

Projekt - część 2

Analiza Statyczna

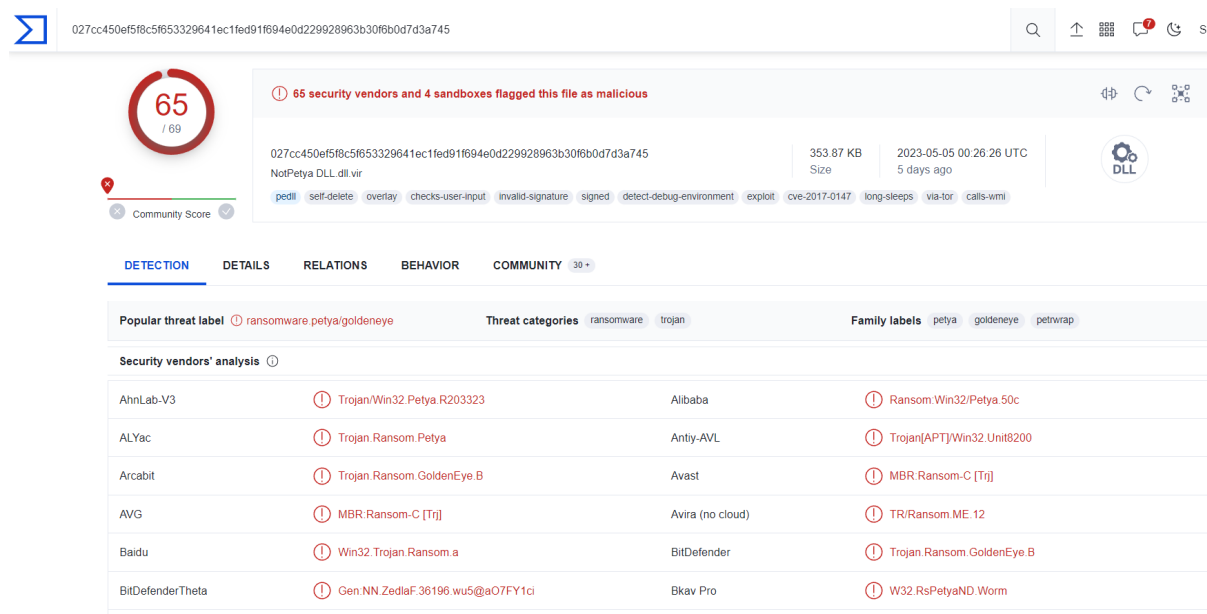
Julia Sadecka, Marcel Trzaskawka, Cyberbezpieczeństwo

NotPetya.....	3
1. VirusTotal.....	3
2. Podstawowa Analiza Statyczna.....	3
3. Podsumowanie podstawowej analizy statycznej.....	11
4. Zaawansowana Analiza Statyczna.....	12
5. Podsumowanie Zaawansowanej Analizy Statycznej.....	16
KeyPass.....	17
1. VirusTotal.....	17
2. Wstępna Analiza Statyczna.....	17
3. Podsumowanie Wstępnej Analizy Statycznej.....	21
4. Zaawansowana Analiza Statyczna.....	21
5. Podsumowanie.....	25

Wirus	MD5	SHA-256
NotPetya	71b6a493388e7d0b40c83ce903bc6b04	027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745
KeyPass Ransomware	6999c944d1c98b2739d015448c99a291	35b067642173874bd2766da0d108401b4cf45d6e2a8b3971d95bf474be4f6282

NotPetya

1. VirusTotal



VirusTotal wykazał, że 65/69 antywirusów zidentyfikowało ten plik jako szkodliwy (malware).

2. Podstawowa Analiza Statyczna

a) Data Kompilacji

TimeDateStamp mówi o tym, że wirus został skompilowany 18 czerwca 2017 o godzinie 7:14 UTC

Member	Value	Comment
Machine	014C	Intel 386
NumberOfSections	0005	
TimeDateStamp	5946285C	Sun, 18 Jun 2017 07:14:36 UTC (2160 days, 8.54 hours ago)
PointerToSymbolTable	00000000	
NumberOfSymbols	00000000	

b) Spakowanie i zaciemnienie

Plik nie jest spakowany. Nie jest także zaciemniony, ponieważ w sekcji nagłówków występują typowe nagłówki dla programu PE.

PPEE - C:\Users\julia\Desktop\MALWARE\NotPetya.bin

File Plugins Help

Name	VirtualAd...	VirtualSize	RawAddr...	RawSize	PtrToRelocs	PtrToLine...	NumOfRe...	NumbOfL...	Characteri...
.text	00001000	0000BD63	00000400	0000BE00	00000000	00000000	0000	0000	60000020
.rdata	0000D000	00008546	0000C200	00008600	00000000	00000000	0000	0000	40000040
.data	00016000	00009B4A	00014800	00005200	00000000	00000000	0000	0000	C0000040
.rsrc	00020000	0003C738	00019A00	0003C800	00000000	00000000	0000	0000	40000040
.reloc	0005D000	00000C02	00056200	00000E00	00000000	00000000	0000	0000	42000040

Section Headers

DIRECTORY_ENTRY_EXPORT

c) Biblioteki

PPEE - C:\Users\julia\Desktop\MALWARE\NotPetya.bin

File Plugins Help

Name RVA	Name	OriginalFirstThunk	TimeDate Stamp	ForwarderChain	FirstThunk	Description (Read from file)
00014F86	KERNEL32.dll	000147A4	00000000	00000000	0000D09C	Biblioteka DLL klienta Windows NT BASE API
00014FBC	USER32.dll	00014958	00000000	00000000	0000D250	Współużytkowana biblioteka DLL klienta Windows
000151EE	ADVAPI32.dll	00014708	00000000	00000000	0000D000	Advanced Windows 32 Base API
00015226	SHELL32.dll	00014918	00000000	00000000	0000D210	Wspólna biblioteka DLL Powłoki systemu Windows
00015264	ole32.dll	000149C0	00000000	00000000	0000D2B8	Microsoft OLE for Windows
000152B4	CRYPT32.dll	00014774	00000000	00000000	0000D06C	Crypto API32
00015366	SHLWAPI.dll	00014924	00000000	00000000	0000D21C	Biblioteka dodatkowych narzędzi powłoki
00015394	IPHLPAPI.DLL	00014798	00000000	00000000	0000D090	IP Helper API
000153A2	WS2_32.dll	00014968	00000000	00000000	0000D260	Biblioteka DLL 32-bitowej wersji usługi Windows So
00015412	MPR.dll	000148F0	00000000	00000000	0000D1E8	Multiple Provider Router DLL
00015452	NETAPI32.dll	00014908	00000000	00000000	0000D200	Net Win32 API DLL
000154B2	DHCPAPI.DLL	00014784	00000000	00000000	0000D07C	Wejściowy DLL interfejsu API serwera DHCP
000154C8	msvcrt.dll	000149A4	00000000	00000000	0000D29C	Windows NT CRT DLL

