Practical4 Report

Name: Rachana Nitinkumar Gugale
UFID: 6353-2454
Email: rgugale@ufl.edu
Date: 26th April 2023
Class: CAP6137
Assignment: Practical 4

Executive Summary

The sample is a **ransomware** and a **keylogger**. It is also a **trojan** as it is distributed as a .bin file but it is actually an executable and if run by the user, it wreaks havoc on their system.

Once the sample is run, it copies itself to **C:\Users\<username>\AppData\Local** and starts numerous other subprocesses in admin mode with the newly copied executable using the Windows API function **ShellExecuteExW**. Before the malware starts a process in admin mode, a privilege escalation user prompt appears on screen. The sample proceeds to deletes the executable file from the original location from where the program was started. Some of the malware's subprocesses take command line arguments like "—Admin", "—Service" and the user's personal ID.

The malware's subprocesses enumerate over all folders and encrypt all files on the system regardless of their filetype. Files are encrypted using **AES** (Advanced Encryption Standard) algorithm. Encrypted files get a .KEYPASS extension. Once the sample finishes execution, it terminates all of its processes and deletes its executable from the machine. The malware drops a file called **!!!DECRYPTION_KEYPASS_INFO!!!.txt** in every folder. This file contains a ransom note and has the attacker's email and the user's personal ID. The program doesn't encrypt files present in *C:\Windows* folder.

The user keeps getting a "Files waiting to be burned to disk" notification periodically as the malware modifies the contents of Burn\Temporary Burn Folder by dropping a ransom note in it. The malware mounts a D: drive.

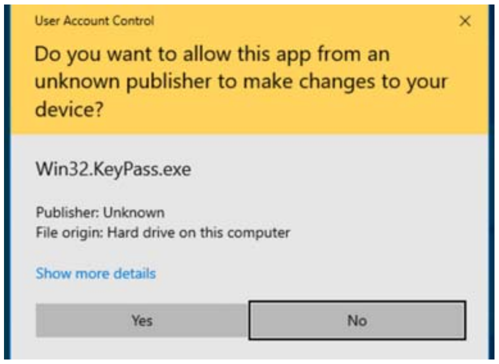
The sample has a hidden GUI. The hidden window contains the ransom note text along with the list of folders the malware skips while encrypting files. The malware installs a *WH_KEYBOARD_LL* hook using the **SetWindowsHookExW** function to capture the user's keystrokes possibly to check if the hotkey for the hidden window was pressed.

The malware is *not persistent*. It deletes itself from the system and any files created after the malware has finished executing don't get encrypted. Some of the subprocesses the malware creates employ anti-debugging using **IsDebuggerPresent** Windows API. The sample uses **ASLR** (Address Space Layout Randomization) which makes it difficult to analyze.

The sample does the following network activity which can be used as a network IoC:

- DNS request for **kronus.pp.ua**
- HTTP GET request to **http://kronus.pp.ua/upwinload/get.php** on port 80

- The presence of !!!DECRYPTION_KEYPASS_INFO!!!.txt or any file with .KEYPASS extension can be a host-based IoC. The presence of HKU\<SID>\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.KEYPASS registry key can also be a host-based IoC. Being unable to open programs installed outside C:\Windows is another indicator.


User Account Control

Do you want to allow this app from an unknown publisher to make changes to your device?

Win32.KeyPass.exe

Publisher: Unknown
File origin: Hard drive on this computer

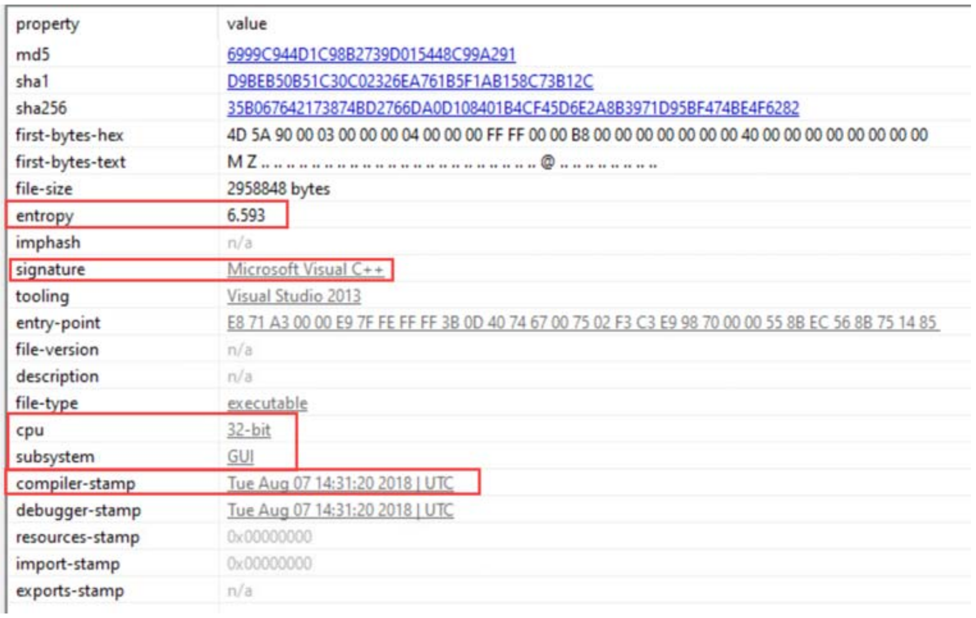
Figure 1 Privilege escalation prompt appears when malware starts its subprocess in admin mode

Static Analysis

PEStudio

I got the following information from PEStudio:

1. **Compilation Date:** Aug 7, 2018 at 14:31:20 UTC.
2. The malware is a 32-bit **Windows GUI** program as can be seen from the "subsystem" value.



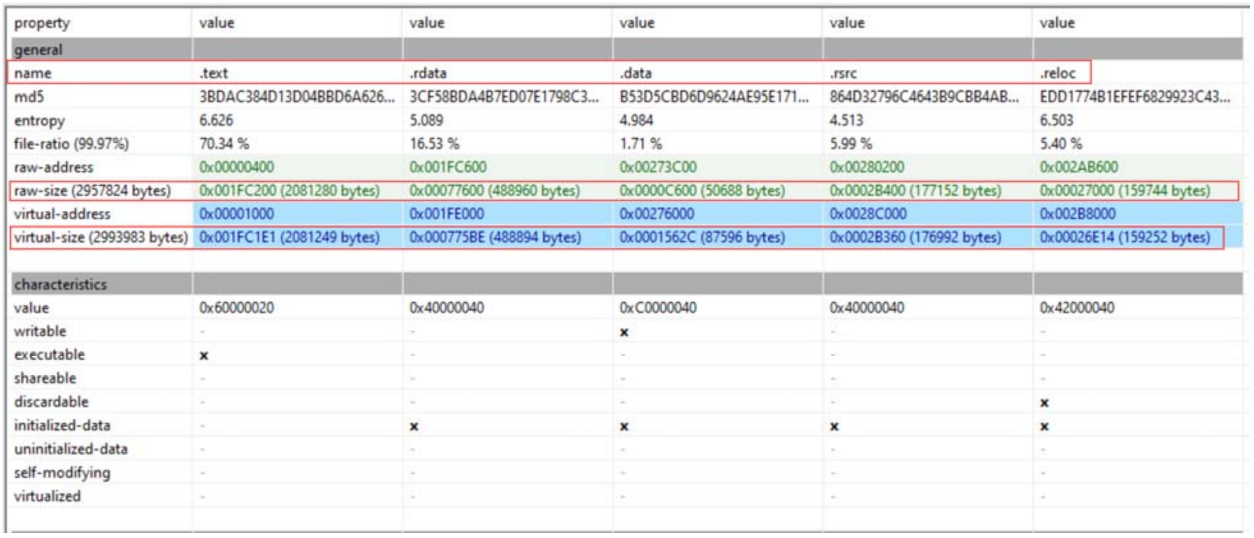
property	value
md5	6999C944D1C98B2739D015448C99A291
sha1	D9BEB50B51C30C02326EA761B5F1AB158C73B12C
sha256	35B067642173874BD2766DA0D108401B4CF45D6E2A8B3971D95BF474BE4F6282
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	2958848 bytes
entropy	6.593
imphash	n/a
signature	Microsoft Visual C++
tooling	Visual Studio 2013
entry-point	E8 71 A3 00 00 E9 7F FE FF FF 3B 0D 40 74 67 00 75 02 F3 C3 E9 98 70 00 00 55 8B EC 56 8B 75 14 85
file-version	n/a
description	n/a
file-type	executable
cpu	32-bit
subsystem	GUI
compiler-stamp	Tue Aug 07 14:31:20 2018 | UTC
debugger-stamp	Tue Aug 07 14:31:20 2018 | UTC
resources-stamp	0x00000000
import-stamp	0x00000000
exports-stamp	n/a

Figure 2 Basic PEStudio Analysis

3. The program is **not packed**. The following indicators were used to reach this conclusion:
    a. **Entropy:** The diversity of a program or file can be quantified by its entropy. A scale of 0 to 8 is used to express entropy values, with 0 indicating no entropy (i.e., all bytes in the file are identical), and 8 indicating maximum entropy (i.e., all bytes in the file are different). Generally, executable programs have an entropy level of around 6-6.5. The entropy of our sample is calculated to be 6.593, which suggests that the malware is not compressed.
    b. **Signature:** The “signature” field in PEStudio is “Microsoft Visual C++” for this sample. Packed samples have the name of the packer or some obfuscated text as signature. The absence of such a string in the signature hints that the malware is unpacked.
    c. **Names of PE Sections:** The program has PE sections named .text, .rdata, .data, .rsrc and .reloc. All these are standard names commonly found in most PE files. The lack of obfuscation in the section names could indicate that the malware is not packed.
    d. **Raw and Virtual Sizes of PE Sections:** When a section's raw size is significantly smaller than its virtual size, it suggests that the malware could be unpacking and extracting new data. In the current sample, the raw and virtual sizes of .text, .rdata, .rsrc, and .reloc sections are quite similar. The .data section has a larger virtual size, but this is typical for Windows executables and does not necessarily indicate packing. These observations provide further evidence that the malware is not packed.

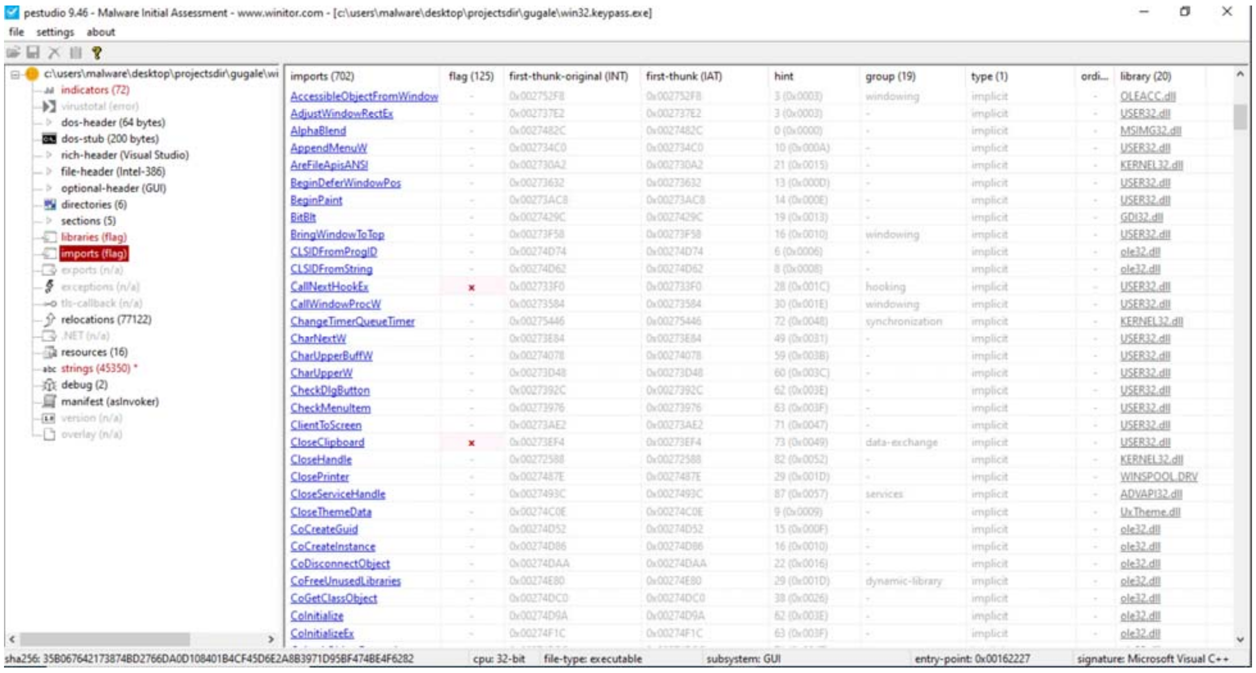
e. **Strings:** Static analysis with PE Studio reveals a large number of strings. Such a huge number of non-obfuscated strings indicate that the sample isn't packed.

f. **Imports:** The program imports many functions from multiple libraries. The presence of such a large import list indicates that the sample isn't packed.



property	value	value	value	value	value
general					
name	.text	.rdata	.data	.rsrc	.reloc
md5	3BDAC384D13D04BBD6A626...	3CF58BDA4B7ED07E1798C3...	B53D5CBD6D9624AE95E171...	864D32796C4643B9CBB4AB...	EDD1774B1EFEF6829923C43...
entropy	6.626	5.089	4.984	4.513	6.503
file-ratio (99.97%)	70.34 %	16.53 %	1.71 %	5.99 %	5.40 %
raw-address	0x00000400	0x001FC600	0x00273C00	0x00280200	0x002AB600
raw-size (2957824 bytes)	0x001FC200 (2081280 bytes)	0x00077600 (488960 bytes)	0x0000C600 (50688 bytes)	0x0002B400 (177152 bytes)	0x00027000 (159744 bytes)
virtual-address	0x00001000	0x001FE000	0x00276000	0x0028C000	0x002B8000
virtual-size (2993983 bytes)	0x001FC1E1 (2081249 bytes)	0x000775BE (488894 bytes)	0x0001562C (87596 bytes)	0x0002B360 (176992 bytes)	0x00026E14 (159252 bytes)
characteristics					
value	0x60000020	0x40000040	0xC0000040	0x40000040	0x42000040
writable	-	-	x	-	-
executable	x	-	-	-	-
shareable	-	-	-	-	-
discardable	-	-	-	-	x
initialized-data	-	x	x	x	x
uninitialized-data	-	-	-	-	-
self-modifying	-	-	-	-	-
virtualized	-	-	-	-	-

Figure 3 Standard PE Section names. Raw and Virtual Sizes are normal too



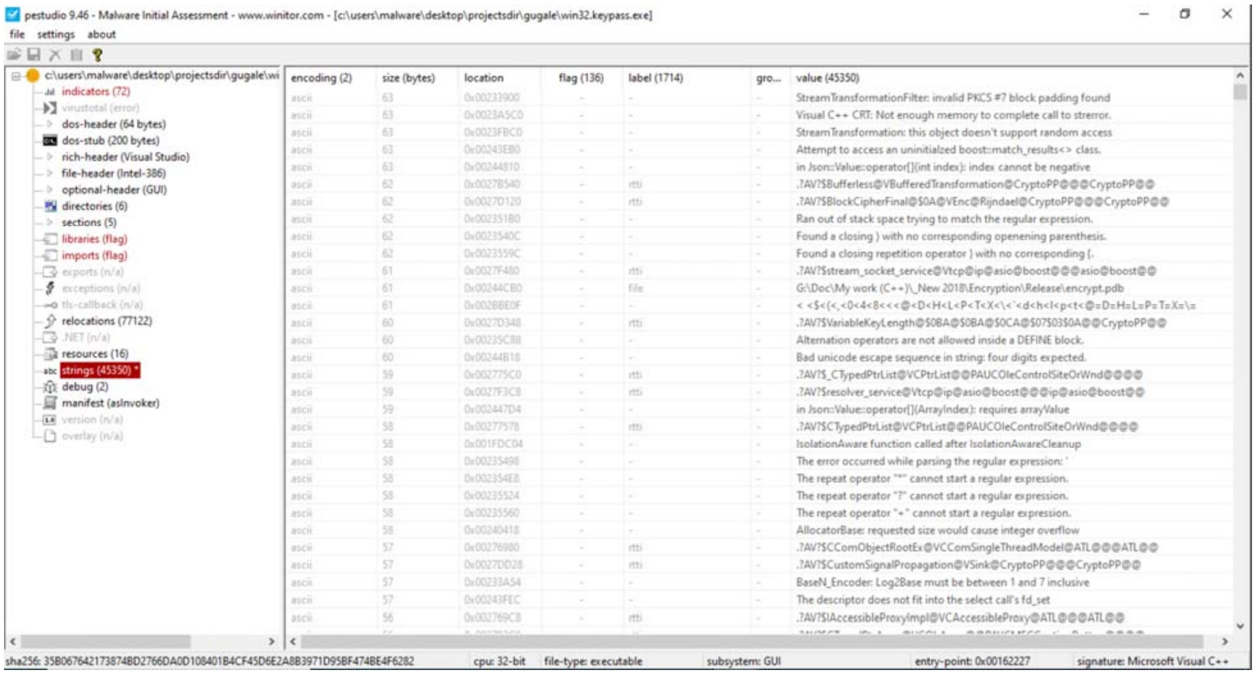
pestudio 9.46 - Malware Initial Assessment - www.winitor.com - [c:\users\malware\desktop\projectsdir\gugale\win32.keypass.exe]

- c:\users\malware\desktop\projectsdir\gugale\wi
  - indicators (72)
  - virustotal (error)
  - dos-header (64 bytes)
  - dos-stub (200 bytes)
  - rich-header (Visual Studio)
  - file-header (Intel-386)
  - optional-header (GUI)
  - directories (6)
  - sections (5)
  - libraries (flag)
  - imports (flag)
  - exports (n/a)
  - exceptions (n/a)
  - tls-callback (n/a)
  - relocations (77122)
  - .NET (n/a)
  - resources (16)
  - strings (45350) *
  - debug (2)
  - manifest (asInvoker)
  - version (n/a)
  - overlay (n/a)

imports (702)	flag (125)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (19)	type (1)	ordi...	library (20)
AccessibleObjectFromWindow	-	0x002752F8	0x002752F8	3 (0x0003)	windowing	implicit	-	OLEACC.dll
AdjustWindowRectEx	-	0x002737E2	0x002737E2	3 (0x0003)	-	implicit	-	USER32.dll
AlphaBlend	-	0x0027482C	0x0027482C	0 (0x0000)	-	implicit	-	MSIMG32.dll
AppendMenuW	-	0x002734C0	0x002734C0	10 (0x000A)	-	implicit	-	USER32.dll
AreFileApisANSI	-	0x002730A2	0x002730A2	21 (0x0015)	-	implicit	-	KERNEL32.dll
BeginDeferWindowPos	-	0x00273632	0x00273632	13 (0x000D)	-	implicit	-	USER32.dll
BeginPaint	-	0x00273AC8	0x00273AC8	14 (0x000E)	-	implicit	-	USER32.dll
BitBlt	-	0x0027429C	0x0027429C	19 (0x0013)	-	implicit	-	GDI32.dll
BringWindowToTop	-	0x00273F58	0x00273F58	16 (0x0010)	windowing	implicit	-	USER32.dll
CLSIDFromProgID	-	0x00274D74	0x00274D74	6 (0x0006)	-	implicit	-	ole32.dll
CLSIDFromString	-	0x00274D62	0x00274D62	8 (0x0008)	-	implicit	-	ole32.dll
CallNextHookEx	x	0x002733F0	0x002733F0	28 (0x001C)	hooking	implicit	-	USER32.dll
CallWindowProcW	-	0x00273584	0x00273584	30 (0x001E)	windowing	implicit	-	USER32.dll
ChangeTimerQueueTimer	-	0x00275446	0x00275446	72 (0x0048)	synchronization	implicit	-	KERNEL32.dll
CharNextW	-	0x00273E84	0x00273E84	49 (0x0031)	-	implicit	-	USER32.dll
CharUpperBuffW	-	0x00274078	0x00274078	59 (0x003B)	-	implicit	-	USER32.dll
CharUpperW	-	0x00273D48	0x00273D48	60 (0x003C)	-	implicit	-	USER32.dll
CheckDlgButton	-	0x0027392C	0x0027392C	62 (0x003E)	-	implicit	-	USER32.dll
CheckMenuItem	-	0x00273976	0x00273976	63 (0x003F)	-	implicit	-	USER32.dll
ClientToScreen	-	0x00273AE2	0x00273AE2	71 (0x0047)	-	implicit	-	USER32.dll
CloseClipboard	x	0x00273EF4	0x00273EF4	73 (0x0049)	data-exchange	implicit	-	USER32.dll
CloseHandle	-	0x00272588	0x00272588	82 (0x0052)	-	implicit	-	KERNEL32.dll
ClosePrinter	-	0x0027487E	0x0027487E	29 (0x001D)	-	implicit	-	WINSPOOL.DRV
CloseServiceHandle	-	0x0027493C	0x0027493C	87 (0x0057)	services	implicit	-	ADVAPI32.dll
CloseThemeData	-	0x00274C0E	0x00274C0E	9 (0x0009)	-	implicit	-	UxTheme.dll
CoCreateGuid	-	0x00274D52	0x00274D52	15 (0x000F)	-	implicit	-	ole32.dll
CoCreateInstance	-	0x00274D86	0x00274D86	16 (0x0010)	-	implicit	-	ole32.dll
CoDisconnectObject	-	0x00274DAA	0x00274DAA	22 (0x0016)	-	implicit	-	ole32.dll
CoFreeUnusedLibraries	-	0x00274E80	0x00274E80	29 (0x001D)	dynamic-library	implicit	-	ole32.dll
CoGetClassObject	-	0x00274DC0	0x00274DC0	38 (0x0026)	-	implicit	-	ole32.dll
CoInitialize	-	0x00274D9A	0x00274D9A	62 (0x003E)	-	implicit	-	ole32.dll
CoInitializeEx	-	0x00274F1C	0x00274F1C	63 (0x003F)	-	implicit	-	ole32.dll

sha256: 35B067642173874BD2766DA0D108401B4CF45D6E2A8B3971D95BF474BE4F6282 | cpu: 32-bit | file-type: executable | subsystem: GUI | entry-point: 0x00162227 | signature: Microsoft Visual C++

Figure 4 The sample has a lot of imports



pestudio 9.46 - Malware Initial Assessment - www.winitor.com - [c:\users\malware\desktop\projectsdir\gugale\win32.keypass.exe]

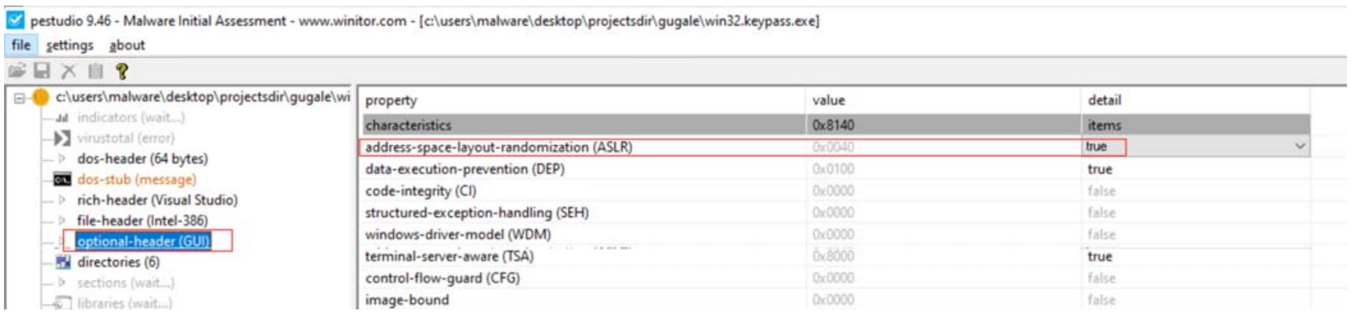
```
c:\users\malware\desktop\projectsdir\gugale\wi
  indicators (72)
  virustotal (error)
  dos-header (64 bytes)
  dos-stub (200 bytes)
  rich-header (Visual Studio)
  file-header (Intel-386)
  optional-header (GUI)
  directories (6)
  sections (5)
  libraries (flag)
  imports (flag)
  exports (n/a)
  exceptions (n/a)
  tls-callback (n/a)
  relocations (77122)
  .NET (n/a)
  resources (16)
  strings (45350) *
  debug (2)
  manifest (asInvoker)
  version (n/a)
  overlay (n/a)
```

encoding (2)	size (bytes)	location	flag (136)	label (1714)	gro...	value (45350)
ascii	63	0x00233900	-	-	-	StreamTransformationFilter: invalid PKCS #7 block padding found
ascii	63	0x0023A5C0	-	-	-	Visual C++ CRT: Not enough memory to complete call to strerror.
ascii	63	0x0023FBC0	-	-	-	StreamTransformation: this object doesn't support random access
ascii	63	0x00243EB0	-	-	-	Attempt to access an uninitialzed boost::match_results<> class.
ascii	63	0x00244810	-	-	-	in Json::Value::operator[](int index): index cannot be negative
ascii	62	0x0027B540	-	rtti	-	.?AV?$Bufferless@VBufferedTransformation@CryptoPP@@@CryptoPP@@
ascii	62	0x0027D120	-	rtti	-	.?AV?$BlockCipherFinal@$0A@VEnc@Rijndael@CryptoPP@@@CryptoPP@@
ascii	62	0x002351B0	-	-	-	Ran out of stack space trying to match the regular expression.
ascii	62	0x0023540C	-	-	-	Found a closing) with no corresponding openening parenthesis.
ascii	62	0x0023559C	-	-	-	Found a closing repetition operator } with no corresponding {.
ascii	61	0x0027F480	-	rtti	-	.?AV?$stream_socket_service@Vtcp@ip@asio@boost@@@asio@boost@@
ascii	61	0x00244CB0	-	file	-	G:\Doc\My work (C++)_New 2018\Encryption\Release\encrypt.pdb
ascii	61	0x002BBE0F	-	-	-	< <$<(<,<0<4<8<<<@<D<H<L<P<T<X<\<`<d<h<l<p<t<@=D=H=L=P=T=X=\=
ascii	60	0x0027D348	-	rtti	-	.?AV?$VariableKeyLength@$0BA@$0BA@$0CA@$07$03$0A@@CryptoPP@@
ascii	60	0x00235C88	-	-	-	Alternation operators are not allowed inside a DEFINE block.
ascii	60	0x00244B18	-	-	-	Bad unicode escape sequence in string: four digits expected.
ascii	59	0x002775C0	-	rtti	-	.?AV?$_CTypedPtrList@VCPtrList@@PAUCOleControlSiteOrWnd@@@@
ascii	59	0x0027F3C8	-	rtti	-	.?AV?$resolver_service@Vtcp@ip@asio@boost@@@ip@asio@boost@@
ascii	59	0x002447D4	-	-	-	in Json::Value::operator[](ArrayIndex): requires arrayValue
ascii	58	0x00277578	-	rtti	-	.?AV?$CTypedPtrList@VCPtrList@@PAUCOleControlSiteOrWnd@@@@
ascii	58	0x001FDC04	-	-	-	IsolationAware function called after IsolationAwareCleanup
ascii	58	0x00235498	-	-	-	The error occurred while parsing the regular expression: '
ascii	58	0x002354E8	-	-	-	The repeat operator "*" cannot start a regular expression.
ascii	58	0x00235524	-	-	-	The repeat operator "?" cannot start a regular expression.
ascii	58	0x00235560	-	-	-	The repeat operator "+" cannot start a regular expression.
ascii	58	0x00240418	-	-	-	AllocatorBase: requested size would cause integer overflow
ascii	57	0x00276980	-	rtti	-	.?AV?$CComObjectRootEx@VCComSingleThreadModel@ATL@@@ATL@@
ascii	57	0x0027DD28	-	rtti	-	.?AV?$CustomSignalPropagation@VSink@CryptoPP@@@CryptoPP@@
ascii	57	0x00233A54	-	-	-	BaseN_Encoder: Log2Base must be between 1 and 7 inclusive
ascii	57	0x00243FEC	-	-	-	The descriptor does not fit into the select call's fd_set
ascii	56	0x002769C8	-	rtti	-	.?AV?$IAccessibleProxyImpl@VCAccessibleProxy@ATL@@@ATL@@

sha256: 35B067642173874BD2766DA0D108401B4CF45D6E2A8B3971D95BF474BE4F6282 cpu: 32-bit file-type: executable subsystem: GUI entry-point: 0x00162227 signature: Microsoft Visual C++

