Sniffing and Detection of Packets

[1]Dr. N Haraprasad, [2]S Nanjundaswamy, [3]Mahadeva Prasad P

[1] Department of Computer Science & Engineering, SJCE, Mysore,India.
[2&3] Department of Computer Science & Engineering, SJCE, Mysore,India.

Abstract - Packet sniffing is the process of checking and retrieving all data packets that pass through the network using a software application or hardware device. Sniffers can be used to control all types of traffic, whether protected or unprotected. Using sniffers, an attacker can obtain information that may be useful to him for the next attack. This article discusses the basic operation of a packet sniffer, sniff-sensitive network protocols, and various software that can be used for sniffing. This article also describes possible defense techniques used to defend against sniffing attacks. Finally, the work ends with a description of some snooping detection techniques. Sniffers are not hacking tools, but they can help a hacker launch other attacks, such as session hijacking, DOS attack, MITM attack, and so on.

Keywords - Wireshark, packet sniffing, network security, HTTP, FTP

I. INTRODUCTION

A sniffer is a program like device that eavesdrops on network traffic by intercepting information traveling on the network. Sniffers are primarily a "Data Interception" technology [1]. They work because Ethernet is built on a shared principle. Most networks use broadcasting technology, in which messages for one computer can be read by another computer on that network. In practice, all computers except the computer listed in the message ignore this message. However, computers can receive messages even if it is not for them. The sniffer [1] is used for this.

Using sniffing, an attacker can intercept packets such as Syslog traffic, DNS traffic, Web traffic, e-mail, and other types of data traffic. By obtaining these packages, an attacker could reveal information such as data, usernames, and password protocols such as HTTP, POP, IMAP, SMTP, FTP, and Telnet. The snooping process is performed using promiscuous ports. This article discusses the basic operations of a packet sniffer, sniff-vulnerable protocols, the different types of devices used for sniffing, sniffing defense techniques, and sniffing detection techniques [2].

II. WORKING WITH SNIFFERS

In the snooping process, an attacker is connected to the target network to sense packets. Using sniffers that turn an attacker's system network card (NIC) into promiscuous mode, the attacker intercepts the packet [3]. Once an attacker retrieves a packet, those packets can be decrypted to obtain information. Sniffers can be widely used to hack a system or network. The steps that an attacker takes when using sniffer to hack a network are listed below and are shown in Figure 1:
a) An attacker who decides to hack the network first discovers the appropriate switch to gain access to the network and connects the system to one of the switch ports.
b) After the switch is successfully connected, the attacker will try to find out network information, such as network topology, using the network discovery tool.
c) By analyzing the network topology, an attacker identifies the victim's computer to control the attacks.
d) After identifying the target, the attacker will use ARP spoofing techniques to send a false (fake)
ARP message
e) The previous step will help the attacker redirect all traffic from the victim's computer to the attacker's computer. It was a man-in-the-middle (MITM) attack. f) The attacker now sees all data packets sent and received by the victim and can obtain confidential information such as username, password, credit card details, PIN, etc.

