ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Vocational Education and Training

MSEI-022
Network Security

Network Security Technology

4

"शिक्षा मानव को बन्धनों से मुक्त करती है और आज के युग में तो यह लोकतंत्र की भावना का आधार भी है। जन्म तथा अन्य कारणों से उत्पन्न जाति एवं वर्गगत विषमताओं को दूर करते हुए मनुष्य को इन सबसे ऊपर उठाती है।"

— इन्दिरा गांधी

"Education is a liberating force, and in our age it is also a democratising force, cutting across the barriers of caste and class, smoothing out inequalities imposed by birth and other circumstances."

– Indira Gandhi

ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Vocational Education and Training

MSEI-022
Network Security

Block

4

NETWORK SECURITY TECHNOLOGY

Programme Expert/Design Committee of Post Graduate Diploma in Information Security (PGDIS)

Prof. K.R. Srivathsan
Pro Vice-Chancellor, IGNOU

Mr. B.J. Srinath, Sr. Director & Scientist 'G',CERT-In, Department of Information Technology, Ministry of Communication and Information Technology, Govt of India

Mr. A.S.A Krishnan, Director, Department of Information Technology, Cyber-Laws and E-Security Group, Ministry of Communication and Information Technology, Govt of India

Mr. S. Balasubramony, Dy. Superintendent of Police, CBI, Cyber Crime Investigation Cell Delhi

Mr. B.V.C. Rao, Technical Director, National Informatics Centre, Ministry of Communication and Information Technology

Prof. M.N. Doja, Professor, Department of Computer Engineering, Jamia Milia Islamia New Delhi

Dr. D.K. Lobiyal, Associate Professor, School of Computer and Systems Sciences, JNU New Delhi

Mr. Omveer Singh, Scientist, CERT-In Department of Information Technology, Cyber-Laws and E-Security Group, Ministry of Communication and Information Technology Govt of India

Dr. Vivek Mudgil, Director, Eninov Systems Noida

Mr. V.V.Subrahmanyam, Assistant Professor School of Computer and Information Science IGNOU

Mr. Anup Girdhar, CEO, Sedulity Solutions & Technologies, New Delhi

Prof. A.K. Saini, Professor, University School of Management Studies, Guru Gobind Singh Indraprastha University, Delhi

Mr. C.S. Rao, Technical Director in Cyber Security Division, National Informatics Centre Ministry of Communication and Information Technology

Prof. C.G. Naidu, Director, School of Vocational Education & Training, IGNOU

Prof. Manohar Lal, Director, School of Computer and Information Science, IGNOU

Prof. K. Subramanian, Director, ACIIL, IGNOU Former Deputy Director General, National Informatics Centre, Ministry of Communication and Information Technology, Govt of India

Prof. K. Elumalai, Director, School of Law IGNOU

Dr. A. Murali M Rao, Joint Director, Computer Division, IGNOU

Mr. P.V. Suresh, Sr. Assistant Professor School of Computer and Information Science IGNOU

Ms. Mansi Sharma, Assistant Professor School of Law, IGNOU

Ms. Urshla Kant
Assistant Professor, School of Vocational Education & Training, IGNOU
Programme Coordinator

Block Preparation

Unit Writers

Prof. Gopi Krishna S Garge
Department of ECE, IISc, Bangalore

Dr. Malati Hegde, Department of ECE IISc, Bangalore

(Unit 1)

Mr. Ashish Shubham
B.Tech & M.Tech (Computer Science & Engineering), IIT Kharagpur
(Unit 2 & 3)

Block Editors

Mr. Anup Girdhar, CEO
Sedulity Solutions & Technologies, New Delhi

Ms. Urshla Kant
Assistant Professor, School of Vocational Education & Training, IGNOU

Proof Reading

Ms. Urshla Kant
Assistant Professor
School of Vocational Education & Training
IGNOU

Production

Mr. B. Natrajan	Mr. Jitender Sethi	Mr. Hemant Parida
Dy. Registrar (Pub.)	Asstt. Registrar (Pub.)	Proof Reader
MPDD, IGNOU, New Delhi	MPDD, IGNOU, New Delhi	MPDD, IGNOU, New Delhi

August, 2011

ISBN: 978–81–266–5615–8

Further information about the School of Vocational Education and Training and the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068. or the website of IGNOU www.ignou.ac.in

Printed and published on behalf of the Indira Gandhi National Open University, New Delhi, by the Registrar, MPDD
Laser typeset by Mctronics Printographics, 27/3 Ward No. 1, Opp. Mother Dairy, Mehrauli, New Delhi-30
Printed at: Berry Art Press A-9, Mayapuri Phase-I New Delhi-64

BLOCK INTRODUCTION

Network security technology refers to the technological safeguards and managerial procedure which can ensure that organizational assets and individual privacy are protected over the network. Network security is needed to secure the data, to prevent it from hacker and to protect the network. This block introduces many of the methods to secure the network such as firewalls, IDS/IPS/Honeypots, Scanning and analysis tools. This block comprises of three units and is designed in the following way;

The **Unit one** introduces firewall and its functions. The security requirements of the network are also discussed. Firewalls is a special program designed for preventing hackers from breaking into the corporate network stopping users on the internet corporate network from gaining access to any internet resources that may prove harmful to the network. A firewall do in terms of security in the generations are Packet Filtering, Application Layer filtering and stateful filter or circuit filtering.

The **Unit two** covers the detailed descriptions of the IDS/IPS/Honeypots. IDS, analyze network traffic and generate alerts when malicious activity is discovered. IPS, perform the same analysis as Intrusion Detection Systems but, because they are inserted in-line, between other network components, they can preempt malicious activity. In contrast to IDS sensors, network traffic flows through an IPS sensor not past it so the IPS sensor can pull or drop traffic from the wire. A honeypot is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems.

The **Unit three** explains Scanning and analysis tools. It describes different IP network scanning methods which allow you to test and effectively identify vulnerable network components. These scanning methods may be classified as Port scanners (TCP/UDP), Vulnerability Scanners (Nesus Scanner), Protocol Analyzers/Sniffers (Ethereal), Formal assessment tools (finSAT). These may be used to remove prospective vulnerabilities and hence strengthen the security of any network but at the same time can be also sued by an attacker to attack and cause disruption. Thus, it is also necessary to take preventive counter measures to restrict these tools also.

Hope you benefit from this block.

ACKNOWLEDGEMENT

The material we have used is purely for educational purposes. Every effort has been made to trace the copyright holders of material reproduced in this book. Should any infringement have occurred, the publishers and editors apologize and will be pleased to make the necessary corrections in future editions of this book.

UNIT 1 FIREWALLS

Structure

1.0 INTRODUCTION

Firewalls are devices that are said to address perimeter security. We will learn what firewalls are, functionally, where they are deployed in a typical network and how the firewall functions. We also understand the typical threats and vulnerabilities that exist in a network and how the firewall mitigates these. Finally, we look at a firewall function implemented on a Linux host and attempt to understand its behaviour.

1.1 OBJECTIVES

After studying this unit you should be able to:

- define Firewalls;
- define security requirements of the network;
- describe various functions of firewalls; and
- explain various types of firewalls.

1.2 WHAT IS A FIREWALL?

Thousands of years ago, the idea of a wall is evolved to protect from the external sources of environment. As such like the chinese build the Great Wall to protect the northern borders of the China against intrusions by tribes.

According to the history of evolution, in the beginning there was no internet, no e-mails, no network and the people entirely relying on telephone and postal mail services to communicate. People used unknown names to send junk messages through postal mail. Later, the internet developed by Advanced Research Projects Agency Network (ARPANET), changed the traditional postal mails. It is the place where the line is borrowed, "where everyone knows your name".

On November 2, 1988, the network was under attack by internet Virus. Later, named it as The Morris Worm.The researchers realized that the internet is no longer a closed community of trusted colleagues. The community started practices to prevent the future attacks and intense focus on network security, the firewall was starting its rapid evolution.

The term firewall is originated from construction of subdividing a building into separate fire areas. A firewall refers to a fire-resistant barrier wall which prevents the fire spreading rapidly from one partition to other. The firewall was in use as early as 1764, by Lightoler to describe the separation of cooking section in a building.

The fire walls are regularly found in cars, transformers or subdividing a building to enhance resistance and prevent the spread of fire. The same firewall is true in steam trains, as described by Schneier:

Coal-powered trains had a large furnace in the engine room, along with a pile of coal. The engineer would shovel coal into the engine. This process created coal dust, which was highly flammable. Occasionally the coal dust would catch fire, causing an engine fire that sometimes spread into the passenger cars. Since dead passengers reduced revenue, train engines were built with iron walls right behind the engine compartment. This stopped fires from spreading into the passenger cars, but didn't protect the engineer between the coal pile and the furnace.

For example, a typical car production will include a metal firewall which seals the fuel tank, seperates the engine from passenger compartment (as shown in Fig. 1). In any event of accidental fire, the firewall can prevent spreading of fire entering passenger compartment.

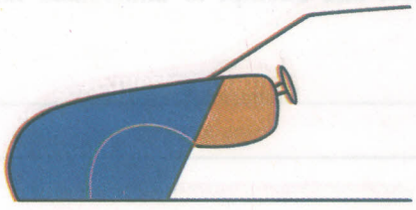


Fig. 1: The firewall in a car (red line)

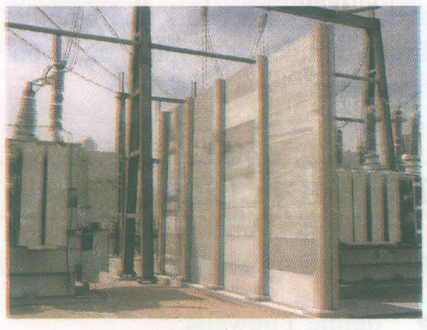


Fig. 2: firewall in electric substation

Similarly, this mechanism is also used in aircrafts. The term firewall is popularly used in many automobile manufacturing to seperate engine from passenger area, in the area of construction and electric substation as shown in Fig. 2, which divides transformers in to seperate fire areas.

www.google.com). The client sends a request to server and the server processes the client request and send the requested website to clients browser (as response).

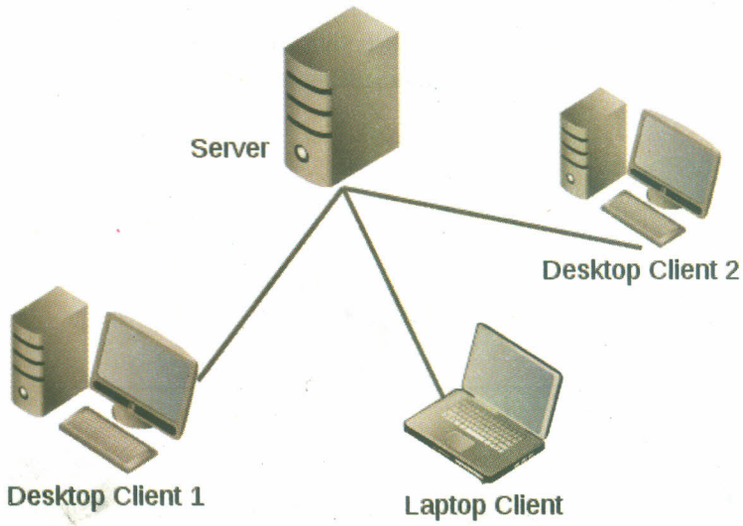


Fig. 5: Client-Server Network Model

Another popular network, thats gaining popularity over the years are the cloud network. Which has a combination of clients, remote servers, isp's, network devices like switches and routers. This network model will be simpler, cheaper and easy to maintain than client-server model. Since the organisation are not buying servers, maintaining and powering. However, they use the remote servers in the cloud.

In the example below, several computers, servers and network devices are interconnected using switch. And this network is connected to servers in the cloud through router. Considering, a calculation that need high computational power, like multiplication of matrix 30x30. Your local server may not be capable of doing it faster, you can use the servers in the cloud which are maintained by third-party (Pay per use).

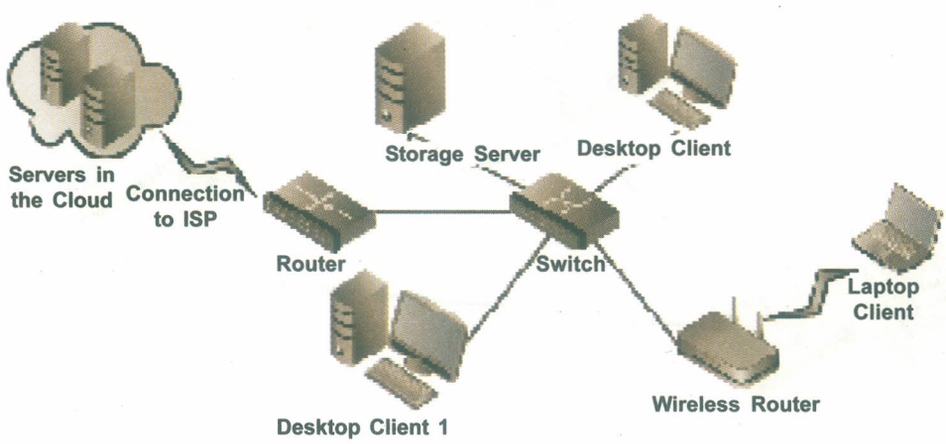


Fig. 6: Cloud-based Network Model

The channel of communication among nodes may be through cables, telephone lines, radio waves, satellites, infrared light or optical fiber.

In a network, each computer or network device is identified by its IP address.

Every packet of information is assigned with a header containing the source and destination information (as shown in Fig. 7).

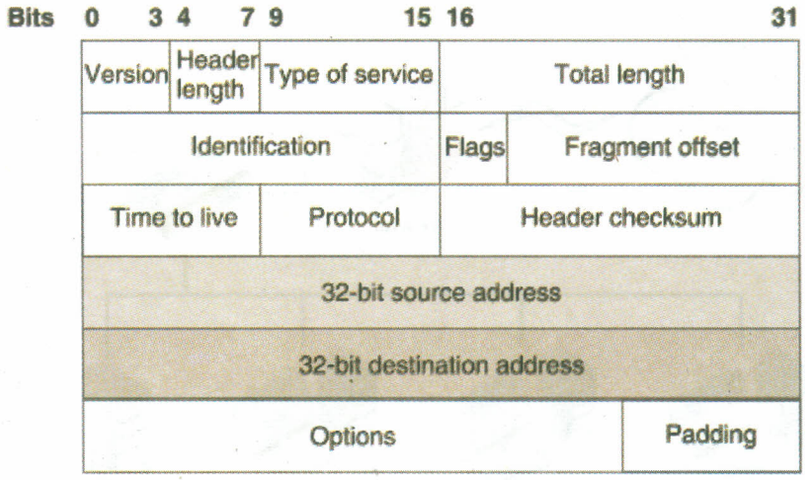


Fig. 7: Internet Protocol (IPv4) header

1.4 WHERE CAN ONE SEE A FIREWALL?

In your home or an organization where security is the preliminary issue to computers connected to a network, they need to be secured from the potential hackers.

The ever first commercial firewall is DEC SEAL by Digital Equipment Corporation (DEC) Network Systems Lab. It was made up of an external system, called Gatekeeper, the only system the Internet could talk to, a filtering gateway, called Gate and an internal Mailhub.The DEC SEAL consisted of three components (as shown in Fig. 8):

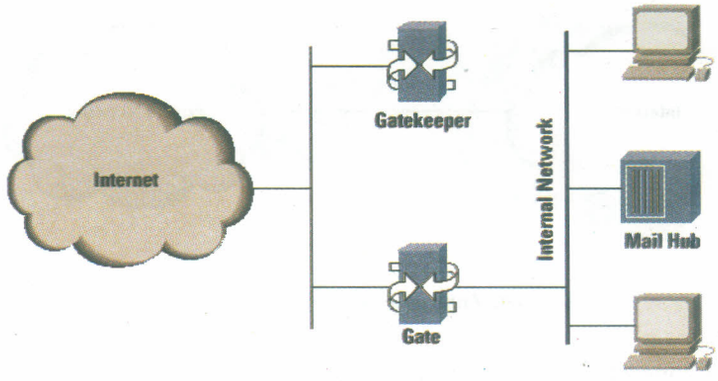


Fig. 8: DEC SEAL -First Commercial Firewall

Gatekeeper: The application proxy server for users who were allowed to access external services. It's also for the uses such as anonymous FTP, the Domain Name System (DNS), etc.

Gate: A packet filtering router, limiting what traffic was allowed to pass between the local and external network. This router was configured so that all traffic to/ from the inside went to a proxy on gatekeeper.

Mailgate: The internal mail server (mail router); this machine was not accessible from the outside. Instead, mail received from the internet is delivered to gatekeeper, which passes it to mailgate.

What a firewall can do? is it allows or block the network traffic between devices based on the set of rules, by the administrator. Each rule defines a specific traffic pattern and the action to be taken, when the pattern is detected.

