International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.161.04

Packet Sniffing and Sniffing Detection

Ruchi Tuli
Department of Computer Science & Engineering, Jubail University College, Jubail, Kingdom of Saudi Arabia

Abstract - Packet sniffing is a process of monitoring and capturing all data packets passing thorough a given network using a software application or a hardware device. Sniffers can be used to monitor all sorts of traffic either protected or unprotected. Using sniffers, attacker can gain information which might be helpful for further attacks. This paper discusses the basic workingof a packet sniffer, network protocols that are vulnerable to sniffing, various software that can be used for sniffing. This paper also describes possible defensive techniques used to defend against sniffing attacks. Finally the papers ends with describing some sniffing detection techniques. Sniffers are not hacking tools but they can help a hacker to launch further attacks such as session hijacking, DOS attacks, MITM attacks etc.

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

I. INTRODUCTION

A sniffer is a program or a device that eavesdrops on the network traffic by grabbing information travelling over a network. Sniffers basically are "Data Interception" technology[1]. They work because Ethernet was built around a principle of sharing. Most networks use broadcast technology wherein messages for one computer can be read by another computer on that network. In practice, all the other computers except the one for which the message is meant, will ignore that message. However, computers can be made to accept messages even if they are not meant for them. This is done by means of a sniffer[1].

Using sniffing, the attacker can capture packets like Syslog traffic, DNS traffic, web traffic, Email and other types of data traffic. By capturing these packets, an attacker can reveal information such as data, username and passwords from protocols such as HTTP, POP, IMAP, SMTP, FTP and Telnet. The process of sniffing is performed by using Promiscuous ports. This paper discusses basic working of a packet sniffer, protocols that are vulnerable to sniffing, various types of tools used for sniffing, defensive techniques to defend against sniffing attacks and sniffing detection techniques[2].

II. WORKING OF SNIFFERS

In the process of sniffing, an attacker gets connected to the target network to sniff the packets. Using sniffers, which turns Network Interface Card (NIC) of the attacker's system into promiscuous mode, attacker captures the packet[3]. Once attacker captures the packet, it can decrypt these packets to extract the information. Sniffers can use used to hack a system or a network. The steps that an attacker follows to make use of sniffers to hack a network are listedbelow and shown in Figure 1:

a) An attacker who decides to hack a network first discovers the appropriate switch to access the network and connects a system to one of the ports on the switch.
b) After succeeding in connecting to the switch, attacker tries to determine network information such as network topology by using network discovery tools.
c) By analyzing the network topology, the attacker identifies the victim's machine to target the attacks.
d) After target identification, the attacker uses ARP spoofing techniques to send a fake (spoofed) ARP message
e) The previous step helps the attacker to divert all the traffic from the victim's computer to the attacker's computer. This is a man-in-the-middle (MITM) attack.
f) Now the attacker can see all the data packets sent and received by the victim and can extract the confidential information such as username, password, credit card details, PIN etc.

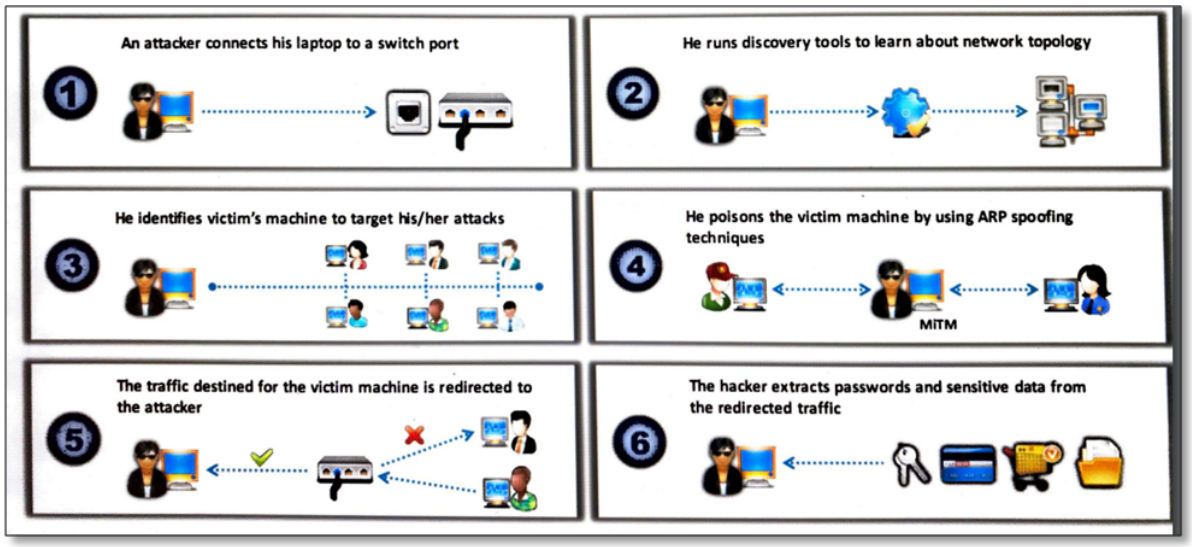

Figure 1: - network hacking using sniffer

III. PROTOCOLS VULNERABLE TO SNIFFING

The following network protocols are vulnerable to sniffing. The main reason for sniffing these protocols is to acquire confidential data like passwords.

