Hands on Project

NotPetya

Introduction

NotPetya is a destructive ransomware that caused widespread damage and disruption across various organizations and industries in June 2017. It is considered one of the most damaging cyberattacks in history, affecting thousands of computers worldwide. This report provides an analysis of NotPetya, including its origins, impact, and mechanisms.

Acquisition of Virus File: The virus file used for analysis in this report was acquired from a GitHub repository maintained by NTFS123, a cybersecurity researcher. The repository, located at https://github.com/NTFS123/MalwareDatabase, contains samples of various malware, including NotPetya, for research and analysis purposes.

Configuration Information

To analyze the NotPetya malware, we utilized a controlled environment with the following configuration:

Virtualization Platform: Virtual Box

VM used : Windows

Analysis Tools: IDA Pro, Process Monitor, Process Hacker

Understanding of the Malware

NotPetya is a type of ransomware that spreads rapidly across networks by exploiting vulnerabilities in Microsoft Windows systems. It encrypts the Master Boot Record (MBR) and individual files on the infected system, rendering it inoperable until a ransom is paid. However, unlike traditional ransomware, NotPetya was designed to cause maximum disruption rather than generate ransom payments.

Understanding of the Malicious Mechanisms

- Infection Vector: NotPetya primarily spreads through the EternalBlue exploit, which targets a vulnerability in the Windows Server Message Block (SMB) protocol (CVE-2017-0144). It also utilizes the EternalRomance and EternalChampion exploits to propagate within networks.
- Propagation: Once inside a network, NotPetya uses legitimate Windows management tools such as PsExec and Windows Management Instrumentation (WMI) to move laterally and infect other machines. It also exploits weak or default credentials to access systems.

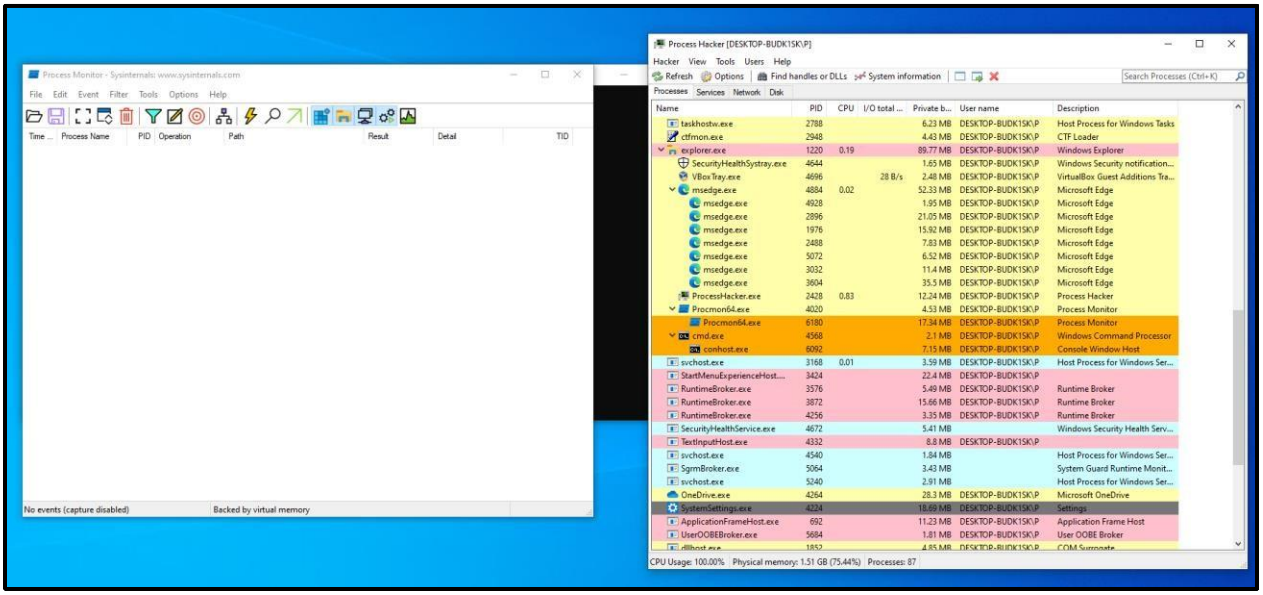
- Encryption: NotPetya encrypts the Master File Table (MFT) and overwrites the Master Boot Record (MBR) with a custom bootloader. It uses a combination of symmetric (AES-128) and asymmetric (RSA-2048) encryption algorithms to encrypt files.
- Payload: After encryption, NotPetya displays a ransom note demanding a Bitcoin payment in exchange for the decryption key. However, analysis suggests that the malware's encryption implementation is flawed, making decryption impossible even with the correct key.
- Destructive Component: NotPetya includes a destructive component that irreversibly damages the infected system. It overwrites the MBR, making it impossible to boot the system, and modifies the system's files and structures, resulting in permanent data loss.

Working

So first we downladed the virus file from the github repo that I have mentioned above in my windows VM.

Then I downladed the required tools.

Once the setup is done, I opened Prochacker tool and Procmon.


Process Monitor - Sysinternals: www.sysinternals.com
File Edit Event Filter Tools Options Help
Process Hacker [DESKTOP-BUDK1SK\P]
Hacker View Tools Users Help
Refresh
Options
Find handles or DLLs
System information
Search Processes (Ctrl+K)
Processes
Services
Network
Disk

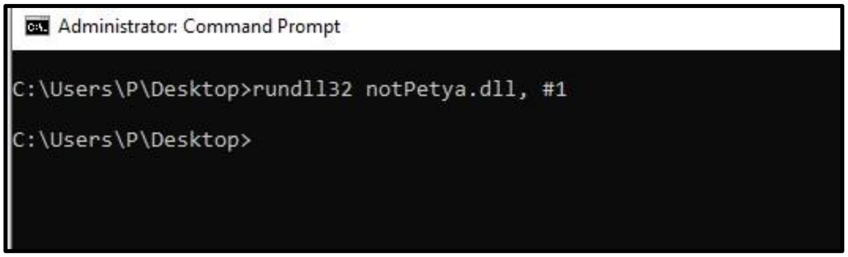
Time ...	Process Name	PID	Operation	Path	Result	Detail	TID

No events (capture disabled)   Backed by virtual memory

Name	PID	CPU	I/O total ...	Private b...	User name	Description
taskhostw.exe	2788			6.23 MB	DESKTOP-BUDK1SK\P	Host Process for Windows Tasks
ctfmon.exe	2948			4.43 MB	DESKTOP-BUDK1SK\P	CTF Loader
explorer.exe	1220	0.19		89.77 MB	DESKTOP-BUDK1SK\P	Windows Explorer
SecurityHealthSystray.exe	4644			1.65 MB	DESKTOP-BUDK1SK\P	Windows Security notification...
VBoxTray.exe	4696		28 B/s	2.48 MB	DESKTOP-BUDK1SK\P	VirtualBox Guest Additions Tra...
msedge.exe	4884	0.02		52.33 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
msedge.exe	4928			1.95 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
msedge.exe	2896			21.05 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
msedge.exe	1976			15.92 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
msedge.exe	2488			7.83 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
msedge.exe	5072			6.52 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
msedge.exe	3032			11.4 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
msedge.exe	3604			35.5 MB	DESKTOP-BUDK1SK\P	Microsoft Edge
ProcessHacker.exe	2428	0.83		12.24 MB	DESKTOP-BUDK1SK\P	Process Hacker
Procmon64.exe	4020			4.53 MB	DESKTOP-BUDK1SK\P	Process Monitor
Procmon64.exe	6180			17.34 MB	DESKTOP-BUDK1SK\P	Process Monitor
cmd.exe	4568			2.1 MB	DESKTOP-BUDK1SK\P	Windows Command Processor
conhost.exe	6092			7.15 MB	DESKTOP-BUDK1SK\P	Console Window Host
svchost.exe	3168	0.01		3.59 MB	DESKTOP-BUDK1SK\P	Host Process for Windows Ser...
StartMenuExperienceHost....	3424			22.4 MB	DESKTOP-BUDK1SK\P	
RuntimeBroker.exe	3576			5.49 MB	DESKTOP-BUDK1SK\P	Runtime Broker
RuntimeBroker.exe	3872			15.66 MB	DESKTOP-BUDK1SK\P	Runtime Broker
RuntimeBroker.exe	4256			3.35 MB	DESKTOP-BUDK1SK\P	Runtime Broker
SecurityHealthService.exe	4672			5.41 MB		Windows Security Health Serv...
TextInputHost.exe	4332			8.8 MB	DESKTOP-BUDK1SK\P	
svchost.exe	4540			1.84 MB		Host Process for Windows Ser...
SgrmBroker.exe	5064			3.43 MB		System Guard Runtime Monit...
svchost.exe	5240			2.91 MB		Host Process for Windows Ser...
OneDrive.exe	4264			28.3 MB	DESKTOP-BUDK1SK\P	Microsoft OneDrive
SystemSettings.exe	4224			18.69 MB	DESKTOP-BUDK1SK\P	Settings
ApplicationFrameHost.exe	692			11.23 MB	DESKTOP-BUDK1SK\P	Application Frame Host
UserOOBEBroker.exe	5684			1.81 MB	DESKTOP-BUDK1SK\P	User OOBE Broker
dllhost.exe	1852			4.85 MB	DESKTOP-BUDK1SK\P	COM Surrogate

CPU Usage: 100.00%   Physical memory: 1.51 GB (75.44%)   Processes: 87

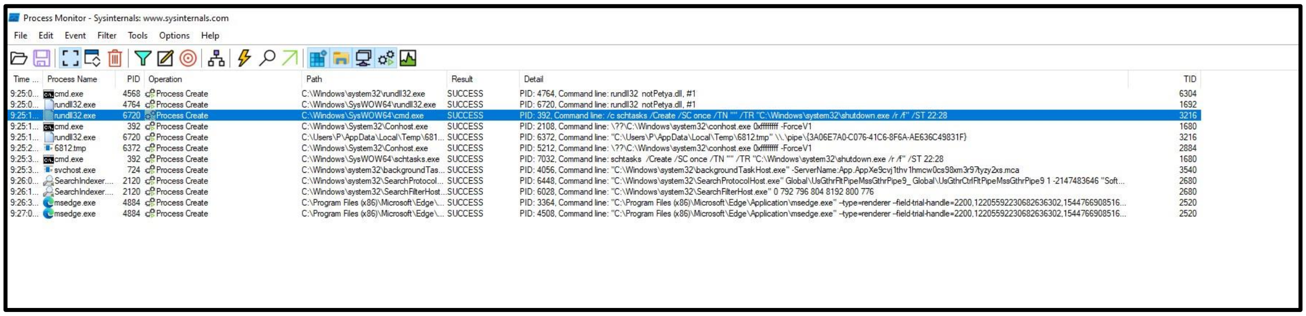
Then we ran the virus file which is notPetya.dll with the following command in the admin command prompt **rundll32 notPetya.dll , #1** we are giving #1 since that is the first point of entry for this file.



