Penetration Testing With Banner Grabbers and Packet Sniffers

[1] Tabu S. Kondo, [2] Leonard J. Mselle

[1] Assistant Lecturer, Department of Computer Technologies and Applications, the University of Dodoma, Tanzania
[2] Senior Lecturer, Department of Computer Science, the University of Dodoma, Tanzania

[1] chifulukwe@yahoo.com, [2] mselel@yahoo.com

ABSTRACT

With looming threats from hackers, identity and corporate data thieves, it is impossible to avoid doing penetration testing at the corporate level so as to ascertain the security of the network. The more ubiquitous penetration testing becomes the more likely secure most organizations become. This paper demonstrates how to use banner grabbers and packet sniffers to collect important information related to security issues at the corporate level. Experiment on the use of banner grabbers and packet sniffers were carried out and the results are discussed. Recommendations are made to motivate organizations to use these tools in order to improve organizational preparedness in performing penetration testing, hence improving network security.

Keywords: *Banner grabber, Network sniffer, Packet sniffer, Penetration testing, Ethical hacking*

1. INTRODUCTION

With cyber attacks becoming the norm, it is more important than ever before to improve the ability of organization staff members to perform penetration testing as part of the effort to enhance network security at organizational level.

Penetration testing can be defined as a legal and authorized attempt to locate and successfully exploit computer systems for the purpose of making those systems more secure. The process includes probing for vulnerabilities as well as providing proof of concept (POC) attacks to demonstrate that the vulnerabilities are real. Proper penetration testing always ends with specific recommendations for addressing and fixing the issues that were discovered during the test. On the whole, this process is used to help secure computers and networks against future attacks [1].

A penetration test is designed to evaluate an information system’s defense and discover weaknesses in the network and its resources. Penetration testing is sometimes called ethical hacking because, in some instances, the entity conducting the penetration test is employing techniques used by crackers. The difference is that the ethical hacker is aiming at acquiring information about the network to improve its security as opposed to causing harm. A penetration test can determine how a system reacts to an attack, whether or not a system’s defenses can be breached, and what information can be acquired from the system [2].

1.1 How Secure is Every one?

On June 14, 2013 the ABC NEWS reported: Alleged NSA leaker Edward Snowden claimed to have evidence that the U.S. government has been hacking into Chinese computer networks since at least 2009 – an effort he said is part of the tens of thousands of hacking operations American cyber spies have launched around the world. On March 11, 2013 the NY Times wrote: “The White House demanded Monday that the Chinese government stop the widespread theft of data from American computer networks and agree to *acceptable norms of behavior in cyberspace*”. As it goes, there is allegation and counter allegation between two world governments pointing to absolute insecurity in the cyberspace. What can be concluded from these accusations and counter accusations is that nobody is secure in the cyberspace. So, what to do?

One of the options is to improve the ability of the organization staff members in using penetration testing tools so as to avoid possible attacks without resorting to professional security agents. Use of such testing tools regularly by corporate network/system administrators will bring some useful techniques, which are hitherto exclusive territories of specialists, to the immediate access of the corporations, hence improving network security.

1.2 Objective

The objective of this paper is to demonstrate how banner grabbers and packet sniffers can be used as defensive tools at the corporate level.

2. BANNER GRABBING

Banner grabbing can be defined as connecting to remote applications and observing the output. This can be surprisingly informative to remote attackers. In a similar way, corporate network/system administrators can exploit this ability to detect how vulnerable their networks are. At the very least, one may have identified the make and model of the running services, which in many cases is enough to set the vulnerability research process in motion [3]. A banner grabber is a tool that can be used to extract information from application banners. The banner grabbing technique can be useful to system/network administrators in cataloguing their systems/networks. Ethical hackers can also use it during penetration tests. Malicious hackers use banner grabbing, since the technique can reveal compromising information about the services that are running on a system. The technique works by using Telnet, FTP, and a proprietary program to establish a connection with a remote machine, after which a bad request is sent. That will cause a vulnerable host to respond with a banner message, which may contain information that malicious hacker can use to further compromise the system.

