ANALYSIS OF MALWARE BEHAVIOR IN WINDOWS AND LINUX OPERATING SYSTEM

[1] Karande Pooja [2] Dr.Priyanka Sharma

[1] Post Graduation, Cyber Security, M.Tech, Raksha Shakti University, Ahmedabad, Gujarat,India

[2] Professor&Head Dept of IT&Telecommunication Dept, Raksha Shakti University, Ahmedabad, Gujarat, India

Abstract: Malware, such as trojan horse, worms and spyware, rootkit, virus is a software which causes harm to computer system without the user's knowledge. we observed that windows as well as linux operating system is also the target for malware attacks. the purpose of this paper is to detect the behavior of malware in the operating systems such as windows and linux. for windows malware behavior analysis we use process hacker, process monitor, noriben sandbox, Inetsim, wireshark and for linux malware behavior analysis we use Limon sandbox, inetsim and wireshark. our approach first analyzes a malware sample in a controlled environment with the help of various tools. This is replete by detecting and analyzing the malware infected packets from a network.

Keywords: Malware Behavior Analysis, Limon sandbox, Noriben sandbox, wireshark , inetsim, Linux and windows operating system

I. INTRODUCTION

Malware is any program or file that is damaging to a computer user. various Types malware are including viruses, worms, Trojan horses and spyware. These malicious code can perform a different functions and monitoring users' computer activity without their Knowledge. Malware can cause massive harm to computer systems and worse yet, may turn a system into a dispatch machine as well. Let us look at Back Office specifically so we can highlight why a tool like this can become ugly if installed on victim systems.

A Malware works by hiding with in a set of evidently useful software programs. Once executed or installed in the system, this type of Malware will start infecting other files in the computer. A Malware is also usually capable of stealing important information from the user's computer. The developer will then be able to gain a level of control over the computer through the Malware. While these things are taking place, the client will notice that the damage computer system has become very slow or unexpected pop up without any activity from the user. Later on, this will result in a computer crash.

Internet security is an important element in networking. It needs protection against intruders. Even though many anti-virus software packets have been designed to analyze malicious codes, they still fail to do so. There are two common methods that an anti-virus software application uses to analyze Malware. The first, and by far the most common method of Malware detection to use a list of virus signature definitions; the second method is to use a dynamic analysis to find Malware based on common behaviors. The use of dynamic analysis involves inspecting the code in a file to see if it contains malware-like instructions.

Back Orifice involve two keys: a client application and a server application. The way in which BackOffice works is that the client application runs on one computer system and the server application runs on a different computer system. The client application connects to another computer system using the server application. The confusing part is the server installed on the victim. The only way for the server application of Back Orifice to be installed on a computer is for it to be installed deliberately. Obviously, the Malware does not come with a default installation of Windows and Unix system, so you must find a way to get the victim to install it.

II. MALWARE PROBLEMS

Malwares are difficult to detect because they appear to be useful programs or application and user tend to download them. malware represents a sophisticated attack because the attack is separated into two parts: the injection of the harmful code and then calling it, - which is one of the reasons for Malware being difficult to track.
This paper focuses on how it is possible to detect and analyze A Malware Sample in A Controlled Environment with The Help Of Various Tools.This Is Replete By Detecting And Analysing The Malware Infected Packets From A Network

III. TYPES OF MALWARE

There are various types of Malware that damage client computer system or threaten data integrity, or harm the functioning of the client’s computer system. Some types of Malware as listed below:

Table 1 malware sample

Windows Malware	**Linux Malware**
Trojan Dropper [5]	Trojan.Linux[1]
Trojan Generic [5]	Mirai.arm[5]
Gen: variant.strictor [3]	HEUR.Trojan Generic Script[2]
HW32 .Packed [3]	

A. Trojan horse or Trojan is a type of malware that is often disguised as legal software. Trojans can be employed by cyber-thieves and hackers trying to gain access to client's systems. Clients are typically tricked by some form of social engineering into loading and executing Trojans on their systems. Once activated, Trojans can enable cyber-criminals to spy on client, steal client’s sensitive data, and gain backdoor access to client system. These actions can include:

- ➢ Deleting data
- ➢ Blocking data
- ➢ Modifying data
- ➢ Copying data
- ➢ Damage the performance of computers or computer networks

B. Backdoors Trojans are the most dangerous type of Trojan and also the most widespread one. These Trojans are remote administration utilities that open affect machines to outside control via a LAN or the Internet. They function in the same way as legal remote administration code used by system administrators. This makes them hard to detect. The only dissimilarity between a legal administration tool and a backdoor is that backdoors are installed and launched without the knowledge or consent of the user of the victim machine.