Administrator: Command Prompt

```
C:\Users\P\Desktop>rundll32 notPetya.dll, #1

C:\Users\P\Desktop>
```

Here we can see in Procmon, a Process Create operation created by this dll to execute the shutdown action

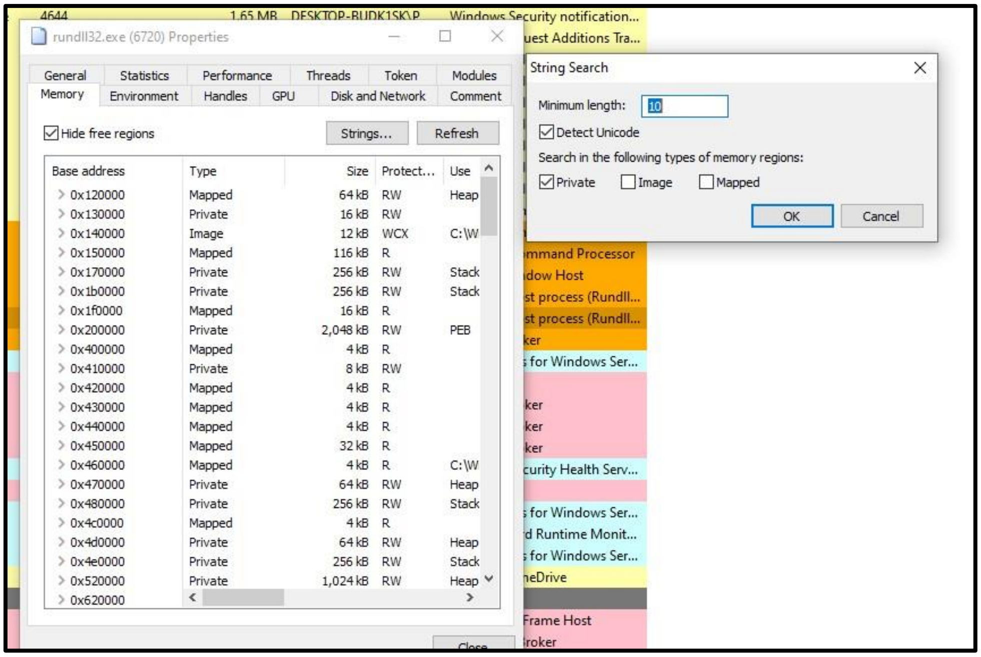


Process Monitor - Sysinternals: www.sysinternals.com

Time ...	Process Name	PID	Operation	Path	Result	Detail	TID
9:25:0...	cmd.exe	4568	Process Create	C:\Windows\system32\rundll32.exe	SUCCESS	PID: 4764, Command line: rundll32 notPetya.dll, #1	6304
9:25:0...	rundll32.exe	4764	Process Create	C:\Windows\SysWOW64\rundll32.exe	SUCCESS	PID: 6720, Command line: rundll32 notPetya.dll, #1	1692
9:25:1...	rundll32.exe	6720	Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 392, Command line: /c schtasks /Create /SC once /TN "" /TR "C:\Windows\system32\shutdown.exe /r /f" /ST 22:28	3216
9:25:1...	cmd.exe	392	Process Create	C:\Windows\System32\Conhost.exe	SUCCESS	PID: 2108, Command line: \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	1680
9:25:1...	rundll32.exe	6720	Process Create	C:\Users\P\AppData\Local\Temp\681...	SUCCESS	PID: 6372, Command line: "C:\Users\P\AppData\Local\Temp\6812.tmp" \\.\pipe\{3A06E7A0-C076-41C6-8F6A-AE636C49831F}	3216
9:25:2...	6812.tmp	6372	Process Create	C:\Windows\System32\Conhost.exe	SUCCESS	PID: 5212, Command line: \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1	2884
9:25:3...	cmd.exe	392	Process Create	C:\Windows\SysWOW64\schtasks.exe	SUCCESS	PID: 7032, Command line: schtasks /Create /SC once /TN "" /TR "C:\Windows\system32\shutdown.exe /r /f" /ST 22:28	1680
9:25:3...	svchost.exe	724	Process Create	C:\Windows\system32\backgroundTas...	SUCCESS	PID: 4056, Command line: "C:\Windows\system32\backgroundTaskHost.exe" -ServerName:App.AppXe9cvj1thv1hmcw0cs98xm3r97tyzy2xs.mca	3540
9:26:0...	SearchIndexer....	2120	Process Create	C:\Windows\system32\SearchProtocol...	SUCCESS	PID: 6448, Command line: "C:\Windows\system32\SearchProtocolHost.exe" Global\UsGthrFltPipeMssGthrPipe9_ Global\UsGthrCtrlFltPipeMssGthrPipe9 1 -2147483646 "Soft...	2680
9:26:1...	SearchIndexer....	2120	Process Create	C:\Windows\system32\SearchFilterHost...	SUCCESS	PID: 6028, Command line: "C:\Windows\system32\SearchFilterHost.exe" 0 792 796 804 8192 800 776	2680
9:26:3...	msedge.exe	4884	Process Create	C:\Program Files (x86)\Microsoft\Edge\...	SUCCESS	PID: 3364, Command line: "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --type=renderer --field-trial-handle=2200,12205592230682636302,1544766908516...	2520
9:27:0...	msedge.exe	4884	Process Create	C:\Program Files (x86)\Microsoft\Edge\...	SUCCESS	PID: 4508, Command line: "C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --type=renderer --field-trial-handle=2200,12205592230682636302,1544766908516...	2520

In ProcHacker, we will now study the properties of rundll32.exe and search for strings Here

we can see the shutdown command.



rundll32.exe (6720) Properties

☑ Hide free regions

Base address	Type	Size	Protect...	Use
> 0x120000	Mapped	64 kB	RW	Heap
> 0x130000	Private	16 kB	RW	
> 0x140000	Image	12 kB	WCX	C:\W
> 0x150000	Mapped	116 kB	R	
> 0x170000	Private	256 kB	RW	Stack
> 0x1b0000	Private	256 kB	RW	Stack
> 0x1f0000	Mapped	16 kB	R	
> 0x200000	Private	2,048 kB	RW	PEB
> 0x400000	Mapped	4 kB	R	
> 0x410000	Private	8 kB	RW	
> 0x420000	Mapped	4 kB	R	
> 0x430000	Mapped	4 kB	R	
> 0x440000	Mapped	4 kB	R	
> 0x450000	Mapped	32 kB	R	
> 0x460000	Mapped	4 kB	R	C:\W
> 0x470000	Private	64 kB	RW	Heap
> 0x480000	Private	256 kB	RW	Stack
> 0x4c0000	Mapped	4 kB	R	
> 0x4d0000	Private	64 kB	RW	Heap
> 0x4e0000	Private	256 kB	RW	Stack
> 0x520000	Private	1,024 kB	RW	Heap
> 0x620000				

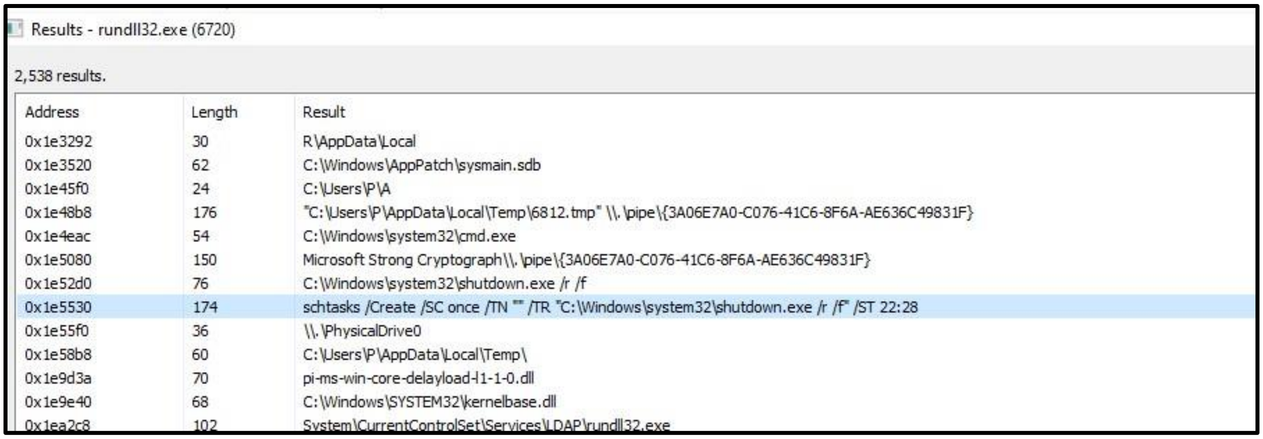
String Search

Minimum length: 10

☑ Detect Unicode

Search in the following types of memory regions:

☑ Private ☐ Image ☐ Mapped

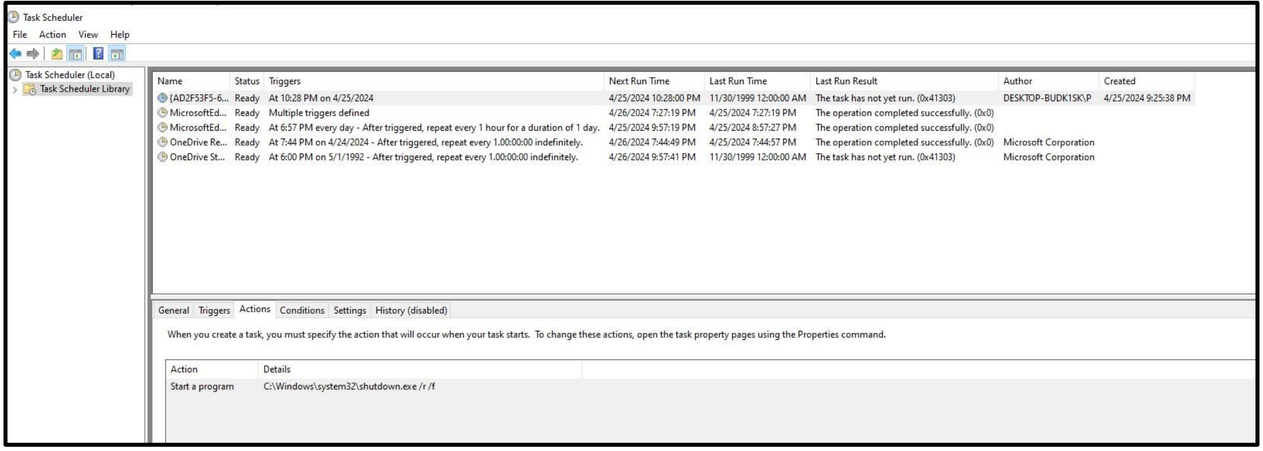


2,538 results.

Address	Length	Result
0x1e3292	30	R\AppData\Local
0x1e3520	62	C:\Windows\AppPatch\sysmain.sdb
0x1e45f0	24	C:\Users\P\A
0x1e48b8	176	"C:\Users\P\AppData\Local\Temp\6812.tmp" \\.\pipe\{3A06E7A0-C076-41C6-8F6A-AE636C49831F}
0x1e4eac	54	C:\Windows\system32\cmd.exe
0x1e5080	150	Microsoft Strong Cryptograph\\.\pipe\{3A06E7A0-C076-41C6-8F6A-AE636C49831F}
0x1e52d0	76	C:\Windows\system32\shutdown.exe /r /f
0x1e5530	174	schtasks /Create /SC once /TN "" /TR "C:\Windows\system32\shutdown.exe /r /f" /ST 22:28
0x1e55f0	36	\\.\PhysicalDrive0
0x1e58b8	60	C:\Users\P\AppData\Local\Temp\
0x1e9d3a	70	pi-ms-win-core-delayload-l1-1-0.dll
0x1e9e40	68	C:\Windows\SYSTEM32\kernelbase.dll
0x1ea2c8	102	System\CurrentControlSet\Services\LDAP\rundll32.exe

We can also see the exact details of this action by using the task scheduler.

In this application we can see the exact time the action is going to be executed.



Name	Status	Triggers	Next Run Time	Last Run Time	Last Run Result	Author	Created
{AD2F53F5-6...	Ready	At 10:28 PM on 4/25/2024	4/25/2024 10:28:00 PM	11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)	DESKTOP-BUDK1SK\P	4/25/2024 9:25:38 PM
MicrosoftEd...	Ready	Multiple triggers defined	4/26/2024 7:27:19 PM	4/25/2024 7:27:19 PM	The operation completed successfully. (0x0)		
MicrosoftEd...	Ready	At 6:57 PM every day - After triggered, repeat every 1 hour for a duration of 1 day.	4/25/2024 9:57:19 PM	4/25/2024 8:57:27 PM	The operation completed successfully. (0x0)		
OneDrive Re...	Ready	At 7:44 PM on 4/24/2024 - After triggered, repeat every 1.00:00:00 indefinitely.	4/26/2024 7:44:49 PM	4/25/2024 7:44:57 PM	The operation completed successfully. (0x0)	Microsoft Corporation	
OneDrive St...	Ready	At 6:00 PM on 5/1/1992 - After triggered, repeat every 1.00:00:00 indefinitely.	4/26/2024 9:57:41 PM	11/30/1999 12:00:00 AM	The task has not yet run. (0x41303)	Microsoft Corporation	

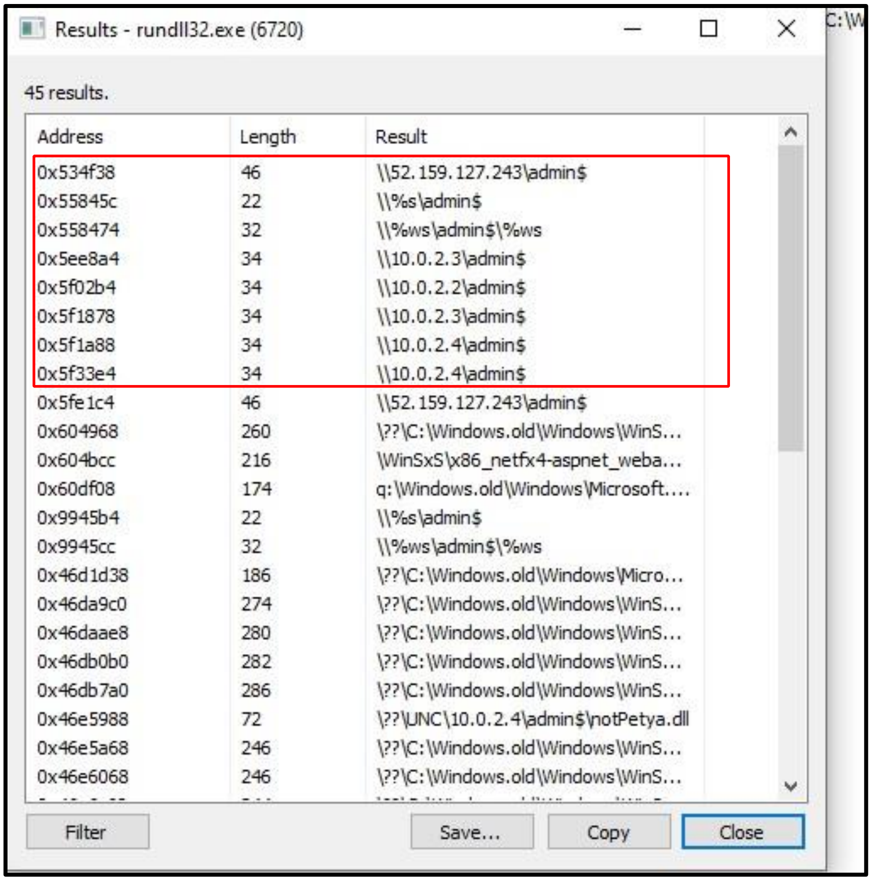
General Triggers Actions Conditions Settings History (disabled)

When you create a task, you must specify the action that will occur when your task starts. To change these actions, open the task property pages using the Properties command.

Action	Details
Start a program	C:\Windows\system32\shutdown.exe /r /f

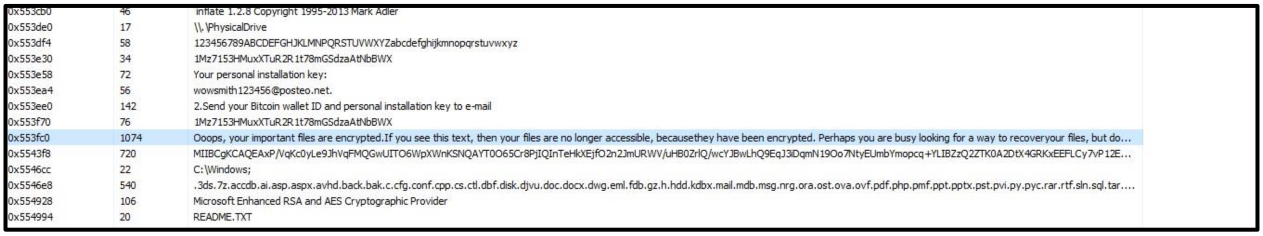
Again coming back to the strings section of Procmon, we filtered for the commands that included admin in it.