Telnet is a plaintext remote management service that provides command-line shell access to multiple server operating systems including UNIX and Windows, and to devices such as Cisco routers and managed switches. From security perspective, the Telnet protocol is weak because all data (including authentication details) are transmitted in plaintext and can be sniffed by attackers. Once authenticated, users are connected through Telnet, their sessions can also be hijacked and commands injected to the underlying operating systems by attackers with access to the same network segment [4].

The File Transfer Protocol (FTP) has been around as long as the Internet. It was the standard tool for moving or copying large files and is still used today, although to a lesser extent because of the popularity of HTTP. FTP uses port 20 for data transfer and port 21 for control. FTP requires entering a logon name and password and is more secure than the Trivial File Transfer Protocol (TFTP) [5].

2.1 Banner Grabbing Tools

2.1.1 Net cat

Net cat is the most useful tool available for interacting with systems across a network. Net cat, which is often referred to as the Swiss Army knife of network tools, can be used by attackers and network/system administrators alike to accomplish a myriad of tasks [6]. Net cat is available for Windows and Linux operating systems.

2.1.2 Nmap

Nmap is an open source program released under the GNU General Public License. It is an invaluable tool for network/system administrators which can be used to discover, monitor, and troubleshoot TCP/IP systems. Nmap is a free cross-platform network scanning utility created by Gordon “Fyodor” Lyon and is actively developed by a community of volunteers [7].

3. PACKET SNIFFING

Packet sniffing is the process of gathering traffic from a network by capturing the data as they pass and store them for later analysis [8]. A packet sniffer is a tool that captures communication among hosts on a particular network. By capturing this network communication, a packet sniffer can reassemble the network packets to view the information originally sent over the network. Any data sent in plaintext, such as user names, passwords, IP addresses, and other sensitive data, are all vulnerable to eavesdropping [9].

A packet sniffer itself is passive [10]. It observes messages being sent and received by applications and protocols running on the computer, but never sends packets itself. Similarly, received packets are never explicitly addressed to the packet sniffer. Instead, a packet sniffer receives a copy of packets that are sent / received from/by applications and protocols executing on the machine.

3.1 Packet Sniffing Tools

3.1.1 Snort

Snort [11] is a freeware lightweight Intrusion Detection System (IDS) and a general-purpose sniffer for various versions of Linux, Unix, and Windows. There are three main modes in which Snort can be configured: sniffer, packet logger, and network intrusion detection system. The sniffer mode simply reads the packets off of the network and displays them in a continuous stream on the console. Packet logger mode logs the packets to the disk. Network intrusion detection mode is the most complex and customizable configuration, allowing Snort to analyze network traffic for matches against a user defined rule set [12].

3.1.2 Windump

Windump [13] is a freeware general-purpose packet sniffer for various versions of Windows. It is a common packet sniffer that runs under the command line. It allows the user to intercept and display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. TCPDump [14] is a freeware general-purpose packet sniffer for various versions of Linux and UNIX.

3.1.3 Wireshark

Wireshark is one of the best packet sniffers available. It is being developed as a free, commercial-quality sniffer. It has numerous features including; a nice graphical user interface (GUI). It can decode over 400 protocols, and is actively being developed and maintained. It runs on Uniplexed Information and Computing (UNIX)-based systems, Mac OS X, and Windows. Wireshark is a great sniffer to use in a production environment, and is available at www.wireshark.org [15].

4. THE EXPERIMENT

Self penetration testing is not a typical case for many corporations. In this experiment, in order to stay away from doing any computer crime in the production network, one laptop and a virtual machine with a virtual computer network was used to explore penetration testing with various banner grabbers and packet sniffers. During the experiments, a laptop with windows 7 Operating system was used. An IPv4 address of 172.17.1.10/24 was configured. A Linux Server virtual machine was installed in a VMware and connected to the host operating system using bridge networking. Various services such as HTTP, SSH, FTP, SMTP, POP, TELNET and IMAP were installed and configured. The Server’s IPv4 address was 172.17.1.1/24.

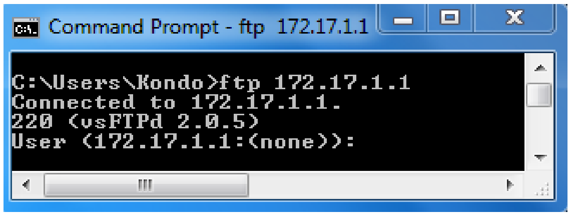
The procedures used in this experiment can be imitated to collect information from any server located in any production network in the world. It can also be used to get credit cards details of innocent people from the compromised machine, get hold of passwords, obtain source code and may obtain email record details illegally. After breaking in, the intruder may conceal their presence from the operating system by using root kit software.