3.1 Telnet and Rlogin

Telnet is a protocol used for communicating with a remote host (via port no. 23) on a network by using command line terminal. Rlogin enables an attacker to log into a network machine remotely via TCP connection. Both protocols fail to provide encryption. So the data traversing between the clients connected through any of these protocols is in plain text and vulnerable to sniff. Attackers can sniff keystrokes including usernames and passwords.

3.2 HTTP

Due to vulnerabilities in the default version of HTTP, websites implementing HTTP transfer user data across the network in plain text, which the attackers can read to steal user credentials

3.3 *SNMP*

SNMP is a TCP/IP based protocol used for exchanging management information between devices connected on a network. The first version of SNMP (SNMPv1) does not offer strong security, which leads to transfer of data in clear text format. Attackers exploit the vulnerabilities in this version in order to acquire passwords in plain text.

3.4 NNTP

Network News Transfer Protocol (NNTP) distributes, inquires, retrieves and posts news articles using a reliable stream-based transmission of news among the ARPA-Internet community. The protocol fails to encrypt the data which gives an attacker the opportunity to sniff sensitive information.

3.5 POP

The Post Office Protocol (POP) allows a user's workstation to access mail from a mailbox server. A user can send mail from workstation to the mailbox server via the Simple Mail Transfer Protocol (SMTP). Attackers can easily sniff the data flowing across a POP network in clear text because of the protocol's weak security implementations.

3.6 *FTP*

File Transfer Protocol (FTP) enables clients to share files between computers in a network. This protocol fails to provide encryption. So attackers sniff data as well as user credentials by running tools like Cain & Abel

3.7 IMAP

Internet Message Access Protocol (IMAP) allows a client to access and manipulate electronic mail messages on a server. This protocol offers inadequate security, which allows attackers to obtain data and user credentials in clear text.

IV. PACKET SNIFFING TOOLS

System administrators use automated tools to monitor their network but attackers misuse these tools to sniff network data. This section describes various packet sniffing tools/software

4.1 Wireshark

Wireshark lets you capture and interactively browse the traffic running on a computer network. This tool uses Winpcap to capture packets on its own supported networks. It captures live network traffic from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token ring, Frame Relay, FDDI networks. The captured files can be programmatically edited via command-line. A set of filters for customized data display can be refined using a display filter[4].

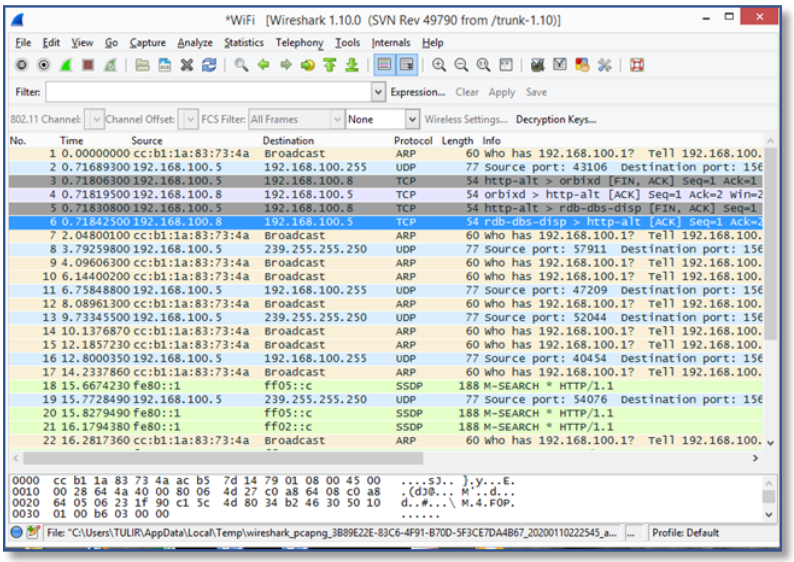

Figure 2 :- Wireshark

4.2 Zenmap

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. With Zenmap you can enable the features of Nmap to help you with: network inventory, managing service upgrade schedules, and monitoring host or service uptime [5]. Features comprise: Host discovery; port scanning; version detection;OS detection;scriptable interface;web scanning; full IPv6 support; Nping support; fast scanning;and much more.Zenmap is the official Nmap Security Scanner GUI. It is a multi-platform (Linux, Windows, Mac OS X, BSD,etc.) free and open source application which aims to make Nmap easy for beginners to use while providing advanced features for experienced Nmap users.