We can see that it will execute various ip addresses with admin



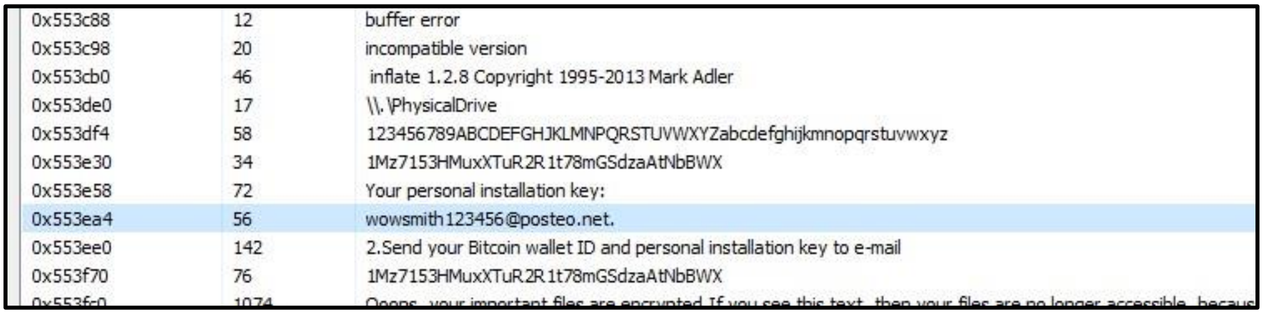
```
Results - rundll32.exe (6720)

45 results.

Address      Length   Result
0x534f38     46       \\52.159.127.243\admin$
0x55845c     22       \\%s\admin$
0x558474     32       \\%ws\admin$\%ws
0x5ee8a4     34       \\10.0.2.3\admin$
0x5f02b4     34       \\10.0.2.2\admin$
0x5f1878     34       \\10.0.2.3\admin$
0x5f1a88     34       \\10.0.2.4\admin$
0x5f33e4     34       \\10.0.2.4\admin$
0x5fe1c4     46       \\52.159.127.243\admin$
0x604968     260      \??\C:\Windows.old\Windows\WinS...
0x604bcc     216      \WinSxS\x86_netfx4-aspnet_weba...
0x60df08     174      q:\Windows.old\Windows\Microsoft....
0x9945b4     22       \\%s\admin$
0x9945cc     32       \\%ws\admin$\%ws
0x46d1d38    186      \??\C:\Windows.old\Windows\Micro...
0x46da9c0    274      \??\C:\Windows.old\Windows\WinS...
0x46daae8    280      \??\C:\Windows.old\Windows\WinS...
0x46db0b0    282      \??\C:\Windows.old\Windows\WinS...
0x46db7a0    286      \??\C:\Windows.old\Windows\WinS...
0x46e5988    72       \??\UNC\10.0.2.4\admin$\notPetya.dll
0x46e5a68    246      \??\C:\Windows.old\Windows\WinS...
0x46e6068    246      \??\C:\Windows.old\Windows\WinS...
```

Next we found out the exact command that it will execute during ransom attack that is the “Oops, your important files are encrypted…” string.



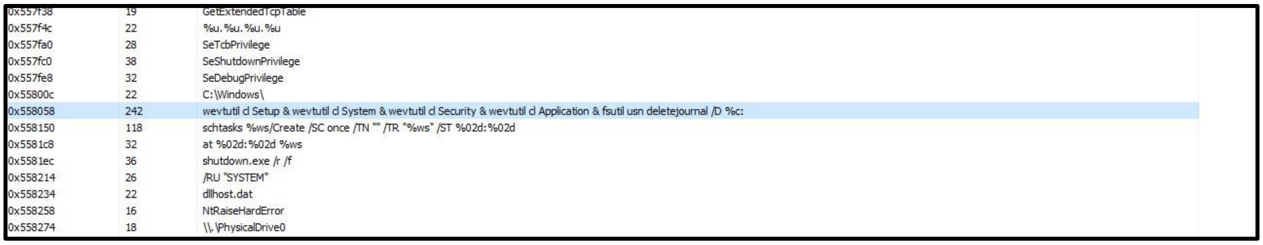
```
0x553cb0   46     inflate 1.2.8 Copyright 1995-2013 Mark Adler
0x553de0   17     \\.\PhysicalDrive
0x553df4   58     123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz
0x553e30   34     1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX
0x553e58   72     Your personal installation key:
0x553ea4   56     wowsmith123456@posteo.net.
0x553ee0   142    2.Send your Bitcoin wallet ID and personal installation key to e-mail
0x553f70   76     1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX
0x553fc0   1074   Ooops, your important files are encrypted.If you see this text, then your files are no longer accessible, becausethey have been encrypted. Perhaps you are busy looking for a way to recoveryour files, but do...
0x5543f8   720    MIIBCgKCAQEAxP/VqKc0yLe9JhVqFMQGwUITO6WpXWnKSNQAYT0O65Cr8PjIQInTeHkXEjfO2n2JmURWV/uHB0ZrlQ/wcYJBwLhQ9EqJ3iDqmN19Oo7NtyEUmbYmopcq+YLIBZzQ2ZTK0A2DtX4GRKxEEFLCy7vP12E...
0x5546cc   22     C:\Windows;
0x5546e8   540    .3ds.7z.accdb.ai.asp.aspx.avhd.back.bak.c.cfg.conf.cpp.cs.ctl.dbf.disk.djvu.doc.docx.dwg.eml.fdb.gz.h.hdd.kdbx.mail.mdb.msg.nrg.ora.ost.ova.ovf.pdf.php.pmf.ppt.pptx.pst.pvi.py.pyc.rar.rtf.sln.sql.tar....
0x554928   106    Microsoft Enhanced RSA and AES Cryptographic Provider
0x554994   20     README.TXT
```

After further analysis we got to know the email address that this attack will display to collect the ransom.



```
0x553c88   12     buffer error
0x553c98   20     incompatible version
0x553cb0   46      inflate 1.2.8 Copyright 1995-2013 Mark Adler
0x553de0   17     \\.\PhysicalDrive
0x553df4   58     123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz
0x553e30   34     1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX
0x553e58   72     Your personal installation key:
0x553ea4   56     wowsmith123456@posteo.net.
0x553ee0   142    2.Send your Bitcoin wallet ID and personal installation key to e-mail
0x553f70   76     1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX
0x553fc0   1074   Ooops, your important files are encrypted.If you see this text, then your files are no longer accessible, becaus
```

This command will delete the files on the system once the attack is successful.

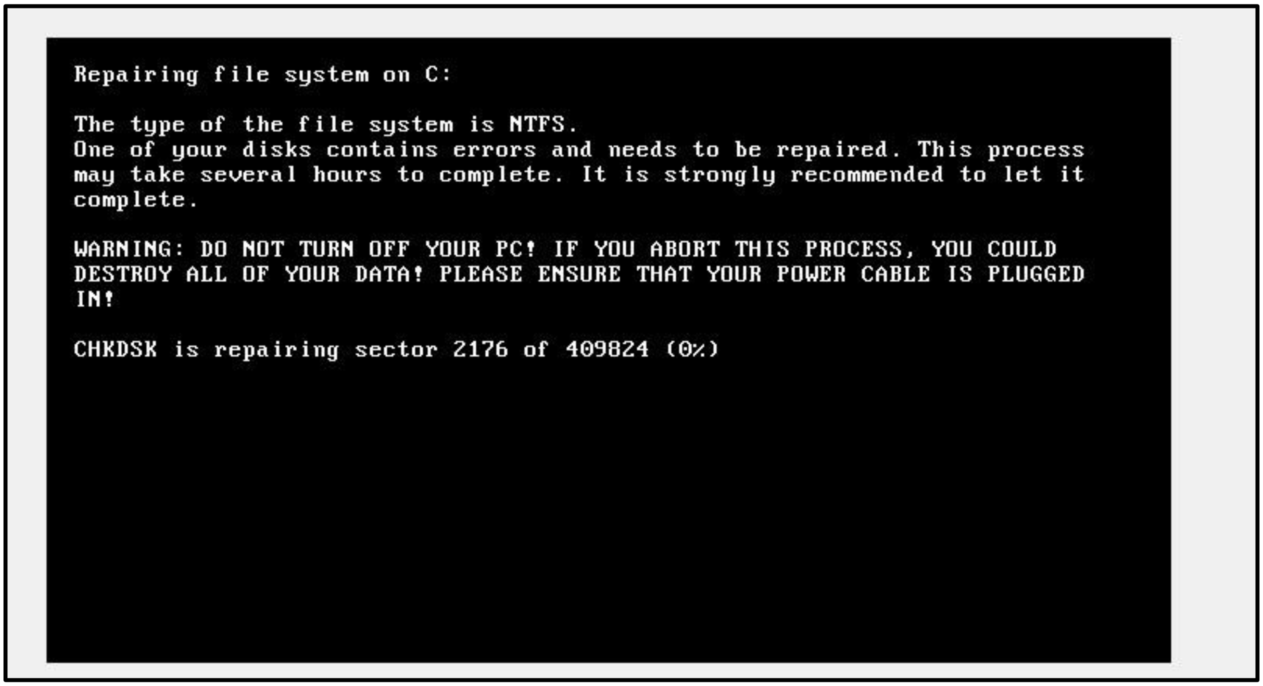


```
0x557f38     19     GetExtendedTcpTable
0x557f4c     22     %u.%u.%u.%u
0x557fa0     28     SeTcbPrivilege
0x557fc0     38     SeShutdownPrivilege
0x557fe8     32     SeDebugPrivilege
0x55800c     22     C:\Windows\
0x558058     242    wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl Application & fsutil usn deletejournal /D %c:
0x558150     118    schtasks %ws/Create /SC once /TN "" /TR "%ws" /ST %02d:%02d
0x5581c8     32     at %02d:%02d %ws
0x5581ec     36     shutdown.exe /r /f
0x558214     26     /RU "SYSTEM"
0x558234     22     dllhost.dat
0x558258     16     NtRaiseHardError
0x558274     18     \\.\PhysicalDrive0
```

Now we rebooted the system to start the attack early instead of waiting for the given time.