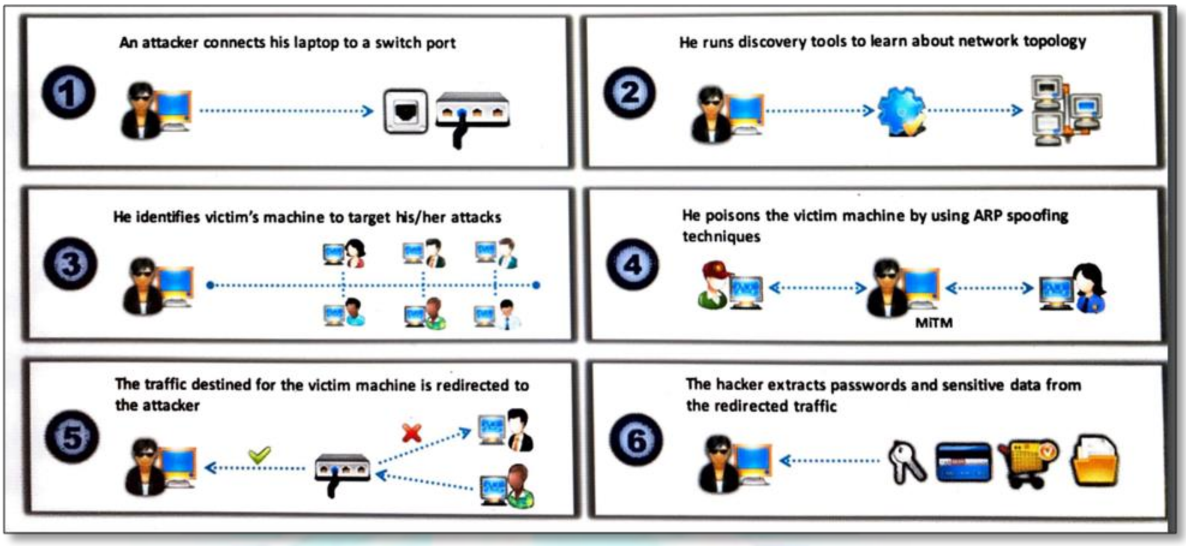

1
An attacker connects his laptop to a switch port
2
He runs discovery tools to learn about network topology
3
He identifies victim's machine to target his/her attacks
4
He poisons the victim machine by using ARP spoofing techniques
MITM
5
The traffic destined for the victim machine is redirected to the attacker
6
The hacker extracts passwords and sensitive data from the redirected traffic

Figure 1: - network hacking using sniffer

III. PROTOCOLS ARE GOOD FOR SNIFFEN

The following network protocols are easy to browse. The main reason for snooping on these protocols is to obtain confidential data such as passwords.

Telnet and Rlogin

Telnet is a protocol used to communicate with a remote host (via port 23) on a network using a command line terminal. Rlogin allows an attacker to log on to a remote network computer over a TCP connection. Both protocols do not provide encryption. This is why the data flowing between clients connected via all protocols is plain text and can be easily sniffed. Attackers can sniff keystrokes, including usernames and passwords.

HTTP

Due to vulnerabilities in the standard version of HTTP, Web sites that implement HTTP transmit user data over a network in clear text that attackers can read and steal user credentials.

SNMP

SNMP is a TCP / IP-based protocol used to exchange management information between devices connected to a network. The first version of SNMP (SNMPv1) did not offer strong security, which leads to data transfer in plain text format. Attackers exploit this version's vulnerabilities to obtain plaintext passwords.

NNTP

The Network News Transfer Protocol (NNTP) distributes, discovers, obtains and distributes news articles with reliable stream-based news coverage among the ARPA Internet community. The protocol fails to encrypt data and gives the attacker the opportunity to sniff out sensitive information.

POP

Post Office Protocol (POP) allows a user's workstation to access an e-mail server. The user can send e-mails from the workstation to the mailbox server via Simple Mail Transfer Protocol (SMTP). Attackers can easily detect plain text data streaming over a POP network due to poor protocol security implementation.

FTP

File Transfer Protocol (FTP) allows clients to share files between computers on a network. This protocol cannot provide encryption. This is why attackers snoop on data and credentials by running tools like Cain & Abel

IMAP

The Internet Message Access Protocol (IMAP) allows a client to access and manipulate e-mail messages on a server. This protocol provides insufficient security and allows attackers to obtain user data and references in clear text.

