Malware Analysis Final Project

Malware: 71B6A493388E7D0B40C83CE903BC6B04

Team Members:
Bradley Dodds
Raymond Tang
Saurabh Mishra
Nathan Nhek
Oscar Castaneda
Justin Nhek

Table of Contents

Summary of the Analysis

In our initial static analysis of the malware we were able to identify this malware as being an encrypting ransomware variety of Petya which had initially wreaked havoc back in early 2016. This ransomware was dubbed NotPetya due to its very similar resemblance to the Petya ransomware. Initially discovered in 2017, the NotPetya ransomware started propagating infecting many organizations. It uses the *EternalBlue* exploit in the Windows server message block (SMB), (entry *CVE-2017-0144* in the Common Vulnerabilities and Exposures catalog) as well as the classic Server Message Block vulnerability in protocol to propagate itself.

A few months after Microsoft had released an update for the EternalBlue exploit, the vulnerability was stolen and leaked by hacker group *Shadow Brokers.* This cyber attack was used to exploit unpatched computers as well as patched computers by grabbing the passwords from those computers to infect others that were patched. A planted backdoor in the tax and accounting software *MEDoc* was used as an initial mechanism to that NotPetya infect computers from *MEDoc*'s servers in Ukraine. After gaining an initial foothold, it tries to spread through lateral movement. It is more effective than a typical ransomware as it not only encrypts files, but also overwrites and encrypts the master boot record.

More than 20 countries were affected by this cyberattack. Governments, banks, nuclear power plants, public transportation systems as well as businesses were impacted. As per a White House assessment, the total damages brought about by NotPetya is more than $10 billion, making it the most destructive cyberattack ever.

Upon further and more through static analysis we were able to discover that upon initial launch, the malware would first determine whether the specified process is running under WOW64 or an Intel64 of x64 process. Following that, a *schtasks* command would execute which we believe is setting a scheduled restart of the system in order for the malware to proceed to its next task. After which the malware's main functionality would kick in and start to encrypt data ending with a force shutdown to which a user would be greeting upon reboot with a ransom note.

Identification

Data	
Names	NotPetya.bin, NotPetya.exe, not_petya.exe, pt2.exe, perfc.dat
MD5	71B6A493388E7D0B40C83CE903BC6B04
SHA-1	34F917AABA5684FBE56D3C57D48EF2A1AA7CF06D
SSDEEP	6144:y/Bt80VmNTBo/x95ZjAetGDN3VFNq7pC+9OqFoK30b3ni5rdQY/CdUOs2:y/X4NTS/x9jNG+w+9OqFoK323qdQYKUG
File Type	Win32 DLL
Magic	PE32 executable for MS Windows (DLL) Intel 80386 32-bit
File Size	353.87 KB (362360 bytes)
Subsystem	Windows Command Line (version 5.1)

PE Header	
Target Machine	Intel 386 or later processors and compatible processors
Compilation Timestamp	2017-06-18 07:14:36
Contained Sections	5

