Ethical Hacking: Network Protocol Analyzer Tool Comparison

Melvin Vélez Villegas
Master in Computer Science
Alfredo Cruz, Ph.D.
Electrical & Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract *— The computer security industry spoke about the term "ethical hacking" to describe a hacker with authorization to, without any bad intentions, attacks a network or other security system – whether private or public – on behalf of its owners. Ethical hackers are also called white hat hackers, and they are distinguished from the black hat hackers. They use a variety of tools and apply them efficiently to discover any vulnerability in the system. This is when a Network* ***Packet Analyzer*** *come into place and is used to monitor, intercept, and decode data packets as they are transmitted across networks. Packet analyzers can be computer programs either software or hardware. Some common alternative names for packet analyzers include* ***packet sniffers****,* ***protocol analyzers****, and* ***network analyzers****. We describe different Network Packet Analyzer tools, how they work and examine which are more susceptible to be detected when analyzed.*

Key Terms *— MITM Attack, Network Analyzer, Packet Sniffers, Protocol Analyzers.*

INTRODUCTION

Network analysis has to do with traffic analysis, protocol analysis, sniffing, packet analysis, eavesdropping, and so on. Is the process of capturing network traffic and taking the time to inspecting and analyze it closely to determine what is really happening on the network.

A **protocol analyzer** decodes – make sense of – the data packets of common protocols and displays the network traffic in a format that is easy to read and analyzed. A **packet sniffer** is a program that monitors the data traveling over a copper wired or Wi-Fi network. Unauthorized sniffers are dangerous to network security because they are difficult to detect and can be inserted almost anywhere, which makes them a favorite weapon of hackers.

A **network analyzer** can be a standalone hardware device with specialized software, or software that is installed on a desktop or laptop computer. The differences between network analyzers depend on features such as the number of supported protocols it can decode, the user interface, and its graphing and statistical capabilities. Other differences include inference capabilities like for example, expert analysis features and the quality of packet decodes. Although several **network analyzers** decode the same protocols, some will work better than others for your environment [1].

WHAT IS NETWORK ANALYSIS?

A computer network is a collection of connected computers. Two or more computer systems are considered as connected if they can send and receive data from each other.

Network analysis happens when network data is capture and decode. Most **network analyzers** can be hardware or software, and are available both free and commercially. A lot of network analyzers interfaces usually have three panes [2] that are as follows in Table 1.

Table 1
Network Analyzers Interfaces Panes

Summary	Detail	Data
Top Pane	*Middle Pane*	*Bottom Pane*
Capture Packets	Logical Breakout of Packets	Packets in Character form
Shows Fields	Selected Packets	Hexadecimal
Summary of Packets		ASCII

WHO USES NETWORK ANALYSIS?

Most administrators use network analysis to troubleshoot their network problems. The bottom

line is to analyze the performance of the network, and to detect any intrusion or disruption in the private network.

When an intruder uses network sniffers is usually to perform a passive attack. In this passive attack, the intruder captures user names and passwords, and also, collects confidential data, and map the network design.

Another use of sniffers for intruders is to create components for a rootkit and to control backdoor program access [3]. As you can see, network sniffers can be use either for malicious intents or for the better good.

HOW DOES IT WORK?

Ethernet is a shared medium that uses MAC or hardware addresses connected to a logical abstract model called the OSI model. This model has seven layers and each layer represents a standard for network communications.

A **network analyzer** is very useful for sniffing packets on a wire cable. A network analyzer is simply software that is running on a computer with a network card. It works by placing the network card in promiscuous mode, which enables the card to see all the traffic on the network, even traffic not destined for the network analyzer's host [4]. It has many functions that can be performed relatively easy and has many advantages such as the ones we can see in Table 2.

Table 2
Network Analyzer Performs the Following Functions

Network Analyzer Functions
It captures all network traffic
Interprets or decodes what is found into a human-readable format
Displays the content in chronological order
View anomalous network traffic and even track down an intruder
Develop a baseline of network activity and performance
Track and isolate malicious network usage
Detect malicious Trojan horse applications
Monitors and track down DoS attacks

NETWORK ANALYZER DRAWBACKS

There are always some drawbacks when we use a network analyzer to scan and protect our network. In order to capture all traffic, you must connect the analyzer to one of the following:

- A hub device on the network.
- A monitor/span/mirror port on a switch.
- A switch that you've performed an ARP poisoning attack on.

If you want to see real traffic similar to what a network sees, you should connect the network analyzer to a hub or switch monitor port or even a network tap on the outside of the firewall [5], as shown in Figure 1. In this way, your testing enables you to view some of the following:

- What's coming in your network before the firewall filters and eliminate the junk traffic.
- What's leaving your network after the traffic passes through the firewall.