W pliku występuje 13 bibliotek

KERNEL32.dll	KERNEL32.dll-kont.	ADVAPI32.dll	SHLWAPI.dll
ConnectNamePipe	InitializeCriticalSection	AdjustTokenPrivileges	StrToIntW
CreateFileA	InterlockedExchange	CreateProcessAsUser	StrStrW
CreateFileMapping	LeaveCriticalSection	CredEnumerate	StrCmpW
CreateFileW	LoadLibrary	CredFree	StrChr, StrCat
CreateNamePipeW	LoadResource	CryptAcquireContext	PathFindFileName
CreateProcessW	LocalAlloc, LocalFree	CryptDestroyKey	PathFindExtension
CreateThread	LockResource	CryptEncrypt	PathFileExists
CreateToolhelp32Snapshot	MapViewOfFile	CryptExportKey	PathCombine, PathAppend
DeleteFile	MultiByteToWideChar	CryptGenKey	
DeviceIoControl	OpenProcess	CryptGenRandom	MPR.dll
DisableThreadLibraryCalls	PeeekNamePipe	CryptImportKey	WNetOpenEnumW

DisconnectNamedPipe	Process32First, Process32Next	CryptReleaseContext	WNetEnumResourceW
EntryCriticalSection	ReadFile	CryptSetKeyParam	WNetCancelConnection, WNetAddConnection
ExitProcess	ResumeThread	DuplicateTokenEx	WNetCloseEnum
FindClose	SetFilePointer	GetSidSubAuthority	
FindFirstFile, FindNextFile, FindResource	SetLastError	GetSidSubAuthorityCo unt	NETAPI32.dll
FlushFileBuffers	SizeOfResource	GetTokenInformation	NetServerEnum
FlushViewOfFile	Sleep	InitializeSecurityDescri ptor	NetApiBufferFree
FreeLibrary	TerminateThread	InitiateSystemShutdow nExW	NetSeerverGetInfo
GetComputerNameExW	UnmapViewOfFile	LookupPrivilegeValue	DHCPAPI.DLL
GetCurrentProcess	VirtualAlloc, VirtualFree, VirtualProtect	OpenProcessToken	DhcpEnumSubnetClients
GetCurrentThread	WaitForMultipleObjects, WaitForSingleObject	OpenThreadToken	DhcpRpcFreeMemory
GetDriveType	WideCharToMultiByte	SetSecurityDescriptorD acl	DhcpGetSubnetInfo
GetEnvironmentVariable W	WriteFile	SetThreadToken	DhcpEnumSubnet
GetExitCodeProcess	IstrcatW	SetTokenInformation	
GetFileSize			msvcrt.dll
GetLastError	USER32.dll	ole32.dll	
GetLocalTime, GetLocalDrives	ExitWindowsEx	CoCreateGuid	WS2_32.dll
GetModuleFileName, GetModuleHandle	wsprintf	CoTaskMemFree	
GetProcAddress, GetProcessHeap		StringFromCLSID	
GetSystemDirectory	SHELL32.dll		
GetTempFileName,	CommandLineToArgv	CRYPT32.dll	

GetTempPath			
GetTickCount	SHGetFolderPath	CryptStringToBinary	
GetVersion		CryptBinaryToString	
GetWindowsDirectoryW	IPHLPAPI.DLL	CryptDecodeObjectEx	
GlobalAlloc, GlobalFree	GetIpNetTable		
HeapAlloc, HeapFree, HeapReAlloc	GetAdaptersInfo		

- **KERNEL32.dll** - W tej bibliotece występują takie funkcje, które
 - manipulują procesami (*CreateProcess*, *OpenProcess*, *Process32First*) - jeżeli malware stworzyło nowy proces trzeba będzie go przeanalizować w dalszej części
 - manipulują plikami (*CreateFile*, *GetFileSize*, *ReadFile*)
 - manipulują pamięcią (*GlobalAlloc*, *HeapFree*). Można zwrócić szczególną uwagę na funkcje *VirtualAlloc/VirtualFree/VirtualProtect*, które są często używane do manipulowania pamięcią wirtualną procesu. Malware może je wykorzystać do wstrzykiwania kodu, omijania zabezpieczeń czy ukrywania się przed wykryciem
 - funkcje mogące służyć do przeszukiwania katalogów: *FindFirstFile*, *FindNextFile*, *GetSystemDirectory*, *GetWindowsDirectoryW*.
 - funkcje, które sprawdzają szczegóły systemu takie jak nazwa komputera (*GetComputerNameEx*), lokalny czas (*GetLocalTime*), numer wersji systemu operacyjnego (*GetVersion*) - mogą one zostać wykorzystane jako część badań ofiary lub do wyboru odpowiedniego offsetu dla danego systemu Windows.
 - funkcja *GetModuleFilename* może zostać użyta do modyfikowania i kopiowania plików w trakcie bieżącego procesu.
- **USER32.dll** - W tej bibliotece znajdują się dwie ciekawe funkcje: *ExitWindowsEx* i *wsprintf*.
 - *ExitWindowsEx* - jest używana do wylogowania użytkownika, restartu systemu lub wyłączenia komputera.
 - *wsprintf* natomiast może być używana do tworzenia sformatowanych komunikatów lub logów. Malware może wykorzystać tę funkcję do generowania złośliwych komunikatów dla użytkownika, wyświetlania fałszywych ostrzeżeń lub wprowadzania w błąd.

- ADVAPI32.dll - Ta biblioteka wydaje się najciekawsza, ponieważ posiada sporo funkcji, które mogą zdradzać co malware wykonuje na komputerze:
 - Manipulacja uprawnieniami i tokenami zabezpieczeń: Funkcje takie jak *AdjustTokenPrivileges*, *DuplicateTokenEx*, *OpenProcessToken* i *SetThreadToken* umożliwiają malware zmianę uprawnień procesów i wątków oraz manipulację tokenami zabezpieczeń. To może prowadzić do podniesienia uprawnień, uzyskania dostępu do poufnych zasobów lub uniknięcia wykrycia przez oprogramowanie antywirusowe.
 - Zarządzanie poświadczeniami: Funkcje takie jak *CredEnumerate*, *CredFree* umożliwiają malware przeglądanie, pobieranie i manipulację informacjami uwierzytelniającymi przechowywanymi w systemie. To może być wykorzystane do kradzieży poświadczeń użytkowników, takich jak hasła czy tokeny dostępowe.
 - Operacje kryptograficzne: Funkcje takie jak *CryptAcquireContext*, *CryptDestroyKey*, *CryptEncrypt*, *CryptExportKey*, *CryptGenKey*, *CryptGenRandom*, *CryptImportKey* umożliwiają malware wykonywanie operacji kryptografic

znych. Malware może używać tych funkcji do szyfrowania danych, generowania lub importowania kluczy kryptograficznych lub generowania losowych danych.

- Kontrola dostępu i uprawnień: Funkcje takie jak *GetTokenInformation*, *LookupPrivilegeValue*, *SetSecurityDescriptorDacl*, *SetTokenInformation* umożliwiają malware manipulację kontrolą dostępu i uprawnieniami. To może obejmować zmianę uprawnień, ustawianie atrybutów lub modyfikację listy kontroli dostępu dla obiektów systemowych.
- SHELL32.dll - funkcja *SHGetFolderPath* - zwraca ścieżkę do folderów, także tych systemowych
- ole32.dll
- CRYPT32.dll - funkcje z tej biblioteki służą do operacji kryptograficznych (konwertuje dane binarne na ciągi kryptograficznie i na odwrót)
- SHLWAPI.dll - zawiera funkcje pomocnicze związane z manipulacją ciągami znaków, ścieżkami plików i innymi operacjami na danych tekstowych. Funkcje w niej wyszukują, porównują i analizują ciągi znaków oraz operacje na ścieżkach plików.
- IPHLPAPI.DLL - funkcje w tej bibliotece mogą dostarczyć informacje na temat połączeń sieciowych, interfejsów sieciowych czy ich konfiguracji.

