VASANTDADA PATIL PRATISHTHAN'S COLLEGE OF ENGINEERING AND VISUAL ARTS

Department of Information Technology

Experiment No: 8

Aim: Study the use of network reconnaissance tools like WHOIS, dig, traceroute, nslookup to gather information about networks and domain registrars.

Theory:

WHOIS :

WHOIS is the Linux utility for searching an object in a WHOIS database. The WHOIS database of a domain is the publicly displayed information about a domains ownership, billing, technical, administrative, and name server information. Running a WHOIS on your domain will look the domain up at the registrar for the domain information. All domains have WHOIS information. WHOIS database can be queried to obtain the following information via WHOIS:

- Administrative contact details, including names, email addresses, and telephone numbers
- Mailing addresses for office locations relating to the target organization
- Details of authoritative name servers for each given domain

Example: Querying Facebook.com

ssc@ssc-OptiPlex-380:~$ whois facebook.com **Whois Server Version 2.0** Domain names in the .com and .net domains can now be registered with many different competing registrars. Go to http://www.internic.net for detailed information. Server Name: FACEBOOK.COM.BRETLANDTRUSTMERCHANDISINGDEPART.COM

IP Address: 69.63.176.11

Registrar: GOOGLE INC.

Whois Server: whois.rrpproxy.net

Referral URL: http://domains.google.com

Server Name: FACEBOOK.COM.DISABLE.YOUR.TIMELINE.NOW.WITH.THE.ORIGINAL.TIMELINE-REMOVE.NET

IP Address: 8.8.8.8

Registrar: ENOM, INC.

Whois Server: whois.enom.com

Referral URL: http://www.enom.com

Server Name: FACEBOOK.COM.GET.ONE.MILLION.DOLLARS.AT.WWW.UNIMUNDI.COM

IP Address: 209.126.190.70

Registrar: PDR LTD. D/B/A PUBLICDOMAINREGISTRY.COM

Whois Server: whois.PublicDomainRegistry.com

Referral URL: http://www.PublicDomainRegistry.com

Dig - Dig is a networking tool that can query DNS servers for information. It can be very helpful for diagnosing problems with domain pointing and is a good way to verify that your configuration is working. The most basic way to use dig is to specify the domain we wish to query:

dig example.com

$ dig example.com

Traceroute - traceroute prints the route that packets take to a network host. Traceroute utility uses the TTL field in the IP header to achieve its operation. For users who are new to TTL field, this field describes how much hops a particular packet will take while traveling on network. So, this effectively outlines the lifetime of the packet on network. This field is usually set to 32 or 64. Each time the packet is held on an intermediate router, it decreases the TTL value by 1. When a router finds the TTL value of 1 in a received packet then that packet is not forwarded but instead discarded. After discarding the packet, router sends an ICMP error message of ―Time exceeded‖ back to the source from where packet generated. The ICMP packet that is sent back contains the IP address of the router. So now it can be easily understood that traceroute operates by sending packets with TTL value starting from 1 and then incrementing by one each time. Each time a router receives the packet, it checks the TTL field, if TTL field is 1 then it discards the packet and sends the ICMP error packet containing its IP address and this is what traceroute requires. So traceroute incrementally fetches the IP of all the routers between the source and the destination.

Example: traceroute example.com

Nslookup - The nslookup command is used to query internet name servers interactively for information. nslookup, which stands for "name server lookup", is a useful tool for finding out information about a named domain. By default, nslookup will translate a domain name to an IP address (or vice versa). For instance, to find out what the IP address of microsoft.com is, you could run the command:

$nslookup microsoft.com

Conclusion: Hence we have successfully studied various network reconnaissance tools.

Experiment No: 9

Aim: Study of packet sniffer tools wireshark: - a. Observe performance in promiscuous as well as non-promiscuous mode. b. Show the packets can be traced based on different filters.

Theory:

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color-coding and other features that let you dig deep into network traffic and inspect individual packets.

Features of Wireshark :

- Available for UNIX and Windows.
- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a
- number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.

Capturing Packets

After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

Installation of Wireshark:

```
sudo apt-get install wireshark
```

After downloading and installing wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. Figure 5 shows the example of interface list in wireshark. For example, if you want to capture traffic on the wireless network, click your wireless interface. You can configure advanced features by clicking Capture Options.

