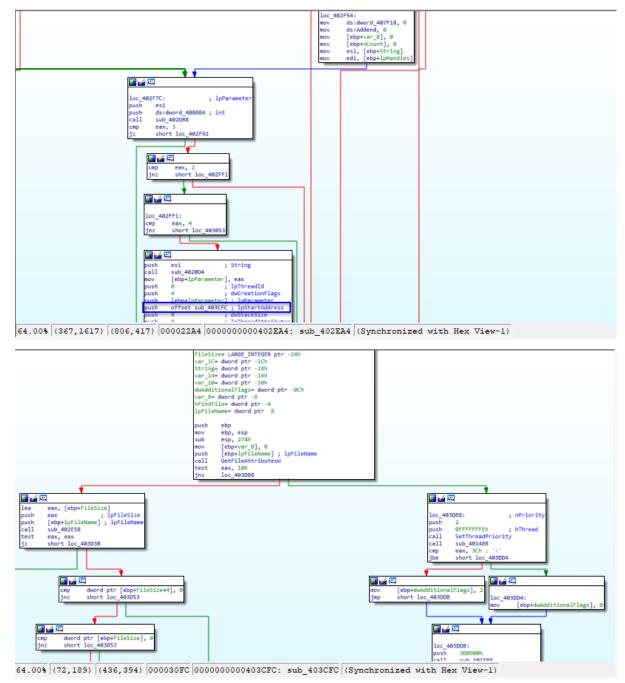




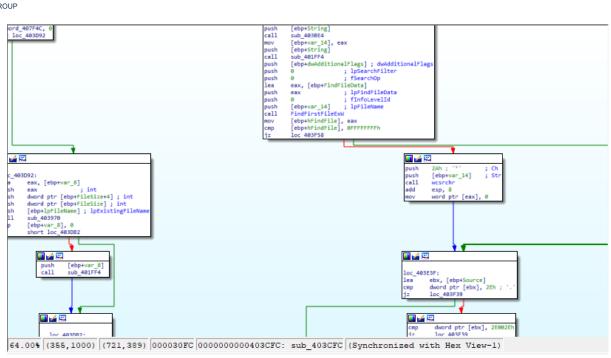


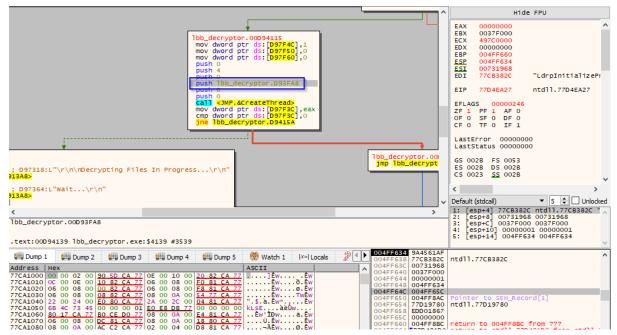
After the execution of the check instructions and validation we arrive to the point of the execution which calls the function **sub_403CFC**, which effectively starts the operations of files gathering (note the function **FindFirstFileExW** and the wildcard "*"), and then continue with the obtaining of the attributes of the encrypted files:













00093F96 × 74 08 100093F98 00093F98 FF75 F8 push dword ptr ssi[ebp-8] 00093F98 > 8855 mov esp.ebp 00093F40 > 64 02 push PFFFFFE 00093F41 > 68 250000 call <100 decryptor.sub_092EA4> 00093F81 > 88 EEEEFFFF call <100 decryptor.sub_092EA4> 00093F81 > 88 250000 call <100 decryptor.sub_092EA4> 00093F81 > 88 250000 call <100 decryptor.sub_092EA4> 00093F81 > 88 25770000 call <100 decryptor.sub_092EA4> 00093F81 > 88 25770000 call <100 decryptor.sub_092EA4> 00093F81 > 88 57570000 call <100 decryptor.sub_092EA4> 00093F81 > 88 57570000 call <100 decryptor.sub_092EA4> 00093F81 > 88 55570000 call <100000 00093F81 > 88 55 mov edi.edi 00093F81 > 88 55	SUD_D93FA6	Hide FPU EAX 00000000 EBX 0037F000 ECX 497C0000 EDX 00000000 EBP 004FF630 ESP 004FF634 ESI 00731966 EDI 77C94EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 OF 0 SF 0 DF 0 CF 0 TF 0 IF 1 LastStaterror 00000000 LastStatey 00000000							
00093FE9 . 53 push ebx 00093FE8 . 51 push ecx 00093FE8 . 52 push ecx 00093FE0 . 56 push ecx 00093FE0 . 57 push edi 00093FE0 . 58 4C720900 push edi 00093FF1 . 58 42720900 push edi 00093FF3 . 58 4720000 call KOMP.4SetConsoleTitlews 00093FF4 . 58 F7100000 call KOMP.4SetConsoleTitlews . call KOMP.4SetConsoleTitlews . call KOMP.4SetConsoleTitlews . call KOMP.4SetConsoleTitl	edi:"LdrpInitia D9724C:L"LockBi > 004FF634 9A4561AF 004FF638 77CB382C 004FF638 00731968	GS 0028 FS 0053 ES 0028 DS 0028 CS 0023 <u>SS</u> 0023 CS 0023 <u>SS</u> 0023							
Add 11 Add 11 Add 11 77CA1000 0000 0000 000	004FF640 0037F000 004FF644 0000001 004FF648 004FF634 004FF650 004FF635 004FF650 004FF88C 004FF658 EDD01867 004FF650 004FF88C	Pointer to SEH_Record[1] ntdll.77D19780 return to 004FF88C from ??? return to ntdll.77D491D0 from ntdll.:							
mov edi,edi push 2 push FFFFFFE call FFFFFFE	push 2 push FFFFFFE call <jmp.&setthreadpriority> [call <idb_decryptor.sub_d92ea4>] cmp_dword_ptr_ds:[D97F4C],0</idb_decryptor.sub_d92ea4></jmp.&setthreadpriority>								
1bb_decryptor.00093FBF call <1bb_decryptor.sub_D93	0								