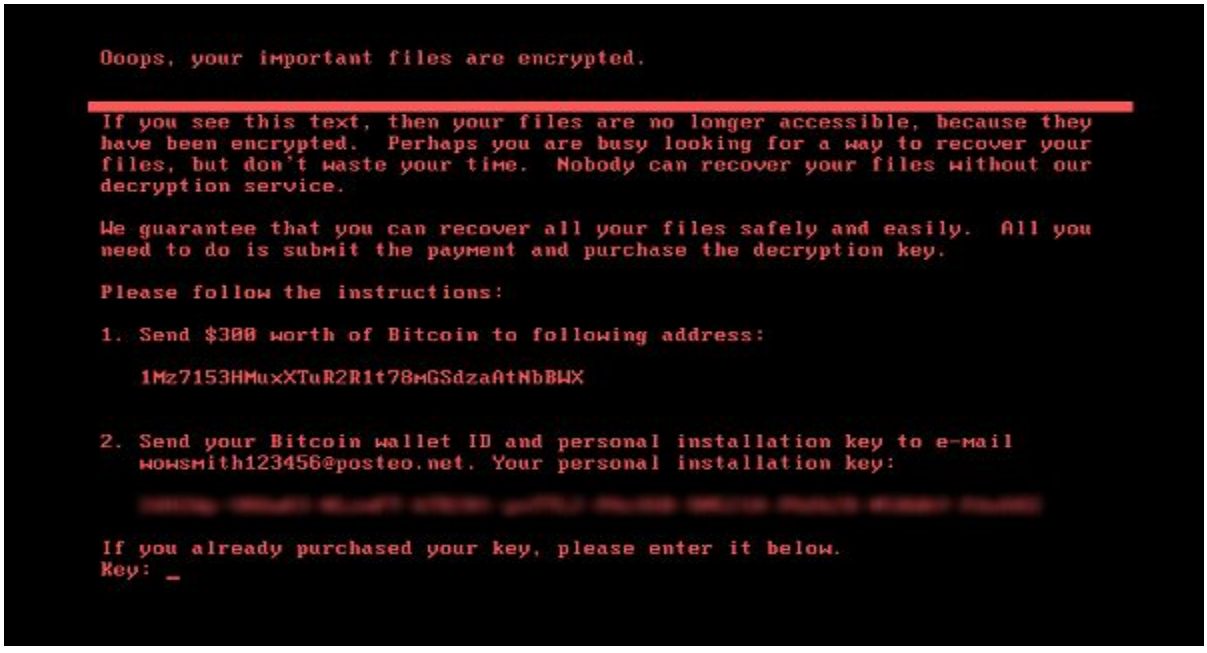
PE Sections			
Name	**MD5**	**Raw Size**	**Entropy**
.text	C5BD3BB710AE377938B17980692B785B	48640	6.55
.rdata	46418E52B546C1F696EB8A524F18C56E	34304	6.99
.data	5216F0C62D1FD41B1D558E129E18D0FE	20992	5.43
.rsrc	F07E68575F50A62382D99E182BAA05D5	247808	8
.reloc	C5D1D4CDADE7DCFBE14EC10DCF66CFB1	3584	4.77

Characteristics

Once this malware is on a computer, it uses a variety of methods to spread across networks. It builds a list of target computers and uses two methods to spread to those computers. Targeted computer include the ones which are on the LAN as well as remote IPs. It checks whether ports *445/139* are open on addresses within the subnet mask as well as on DHCP clients, in addition to computers connected on the same network, those present in the ARP cache and active directory, and neighboring networks. It then builds a list of usernames and passwords from the Windows Credential Manager or by dropping and executing a credential dumper to spread to those targets.

On the target computer, it is executed remotely using either *PsExec* or the *Windows Management Instrumentation Command-line* (WMIC) tool. After execution, contents of *run32.dll* are overwritten with null bytes and then deleted from the disk. In directory *C:\Windows*, a file *perfc* gets created to indicate that the computer has been infected.

Once NotPetya has been installed on a computer, it tries to spread to as many computers as possible, and schedules a reboot. It begins by user-mode encryption, where files having the most common extensions are encrypted. After this, the master boot record (MBR) is modified to add a custom loader which displays a CHKDSK simulator. This simulator is shown during full disk encryption. When both stages of the encryption: user-level and full disk encryption are complete, the computer is restarted with: "*/c at 00:49 C:\Windows\system32\shutdown.exe /r /f*". After reboot, a ransom screen demanding $300 in bitcoin is displayed to the user.