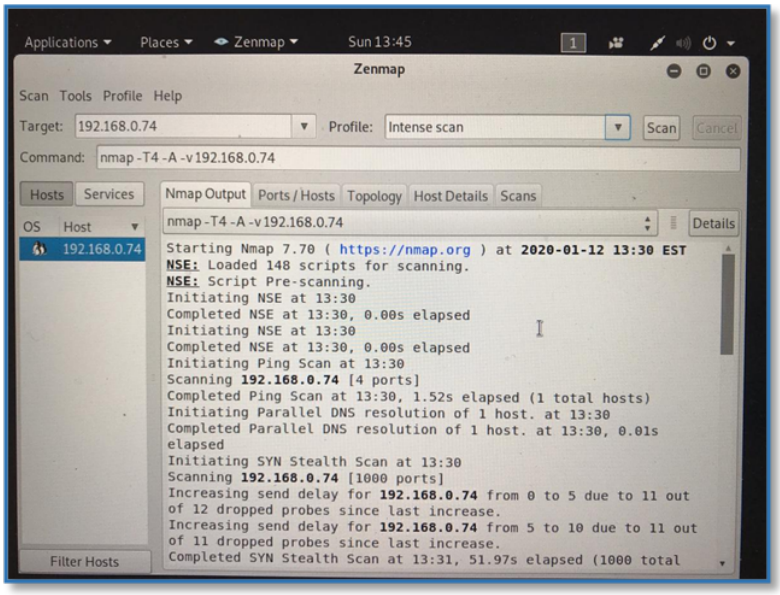

Figure 3:- Zenmap

4.3 AngryIPScanner

AngryIP Scanner [6] is an open source, snappy platform scanner that is designed to be incredibly fast and very easy to use. AngryIP deals the following features: Portable zero installation on certain platforms; ping checks; NetBIOS information; resolveshostnames; determines MAC address; can determine currently logged-in user; plug in system; scan results can be saved as CSV, TXT, XML, or IP-Portlist; and fast, multi-threaded scanning. AngryIP Scanner maintained by angryziber.

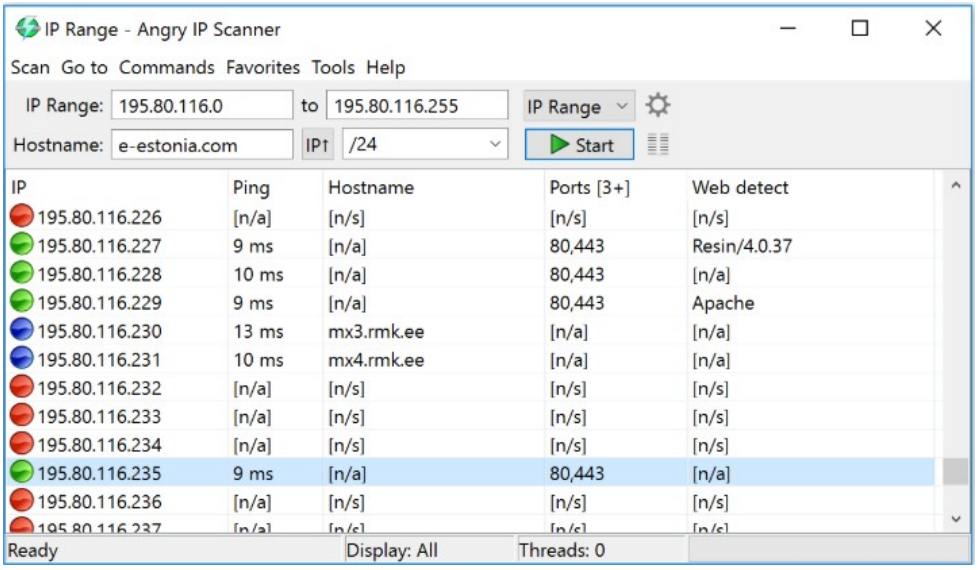

Figure 4 : - Angry IP Scanner

4.4 Cain & Abel

Cain&Abel is a password recovery tool for Microsoft Operating Systems. It permits simple recovery of different kinds of passwords by sniffing the network, cracking encrypted passwords with Dictionary, Brute-Force and Cryptanalysis attacks, recording VoIP conversations, decoding scrambled passwords,

recovering wireless network keys, revealing password boxes, uncovering cached passwords and analyzing routing protocols. The program does not utilize any software vulnerabilities or bugs that could not be fixed with little attempt. It covers some security aspects/weakness present in protocol's standards, authentication methods and caching mechanisms; its main purpose is the simplified recovery of passwords and credentials from various sources, however it also ships some "non-standard" utilities for Microsoft Windows users. Cain &Abel has been developed in the hope that it will be useful for network administrators, teachers, securityconsultants/professionals,forensicstaff,securitysoftwarevendors,professionalpenetrationtesterande veryoneelsethatplanstouseitforethicalreasons.Thelatestversion(Cain&Abelv4.9.56)[7]isfasterandcontainsa lotofnewfeatureslikeAPR(ArpPoisonRouting)whichenablessniffingonswitchedLANsandMan-in-the-Middleattacks.Thesnifferinthisversioncanalsoexamineencryptedprotocolslike SSH-1andHTTPS, and contains filters to capture credentials from a large range of authentication mechanisms.