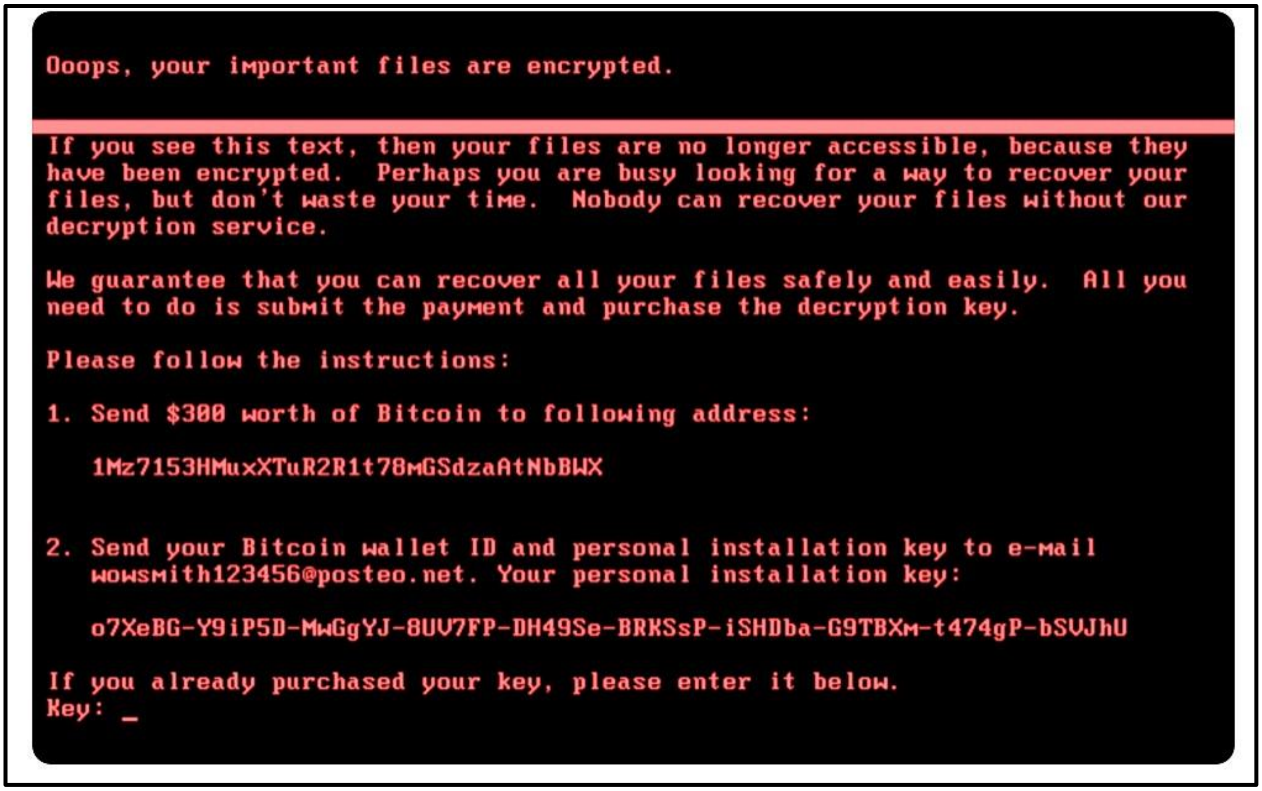
This screen is just a cover.

To be safe from the attack a person should turn his system off at this point right away and not wait for the sectors to be completed.



```
Repairing file system on C:

The type of the file system is NTFS.
One of your disks contains errors and needs to be repaired. This process
may take several hours to complete. It is strongly recommended to let it
complete.

WARNING: DO NOT TURN OFF YOUR PC! IF YOU ABORT THIS PROCESS, YOU COULD
DESTROY ALL OF YOUR DATA! PLEASE ENSURE THAT YOUR POWER CABLE IS PLUGGED
IN!

CHKDSK is repairing sector 2176 of 409824 (0%)
```

Once this is loaded, that is when the attack has successfully entered your system and this screen appears



```
Ooops, your important files are encrypted.

If you see this text, then your files are no longer accessible, because they
have been encrypted.  Perhaps you are busy looking for a way to recover your
files, but don't waste your time.  Nobody can recover your files without our
decryption service.

We guarantee that you can recover all your files safely and easily.  All you
need to do is submit the payment and purchase the decryption key.

Please follow the instructions:

1. Send $300 worth of Bitcoin to following address:

   1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX


2. Send your Bitcoin wallet ID and personal installation key to e-mail
   wowsmith123456@posteo.net. Your personal installation key:

   o7XeBG-Y9iP5D-MwGgYJ-8UV7FP-DH49Se-BRKSsP-iSHDba-G9TBXm-t474gP-bSVJhU

If you already purchased your key, please enter it below.
Key: _
```

This is the ransom attack screen where it has asked for $300 worth of Bitcoin.

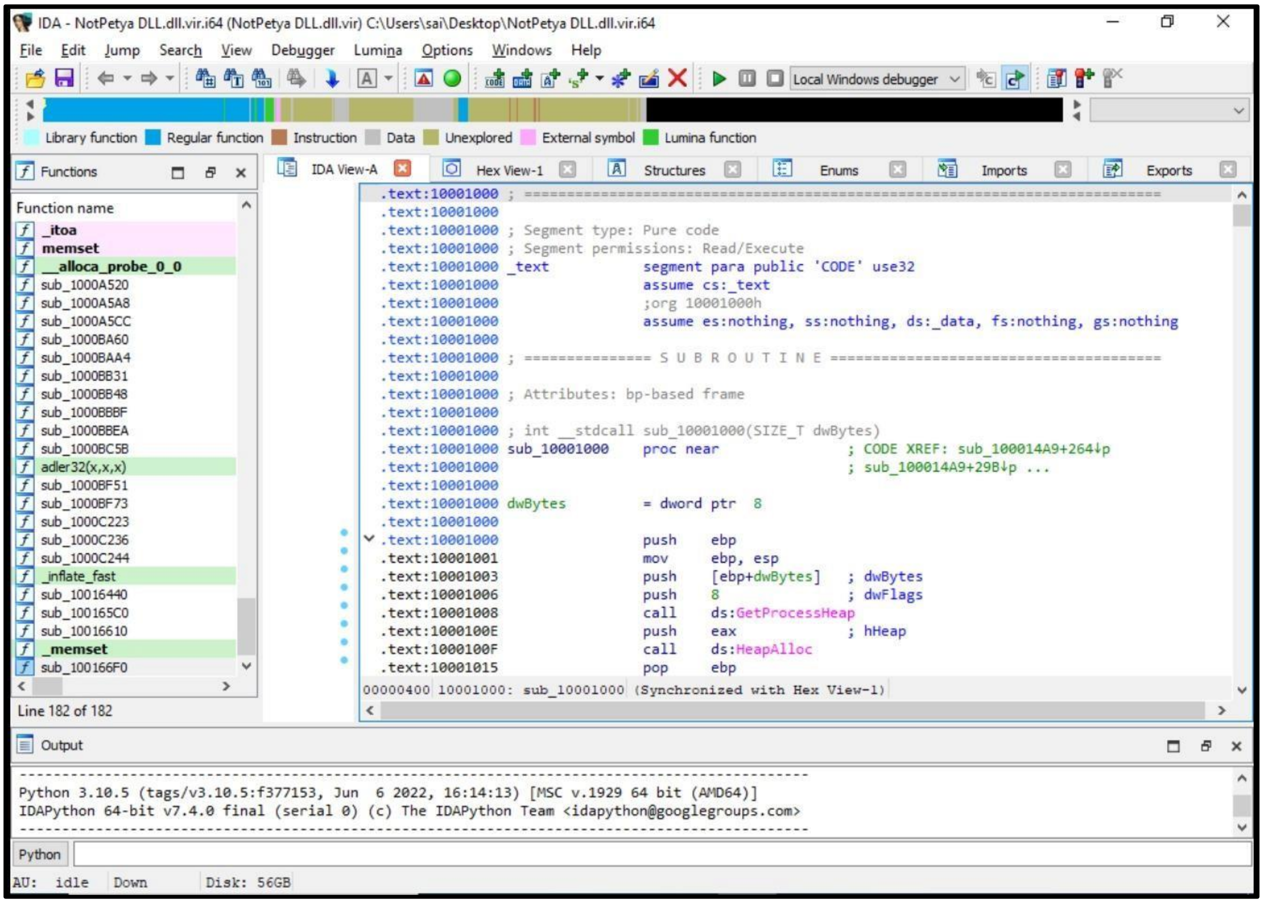
And also provides the victim with a personal key that he has to give along with his wallet ID to the attacker’s email address that is given.

No matter which key the victim puts the system wont accept it.

This is how you come to know that you have become a victim of this attack.

Analysis

We analyzed this virus in IDA pro,


IDA - NotPetya DLL.dll.vir.i64 (NotPetya DLL.dll.vir) C:\Users\sai\Desktop\NotPetya DLL.dll.vir.i64
File Edit Jump Search View Debugger Lumina Options Windows Help
Local Windows debugger
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions
IDA View-A
Hex View-1
Structures
Enums
Imports
Exports
Output
Python