4.1 Banner Grabbing

4.1.1 Banner Grabbing with FTP

In order to do banner grabbing in the FTP service, a connection through command prompt of the host laptop to the Linux FTP server was established. The ftp command used to extract information from FTP application banner is shown in figure 1.

As it can be seen in figure 1, the FTP service running in the Linux server is named vsFTPd whose version is 2.0.5. This is the information that is needed when searching for vulnerabilities of the software in the system/network.



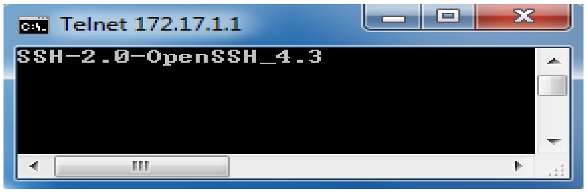
```
C:\Users\Kondo>ftp 172.17.1.1
Connected to 172.17.1.1.
220 (vsFTPd 2.0.5)
User (172.17.1.1:(none)):
```

Fig 1: Banner grabbing with ftp

4.1.2 Banner Grabbing with Telnet

Banner grabbing using Telnet is successful just because when we send wrong information to wrong port the victim returns with error message which also has banner information. The general command to banner grab with telnet is *telnet IP_Address Port*. This command generates the banner of the port that is requested. For example, to find out what SSH (by default, SSH uses port number 22) service is running in the Linux Server, *telnet* was used by running the command:
C:\>Users\Kondo>telnet 172.17.1.1 22

This established a connection to the target machine and displayed a banner revealing the service (i.e. OpenSSH 4.3) and its protocol (i.e. SSH 2.0) as shown in figure 2. The displayed information in figure 2 is useful when someone wants to begin searching for vulnerabilities of the discovered software.

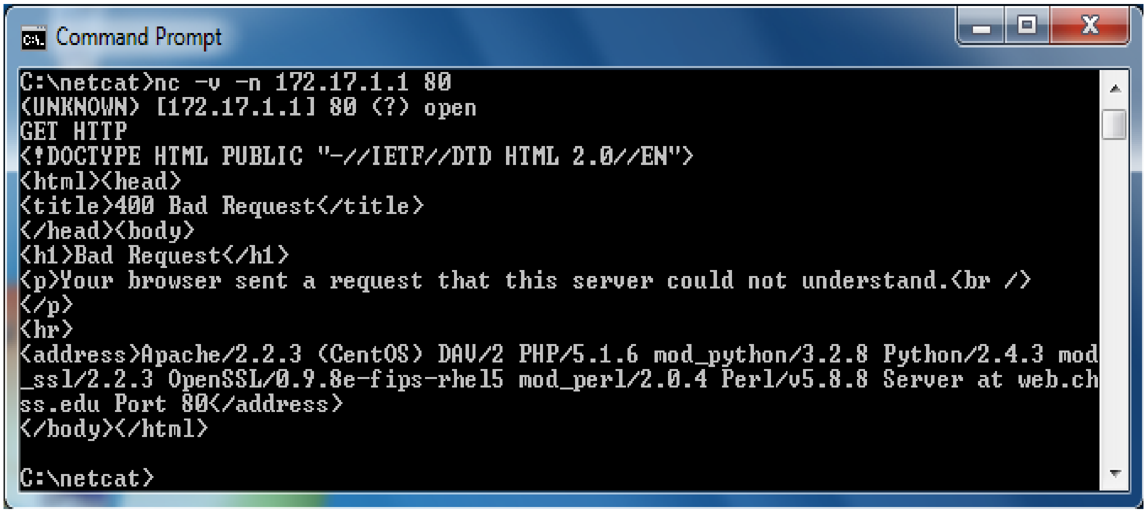


```
SSH-2.0-OpenSSH_4.3
```

Fig 2: Banner grabbing with telnet

4.1.3 Banner Grabbing with Net cat

To banner-grab with netcat, the following general command for the command line can be used: *nc -v -n IP_Address Port*. This command will reveal the banner of the port that is being requested. For example, to query port 80 in the Linux server, the netcat command shown in figure 3 was used. After typing the appropriate command for querying the web services, a prompt waiting for input is displayed. The command *GET HTTP* is typed as shown in figure 3. The information obtained in figure 3 reveals that the Linux server is using Apache web server along with its version (2.2.3). As a matter of fact, this information can be used as an initial stage of finding vulnerabilities of the discovered software.