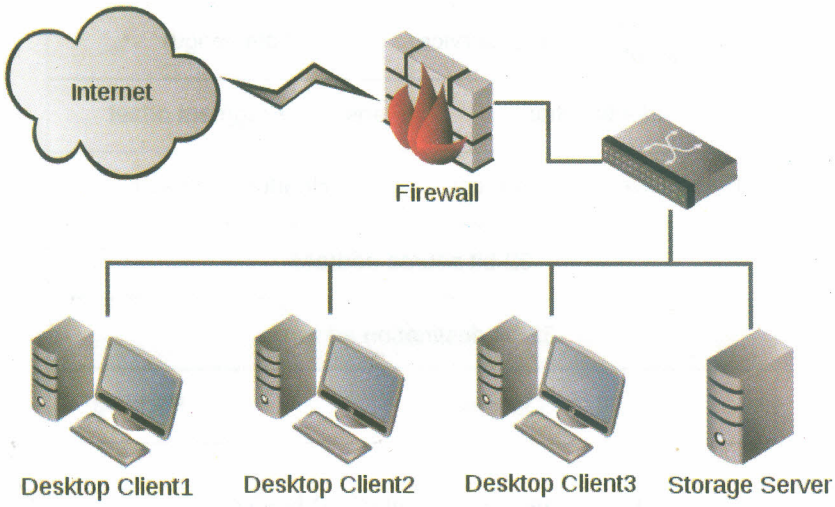


Fig. 9: Firewall in a Network

A firewall can only operate on the traffic that physically passes through it. It has no impact on the traffic between the devices on the same side of the firewall (as shown in Fig. 9). For example, traffic from host Desktop Client 1 to Desktop Client 2. It will apply the traffic rules on packets that passes through it. Like, when host Desktop Clients access some information on internet.

When an organization is connected to internet without firewall (as shown in Fig. 10), the exposure to attack is called the "zone of risk". Every host on the internet are accessible and can attack every host on the private network.

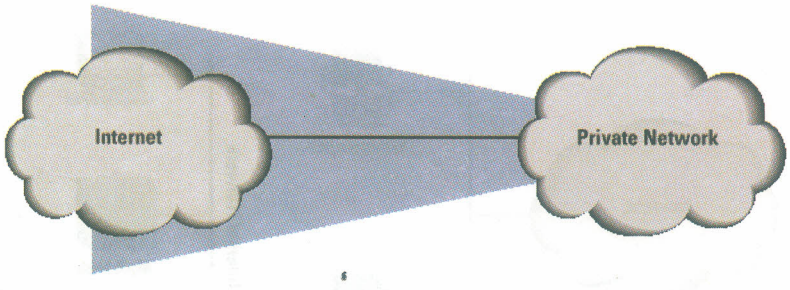


Fig. 10: Zone of risk to the Network

To reduce the zone of risk, we require to implement a firewall system. As we see in the Fig. 11, the zone of risk will now be the firewall system itself. Now, every host in the internet can attack the firewall system, it becomes easy task to monitor all the risk at one place (firewall).

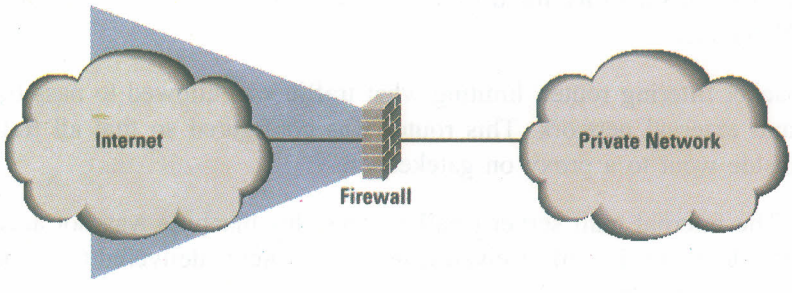


Fig. 11: Zone of risk reduced to the Firewall

A firewall cannot detect packets with "bad intent". They cannot protect against an insider attack (can be tracked with logs, if the user uses internet as part of his crime). Firewalls cannot protect the connections or traffic that do not go through the firewall. It provides little protection from unknown or new attacks.

1.5 SECURITY REQUIREMENTS OF THE NETWORK

Network security is preliminary act of organizations, enterprises and institutions to protect their valuable information across the network. And it involves the process of preventing and detecting unauthorized users using or accessing the communication channel.

We require to identify threats and implement a set of tools to combat them. Computer network are vulnerable to many threats, including

- Social engineering, wherein someone tries to gain access through social means (pretending to be a legitimate system user or administrator, tricking people into revealing secrets etc.)
- War dialing, wherein someone uses computer software and a modem to search for desktop computers equipped with modems that answer, providing a potential path into a corporate network
- Denial-of-service attacks, including all types of attacks intended to overwhelm a computer or a network in such a way that legitimate users of the computer or network cannot use it
- Protocol-based attacks, which take advantage of known (or unknown) weaknesses in network services
- Host attacks, which attack vulnerabilities in particular computer operating systems or in how the system is set up and administered
- Password guessing
- Eavesdropping of all sorts, including stealing e-mail messages, files, passwords and other information over a network connection by listening in on the connection.

Firewalls can help protect against from some of these attacks, but not all. Network security includes the tools, that provide safety from all the above activities. They include

- antivirus software packages
- virtual private network
- secure network infrastructure
- encryption and
- security management

None of the tools alone can protect the entire security requirement, they must be layered together to achieve expected level of security.

Check Your Progress 1

Note: a) Space is given below for writing your answer.

b) Compare your answer with the one given at the end of the Unit.

Explain security requirements of the network.

.....

.....

.....

.....

1.6 TYPICAL NETWORK VULNERABILITIES

Network vulnerabilities are the most common news in the domain of network security. Any breach in the network firewall configuration or settings will result in the theft of sensitive information.

Most common vulnerabilities cause by incomplete or incorrect implementation of intrusion detection/ prevention system. These attacks can never be stopped by a network administrator, but they must be always one step ahead of hackers' tricks.

Every network administrator must follow some basic principles as provided in the following sections.

1.6.1 Access to Unwanted Services/Ports

A Full installation of the any enterprise linux operating system will install thousands of packages and libraries. Most of the administrators will not prefer to install the selected packages, instead the basic selection itself will include several server applications.

This will install several unwanted packages and services running on the machine. With the default settings and possibly service is turned on, which will create problem with the unwanted services such as DHCP, Telnet so on without the administrator realizing it, which in turn creates unwanted traffic to the server or a breach to the hackers. Network administrators must keep track of unwanted services accessing the internet and unwanted services running on the local machine.

1.6.2 Well Known Services

The vulnerabilities may also happen even on the well known services running on your server, if they are incorrect or incompletely configured. Best example like, administrators often install services like Network monitoring tools, port scanner, ping tools etc. and often forget to change the default password provided. These constitute a vulnerability on the system since they provide a easy way of attacking the network.

Similarly, weak passwords or passwords resemble username or dictionary words always encourage hackers to attack on your network.

1.6.3 Generating Excessive Traffic

When hackers get access to a system, the first thing they intend to do is to attempt to disrupt the functional network. An easy means of doing that is by generating excessive traffic on the network from the host that they have penetrated. Tools like ping can be used to achieve this.

1.6.4 Other Implementation Vulnerabilities

Another category of insecure services include NFS (Network File System) and NIS (Network Information Services), which are inherently developed for usage across LAN. But unfortunately, extended to the WAN for remote users, NFS/NIS does not, have any authentication or security by default. Administrator must configure it to prevent the hacker form mounting NFS shared files and accessing vital information.

NIS has the vital information including passwords and file permissions within a plain text ASCII or DBM (ASCII) database. A cracker who gains access to the file will have every user account details on the network including administrator's account.

There are several such implementation vulnerabilities. A large percentage of vulnerabilities arise fomr the fact that systems are not configured appropriately. Then, there are implementation vulnerabilities such as buffer overflows etc. that indicate problems with the protocol implementations or application implementations.

1.7 FUNCTIONS OF THE FIREWALL

In data networking, a firewall is a device with set of rules to permit or deny network access by unauthorized services. It is as similar to the originated fire wall in terms of functionality.

Many operating systems support software based firewall to deny access against the private internet. Software firewalls acts between network card drivers and operating system.

The firewall must be positioned in the network to control all the incoming and outgoing traffic. Usually firewall is positioned as shown in the diagram, which have the control of entire network traffic filtering the packets that physically passes through it.

There are many ways of unsuspicious people use to access the unprotected computers over the network by remote login, SMTP session hijacking, operating system bugs, denial of service, viruses, spam and many more. Some of the attacks are hard and some of them are impossible to filter using a firewall. Example like, spam mails can not be stopped in a firewall as long as you accept e-mail.

The level of security you establish, will determine how many of the threats can be stopped. The highest level of security is achieved by simply blocking everything. Obviously, that is not the purpose of having a network with firewall.

So, what does a firewall do in terms of security in the generations?

- The most basic firewall performs Packet Filtering
- The second most performs Application Layer filtering
- The third most performs stateful filter or circuit filtering

The disadvantages in setting up the firewall is obvious, restricting the access to internal application from outside world. To maintain high level of security we need to allow services like DNS naming servers, e-mail forwarding, mirror repository or proxy services. Only selective services need to communicate with external network and restrict access to services (like NFS, NIS, finger, telnet, etc.).

1.8 TYPES OF FIREWALL

Basically, a firewall is a barrier to keep destructive forces away from your property. In fact, that's why it's called a firewall. Its job is similar to a physical firewall that keeps a fire from spreading from one area to the next. A firewall is actually a device or program that blocks undesired Internet traffic, including viruses, from accessing your computer. Both Windows and Mac OS X have built-in firewall programs that are easy to set up. By blocking unwanted Internet traffic, a lot of viruses and bugs can be stopped dead in their tracks.

Firewalls make it possible to filter incoming and outgoing traffic that flows through your system. A firewall can use one or more sets of "rules" to inspect the network packets as they come in or go out of your network connections and either allows the traffic through or blocks it. The rules of a firewall can inspect one or more characteristics of the packets, including but not limited to the protocol type, the source or destination host address and the source or destination port.

Firewalls can greatly enhance the security of a host or a network. They can be used to do one or more of the following things:

- To protect and insulate the applications, services and machines of your internal network from unwanted traffic coming in from the public Internet.
- To limit or disable access from hosts of the internal network to services of the public Internet.
- To support network address translation (NAT), which allows your internal network to use private IP addresses and share a single connection to the public Internet (either with a single IP address or by a shared pool of automatically assigned public addresses).

The primary purpose of a firewall is to filter traffic. Firewalls inspect packets as they pass through and based on the criteria that the administrator has defined, the firewall allows or denies each packet.

Firewalls block everything that you haven't specifically allowed. Routers with filtering capabilities are a simplified example of a firewall. Administrators often configure them to allow all outbound connections from the internal network, but to block all incoming traffic. So, a user on the internal network would be able to download e-mail without a problem, but an administrator would need to customize the router configuration to connect to your home PC from work by using Remote Desktop. Other applications that might require special firewall configuration are WebCam servers, collaboration software and multiplayer online games.

You use packet filters to instruct a firewall to drop traffic that meets certain criteria. For example, you could create a filter that would drop all ping requests. You can also configure filters with more complex exceptions to a rule. For example, a filter might assist with troubleshooting the firewall by allowing the firewall to respond to ping requests coming from a monitoring station's IP address. By default, Microsoft ISA Server doesn't respond to ping queries on its external interface. You would need to create a packet filter on the ISA Server computer for it to respond to a ping request.

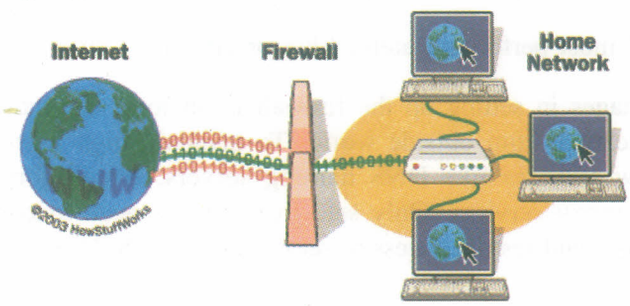


Fig. 12

There are five most popular generations of firewall, each have different method of filtering network packets.

1.8.1 Packet Filtering

All internet traffic in the network is of the packets form. A packet is small quantity of information, for easy handling. When a large amount of data must be sent across the network, it is broken in to large number of packets(pieces) for easy and effective transmission.

All your file downloads, communications, e-mails, webpages across the internet are transfered in the form of packets. A packet is a series of numbers, basically having the information of

- Source IP address
- Destination IP address
- The data
- Error checking information
- Protocol information

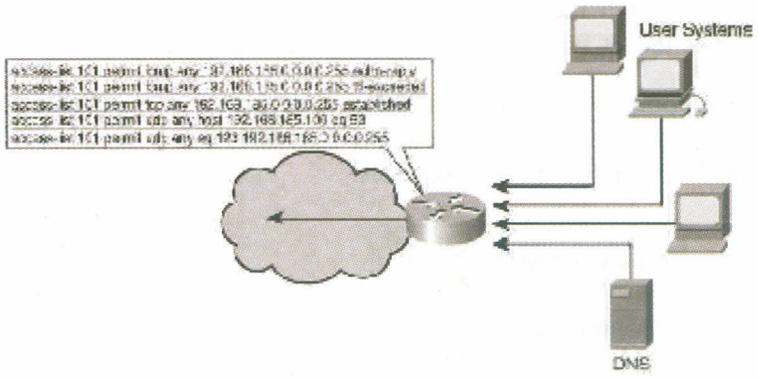


Fig. 13: Packet Filtering

- And additional options.

In packet filtering, protocol and address information in each packet is considered, this type of filtering pays no attention to the existing stream of packets. Instead, it filters depending on examining incoming or outgoing packets, it allows or deny the packets, relying on the acceptance policy in the configuration rules.

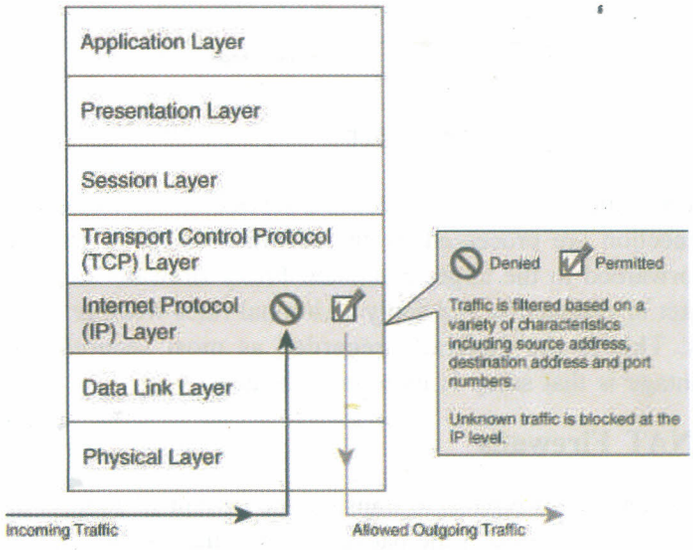


Fig. 14: Packet Filtering in OSI Model

Packet filtering firewall, operates at the IP layer of the protocol stack (as shown in Fig. 14). Traffic is filtered in this layer, based on the characteristics including source address, destination address and port numbers.

Filtering policies rely completely on allowing or disallowing the IP address, Port or Protocol.

A simple access list from example,
access-list 101 permit tcp any 192.168.10.1 0.0.0.255

The above access rule allows traffic from any system to the subnet 192.168.10.1/24

1.8.2 Application Layer Filtering

In this approach, firewall can understand the regulation of traffic further than packet filtering. This means that the firewall does not inspect headers alone but looks at the payload of the IP packet that contains a TCP segment within which it inspects the application layer data.

However, in this case of proxy firewall, the interaction is controlled at the application layer (as shown in Fig. 15). It forces both the end communications to be processed through the proxy. In the case of the application using TCP, the proxy becomes the TCP end-point for the systems at either end.

To provide support for multiple services at the proxy, it must be running with specific service for each protocol. Example like, SMTP for e-mail, HTTP for web services and FTP for file transfer services.

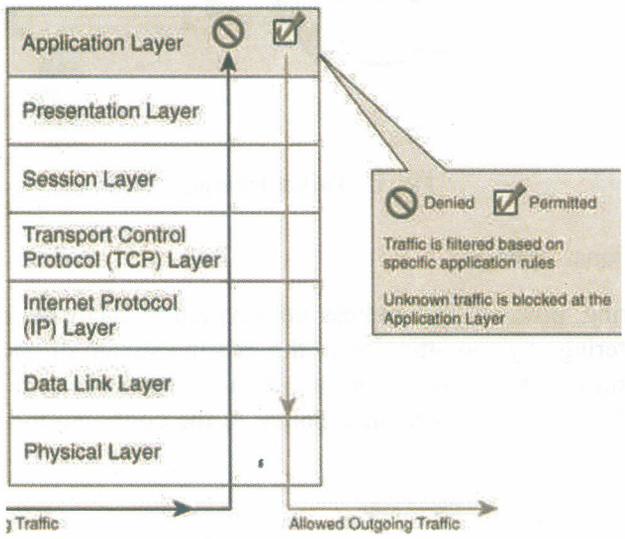


Fig. 15: Application Layer Filtering

Whenever a client wants to connect to a service on the internet, the packet making the connection are processed by the specific service at the proxy server before being forwarded to the target computer. Proxy firewalls can look more deep into the packet of connection and apply additional rules weather to forward or drop the packets. This is the generally regarded as most secured type of firewall. A disadvantage is that setup is more complicated.