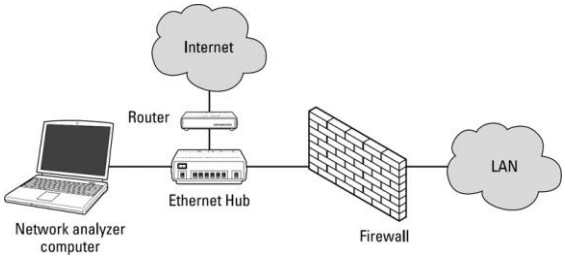


Figure 1
Connecting a Network Analyzer outside the Firewall

It doesn't matter really where you ultimately connect your network analyzer, whether inside or outside your firewall, you will see immediate results. This can be an overwhelming amount of information at once, but always try to look for these traits first:

Always look for Odd traffic first:

- Unusual amount of ICMP packets.
- Excessive amounts of multicast or broadcast traffic.
- Protocols that aren't permitted by policy or shouldn't exist given your current network configuration.
- Internet usage habits
- Web surfing and social media.
- E-mail.

- Instant messaging or P2P software.

Second, look for Questionable usage:

- Many lost or oversized packets, indicating hacking tools or malware are present.
- High bandwidth consumption that might point to a web or FTP server that doesn't belong.
- Significant amount of inbound traffic from unknown hosts, like FTP or telnet.
- Tons of inbound UDP or ICMP echo requests, SYN floods, or excessive broadcasts.
- Nonstandard hostnames on your network.
- Hidden servers especially web, SMTP, FTP, DNS, and DHCP.

You will need to let your network analyzer run for quite a while. It could be several hours to several days, depending on what you're looking for. Before getting started, configure your network analyzer to capture and store the most relevant data.

You can easily fill hundreds of gigabytes worth of data that is hard drive space in a short period. It is recommended to run the network analyzer in what OmniPeek calls monitor mode. This allows the analyzer to keep track of what's going on but not capture and store every single packet, which is very convenient. Monitor mode — if supported by the analyzer — is very beneficial and is often all we need.

It is always recommended to run a baseline when your network is working normally. When you have a baseline, you can see any obvious abnormalities when an attack occurs. Make sure you have permission to use a sniffer on a network that is not your own [6].

PURPOSE OF THE PROJECT

We worked with three labs and we applied each tool used in this project.

The first one has to do with capturing packets that comes from a connection made with a Dynamic Host Configuration Protocol (DHCP) server leasing out IPs for a specific time frame. With administrative access to the DHCP server, we will make some changes to the configuration, such as changing the least time to have different results.

The second lab has to do also with capturing packets from within a SSL session with a particular connection to a secure site by beginning a process of purchasing an item. In this lab, we'll investigate the Secure Sockets Layer (SSL) protocol, focusing on the SSL records sent over a TCP connection. We'll do so by analyzing a trace of the SSL records sent between your host and an e-commerce server. We'll investigate the various SSL record types as well as the fields in the SSL messages.

The third lab has to do with the MITM (man in the middle) Attack. This is to prove that an attacker or intruder can put himself in the logical way between two computers speaking together. Once in this position, the intruder can launch a lot of different and very dangerous attacks because he is in the way between the two normal computers.

We will be using for this third lab tools like Ettercap and Cain & Able. Both tools can be used as a console application and Ettercap both user interface and console.

What we are trying to accomplish in this project is to test to see the packet that an IP segment is produced during a session of communication utilizing this various tools as the main packet analyzing programs. Capturing the data that goes through from one end to another will let us know is the communication was successful or not, secure or not, and safe.

Although, these labs can be accomplished in a contained environment or a test environment, does not guarantee they will work on an open network like the Internet.

NETWORK ANALYZER TOOLS

There were six tools in total, chosen randomly, and tested for speed of execution, detectability and batch or filter capability.

Wireshark

Formerly known as Ethereal, is a free alternative. Wireshark is a feature-rich **network**

analyzer that rivals commercial counterparts. It can decode more than 750 protocols and is compatible with more than 25 other sniffers and capture utilities. It's not as user-friendly as most of the commercial products, but it is very powerful if you're willing to learn its ins and outs. Wireshark display and capture filters can be used to sort through network traffic. Wireshark is available for both Windows and OS X. It is free to distribute and you are free to modify it [7].

OmniPeek

OmniPeek is one of the favorite network analyzer out in the market today. It does probably everything you will ever need and more and is very simple to use. OmniPeek is available from WildPackets for the Windows operating systems. As a portable analyzer, OmniPeek offers an intuitive, easy-to-use graphical interface that engineers can use to rapidly analyze and troubleshoot enterprise networks. OmniPeek supports local captures from multiple interfaces and data collection from any network topology, including 10 Gigabit and Gigabit networks, wireless networks, and local matrix switches [8]. OmniPeek is not free, certainly not cheap, and because of that small business will venture to other analyzers that are either cheaper or free.