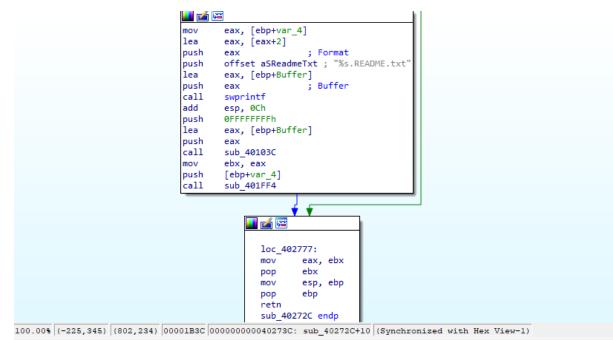
import of the set of the				Hide FPU
e developing the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i general prime the developing the developing the developing i </td <td></td> <td>push ebp mov ebp,esp add esp,FFFFFE0 push ebx push ecx push ecx push edi ; edi:"LdrpInitializeProcess" xor ebx,ebx mov dword ptr ss:[ebp-14],ebx call <lbb_decryptor.sub_d91428s cmp dword ptr ds:[D97F4C],0 je lbb_decryptor.00D92EC2 cmp eax,4 jb lbb_decryptor.0D92EC2 lbb_decryptor.00D92EC2</lbb_decryptor.sub_d91428s </td> <td></td> <td>EBX 0037F000 ECX 497C0000 EDX 00000000 ESP 004FF630 EDI 77CB382C "LdrpInitializePr EIP 77CB382C "LdrpInitializePr EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 0F 0 SF 0 DF 0 CF 0 TF 0 IF 1 LastError 00000000 LastStatus 00000000 LastStatus 00000000 CG 002B FS 0053 ES 002B FS 0053 ES 002B DS 0028 CS 0023 <u>SS</u> 0028</td>		push ebp mov ebp,esp add esp,FFFFFE0 push ebx push ecx push ecx push edi ; edi:"LdrpInitializeProcess" xor ebx,ebx mov dword ptr ss:[ebp-14],ebx call <lbb_decryptor.sub_d91428s cmp dword ptr ds:[D97F4C],0 je lbb_decryptor.00D92EC2 cmp eax,4 jb lbb_decryptor.0D92EC2 lbb_decryptor.00D92EC2</lbb_decryptor.sub_d91428s 		EBX 0037F000 ECX 497C0000 EDX 00000000 ESP 004FF630 EDI 77CB382C "LdrpInitializePr EIP 77CB382C "LdrpInitializePr EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 0F 0 SF 0 DF 0 CF 0 TF 0 IF 1 LastError 00000000 LastStatus 00000000 LastStatus 00000000 CG 002B FS 0053 ES 002B FS 0053 ES 002B DS 0028 CS 0023 <u>SS</u> 0028
C Description Description Description Description 1 image and provide in the set of the		Shr eax,1		
1:ext:0002561 lbb_decryptor.ext:5261 #2261 ext:0002561 lbb_decryptor.ext:5261 #2261 ext:0002561 lbb_decryptor.ext:5261 #2261 ext:0002561 lbb_decryptor.ext:5261 #2261 #2671	<		>	1: [esp+4] 77CB382C ntd]] 77CB382C "
1: Ext::00092E81 1bb_decryptor.exe: 12E81 #22E1 <cub_92e4+e0< td=""> \$\$1 [255:14] 00#FE33 00#FE33 ##Dowp 1 #Dowp 2 #Dowp 3 #Dowp 4 #Dowp 5 @ Weth 1 M=Locale \$\$1 [255:14] 00#FE33 00#FE33 #772A000 M8 00 00 00 00 00 00 00 00 00 00 00 00 00</cub_92e4+e0<>	dword ptr ss:[ebp-14]=[004FF64C]=004F ebx=0037F000	F65C		3: [esp+C] 0037F000 0037F000 4: [esp+C] 00000001 00000001
Address Less Ad	.text:00D92EB1 1bb_decryptor.exe:\$2EB	1 #22B1 <sub_d92ea4+d></sub_d92ea4+d>		5: [esp+14] 004FF634 004FF634
772A1830 (b 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Address Hex 77CA1000 00 00 02 00 90 5D CA 77 0E 0 77CA1010 0C 00 0E 00 10 82 CA 77 06 0	ASCII 0 10 00 <u>20 82 CA 77</u> ∗]ÊwÊw 0 08 00 <u>F0 81 CA 77</u> ×Êw	004FF63C 00731968 004FF640 0037F000 004FF644 00000001 004FF648 004FF634	ntdll.77CB382C 3 4
• 00032550 • 0003254 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 0003255 • 00003255 • 0000000 • 0000000	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	0 0A 00 <u>54 77 CA 77</u> Êw Têw. 0 CO 00 <u>04 81 CA 77</u> '. S.a. Êw Têw 8 <u>DB 77</u> 00 00 00 00 kLsE àèÛw 0 0A 00 <u>54 81 CA 77</u> Û. Êw Êw 0 0A 00 <u>DS 81 CA 77</u> Âw Êw 0 04 00 <u>DS 81 CA 77</u> Âw Êw	004FF650 004FF840 004FF654 77D19780 004FF658 EDD01865 004FF650 0000000 004FF660 004FF860 004FF664 77D491D	Pointer to SEH_Record[1] ntdl1.77D19780 - return to 004FF8BC from ??? oreturn to ntdl1.72D491D0 from ntdl1.
•••••••••••••••••••••••••••••	77CA10A0 00 00 00 00 57 14 01 E2 46 1	5 C5 43 A5 FE 00 8DWâF.ÅC¥þ		
• 00092EAC • 526 • 556 push edx push push edx push edx	77CA10A0 00 00 00 00 00 00 00 00 00 00 00 00	5 C5 43 A5 FE 00 8DWâF.ÁC¥þ 7 CA 77 01 00 00 00 îâÓðØwÉw		
 	77CA10A0 00 00 00 00 00 01 57 14 01 E2 46 1 77CA10B0 EE E3 D3 F0 06 00 00 00 00 00 00 08 7 00092EA0 50 00092EA1 5 55 00092EA4 5 55 00092EA4 5 55 00092EA7 53	5 C5 43 A5 FE 00 8DWär.ACYD 7 CA ZZ 01 00 00 00 jädðØwew pop ebp pop ebp ret 8 push ebp mov ebp.esp add esp.FFFFFE0 push ebx	< <	Hide FPU EAX 0000000 EBX 0037F000 ECX 497C0000 EDX 0000000
Image: Construction of the set of t	77CA10A0 00 00 00 00 57 14 01 E2 46 1 77CA10B0 EE E3 D3 F0 06 00 00 00 00 00 00 28 7 00D 92E9E . 88E5 00D 92EA4 50 00D 92EA4 51 00D 92EA4 53 00D 92EA5 55 00D 92	5 C5 43 A5 FE 00 8DWär.AC¥P CA ZZ 01 00 00 00 jädöØwEw mov esp.ebp pop ebp pop ebp push ebp add esp.FFFFFE0 push ebx push ecx push ecx push edx push edi xor elx.ebx	< <	Hide FPU EAX 0000000 EBX 0037F000 ECX 497C0000 EDX 00000000 EBX 00000000 EBX 0037F060 EST 004FF661 EST 004FF654 EDI 77CB382C "LdrpInitializePr
• 000 922DB • 000 922DB • 000 922DB • 000 922DF • 000 922DF • 000 922EF • E 9 D2010000 • E 8 CSF0FFFF • call <1bc.decryptor.b39308 1ea eax, dword ptr ds:[ebx*2] vond	77CA1080 00	5 C5 43 A5 FE 00 8D	v <	Hide FPU EAX 00000000 EBX 0037F000 ECX 497C0000 EDX 00000000 ESP 004FF634 ESI 00731968 EDI 77C8382C "LdrpInitializer EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 0F 0 SF 0 DF 0
	77CA10A0 0 00 0 00	5 C5 43 A5 FE 00 8D	v c	Hide FPU EAX 00000000 EBX 0037F000 ECX 497C0000 EDX 00000000 ESP 004FF650 EST 00731568 EDI 77C8382C "LdrpInitializePr EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 OF 0 SF 0 DF 0 CF 0 TF 0 IF 1 LastError 00000000
dword ptr ss:[ebp-14]=[004FF64C]=004FF65C 2: [esp+3] 00731968 dword ptr ss:[ebp-14]=[004FF64C]=004FF65C 2: [esp+3] 00731968 itext:0092EB1 lbb_decryptor.exe:\$2EB1 #22B1 <sub_02e44+b> 2: [esp+14] 004FF634 upunp 1 upunp 2 upunp 3 upunp 4 upunp 5 W Watch 1 lx=locals 004FF638 00750508 l*indows/\System32\\CRYPT32.dll" Address Hex 004FF638 00000000 00000000 00000000 0000</sub_02e44+b>	77CA10A0 00	5 C5 43 A5 FE 00 8D	v c	Hide FPU EAX 00000000 EBX 0037F000 ECX 497C0000 EDX 0000000 ESP 004FF650 EST 00731968 EDI 77CB382C "LdrpInitializer EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 OF 0 SF 0 DF 0 CF 0 TF 0 IF 1 LastError 00000000 LastStatus 0000000 GS 0028 FS 0053 ES 0028 DS 0028
.text:00092EB1 10b_decryptor.exe:52E81 #22B1 <sub_92ea4ho></sub_92ea4ho>	77CA10A0 00	5 C5 43 A5 FE 00 8D , W är. ACYP 7 C4 72 01 00 00 00 jiaob 0wfw mov esp,ebp pop ebp ret push ebp mov dep,esp add esp, FFFFFFE0 push ebp push ect push esi push ect push esi push esi push esi push esi push esi push esi add esp,ebp Push esi push esi B300 00 FF Call <1bb_decryptor.sub_991428	v c	Hide FPU EAX 0000000 EBX 0037F000 ECX 497C0000 EDX 0000000 EBP 004FF634 EDI 77068382C EIP 7704EA27 EFLAGS 00000246 ZF1 PF1 AF0 0F0 CF0 0F1 LastError 00000000 LastStatus 00000000 GS 002B CS<002B
ewp Cump 2 ewp Cump 2 <td>77CA10A0 00</td> <td>5 C5 43 A5 FE 00 8D </td> <td>v c</td> <td>Hide FPU EAX 00000000 EBX 0037F000 ECX 437C0000 EDX 00000000 ESP 004FF634 ESI 00731968 EDI 77C8832C "LdrpInitializer EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 0F 0 SF 0 DF 0 CF 0 TF 0 IF 1 LaStError 00000000 LastStatus 000000000 GS 0028 FS 0053 ES 0028 DS 0028 CS 0028 DS 0028 CS 0023 SS 0028 C Default (stdcal) 1: [esp+4] 77C8382C ntdll.77C8382C if 0 3: [esp-C] 0037F000 0037F000</td>	77CA10A0 00	5 C5 43 A5 FE 00 8D	v c	Hide FPU EAX 00000000 EBX 0037F000 ECX 437C0000 EDX 00000000 ESP 004FF634 ESI 00731968 EDI 77C8832C "LdrpInitializer EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 0F 0 SF 0 DF 0 CF 0 TF 0 IF 1 LaStError 00000000 LastStatus 000000000 GS 0028 FS 0053 ES 0028 DS 0028 CS 0028 DS 0028 CS 0023 SS 0028 C Default (stdcal) 1: [esp+4] 77C8382C ntdll.77C8382C if 0 3: [esp-C] 0037F000 0037F000
0263F888 00 00 00 00 00 00 00 00 00 00 00 00 00	77CA10A0 00	5 C5 43 A5 FE 00 8D	v < sub_D92EA4	Nide FPU EAX 00000000 EBX 0037F000 EBX 0037F000 EAX 0000000 EDX 00071986 EDI 77D4EA27 FLAGS 00000000 LastError 00000000 Color 5 Default (stdcall) TCB322C 0037F000 Lespt-1 0037F000 Lespt-1 0037F000 Lespt-1 004FF634 Lespt-1 004FF634
0263FBF8 00 00 00 00 00 00 00 00 00 19 04 A7 76 00 00 00 00 \$vhuc\$v	77CA10A0 00	S CS 43 AS FE 00 8D, w är. ACYD CA 22 01 00 00 00 1 à00 @wEw mov esp,ebp pop ebp res 8 pop ebp res 8 push esp push esp push esp push ecx push ecx push esi xor ebx,ebx fFF CAT Call Chock Decryptor All Encrypted Files 0 All Decrypted Files 0 All Decrypted Files 0 Call Chock Bisk Decryptor FF CALL Concertion Co		Hide FPU EAX 00000000 EBX 0037F000 ECX 437C0000 EDX 00000000 ESP 004FF634 EDI 77C8832C "LdrpInitializer EIP 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 0F 0 SF 0 DF 0 CF 0 TF 0 IF 1 LaStError 00000000 LastStatus 000000000 GS 0028 FS 0053 ES 0028 DS 0028 CF 0 TF 0 IF 1 LaStError 0000000 GS 0028 FS 0053 ES 0028 DS 0028 Default (stdcal) 1: [esp+4] 77C8382C ntdll.77C8382C "Unlocked I: [esp-10] 00000001 0000001 S: [esp-11] 004FF634 004FF634 2: [esp-11] 004FF634 004FF634 2: [esp-11] 004FF634 004FF634 2: [esp-11] 004FF634 004FF634
Command: Commands are comma separated (like assembly instructions): mov eax, ebx Default	77CA10A0 00	5 C5 43 A5 FE 00 8D		Hide FPU EAX 00000000 EBX 0037F000 ECX 437C0000 EDX 00000000 EDX 00000000 ESP 004FF634 ESI 00731986 EDI 77CB382C "LdrpInitializer EIF 77D4EA27 ntdll.77D4EA27 EFLAGS 00000246 ZF 1 PF 1 AF 0 0F 0 SF 0 DF 0 CF 0 TF 0 IF 1 LaStError 00000000 LaStStatus 000000000 LaStStatus 000000000 GS 0028 FS 0053 ES 0028 DS 0028 C 0023 SS 0028 C Default (stdcall) J Esp 14] 07CB382C ntdll.77CB382C 1: [esp+4] 77CB382C ntdll.77CB382C 2: [esp+1] 004FF634 004FF634 3: [esp-10] 0000001 00000001 S: [esp-14] 004FF634 004FF634 3: [esp-14] 004FF634 004FF634 4: [esp-14] 77CB382C from ntdll. Feturn to ntdll.77CB3530 from ntdll. Pointer to SEH_Record[1] ntdll.77D19780 4L"indows\\System32\\CRYPT32.dll"