1.8.3 NAT Firewalls

The basic purpose of NAT is to translate the current IP address to a new IP address at the firewall, to represent the packet receiver that as though it were coming from a single IP address. This prevents the attacker to know the original IP addresses in the network.

The NAT creates a table in memory that holds all these information of translation and connections. The ability of mapping the entire network behind a single ip is based on the port number assigned by NAT firewall (as shown in Fig. 16).

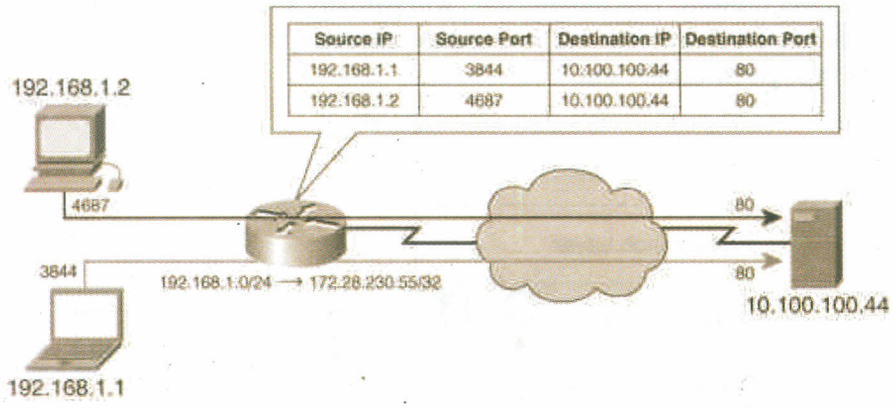


Fig. 16: NAT Firewall

Example of the NAT IP address:

Source IP	Source Port	NAT IP	NAT port	Destination IP	Destination Port
192.168.0.1	3144	172.28.230.55	3144	10.100.100.44	80

Here, when a packet is originated from source IP(192.168.0.1), NAT changes the source IP address to 172.28.230.55 in each packet and forwarded to destination IP. The destination IP can never trace the original source IP address.

The IP header will be changed as below,

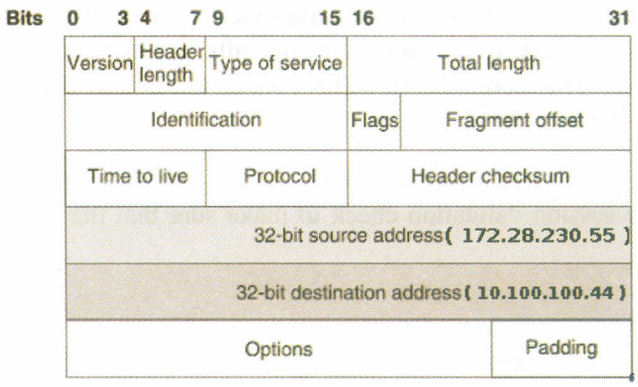


Fig. 17: Packet with Source IP

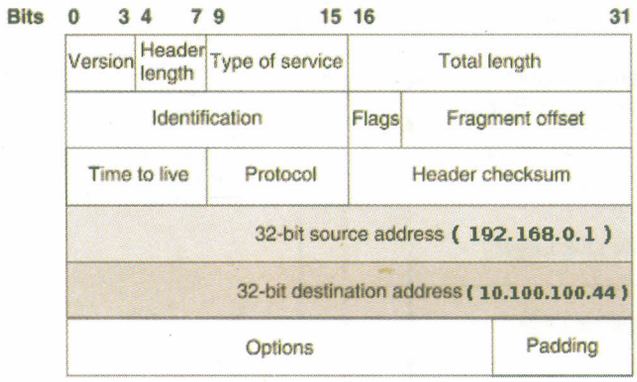


Fig. 18: Packet with NAT IP

1.8.4 Circuit Level Firewalls

Circuit level filtering works at the session layer of OSI model (as shown in Fig. 19). Traffic to the remote compute is made as though the traffic is originated from a circuit level firewall. This modification will partially allow to hide the information about the protected network but has a drawback that it does not filter individual packets in a given connection.

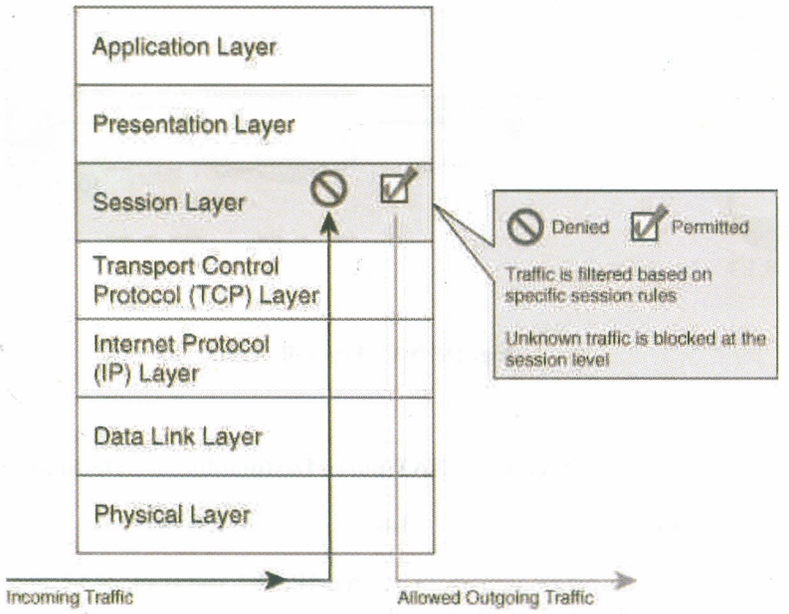


Fig. 19: Circuit level Firewall

1.8.5 Stateful Filtering

Stateful filtering are the most modern approach of firewall, it combines the capabilities of NAT firewalls, circuit level firewalls and application firewalls into a common system. This approach validates connection before allowing data to be transferred. See the Fig. 20.

These firewalls filters traffic initially with packet characteristics and rules and also includes the session validation check to make sure that the specific session is allowed.

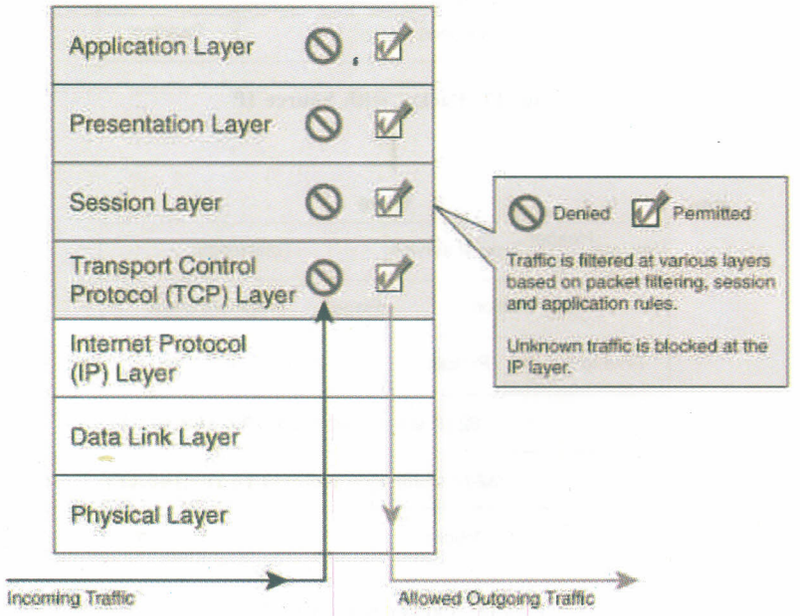


Fig. 20: Stateful Firewall

Stateless firewalls watch the traffic packet by packet and filter them based on individual rules. Each packet is individually checked and filtered. They do not atempt to corelate the packets that came before and then judge if there is a malicious potential or intention. However, it is necessary to watch a set of packets between a source and a destination to infer any malicious intent.

Stateful firewalls can watch traffic streams from end to end. They are are aware of communication paths. This implies that the firewall can identify flows. A flow table that provides the source and destination IP addresses is built dynamically in the firewall. The firewall then monitors packets pertaining to each flow in both directions and applies filtering rules. Since the firewall now tracks flows rather than individual packets, it can tell what stage a TCP connection is in (open, open sent, synchronized, synchronization acknowledge or established), it can tell if the MTU has changed, whether packets have fragmented etc. Flow monitoring enables the firewall to detect many packet sequences and patterns that could be malicious.

The firewall does not limit its packet inspection to the IP headers alone. It inspects the IP packet's payload part that contains the TCP segment and the application layer data. By doing this, it is able to analyse the application data (if the application data is not encrypted) and attempt to detect malware that may be present. Such a functionality is termed as Deep Packet Inspection (DPI). The firewall compares the application data with well known malware signatures and then filters the traffic.

Often, the firewall's capabilities are extended to functionally include Intrusion Detection.

Network and Host based Firewalls

There are two main types of firewalls: network firewalls and host-based firewalls.

Network Firewalls

Network firewalls protect an entire network by guarding the perimeter of that network. Network firewalls forward traffic to and from computers on an internal network and filter that traffic based on the criteria the administrator has set. Network firewalls come in two flavors: hardware firewalls and software firewalls. Hardware-based network firewalls are generally cheaper than software-based network firewalls and are the right choice for home users and many small businesses. Software-based network firewalls often have a larger feature set than hardware-based firewalls and might fit the needs of larger organizations. Software-based firewalls can also run on the same server as other services, such as e-mail and file sharing, allowing small organizations to make better use of existing servers. Network firewalls often include additional features that aren't necessary for host-based firewalls, as described in the following sections. Network firewalls, such as the software-based Microsoft's Internet Security and Acceleration (ISA) Server or the hardware-based Nortel Networks Alteon Switched Firewall System, protect the perimeter of a network by watching traffic that enters and leaves.

Host-Based Firewalls

Host-based firewalls are software firewalls installed on each individual system. Depending on the software you choose, a host-based firewall can offer features beyond those of network firewalls, such as protecting your computer from spyware (a component of some free software that tracks your Web browsing habits) and Trojan horses (a program that claims to do one thing, but does another, malicious thing, such as recording your passwords). If you travel with a laptop, a host-based firewall is a necessity-you need protection wherever you connect to the Internet and your hardware firewall can protect you only at home.

Why would you buy third-party firewall software when Windows XP includes ICF for free? ICF is designed to provide basic intrusion prevention, but doesn't include

the rich features of a third-party firewall application. Most third-party firewalls protect you from software that could violate your privacy or allow an attacker to misuse your computer-features not found in ICF. Also, you can install third-party firewall programs on systems that have older versions of Windows. Note that firewall software doesn't replace antivirus software. You should use both.

Popular host-based firewall products include ZoneAlarm, Tiny Personal Firewall, Agnitum Outpost Firewall, Kerio Personal Firewall and Internet Security Systems' BlackICE PC Protection. Most host-based firewall software is available in free or trial versions, so it won't cost you anything to download these packages and determine whether they meet your needs better than ICF.

Host-based firewalls, such as Internet Connection Firewall (ICF included with Windows XP and Windows Server 2003), protect an individual computer regardless of the network it's connected to.

1.9 IPTABLES IN LINUX

iptables is a stateless packet filter used to set up, maintain and configure the packet filter tables in the Linux kernel. It is installed by default in all the Linux distributions. It is also a powerful administration tool for packet filtering and NAT.

iptables can be used to configure custom rules to forward or drop or NAT an incoming packet. Rules are constructed using command line options (refer the manual page). A set of such rules can be included into a script which can be run at the time of starting up the host.

iptables require root previleges to operate and execute. Some examples of the command line are included as below:

To limit FTP access to hosts from specific subnets with source address ranges 14.122.128.0/24 and with source address range 144.33.28.0/24 and specifically deny ftp access to the host 192.168.1.6, the following are the rules required. We need to add a rule to retain the TCP connections that are already in progress when the other three rules are applied. The rules are defined using the following command lines

```
iptables -I FORWARD -p tcp -d 192.168.1.6 --dport 21 -j DROP

iptables -I FORWARD -p tcp -s 198.133.219.0/24 --dport 21 -j ACCEPT

iptables -I FORWARD -p tcp -s 69.147.64.0/18 --dport 21 -j ACCEPT

iptables -I FORWARD -p tcp --dport 21 -m state --state
RELATED,ESTABLISHED -j ACCEPT
```

Deny access to a specific Subnet

```
iptable s -I FORWARD -s 192.168.2.0/255.255.255.0 -j DROP
```

Deny access to a specific IP address range with Logging

```
iptables -I FORWARD -m iprange --src-range 192.168.1.10-192.168.1.13
-j logdrop
```

Block outgoing SMTP traffic except from specified hosts. Simple Mail Transfer Protocol operates on tcp port 25.

```
iptables -I FORWARD 1 -p tcp -s 192.168.1.2 --dport 25 -j ACCEPT

iptables -I FORWARD 2 -p tcp -s 192.168.1.1/24 --dport 25 -j REJECT
```

Check Your Progress 2

Note: a) Space is given below for writing your answer.

b) Compare your answer with the one given at the end of this Unit.

What are the functions of the firewalls?

.....

.....

.....

.....

.....

.....

.....

1.10 LET US SUM UP

In this unit, we have understood the origin of the term firewall. We have seen its connotation in the context of data network security. We have looked at the evolution of firewalls and how they have come about starting from the early implementations. They are broadly classified into stateful and stateless firewalls. Chronolgically, stateless firewalls came first and stateful firewalls followed. Stateful firewalls include the capability to monitor flows as well as perform deep packet inspection (DPI). DPI, along with flow based monitoring provides firewalls with a capability to inspect the application payload and look for malware signatures or for traffic anamolies. Finally, we have seen an example that illustrates a stateless firewall implementation, iptables, on Linux.

1.11 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

Security Requirements of the Network

Network security is preliminary act of organizations, enterprises and institutions to protect their valuable information across the network. And it involves the process of preventing and detecting unauthorized users using or accessing the communication channel.

To prevent the unknown users, we require to identify threats and implementation the set of tools to combat them. Computer network include many vulnerable threats, including

- Social engineering, wherein someone tries to gain access through social means (pretending to be a legitimate system user or administrator, tricking people into revealing secrets, etc.)
- War dialing, wherein someone uses computer software and a modem to search for desktop computers equipped with modems that answer, providing a potential path into a corporate network
- Denial-of-service attacks, including all types of attacks intended to overwhelm a computer or a network in such a way that legitimate users of the computer or network cannot use it

- Protocol-based attacks, which take advantage of known (or unknown) weaknesses in network services
- Host attacks, which attack vulnerabilities in particular computer operating systems or in how the system is set up and administered
- Password guessing
- Eavesdropping of all sorts, including stealing e-mail messages, files, passwords and other information over a network connection by listening in on the connection.

Firewalls can help protect against from some of these attacks, but not all. Network security includes the tools, that provide safety from all the above activities. They include

- antivirus software packages
- virtual private network
- secure network infrastructure
- encryption and
- security management

None of the tools alone can protect the entire security requirement, they must be layered together to achieve expected level of security.

Check Your Progress 2

Functions of the Firewalls

In computer networking, a firewall is a device with set of rules to permit or deny network access by unauthorized services. It is as similar to the originated fire wall in terms of functionality.

Many operating systems support software based firewall to deny access against the private internet. Software firewalls acts between network card drivers and operating system. The firewall must be positioned in the network to control all the incoming and outgoing traffic.

There are many ways of unsuspicious people use to access the unprotected computers over the network by remote login, SMTP session hijacking, operating system bugs, denial of service, viruses, spam and many more. Some of the attacks are hard and some of them are impossible to filter using a firewall. Example like, spam mails can not be stopped in a firewall as long as you accept e-mail.

The level of security you establish, will determine how many of the threats can be stopped. The highest level of security is achieved by simply blocking everything. Obviously, that is not the purpose of having a network with firewall.

So, firewall do in terms of security in the generations as below

- The most basic firewall performs Packet Filtering
- The second most performs Application Layer filtering
- The third most performs stateful filter or circuit filtering

The disadvantages in setting up the firewall is obvious, restricting the access to internal application from outside world. To maintain high level of security we need to allow services like DNS naming servers, e-mail forwarding, mirror repository or proxy services. Only selective services need to communicate with external network and restrict access to services (like NFS, NIS, finger, telnet, etc).

1.12 SUGGESTED READINGS

- Firewalls and Internet Security: www.cisco.com.
- http://codeidol.com/.
- http://www.techsoup.org/.
- http://www.wisc-online.com/objects/viewobjects.aspx?ID=C14105.
- Schneier. Digital Security in a Networked World, John Wiley & Sons, New York, NY, 188-193.

UNIT 2 IDS/IPS/HONEYPOTS

Structure

2.0 INTRODUCTION

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are

violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. Incidents have many causes, such as malware (e.g. worms, spyware), attackers gaining unauthorized access to systems from the Internet, and authorized users of systems who misuse their privileges or attempt to gain additional privileges for which they are not authorized.

An intrusion detection system (IDS) is software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of an intrusion detection system and can also attempt to stop possible incidents.