IV. THE PROPOSED SOLUTION

The aim of this paper is mentioned before analyze the malware behavior in windows and linux operating system. The methodology includes three main parts:

1. Set up lab environment for windows malware analysis then analyze the malware behavior using tools
2. Set up lab environment for Linux malware analysis then analyze the malware behavior using tools
3. Compare the network behavior of malware in windows and Linux operating system

Analyze windows Malware Behavior

The following diagram shows the lab environment that will be used to perform behavior analysis in windows operating system.

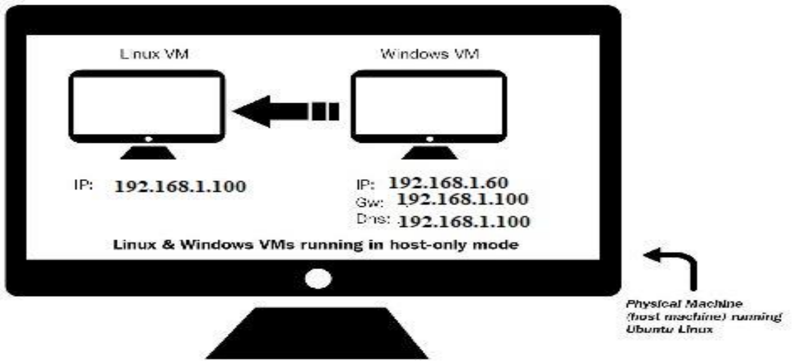

Linux VM
Windows VM
IP: 192.168.1.100
IP: 192.168.1.60
Gw: 192.168.1.100
Dns: 192.168.1.100
Linux & Windows VMs running in host-only mode
Physical Machine (host machine) running Ubuntu Linux

Fig 1 lab architecture for windows malware analysis

In this setup, both the Linux and Windows Virtual Machine were configured to use the host-only network configuration mode. The Linux Virtual Machine was preconfigured to an IP address of 192.168.1.100, and the IP address of the Windows Virtual Machine was set to 192.168.1.60. The default gateway and the DNS of the Windows Virtual Machine were set to the IP address of the Linux VM (192.168.1.100), so all the Windows network traffic is going through the Linux Virtual Machine. The Windows Virtual Machine will be used to run the malware sample during analysis, and the Linux Virtual Machine will be used to examine the network traffic.

Tools used for Windows Malware Analysis

1. Process Inspection with Process Hacker
2. Determining System Interaction with Process Monitor
3. Logging System Activities Using Noriben
4. Capturing Network Traffic with Wireshark
5. Simulating Services with INetSim

Analyze the Malware Sample in Windows Operating System

The following behavior analysis steps were followed:

1. Both the Windows VirtualMachine and the Linux VirtualMachine were reverted to the clean snapshots.
2. On Windows VirtualMachine, Process Hacker was started with administrator privileges to determine process attributes, and the Noriben sandbox was executed (which in turn started Process Monitor) to analyze the malware's interaction with the system.
3. INetSim was launched to simulate network services on Linux VirtualMachine, and Wireshark was executed and configured to capture the network traffic on the network interface.
4. With all the monitoring tools running, the malware was launched with administrator privileges for around 40 seconds.
5. After 40 seconds, Noriben was stopped on the Windows VirtualMachine. INetSim and Wireshark were stopped on the Linux VirtualMachine.

This involves collecting the data/reports from the monitoring tools and analyzing them to determine the malware's Network behavior.

Analyze Linux Malware Behavior

The following diagram shows the lab environment that will be used to perform behavior analysis in Linux operating system.

