Advance Malware Analysis Using Static And Dynamic Methodology

Saurabh Chaudhary

May 11, 2020

Advance Malware Analysis Using Static and Dynamic Methodology

Saurabh
Dept. Of Computer science and engineering
Lakshmi Narain College of Technology & Science
Bhopal, India
sudosaurabh@gmail.com

***Abstract*— As we are becoming more and more dependent on computers the attack vectors on them are increasing day by day. The cyberspace is becoming the battlefield of the 21st century as we are witnessing the increasing potential of a cyber-attack on the critical infrastructure. Malware are the most sophisticated evil code It is designed to damage computer systems without the knowledge of the owner these days malware are made up with special arbitrary to evade detection from the antivirus [1] with a huge potential to damage computer systems. Malware analysis is a process for studying the components and the behavior of malware. For analyzing malware we will use two types of methods static analysis and the dynamic analysis. In the static analysis the malware are examined without running it, whereas in dynamic analysis the malware is analyzed while running it in a virtual and controlled environment. In this research we are going to focus on malware analysis using the static and the dynamic method which will help us to access damage, to know the indicators of compromise and to determine the sophistication level of an intruder and to catch the creator of the malware.**

Keywords*— malware; cyber-attack; static analysis; cyberspace; dynamic analysis;

I. INTRODUCTION

"Malware" is the malicious software which are used as a single term to refer to the virus, spyware, worm, Trojan etc. Malware [2] is designed to damage a computer ,anything connected to computer or a networked pc. So malware are computer program which are made to damage your computer. There are a wide number and variety of malware it can be a virus, worm or Trojan [3]. Malware are created for evil purpose criminals can take advantage of the malware to get into computers and steal personal data, confidential data or use such information for profit.There has been a huge exponential growth in malware and the evil malware creators are using new complicated and advanced methods to program a malware which is hard to detect Also 285,000 new malware samples are seen every day over the internet. The increase in number of these malicious file has increased the demand for malware analysis or malware forensics [4] researcher. These days malware have the ability to bypass detection i.e. to evade detection. So it is important to get the complete analysis to access the impact or damage caused by malware sample.In this research, we are going to analyze a portable executable (P.E.) [5] sample file QQQ.exe this malware is recently discovered and is running in the wild and has an ability to evade maximum antivirus or malware detection system which uses signature-based analysis. Most antivirus uses signature-based detection of malware which only works when the signature of the malware is already present in the antivirus database.

In the research Malware sample, QQQ.exe will be analyzed using both advanced static and dynamic analysis to explain the processing of the malware.

II. METHODS FOR MALWARE ANALYSIS

Malware analysis is a method of analyzing a sample which can be a malware and to study the components and its behavior .There are two common methods for analyzing a malware static analysis and dynamic analysis.

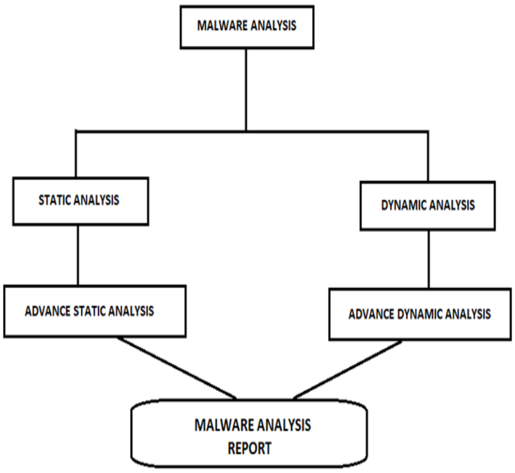


Fig. 1. Methods for Malware Analysis

Static analysis is the most common method for malware analysis. It is done by analyzing the malware without actually running it. Whereas in dynamic analysis the actual malware sample is observed while running the sample in a controlled environment generally in a virtual machine [6]. Static analysis is considered safe because in this method the malware is analyzed without the actual execution of the malware sample.

Static analysis can be split into two parts these are basic static analysis and the advanced static analysis. Like that dynamic analysis can also be further divided into two parts, the basic dynamic analysis and the advanced dynamic analysis. The method which we are going to use in our research are advanced dynamic analysis and advanced static analysis.

A. Advance Static Analysis

In advance static analysis further analysis are done including the basic static analysis for the strings , the linked library ,the exported and imported functions using disassembler like IDA or RADARE .