The basic difference between the two technologies lies in how they provide protection for network environments.

IDS, analyze network traffic and generate alerts when malicious activity is discovered. They are generally able to reset TCP connections by issuing specially crafted packets after an attack begins and some are even able to interface with firewall systems to re-write firewall rules gets on-the-fly.

IPS, perform the same analysis as Intrusion Detection Systems but, because they are inserted in-line, between other network components, they can preempt malicious activity. In contrast to IDS sensors, network traffic flows through an IPS sensor not past it so the IPS sensor can pull or drop traffic from the wire.

2.1 OBJECTIVES

After studying this unit you should be able to:

- identify uses and key functions of IDPS technologies;
- understand the various detection methods used by IDS;
- differentiate among different IDS systems viz. Network Based, Host-Based and Distributed IDS;
- understand various IPS techniques;
- understand other similar technologies such as HoneyPots;
- select an IDP System based on personal or organizational requirements and needs; and
- install and use Snort IDPS which is an open source IDPS for windows.

2.2 USES OF IDPS

IDPSs are primarily focused on identifying possible incidents. For example, an IDPS could detect when an attacker has successfully compromised a system by exploiting a vulnerability in the system. The IDPS could then report the incident to security administrators, who could quickly initiate incident response actions to minimize the damage caused by the incident. The IDPS could also log information that could be used by the incident handlers.

Many IDPSs can also identify reconnaissance activity, which may indicate that an attack is imminent. For example, some attack tools and forms of malware, particularly worms, perform reconnaissance activities such as host and port scans to identify targets for subsequent attacks. An IDPS might be able to block reconnaissance and notify security administrators, who can take actions if needed to alter other security controls to prevent related incidents.

In addition to identifying incidents and supporting incident response efforts, organizations have found other uses for IDPSs, including the following:

- **Identifying security policy problems**. An IDPS can provide some degree of quality control for security policy implementation, such as duplicating firewall rulesets and alerting when it sees network traffic that should have been blocked by the firewall but was not because of a firewall configuration error.
- **Documenting the existing threat to an organization**. IDPSs log information about the threats that they detect. Understanding the frequency and characteristics of attacks against an organization's computing resources is helpful in identifying the appropriate security measures for protecting the resources. The information can also be used to educate management about the threats that the organization faces.
- **Deterring individuals from violating security policies**. If individuals are aware that their actions are being monitored by IDPS technologies for security policy violations, they may be less likely to commit such violations because of the risk of detection.

2.3 INTRUSION DETECTION SYSTEM (IDS)

2.3.1 Functions of IDS

An Intrusion Detection System has three actions:

- Obtain audit data from the system monitored
  - Network traffic
  - System logs
  - System properties (load average, file use, login times, etc)

Information is usually recorded locally, and might also be sent to separate systems such as centralized logging servers, security information and event management (SIEM) solutions, and enterprise management systems.

- Analyze the data, searching for evidence of attack
- Report any attack evidence to a human operator ('alert')

This notification, known as an alert, occurs through any of several methods, including the following: e-mails, pages, messages on the IDPS user interface, Simple Network Management Protocol (SNMP) traps, syslog messages, and

user-defined programs and scripts. A notification message typically includes only basic information regarding an event; administrators need to access the IDPS for additional information.

2.3.2 Common Detection Methods

IDS technologies use many methodologies to detect incidents. Most IDPS technologies use multiple detection methodologies, either separately or integrated, to provide more broad and accurate detection.

2.3.2.1 Signature Based Detection

A signature is a pattern that corresponds to a known threat. Signature-based detection is the process of comparing signatures against observed events to identify possible incidents.

Examples of signatures are as follows:

- A telnet attempt with a username of "root", which is a violation of an organization's security policy
- An e-mail with a subject of "Free pictures!" and an attachment filename of "freepics.exe", which are characteristics of a known form of malware
- An operating system log entry with a status code value of 645, which indicates that the host's auditing has been disabled.

Signature-based detection is very effective at detecting known threats but largely ineffective at detecting previously unknown threats, threats disguised by the use of evasion techniques, and many variants of known threats. For example, if an attacker modified the malware in the previous example to use a filename of "freepics2.exe", a signature looking for "freepics.exe" would not match it.

Pros

- Fast (given a small rule set)
- Rules are customizable and easy to write
- Easily identify intrusion

Cons

- Weak against newly-discovered vulnerabilities
- Requires constant updating to be effective

Also known as misuse detection.

2.3.2.2 Anomaly Based Detection

Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations. An IDPS using anomaly-based detection has profiles that represent the normal behavior of such things as users, hosts, network connections, or applications. The profiles are developed by monitoring the characteristics of typical activity over a period of time. For example, a profile for a network might show that Web activity comprises an average of 13% of network bandwidth at the Internet border during typical workday hours. The IDPS then uses statistical methods to compare the characteristics of current activity to thresholds related to the profile, such as detecting when Web activity comprises significantly more bandwidth than expected and alerting an administrator of the anomaly. Profiles can be developed for many behavioral attributes, such as the number of e-mails sent by a user, the number of failed login attempts for a host, and the level of processor usage for a host in a given period of time.

Pros

- Customizable to system
- More flexible than rule-based; can detect new types of intrusions
- Low chance of missing an intrusion

Cons

- Requires complete profile of system behavior
- High chance of false alarms
- Must be updated if system changes

The major benefit of anomaly-based detection methods is that they can be very effective at detecting previously unknown threats.

2.3.2.3 Stateful Protocol Analysis

Stateful protocol analysis is the process of comparing predetermined profiles of generally accepted definitions of benign protocol activity for each protocol state against observed events to identify deviations. Unlike anomaly-based detection, which uses host or network-specific profiles, stateful protocol analysis relies on vendor-developed universal profiles that specify how particular protocols should and should not be used.

The "stateful" in stateful protocol analysis means that the IDS is capable of understanding and tracking the state of network, transport, and application protocols that have a notion of state. For example, when a user starts a File Transfer Protocol (FTP) session, the session is initially in the unauthenticated state. Unauthenticated users should only perform a few commands in this state, such as viewing help information or providing usernames and passwords.

An important part of understanding state is pairing requests with responses, so when an FTP authentication attempt occurs, the IDPS can determine if it was successful by finding the status code in the corresponding response. Once the user has authenticated successfully, the session is in the authenticated state, and users are expected to perform any of several dozen commands. Performing most of these commands while in the unauthenticated state would be considered suspicious, but in the authenticated state performing most of them is considered benign.

The "protocol analysis" performed by stateful protocol analysis methods usually includes reasonableness checks for individual commands, such as minimum and maximum lengths for arguments. If a command typically has a username argument, and usernames have a maximum length of 20 characters, then an argument with a length of 1000 characters is suspicious. If the large argument contains binary data, then it is even more suspicious.

Pros

- More flexible than rule-based; can detect new types of intrusions

Cons

- Assumes that all applications adhere to protocol standards
- Extensive and difficult to write
- Will miss intrusions that do not violate protocol

2.3.2.4 Specification Based Detection

IDS is given a defined description (specification) of the normal system state, very similar to protocol based detection.

Alarm is raised if system deviates from specification.

Example: Raise alarm if CPU load on a host goes higher than value in specification

Pros

- Customizable to system
- More flexible than rule-based; can detect new types of intrusions
- Low chance of missing an intrusion

Cons

- Requires syntax for description of system
- Requires complete knowledge of system behavior
- High chance of false alarms
- Must be updated if system changes

2.3.2.5 Policy-Based Detection

IDS has list of predefined acceptable actions (policy)

Alarm is raised if policy is violated

Example: Raise alarm if a GET request to a specific HTTP server is for any file other

than those allowed to be viewed by users

Pros

- Customizable to system
- More flexible than rule-based; can detect new types of intrusions

Cons

- Requires complete policy of system
- Must be updated if policy changes

2.3.3 Types of Intrusion Detection System (IDS)

2.3.3.1 Network Based IDS

Network-Based, which monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity. It can identify many different types of events of interest. It is most commonly deployed at a boundary between networks, such as in proximity to border firewalls or routers, virtual private network (VPN) servers, remote access servers, and wireless networks.

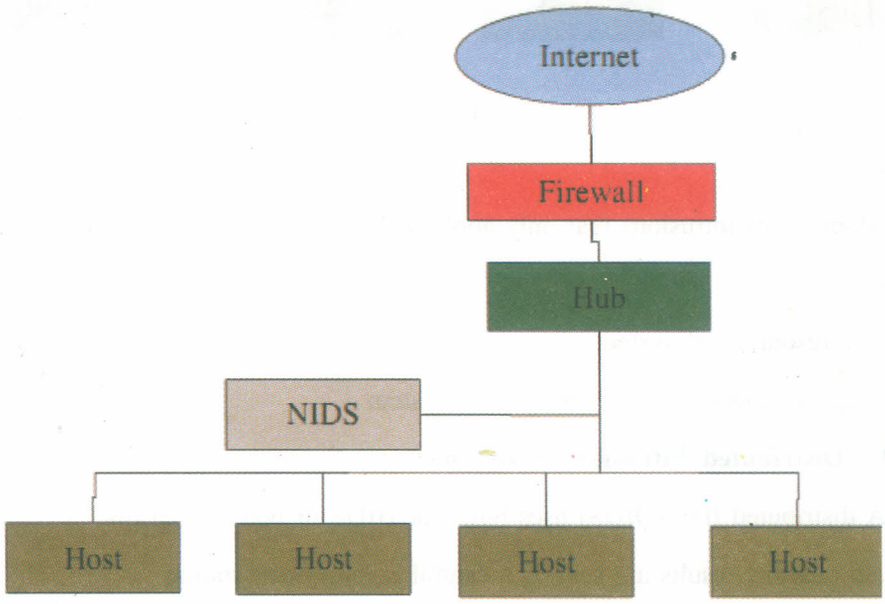


Fig. 1

Pros

- Does not affect network performance
- Can be hidden

Cons

- Weak against denial of service attack
- Single point of failure

2.3.3.2 Host-Based

Host based IDS monitors the characteristics of a single host and the events occurring within that host for suspicious activity. Examples of the types of characteristics a host-based IDPS might monitor are network traffic (only for that host), system logs, running processes, application activity, file access and modification, and system and application configuration changes. Host-based IDPSs are most commonly deployed on critical hosts such as publicly accessible servers and servers containing sensitive information.

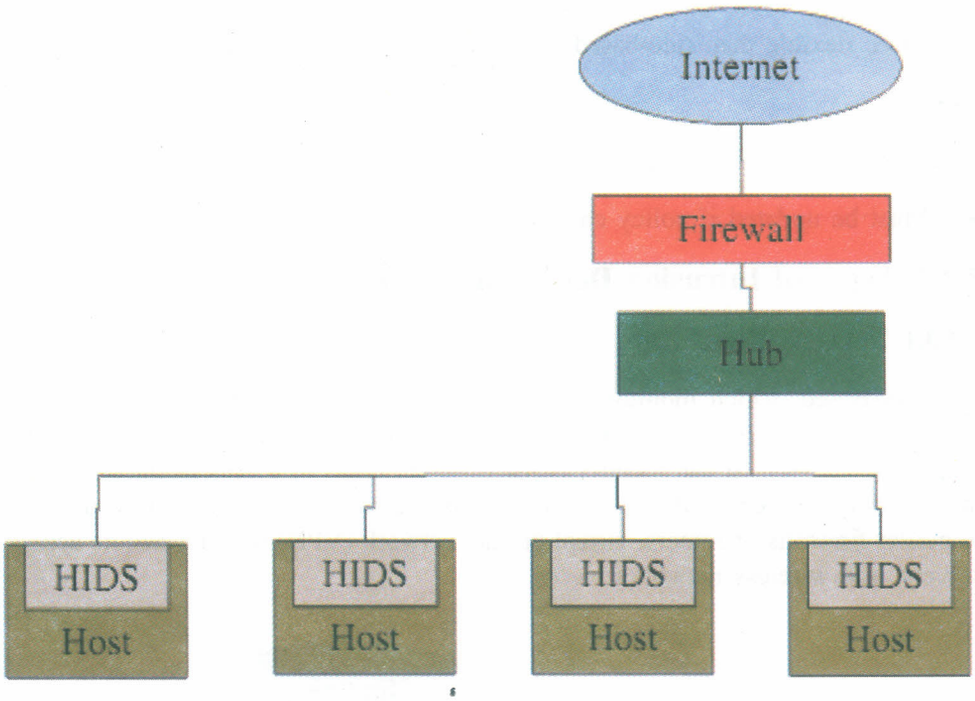


Fig. 2

Pros

- Monitor for intrusions that only apply to host

Cons

- Use resources of system
- Requires specific HIDS for specific system

2.3.3.3 Distributed Intrusion Detection

- A distributed IDS (DIDS) uses NIDS or HIDS or both as sensors
- All analysis results are sent to a central management station
- Also known as a hybrid IDS

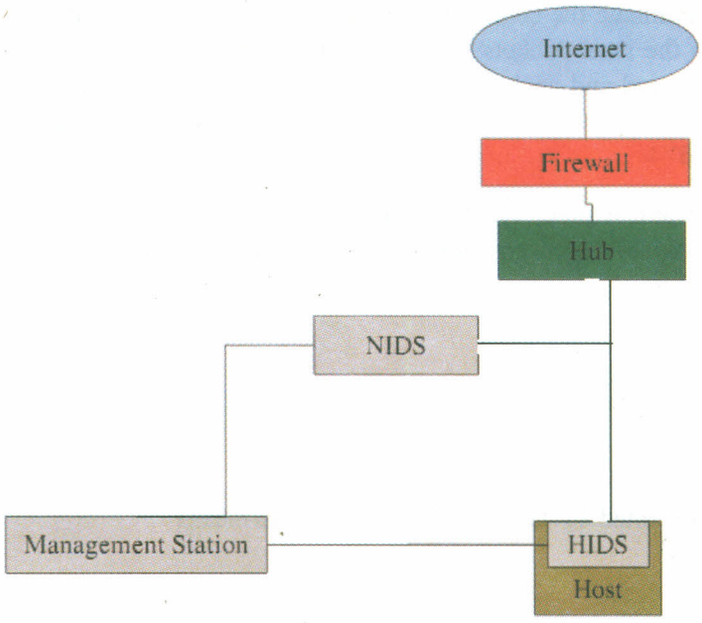


Fig. 3

Pros

- Widest in range
- Sensors can share and coordinate results

Cons

- Results from sensors are sent across the [potentially compromised
- Results from sensors may not be sent if network is flooded

Examples of DIDS

Prelude – rule-based packet sniffer, log analyzer, buffer overflow detection

Osiris – file integrity monitor

2.4 INTRUSION PREVENTION SYSTEM (IPS)

Many network intrusions take little time to execute

- Worms
- Backdoor exploits

An IDS often does not provide enough time for an administrator to respond before the intrusion has damaged the system

An Intrusion Prevention System (IPS) is designed to identify potential attacks and autonomously execute countermeasures to inhibit them, without affecting normal system operation.

IPS is the 'next step' following IDS.

2.4.1 Different Countermeasures taken by IPS (Network Based)

2.4.1.1 Session Sniping

A passive sensor can attempt to end an existing TCP session by sending TCP reset packets to both endpoints. The sensor does this to make it appear to each endpoint that the other endpoint is trying to end the connection. The goal is for one of the

endpoints to terminate the connection before an attack can succeed. Unfortunately, in many cases the reset packets are not received in time because the attack traffic has to be monitored and analyzed, the attack detected, and the packets sent across networks to the endpoints. Also, since this technique is only applicable to TCP, it cannot be used

for attacks carried in other types of packets, including UDP and ICMP. Session sniping is not widely used anymore because other, newer prevention capabilities are more effective.

Pros

- Easy to implement and use

Cons

- Can be prevented by attacker through various means

Examples

- IPtables (Linux firewall) can REJECT traffic from a given host using this Method.
- Snort has a 'flexible response' option, which allows it to send resets when a rule is triggered.

2.4.1.2 Packet Filtering

Most inline IPS sensors offer firewall capabilities that can be used to drop or reject suspicious network activity.

Usually implemented as an Inline NIDS

- NIDS positioned inline on network and acts as a bridge between subnets
- NIDS redesigned to drop malicious traffic.

Disruption of Inline NIDS will disrupt network. Also deployed as 'smart' switches.