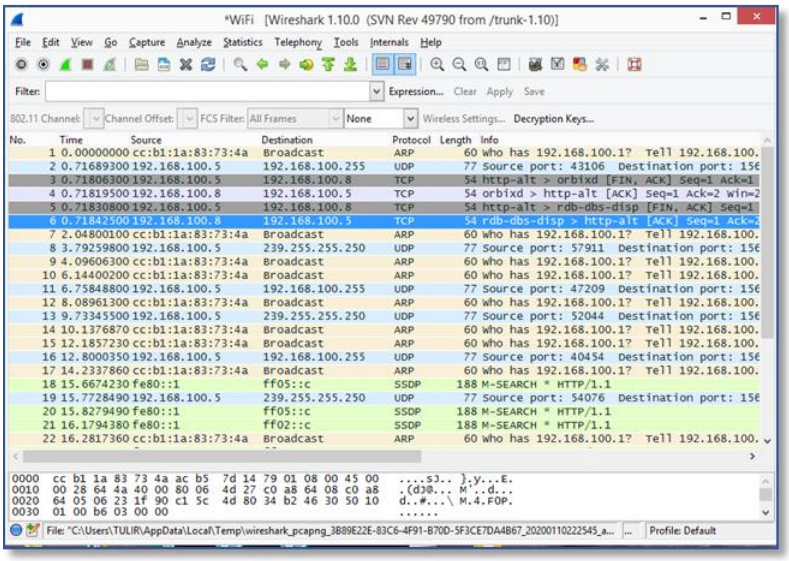
IV. PANDING TOOLS TO THE PACKAGING

System administrators use automated tools to control their network, but attackers use these tools to steal network

data. This section describes the various package tracking tools / software
Wireshark

Figure 2

Wireshark allows you to capture and interactively browse traffic running on a computer network. This tool uses


*WiFi [Wireshark 1.10.0 (SVN Rev 49790 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
802.11 Channel: Channel Offset: FCS Filter: All Frames None
Wireless Settings... Decryption Keys...

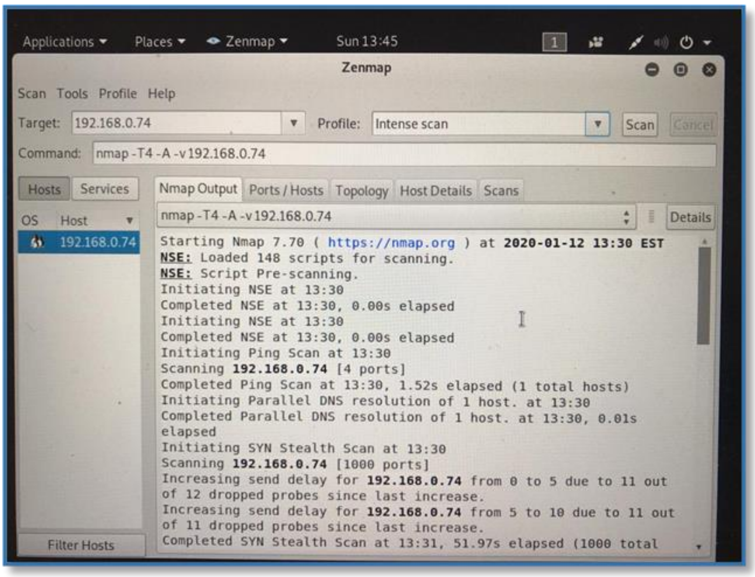
No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
2	0.71689300	192.168.100.5	192.168.100.255	UDP	77	Source port: 43106 Destination port: 156
3	0.71806300	192.168.100.5	192.168.100.8	TCP	54	http-alt > orbixd [FIN, ACK] Seq=1 Ack=1
4	0.71819500	192.168.100.8	192.168.100.5	TCP	54	orbixd > http-alt [ACK] Seq=1 Ack=2 Win=2
5	0.71830800	192.168.100.5	192.168.100.8	TCP	54	http-alt > rdb-dbs-disp [FIN, ACK] Seq=1
6	0.71842500	192.168.100.8	192.168.100.5	TCP	54	rdb-dbs-disp > http-alt [ACK] Seq=1 Ack=2
7	2.04800100	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
8	3.79259800	192.168.100.5	239.255.255.250	UDP	77	Source port: 57911 Destination port: 156
9	4.09606300	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
10	6.14400200	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
11	6.75848800	192.168.100.5	192.168.100.255	UDP	77	Source port: 47209 Destination port: 156
12	8.08961300	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
13	9.73345500	192.168.100.5	239.255.255.250	UDP	77	Source port: 52044 Destination port: 156
14	10.1376870	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
15	12.1857230	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
16	12.8000350	192.168.100.5	192.168.100.255	UDP	77	Source port: 40454 Destination port: 156
17	14.2337860	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.
18	15.6674230	fe80::1	ff05::c	SSDP	188	M-SEARCH * HTTP/1.1
19	15.7728490	192.168.100.5	239.255.255.250	UDP	77	Source port: 54076 Destination port: 156
20	15.8279490	fe80::1	ff05::c	SSDP	188	M-SEARCH * HTTP/1.1
21	16.1794380	fe80::1	ff02::c	SSDP	188	M-SEARCH * HTTP/1.1
22	16.2817360	cc:b1:1a:83:73:4a	Broadcast	ARP	60	Who has 192.168.100.1? Tell 192.168.100.

```
0000  cc b1 1a 83 73 4a ac b5  7d 14 79 01 08 00 45 00   ....sJ.. }.y...E.
0010  00 28 64 4a 40 00 80 06  4d 27 c0 a8 64 08 c0 a8   .(dJ@... M'..d...
0020  64 05 06 23 1f 90 c1 5c  4d 80 34 b2 46 30 50 10   d..#...\ M.4.F0P.
0030  01 00 b6 03 00 00                                  ......
```