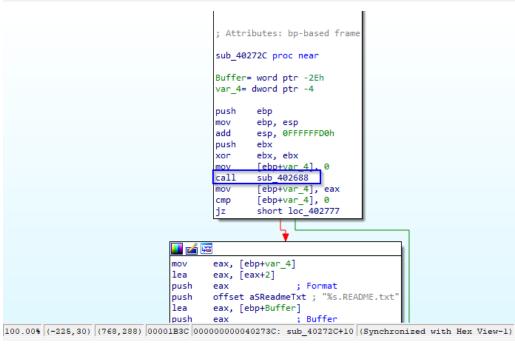


	; Attributes: bp-based frame
	sub_40272C proc_near
	Buffer= word ptr -2Eh var_4= dword ptr -4
	<pre>push ebp mov ebp, esp add esp, 0FFFFFD0h push ebx xor ebx, ebx mov [ebp+var_4], 0</pre>
	call sub_402688 mov [ebp+var_4], eax cmp [ebp+var_4], 0 jz short loc_402777
🛄 🛃 [
mov	eax, [ebp+var_4]
lea push	eax, [eax+2] eax ; Format
pusn	

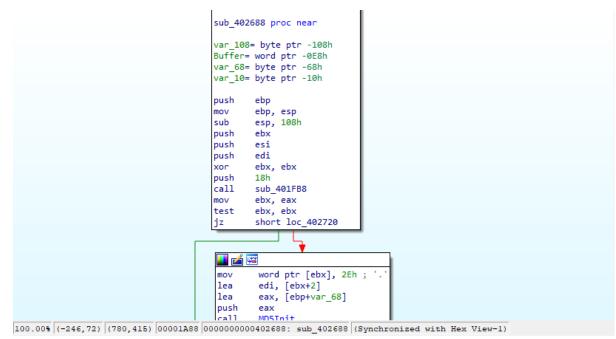
At the moment in which the content of the file README is read is possible to highlight how the function **sub_402688** is called, which consequently calls the function of hashing digest (specifically the MD5 context is created) and, then, the function **sub_4011DC** is executed:

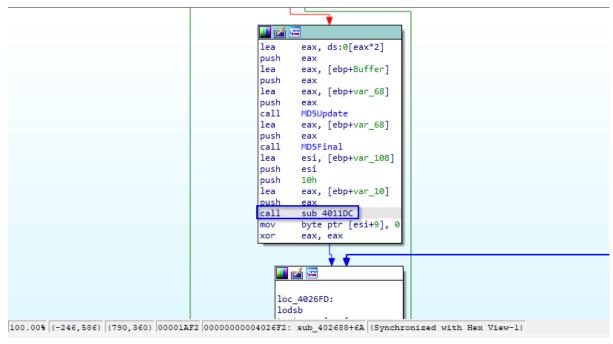








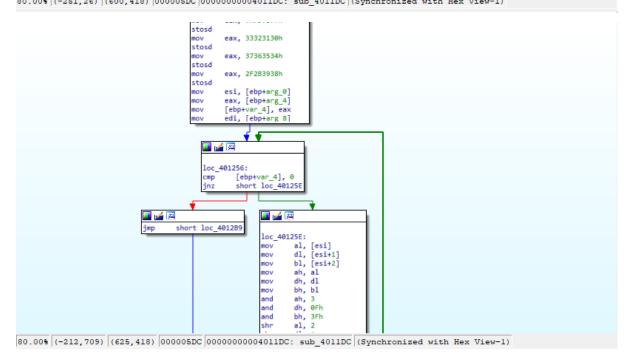




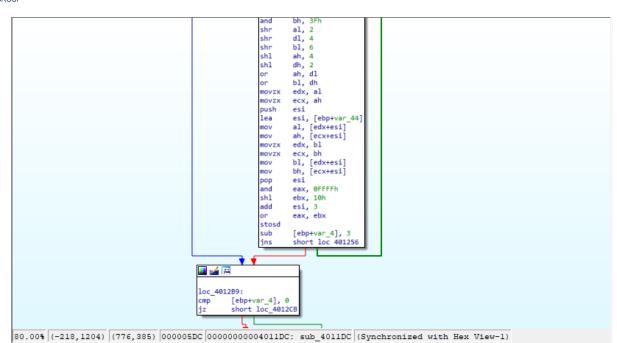
In this function there are insertions of plenty of **hardcoded** values which are added to the eax register. Then various AND and OR operations are executed as follows, after a compare and jnz construct to the label loc_401256. In particular the values are referred to a dictionary, in hexadecimal:

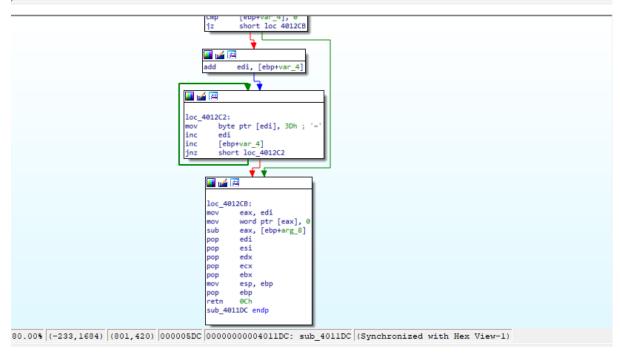


	; Attributes: bp-based frame
	sub_4011DC proc near
	var 44= byte ptr -44h
	var 4= dword ptr -4
	arg_0= dword ptr 8
	arg_4= dword ptr 0Ch
	arg_8= dword ptr 10h
	push ebp
	mov ebp, esp
	add esp, 0FFFFFBCh
	push ebx
	push ecx
	push edx push esi
	push edi
	lea edi, [ebp+var 44]
	mov eax, 44434241h
	stosd
	mov eax, 48474645h
	stosd mov eax, 4C484A49h
	stosd
	mov eax, 504F4E4Dh
	stosd
	mov eax, 54535251h
	stosd mov eax, 58575655h
	stosd
	mov eax, 62615A59h
	stosd
80 00% (-251 26) (600 418) 000005DC 000	000000004011DC: sub 4011DC (Synchronized with Hex View-1)









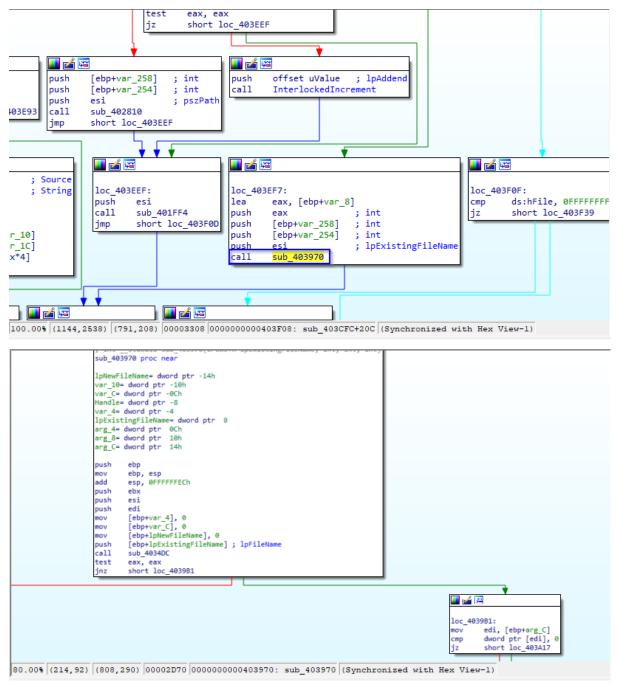
Swascan
TINEXTA GROUP

Input 44434241h

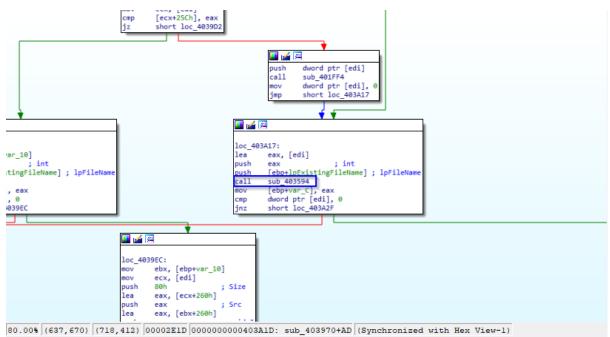
Output	×××	
DCBA		

Here are the details of the function calling in the context of files gathering for the setting of the file pointer and the attributes of the files taken into consideration.