CommView

CommView is a low-cost, Windows-based alternative. CommView is also available from Tamosoft. Network security administrators require advanced software tools to capture and analyze both sent and received packets. CommView is an intuitive application that serves this exact purpose, enabling users to monitor the network traffic. Within its user-friendly and well-organized window, you can view a list of the active network connections and network statistical data [9]. Combining ease of use with an advanced feature set, CommView provides a complete set of tools for monitoring the traffic, analyzing the content of the transferred data and viewing network statistics.

Cain & Abel

Is a free multifunctional password recovery tool for performing ARP poisoning, capturing packets, cracking passwords, and much more [10]. Cain & Abel is for Microsoft Operating Systems. Cain & Abel has been developed in the hope that it will be useful for network administrators, teachers, security consultants/professionals, forensic staff, security software vendors, professional penetration tester and everyone else that plans to use it for ethical reasons. New features like APR (Arp Poison Routing) which enables sniffing on switched LANs and Man-in-the-Middle attacks. The sniffer can also analyze encrypted protocols such as SSH-1 and HTTPS, and contains filters to capture credentials from a wide range of authentication mechanisms.

Ettercap

It is another powerful and free utility for performing network analysis on Windows, Linux, and other Operating Systems as well. Ettercap is a tool made by Alberto Ornaghi (ALoR) and Marco Valleri (NaGA) and is basically a suite for man in the middle attacks on a LAN. For those who do not like the Command like Interface (CLI), it is provided with an easy graphical interface.

Ettercap is able to perform attacks against the ARP protocol by positioning itself as "man in the middle" and, once positioned as this, it is able to infect, replace, delete data in a connection, discover passwords for protocols such as FTP, HTTP, POP, SSH1, etc, and provide fake SSL certificates in HTTPS sections to the victims. Plugins are also available for attacks such as DNS spoofing [11].

Microsoft Message Analyzer

Microsoft Message Analyzer is a new and free tool for capturing, displaying, and analyzing protocol messaging traffic and other system messages. Message Analyzer also enables you to import, aggregate, and analyzes data from log and trace files. With Message Analyzer, you can choose to capture data live or load archived message collections from multiple data sources as well as the ability to been able to gather the information

simultaneously. Message Analyzer enables you to display trace, log, and other message data in numerous data viewer formats, including a default tree grid view and other selectable graphical views that employ grids, charts, and timeline visualizer components which provide high-level data summaries and other statistics. It also enables you to configure your own custom data viewers [12].

In Table 3 we have a comparison of the network tools that were briefly describe alone with their creator, if they are graphical friendly or console only.

Table 3
Packet Analyzers Comparison

Name	Creator	GUI/ Console	License
Cain & Abel	Massimiliano Montoro	GUI	Proprietary
CommView	TamoSoft	GUI	Proprietary
Ettercap	ALoR & NaGA	Both	GNU
Microsoft Message Analyzer	Microsoft	GUI	Proprietary
OmniPeek	WildPackets	GUI	Proprietary
Wireshark	Wireshark Team	Both	GNU

TOOLS & SOURCES OF INFORMATION

The tools that we used were already mentioned alone with their tools, we used a computer running Windows 7 with Internet Explorer 11, VirtualBox and VMWare to create a virtual lab for testing. Also we used the actual wireless connection, and as an alternative, a network card 10/100/1G Ethernet. Within Windows, we performed several DHCP-related commands and capture the DHCP messages exchanged as a result of executing these commands. The command that we executed was “ipconfig” and two switches such as “/release” (to release the current IP lease) and “/renew” (to renew a new IP lease).

Also, we did start a session within Internet Explorer to capture packets from a secure web site using all different tools to model its behavior. After capturing the packets, we set a filter to display only the Ethernet frames that contain SSL records sent from and received by our host.

The goal of our tool comparison was to provide warning about the danger of "man in the middle" attacks by ARP spoofing. We explained how to configure the MITM Proxy and Ettercap computer as "man in the middle", and we showed some attacks. Finally, some countermeasures were given to fight against these types of very dangerous ARP poisoning attacks.

EXPERIMENTAL WORK

In order to evaluate each of the **Network Analyzer** tools, a particular home network environment was created and was exploited by chosen with different scenarios with different transitions of data across the wired network. The tests chosen were a very important component in benchmarking network analysis techniques. At first the tests implemented were conducted on DHCP related transactions from the Internet while sending the following messages:

DHCP
(Dynamic Host Configuration Protocol)

In this lab, we took a quick look at DHCP. DHCP is use to dynamically assign IP addresses to hosts and also to configure other network complex configuration information.

DHCP Experiment

In order to observe DHCP in action, we performed several DHCP-related commands and capture the DHCP messages exchanged as a result of executing these commands.