- WS2_32.dll - biblioteki zapewniające dostęp do sieci. Prawdopodobnie program łączy się z siecią dzięki niej
- MPR.dll
- NETAPI32.dll
- Biblioteka DHCPAPI.DLL i jej funkcje są używane w kontekście zarządzania usługą DHCP, analizy konfiguracji sieci, monitorowania klientów DHCP i manipulacji informacjami dotyczącymi podsieci.
- msvcr7.dll

d) Podejrzane stringi

Ze stringów możemy wyciągnąć, że malware szyfrował ważne pliki i żądał okupu w postaci \$300 w postaci bitcoinów.

0000F1B2	Your personal installation key:
0000F1FC	wowsmith123456@posteo.net.
0000F23E	Send your Bitcoin wallet ID and personal installation key to e-mail
0000F318	Ooops, your important files are encrypted.
0000F374	If you see this text, then your files are no longer accessible, because
0000F406	they have been encrypted. Perhaps you are busy looking for a way to recover
0000F4A0	your files, but don't waste your time. Nobody can recover your files without
0000F53C	our decryption service.
0000F572	We guarantee that you can recover all your files safely and easily.
0000F5FC	All you need to do is submit the payment and purchase the decryption key.
0000F696	Please follow the instructions:
0000F6E2	Send \$300 worth of Bitcoin to following address:

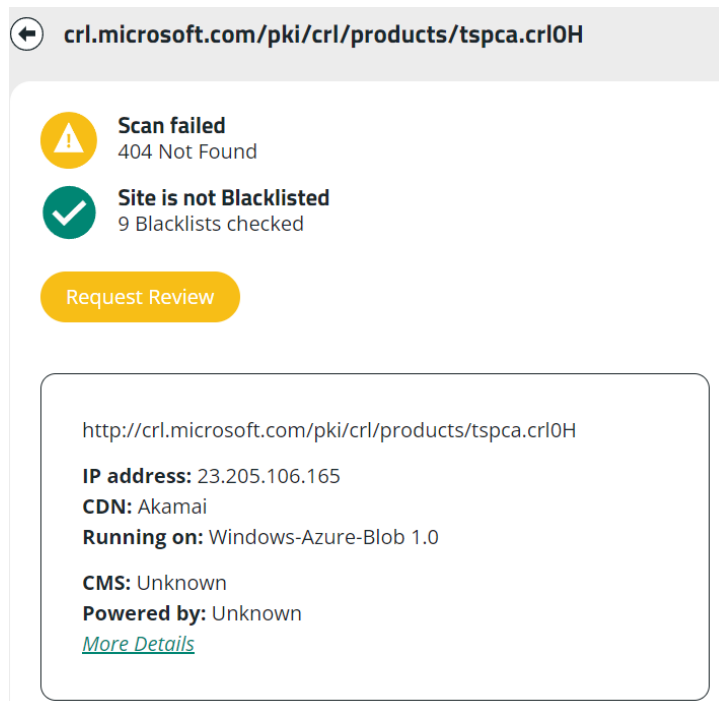
Microsoft Enhanced RSA and AES Cryptographic Provider to narzędzie, które dostarcza usprawnione algorytmy kryptograficzne RSA i AES w systemach operacyjnych Windows. Bardzo możliwe, że właśnie tym narzędziem zostały zaszyfrowane pliki.

0000FC80	Microsoft Enhanced RSA and AES Cryptographic Provider
----------	-------------------------------------------------------

e) Połączenie z Internetem

Offset	Type	Strings recognized URL
000577EE	ASCII	http://crl.microsoft.com/pki/crl/products/CSPCA.crl0H
00058118	ASCII	http://crl.microsoft.com/pki/crl/products/tspca.crl0H
00058415	ASCII	http://technet.microsoft.com/sysinternals 0
0005783F	ASCII	http://www.microsoft.com/pki/certs/CSPCA.crt0
00058169	ASCII	http://www.microsoft.com/pki/certs/tspca.crt0

Po sprawdzeniu stron URL okazało się, że tylko jeden link działał i przenosi na stronę microsoftu. Reszta była nieaktywna - strony nie istniały.



f) Suspicious

W Suspicious znajdujemy także takie podejrzane wyniki jak:

- Key
- Root
- ping
- rundll32.exe
- \\.\C:

Offset	Type	Strings found
000132D0	UNICODE	127.0.0.1
0000FA24	UNICODE	C:\Windows;
00013364	UNICODE	C:\Windows\
00013633	UNICODE	C:\Windows\System32\rundll32.exe "C:\Windows\%s",#1
0001372D	UNICODE	C:\Windows\System32\rundll32.exe \"C:\Windows\%s\" #1
000146AA	ASCII	Clients
0001F811	ASCII	DEL
00013F08	ASCII	DeleteFileW
00014215	ASCII	Key
00014253	ASCII	Key
00014272	ASCII	Key
00014284	ASCII	Key
000195FF	ASCII	Key:
0001423C	ASCII	KeyParam
00057395	ASCII	Root Authority
00057C83	ASCII	Root Authority
00057125	ASCII	Root Authority0
0001461F	ASCII	ServerEnum
00014643	ASCII	ServerGetInfo
000135C4	ASCII	\\.\C:
0000F138	ASCII	\\.\PhysicalDrive
000135CC	ASCII	\\.\PhysicalDrive0
00000496	ASCII	\\.\f
00013244	UNICODE	\\.\pipe\%ws
000137B9	UNICODE	admin\$
000137D2	UNICODE	admin\$\%ws

Aktywuj system Windows
Przejdź do ustawień, aby aktywow

Offset	Type	Strings found
0000F138	ASCII	\\.\PhysicalDrive
000135CC	ASCII	\\.\PhysicalDrive0
00000496	ASCII	\\.\f
00013244	UNICODE	\\.\pipe\%ws
000137B9	UNICODE	admin\$
000137D2	UNICODE	admin\$\%ws
00015468	UNICODE	c:\Windows\
00016CF0	UNICODE	c:\Windows\
0001339D	UNICODE	cmd.exe
00013415	UNICODE	deletejournal /D %c:
0001956A	ASCII	key to e-mail
0000F274	UNICODE	key to e-mail
00019612	ASCII	key! Please try again.
000195E0	ASCII	key, please enter it below.
000194B8	ASCII	key.
0000F641	UNICODE	key.
000195B3	ASCII	key:
0000F1CD	UNICODE	key:
000136DC	UNICODE	password:"%ws"
000585E6	ASCII	ping PCA
00057AB2	ASCII	ping PCA0
00057EBB	ASCII	ping PCA0
00013D49	ASCII	pingW
00015480	UNICODE	rundll32.exe
00016CD0	UNICODE	rundll32.exe
00013647	UNICODE	rundll32.exe "C:\Windows\%s",#1
00013741	UNICODE	rundll32.exe \"%C:\Windows\%s\" #1
0000FB08	UNICODE	vbox.vbs.vcb.vdi.vfd.vmc.vmdk.vmsd.vmx.vsd.vsv.vsw.work.xls.xlsx.xvd.zip.