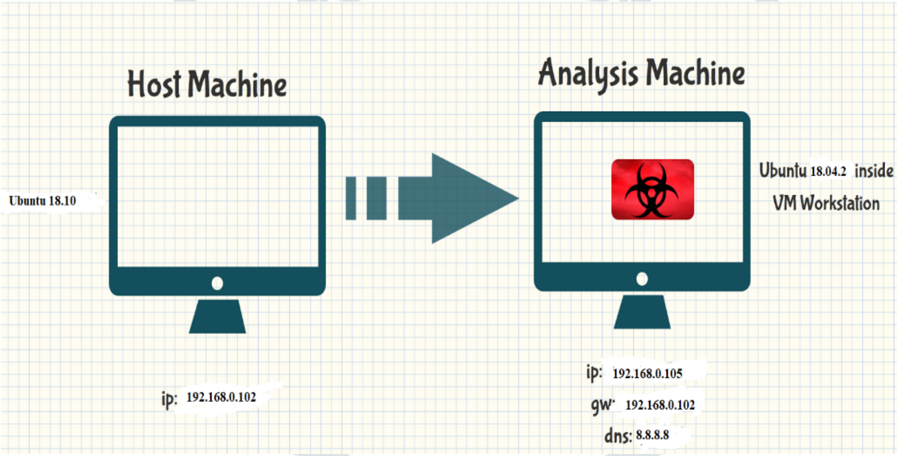


Fig 2 lab architecture for linux malware analysis

The setup consists of a host system running Ubuntu 18.10 (64 bit) with VMware Workstation Installed and analysis machine running Ubuntu 18.04.2 inside VMware Workstation in the bridge mode. malware will be run on the analysis machine, both host machine and analysis machine run with root privileges. The ip address of the host system is set to 192.168.0.102 and the ip address of the analysis machine is set to 192.168.0.105 and the gateway of the analysis machine is set to 192.168.0.102 (ip address of host system) and the dns server on the analysis machine is set to 8.8.8.8 The idea of this setup is to make sure that when the malware is run in the analysis machine all traffic of network goes through the host machine where the packets will be captured.

Tools used for Linux Malware Analysis

1. Limon sandbox
2. Wireshark

Setting up Limon for Linux malware analysis.

1. Configuring and Installing tools on Host Machine

   - VMware Workstation
   - YARA and YARA-Python
   - Ssdeep
   - Sysdig
   - INetSim
   - Create a directory to store YARA rules
   - create a directory to store analysis results

2. Configuring and Installing tools on Analysis Machine

   - Set Root password and enable graphical root login
   - Sysdig
   - Strace
   - PHP
   - Install packages to run 32 bits executable on 64-bit Ubuntu system
   - Create directory to transfer malware sample
   - Clean Bash History
   - Take a CleanSnapShot

3. Configuring Limon

This involves collecting the data/reports from the monitoring tools and analyzing them to determine the malware's Network behavior.

The Algorithm Process for Packet Information Grabbing

1. Determine IP, TCP and UDP alongside structure
2. Capture packet length
3. Capture source and destination MAC address
4. Examine the IP and ARP Protocol
5. If packets are infected by Malware then

   - Packet grabbing Date and Time
   - Examine Packet Type whether the Packet is TCP or UDP
   - Packet Source and Destination IP address
   - Payload is displayed