B. Advanced Dynamic Analysis

In advanced dynamic analysis method further analysis are done including the basic dynamic analysis. Advance debugging on malware is also done also the registry analysis and the windows system analysis is done

C. Report

The analysis report is generated on the characteristics and behavior of the malware sample keeping the results of advanced static and advanced dynamic analysis.

III. TOOLS USED

For our Research we will use several computer programs to understand more about the malware and in our process to analyze malware. For Basic static malware analysis we will be using these tools in Table1.

TABLE 1

Tools for basic static analysis	**Description**
PEid	It stands for portable executable identification It is used for detecting weather the PE file is packed or obfuscated
PEView	It is used for viewing the header and the section of the portable executables
MD5Deep	It is used for MD5 hash generation
Virustotal.com	Equipped with more than 50 antivirus ,it is used for antivirus scanning and other purposes

Basic dynamic analysis are done using these tools given in the Table2.

TABLE 2

Tools for basic dynamic analysis	**Description**
Virtualbox	It is a virtually controlled platform. It can load multiple guest Operating systems under just a single host operating-system (host OS). individual guest OS can be controlled within its own virtual machine
Sandboxie	It is a testing environment that does isolates untested computer program or code changes and outright experimentation from the production environment
Resource hacker	It shows the real time resource utilization of programs for a window machine
Processs hacker	It Monitors the system resources.it also debug software and detect malware.
Wireshark	Wire shark monitors the inbound and outbound network traffic

Also the Tools for advanced static analysis are depicted in Table 3.

TABLE 3

Tools for advanced static analysis	**Description**
Ida pro	IDA is a multi-processor disassembler and debugger that offers so many features to play with programs.
Dependency walker	It's a tool that scans Windows module (exe, dll,sys, etc.) and makes a hierarchical tree diagram of all dependent modules.

For advanced dynamic analysis these tools are used which can be seen in the table 4

TABLE 4

Tools for advanced dynamic analysis	Description
Immunity debugger	It's a tool used for performing debugging and reverse engineering to a sample.
Regshot	Regshot is used in analyzing the registry
PTFinder	This tool search for memory dump of a Microsoft Windows system for traces of processes and threads.

In the research will be using malware QQQ.exe as a sample malware, and all the test is to be performed using the advance static and the advance dynamic analysis method. Details of the process of malware analysis method can be seen in Fig.1

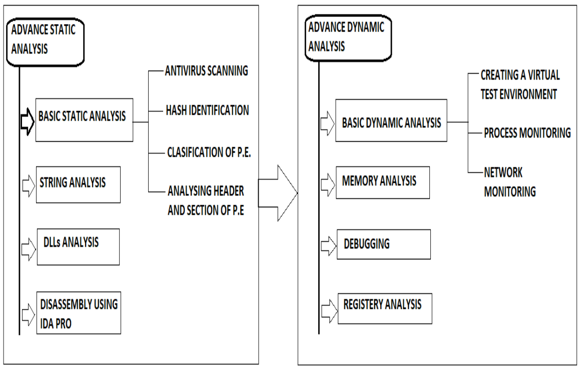


Fig. 1. Malware analysis method

IV. MALWARE ANALYSIS METHOD

For our Research we will use several computer programs to understand more about the malware and in our process to analyze malware.

A. Reverse engineering

Reverse engineering [7] is the method of taking apart the software for analyzing its parts it is the process by which hardware or software is de-constructed to reveal its designs, architecture, or to extract knowledge from the object. Reverse engineering's purpose is to improve the functionality by understanding its architecture whereas in malware analysis reverse engineering is done to understand the architecture and working of the malware.

B. Debugging

Debugging is the process for finding the instruction cycle of the machine language that is obtained while running the program. Debugging for malware is done to understand its working by observing the instructions made by the malware.

C. Disassembler

It is a computer program that makes the machine level language into a slight higher level language which can be understand easily by the user who is using it. By disassembling the malware file using a disassembler like IDA Pro researcher will be able to collect information from malware program which will be useful in identification of the characteristics of malware.

D. Packers (runtime)

These are computer programs that unpacks itself when the "packed file" is executed. It is also called as self-extracting files.