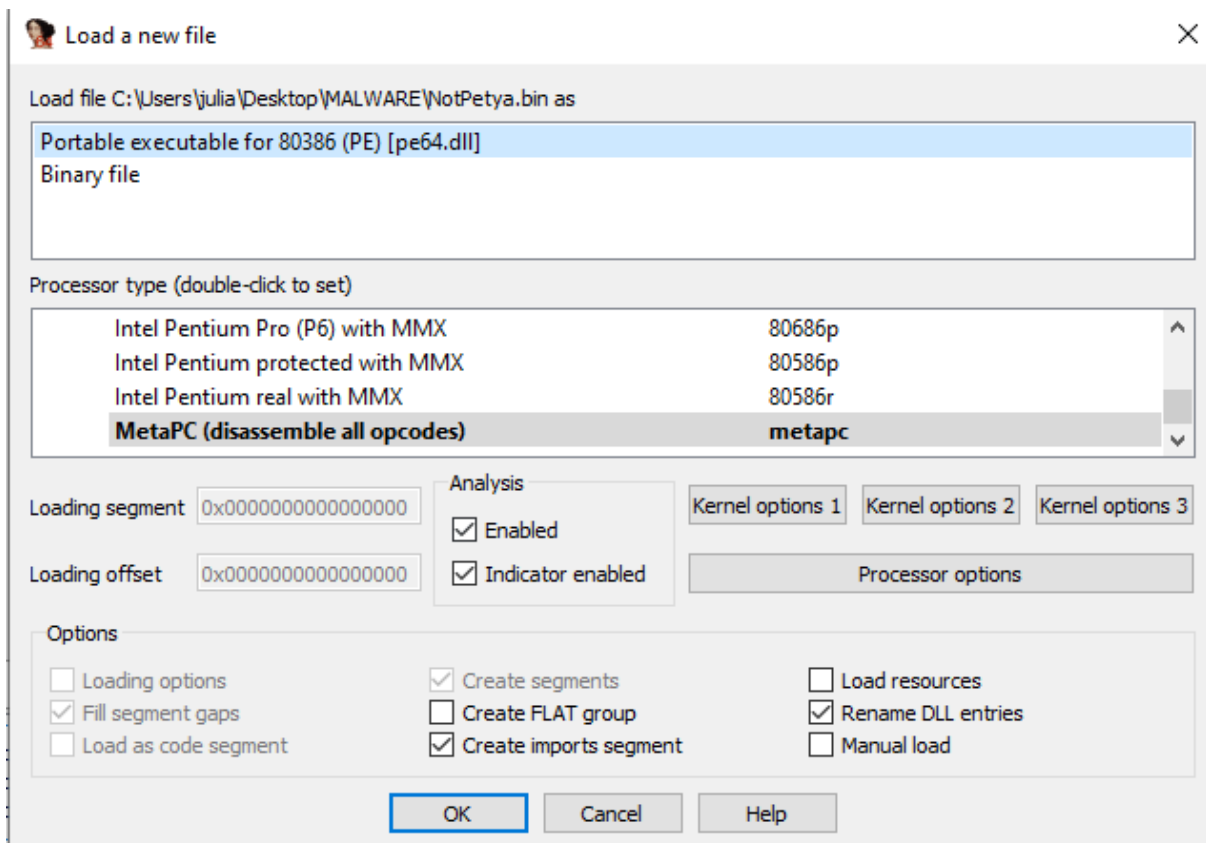
3. Podsumowanie podstawowej analizy statycznej

Z podstawowej analizy możemy wnioskować, że malware przeszukuje foldery zakażonego systemu. Stara się także zwiększyć swoje uprawnienia. Szuka odpowiednich folderów i następnie je szyfruje np. za pomocą biblioteki CRYPT32.dll. Malware posiada także sporo bibliotek sugerujących, że łączy się z internetem (NETAPI32.dll, DHCPAPI.DLL). Może on tym źródłem np. przysyłać klucz do zaszyfrowania plików. Następnie, gdy pliki są już zaszyfrowane wysyła komunikat (np. za pomocą funkcji wsprintf) do użytkownika z żądaniem okupu.

4. Zaawansowana Analiza Statyczna

a) Wstęp

Do Zaawansowanej Analizy Dynamicznej używamy programu IDA. W ustawieniach dodaję Auto Komentarze, Line Prefix i Number of Opcode Bytes (6).



Plik ma format PE, Processor type - MetaPC (deasemblacja dla wszystkich typów)

b) Exporty

Pierwszy z nich prowadzi do początku programu wykonywalnego.

Name	Address	Ordinal
f	0000000010007DEB	1
f DllEntryPoint	0000000010007D39	[main entry]

c) Zwiększenie uprawnień

Malware wywołuje takie funkcje jak:

- SeshutdownPrivilage
- SeTBCPrivilege
- SeDebugPrivilage

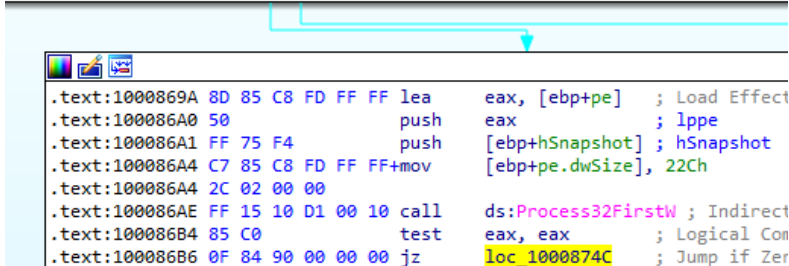
Dzięki nim zwiększa swoje uprawnienia

Po tych próbach wywołuje funkcje o adresie sub_10008677. W tej funkcji tworzy migawkę procesów, a następnie prawdopodobnie za pomocą funkcji Process32First i Process32Next pobiera informacje o procesach.

```

push    ebp
mov     ebp, esp
02 00 00 sub     esp, 238h          ; Integer Subtraction
FF      or      [ebp+var_4], 0FFFFFFFh ; Logical Inclusive OR
push    0          ; th32ProcessID
push    2          ; dwFlags
D1 00 10 call    ds:CreateToolhelp32Snapshot ; Indirect Call Near Procedure
mov     [ebp+hSnapshot], eax
cmp     eax, 0FFFFFFFh ; Compare Two Operands
00 00 00 jz      loc_10008755 ; Jump if Zero (ZF=1)

```

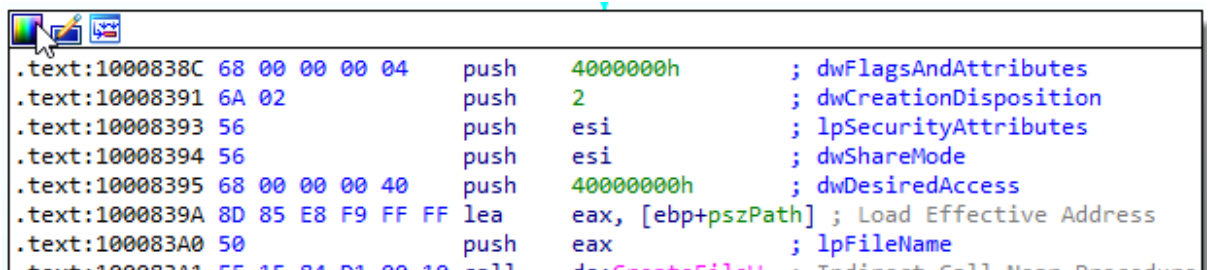


```

.text:1000869A 8D 85 C8 FD FF FF lea     eax, [ebp+pe] ; Load Effective Address
.text:100086A0 50                push    eax ; lppe
.text:100086A1 FF 75 F4        push    [ebp+hSnapshot] ; hSnapshot
.text:100086A4 C7 85 C8 FD FF FF mov     [ebp+pe.dwSize], 22Ch
.text:100086A4 2C 02 00 00
.text:100086AE FF 15 10 D1 00 10 call    ds:Process32FirstW ; Indirect Call Near Procedure
.text:100086B4 85 C0            test     eax, eax ; Logical Compare
.text:100086B6 0F 84 90 00 00 00 jz      loc_1000874C ; Jump if Zero (ZF=1)