Steps to Run the Experiment

We open the windows command prompt and enter “ipconfig/release”. This command releases the IP address, so that the host’s IP address becomes 0.0.0.0. See Figure 2.

Next, start the tool to begin capturing. From the command prompt enter “ipconfig/renew”. This instructs your network host to obtain a network configuration, including a new IP address. See Figure 3.

Wait until the "*ipconfig/renew*" has terminated. Then enter the same command "*ipconfig /renew*" again. When you run the second *"ipconfig/renew"* and then terminates, enter the command "ipconfig/release" to release the previously-allocated IP address to your computer.

Finally, enter "*ipconfig/renew*" to again be allocated an IP address for your computer. Stop the tool packet capture.

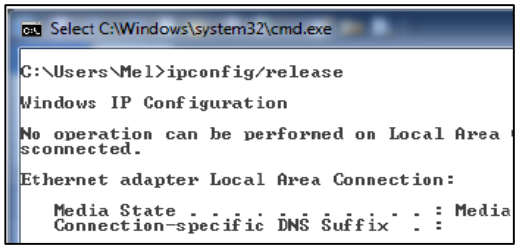


Figure 2
Ipconfig/Release Command Executed

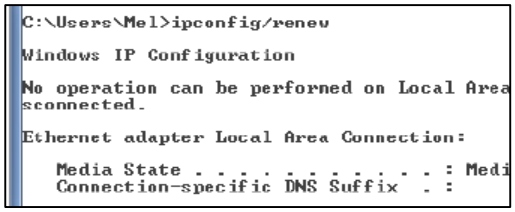
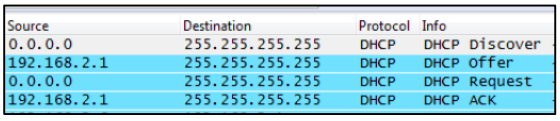


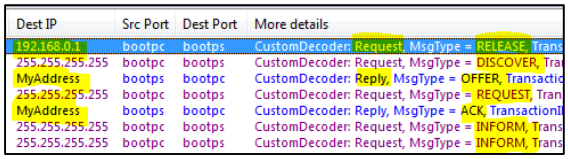
Figure 3
Ipconfig/Renew Command Executed

Now let's take a look at the resulting window screen shots from the network analyzer tools. To see only the DHCP packets, enter into the filter field "bootp" or "dhcp" depending on the tool used. (DHCP derives from an older protocol called BOOTP. Both BOOTP and DHCP use the same port numbers, 67 and 68. To see DHCP packets in the current version of Wireshark, you need to enter "bootp" and not "dhcp" in the filter). We see from Figure 4, 5, 6 and 7 that the first *ipconfig* renew command caused four DHCP packets to be generated: a DHCP Discover packet, a DHCP Offer packet, a DHCP Request packet, and a DHCP ACK packet.



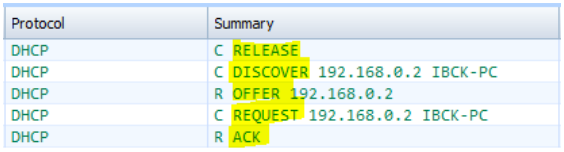
Source	Destination	Protocol	Info
0.0.0.0	255.255.255.255	DHCP	DHCP Discover
192.168.2.1	255.255.255.255	DHCP	DHCP Offer
0.0.0.0	255.255.255.255	DHCP	DHCP Request
192.168.2.1	255.255.255.255	DHCP	DHCP ACK

Figure 4
Wireshark DHCP Capture



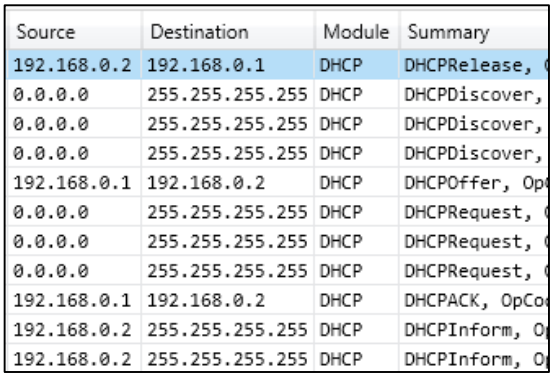
Dest IP	Src Port	Dest Port	More details
192.168.0.1	bootpc	bootps	CustomDecoder: Request, MsgType = RELEASE, Trans
255.255.255.255	bootpc	bootps	CustomDecoder: Request, MsgType = DISCOVER, Tra
MyAddress	bootps	bootpc	CustomDecoder: Reply, MsgType = OFFER, Transactio
255.255.255.255	bootpc	bootps	CustomDecoder: Request, MsgType = REQUEST, Tran
MyAddress	bootps	bootpc	CustomDecoder: Reply, MsgType = ACK, TransactionI
255.255.255.255	bootpc	bootps	CustomDecoder: Request, MsgType = INFORM, Trans
255.255.255.255	bootpc	bootps	CustomDecoder: Request, MsgType = INFORM, Trans