File: "C:\Users\TULIR\AppData\Local\Temp\wireshark_pcapng_3B89E22E-83C6-4F91-B70D-5F3CE7DA4B67_20200110222545_a... Profile: Default

Winpcap to retrieve packets on supported networks. Captures live network traffic from Ethernet, IEEE 802.11, PPP / HDLC, ATM, Bluetooth, USB, Token ring, Frame Relay, FDDI networks. The program can edit the downloaded files using the command line. The filter set for the custom data display can be specified using the display filter [4].

Zenmap is the authorized graphical user interface(GUI) for the Nmap Security Scanner. Zenmap is accessible for Windows, Linux, Mac, and BSD. Zenmap may be used to read live captures or save captures for later viewing.


Applications Places Zenmap Sun 13:45
Zenmap
Scan Tools Profile Help
Scan
Cancel
Hosts Services
OS Host
Nmap Output Ports / Hosts Topology Host Details Scans
Details
Filter Hosts

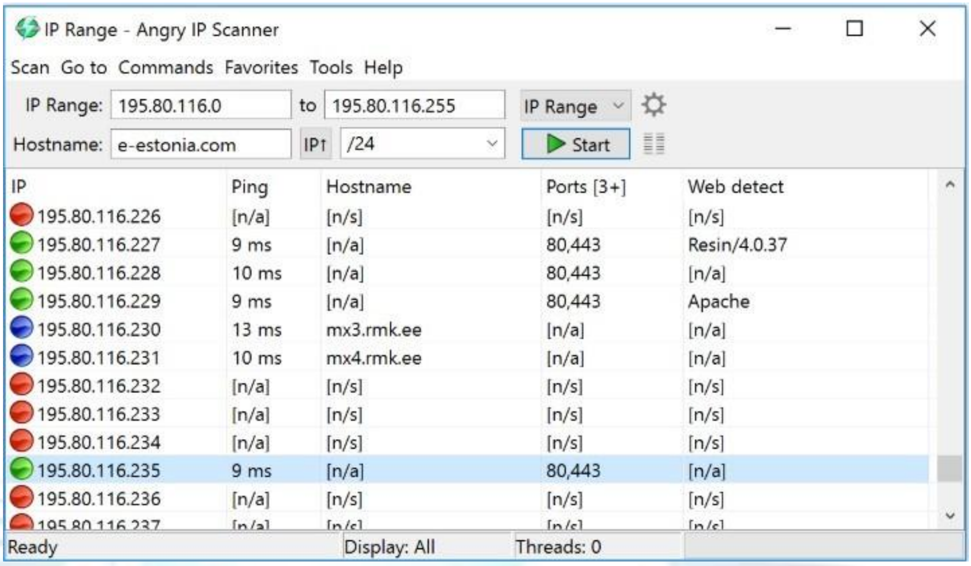
Target: 192.168.0.74
Profile: Intense scan
Command: nmap -T4 -A -v 192.168.0.74