```

W późniejszym czasie przeszukując pliki w systemie



```

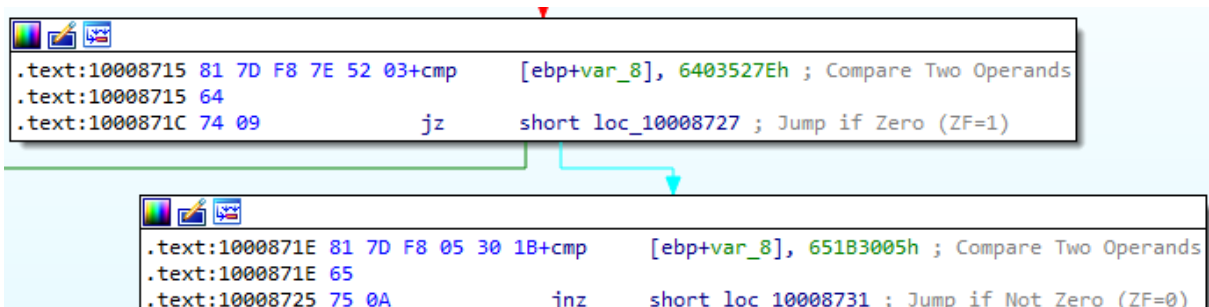
.text:1000838C 68 00 00 00 04 push    4000000h ; dwFlagsAndAttributes
.text:10008391 6A 02          push    2 ; dwCreationDisposition
.text:10008393 56            push    esi ; lpSecurityAttributes
.text:10008394 56            push    esi ; dwShareMode
.text:10008395 68 00 00 00 40 push    40000000h ; dwDesiredAccess
.text:1000839A 8D 85 E8 F9 FF FF lea     eax, [ebp+pszPath] ; Load Effective Address
.text:100083A0 50            push    eax ; lpFileName
.text:100083A1 FF 15 10 D1 00 10 call    ds:CreateFileW ; Indirect Call Near Procedure

```

d) Sprawdzenie posiadania antywirusa

Malware sprawdza czy na komputerze jest zainstalowany antywirus:

- 6403527Eh - Kaspersky
- 651B3005 - Norton



```

.text:10008715 81 7D F8 7E 52 03+cmp [ebp+var_8], 6403527Eh ; Compare Two Operands
.text:10008715 64
.text:1000871C 74 09          jz      short loc_10008727 ; Jump if Zero (ZF=1)

```

```

.text:1000871E 81 7D F8 05 30 1B+cmp [ebp+var_8], 651B3005h ; Compare Two Operands
.text:1000871E 65
.text:10008725 75 0A          jnz     short loc_10008731 ; Jump if Not Zero (ZF=0)

```

e) Wyłączenie komputera

Malware pobiera lokalny czas, a także informację o tym jak dawno system został uruchomiony. Zdobywa także folder systemowy.

```
call    ds:GetLocalTime ;
call    ds:GetTickCount
call    ds:GetSystemDirectoryW ; Indirect Call Near Proceed
test    eax, eax        ; Logical Compare
jz      loc_100085C9     ; Jump if Zero (ZF=1)
0      push    offset pszMore ; "shutdown.exe /r /f"
```

Następnie sprawdza wersję systemu. Wszystko po to, aby dobrać odpowiednią komendę do wyłączenia komputera np. "schtasks %ws/Create ..."

```
call    ds:GetVersionExW ;
test    eax, eax        ; L
jz      short loc_100084DA
text "UTF-16LE", 'schtasks %ws/Create /SC once /TN "" /TR "%ws" /ST %'
```

f) Sprawdzenie połączenia z siecią

Malware zbiera informacje o użytkowniku: sprawdza nazwę komputera (GetComputerNameEx),

```
call    ds:GetComputerNameExW
```

domenę, nazwę serwera, tablicę routingu (a w niej można znaleźć informacje o sieciach docelowych, bramach domyślnych, interfejsach sieciowych)(GetIpNetTable).

```
push    esi                ; servername

push    [ebp+domain]      ; domain

push    eax                ; SizePointer
push    edi                ; IpNetTable
mov     [ebp+var_10], edi
mov     [ebp+SizePointer], edi
call    esi ; GetIpNetTable ; Indirect Cal
```

Znajduje też informacje o konfiguracji i stanie serwera sieciowego (NetServerGetInfo). Sprawdza czy serwer DHCP jest włączony

```

push    ebx            ; lpThreadId
push    ebx            ; dwCreationFlags
push    edi            ; lpParameter
push    offset sub_10008E7F ; lpStartAddress
push    ebx            ; dwStackSize
push    ebx            ; lpThreadAttributes
call    ds:CreateThread ; Indirect Call Near Proc
xor     esi, esi       ; Logical Exclusive OR
call    ds:NetServerGetInfo ; !
xor     ebx, ebx       ; Logical Exclusive OR
push    65h ; 'e'      ; level
lea     ecx, [edi-55h] ; Load Effective Address
push    ebx            ; servername
and     esi, esi       ; Logical AND

```

g) Procesy

Funkcją GetCurrentProcess wirus wywołuje proces, a następnie sprawdza czy maszyna jest 64 bitowa (IsWow64Process)

```

call    ds:GetCurrentProcess ; Indirect Call Near Proc
push    offset ProcName ; "IsWow64Process"
push    offset ModuleName ; "kernel32.dll"

```

Jeśli maszyna jest 32-bitowa, odblokowuje zasób z sekcji RT_RCDATA w pamięci, (który jest kopią malwaru dla tego typu procesora), aby go uruchomić na zdalnych maszynach.

Tworzy plik o losowej nazwie w ścieżce

C:\DOCUMENT1\ADMINI1\LOCALS~1\Temp\B0.tmp i zapisuje w nim zasób.

h) dllhost.dat

Plik dllhost.dll, którego to rozszerzenie powszechnie jest uważane za niebezpieczne.

```

call    ds:GetWindowsDirectoryW
push    offset aDllhostDat ; "dllhost.dat"
push    lpMem              ; pszPath
call    ds:PathAppendW ; Indirect Call Near Procedure
jmp     short loc_10008A6D ; Jump