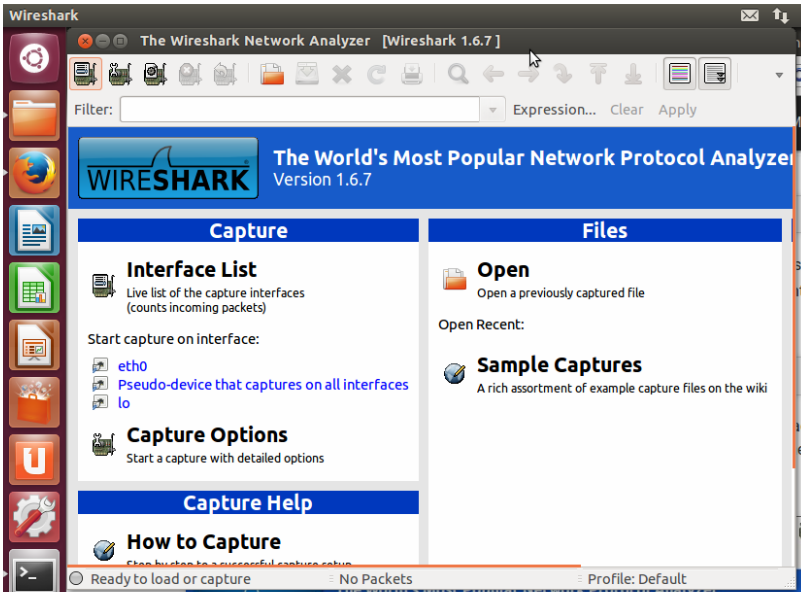

Wireshark
The Wireshark Network Analyzer [Wireshark 1.6.7]
Filter:
Expression...
Clear
Apply
WIRESHARK
The World's Most Popular Network Protocol Analyzer
Version 1.6.7
Capture
Interface List
Live list of the capture interfaces (counts incoming packets)
Start capture on interface:
eth0
Pseudo-device that captures on all interfaces
lo
Capture Options
Start a capture with detailed options
Capture Help
How to Capture
Files
Open
Open a previously captured file
Open Recent:
Sample Captures
A rich assortment of example capture files on the wiki
Ready to load or capture
No Packets
Profile: Default

Figure 5: Interface List using Wireshark

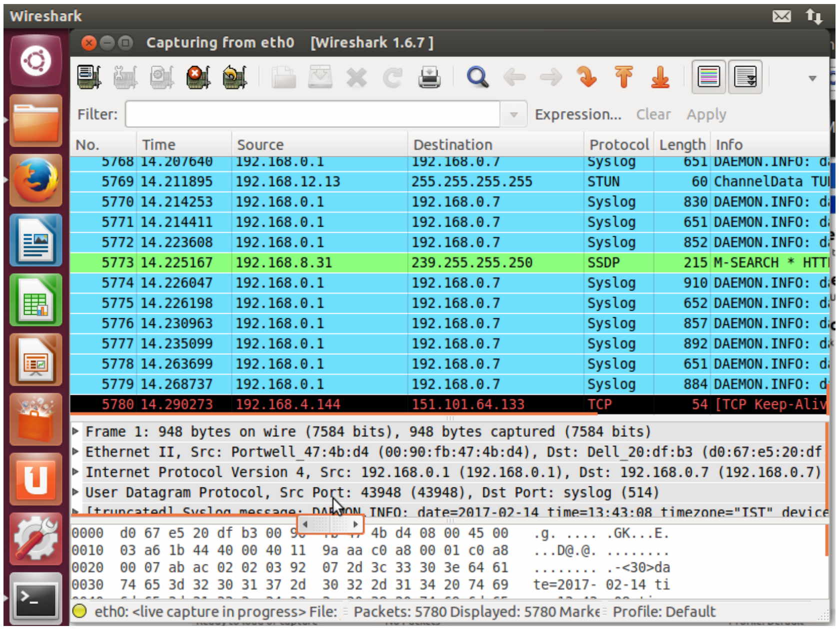
As soon as you click the interface‘s name, you‘ll see the packets start to appear in real time. Wireshark captures each packet sent to or from your system. If you‘re capturing on a wireless interface and have promiscuous mode enabled in your capture options, you‘ll also see other the other packets on the network.

Click the stop capture button near the top left corner of the window when you want to stop capturing traffic. Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.

Filtering Packets

If you‘re trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you‘ll likely have a large amount of packets to sift through. That‘s where filters of Wireshark come in. Figure 6 shows filter in Wireshark and traffic running in Wireshark.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type ―dns and you‘ll see only DNS packets (Figure 7). When you start typing, Wireshark will help to you auto complete your filter.


