Abertay University

Malware Analysis

Szymon Lewandowski - 2002026

CMP320: Advanced Ethical Hacking

2022/23

.

Abstract

The contents of this report covers a full analysis of a malware sample conducted by a malware analyst. The purpose of this report was to investigate an unknown malware sample to determine its type, characteristics, components and functionality while making use of appropriate malware analysis techniques.

The malware analyst has successfully documented their findings in the procedure part of the report including static and dynamic malware analysis. This was done using various pieces of software to investigate the sample's meta data as well as executing the malware onto a virtual machine to observe and analyse it in action.

The analysis technique sections in the procedure part of the report allowed the malware analyst to determine the type of malware being ransom-ware software called ,"wanaCryptor", that infected many computer systems and organization in 2017. The process of finding out the characteristics of this type of malware, its components and functionality can be found documented in the procedure stage of this report.

Suitable countermeasures have been suggested in the discussion part of the report and how to protect computer systems from this specific malware sample. It is important that people are aware of this type of malware and take steps to mitigate the chances of it infecting computer systems and putting large amounts of data at risk.

.

Contents

.

1 Introduction

1.1 Background

Viruses, otherwise known as malware is a software type used by hackers to gain unauthorized access to a computer system with malicious intent. This is done for the purpose of damaging and/or disrupting the computer system for illicit gain.

As a result, malware analysts are tasked with identifying this kind of software and determine its type, components, characteristics and functionality to fully understand how it works and prevent it from infecting a computer system in the future. Malware analysts are usually part of larger organizations and teams specialized in analyzing malicious software.

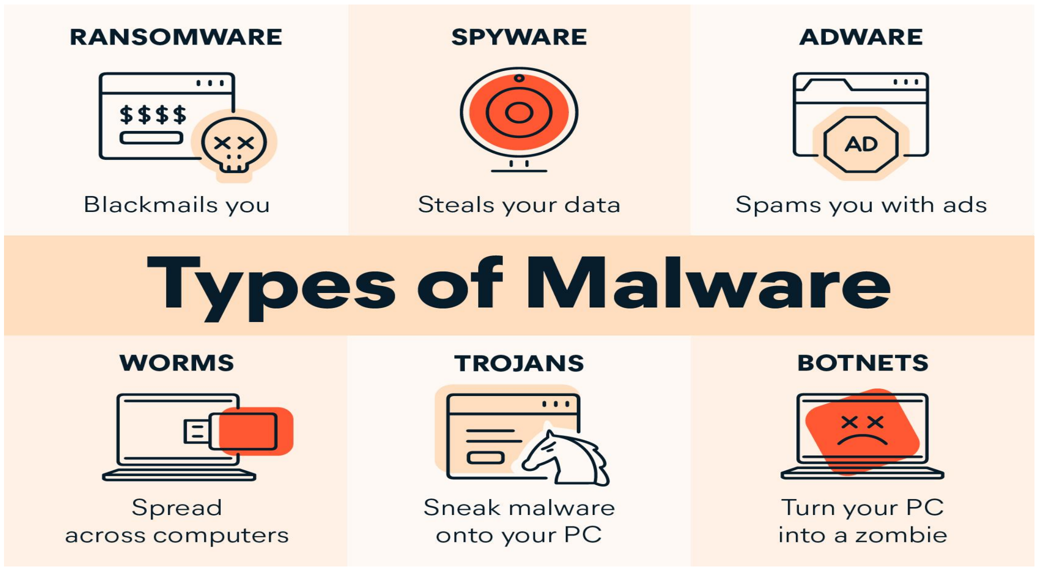


Figure 1: Types of malware (Avast, 2023).

There are a few different types of malicious software as seen by figure 1. It is important that malware analysts know the characteristics of each of these types of malware and understand how malicious software can be executed on the target system. This is extremely important for preventing and reducing the amount of annual malware attacks on computer systems.

Nowadays, hackers are aware of malware analysis and people trying to prevent such attacks. This is why malware analysis is not always that simple. Malware developers design the malware payloads in such a way that it cannot be detected by the system or the user. This can be done by packing files, disguising files as legitimate ones etc.

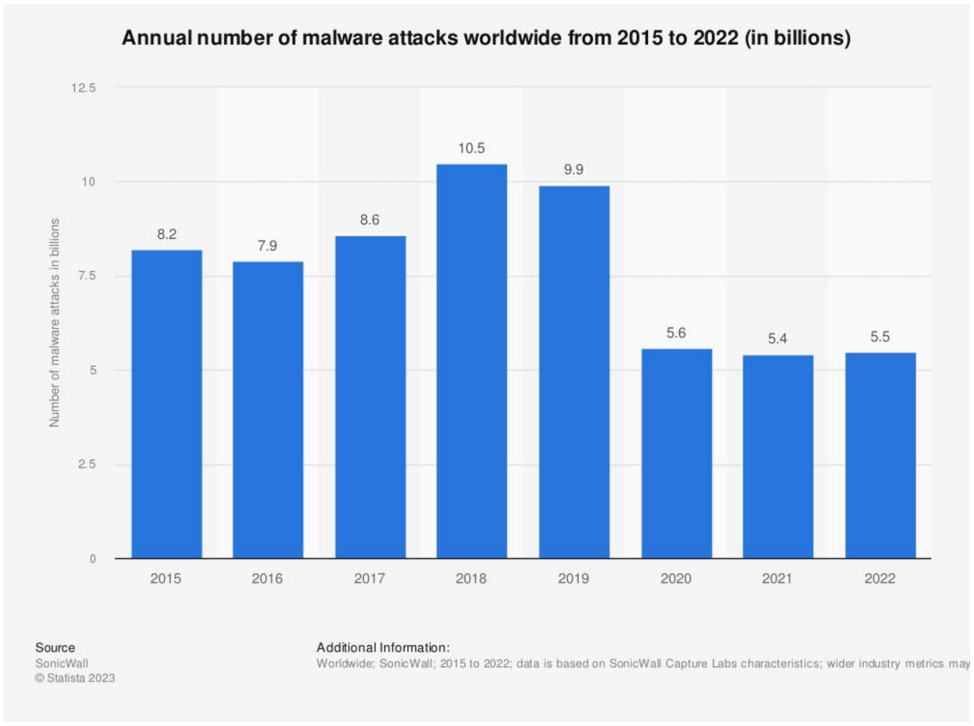


Figure 2: Worldwide annual malware attacks statistics (statista.com, 2023).

Malware attacks have been an on-going issue ever since the first computer system and with technology constantly advancing, it is important to ensure systems are as secure and up-to-date as much as possible. The figure above shows the number of malware attacks in billions between the years 2015 and 2022. The graph illustrating the statistics has a downward trend after 2019, reducing the total worldwide malware attacks by 4.3 billion in 2020. This shows that we're making a step in the right direction, focusing on system security and attack prevention. However, this still means that on average during 2020 to 2022, there are still 5.5 billion annual malware attacks in the world. Cyber security is improving but we've still got a long way to go. This is why malware analysis is crucial to improving and safe-guarding computer systems all over the world.

The malware analyst has been tasked with providing thorough analysis and investigation into a malware sample. Determining its characteristics, components, type and functionality. In this report, the analyst will go through their procedure and step-by-step analysis using static and dynamic techniques.

The static analysis part of the report will feature passive examination of the target malware sample. The malware analyst will examine the sample without running it, focusing on gathering as much information about the sample in a safe manner.

Dynamic analysis will be featured later in the report and will feature the malware analyst running the target executable file. This technique is useful to understand what the malware sample actually does as it runs in real-time and infects the computer system.

1.2 AIMS

The aims of the report are as follows:

- Perform static and dynamic malware analysis and document the procedure.
- Determine the type, characteristics, components and functionality of the malware sample.
- Discuss and analyse the results and provide mitigation for the determined malware.

2 Procedure

2.1 Overview of Procedure

For this report, the malware analyst has been given a malware sample to analyse on the CMP320 Malware virtual machine. This is a crucial part in the procedure as malware analysis can be very dangerous if not isolated in a virtual machine.

The malware analyst has used the VMware Workstation virtual machine and opened the malware machine in VMware.