```

i) szyfrowanie

Klucz publiczny, który jest generowany przez Microsoft Enhanced RSA and AES Cryptographic Provider

```

; DATA XREF: sub_10001BA0+53↑o
; sub_10001EEF+5B↑o ...
text "UTF-16LE", 'MIIBCgKCAQEAxP/VqKc0yLe9JhVqFMQGWUITO6WpXWnKSNQAYT0'
text "UTF-16LE", '065Cr8PjIQInTeHkXEjF02n2JmURlW/uHB0Zr1Q/wcYJBwLhQ9E'
text "UTF-16LE", 'qJ3iDqmN190o7NtyEUmbYmopcq+YLIBZzQ2ZTK0A2DtX4GRKxEE'
text "UTF-16LE", 'FLCy7vP12EYOPXknVy/+mf0JFWixz29QiTF5oLu15wVLONCuEib'
text "UTF-16LE", 'GaNNpgq+CXsPwfITDbDDmdrRIiUEUw6o3pt5pN0skf0JbMan2TZ'
text "UTF-16LE", 'u6zfHzuts7KafP5UA8/0Hmf5K3/F9Mf9SE68EZjK+cIiFlKewnd'
text "UTF-16LE", 'P0XfRCYXI9AJYCea0u7CXF6U0AVNnNjvLeOn42LHFUK4o6JwIDA'
text "UTF-16LE", 'QAB',0
align 4

```

Malware wybiera tylko konkretne rozszerzenia plików:

```

; .data:10018BD4+0
text "UTF-16LE", '.3ds.7z.accdb.ai.aspx.avhd.back.bak.c.cfg.conf.'
text "UTF-16LE", 'cpp.cs.ctl.dbf.disk.djvu.doc.docx.dwg.eml.fdb.gz.h.'
text "UTF-16LE", 'hdd.kdbx.mail.mdb.msg.nrg.ora.ost.ova.ovf.pdf.php.p'
text "UTF-16LE", 'mf.ppt.pptx.pst.pvi.py.pyc.rar.rtf.sln.sql.tar.vbox'
text "UTF-16LE", '.vbs.vcb.vdi.vfd.vmc.vmdk.vmsd.vmx.vsd.vsv.work.xl'
text "UTF-16LE", 's.xlsx.xvd.zip.',0
align 10h

```

j) README.TXT

Malware tworzy plik o nazwie README.TXT. Następnie zapisuje w nim notatkę:

```

; .data:10018C3C+0
text "UTF-16LE", 'Oops, your important files are encrypted.',0Dh,0Ah
text "UTF-16LE", 0Dh,0Ah
text "UTF-16LE", 'If you see this text, then your files are no longer'
text "UTF-16LE", ' accessible, because',0Dh,0Ah
text "UTF-16LE", 'they have been encrypted. Perhaps you are busy look'
text "UTF-16LE", 'ing for a way to recover',0Dh,0Ah
text "UTF-16LE", 'your files, but don',27h,'t waste your time. Nobody'
text "UTF-16LE", ' can recover your files without',0Dh,0Ah
text "UTF-16LE", 'our decryption service.',0Dh,0Ah
text "UTF-16LE", 0Dh,0Ah
text "UTF-16LE", 'We guarantee that you can recover all your files sa'
text "UTF-16LE", 'fely and easily.',0Dh,0Ah
text "UTF-16LE", 'All you need to do is submit the payment and purcha'
text "UTF-16LE", 'se the decryption key.',0Dh,0Ah
text "UTF-16LE", 0Dh,0Ah
text "UTF-16LE", 'Please follow the instructions:',0Dh,0Ah
text "UTF-16LE", 0Dh,0Ah
text "UTF-16LE", '1.',9,'Send $300 worth of Bitcoin to following addr'
text "UTF-16LE", 'ess:',0Dh,0Ah
text "UTF-16LE", 0Dh,0Ah,0

```

5. Podsumowanie Zaawansowanej Analizy Statycznej

Po zaawansowanej analizie statycznej możemy powiedzieć, że NotPetya to wirus szantażujący, który zależnie od znalezionych antywirusów może działać inaczej. Może on wykorzystywać podatność związaną z Eternal Blue. Po zainfekowaniu wirus przeszukuje system np. sprawdza liczbę serwerów i sprawdza czy włączone są

serwery DHCP. Prawdopodobnie chce ona się przez nie przenieść do innych komputerów. Następnie szyfruje pliki z odpowiednim rozszerzeniem przy użyciu losowego klucza. Po tym wyświetla wiadomość z żądaniem okupu. W tej wiadomości jest podany klucz instalacyjny.

KeyPass

1. VirusTotal

35b067642173874bd2766da0d108401b4c45d6e2a8b3971d95bf474be4f6282

59 / 70

59 security vendors and 3 sandboxes flagged this file as malicious

Win32.KeyPass.bin

2.82 MB Size

2023-04-27 13:32:56 UTC 13 days ago

EXE

peexe runtime-modules detect-debug-environment long-sleeps direct-cpu-clock-access checks-user-input

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 15

Popular threat label **ransomware.keypass/encoder** Threat categories ransomware trojan Family labels keypass encoder stop

Security vendors' analysis

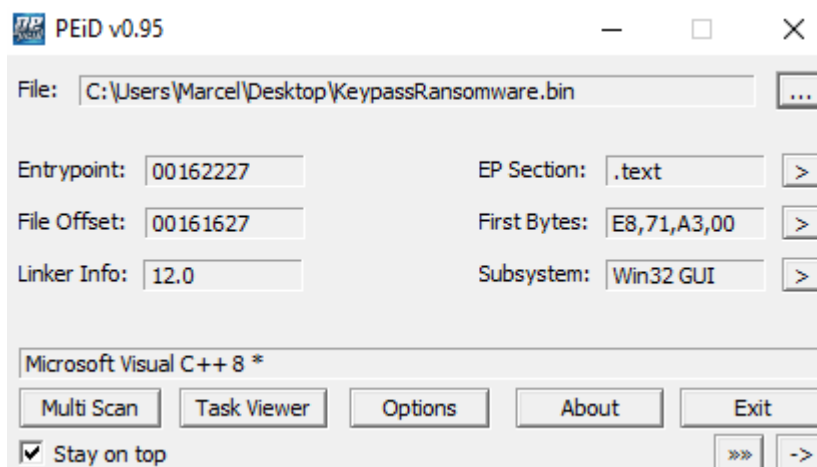
AhnLab-V3	Trojan.Win32.Ransom.R233970	Alibaba	Trojan.Win32/Filecoder.999ee560
ALYac	Trojan.Ransom.Filecoder	Antiy-AVL	Trojan[Ransom]Win32.Encoder
Arcabit	Generic.Ransom.KeyPass.887F95AB	Avast	Win32.Trojan-gen
AVG	Win32.Trojan-gen	Avira (no cloud)	TR/FileCoder.pfzsh
BitDefender	Dropped Generic.Ransom.KeyPass.887...	Bkav Pro	W32.CaiegiD.Trojan
ClamAV	Win.Ransomware.Keypass-6731956-0	CrowdStrike Falcon	Win/malicious_confidence_100% (W)

VirusTotal wykazał, że 59/70 antywirusów zidentyfikowało ten plik jako szkodliwy (malware).