Figure 5
Commview DHCP Capture



Protocol	Summary
DHCP	C RELEASE
DHCP	C DISCOVER 192.168.0.2 IBCK-PC
DHCP	R OFFER 192.168.0.2
DHCP	C REQUEST 192.168.0.2 IBCK-PC
DHCP	R ACK

Figure 6
OmniPeek DHCP Capture



Source	Destination	Module	Summary
192.168.0.2	192.168.0.1	DHCP	DHCPRelease,
0.0.0.0	255.255.255.255	DHCP	DHCPDiscover,
0.0.0.0	255.255.255.255	DHCP	DHCPDiscover,
0.0.0.0	255.255.255.255	DHCP	DHCPDiscover,
192.168.0.1	192.168.0.2	DHCP	DHCPOffer, Op
0.0.0.0	255.255.255.255	DHCP	DHCPRequest,
0.0.0.0	255.255.255.255	DHCP	DHCPRequest,
0.0.0.0	255.255.255.255	DHCP	DHCPRequest,
192.168.0.1	192.168.0.2	DHCP	DHCPACK, OpCo
192.168.0.2	255.255.255.255	DHCP	DHCPInform, O
192.168.0.2	255.255.255.255	DHCP	DHCPInform, O

Figure 7
Microsoft Message Analyzer DHCP Capture

SSL (SECURE SOCKET LAYER)

In this lab, we investigated the Secure Socket Layer (SSL) protocol, focusing on the SSL records sent over a TCP connection. We did so by analyzing a trace of the SSL records sent between a host and an e-commerce server. We investigated the various SSL record types as well as the fields in the SSL messages.

Once again all **packet sniffing** tools were tested in the lab to reach a conclusion of how each tool operates and we saw the similarities and differences.

We started the test with Wireshark to see how a particular SSL session is capture and analyzed. The following was a SSL session initiated on my computer using Internet Explorer 11 to get into the Amazon.com e-commerce website and accessing my private account.

All data was capture, in this case using Wireshark, see Figure 8, but we are going to emulate the exact same test with some of the tools because

not all of them have the capacity or were designed with the same functionality. Four of them had the capacity to capture and analyzed live data and those are the ones showed within Figure 9, 10 and 11. All six tools were tested to show how each one of them can easily capture and analyze the traffic that can generate a SSL session between the computer and the Internet.