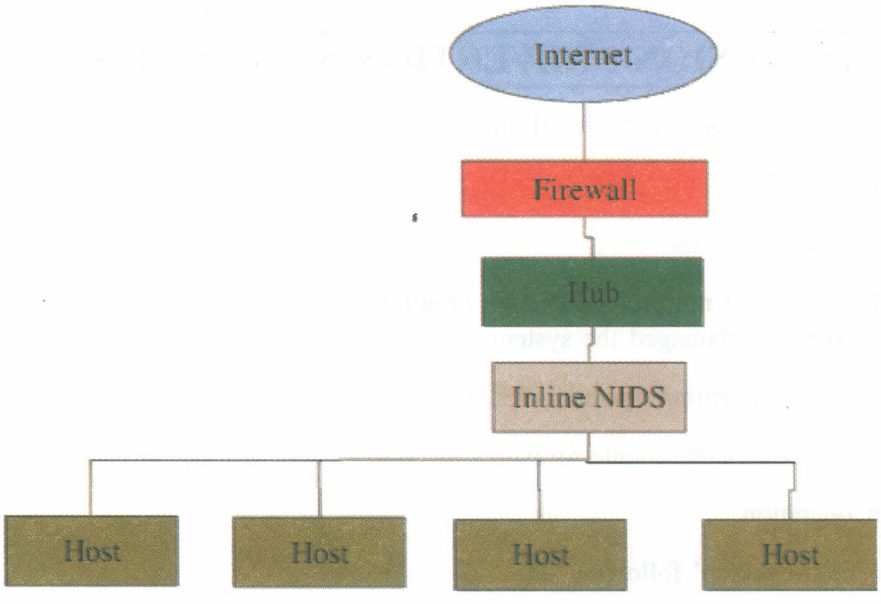


Fig. 4

Pros

- Removes malicious traffic from the network

Cons

- Possibility of removing legitimate traffic

Examples

- Hogwash drops traffic when a rule is triggered
- Latest version of Snort has an 'inline' feature that allows it to drop traffic when a rule is triggered.

2.4.1.3 Packet Scrubbing

Some inline IPS sensors can sanitize part of a packet, which means that malicious content is replaced with benign content and the sanitized packet sent to its destination. A sensor that acts as a proxy might perform automatic normalization of all traffic, such as repackaging application payloads in new packets. This has the effect of sanitizing some attacks involving packet headers and some application headers, whether or not the IDPS has detected an attack. Some sensors can also strip infected attachments from e-mails and remove other discrete pieces of malicious content from network traffic.

Pros

- Does not indicate that IPS detected intrusion

Cons

- Only works if the intrusion involves malicious packet content

Example

- Snort has a 'replace' option that changes the malicious content to user-defined content of the same length

2.4.1.4 IP Blocking

Many IDPS sensors can instruct network security devices such as firewalls, routers, and switches to reconfigure themselves to block certain types of activity or route it elsewhere. This can be helpful in several situations, such as keeping an external attacker out of a network and quarantining an internal host that has been compromised (e.g. moving it to a quarantine VLAN). This prevention technique is useful only

for network traffic that can be differentiated by packet header characteristics typically recognized by network security devices, such as IP addresses and port numbers.

Pros

- Effectively removes attacker from the network

Cons

- Possibility of blocking legitimate hosts

Example

- SnortSAM firewall agent receives information from Snort rules and changes settings of firewalls (such as IPtables) accordingly

2.4.1.5 Deception

The IPS sends traffic to the attacker indicating that the attack against a host succeeded or that the connection to the host is no longer available Usually a

dedicated host (a 'honeypot') on the network purposely exhibits vulnerabilities, drawing the attacker away from the rest of the network can also fake service applications on the hosts.

Pros

- Attacker proceeds with attack; obtain information about attack and attacker

Cons

- Attacker is still on the network; may be able to launch attacks on other hosts
- Possible legal implications

Examples

- The Deception Toolkit (DTK) simulates services on a host
- HoneyPots creates virtual hosts for the attacker to attack

2.4.2 Different Countermeasures taken by IPS (Host Based)

Host-based IPS agents offer various intrusion prevention capabilities. Because the capabilities vary based on the detection techniques used by each product, the following items describe the capabilities by detection technique.

2.4.2.1 Code Analysis

The code analysis techniques can prevent code from being executed, including

malware and unauthorized applications. Some host-based IDPSs can also stop network applications from invoking shells, which could be used to attempt to perform certain types of attacks. If configured and tuned well, code analysis can be very effective, particularly at stopping previously unknown attacks.

2.4.2.2 Network Traffic Analysis

This can stop incoming network traffic from being processed by the host and outgoing network traffic from exiting it. This might be done to stop network, transport, and

application layer attacks (and in some cases, wireless networking protocol attacks), as well as to stop the use of unauthorized applications and protocols. Analysis can also identify malicious files being downloaded or transferred and prevent those files from being placed on the host. The network traffic might be dropped or rejected, and the host's personal firewall (which might be built into the agent) could be reconfigured to shun additional traffic related to the suspicious traffic. Network traffic analysis is effective at stopping many known and previously unknown attacks.

2.4.2.3 Network Traffic Filtering

Working as a host-based firewall, this can stop unauthorized access and acceptable use policy violations (e.g. use of inappropriate external services). It is effective only

against stopping activity that is identifiable by IP address and TCP port, UDP port, or ICMP type and code.

2.4.2.4 File System Monitoring

This can prevent files from being accessed, modified, replaced, or deleted, which could stop malware installation, including Trojan horses and rootkits, as well as other attacks involving inappropriate file access. This technique can provide an

additional layer of access control to supplement the existing access control technologies on a host.

Other host-based IDPS detection techniques, such as log analysis, network configuration monitoring, and file integrity and attribute checking, generally do not support prevention actions because they identify events well after they have occurred.

Check Your Progress 1

Note: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of the Unit.

1) What detection method would you use and why, if the type of attack is unknown?

.....

.....

.....

.....

.....

2) If you have a LAN system consisting of 2-3 terminals, which IDS type would you employ and why ?

.....

.....

.....

.....

.....

.....

3) What are the different Preventive Countermeasures employed by IPS?

.....

.....

.....

.....

2.5 HONEYPOTS

A honeypot is a trap set to detect, deflect, or in some manner counteract attempts at unauthorized use of information systems. Generally it consists of a computer, data, or a network site that appears to be part of a network, but is actually isolated and monitored, and which seems to contain information or a resource of value to attackers.

Using Honeypots may attract legal issues.

A honeypot is valuable as a surveillance and early-warning tool. While it is often a computer, a honeypot can take other forms, such as files or data records, or even unused IP address space. A honeypot that masquerades as an open proxy to monitor

and record those using the system is known as a "sugarcane". Honeypots should have no production value, and hence should not see any legitimate traffic or activity. Whatever they capture is therefore malicious or unauthorized. One practical application of this is the spamtrap - a honeypot that thwarts spam by masquerading as a type of system abused by spammers. These honeypots categorize trapped material 100% accurately: it is all illicit.

Honeypots can carry risks to a network, and must be handled with care. If they are not properly walled off, an attacker can use them to break into a system.

Victim hosts are an active network counter-intrusion tool. These computers run special software, designed to appear to an intruder as being important and worth looking into. In reality, these programs are dummies, and their patterns are constructed specifically to foster interest in attackers. The software installed on, and run by, victim hosts is dual purpose. First, these dummy programs keep a network intruder occupied looking for valuable information where none exists, effectively convincing an intruder to isolate themselves in what is truly an unimportant part of the network. This decoy strategy is designed to keep an intruder from getting bored and heading into truly security-critical systems. The second part of the victim host strategy is intelligence gathering. Once an intruder has broken into the victim host, the machine or a network administrator can examine the intrusion methods used by the intruder. This intelligence can be used to build specific countermeasures to intrusion techniques, making truly important systems on the network less vulnerable to intrusion.

2.5.1 Types of Honeypots

Honeypots come in many shapes and sizes, making them difficult to get a grasp of. To help us better understand honeypots and all the different types, we break them down into two general categories, low-interaction and high-interaction honeypots. These categories helps us understand what type of honeypot you are dealing with, its strengths, and weaknesses. Interaction defines the level of activity a honeypot allows an attacker. Low-interaction honeypots have limited interaction, they normally work by emulating services and operating systems. Attacker activity is limited to the level of emulation by the honeypot. For example, an emulated FTP service listening on port 21 may just emulate a FTP login, or it may support a variety of additional FTP commands. The advantages of a low-interaction honeypot is their simplicity. These honeypots tend to be easier to deploy and maintain, with minimal risk. Usually they involve installing software, selecting the operating systems and services you want to emulate and monitor, and letting the honeypot go from there. This plug and play approach makes deploying them very easy for most organizations. Also, the emulated services mitigate risk by containing the attacker's activity, the attacker never has access to an operating system to attack or harm others. The main disadvantages with low interaction honeypots is that they log only limited information and are designed to capture known activity. The emulated services can only do so much. Also, its easier for an attacker to detect a low-interaction honeypot, no matter how good the emulation is, skilled attacker can eventually detect their presence. Examples of low-interaction honeypots include Specter, Honeyd, and KFSensor.

High-interaction honeypots are different, they are usually complex solutions as they involve real operating systems and applications. Nothing is emulated, we give attackers the real thing. If you want a Linux honeypot running an FTP server, you build a real Linux system running a real FTP server. The advantages with such a solution are two fold. First, you can capture extensive amounts of information. By giving attackers real systems to interact with, you can learn the full extent of their behavior, everything from new rootkits to international IRC sessions. The second advantage is high-interaction honeypots make no assumptions on how an attacker will behave. Instead, they provide an open environment that

captures all activity. This allows high-interaction solutions to learn behavior we would not expect. An excellent example of this is how a Honeynet captured encoded back door commands on a non-standard IP protocol (specifically IP protocol 11, Network Voice Protocol). However, this also increases the risk of the honeypot as attackers can use these real operating system to attack non-honeypot systems. As result, additional technologies have to be implement that prevent the attacker from harming other non-honeypot systems. In general, high-interaction honeypots can do everything low-interaction honeypots can do and much more. However, they can be more complext to deploy and maintain. Examples of high-interaction honeypots include Symantec Decoy Server and Honeynets. You can find a complete listing of both low and high interaction honeypots at Honeypot Solutions page. To better understand both low and high interaction honeypots lets look at two examples. We will start with the low-interaction honeypot Honeyd.

2.5.1.1 Honeyd: Low-Interaction Honeypot

Honeyd is a low-interaction honeypot. Developed by Niels Provos, Honeyd is OpenSource and designed to run primarily on Unix systems (though it has been ported to Windows). Honeyd works on the concept of monitoring unused IP space. Anytime it sees a connection attempt to an unused IP, it intercepts the connection and then interacts with the attacker, pretending to be the victim.

By default, Honeyd detects and logs any connection to any UDP or TCP port. In addition, you can configure emulated services to monitor specific ports, such as an emulated FTP server monitoring TCP port 21. When an attacker connects to the emulated service, not only does the honeypot detect and log the activity, but it captures all of the attacker's interaction with the emulated service. In the case of the emulated FTP server, we can potentially capture the attacker's login and password, the commands they issue, and perhaps even learn what they are looking for or their identity.

It all depends on the level of emulation by the honeypot. Most emulated services work the same way. They expect a specific type of behavior, and then are programmed to react in a predetermined way. If attack A does this, then react this way. If attack B does this, then respond this way. The limitation is if the attacker does something that the emulation does not expect, then it does not know how to respond. Most low-interaction honeypots, including Honeyd, simply generate an error message.

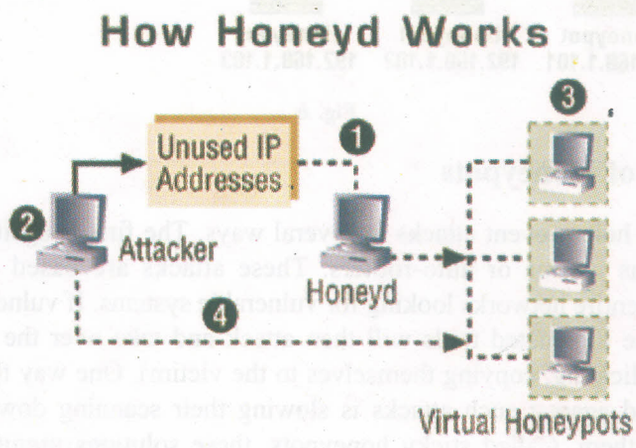


Fig. 5

2.5.1.2 Honeynets: High Interaction Honeypot

Honeynets are a prime example of high-interaction honeypot. Honeynets are not a product, they are not a software solution that you install on a computer. Instead, Honeyents are an architecture, an entire network of computers designed to attacked.

The idea is to have an architecture that creates a highly controlled network, one where all activity is controlled and captured. Within this network we place our intended victims, real computers running real applications. The bad guys find, attack, and break into these systems on their own initiative. When they do, they do not realize they are within a Honeynet. All of their activity, from encrypted SSH sessions to emails and files uploads, are captured without them knowing it.

This is done by inserting kernel modules on the victim systems that capture all of the attacker's actions. At the same time, the Honeynet controls the attacker's activity. Honeynets do this using a Honeywall gateway. This gateway allows inbound traffic to the victim systems, but controls the outbound traffic using intrusion prevention technologies. This gives the attacker the flexibility to interact with the victim systems, but prevents the attacker from harming other non-Honeynet computers.

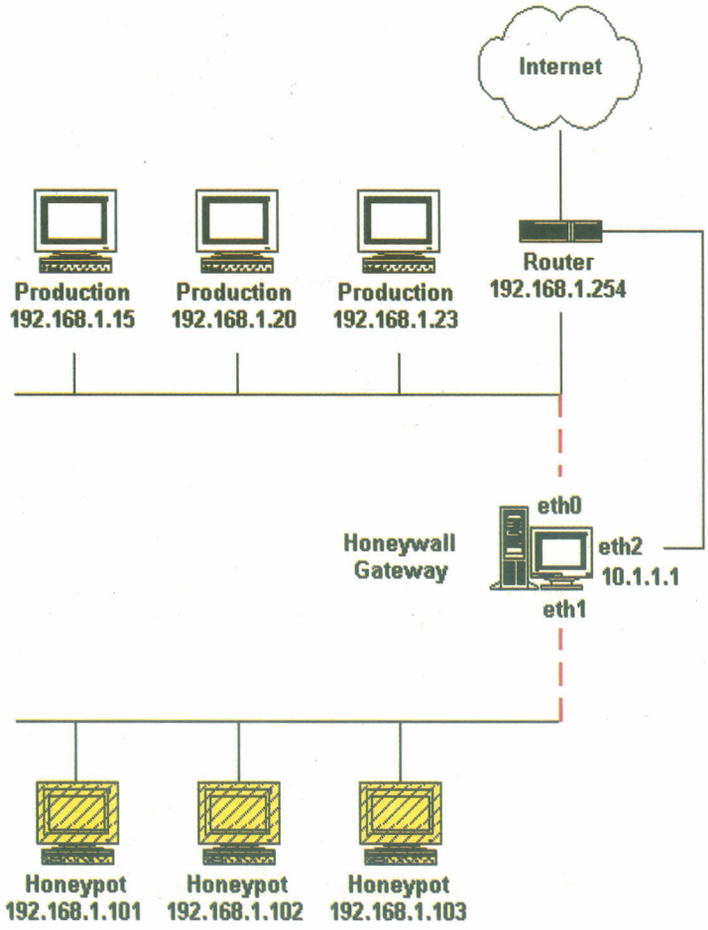


Fig. 6

2.5.2 Value of Honeypots

Honeypots can help prevent attacks in several ways. The first is against automated attacks, such as worms or auto-rooters. These attacks are based on tools that randomly scan entire networks looking for vulnerable systems. If vulnerable systems are found, these automated tools will then attack and take over the system (with worms self-replicating, copying themselves to the victim). One way that honeypots can help defend against such attacks is slowing their scanning down, potentially even stopping them. Called sticky honeypots, these solutions monitor unused IP space. When probed by such scanning activity, these honeypots interact with and slow the attacker down.

The second way honeypots can help protect an organization is through detection. Detection is critical, its purpose is to identify a failure or breakdown in prevention. Regardless of how secure an organization is, there will always be failures, if for no other reasons then humans are involved in the process. By detecting an attacker,

you can quickly react to them, stopping or mitigating the damage they do. Tradtionally, detection has proven extremely difficult to do.

The third and final way a honeypot can help protect an organization is in reponse. Once an organization has detected a failure, how do they respond? This can often be one of the greatest challenges an organization faces. There is often little information on who the attacker is, how they got in, or how much damage they have done. In these situations detailed information on the attacker's activity are critical. There are two problems compounding incidence response.

2.5.3 Advantages and Disadvantages of Honeypots

Advantages: Honeypots are a tremendously simply concept, which gives them some very powerful strengths.

- *Small data sets of high value:* Honeypots collect small amounts of information. Instead of logging a one GB of data a day, they can log only one MB of data a day. Instead of generating 10,000 alerts a day, they can generate only 10 alerts a day. Remember, honeypots only capture bad activity, any interaction with a honeypot is most likely unauthorized or malicious activity. As such, honeypots reduce 'noise' by collectin only small data sets, but information of high value, as it is only the bad guys. This means its much easier (and cheaper) to analyze the data a honeypot collects and derive value from it.
- *New tools and tactics:* Honeypots are designed to capture anything thrown at them, including tools or tactics never seen before.
- Minimal resources: Honeypots require minimal resources, they only capture bad activity. This means an old Pentium computer with 128MB of RAM can easily handle an entire class B network sitting off an OC-12 network.
- *Encryption or IPv6:* Unlike most security technologies (such as IDS systems) honeypots work fine in encrypted or IPv6 environments. It does not matter what the bad guys throw at a honeypot, the honeypot will detect and capture it.
- *Information:* Honeypots can collect in-depth information that few, if any other technologies can match.
- *Simplicty:* Finally, honeypots are conceptually very simple. There are no fancy algorithms to develop, state tables to maintain, or signatures to update. The simpler a technology, the less likely there will be mistakes or misconfigurations.