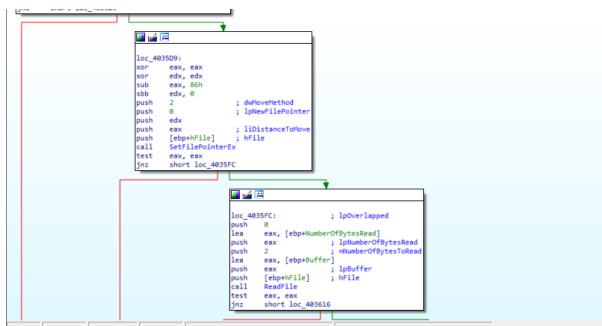


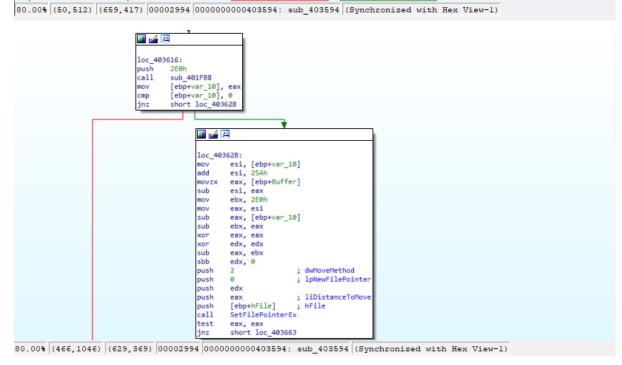




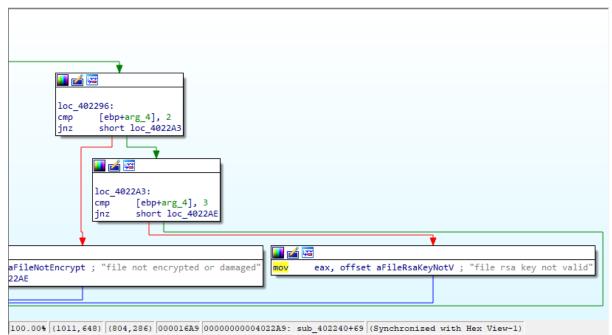
	: Attributes: bp-based frame	
	; Accribuces: op-based frame	
	; int stdcall sub 403594(LPCWSTR lpFileName, int)	
	sub 403594 proc near	
	var_10= dword ptr -10h	
	Buffer= word ptr -0Ah	
	NumberOfBytesRead= dword ptr -8	
	hFile= dword ptr -4	
	lpFileName= dword ptr 8 arg 4= dword ptr 0Ch	
	arg_4= dword per och	
	push ebp	
	mov ebp, esp	
	add esp, ØFFFFFFØh	
	push ebx	
	push esi	
	mov [ebp+var_10], 0	
	mov [ebp+NumberOfBytesRead], 0	
	mov [ebp+hFile], ØFFFFFFFh	
	push 0 ; hTemplateFile	
	push 80h ; dwFlagsAndAttributes push 3 ; dwCreationDisposition	
	push 0 ; lpSecurityAttributes	
	push 0 ; dwShareMode	
	push 80000000h ; dwDesiredAccess	
	push [ebp+lpFileName]; lpFileName	
	call CreateFileW	
	mov [ebp+hFile], eax	
	cmp [ebp+hFile], ØFFFFFFFh	
	jnz short loc 4035D9	
	The second se	
0 008 (-32	30) (619 420) 00002994 000000000403594	wh 403594 (Sunchronized with Hey View-1)





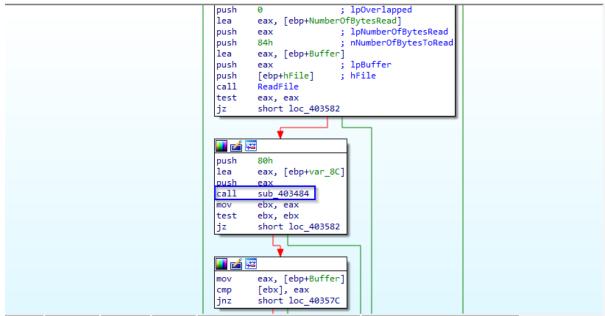




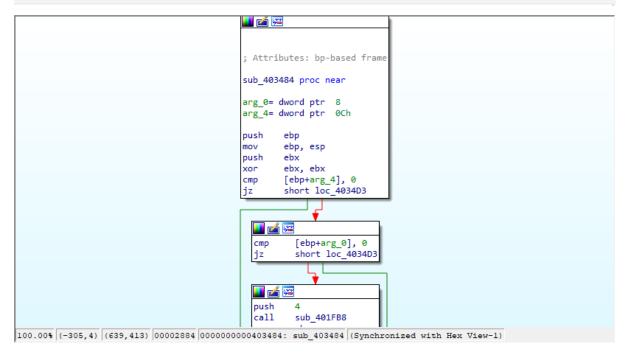


When the encrypted files in input are read, the function **sub_403484** is executed, then a buffer is called and inserted in the eax register. The function sub_403484 contains XOR operations and calls for 3 times (intermittently to bswap operations) the function *sub_401334*. The *bswap* instruction permits to perform the swap of bytes taken in input considered during the execution.

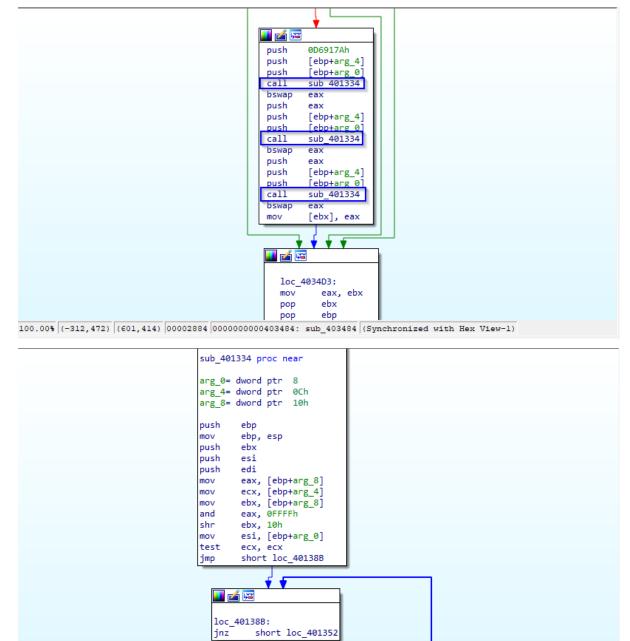




100.00% (-225,813) (660,413) 00002962 00000000403562: sub_4034DC+86 (Synchronized with Hex View-1)







Follows, instead, the execution of the **XOR** instructions for the value **20039FEFh**:

100.00% (-157,77) (656,364) 00000734 000000000000001334: sub_401334 (Synchronized with Hex View-1)



inc add

dec

jnz

ebx, eax

edi

edx

edi

eax

eax, ebx

edx, edx

ebx, edx

ecx, ecx

short loc 401365

edi, 20029FE0h

edi, 20039FEFh edx, edx

edi

🚺 🚄 🔛

mov xor

xor div

push

mov

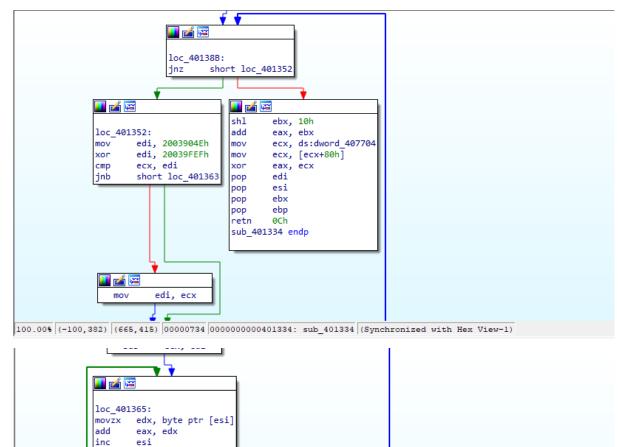
xor

div

mov

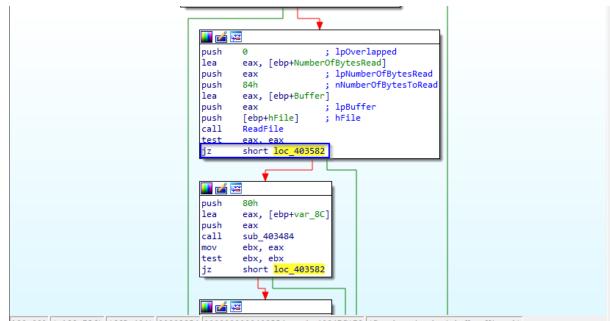
рор

test

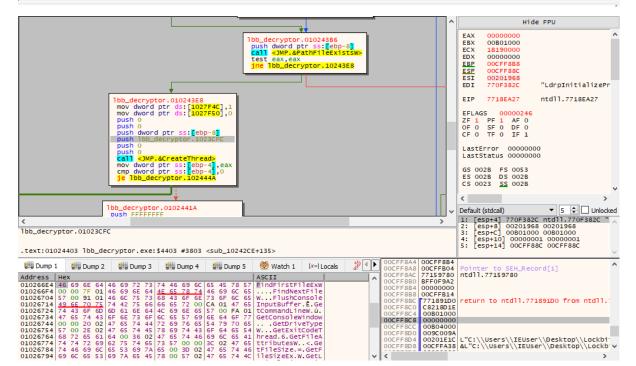


100.00% (-105,872) (672,419) 00000734 00000000401334: sub_401334 (Synchronized with Hex View-1)