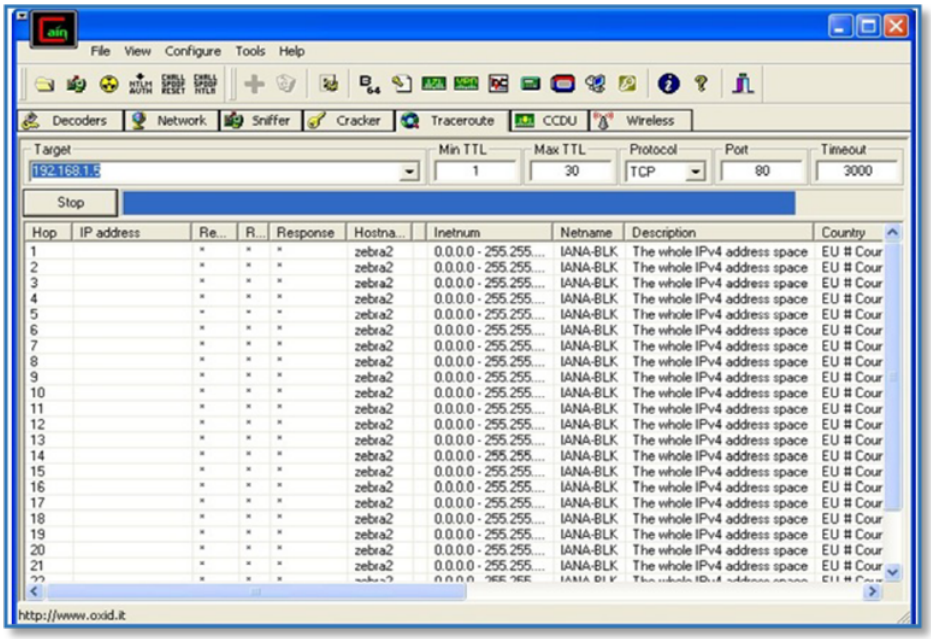

Figure 5 : - Cain & Abel

4.5 Tcpdump

Tcpdump is the network sniffer used before Wireshark came. It may not have the bells and whistles (like a GUI and logic for various application protocols) that Wireshark has, but it works well and with fewer security risks. It also requires fewer system resources. While Tcpdump doesn't receive new features often, it is actively maintained to fix bugs and portability problems. It is great fortrackingdownnetworkproblemsormonitoringactivity.ThereisaseparateWindowsportnamedWinDump.T cpdumpisthesourceoftheLibpcap/WinPcappacketcapturelibrary,whichisusedbyNmap andmanyothertools[8].

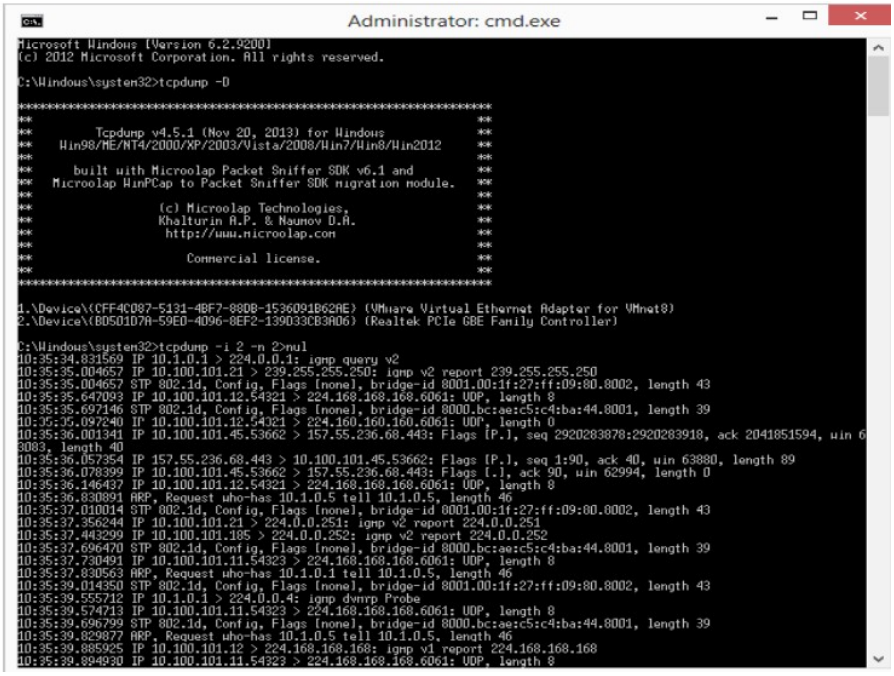

Figure 6 :- TCPdump

4.6 Kismet

Kismetisan802.11layer2wirelessnetworkdetector,sniffer,andintrusiondetectionsystem.Kismetwillworkwit hanywirelesscardwhichsupportsrawmonitoring(RFMON)mode,and(withappropriatehardware)cansniff802 .11b,802.11a,802.11g,and802.11ntraffic.Kismetalsosupportspluginswhichallowsniffingothermediasuchas DECT.Kismetidentifiesnetworksbypassivelycollectingpacketsanddetectingstandardnamednetworks,detect inghiddennetworks,andinferringthepresenceofnon-beaconingnetworksviadatatraffic. InSep25,2013 ReleasedthefirstversionofSmarterWi-FiManagerforAndroid [9].