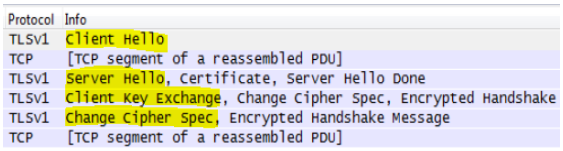


Figure 8
Wireshark SSL Capture

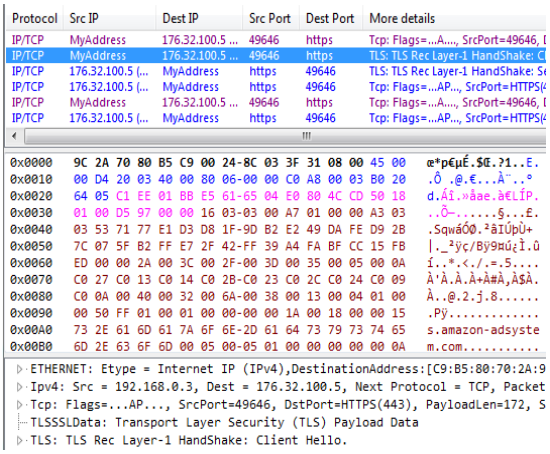


Figure 9
Commview SSL Capture

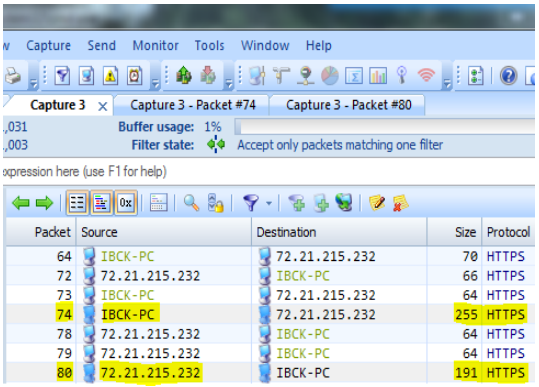


Figure 10
OmniPeek SSL Capture

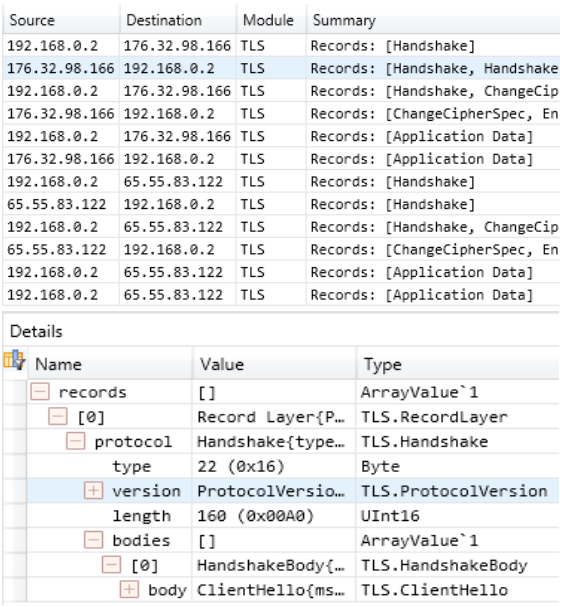


Figure 11
Microsoft Message Analyzer SSL Capture

MITM (Man In The Middle) Attacks

The **MITM Attack** name stands for Man-In-The-Middle and is a reference to the process we use to intercept and interfere with the data transfers in a private network. This is an attack in where a hacker put its computer in the logical way between two computers speaking together as shown in the Figure 12 and 13.

Once in this position, the hacker can launch a lot of different and very dangerous attacks because he is in the way between to two normal computers. The basic idea is to pretend to be the server to the client, and pretend to be the client to the server, while we sit in the middle decoding traffic from both sides. Anyone who is trying to do such a thing without the proper permissions is incriminating itself into a crime or violation as best.

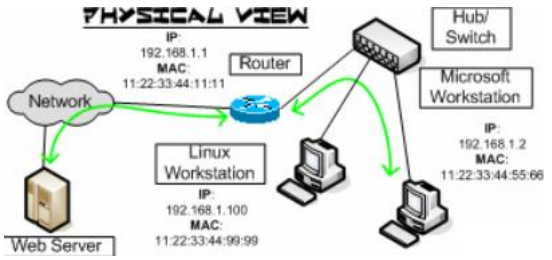


Figure 12
Normal Network View

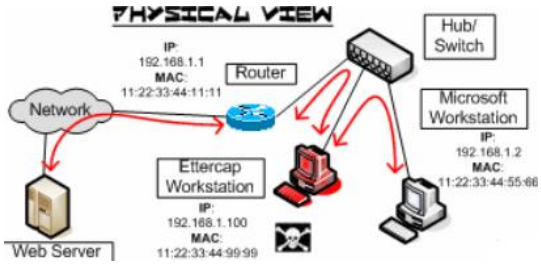


Figure 13
Ettercap ARP Poisoning

To constrain these kinds of actions it was implemented the Certificate Authority system that was designed to prevent exactly this attack, by allowing a trusted third-party to cryptographically sign a server's SSL certificates to verify that they are legit. If this signature doesn't match or is from a non-trusted party, a secure client will simply drop the connection and refuse to proceed. Despite the many shortcomings of the CA system as it exists today, this is usually fatal to attempts to MITM an SSL connection for analysis [13]. In order for an intruder to complete this kind of attack; the hacker has to become a trusted Certificate Authority themselves. The Mitmproxy (another network sniffer/MITM) includes a full CA implementation that generates interception certificates on the fly. According to Table 4 this is how it works.

Table 4
HTTPS Request and Mitmproxy Flow

Client	MITM Proxy	Server
(1)CONNECT Request	(2)Connection established	
(3)Initiate SSL handshake with SNI		(4)Initiate SSL handshake with SNI
(6)Complete SSL handshake		(5)CN & SANs
(7)Request		(8)Request

To show how this works, we used Cain & Abel and Ettercap to act as a MITM to capture/sniff all network data traveling across the Internet, an Internet Router, a computer and another computer with Cain & Abel performing ARP (Address Resolution Protocol) poisoning.

The result was a data stream capture of the user name and password pertaining to an email account in google, see Figure 14and 15. This will only work if you are located inside the same subnet as the computers trying to communicate with each other or with the Internet.

