

Reverse Engineering the Behaviour of NotPetya Ransomware

R Lakshmi Prasanna Sai, T. Pavan Kumar

Abstract—Recently Ransomware attack had a great impact on several sectors like, Banking & finance, Insurance, Healthcare, utility and energy, Manufacturing, Education, Public and Government sectors etc. One of the prominent type of ransomware that effected several computers across the world, including Ukraine, France, Russia, and England which hit the big time in 2017, however its effect still persists in 2018, and is referred to as NotPetya. This is destructive because it combines regular ransomware behaviour with stealthy transmission techniques. NotPetya encrypts the files and also master boot loader (MBR) which intercepts the booting process with a ransom note. Eventhough by paying the ransom, the data couldn't have been recovered from the machine. This paper gives comprehensive technical analysis and reverse engineering of NotPetya ransomware.

Keywords—Ransom, Ransomware, NotPetya, Encryption, Reverse Engineering.

I. INTRODUCTION

Ransomware is one of the biggest threats in the Digital world. It is a type of malware that encrypts all the files or documents on the PC and it has the capability to spread across the network. Victim's can only get back to their files only if they pay ransom to the attacker. Data from the statistics shows that Public/Private sector is not immune to attack. Most of the attacks are targeting Financial services, Education, IT/Telecoms, Power grids, Oil and gas, Government etc have been hit as well. All these ransomware attacks are mainly carried by using Trojan that is a malicious code is masked as a legitimate file which comes as an email attachment where the victim is tricked to open it or download it. Around from 2012, ransomware scams are growing internationally.^[3] The victims who confronts with ransomware between 2016 to 2017 increased by 11.4% when compared with 2015-16. The average ransom is up to \$1,000. Adding strength to the effect, about 20% of the victim's who have paid the ransom demands, never retrieved their files back from effect. They disconnected with the network without providing decryption key. About 72% of the infected companies lost their access to data for two to three days which is a great loss to the revenue.^[5] In the first six months of 2018 there have been 181.5 million ransomware attacks^[4]. According to Kaspersky, for every 40 seconds, a company gets shot by a ransomware.^[6]

In the ransomware families one of the devastating type of ransomware is NotPetya which is currently spreading across

the world which stood top second in its effect. According to reports it first originated from Russia and Ukraine, but now reached to U.S, the U.K, Denmark, Poland, Italy, India, Japan, Germany, France. In other words, it's almost everywhere in the world. The "NotPetya" attacks is similar to the very recent WannaCry ransomware which uses NSA exploit EternalBlue for spreading through network. But in addition to this, NotPetya uses multiple propagation techniques to spread through the computers. It includes Credential stealer to grab passwords and PsExec which use those collected usernames and passwords to gain access to other systems that are connected in that domain in the same network.^[7] It is not usual type of ransomware because instead of directly encrypting the victim's files, it encrypts the MFT(Master File Table) which holds the information related to the file names, size and location on the physical drive. Prior encrypting MFT, it replaces MBR(Master Boot Record), which stores the code that initiates the OS bootloader and replaces it with malicious code that displays the ransom note with instructions. So it stops the system from booting and displays the ransom note whenever the system is started.^[8]

So, to analyze the functionality of malware we need to reverse engineer it. Reverse Engineering is a challenging task for the malware analyst. Reverse Engineering involves mainly two important techniques for analysis of malware they are static and dynamic analysis. Static analysis is done without running the the malware, so it is much safer than dynamic analysis. Whereas in dynamic analysis the malware is executed in separate/isolated environment to examine its behaviour^[9]. Most of the literatures are based on static analysis or dynamic analysis. Whereas my work will collectively represents static, dynamic and characteristics of NotPetya malware. This paper will cover in-depth technical analysis of NotPetya, which is structured as follows: Sec. 2 describes how NotPetya spreads. In Sec. 3 Flow of the malware execution in secured environment. In Sec. 4 reports static and dynamic analysis results done with malware. Sec. 5 Summarises the related work. Sec. 6 Concludes.

II. RELATED WORK

NotPetya malware combines ransomware functionality with an ability to propagate itself in network. This is initially identified on the systems running a document management software that is M.E.Doc. This software is mostly used for tax and payroll accounting. Based on analysing the M.E.Doc software, and from reports by anti-virus companies, it was

Revised Manuscript Received on March 08, 2019.

R Lakshmi Prasanna Sai, M.Tech Student, Department of Computer Science and Engineering, Koneru Lakshmaiah Educational Foundation, Vaddeswaram, Guntur District, Andhra Pradesh, India. (laxmiprasannasai@hotmail.com)

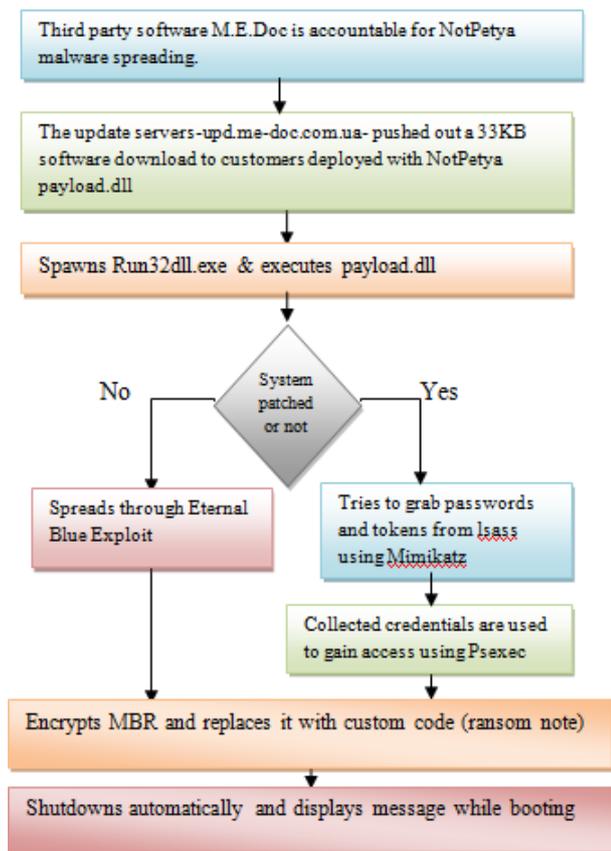
Dr.T. Pavan Kumar, Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Educational Foundation, Vaddeswaram, Guntur District, Andhra Pradesh, India. (pavanakumat_ist@kluniversity.in)



first deployed as a software update. And it started distributing through network slowly. It combines traditional ransomware with propagating through network functionality^[10].