Wireshark
Capturing from eth0 [Wireshark 1.6.7]
Expression... Clear Apply

Filter:

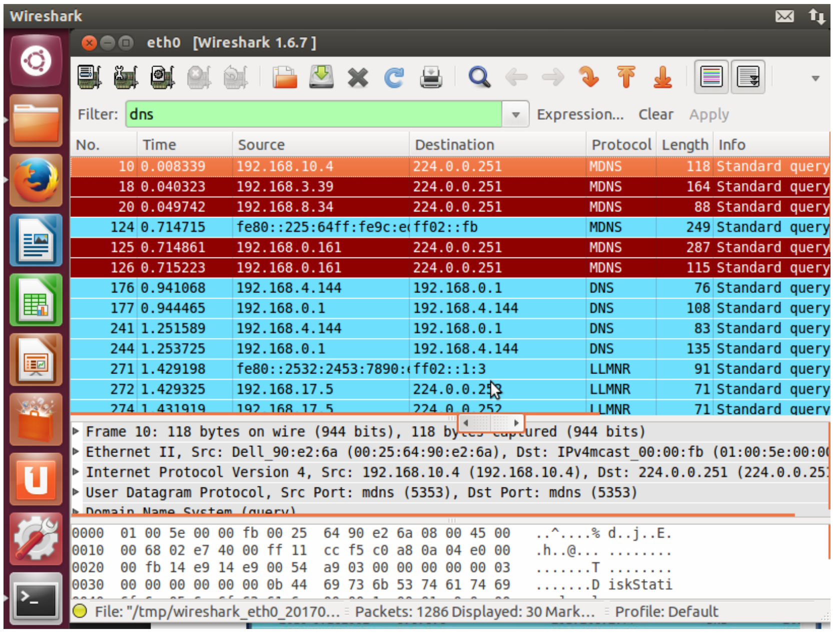
No.	Time	Source	Destination	Protocol	Length	Info
5768	14.207640	192.168.0.1	192.168.0.7	Syslog	651	DAEMON.INFO: d
5769	14.211895	192.168.12.13	255.255.255.255	STUN	60	ChannelData TU
5770	14.214253	192.168.0.1	192.168.0.7	Syslog	830	DAEMON.INFO: d
5771	14.214411	192.168.0.1	192.168.0.7	Syslog	651	DAEMON.INFO: d
5772	14.223608	192.168.0.1	192.168.0.7	Syslog	852	DAEMON.INFO: d
5773	14.225167	192.168.8.31	239.255.255.250	SSDP	215	M-SEARCH * HTT
5774	14.226047	192.168.0.1	192.168.0.7	Syslog	910	DAEMON.INFO: d
5775	14.226198	192.168.0.1	192.168.0.7	Syslog	652	DAEMON.INFO: d
5776	14.230963	192.168.0.1	192.168.0.7	Syslog	857	DAEMON.INFO: d
5777	14.235099	192.168.0.1	192.168.0.7	Syslog	892	DAEMON.INFO: d
5778	14.263699	192.168.0.1	192.168.0.7	Syslog	651	DAEMON.INFO: d
5779	14.268737	192.168.0.1	192.168.0.7	Syslog	884	DAEMON.INFO: d
5780	14.290273	192.168.4.144	151.101.64.133	TCP	54	[TCP Keep-Aliv

```
▸ Frame 1: 948 bytes on wire (7584 bits), 948 bytes captured (7584 bits)
▸ Ethernet II, Src: Portwell_47:4b:d4 (00:90:fb:47:4b:d4), Dst: Dell_20:df:b3 (d0:67:e5:20:df
▸ Internet Protocol Version 4, Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.7 (192.168.0.7)
▸ User Datagram Protocol, Src Port: 43948 (43948), Dst Port: syslog (514)
▸ [truncated] Syslog message: DAEMON.INFO: date=2017-02-14 time=13:43:08 timezone="IST" device
```

```
0000  d0 67 e5 20 df b3 00 90  fb 47 4b d4 08 00 45 00   .g. .... .GK...E.
0010  03 a6 1b 44 40 00 40 11  9a aa c0 a8 00 01 c0 a8   ...D@.@. ........
0020  00 07 ab ac 02 02 03 92  07 2d 3c 33 30 3e 64 61   ........ .-<30>da
0030  74 65 3d 32 30 31 37 2d  30 32 2d 31 34 20 74 69   te=2017- 02-14 ti
```

eth0: <live capture in progress> File: Packets: 5780 Displayed: 5780 Marke Profile: Default