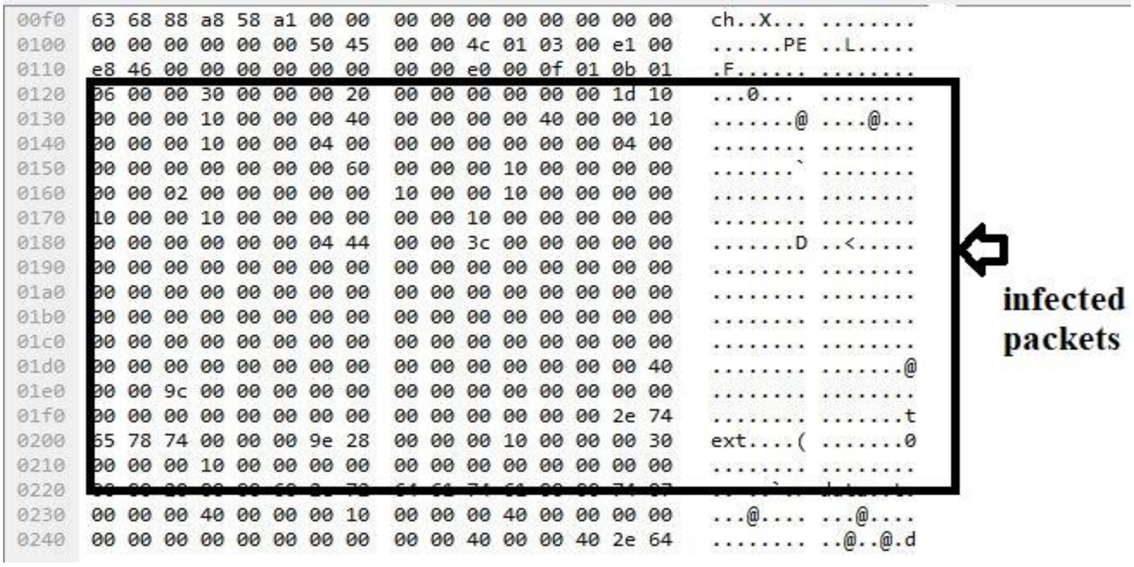
6. Exit

V. RESULTS

The analyses of a Malware packet comparing two normal and abnormal packets. After the abnormal packet had been is identify, the packets were analyzed to determine whether they are infected packets or not. Wireshark can capture and extract all the packet information that has been describe, then it will analyze all infected captured packet information and store all necessary required information into files for viewing. The possible information contains the protocols being used on a network fragment, but it concerns mainly the behavior of network TCP header protocol, IP header protocol and traffic between each source and destination.

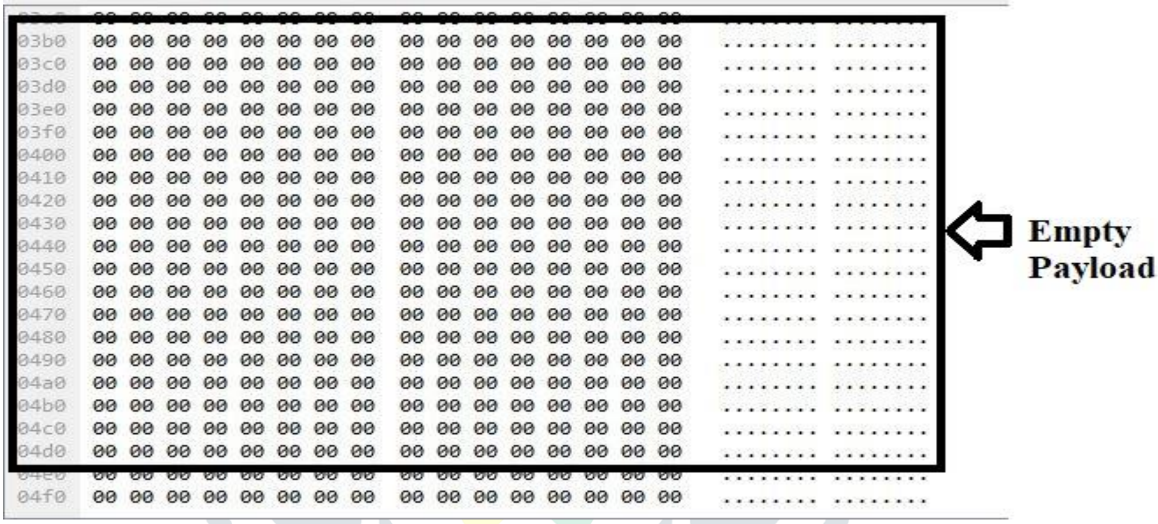
The results of the implementation to the Wireshark packet network for the operating systems Linux and Windows are as follows:

- The attack payload is obtained by malicious exe file in windows operating system whereas attack payload is obtained by elf file in Linux operating system. The attached file which contains the Malware has the following behaviors in operating systems.
- The infected packets in Windows Operating System, i.e. the Trojan dropper that was tested for this evaluation has certain information, such as length or file size. Figure shows the states from the TCP segment byte 7300 the payloads are not infected However, when the information is encrypted or the file is in process, the payloads after that are infected. The same thing occurs to the information available for the Trojan dropper where file size is 7354, when the network protocols processes the file information it becomes infected.
- The results of Windows based Wireshark show the Trojan dropper attack payload. the Trojan dropper produce a different pattern of behavior compared with normal behavior. The Trojan dropper output is empty payload. No data are -Trojan based on the results the empty payload is defined as an attack, compared with normal packets. Empty data are transmitted through the TCP flow. According to this behavior, it can be concluded or there can be strong accordance that this is an abnormal behavior of TCP flow caused by a harmful code.
- A Trojan dropper change the behavior of packets are infected after the information enters the network. The packets are not fully infected, only certain parts are infected, 00 means that no packet has been sent to the destination port from the source port.



```
00f0  63 68 88 a8 58 a1 00 00  00 00 00 00 00 00 00 00   ch..X... ........
0100  00 00 00 00 00 00 50 45  00 00 4c 01 03 00 e1 00   ......PE ..L.....
0110  e8 46 00 00 00 00 00 00  00 00 e0 00 0f 01 0b 01   .F...... ........
0120  06 00 00 30 00 00 00 20  00 00 00 00 00 00 1d 10   ...0...  ........
0130  00 00 00 10 00 00 00 40  00 00 00 00 40 00 00 10   .......@ ....@...
0140  00 00 00 10 00 00 04 00  00 00 00 00 00 00 04 00   ........ ........
0150  00 00 00 00 00 00 00 60  00 00 00 10 00 00 00 00   .......` ........
0160  00 00 02 00 00 00 00 00  10 00 00 10 00 00 00 00   ........ ........
0170  10 00 00 10 00 00 00 00  00 00 10 00 00 00 00 00   ........ ........
0180  00 00 00 00 00 00 04 44  00 00 3c 00 00 00 00 00   .......D ..<.....
0190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 40   ........ .......@
01e0  00 00 9c 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
01f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 2e 74   ........ .......t
0200  65 78 74 00 00 00 9e 28  00 00 00 10 00 00 00 30   ext....( .......0
0210  00 00 00 10 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0220  00 00 00 00 00 60 2e 72  64 61 74 61 00 00 74 07   .....`.r data..t.
0230  00 00 00 40 00 00 00 10  00 00 00 40 00 00 00 00   ...@.... ...@....
0240  00 00 00 00 00 00 00 00  00 00 40 00 00 40 2e 64   ........ ..@..@.d
```

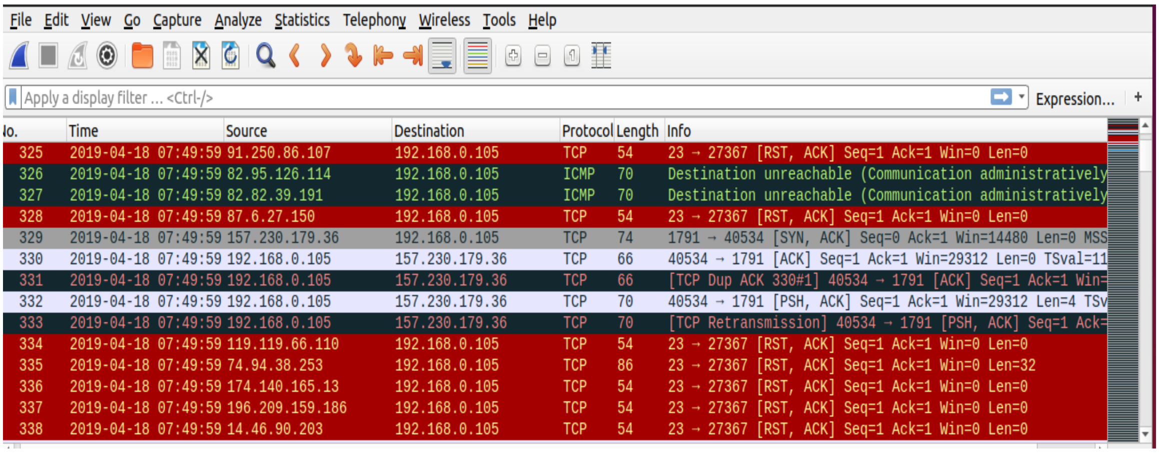
Fig 3 infected packets in windows operating system



```
03a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
03f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0400  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0410  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0420  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0430  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0440  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0450  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0460  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0470  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0480  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0490  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
04a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
04b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
04c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
04d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
04e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
04f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

Fig 4 empty payload in windows operating system

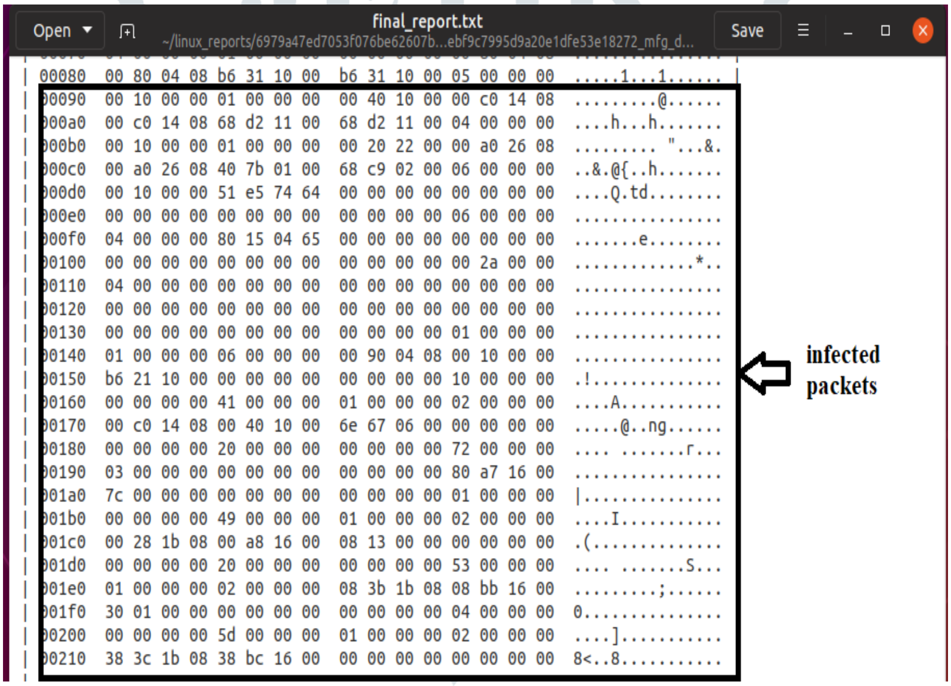
- The results for the Trojan Linux attack payload for Linux based Wireshark, show that the Trojan Linux produce a different pattern of behavior compared with normal behavior. The Trojan Linux output is also giving empty payload. No data are - Trojan based on the results the empty payload is defined as an attack, compared with normal packets. Empty data are transmitted through the TCP flow. in a normal packet capture TCP connection starts, a destination device receives a SYN packet from a source device and sends back a SYN ACK. The destination device must then hear an ACK of the SYN ACK before the connection is established. This is mentioned to as the "TCP three-way handshake."
- Bad packets received are packets that may include ICMP unreachable status or TCP Retransmission that may be indicative of an unusual number of failures which may be evidence of scanning of multiple destinations by the source device.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Expression…
+