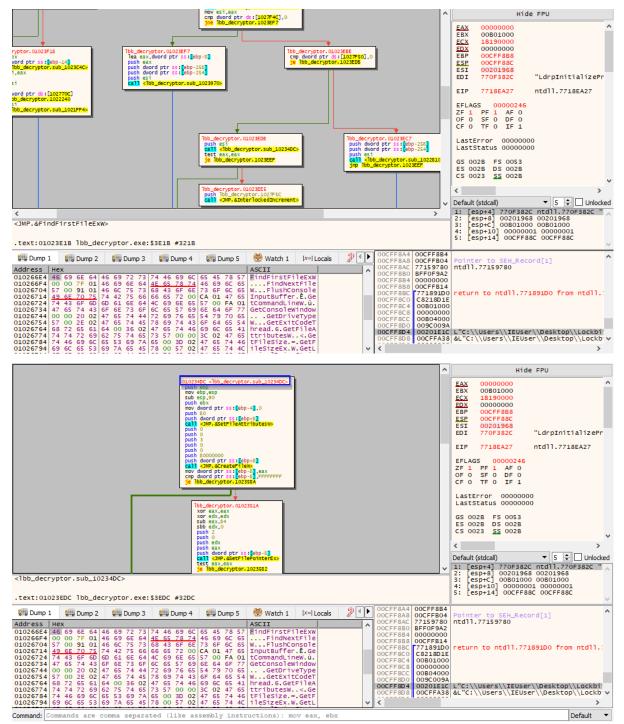




100.00% (-199,756) (665,404) 00002954 00000000403554: sub_4034DC+78 (Synchronized with Hex View-1)

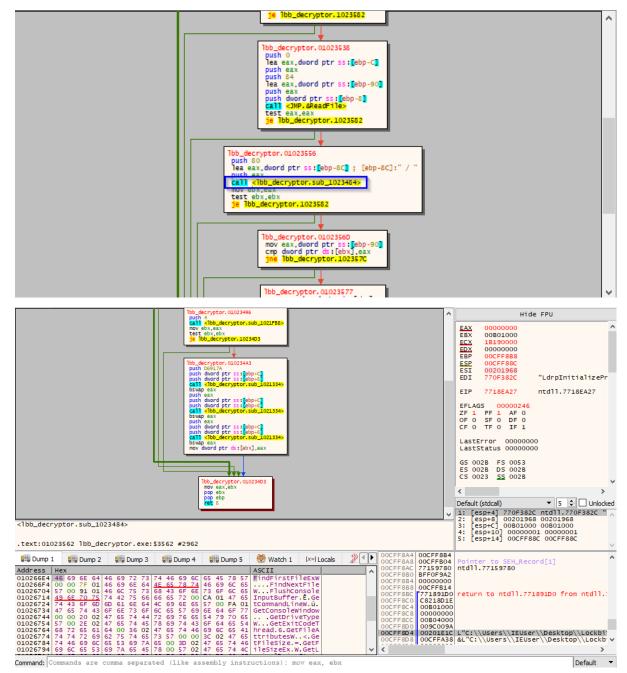






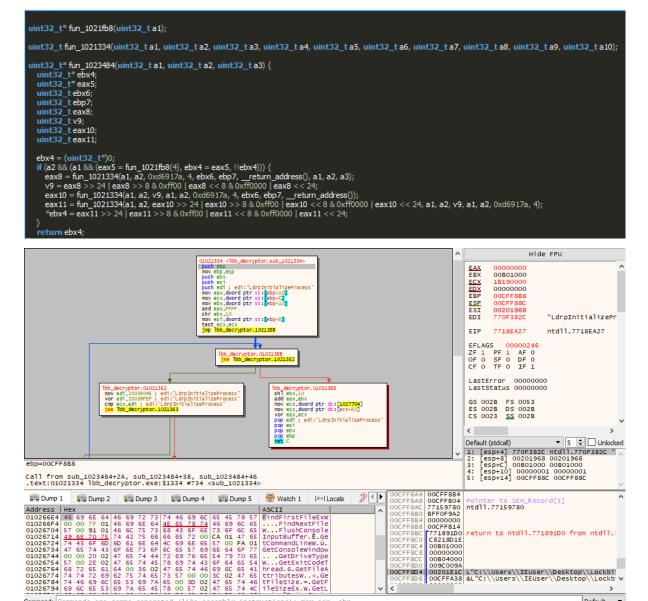
Following is the identification of the same execution context in debugging environment:





By decompiling in native C++ is possible to have the evidence of bit shifting performed on the attributes of the files taken in input:





۲.

Default 🔻

Command: Commands comma separated (like assembly instructions): mov eax, ebx are

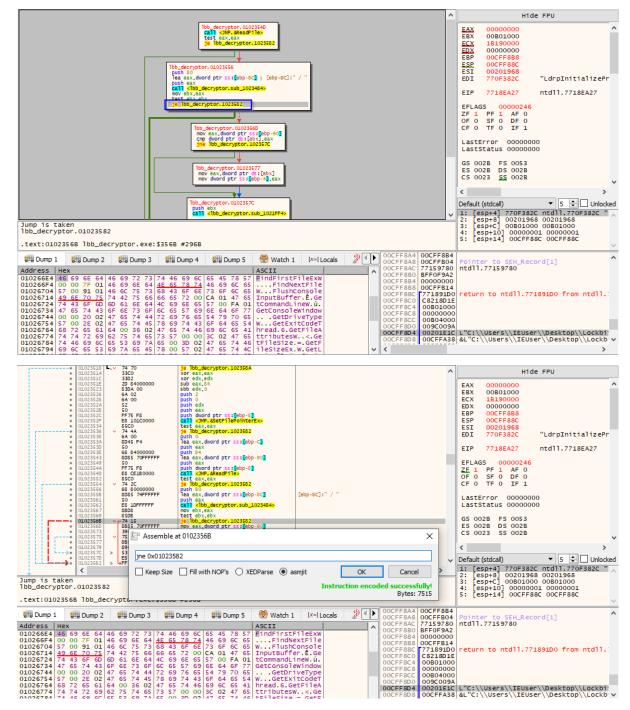
74



```
struct s0 {
    int8_t[128] pad128;
    uint32_t f128;
struct s0* g1027704 = (struct s0*)0;
uint32_t fun_1021334(uint8_t* a1, uint32_t a2, uint32_t a3) {
  uint32_t ecx4;
uint32_t eax5;
  uint32_t ebx6;
uint8_t* esi7;
  int1_t zf8;
  uint32_t edi9;
  struct s0* ecx10;
  ecx4 = a2;
  eax5 = a3 & 0xffff;
  ebx6 = a3 >> 16;
  esi7 = a1;
  zf8 = ecx4 == 0;
  while (!zf8) {
     edi9 = 0xfa1;
     if (ecx4 < 0xfa1) {
        edi9 = ecx4;
     ecx4 = ecx4 - edi9;
       eax5 = eax5 + *esi7;
       ++esi7;
ebx6 = ebx6 + eax5;
       --edi9;
     } while (edi9);
ebx6 = ebx6 % 0x1000f;
eax5 = eax5 % 0x1000f;
     zf8 = ecx4 == 0;
   ecx10 = g1027704;
   return eax5 + (ebx6 << 16) ^ ecx10->f128;
```

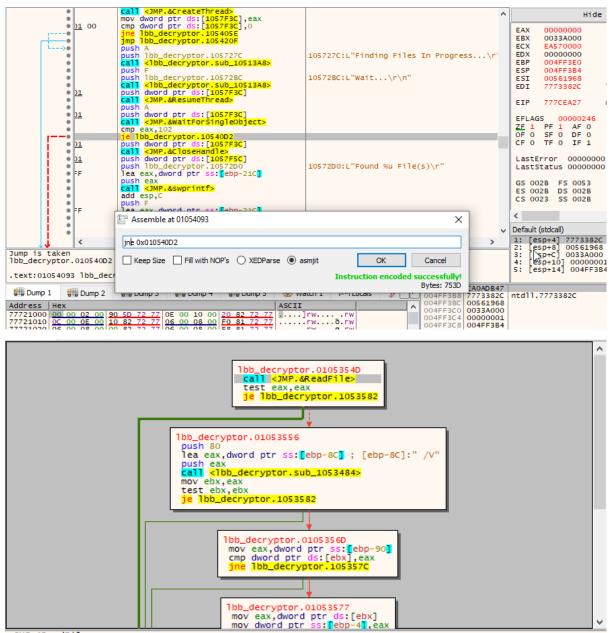
Here's an example of assembly modification related to the execution context in which is performed the call to XOR and bitswapping function.





Here are the details of an attempt to modify a jump instruction in the context of files gathering to **redirect** the execution to the *modification of attributes* of the encrypted files:

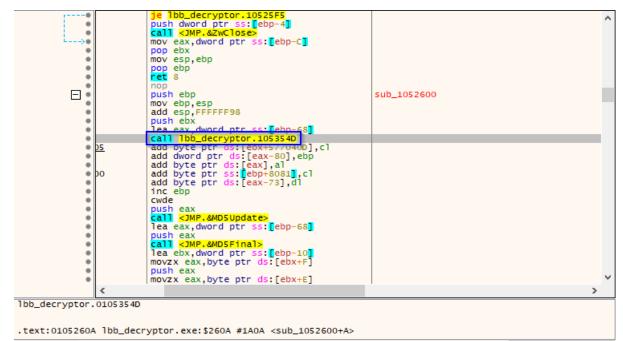




<JMP.&ReadFile>

.text:0105354D lbb_decryptor.exe:\$354D #294D

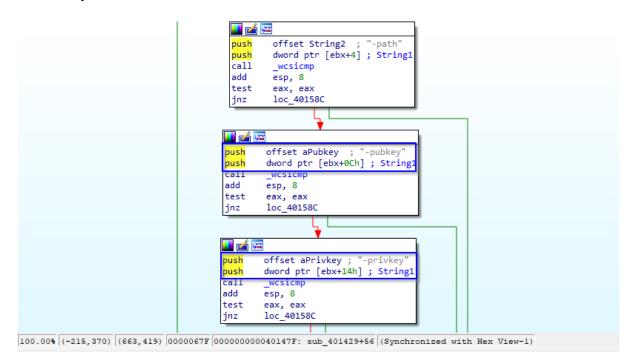




To generate the couple of public and private keys (where the public key is used by the ransomware and the private key by the decryptor) is called the executable **keygen.exe**, which creates the variables of the couple of keys respectively as follows:

Public key -> **ebx+14h**

Private key -> ebx+0Ch



Subsequently it is called the function sub_401000 to call the execution of the cryptography