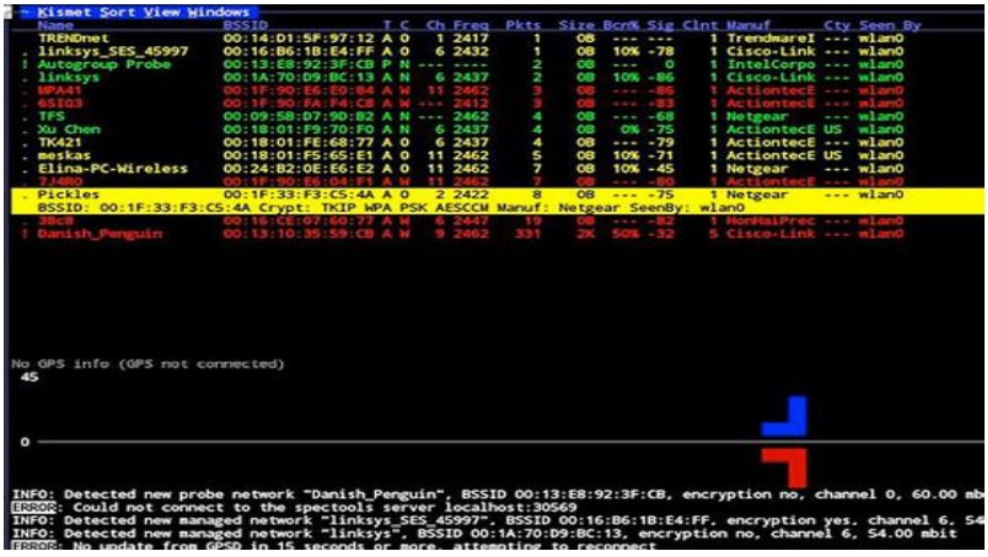

Figure 7 :- Kismet

4.7 Ettercap :-

Ettercap is a complete suite for man in the middle attacks. It features sniffing of live connections, content filtering on the fly and many other interesting tricks. It supports active and passive dissection of many protocols and includes many features for network and host analysis [10].

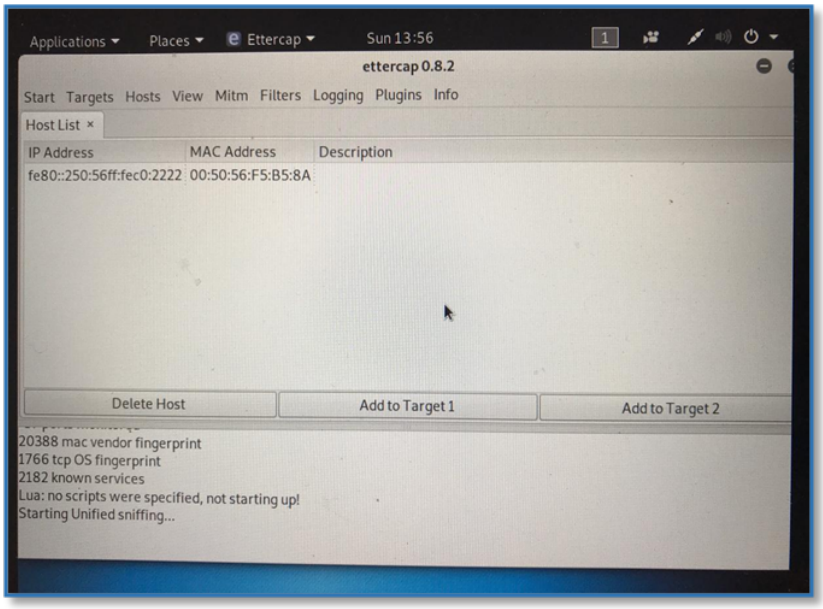

Figure 8 :- Ettercap

4.8 Dsniff

This is well-known and well-designed suite which includes many tools : dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.); arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). The suite suffers from the lack of any updates in the last decade, but it is still a great toolset for handling your password sniffing needs [11].