```
No. Time                Source          Destination     Protocol Length Info
325 2019-04-18 07:49:59 91.250.86.107   192.168.0.105   TCP      54     23 → 27367 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
326 2019-04-18 07:49:59 82.95.126.114   192.168.0.105   ICMP     70     Destination unreachable (Communication administratively
327 2019-04-18 07:49:59 82.82.39.191    192.168.0.105   ICMP     70     Destination unreachable (Communication administratively
328 2019-04-18 07:49:59 87.6.27.150     192.168.0.105   TCP      54     23 → 27367 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
329 2019-04-18 07:49:59 157.230.179.36  192.168.0.105   TCP      74     1791 → 40534 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS
330 2019-04-18 07:49:59 192.168.0.105   157.230.179.36  TCP      66     40534 → 1791 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=11
331 2019-04-18 07:49:59 192.168.0.105   157.230.179.36  TCP      66     [TCP Dup ACK 330#1] 40534 → 1791 [ACK] Seq=1 Ack=1 Win=
332 2019-04-18 07:49:59 192.168.0.105   157.230.179.36  TCP      70     40534 → 1791 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=4 TSv
333 2019-04-18 07:49:59 192.168.0.105   157.230.179.36  TCP      70     [TCP Retransmission] 40534 → 1791 [PSH, ACK] Seq=1 Ack=
334 2019-04-18 07:49:59 119.119.66.110  192.168.0.105   TCP      54     23 → 27367 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
335 2019-04-18 07:49:59 74.94.38.253    192.168.0.105   TCP      86     23 → 27367 [RST, ACK] Seq=1 Ack=1 Win=0 Len=32
336 2019-04-18 07:49:59 174.140.165.13  192.168.0.105   TCP      54     23 → 27367 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
337 2019-04-18 07:49:59 196.209.159.186 192.168.0.105   TCP      54     23 → 27367 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
338 2019-04-18 07:49:59 14.46.90.203    192.168.0.105   TCP      54     23 → 27367 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