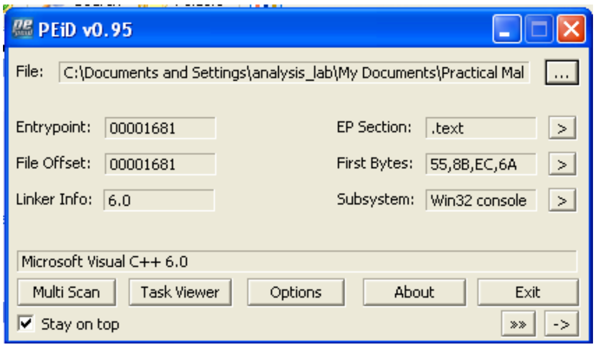
This technique is also called "executable compression". This compression technique was made to make files smaller. So users wouldn't have to unpack them manually. But in this era of computer and internet speed the need for smaller files is no more. So whenever we come across a packers being used nowadays, it is almost always for malicious purposes. It makes reverse engineering difficult with the advantage of low level digital foot-printing and leaving less trails.

E. Obfuscation

Obfuscation is a technique to make source code difficult to understand. Programmers may deliberately obfuscate code to conceal its purpose (security through obscurity) or its logic or implicit values embedded in it. The main purpose of it is to make reverse engineering more difficult and to make the architecture of the code a puzzle. It can be done manually or by using automated tool.

V. RESULT

For our Research we have used several computer programs to understand more about the malware and in our process to analyze malware. We have done experiments by running the malware sample and getting the output of the system.



File: C:\Documents and Settings\analysis_lab\My Documents\Practical Mal

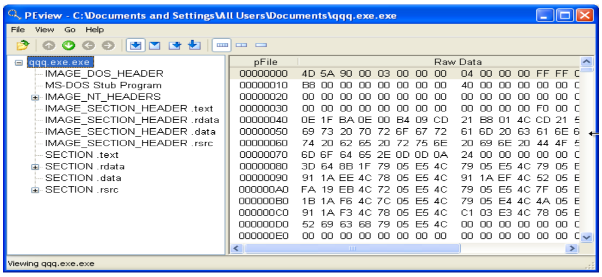
Entrypoint: 00001681
EP Section: .text
File Offset: 00001681
First Bytes: 55,8B,EC,6A
Linker Info: 6.0
Subsystem: Win32 console

Microsoft Visual C++ 6.0

☑ Stay on top

Fig. 2.PEiD detector to identify packers and compilers for PE files.

In the figure 2 we can see the entry point and the compiler used to compile the program once the type of obfuscated is known then we will find a way to deobfuscate protection of malware.

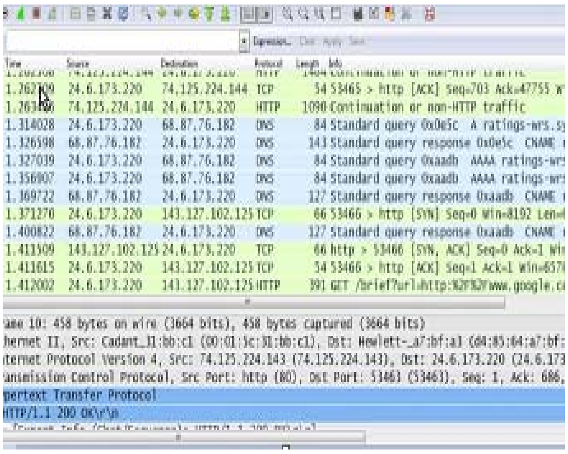


```
PEview - C:\Documents and Settings\All Users\Documents\qqq.exe.exe
qqq.exe.exe
  IMAGE_DOS_HEADER
  MS-DOS Stub Program
  IMAGE_NT_HEADERS
  IMAGE_SECTION_HEADER .text
  IMAGE_SECTION_HEADER .rdata
  IMAGE_SECTION_HEADER .data
  IMAGE_SECTION_HEADER .rsrc
  SECTION .text
  SECTION .rdata
  SECTION .data
  SECTION .rsrc

pFile     Raw Data
00000000  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF C
00000010  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 C
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 C
00000030  00 00 00 00 00 00 00 00  00 00 00 00 F0 00 C
00000040  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 5
00000050  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6
00000060  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 5
00000070  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 C
00000080  3D 64 8B 1F 79 05 E5 4C  79 05 E5 4C 79 05 E
00000090  91 1A EE 4C 78 05 E5 4C  91 1A EF 4C 52 05 E
000000A0  FA 19 EB 4C 72 05 E5 4C  79 05 E5 4C 7F 05 E
000000B0  1B 1A F6 4C 7C 05 E5 4C  79 05 E4 4C 4A 05 E
000000C0  91 1A F3 4C 78 05 E5 4C  C1 03 E3 4C 78 05 E
000000D0  52 69 63 68 79 05 E5 4C  00 00 00 00 00 00 C
000000E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 C
Viewing qqq.exe.exe
```

Fig. 3.Analyzing PE QQQ.exe with program PEview

In the above picture we can see PEview analyzing portable executable. From this experiment we will obtain the creation time of malware however it can also be false.