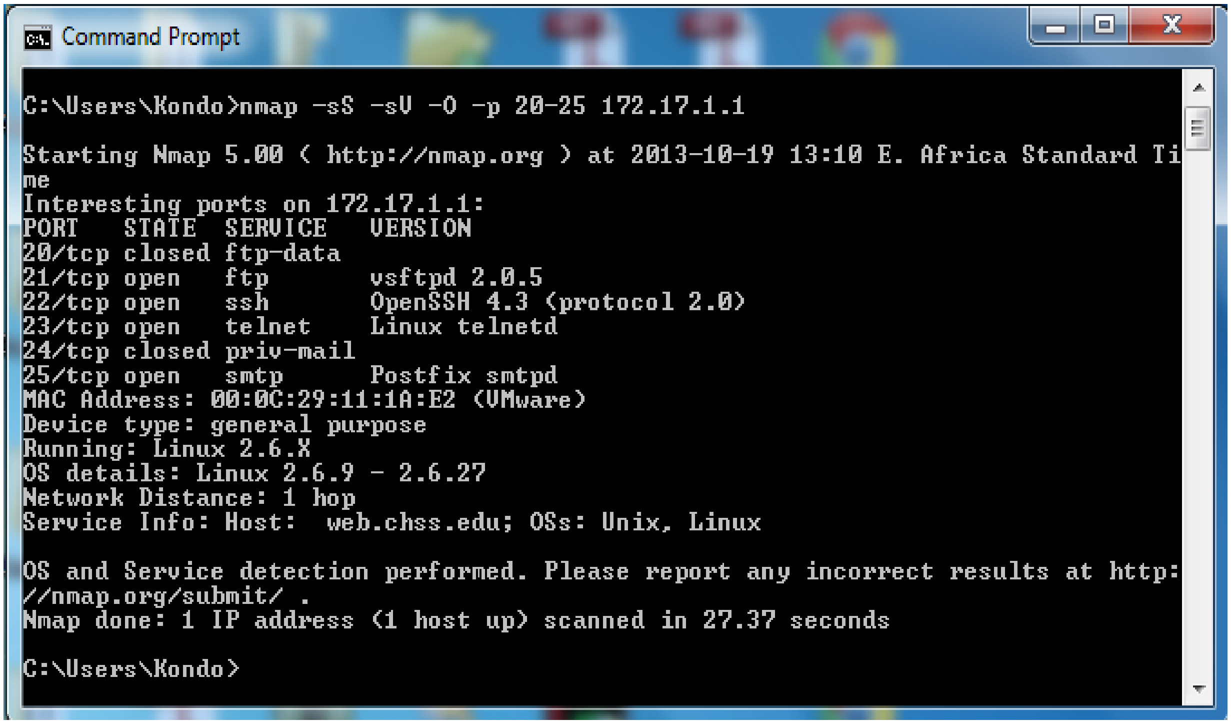


```
C:\netcat>nc -v -n 172.17.1.1 80
(UNKNOWN) [172.17.1.1] 80 (?) open
GET HTTP
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.2.3 (CentOS) DAV/2 PHP/5.1.6 mod_python/3.2.8 Python/2.4.3 mod
_ssl/2.2.3 OpenSSL/0.9.8e-fips-rhel5 mod_perl/2.0.4 Perl/v5.8.8 Server at web.ch
ss.edu Port 80</address>
</body></html>

C:\netcat>
```

Fig 3: Banner grabbing with netcat

4.1.4 Banner Grabbing with Nmap

The version-scanning feature of nmap is invoked with the -sV flag. Based on a returned banner, or on a specific response to an nmap-provided probe, a match is made between the service response and the nmap service fingerprints [5]. This type of enumeration can be very noisy as unusual packets are sent to guess the service version. As such, Intrusion detection systems (IDSs) alerts will likely be generated unless some other type of mechanism is used to mask it. Figure 4 shows a successful scan using *nmap -sS -sV -O* against a Linux server. This performs a SYN-based port-scan with a version scan and uses the OS fingerprinting function. The version scanner picks up the version (4.3) and protocol (2.0) of OpenSSH in use, along with the Linux kernel level range (2.6.x), etc. The information picked up by the nmap scanner can help to use the available search engines so as to find out vulnerabilities of the discovered software of the target machine.