Disadvantages: Like any technology, honeypots also have their weaknesses. It is because of this they do not replace any current technology, but work with existing technologies.

- *Limited view:* Honeypots can only track and capture activity that directly interacts with them. Honeypots will not capture attacks against other systems, unless the attacker or threat interacts with the honeypots also.
- *Risk:* All security technologies have risk. Firewalls have risk of being penetrated, encryption has the risk of being broken, IDS sensors have the risk of failing to detect attacks. Honeypots are no different, they have risk also. Specifically, honeypots have the risk of being taken over by the bad guy and being used to harm other systems.

2.6 HANDS-ON, SNORT AN IPS

Snort is an open source intrusion detection/prevention system created by Martin "Marty" Roesch, founder of Sourcefire. It is capable of performing real-time traffic analysis and logging. It is the most widely used IDS/IPS system. It can monitor

for, detect and respond to various attack strategies by using signature, protocol and anomaly-based inspection techniques.

2.6.1 Installing and Using Snort

Snort: Snort's official web site is: http://www.snort.org. The site has links to the tools we will need to get snort up and running. there are core components that have been ported to Windows platforms. We will be installing version 2.8.5.1, which is the current stable version. The binary needed to install Snort can be found in the downloads section of the website, or directly at http://dl.snort.org/snort-current/Snort_2_8_5_1_Installer.exe.

WinPCap: WinPCap is a third party library that is REQUIRED by Snort. Installation is easy and straight forward. The installer for WinPCap can be downloaded from the WinPCap website at http://www.winpcap.org/install/default.htm. We will be installing version 4.1.1, which is the current stable version.

First Install WinPcap.

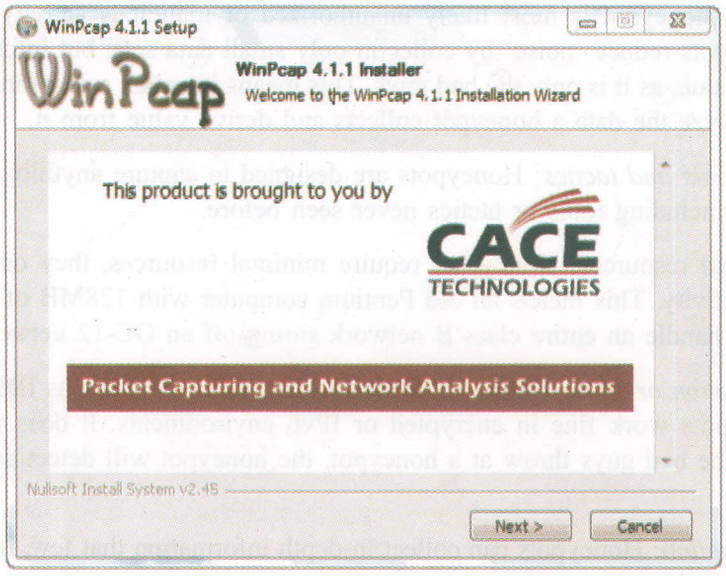


Fig. 7

Click Next and Agree the License agreement to bring this screen.

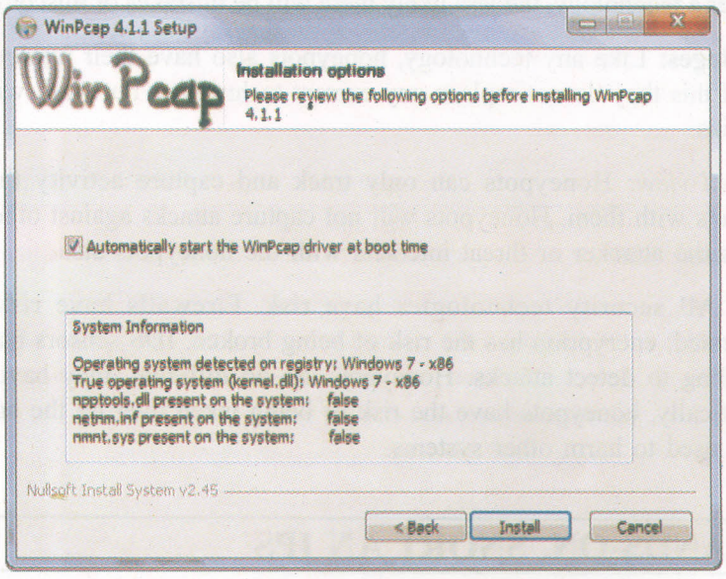


Fig. 8

Click Install.

Click "Finish" to exit the WinPCap setup application. You will need to reboot your computer at this time.

To install Snort, navigate to the location of the Snort Installer file. Right click the file and select "Run as Administrator". As we did with WinPCap, we will also be installing Snort with all the default settings. The setup application will launch and prompt you to read and agree to the License Agreement.

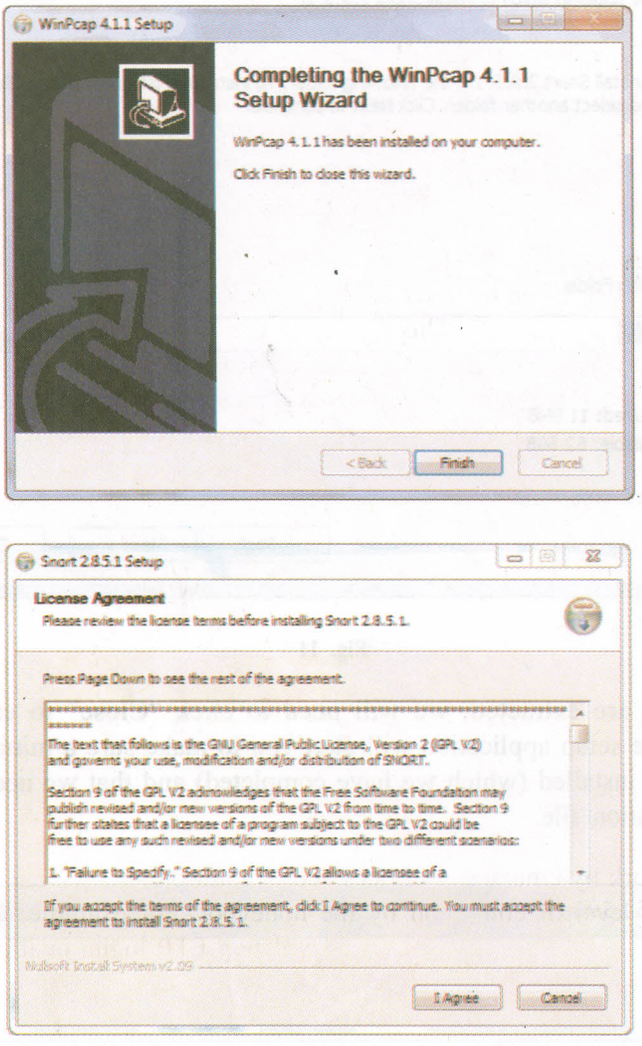


Fig. 9

Select the first radio button as seen above and click "Next" to select the Snort components to install.

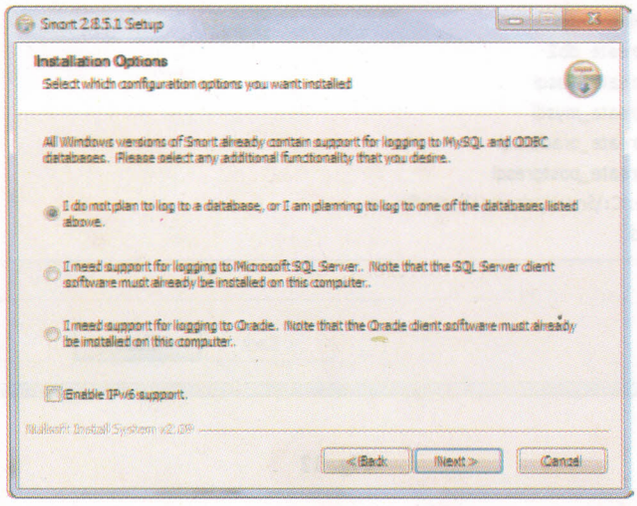


Fig. 10

For our install, let's take the default location of "c:\snort" and click "Next". At this point we have given the setup application all of the information it needs to extract the files necessary for our installation.

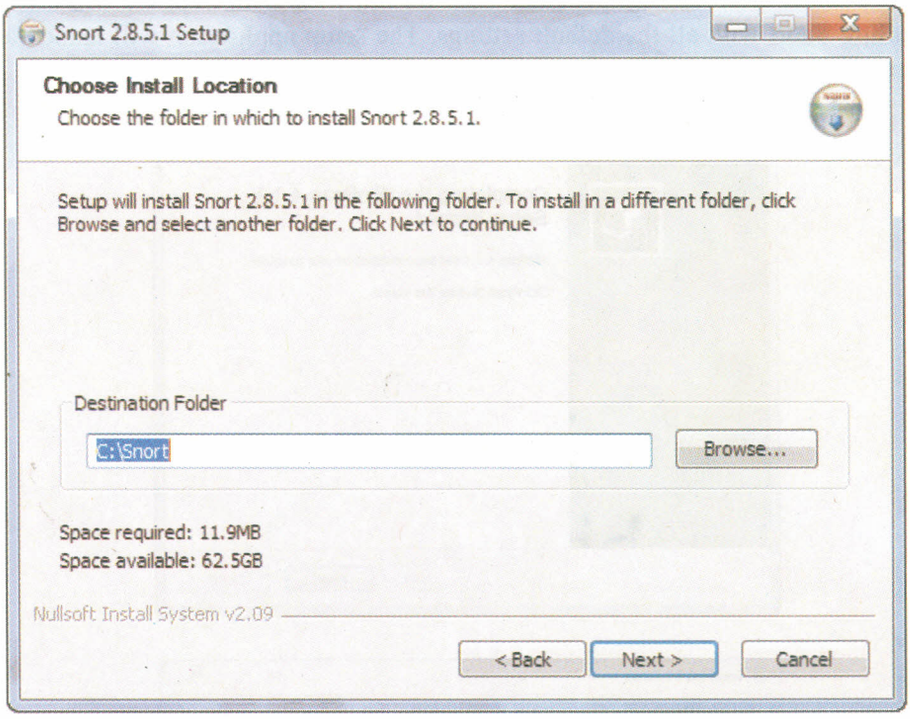


Fig. 11

Once the files are extracted, we will need to click "Close" to exit the setup application. The setup application will alert you to make sure a minimum version of WinPCap is installed (which we have completed) and that we need to edit the Snort configuration file.

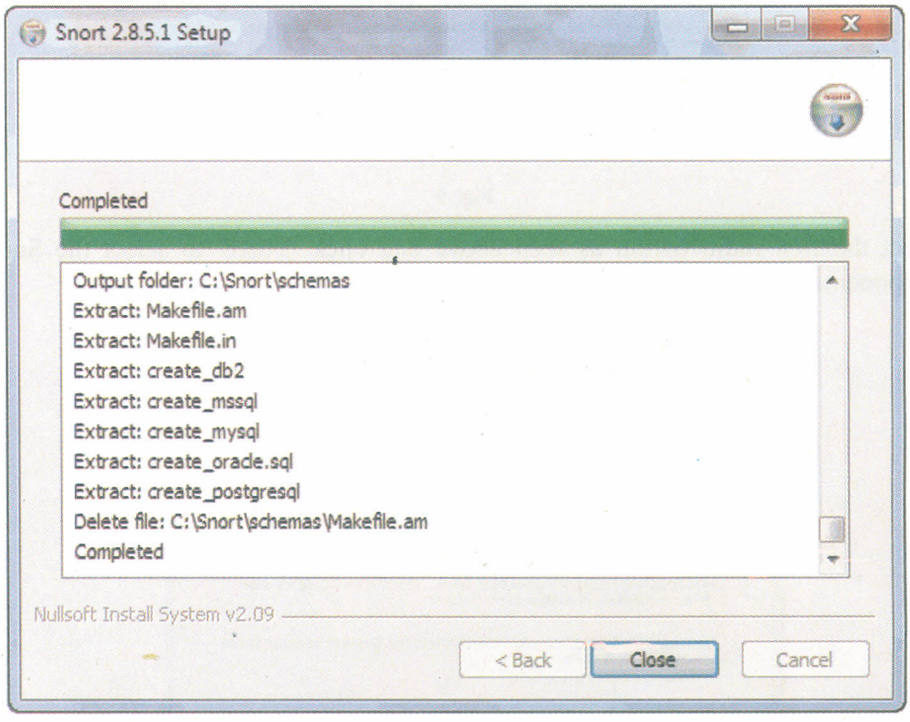


Fig. 12

Click "OK" to acknowledge this and close the setup application.

As we were told by the Snort setup application, we will need to change a couple of parameters in the c:\snort\etc\snort.conf file. To do so, let's use Microsoft's Wordpad application. Open the snort.conf file and find the lines highlighted below:

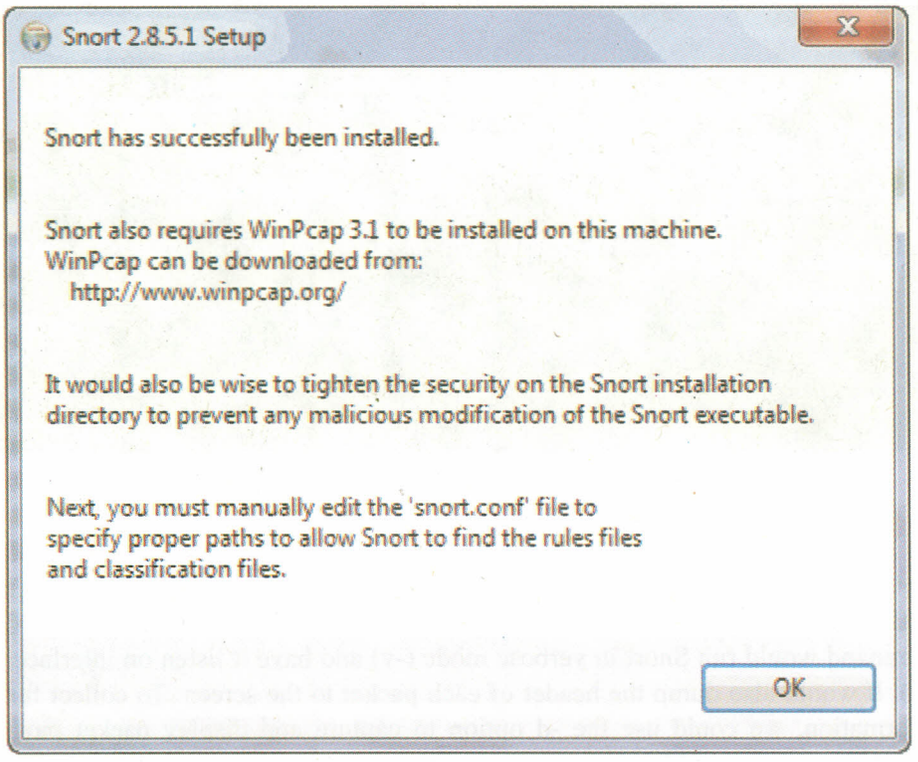


Fig. 13

Once you find these lines, modify them to reflect our default install path (c:\snort) as seen below:

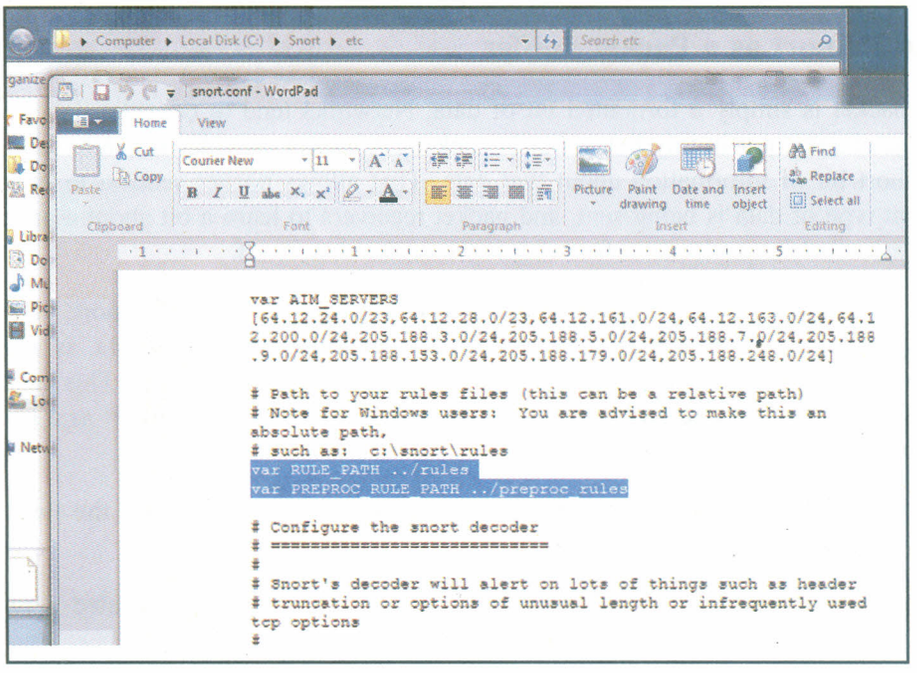


Fig. 14

Save this file and close Wordpad. We are now ready to use our installation of Snort!