Figure 5 Many readable strings found by PEStudio

4. **Address space layout randomization (ASLR):** PEStudio shows that the ASLR byte in *optional-header* of the PE file is set to true. This means the executable loads at a different address each time and makes analysis much more difficult.



pestudio 9.46 - Malware Initial Assessment - www.winitor.com - [c:\users\malware\desktop\projectsdir\gugale\win32.keypass.exe]

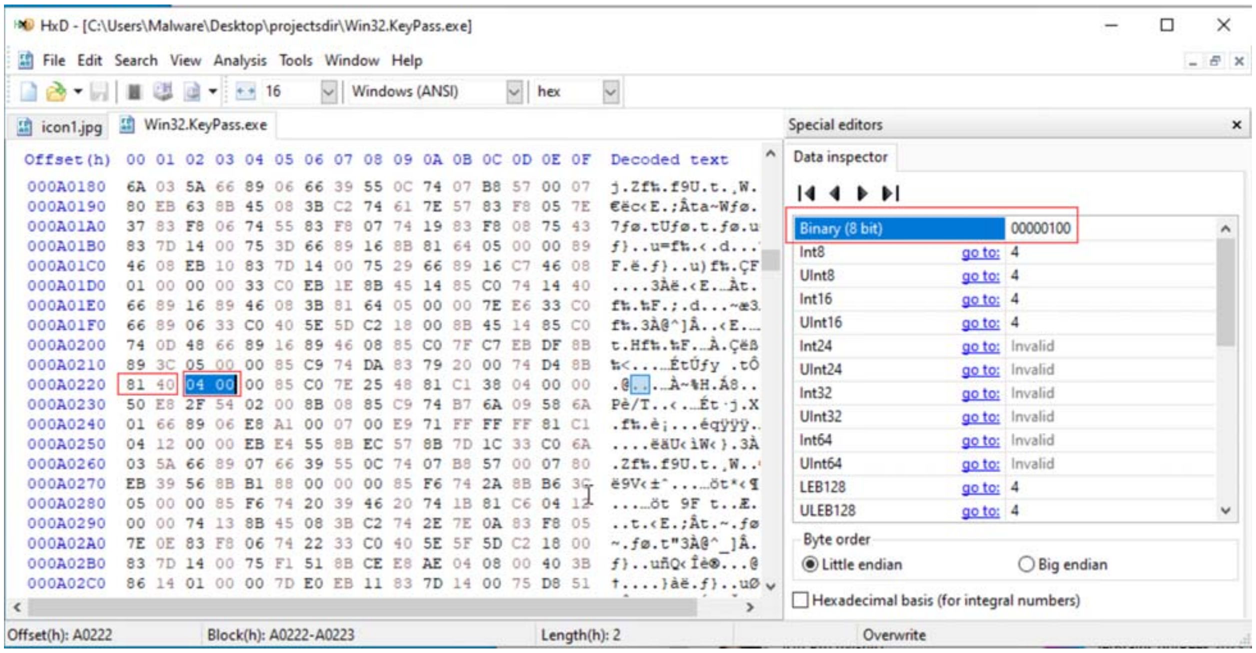
```
c:\users\malware\desktop\projectsdir\gugale\wi
  indicators (wait...)
  virustotal (error)
  dos-header (64 bytes)
  dos-stub (message)
  rich-header (Visual Studio)
  file-header (Intel-386)
  optional-header (GUI)
  directories (6)
  sections (wait...)
  libraries (wait...)
```

property	value	detail
characteristics	0x8140	items
address-space-layout-randomization (ASLR)	0x0040	true
data-execution-prevention (DEP)	0x0100	true
code-integrity (CI)	0x0000	false
structured-exception-handling (SEH)	0x0000	false
windows-driver-model (WDM)	0x0000	false
terminal-server-aware (TSA)	0x8000	true
control-flow-guard (CFG)	0x0000	false
image-bound	0x0000	false

Figure 6 Executable uses ASLR

I tried turning off ASLR by patching ASLR byte using HxD and setting it to 0 but it didn't turn ASLR off. Probably because the ASLR mechanism also uses other values in the PE file to determine whether to enable it or not.

Since I couldn't turn ASLR off, I rebased the file in Ghidra with the x32dbg address each time I was analyzing the sample.


HxD - [C:\Users\Malware\Desktop\projectsdir\Win32.KeyPass.exe]
File Edit Search View Analysis Tools Window Help
16
Windows (ANSI)
hex
icon1.jpg
Win32.KeyPass.exe
Special editors
Data inspector

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
000A0180  6A 03 5A 66 89 06 66 39 55 0C 74 07 B8 57 00 07  j.Zf‰.f9U.t.¸W..
000A0190  80 EB 63 8B 45 08 3B C2 74 61 7E 57 83 F8 05 7E  €ëc‹E.;Âta~Wfø.~
000A01A0  37 83 F8 06 74 55 83 F8 07 74 19 83 F8 08 75 43  7fø.tUfø.t.fø.uC
000A01B0  83 7D 14 00 75 3D 66 89 16 8B 81 64 05 00 00 89  f}..u=f‰.‹.d...‰
000A01C0  46 08 EB 10 83 7D 14 00 75 29 66 89 16 C7 46 08  F.ë.f}..u)f‰.ÇF.
000A01D0  01 00 00 00 33 C0 EB 1E 8B 45 14 85 C0 74 14 40  ....3Àë.‹E.…Àt.@
000A01E0  66 89 16 89 46 08 3B 81 64 05 00 00 7E E6 33 C0  f‰.‰F.;.d...~æ3À
000A01F0  66 89 06 33 C0 40 5E 5D C2 18 00 8B 45 14 85 C0  f‰.3À@^]Â..‹E.…À
000A0200  74 0D 48 66 89 16 89 46 08 85 C0 7F C7 EB DF 8B  t.Hf‰.‰F.…À.ÇëB‹
000A0210  89 3C 05 00 00 85 C9 74 DA 83 79 20 00 74 D4 8B  ‰<....…ÉtÚfy .tÔ‹
000A0220  81 40 04 00 00 85 C0 7E 25 48 81 C1 38 04 00 00  .@....…À~%H.Á8...
000A0230  50 E8 2F 54 02 00 8B 08 85 C9 74 B7 6A 09 58 6A  Pè/T..‹.…Ét·j.Xj
000A0240  01 66 89 06 E8 A1 00 07 00 E9 71 FF FF FF 81 C1  .f‰.è¡...éqÿÿÿ.Á
000A0250  04 12 00 00 EB E4 55 8B EC 57 8B 7D 1C 33 C0 6A  ....ëäU‹ìW‹}.3Àj
000A0260  03 5A 66 89 07 66 39 55 0C 74 07 B8 57 00 07 80  .Zf‰.f9U.t.¸W...€
000A0270  EB 39 56 8B B1 88 00 00 00 85 F6 74 2A 8B B6 3C  ë9V‹±ˆ....…öt*‹¶<
000A0280  05 00 00 85 F6 74 20 39 46 20 74 1B 81 C6 04 12  ...…öt 9F t..Æ..
000A0290  00 00 74 13 8B 45 08 3B C2 74 2E 7E 0A 83 F8 05  ..t.‹E.;Ât.~.fø.
000A02A0  7E 0E 83 F8 06 74 22 33 C0 40 5E 5F 5D C2 18 00  ~.fø.t"3À@^_]Â..
000A02B0  83 7D 14 00 75 F1 51 8B CE E8 AE 04 08 00 40 3B  f}..uñQ‹Îè®...@;
000A02C0  86 14 01 00 00 7D E0 EB 11 83 7D 14 00 75 D8 51  †....}àë.f}..uØQ
```

Binary (8 bit)		00000100
Int8	go to:	4
UInt8	go to:	4
Int16	go to:	4
UInt16	go to:	4
Int24	go to:	Invalid
UInt24	go to:	Invalid
Int32	go to:	Invalid
UInt32	go to:	Invalid
Int64	go to:	Invalid
UInt64	go to:	Invalid
LEB128	go to:	4
ULEB128	go to:	4

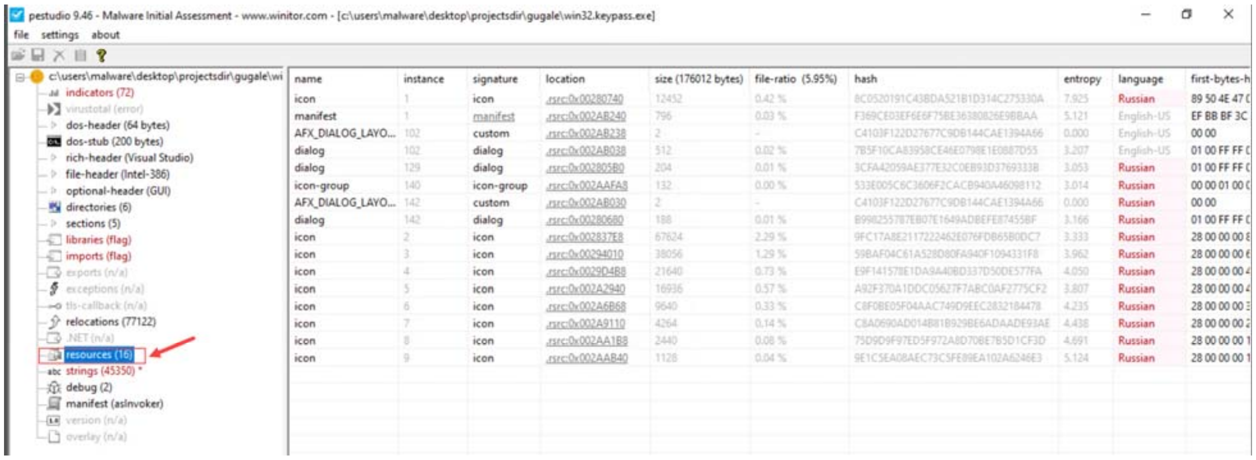
Byte order: [✓ Little endian] Big endian

☐ Hexadecimal basis (for integral numbers)

Offset(h): A0222 Block(h): A0222-A0223 Length(h): 2 Overwrite

Figure 7 Changing the value 40 to 00 is not sufficient to turn ASLR off

Resources



name	instance	signature	location	size (176012 bytes)	file-ratio (5.95%)	hash	entropy	language	first-bytes-h
icon	1	icon	.rsrc:0x00280740	12452	0.42 %	8C0520191C43BDA521B1D314C275330A	7.925	Russian	89 50 4E 47 (
manifest	1	manifest	.rsrc:0x002AB240	796	0.03 %	F369CE03EF6E6F75BE36380826E9BBAA	5.121	English-US	EF BB BF 3C
AFX_DIALOG_LAYO...	102	custom	.rsrc:0x002AB238	2	-	C4103F122D27677C9DB144CAE1394A66	0.000	English-US	00 00
dialog	102	dialog	.rsrc:0x002AB038	512	0.02 %	7B5F10CA83958CE46E0798E1E0887D55	3.207	English-US	01 00 FF FF (
dialog	129	dialog	.rsrc:0x002805B0	204	0.01 %	3CFA42059AE377E32C0EB93D3769333B	3.053	Russian	01 00 FF FF (
icon-group	140	icon-group	.rsrc:0x002AAFA8	132	0.00 %	533E005C6C3606F2CACB940A46098112	3.014	Russian	00 00 01 00 (
AFX_DIALOG_LAYO...	142	custom	.rsrc:0x002AB030	2	-	C4103F122D27677C9DB144CAE1394A66	0.000	Russian	00 00
dialog	142	dialog	.rsrc:0x00280680	188	0.01 %	B998255787EB07E1649ADBEFE87455BF	3.166	Russian	01 00 FF FF (
icon	2	icon	.rsrc:0x002837E8	67624	2.29 %	9FC17A8E2117222462E076FDB65B0DC7	3.333	Russian	28 00 00 00 8
icon	3	icon	.rsrc:0x00294010	38056	1.29 %	59BAF04C61A528D80FA940F1094331F8	3.962	Russian	28 00 00 00 6
icon	4	icon	.rsrc:0x0029D4B8	21640	0.73 %	E9F141578E1DA9A40BD337D50DE577FA	4.050	Russian	28 00 00 00 4
icon	5	icon	.rsrc:0x002A2940	16936	0.57 %	A92F370A1DDC05627F7ABC0AF2775CF2	3.807	Russian	28 00 00 00 4
icon	6	icon	.rsrc:0x002A6B68	9640	0.33 %	C8F0BE05F04AAC749D9EEC2832184478	4.235	Russian	28 00 00 00 3
icon	7	icon	.rsrc:0x002A9110	4264	0.14 %	C8A0690AD014B81B929BE6ADAADE93AE	4.438	Russian	28 00 00 00 2
icon	8	icon	.rsrc:0x002AA1B8	2440	0.08 %	75D9D9F97ED5F972A8D70BE7B5D1CF3D	4.691	Russian	28 00 00 00 1
icon	9	icon	.rsrc:0x002AAB40	1128	0.04 %	9E1C5EA08AEC73C5FE89EA102A6246E3	5.124	Russian	28 00 00 00 1

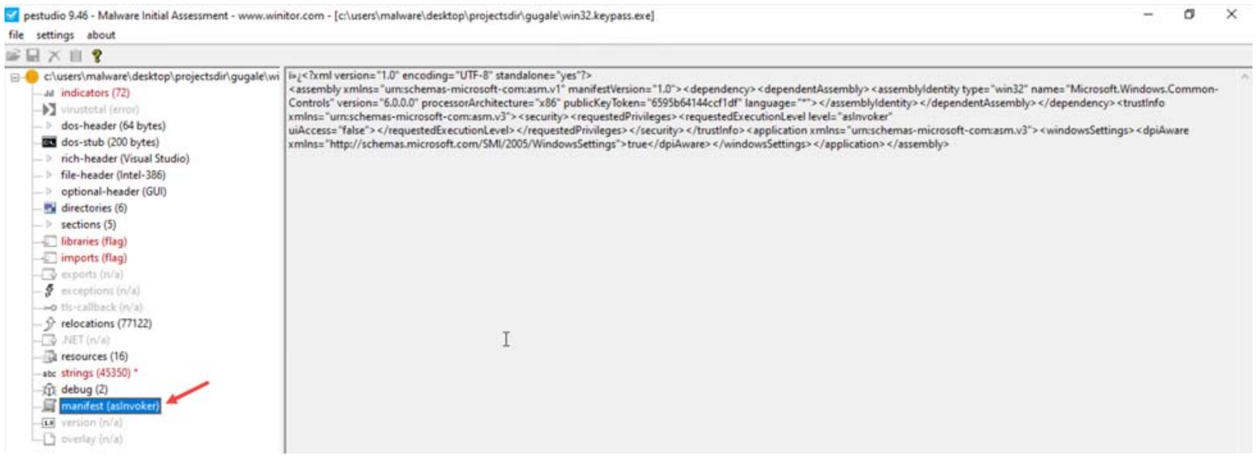
Figure 8 Resources present in the binary

The sample includes 16 resources as shown in the image above. The language for most resources is set to Russian indicating that the malware might have Russian origins. Following types of resources are present:

1. **Icon:** The first resource is a .png file containing the file icon. Dumping the resource shows a Bitcoin like icon. There are other resources listed as icons but dumping them as .jpg/.png/.ico files doesn't give a valid image.

Figure 9 First resource is the file icon

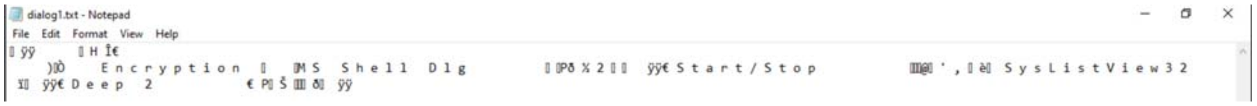
2. **Manifest:** The resource section contains an XML asInvoker manifest file. The “asInvoker” manifest is used to launch applications which require the same level of privilege as the current user. For any tasks which require privilege escalation, the program asks for permission separately through a privilege escalation dialog box.



```
ï»¿<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0"><dependency><dependentAssembly><assemblyIdentity type="win32" name="Microsoft.Windows.Common-Controls" version="6.0.0.0" processorArchitecture="x86" publicKeyToken="6595b64144ccf1df" language="*"></assemblyIdentity></dependentAssembly></dependency><trustInfo xmlns="urn:schemas-microsoft-com:asm.v3"><security><requestedPrivileges><requestedExecutionLevel level="asInvoker" uiAccess="false"></requestedExecutionLevel></requestedPrivileges></security></trustInfo><application xmlns="urn:schemas-microsoft-com:asm.v3"><windowsSettings><dpiAware xmlns="http://schemas.microsoft.com/SMI/2005/WindowsSettings">true</dpiAware></windowsSettings></application></assembly>
```

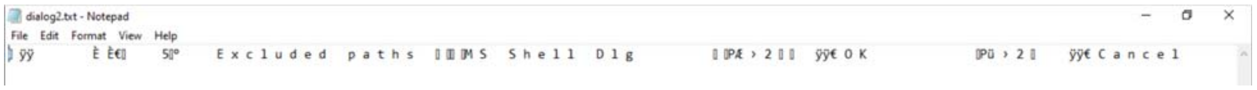
Figure 10 asInvoker manifest

3. **Dialogs:** The resources contain two dialogs. Dumping them as txt files shows some text which might be displayed to the user in a dialog box. But dynamic analysis on the program shows that dialog boxes with these messages are never shown to the user. Hence, the exact purpose of these dialog files is not clear.



```
 ÿÿ      H Î€
    )Ò    E n c r y p t i o n       M S   S h e l l   D l g        Pð ¼ 2     ÿÿ€ S t a r t / S t o p        @ ' ,  è  S y s L i s t V i e w 3 2
 ï  ÿÿ€ D e e p   2            € P Š   ð  ÿÿ
```

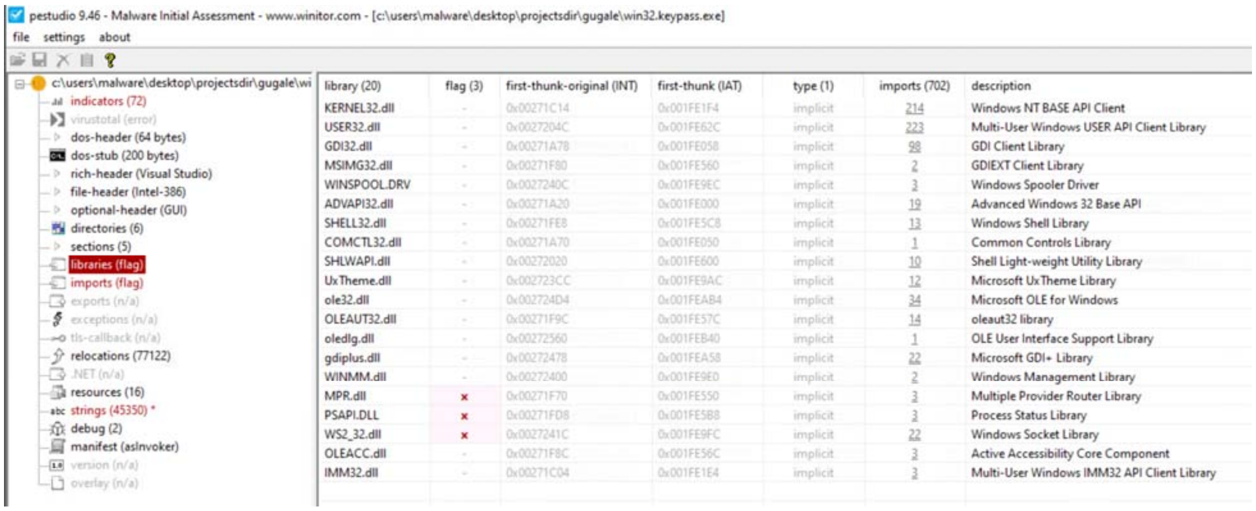
Figure 11 Dialog file 1 dump