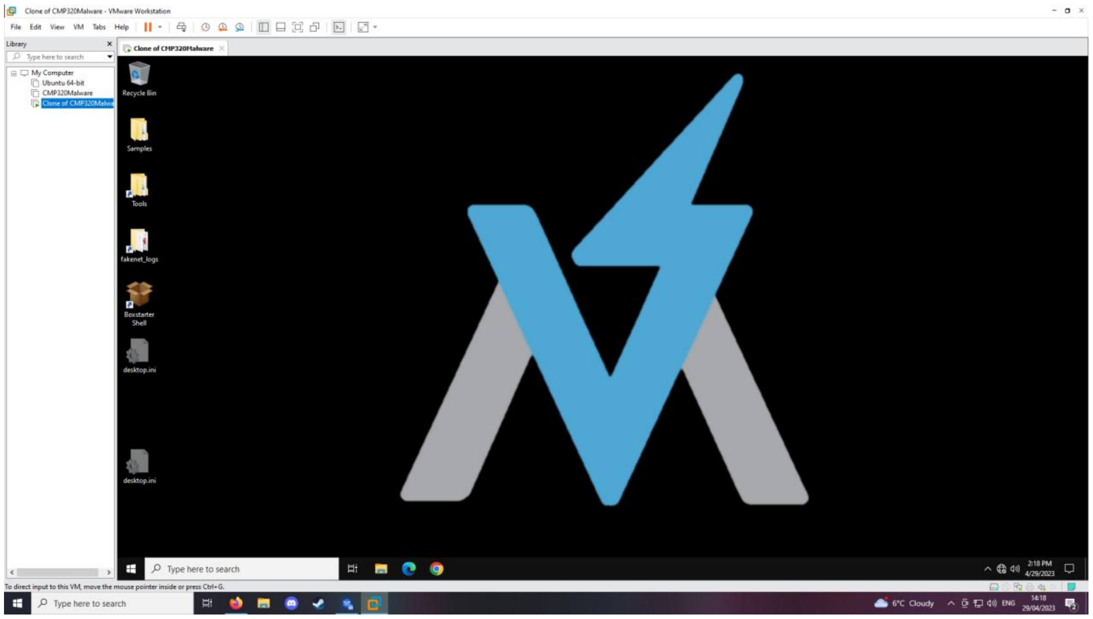


Figure 3: Malware virtual machine.

The malware analyst made sure to take all necessary precautions when performing this task. Clean boot snapshots have been used to ensure the virtual machine can be reverted to its original state. This was especially useful for dynamic analysis.

<u>Section Overview</u>

- Static Analysis
- Dynamic Analysis

Once the environment was all setup and ready to go, the malware analyst selected a malware sample for analysis. The first stage in malware analysis was static analysis of the target sample. This section of the report focused on passively examining the malware sample with the intent to find as much information about it without running the malware sample.

The dynamic analysis section built upon the previous section of static analysis. Dynamic analysis focused on running the executable file and observing what the target malware sample does. This section gave the malware analyst a better idea of the functionality and type of malware that the sample is.

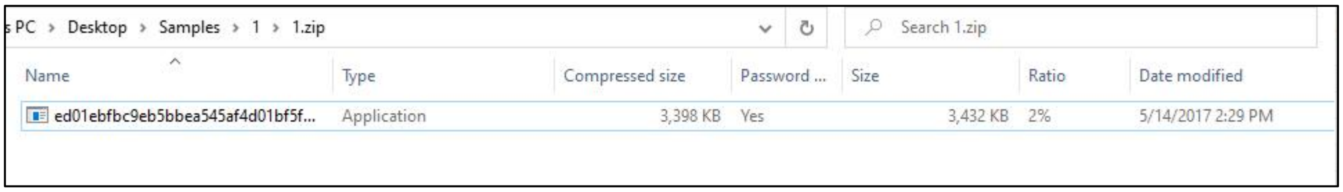
<u>Tools used during analysis of malware sample:</u>

- VMware Workstation
- CMP320 Malware Virtual Machine
- VirusTotal - Useful for file has comparison of other malware signatures.
- HashMyFiles - Useful for obtaining the hash of files.
- Strings - Useful for static analysis of strings found in the malware file.
- Pestudio - Useful for header analysis
- Peid - Software to check for software used to pack the malware payload.
- CFF Explorer - Check for compilation and modification time stamps.
- Dependency Walker - Component investigation.
- Regshot-x64-Unicode - Snapshots of registry.
- Process Monitor - Monitor for running processes.
- Task Explorer - Allows for individual task analysis.
- UPX - check for packed files.

2.2 STATIC MALWARE ANALYSIS

The first part of the procedure featured static analysis. This malware analysis technique focused on passive analysis, meaning that the malware sample has not been ran and tested at this stage in the report.

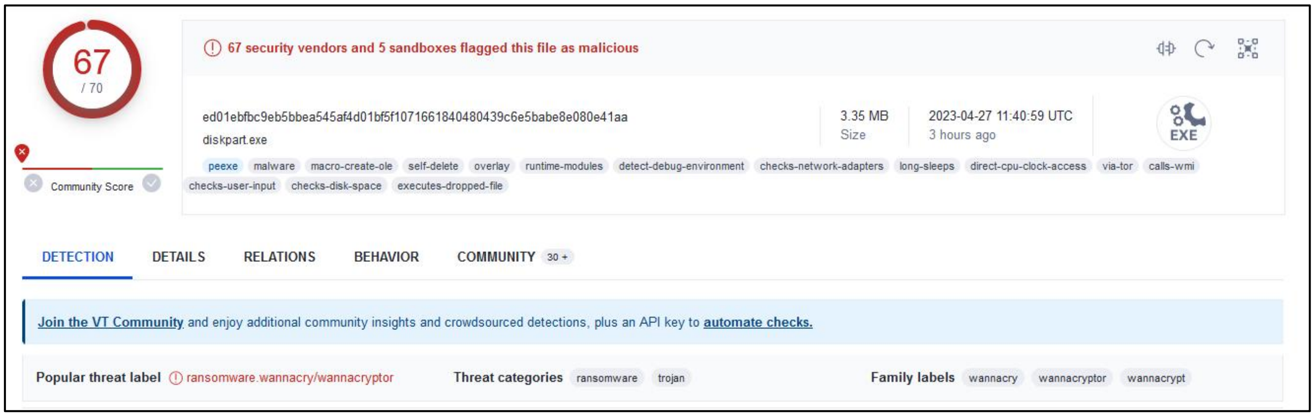
Upon discovery of the malware sample, the analyst found a single .exe file named, "ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe"**.**


s PC › Desktop › Samples › 1 › 1.zip

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
ed01ebfbc9eb5bbea545af4d01bf5f...	Application	3,398 KB	Yes	3,432 KB	2%	5/14/2017 2:29 PM

Figure 4: Malware Sample Folder

The executable file has been hashed using ‘HashMyFiles’ tool and uploaded to virustotal.com to compare this hash to existing malware hashes. Doing this, the malware analyst got a good indication on what type of malware they were dealing with.



67 / 70

Community Score

67 security vendors and 5 sandboxes flagged this file as malicious

ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa
diskpart.exe

3.35 MB
Size

2023-04-27 11:40:59 UTC
3 hours ago

EXE

peexe · malware · macro-create-ole · self-delete · overlay · runtime-modules · detect-debug-environment · checks-network-adapters · long-sleeps · direct-cpu-clock-access · via-tor · calls-wmi · checks-user-input · checks-disk-space · executes-dropped-file