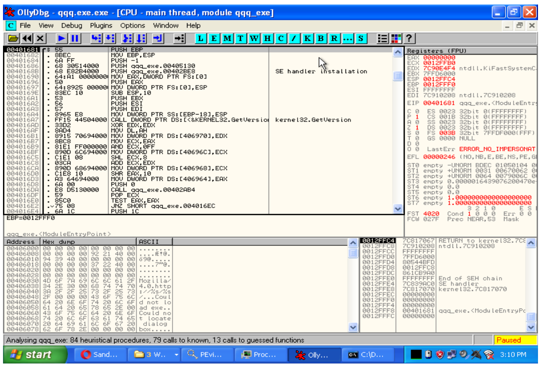


```
Time      Source          Destination     Protocol Length Info
1.262709  24.6.173.220    74.125.224.144  TCP      54 53465 > http [ACK] Seq=703 Ack=47755 W
1.263406  74.125.224.144  24.6.173.220    HTTP     1090 Continuation or non-HTTP traffic
1.314028  24.6.173.220    68.87.76.182    DNS      84 Standard query 0x0e5c  A ratings-wrs.s
1.326598  68.87.76.182    24.6.173.220    DNS      143 Standard query response 0x0e5c  CNAME
1.327039  24.6.173.220    68.87.76.182    DNS      84 Standard query 0xaadb  AAAA ratings-wr
1.356907  24.6.173.220    68.87.76.182    DNS      84 Standard query 0xaadb  AAAA ratings-wr
1.369722  68.87.76.182    24.6.173.220    DNS      127 Standard query response 0xaadb  CNAME
1.371270  24.6.173.220    143.127.102.125 TCP      66 53466 > http [SYN] Seq=0 Win=8192 Len=
1.400822  68.87.76.182    24.6.173.220    DNS      127 Standard query response 0xaadb  CNAME
1.411509  143.127.102.125 24.6.173.220    TCP      66 http > 53466 [SYN, ACK] Seq=0 Ack=1 Wi
1.411615  24.6.173.220    143.127.102.125 TCP      54 53466 > http [ACK] Seq=1 Ack=1 Win=657
1.412002  24.6.173.220    143.127.102.125 HTTP     391 GET /brief?url=http:%2F%2Fwww.google.c

ame 10: 458 bytes on wire (3664 bits), 458 bytes captured (3664 bits)
hernet II, Src: Cadant_31:bb:c1 (00:01:5c:31:bb:c1), Dst: Hewlett-_a7:bf:a1 (d4:85:64:a7:bf:
ternet Protocol Version 4, Src: 74.125.224.143 (74.125.224.143), Dst: 24.6.173.220 (24.6.173
ansmission Control Protocol, Src Port: http (80), Dst Port: 53463 (53463), Seq: 1, Ack: 686,
pertext Transfer Protocol
HTTP/1.1 200 OK\r\n
```

Fig. 4. Analyzing malware network activity with wireshark

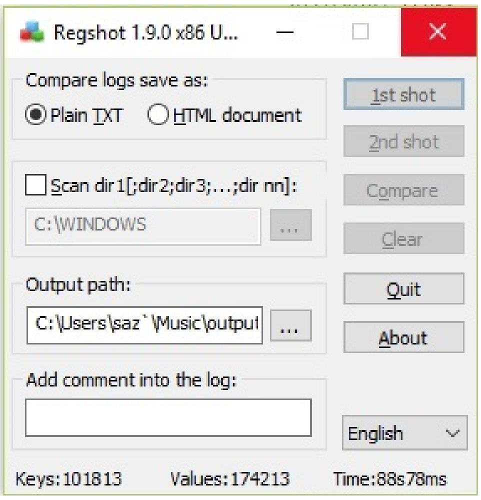
In fig.4. We are using wireshark for detecting network activity (all the inbound and outbound traffic) performed by malware. At this process we found malware QQQ.exe trying to communicate with a remote server.



```
OllyDbg - qqq.exe.exe - [CPU - main thread, module qqq_exe]
Analysing qqq_exe: 84 heuristical procedures, 79 calls to known, 13 calls to guessed functions
```

Fig. 5. Analyzing malware with ollydbg

We can also see the malware qqq.exe also calls a dll which is needed for using Graphical user interface application.