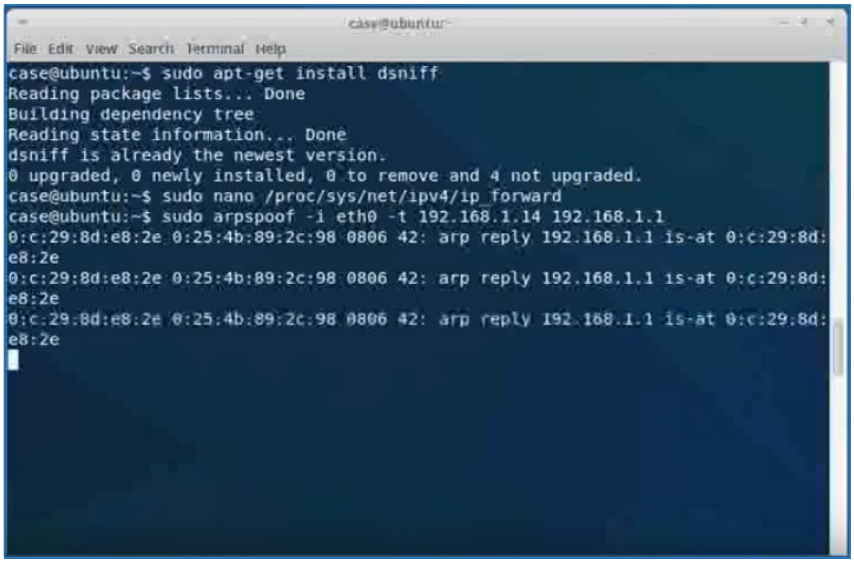

Figure 9 :- Dsniff

4.9 NetworkMiner

NetworkMiner is a Network Forensic Analysis Tool (NFAT) for Windows (but also works in Linux/MacOSX/FreeBSD). NetworkMiner can be used as a passive network sniffer/packetcapturingtoolinordertodetectoperatingsystems,sessions,hostnames,openportsetc.withoutputtinganytrafficonthenetwork.NetworkMinercanalsoparsePCAPfilesforoff-lineanalysis and to regenerate/reassemble transmitted files and certificates from PCAP files.NetworkMinercollectsdatalikeforensicevidenceabouthostsonthenetworkratherthantocollectdataregardingthetrafficonthenetwork.Themainuserinterfaceviewishostcentrici.e.informationgroupedperhostratherthanpacketcentrici.e.informationshowedasalistofpackets/frames.NetworkMinerhas,sincethefirstreleasein2007,becomeapopulartoolamongincidentresponseteamsaswellaslawenforcement.NetworkMineristodayusedbycompaniesandorganizationsallovertheworld[12].

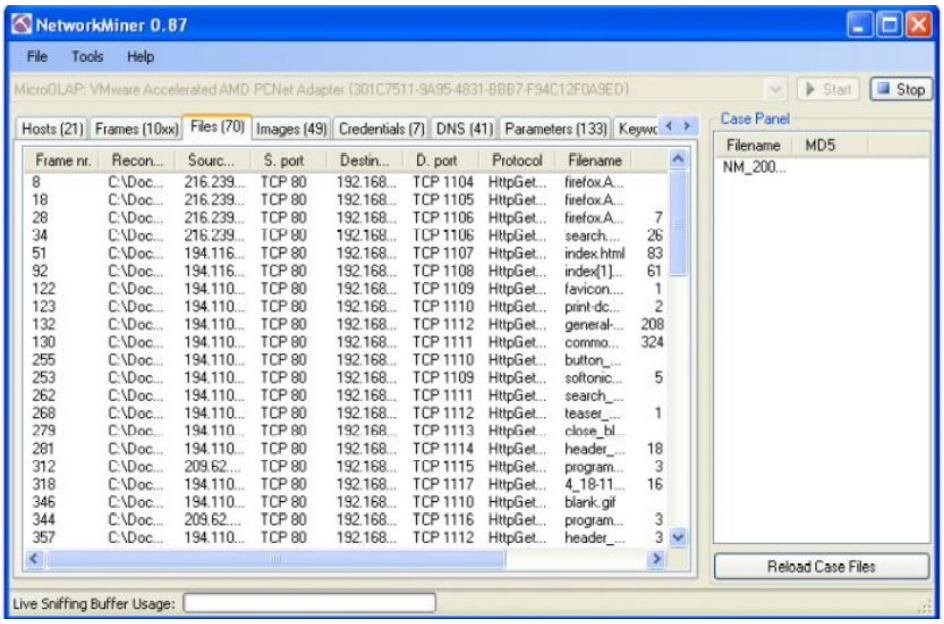

Figure 10 :- NetworkMiner

4.10Capsa Network Analyzer

Capsa network analyzer is a network-monitoring tool that captures all the data transmitted over the network and provides a wide range of analysis statistics in an intuitive and graphic way. The tool helps to analyze and troubleshoot the problem that has occurred (if any) in the network. It is also able to perform reliable network forensics, advanced protocol analyzing, in-depth packet decoding and automatic expert diagnosing. It helps to detect network vulnerabilities. An attacker can use this tool to sniff packets from the target network[13].