DETECTION · DETAILS · RELATIONS · BEHAVIOR · COMMUNITY 30 +

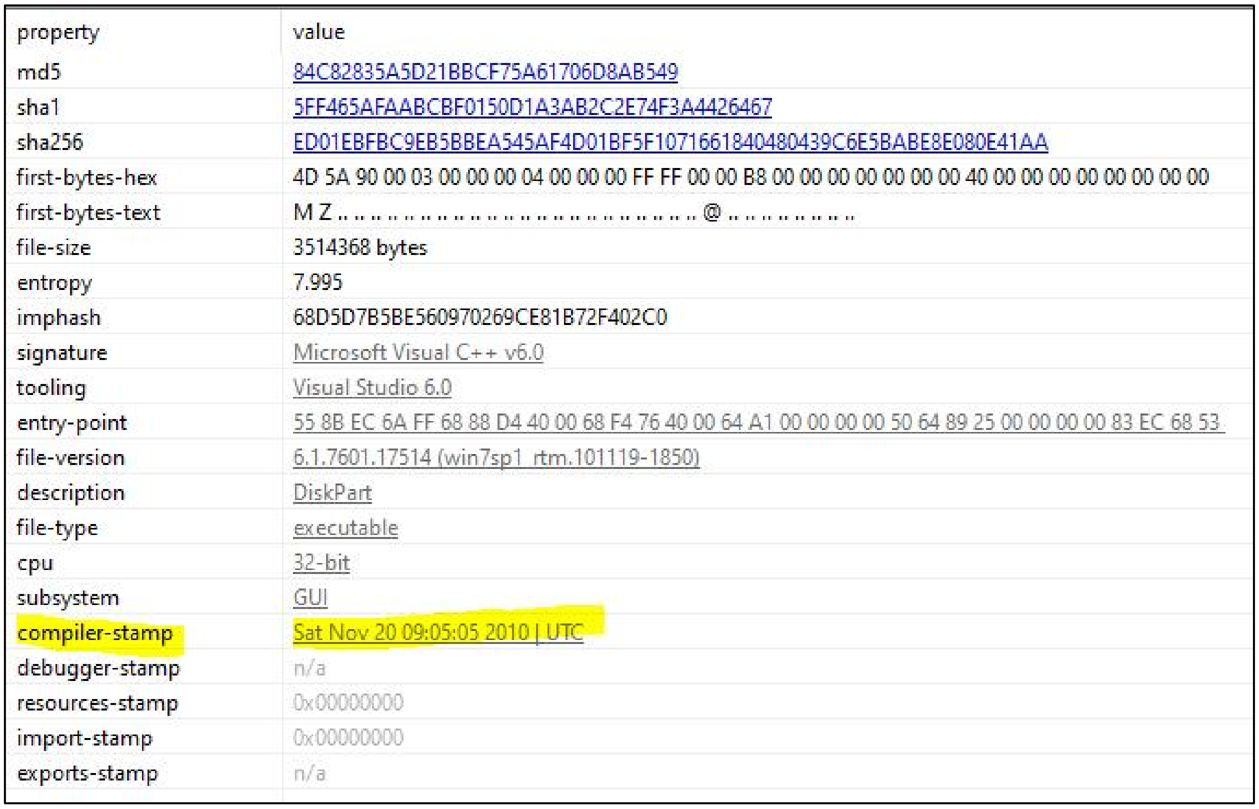
Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label ransomware.wannacry/wannacryptor · Threat categories ransomware trojan · Family labels wannacry wannacryptor wannacrypt

Figure 5: Hash upload on virustotal.

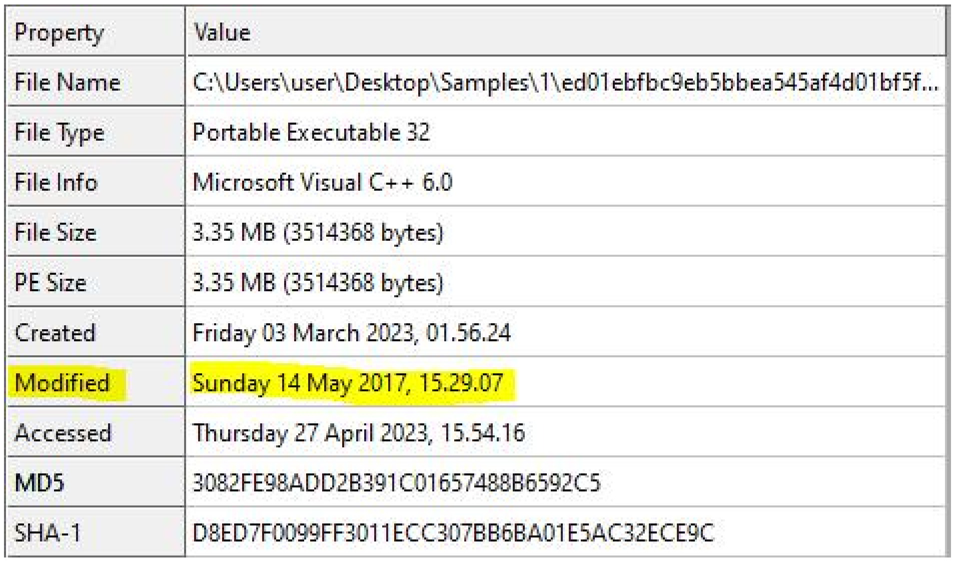
As seen by the figure above, virustotal.com flagged this malware sample as malicious. This malware has been given the threat categories of ransomeware and trojan with the label of ‘wannacry’.

The next step the malware analyst took was to determine the complication and modification dates of the malware sample. This was achieved using CFF Explorer as well as pestudio. Pestudio has been used to find the compiler-stamp and CFF Explorer to find the modification date.



property	value
md5	84C82835A5D21BBCF75A61706D8AB549
sha1	5FF465AFAABCBF0150D1A3AB2C2E74F3A4426467
sha256	ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA
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
first-bytes-text	M Z .. .. @ .. ..
file-size	3514368 bytes
entropy	7.995
imphash	68D5D7B5BE560970269CE81B72F402C0
signature	Microsoft Visual C++ v6.0
tooling	Visual Studio 6.0
entry-point	55 8B EC 6A FF 68 88 D4 40 00 68 F4 76 40 00 64 A1 00 00 00 00 50 64 89 25 00 00 00 00 83 EC 68 53
file-version	6.1.7601.17514 (win7sp1_rtm.101119-1850)
description	DiskPart
file-type	executable
cpu	32-bit
subsystem	GUI
compiler-stamp	Sat Nov 20 09:05:05 2010 | UTC
debugger-stamp	n/a
resources-stamp	0x00000000
import-stamp	0x00000000
exports-stamp	n/a

Figure 6: Compiler-stamp using pestudio.

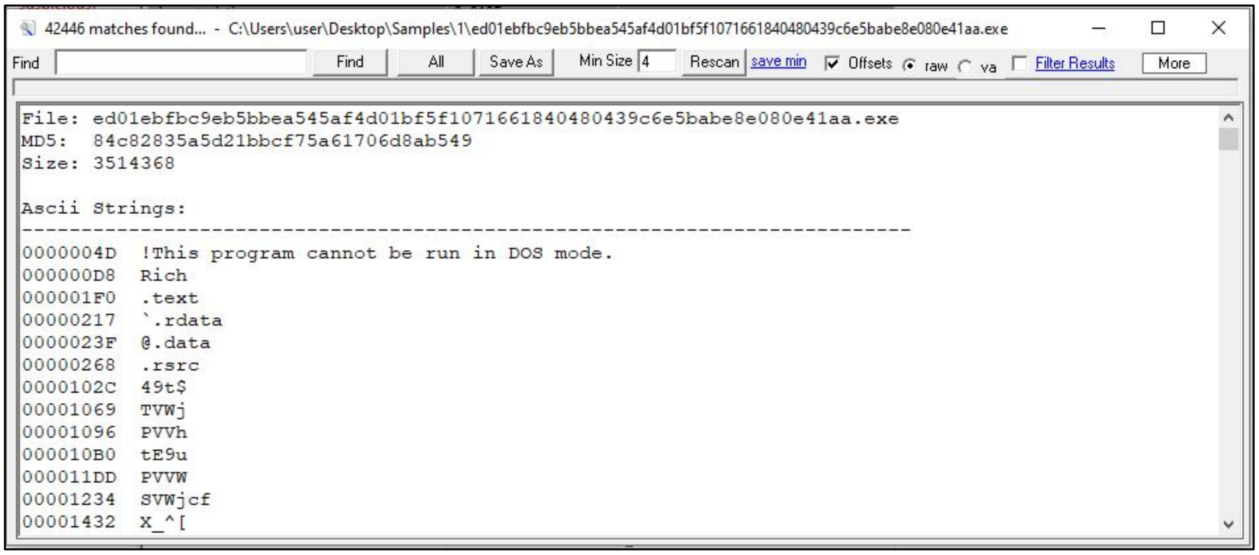


Property	Value
File Name	C:\Users\user\Desktop\Samples\1\ed01ebfbc9eb5bbea545af4d01bf5f...
File Type	Portable Executable 32
File Info	Microsoft Visual C++ 6.0
File Size	3.35 MB (3514368 bytes)
PE Size	3.35 MB (3514368 bytes)
Created	Friday 03 March 2023, 01.56.24
Modified	Sunday 14 May 2017, 15.29.07
Accessed	Thursday 27 April 2023, 15.54.16
MD5	3082FE98ADD2B391C01657488B6592C5
SHA-1	D8ED7F0099FF3011ECC307BB6BA01E5AC32ECE9C

Figure 7: Modification date using CFF Explorer.