```
 ÿÿ        È È€     5 °    E x c l u d e d   p a t h s      M S   S h e l l   D l g         PÆ › 2     ÿÿ€ O K          Pü › 2     ÿÿ€ C a n c e l
```

Figure 12 Dialog file 2 dump

Imported Libraries


pestudio 9.46 - Malware Initial Assessment - www.winitor.com - [c:\users\malware\desktop\projectsdir\gugale\win32.keypass.exe]

library (20)	flag (3)	first-thunk-original (INT)	first-thunk (IAT)	type (1)	imports (702)	description
KERNEL32.dll	-	0x00271C14	0x001FE1F4	implicit	214	Windows NT BASE API Client
USER32.dll	-	0x0027204C	0x001FE62C	implicit	223	Multi-User Windows USER API Client Library
GDI32.dll	-	0x00271A78	0x001FE058	implicit	98	GDI Client Library
MSIMG32.dll	-	0x00271F80	0x001FE560	implicit	2	GDIEXT Client Library
WINSPOOL.DRV	-	0x0027240C	0x001FE9EC	implicit	3	Windows Spooler Driver
ADVAPI32.dll	-	0x00271A20	0x001FE000	implicit	19	Advanced Windows 32 Base API
SHELL32.dll	-	0x00271FE8	0x001FE5C8	implicit	13	Windows Shell Library
COMCTL32.dll	-	0x00271A70	0x001FE050	implicit	1	Common Controls Library
SHLWAPI.dll	-	0x00272020	0x001FE600	implicit	10	Shell Light-weight Utility Library
UxTheme.dll	-	0x002723CC	0x001FE9AC	implicit	12	Microsoft UxTheme Library
ole32.dll	-	0x002724D4	0x001FEAB4	implicit	34	Microsoft OLE for Windows
OLEAUT32.dll	-	0x00271F9C	0x001FE57C	implicit	14	oleaut32 library
oledlg.dll	-	0x00272560	0x001FEB40	implicit	1	OLE User Interface Support Library
gdiplus.dll	-	0x00272478	0x001FEA58	implicit	22	Microsoft GDI+ Library
WINMM.dll	-	0x00272400	0x001FE9E0	implicit	2	Windows Management Library
MPR.dll	x	0x00271F70	0x001FE550	implicit	3	Multiple Provider Router Library
PSAPI.DLL	x	0x00271FD8	0x001FE5B8	implicit	3	Process Status Library
WS2_32.dll	x	0x0027241C	0x001FE9FC	implicit	22	Windows Socket Library
OLEACC.dll	-	0x00271F8C	0x001FE56C	implicit	3	Active Accessibility Core Component
IMM32.dll	-	0x00271C04	0x001FE1E4	implicit	3	Multi-User Windows IMM32 API Client Library

Figure 13 Statically imported libraries

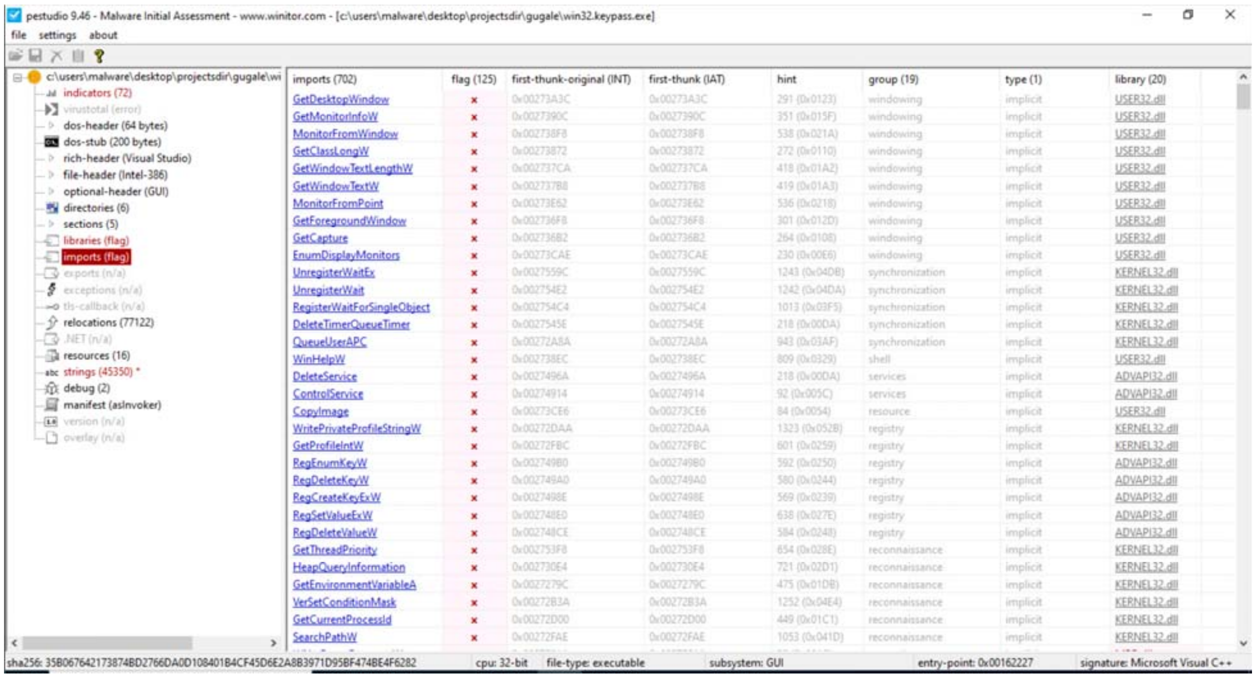
The sample imports numerous libraries statically as can be seen from the image above. Some of the statically imported libraries are:

1. **MPR.dll** - MPR stands for Multiple Provider Router, and the MPR.dll file contains functions related to network connections and remote access. It is flagged by PEStudio as it can be used for malicious tasks.
2. **PSAPI.dll** - PSAPI.dll (Process Status API) file contains functions related to the management and monitoring of processes and their performance. It contains functions for enumerating the processes and modules currently running.
3. **WS2_32.dll** – This DLL contains Berkley socket API functions for communication between two machines. The malware might be using this DLL to send information to its C&C server.
4. **GDIPLUS.dll** – This library is used for rendering 2D graphics and images. It contains functions for creating and manipulating graphics objects, working with fonts and colors, and rendering images and text. The malware might be using this library to display dialog boxes and change the desktop wallpaper.
5. **OLE32.dll –** This DLL contains functions for manipulating COM (Component Object Model) objects. It contains functions for object creation, memory management, and interprocess communication between COM objects.
6. **COMCTL32.dll** – This library contains functions and resources that are used by the Windows operating system and applications to create and manage common controls, such as buttons, menus, toolbars, and status bars. The malware might be using this to display the privilege escalation pop-up.
7. **SHLWAPI.dll** – Contains functions related to file and folder management, path handling, and string manipulation. The malware might be using them to traverse folders to get paths of files to encrypt.

8. **SHELL32.dll** – This DLL contains functions like ShellExecuteW and ShellExecuteExW which are used to launch another process. The malware might be using it to launch its subprocesses.

Interesting or suspicious functions

PEStudio identifies 125 suspicious functions.



pestudio 9.46 - Malware Initial Assessment - www.winitor.com - [c:\users\malware\desktop\projectsdir\gugale\win32.keypass.exe]

file settings about

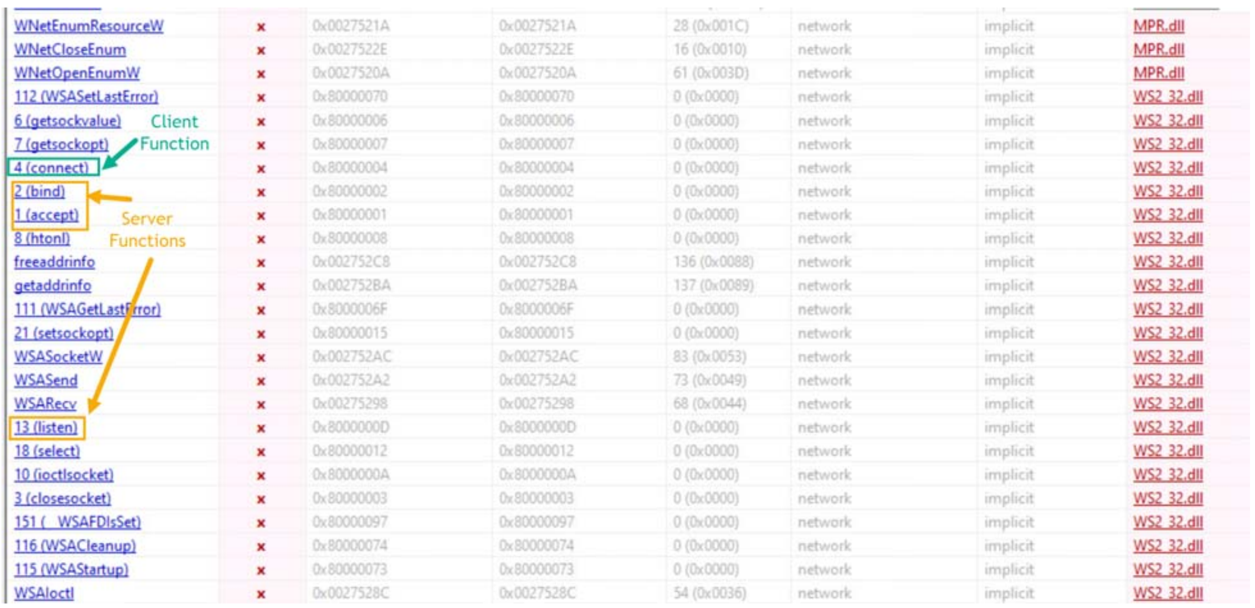
- c:\users\malware\desktop\projectsdir\gugale\wi
  - indicators (72)
  - virustotal (error)
  - dos-header (64 bytes)
  - dos-stub (200 bytes)
  - rich-header (Visual Studio)
  - file-header (Intel-386)
  - optional-header (GUI)
  - directories (6)
  - sections (5)
  - libraries (flag)
  - imports (flag)
  - exports (n/a)
  - exceptions (n/a)
  - tls-callback (n/a)
  - relocations (77122)
  - .NET (n/a)
  - resources (16)
  - strings (45350) *
  - debug (2)
  - manifest (asInvoker)
  - version (n/a)
  - overlay (n/a)

imports (702)	flag (125)	first-thunk-original (INT)	first-thunk (IAT)	hint	group (19)	type (1)	library (20)
GetDesktopWindow	x	0x00273A3C	0x00273A3C	291 (0x0123)	windowing	implicit	USER32.dll
GetMonitorInfoW	x	0x0027390C	0x0027390C	351 (0x015F)	windowing	implicit	USER32.dll
MonitorFromWindow	x	0x002738F8	0x002738F8	538 (0x021A)	windowing	implicit	USER32.dll
GetClassLongW	x	0x00273872	0x00273872	272 (0x0110)	windowing	implicit	USER32.dll
GetWindowTextLengthW	x	0x002737CA	0x002737CA	418 (0x01A2)	windowing	implicit	USER32.dll
GetWindowTextW	x	0x002737B8	0x002737B8	419 (0x01A3)	windowing	implicit	USER32.dll
MonitorFromPoint	x	0x00273E62	0x00273E62	536 (0x0218)	windowing	implicit	USER32.dll
GetForegroundWindow	x	0x002736F8	0x002736F8	301 (0x012D)	windowing	implicit	USER32.dll
GetCapture	x	0x002736B2	0x002736B2	264 (0x0108)	windowing	implicit	USER32.dll
EnumDisplayMonitors	x	0x00273CAE	0x00273CAE	230 (0x00E6)	windowing	implicit	USER32.dll
UnregisterWaitEx	x	0x0027559C	0x0027559C	1243 (0x04DB)	synchronization	implicit	KERNEL32.dll
UnregisterWait	x	0x002754E2	0x002754E2	1242 (0x04DA)	synchronization	implicit	KERNEL32.dll
RegisterWaitForSingleObject	x	0x002754C4	0x002754C4	1013 (0x03F5)	synchronization	implicit	KERNEL32.dll
DeleteTimerQueueTimer	x	0x0027545E	0x0027545E	218 (0x00DA)	synchronization	implicit	KERNEL32.dll
QueueUserAPC	x	0x00272A8A	0x00272A8A	943 (0x03AF)	synchronization	implicit	KERNEL32.dll
WinHelpW	x	0x002738EC	0x002738EC	809 (0x0329)	shell	implicit	USER32.dll
DeleteService	x	0x0027496A	0x0027496A	218 (0x00DA)	services	implicit	ADVAPI32.dll
ControlService	x	0x00274914	0x00274914	92 (0x005C)	services	implicit	ADVAPI32.dll
CopyImage	x	0x00273CE6	0x00273CE6	84 (0x0054)	resource	implicit	USER32.dll
WritePrivateProfileStringW	x	0x00272DAA	0x00272DAA	1323 (0x052B)	registry	implicit	KERNEL32.dll
GetProfileIntW	x	0x00272FBC	0x00272FBC	601 (0x0259)	registry	implicit	KERNEL32.dll
RegEnumKeyW	x	0x002749B0	0x002749B0	592 (0x0250)	registry	implicit	ADVAPI32.dll
RegDeleteKeyW	x	0x002749A0	0x002749A0	580 (0x0244)	registry	implicit	ADVAPI32.dll
RegCreateKeyExW	x	0x0027498E	0x0027498E	569 (0x0239)	registry	implicit	ADVAPI32.dll
RegSetValueExW	x	0x002748E0	0x002748E0	638 (0x027E)	registry	implicit	ADVAPI32.dll
RegDeleteValueW	x	0x002748CE	0x002748CE	584 (0x0248)	registry	implicit	ADVAPI32.dll
GetThreadPriority	x	0x002753F8	0x002753F8	654 (0x028E)	reconnaissance	implicit	KERNEL32.dll
HeapQueryInformation	x	0x002730E4	0x002730E4	721 (0x02D1)	reconnaissance	implicit	KERNEL32.dll
GetEnvironmentVariableA	x	0x0027279C	0x0027279C	475 (0x01DB)	reconnaissance	implicit	KERNEL32.dll
VerSetConditionMask	x	0x00272B3A	0x00272B3A	1252 (0x04E4)	reconnaissance	implicit	KERNEL32.dll
GetCurrentProcessId	x	0x00272D00	0x00272D00	449 (0x01C1)	reconnaissance	implicit	KERNEL32.dll
SearchPathW	x	0x00272FAE	0x00272FAE	1053 (0x041D)	reconnaissance	implicit	KERNEL32.dll

sha256: 35B067642173874BD2766DA0D108401B4CF45D6E2A8B3971D95BF474BE4F6282 | cpu: 32-bit | file-type: executable | subsystem: GUI | entry-point: 0x00162227 | signature: Microsoft Visual C++

Figure 14 Suspicious Imports

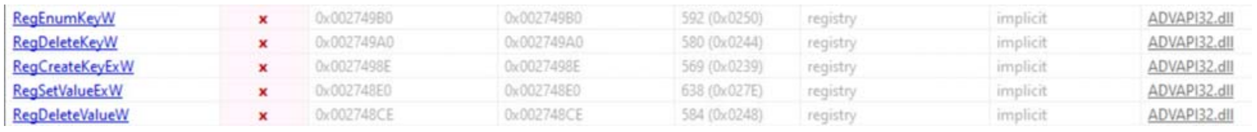
- **Network related functions:** The malware imports both client and server related functions from WS2_32.dll. Functions imported from MPR.dll suggest that the sample is iterating over network resources like shared folders, printers, and servers and retrieving information about them. The malware might be trying to find other computers on the network to infect them.



WNetEnumResourceW	✗	0x0027521A	0x0027521A	28 (0x001C)	network	implicit	MPR.dll
WNetCloseEnum	✗	0x0027522E	0x0027522E	16 (0x0010)	network	implicit	MPR.dll
WNetOpenEnumW	✗	0x0027520A	0x0027520A	61 (0x003D)	network	implicit	MPR.dll
112 (WSASetLastError)	✗	0x80000070	0x80000070	0 (0x0000)	network	implicit	WS2 32.dll
6 (getsockvalue)	✗	0x80000006	0x80000006	0 (0x0000)	network	implicit	WS2 32.dll
7 (getsockopt)	✗	0x80000007	0x80000007	0 (0x0000)	network	implicit	WS2 32.dll
4 (connect)	✗	0x80000004	0x80000004	0 (0x0000)	network	implicit	WS2 32.dll
2 (bind)	✗	0x80000002	0x80000002	0 (0x0000)	network	implicit	WS2 32.dll
1 (accept)	✗	0x80000001	0x80000001	0 (0x0000)	network	implicit	WS2 32.dll
8 (htonl)	✗	0x80000008	0x80000008	0 (0x0000)	network	implicit	WS2 32.dll
freeaddrinfo	✗	0x002752C8	0x002752C8	136 (0x0088)	network	implicit	WS2 32.dll
getaddrinfo	✗	0x002752BA	0x002752BA	137 (0x0089)	network	implicit	WS2 32.dll
111 (WSAGetLastError)	✗	0x8000006F	0x8000006F	0 (0x0000)	network	implicit	WS2 32.dll
21 (setsockopt)	✗	0x80000015	0x80000015	0 (0x0000)	network	implicit	WS2 32.dll
WSASocketW	✗	0x002752AC	0x002752AC	83 (0x0053)	network	implicit	WS2 32.dll
WSASend	✗	0x002752A2	0x002752A2	73 (0x0049)	network	implicit	WS2 32.dll
WSARecv	✗	0x00275298	0x00275298	68 (0x0044)	network	implicit	WS2 32.dll
13 (listen)	✗	0x8000000D	0x8000000D	0 (0x0000)	network	implicit	WS2 32.dll
18 (select)	✗	0x80000012	0x80000012	0 (0x0000)	network	implicit	WS2 32.dll
10 (ioctlsocket)	✗	0x8000000A	0x8000000A	0 (0x0000)	network	implicit	WS2 32.dll
3 (closesocket)	✗	0x80000003	0x80000003	0 (0x0000)	network	implicit	WS2 32.dll
151 (WSAFDIsSet)	✗	0x80000097	0x80000097	0 (0x0000)	network	implicit	WS2 32.dll
116 (WSACleanup)	✗	0x80000074	0x80000074	0 (0x0000)	network	implicit	WS2 32.dll
115 (WSAStartup)	✗	0x80000073	0x80000073	0 (0x0000)	network	implicit	WS2 32.dll
WSAIoctl	✗	0x0027528C	0x0027528C	54 (0x0036)	network	implicit	WS2 32.dll

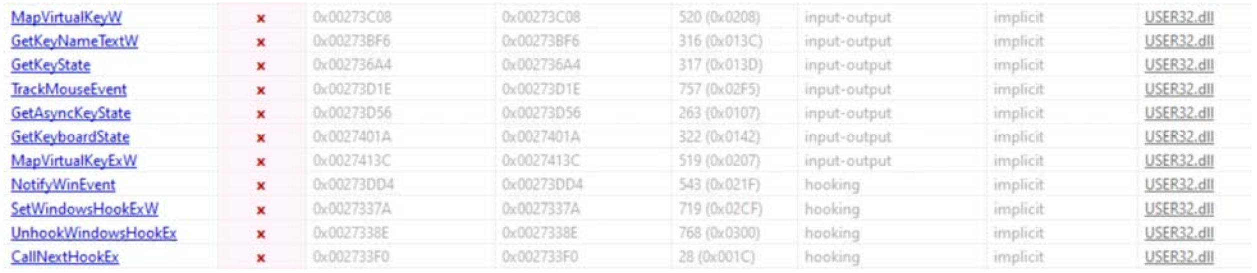
Figure 15 Network related imports

- **Registry related functions:** The sample has imports related to registry modification. These might be related to malware's persistence mechanisms.



RegEnumKeyW	✗	0x002749B0	0x002749B0	592 (0x0250)	registry	implicit	ADVAPI32.dll
RegDeleteKeyW	✗	0x002749A0	0x002749A0	580 (0x0244)	registry	implicit	ADVAPI32.dll
RegCreateKeyExW	✗	0x0027498E	0x0027498E	569 (0x0239)	registry	implicit	ADVAPI32.dll
RegSetValueExW	✗	0x002748E0	0x002748E0	638 (0x027E)	registry	implicit	ADVAPI32.dll
RegDeleteValueW	✗	0x002748CE	0x002748CE	584 (0x0248)	registry	implicit	ADVAPI32.dll

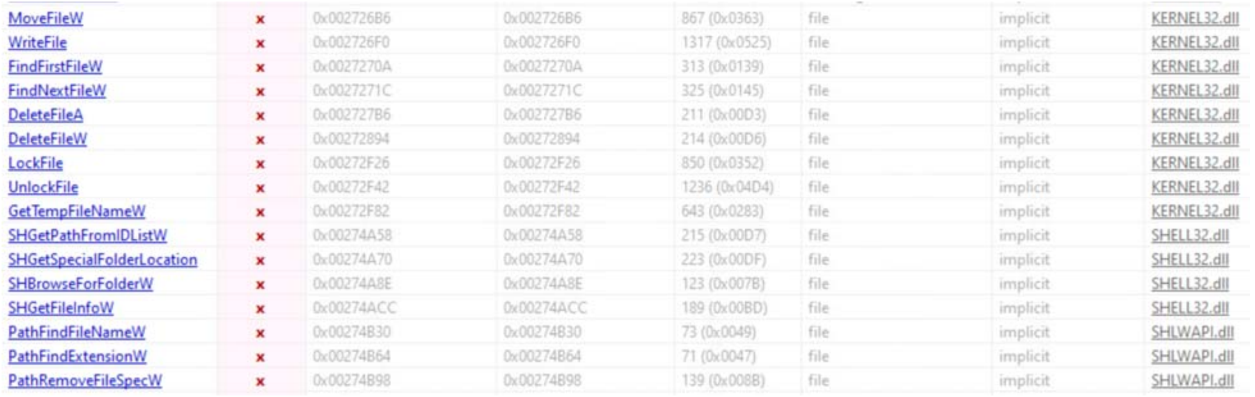
- **Keylogging related functions:** The malware uses various suspicious functions from USER32.dll like GetKeyState, GetKeyboardState and SetWindowsHookEx. This indicates that the malware monitors the keys pressed by the user and calls a hooking function if a particular key combination is pressed.



MapVirtualKeyW	✗	0x00273C08	0x00273C08	520 (0x0208)	input-output	implicit	USER32.dll
GetKeyNameTextW	✗	0x00273BF6	0x00273BF6	316 (0x013C)	input-output	implicit	USER32.dll
GetKeyState	✗	0x002736A4	0x002736A4	317 (0x013D)	input-output	implicit	USER32.dll
TrackMouseEvent	✗	0x00273D1E	0x00273D1E	757 (0x02F5)	input-output	implicit	USER32.dll
GetAsyncKeyState	✗	0x00273D56	0x00273D56	263 (0x0107)	input-output	implicit	USER32.dll
GetKeyboardState	✗	0x0027401A	0x0027401A	322 (0x0142)	input-output	implicit	USER32.dll
MapVirtualKeyExW	✗	0x0027413C	0x0027413C	519 (0x0207)	input-output	implicit	USER32.dll
NotifyWinEvent	✗	0x00273DD4	0x00273DD4	543 (0x021F)	hooking	implicit	USER32.dll
SetWindowsHookExW	✗	0x0027337A	0x0027337A	719 (0x02CF)	hooking	implicit	USER32.dll
UnhookWindowsHookEx	✗	0x0027338E	0x0027338E	768 (0x0300)	hooking	implicit	USER32.dll
CallNextHookEx	✗	0x002733F0	0x002733F0	28 (0x001C)	hooking	implicit	USER32.dll

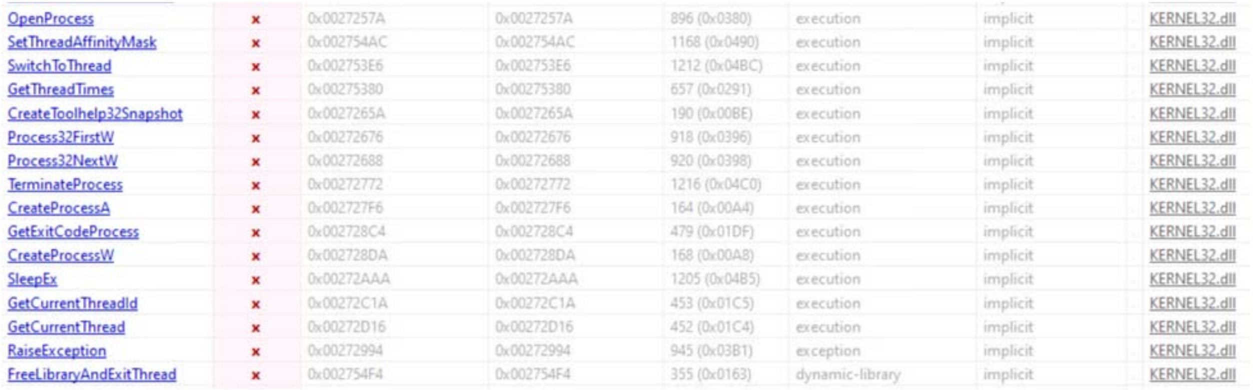
- **File and folder iteration related functions:** As can be seen from the following imports, the malware tries to find all the files in a directory using functions like FindFirstFile and FindNextFile to encrypt them. It fetches the file extension through PathFindExtension possibly to exclude files which already have .KEYPASS extension. GetSpecialFolderLocation is used to get location of

C:\Users\<username>\AppData\Local folder. The sample then copies itself at this location. It uses DeleteFile to delete its executable from the initial location.



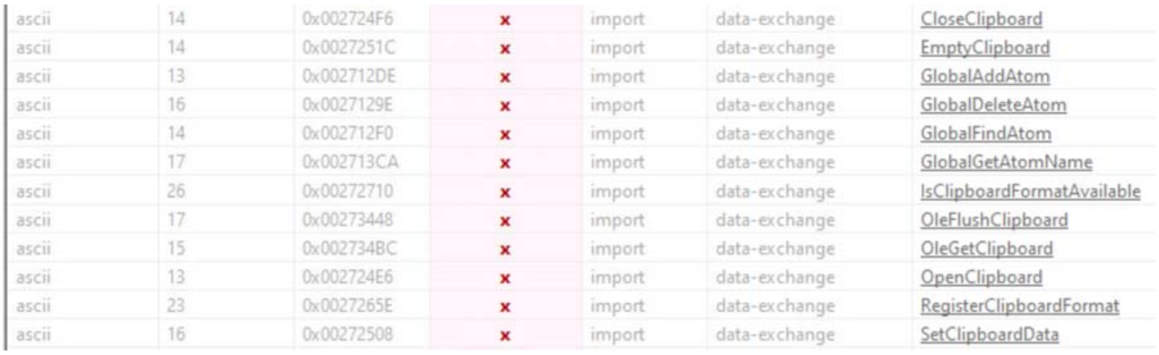
MoveFileW	x	0x002726B6	0x002726B6	867 (0x0363)	file	implicit	KERNEL32.dll
WriteFile	x	0x002726F0	0x002726F0	1317 (0x0525)	file	implicit	KERNEL32.dll
FindFirstFileW	x	0x0027270A	0x0027270A	313 (0x0139)	file	implicit	KERNEL32.dll
FindNextFileW	x	0x0027271C	0x0027271C	325 (0x0145)	file	implicit	KERNEL32.dll
DeleteFileA	x	0x002727B6	0x002727B6	211 (0x00D3)	file	implicit	KERNEL32.dll
DeleteFileW	x	0x00272894	0x00272894	214 (0x00D6)	file	implicit	KERNEL32.dll
LockFile	x	0x00272F26	0x00272F26	850 (0x0352)	file	implicit	KERNEL32.dll
UnlockFile	x	0x00272F42	0x00272F42	1236 (0x04D4)	file	implicit	KERNEL32.dll
GetTempFileNameW	x	0x00272F82	0x00272F82	643 (0x0283)	file	implicit	KERNEL32.dll
SHGetPathFromIDListW	x	0x00274A58	0x00274A58	215 (0x00D7)	file	implicit	SHELL32.dll
SHGetSpecialFolderLocation	x	0x00274A70	0x00274A70	223 (0x00DF)	file	implicit	SHELL32.dll
SHBrowseForFolderW	x	0x00274A8E	0x00274A8E	123 (0x007B)	file	implicit	SHELL32.dll
SHGetFileInfoW	x	0x00274ACC	0x00274ACC	189 (0x00BD)	file	implicit	SHELL32.dll
PathFindFileNameW	x	0x00274B30	0x00274B30	73 (0x0049)	file	implicit	SHLWAPI.dll
PathFindExtensionW	x	0x00274B64	0x00274B64	71 (0x0047)	file	implicit	SHLWAPI.dll
PathRemoveFileSpecW	x	0x00274B98	0x00274B98	139 (0x008B)	file	implicit	SHLWAPI.dll

- **Process related functions:** The malware imports multiple process and thread related functions from KERNEL32.dll. It might be using these to manipulate its subprocesses and their threads.