Figure 6: Filter in Wireshark and traffic running in Wireshark



Filter: dns

No.	Time	Source	Destination	Protocol	Length	Info
10	0.008339	192.168.10.4	224.0.0.251	MDNS	118	Standard query
18	0.040323	192.168.3.39	224.0.0.251	MDNS	164	Standard query
20	0.049742	192.168.8.34	224.0.0.251	MDNS	88	Standard query
124	0.714715	fe80::225:64ff:fe9c:e	ff02::fb	MDNS	249	Standard query
125	0.714861	192.168.0.161	224.0.0.251	MDNS	287	Standard query
126	0.715223	192.168.0.161	224.0.0.251	MDNS	115	Standard query
176	0.941068	192.168.4.144	192.168.0.1	DNS	76	Standard query
177	0.944465	192.168.0.1	192.168.4.144	DNS	108	Standard query
241	1.251589	192.168.4.144	192.168.0.1	DNS	83	Standard query
244	1.253725	192.168.0.1	192.168.4.144	DNS	135	Standard query
271	1.429198	fe80::2532:2453:7890:e	ff02::1:3	LLMNR	91	Standard query
272	1.429325	192.168.17.5	224.0.0.252	LLMNR	71	Standard query
274	1.431919	192.168.17.5	224.0.0.252	LLMNR	71	Standard query

```
▸ Frame 10: 118 bytes on wire (944 bits), 118 bytes captured (944 bits)
▸ Ethernet II, Src: Dell_90:e2:6a (00:25:64:90:e2:6a), Dst: IPv4mcast_00:00:fb (01:00:5e:00:00
▸ Internet Protocol Version 4, Src: 192.168.10.4 (192.168.10.4), Dst: 224.0.0.251 (224.0.0.25
▸ User Datagram Protocol, Src Port: mdns (5353), Dst Port: mdns (5353)
▸ Domain Name System (query)
```

```
0000  01 00 5e 00 00 fb 00 25  64 90 e2 6a 08 00 45 00   ..^....% d..j..E.
0010  00 68 02 e7 40 00 ff 11  cc f5 c0 a8 0a 04 e0 00   .h..@... ........
0020  00 fb 14 e9 14 e9 00 54  a9 03 00 00 00 00 00 03   .......T ........
0030  00 00 00 00 00 00 0b 44  69 73 6b 53 74 61 74 69   .......D iskStati
```

File: "/tmp/wireshark_eth0_20170... Packets: 1286 Displayed: 30 Mark... Profile: Default

Figure 7: DNS Filter in Wireshark and traffic running in Wireshark

Program: Installation of Wireshark and running of Wireshark.

Output: Observe and analyze the traffic using Wireshark.

Conclusion: Hence we have successfully studied packet sniffer tool Wireshark.

Experiment No: 10

Aim: Download and install nmap. Use it with different options to scan open ports, perform OS fingerprinting, do a ping scan, tcp port scan, udp port scan, etc.

Theory:

Nmap (Network Mapper) is a security scanner originally written by Gordon Lyon (also known by his pseudonym Fyodor Vaskovich) used to discover hosts and services on a computer network, thus creating a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host and then analyzes the responses. Unlike many simple port scanners that just send packets at some predefined constant rate, Nmap accounts for the network conditions (latency fluctuations, network congestion, the target interference with the scan) during the run. Also, owing to the large and active user community providing feedback and contributing to its features, Nmap has been able to extend its discovery capabilities beyond simply figuring out whether a host is up or down and which ports are open and closed; it can determine the operating system of the target, names and versions of the listening services, estimated uptime, type of device, and presence of a firewall.

Nmap features include:

- Host Discovery – Identifying hosts on a network. For example, listing the hosts which respond to pings or have a particular port open.
- Port Scanning – Enumerating the open ports on one or more target hosts.
- Version Detection – Interrogating listening network services listening on remote devices to determine the application name and version number.
- OS Detection – Remotely determining the operating system and some hardware characteristics of network devices.

Basic commands working in Nmap:

- For target specifications: nmap <target‘s URL or IP with spaces between them>
- For OS detection: nmap -O <target-host's URL or IP>
- For version detection: nmap -sV <target-host's URL or IP>

SYN scan is the default and most popular scan option for good reasons. It can be performed quickly, scanning thousands of ports per second on a fast network not hampered by restrictive firewalls. It is also relatively unobtrusive and stealthy since it never completes TCP connections

Algorithm\Implementation Steps\Installation Steps:

- Installing Nmap from the link.

  ```
  sudo apt-get install nmap
  ```

- Obtaining Your IP addresses.

  Use the ifconfig command in Linux.

- Performing a Scan of the Local Network.