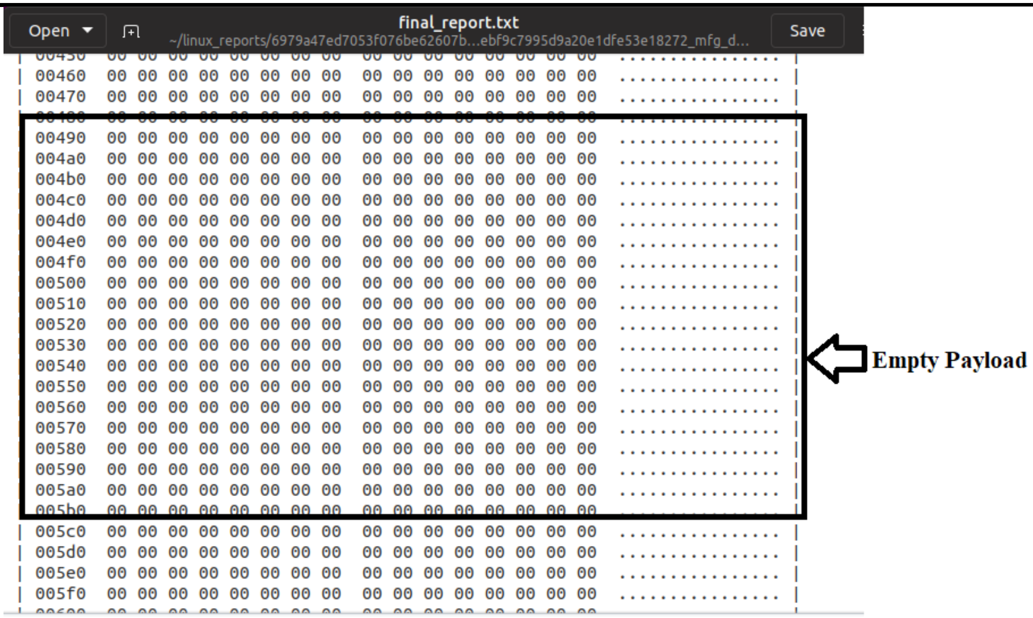
Fig 5 bad packets captured via wireshark in linux operating system

- ➢ Linux and Windows have the same output for a Trojan attack through the infected packet-based Wireshark output. The Linux and Windows Operating Systems have the same behavior for captured infected Packets. Empty data are transmitted through the TCP flow. According to this behavior, it can be concluded or there can be strong accordance that this is an abnormal behavior of TCP flow caused by a harmful code in Linux Operating System.


Open
final_report.txt
~/linux_reports/6979a47ed7053f076be62607b...ebf9c7995d9a20e1dfe53e18272_mfg_d...
Save
infected packets

```
| 00080  00 80 04 08 b6 31 10 00  b6 31 10 00 05 00 00 00  .....1...1......
| 00090  00 10 00 00 01 00 00 00  00 40 10 00 00 c0 14 08  .........@......
| 000a0  00 c0 14 08 68 d2 11 00  68 d2 11 00 04 00 00 00  ....h...h.......
| 000b0  00 10 00 00 01 00 00 00  00 20 22 00 00 a0 26 08  ......... "...&.
| 000c0  00 a0 26 08 40 7b 01 00  68 c9 02 00 06 00 00 00  ..&.@{..h.......
| 000d0  00 10 00 00 51 e5 74 64  00 00 00 00 00 00 00 00  ....Q.td........
| 000e0  00 00 00 00 00 00 00 00  00 00 00 00 06 00 00 00  ................
| 000f0  04 00 00 00 80 15 04 65  00 00 00 00 00 00 00 00  .......e........
| 00100  00 00 00 00 00 00 00 00  00 00 00 00 00 2a 00 00  .............*..
| 00110  04 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
| 00120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
| 00130  00 00 00 00 00 00 00 00  00 00 00 00 01 00 00 00  ................
| 00140  01 00 00 00 06 00 00 00  00 90 04 08 00 10 00 00  ................
| 00150  b6 21 10 00 00 00 00 00  00 00 00 00 10 00 00 00  .!..............
| 00160  00 00 00 00 41 00 00 00  01 00 00 00 02 00 00 00  ....A...........
| 00170  00 c0 14 08 00 40 10 00  6e 67 06 00 00 00 00 00  .....@..ng......
| 00180  00 00 00 00 20 00 00 00  00 00 00 00 72 00 00 00  .... .......r...
| 00190  03 00 00 00 00 00 00 00  00 00 00 00 80 a7 16 00  ................
| 001a0  7c 00 00 00 00 00 00 00  00 00 00 00 01 00 00 00  |...............
| 001b0  00 00 00 00 49 00 00 00  01 00 00 00 02 00 00 00  ....I...........
| 001c0  00 28 1b 08 00 a8 16 00  08 13 00 00 00 00 00 00  .(..............
| 001d0  00 00 00 00 20 00 00 00  00 00 00 00 53 00 00 00  .... .......S...
| 001e0  01 00 00 00 02 00 00 00  08 3b 1b 08 08 bb 16 00  .........;......
| 001f0  30 01 00 00 00 00 00 00  00 00 00 00 04 00 00 00  0...............
| 00200  00 00 00 00 5d 00 00 00  01 00 00 00 02 00 00 00  ....]...........
| 00210  38 3c 1b 08 38 bc 16 00  00 00 00 00 00 00 00 00  8<..8...........
```