2. Wstępna Analiza Statyczna

a. Pakowanie

Na sam początek analizy sprawdziłem czy program jest spakowany
Program PEiD nie wykrył żadnych programów pakujących



b. Header

Nagłówek zawiera informacje o architekturze i cechach programu

Korzysta z instrukcji Intel 386, które są używane na znacznej większości procesorów (wsteczna kompatybilność)

Skompilowany został 7.08.2018 o godzinie 14:31:21

Characteristics zawiera mało, bo tylko dwie cechy

0x0100 - Program korzysta z instrukcji 32-bitowych (x86)

0x0002 - Program jest wykonywalny

c. Sekcje

Program zawiera dużo sekcji i na podstawie flag można wyciągnąć kilka informacji

- .text - Instrukcje wykonywane przez procesor
- .rdata - Dane tylko do odczytu
- .data - Dane w tej sekcji mogą zostać zmienione przez program
- .rsrc - Dane tylko do odczytu, zawiera zasoby dla modułu
- .reloc - Dane tylko do odczytu, sekcja może zostać odrzucona

d. Importy

Program zawiera bardzo dużo importowanych bibliotek w tym:

- KERNEL32.dll - Funkcje Systemowe
- USER32.dll - Interfejs użytkownika. Program importuje co najmniej **200** funkcji z tej biblioteki. Większość służy do obsługi okien. Najbardziej zaintrygowały mnie te
 - BeginDeferWindowPos - Handle do struktury złożonej z kilku okien
 - GetSystemMetrics - Informacje czy system jest załadowany w trybie bezpiecznym, czy jest właśnie wyłączany. Wartości jest dużo, zależy czego potrzebuje aplikacja
 - Load*W - Ładuje wybrany zasób z sekcji .rsrc
 - OpenClipboard - Otwiera schowek i zapobiega przed zmianami go przez inne aplikacje
- GDI32.dll - Interfejsy GUI
- WINSPOOL.DRV - Sterownik do drukarek (LOL)
- SHELL32.dll - Wykonywanie poleceń w powłocie
- ADVAPI32.dll - API systemu windows (Rejestry, usługi itp.)
- UxTheme.dll - Rysowanie kolorów i tła
- ole32.dll - Biblioteka do interakcji przy użyciu Component Object Model (COM)
- gdiplus.dll - Wyświetlanie zdjęć

- WINMM.dll - Odtwarzanie dźwięków
- MPR.dll - Enumeracja istniejących połączeń sieciowych
- PSAPI.DLL - Enumeracja procesów i modułów
- WS2_32.dll - Funkcje sieciowe i internetowe
- IMM32.dll - Umiędzynarodowienie aplikacji (tłumaczenie na inne języki itp)

Nie są to wszystkie biblioteki importowane, a jedynie te najciekawsze

e. Zasoby

W sekcji .rsrc PPEE wykrywa plik PNG, Ikonkę oraz plik XML

f. Reloc

Jakieś tabele relokacji danych czy coś **TODO**

g. Safe Exception Handling

h. Debug

Program posiada również plik do debugowania, niestety nie jest on załączony z próbką.

Ścieżka to G:\Doc\My work (C++)_New 2018\Encryption\Release\encrypt.pdb
Prawdopodobnie tak nazywał się plik kiedy był pisany przez jego autora.

i. Podejrzone stringi

- Ścieżki do głównych folderów wyszukiwarek (np. C:\Program Files (x86)\Mozilla Firefox\
- Ścieżka do pliku w folderze systemowym - C:\Windows\System32\rdpclip.exe - Służy do połączeń RDP (Remote Desktop Protocol)
- Linki do stron internetowych. Obie strony nie istnieją nawet w wayback machine:
 - <http://kronus.pp.ua/upwinload/get.php>
 - <http://schemas.microsoft.com/SMI/2005/WindowsSettings>”>true</dpiAware></windowsSettings></application></assembly>
- Losowe pliki tekstowe

C:\windows\12300.txtt

C:\windows\12344.txtt

C:\windows\12366.txtt

C:\windows\12388.txtt

C:\windows\123____.txtt

C:\windows\12__3.txtt

- Potencialny Manifest

- | |
|--------------------------------------------------------------------------------------------------------------------------|
| eyLength@\$0BA@\$0BA@\$0CA@\$07\$03\$0A@@@CryptoPP@@@ |
| eyLength@CryptoPP@@@ |
| eyNameTextW |
| eyState |
| eyState |
| eyTip@@@ |
| eyTip@@PAV1@@@ |
| eyToken="6595b64144ccf1df" language="*"></assemblyIdentity></dependentAssembly></dependency><trustInfo xmlns="urn:schem |
| eyTransactedW |
| eyTransactedW |
| eyTransactedW |
| eyW |
| eyW |
| eyW |
| eyboardLayout |
| eyboardPropertyPage |
| eyboardPropertyPage@@@ |
| eyboardShortcut |
| eyboardState |
| eyingInterface@CryptoPP@@@ |
| eyingInterfaceImpl@V?\$TwoBases@VBlockCipher@CryptoPP@@@URijndael_Info@2@@@CryptoPP@@@V12@@@CryptoPP@@@ |
| eyingInterfaceImpl@V?\$TwoBases@VBlockCipher@CryptoPP@@@URijndael_Info@2@@@CryptoPP@@@V12@@@CryptoPP@@@V12@@@CryptoPP@@@ |

- key and it will decrypt all your data.
- key and only we can recover your files.
- key length

3. Podsumowanie Wstępnej Analizy Statycznej

Po wstępnej analizie jestem w stanie stwierdzić, że program jest aplikacją GUI i może składać się z kilku okien na raz. Może on:

- a. wykonywać polecenia w powłoce
- b. Zmieniać rejestry
- c. Korzystać z COM
- d. Zbierać informacje na temat procesów, połączeń internetowych i systemu.
- e. Program importuje również sterowniki do drukarek. Bazując na znajdujących się zdjęciach w sekcji `.rsrc` przypuszczam, że może on drukować je z drukarki lub wyświetlać je w oknach.
- f. Prawdopodobnie korzysta z proxy i łączy się ze stronami internetowymi.
- g. Najbardziej podejrzaną rzeczą są stringi, które wskazują na potencjalne szyfrowanie danych i plików.

Nazwa pliku Debug jest również podejrzana.

Podejrzenia wskazują na **Ransomware**.

4. Zaawansowana Analiza Statyczna

a. Załadowanie pliku do IDA Free

W celu dalszej analizy pliku użyję do tego IDA. Podczas wstępnej analizy zauważyłem, że jest to plik wykonywalny, dlatego załaduję go jako PE. Typ procesora MetaPC (Wszystkie rodzaje) oraz zaznaczam opcje *Load resources* oraz *Manual Load*.

Ważne jest również załadowanie pliku FLIRT `vc32rtf`, który rozpoznaje bardzo dużo funkcji co znacznie ułatwia analizę

b. Analiza sekcji `.data`, `.rdata` i `.rsrc`

Po przejrzaniu sekcji `.rdata` znalazłem String, który prawdopodobnie wyświetla się użytkownikowi zaraz po zaszyfrowaniu jego wszystkich plików

```
```txt
Attention!
All you files, documents, photos, databases and other important files are encrypted and have the
extension: .KEYPASS
The only method of recovering files is to purchase an decrypt software and unique private key.
After purchase you will start decrypt software, enter your unique private key and it will decrypt
all your data
Only we can give you this key and only we can recover your files.
You need to contact us by e-mail
BM-2cUMY51WfNRG8jGrWcMzTASeUGX84yX74l@bitmessage.ch send us your personal
ID and wait for further instructions.
For you to be sure, that we can decrypt your files - you can send us a 1-3 any not very big
encrypted files and we will send you back it in a original form FREE.
Price for decryption $300.
This price available if you contact us first 72 hours.