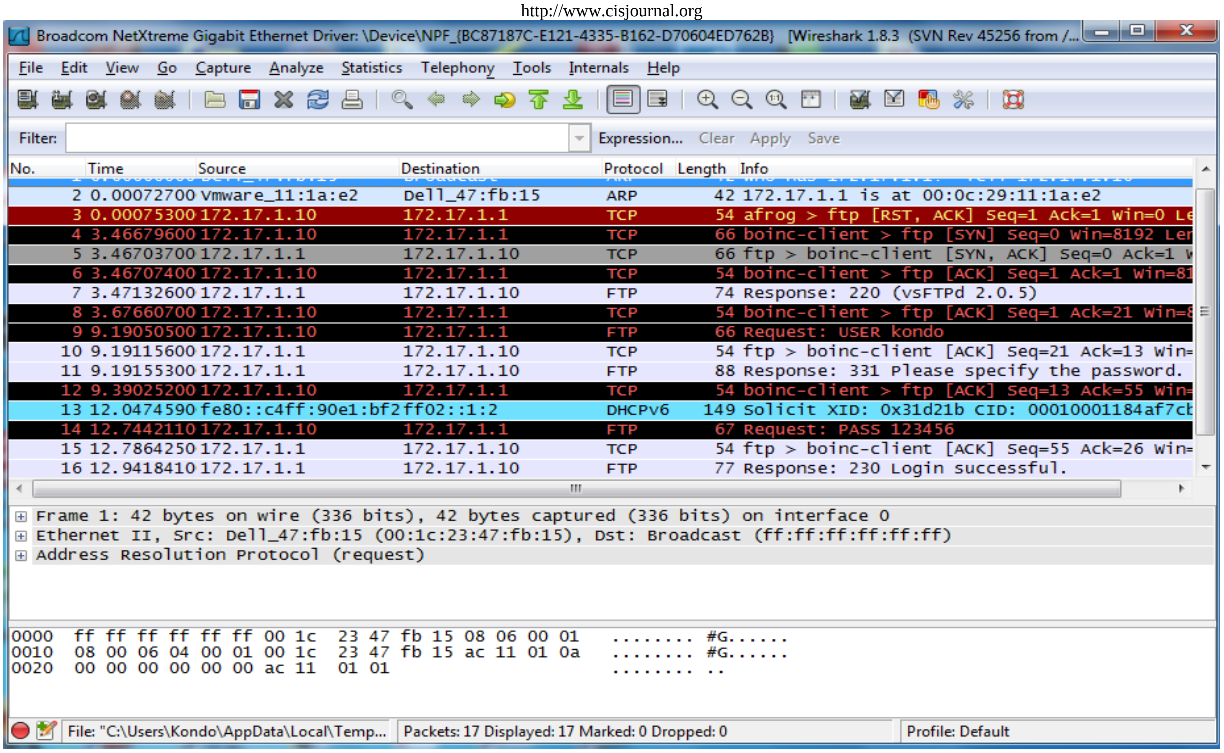
```
C:\Users\Kondo>nmap -sS -sV -O -p 20-25 172.17.1.1

Starting Nmap 5.00 ( http://nmap.org ) at 2013-10-19 13:10 E. Africa Standard Ti
me
Interesting ports on 172.17.1.1:
PORT   STATE  SERVICE  VERSION
20/tcp closed ftp-data
21/tcp open   ftp      vsftpd 2.0.5
22/tcp open   ssh      OpenSSH 4.3 (protocol 2.0)
23/tcp open   telnet   Linux telnetd
24/tcp closed priv-mail
25/tcp open   smtp     Postfix smtpd
MAC Address: 00:0C:29:11:1A:E2 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.9 - 2.6.27
Network Distance: 1 hop
Service Info: Host:  web.chss.edu; OSs: Unix, Linux

OS and Service detection performed. Please report any incorrect results at http:
//nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.37 seconds

C:\Users\Kondo>
```