192.168.0.74

nmap -T4 -A -v 192.168.0.74

```
Starting Nmap 7.70 ( https://nmap.org ) at 2020-01-12 13:30 EST
NSE: Loaded 148 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 13:30
Completed NSE at 13:30, 0.00s elapsed
Initiating NSE at 13:30
Completed NSE at 13:30, 0.00s elapsed
Initiating Ping Scan at 13:30
Scanning 192.168.0.74 [4 ports]
Completed Ping Scan at 13:30, 1.52s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:30
Completed Parallel DNS resolution of 1 host. at 13:30, 0.01s
elapsed
Initiating SYN Stealth Scan at 13:30
Scanning 192.168.0.74 [1000 ports]
Increasing send delay for 192.168.0.74 from 0 to 5 due to 11 out
of 12 dropped probes since last increase.
Increasing send delay for 192.168.0.74 from 5 to 10 due to 11 out
of 11 dropped probes since last increase.
Completed SYN Stealth Scan at 13:31, 51.97s elapsed (1000 total
```

Figure 3


IP Range - Angry IP Scanner
Scan Go to Commands Favorites Tools Help
IP Range
Start

IP Range: 195.80.116.0 to 195.80.116.255
Hostname: e-estonia.com IP↑ /24

IP	Ping	Hostname	Ports [3+]	Web detect
🔴 195.80.116.226	[n/a]	[n/s]	[n/s]	[n/s]
🟢 195.80.116.227	9 ms	[n/a]	80,443	Resin/4.0.37
🟢 195.80.116.228	10 ms	[n/a]	80,443	[n/a]
🟢 195.80.116.229	9 ms	[n/a]	80,443	Apache
🔵 195.80.116.230	13 ms	mx3.rmk.ee	[n/a]	[n/a]
🔵 195.80.116.231	10 ms	mx4.rmk.ee	[n/a]	[n/a]
🔴 195.80.116.232	[n/a]	[n/s]	[n/s]	[n/s]
🔴 195.80.116.233	[n/a]	[n/s]	[n/s]	[n/s]
🔴 195.80.116.234	[n/a]	[n/s]	[n/s]	[n/s]
🟢 195.80.116.235	9 ms	[n/a]	80,443	[n/a]
🔴 195.80.116.236	[n/a]	[n/s]	[n/s]	[n/s]
🔴 195.80.116.237	[n/a]	[n/s]	[n/s]	[n/s]

Ready Display: All Threads: 0

Figure 4

AngryIP Scanner [6] is an open source, fast platform scanner designed to be extremely fast and very easy to use. AngryIP addresses the following features: Portable zero installation on some platforms; ping control; NetBIOS information; resolutions; specifies the MAC address; identify the currently logged in user; engage in the system; Scan results can be saved as CSV, TXT, XML or IP Portlist; and fast multi-threaded scanning. AngryIP Scanner powered by angryziber.