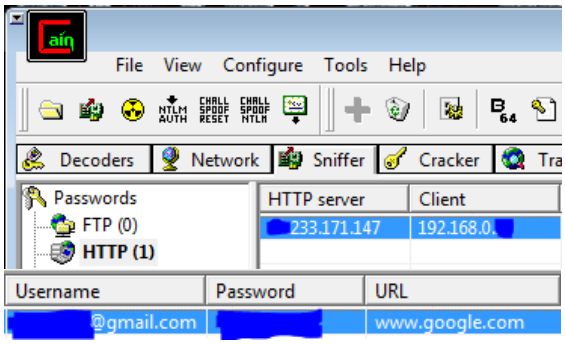


Figure 14
Cain & Abel MITM ARP Poisoning

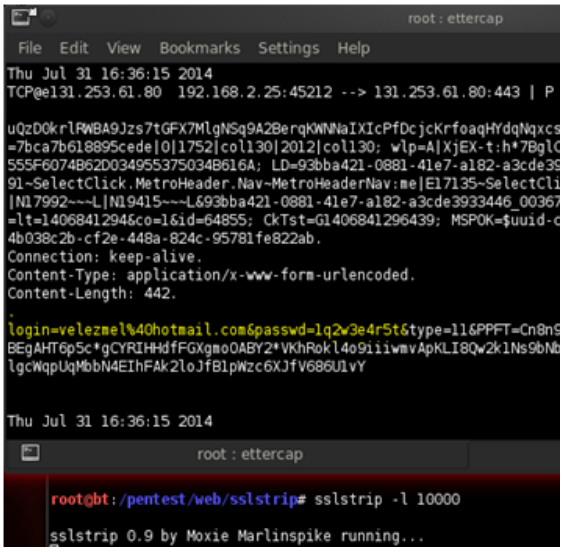


Figure 15
Ettercap MITM ARP Poisoning

EXPERIMENTAL RESULT

After the experiment, the tests performed provided in our study a clearer perspective in the capacities of the tools according to the objectives presented in this investigation.

Most software-based network **protocol analyzers** work in about the same way as is showed in Figure 16. It shows and displays, at least initially, the same basic information. The analyzer runs on a host system. When you start the analyzer (in promiscuous mode), the host NIC's software driver intercepts all traffic that passes through the NIC. The

protocol analyzer passes the intercepted traffic to the analyzer's packet-decoder engine, which identifies and splits packets into their respective layers. The protocol analyzer software analyzes the packets and displays packet information on the analyzer host's screen. Depending on the product's capabilities, you can then analyze and filter the traffic further.

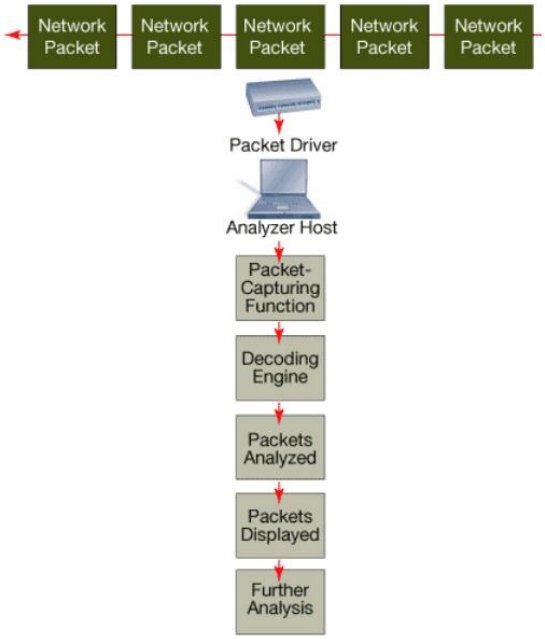


Figure 16
How a Network Analyzer Monitors Traffic

CHALLENGES TO LAW ENFORCEMENT

Although there have been some advances in **network analysis** detection and breaking, there is currently no single easy-to-use tool available to law enforcement. Several factors intensify the challenge faced by law enforcement in detecting network intrusion and penetration: Network intrusion detection is usually handled separately from network penetration [14].

An automated tool that integrates detection and prevention in a way that is familiar and easily accessible to law enforcement has not been yet developed. Newer forms of forcing the entry into the network techniques are being rapidly developed, rendering the current detection tools almost inefficient and ineffective in some cases.

DISCUSSION

Network sniffers, in media circles, are defined as a potential threat. We were able to see the packets sniffers of some of this open source application, which in principle have the same concept of capturing and dissecting Ethernet packets of those of commercially labeled. The task presented before us was actually to test and make a comparison of these tools to see their capabilities and make a report of how they handle the burden of reveling what is happing in our networks. Even though some of the tools used were mostly free, we can say that they perform very well, and most of the public users, developers and web sites engineers agree that they can be used and trust.

As in any computer based company, there is not one tool, that can cope with every user demand, still by evaluating the options available and how they work, will provide us with better knowledge when the moment to make a decision arrives, in order to determine buying and using an out-of-the-box solution or simply choose another that can be customizable [15].

CONCLUSION AND FUTURE WORK

All the Network **Protocol Analyzers** and **Network Sniffers** tools have their advantages and drawbacks as our study revealed.