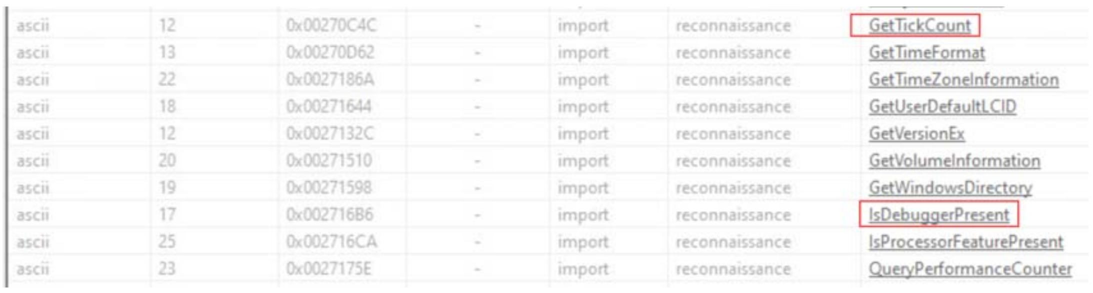
OpenProcess	x	0x0027257A	0x0027257A	896 (0x0380)	execution	implicit	KERNEL32.dll
SetThreadAffinityMask	x	0x002754AC	0x002754AC	1168 (0x0490)	execution	implicit	KERNEL32.dll
SwitchToThread	x	0x002753E6	0x002753E6	1212 (0x04BC)	execution	implicit	KERNEL32.dll
GetThreadTimes	x	0x00275380	0x00275380	657 (0x0291)	execution	implicit	KERNEL32.dll
CreateToolhelp32Snapshot	x	0x0027265A	0x0027265A	190 (0x00BE)	execution	implicit	KERNEL32.dll
Process32FirstW	x	0x00272676	0x00272676	918 (0x0396)	execution	implicit	KERNEL32.dll
Process32NextW	x	0x00272688	0x00272688	920 (0x0398)	execution	implicit	KERNEL32.dll
TerminateProcess	x	0x00272772	0x00272772	1216 (0x04C0)	execution	implicit	KERNEL32.dll
CreateProcessA	x	0x002727F6	0x002727F6	164 (0x00A4)	execution	implicit	KERNEL32.dll
GetExitCodeProcess	x	0x002728C4	0x002728C4	479 (0x01DF)	execution	implicit	KERNEL32.dll
CreateProcessW	x	0x002728DA	0x002728DA	168 (0x00A8)	execution	implicit	KERNEL32.dll
SleepEx	x	0x00272AAA	0x00272AAA	1205 (0x04B5)	execution	implicit	KERNEL32.dll
GetCurrentThreadId	x	0x00272C1A	0x00272C1A	453 (0x01C5)	execution	implicit	KERNEL32.dll
GetCurrentThread	x	0x00272D16	0x00272D16	452 (0x01C4)	execution	implicit	KERNEL32.dll
RaiseException	x	0x00272994	0x00272994	945 (0x03B1)	exception	implicit	KERNEL32.dll
FreeLibraryAndExitThread	x	0x002754F4	0x002754F4	355 (0x0163)	dynamic-library	implicit	KERNEL32.dll

- **Clipboard related functions:** The malware might be manipulating clipboard contents judging from all the clipboard functions it imports.



ascii	14	0x002724F6	x	import	data-exchange	CloseClipboard
ascii	14	0x0027251C	x	import	data-exchange	EmptyClipboard
ascii	13	0x002712DE	x	import	data-exchange	GlobalAddAtom
ascii	16	0x0027129E	x	import	data-exchange	GlobalDeleteAtom
ascii	14	0x002712F0	x	import	data-exchange	GlobalFindAtom
ascii	17	0x002713CA	x	import	data-exchange	GlobalGetAtomName
ascii	26	0x00272710	x	import	data-exchange	IsClipboardFormatAvailable
ascii	17	0x00273448	x	import	data-exchange	OleFlushClipboard
ascii	15	0x002734BC	x	import	data-exchange	OleGetClipboard
ascii	13	0x002724E6	x	import	data-exchange	OpenClipboard
ascii	23	0x0027265E	x	import	data-exchange	RegisterClipboardFormat
ascii	16	0x00272508	x	import	data-exchange	SetClipboardData

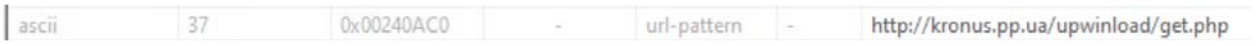
- **Anti-analysis related functions:** The sample imports **GetTickCount** and **IsDebuggerPresent** functions which are used to thwart debugging. This suggests that the malware might be doing some anti-debugging.



ascii	12	0x00270C4C	-	import	reconnaissance	GetTickCount
ascii	13	0x00270D62	-	import	reconnaissance	GetTimeFormat
ascii	22	0x0027186A	-	import	reconnaissance	GetTimeZoneInformation
ascii	18	0x00271644	-	import	reconnaissance	GetUserDefaultLCID
ascii	12	0x0027132C	-	import	reconnaissance	GetVersionEx
ascii	20	0x00271510	-	import	reconnaissance	GetVolumeInformation
ascii	19	0x00271598	-	import	reconnaissance	GetWindowsDirectory
ascii	17	0x002716B6	-	import	reconnaissance	IsDebuggerPresent
ascii	25	0x002716CA	-	import	reconnaissance	IsProcessorFeaturePresent
ascii	23	0x0027175E	-	import	reconnaissance	QueryPerformanceCounter

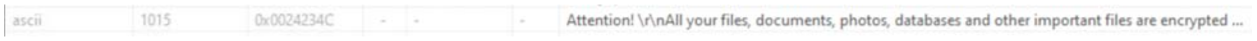
Interesting or suspicious strings

1. **URL**: The binary has a string referring to a URL. This could be the URL of the C&C server from where the malware might be getting instructions and where the sample might be sending some exfiltrated data.



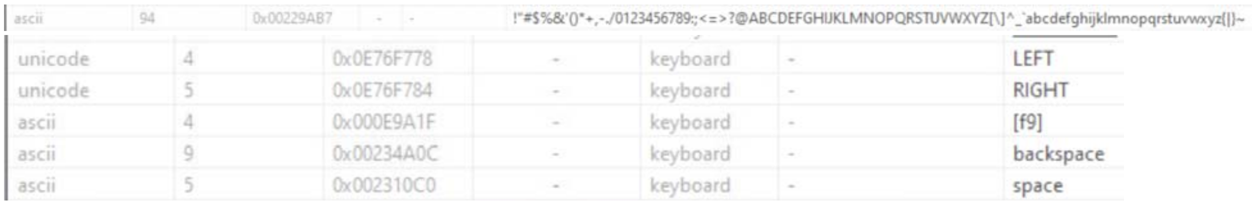
ascii	37	0x00240AC0	-	url-pattern	-	http://kronus.pp.ua/upwinload/get.php

2. **Ransom note string:** The sample drops ransom notes with the following text in every folder.



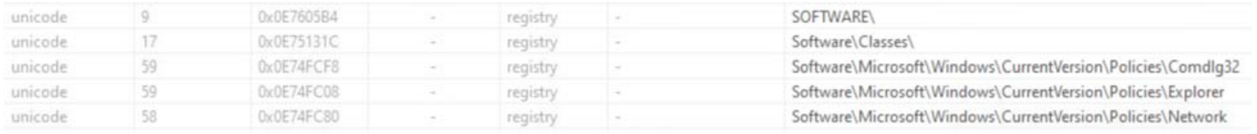
ascii	1015	0x0024234C	-	-	-	Attention! \r\nAll your files, documents, photos, databases and other important files are encrypted ...

3. **Keyboard related strings:** The sample contains a string with all printable characters on the keyboard. It also contains strings for non-printable characters like LEFT, RIGHT and backspace. This hints that the malware might be monitoring the user's key strokes.



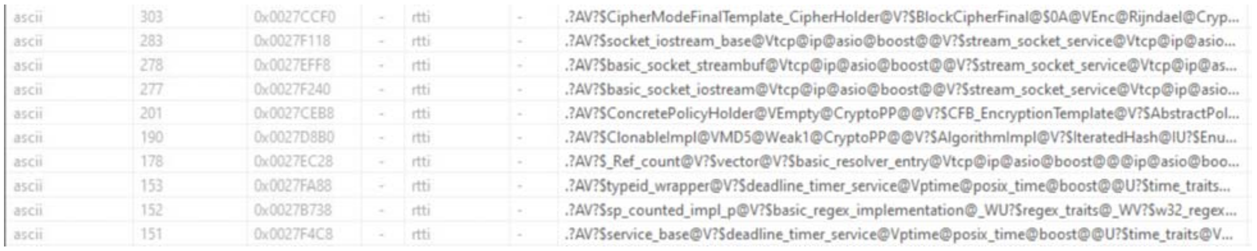
ascii	94	0x00229AB7	-	-	-	!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
unicode	4	0x0E76F778	-	keyboard	-	LEFT
unicode	5	0x0E76F784	-	keyboard	-	RIGHT
ascii	4	0x000E9A1F	-	keyboard	-	[f9]
ascii	9	0x00234A0C	-	keyboard	-	backspace
ascii	5	0x002310C0	-	keyboard	-	space

4. **Registry keys:** The sample probably tries to change the access permission of Explorer, Network and Comdlg32 functions by modifying their registry keys.



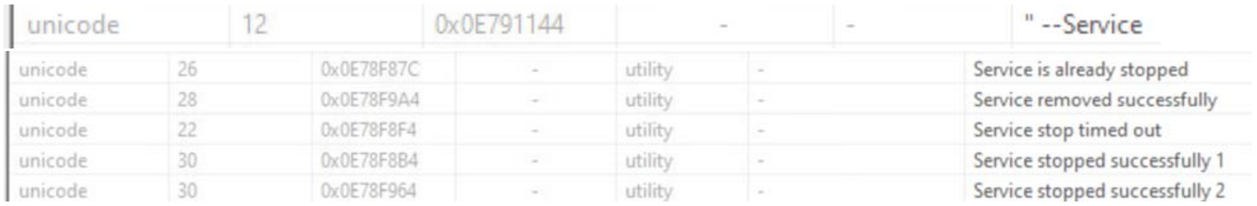
unicode	9	0x0E7605B4	-	registry	-	SOFTWARE\
unicode	17	0x0E75131C	-	registry	-	Software\Classes\
unicode	59	0x0E74FCF8	-	registry	-	Software\Microsoft\Windows\CurrentVersion\Policies\Comdlg32
unicode	59	0x0E74FC08	-	registry	-	Software\Microsoft\Windows\CurrentVersion\Policies\Explorer
unicode	58	0x0E74FC80	-	registry	-	Software\Microsoft\Windows\CurrentVersion\Policies\Network

5. **RTTI strings:** The sample has a lot of RTTI (Run-time type information) strings. RTTI is used by C++ programs to determine the type of an object during runtime. This indicates that the malware was written in C++.



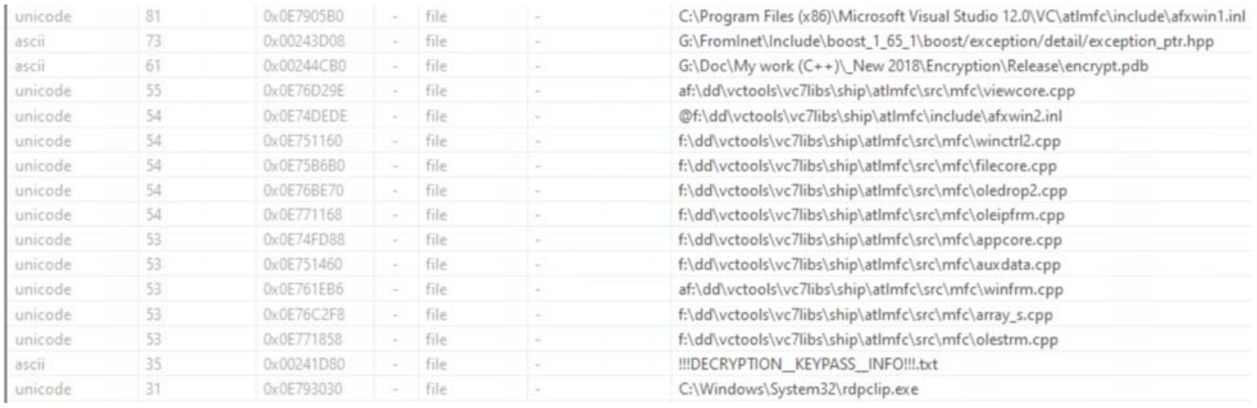
ascii	303	0x0027CCF0	-	rtti	-	.?AV?$CipherModeFinalTemplate_CipherHolder@V?$BlockCipherFinal@$0A@VEnc@Rijndael@Cryp...
ascii	283	0x0027F118	-	rtti	-	.?AV?$socket_iostream_base@Vtcp@ip@asio@boost@@V?$stream_socket_service@Vtcp@ip@asio...
ascii	278	0x0027EFF8	-	rtti	-	.?AV?$basic_socket_streambuf@Vtcp@ip@asio@boost@@V?$stream_socket_service@Vtcp@ip@as...
ascii	277	0x0027F240	-	rtti	-	.?AV?$basic_socket_iostream@Vtcp@ip@asio@boost@@V?$stream_socket_service@Vtcp@ip@asio...
ascii	201	0x0027CEB8	-	rtti	-	.?AV?$ConcretePolicyHolder@VEmpty@CryptoPP@@V?$CFB_EncryptionTemplate@V?$AbstractPol...
ascii	190	0x0027D8B0	-	rtti	-	.?AV?$ClonableImpl@VMD5@Weak1@CryptoPP@@V?$AlgorithmImpl@V?$IteratedHash@IU?$Enu...
ascii	178	0x0027EC28	-	rtti	-	.?AV?$_Ref_count@V?$vector@V?$basic_resolver_entry@Vtcp@ip@asio@boost@@@ip@asio@boo...
ascii	153	0x0027FA88	-	rtti	-	.?AV?$typeid_wrapper@V?$deadline_timer_service@Vptime@posix_time@boost@@U?$time_traits...
ascii	152	0x0027B738	-	rtti	-	.?AV?$sp_counted_impl_p@V?$basic_regex_implementation@_WU?$regex_traits@_WV?$w32_regex...
ascii	151	0x0027F4C8	-	rtti	-	.?AV?$service_base@V?$deadline_timer_service@Vptime@posix_time@boost@@U?$time_traits@V...

6. **Service:** The sample might be starting a service with the –Service flag. Some other strings are also present which seem like messages recorded when services are stopped.



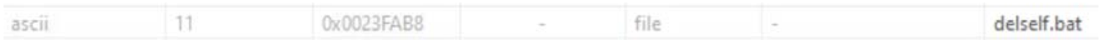
unicode	12	0x0E791144	-	-		" --Service
unicode	26	0x0E78F87C	-	utility	-	Service is already stopped
unicode	28	0x0E78F9A4	-	utility	-	Service removed successfully
unicode	22	0x0E78F8F4	-	utility	-	Service stop timed out
unicode	30	0x0E78F8B4	-	utility	-	Service stopped successfully 1
unicode	30	0x0E78F964	-	utility	-	Service stopped successfully 2

7. **File names:**



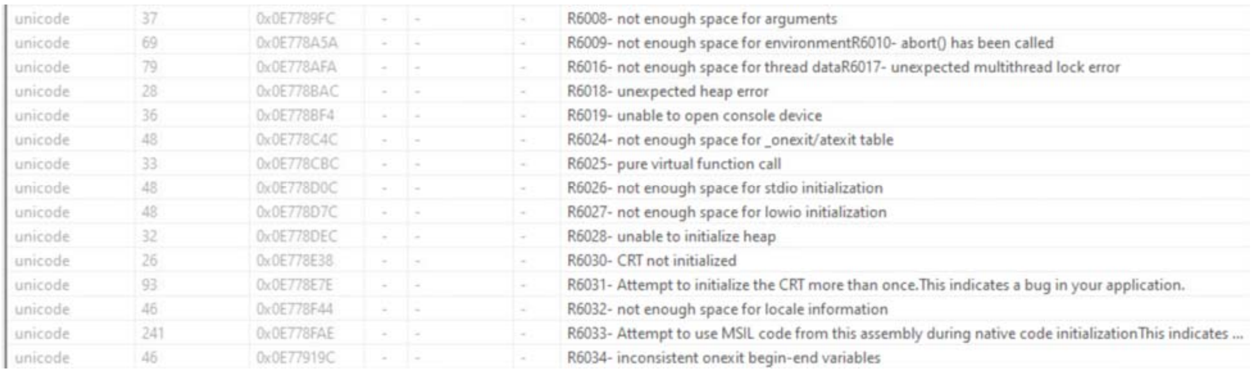
unicode	81	0x0E7905B0	-	file	-	C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\atlmfc\include\afxwin1.inl
ascii	73	0x00243D08	-	file	-	G:\FromInet\Include\boost_1_65_1\boost/exception/detail/exception_ptr.hpp
ascii	61	0x00244CB0	-	file	-	G:\Doc\My work (C++)_New 2018\Encryption\Release\encrypt.pdb
unicode	55	0x0E76D29E	-	file	-	af:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\viewcore.cpp
unicode	54	0x0E74DEDE	-	file	-	@f:\dd\vctools\vc7libs\ship\atlmfc\include\afxwin2.inl
unicode	54	0x0E751160	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\winctrl2.cpp
unicode	54	0x0E75B6B0	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\filecore.cpp
unicode	54	0x0E76BE70	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\oledrop2.cpp
unicode	54	0x0E771168	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\oleipfrm.cpp
unicode	53	0x0E74FD88	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\appcore.cpp
unicode	53	0x0E751460	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\auxdata.cpp
unicode	53	0x0E761EB6	-	file	-	af:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\winfrm.cpp
unicode	53	0x0E76C2F8	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\array_s.cpp
unicode	53	0x0E771858	-	file	-	f:\dd\vctools\vc7libs\ship\atlmfc\src\mfc\olestrm.cpp
ascii	35	0x00241D80	-	file	-	!!!DECRYPTION__KEYPASS__INFO!!!.txt
unicode	31	0x0E793030	-	file	-	C:\Windows\System32\rdpclip.exe

   a. **Afxwin1.inl -** This is a header file that is part of the MFC (Microsoft Foundation Classes) library in C++. It contains functions related to Windows UI programming. This hints that the malware might have a GUI interface.
   b. **Exception_ptr.hpp -** This seems like a Boost C++ library related header file used for exception handling related functions. Boost libraries cover a wide range of areas, including algorithms, containers, concurrency, cryptography, file systems, graphics, math, networking, serialization, etc. The path string for this file points to the G drive which is highly suspicious.
   c. **Encrypt.pdb** – pdb file extension is used in Microsoft Visual C++ to indicate a Program Database file. This might be the malware's program database. But since the path is so unusual and points to the G drive, the sample probably will never use it.
   d. **rdpclip.exe** – It is a Windows system process that runs in the background and is responsible for managing the clipboard functionality in Remote Desktop sessions. This hints that the malware might be using Remote Desktop.
   e. **DECRYPTION_KEYPASS_INFO.txt** – This is the name of the ransom note file the malware drops in each folder.
   f. **delself.bat** – The malware might be using this file to delete itself.



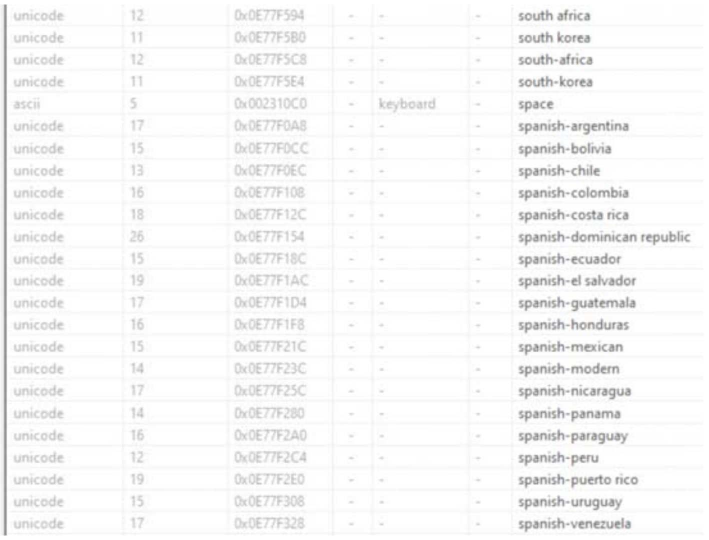
ascii	11	0x0023FAB8	-	file	-	delself.bat

8. **C Runtime error strings**: The sample contains C runtime error codes which tells us that it was probably written in C/C++.



unicode	37	0x0E7789FC	-	-	-	R6008- not enough space for arguments
unicode	69	0x0E778A5A	-	-	-	R6009- not enough space for environmentR6010- abort() has been called
unicode	79	0x0E778AFA	-	-	-	R6016- not enough space for thread dataR6017- unexpected multithread lock error
unicode	28	0x0E778BAC	-	-	-	R6018- unexpected heap error
unicode	36	0x0E778BF4	-	-	-	R6019- unable to open console device
unicode	48	0x0E778C4C	-	-	-	R6024- not enough space for _onexit/atexit table
unicode	33	0x0E778CBC	-	-	-	R6025- pure virtual function call
unicode	48	0x0E778D0C	-	-	-	R6026- not enough space for stdio initialization
unicode	48	0x0E778D7C	-	-	-	R6027- not enough space for lowio initialization
unicode	32	0x0E778DEC	-	-	-	R6028- unable to initialize heap
unicode	26	0x0E778E38	-	-	-	R6030- CRT not initialized
unicode	93	0x0E778E7E	-	-	-	R6031- Attempt to initialize the CRT more than once.This indicates a bug in your application.
unicode	46	0x0E778F44	-	-	-	R6032- not enough space for locale information
unicode	241	0x0E778FAE	-	-	-	R6033- Attempt to use MSIL code from this assembly during native code initializationThis indicates ...
unicode	46	0x0E77919C	-	-	-	R6034- inconsistent onexit begin-end variables

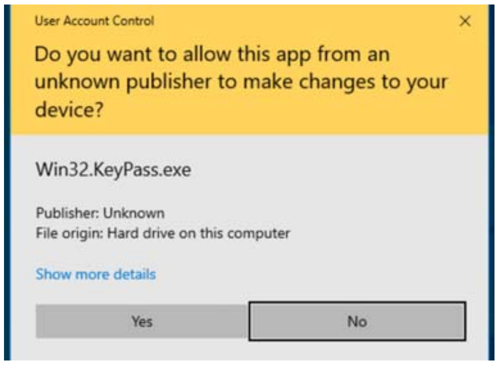
9. **Country and language related strings:** The sample might be checking the host machine's region and default language and might be performing different set of actions based on the region and language values it obtains.



unicode	12	0x0E77F594	-	-	-	south africa
unicode	11	0x0E77F5B0	-	-	-	south korea
unicode	12	0x0E77F5C8	-	-	-	south-africa
unicode	11	0x0E77F5E4	-	-	-	south-korea
ascii	5	0x002310C0	-	keyboard	-	space
unicode	17	0x0E77F0A8	-	-	-	spanish-argentina
unicode	15	0x0E77F0CC	-	-	-	spanish-bolivia
unicode	13	0x0E77F0EC	-	-	-	spanish-chile
unicode	16	0x0E77F108	-	-	-	spanish-colombia
unicode	18	0x0E77F12C	-	-	-	spanish-costa rica
unicode	26	0x0E77F154	-	-	-	spanish-dominican republic
unicode	15	0x0E77F18C	-	-	-	spanish-ecuador
unicode	19	0x0E77F1AC	-	-	-	spanish-el salvador
unicode	17	0x0E77F1D4	-	-	-	spanish-guatemala
unicode	16	0x0E77F1F8	-	-	-	spanish-honduras
unicode	15	0x0E77F21C	-	-	-	spanish-mexican
unicode	14	0x0E77F23C	-	-	-	spanish-modern
unicode	17	0x0E77F25C	-	-	-	spanish-nicaragua
unicode	14	0x0E77F280	-	-	-	spanish-panama
unicode	16	0x0E77F2A0	-	-	-	spanish-paraguay
unicode	12	0x0E77F2C4	-	-	-	spanish-peru
unicode	19	0x0E77F2E0	-	-	-	spanish-puerto rico
unicode	15	0x0E77F308	-	-	-	spanish-uruguay
unicode	17	0x0E77F328	-	-	-	spanish-venezuela

Dynamic Analysis

After running the malware sample, a privilege escalation dialog box pops up.



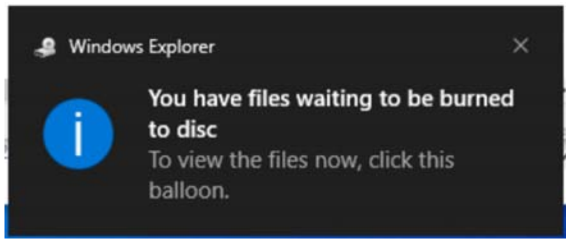
User Account Control

Do you want to allow this app from an unknown publisher to make changes to your device?

Win32.KeyPass.exe

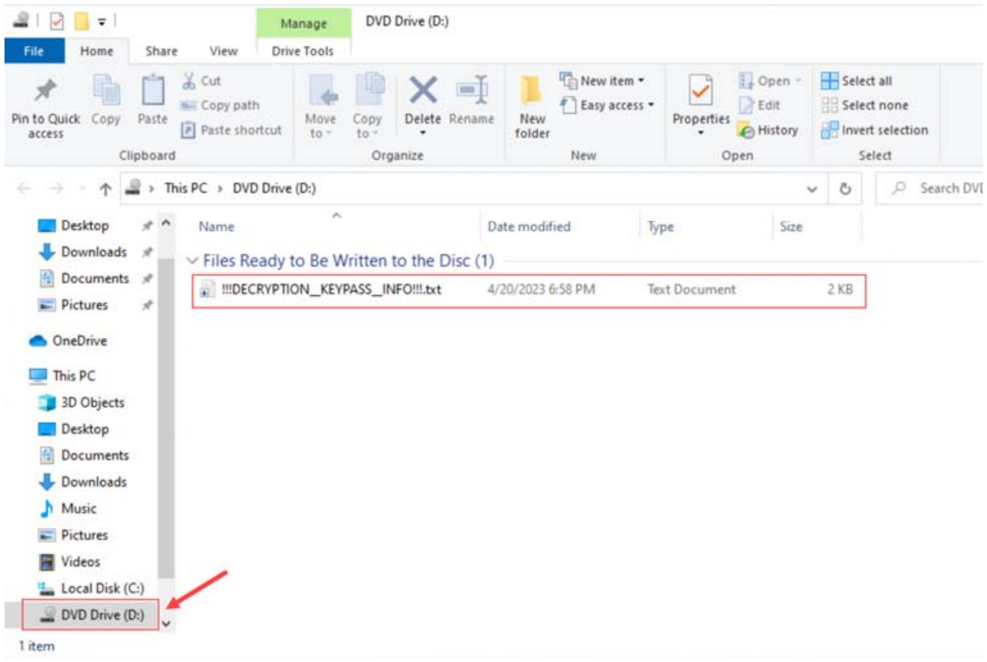
Publisher: Unknown
File origin: Hard drive on this computer

A few seconds later, a notification appears saying there are files waiting to be burned to disk. The File Explorer mounts a D: drive and shows the ransom note file waiting to be burned to disk.


Windows Explorer

You have files waiting to be burned to disc

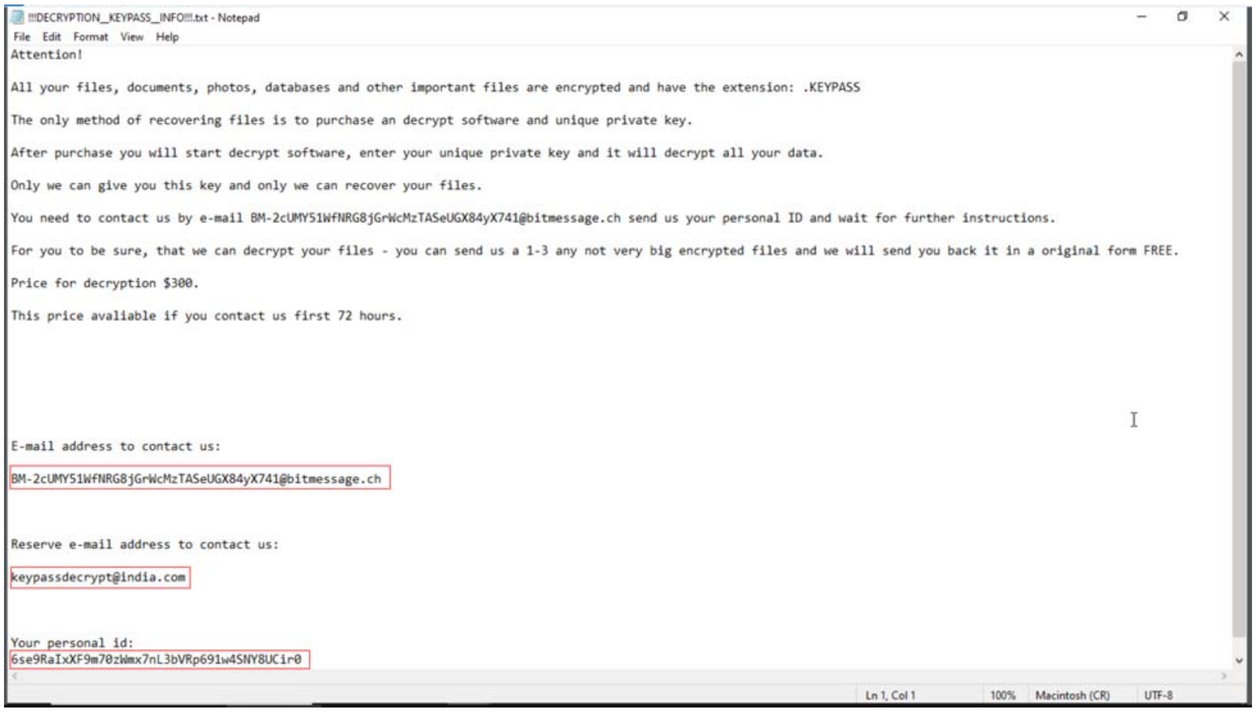
To view the files now, click this balloon.



Name	Date modified	Type	Size
Files Ready to Be Written to the Disc (1)			
!!!DECRYPTION_KEYPASS_INFO!!!.txt	4/20/2023 6:58 PM	Text Document	2 KB

Within the next few seconds, the desktop background turns black and all the files get encrypted and get renamed with a .KEYPASS extension.

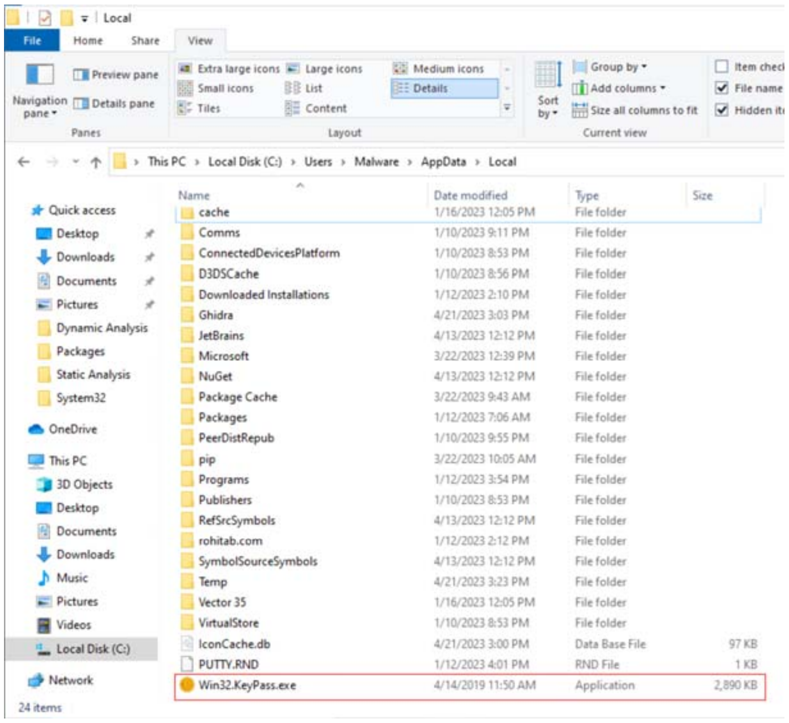
Ransom note file gets dropped in every folder. The ransom note file contains two email address to contact, one of them being keypassdecrypt@india.com. It also contains the user's personal ID which could be the user's private key. For every run of the malware this key remained the same. It could be because the malware isn't being able to contact its C&C server.


!!!DECRYPTION__KEYPASS__INFO!!!.txt - Notepad
File Edit Format View Help
Ln 1, Col 1
100%
Macintosh (CR)
UTF-8