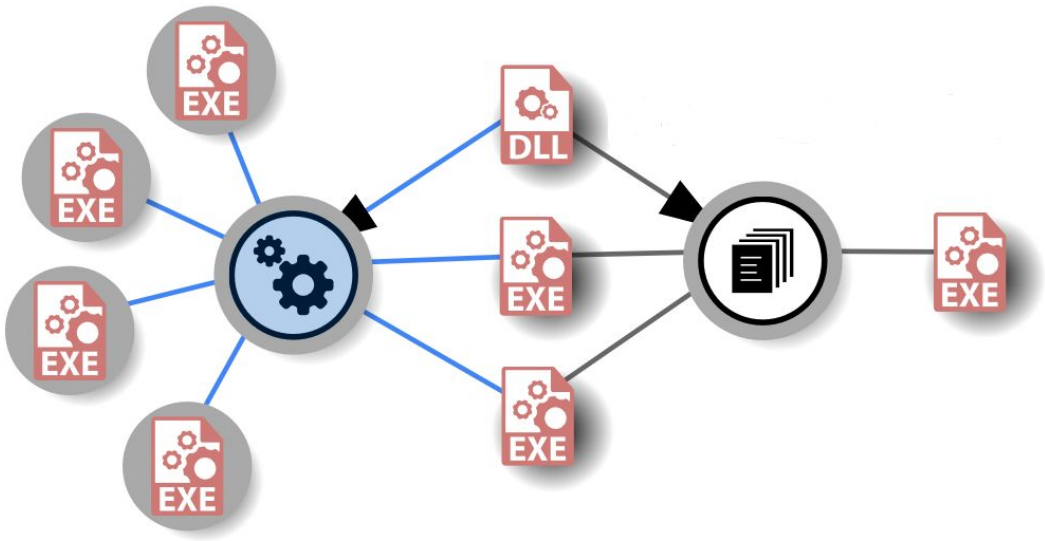


Mitre Attack Matrix

Initial Access, Execution, Persistence, Privilege Escalation, Defense Evasion, Credential Access, Discovery, Lateral Movement, Collection, Exfiltration, Command & Control, Impact

Process Flow

Summary Graph



There are six execution parents (EXEs connected to the gear icon node):

1. Done.exe
2. UniCrypt.exe (4)
3. Scammer-hope.exe

And three PE resource parents (EXEs connected to the files icon node), two of which are *done.exe*, and *scammer-hope.exe* (also common to execution parents) and *dttcodexgigas*.cc*hash*
The DLL file is *NotPetya.bin*

Dependencies

Imports: (List of functions in Import Table (pg. 13))

Dynamic-Link Library (.dll)	Description
KERNEL32.dll	Low-level operating system functions for memory management and resource handling
USER32.dll	Windows management functions for message handling, timers, menus, and communications
ADVAPI32.dll	Designed to support several API's including security and registry calls, restarting and shutting down the system, managing user accounts and starting/stopping/creating Windows services.
SHELL32.dll	Controls certain API functions of the Windows Shell which is a graphical user interface for Windows operating systems consisting of several elements such as the taskbar, desktop and the Start Menu
ole32.dll	Allows objects created in one application to be embedded in documents/objects created by a different applications e.g. embedding
CRYPT32.dll	Module that implements many of the Certificate and Cryptographic Messaging functions in the CryptoAPI
SHLWAPI.dll	Dynamic link library which contains functions for UNC and URL paths, registry entries, and color settings.
IPHLPAPI.dll	Module containing the functions used by the Windows IP Helper API
WS2_32.dll	Module containing Windows Sockets API used by most Internet and network applications to handle network connections
MPR.dll	Dynamic link library associated with Multiple Provider Router, this process enables the OS to interpret information regarding network providers
NETAPI32.dll	Module that contains the Windows NET API used by applications to access a Microsoft network
DHCPSAPI.dll	Dynamic link library associated with configure and management of DHCP services on a Windows which enables developers to create, modify, and delete client leases, view DHCP subnets/scopes and associated elements and view DHCP server bindings
msvcrt.dll	Dynamic link library which provides programs compiled with

	most of the standard C library functions including string manipulation, memory allocation, C-style input/output calls, and others

Behavioral and Code Analysis Findings

Function Calls:

Function	Address	Info
IsWow64Process	[10014034]	Determines whether the specified process is running under WOW64 or an Intel64 of x64 processor
https://docs.microsoft.com/en-us/windows/win32/api/wow64apiset/nf-wow64apiset-iswow64process		
GetExtendedTcpTable	[10014090]	The GetExtendedTcpTable function retrieves a table that contains a list of TCP endpoints available to the application
https://docs.microsoft.com/en-us/windows/win32/api/iphlpapi/nf-iphlpapi-getextendedtcptable		

Commands:

- schtasks %ws/Create /SC once /TN "" /TR "%ws" /ST %02d:%02d
- Address: [100142A8]
- Description:
    - The schtasks command enables an administrator to create, delete, query, change, run, and end scheduled tasks on a local or remote system.

/parameter [arguments]		Description
/Create		Create a new scheduled task
/SC [once]	schedule	Specifies the schedule frequency. Valid schedule types: MINUTE, HOURLY, DAILY, WEEKLY, MONTHLY, ONCE, ONSTART, ONLOGON, ONIDLE, ONEVENT.
/TN [“”]	taskname	Specifies a name which uniquely identifies this scheduled task.
/TR [“%ws”]	taskrun	Specifies the path and file name of the program to be run at the scheduled time. Example: C:\windows\system32\calc.exe
/ST []	starttime	Specifies the start time to run the task. The time format is HH:mm (24 hour time) for example, 14:30 for 2:30 P.M. Defaults to the current time if /ST is not specified. This option is required with /SC ONCE.
Source: https://www.computerhope.com/schtasks.htm		

- shutdown.exe /r /f
- Address: [10014344]
- Description:
    - Enables you to shut down or restart local or remote computers one at a time.

Parameter	Description
/r	Restarts the computer after shutdown
/f	Forces running applications to close without warning users. ***Caution:*** Using the /f option might result in loss of unsaved data
Source: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/shutdown	

Other Findings:

Address	Text
[1000FE11] → [1000FE16]	"1.2.8"
[1000FE17] → [1000FE20]	"Copyright"
[1000FE21] → [1000FE2A]	"1995-2013"
[1000FE2B] → [1000FE2F]	"Mark"
[1000FE30] → [1000FE35]	"Adler"
[1001A910] → [1001A92D]	"Repairing file system on C:"
[1001A932] → [1001A957]	"The type of the file system is NTFS."
[1001A95A] → [1001A996]	"One of your disks contains errors and needs to be repaired."
[1001A997] → [1001A9CA]	"This process may take several hours to complete."
[1001A9CB] → [1001A9FC]	"It is strongly recommended to let it complete."
[1001AA01] → [1001AA23]	"WARNING: DO NOT TURN OFF YOUR PC!"
[1001AA24] → [1001AA65]	"IF YOU ABORT THIS PROCESS, YOU COULD DESTROY ALL OF YOUR DATA!"
[1001AA66] → [1001AA9B]	"PLEASE ENSURE THAT YOUR POWER CABLE IS PLUGGED IN!"
[1001AAA2] → [1001AABD]	"CHKDSK is repairing sector"
[1001AAC0] → [1001AADB]	"Please reboot your computer!"