Regshot 1.9.0 x86 U...

Compare logs save as:

● Plain TXT ○ HTML document

□ Scan dir1[;dir2;dir3;...;dir nn]:

C:\WINDOWS

Output path:

C:\Users\saz`\Music\output

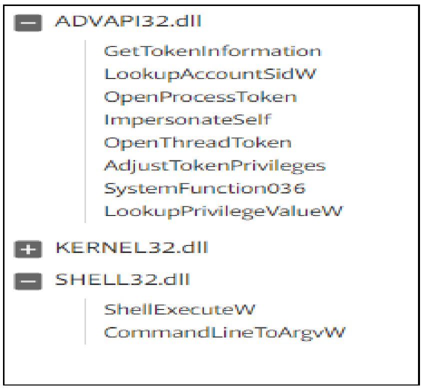
Add comment into the log:

Keys:101813 Values:174213 Time:88s78ms

Fig. 6. Analyzing registry with regshot

Based on our experiments done to malware QQQ.exe with both the methods of the advance static analysis and the advance dynamic analysis, the following was concluded from the analysis which are as follows: Malware QQQ.exe was created on 2018-08-17 23:31:11,it is a Trojan type malware targeting Intel 386 or later processors and compatible processors. It is importing ADVAPI32.dll, KERNEL32.dll, and SHELL32.dll having the MD5 as - (6cdcb9f86972efc4cfce4b06b6be053a) with file size 140kb.

When malware QQQ.exe already infected the computer system, now the malware usage lot of memory to run the program as well as infect another programs which runs in victim's computer. Malware QQQ.exe is also switching off the windows security system such as windows defender, firewall as well as that it is also contacting to remote malware server.



- ADVAPI32.dll
  - GetTokenInformation
  - LookupAccountSidW
  - OpenProcessToken
  - ImpersonateSelf
  - OpenThreadToken
  - AdjustTokenPrivileges
  - SystemFunction036
  - LookupPrivilegeValueW
- KERNEL32.dll
- SHELL32.dll
  - ShellExecuteW
  - CommandLineToArgvW

Fig. 7. Imports made by the malware

For the malware QQQ.exe we were able to find that it is coded in C++. It is a ransomware which is a type of Trojan which is asking for ransom on this bit coin address: (14hVKm7Ft2rxDBFTNkkRC3kGstMGp2A4hk) also it Drops file to %Public% and executes it.

VI. CONCLUSION

Throughout this research, we implemented malware analysis using the Advance static and the advance dynamic analysis methods to get a better result than the ordinary analysis methods. When analyzing the malware using advance method of static analysis which includes the basic static analysis as well as some advancements to it, we started with the identification of the program weather it is a malware or not, with this method we were able to find the malware creation time and date and the headers of the portable executable.

Meanwhile, on the analysis with advance static analysis methods which is capable of providing more complete information about characteristics, the architecture and capabilities of malware, such as the information about malware to infect another programs, as well as modifying the registry and create new files and folders.

On the other hand the dynamic methods of malware analysis can discover all the DLL imports of malware, the process of malware inside the system, its working and also the network traffic connection performed by malware against the server. Based on this research, advance static malware analysis and advance dynamic malware analysis combined are able to provide a more vivid and fascinating picture of the characteristics of malware QQQ.exe.

VII. REFERENCES

[1] Vigna, G. 2014. Antivirus Isn't Dead, It Just Can't Keep Up. Technical Report. Lastline Labs, May 2014.

[2] www.wikipedia.org/wiki/Malware

[3] www.kaspersky.co.in/resource-center/threats/trojans

[4] Almarri, S., & Sant, P. 2014. Optimised Malware Detection in Digital Forensics. International Journal of Network Security & Its Applications (IJNSA), Vol.6, No.1, January 2014.

[5] www.wikipedia.org/wiki/Portable_Executable

[6] www.wikipedia.org/wiki/Virtual_machine

[7] Eilam, E. 2003. Reversing - Secrets of Reverse Engineering. Indianapolis: Wiley Publishing, Inc.