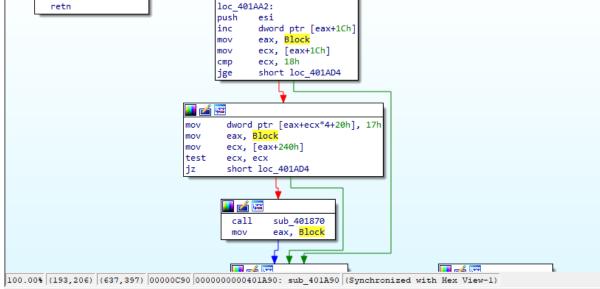
library and the randomic generation with MIRACL module. Furthermore, there are **rdrand** instructions for the generation of randomic numbers. The functions involved in the MIRACL execution are also **sub_4022C0** and **sub_401A90**.

bush	offset unk_409500
bush	offset unk 4093F0
all	sub 401000
lea	eax, [ebp+Buffer]
bush	eax ; 1pBuffer
bush	208h ; nBufferLength
all	GetCurrentDirectoryW
bush	dword ptr [ebx+8] ; lpPathName
all	SetCurrentDirectoryW
bush	offset Buffer
bush	100h
	offset unk_4093F0
all	sub_401274
bush	eax ; nNumberOfBytesToWrite
bush	offset Buffer ; 1pBuffer
bush	dword ptr [ebx+10h] ; lpFileName
all	sub_4013D0
bush	offset unk_409790
bush	100h
	offset unk_409500
all	sub_401274
bush	eax ; nNumberOfBytesToWrite
bush	offset unk_409790 ; lpBuffer
bush	dword ptr [ebx+18h] ; lpFileName
all	sub_4013D0

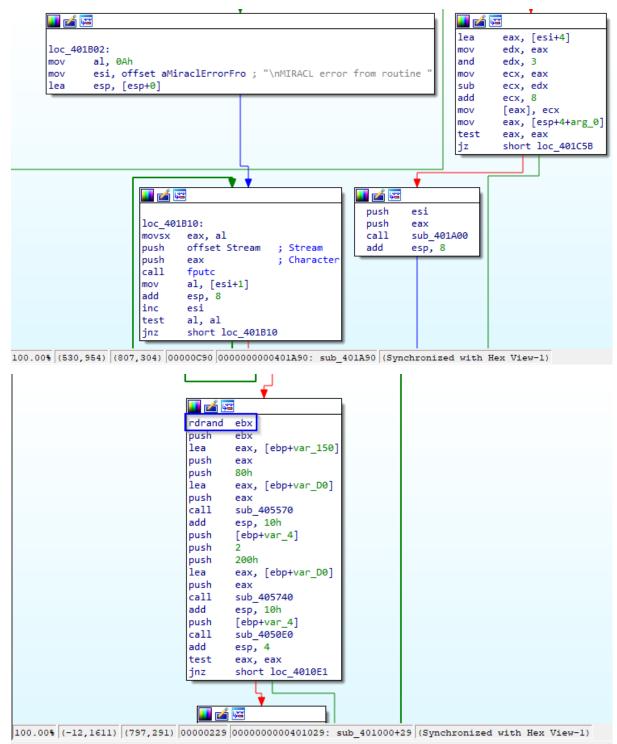
sub_ var_ var var Bloc var_ var_ arg_ arg_ push	
mov sub push push	esp, 150h n ebx n ecx
push push push	n esi n edi
push push call	400h



push	edi
push	10h
push	400h
call	sub 4022C0
add	esp, 8
mov	dword ptr [eax+234h], 10h
push	0
call	sub 401A90
add	esp, 4
mov	[ebp+Block], eax
push	0
call	sub_401A90
add	esp, 4
mov	[ebp+var_4], eax
push	0
call	sub_401A90
add	esp, 4
mov	[ebp+var_8], eax
push	10001h
call	sub_401A90
add	esp, 4
mov	[ebp+var_10], eax
push	0
call	sub_401A90
add	esp, 4
mov	[ebp+var_14], eax
push	0
call	sub_401A90
.00% (-123,877) (811,	285) 00000200 000000000401000: sub_401000 (Synchronized with Hex View-1)
•	
🗾 🚄 🖼	
xor eax, eax	
retn	loc_401AA2:
	push esi
	inc dword_ptr [eax+1Ch]





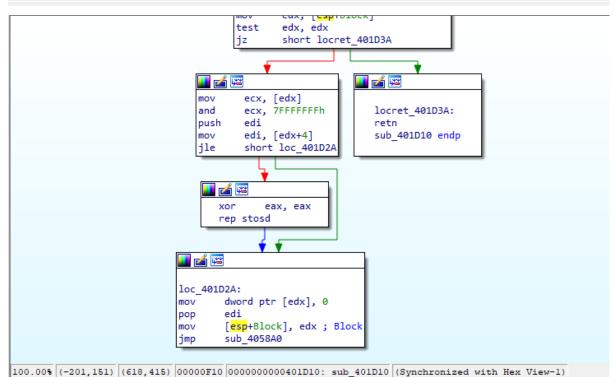


The function sub_401D10 is called recurrently and by having as attribute the variable associated to the public key. This function performs AND operations on the variable Block:



pusn call	eax memcpy	j	νοτα
add	esp, 0Ch		
	[ebp+Block]		Block
call	sub 401D10	,	DICCK
add	esp, 4		
push	[ebp+var 4]		Block
call	sub 401D10	,	DIOCK
add	esp, 4		
push	[ebp+var 8]		Block
call	sub 401D10		DIOCK
add	_		
push	esp, 4		Block
call	[ebp+var_10]	;	DIOCK
	sub_401D10		
add	esp, 4		
push	[ebp+var_14]	;	Block
call	sub_401D10		
add	esp, 4		
	[ebp+var_18]	;	Block
call	sub_401D10		
add	esp, 4		
call	sub_401D40		
pop	edi		
pop	esi		
pop	edx		
pop	ecx		
pop	ebx		
mov	esp, ebp		
non	ohn		

100.00% (-12,4026) (567,413) 00000229 00000000401029: sub_401000+29 (Synchronized with Hex View-1)



About the ELF file **decrypt_ESXI_Linux** is possible to verify that the given score by ELF Parser is **34**, so a quite "low" value which is an index that the file in question is heuristically known as little suspicious. There are evidences related to antidebugging (through ptrace) and some references that seem to be part of the phase of identification of the victim ID (also in this case there are details related to hashing functions).



🗑 ELF Parser		- 0
Score	Overview Filename Size	ELF Header SHeaders PHeaders Symbols Capabilities Scoring C:/Users/IEUser/Desktop/Lockbit3-Decryptor/decrypt_ESXI_Linux_64/decrypt_ESXI_Linux_64/decrypt_ESXI_Linux_64 237448
Controls	md5	6c210b23c3d98642f0b619bead52439c
Open Reset	sha1	a3d5667d676c10f515ffbb7149399a06c26ae92c
About Close	sha256	6dbe98b0a6f0a6063c85175a261a4fcffc7b8b4e8526a67fbccbb11976250f31
	Family	Undetermined

Following some details extractable from the Read Only of the file, in which is possible to identify references to **BLAKE2B_BLOCKBYTES**, **crypt_generichash_blake2b_final**, similarly to the Portable Executable version for Windows also in this case it is performed a call to hashing functionalities.



```
Read Only Segment (offset=0x28b40, size=43656, strings=37)
               String="""fD**~T"
               String=""fD"*~T*"
               String="%%oJ..r\"
               String="&jL&6Zl6?A~?"
               String="/dev/random"
               String="/dev/urandom"
               String="/proc/%d/cmdline"
               String="22Vd::Nt"
               String="2Vd2:Nt:"
               String=";d22Vt::N"
               String="=&&jL66Zl??A~"
               String="D""fT**~;"
               String="J%%o\..r8"
               String="L&&jl66Z~??A"
               String="LibsodiumDRGsodium/core.c"
               String="S->buflen <= BLAKE2B_BLOCKBYTES"
```

```
String="LibsodiumDRGsodium/core.c"
String="S->buflen <= BLAKE2B_BLOCKBYTES"
String="Vd22Nt::'
String="_sodium_malloc"
String="_unprotected_ptr_from_user_ptr(user_ptr) == unprotected_ptr"
String="blake2b_final"
String="crypto_generichash/blake2b/ref/blake2b-ref.c"
String="crypto_generichash/blake2b/ref/generichash_blake2b.c"
String="crypto_generichash_blake2b_final"
String="curve25519xsalsa20poly1305"
String="fD""~T**"
String="jL&&Zl66A~??"
String="locked == 0"
String="outlen <= (255)"
String="randombytes/sysrandom/randombytes_sysrandom.c"
String="safe_read"
String="size <= 9223372036854775807L"
```

Following the classification of various patterns identified in the ELF file, for example references

to system and sysinfo(), randomic generation functions, manipulation of the processes (for example **fork()**) and antidebugging:



Category		Details
~	Shell	
		system() found
~	Random Functions	-
		rand() found
		srand() found
~	Process Manipulation	
		daemon() found
		fork() found
		raise() found
×	Pipe Functions	
		pclose() found
		popen() found
~	Information Gathering	
		access() found
		sysinfo() found
~	File Functions	
		fclose() found
		unlink() found
×	Anti-Debug	
		ptrace detection found

Score	Reason
12	Process manipulation functions
2	Information gathering
10	Shell commands
10	Anti debug techniques

Here are the details of the magic number of the ELF analyzed, so **7f 45 4c 46** which identifies the typology of the ELF file:



Magic	7f 45 4c 46
Class	64-bit
Encoding	Little Endian
ELF Version	1
OS ABI	System V
ABI Version	0
Туре	ET_EXEC
Machine	x86_64
Version	1
Entry Point	0x4023e0
PH Offset	64
SH Offset	235720
Flags	0x0
Header Size	64
PH Entry Size	56
PH Entries	9
SH Entry Size	64
SH Entries	27
String I	ndex 26