1. For the following steps, please use the nmap command line tool installed on Ubuntu
2. Scan your subnet to determine how many hosts can be found. For example, if you are on the 192.168.1.0 subnet, you would enter the following command: nmap –sP 192.168.1.*

   i. What is your subnet? _________________

   ii. How many hosts were found? _________________
3. Next perform a stealth scan (Please use the IP for your subnet): nmap –sS –P0 –p 192.169.1.*
4. Now, you'll perform an OS identification. Use the Linux O/S to scan your Windows machine:

   i. nmap –O Windows_IP_ADDRESS

   ii. OS Type 1:________________________________

   iii. Now we want to use the Windows machine to scan the Linux O/S. Go to a Windows DOS prompt and enter the following command:

   iv. nmap –O Linux_IP_ADDRESS

   v. Now we will perform a service selection scan. Let's scan for all computers with FTP running. We would do that as follows:

   nmap –p21 192.168.1.*

5. List the IP addresses with that has the FTP open: ______________________

Program: Execution of nmap.

Output: Observe the output of namp.

Conclusion: Hence we have successfully studied and used nmap.

Experiment No: 11

Aim: Study of malicious software using different tools e.g. make use of the NESSUS to scan the network for vulnerabilities.

Theory:

Nessus is one of the most popular and capable vulnerability scanners, particularly for UNIX systems. Nessus is a remote security scanning tool, which scans a computer and raises an alert if it discovers any vulnerabilities that malicious hackers could use to gain access to any computer we have connected to a network.

Nessus is not a complete security solution, rather it is one small part of a good security strategy. NESSUS does not actively prevent attacks; it is only a tool that checks your computers to find vulnerabilities that hackers could exploit. It is up to the System Administrator to patch these Vulnerabilities in order to create a security solution.

To learn how NESSUS and other port-scanning security tools work, it is necessary to understand different services (such as a web server, SMTP server, FTP server, etc) are accessed on a remote server. Most high-level network traffic, such as email, web pages, etc reach a server via a high-level protocol that is transmitted reliably by a TCP stream. To keep different streams from interfering with each other, a computer divides its physical connection to the network into thousands of logical paths, called ports. So if we want to talk to a web server on a given machine, we would connect to port #80 (the standard HTTP port), but if we want to connect to an SMTP server on that same machine we would instead connect to port #25.

Each computer has thousands of ports, all of which may or may not have services (ie: a server for a specific high-level protocol) listening on them. Nessus works by testing each port on a computer, determining what service it is running, and then testing this service to make sure there are no vulnerabilities in it that could be used by a hacker to carry out a malicious attack. Nessus is called a "remote scanner" because it does not need to be installed on a computer for it to test that computer. Instead, we can install it on only one computer and test as many computers as we would like.

Conclusion: Hence we have successfully studied a security vulnerability scanning tool Nessus.

Experiment No: 12

Aim: Study of Network security by setting up IPSEC under LINUX

Theory:

Internet Protocol Security (**IPsec**) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a communication session. IPsec includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used in protecting data flows between a pair of hosts (*host-to-host*), between a pair of security gateways (*network-to-network*), or between a security gateway and a host (*network-to-host*)

Internet Protocol security (IPsec) uses cryptographic security services to protect communications over Internet Protocol (IP) networks. IPsec supports network-level peer authentication, data origin authentication, data integrity, and data confidentiality (encryption), and replay protection.

IPsec is an end-to-end security scheme operating in the Internet Layer of the Internet Protocol Suite, while some other Internet security systems in widespread use, such as Transport Layer Security (TLS) and Secure Shell (SSH), operate in the upper layers at Application layer. Hence, only IPsec protects any application traffic over an IP network. Applications can be automatically secured by IPsec at the IP layer.

ipsec can be implemented using strongSwan tool. strongSwan is a IPsec implementation. It uses openSSL pluugin (Eliptic Curve Cryptography).

Let two servers be red server (192.168.4.144) and blue server (192.168.4.145)

Installation and configuration on red server

Step 1: Installation of strongswan

```
project@project-OptiPlex-360:~$ sudo apt-get install ipsec-tools strongswan-starter
```

Step 2: Configuration of ipsec.conf

```
project@project-OptiPlex-360:~$ sudo gedit /etc/ipsec.conf
```

Step 3: Configuration of ipsec.secrets

```
project@project-OptiPlex-360:~$ sudo gedit /etc/ipsec.secrets
```

Step 4: Start ipsec

```
project@project-OptiPlex-360:~$ sudo ipsec restart
```

```
Stopping strongSwan IPsec...
Starting strongSwan 5.3.5 IPsec [starter]...
```

Step 5: Checking status information of ipsec

```
project@project-OptiPlex-360:~$ sudo ipsec statusall
```

Conclusion: Hence we have successfully studied IPSec.