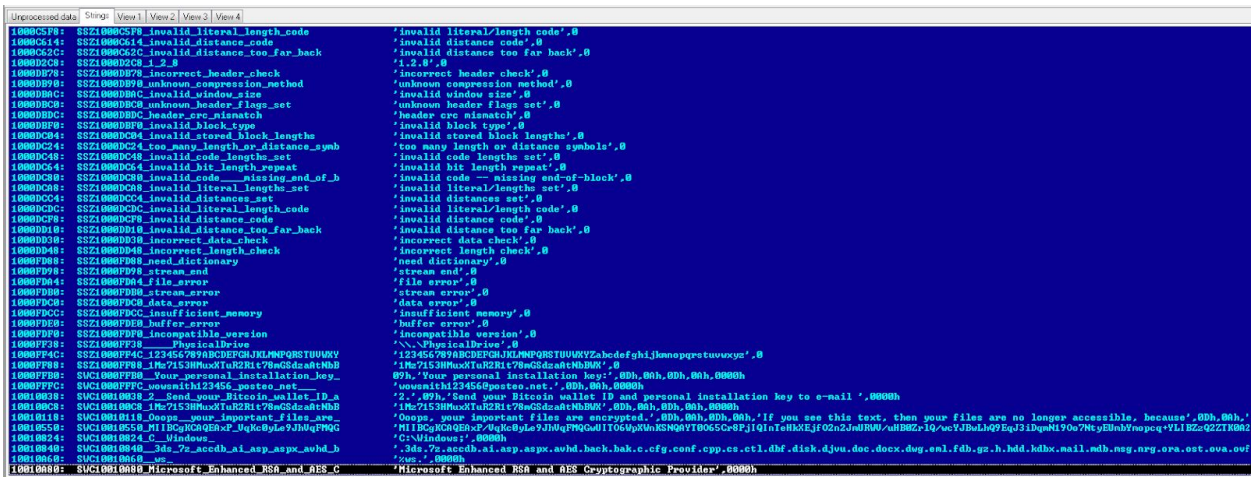
[1001AADE] → [1001AAEF]	“Decrypting sector”
[1001AAF6] → [1001AB20]	“Ooops, your important files are encrypted”
[1001AB26] → [1001AB8A]	“If you see this text, then your files are no longer accessible, because The have been encrypted.”
[1001AB8B] → [1001ABE6]	“Perhaps you are busy looking for a way to recover your files, but don't waste your time.”
[1001ABE7] → [1001AC26]	“Nobody can recover your files without our decryption service.”
[1001AC2B] → [1001AC6F]	“We guarantee that you can recover all your files safely and easily.”
[1001AC70] → [1001ACBB]	“All you need to do is submit the payment and purchase the decryption key.”
[1001ACC0] → [1001ACDF]	“Please follow the instructions:”
[1001ACE4] → [1001AD17]	“1-Send $300 worth of Bitcoin to following address:”
[1001AD30] → [1001AD96]	“2-Send your Bitcoin wallet ID and personal installation key to e-mail wowsmith123456@posteo.net.”
[1001AD97] → [1001ADB6]	“Your personal installation key:”
[1001ADC1] → [1001ADFA]	“If you already purchased your key, please enter the below.”
[1001ADFE] → [1001AE02]	“Key:”
[1001AE07] → [1001AE15]	“Incorrect key!”
[1001AE16] → [1001AE27]	“Please try again.”
[1001B0F8] → [1001B0FD]	“ERROR!”

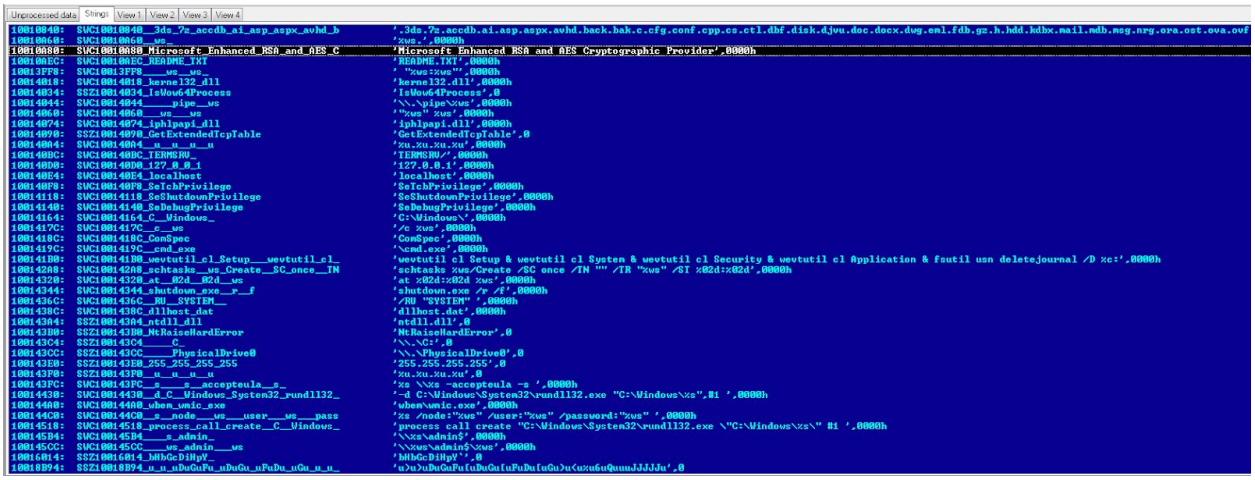
HTTP Network Signatures:

Source IP	Source Port	Destination IP	Destination Port
192.168.2.5	49727	40.90.137.126	80
192.168.2.5	49728	104.103.107.203	80
192.168.2.5	49729	23.37.52.81	80

Supporting Figures

Strings





Problem and Message List

// Generated by PE Explorer 1.99 (http://www.heaventools.com)
// File name: C:\Users\IEUser\Downloads\petya_sample\027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745
// Created : 10.12.2019 19:57
// Type : Problem and Messages List

Line caused error (F3 jmp L1000CC5C) at: 1000CC1Ah

Import Table:

+ KERNEL32.dll

- ConnectNamedPipe
- GetModuleHandleW
- CreateNamedPipeW
- TerminateThread
- DisconnectNamedPipe
- FlushFileBuffers
- GetTempPathW
- GetProcAddress
- DeleteFileW
- FreeLibrary
- GlobalAlloc
- LoadLibraryW
- GetComputerNameExW
- GlobalFree
- ExitProcess
- GetVerionExW
- GetModuleFileNameW
- DisableThreadLibraryCalls
- ResumeThread
- GetEnviornmentVariableW
- GetFileSize
- SetFilePointer
- FindFirstFileW
- CloseHandle
- DeviceControl1
- GetLastError
- GetSystemDirectoryA
- FindResourceW
- SetLastError
- LoadResource
- GetCurrentThread
- OpenProcess
- GetSystemDirectoryW
- SizeofResource
- GetLocalTime
- Process32FirstW
- LockResource
- Process32NextW
- GetModuleHandleA
- 1strcatW
- CreateToolhelp32Snashot
- GetCurrentProcess
- VirtualFree
- VirtualAlloc
- LoadLibraryA
- VirtualProtect
- WideCharToMultiByte
- GetExitCodeProcess
- WaitForMultipleObjects
- CreateProcessW
- ReadFile
- WriteFile
- GetProcessHeap
- InitializeCriticalSection
- HeapReAlloc
- PeekNamedPipe
- GetTempFileNameW
- InterlockedExchange
- LeaveCriticalSection
- MultiByteToWideChar
- CreateFileA
- GetTickCount
- CreateThread
- LocalFree
- FindNextFileW
- CreateFileMappingW
- LocalAlloc
- FindClose
- GetFileSizeEx
- CreateFileW
- Sleep
- FlushViewOfFile
- GetLogicalDrives
- WaitForSingleObject
- GetDriveTypeW
- UnmapViewOfFile
- MapViewOfFile
- GetWindowsDirectoryW
- EnterCriticalSection
- HeapFree
- SetFilePointerEx
- HeapAlloc

+ ADVAPI32.dll

- CryptGenRandom
- CryptAcquireContextA
- CryptExportKey
- CryptAcquireContextW
- CreateProcessAsUserW
- InitiateSystemShutdownEx
- DuplicateTokenEx
- CryptGenKey
- CryptEncrypt
- SetTokenInformation
- GetTokenInformation
- GetSidSubAuthorityCount
- OpenThreadToken
- GetSidSubAuthority
- AdjustTokenPrivleges
- LookupPrivilegeValueW
- CryptImportKey
- CryptSetKeyParam
- OpenProcessToken
- SetThreadToken
- CredEnumerateW
- CredFree
- SetSecurityDescriptorDac1
- InitializeSecurityDescriptor
- CryptDestroyKey
- CryptReleaseContext