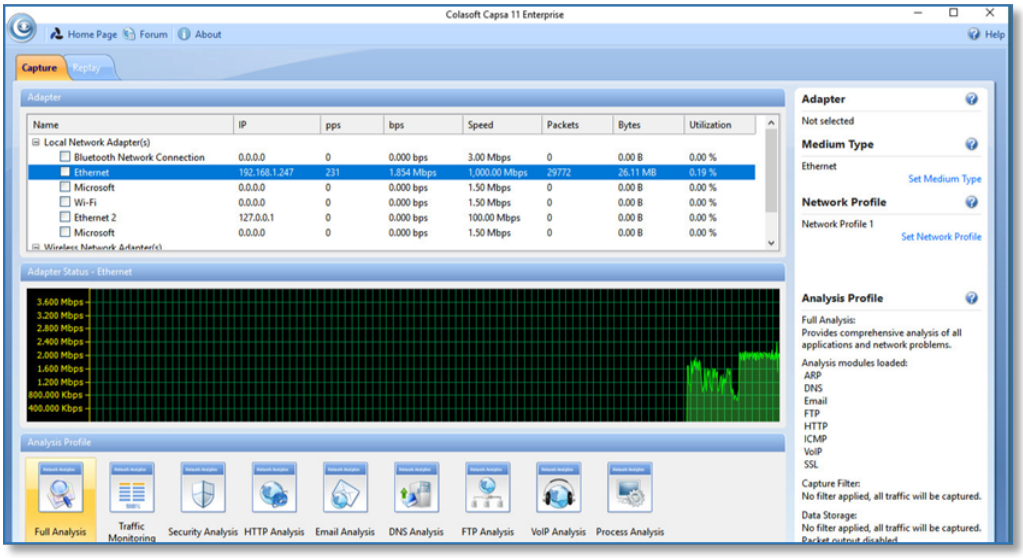

Figure 11 :- Capsa Network Analyzer

V.DEFENSIVE TECHNIQUES

This section describes countermeasures and possible defensive techniques that can be used to defend a target network against sniffing attacks. Listed below are some of the countermeasures that can be followed to defend against sniffing :-

- Restrict the physical access to the network media to ensure that a packet sniffer cannot be installed.
- Use end-to-end encryption to protect confidential information.
- Permanently add the MAC address of the gateway to the ARP cache.
- Use static IP addresses and ARP tables to prevent attackers from adding the spoofed ARP entries for machines in the network.
- Turn off network identification broadcasts and if possible restrict the network to authorized users in order to protect the network from being discovered with sniffing tools.
- Use IPv6 instead of IPv4 protocol
- Use encrypted sessions such as SSH instead of Telnet, SecureCopy (SCP) instead of FTP, SSL for email connection, etc. to protect wireless network users against sniffing attacks.
- Use HTTPS instead of HTTP to protect usernames and passwords.
- Use switch instead of the hub as switch delivers data only to the intended recipient.
- Use Secure File Transfer Protocol (SFTP) instead of FTP for secure transfer of files.
- Use PGP and S/MIME, VPN, IPSec, SSL/TLS, SecureShell (SSH) and one time passwords (OTP).
- Always encrypt the wireless traffic with a strong encryption protocol such as WPA and WPA2.
- Retrieve MAC directly from NIC instead of OS; this prevents MAC address spoofing.
- Use tools to determine if any NICs are running in the promiscuous mode.
- Use a concept of ACL (Access Control List) to allow access to only a fixed range of trusted IP addresses in a network.
- Change default passwords to complex passwords.
- Avoid broadcasting SSID (Session Set Identifier).
- Implement MAC filtering mechanism on your router.

VI. SNIFFER DETECTION TECHNIQUES

It is not easy to detect a sniffer on a network as it only captures 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. There are many tools such as Nmap that are available to use for the detection of promiscuous mode. Run IDS and notice if the MAC address of certain machines has changed (Example : router's MAC address). IDS can detect sniffing activities on a network. It notifies or alerts the administrator when a suspicious activity such as sniffing or MAC spoofing occurs. Network tools such as Capsa Network Analyzer monitors the network for strange packets such as packets with spoofed addresses. This tool can collect, consolidate centralize and analyze traffic data across different network resources and technologies. Following are the techniques to detect sniffing[14][2][1]:-

6.1 Sniffing Detection Methods

6.1.1 Ping Method

To detect a sniffer on a network, identify the system on the network running in promiscuous mode. The ping method is useful in detecting a system that runs in promiscuous mode, which in turns helps to detect sniffers installed on the network.

Just send a ping request to the suspected machine with its IP address and incorrect MAC address. The adapter will reject it as the MAC address does not match, whereas the suspect machine running the sniffer responds to it, as it does not reject packets with a different MAC address. Thus, this response will identify the sniffer in the network.

6.1.2 DNS Method

The reverse DNS lookup is the opposite of the DNS lookup method. Sniffers using reverse DNS lookup increase network traffic. This increase in network traffic can be an indication of the presence of a sniffer on the network.

Users can perform a reverse DNS lookup remotely or locally. Monitor the organization's DNS server to identify incoming reverse DNS lookups. The method sending ICMP requests to a non-existing IP address can also monitor reverse DNS lookups. The computer performing the reverse DNS lookup would respond to the ping, thus identifying it as hosting a sniffer.