Figure 4: - Angry IP Scanner

Cain and Abel

Cain & Abel is a password recovery tool for Microsoft operating systems. It facilitates the acquisition of various types of passwords through network sniffing, cracking of encrypted passwords by means of dictionary attacks, Brute-Force and Cryptanalysis attacks, recording of VoIP call attacks, decoding of encrypted passwords,

retrieving wireless network keys, discovering password boxes, retrieving cache passwords, and analyzing routing protocols. The program does not use any software vulnerabilities or errors that cannot be fixed by a small test. It covers some aspects of security / vulnerabilities present in protocol standards, authentication methods, and caching mechanisms; its main purpose is to make it easier to obtain passwords and links from various sources, but it also provides some "non-standard" tools for Microsoft Windows users. Cain & Abel was developed in the hope of being useful to network administrators, trainers,

securityconsultants/professionals,forensicstaff,securitysoftwarevendors,professionalpenetrationtesterand e veryoneelsethatplanstouseitforethicalreasons.Thelatestversion(Cain&Abelv4.9.56)[7]isfasterandcontainsa lotofnewfeatureslikeAPR(ArpPoisonRouting)whichenablessniffingonswitchedL ANsandMan-in-the-Middleattacks.ThesnifferinthisversioncanalsoexamineencryptedprotocolslikeSSH-1andHTTPS, and contains filters to capture credentials from a large range of authentication mechanisms.

V. READING DETECTION TECHNIQUES

It's not easy to find a sniffer on the net because it catches up and runs in promiscuous mode. Promiscuous mode allows a network device to intercept and read each network packet that arrives in its entirety. The sniffer leaves no trace since it does not transmit data. To find sniffers, check for systems that are running in promiscuous mode which is a NIC mode that allows all packets (traffic) to pass, without validating its destination address. Standalone sniffers are difficult to detect because they do not transmit data traffic. The reverse DNS lookup method helps to detect non-standalone sniffers. Many tools, such as Nmap, are available to detect promiscuous mode. Run IDS to see if the MAC address of some machines has changed (For example: MAC address of the router). IDS detects snooping activities on the network. Alerts or warns the administrator if suspicious activity, such as sniffing or MAC spoofing, occurs. Network tools such as Capsa Network Analyzer check the network for foreign packets, such as packets with spoofed addresses. This tool can collect, centralize, and analyze traffic data across a variety of network resources and technologies. The following are snooping detection techniques [14] [2] [1]: -

Sniffing test methods

Ping method

To identify the sniffer on the network, identify the network system running in promiscuous mode. The ping method is useful for identifying a system running in promiscuous mode, which in turn helps identify sniffers installed on the network.

Just send a ping request to the suspicious machine with its IP address and incorrect MAC address. Reject it from the adapter because the MAC address does not match, while the suspicious machine running the sniffer responds because it does not reject packets with a different MAC address. Therefore, this response identifies the network sniffer.

DNS method

Reverse DNS lookups are the opposite of DNS lookup methods. Sniffers that use reverse DNS lookups increase network traffic. This increase in network traffic may be a sign of the presence of a network sniffer.

Users can perform reverse DNS lookups remotely or locally. Check your organization's DNS server for future reverse DNS lookups. The way to send ICMP requests to the missing IP address can also be checked by a reverse DNS lookup. The computer performing the reverse DNS lookup responds to the ping, which means that it hosts the sniffer.

ARP method: -

This technique sends untransmitted ARPs to all nodes on the network. A node running in promiscuous mode on the network caches the local ARP address. It then sends a ping message to the network with a local IP address but a different MAC address. In this case, only the node with the MAC address (previously cached) can respond to your broadcast request. The promiscuous machine responds to the ping message because it has the correct information about the host that sent the ping request to its cache; Some computers send an ARP test to identify the source of the ping request. Detects the node where the sniffer is running.

Sniffing analysis tools

In addition to the above methods, there are several tools that can detect sniffing. These tools are listed below [1]: -

Anti-snuff

AntiSniff with network card promiscuous mode detector. It works by sending a series of carefully created packets to the target machine in a certain order, scanning the results and performing time tests against the target. By measuring the timing results and checking the responses of the network target, it is possible to determine whether the target is in promiscuous mode, ie sniffing the network. Network card detection in promiscuous mode is a good way to determine if your computer network has been compromised.