To verify that Snort is installed and running correctly you can run a couple of commands from the Command Prompt. Open a command prompt as Administrator, switch to the "C:\Snort\Bin" directory and run "snort.exe -W" to see a list of

interfaces available to Snort. The following is output from the command on Windows 7:

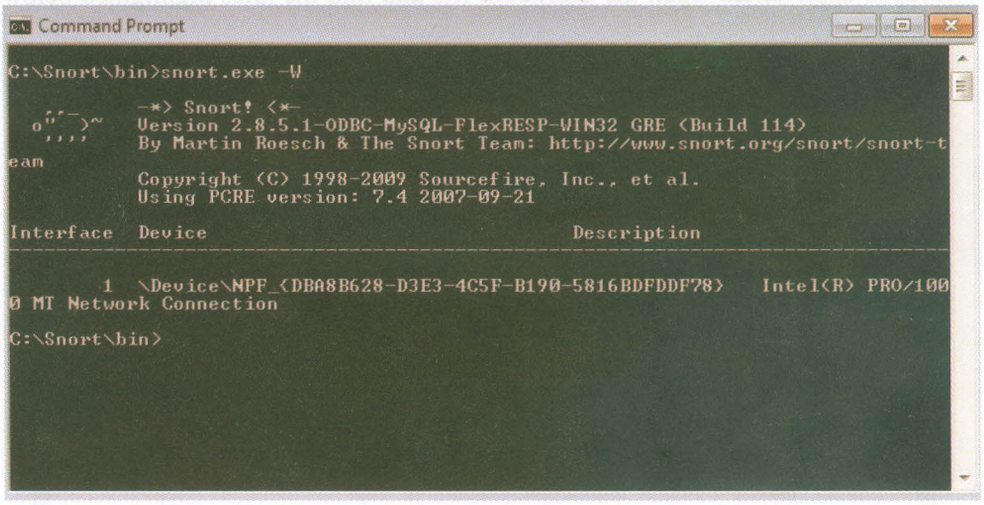


Fig. 15

As you can see, the computer in the example has only one interface with an "Interface" number of "1". If we wanted to use Snort as a sniffer and watch all traffic on this interface, we could issue the command "snort.exe -i 1 -v". This command would run Snort in verbose mode (-v) and have it listen on interface 1 (-i 1). It would also dump the header of each packet to the screen. To collect further information, we could use the -d option to capture and display packet payload. Note: You can use CTRL-C to interrupt the running program.

Let's start Snort as a sniffer to display packet headers and contents. The command we want to enter at our command prompt is "snort.exe -i 1 -vd". You can always run Snort with the "-?" option to get a full list of options available. To stop sniffing packets, break out of the program by pressing Ctrl-C.

2.6.2 Snort as a Packet Sniffer

A packet is simply a formatted bit of data. Networks tend to consider data not in large chunks, nor in tiny bytes, but in packets. A packet provides not only data, but a short bit of information about that data. That allows packet senders to indicate what type of data is being sent, and packet receivers to learn a bit about a packet's data (called its payload) before digging into the data itself.

A packet sniffer is a tool that sniffs, or investigates, packets. With Snort, you can inspect network traffic at the packet level, allowing you to see raw data, as well as the information the packet receiver attached to the raw data itself. This is what you got when you used the snort with the -v flag. When you run Snort as a packet sniffer, you get three things:

1) Information about where Snort is logging information to and the network interface it's inspecting.

2) The packets Snort is sniffing, being sent to and from the indicated network interface.

3) A summary of everything Snort did in this running (printed when you end the sniffing, usually with Ctrl+C).

In the next section, you'll learn about exactly what's in these packets and how to start making rudimentary analyses on these packets. For now, see if you can figure out, for each packet, what each bit of information is that's reported for a packet. Here's a single packet's output:

2.6.3 Snort as a Packet Logger

Packet sniffing is great, but as you've seen, Snort's packet sniffing mode assumes you're hunched over your display, eagerly watching as thousands of lines of network data flies by. Of course, you've got sites to design and worlds to take over, so that's not optimal. By adding the -l switch to Snort, you tell it to log packets to a directory of your choosing:

```
[bdm0509:~/Documents/developerworks/snort_1] sudo snort -l myLogDir/

Password:

Running in packet logging mode

Log directory = myLogDir/

        --== Initializing Snort ==--

Initializing Output Plugins!

Verifying Preprocessor Configurations!

***

*** interface device lookup found: en0

***

Initializing Network Interface en0

Decoding Ethernet on interface en0

        --== Initialization Complete ==--


   ,,_     -*> Snort! <*-

  o"  )~   Version 2.8.0.2 (Build 75)

   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/team.html

           (C) Copyright 1998-2007 Sourcefire Inc., et al.

           Using PCRE version: 7.6 2008-01-28

Not Using PCAP_FRAMES
```

2.6.4 Snort as an IDS (or NIDS)

Ultimately, both packet sniffing and packet logging -- as well as analysis on those logs -- are subsystems of an Intrusion Detection System (or, as Snort is sometimes called, an NIDS, Network Intrusion Detection System). This is where Snort really shines. It's also where things get quite involved and fairly tricky. Because the types of intrusions change rapidly, Snort has a set of rules that you can download from the Snort site that details these intrusions and allows Snort to look for them. The rules change frequently, keeping up with (or at least trying to) the various types of attacks that are going on.

Additionally, you'll need to configure Snort and tell it what to do when it senses an attack. This is where that network and system admin you were being nice to earlier is going to save your site. let them do the work of responding to attacks. But if you can make them aware of attacks by handing them configuration and rules files, you're already way ahead of the game.

If you're impatient, and simply can't wait to try out Snort as an IDS, try running snort-A, which puts Snort in alert mode. You'll have to fiddle a bit to get things working, but this is a great testbed for Snort while you're waiting on the next article.

Check Your Progress 2

Note: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of the Unit.

1) What are the different kinds of Honey Pots and when are they used?

.....

.....

.....

.....

2) Discuss the advantages and the disadvantages of HoneyPots.

.....

.....

.....

.....

3) How would you use Snort to as a packet sniffer and detect suspicious hacker activity?

.....

.....

.....

.....

2.7 LET US SUM UP

Different IDS use different Detection Methods, these can be:

- Signature Based
- Anomaly Based
- Policy Based
- Protocol Analysis
- Specification Based.

Different IPS countermeasures are:

Network Based

- Session Sniping
- Packet Filtering
- Packet Scrubbing

- IP Blocking
- Deception

Host Based

- Code Analysis
- Network Traffic Analysis
- Network Traffic filtering
- File system Monitoring

2.8 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) We use Anomaly based detection method, if the type of attack is unknown.

   Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations. An IDPS using anomaly-based detection has profiles that represent the normal behavior of such things as users, hosts, network connections, or applications. The profiles are developed by monitoring the characteristics of typical activity over a period of time. For example, a profile for a network might show that Web activity comprises an average of 13% of network bandwidth at the Internet border during typical workday hours. The IDPS then uses statistical methods to compare the characteristics of current activity to thresholds related to the profile, such as detecting when Web activity comprises significantly more bandwidth than expected and alerting an administrator of the anomaly. Profiles can be developed for many behavioral attributes, such as the number of e-mails sent by a user, the number of failed login attempts for a host, and the level of processor usage for a host in a given period of time. The major benefit of anomaly-based detection methods is that they can be very effective at detecting previously unknown threats.

2) If we have a LAN system consisting of 2-3 terminals, we employ Network based IDS type. Network-Based, which monitors network traffic for particular network segments or devices and analyzes the network and application protocol activity to identify suspicious activity. It can identify many different types of events of interest. It is most commonly deployed at a boundary between networks, such as in proximity to border firewalls or routers, virtual private network (VPN) servers, remote access servers, and wireless networks.

3) **Different Countermeasures taken by IPS (Host Based)**

   Host-based IPS agents offer various intrusion prevention capabilities. Because the capabilities vary based on the detection techniques used by each product, the following items describe the capabilities by detection technique.

   Code Analysis

   The code analysis techniques can prevent code from being executed, including malware and unauthorized applications. Some host-based IDPSs can also stop network applications from invoking shells, which could be used to attempt to perform certain types of attacks. If configured and tuned well, code analysis can be very effective, particularly at stopping previously unknown attacks.

Network Traffic Analysis

This can stop incoming network traffic from being processed by the host and outgoing network traffic from exiting it. This might be done to stop network, transport, and application layer attacks (and in some cases, wireless networking protocol attacks), as well as to stop the use of unauthorized applications and protocols. Analysis can also identify malicious files being downloaded or transferred and prevent those files from being placed on the host. The network traffic might be dropped or rejected, and the host's personal firewall (which might be built into the agent) could be reconfigured to shun additional traffic related to the suspicious traffic. Network traffic analysis is effective at stopping many known and previously unknown attacks.

Network Traffic Filtering

Working as a host-based firewall, this can stop unauthorized access and acceptable use policy violations (e.g. use of inappropriate external services). It is effective only against stopping activity that is identifiable by IP address and TCP port, UDP port, or ICMP type and code.

File system Monitoring

This can prevent files from being accessed, modified, replaced, or deleted, which could stop malware installation, including Trojan horses and rootkits, as well as other attacks involving inappropriate file access. This technique can provide an additional layer of access control to supplement the existing access control technologies on a host.

Other host-based IDPS detection techniques, such as log analysis, network configuration monitoring, and file integrity and attribute checking, generally do not support prevention actions because they identify events well after they have occurred.

Check Your Progress 2

1) Types of Honeypots

Honeypots come in many shapes and sizes, making them difficult to get a grasp of. There are two general categories, low-interaction and high-interaction honeypots. These categories helps us understand what type of honeypot you are dealing with, its strengths, and weaknesses. Interaction defines the level of activity a honeypot allows an attacker. Low-interaction honeypots have limited interaction, they normally work by emulating services and operating systems. Attacker activity is limited to the level of emulation by the honeypot. For example, an emulated FTP service listening on port 21 may just emulate a FTP login, or it may support a variety of additional FTP commands. The advantages of a low-interaction honeypot is their simplicity. These honeypots tend to be easier to deploy and maintain, with minimal risk. Usually they involve installing software, selecting the operating systems and services you want to emulate and monitor, and letting the honeypot go from there. This plug and play approach makes deploying them very easy for most organizations. Also, the emulated services mitigate risk by containing the attacker's activity, the attacker never has access to an operating system to attack or harm others. The main disadvantages with low interaction honeypots is that they log only limited information and are designed to capture known activity. The emulated services can only do so much. Also, its easier for an attacker to detect a low-interaction honeypot, no matter how good the emulation is, skilled attacker can eventually detect their presence. Examples of low-interaction honeypots include Specter, Honeyd, and KFSensor.

High-interaction honeypots are different, they are usually complex solutions as they involve real operating systems and applications. Nothing is emulated, we give attackers the real thing. If you want a Linux honeypot running an FTP server, you build a real Linux system running a real FTP server. The advantages with such a solution are two fold. First, you can capture extensive amounts of information. By giving attackers real systems to interact with, you can learn the full extent of their behavior, everything from new rootkits to international IRC sessions. The second advantage is high-interaction honeypots make no assumptions on how an attacker will behave. Instead, they provide an open environment that captures all activity. This allows high-interaction solutions to learn behavior we would not expect. An excellent example of this is how a Honeynet captured encoded back door commands on a non-standard IP protocol (specifically IP protocol 11, Network Voice Protocol). However, this also increases the risk of the honeypot as attackers can use these real operating system to attack non-honeypot systems. As result, additional technologies have to be implement that prevent the attacker from harming other non-honeypot systems. In general, high-interaction honeypots can do everything low-interaction honeypots can do and much more. However, they can be more complext to deploy and maintain. Examples of high-interaction honeypots include Symantec Decoy Server and Honeynets. You can find a complete listing of both low and high interaction honeypots at Honeypot Solutions page. To better understand both low and high interaction honeypots lets look at two examples. We will start with the low-interaction honeypot Honeyd.

2) Advantages and Disadvantages of Honeypots

Advantages: Honeypots are a tremendously simply concept, which gives them some very powerful strengths.

- *Small data sets of high value:* Honeypots collect small amounts of information. Instead of logging a one GB of data a day, they can log only one MB of data a day. Instead of generating 10,000 alerts a day, they can generate only 10 alerts a day. Remember, honeypots only capture bad activity, any interaction with a honeypot is most likely unauthorized or malicious activity. As such, honeypots reduce 'noise' by collectin only small data sets, but information of high value, as it is only the bad guys. This means its much easier (and cheaper) to analyze the data a honeypot collects and derive value from it.

- *New tools and tactics:* Honeypots are designed to capture anything thrown at them, including tools or tactics never seen before.

- *Minimal resources:* Honeypots require minimal resources, they only capture bad activity. This means an old Pentium computer with 128MB of RAM can easily handle an entire class B network sitting off an OC-12 network.

- *Encryption or IPv6:* Unlike most security technologies (such as IDS systems) honeypots work fine in encrypted or IPv6 environments. It does not matter what the bad guys throw at a honeypot, the honeypot will detect and capture it.

- *Information:* Honeypots can collect in-depth information that few, if any other technologies can match.

- *Simplicty:* Finally, honeypots are conceptually very simple. There are no fancy algorithms to develop, state tables to maintain, or signatures to update. The simpler a technology, the less likely there will be mistakes or misconfigurations.

Disadvantages: Like any technology, honeypots also have their weaknesses. It is because of this they do not replace any current technology, but work with existing technologies.

- *Limited view:* Honeypots can only track and capture activity that directly interacts with them. Honeypots will not capture attacks against other systems, unless the attacker or threat interacts with the honeypots also.
- *Risk:* All security technologies have risk. Firewalls have risk of being penetrated, encryption has the risk of being broken, IDS sensors have the risk of failing to detect attacks. Honeypots are no different, they have risk also. Specifically, honeypots have the risk of being taken over by the bad guy and being used to harm other systems.

3) Snort as a packet sniffer

A packet is simply a formatted bit of data. Networks tend to consider data not in large chunks, nor in tiny bytes, but in packets. A packet provides not only data, but a short bit of information about that data. That allows packet senders to indicate what type of data is being sent, and packet receivers to learn a bit about a packet's data (called its payload) before digging into the data itself.

A packet sniffer is a tool that sniffs, or investigates, packets. With Snort, you can inspect network traffic at the packet level, allowing you to see raw data, as well as the information the packet receiver attached to the raw data itself. This is what you got when you used the snort with the -v flag. When you run Snort as a packet sniffer, you get three things:

1) Information about where Snort is logging information to and the network interface it's inspecting.
2) The packets Snort is sniffing, being sent to and from the indicated network interface.
3) A summary of everything Snort did in this running (printed when you end the sniffing, usually with Ctrl+C).

UNIT 3 SCANNING AND ANALYSIS TOOLS

Structure

3.0 INTRODUCTION

Network scanning builds a clearer picture of accessible hosts and their network services. Network scanning and analysis is the real data gathering exercise of any network security assessment. The motive behind IP network scanning is to gain insight into the following elements of a given network:

- Areas of vulnerability within target host IP stack implementations
- Operating platforms of target hosts and their configuration
- Accessible TCP and UDP network services running on the target hosts
- Configuration of filtering and security systems (including firewalls, border routers, switches and IDS sensors).

Remember in computer science *platform* means computer hardware and/or operating system.

3.1 OBJECTIVES

After studying this unit, you should be able to:

- identify the need of using network scanning and analysis tools;
- recognize different software tools used for scanning and analyzing a network;
- to use standard tools to scan and analyze network components;
- explain how these tools work and are able to give meaningful results;

- elucidate different techniques used to scan a target host or check compliance to a given security policy;
- identify basic vulnerabilities within a network and be able to exploit them; and
- understand the countermeasures that can be taken to prevent attackers from scanning and exploiting a network.

3.2 BASIC SCANNING TECHNIQUES

Performing a network scan paints a clear picture of the network security mechanisms and its topology. Further analysis steps involve collecting information about the UDP and TCP services that are running on the network. Over time, a number of techniques have been developed for surveying the protocols and ports on which a target machine is listening. They all offer different benefits and problems. Lets take a look at some of those techniques:

There are a total of 42 types of ICMP messages as defined in RFC 792.

3.2.1 ICMP Probe

The Internet Control Message Protocol (ICMP) is one of the core protocols of the Internet Protocol Suite. It is chiefly used by the operating systems of networked computers to send error messages indicating, for example, that a requested service is not available or that a host or router could not be reached. ICMP can also be used to relay query messages. It can be used to identify potentially weak and poorly connected networks.

The following types of ICMP messages are useful for performing network scans:

Type 8 (echo request)

Echo request messages are also known as ping packets. You can use a scanning tool such as nmap to perform ping sweeping and easily identify hosts that are accessible.

Type 13 (timestamp request)

A timestamp request message requests system time information from the target host. The response is in a decimal format and is the number of milliseconds elapsed since midnight GMT.

Type 15 (information request)

The ICMP information request message was intended to support self-configuring systems such as diskless workstations at boot time, to allow them to discover their network address. Protocols such as RARP, BOOTP or DHCP do so more robustly, so type 15 messages are rarely used.