+ WS2_32.dll

- WS2_32.12
- WS2_32.52
- WS2_32.151
- WS2_32.14
- WS2_32.3
- WS2_32.11
- WS2_32.18
- WS2_32.16
- WS2_32.19
- WS2_32.9
- WS2_32.23
- WS2_32.115
- WS2_32.10
- WS2_32.4

+ **SHLWAPI.dll**
  - PathAppendW
  - StrToIntW
  - PathFindFileNameW
  - PathFileExistsW
  - StrCmpW
  - StrCmpIW
  - StrChrW
  - StrCatW
  - StrStrW
  - PathFindExtentionW
  - PathCombineW
  - StrStrIW

+ **MPR.dll**
  - WNetOpenEnumW
  - WNetEnumResourceW
  - WNetCancelConnection2W
  - WNetCloseEnum
  - WNetAddConnection2W

+ **DHCPSAPI.dll**
  - DhcpEnumSubnetClient
  - DhcpRpcFreeMemory
  - DhcpGetSubnetInfo
  - DhcpEnumSubnets

+ **CRYPT32.dll**
  - CryptStringToBinaryW
  - CryptBinaryToStringW
  - CryptDecodeObjectEx

+ **USER32.dll**
  - ExitWindowsEx
  - wsprintfA
  - wsprintfW

+ **NETAPI32.dll**
  - NetServerEnum
  - NetApiBufferFree
  - NetServerGetInfo

+ **ole32.dll**
  - CoCreateGuid
  - GoTaskMemFree
  - StringFromCLSID

+ **SHELL32.dll**
  - CommandLineToArgvW
  - SHGetFolderPathW

+ **IPHLPAPI.dll**
  - GetIpNetTable
  - GetAdaptersInfo

+ **msvcrt.dll**
  - malloc
  - _itoa
  - memset
  - free
  - rand
  - memcpy

Incident Recommendations

The best way to mitigate this malware is to have an updated OS. This includes downloading security patches for the operating system at regular intervals as well as not using out of support operating system. These measures prevent its spread across the network and help contain this malware to limited instances. Security analysts have also pointed out that it may be possible to stop encryption if an infected computer is immediately shut down when the *CHKDSK* screen appears. They also recommended to create read-only files named *perf.c* and/or *perfc.dat* in *C:\Windows* directory to possibly prevent the payload from executing.

Additionally, anti-virus software, if installed on a computer should be able to detect and prevent user-level encryption. It is also important to remove any legacy software that are no longer supported. Using them without an updated patch can expose the computer to the malware. Be careful on clicking on phishing emails and popups that could appear online. And always remember to back up any important files or the entire system image. That way, if an attack does occur, there is always something to fall back on.

References

- https://www.virustotal.com/gui/file/027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745/detection

- https://www.crowdstrike.com/blog/petrwrap-ransomware-technical-analysis-triple-threat-file-encryption-mft-encryption-credential-theft/

- https://www.fireeye.com/blog/threat-research/2017/06/petya-ransomware-spreading-via-eternalblue-exploit.html

- https://www.joesandbox.com/analysis/189349/0/pdf

- https://www.vmray.com/analyses/027cc450ef5f/report/overview.html

- https://app.any.run/tasks/6039e784-d260-47d5-af17-787621d9920f/

- https://www.symantec.com/blogs/threat-intelligence/petya-ransomware-wiper

- https://en.wikipedia.org/wiki/Petya_(malware)

- https://www.theguardian.com/technology/2017/jun/27/petya-ransomware-cyber-attack-who-what-why-how