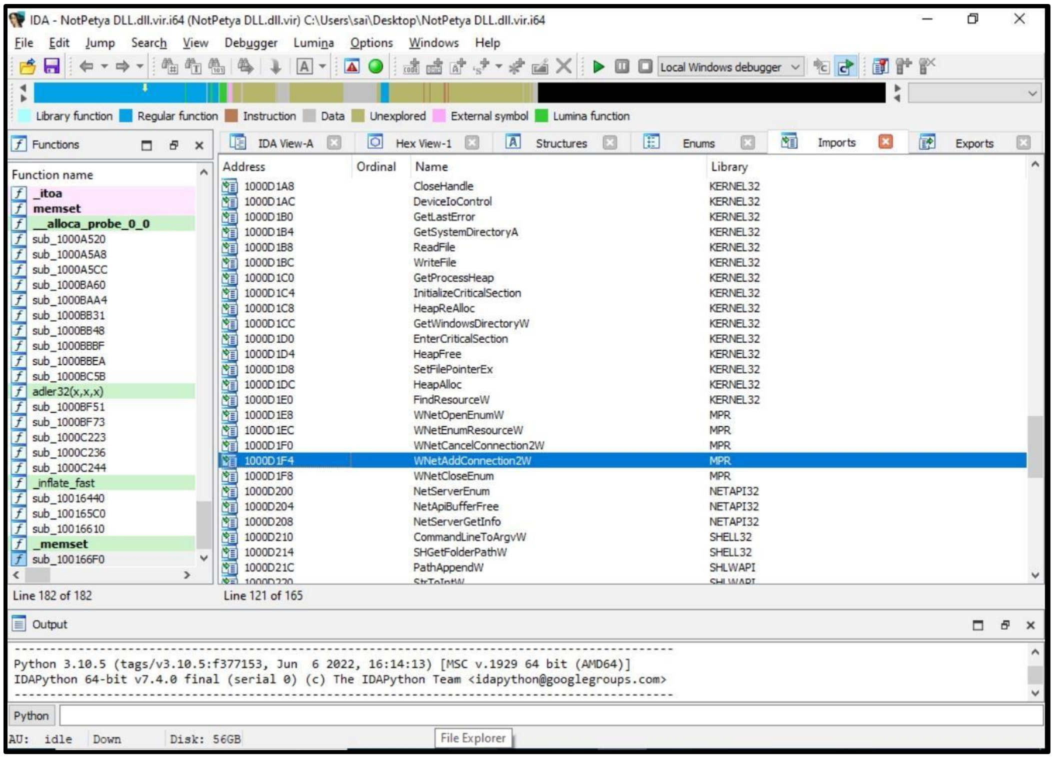
```
Function name
_itoa
memset
__alloca_probe_0_0
sub_1000A520
sub_1000A5A8
sub_1000A5CC
sub_1000BA60
sub_1000BAA4
sub_1000BB31
sub_1000BB48
sub_1000BBBF
sub_1000BBEA
sub_1000BC5B
adler32(x,x,x)
sub_1000BF51
sub_1000BF73
sub_1000C223
sub_1000C236
sub_1000C244
_inflate_fast
sub_10016440
sub_100165C0
sub_10016610
_memset
sub_100166F0
Line 182 of 182
```

```
.text:10001000 ; ===========================================================================
.text:10001000
.text:10001000 ; Segment type: Pure code
.text:10001000 ; Segment permissions: Read/Execute
.text:10001000 _text           segment para public 'CODE' use32
.text:10001000                 assume cs:_text
.text:10001000                 ;org 10001000h
.text:10001000                 assume es:nothing, ss:nothing, ds:_data, fs:nothing, gs:nothing
.text:10001000
.text:10001000 ; =============== S U B R O U T I N E =======================================
.text:10001000
.text:10001000 ; Attributes: bp-based frame
.text:10001000
.text:10001000 ; int __stdcall sub_10001000(SIZE_T dwBytes)
.text:10001000 sub_10001000    proc near               ; CODE XREF: sub_100014A9+264↓p
.text:10001000                                         ; sub_100014A9+29B↓p ...
.text:10001000
.text:10001000 dwBytes         = dword ptr  8
.text:10001000
.text:10001000                 push    ebp
.text:10001001                 mov     ebp, esp
.text:10001003                 push    [ebp+dwBytes]   ; dwBytes
.text:10001006                 push    8               ; dwFlags
.text:10001008                 call    ds:GetProcessHeap
.text:1000100E                 push    eax             ; hHeap
.text:1000100F                 call    ds:HeapAlloc
.text:10001015                 pop     ebp
00000400 10001000: sub_10001000 (Synchronized with Hex View-1)
```

```
---------------------------------------------------------------------------------------
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
---------------------------------------------------------------------------------------
AU:  idle   Down     Disk: 56GB
```

We started analyzing the virus by going through the names list that consists of the names of the imports of the API calls which the malware uses and there are network connections being created.

The one that is important for us is the WNetAddConnection2W



IDA - NotPetya DLL.dll.vir.i64 (NotPetya DLL.dll.vir) C:\Users\sai\Desktop\NotPetya DLL.dll.vir.i64

File Edit Jump Search View Debugger Lumina Options Windows Help
Local Windows debugger
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions
IDA View-A
Hex View-1
Structures
Enums
Imports
Exports

Function name
- _itoa
- memset
- __alloca_probe_0_0
- sub_1000A520
- sub_1000A5A8
- sub_1000A5CC
- sub_1000BA60
- sub_1000BAA4
- sub_1000BB31
- sub_1000BB48
- sub_1000BBBF
- sub_1000BBEA
- sub_1000BC5B
- adler32(x,x,x)
- sub_1000BF51
- sub_1000BF73
- sub_1000C223
- sub_1000C236
- sub_1000C244
- _inflate_fast
- sub_10016440
- sub_100165C0
- sub_10016610
- _memset
- sub_100166F0

Line 182 of 182

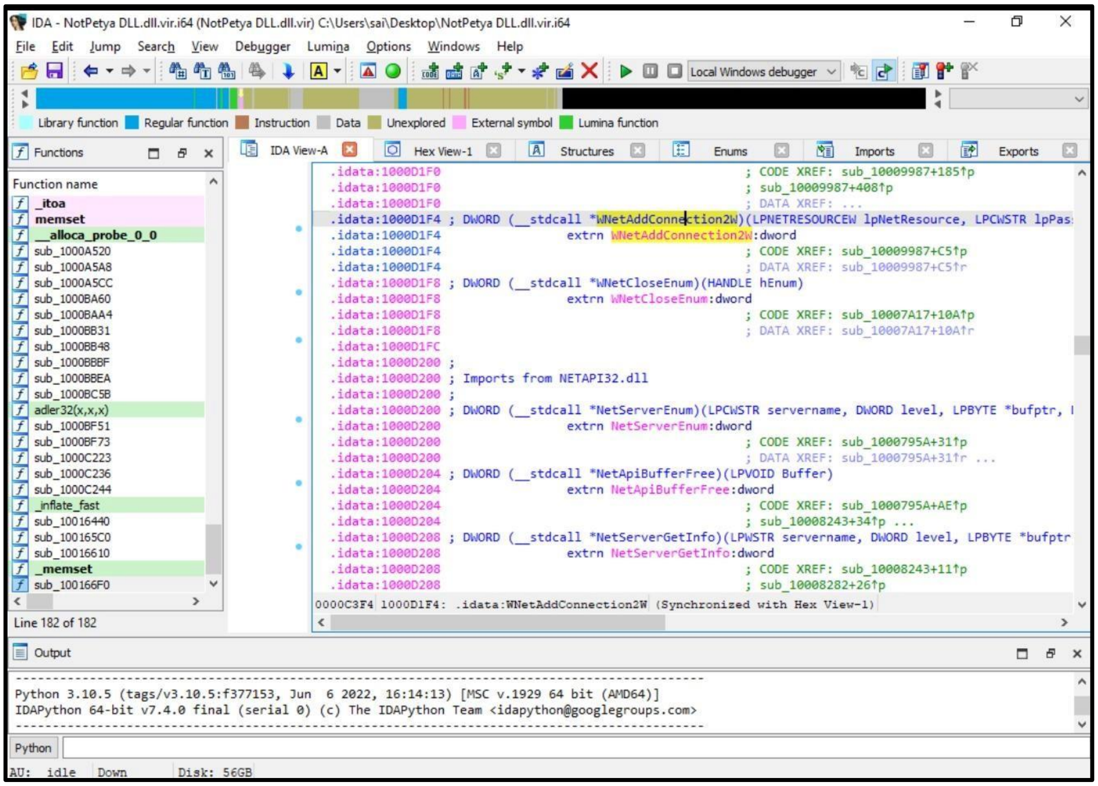
Address	Ordinal	Name	Library
1000D1A8		CloseHandle	KERNEL32
1000D1AC		DeviceIoControl	KERNEL32
1000D1B0		GetLastError	KERNEL32
1000D1B4		GetSystemDirectoryA	KERNEL32
1000D1B8		ReadFile	KERNEL32
1000D1BC		WriteFile	KERNEL32
1000D1C0		GetProcessHeap	KERNEL32
1000D1C4		InitializeCriticalSection	KERNEL32
1000D1C8		HeapReAlloc	KERNEL32
1000D1CC		GetWindowsDirectoryW	KERNEL32
1000D1D0		EnterCriticalSection	KERNEL32
1000D1D4		HeapFree	KERNEL32
1000D1D8		SetFilePointerEx	KERNEL32
1000D1DC		HeapAlloc	KERNEL32
1000D1E0		FindResourceW	KERNEL32
1000D1E8		WNetOpenEnumW	MPR
1000D1EC		WNetEnumResourceW	MPR
1000D1F0		WNetCancelConnection2W	MPR
1000D1F4		WNetAddConnection2W	MPR
1000D1F8		WNetCloseEnum	MPR
1000D200		NetServerEnum	NETAPI32
1000D204		NetApiBufferFree	NETAPI32
1000D208		NetServerGetInfo	NETAPI32
1000D210		CommandLineToArgvW	SHELL32
1000D214		SHGetFolderPathW	SHELL32
1000D21C		PathAppendW	SHLWAPI
1000D220		StrToIntW	SHLWAPI

Line 121 of 165

Output

```
-----------------------------------------------------------------------
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
-----------------------------------------------------------------------
```

This is the reference to the API call. We then press X, we can see where that call is being cross referenced and doubleclick to go to that part of the code.