Section Headers

Index	Name	Туре	Flags	Virtual Address	Offset	Size	Link
0		K_NULL		0x0	0	0	0
1	.interp	K_PROGBITS	Alloc	0x400238	568	28	0
2	.note.ABI-tag	K_NOTE	Alloc	0x400254	596	32	0
3	.note.gnu.build-id	K_NOTE	Alloc	0x400274	628	36	0
4	.gnu.hash	K_GNU_HASH	Alloc	0x400298	664	64	5
5	.dynsym	K_DYNSYM	Alloc	0x4002d8	728	2688	6
6	.dynstr	K_STRTAB	Alloc	0x400d58	3416	1078	0
7	.gnu.version	K_VERSYM	Alloc	0x40118e	4494	224	5
8	anu version r	K VERNEED	Alloc	0v401270	4720	128	6

Details

Interpreter (offset=0x238, size=28) Value="/lib64/ld-linux-x86-64.so.2"

6	.dynstr	K_STRTAB	Alloc	0x400d58	3416	1078	0
7	.gnu.version	K_VERSYM	Alloc	0x40118e	4494	224	5
8	anu version r	K VERNEED	Alloc	0x401270	4720	128	6

Details

```
String Table (offset=0xd58, size=1078, entries=108)

String=""

String="GLIBC_2.5"

String="GLIBC_2.3"

String="GLIBC_2.3.4"

String="GLIBC_2.4"

String="_Jv_RegisterClasses"

String="__assert_fail"

String="__ctype_tolower_loc"

String="__ctype_tolower_loc"

String="__fxstat"

String="__fxstat"

String="__fxstat64"

String="__fxstat64"

String="__libc_start_main"

String="__istack_chk_fail"
```

Physiological, instead, is the presence of the call to a **fgets** function, so the ability to read the streams. In this specific case the possibility to read data and bytes to submit to the decryption phase:



String="xpg_basename"
String="xstat64"
String="abort"
String="access"
String="calloc"
String="chdir"
String="daemon"
String="dirname"
String="exit"
String="fclose"
String="fcntl"
String="fgets"
String="flock"
String="fopen64"
String="fork"
String="fread"
String="ftruncate64"
 String="getopt_long"

Also in this case is possible to have the evidence of the creation of concurrential executions

through threads (for example the functionality **pthread_cond_wait** can be used to put a

String="mmap64"
String="mprotect"
String="munlock"
String="munmap"
String="nanosleep"
String="optarg"
String="opterr"
String="optind"
String="optopt"
String="pclose"
String="poll"
String="popen"
String="pthread_cond_broadcast"
String="pthread_cond_destroy"
String="pthread_cond_init"
String="pthread_cond_signal"
String="pthread_cond_wait"
String="othread_create"

Here are some references to randomic generations, for example

randombytes_sysrandom_implementation



String="ptrace" String="raise" String="randombytes_sysrandom_implementation" String="readlink" String="rename" String="setsid" String="setvbuf" String="snprintf" String="sprintf" String="srand" String="stderr" String="stdout" String="strcasecmp" String="strcat" String="strchr" String="strcmp" String="strcpy" String="strftime"

String="strftime" String="strlen" String="strncmp" String="strncpy" String="strrchr" String="strstr" String="strtok" String="strtol" String="sysconf" String="sysinfo" String="system" String="size > (size_t) 0U" String="sodium/utils.c" String="sodium crit enter" String="sysrandom" String="x%oJ%.r\." String="xxoJ%%r\..\$8"

String Table (offset=0x397dd, size=229, entries=26) String="" String=".bss" String=".comment" String=".ctors" String=".data" String=".dtors" String=".dynamic" String=".dynstr" String=".dynsym" String=".eh_frame" String=".eh_frame_hdr" String=".fini" String=".gnu.hash" String=".gnu.version" String=".gnu.version_r" String=".got"



String=".init" String=".interp" String=".jcr" String=".note.ABI-tag" String=".note.gnu.build-id" String=".rela.dyn" String=".rela.plt" String=".rela.plt" String=".rodata" String=".shstrtab" String=".text"

The call to a **chdir** function could permit to refer to different folders containing the files to manage:



	Туре	Binding	Value
1	STT_NOTYPE	STB_LOCAL	0x0
2	STT_FUNC	STB_GLOBAL	daemon
3	STT_FUNC	STB_GLOBAL	mprotect
4	STT_FUNC	STB_GLOBAL	chdir
5	STT_FUNC	STB_GLOBAL	dirname
6	STT_FUNC	STB_GLOBAL	pthread_cond_destroy
7	STT_FUNC	STB_GLOBAL	memset
8	STT_FUNC	STB_GLOBAL	snprintf
9	STT_FUNC	STB_GLOBAL	setsid

	Туре	Binding	Value
22	STT_FUNC	STB_GLOBAL	malloc
23	STT_FUNC	STB_GLOBAL	libc_start_main
24	STT_FUNC	STB_GLOBAL	system
25	STT_FUNC	STB_GLOBAL	unlink
26	STT_FUNC	STB_GLOBAL	memcpy_chk
27	STT_FUNC	STB_GLOBAL	gmtime
28	STT_FUNC	STB_GLOBAL	sysconf
29	STT_FUNC	STB_GLOBAL	pthread_mutex_init
30	STT_FUNC	STB_GLOBAL	fgets
31	STT_FUNC	STB_GLOBAL	_fxstat64
32	STT_FUNC	STB_GLOBAL	vfprintf
33	STT_FUNC	STB_GLOBAL	strdup



73	STT_FUNC	STB_GLOBAL	srand
74	STT_FUNC	STB_GLOBAL	pthread_cond_wait
75	STT_FUNC	STB_GLOBAL	pthread_detach
76	STT_FUNC	STB_GLOBAL	ctype_tolower_loc
77	STT_FUNC	STB_GLOBAL	memcmp
78	STT_FUNC	STB_GLOBAL	calloc
79	STT_FUNC	STB_GLOBAL	munmap
80	STT_FUNC	STB_GLOBAL	fclose
81	STT_FUNC	STB_GLOBAL	strncpy
82	STT_FUNC	STB_GLOBAL	_xstat64
83	STT_FUNC	STB_GLOBAL	lseek64
84	STT_FUNC	STB_GLOBAL	access

Following are the details of the hexadecimal of the ELF file in question, where is possible to identify the references to cryptography and *Blake2B* hash generation.



	decryp	ot_ES	XI_L	inux_	64														
	ħ	ß		Ĩ	(M	P		ŝ										
	ffset 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00000 00001 00001 00001 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00011 00012 00021 00022 00022 00022	00 10 20 30 40 50 60 70 80 90 80 80 20 80 20 20 20 20 20 20 20 20 20 20 20 20 20	0 7F24006408388 33C11009406608208 66005280844508 640000052608 640000052600 640000052600 640000052600 6400000526000052600000000000000000000000	1 45 00 00 00 00 00 00 00 00 00 00 00 00 00	$\begin{array}{c} 2\\ 4C\\ 3E\\ 00\\ 00\\ 00\\ 40\\ 00\\ 00\\ 40\\ 00\\ 00\\ 40\\ 00\\ 0$	$\begin{array}{c} 3\\ 46\\ 00\\ 00\\ 00\\ 00\\ 00\\ 00\\ 00\\ 00\\ 00\\ 0$	4 02 01 00 40 00 00 00 00 00 00 00 00 00 00 00	5 01 000 000 000 000 000 000 000 000 000	$egin{array}{c} 6 \\ 01 \\ 00 \\ 00 \\ 38 \\ 00 \\ 00 \\ 00 \\ 00 \\ 00$	7 000 000 000 000 000 000 000 000 000 0	8 00009400000000000000000000000000000000	9 00 23 98 00 00 00 00 00 00 00 00 00 00 00 00 00	A 00 40 03 40 00 40 00 40 00 40 00 40 00 40 00 63 00 00 40 00 00 40 00 00 40 00 00 40 00 0	B 000 000 000 000 000 000 000 000 000 0	C 000 000 000 000 000 000 000 000 000 0	D 000 000 000 000 000 000 000 000 000 0	E 000 000 000 000 000 000 000 000 000 0	F 00 00 00 00 00 00 00 00 00 00 00 00 00	Ascii ■ELF 00 0
000000000000000000000000000000000000000		50 60 70 80 90																	