6.1.3 ARP Method :-

This technique sends a non-broadcast ARP to all the nodes in the network. The node that runs in promiscuous mode on the network will cache the local ARP address. Then it will broadcast a ping message on the network with the local IP address but a different MAC address. In this case, only the node that has the MAC address (cached earlier) will be able to respond to your broadcast ping request. A machine in promiscuous mode replies to the ping message as it has correct information about the host that is sending ping request in its cache; rest of the machines will send ARP probe to identify the source of the ping request. This will detect the node on which the sniffer is running.

6.2 Sniffing Detection Tools

Apart from the methods discussed above, there are few available tools which can also detect sniffing. These tools are listed below[1] :-

6.2.1 Anti-sniff

AntiSniff is network card promiscuous mode detector. It works by sending a series of carefully crafted packets in a certain order to a target machine, sniffing the results, and performing timing tests against the target. By measuring timing results and monitoring the target's responses on the network, it can be determined if the target is in promiscuous mode, i.e. sniffing the network. Detecting a network card in promiscuous mode is a good way to determine if your computer network has been compromised.

6.2.2 ARPWatch

ARWatch is a computer software tool for monitoring Address Resolution Protocol traffic on a computer network. It generates a log of observed pairing of IP addresses with MAC addresses along with a time stamp when the pairing appeared on the network. It also has the option of sending an email to an administrator when a pairing changes or is added. Network administrators monitor ARP activity to detect ARP spoofing [14].

6.2.3 Snort :-

Snort is a free and open source network intrusion prevention system and network intrusion detection system created by Martin Roesch in 1998. Snort is now developed by Sourcefire, of which Roesch is the founder and CTO. In 2009, Snort entered InfoWorld's Open Source Hall of Fame as one of the open source software of all time. Snort's open source network-based intrusion detection system (NIDS) has the ability to perform real time traffic analysis and packet logging on Internet Protocol networks. Snort performs protocol analysis, content searching, and content matching. These basic services have many purposes including application-aware triggered quality of service, to de-prioritize bulk traffic when latency sensitive applications are in use. The program can also be used to detect probes or attacks, including, but not limited to, operating system finger printing attempts, common gateway interface, buffer overflows, server message block probes, and stealth port scans. Snort can be configured in three main modes : sniffer, packet logger, and network intrusion detection [15].

VII. 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. As the main aim of deploying sniffers is to capture the confidential information such as passwords, so the packet sniffing is a serious matter for network security. Sniffers can be deployed in any environment, so the best practice is to send the data in an encrypted form. Users can also deploy a number of techniques to detect the sniffers on the network and protect the data from sniffing, which has been discussed in the latter part of this paper. Sniffers are called network administrator's nightmare as it may be difficult in certain situations to detect the presence of sniffers.

REFERENCES

[1] S. Dhar, I. Security, M. Team, and R. Infocomm, "Sniffers Basics and Detection Information Security Management Team," *Secur. Manag.*, 2007.
[2] D. D. R. P. Nimisha P. Patel, Rajan G. Patel, "Packet Sniffing : Network Wiretapping Packet Sniffing : Network Wiretapping," *Pack. Sniff. Netw. Wiretapping*, vol. 2, no. February, pp. 6–7, 2009.
[3] I. Kaur, H. Kaur, and E. G. Singh, "Analysing Various Packet Sniffing Tools," *Int. J. Electr. Electron. Comput. Sci. Eng.*, vol. 1, no. 5, pp. 65–69, 2014.
[4] Wireshark, "https://www.wireshark.org/docs/wsug_html_chunked/." [Online]. Available: https://www.wireshark.org/docs/wsug_html_chunked/.
[5] Zenmap, "http://nmap.org/book/zenmap.html." [Online]. Available: http://nmap.org/book/zenmap.html%0A.
[6] AngryIP, "http://angryip.org/." [Online]. Available: http://angryip.org/%0A.
[7] Cain, "https://web.archive.org/web/20190603235413if_/http://www.oxid.it/cain.html." [Online]. Available: https://web.archive.org/web/20190603235413if_/http://www.oxid.it/cain.html.
[8] TCPdump, "TCPdump.org." [Online]. Available: https://www.tcpdump.org/.
[9] Kismet, "https://www.kismetwireless.net/." [Online]. Available: https://www.kismetwireless.net/.
[10] Ettercap, "https://www.ettercap-project.org/." [Online]. Available: https://www.ettercap-project.org/.
[11] Dsniff, "https://github.com/tecknicaltom/dsniff." [Online]. Available: https://github.com/tecknicaltom/dsniff.
[12] NetworkMiner, "https://www.netresec.com/." [Online]. Available: https://www.netresec.com/.
[13] Capsa, "https://www.colasoft.com/capsa/." [Online]. Available: https://www.colasoft.com/capsa/.
[14] M. A. Qadeer, M. Zahid, A. Iqbal, and M. R. Siddiqui, "Network traffic analysis and intrusion detection using packet sniffer," *2nd Int. Conf. Commun. Softw. Networks, ICCSN 2010*, pp. 313–317, 2010.