IDA - NotPetya DLL.dll.vir.i64 (NotPetya DLL.dll.vir) C:\Users\sai\Desktop\NotPetya DLL.dll.vir.i64

File Edit Jump Search View Debugger Lumina Options Windows Help
Local Windows debugger
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions
IDA View-A
Hex View-1
Structures
Enums
Imports
Exports

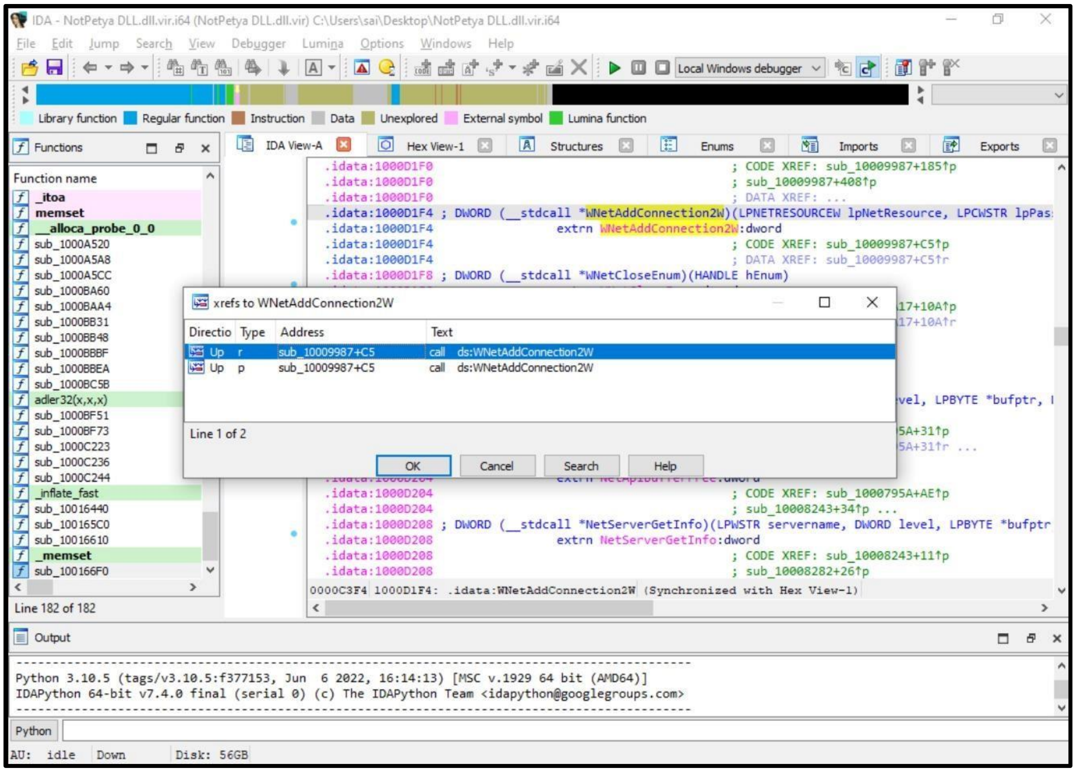
Function name
- _itoa
- memset
- __alloca_probe_0_0
- sub_1000A520
- sub_1000A5A8
- sub_1000A5CC
- sub_1000BA60
- sub_1000BAA4
- sub_1000BB31
- sub_1000BB48
- sub_1000BBBF
- sub_1000BBEA
- sub_1000BC5B
- adler32(x,x,x)
- sub_1000BF51
- sub_1000BF73
- sub_1000C223
- sub_1000C236
- sub_1000C244
- _inflate_fast
- sub_10016440
- sub_100165C0
- sub_10016610
- _memset
- sub_100166F0

Line 182 of 182

```
.idata:1000D1F0                                         ; CODE XREF: sub_10009987+185↑p
.idata:1000D1F0                                         ; sub_10009987+408↑p
.idata:1000D1F0                                         ; DATA XREF: ...
.idata:1000D1F4 ; DWORD (__stdcall *WNetAddConnection2W)(LPNETRESOURCEW lpNetResource, LPCWSTR lpPas
.idata:1000D1F4                 extrn WNetAddConnection2W:dword
.idata:1000D1F4                                         ; CODE XREF: sub_10009987+C5↑p
.idata:1000D1F4                                         ; DATA XREF: sub_10009987+C5↑r
.idata:1000D1F8 ; DWORD (__stdcall *WNetCloseEnum)(HANDLE hEnum)
.idata:1000D1F8                 extrn WNetCloseEnum:dword
.idata:1000D1F8                                         ; CODE XREF: sub_10007A17+10A↑p
.idata:1000D1F8                                         ; DATA XREF: sub_10007A17+10A↑r
.idata:1000D1FC
.idata:1000D200 ;
.idata:1000D200 ; Imports from NETAPI32.dll
.idata:1000D200 ;
.idata:1000D200 ; DWORD (__stdcall *NetServerEnum)(LPCWSTR servername, DWORD level, LPBYTE *bufptr, I
.idata:1000D200                 extrn NetServerEnum:dword
.idata:1000D200                                         ; CODE XREF: sub_1000795A+31↑p
.idata:1000D200                                         ; DATA XREF: sub_1000795A+31↑r ...
.idata:1000D204 ; DWORD (__stdcall *NetApiBufferFree)(LPVOID Buffer)
.idata:1000D204                 extrn NetApiBufferFree:dword
.idata:1000D204                                         ; CODE XREF: sub_1000795A+AE↑p
.idata:1000D204                                         ; sub_10008243+34↑p ...
.idata:1000D208 ; DWORD (__stdcall *NetServerGetInfo)(LPWSTR servername, DWORD level, LPBYTE *bufptr
.idata:1000D208                 extrn NetServerGetInfo:dword
.idata:1000D208                                         ; CODE XREF: sub_10008243+11↑p
.idata:1000D208                                         ; sub_10008282+26↑p
0000C3F4 1000D1F4: .idata:WNetAddConnection2W (Synchronized with Hex View-1)
```

Output

```
-----------------------------------------------------------------------
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
-----------------------------------------------------------------------
```



IDA - NotPetya DLL.dll.vir.i64 (NotPetya DLL.dll.vir) C:\Users\sai\Desktop\NotPetya DLL.dll.vir.i64

Function name
_itoa
memset
__alloca_probe_0_0
sub_1000A520
sub_1000A5A8
sub_1000A5CC
sub_1000BA60
sub_1000BAA4
sub_1000BB31
sub_1000BB48
sub_1000BBBF
sub_1000BBEA
sub_1000BC5B
adler32(x,x,x)
sub_1000BF51
sub_1000BF73
sub_1000C223
sub_1000C236
sub_1000C244
_inflate_fast
sub_10016440
sub_100165C0
sub_10016610
_memset
sub_100166F0

Line 182 of 182

```
.idata:1000D1F0                                          ; CODE XREF: sub_10009987+185↑p
.idata:1000D1F0                                          ; sub_10009987+408↑p
.idata:1000D1F0                                          ; DATA XREF: ...
.idata:1000D1F4 ; DWORD (__stdcall *WNetAddConnection2W)(LPNETRESOURCEW lpNetResource, LPCWSTR lpPas
.idata:1000D1F4                 extrn WNetAddConnection2W:dword
.idata:1000D1F4                                          ; CODE XREF: sub_10009987+C5↑p
.idata:1000D1F4                                          ; DATA XREF: sub_10009987+C5↑r
.idata:1000D1F8 ; DWORD (__stdcall *WNetCloseEnum)(HANDLE hEnum)
                                                         17+10A↑p
                                                         17+10A↑r
                                                         vel, LPBYTE *bufptr, l
                                                         5A+31↑p
                                                         5A+31↑r ...
.idata:1000D204                                          ; CODE XREF: sub_1000795A+AE↑p
.idata:1000D204                                          ; sub_10008243+34↑p ...
.idata:1000D208 ; DWORD (__stdcall *NetServerGetInfo)(LPWSTR servername, DWORD level, LPBYTE *bufptr
.idata:1000D208                 extrn NetServerGetInfo:dword
.idata:1000D208                                          ; CODE XREF: sub_10008243+11↑p
.idata:1000D208                                          ; sub_10008282+26↑p
0000C3F4 1000D1F4: .idata:WNetAddConnection2W (Synchronized with Hex View-1)
```

xrefs to WNetAddConnection2W

Directio	Type	Address	Text
Up	r	sub_10009987+C5	call ds:WNetAddConnection2W
Up	p	sub_10009987+C5	call ds:WNetAddConnection2W

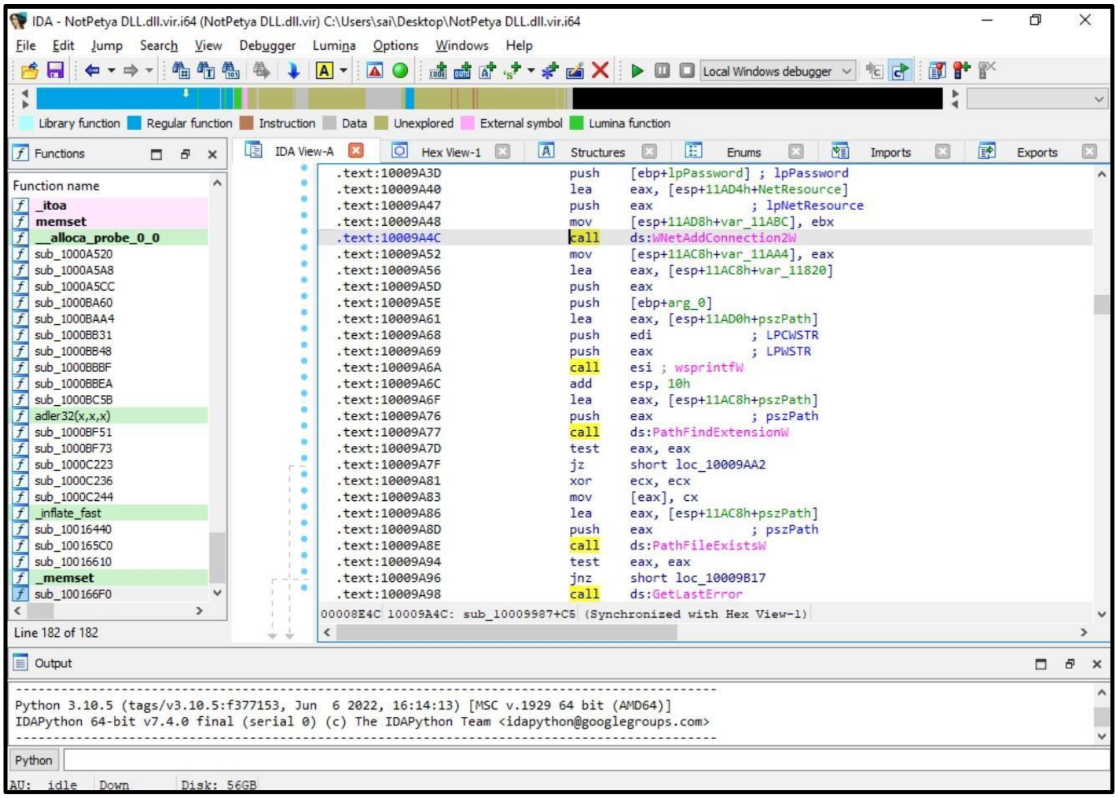
Line 1 of 2

OK Cancel Search Help

```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
```

AU: idle Down Disk: 56GB

Just before the call we can see the username, password is being passed to function as arguments on the stack.