The system infected with NotPetya has three methods of spreading as discussed in the flowchart,

1. Remote exploit (EternalBlue, EternalRomance) for MS17-010.
2. Windows Management Instrumentation(WMI).
3. The psexec tool.



Flow of NotPetya ransomware

It spreads to Windows Operating System through several methods. One of the prominent way is SMB service exploit (EternalBlue) which is previously exploited by WannaCry. It is the same vulnerability reported by Microsoft as MS17-010. It also uses Mimikatz, a technique to collect the credentials from the windows lsass (Local Security Authority Subsystem Service). The collected credentials are used to make an attempt to compromise other systems by using Microsoft tools, PsExec and Windows Management Instrumentation (WMI). Not Petya malware uses MS17-010 vulnerability to infect the unpatched systems. It uses PsExec and WMI tools to exploit the patched systems by extracting credentials from infected system's lsass process to gain access to systems^{[10][11]}.

Then it overwrites the MFT table and replaces the MBR with hostile code which prevents system from booting and displays the ransom demanding note. The encryption algorithms used by this ransomware are 128-bit AES in CBC mode and 2048-bit RSA to encrypt files. The ransom note demands \$300 USD for each infected machine, and established Bitcoin workflow with the email address(

wowsmith123456@posteo.net

wowsmith123456@posteo.net). According to research reports, there are no such evidences of providing decryption keys by the attackers for recovering files after payment.

So to analyze the actual infection that is caused by the malware, Reverse Engineering is preferred. As discussed there are two methods for analysing a malware. They are static and dynamic analysis which are once again divided into two sub parts.

1.1 Static Analysis

1.1.1 Basic Static Analysis

It will help to make sure that the file is malicious or not. It is mainly used to know the functionality of the malware because it is a process of investigating the executable file without viewing the actual code. It is a straightforward process and very quick, but it is mostly ineffective against sophisticated malware.

1.1.2 Advanced Static Analysis

Advanced static analysis, is looking at the program's instructions to know the functionality of malware by loading the PE file into a disassembler. Disassembler will tell exactly what the program does by executing the instructions through CPU. It is a deeper learning process than basic static analysis and requires knowledge to understand the assembly-level code and also windows OS concepts.

1.2 Dynamic Analysis

1.2.1 Basic Dynamic Analysis

It involves running the malware on the system and noticing its behaviour in order to remove the infection. But to run the malware a separate environment must be setted up that will decrease the risk of damage to system and also to network. Like Basic Static analysis, it can be performed without having deep programming knowledge. But through this approach they may miss the important functionality.

1.2.2 Advanced Dynamic Analysis

It involves running the malware using debugger to examine the internal state of the executable. This technique provides an appropriate way to know the behaviour of malware functionality. This technique will be most useful to obtain information that is difficult to gather from other techniques.

III. MALWARE ANALYSIS & RESULTS

- ❖ 027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745-----**Main DLL**
- ✦ 02ef73bd2458627ed7b397ec26ee2de2e92c71a0e7588f78734761d8edbdcd9f-----
(embedded 64-bit credential dumper)
- ✦ eae9771e2eeb7ea3c6059485da39e77b8c0c369232f01334954fbac1c186c998-----
(embedded 32-bit credential dumper)
- ✦ f8dbabdfa03068130c277ce49c60e35c029ff29d9e3c74c362521f3fb02670d5-----



embedded psexec.exe (not harmful).

The above are the hash values of the analyzed samples. First one is the Main dll which contains the code of the EternalBlue and EternalRomance exploit. Second and third is the 32-bit and 64-bit embedded credential dumper similar to Mimikatz. The last one is the Sysinternals PsExec.exe file which is used to gain remote access to other systems for spreading the infection. Further technical analysis is done in the below sections accordingly.

Basic Static Analysis

The sample that is used for basic static analysis is 32-bit DLL with an unnamed export as in Fig 1. It is not packed, as shown in fig 2. As shown in fig 3, the resource section contains four obfuscated binaries. In those binaries, one is PsExec utility, two are 32-bit and 64-bit credential harvesters and the fourth one is a component of exploit (Eternal Blue).

E	Ordinal ^	Hint	Function	Entry Point
0x0	1 (0x0001)	N/A	N/A	0x00007DEB

Figure 1



Figure 2

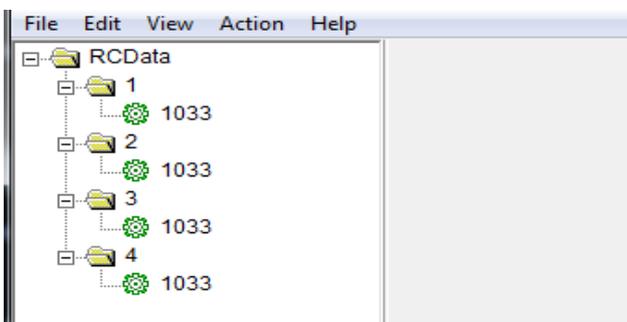


Figure 3

In this work, I have developed a tool named Basic Static analysis Report, which gives the information in the file. It displays the details like MD5, SHA1, PE file entropy, list of sections in the PE file, windows functions that are used by malware. Tool has the capability to show entropy of a given sample. It may detect the type of malware family according to the given yara rules. It also generates results according to the malware behavior as shown in Fig 4.

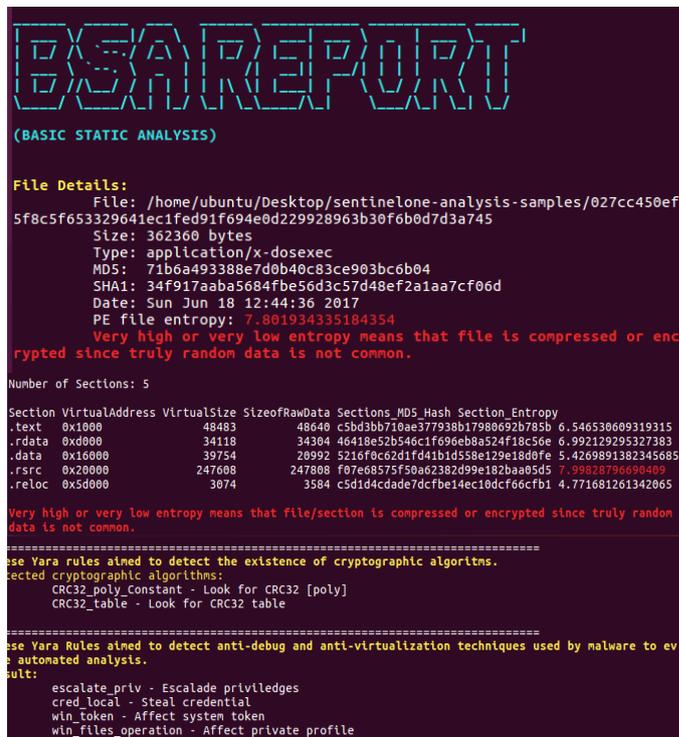


Figure 4

Basic Dynamic Analysis

In this analysis, the sample is executed in a safe or isolated environment. The file that is dropped by the malware is as follows:

- C:\Windows\perfc.dat

Whenever the sample gets installed, it will check whether the main dll is present in “C:\Windows” directory. This technique is commonly used to thwart the analysis efforts.

- C:\Windows\System32\rundll32 perfc.dat, #1

So, through Process Monitor we can check the processes that are created by the malware, shown in Fig 5. A temp file named 3FC0.tmp is created in the %temp% folder which is 32-bit or 64-bit credential harvester. It drops the file C:\Windows\dllhost.dat, a copy of the PsExec, which allows execution of process remotely. And also copies itself in to the memory and free the original one, removing the lock of the file on the disk.

Time	Process Name	PID	Operation	Path	Result	Detail
9:29.4	cmd.exe	3208	Process Create	C:\Windows\system32\rundll32.exe	SUCCESS	PID: 324, Comman...
9:29.4	rundll32.exe	324	Process Create	C:\Windows\system32\cmd.exe	SUCCESS	PID: 2884, Comma...
9:29.4	cmd.exe	380	Process Create	C:\Windows\system32\control.exe	SUCCESS	PID: 2804, Comma...
9:29.4	cmd.exe	2984	Process Create	C:\Windows\system32\uchost.exe	SUCCESS	PID: 1912, Comma...
9:29.4	rundll32.exe	324	Process Create	C:\Users\Deif\AppData\Local\Temp\3FC0.tmp	SUCCESS	PID: 3500, Comma...
9:29.4	cmd.exe	380	Process Create	C:\Windows\system32\control.exe	SUCCESS	PID: 3924, Comma...

Figure 5

As shown in Fig 6, the files that are created by the malware after execution are dllhost.dat and also perfc.

dllhost.dat	C:\Windows	Date modified: 11/13/2018 10:48 AM	Size: 372 KB
perfc	C:\Windows	Date modified: 11/13/2018 10:48 AM	Size: 0 bytes

Figure 6

As shown in Fig 7, the result obtained by regshot helps to view the changes in the registry values after running the malware. It lists the number of modified keys, newly added keys and the total number of changes done in the registry.

```

Regshot 1.9.0 x86 ANSI
Comments:
Datetime: 2018/11/12 09:32:44 , 2018/11/12 09:33:59
Computer: WIN-K3LMKMB0VRM , WIN-K3LMKMB0VRM
Username: Dell , Dell
-----
Keys added: 14
-----
Values added: 54
-----
Total changes: 93
    
```

Figure 7

Advanced Static Analysis

Here, we need to disassemble the code of malware to know its functionality. As shown in the Fig 8, it is the main EternalBlue exploit code i.e., core_MS17_010. If the exploit condition exits, the actual code is called in order to gain remote code execution abilities on victim computers.

```

push [ebp+_lpIPAddress] ; cp
push eax ; _array_ConnectedSockets
call core_MS17_010 ; Check for vulnerable connections
mov edi, eax
lea eax, [ebp+array_ConnectedSockets]
test edi, edi
jz short loc_100066ED

call net_OpenSocketCleanUp
mov eax, edi
jmp short loc_10006721

loc_100066ED: ; int
push [ebp+arg_18]
mov byte_1001F8FD, 0
push [ebp+arg_14] ; int
push [ebp+arg_10] ; int
push [ebp+arg_0] ; int
push [ebp+arg_8] ; int
push [ebp+arg_4] ; int
push offset sub_10001F74 ; int
push esi ; hostshort
push [ebp+_lpIPAddress] ; cp
push eax ; int
call core_MS17_010 ; Exploit vulnerability
    
```

Figure 8

The process of exploitation starts from core_MS17_010 (sub_10005A7E).It sets up connections to vulnerable victims and then calls sub_10003CA0 which is responsible for decrypting and delivering payloads to victims. In exploit, the construction of payload is finished by decrypting and adding two packed resources to the malware's resource section.

Fig 9 represents the flow of decrypting EternalBlue packers packed in malware's resource section.

```

loc_10003D80:
mov cl, ds:byte_100123B0[eax]
xor cl, 0CCh
mov [esi+eax+1F1h], cl
inc eax
cmp eax, 977h
jb short loc_10003D80

loc_10003D98:
push edi
cmp bl, 2
jnz short loc_10003D88

loc_10003D9C:
mov ecx, offset unk_10012D27
mov eax, esi
sub ecx, esi
mov edi, 478h

loc_10003DAC:
mov dl, [ecx+eax]
xor dl, 0CCh
mov [eax], dl
inc eax
dec edi
jnz short loc_10003DAC
    
```

Figure 9

In the Fig 10, we can see how the previously constructed packet is delivered through the open socket.

```

loc_10003DB8: ; len
push [ebp+len]
mov ebx, [ebp+arg_0]
push esi ; buf
call NI_SelectWriteAndSend
lea esi, [ebp+NI_PayloadBuffer]
mov edi, eax
call NI_Free
mov eax, edi
pop edi
    
```

Figure 10

Advanced Dynamic Analysis

In this we use OllyDbg to debug the malware for knowing its internal functionality. For patched systems to spread the malware, a copy of windows sysinternals PsExec tool is written to %WinDir%\dllhost.dat. It uses the tool for gaining access to remote system to run malware on it with the following command.

- *Psexec -accepteula -s -d c:\windows\system32\rundll32.exe "C:\Windows\<filename> ", #1*

```

00401000 75 05 JNZ     00401007 ; if not zero
00401002 50 PUSH  0
00401003 50 PUSH  0
00401004 50 PUSH  0
00401005 50 PUSH  0
00401006 50 PUSH  0
00401007 50 PUSH  0
00401008 50 PUSH  0
00401009 50 PUSH  0
0040100A 50 PUSH  0
0040100B 50 PUSH  0
0040100C 50 PUSH  0
0040100D 50 PUSH  0
0040100E 50 PUSH  0
0040100F 50 PUSH  0
00401010 50 PUSH  0
00401011 50 PUSH  0
00401012 50 PUSH  0
00401013 50 PUSH  0
00401014 50 PUSH  0
00401015 50 PUSH  0
00401016 50 PUSH  0
00401017 50 PUSH  0
00401018 50 PUSH  0
00401019 50 PUSH  0
0040101A 50 PUSH  0
0040101B 50 PUSH  0
0040101C 50 PUSH  0
0040101D 50 PUSH  0
0040101E 50 PUSH  0
0040101F 50 PUSH  0
00401020 50 PUSH  0
00401021 50 PUSH  0
00401022 50 PUSH  0
00401023 50 PUSH  0
00401024 50 PUSH  0
00401025 50 PUSH  0
00401026 50 PUSH  0
00401027 50 PUSH  0
00401028 50 PUSH  0
00401029 50 PUSH  0
0040102A 50 PUSH  0
0040102B 50 PUSH  0
0040102C 50 PUSH  0
0040102D 50 PUSH  0
0040102E 50 PUSH  0
0040102F 50 PUSH  0
00401030 50 PUSH  0
00401031 50 PUSH  0
00401032 50 PUSH  0
00401033 50 PUSH  0
00401034 50 PUSH  0
00401035 50 PUSH  0
00401036 50 PUSH  0
00401037 50 PUSH  0
00401038 50 PUSH  0
00401039 50 PUSH  0
0040103A 50 PUSH  0
0040103B 50 PUSH  0
0040103C 50 PUSH  0
0040103D 50 PUSH  0
0040103E 50 PUSH  0
0040103F 50 PUSH  0
00401040 50 PUSH  0
00401041 50 PUSH  0
00401042 50 PUSH  0
00401043 50 PUSH  0
00401044 50 PUSH  0
00401045 50 PUSH  0
00401046 50 PUSH  0
00401047 50 PUSH  0
00401048 50 PUSH  0
00401049 50 PUSH  0
0040104A 50 PUSH  0
0040104B 50 PUSH  0
0040104C 50 PUSH  0
0040104D 50 PUSH  0
0040104E 50 PUSH  0
0040104F 50 PUSH  0
00401050 50 PUSH  0
00401051 50 PUSH  0
00401052 50 PUSH  0
00401053 50 PUSH  0
00401054 50 PUSH  0
00401055 50 PUSH  0
00401056 50 PUSH  0
00401057 50 PUSH  0
00401058 50 PUSH  0
00401059 50 PUSH  0
0040105A 50 PUSH  0
0040105B 50 PUSH  0
0040105C 50 PUSH  0
0040105D 50 PUSH  0
0040105E 50 PUSH  0
0040105F 50 PUSH  0
00401060 50 PUSH  0
00401061 50 PUSH  0
00401062 50 PUSH  0
00401063 50 PUSH  0
00401064 50 PUSH  0
00401065 50 PUSH  0
00401066 50 PUSH  0
00401067 50 PUSH  0
00401068 50 PUSH  0
00401069 50 PUSH  0
0040106A 50 PUSH  0
0040106B 50 PUSH  0
0040106C 50 PUSH  0
0040106D 50 PUSH  0
0040106E 50 PUSH  0
0040106F 50 PUSH  0
00401070 50 PUSH  0
00401071 50 PUSH  0
00401072 50 PUSH  0
00401073 50 PUSH  0
00401074 50 PUSH  0
00401075 50 PUSH  0
00401076 50 PUSH  0
00401077 50 PUSH  0
00401078 50 PUSH  0
00401079 50 PUSH  0
0040107A 50 PUSH  0
0040107B 50 PUSH  0
0040107C 50 PUSH  0
0040107D 50 PUSH  0
0040107E 50 PUSH  0
0040107F 50 PUSH  0
00401080 50 PUSH  0
00401081 50 PUSH  0
00401082 50 PUSH  0
00401083 50 PUSH  0
00401084 50 PUSH  0
00401085 50 PUSH  0
00401086 50 PUSH  0
00401087 50 PUSH  0
00401088 50 PUSH  0
00401089 50 PUSH  0
0040108A 50 PUSH  0
0040108B 50 PUSH  0
0040108C 50 PUSH  0
0040108D 50 PUSH  0
0040108E 50 PUSH  0
0040108F 50 PUSH  0
00401090 50 PUSH  0
00401091 50 PUSH  0
00401092 50 PUSH  0
00401093 50 PUSH  0
00401094 50 PUSH  0
00401095 50 PUSH  0
00401096 50 PUSH  0
00401097 50 PUSH  0
00401098 50 PUSH  0
00401099 50 PUSH  0
0040109A 50 PUSH  0
0040109B 50 PUSH  0
0040109C 50 PUSH  0
0040109D 50 PUSH  0
0040109E 50 PUSH  0
0040109F 50 PUSH  0
004010A0 50 PUSH  0
004010A1 50 PUSH  0
004010A2 50 PUSH  0
004010A3 50 PUSH  0
004010A4 50 PUSH  0
004010A5 50 PUSH  0
004010A6 50 PUSH  0
004010A7 50 PUSH  0
004010A8 50 PUSH  0
004010A9 50 PUSH  0
004010AA 50 PUSH  0
004010AB 50 PUSH  0
004010AC 50 PUSH  0
004010AD 50 PUSH  0
004010AE 50 PUSH  0
004010AF 50 PUSH  0
004010B0 50 PUSH  0
004010B1 50 PUSH  0
004010B2 50 PUSH  0
004010B3 50 PUSH  0
004010B4 50 PUSH  0
004010B5 50 PUSH  0
004010B6 50 PUSH  0
004010B7 50 PUSH  0
004010B8 50 PUSH  0
004010B9 50 PUSH  0
004010BA 50 PUSH  0
004010BB 50 PUSH  0
004010BC 50 PUSH  0
004010BD 50 PUSH  0
004010BE 50 PUSH  0
004010BF 50 PUSH  0
004010C0 50 PUSH  0
004010C1 50 PUSH  0
004010C2 50 PUSH  0
004010C3 50 PUSH  0
004010C4 50 PUSH  0
004010C5 50 PUSH  0
004010C6 50 PUSH  0
004010C7 50 PUSH  0
004010C8 50 PUSH  0
004010C9 50 PUSH  0
004010CA 50 PUSH  0
004010CB 50 PUSH  0
004010CC 50 PUSH  0
004010CD 50 PUSH  0
004010CE 50 PUSH  0
004010CF 50 PUSH  0
004010D0 50 PUSH  0
004010D1 50 PUSH  0
004010D2 50 PUSH  0
004010D3 50 PUSH  0
004010D4 50 PUSH  0
004010D5 50 PUSH  0
004010D6 50 PUSH  0
004010D7 50 PUSH  0
004010D8 50 PUSH  0
004010D9 50 PUSH  0
004010DA 50 PUSH  0
004010DB 50 PUSH  0
004010DC 50 PUSH  0
004010DD 50 PUSH  0
004010DE 50 PUSH  0
004010DF 50 PUSH  0
004010E0 50 PUSH  0
004010E1 50 PUSH  0
004010E2 50 PUSH  0
004010E3 50 PUSH  0
004010E4 50 PUSH  0
004010E5 50 PUSH  0
004010E6 50 PUSH  0
004010E7 50 PUSH  0
004010E8 50 PUSH  0
004010E9 50 PUSH  0
004010EA 50 PUSH  0
004010EB 50 PUSH  0
004010EC 50 PUSH  0
004010ED 50 PUSH  0
004010EE 50 PUSH  0
004010EF 50 PUSH  0
004010F0 50 PUSH  0
004010F1 50 PUSH  0
004010F2 50 PUSH  0
004010F3 50 PUSH  0
004010F4 50 PUSH  0
004010F5 50 PUSH  0
004010F6 50 PUSH  0
004010F7 50 PUSH  0
004010F8 50 PUSH  0
004010F9 50 PUSH  0
004010FA 50 PUSH  0
004010FB 50 PUSH  0
004010FC 50 PUSH  0
004010FD 50 PUSH  0
004010FE 50 PUSH  0
004010FF 50 PUSH  0
00401100 50 PUSH  0
00401101 50 PUSH  0
00401102 50 PUSH  0
00401103 50 PUSH  0
00401104 50 PUSH  0
00401105 50 PUSH  0
00401106 50 PUSH  0
00401107 50 PUSH  0
00401108 50 PUSH  0
00401109 50 PUSH  0
0040110A 50 PUSH  0
0040110B 50 PUSH  0
0040110C 50 PUSH  0
0040110D 50 PUSH  0
0040110E 50 PUSH  0
0040110F 50 PUSH  0
00401110 50 PUSH  0
00401111 50 PUSH  0
00401112 50 PUSH  0
00401113 50 PUSH  0
00401114 50 PUSH  0
00401115 50 PUSH  0
00401116 50 PUSH  0
00401117 50 PUSH  0
00401118 50 PUSH  0
00401119 50 PUSH  0
0040111A 50 PUSH  0
0040111B 50 PUSH  0
0040111C 50 PUSH  0
0040111D 50 PUSH  0
0040111E 50 PUSH  0
0040111F 50 PUSH  0
00401120 50 PUSH  0
00401121 50 PUSH  0
00401122 50 PUSH  0
00401123 50 PUSH  0
00401124 50 PUSH  0
00401125 50 PUSH  0
00401126 50 PUSH  0
00401127 50 PUSH  0
00401128 50 PUSH  0
00401129 50 PUSH  0
0040112A 50 PUSH  0
0040112B 50 PUSH  0
0040112C 50 PUSH  0
0040112D 50 PUSH  0
0040112E 50 PUSH  0
0040112F 50 PUSH  0
00401130 50 PUSH  0
00401131 50 PUSH  0
00401132 50 PUSH  0
00401133 50 PUSH  0
00401134 50 PUSH  0
00401135 50 PUSH  0
00401136 50 PUSH  0
00401137 50 PUSH  0
00401138 50 PUSH  0
00401139 50 PUSH  0
0040113A 50 PUSH  0
0040113B 50 PUSH  0
0040113C 50 PUSH  0
0040113D 50 PUSH  0
0040113E 50 PUSH  0
0040113F 50 PUSH  0
00401140 50 PUSH  0
00401141 50 PUSH  0
00401142 50 PUSH  0
00401143 50 PUSH  0
00401144 50 PUSH  0
00401145 50 PUSH  0
00401146 50 PUSH  0
00401147 50 PUSH  0
00401148 50 PUSH  0
00401149 50 PUSH  0
0040114A 50 PUSH  0
0040114B 50 PUSH  0
0040114C 50 PUSH  0
0040114D 50 PUSH  0
0040114E 50 PUSH  0
0040114F 50 PUSH  0
00401150 50 PUSH  0
00401151 50 PUSH  0
00401152 50 PUSH  0
00401153 50 PUSH  0
00401154 50 PUSH  0
00401155 50 PUSH  0
00401156 50 PUSH  0
00401157 50 PUSH  0
00401158 50 PUSH  0
00401159 50 PUSH  0
0040115A 50 PUSH  0
0040115B 50 PUSH  0
0040115C 50 PUSH  0
0040115D 50 PUSH  0
0040115E 50 PUSH  0
0040115F 50 PUSH  0
00401160 50 PUSH  0
00401161 50 PUSH  0
00401162 50 PUSH  0
00401163 50 PUSH  0
00401164 50 PUSH  0
00401165 50 PUSH  0
00401166 50 PUSH  0
00401167 50 PUSH  0
00401168 50 PUSH  0
00401169 50 PUSH  0
0040116A 50 PUSH  0
0040116B 50 PUSH  0
0040116C 50 PUSH  0
0040116D 50 PUSH  0
0040116E 50 PUSH  0
0040116F 50 PUSH  0
00401170 50 PUSH  0
00401171 50 PUSH  0
00401172 50 PUSH  0
00401173 50 PUSH  0
00401174 50 PUSH  0
00401175 50 PUSH  0
00401176 50 PUSH  0
00401177 50 PUSH  0
00401178 50 PUSH  0
00401179 50 PUSH  0
0040117A 50 PUSH  0
0040117B 50 PUSH  0
0040117C 50 PUSH  0
0040117D 50 PUSH  0
0040117E 50 PUSH  0
0040117F 50 PUSH  0
00401180 50 PUSH  0
00401181 50 PUSH  0
00401182 50 PUSH  0
00401183 50 PUSH  0
00401184 50 PUSH  0
00401185 50 PUSH  0
00401186 50 PUSH  0
00401187 50 PUSH  0
00401188 50 PUSH  0
00401189 50 PUSH  0
0040118A 50 PUSH  0
0040118B 50 PUSH  0
0040118C 50 PUSH  0
0040118D 50 PUSH  0
0040118E 50 PUSH  0
0040118F 50 PUSH  0
00401190 50 PUSH  0
00401191 50 PUSH  0
00401192 50 PUSH  0
00401193 50 PUSH  0
00401194 50 PUSH  0
00401195 50 PUSH  0
00401196 50 PUSH  0
00401197 50 PUSH  0
00401198 50 PUSH  0
00401199 50 PUSH  0
0040119A 50 PUSH  0
0040119B 50 PUSH  0
0040119C 50 PUSH  0
0040119D 50 PUSH  0
0040119E 50 PUSH  0
0040119F 50 PUSH  0
004011A0 50 PUSH  0
004011A1 50 PUSH  0
004011A2 50 PUSH  0
004011A3 50 PUSH  0
004011A4 50 PUSH  0
004011A5 50 PUSH  0
004011A6 50 PUSH  0
004011A7 50 PUSH  0
004011A8 50 PUSH  0
004011A9 50 PUSH  0
004011AA 50 PUSH  0
004011AB 50 PUSH  0
004011AC 50 PUSH  0
004011AD 50 PUSH  0
004011AE 50 PUSH  0
004011AF 50 PUSH  0
004011B0 50 PUSH  0
004011B1 50 PUSH  0
004011B2 50 PUSH  0
004011B3 50 PUSH  0
004011B4 50 PUSH  0
004011B5 50 PUSH  0
004011B6 50 PUSH  0
004011B7 50 PUSH  0
004011B8 50 PUSH  0
004011B9 50 PUSH  0
004011BA 50 PUSH  0
004011BB 50 PUSH  0
004011BC 50 PUSH  0
004011BD 50 PUSH  0
004011BE 50 PUSH  0
004011BF 50 PUSH  0
004011C0 50 PUSH  0
004011C1 50 PUSH  0
004011C2 50 PUSH  0
004011C3 50 PUSH  0
004011C4 50 PUSH  0
004011C5 50 PUSH  0
004011C6 50 PUSH  0
004011C7 50 PUSH  0
004011C8 50 PUSH  0
004011C9 50 PUSH  0
004011CA 50 PUSH  0
004011CB 50 PUSH  0
004011CC 50 PUSH  0
004011CD 50 PUSH  0
004011CE 50 PUSH  0
004011CF 50 PUSH  0
004011D0 50 PUSH  0
004011D1 50 PUSH  0
004011D2 50 PUSH  0
004011D3 50 PUSH  0
004011D4 50 PUSH  0
004011D5 50 PUSH  0
004011D6 50 PUSH  0
004011D7 50 PUSH  0
004011D8 50 PUSH  0
004011D9 50 PUSH  0
004011DA 50 PUSH  0
004011DB 50 PUSH  0
004011DC 50 PUSH  0
004011DD 50 PUSH  0
004011DE 50 PUSH  0
004011DF 50 PUSH  0
004011E0 50 PUSH  0
004011E1 50 PUSH  0
004011E2 50 PUSH  0
004011E3 50 PUSH  0
004011E4 50 PUSH  0
004011E5 50 PUSH  0
004011E6 50 PUSH  0
004011E7 50 PUSH  0
004011E8 50 PUSH  0
004011E9 50 PUSH  0
004011EA 50 PUSH  0
004011EB 50 PUSH  0
004011EC 50 PUSH  0
004011ED 50 PUSH  0
004011EE 50 PUSH  0
004011EF 50 PUSH  0
004011F0 50 PUSH  0
004011F1 50 PUSH  0
004011F2 50 PUSH  0
004011F3 50 PUSH  0
004011F4 50 PUSH  0
004011F5 50 PUSH  0
004011F6 50 PUSH  0
004011F7 50 PUSH  0
004011F8 50 PUSH  0
004011F9 50 PUSH  0
004011FA 50 PUSH  0
004011FB 50 PUSH  0
004011FC 50 PUSH  0
004011FD 50 PUSH  0
004011FE 50 PUSH  0
004011FF 50 PUSH  0
00401200 50 PUSH  0
00401201 50 PUSH  0
00401202 50 PUSH  0
00401203 50 PUSH  0
00401204 50 PUSH  0
00401205 50 PUSH  0
00401206 50 PUSH  0
00401207 50 PUSH  0
00401208 50 PUSH  0
00401209 50 PUSH  0
0040120A 50 PUSH  0
0040120B 50 PUSH  0
0040120C 50 PUSH  0
0040120D 50 PUSH  0
0040120E 50 PUSH  0
0040120F 50 PUSH  0
00401210 50 PUSH  0
00401211 50 PUSH  0
00401212 50 PUSH  0
00401213 50 PUSH  0
00401214 50 PUSH  0
00401215 50 PUSH  0
00401216 50 PUSH  0
00401217 50 PUSH  0
00401218 50 PUSH  0
00401219 50 PUSH  0
0040121A 50 PUSH  0
0040121B 50 PUSH  0
0040121C 50 PUSH  0
0040121D 50 PUSH  0
0040121E 50 PUSH  0
0040121F 50 PUSH  0
00401220 50 PUSH  0
00401221 50 PUSH  0
00401222 50 PUSH  0
00401223 50 PUSH  0
00401224 50 PUSH  0
00401225 50 PUSH  0
00401226 50 PUSH  0
00401227 50 PUSH  0
00401228 50 PUSH  0
00401229 50 PUSH  0
0040122A 50 PUSH  0
0040122B 50 PUSH  0
0040122C 50 PUSH  0
0040122D 50 PUSH  0
0040122E 50 PUSH  0
0040122F 50 PUSH  0
00401230 50 PUSH  0
00401231 50 PUSH  0
00401232 50 PUSH  0
00401233 50 PUSH  0
00401234 50 PUSH  0
00401235 50 PUSH  0
00401236 50 PUSH  0
00401237 50 PUSH  0
00401238 50 PUSH  0
00401239 50 PUSH  0
0040123A 50 PUSH  0
0040123B 50 PUSH  0
0040123C 50 PUSH  0
0040123D 50 PUSH  0
0040123E 50 PUSH  0
0040123F 50 PUSH  0
00401240 50 PUSH  0
00401241 50 PUSH  0
00401242 50 PUSH  0
00401243 50 PUSH  0
00401244 50 PUSH  0
00401245 50 PUSH  0
00401246 50 PUSH  0
00401247 50 PUSH  0
00401248 50 PUSH  0
00401249 50 PUSH  0
0040124A 50 PUSH  0
0040124B 50 PUSH  0
0040124C 50 PUSH  0
0040124D 50 PUSH  0
0040124E 50 PUSH  0
0040124F 50 PUSH  0
00401250 50 PUSH  0
00401251 50 PUSH  0
00401252 50 PUSH  0
00401253 50 PUSH  0
00401254 50 PUSH  0
00401255 50 PUSH  0
00401256 50 PUSH  0
00401257 50 PUSH  0
00401258 50 PUSH  0
00401259 50 PUSH  0
0040125A 50 PUSH  0
0040125B 50 PUSH  0
0040125C 50 PUSH  0
0040125D 50 PUSH  0
0040125E 50 PUSH  0
0040125F 50 PUSH  0
00401260 50 PUSH  0
00401261 50 PUSH  0
00401262 50 PUSH  0
00401263 50 PUSH  0
00401264 50 PUSH  0
00401265 50 PUSH  0
00401266 50 PUSH  0
00401267 50 PUSH  0
00401268 50 PUSH  0
00401269 50 PUSH  0
0040126A 50 PUSH  0
0040126B 50 PUSH  0
0040126C 50 PUSH  0
0040126D 50 PUSH  0
0040126E 50 PUSH  0
0040126F 50 PUSH  0
00401270 50 PUSH  0
00401271 50 PUSH  0
00401272 50 PUSH  0
00401273 50 PUSH  0
00401274 50 PUSH  0
00401275 50 PUSH  0
00401276 50 PUSH  0
00401277 50 PUSH  0
00401278 50 PUSH  0
00401279 50 PUSH  0
0040127A 50 PUSH  0
0040127B 50 PUSH  0
0040127C 50 PUSH  0
0040127D 50 PUSH  0
0040127E 50 PUSH  0
0040127F 50 PUSH  0
00401280 50 PUSH  0
00401281 50 PUSH  0
00401282 50 PUSH  0
00401283 50 PUSH  0
00401284 50 PUSH  0
00401285 50 PUSH  0
00401286 50 PUSH  0
00401287 50 PUSH  0
00401288 50 PUSH  0
00401289 50 PUSH  0
0040128A 50 PUSH  0
0040128B 50 PUSH  0
0040128C 50 PUSH  0
0040128D 50 PUSH  0
0040128E 50 PUSH  0
0040128F 50 PUSH  0
00401290 50 PUSH  0
00401291 50 PUSH  0
00401292 50 PUSH  0
00401293 50 PUSH  0
00401294 50 PUSH  0
00401295 50 PUSH  0
00401296 50 PUSH  0
00401297 50 PUSH  0
00401298 50 PUSH  0
00401299 50 PUSH  0
0040129A 50 PUSH  0
0040129B 50 PUSH  0
0040129C 50 PUSH  0
0040129D 50 PUSH  0
0040129
```