```
Attention!

All your files, documents, photos, databases and other important files are encrypted and have the extension: .KEYPASS

The only method of recovering files is to purchase an decrypt software and unique private key.

After purchase you will start decrypt software, enter your unique private key and it will decrypt all your data.

Only we can give you this key and only we can recover your files.

You need to contact us by e-mail BM-2cUMY51WfNRG8jGrWcMzTASeUGX84yX741@bitmessage.ch send us your personal ID and wait for further instructions.

For you to be sure, that we can decrypt your files - you can send us a 1-3 any not very big encrypted files and we will send you back it in a original form FREE.

Price for decryption $300.

This price avaliable if you contact us first 72 hours.




E-mail address to contact us:

BM-2cUMY51WfNRG8jGrWcMzTASeUGX84yX741@bitmessage.ch


Reserve e-mail address to contact us:

keypassdecrypt@india.com


Your personal id:
6se9RaIxXF9m70zWmx7nL3bVRp691w4SNY8UCir0
```

Figure 16 Ransom note

- It can be observed that the malware does not kill processes that are currently running.
- It also doesn't encrypt the executables of currently running processes.
- The files in **C:\Windows** folder aren't encrypted by the program.
- If the file is renamed with .KEYPASS extension before the malware is run, the malware skips over it and doesn't encrypt it. Renaming the files with .KEYPASS extension before running the malware and then renaming them back to their original extension can be used to beat the malware's encryption.
- Malware deletes itself from the location where its file was originally present. It copies itself to **C:\Users\<username>\AppData\Local** and executes from there. Once it finishes execution, it deletes itself from everywhere.

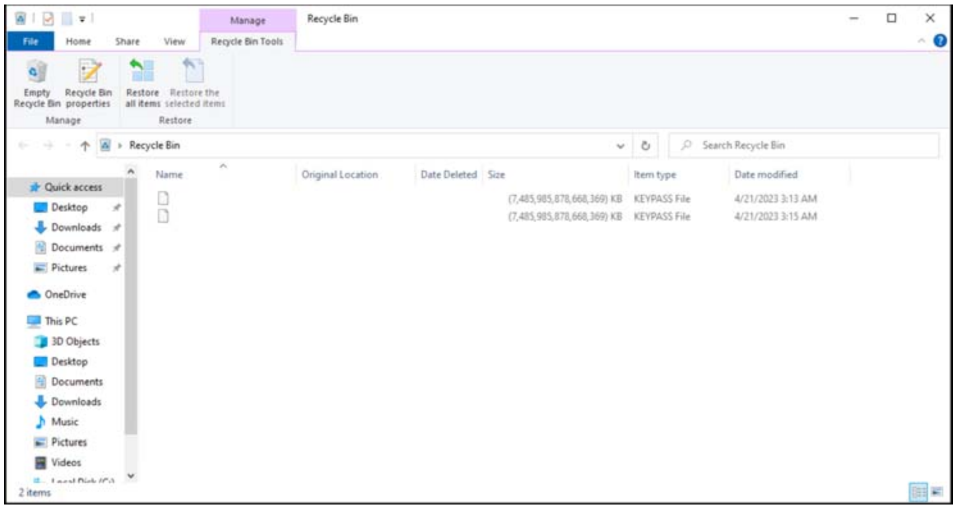

This PC › Local Disk (C:) › Users › Malware › AppData › Local

Name	Date modified	Type	Size
cache	1/16/2023 12:05 PM	File folder	
Comms	1/10/2023 9:11 PM	File folder	
ConnectedDevicesPlatform	1/10/2023 8:53 PM	File folder	
D3DSCache	1/10/2023 8:56 PM	File folder	
Downloaded Installations	1/12/2023 2:10 PM	File folder	
Ghidra	4/21/2023 3:03 PM	File folder	
JetBrains	4/13/2023 12:12 PM	File folder	
Microsoft	3/22/2023 12:39 PM	File folder	
NuGet	4/13/2023 12:12 PM	File folder	
Package Cache	3/22/2023 9:43 AM	File folder	
Packages	1/12/2023 7:06 AM	File folder	
PeerDistRepub	1/10/2023 9:55 PM	File folder	
pip	3/22/2023 10:05 AM	File folder	
Programs	1/12/2023 3:54 PM	File folder	
Publishers	1/10/2023 8:53 PM	File folder	
RefSrcSymbols	4/13/2023 12:12 PM	File folder	
rohitab.com	1/12/2023 2:12 PM	File folder	
SymbolSourceSymbols	4/13/2023 12:12 PM	File folder	
Temp	4/21/2023 3:23 PM	File folder	
Vector 35	1/16/2023 12:05 PM	File folder	
VirtualStore	1/10/2023 8:53 PM	File folder	
IconCache.db	4/21/2023 3:00 PM	Data Base File	97 KB
PUTTY.RND	1/12/2023 4:01 PM	RND File	1 KB
Win32.KeyPass.exe	4/14/2019 11:50 AM	Application	2,890 KB

24 items

Figure 17 WinKeyPass's Exe file copied to AppData/Local

- The sample modifies files present in the recycle bin too.

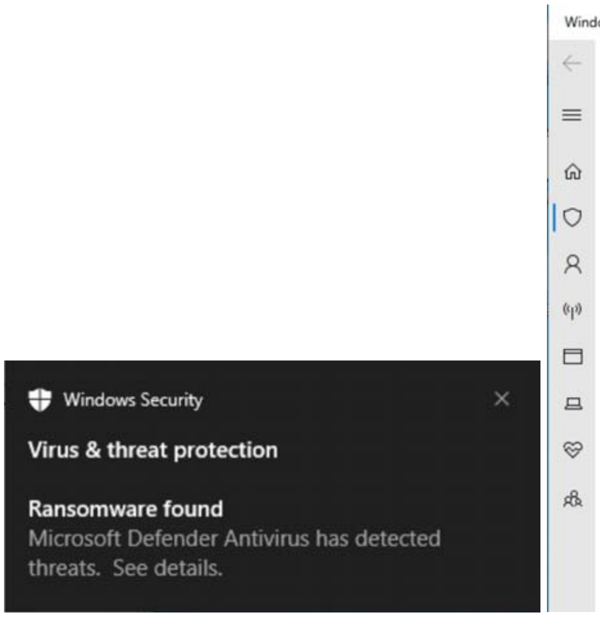

Recycle Bin

Name	Original Location	Date Deleted	Size	Item type	Date modified
			(7,485,985,878,668,369) KB	KEYPASS File	4/21/2023 3:13 AM
			(7,485,985,878,668,369) KB	KEYPASS File	4/21/2023 3:15 AM

2 items

Figure 18 Files in recycle bin modified by the sample

- Windows reports that it has found a ransomware on the machine.



Windows Security

Virus & threat protection

Ransomware found

Microsoft Defender Antivirus has detected threats. See details.

Windows Security

Virus & threat protection

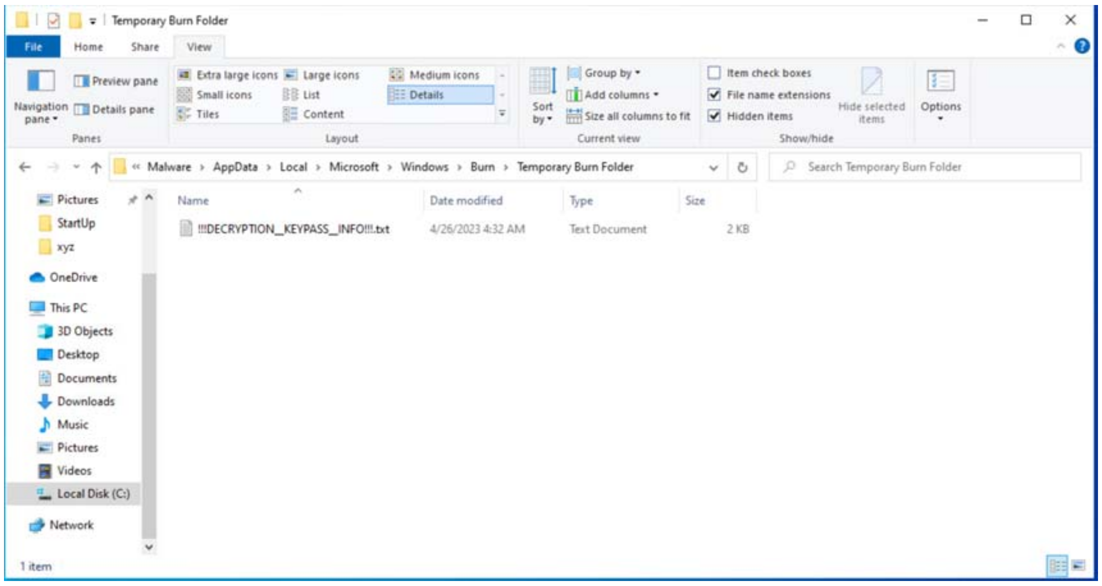
Protection for your device against threats.

Current threats

Threats found. Start the recommended actions.

Ransom:MacOS/Filecoder 4/15/2023 10:29 PM (Active)	Severe

- The “Files waiting to be burned” message appears frequently as the malware drops a ransom note in C:\Users\<User>\AppData\Local\Microsoft\Windows\Burn\Temporary Burn Folder. When a user tries to burn a file to disk, it first goes to the temporary burn folder. The OS periodically checks this folder to see if any files are present in it. If there are, it displays the “Files waiting to be burned” message and writes the files to disk. Since the malware adds a file to this folder, the OS thinks there is a file to be burned to disk.



Temporary Burn Folder

« Malware › AppData › Local › Microsoft › Windows › Burn › Temporary Burn Folder

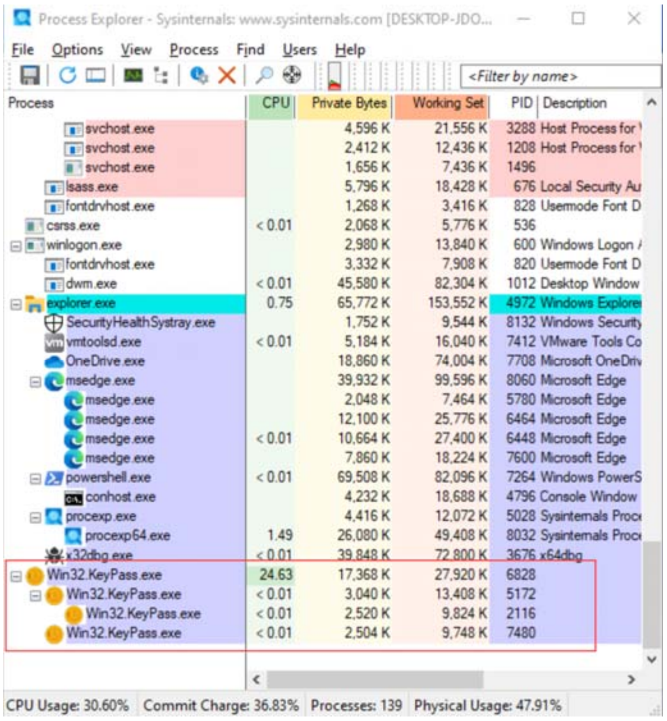
Name	Date modified	Type	Size
!!!DECRYPTION__KEYPASS__INFO!!!.txt	4/26/2023 4:32 AM	Text Document	2 KB

1 item

Figure 19 Ransom note file in Temporary Burn Folder responsible for “Files waiting to be burned” message

Process Explorer

- Process Explorer shows that the malware creates multiple sub processes and sub-sub processes.



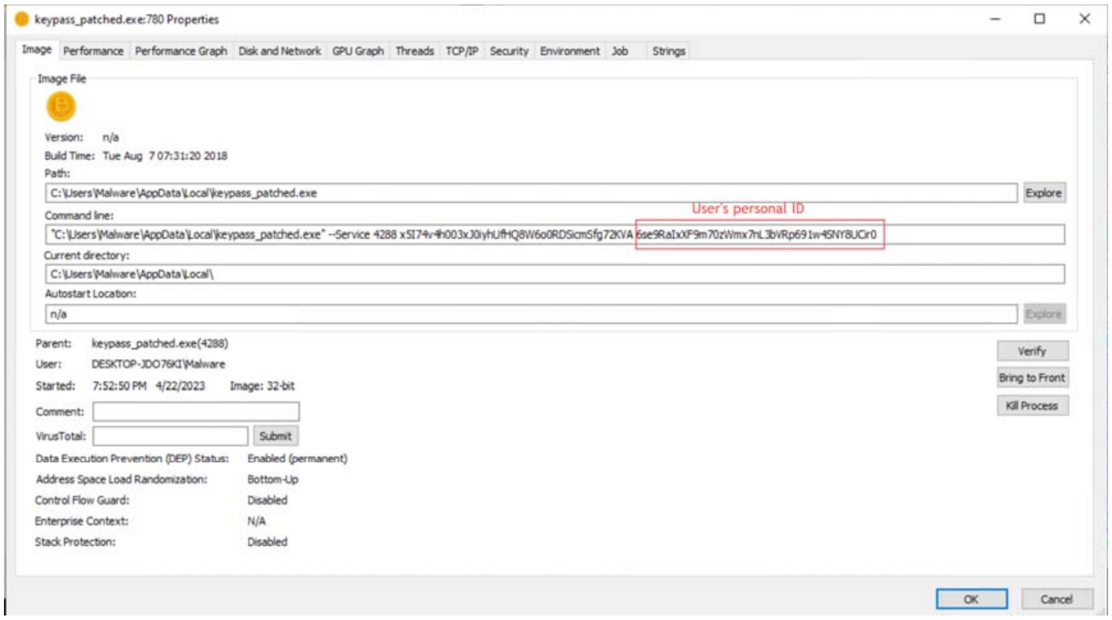
Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-JDO...

Process	CPU	Private Bytes	Working Set	PID	Description
svchost.exe		4,596 K	21,556 K	3288	Host Process for
svchost.exe		2,412 K	12,436 K	1208	Host Process for
svchost.exe		1,656 K	7,436 K	1496	
lsass.exe		5,796 K	18,428 K	676	Local Security Au
fontdrvhost.exe		1,268 K	3,416 K	828	Usermode Font D
csrss.exe	< 0.01	2,068 K	5,776 K	536	
winlogon.exe		2,980 K	13,840 K	600	Windows Logon
fontdrvhost.exe		3,332 K	7,908 K	820	Usermode Font D
dwm.exe	< 0.01	45,580 K	82,304 K	1012	Desktop Window
explorer.exe	0.75	65,772 K	153,552 K	4972	Windows Explorer
SecurityHealthSystray.exe		1,752 K	9,544 K	8132	Windows Security
vmtoolsd.exe	< 0.01	5,184 K	16,040 K	7412	VMware Tools Co
OneDrive.exe		18,860 K	74,004 K	7708	Microsoft OneDriv
msedge.exe		39,932 K	99,596 K	8060	Microsoft Edge
msedge.exe		2,048 K	7,464 K	5780	Microsoft Edge
msedge.exe		12,100 K	25,776 K	6464	Microsoft Edge
msedge.exe	< 0.01	10,664 K	27,400 K	6448	Microsoft Edge
msedge.exe		7,860 K	18,224 K	7600	Microsoft Edge
powershell.exe	< 0.01	69,508 K	82,096 K	7264	Windows PowerS
conhost.exe		4,232 K	18,688 K	4796	Console Window
procexp.exe		4,416 K	12,072 K	5028	Sysinternals Proc
procexp64.exe	1.49	26,080 K	49,408 K	8032	Sysinternals Proc
x32dbg.exe	< 0.01	39,848 K	72,800 K	3676	x64dbg
Win32.KeyPass.exe	24.63	17,368 K	27,920 K	6828	
Win32.KeyPass.exe	< 0.01	3,040 K	13,408 K	5172	
Win32.KeyPass.exe	< 0.01	2,520 K	9,824 K	2116	
Win32.KeyPass.exe	< 0.01	2,504 K	9,748 K	7480	

CPU Usage: 30.60% | Commit Charge: 36.83% | Processes: 139 | Physical Usage: 47.91%

Figure 20 Malware creates multiple processes

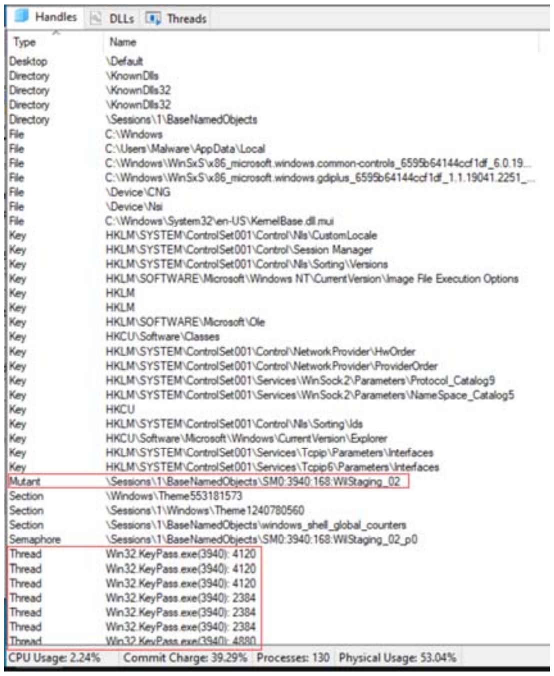
- Checking the properties of one of the subprocesses the malware generates shows that the user's personal ID is passed as a command line parameter to the process while starting it. The main process doesn't take any command line parameters.



```
Version:    n/a
Build Time: Tue Aug  7 07:31:20 2018
Path:
C:\Users\Malware\AppData\Local\keypass_patched.exe
Command line:
"C:\Users\Malware\AppData\Local\keypass_patched.exe" --Service 4288 x5I74v4h003xJ0iyhUfHQ8W6o0RDSicmSfg72KVA 6se9RaIxXF9m70zWmx7nL3bVRp691w4SNY8UCir0
Current directory:
C:\Users\Malware\AppData\Local\
Autostart Location:
n/a

Parent:   keypass_patched.exe(4288)
User:     DESKTOP-JDO76KI\Malware
Started:  7:52:50 PM  4/22/2023     Image: 32-bit
Comment:
VirusTotal:
Data Execution Prevention (DEP) Status:   Enabled (permanent)
Address Space Load Randomization:         Bottom-Up
Control Flow Guard:                       Disabled
Enterprise Context:                       N/A
Stack Protection:                         Disabled
```

Figure 21 Subprocesses take command line parameters

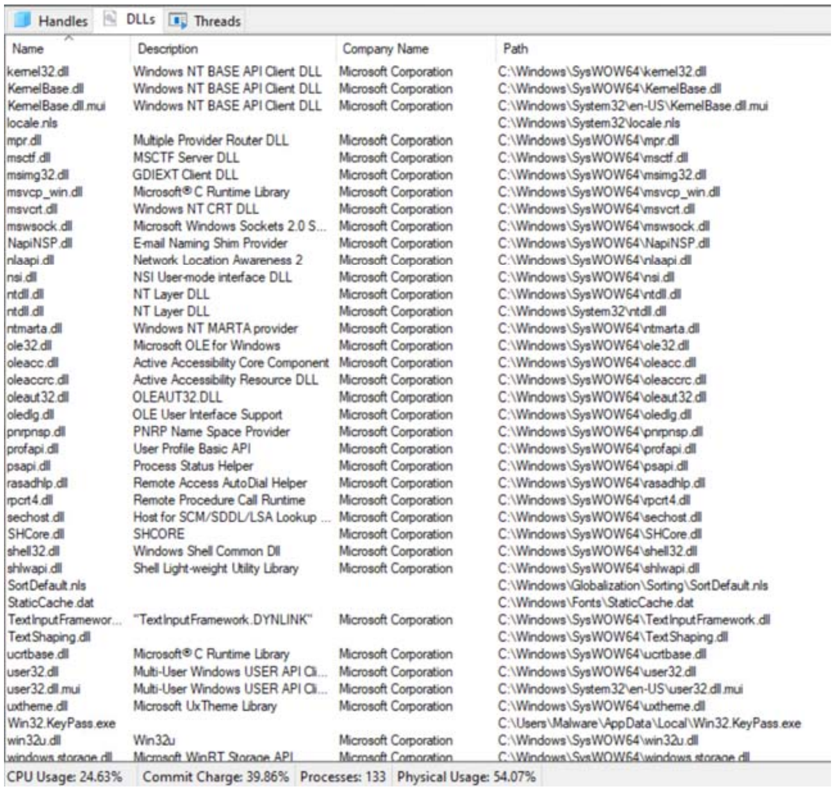
- The malware's processes use a mutant that is commonly used by most Windows processes. It can also be seen that the processes use multiple threads.



```
Type        Name
Desktop     \Default
Directory   \KnownDlls
Directory   \KnownDlls32
Directory   \KnownDlls32
Directory   \Sessions\1\BaseNamedObjects
File        C:\Windows
File        C:\Users\Malware\AppData\Local
File        C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.19...
File        C:\Windows\WinSxS\x86_microsoft.windows.gdiplus_6595b64144ccf1df_1.1.19041.2251_...
File        \Device\CNG
File        \Device\Nsi
File        C:\Windows\System32\en-US\KernelBase.dll.mui
Key         HKLM\SYSTEM\ControlSet001\Control\Nls\CustomLocale
Key         HKLM\SYSTEM\ControlSet001\Control\Session Manager
Key         HKLM\SYSTEM\ControlSet001\Control\Nls\Sorting\Versions
Key         HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
Key         HKLM
Key         HKLM
Key         HKLM\SOFTWARE\Microsoft\Ole
Key         HKCU\Software\Classes
Key         HKLM\SYSTEM\ControlSet001\Control\NetworkProvider\HwOrder
Key         HKLM\SYSTEM\ControlSet001\Control\NetworkProvider\ProviderOrder
Key         HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\Protocol_Catalog9
Key         HKLM\SYSTEM\ControlSet001\Services\WinSock2\Parameters\NameSpace_Catalog5
Key         HKCU
Key         HKLM\SYSTEM\ControlSet001\Control\Nls\Sorting\Ids
Key         HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer
Key         HKLM\SYSTEM\ControlSet001\Services\Tcpip\Parameters\Interfaces
Key         HKLM\SYSTEM\ControlSet001\Services\Tcpip6\Parameters\Interfaces
Mutant      \Sessions\1\BaseNamedObjects\SM0:3940:168:WilStaging_02
Section     \Windows\Theme553181573
Section     \Sessions\1\Windows\Theme1240780560
Section     \Sessions\1\BaseNamedObjects\windows_shell_global_counters
Semaphore   \Sessions\1\BaseNamedObjects\SM0:3940:168:WilStaging_02_p0
Thread      Win32.KeyPass.exe(3940): 4120
Thread      Win32.KeyPass.exe(3940): 4120
Thread      Win32.KeyPass.exe(3940): 4120
Thread      Win32.KeyPass.exe(3940): 2384
Thread      Win32.KeyPass.exe(3940): 2384
Thread      Win32.KeyPass.exe(3940): 2384
Thread      Win32.KeyPass.exe(3940): 4880
CPU Usage: 2.24%   Commit Charge: 39.29%   Processes: 130   Physical Usage: 53.04%
```

Figure 22 Threads and mutants used by the malware's processes

- The DLL list suggests that the malware might be doing some network activity, it uses COM (Component Object Model) and it might be manipulating GUI elements on the screen.



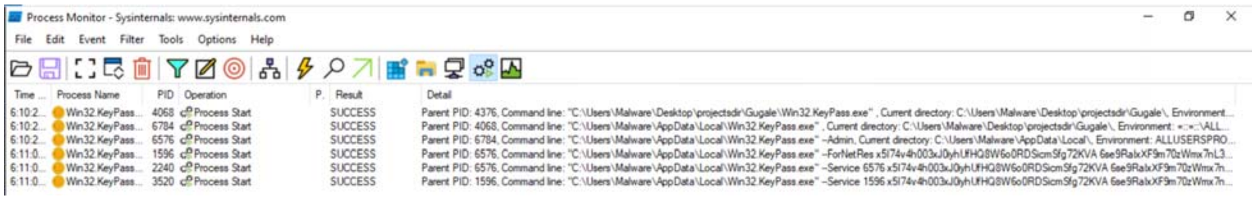
Name	Description	Company Name	Path
kernel32.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\Windows\SysWOW64\kernel32.dll
KernelBase.dll	Windows NT BASE API Client DLL	Microsoft Corporation	C:\Windows\SysWOW64\KernelBase.dll
KernelBase.dll.mui	Windows NT BASE API Client DLL	Microsoft Corporation	C:\Windows\System32\en-US\KernelBase.dll.mui
locale.nls			C:\Windows\System32\locale.nls
mpr.dll	Multiple Provider Router DLL	Microsoft Corporation	C:\Windows\SysWOW64\mpr.dll
msctf.dll	MSCTF Server DLL	Microsoft Corporation	C:\Windows\SysWOW64\msctf.dll
msimg32.dll	GDIEXT Client DLL	Microsoft Corporation	C:\Windows\SysWOW64\msimg32.dll
msvcp_win.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:\Windows\SysWOW64\msvcp_win.dll
msvcrt.dll	Windows NT CRT DLL	Microsoft Corporation	C:\Windows\SysWOW64\msvcrt.dll
mswsock.dll	Microsoft Windows Sockets 2.0 S...	Microsoft Corporation	C:\Windows\SysWOW64\mswsock.dll
NapiNSP.dll	E-mail Naming Shim Provider	Microsoft Corporation	C:\Windows\SysWOW64\NapiNSP.dll
nlaapi.dll	Network Location Awareness 2	Microsoft Corporation	C:\Windows\SysWOW64\nlaapi.dll
nsi.dll	NSI User-mode interface DLL	Microsoft Corporation	C:\Windows\SysWOW64\nsi.dll
ntdll.dll	NT Layer DLL	Microsoft Corporation	C:\Windows\SysWOW64\ntdll.dll
ntdll.dll	NT Layer DLL	Microsoft Corporation	C:\Windows\System32\ntdll.dll
ntmarta.dll	Windows NT MARTA provider	Microsoft Corporation	C:\Windows\SysWOW64\ntmarta.dll
ole32.dll	Microsoft OLE for Windows	Microsoft Corporation	C:\Windows\SysWOW64\ole32.dll
oleacc.dll	Active Accessibility Core Component	Microsoft Corporation	C:\Windows\SysWOW64\oleacc.dll
oleaccrc.dll	Active Accessibility Resource DLL	Microsoft Corporation	C:\Windows\SysWOW64\oleaccrc.dll
oleaut32.dll	OLEAUT32.DLL	Microsoft Corporation	C:\Windows\SysWOW64\oleaut32.dll
oledlg.dll	OLE User Interface Support	Microsoft Corporation	C:\Windows\SysWOW64\oledlg.dll
pnrpnsp.dll	PNRP Name Space Provider	Microsoft Corporation	C:\Windows\SysWOW64\pnrpnsp.dll
profapi.dll	User Profile Basic API	Microsoft Corporation	C:\Windows\SysWOW64\profapi.dll
psapi.dll	Process Status Helper	Microsoft Corporation	C:\Windows\SysWOW64\psapi.dll
rasadhlp.dll	Remote Access AutoDial Helper	Microsoft Corporation	C:\Windows\SysWOW64\rasadhlp.dll
rpcrt4.dll	Remote Procedure Call Runtime	Microsoft Corporation	C:\Windows\SysWOW64\rpcrt4.dll
sechost.dll	Host for SCM/SDDL/LSA Lookup ...	Microsoft Corporation	C:\Windows\SysWOW64\sechost.dll
SHCore.dll	SHCORE	Microsoft Corporation	C:\Windows\SysWOW64\SHCore.dll
shell32.dll	Windows Shell Common Dll	Microsoft Corporation	C:\Windows\SysWOW64\shell32.dll
shlwapi.dll	Shell Light-weight Utility Library	Microsoft Corporation	C:\Windows\SysWOW64\shlwapi.dll
SortDefault.nls			C:\Windows\Globalization\Sorting\SortDefault.nls
StaticCache.dat			C:\Windows\Fonts\StaticCache.dat
TextInputFramewor...	"TextInputFramework.DYNLINK"	Microsoft Corporation	C:\Windows\SysWOW64\TextInputFramework.dll
TextShaping.dll			C:\Windows\SysWOW64\TextShaping.dll
ucrtbase.dll	Microsoft® C Runtime Library	Microsoft Corporation	C:\Windows\SysWOW64\ucrtbase.dll
user32.dll	Multi-User Windows USER API Cli...	Microsoft Corporation	C:\Windows\SysWOW64\user32.dll
user32.dll.mui	Multi-User Windows USER API Cli...	Microsoft Corporation	C:\Windows\System32\en-US\user32.dll.mui
uxtheme.dll	Microsoft UxTheme Library	Microsoft Corporation	C:\Windows\SysWOW64\uxtheme.dll
Win32.KeyPass.exe			C:\Users\Malware\AppData\Local\Win32.KeyPass.exe
win32u.dll	Win32u	Microsoft Corporation	C:\Windows\SysWOW64\win32u.dll
windows.storage.dll	Microsoft WinRT Storage API	Microsoft Corporation	C:\Windows\SysWOW64\windows.storage.dll

CPU Usage: 24.63% | Commit Charge: 39.86% | Processes: 133 | Physical Usage: 54.07%

Figure 23 DLLs imported by the malware processes

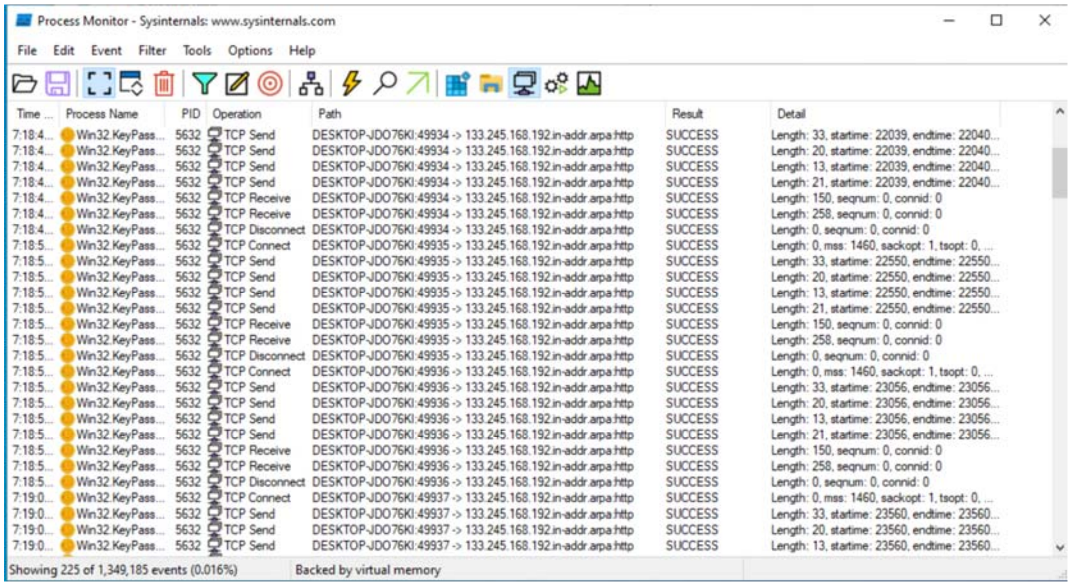
Process Monitor

- The sample creates multiple processes with various parameters like "—Admin", "—Service" and "—ForNetRes".



```
Time ...  Process Name       PID   Operation       P.  Result    Detail
6:10:2... Win32.KeyPass...   4068  Process Start       SUCCESS   Parent PID: 4376, Command line: "C:\Users\Malware\Desktop\projectsdir\Gugale\Win32.KeyPass.exe" , Current directory: C:\Users\Malware\Desktop\projectsdir\Gugale\, Environment...
6:10:2... Win32.KeyPass...   6784  Process Start       SUCCESS   Parent PID: 4068, Command line: "C:\Users\Malware\AppData\Local\Win32.KeyPass.exe" , Current directory: C:\Users\Malware\Desktop\projectsdir\Gugale\, Environment: =::=::\ALL...
6:10:2... Win32.KeyPass...   6576  Process Start       SUCCESS   Parent PID: 6784, Command line: "C:\Users\Malware\AppData\Local\Win32.KeyPass.exe" --Admin, Current directory: C:\Users\Malware\AppData\Local\, Environment: ALLUSERSPRO...
6:11:0... Win32.KeyPass...   1596  Process Start       SUCCESS   Parent PID: 6576, Command line: "C:\Users\Malware\AppData\Local\Win32.KeyPass.exe" --ForNetRes x5I74v4h003xJ0yhUfHQ8W6o0RDSicmSfg72KVA 6se9RaIxXF9m70zWmx7nL3...
6:11:0... Win32.KeyPass...   2240  Process Start       SUCCESS   Parent PID: 6576, Command line: "C:\Users\Malware\AppData\Local\Win32.KeyPass.exe" --Service 6576 x5I74v4h003xJ0yhUfHQ8W6o0RDSicmSfg72KVA 6se9RaIxXF9m70zWmx7n...
6:11:0... Win32.KeyPass...   3520  Process Start       SUCCESS   Parent PID: 1596, Command line: "C:\Users\Malware\AppData\Local\Win32.KeyPass.exe" --Service 1596 x5I74v4h003xJ0yhUfHQ8W6o0RDSicmSfg72KVA 6se9RaIxXF9m70zWmx7n...
```

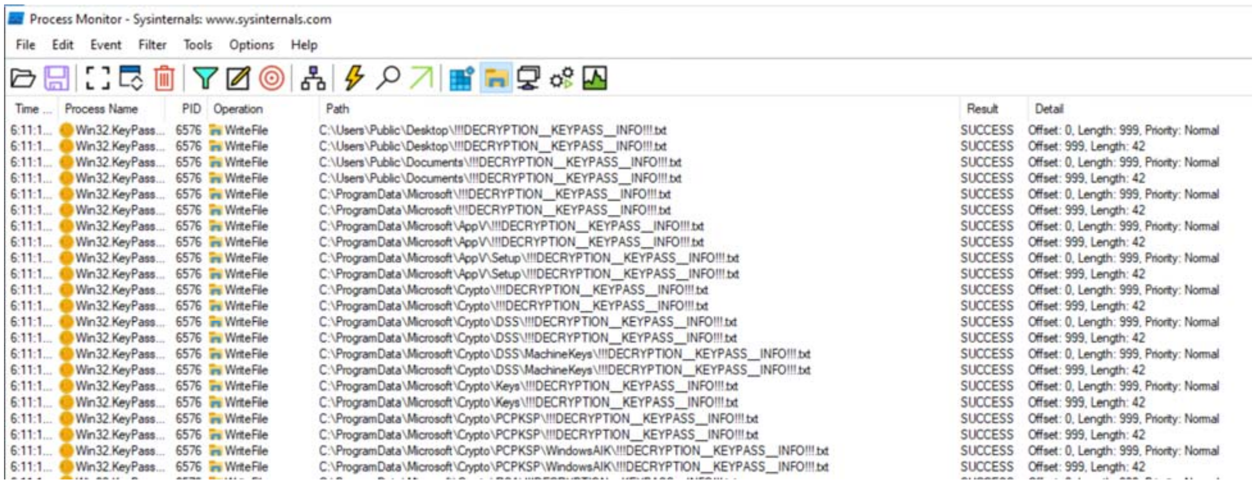
- The sample's HTTP communication can be seen in the ProcMon logs as TCP operations.



```
Process Monitor - Sysinternals: www.sysinternals.com
File  Edit  Event  Filter  Tools  Options  Help

Time ...  Process Name    PID  Operation       Path                                                   Result   Detail
7:18:4...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49934 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 33, startime: 22039, endtime: 22040...
7:18:4...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49934 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 20, startime: 22039, endtime: 22040...
7:18:4...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49934 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 13, startime: 22039, endtime: 22040...
7:18:4...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49934 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 21, startime: 22039, endtime: 22040...
7:18:4...  Win32.KeyPass...  5632  TCP Receive     DESKTOP-JDO76KI:49934 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 150, seqnum: 0, connid: 0
7:18:4...  Win32.KeyPass...  5632  TCP Receive     DESKTOP-JDO76KI:49934 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 258, seqnum: 0, connid: 0
7:18:4...  Win32.KeyPass...  5632  TCP Disconnect  DESKTOP-JDO76KI:49934 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 0, seqnum: 0, connid: 0
7:18:5...  Win32.KeyPass...  5632  TCP Connect     DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 0, mss: 1460, sackopt: 1, tsopt: 0, ...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 33, startime: 22550, endtime: 22550...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 20, startime: 22550, endtime: 22550...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 13, startime: 22550, endtime: 22550...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 21, startime: 22550, endtime: 22550...
7:18:5...  Win32.KeyPass...  5632  TCP Receive     DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 150, seqnum: 0, connid: 0
7:18:5...  Win32.KeyPass...  5632  TCP Receive     DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 258, seqnum: 0, connid: 0
7:18:5...  Win32.KeyPass...  5632  TCP Disconnect  DESKTOP-JDO76KI:49935 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 0, seqnum: 0, connid: 0
7:18:5...  Win32.KeyPass...  5632  TCP Connect     DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 0, mss: 1460, sackopt: 1, tsopt: 0, ...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 33, startime: 23056, endtime: 23056...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 20, startime: 23056, endtime: 23056...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 13, startime: 23056, endtime: 23056...
7:18:5...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 21, startime: 23056, endtime: 23056...
7:18:5...  Win32.KeyPass...  5632  TCP Receive     DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 150, seqnum: 0, connid: 0
7:18:5...  Win32.KeyPass...  5632  TCP Receive     DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 258, seqnum: 0, connid: 0
7:18:5...  Win32.KeyPass...  5632  TCP Disconnect  DESKTOP-JDO76KI:49936 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 0, seqnum: 0, connid: 0
7:19:0...  Win32.KeyPass...  5632  TCP Connect     DESKTOP-JDO76KI:49937 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 0, mss: 1460, sackopt: 1, tsopt: 0, ...
7:19:0...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49937 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 33, startime: 23560, endtime: 23560...
7:19:0...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49937 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 20, startime: 23560, endtime: 23560...
7:19:0...  Win32.KeyPass...  5632  TCP Send        DESKTOP-JDO76KI:49937 -> 133.245.168.192.in-addr.arpa:http  SUCCESS  Length: 13, startime: 23560, endtime: 23560...
Showing 225 of 1,349,185 events (0.016%)    Backed by virtual memory
```

Figure 24 The sample's network activity

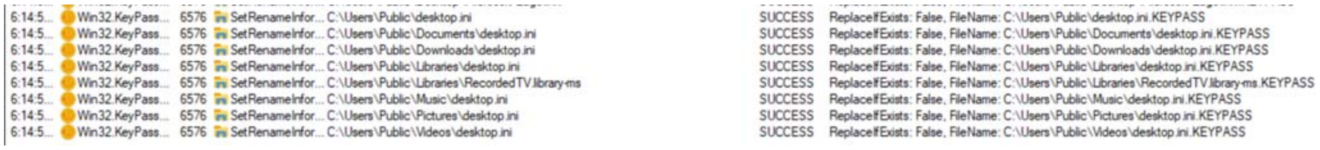
- The sample drops ransom notes in all folders using WriteFile.



```
Process Monitor - Sysinternals: www.sysinternals.com
File  Edit  Event  Filter  Tools  Options  Help

Time ...  Process Name      PID   Operation  Path                                                                          Result   Detail
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\Users\Public\Desktop\!!!DECRYPTION__KEYPASS__INFO!!!.txt                        SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\Users\Public\Desktop\!!!DECRYPTION__KEYPASS__INFO!!!.txt                        SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\Users\Public\Documents\!!!DECRYPTION__KEYPASS__INFO!!!.txt                      SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\Users\Public\Documents\!!!DECRYPTION__KEYPASS__INFO!!!.txt                      SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\!!!DECRYPTION__KEYPASS__INFO!!!.txt                       SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\!!!DECRYPTION__KEYPASS__INFO!!!.txt                       SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\AppV\!!!DECRYPTION__KEYPASS__INFO!!!.txt                  SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\AppV\!!!DECRYPTION__KEYPASS__INFO!!!.txt                  SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\AppV\Setup\!!!DECRYPTION__KEYPASS__INFO!!!.txt            SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\AppV\Setup\!!!DECRYPTION__KEYPASS__INFO!!!.txt            SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\!!!DECRYPTION__KEYPASS__INFO!!!.txt                SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\!!!DECRYPTION__KEYPASS__INFO!!!.txt                SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\DSS\!!!DECRYPTION__KEYPASS__INFO!!!.txt            SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\DSS\!!!DECRYPTION__KEYPASS__INFO!!!.txt            SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\DSS\MachineKeys\!!!DECRYPTION__KEYPASS__INFO!!!.txt SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\DSS\MachineKeys\!!!DECRYPTION__KEYPASS__INFO!!!.txt SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\Keys\!!!DECRYPTION__KEYPASS__INFO!!!.txt           SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\Keys\!!!DECRYPTION__KEYPASS__INFO!!!.txt           SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\PCPKSP\!!!DECRYPTION__KEYPASS__INFO!!!.txt         SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\PCPKSP\!!!DECRYPTION__KEYPASS__INFO!!!.txt         SUCCESS  Offset: 999, Length: 42
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\PCPKSP\WindowsAIK\!!!DECRYPTION__KEYPASS__INFO!!!.txt SUCCESS  Offset: 0, Length: 999, Priority: Normal
6:11:1...  Win32.KeyPass...  6576  WriteFile  C:\ProgramData\Microsoft\Crypto\PCPKSP\WindowsAIK\!!!DECRYPTION__KEYPASS__INFO!!!.txt SUCCESS  Offset: 999, Length: 42
```

Figure 25 Ransom note dropped with WriteFile

- Encrypted files are written back with WriteFile and renamed with the .KEYPASS extension using SetRenameInformation operation.

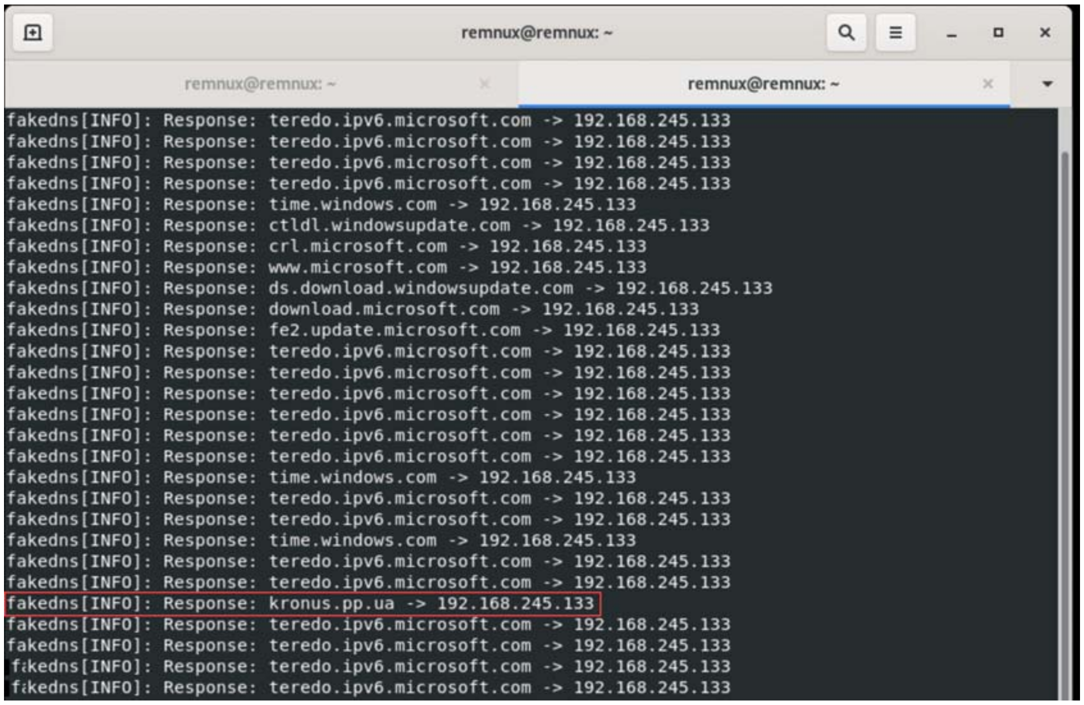


```
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\desktop.ini                      SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\desktop.ini.KEYPASS
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\Documents\desktop.ini            SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\Documents\desktop.ini.KEYPASS
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\Downloads\desktop.ini            SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\Downloads\desktop.ini.KEYPASS
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\Libraries\desktop.ini            SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\Libraries\desktop.ini.KEYPASS
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\Libraries\RecordedTV.library-ms  SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\Libraries\RecordedTV.library-ms.KEYPASS
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\Music\desktop.ini                SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\Music\desktop.ini.KEYPASS
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\Pictures\desktop.ini             SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\Pictures\desktop.ini.KEYPASS
6:14:5...  Win32.KeyPass...  6576  SetRenameInfor...  C:\Users\Public\Videos\desktop.ini               SUCCESS  ReplaceIfExists: False, FileName: C:\Users\Public\Videos\desktop.ini.KEYPASS
```

Figure 26 Renaming files after encryption

FakeDNS

- FakeDNS is a tool available on Remnux designed to intercept DNS resolution requests and respond by providing the requester with a preconfigured IP address. As can be seen from the image below, there are a lot of DNS requests for resolution of domains used by standard Microsoft services like time.windows.com and teredo.ipv6.microsoft.com.
- There is one suspicious DNS request in the logs – IP resolution request for **kronus.pp.ua.** The malware's CNC server might be located here.

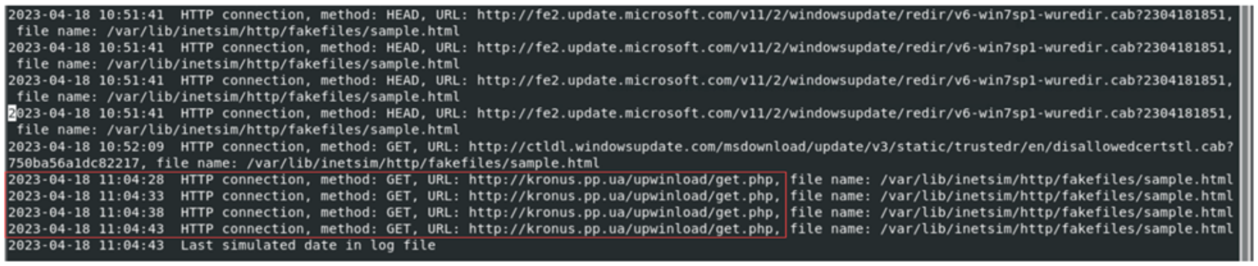


```
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: time.windows.com -> 192.168.245.133
fakedns[INFO]: Response: ctldl.windowsupdate.com -> 192.168.245.133
fakedns[INFO]: Response: crl.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: www.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: ds.download.windowsupdate.com -> 192.168.245.133
fakedns[INFO]: Response: download.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: fe2.update.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: time.windows.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: time.windows.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: kronus.pp.ua -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
fakedns[INFO]: Response: teredo.ipv6.microsoft.com -> 192.168.245.133
```

Figure 27 DNS request for kronus.pp.ua

InetSim

- InetSim is a command line utility used to simulate various internet services such as HTTP, DNS, FTP, and SMTP. InetSim creates a virtual network environment that emulates the behavior of various internet services, allowing users to test their applications and security tools in a safe and controlled manner.
- For the Inetsim logs, it can be seen that the malware sends multiple HTTP GET requests to the URL **http://kronus.pp.ua/upwinload/get.php**
- The C&C server of the sample could be located here and the malware might be sending the information it collects to the server.

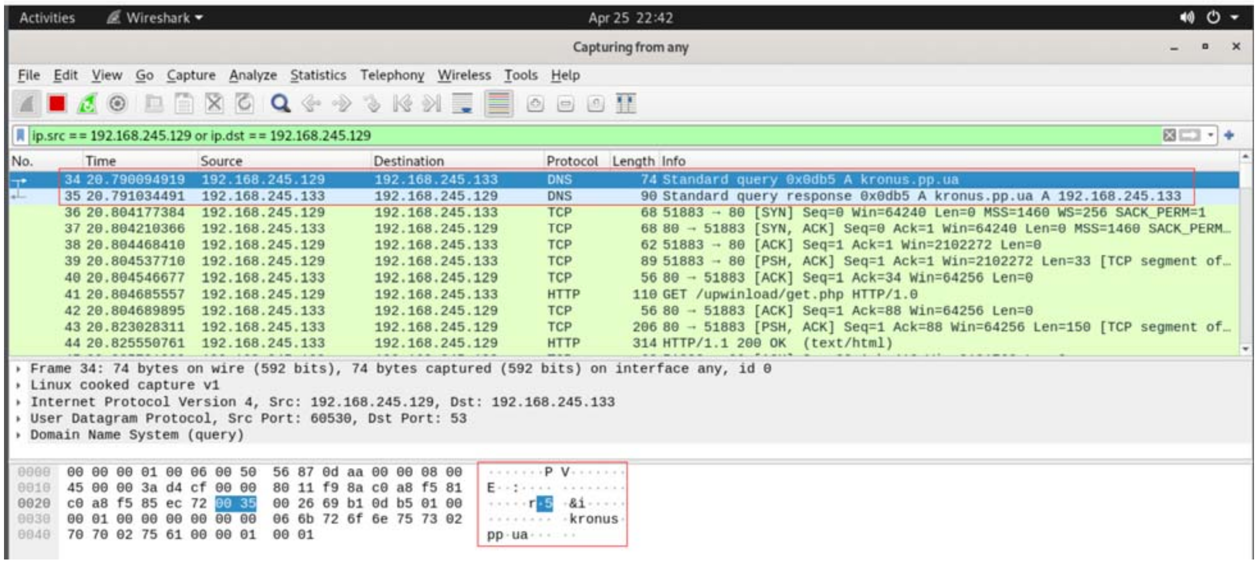


```
2023-04-18 10:51:41  HTTP connection, method: HEAD, URL: http://fe2.update.microsoft.com/v11/2/windowsupdate/redir/v6-win7sp1-wuredir.cab?2304181851,
 file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 10:51:41  HTTP connection, method: HEAD, URL: http://fe2.update.microsoft.com/v11/2/windowsupdate/redir/v6-win7sp1-wuredir.cab?2304181851,
 file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 10:51:41  HTTP connection, method: HEAD, URL: http://fe2.update.microsoft.com/v11/2/windowsupdate/redir/v6-win7sp1-wuredir.cab?2304181851,
 file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 10:51:41  HTTP connection, method: HEAD, URL: http://fe2.update.microsoft.com/v11/2/windowsupdate/redir/v6-win7sp1-wuredir.cab?2304181851,
 file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 10:52:09  HTTP connection, method: GET, URL: http://ctldl.windowsupdate.com/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?
750ba56a1dc82217, file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 11:04:28  HTTP connection, method: GET, URL: http://kronus.pp.ua/upwinload/get.php, file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 11:04:33  HTTP connection, method: GET, URL: http://kronus.pp.ua/upwinload/get.php, file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 11:04:38  HTTP connection, method: GET, URL: http://kronus.pp.ua/upwinload/get.php, file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 11:04:43  HTTP connection, method: GET, URL: http://kronus.pp.ua/upwinload/get.php, file name: /var/lib/inetsim/http/fakefiles/sample.html
2023-04-18 11:04:43  Last simulated date in log file
```

Figure 28 Suspicious GET requests

Wireshark

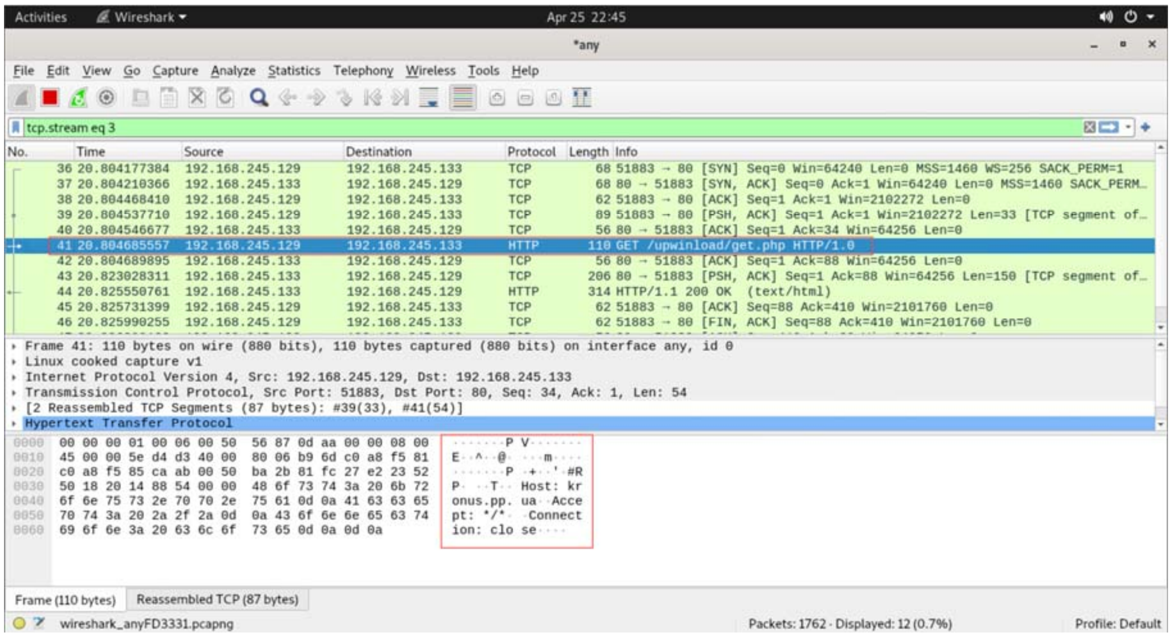
- Wireshark allows us to capture and analyze network traffic in real-time, and view the packets that are being transmitted over the network.
- Analyzing the malware sample with Wireshark shows a DNS request for kronus.pp.ua which is probably the malware's C&C server. This is the same domain name that was seen with FakeDNS.


Activities
Wireshark
Apr 25 22:42
Capturing from any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.src == 192.168.245.129 or ip.dst == 192.168.245.129

No.  Time          Source           Destination      Protocol Length Info
  34 20.790094919  192.168.245.129  192.168.245.133  DNS       74 Standard query 0x0db5 A kronus.pp.ua
  35 20.791034491  192.168.245.133  192.168.245.129  DNS       90 Standard query response 0x0db5 A kronus.pp.ua A 192.168.245.133
  36 20.804177384  192.168.245.129  192.168.245.133  TCP       68 51883 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  37 20.804210366  192.168.245.133  192.168.245.129  TCP       68 80 → 51883 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM…
  38 20.804468410  192.168.245.129  192.168.245.133  TCP       62 51883 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
  39 20.804537710  192.168.245.129  192.168.245.133  TCP       89 51883 → 80 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=33 [TCP segment of…
  40 20.804546677  192.168.245.133  192.168.245.129  TCP       56 80 → 51883 [ACK] Seq=1 Ack=34 Win=64256 Len=0
  41 20.804685557  192.168.245.129  192.168.245.133  HTTP     110 GET /upwinload/get.php HTTP/1.0
  42 20.804689895  192.168.245.133  192.168.245.129  TCP       56 80 → 51883 [ACK] Seq=1 Ack=88 Win=64256 Len=0
  43 20.823028311  192.168.245.133  192.168.245.129  TCP      206 80 → 51883 [PSH, ACK] Seq=1 Ack=88 Win=64256 Len=150 [TCP segment of…
  44 20.825550761  192.168.245.133  192.168.245.129  HTTP     314 HTTP/1.1 200 OK  (text/html)

› Frame 34: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface any, id 0
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 192.168.245.129, Dst: 192.168.245.133
› User Datagram Protocol, Src Port: 60530, Dst Port: 53
› Domain Name System (query)

0000  00 00 00 01 00 06 00 50  56 87 0d aa 00 00 08 00   ·······P V·······
0010  45 00 00 3a d4 cf 00 00  80 11 f9 8a c0 a8 f5 81   E··:···· ········
0020  c0 a8 f5 85 ec 72 00 35  00 26 69 b1 0d b5 01 00   ·····r·5 ·&i·····
0030  00 01 00 00 00 00 00 00  06 6b 72 6f 6e 75 73 02   ········ ·kronus·
0040  70 70 02 75 61 00 00 01  00 01                     pp·ua··· ··
```

Figure 29 DNS request

- Wireshark also shows an HTTP GET request to *kronus.pp/upwinload/get.php*. The malware might be trying to fetch something like an encryption key from its C&C server. This is the same request as we saw in InetSim logs.

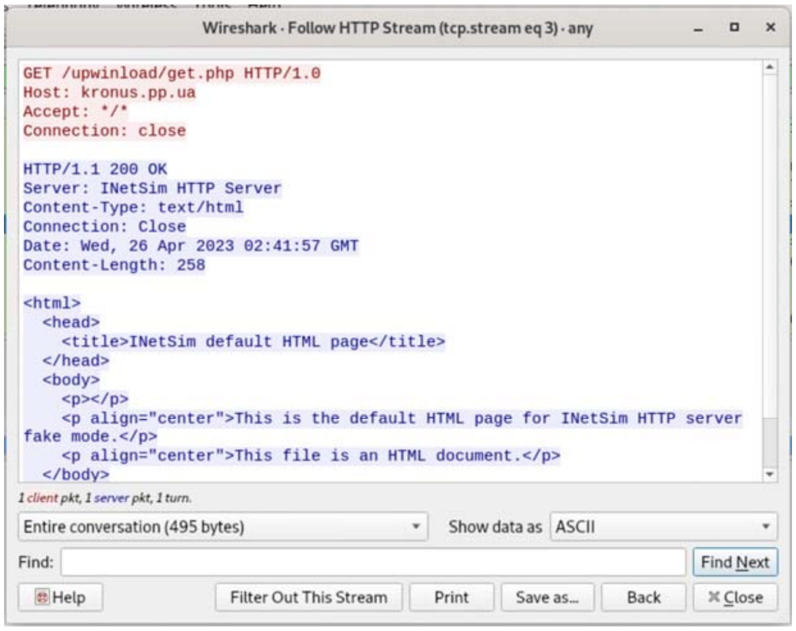


```
No.  Time          Source           Destination      Protocol Length Info
  36 20.804177384  192.168.245.129  192.168.245.133  TCP        68 51883 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
  37 20.804210366  192.168.245.133  192.168.245.129  TCP        68 80 → 51883 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM…
  38 20.804468410  192.168.245.129  192.168.245.133  TCP        62 51883 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
  39 20.804537710  192.168.245.129  192.168.245.133  TCP        89 51883 → 80 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=33 [TCP segment of…
  40 20.804546677  192.168.245.133  192.168.245.129  TCP        56 80 → 51883 [ACK] Seq=1 Ack=34 Win=64256 Len=0
  41 20.804685557  192.168.245.129  192.168.245.133  HTTP      110 GET /upwinload/get.php HTTP/1.0
  42 20.804689895  192.168.245.133  192.168.245.129  TCP        56 80 → 51883 [ACK] Seq=1 Ack=88 Win=64256 Len=0
  43 20.823028311  192.168.245.133  192.168.245.129  TCP       206 80 → 51883 [PSH, ACK] Seq=1 Ack=88 Win=64256 Len=150 [TCP segment of…
  44 20.825550761  192.168.245.133  192.168.245.129  HTTP      314 HTTP/1.1 200 OK  (text/html)
  45 20.825731399  192.168.245.129  192.168.245.133  TCP        62 51883 → 80 [ACK] Seq=88 Ack=410 Win=2101760 Len=0
  46 20.825990255  192.168.245.129  192.168.245.133  TCP        62 51883 → 80 [FIN, ACK] Seq=88 Ack=410 Win=2101760 Len=0

Frame 41: 110 bytes on wire (880 bits), 110 bytes captured (880 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.245.129, Dst: 192.168.245.133
Transmission Control Protocol, Src Port: 51883, Dst Port: 80, Seq: 34, Ack: 1, Len: 54
[2 Reassembled TCP Segments (87 bytes): #39(33), #41(54)]
Hypertext Transfer Protocol

0000  00 00 00 01 00 06 00 50  56 87 0d aa 00 00 08 00   ·······P V·······
0010  45 00 00 5e d4 d3 40 00  80 06 b9 6d c0 a8 f5 81   E··^··@· ···m····
0020  c0 a8 f5 85 ca ab 00 50  ba 2b 81 fc 27 e2 23 52   ·······P ·+··'·#R
0030  50 18 20 14 88 54 00 00  48 6f 73 74 3a 20 6b 72   P· ··T·· Host: kr
0040  6f 6e 75 73 2e 70 70 2e  75 61 0d 0a 41 63 63 65   onus.pp. ua··Acce
0050  70 74 3a 20 2a 2f 2a 0d  0a 43 6f 6e 6e 65 63 74   pt: */*· ·Connect
0060  69 6f 6e 3a 20 63 6c 6f  73 65 0d 0a 0d 0a         ion: clo se····
```

Activities
Wireshark
Apr 25 22:45
*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 3
Frame (110 bytes)
Reassembled TCP (87 bytes)
wireshark_anyFD3331.pcapng
Packets: 1762 · Displayed: 12 (0.7%)
Profile: Default

Figure 30 HTTP GET request

- Opening the GET request shows us its parameters in detail.



```
GET /upwinload/get.php HTTP/1.0
Host: kronus.pp.ua
Accept: */*
Connection: close

HTTP/1.1 200 OK
Server: INetSim HTTP Server
Content-Type: text/html
Connection: Close
Date: Wed, 26 Apr 2023 02:41:57 GMT
Content-Length: 258

<html>
  <head>
    <title>INetSim default HTML page</title>
  </head>
  <body>
    <p></p>
    <p align="center">This is the default HTML page for INetSim HTTP server
fake mode.</p>
    <p align="center">This file is an HTML document.</p>
  </body>
```

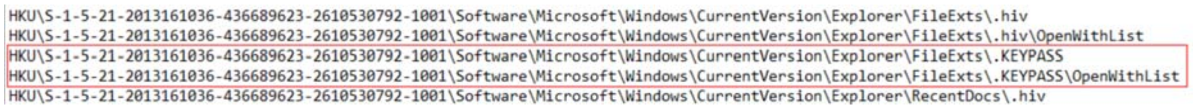
1 client pkt, 1 server pkt, 1 turn.

Entire conversation (495 bytes) Show data as ASCII

Figure 31 Contents of the GET request

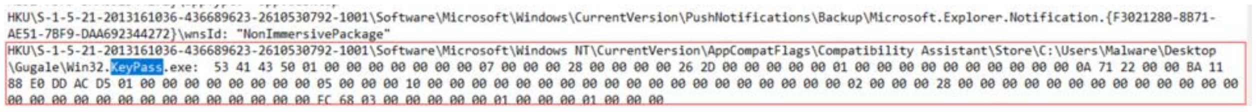
Regshot

- Regshot shows that following key gets added to the registry:
    - HKU\<SID>\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.KEYPASS

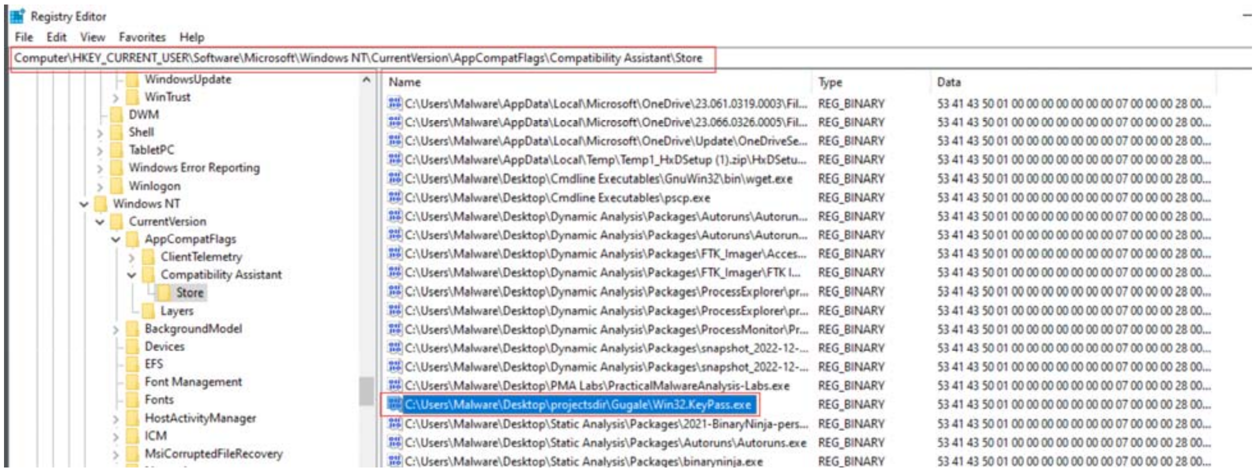


```
HKU\S-1-5-21-2013161036-436689623-2610530792-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv
HKU\S-1-5-21-2013161036-436689623-2610530792-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.hiv\OpenWithList
HKU\S-1-5-21-2013161036-436689623-2610530792-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.KEYPASS
HKU\S-1-5-21-2013161036-436689623-2610530792-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.KEYPASS\OpenWithList
HKU\S-1-5-21-2013161036-436689623-2610530792-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs\.hiv
```

- A key to malware's original executable path is added to HKU\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store



```
HKU\S-1-5-21-2013161036-436689623-2610530792-1001\Software\Microsoft\Windows\CurrentVersion\PushNotifications\Backup\Microsoft.Explorer.Notification.{F3021280-8B71-
AE51-7BF9-DAA692344272}\wnsId: "NonImmersivePackage"
HKU\S-1-5-21-2013161036-436689623-2610530792-1001\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store\C:\Users\Malware\Desktop
\Gugale\Win32.KeyPass.exe:  53 41 43 50 01 00 00 00 00 00 00 07 00 00 00 28 00 00 00 00 26 2D 00 00 00 00 01 00 00 00 00 00 00 00 00 0A 71 22 00 00 BA 11
88 E0 DD AC D5 01 00 00 00 00 00 00 00 05 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 02 00 00 00 28 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 FC 68 03 00 00 00 00 00 01 00 00 00 01 00 00 00
```



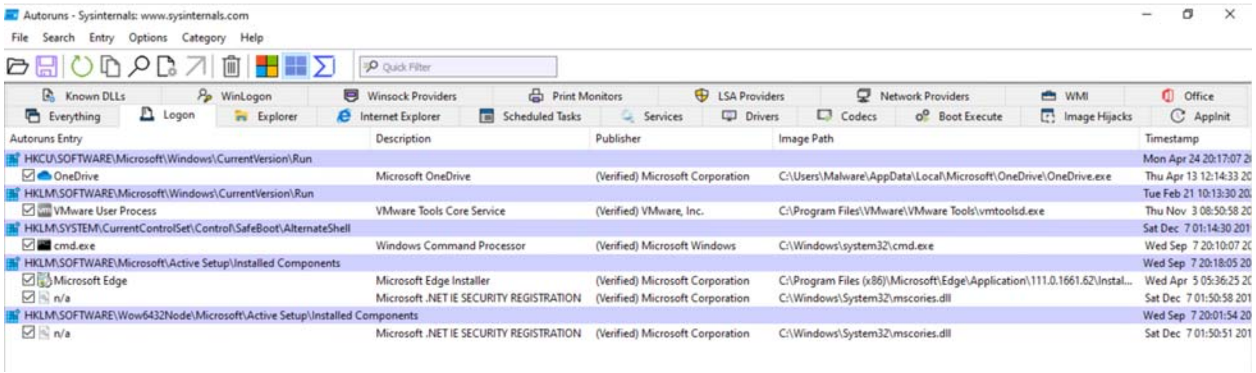
Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store

Name	Type	Data
C:\Users\Malware\AppData\Local\Microsoft\OneDrive\23.061.0319.0003\Fil...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\AppData\Local\Microsoft\OneDrive\23.066.0326.0005\Fil...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\AppData\Local\Microsoft\OneDrive\Update\OneDriveSe...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\AppData\Local\Temp\Temp1_HxDSetup (1).zip\HxDSetu...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Cmdline Executables\GnuWin32\bin\wget.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Cmdline Executables\pscp.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\Autoruns\Autorun...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\Autoruns\Autorun...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\FTK_Imager\Acces...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\FTK_Imager\FTK I...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\ProcessExplorer\pr...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\ProcessExplorer\pr...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\ProcessMonitor\Pr...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\snapshot_2022-12-...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Dynamic Analysis\Packages\snapshot_2022-12-...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\PMA Labs\PracticalMalwareAnalysis-Labs.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\projectsdir\Gugale\Win32.KeyPass.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Static Analysis\Packages\2021-BinaryNinja-pers...	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Static Analysis\Packages\Autoruns\Autoruns.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...
C:\Users\Malware\Desktop\Static Analysis\Packages\binaryninja.exe	REG_BINARY	53 41 43 50 01 00 00 00 00 00 00 00 07 00 00 00 28 00...

- The malware doesn't modify any persistence related keys.

Autoruns

- Autoruns shows nothing suspicious from which provides further evidence that the malware is not persistent.



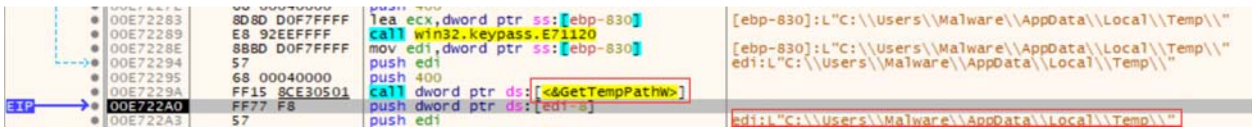
Autoruns Entry	Description	Publisher	Image Path	Timestamp
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				Mon Apr 24 20:17:07 2
OneDrive	Microsoft OneDrive	(Verified) Microsoft Corporation	C:\Users\Malware\AppData\Local\Microsoft\OneDrive\OneDrive.exe	Thu Apr 13 12:14:33 20
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run				Tue Feb 21 10:13:30 20
VMware User Process	VMware Tools Core Service	(Verified) VMware, Inc.	C:\Program Files\VMware\VMware Tools\vmtoolsd.exe	Thu Nov 3 08:50:58 20
HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\AlternateShell				Sat Dec 7 01:14:30 201
cmd.exe	Windows Command Processor	(Verified) Microsoft Windows	C:\Windows\system32\cmd.exe	Wed Sep 7 20:10:07 20
HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components				Wed Sep 7 20:18:05 20
Microsoft Edge	Microsoft Edge Installer	(Verified) Microsoft Corporation	C:\Program Files (x86)\Microsoft\Edge\Application\111.0.1661.62\Instal...	Wed Apr 5 05:36:25 20
n/a	Microsoft .NET IE SECURITY REGISTRATION	(Verified) Microsoft Corporation	C:\Windows\System32\mscories.dll	Sat Dec 7 01:50:58 201
HKLM\SOFTWARE\Wow6432Node\Microsoft\Active Setup\Installed Components				Wed Sep 7 20:01:54 20
n/a	Microsoft .NET IE SECURITY REGISTRATION	(Verified) Microsoft Corporation	C:\Windows\System32\mscories.dll	Sat Dec 7 01:50:51 201

Figure 32 Nothing suspicious shown by Autoruns

Ghidra Analysis and x32dbg Analysis

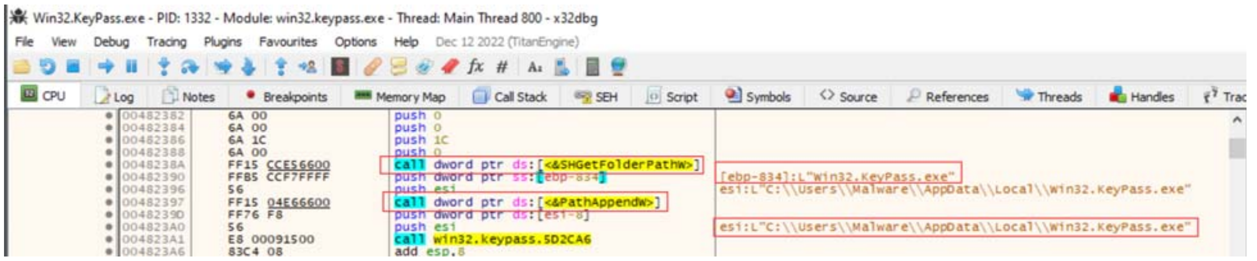
1. **Creation of path to copy executable to:** The malware creates the path *C:\Users\<username>\AppData\Local* using calls to **GetTempPathW** and

SHGetFolderPathW to get the path to *AppData\Local* folder and appends the executable's name to it using **PathAppendW** to create the path where the executable would be copied.



```
00E72283    8D8D D0F7FFFF    lea ecx,dword ptr ss:[ebp-830]      [ebp-830]:L"C:\\Users\\Malware\\AppData\\Local\\Temp\\"
00E72289    E8 92EEFFFF      call win32.keypass.E71120
00E7228E    8B8D D0F7FFFF    mov edi,dword ptr ss:[ebp-830]      [ebp-830]:L"C:\\Users\\Malware\\AppData\\Local\\Temp\\"
00E72294    57               push edi                            edi:L"C:\\Users\\Malware\\AppData\\Local\\Temp\\"
00E72295    68 00040000      push 400
00E7229A    FF15 8CE30501    call dword ptr ds:[<&GetTempPathW>]
00E722A0    FF77 F8          push dword ptr ds:[edi-8]
00E722A3    57               push edi                            edi:L"C:\\Users\\Malware\\AppData\\Local\\Temp\\"
```

Figure 33 Obtaining path to AppData\Local via GetTempPathW()



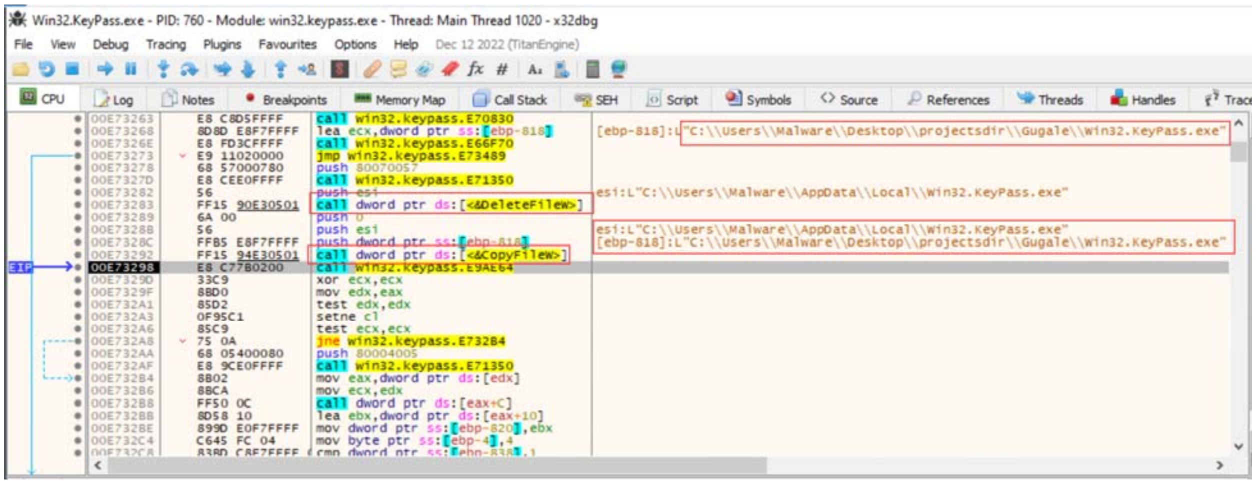
Win32.KeyPass.exe - PID: 1332 - Module: win32.keypass.exe - Thread: Main Thread 800 - x32dbg

File View Debug Tracing Plugins Favourites Options Help Dec 12 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trac

```
00482382    6A 00            push 0
00482384    6A 00            push 0
00482386    6A 1C            push 1C
00482388    6A 00            push 0
0048238A    FF15 CCE56600    call dword ptr ds:[<&SHGetFolderPathW>]
00482390    FFB5 CCF7FFFF    push dword ptr ss:[ebp-834]         [ebp-834]:L"Win32.KeyPass.exe"
00482396    56               push esi                            esi:L"C:\\Users\\Malware\\AppData\\Local\\Win32.KeyPass.exe"
00482397    FF15 04E66600    call dword ptr ds:[<&PathAppendW>]
0048239D    FF76 F8          push dword ptr ds:[esi-8]
004823A0    56               push esi                            esi:L"C:\\Users\\Malware\\AppData\\Local\\Win32.KeyPass.exe"
004823A1    E8 00091500      call win32.keypass.5D2CA6
004823A6    83C4 08          add esp,8
```

Figure 34 Construction of path to copy executable to

2. **Deleting executable from original location and copying it to AppData\Local folder:** The malware deletes the original executable with a call to **DeleteFileW** and copies the contents of the original executable to C:\Users\<username>\AppData\Local by a call to **CopyFilesW**.



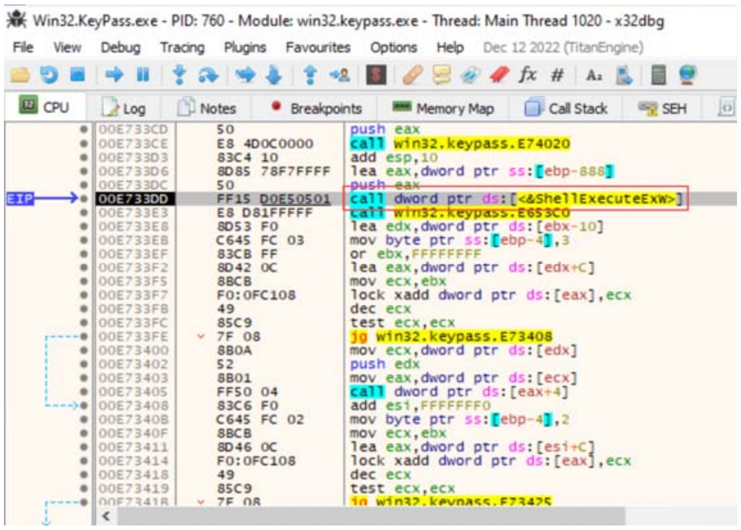
Win32.KeyPass.exe - PID: 760 - Module: win32.keypass.exe - Thread: Main Thread 1020 - x32dbg

File View Debug Tracing Plugins Favourites Options Help Dec 12 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trac

```
00E73263    E8 C8D5FFFF      call win32.keypass.E70830
00E73268    8D8D E8F7FFFF    lea ecx,dword ptr ss:[ebp-818]      [ebp-818]:L"C:\\Users\\Malware\\Desktop\\projectsdir\\Gugale\\Win32.KeyPass.exe"
00E7326E    E8 FD3CFFFF      call win32.keypass.E66F70
00E73273    E9 11020000      jmp win32.keypass.E73489
00E73278    68 57000780      push 80070057
00E7327D    E8 CEE0FFFF      call win32.keypass.E71350
00E73282    56               push esi                            esi:L"C:\\Users\\Malware\\AppData\\Local\\Win32.KeyPass.exe"
00E73283    FF15 90E30501    call dword ptr ds:[<&DeleteFileW>]
00E73289    6A 00            push 0
00E7328B    56               push esi                            esi:L"C:\\Users\\Malware\\AppData\\Local\\Win32.KeyPass.exe"
00E7328C    FFB5 E8F7FFFF    push dword ptr ss:[ebp-818]         [ebp-818]:L"C:\\Users\\Malware\\Desktop\\projectsdir\\Gugale\\Win32.KeyPass.exe"
00E73292    FF15 94E30501    call dword ptr ds:[<&CopyFileW>]
00E73298    E8 C77B0200      call win32.keypass.E9AE64
00E7329D    33C9             xor ecx,ecx
00E7329F    8BD0             mov edx,eax
00E732A1    85D2             test edx,edx
00E732A3    0F95C1           setne cl
00E732A6    85C9             test ecx,ecx
00E732A8    75 0A            jne win32.keypass.E732B4
00E732AA    68 05400080      push 80004005
00E732AF    E8 9CE0FFFF      call win32.keypass.E71350
00E732B4    8B02             mov eax,dword ptr ds:[edx]
00E732B6    8BCA             mov ecx,edx
00E732B8    FF50 0C          call dword ptr ds:[eax+C]
00E732BB    8D58 10          lea ebx,dword ptr ds:[eax+10]
00E732BE    899D E0F7FFFF    mov dword ptr ss:[ebp-820],ebx
00E732C4    C645 FC 04       mov byte ptr ss:[ebp-4],4
00E732C8    83BD C8F7FFFF    cmp dword ptr ss:[ebp-838],1
```

Figure 35 Copying file to AppData\Local and deleting original file

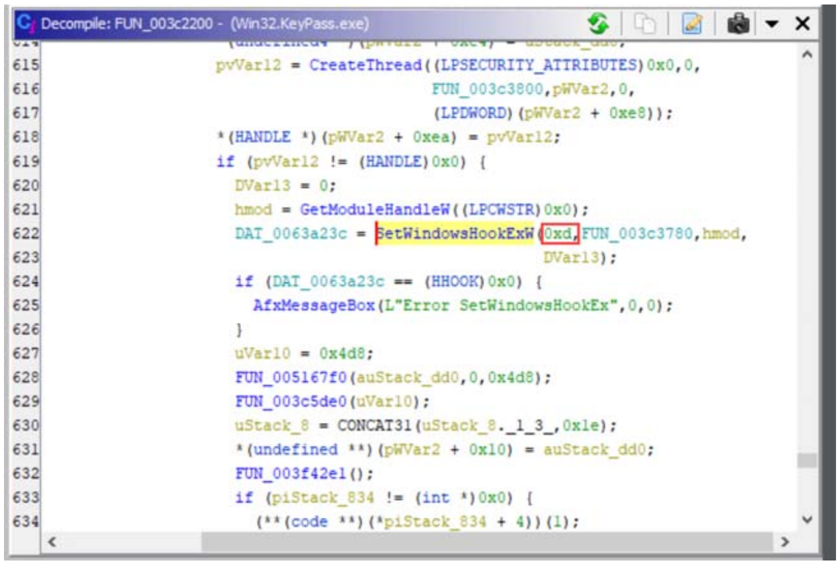
3. **Executing a new process:** The sample calls **ShellExecuteExW** with the path to the malware's executable file copied to *C:\Users\<username>\AppData\Local.*


Win32.KeyPass.exe - PID: 760 - Module: win32.keypass.exe - Thread: Main Thread 1020 - x32dbg
File View Debug Tracing Plugins Favourites Options Help Dec 12 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH
EIP

```
00E733CD  50              push eax
00E733CE  E8 4D0C0000     call win32.keypass.E74020
00E733D3  83C4 10         add esp,10
00E733D6  8D85 78F7FFFF   lea eax,dword ptr ss:[ebp-888]
00E733DC  50              push eax
00E733DD  FF15 D0E50501   call dword ptr ds:[<&ShellExecuteExW>]
00E733E3  E8 D81FFFFF     call win32.keypass.E653C0
00E733E8  8D53 F0         lea edx,dword ptr ds:[ebx-10]
00E733EB  C645 FC 03      mov byte ptr ss:[ebp-4],3
00E733EF  83CB FF         or ebx,FFFFFFFF
00E733F2  8D42 0C         lea eax,dword ptr ds:[edx+C]
00E733F5  8BCB            mov ecx,ebx
00E733F7  F0:0FC108       lock xadd dword ptr ds:[eax],ecx
00E733FB  49              dec ecx
00E733FC  85C9            test ecx,ecx
00E733FE  7F 08           jg win32.keypass.E73408
00E73400  8B0A            mov ecx,dword ptr ds:[edx]
00E73402  52              push edx
00E73403  8B01            mov eax,dword ptr ds:[ecx]
00E73405  FF50 04         call dword ptr ds:[eax+4]
00E73408  83C6 F0         add esi,FFFFFFF0
00E7340B  C645 FC 02      mov byte ptr ss:[ebp-4],2
00E7340F  8BCB            mov ecx,ebx
00E73411  8D46 0C         lea eax,dword ptr ds:[esi+C]
00E73414  F0:0FC108       lock xadd dword ptr ds:[eax],ecx
00E73418  49              dec ecx
00E73419  85C9            test ecx,ecx
00E7341B  7F 08           jg win32.keypass.E73425
```