Fig 6 infected packets in linux operating system


Open
final_report.txt
~/linux_reports/6979a47ed7053f076be62607b...ebf9c7995d9a20e1dfe53e18272_mfg_d...
Save
Empty Payload

```
| 00460  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00470  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00480  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00490  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 004a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 004b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 004c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 004d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 004e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 004f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00500  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00510  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00520  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00530  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00540  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00550  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00560  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00570  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00580  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 00590  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 005a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 005b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 005c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 005d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 005e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
| 005f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................ |
```

Fig 7 empty payload in linux operating system

Comparison of Malware Behavior Between Linux and Windows Operating System

Table 2 comparison of malware behavior in windows and linux operating system

Events	Windows Operating System	Linux Operating System
Operating System Behavior	Windows Operating System Unsteady as compared to Linux Operating System because Windows is easily infected by virus without user's knowledge.	Linux is Steadier than Windows Operating System.
Malware Execution file	.exe file can only run in Windows Operating System	Only elf file can run in Linux Operating System. Exe file does not execute in Linux Operating System it's shows an error message.
Network Behavior of Malware	As an information enter the network packets are infected.	As an information enter the network Packets are infected.
Trojan attack Packets Infected Payload	Packets are infected only certain parts. They are not fully infected	Same result gives the wireshark, packets are infected only certain parts.

VI. CONCLUSION

From the implementation we can conclude that:

- Linux and Windows have the same output for a Trojan attack through the infected packet-based Wireshark. The infected captured packet for both Linux and Windows has the same behavior. The designed code is able to capture TCP, UDP and also ICMP protocol information.
- The objectives of this paper have been partially achieved in the following perception first infected packet-based Wireshark output has been captured and network information has discovered and secondly Trojan attack from a computer network fragmented has been detected and monitored.

REFERENCES

[1] E. a. G. M. a. F. Y. a. B. D. Cozzi, "Understanding linux malware," *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 161--175, 2018.

[2] M. K. A, "Learning Malware Analysis," 2018.

[3] S. S. a. K. M. A. a. J. C. Darshan, "Windows malware detection based on cuckoo sandbox generated report using machine learning algorithm," *Industrial and Information Systems (ICIIS), 2016 11th International Conference on*, pp. 534--539, 2016.

[4] K. Monnappa, "Automating linux malware analysis using limon sandbox," 2015.

[5] S. a. G. A. A. Nari, "Automated malware classification based on network behavior," 2013.

[6] G. a. A.-B. H. M. a. o. Al-Saadoon, "A comparison of trojan virus behavior in Linux and Windows operating systems," *arXiv preprint arXiv:1105.1234*, 2011.

[7] J. A. a. A.-B. A. a. X. S. a. S. R. Morales, "Analyzing and exploiting network behaviors of malware," 2010.

[8] U. a. H. I. a. B. D. a. K. E. a. K. C. Bayer, "A View on Current Malware Behaviors," 2009.

[9] K. a. H. T. a. W. C. a. D. P. a. L. P. Rieck, "Learning and classification of malware behavior," 2008.

[10] "http://malware-unplugged.blogspot.com/2015/11/setting-up-limon-sandbox-for-analyzing.html".

[11] M. Bishop, "An Overview of Computer Viruses in a Paper Environment".