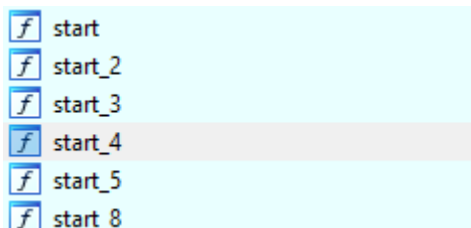
E-mail address to contact us:
BM-2cUMY51WfNRG8jGrWcMzTASeUGX84yX74l@bitmessage.ch

Reserve e-mail address to contact us:
keypassdecrypt@india.com

Your personal id:
```
```

c. Funkcja Main

Po załadowaniu pliku FLIRT IDA rozpoznała 6 funkcji start



```
f start
f start_2
f start_3
f start_4
f start_5
f start_8
```

Wszystkie te funkcje poza *start* wykonują dużo operacji matematycznych np. sinusy

Funkcja *start* wywołuje funkcje takie jak `__heap_init`, `__mtinit`, `__ioint` oraz takie które np. inicjalizują zmienne środowiskowe. Z pewnością mogę stwierdzić jest to miejsce, w którym program rozpoczyna swoje działanie.

Drugim krokiem jest zlokalizowanie funkcji *main*. Powinna ona zwracać wartość 0, dlatego będę analizował program od tyłu w poszukiwaniu rejestru *eax* (rejestr przechowujący pierwszą zwracaną wartość przez funkcję, według AMD ABI) o wartości 0

Dodatkowo znajdują się tu jedynie dwie niezidentyfikowane funkcje:

- *sub_56C639*
- *sub_5C96C1*

Pierwsza funkcja zawiera tylko jedną prostą pętlę i żadnych podfunkcji więc nie może być to funkcja *main*

Po długiej analizie funkcji *main* nie znalazłem ciekawych informacji.

d. Manifest

Na szczęście funkcja *main* to nie jedyne miejsce z którego można zacząć

Poprzez spojrzenie w cross-references do wiadomości o żądaniu okupu

Wykonuje się tam bardzo dużo kodu, który ciężko jest mi zrozumieć. Widziałem tam jedynie informacje o przeglądarek internetowych oraz możliwego zapisywania żądania okupu do pliku

e. Ghidra

W celu dalszej analizy posłużyłem się Ghidrą, która wyposażona jest w dekompiletor. Kod C będzie łatwiejszy do analizy niż assembler. Po załadowaniu wirusa ze standardowymi opcjami Ghidra szybko rozpoznała *entry point*.

Znajdują się tu dwie funkcje. Jedna z nich to *__security_init_cookie*, która nie jest interesująca. Druga funkcja *FUN_00562b0* jest prawdopodobnie funkcją *main*. Wirus jest bardzo dużym programem i nie udało mi się do końca przeanalizować tej funkcji. Niektóre funkcje były wykorzystane w celu zaciemnienia kodu np. *EncodePointer*. Program używa również zabezpieczeń przed podatnościami Buffer Overflow w postaci *SecurityCookie*.

Znalazłem również stringi, dotyczące szyfrowania, IV oraz CFB-Mode. Znajduje się tu też klasa

```

aes2 = CryptoPP::
    CipherModeFinalTemplate_CipherHolder<class_CryptoPP::BlockCipherFinal<0,class_CryptoPP::Rijndael::Enc>,class_CryptoPP::ConcretePolicyHolder<class_CryptoPP::Empty,class_CryptoPP::CFB_EncryptionTemplate<class_CryptoPP::AbstractPolicyHolder<class_CryptoPP::CFB_CipherAbstractPolicy,class_CryptoPP::CFB_ModePolicy>_>,class_CryptoPP::CFB_CipherAbstractPolicy>_>::vftable;

```

Zawarte są tam słowa CFB Encryption oraz Rijndael, czyli oryginalna nazwa algorytmu AES.

Wiem więc, że pliki są szyfrowane za pomocą AESa w trybie CFB.

Nie udało mi się jednak znaleźć funkcji, która generuje klucz.

Nie wiem też ilu bitowy jest ten algorytm oraz z jakiego IV korzysta.

f. Linux

Użyłem Linuxa do ekstrakcji zasobów za pomocą polecenia *wrestool*

```

(kali@kali)-[~/Desktop]
$ wrestool KeypassRansomware.bin
--type='AFX_DIALOG_LAYOUT' --name=102 --language=1033 [offset=0x2b7038 size=2]
--type='AFX_DIALOG_LAYOUT' --name=142 --language=1049 [offset=0x2b6e30 size=2]
--type=3 --name=1 --language=1049 [type=icon offset=0x28c540 size=12452]
--type=3 --name=2 --language=1049 [type=icon offset=0x28f5e8 size=67624]
--type=3 --name=3 --language=1049 [type=icon offset=0x29fe10 size=38056]
--type=3 --name=4 --language=1049 [type=icon offset=0x2a92b8 size=21640]
--type=3 --name=5 --language=1049 [type=icon offset=0x2ae740 size=16936]
--type=3 --name=6 --language=1049 [type=icon offset=0x2b2968 size=9640]
--type=3 --name=7 --language=1049 [type=icon offset=0x2b4f10 size=4264]
--type=3 --name=8 --language=1049 [type=icon offset=0x2b5fb8 size=2440]
--type=3 --name=9 --language=1049 [type=icon offset=0x2b6940 size=1128]
--type=5 --name=102 --language=1033 [type=dialog offset=0x2b6e38 size=512]
--type=5 --name=129 --language=1049 [type=dialog offset=0x28c3b0 size=204]
--type=5 --name=142 --language=1049 [type=dialog offset=0x28c480 size=188]
--type=14 --name=140 --language=1049 [type=group_icon offset=0x2b6da8 size=132]
--type=24 --name=1 --language=1033 [offset=0x2b7040 size=796]

```

Następnie użyłem *binwalk* do identyfikacji typu plików.

Większości nie udało się rozpoznać, ale udało się wydobyć plik PNG oraz XML



Plik PNG przedstawia bitcoina

```
--<assembly manifestVersion="1.0">
--<dependency>
--<dependentAssembly>
--<assemblyIdentity type="win32" name="Microsoft.Windows.Common-Controls" version="6.0.0.0" processorArchitecture="x86" publicKeyToken="6595b64144ccf1df" language="**"/>
--</dependentAssembly>
--</dependency>
--<trustInfo>
--<security>
--<requestedPrivileges>
--<requestedExecutionLevel level="asInvoker" uiAccess="false"/>
--</requestedPrivileges>
--</security>
--</trustInfo>
--<application>
--<windowsSettings>
--<dpiAware>true</dpiAware>
--</windowsSettings>
--</application>
</assembly>
```

Natomiast plik XML wygląda tak.

Binwalk wykrył w programie również sygnatury AESa (S-boxy) oraz stałe algorytmu haszującego SHA-256.

5. Podsumowanie

Po Statycznej Analizie jestem w stanie stwierdzić, że wirus jest zaciemniony i skomplikowany.

Po uruchomieniu wirus zacznie szyfrować pliki użytkownika za pomocą algorytmu AES w trybie CFB oraz do każdego folderu doda plik KEYPASS_INFO!!!.txt, w którym będzie wiadomość z żądaniem okupu. Zasyfrowane pliki mają rozszerzenie .KEYPASS

Co ciekawe podczas analizy zauważyłem rozszerzenia i metadane należące do różnych krajów

Jedna strona internetowa jest z rozszerzeniem .ua - Ukraina

Niektóre metadane znalezione podczas analizy w PPEE wskazywały język rosyjski

Są również dwa maile:

Jeden z rozszerzeniem .ch - Szwajcaria

Drugi india.com - India

Komputer po uruchomieniu wirusa wygląda tak:

