PAPER • OPEN ACCESS

Malware Analysis and Detection Using Reverse **Engineering Technique**

To cite this article: S Megira et al 2018 J. Phys.: Conf. Ser. 1140 012042

View the article online for updates and enhancements.

You may also like

- A stacking-based classification approach to android malware using host-level encrypted traffic Zhixing Xue, Weina Niu, Xixuan Ren et al.
- An Analysis of Machine Learning-Based Android Malware Detection Approaches R. Srinivasan, S Karpagam, M. Kavitha et al.
- PAM Clustering Aided Android Malicious Apps Detection Nibras Talib Mohammed, Mohsin Hasan Hussein and Abbas Jabber Rashid





DISCOVER how sustainability intersects with electrochemistry & solid state science research



This content was downloaded from IP address 18.226.150.175 on 24/04/2024 at 09:58

Malware Analysis and Detection Using Reverse Engineering Technique

S Megira¹, A R Pangesti² and F W Wibowo³

^{1, 2, 3} Magister Teknik Informatika Universitas AMIKOM Yogyakarta, Indonesia.

IOP Publishing

E-mail: selvy.megira@students.amikom.ac.id

Abstract. The increasing use of internet and technology today cannot be separated from cybercrime that can threaten its users. The cyber threat like malware attempts to infiltrate the computer or mobile device offline or the internet, chat (online) and anyone can be a potential target. Malware, also known as malicious software, is often used by cybercriminals to achieve their goals by tracking internet activity, capturing sensitive information or block computer access. In the past two years, the more malicious software has been created than in the previous ten years combined. Malware has its own defense system and it is possible to hide from antivirus or even infect the antivirus itself. Malware can be handled by knowing how to work when doing an attack into a computer system. This research aims to analyze malware by using malware sample to better understanding how they can infect computers and devices, the level of threats they pose, and how to protect devices against them.

1. Introduction

As time moves forward, everything around us changes. We are almost entirely dependent on technology, and it becomes an important role in modern life. From photo memories to important documents, we store everything on our computers and mobile devices. But, although technology has made our lives convenient, it has also allowed a new form of crime, cyber threat. Even the business processes in the enterprise need the third parties in facing the security threat. The data security services must be specified [1]. Cybercriminals can attack computer by using malware to track internet activities and capturing sensitive information such as username and password from financial websites.

Malware, or malicious software, is any program or file that intentionally designed to harm, infiltrate, or damage a computer, server or computer network. Malware is also commonly defined as malicious code. This software can disable or disrupt the operation of a system, allowing hackers to gain access to confidential and sensitive information and to spy on the computer and the owner of the computer itself. Malware is specifically made to be hidden so that they can remain inside a system for a certain period of time without the knowledge of the system owner. Usually, they disguise themselves into a clean program; even some of the latest

IOP Publishing

malware has the ability to avoid detection of antivirus. The effect of malicious software is much more dangerous for corporates than for personal users. If malware attacks system's network, they can cause widespread damage and disruption, requiring extensive recovery efforts within the organization.

Malware analysis by using reverse engineering method become one solution that can be used to extract data in a malware to find out how the malware is working when it attacks into the system. Therefore, this study aims to perform malware analysis so as to know the dangers of malware and how to prevent it and protect our devices against it. In this study, a file named best.exe will be used as a malware sample to find out information about malware contained in it.

2. Related Works

Analyzed malware sample using a static and dynamic method has been researched [2]. The results of the use method can provide complete information on TT.exe malware, but it takes a long process of time. The first thing to do is to identify a suspected malware program, then detect techniques packaged/disguised by malware, and find the malware. The method used to provide information that can actually be used malware, network connections made by malware against servers, and malware processes in the system. This malware can shut down windows systems like antivirus, firewall, and recovery system. Then the study to analyzed malware and proves that using Reverse Engineering technique has a high success rate in analyzing malware. However, it causes high complexity [3]. Malware can be handled if knowledge of the malware movement when attacking a system is revealed.

Malware is malicious software that is currently increasing at an alarming rate [4]. Meanwhile, another research discussed developments in the detection of android malware, which uses static analysis techniques, dynamic analysis, and hybrid. The main problem in evaluating approaches to malware attacks is due to the absence of more complete malware datasets [5]. Another research combines two methods of static analysis and dynamic analysis that can be used as an indicator to analyze Trojan based on its behavior. Therefore, how the behavior of Trojan when infected windows operating system can be depicted [6].

3. Malware Analysis

Malware analysis is done to provide the necessary information to deal with malware attacks by knowing what's going on in the system, the location of the infected file, detecting how the malware works, and which types of malware it belongs to. Malware can be categorized into several types, and to perform malware analysis [4,7,8]. Precise technique and method are required so the purpose of analysis can be achieved.

3.1 Types of Malware

The following are some common types of malware:

1. Adware

An adware is a type of malware that automatically delivers advertisements and displays ads on your computer. Adware is the most lucrative malware and the least dangerous.

2. Spyware

A spyware is designed to monitor, tracking internet activities, and other activities. Spyware just like adware often sends activities to advertisers. Spyware can violate privacy and has the potential to be abused this becomes controversial.

3. Virus

A virus is malicious software that cybercriminals program to reproduce. Viruses can be used to create botnets, steal information, steal money, harm host computers and network and more. The virus most often is spread by files between computers or sharing software.

4. Worm

A worm like a virus, a program that replicates itself and worms, are infectious. Worm destroys files on the computer and data. They spread over computer networks by exploiting operating system vulnerabilities. Type of computer virus can be classified as computer worms, and worms often spread by sending mass emails with infected attachments to users contact.

5. Trojan

A trojan is a type of malware that trick users into downloading and installing malware by disguise. Trojan is the most dangerous malware. Trojans are used for discovering your financial information, steal data (logins, electronic money), modify files, taking over your computer's system resources, and more.

6. Rootkit

A rootkit is a type of malicious software that gives an unauthorized user privileged access to a computer and designed to remotely access a computer without being detected by security programs or users. It is the hardest of all malware to detect and therefore to remove.

7. Backdoors

A backdoor provides a network connection for other malware to enter or for viruses or spam to be sent or hackers.

8. Keyloggers

The keylogger records everything entry made on your computer without the permission of the user in order to glean your login passwords, names, and other sensitive information.

9. Ransomware

Ransomware is the software finds all your files and encrypts them and then leaves messages for you. To regain access to data, then we have to pay a ransom. Ransomware often spreads like a normal computer worm through some other vulnerability in a

network service or downloaded file. Ransomware encrypts data on the computer and used an encryption key that only attackers know. If the ransom is not paid, often the data is permanently deleted.

3.2 Malware Analysis Method

When performing malware analysis, the malware sample used is an executable file format, which won't be human-readable. Therefore, some methods are used to extract the file so that it can get information from malware. There are two main malware analysis approaches, namely static analysis and dynamic analysis; both methods are subsequently categorized as basic or advanced [9] as seen in Figure 1.

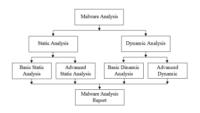


Figure 1. Malware analysis method

From Figure 1, we can describe the methods as follow:

1. Basic Static Analysis

Programs that are suspected of being malware will be tested with scanning using antivirus, then hashing and detection packed or obfuscated in the program. To detect the packer program used so that we can unload the malware file is PEiD. The portable executable structure of the program will be analyzed.

2. Advanced Static Analysis

Advanced static analysis stage includes disassembly or debugging to analyze strings, libraries, and functions linked by using IDA disassembler.

3. Basic Dynamic Analysis

The basic method in dynamic analysis, in observing the work of a system or behavior of malware, a virtual machine is used. So, if the executed malware is damaging the system then the main system is not damaged due to malware that is running.

- 4. Advanced Dynamic Analysis In advanced dynamic analysis methods, an analysis of the Windows operating system and registrialware analysis monitoring process and data analysis packages created by malware.
- 5. Malware Analysis Report

Reports of malware analysis results obtained are based on static analysis and dynamic analysis. Therefore, the report information about the characteristics of malware is obtained.

3.3 Reverse Engineering Technique

According to [10], the reverse engineering process in software or application can be implemented with the steps:

1. Assembly

Assembly language is used for microprocessors and other programmable devices which any low-level programming language. An assembly language is not just a single language but rather a group of languages and the most basic programming language available for any processor. Assembly language cannot recognize high-level languages like Java and Pascal.

2. Disassembly

Disassembly is used for transforming assembly language into machine code. Disassembly is a reverse assembly process [2].

3. Debugging

Debugging is a method which developers can implement to search bugs, subtract bugs, and isolate the source of the problem. Debugging is used for executing testing from each core process in malware [11].

4. X86 Architecture

The X86 architecture is a design of complex instruction set computer with varying instruction length. Basically, on the internal; of most modern computer architectures including x86 follow the Von Neumann architecture. In the design of the reconfigurable system, the interoperability could also be an issue in the architecture [12].

5. Instruction

Instruction is construction from assembly program. An assembly of x86 instruction consists of mnemonic and zero or operands.

6. Hashing

The hash process is executed for verification before and after the malware analysis process. Verification is executed to determine the absence of hash changes in the sample malware after the analysis process.

7. String analysis

Strings in a program are values that will be loaded from malware sample when executed. Reverse engineering process must be done for string analysis to get strong evidence from malware sample.

4. Conclusion

In this research, we use best.exe as a malware sample and perform some experiments by executing it to observe how the malware works. To prevent other systems from getting infected by malware sample, the virtual machine was used for the analysis. On the laptops, VMWare Workstation was installed with Windows 7 as guest OS. With VMWare Workstation, the system can go back to a non-infected system without reinstalling the guest OS using snapshot.

The following tools were used for help malware analysis which is categorized by malware analysis methods. Static method tools can be seen in Table I, and dynamic tools in Table II.

	Basic Static Analysis
Tools	Description
	is used for analyzing
CFF	portable .exe file without
explo	affecting the internal
rer	structure
	is used for displaying the
PEvie	structure and content of
W	the .exe file
Virus	is a website for checking
total.	malware against a
com	program
Adva	nced Static Analysis
Tools	Description
	The Interactive
	Disassembler (IDA) is
IDA	used for disassembling
	binary code
Depe	is used for analyzing the
ndenc	dependencies used by the
У	malware sample
Walk	marware sample
er	

Table 2. Dynamic Tools

Basic Dynamic Analysis		
Tools	Description	
VM Ware Work statio n	is used as a virtual machine to run the malware sample	
Proce	is used for monitoring and	
SS	displaying all activities	

	Basic Dynamic Analysis
Tools	Description
Moni	within the system in real-
tor	time
Proce	is used for monitoring the
SS	processes that are
Explo	currently running in a
er	system path
Wires	is used for capturing and
nark	analyzing network traffic
Advanc	ed Dynamic Analysis
Tools	Description
Olly	is used for debugging
Dbg	binary code

There are many tools that can be used to perform malware analysis. We can choose these tools according to our needs. Some of these tools can be downloaded for free through their official website, and some of them require us to pay the full version of the software.

4.1 Static Analysis

After choosing the tools, malware analysis began with static analysis by using static analysis tools to get the information that can be retrieved by looking at the .exe file's PE header information. Information that will be obtained such as whether the file is really malware or not, what kind of malware it is, what programming language it contains. For the first step, we use CFF Explorer to open the malware sample. This tool can give us the information related to the malware that we want to investigate. The process of this step can be seen in the figures 2.

ile Settings ?					
🔊 🦉 🖏		best.exe	_		×
	٦.	Property	Val	ue	
File: best.exe	1	File Type	Por	table Executable 32 .NET Assembly	
- E ill Nt Headers		File Info	No	match found.	_
File Header Gotional Header		File Size	626	.00 KB (641024 bytes)	_
Date Directories [x]		PE Size	626	.00 KB (541024 bytes)	
Bection Headers [x] Control Directory		Created	Frid	lay 13 July 2018, 12:55:52	
mont Directory Resource Directory		Modified	Frid	lay 13 July 2018, 12:51:52	_
- Relocation Directory		Accessed	Frid	lay 13 July 2018, 12:55:52	
- I C NET Directory		MD5	D70	178F34BFA30EC5781FFB8F4508365E	
MetaData Streams		SHA-1	DS8	6787F851C548D151C17326E866CE025485CD9	
Tables Header		Property		Value	
Tables		CompanyNam	e	Lincoln National Corporation	
- 🔳 #GUID		FileDescription	1.	Optimizes and tweaks your Windows	
- Address Converter	11	FileVersion		6.13.24.4	
— 🐁 Dependency Walker	11	InternalName		besttt.exe	
- 🐁 Hex Editor		LegalCopyrigh	t	Copyright © 2018 Lincoln National Corporation	
- 🐁 Import Adder		OriginalFilenar	ne	besttt.exe	_
- Ouick Disassembler		ProductName		Optimizes and tweaks your Windows	
- A Resource Editor		ProductVersion		6.13.24.4	

Figure 2. Malware static analysis using CFF Explorer

Figure 2 shows information about the identity of the best.exe file. From the analysis obtained information about the file size which is 626 KB / 641024 bytes, with 32.NET assembly Portable Executable (PE) file type, and the manufacturer is Lincoln National Corporation. The file has the original name of best.exe with file version 6.13.24.4 (see Figure 3).

File View Go Help				
2000000000				
⊡-best.exe	pFile	Data	Description	Value
-IMAGE_DOS_HEADER	00000084	014C	Machine	IMAGE_FILE_MACHINE_I386
-MS-DOS Stub Program	00000086	0005	Number of Sections	
E-IMAGE_NT_HEADERS	88000000	59D0B34A	Time Date Stamp	2017/10/01 Sun 09:20:10 UTC
- Signature	0000008C	00000000	Pointer to Symbol Table	
- IMAGE_FILE_HEADER	00000090	00000000	Number of Symbols	
- IMAGE_OPTIONAL_HEADER	00000094	00E0	Size of Optional Header	
- IMAGE_SECTION_HEADER I 1%0_7	00000096	0102	Characteristics	
-IMAGE_SECTION_HEADER_text			0002	IMAGE_FILE_EXECUTABLE_IMAGE
-IMAGE_SECTION_HEADER .rsrc			0100	IMAGE_FILE_32BIT_MACHINE
-IMAGE_SECTION_HEADER .reloc				
- IMAGE_SECTION_HEADER				
- SECTION II%o_7				
IN-SECTION .text				
SECTION .rsrc				
SECTION .reloc				
# SECTION	1		88	

Figure 3. Analyzing PE structure with Preview

In Figure 3, it is shown that the best.exe file has a file structure consisting of several sections. The analysis using PEview above also obtained the date of the file made. The file is made on Sunday, 01/10/2017 at 09:20 UTC. Details of each section can be seen in the table III.

In addition to using software, this paper also uses a website that is virustotal.com to analyze malware samples so it can find out whether the file best.exe really is a malware and includes which type of malware the file as shown in Figure 4.

SHA256	0cfe9c1725	dfc5f73bb36ae2b1	68958f8ee8cf008f1240cf2808a91a513e22d4	
File name	besttt.exe			(2 ()
Detection	ratio: 37 / 67			🕑 2 🙂
Analysis d	ate: 2018-07-12	15:35:32 UTC (14	l hours, 17 minutes ago)	
🗊 Analysis	Q, File detail	🗙 Relationships	Additional information P Comments O Vote	15
Antivirus			Result	Update
Ad-Aware			Gen: Variant.Razy.362440	20180712
AegisLab			Troj Msil Agentic	20180712
Antiy-AVL			Trojan[Spy]/MSIL AGeneric	20180712
Arcabit			Trojan Razy D587C8	20180712
			Win32 Malware-gen	20180712
Avast				

Figure 4. Malware static analysis using virustotal.com

Based on the scanning of some antivirus on virustotal.com site, we obtained some information that the best.exe file is a malware that is included in Trojan type with the code SHA256 of 0cfe9c1725dfc5f73bb36ae2b168958f8ee8cf008f1240cf2 808a91a513e22d4.¬ The next step is to disassemble the malware sample to find out the commands used by the malware. IDA software is used to perform disassembly processes resulting in assembly language source code from machine-execution code. In this study, IDA software used is IDA freeware. Example of disassembling binary code can be seen in Figure 5.

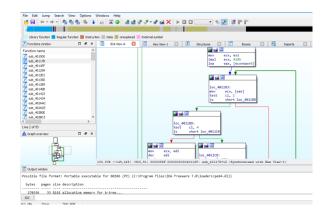


Figure 5. Example of disassembling binary code with IDA

4.2 Dynamic Analysis

The first step in the dynamic analysis is to record all malware activity when running the sample. To do so, Process Monitor and Wireshark are run first before running the sample. So that as soon as the sample is executed, the process the monitor shows all disk and registry activity of the currently running sample as shown in Figure 6 and 7 respectively.

					🔁 🗂 🔹 toress
			Destination	Protocol I	Length (arfo
	M-HIMAT	Source 10201201201010-002	245125 X 45138	108	55 [TCP Karp-Alive] 55615 + 441 [ACK] Seg-1 Ack+1 Min+63 Lan-1
	54.001949	74.125.24.139	192.168.43.247	TCP	54 443 - 35615 [RST, ACK] Seg=1 Ack=2 WLn+0 Len+0
	54,858734	192, 168, 43, 247	102,168,45,1	04/5	#2 Standard guery 0x75c7 A pa.sockets.stackexchange.com
	54.504788	192,168,43,1	192.168.43.247	DHS	104 Standard query response 0xf5c7 A qu.sockets.stackexchange.com A 190.252.
64.7	4, 307481	192,168,43,247	188.252,286.25	TCP	86 55620 + 443 [SVN] Seg+0 Wile+8102 Lan+0 PSS+1468 WS+256 SACK PERM+1
49.1	54.963496	198.252.286.25	192.155.43.247	TCP	86 445 + 55620 [599, ACK] Segue Ack-1 M10-4200 Len-8 M55-1400 M5+4 SACK PE.
78.1	14.963645	192.168.43.247	198.252.286.25	TEP	54 55628 + 443 [ACK] Seg=1 Ack=1 Win=16384 Len=8
	54.963984	192.168.43.247	198.252.286.25	TL5v1.2	571 Client Hello
	15.211353	198.252.286.25	192.168.43.247	TEP	54 443 + 55620 [ACK] Seq=1 Ack=518 Min=4716 Len=8
	15.500436	198.252.286.23	192-168-41-247	168	102 [ICP Previous segment not captured] 443 - 55620 [PSH, ACK] Seg-1401 Ack.
- 79	11.500810	198-252-286-25	192-108-437247	TCP	68 [TCP Day ACC 7001] \$1620 + 443 [ACK] Seq-\$10 Act+1 bin-15504 Lemon SLS- 102 [TCP Previous segment nut captured] 043 + 35620 [PSH, ACK] Seq-2009 Act De Alter Act Act Act Sector Act Act Cost Terration Science (Cost Act Act Act Act Act Act Act Act Act Ac
Frame 1 Inte Enca Arris [Tis [Tis [Tis [Tis]	Piconna rface id: 0 psulation ty yeal time: Ju a shift for h Time: 1531 a delta from a since refu since refu	In sine (432 bits), (\Device\MFF_(23704) pp: Ethernat (1) 1 33, 2018 16 (36:09) 474683, 33356900 see previous captured to previous displayed runce or first fram	102.102.43,242 H bytes: captured (45 100.8084-4054-5455-711 335569000 SE Asia Sta 10000 seconds] onds frame: 0.00000000 seconds 1: 0.00000000 seconds	TCP TCD 2 bits) on i procc88200}) andard Time conds] tconds] 1]	48 (Dr buy ACS 1981) Stolp + 46 (ACS) (synth Aniva viscotiski kove tata B) (Dr brechning storet or query) ACS - 5548 (Imp. 354, Storet ACS B) (Dr brechning storet or an even of a storet or an even of a storet or a storet as a storet or an even of a storet or a storet or a storet or a storet as a storet or an even of a storet or a storet or a storet as a storet or a storet as a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet or a storet
Frame 1 Frame 1 Frame 1 Inte Enca Arrs [Tim Epoc [Tim [Tim [Tim]Tim	ricestation of the second seco	<pre>in wire (432 bits), i ()Device NUP _(2)FOG per Ethernet (1) 1 15, 2018 10(38:03) this packet: 0.0000 s previous captured to previous displayed remens or first fram 70 bit 03 05 32 hs 5</pre>	102.103.43,245 54 bytes: captured: (45: 60-8064-4054-645(-711 335569000 52 Asia 5t; 60000 seconds] 00045 frame: 0.00000000 iscond; 10.00000000 iscond; 10.000000000 iscond; 10.000000000 iscond; 10.00000000 iscond; 10.00000000 iscond; 10.00000000 iscond; 10.00000000 iscond; 10.00000000 iscond; 10.00000000 iscond; 10.0000000 iscond; 10.0000000 iscond; 10.0000000 iscond; 10.0000000 iscond; 10.0000000 iscond; 10.0000000 iscond; 10.000000000000 iscond; 10.00000000000000000000000000000000000	TCP TCD 2 bits) on i procceszoo)) endard Time conds] conds] conds] conds] conds]	(B) (TO Sept. ACT (Ref)) State - 4/B (ACT) Sept. B. Ani-1 Min-Holis Low-Exc. B (B) (TO Section adjustment - expressed AC - TAke (Ref), ACT (Section Act (Section adjustment - expressed AC - TAke (Ref), ACT (Section Act (Section Act)) (Section Act) (Sec
Frame 1 Frame 1 Fra	Piconna rface id: 0 prolation ty yeal time: Ju a shift for h Time: JSJ a delta from a delta from a since refu since refu a4 ed 00 fi a4 ed 00 fi	<pre>in wire (432 bits), i ()Device NUP _(2)FOG per Ethernet (1) 1 15, 2018 10(38:03) this packet: 0.0000 s previous captured to previous displayed remens or first fram 70 bit 03 05 32 hs 5</pre>	102.004.432.243 100.0004.4034.4035.411 1000-0000.55.4.514.11 1355500000.55.4.514.11 100000 seconds] conds 10.0000000000 second 11.0.000000000 second 11.0.000000000 second 11.0.000000000 second 11.0.000000000 second 12.0.00000000 second 13.0.00000000 second 13.0.00000000 second 13.0.00000000 second 13.0.00000000 second 14.0.00000000 second 15.0.00000000 second 14.0.00000000 second 15.0.00000000 second 15.0.00000000 second 15.0.00000000 second 15.0.00000000 second 15.0.00000000 second 15.0.00000000 second 15.0.000000000 second 15.0.0000000000 second 15.0.0000000000 second 15.0.000000000 second 15.0.000000000 second 15.0.000000000 second 15.0.000000000 second 15.0.0000000000 second 15.0.000000000000 second 15.0.0000000000000000000000000000000000	TCP TCD 2 bits) on i procc88200}) andard Time conds] tconds] 1]	(1) (TO KO AGY (PH)) STOLE + 40 (AGY) Sayetta Ada's sin-toliki lawet and (1) (TO Produce and Stole + 40) (AGY) Sayetta Ada's sin-toliki lawet (Sayetta Ada's Ad

Figure 6. Analyzing malware network traffic with Wireshark

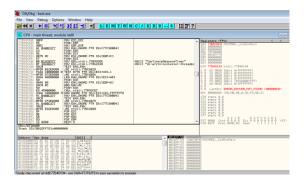


Figure 7. Debugging malware sample with Ollydbg

5. Result Analysis

Malware analysis that has been done with the best.exe file as malware sample using some malware analysis tools got a result that file of the best.exe is a malware which is virus Gen:Variant.Razy with the file size of 626 KB. Virus Gen:Variant.Razy is a virus that can be detected by some anti-virus and anti-malware because the virus is attacking computer systems with Windows OS. Some capabilities of the virus Gen:Variant.Razy according to the results of the analysis are as follows:

1. Can hide traces after download

The code is used:
"<Input Sample>" opened
"C:\best.exe:Zone.Identifier" (with delete access)

2. Know the computer name that is active

```
The code is used:

"<Input Sample>" (Path: "HKLM\SYSTEM\

CONTROLSET001\CONTROL\COMPUTERNAME\ACTIVECOMPUTERNAME";

Key: "COMPUTER NAME")
```

3. Make the computer sleep mode with a long

The code is used:

```
"<Input Sample>" sleeping for "1566804069" milliseconds
```

4. Create the new processes/tasks

The code is used:

"<Input Sample>" is creating a new process (Name:
"C:\best.exe", Handle: 800)

"<Input Sample>" is creating a new process (Name: "%WINDIR%\SysWOW64\wscript.exe", Handle: 612)

"wscript.exe" is creating a new process (Name: "%WINDIR%\SysWOW64 \cmd.exe", Handle: 772)

"cmd.exe" is creating a new process (Name: "%APPDATA%\remcos\remcos.exe", Handle: 124)

- 5. Can send information about infected computer to hacker
- 6. Record browsing history

This virus can be downloaded while searching the internet. Most of the users are unaware of how the virus has been installed in their computer until the antivirus software detects any

malware or virus threats. Based on the analysis that has been done, some common symptoms that occur when the computer has been infected with the virus Gen: Variant.Razy include:

- 1. High and abnormal CPU and VGA usage
- 2. Windows slows down
- 3. All programs work slower than before
- 4. Appears browser popups that recommend fake updates

To overcome these viruses, they can be checked using antivirus software and by resetting the browser to default settings.

6. Conclusion

Based on the analysis of malware using reverse engineering techniques that have been done in this study, the following conclusions are obtained:

- 1. Reverse engineering is an appropriate technique for use in analyzing malware
- 2. Static analysis and dynamic analysis methods each have advantages in the process of analyzing malware, then by combining the two methods will be able to provide more accurate results.
- 3. Each type of malware has its own way of working and threats, therefore malware analysis is an important thing to do in order to find the right steps to overcome and prevent malware attacks.

7. Reference

- H Utama, F W Wibowo 2015 Security Specification of WS-SecureConversation, 2015 IEEE Student Conference on Research and Development (SCOReD 2015), pp. 690 – 695
- [2] S Yusirwan, Y Prayudi and I Riadi 2015 Implementation of Malware Analysis using Static and D Dynamic Analysis Method (International Journal of Computer Applications) vol 117 no 6
- [3] H A Nugroho and Hamid 2013 *Reverse Engineering Techniques for Malware Analysis* (International Symposium on Digital Forensics and Security)
- [4] D Uppa, V Mehra and V Verma 2014 Basic Survey on Malware Analysis, Tools and Techniques (International Journal on Computational Sciences & Applications (IJCSA)) vol 4 no1
- [5] B Baskaran and A Ralescu 2016 A Study of Android Malware Detection and Machine Learning (MAICS)
- [6] S C Y Hutauruk, F A Yulianto and G B Satrya 2016 Malware Analysis pada Windows Operating System untuk Mendeteksi Trojan (e-Proceeding of Engineering) vol 3 no 2
- [7] M Agrawal, H Singh, N Gour and Mr A Kumar 2014 Evaluation on Malware Analysis (International Journal of Computer Science and Information Technologies (IJCSIT))

- [8] W Nurdiyanto 2013 Skenario Kombinasi Tools yang Efektif dalam Analisis Malware
- [9] R Alghamdi, K Alfalqi and M Waqdam 2015 Android Platform Malware Analysis (International Journal of Advanced Computer Science and Applications) vol 6 no 1
- [10] Z Lenny 2001 Reverse Engineering Malware
- [11] S Almarri and P Sant 2014 *Optimised Malware Detection in Digital Forensics* (International Journal of Network Security & Its Applications (IJNSA)) Vol 6 No 1
- F W Wibowo 2011 Interoperability of reconfiguring system on FPGA using a design entry of hardware description language (Computation and Communication Technologies: 3rd International Conference on Advances in Computing, Control, and Telecommunication Technologies, ACT Computer Science Series 1) pp. 79-83