ARPWatch

ARWatch is a computer software tool for monitoring Address Resolution Protocol traffic on a computer network. Records the observed association of IP addresses with MAC ads along with a timestamp when the link appears on the network. It also has the ability to send an email to the administrator when a link is changed or added. Network administrators monitor ARP activity to detect ARP fraud [14].

Snort is a free and open source intrusion prevention and intrusion detection system created by Martin Roesch in 1998. Snort is now created by Sourcefire, of which Roesch is the founder and CTO. In 2009, Snort was inducted into the InfoWorld Open Source Hall of Fame as one of the most open source software of all time. Snort's open source network intrusion detection (NIDS) system has the ability to perform real-time traffic analysis and packet registration on Internet Protocol networks. Snort performs log analysis, content search, and content matching. These basic services serve several purposes, including knowledge of the quality of service triggered by the application, in order to prioritize large volumes of traffic when using latency-sensitive applications. The program can also be used to identify checks or attacks, including but not limited to operating system fingerprint tests, common gateway interfaces, buffer overflows, server message block tests, and stealthport scans. Snort can be configured in three main modes : sniffer, packet logger, and network intrusion detection [15].

VI. CONCLUSION

In this paper, some important packet sniffing tools that monitor and capture the traffic between legitimate users are discussed. Each tool has a different way of working and its own strengths. As there is a saying – “Prevention is better than cure”. So, some countermeasure to prevent sniffing are also discussed. If the main purpose of deploying sniffers is to intercept confidential information, such as passwords, then packet snooping is a serious issue for network security. Sniffers can be used in any environment, so the best practice is to send data in encrypted form. Users can also deploy several methods for detecting network sniffers and protecting data from sniffing, which will be discussed later in this document. Sniffer is called a network administrator's nightmare because in some situations it can be difficult to detect the presence of sniffer.

VII. REFERENCE

[1] S. Dhar, I. Feiligens, M. Team, ug R. Infocomm, "Sniffers Basics en Detection Information Security Management Team," Secur. Beear, 2007. [2]D. D.R.P. Nimisha P. Patel, Rajan G. Patel, "Packet Sniffing: Network Wiretapping Packet Sniffing: Network Wiretapping," Pack. Simhot. Netw. Odposlech, sv. 2, no. February, p. 6-7, 2009

[3] I. Kear, H. Kaur and E.G. Singh, "Analysis of Different Tools for Sniffing Packages," Int. J. electr. Electron. Count. De Sci. Eng., Vol. 1, no. 5, s. 65–69, 2014.

[4]Wireshark, "https://www.wireshark.org/docs/wsug_html_chunked/." [Online]. Anaa: https://www.wireshark.org/docs/wsug_html_chunked/.

[5]Zenmap, "http://nmap.org/book/zenmap.html." [Online]. Anaa: http://nmap.org/book/zenmap.html%0A.

[6] AngryIP, "http://angryip.org/." [Online]. Anaa: http://angryip.org/%0A.

[7] Kain, "https://web.archive.org/web/20190603235413if_/http://www.oxid.it/cain.html." [Online]. Anaa: https://web.archive.org/web/20190603235413if_/http://www.oxid.it/cain.html. [8] TCPdump, "TCPdump.org." [Online]. Anaa: https://www.tcpdump.org/.

[9] Kismet, "https://www.kismetwireless.net/." [Online]. Anaa: https://www.kismetwireless.net/.

[10] Ettercap, "https://www.ettercap-project.org/." [Online]. Anaa: https://www.ettercap-project.org/.

[11] Dsniff, "https://github.com/tecknicaltom/dsniff." [Online]. Anaa: https://github.com/tecknicaltom/dsniff.

[12] NetworkMiner, "https://www.netresec.com/." [Online]. Anaa: https://www.netresec.com/. [13]Capsa, "https://www.colasoft.com/capsa/." [Online]. Já: https://www.colasoft.com/capsa/.

[14]M. A. Kadeer, Mgr. Zahid, A. Iqbal and M. R. Siddiqui, "Network Traffic and Burglary Analysis d