Fig 4: Banner grabbing with nmap

4.2 Packet Sniffing

4.2.1 Basic Sniffing with Wireshark

Wireshark has many features such as ability to capture data from live network connection or read from a file that contains already-captured packets. Wireshark is able to read data from a wide variety of networks, from Ethernet, IEEE 802.11, PPP, and even loopback. The captured network data can be monitored and managed via a GUI – which allows for plug-ins to be inserted and used. Wireshark can capture VoIP packets and raw USB traffic.. To start capturing traffic, the wireshark software is opened, and from the capture menu the intended interface(s) is/are selected. Figure 5 shows the captured FTP traffic when an ftp login process is invoked.


Broadcom NetXtreme Gigabit Ethernet Driver: \Device\NPF_{BC87187C-E121-4335-B162-D70604ED762B} [Wireshark 1.8.3 (SVN Rev 45256 from /...
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

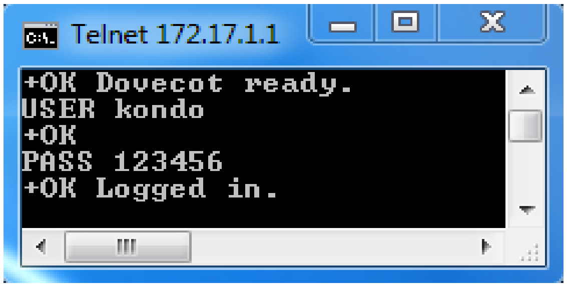
```
No.   Time       Source                 Destination  Protocol Length Info
  2 0.00072700 Vmware_11:1a:e2          Dell_47:fb:15 ARP      42 172.17.1.1 is at 00:0c:29:11:1a:e2
  3 0.00075300 172.17.1.10              172.17.1.1    TCP      54 afrog > ftp [RST, ACK] Seq=1 Ack=1 Win=0 Le
  4 3.46679600 172.17.1.10              172.17.1.1    TCP      66 boinc-client > ftp [SYN] Seq=0 Win=8192 Len
  5 3.46703700 172.17.1.1               172.17.1.10   TCP      66 ftp > boinc-client [SYN, ACK] Seq=0 Ack=1 W
  6 3.46707400 172.17.1.10              172.17.1.1    TCP      54 boinc-client > ftp [ACK] Seq=1 Ack=1 Win=81
  7 3.47132600 172.17.1.1               172.17.1.10   FTP      74 Response: 220 (vsFTPd 2.0.5)
  8 3.67660700 172.17.1.10              172.17.1.1    TCP      54 boinc-client > ftp [ACK] Seq=1 Ack=21 Win=8
  9 9.19050500 172.17.1.10              172.17.1.1    FTP      66 Request: USER kondo
 10 9.19115600 172.17.1.1               172.17.1.10   TCP      54 ftp > boinc-client [ACK] Seq=21 Ack=13 Win=
 11 9.19155300 172.17.1.1               172.17.1.10   FTP      88 Response: 331 Please specify the password.
 12 9.39025200 172.17.1.10              172.17.1.1    TCP      54 boinc-client > ftp [ACK] Seq=13 Ack=55 Win=
 13 12.0474590 fe80::c4ff:90e1:bf2      ff02::1:2     DHCPv6  149 Solicit XID: 0x31d21b CID: 00010001184af7cb
 14 12.7442110 172.17.1.10              172.17.1.1    FTP      67 Request: PASS 123456
 15 12.7864250 172.17.1.1               172.17.1.10   TCP      54 ftp > boinc-client [ACK] Seq=55 Ack=26 Win=
 16 12.9418410 172.17.1.1               172.17.1.10   FTP      77 Response: 230 Login successful.

⊞ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
⊞ Ethernet II, Src: Dell_47:fb:15 (00:1c:23:47:fb:15), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ Address Resolution Protocol (request)

0000  ff ff ff ff ff ff 00 1c  23 47 fb 15 08 06 00 01   ........ #G......
0010  08 00 06 04 00 01 00 1c  23 47 fb 15 ac 11 01 0a   ........ #G......
0020  00 00 00 00 00 00 ac 11  01 01                     ........ ..

File: "C:\Users\Kondo\AppData\Local\Temp... | Packets: 17 Displayed: 17 Marked: 0 Dropped: 0 | Profile: Default
```