- C:\windows\system32\wbem\wmic.exe /node:"<node>" /user:"<user>" /password:"<password>" process call create "C:\Windows\System32\rundll32.exe "C:\Windows\<file>", #1

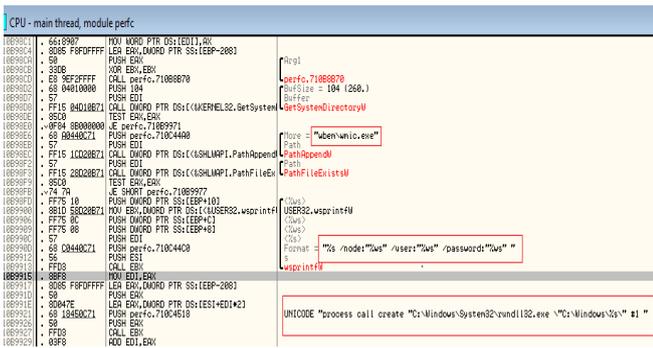


Figure 12

NotPetya engages the following method to reboot the system so that MFT encryptor code loads in the boot loader and displays the ransom note.

It schedules shutdown through cmd with the following command as shown in Fig 13.

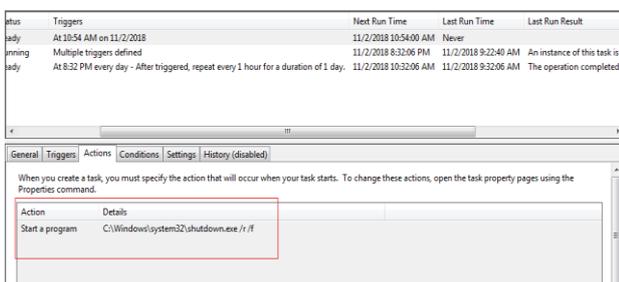
- /c schtasks /Create/SC once /TN "" /TR "C:\Windows\system32\shutdown.exe /r /f" /ST <HH:MM>

/r → reboot after shutdown

/f → forces running applications to close



Figure 13



Scheduled shutdown in system

At last, after encrypting MBR and replacing MFT, it restarts at a particular time scheduled by malware and displays the message shown in the Fig 14.

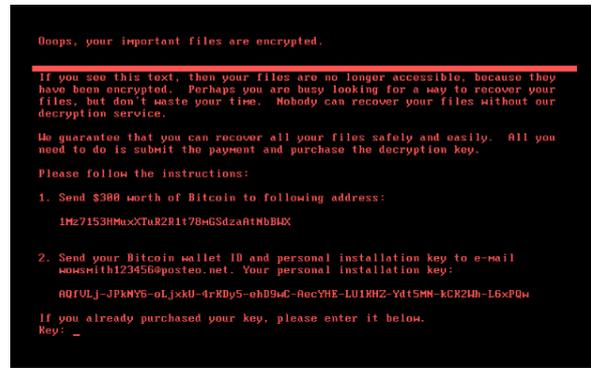


Figure 14

IV. CONCLUSION

By analyzing the NotPetya Ransomware, we have found many important factors like encryption mechanism used by the malware, the types of files it will effect, the network connection that is established for transmitting payload to exploit vulnerabilities in the network system, the registry changes performed by the malware and the new process created by it.

To safeguard from this type of ransomware we always have to maintain a backup copy of the data. Place the firewall for filtering malicious traffic and make sure that the installed softwares are up to date by patching it regularly.

Further analysis is carried on different kinds of ransomware and the signature needs to be added to the antivirus database so that they can be blocked whenever they are detected in any of the system.

V. ACKNOWLEDGMENT

This work is supported by the Department of Science and Technology, India through the fund sanctioned for improvement of Science & Technology infrastructure, at department of CSE, K.L University, by order number SR/FST/ESI-332/2013.

REFERENCES

1. DAN DAHLBERG "ransomware cyber attacks " blog on Bitsight
2. Online "Ransomware" wikipedia
3. Online "New Internet Scam" news on FBI 2012
4. "sonicwall cyber threat report" article on helpnet security 2018
5. Phillip Long "5 Ransomware Statistics Every Business Owner Needs to Know" blog on BIS
6. "Attacks on Business Now Equal One Every 40 Seconds" press release on kaspersky lab 2016
7. Online "Petya" wikipedia
8. Lucian Constantin "Petya ransomware is now double the trouble" article on network world
9. Syarif Yusirwan S, Yudi Prayudi, Imam Riadi "Implementation of Malware Analysis using Static and Dynamic Analysis Method" International Journal of Computer Applications (0975 – 8887)Volume 117 – No. 6, 2015
10. Falcon Intelligence Team "fast spreading petwrap ransomware attack combines eternalblue exploit credential stealing" blog on CrowdStrike "malware analysis basics static analysis" InfoSec Resources