Using pestudio and CFF Explorer, the malware analyst found that the .exe file has a compilation stamp with the date 20th of November 2010 and has been modified in the 14th of May 2017.

The next step the malware analyst took was to run a string search on the target application file to search for common or specific strings in the source. This analysis technique can be useful to look for any clues on the functionality of the malware payload.

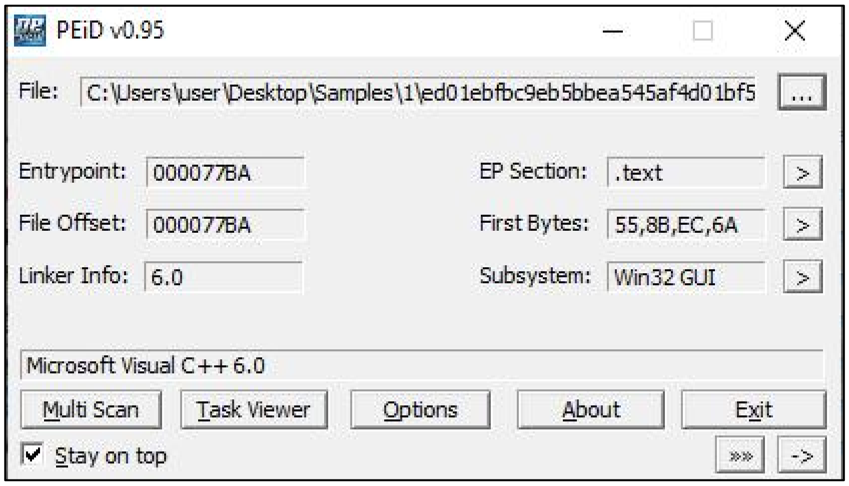

42446 matches found... - C:\Users\user\Desktop\Samples\1\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe
Find
Find
All
Save As
Min Size 4
Rescan
save min
Offsets
raw
va
Filter Results
More

```
File: ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe
MD5:  84c82835a5d21bbcf75a61706d8ab549
Size: 3514368

Ascii Strings:
---------------------------------------------------------------------------
0000004D  !This program cannot be run in DOS mode.
000000D8  Rich
000001F0  .text
00000217  `.rdata
0000023F  @.data
00000268  .rsrc
0000102C  49t$
00001069  TVWj
00001096  PVVh
000010B0  tE9u
000011DD  PVVW
00001234  SVWjcf
00001432  X_^[
```

Figure 8: String search used on source.

Running a string search on the target file has given us '42446' matches. This can be used and filtered through to get an idea of the file functionality and extra information about the executable file. Running strings and getting a large body of text with over forty-thousand results has given the malware analyst an indication that the file may not be packed/compressed. The malware analyst has attempted to run upx on the target exe file. Upx software allows the user to pack and unpack files. This can also be used to determine whether the file the analyst is investigating, actually packed or not.



File: C:\Users\user\Desktop\Samples\1\ed01ebfbc9eb5bbea545af4d01bf5

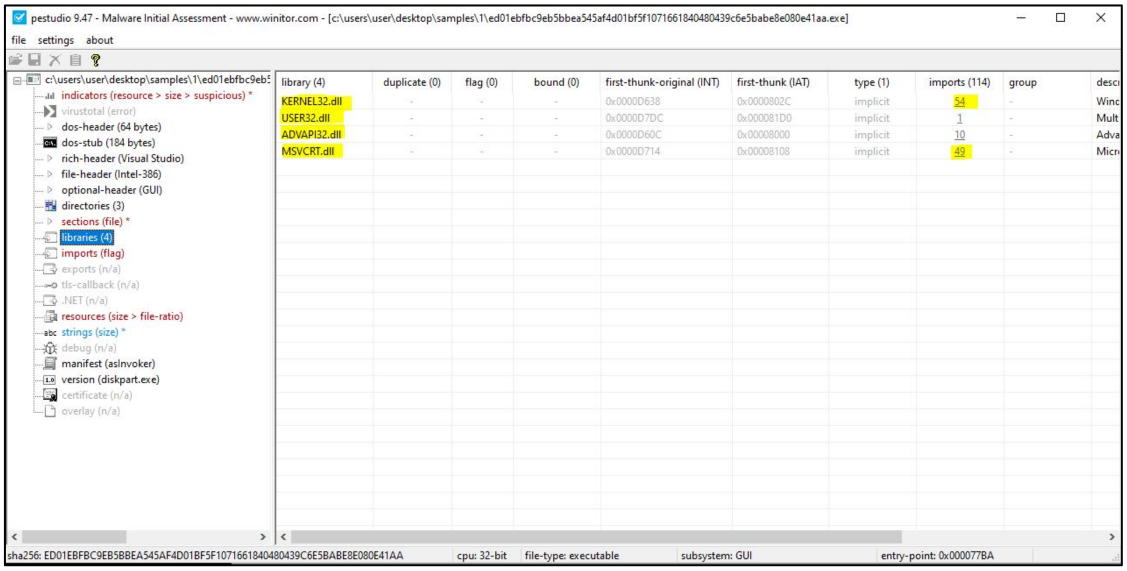
Entrypoint: 000077BA
EP Section: .text
File Offset: 000077BA
First Bytes: 55,8B,EC,6A
Linker Info: 6.0
Subsystem: Win32 GUI

Microsoft Visual C++ 6.0

Figure 9: File scanned using PEid.

The malware analyst used a tool called 'PEid' to scan the executable file to potentially find out the software used to pack the malware sample. In this case, the malware analyst used this tool to double check whether the sample has been previously packed. As seen by figure 9, This confirmed that this malware sample was not packed. This made the analysts job easier as there was no need to go through the unpacking process to be able to thoroughly analyse the sample.

The malware analyst then scanned the executable file in a tool called 'PeStudio' to find relevant information about the libraries in use, imports and functionality of the malware payload. Discovering the libraries gave insight into what DLL's are used with the malware payload.


pestudio 9.47 - Malware Initial Assessment - www.winitor.com - [c:\users\user\desktop\samples\1\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

```
c:\users\user\desktop\samples\1\ed01ebfbc9eb5
  indicators (resource > size > suspicious) *
  virustotal (error)
  dos-header (64 bytes)
  dos-stub (184 bytes)
  rich-header (Visual Studio)
  file-header (Intel-386)
  optional-header (GUI)
  directories (3)
  sections (file) *
  libraries (4)
  imports (flag)
  exports (n/a)
  tls-callback (n/a)
  .NET (n/a)
  resources (size > file-ratio)
  strings (size) *
  debug (n/a)
  manifest (asInvoker)
  version (diskpart.exe)
  certificate (n/a)
  overlay (n/a)
```

library (4)	duplicate (0)	flag (0)	bound (0)	first-thunk-original (INT)	first-thunk (IAT)	type (1)	imports (114)	group	desc
KERNEL32.dll	-	-	-	0x0000D638	0x0000802C	implicit	54	-	Winc
USER32.dll	-	-	-	0x0000D7DC	0x000081D0	implicit	1	-	Mult
ADVAPI32.dll	-	-	-	0x0000D60C	0x00008000	implicit	10	-	Adva
MSVCRT.dll	-	-	-	0x0000D714	0x00008108	implicit	49	-	Micr

sha256: ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA | cpu: 32-bit | file-type: executable | subsystem: GUI | entry-point: 0x000077BA

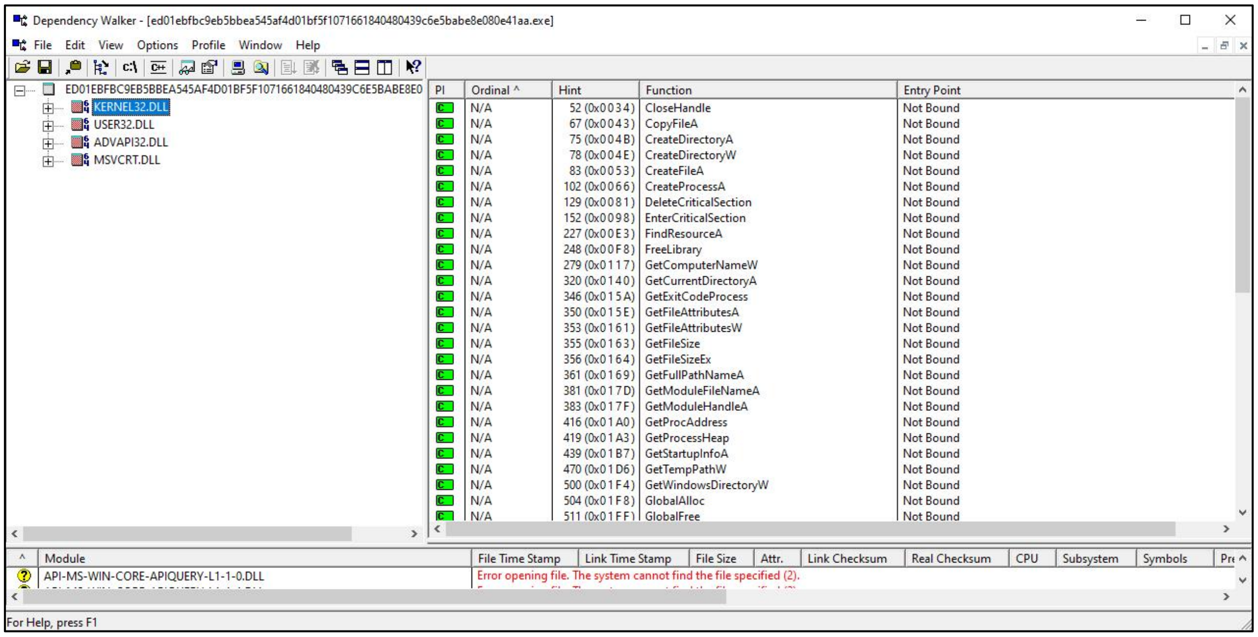
Figure 10: DLL library scan using PEStudio.

From figure 10, the analyst found that the executable file used four different DLL files on run-time. These DLL's are 'KERNAL32.dll', 'USER32.dll', ADVAPI32.dll', and 'MSVCRT.dll'. The imports column of the libraries show that KERNAL32.dll has 54 imports and MSVCRT.dll has 49 imports.

<u>DLL'S & Usage:</u>

KERNAL32.dll - Core functionality. Allows manipulation of files, memory and hardware.
USER32.dll - UI components for controlling and responding to user actions.
ADVAPI.dll - Provides access to service manager and registry.
MSVCRT.dll - Standard C library for visual C++ compiler.

The next step was to use 'Dependency Walker' to look at the function imported by the program through the dll's. Full DLL function imports can be found in Appendix A - DLL Import Functions.



Dependency Walker - [ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E0
- KERNEL32.DLL
- USER32.DLL
- ADVAPI32.DLL
- MSVCRT.DLL

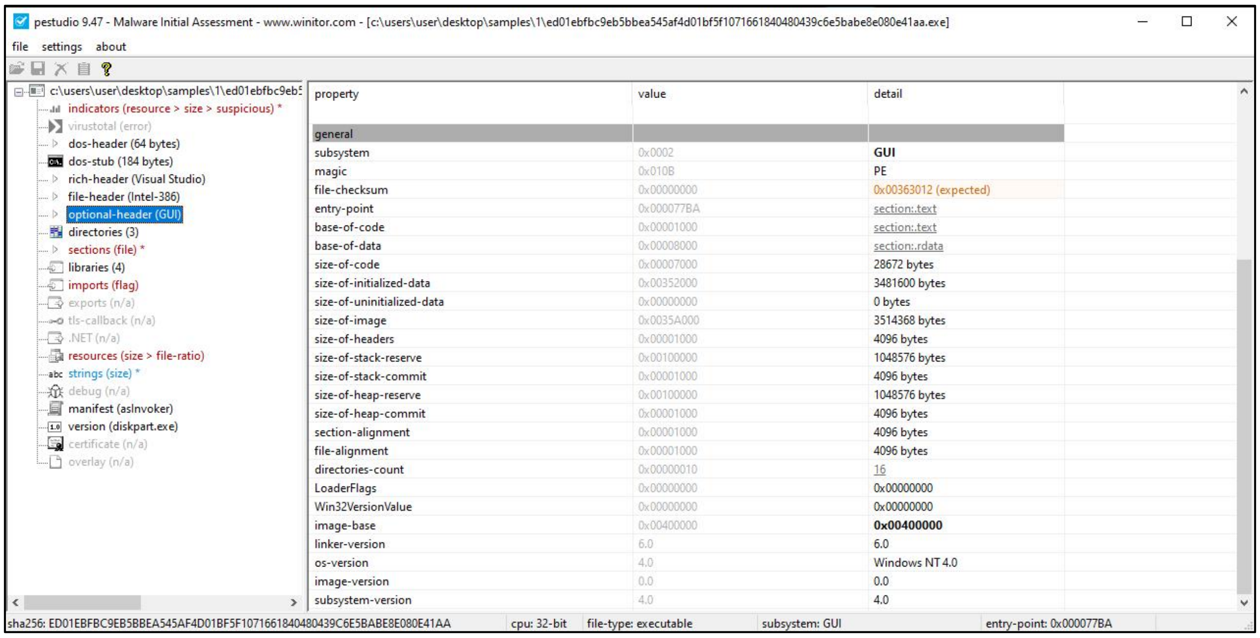
PI	Ordinal ^	Hint	Function	Entry Point
🟩	N/A	52 (0x0034)	CloseHandle	Not Bound
🟩	N/A	67 (0x0043)	CopyFileA	Not Bound
🟩	N/A	75 (0x004B)	CreateDirectoryA	Not Bound
🟩	N/A	78 (0x004E)	CreateDirectoryW	Not Bound
🟩	N/A	83 (0x0053)	CreateFileA	Not Bound
🟩	N/A	102 (0x0066)	CreateProcessA	Not Bound
🟩	N/A	129 (0x0081)	DeleteCriticalSection	Not Bound
🟩	N/A	152 (0x0098)	EnterCriticalSection	Not Bound
🟩	N/A	227 (0x00E3)	FindResourceA	Not Bound
🟩	N/A	248 (0x00F8)	FreeLibrary	Not Bound
🟩	N/A	279 (0x0117)	GetComputerNameW	Not Bound
🟩	N/A	320 (0x0140)	GetCurrentDirectoryA	Not Bound
🟩	N/A	346 (0x015A)	GetExitCodeProcess	Not Bound
🟩	N/A	350 (0x015E)	GetFileAttributesA	Not Bound
🟩	N/A	353 (0x0161)	GetFileAttributesW	Not Bound
🟩	N/A	355 (0x0163)	GetFileSize	Not Bound
🟩	N/A	356 (0x0164)	GetFileSizeEx	Not Bound
🟩	N/A	361 (0x0169)	GetFullPathNameA	Not Bound
🟩	N/A	381 (0x017D)	GetModuleFileNameA	Not Bound
🟩	N/A	383 (0x017F)	GetModuleHandleA	Not Bound
🟩	N/A	416 (0x01A0)	GetProcAddress	Not Bound
🟩	N/A	419 (0x01A3)	GetProcessHeap	Not Bound
🟩	N/A	439 (0x01B7)	GetStartupInfoA	Not Bound
🟩	N/A	470 (0x01D6)	GetTempPathW	Not Bound
🟩	N/A	500 (0x01F4)	GetWindowsDirectoryW	Not Bound
🟩	N/A	504 (0x01F8)	GlobalAlloc	Not Bound
🟩	N/A	511 (0x01FF)	GlobalFree	Not Bound

Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symbols
API-MS-WIN-CORE-APIQUERY-L1-1-0.DLL	Error opening file. The system cannot find the file specified (2).								

For Help, press F1

Figure 11: Dependency Walker used to find imported functions.

The next step the malware analyst took was to inspect the headers in 'PeStudio' to gain additional information on how the malware sample functions. Investigating headers gave the analyst useful insight whether or not the malware sample is GUI or command line based. This provides additional information on how the malware sample functions.



pestudio 9.47 - Malware Initial Assessment - www.winitor.com - [c:\users\user\desktop\samples\1\ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa.exe]

- c:\users\user\desktop\samples\1\ed01ebfbc9eb5
  - indicators (resource > size > suspicious) *
  - virustotal (error)
  - dos-header (64 bytes)
  - dos-stub (184 bytes)
  - rich-header (Visual Studio)
  - file-header (Intel-386)
  - optional-header (GUI)
  - directories (3)
  - sections (file) *
  - libraries (4)
  - imports (flag)
  - exports (n/a)
  - tls-callback (n/a)
  - .NET (n/a)
  - resources (size > file-ratio)
  - strings (size) *
  - debug (n/a)
  - manifest (asInvoker)
  - version (diskpart.exe)
  - certificate (n/a)
  - overlay (n/a)

property	value	detail
general		
subsystem	0x0002	GUI
magic	0x010B	PE
file-checksum	0x00000000	0x00363012 (expected)
entry-point	0x000077BA	section:.text
base-of-code	0x00001000	section:.text
base-of-data	0x00008000	section:.rdata
size-of-code	0x00007000	28672 bytes
size-of-initialized-data	0x00352000	3481600 bytes
size-of-uninitialized-data	0x00000000	0 bytes
size-of-image	0x0035A000	3514368 bytes
size-of-headers	0x00001000	4096 bytes
size-of-stack-reserve	0x00100000	1048576 bytes
size-of-stack-commit	0x00001000	4096 bytes
size-of-heap-reserve	0x00100000	1048576 bytes
size-of-heap-commit	0x00001000	4096 bytes
section-alignment	0x00001000	4096 bytes
file-alignment	0x00001000	4096 bytes
directories-count	0x00000010	16
LoaderFlags	0x00000000	0x00000000
Win32VersionValue	0x00000000	0x00000000
image-base	0x00400000	0x00400000
linker-version	6.0	6.0
os-version	4.0	Windows NT 4.0
image-version	0.0	0.0
subsystem-version	4.0	4.0

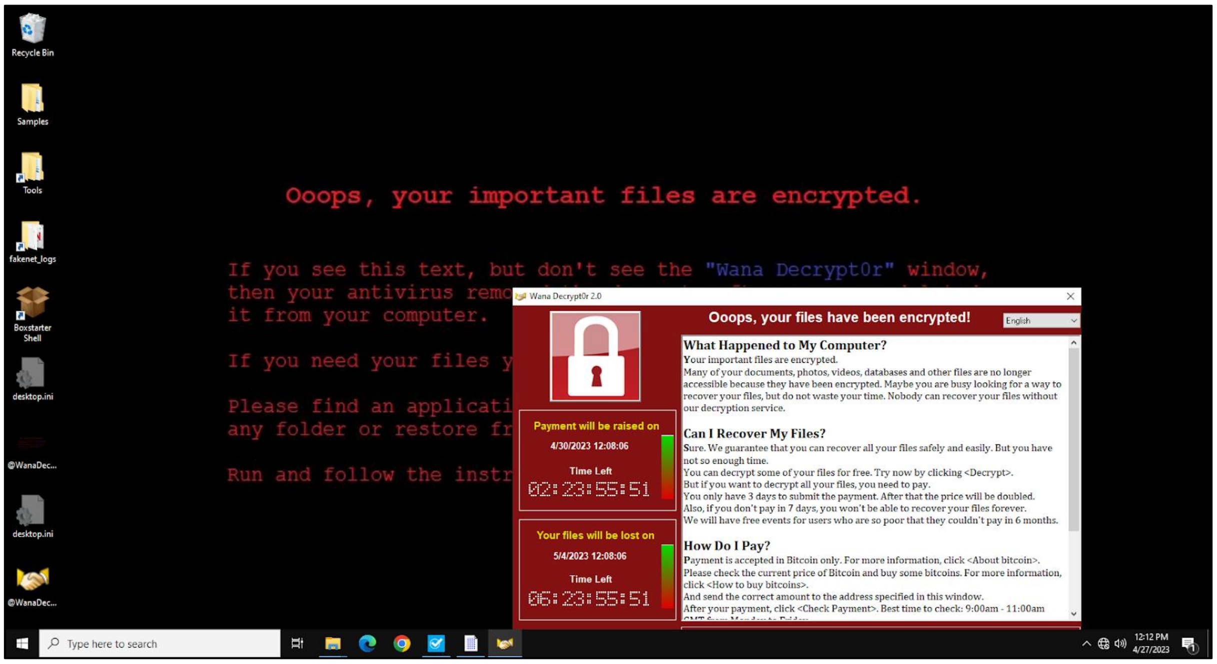
sha256: ED01EBFBC9EB5BBEA545AF4D01BF5F1071661840480439C6E5BABE8E080E41AA cpu: 32-bit file-type: executable subsystem: GUI entry-point: 0x000077BA

Figure 12: Header inspected using PeStudio.

From this information, the malware analyst can determine that the malware sample is GUI based and runs GUI elements on run-time.

2.3 Dynamic Malware Analysis

In the next stage in the procedure, the malware analyst had to dynamically analyse the malware sample. As mentioned previously, this stage is necessary to figure out the full functionality of the malware sample and its intent.



Ooops, your important files are encrypted.

If you see this text, but don't see the "Wana Decrypt0r" window,
then your antivirus remo
it from your computer.

If you need your files y

Please find an applicati
any folder or restore fr

Run and follow the instr

Wana Decrypt0r 2.0

Ooops, your files have been encrypted!

English

Payment will be raised on
4/30/2023 12:08:06
Time Left
02:23:55:51

Your files will be lost on
5/4/2023 12:08:06
Time Left
06:23:55:51

What Happened to My Computer?
Your important files are encrypted.
Many of your documents, photos, videos, databases and other files are no longer accessible because they have been encrypted. Maybe you are busy looking for a way to recover your files, but do not waste your time. Nobody can recover your files without our decryption service.

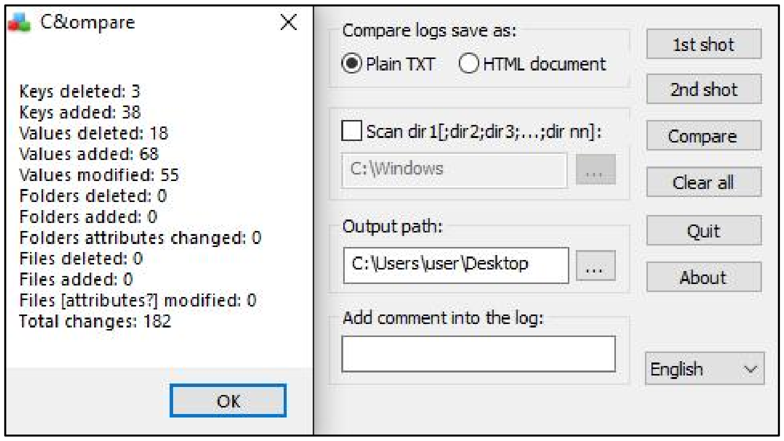
Can I Recover My Files?
Sure. We guarantee that you can recover all your files safely and easily. But you have not so enough time.
You can decrypt some of your files for free. Try now by clicking <Decrypt>.
But if you want to decrypt all your files, you need to pay.
You only have 3 days to submit the payment. After that the price will be doubled.
Also, if you don't pay in 7 days, you won't be able to recover your files forever.
We will have free events for users who are so poor that they couldn't pay in 6 months.

How Do I Pay?
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment, click <Check Payment>. Best time to check: 9:00am - 11:00am

Figure 13: Malware sample executed.

The malware analyst ran the malware sample executable and the malware payload executed a ransom-ware attack on the virtual machine. This can be seen in figure 13. Once the malware analyst completed this and got evidence of the type of malware the sample was, the virtual machine was reverted to its clean state.

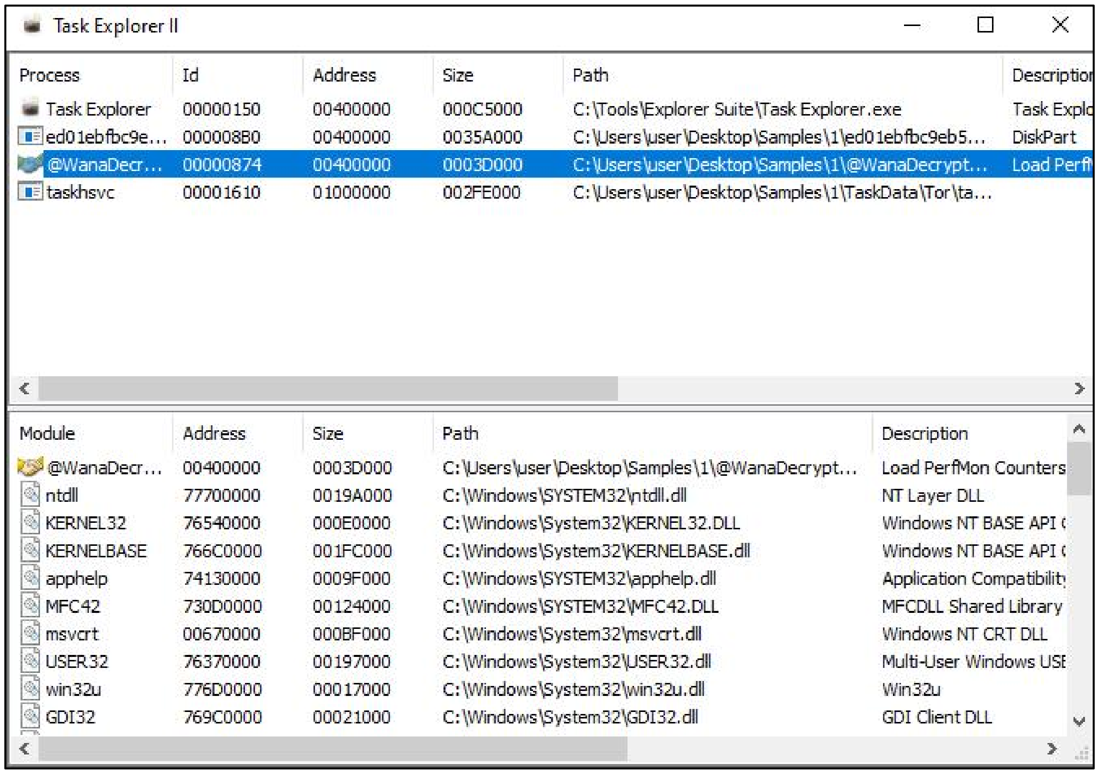
The next step the malware analyst has taken was to analyse the process from before and after the malware sample was executed. For this, the analyst used Regshot to take snapshots of the machine's state both before and after the malicious software was ran. The analyst also used a Process Monitor to inspect what the malware sample is executing on run-time. This information has been used as evidence. This can be seen in figure 14.


```
C&compare

Keys deleted: 3
Keys added: 38
Values deleted: 18
Values added: 68
Values modified: 55
Folders deleted: 0
Folders added: 0
Folders attributes changed: 0
Files deleted: 0
Files added: 0
Files [attributes?] modified: 0
Total changes: 182
```

Figure 14: Snapshots compared.

Two snapshots have been created. One before executing the malware sample and one after. These two snapshots have been saved as ,'regshot1.hivu', and ,'regshot2.hivu'. After that was complete, both regshots have been compared to give the analyst an output.

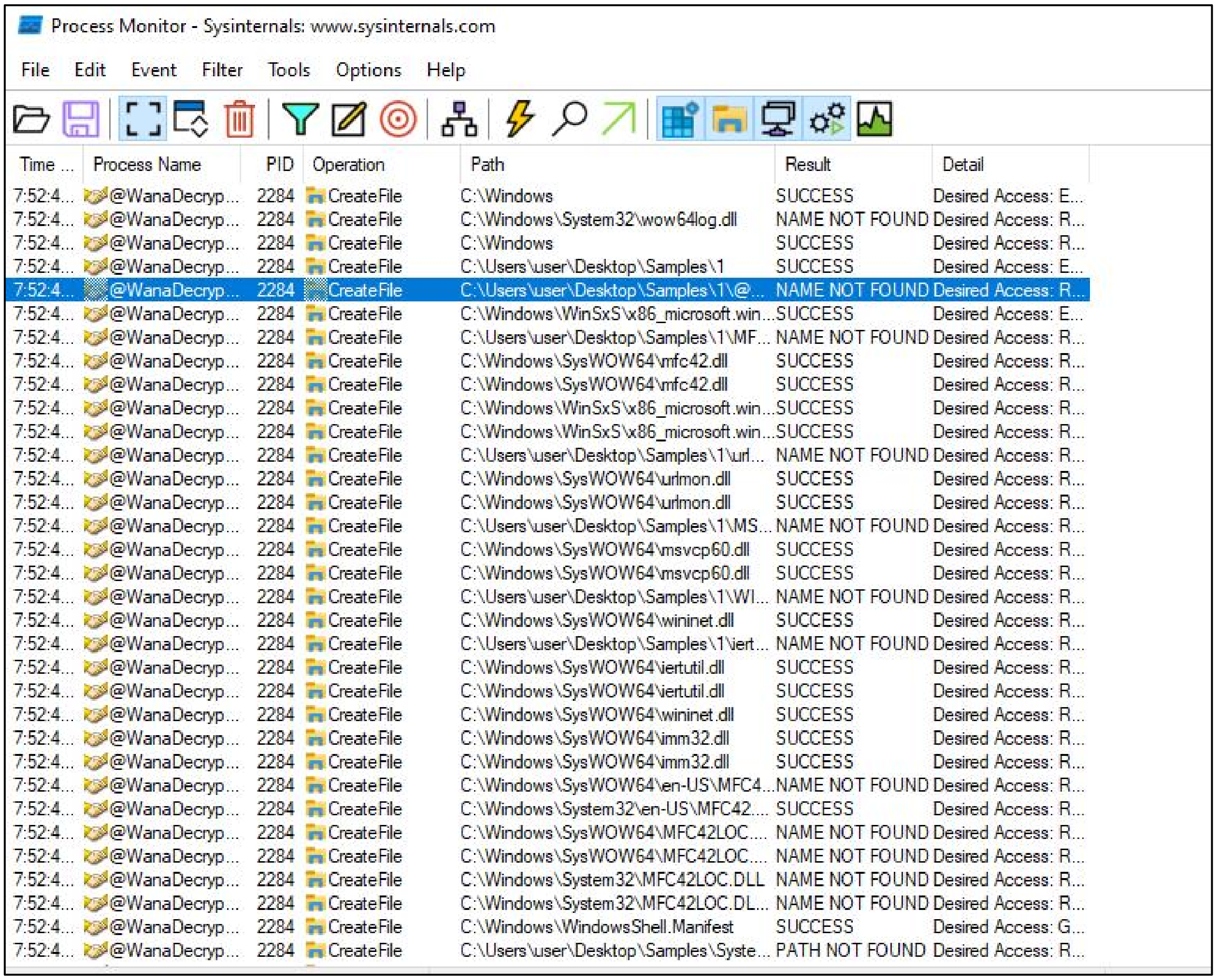

Task Explorer II

Process	Id	Address	Size	Path	Description
Task Explorer	00000150	00400000	000C5000	C:\Tools\Explorer Suite\Task Explorer.exe	Task Explo
ed01ebfbc9e...	000008B0	00400000	0035A000	C:\Users\user\Desktop\Samples\1\ed01ebfbc9eb5...	DiskPart
@WanaDecr...	00000874	00400000	0003D000	C:\Users\user\Desktop\Samples\1\@WanaDecrypt...	Load Perfi
taskhsvc	00001610	01000000	002FE000	C:\Users\user\Desktop\Samples\1\TaskData\Tor\ta...	

Module	Address	Size	Path	Description
@WanaDecr...	00400000	0003D000	C:\Users\user\Desktop\Samples\1\@WanaDecrypt...	Load PerfMon Counters
ntdll	77700000	0019A000	C:\Windows\SYSTEM32\ntdll.dll	NT Layer DLL
KERNEL32	76540000	000E0000	C:\Windows\System32\KERNEL32.DLL	Windows NT BASE API (
KERNELBASE	766C0000	001FC000	C:\Windows\System32\KERNELBASE.dll	Windows NT BASE API (
apphelp	74130000	0009F000	C:\Windows\SYSTEM32\apphelp.dll	Application Compatibility
MFC42	730D0000	00124000	C:\Windows\SYSTEM32\MFC42.DLL	MFCDLL Shared Library
msvcrt	00670000	000BF000	C:\Windows\System32\msvcrt.dll	Windows NT CRT DLL
USER32	76370000	00197000	C:\Windows\System32\USER32.dll	Multi-User Windows USE
win32u	776D0000	00017000	C:\Windows\System32\win32u.dll	Win32u
GDI32	769C0000	00021000	C:\Windows\System32\GDI32.dll	GDI Client DLL

Figure 15: Task explorer running during execution.

Task explorer was a useful tool to monitor what tasks the malware sample has executed during run-time. From the above figure, the malware sample executes its executable file as well as another executable file named, 'taskhsvc'. Another task was run named @WanaDecryptor which was the GUI based application the malware sample was running.


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

Time ...	Process Name	PID	Operation	Path	Result	Detail
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows	SUCCESS	Desired Access: E...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\1	SUCCESS	Desired Access: E...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\1\@...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\WinSxS\x86_microsoft.win...	SUCCESS	Desired Access: E...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\1\MF...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\mfc42.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\mfc42.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\WinSxS\x86_microsoft.win...	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\WinSxS\x86_microsoft.win...	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\1\url...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\urlmon.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\urlmon.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\1\MS...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\msvcp60.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\msvcp60.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\1\WI...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\wininet.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\1\iert...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\iertutil.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\iertutil.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\wininet.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\imm32.dll	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\en-US\MFC4...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\System32\en-US\MFC42....	SUCCESS	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\MFC42LOC....	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\SysWOW64\MFC42LOC....	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\System32\MFC42LOC.DLL	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\System32\MFC42LOC.DL...	NAME NOT FOUND	Desired Access: R...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Windows\WindowsShell.Manifest	SUCCESS	Desired Access: G...
7:52:4...	@WanaDecryp...	2284	CreateFile	C:\Users\user\Desktop\Samples\Syste...	PATH NOT FOUND	Desired Access: R...

Figure 16: Process monitor running during execution.

The process monitor tool allowed the malware analyst to inspect and analyse the processes being run by the malware sample at run-time. Appropriate filters have been set to monitor the target malware sample. This tool was useful as each event could have been double clicked and further inspected by the malware analyst.

3 DISCUSSION

3.1 GENERAL DISCUSSION

Throughout the procedure, the malware analyst found valuable information on the target malware sample. Based on static and dynamic analysis techniques, the analyst determined that the malware sample in question is ransom-ware software. The intention of this type of malware software is to force infected users to pay money for their data stored on their computer system to be decrypted and not deleted; blackmailing the victim.

The analyst looked further into the type of malware throughout the procedure stage of this report. The procedure features many indications as to what type the malware sample is as well as specifically what kind of ransom-ware software it is. This indicators were the virus-total hash upload which returned with labels of ransom-ware, Trojan and most importantly, “wanaCryptor”. The modification stamps show the year 2017. This specific malware has been identified as the ransom-ware attack called WannaCry which was a ransom-ware attack that disrupted banks, hospitals and communication companies around the globe in 2017.

When executed, this ransom-ware attack first searches the memory for any important files such as videos, images, sound files and more. The malware software then runs an encryption on those files that deny any access to them from the user. These files can only be decrypted using a digital key. The malicious software then executes a GUI application on the target’s desktop which informs the user that they’ve been infected and told to pay a ransom for their files with instructions. There is a timer running on the application with a time constraint. This forces the user to pay before the specified time. Otherwise, the data will be deleted.

The malware sample features several components. This has been analyzed during the static and dynamic analysis. The WanaCryptor uses four DLL imports that helps it execute on the target system. These can be seen in the list below as well as all the imported functions in Appendix A - DLL Import Functions.

DLL’S & Usage:
KERNAL32.dll - Core functionality. Allows manipulation of files, memory and hardware.
USER32.dll - UI components for controlling and responding to user actions.
ADVAPI.dll - Provides access to service manager and registry.
MSVCRT.dll - Standard C library for visual C++ compiler. This library has been used to execute c++ code onto the target system.

The aims of the report have been met by the malware analyst. Throughout the procedure, the analyst managed to analyse the target malware sample using static and dynamic analysis techniques.
The malware sample has been identified in its type, characteristics, components and core functionality by the malware analyst. A discussion of the results and findings have been provided by the analyst as part of this report. The countermeasures for this malware sample will be featured in the next section of the discussion - 3.2 Countermeasures.

3.2 COUNTERMEASURES

It is important to understand how to keep yourself protected from malware attacks on a day to day basic. There is a number of actions people can take to increase their security and minimize the chances of getting infected with any malware software.

First of all, ensure that you're running an anti-virus on your computer. This is crucial in attack prevention and useful for scanning the machine for any potential malware. Keep software up-to-date as much as possible. Especially the operating system of the machine. Updates are there for a reason. Often to patch any security vulnerabilities and bugs.

In cyber-security, it is said that one of the main vulnerabilities are people. The use of social engineering, phishing emails, suspicious links and more can lead to a whole organization to get infected by malware. It is recommended that all employees in an organization understand this and are trained to reduce the risk of a computer system(s) getting infected by malicious software.

This ransom-ware software encrypts and later deletes the data on the computer system if not paid. In order to eliminate the risk of important information getting lost, regular data back-ups are recommended.

3.3 FUTURE WORK

Given more time and resources, the malware analyst will perform additional analysis on the target malware sample. Given the safety precautions of the procedure, the network adapter on the malware virtual machine was set to host only meaning that the virtual machine was not connected to the internet and could not spread across it. Given the type and functionality of the malware sample investigated in this report, the analyst knows that this, 'wanaCryptor', malware software could potentially be a worm and could spread to other machines if not contained properly. This will be tested in a safe, controlled environment.

The malware analyst would reverse-engineer the code as much as possible. Implementing more advanced analysis techniques to figure out how the sample functions in more depth by analyzing the base code and using disassemblers to understand exactly how the malware executed and its functions step-by-step.

REFERENCES

-Avast. 2023. c-malware. [ONLINE] Available at: https://www.avast.com/c-malware. [Accessed 2 May 2023].

-Statista. 2023. malware-attacks-per-year-worldwide. [ONLINE] Available at: https://www.statista.com/statistics/873097/malware-attacks-per-year-worldwide/. [Accessed 2 May 2023].

- intezer. 2023. the-role-of-malware-analysis-in-cybersecurity. [ONLINE] Available at: https://www.intezer.com/blog/malware-analysis/the-role-of-malware-analysis-in-cybersecurity/. [Accessed 2 May 2023].

-minecast. 2023. all-you-need-to-know-about-wannacry-ransomware. [ONLINE] Available at: https://www.mimecast.com/blog/all-you-need-to-know-about-wannacry-ransomware/. [Accessed 2 May 2023].

APPENDICES

APPENDIX A - DLL IMPORT FUNCTIONS

KERNAL32.dll

```
CloseHandle
CopyFileA
CreateDirectoryA
CreateDirectoryW
CreateFileA
CreateProcessA
DeleteCriticalSection
EnterCriticalSection
FindResourceA
FreeLibrary
GetComputerNameW
GetCurrentDirectoryA
GetExitCodeProcess
GetFileAttributesA
GetFileAttributesW
GetFileSize
GetFileSizeEx
GetFullPathNameA
GetModuleFileNameA
GetModuleHandleA
GetProcAddress
GetProcessHeap
GetStartupInfoA
GetTempPathW
GetWindowsDirectoryW
GlobalAlloc
GlobalFree
HeapAlloc
HeapFree
InitializeCriticalSection
IsBadReadPtr
LeaveCriticalSection
LoadLibraryA
LoadResource
LocalFileTimeToFileTime
LockResource
MultiByteToWideChar
OpenMutexA
ReadFile
SetCurrentDirectoryA
SetCurrentDirectoryW
SetFileAttributesW
SetFilePointer
SetFileTime
SetLastError
SizeofResource
Sleep
SystemTimeToFileTime
TerminateProcess
VirtualAlloc
VirtualFree
VirtualProtect
WaitForSingleObject
WriteFile
```

ADVAPI32.dll

```
CloseServiceHandle
CreateServiceA
CryptReleaseContext
OpenSCManagerA
OpenServiceA
RegCloseKey
RegCreateKeyW
RegQueryValueExA
RegSetValueExA
StartServiceA
```

MSVCRT.dll

```
??0exception@@QAE@ABQBD@Z
??0exception@@QAE@ABV0@@Z
??1exception@@UAE@XZ
??1type_info@@UAE@XZ
??2@YAPAXI@Z
??3@YAXPAX@Z
_CxxThrowException
_XcptFilter
__CxxFrameHandler
__getmainargs
__p___argc
__p___argv
__p__commode
__p__fmode
__set_app_type
__setusermatherr
_acmdln
_adjust_fdiv
_controlfp
_except_handler3
_exit
_initterm
_local_unwind2
_mbsstr
_stricmp
calloc
exit
fclose
fopen
fread
free
fwrite
malloc
memcmp
memcpy
memset
rand
realloc
sprintf
srand
strcat
strcmp
strcpy
strlen
strrchr
swprintf
wcscat
wcslen
wcsrchr
```