0002AEA0 0002AEA0 0002AEA0 0002AEC0 00	Offset	0	1	2	3	4	5	6	7	8	9	A	В	С	D	Е	F	Ascii
0002AFE0 73 68 57 62 67 66 69 62 73 67 64 69 75 60 24 52 47 73 67 64 11	0002AE90										00						00	
0002AEC0 00 46 69 75 6D 44 52 47 73 6F 64 1.1bsodiumDRGsod 0002AEF0 64 97 6D 2F 63 60 62 63 6B 65 ium <core.c.locke< td=""> 0002AEF0 72 6F 64 69 75 6D 2F 74 5F 65 62 74 65 62 74 65 62 74 65 62 74 65 62 74 65 62 74 65 62 74 65 62 74 65 62 74 65 62 77 74 72 67 74 72 73 77 74 72 75 77 74 72 74 75 77 77 72 77 77 72 77 77 72 20 75 77 77 77 73 72 61 64 67 60 74 65 74 65 74 65 67 67<!--</td--><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></core.c.locke<>																		
0002AEE0 69 75 6D 2F 63 6F 72 65 2E 63 00 6C 6F 63 6B 65 ium/core.c.locke 0002AEF0 73 6F 64 69 75 6D 75 65 74 65 6E 74 65 6E 74 65 6E 74 65 6E 74 65 72 67 74 65 6E 74 65 74 74 75 76 67 74 75 74 65 65 77 74 72 26 74 65 64 67 74 75 77 74 72 29 77 74 72 29 77 74 72 29 77 74 72 20 75 73 76 74 75 74 75 74 75 74 75 74 75 74 75 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>																		
0002AEE0 64 20 3D 2D 3D 0D <																		
0002AFF0 73 6F 64 69 75 63 72 69 74 65 65 62 74 65 65 74 65 65 74 65 65 74 65 62 73 22 65 74 65 62 73 25 75 74 69 67 72 26 74 65 74 65 75 75 75 73 74 75 75 73 74 75 73 74 75 73 74 75 73 74 75 73 74 75 73 74 75 74 75																		
0002AF00 72 00 73 6F 64 69 75 74 69 6C 73 2E 63 r.sodium/utils.c 0002AF10 63 74 65 65 72 67 74 65 77 74 72 25 66 72 67 74 72 74 72 75 65 72 67 74 72 75 65 72 67 74 72 74 72 74 72 74 72 74 75 65 75 75 65 74 75 64 67 74 72 74 75 65 75 75 66 74 65 64 67 74 72 74 74 72 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 74 75 75 74 74 75 72 74 65 73 77 73 73 73 73																		
0002AF10 00 <																		
0002AF20 63 74 65 64 5F 70 74 72 28 75 73 65 72 5F 70 74 72 29 cted_ptr_from_us 0002AF30 20 3 3D 20 75 6E 70 72 6F 74 65 63 74 65 64 5F =:unprotected_ 0002AF50 70 74 72 20 5F 73 6F 64 69 75 6D 5F 6D 61 6C 6C ptrsodium_mall 0002AF60 26 72 6F 72 61 6E 64 6F 6D 00 73 69 7A 65 20 3E 20 v/random.size.>. 0002AF70 76 2F 72 61 6E 64 6F 6D 00 73 69 7A 65 20 3E 20 v/random.size.>. 0002AF80 28 73 69 7A 65 5F 74 29 20 30 55 00 73 69 7A 65 (size_t).0U.size 0002AF70 76 2F 72 61 6E 64 6F 6D 00 2F 64 65 76 2F 75 72 v/random.size.>. 0002AF80 28 73 69 7A 65 5F 74 29 20 30 55 00 73 69 7A 65 (size_t).0U.size 0002AF80 34 37 37 35 38 30 37 4C 00 2F 64 65 76 2F 72 72 andomrandomby 0002AF80 64 6F 6D 02 79 74 65 73 5F 73 79 73 72 61 6E 64 6F 6D 22 63 00 73 61 66 65 5F 72 65 61 64 00 dom.c.safe_read. 0002AFF0 63 75 72 76 65 32 35 35 31 39 78 73 61 66 73 61 curve25519xsalsa curve25519xsalsa 0002B000 32 07 0 6F 64 6F 5D 2E 63 20 07 00 00 00 00 00 00 00 00 00 00 curve1519xsalsa curve1519xsalsa 0002B010 30 3E 27 79 70 74 6F 75 66 6E 65 72 69 64 88 61 curvpto_genericha curve1519xsalsa																		
0002AF40 20 3D 3D 20 75 6E 70 72 6F 74 65 63 74 65 64 5F 0002AF50 70 74 72 00 5F 73 6F 64 69 75 6D 5F 6D 61 6C cc ptrsodium_mall 0002AF70 76 2F 72 61 6E 64 6F 6D 00 73 69 7A 65 20 3E 20 33 34 37 75 77 77 77 77 77 77 77 73 72 61 62 77 77 73 72 61	0002AF20	63	74	65	64	5F	70	74	72	5F	66	72	6F	6D	5F	75	73	
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		65									65							er_ptr(user_ptr)
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																		
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		· · -																
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																		
$\begin{array}{c c c c c c c c c c c c c c c c c c c $		· · -																
0002AFA0 34 37 37 35 38 30 37 4C 00 2F 64 65 76 2F 75 72 andomrandomby 0002AFD0 61 6E 64 6F 6D 02 73 79 73 72 61 6E 64 6F 6D 279 14 65 73 79 73 72 61 6E 00 00 00 73 79 73 72 65 61 64 00 00mbytes_sysran 0002AFF0 63 75 72 76 65 32 35 31 39 78 73 61 66 67 61 00 <t< td=""><td></td><td>. – -</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>		. – -																
$\begin{array}{c} 0002AFB0 \\ 0002AFC0 \\ 074 \\ 65 \\ 73 \\ 2F \\ 73 \\ 2F \\ 73 \\ 77 \\ 73 \\ 72 \\ 65 \\ 73 \\ 77 \\ 73 \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 64 \\ 6F \\ 6D \\ 2F \\ 72 \\ 61 \\ 6E \\ 61 \\ 6E \\ 65 \\ 72 \\ 77 \\ 77 \\ 74 \\ 6F \\ 6F \\ 79 \\ 77 \\ 74 \\ 6F \\ 6F \\ 75 \\ 72 \\ 65 \\ 6E \\ 65 \\ 72 \\ 65 \\ 6E \\ 65 \\ 72 \\ 65 \\ 62 \\ 66 \\ 2F \\ 72 \\ 65 \\ 62 \\ 66 \\ 2F \\ 72 \\ 65 \\ 62 \\ 66 \\ 2F \\ 62 \\ 62 \\ 62 \\ 62 \\ 62$																		
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																		
$\begin{array}{c} 0002AFE0 \\ 0002AFE0 \\ 64 \\ 6F \\ 6D \\ 2E \\ 63 \\ 75 \\ 72 \\ 76 \\ 65 \\ 32 \\ 35 \\ 35 \\ 35 \\ 31 \\ 39 \\ 78 \\ 73 \\ 61 \\ 6C \\ 73 \\ 61 \\ 70 \\ 70 \\ 70 \\ 70 \\ 70 \\ 70 \\ 70 \\ 7$		74	65	73		73											61	
0002AFF0 63 75 72 76 65 32 35 31 39 78 73 61 6C 73 61 73 61 6C 73 61 73 63 73 73 73 70 74 6F 5F 67 65 66 2F 62 68 61 66 66 66 2F 62 62 72 65 66 2F 62 62 74 65 66 2F 62 62 64 41 42 42 42 42 42 42 42 42 42 42 42 42 42 42 44 42	0002AFD0	6E	64	6F	6D				65					73	72	61	6E	ndombytes_sysran
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																		
0002B010 00																		
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																		20poly1305
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																		····
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																		
0002B050 53 2D 3E 62 75 66 6C 65 6E 20 3C 3D 20 42 4C 41 S->buflen.<=.BLA																		
0002B070 00 00 00 00 00 00 00 00 00 00 00 00 00																		
0002B080 62 6C 61 6B 65 32 62 5F 66 69 6E 61 6C 00	0002B060	4B	45	32	42	5F	42	4C	4F	43	4B	42	59	54	45	53	00	KE2B_BLOCKBYTES.
0002B090 00				00	00						00			00	00	00	00	
0002B0A0 08 C9 BC F3 67 E6 09 6A 3B A7 CA 84 85 AE 67 BB 0 ɼógæ.j;SÉI®g» 0002B0B0 2B F8 94 FE 72 F3 6E 3C F1 36 1D 5F 3A F5 4F A5 +elprón<ñ6_:ôO¥																		blake2b_final
0002B0B0 2B F8 94 FE 72 F3 6E 3C F1 36 1D 5F 3A F5 4F A5 +alpron<ñ6 :ôO¥																		
0002B0C0 D1 82 E6 AD 7F 52 0E 51 1F 6C 3E 2B 8C 68 05 9B N∎æ-∎R0Q 1>+∎b0∎ 0002B0D0 6B BD 41 FB AB D9 83 1F 79 21 7E 13 19 CD E0 5B k%Aû«Ŭ∎ y!^o00 1à[0002B0F0 00																		ULMOGæ.j;SLII®g»>
0002B0D0 6B BD 41 FB AB D9 83 1F 79 21 7E 13 19 CD E0 5B k*Aû«Ůľ y!^00 Îà[0002B0E0 80 00																		₩ m = 150 C l l + 160
0002B0E0 80 00 <																		
0002B0F0 00 <																		
0002B100 <																		••••••
0002B120 00 00 00 00 00 00 00 00 00 00 00 00 0					ŌŌ												00	
UUU2B130 UU OO																		
	0002B130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	



CONCLUSIONS:

The decryption tools analyzed perform a dynamic recognition of the cryptography context used on the compromised machine, in which the files were encrypted. It is present also an environment information gathering by the tool, in particular also of the user's settings NTUSER.dat. Some "cleaning" evidences related to some artifacts caused by the LockBit 3.0 infections, for example the restore of the Wallpaper and the icons, but also the remanipulation of some registry keys and some security identifiers. It is possible also to note that similarly to what happens with the encryption phase, one of the points that majorly characterizes the tool is the efficiency: specific threads are created for the execution of the decryption subroutine and to "point" to files obtained with an enumeration loop that will have to decrypt, by using the more efficient function SetFilePointerEx, instead of SetFilePointer.

Two fundamental evidences are securely related to the executions of bitswapping the files read in input by calling recurrently the function sub_401334, which includes also the XOR operations execution. The second evidence is instead related to the comparison between the value of eax register containing the MD5 hashing digest with hardcoded values in the decryptor with also the adding of AND and OR operations.

From the analysis of keygen.exe was possible to understand how, at the moment of the contextual creation of the ransomware and the decryptor, two keys are created: priv.key and pub.key.

The public key is used in the ransomware to encrypt the files of the victim. The private key, encoded in the decryptor, is used to decrypt the encrypted files only with the ransomware "linked" to the decryptor.

The decryptors are not "universal" but they are strictly related to the couple of public and private keys generated by keygen.exe



Technical Contributors:

Fabio Pensa SoC Team Swascan

Contact Info

Milano +39 0278620700 www.swascan.com info@swascan.com Via Fabio Filzi, 2b, 20063, Cernusco sul Naviglio, MI