A protocol analyzer window typically consists of three panes. The top pane displays a summary of the captured packets. Typically, this pane shows at minimum the following fields: date; time that the packet was captured; source and destination IP addresses; source and destination port addresses; protocol type and a summary of the captured data. The middle pane shows the logical breakout of a selected packet, and the bottom pane shows the packet in hexadecimal, ASCII or text-character form.

The analyzer organizes captured packets by layer and protocol. The best **packet analyzers** can recognize a protocol by its most definitive layer—the upper layer—and display the captured information on a field-by-field basis. This type of

information is typically displayed in the analyzer window's second pane. For example, any protocol analyzer can recognize TCP traffic.

Network analyzers like Wireshark, OmniPeek, CommView and Microsoft Message Analyzer are the ones who demonstrated versatility and customization. Also they have a good filtering functionality that let you swift through tons of data that you don't need to look at.

In regard to the **MITM attack** the tools used like Ettercap can provide some statistics such as the traffic on the network interfaces, the weak passwords discovery or the network connections status.

Fighting effectively against ARP poisoning with efficiency is not an easy task because the ARP protocol provides no possibilities to establish the authenticity of the source of incoming packets.

ACKNOWLEDGMENTS

First of all I would like to thank God, who was the one that gave me the courage to pursued my goals of starting and completing this laborious task of getting a master's degree in computer science for His eternal purpose. I will also like to thank my family which is my loving wife Tamara and my two lovely daughters Arianne and Arianell. Without their support and love, for sure, this would have been an impossible task to accomplishment.

REFERENCES

[1] S. McClure, J. Scambray and G. Kurtz, "Hacking Windows," in *Hacking Exposed 7: Network Security Secrets & Solutions*. New York: McGraw-Hill. 2012. Books24x7. [Online]. Available: http://common.books24x7.com/toc.aspx?bookid=47609. [Accessed August 9, 2014].

[2] M. Collier and D. Endler, "UC Network Eavesdropping", in *Hacking Exposed Unified Communications & VOIP Security Secrets & Solutions*, 2nd ed. New York: McGraw-Hill. 2014. Books24x7. [Online]. Available: http://common.books24x7.com/toc.aspx?bookid =59135. [Accessed: August 16, 2014].

[3] J. Fichera and S. Bolt, "Network Analysis," in *Network Intrusion Analysis: Methodologies, Tools, & Techniques for Incident Analysis and Response*. Waltham, MA: Syngress. 2013. Books24x7. [Online]. Available: http://common.books24x7.com/toc.aspx?bookid=47318 [Accessed: August 24, 2014].

[4] K. Beaver, "Network Infrastructure," in *Hacking for Dummies*, 4th ed. Hoboken, NJ: Wiley, 2013. Books24x7. [Online]. Available: http://common.books24x7.com/toc.aspx?bookid=51155 [Accessed: August 24, 2014].

[5] R. Bejtlich, "Collecting Network Traffic," in *The Practice of Network Security Monitoring: Understanding Incident Detection and Response*, San Francisco, CA: No Starch Press, 2013. Books24x7. [Online]. Available: http://common.books24x7.com/toc.aspx?bookid =58348.

[6] T. Wilhelm, "Ethics and Hacking," in *Professional Penetration Testing: Creating and Learning in a Hacking Lab*, 2nd ed. Waltham, MA: Syngress, 2013. Books24x7. [Online]. Available: http://common.books24x7.com/toc.aspx?bookid=56567 [Accessed: August 28, 2014]

[7] G. Combs, (2006). Features. [Online]. Available: https://www.wireshark.org/about.html. [Accessed: November 12, 2014].

[8] WILDPACKETS, Inc. (1998). Products. [Online]. Available: http://www.wildpackets.com/products/omnipeek_network_analyzer. [Accessed: November 12, 2014].

[9] CommView. (1998). [Online] Available: http://www.tamos.com/products/commview. [Accessed: November 13, 2014].

[10] Massimiliano Montoro. (1998). About. [Online]. Available: http://www.oxid.it/cain.html. [Accessed: November 14, 2014].

[11] A. Ornaghi and M. Valleri, (2001, January 25). Ettercap. Home. [Online]. Available: http://ettercap.github.io/ettercap//index.html [Accessed: November 15, 2014].

[12] Microsoft. (1975). Guide. [Online]. Available: http://technet.microsoft.com/en-us/library/jj649776.aspx [Accessed: November 12, 2014].

[13] A. Cortesi, (2014). MITMProxy Console. [Online]. Available: http://mitmproxy.org/doc/mitmproxy.html. [Accessed: November 14, 2014].

[14] C. Easttom and J. Taylor, *Computer Crime, Investigation, and the Law*, Boston, MA: Course Technology, 2011, pp. 445-447.

[15] C. P. Pfleeger and S. L. Pfleeger, *Analyzing Computer Security: A Threat / Vulnerability / Countermeasure Approach*, New Jersey: Prentice Hall, 2012.