IDA - NotPetya DLL.dll.vir.i64 (NotPetya DLL.dll.vir) C:\Users\sai\Desktop\NotPetya DLL.dll.vir.i64

Function name
_itoa
memset
__alloca_probe_0_0
sub_1000A520
sub_1000A5A8
sub_1000A5CC
sub_1000BA60
sub_1000BAA4
sub_1000BB31
sub_1000BB48
sub_1000BBBF
sub_1000BBEA
sub_1000BC5B
adler32(x,x,x)
sub_1000BF51
sub_1000BF73
sub_1000C223
sub_1000C236
sub_1000C244
_inflate_fast
sub_10016440
sub_100165C0
sub_10016610
_memset
sub_100166F0

Line 182 of 182

```
.text:10009A3D                 push    [ebp+lpPassword] ; lpPassword
.text:10009A40                 lea     eax, [esp+11AD4h+NetResource]
.text:10009A47                 push    eax             ; lpNetResource
.text:10009A48                 mov     [esp+11AD8h+var_11ABC], ebx
.text:10009A4C                 call    ds:WNetAddConnection2W
.text:10009A52                 mov     [esp+11AC8h+var_11AA4], eax
.text:10009A56                 lea     eax, [esp+11AC8h+var_11820]
.text:10009A5D                 push    eax
.text:10009A5E                 push    [ebp+arg_0]
.text:10009A61                 lea     eax, [esp+11AD0h+pszPath]
.text:10009A68                 push    edi             ; LPCWSTR
.text:10009A69                 push    eax             ; LPWSTR
.text:10009A6A                 call    esi ; wsprintfW
.text:10009A6C                 add     esp, 10h
.text:10009A6F                 lea     eax, [esp+11AC8h+pszPath]
.text:10009A76                 push    eax             ; pszPath
.text:10009A77                 call    ds:PathFindExtensionW
.text:10009A7D                 test    eax, eax
.text:10009A7F                 jz      short loc_10009AA2
.text:10009A81                 xor     ecx, ecx
.text:10009A83                 mov     [eax], cx
.text:10009A86                 lea     eax, [esp+11AC8h+pszPath]
.text:10009A8D                 push    eax             ; pszPath
.text:10009A8E                 call    ds:PathFileExistsW
.text:10009A94                 test    eax, eax
.text:10009A96                 jnz     short loc_10009B17
.text:10009A98                 call    ds:GetLastError
00008E4C 10009A4C: sub_10009987+C5 (Synchronized with Hex View-1)
```

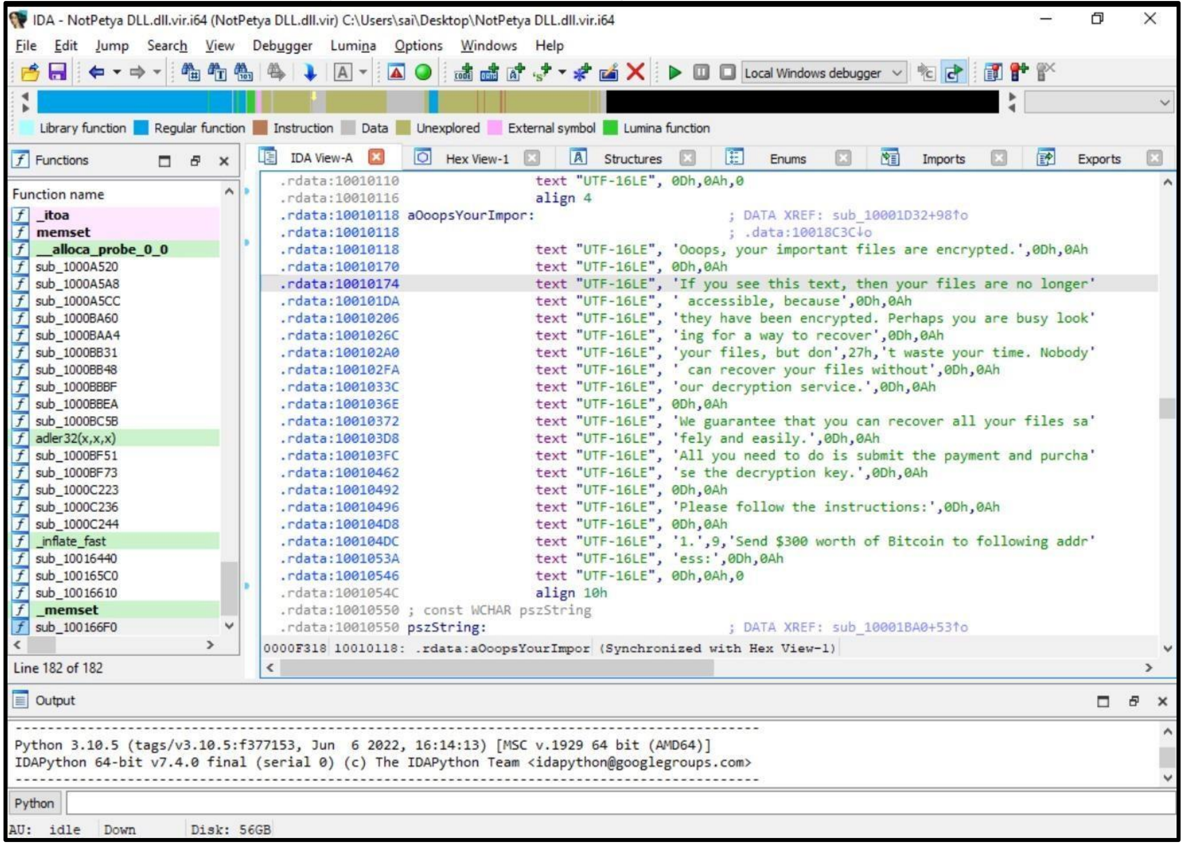
```
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
```

AU: idle Down Disk: 56GB

Now let’s go through the data section of the malware.

We can see the strings that the malware will execute.

So this how we have analyzed the malware using IDA pro.



```
.rdata:10010110                 text "UTF-16LE", 0Dh,0Ah,0
.rdata:10010116                 align 4
.rdata:10010118 aOoopsYourImpor:                        ; DATA XREF: sub_10001D32+98↑o
.rdata:10010118                                         ; .data:10018C3C↓o
.rdata:10010118                 text "UTF-16LE", 'Ooops, your important files are encrypted.',0Dh,0Ah
.rdata:10010170                 text "UTF-16LE", 0Dh,0Ah
.rdata:10010174                 text "UTF-16LE", 'If you see this text, then your files are no longer'
.rdata:100101DA                 text "UTF-16LE", ' accessible, because',0Dh,0Ah
.rdata:10010206                 text "UTF-16LE", 'they have been encrypted. Perhaps you are busy look'
.rdata:1001026C                 text "UTF-16LE", 'ing for a way to recover',0Dh,0Ah
.rdata:100102A0                 text "UTF-16LE", 'your files, but don',27h,'t waste your time. Nobody'
.rdata:100102FA                 text "UTF-16LE", ' can recover your files without',0Dh,0Ah
.rdata:1001033C                 text "UTF-16LE", 'our decryption service.',0Dh,0Ah
.rdata:1001036E                 text "UTF-16LE", 0Dh,0Ah
.rdata:10010372                 text "UTF-16LE", 'We guarantee that you can recover all your files sa'
.rdata:100103D8                 text "UTF-16LE", 'fely and easily.',0Dh,0Ah
.rdata:100103FC                 text "UTF-16LE", 'All you need to do is submit the payment and purcha'
.rdata:10010462                 text "UTF-16LE", 'se the decryption key.',0Dh,0Ah
.rdata:10010492                 text "UTF-16LE", 0Dh,0Ah
.rdata:10010496                 text "UTF-16LE", 'Please follow the instructions:',0Dh,0Ah
.rdata:100104D8                 text "UTF-16LE", 0Dh,0Ah
.rdata:100104DC                 text "UTF-16LE", '1.',9,'Send $300 worth of Bitcoin to following addr'
.rdata:1001053A                 text "UTF-16LE", 'ess:',0Dh,0Ah
.rdata:10010546                 text "UTF-16LE", 0Dh,0Ah,0
.rdata:1001054C                 align 10h
.rdata:10010550 ; const WCHAR pszString
.rdata:10010550 pszString:                              ; DATA XREF: sub_10001BA0+53↑o
0000F318 10010118: .rdata:aOoopsYourImpor (Synchronized with Hex View-1)
```

```
---------------------------------------------------------------------------------------
Python 3.10.5 (tags/v3.10.5:f377153, Jun  6 2022, 16:14:13) [MSC v.1929 64 bit (AMD64)]
IDAPython 64-bit v7.4.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
---------------------------------------------------------------------------------------
```

```
AU:  idle   Down      Disk: 56GB
```

Conclusion

In conclusion, the analysis of NotPetya provides valuable insights into the workings of one of the most destructive ransomware attacks in recent history. By acquiring the malware file from a reputable source and performing static analysis using IDA Pro, we gained a deeper understanding of its behavior and mechanisms.

Our implementation of NotPetya on a controlled system allowed us to observe its impact and behavior firsthand. We confirmed its ability to encrypt files and overwrite the Master Boot Record, rendering the system inoperable. Additionally, the analysis revealed the sophisticated propagation methods used by NotPetya to spread within networks and exploit vulnerabilities.

Through static analysis with IDA Pro, we dissected the malware's code and identified key functions responsible for encryption, propagation, and payload execution. This analysis provided crucial insights into how NotPetya operates and allowed us to understand its malicious mechanisms.