Fig 5: Captured FTP traffic with wireshark

4.2.2 Using Snort as a Packet Sniffer and Logger

In its simplest form, snort is a packet sniffer. The command line interface used for sniffing network traffic in this experiment is:
C:\Snort\bin>snort -vde -i 5 –l c:\snort\log.

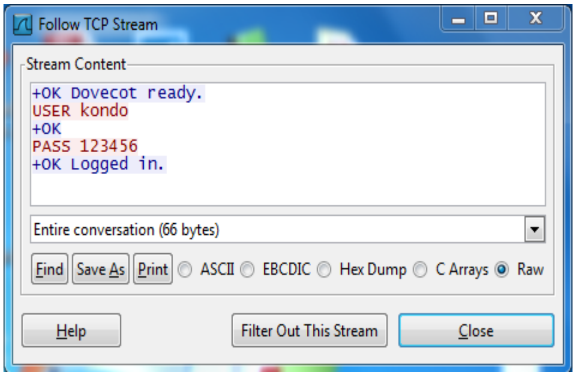
In order to see the TCP and IP packet header information, application-layer headers and the data link-layer headers –vde options were used in the snort command. Furthermore, the c:\snort\log option was used to save the captured data to the hard disk. At the command line, the telnet command was used through port 110 (POP) to log in to the Linux Server as shown in figure 6.



```
+OK Dovecot ready.
USER kondo
+OK
PASS 123456
+OK Logged in.
```

Fig 6: Logging in the POP server through telnet

The Snort software captured the traffic, and when the captured file is viewed in the wireshark, the output is as shown in figure 7.



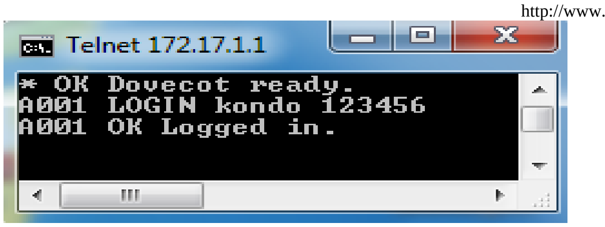
```
+OK Dovecot ready.
USER kondo
+OK
PASS 123456
+OK Logged in.
```

Entire conversation (66 bytes)
○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

Fig 7: Retrieved POP traffic with wireshark

4.2.3 Basic Sniffing with Windump

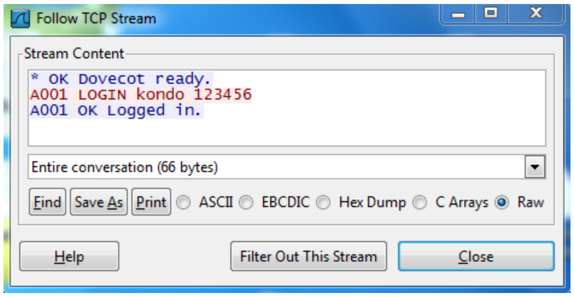
The windump command used to capture the network traffic with windump software is *C:\>windump –i 5 –s 5000 –w c:\windump\windump.log*. This command directs tcpdump to sniff traffic and write it to a file called *windump.log*. At the command line, a telnet command is issued through port 143 (IMAP), to facilitate logon process to the Linux server as shown in figure 8.



```
* OK Dovecot ready.
A001 LOGIN kondo 123456
A001 OK Logged in.
```

Fig 8: Logging in the IMAP server through telnet

The windump software captured the traffic, and when the captured file is viewed in the wireshark, the output is as shown in figure 9.



```
* OK Dovecot ready.
A001 LOGIN kondo 123456
A001 OK Logged in.
```

Entire conversation (66 bytes)

Fig 9: Retrieved IMAP traffic with wireshark

5. DISCUSSIONS OF THE RESULTS

The experiment reveals two main scenarios; the first scenario demonstrates how banner grabbers (such as FTP, TELNET, netcat and nmap) can be used by network/system administrators and network security professionals for OS and application fingerprinting at organizational level. The second scenario demonstrates how packet sniffers (such as windump, snort and wireshark) can be used by network/system administrators and network security professionals to collect necessary security information of a network at the organizational level. The capabilities of wireshark to retrieve network traffic data captured by other packet sniffers such as windump/tcpdump and snort were also demonstrated. This has vast implications for network and information security at organizational domain.