Figure 36 Creating a subprocess with ShellExecuteExW

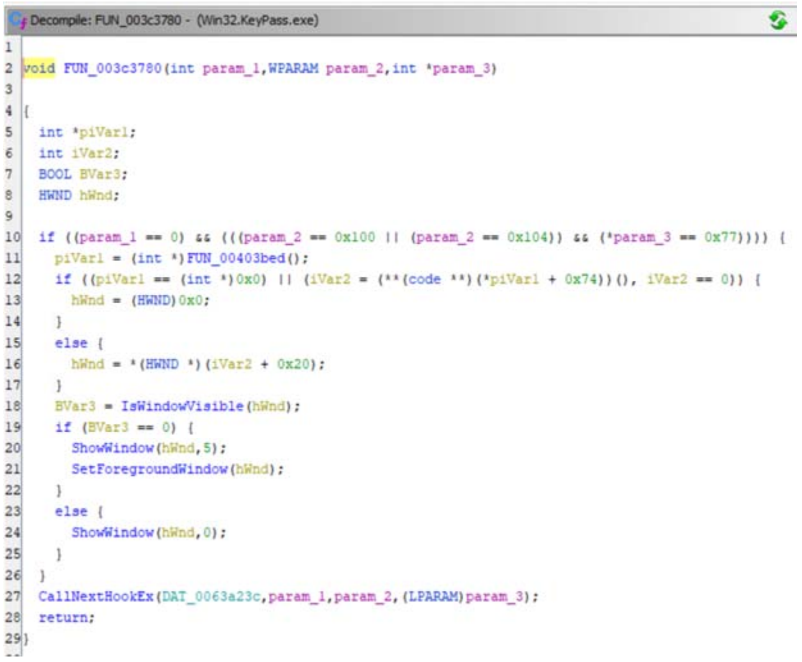
4. **Installs a KeyBoard hook:** The malware installs a **WH_KEYBOARD_LL** hook. As can be seen from the image below, the keyboard hook is installed using the **SetWindowsHookExW** function. 0xd which is the integer value of WH_KEYBOARD_LL is passed as a parameter to this function. The hooking function to be called after a key is pressed is also passed to the SetWindowsHookExW method.



```
    pvVar12 = CreateThread((LPSECURITY_ATTRIBUTES)0x0,0,
                           FUN_003c3800,pWVar2,0,
                           (LPDWORD)(pWVar2 + 0xe8));
    *(HANDLE *)(pWVar2 + 0xea) = pvVar12;
    if (pvVar12 != (HANDLE)0x0) {
      DVar13 = 0;
      hmod = GetModuleHandleW((LPCWSTR)0x0);
      DAT_0063a23c = SetWindowsHookExW(0xd,FUN_003c3780,hmod,
                                       DVar13);
      if (DAT_0063a23c == (HHOOK)0x0) {
        AfxMessageBox(L"Error SetWindowsHookEx",0,0);
      }
      uVar10 = 0x4d8;
      FUN_005167f0(auStack_dd0,0,0x4d8);
      FUN_003c5de0(uVar10);
      uStack_8 = CONCAT31(uStack_8._1_3_,0x1e);
      *(undefined **)(pWVar2 + 0x10) = auStack_dd0;
      FUN_003f42e1();
      if (piStack_834 != (int *)0x0) {
        (**(code **)(*piStack_834 + 4))(1);
```

Figure 37 The sample installs a keyboard hook

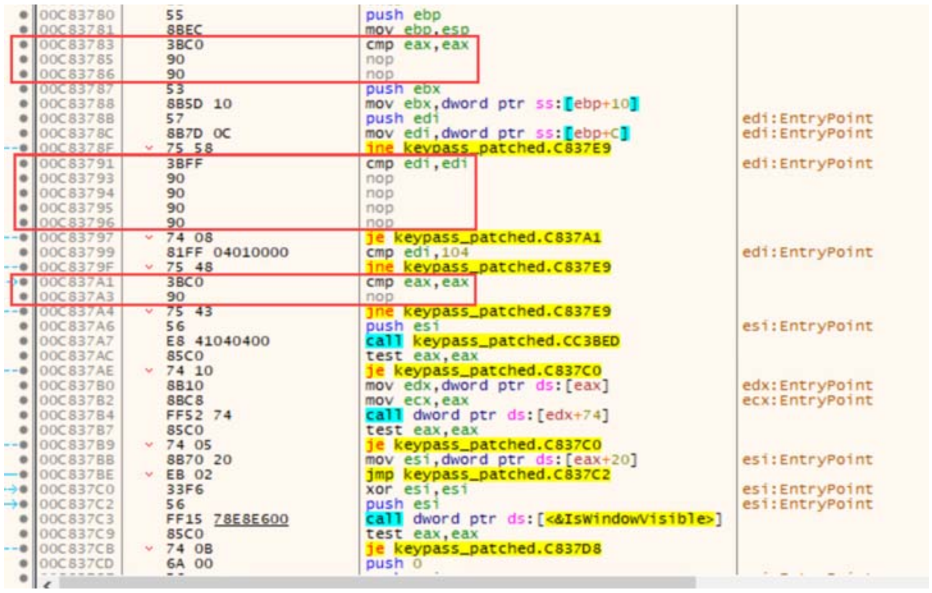
Hooking function: The hooking function seems to be responsible for displaying a window if param1, param2 and param3 have the required values. Since function FUN_003c3780 is a LowLevelKeyboardProc callback function, the parameters passed to it are defined in MSDN.



```

void FUN_003c3780(int param_1,WPARAM param_2,int *param_3)

{
  int *piVar1;
  int iVar2;
  BOOL BVar3;
  HWND hWnd;

  if ((param_1 == 0) && (((param_2 == 0x100 || (param_2 == 0x104)) && (*param_3 == 0x77)))) {
    piVar1 = (int *)FUN_00403bed();
    if ((piVar1 == (int *)0x0) || (iVar2 = (**(code **)(*piVar1 + 0x74))(), iVar2 == 0)) {
      hWnd = (HWND)0x0;
    }
    else {
      hWnd = *(HWND *)(iVar2 + 0x20);
    }
    BVar3 = IsWindowVisible(hWnd);
    if (BVar3 == 0) {
      ShowWindow(hWnd,5);
      SetForegroundWindow(hWnd);
    }
    else {
      ShowWindow(hWnd,0);
    }
  }
  CallNextHookEx(DAT_0063a23c,param_1,param_2,(LPARAM)param_3);
  return;
}
```

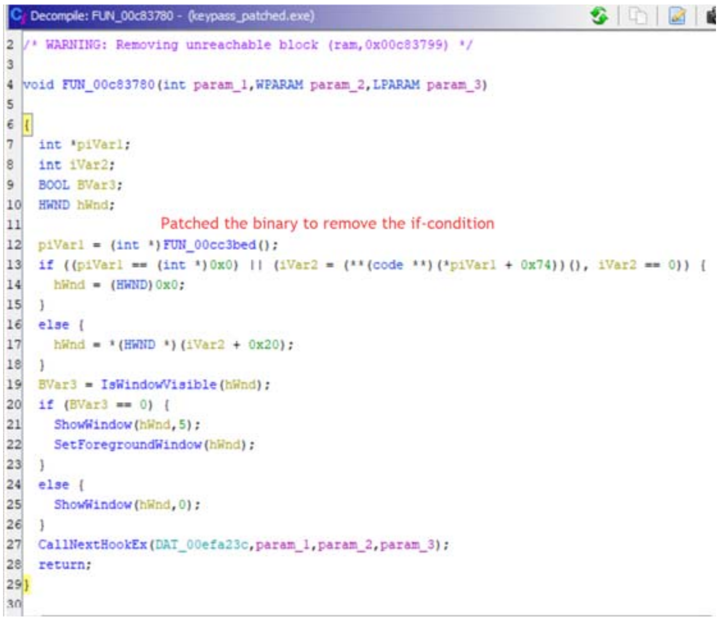
Figure 38 Hooking function to be called after key press

To get rid of the if-condition and to get this window to display after pressing any key, I patched the binary in x32dbg as per the following image:



```
00C83780      55              push ebp
00C83781      8BEC            mov ebp,esp
00C83783      3BC0            cmp eax,eax
00C83785      90              nop
00C83786      90              nop
00C83787      53              push ebx
00C83788      8B5D 10         mov ebx,dword ptr ss:[ebp+10]
00C8378B      57              push edi                                   edi:EntryPoint
00C8378C      8B7D 0C         mov edi,dword ptr ss:[ebp+C]               edi:EntryPoint
00C8378F      75 58           jne keypass_patched.C837E9
00C83791      3BFF            cmp edi,edi                                edi:EntryPoint
00C83793      90              nop
00C83794      90              nop
00C83795      90              nop
00C83796      90              nop
00C83797      74 08           je keypass_patched.C837A1
00C83799      81FF 04010000   cmp edi,104                                edi:EntryPoint
00C8379F      75 48           jne keypass_patched.C837E9
00C837A1      3BC0            cmp eax,eax
00C837A3      90              nop
00C837A4      75 43           jne keypass_patched.C837E9
00C837A6      56              push esi                                   esi:EntryPoint
00C837A7      E8 41040400     call keypass_patched.CC3BED
00C837AC      85C0            test eax,eax
00C837AE      74 10           je keypass_patched.C837C0
00C837B0      8B10            mov edx,dword ptr ds:[eax]                 edx:EntryPoint
00C837B2      8BC8            mov ecx,eax                                ecx:EntryPoint
00C837B4      FF52 74         call dword ptr ds:[edx+74]
00C837B7      85C0            test eax,eax
00C837B9      74 05           je keypass_patched.C837C0
00C837BB      8B70 20         mov esi,dword ptr ds:[eax+20]              esi:EntryPoint
00C837BE      EB 02           jmp keypass_patched.C837C2
00C837C0      33F6            xor esi,esi                                esi:EntryPoint
00C837C2      56              push esi                                   esi:EntryPoint
00C837C3      FF15 78E8E600   call dword ptr ds:[<&IsWindowVisible>]
00C837C9      85C0            test eax,eax
00C837CB      74 0B           je keypass_patched.C837D8
00C837CD      6A 00           push 0
```

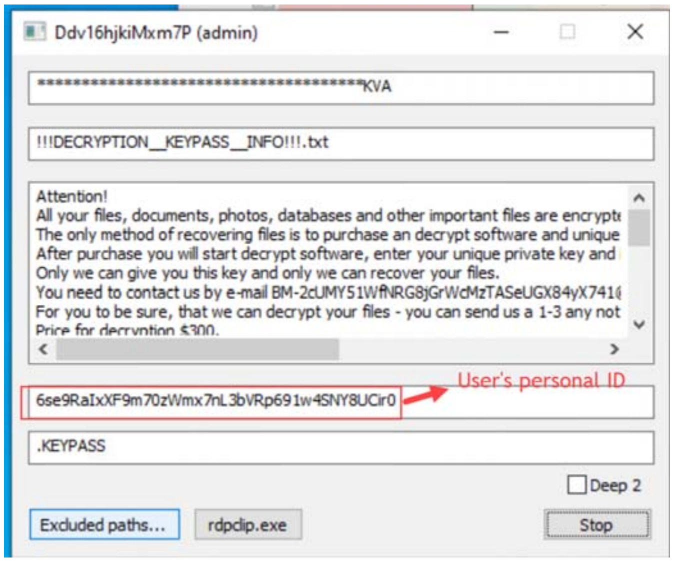
Figure 39 Patched code in x32dbg


Decompile: FUN_00c83780 - (keypass_patched.exe)
Patched the binary to remove the if-condition

```
/* WARNING: Removing unreachable block (ram,0x00c83799) */

void FUN_00c83780(int param_1,WPARAM param_2,LPARAM param_3)

{
  int *piVar1;
  int iVar2;
  BOOL BVar3;
  HWND hWnd;

  piVar1 = (int *)FUN_00cc3bed();
  if ((piVar1 == (int *)0x0) || (iVar2 = (**(code **)(*piVar1 + 0x74))(), iVar2 == 0)) {
    hWnd = (HWND)0x0;
  }
  else {
    hWnd = *(HWND *)(iVar2 + 0x20);
  }
  BVar3 = IsWindowVisible(hWnd);
  if (BVar3 == 0) {
    ShowWindow(hWnd,5);
    SetForegroundWindow(hWnd);
  }
  else {
    ShowWindow(hWnd,0);
  }
  CallNextHookEx(DAT_00efa23c,param_1,param_2,param_3);
  return;
}
```

Figure 40 Patched code in Ghidra

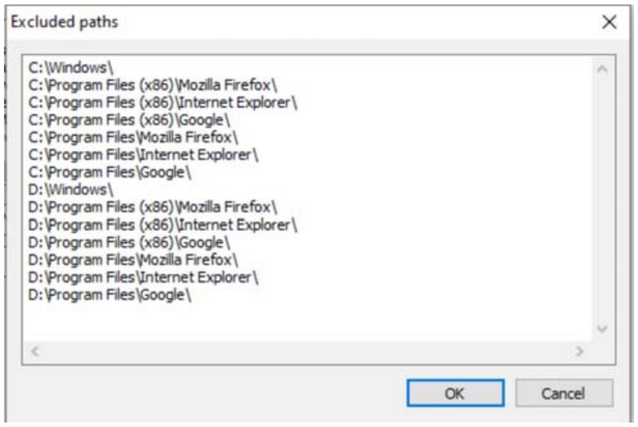
Running the patched binary and pressing any keyboard key brings up the window in the image below. This window contains the ransom note message (The same as the one present in !!!DECRYPTION_KEYPASS_INFO!!!.txt). It also shows the user's personal ID.



```
***************************************KVA
!!!DECRYPTION__KEYPASS__INFO!!!.txt
Attention!
All your files, documents, photos, databases and other important files are encrypte
The only method of recovering files is to purchase an decrypt software and unique
After purchase you will start decrypt software, enter your unique private key and
Only we can give you this key and only we can recover your files.
You need to contact us by e-mail BM-2cUMY51WfNRG8jGrWcMzTASeUGX84yX741(
For you to be sure, that we can decrypt your files - you can send us a 1-3 any not
Price for decryption $300.
6se9RaIxXF9m70zWmx7nL3bVRp691w4SNY8UCir0
.KEYPASS
```

Figure 41 Malware's hidden window

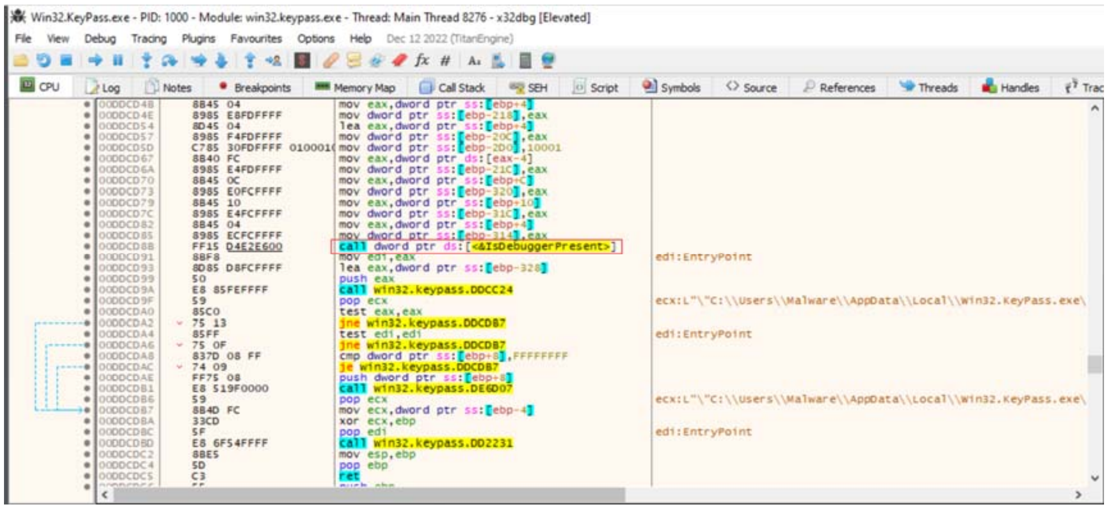
Clicking on the "Excluded paths" button shows another window containing a list of the directories whose files the malware doesn't encrypt.



```
C:\Windows\
C:\Program Files (x86)\Mozilla Firefox\
C:\Program Files (x86)\Internet Explorer\
C:\Program Files (x86)\Google\
C:\Program Files\Mozilla Firefox\
C:\Program Files\Internet Explorer\
C:\Program Files\Google\
D:\Windows\
D:\Program Files (x86)\Mozilla Firefox\
D:\Program Files (x86)\Internet Explorer\
D:\Program Files (x86)\Google\
D:\Program Files\Mozilla Firefox\
D:\Program Files\Internet Explorer\
D:\Program Files\Google\
```

Figure 42 Excluded paths

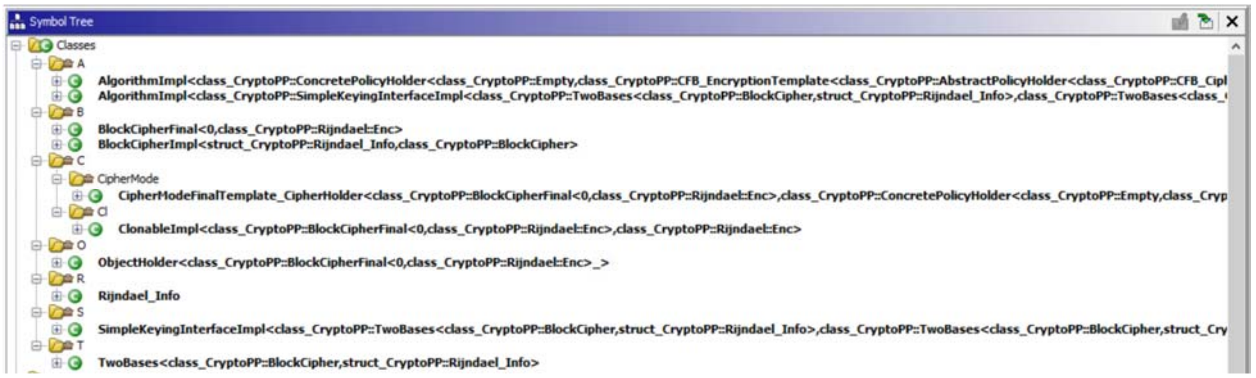
5. **Anti-Debugging:** Some of the subprocesses the malware creates employ anti-debugging using **IsDebuggerPresent** Windows API function. It is very difficult to attach a debugger to the subprocesses created by the malware as they are transient and disappear in a few milliseconds.


Win32.KeyPass.exe - PID: 1000 - Module: win32.keypass.exe - Thread: Main Thread 8276 - x32dbg [Elevated]
File View Debug Tracing Plugins Favourites Options Help Dec 12 2022 (TitanEngine)
CPU Log Notes Breakpoints Memory Map Call Stack SEH Script Symbols Source References Threads Handles Trac

```
00DDCD4B   8B45 04             mov eax,dword ptr ss:[ebp+4]
00DDCD4E   8985 E8FDFFFF       mov dword ptr ss:[ebp-218],eax
00DDCD54   8D45 04             lea eax,dword ptr ss:[ebp+4]
00DDCD57   8985 F4FDFFFF       mov dword ptr ss:[ebp-20C],eax
00DDCD5D   C785 30FDFFFF 010001( mov dword ptr ss:[ebp-2D0],10001
00DDCD67   8B40 FC             mov eax,dword ptr ds:[eax-4]
00DDCD6A   8985 E4FDFFFF       mov dword ptr ss:[ebp-21C],eax
00DDCD70   8B45 0C             mov eax,dword ptr ss:[ebp+C]
00DDCD73   8985 E0FCFFFF       mov dword ptr ss:[ebp-320],eax
00DDCD79   8B45 10             mov eax,dword ptr ss:[ebp+10]
00DDCD7C   8985 E4FCFFFF       mov dword ptr ss:[ebp-31C],eax
00DDCD82   8B45 04             mov eax,dword ptr ss:[ebp+4]
00DDCD85   8985 ECFCFFFF       mov dword ptr ss:[ebp-314],eax
00DDCD8B   FF15 D4E2E600       call dword ptr ds:[<&IsDebuggerPresent>]
00DDCD91   8BF8                mov edi,eax                            edi:EntryPoint
00DDCD93   8D85 D8FCFFFF       lea eax,dword ptr ss:[ebp-328]
00DDCD99   50                  push eax
00DDCD9A   E8 85FEFFFF         call win32.keypass.DDCC24
00DDCD9F   59                  pop ecx                                ecx:L"\"C:\\Users\\Malware\\AppData\\Local\\Win32.KeyPass.exe\
00DDCDA0   85C0                test eax,eax
00DDCDA2   75 13               jne win32.keypass.DDCDB7
00DDCDA4   85FF                test edi,edi                           edi:EntryPoint
00DDCDA6   75 0F               jne win32.keypass.DDCDB7
00DDCDA8   837D 08 FF          cmp dword ptr ss:[ebp+8],FFFFFFFF
00DDCDAC   74 09               je win32.keypass.DDCDB7
00DDCDAE   FF75 08             push dword ptr ss:[ebp+8]
00DDCDB1   E8 519F0000         call win32.keypass.DE6D07
00DDCDB6   59                  pop ecx                                ecx:L"\"C:\\Users\\Malware\\AppData\\Local\\Win32.KeyPass.exe\
00DDCDB7   8B4D FC             mov ecx,dword ptr ss:[ebp-4]
00DDCDBA   33CD                xor ecx,ebp
00DDCDBC   5F                  pop edi                                edi:EntryPoint
00DDCDBD   E8 6F54FFFF         call win32.keypass.DD2231
00DDCDC2   8BE5                mov esp,ebp
00DDCDC4   5D                  pop ebp
00DDCDC5   C3                  ret
```

Figure 43 Anti-debugging using IsDebuggerPresent

6. **Encryption:** A lot of classes in the sample include the word Rijndael which is another name for the **Advanced Encryption Standard (AES)** algorithm. The sample thus uses AES for encryption of files.



```
Symbol Tree
Classes
  A
    AlgorithmImpl<class_CryptoPP::ConcretePolicyHolder<class_CryptoPP::Empty,class_CryptoPP::CFB_EncryptionTemplate<class_CryptoPP::AbstractPolicyHolder<class_CryptoPP::CFB_Cipl
    AlgorithmImpl<class_CryptoPP::SimpleKeyingInterfaceImpl<class_CryptoPP::TwoBases<class_CryptoPP::BlockCipher,struct_CryptoPP::Rijndael_Info>,class_CryptoPP::TwoBases<class_
  B
    BlockCipherFinal<0,class_CryptoPP::Rijndael::Enc>
    BlockCipherImpl<struct_CryptoPP::Rijndael_Info,class_CryptoPP::BlockCipher>
  C
    CipherMode
      CipherModeFinalTemplate_CipherHolder<class_CryptoPP::BlockCipherFinal<0,class_CryptoPP::Rijndael::Enc>,class_CryptoPP::ConcretePolicyHolder<class_CryptoPP::Empty,class_Cryp
    Cl
      ClonableImpl<class_CryptoPP::BlockCipherFinal<0,class_CryptoPP::Rijndael::Enc>,class_CryptoPP::Rijndael::Enc>
  O
    ObjectHolder<class_CryptoPP::BlockCipherFinal<0,class_CryptoPP::Rijndael::Enc>_>
  R
    Rijndael_Info
  S
    SimpleKeyingInterfaceImpl<class_CryptoPP::TwoBases<class_CryptoPP::BlockCipher,struct_CryptoPP::Rijndael_Info>,class_CryptoPP::TwoBases<class_CryptoPP::BlockCipher,struct_Cry
  T
    TwoBases<class_CryptoPP::BlockCipher,struct_CryptoPP::Rijndael_Info>
```

Figure 44 Lots of classes include "Rijndael"

Indicators of Compromise

Host Based Indicators

Following are the host-based indicators can be used to detect the malware:

- Presence of file !!!DECRYPTION_KEYPASS_INFO!!!.txt
- Presence of files with .KEYPASS extension
- Presence of the following registry key:
  - HKU\<SID>\Software\Microsoft\Windows\CurrentVersion\Explorer\FileExts\.KEYPASS
- Periodic notifications of “Files waiting to be burned to disk” even when the user hasn’t issued a burn command.
- Being unable to open programs installed outside C:\Windows.

Network Based Indicators

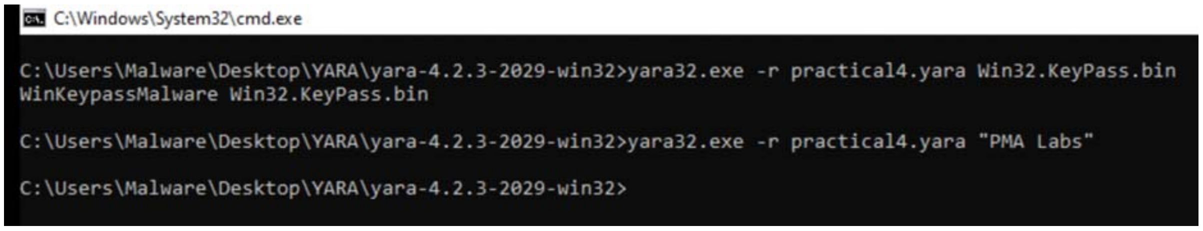
The following network based indicators of compromise can be used:

- DNS request for **kronus.pp.ua**
- HTTP GET request to **http://kronus.pp.ua/upwinload/get.php** on port 80

YARA Rule

```
rule WinKeypassMalware {
    meta:
        description = "Rule to detect WinKeypass malware"
        author = "Rachana"
        date = "4/25/2023"
    strings:
        $s1 = "kronus.pp.ua/upwinload/get.php" ascii wide nocase
        $s2 = "delself.bat" ascii wide nocase
        $s3 = "DECRYPTION_KEYPASS_INFO.txt" ascii wide nocase
        $s4 = "G:\\Doc\\My work (C++)\\_New
2018\\Encryption\\Release\\encrypt.pdb" ascii wide nocase
        $s5 = "G:\\FromInet\\Include\\boost_1_65_1" ascii wide nocase
        $s6 = "KEYPASS" ascii wide nocase
        $s7 = "keypassdecrypt@india.com" ascii wide nocase
        $s8 = "Ramsil" ascii wide nocase

        $x1 = "rdpclip.exe" ascii wide nocase
        $x2 = "GetTickCount" ascii wide nocase
        $x3 = "IsDebuggerPresent" ascii wide nocase
        $x4 = "SetWindowsHookEx" ascii wide nocase
        $x5 =
"Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Network"
ascii wide nocase
        $x6 = "asInvoker" ascii wide nocase

    condition:
        uint16(0) == 0x5a4d and filesize < 2890KB and (5 of ($s*) and
4 of ($x*))
}
```

- The YARA rule fires on Win32.KeyPass's binary but it doesn't fire on any other PMA Lab binaries.



```
C:\Windows\System32\cmd.exe

C:\Users\Malware\Desktop\YARA\yara-4.2.3-2029-win32>yara32.exe -r practical4.yara Win32.KeyPass.bin
WinKeypassMalware Win32.KeyPass.bin

C:\Users\Malware\Desktop\YARA\yara-4.2.3-2029-win32>yara32.exe -r practical4.yara "PMA Labs"

C:\Users\Malware\Desktop\YARA\yara-4.2.3-2029-win32>
```

Figure 45 Result of running the YARA rule on Keypass binary and PMA labs