Although the experiment was performed on the Linux server, the same procedures can be applied on the other network operating systems such as Windows server 2003, 2008, 2012, etc. Furthermore, the experiment was performed in a virtual networking environment, but the same procedures can be applied on the production network environment.

6. CONCLUSION AND RECOMMENDATIONS

Historically, packet sniffers were dedicated hardware devices that were expensive and difficult to use. However, new advances in technology have allowed for the development of software-based packet sniffers, which make it more convenient and affordable for network/system administrators to effectively troubleshoot a network. It also brings the capability of network security analysis in the hands of system administrators at corporate level. Professional penetration testers use network traffic data to understand the activity of hosts attached to the network, especially malicious activities. With the increasing demand for security expertise in the Internet, there is a strong need for penetration testers in the industry, government and the military.

System administrators, network engineers, security engineers, system operators, and programmers at corporate level, can easily resort into the use packet sniffers, which are invaluable tools for diagnosing and troubleshooting network problems, system configuration issues, and application difficulties without necessarily outsourcing this function to specialized firms.

Banner grabbing allows system hackers to discover valuable information of various services and the operating system of the specified target machine. To improve security, system administrators, network engineers, security engineers, system operators and programmers who are responsible for managing the security of the organization network are advised to change the default banners (after verification) so as to improve the security of their application software and operating systems. Moreover, unnecessary services should be disabled. Furthermore, it is advised to use the available techniques such as Secure Socket Layer (SSL), Secure Shell (SSH), Virtual Private networks (VPNs) and Pretty Good Privacy (PGP)/Secure MIME (SMIME) to encrypt the data at corporate level so that intruders are denied reading capabilities even if they can sniff inside.

In general, we recommend that using common network security tools to perform self penetration testing at corporate level will help corporations to detect vulnerabilities without resorting to outside expertise and hence improve their security posture.

REFERENCES

[1] P. Engebretson, The Basics of hacking and penetration testing: Ethical Hacking and Penetration Testing Made Easy, Syngress, 2011.

[2] E. Cole, R. Krutz, and J. W. Conley, Network Security Bible, Wiley Publishing, Inc., 2005.

[3] S. Mcclure, J. Scambray, G. Kurtz, Hacking exposed™ 7: network security secrets & solutions, McGraw Hill, 2012.

[4] C. McNab, Network Security Assessment, O'Reilly, 2008.

[5] M. T. Simpson, K. Backman, and J. E. Corley, Hands-On Ethical Hacking and Network Defense, Course technology, 2011.

[6] E. Skoudis and T. Liston, Counter Hack Reloaded, Second Edition: A Step-by-Step Guide to

Computer Attacks and Effective Defenses, Prentice Hall, 2005.

[7] N. Marsh. (2010) Nmap® Cookbook The fat-free guide to network scanning [Online]. Available: www.NmapCookbook.com.

[8] R. L. Krutz and R. D. Vines, The CEH™ Prep Guide: The Comprehensive Guide to Certified Ethical Hacking, Wiley Publishing, 2007.

[9] D. Mackey, Web Security for Network and System Administrators, 1st edition, Thomson, 2003.

[10] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 6th edition, Pearson, 2013.

[11] Snort software [Online]. Available: www.snort.org.

[12] T. J. Klevinsky, S. Laliberte, and A. Gupta, Hack I.T: Security Through Penetration Testing, Addison Wesley, 2002.

[13] Windump software [Online]. Available: http://netgroup-serv.polito.it/windump.

[14] TCPDump software [Online]. Available: www-nrg.ee.lbl.gov.

[15] M. Krishnamurthy, E. S. Seagren, R. Alder, A. W. Bayles, J. Burke, S. Carter, and E. Faskha, How to Cheat at Securing Linux, Syngress publishing, 2008.