Type 17 (subnet address mask request)

An address mask request message reveals the subnet mask used by the target host. This information is useful when mapping networks and identifying the size of subnets and network spaces used by organizations.

Tools using ICMP PROBE technique

PING

Ping is arguably the most commonly used utilities used in network scanning. It is available both on Linux and Windows. It is used to determine whether a target host is reachable from a source and estimate a round trip time to the same.

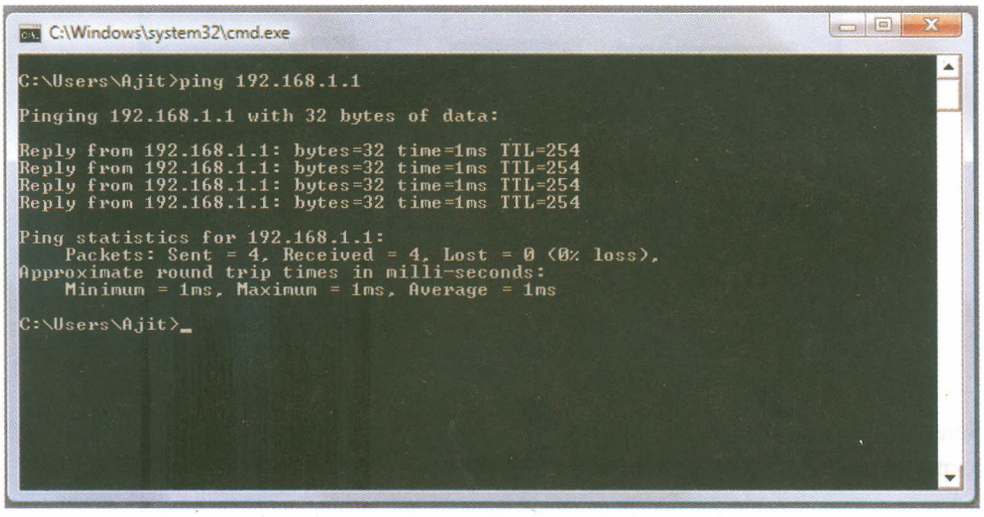


Fig. 1

SING

Send ICMP Nasty Garbage (SING) is a command-line tool that sends fully customizable ICMP packets. The main purpose of the tool is to replace the ping commandwith certain enhancements, including the ability to transmit and receive spoofed packets, send MAC-spoofed packets and support the transmission of many other message types, including ICMP address mask, timestamp and information requests, router solicitation and router advertisement messages.

Using sing to send ICMP address mask request messages:

sing -mask 192.168.0.25

SINGing to 192.168.0.25 (192.168.0.25): 12 data bytes

12 bytes from 192.168.0.25: seq=0 ttl=236 mask=255.255.255.0

12 bytes from 192.168.0.25: seq=1 ttl=236 mask=255.255.255.0

12 bytes from 192.168.0.25: seq=2 ttl=236 mask=255.255.255.0

12 bytes from 192.168.0.25: seq=3 ttl=236 mask=255.255.255.0

NMAP

Nmap is one of the most widely used network scanning and analysis tools. Nmap can perform ICMP ping-sweep scans of target address spaces easily and relatively quickly.

Installing NMAP

Download the nmap for windows setup from http://www.nmap.org/download.html

Then run it to get the following window.

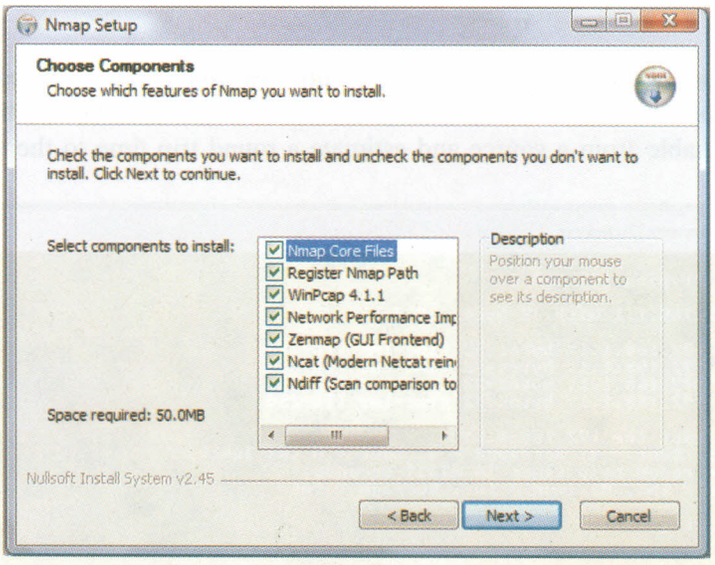


Fig. 2

Select the components you want to install, click Next to continue. You might need to perform some intermediate steps.

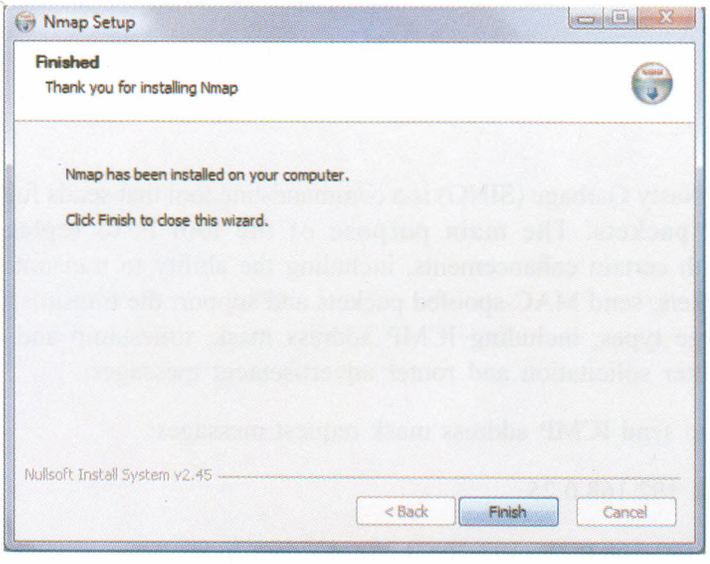


Fig. 3

Click Finish to exit the setup.

Performing a ping sweep with Zenmap (nmap GUI utility):

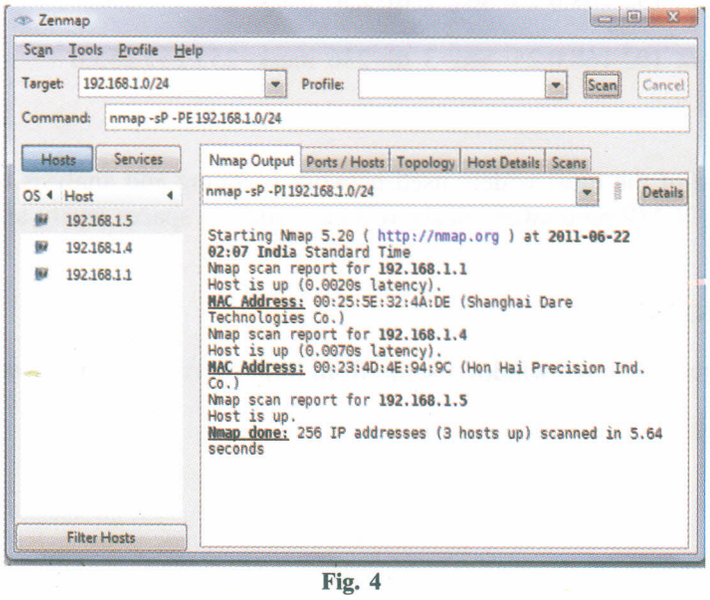


Fig. 4

3.2.2 TCP Port Scanning

Accessible TCP ports can be identified by port scanning target IP addresses.

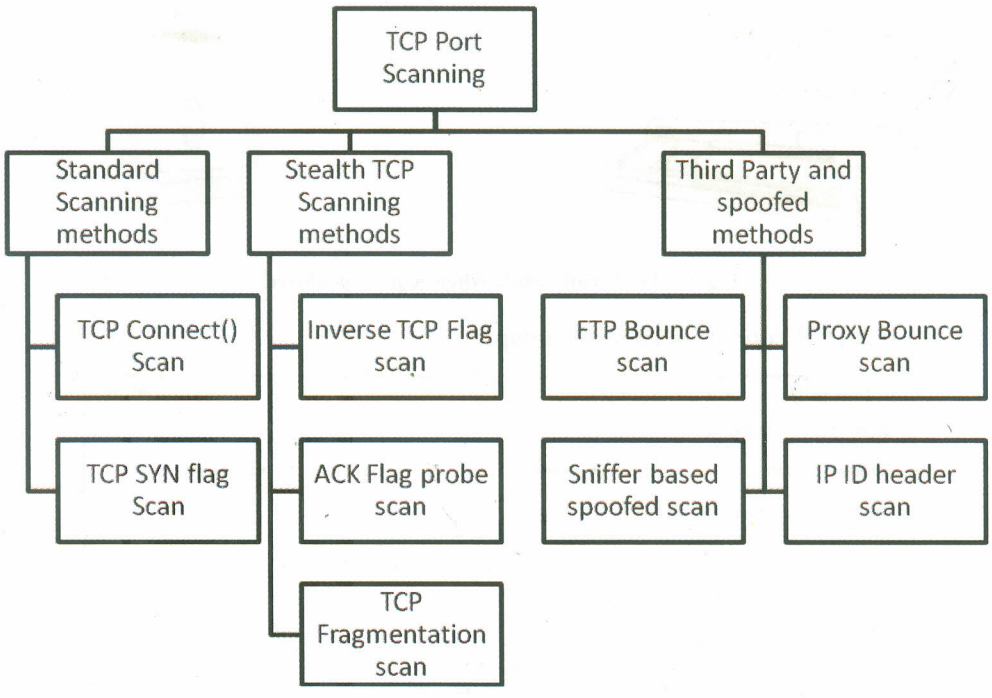


Fig. 5

3.2.2.1 Standard Scanning Methods

Extremely simple direct techniques used to identify accessible services accurately. These scanning methods are reliable but are easily logged and identified.

TCP Connect() scan

This is the most basic form of TCP scanning. The connect() system call provided by your operating system is used to open a connection to every interesting port on the machine. If the port is listening, connect() will succeed, otherwise the port isn't reachable. One strong advantage to this technique is that you don't need any special privileges. Another advantage is speed. While making a separate connect() call for every targeted port in a linear fashion would take ages over a slow connection, you can hasten the scan by using many sockets in parallel. Using non-blocking I/O allows you to set a low time-out period and watch all the sockets at once.

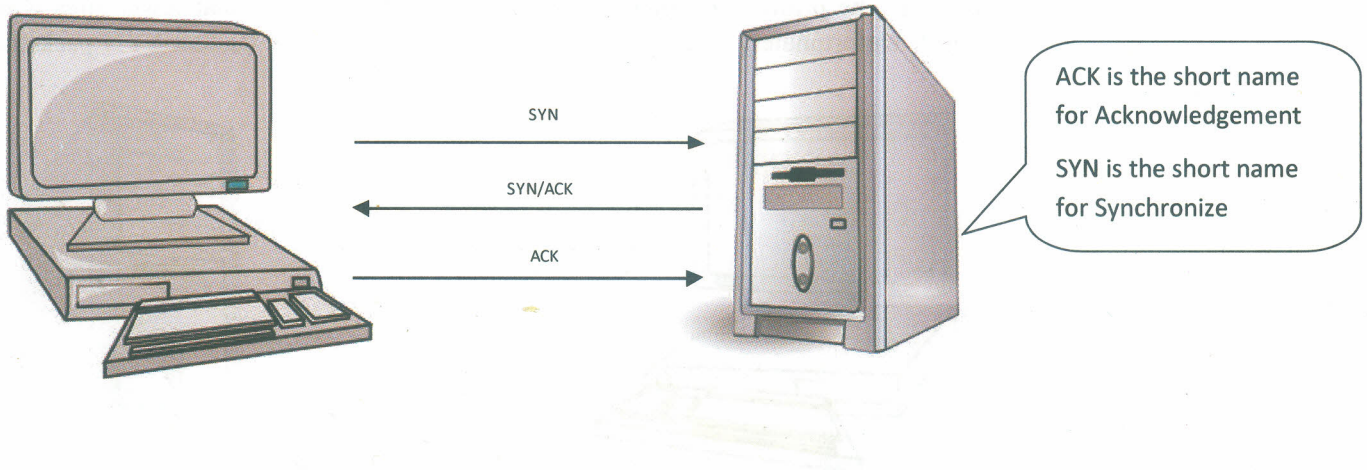


Fig. 6: TCP scan result when a port is open

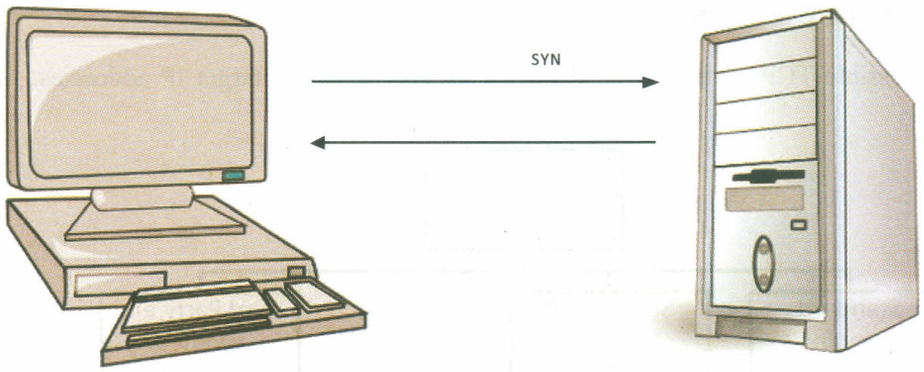


Fig. 7: TCP scan result when a port is closed

Performing TCP connect scan with nmap:

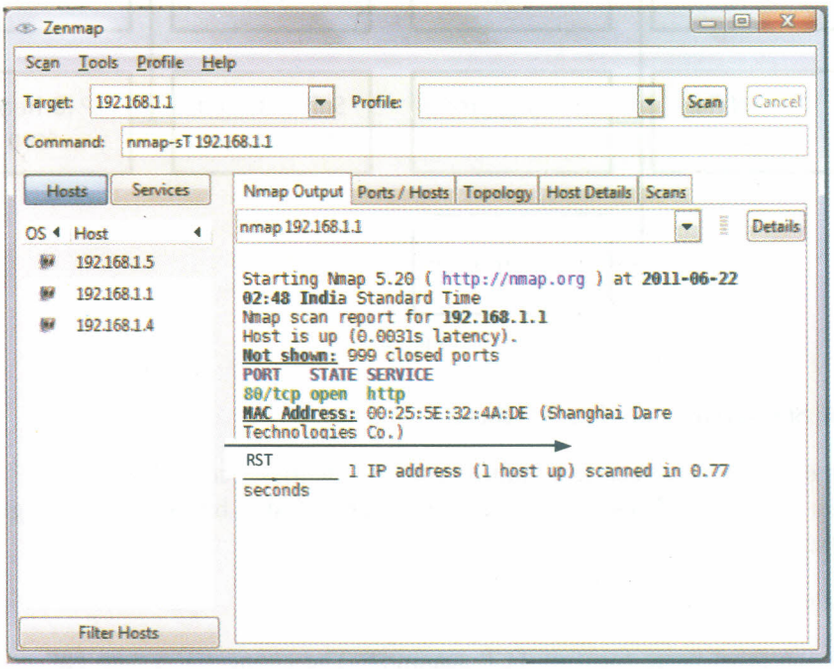


Fig. 8

TCP SYN Scan

This technique is often referred to as "half-open" scanning because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and wait for a response. A SYN/ACK indicates the port is listening. A RST is indicative of a non-listener. If a SYN/ACK is received, you immediately send a RST to tear down the connection (actually the kernel does this for us). The primary advantage to this scanning technique is that fewer sites will log it. Unfortunately you need root privileges to build these custom SYN packets.

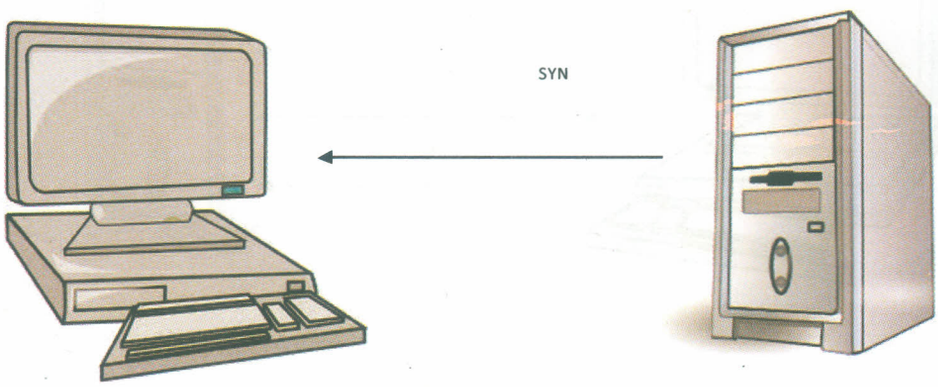


Fig. 9: TCP scan result when a port is open

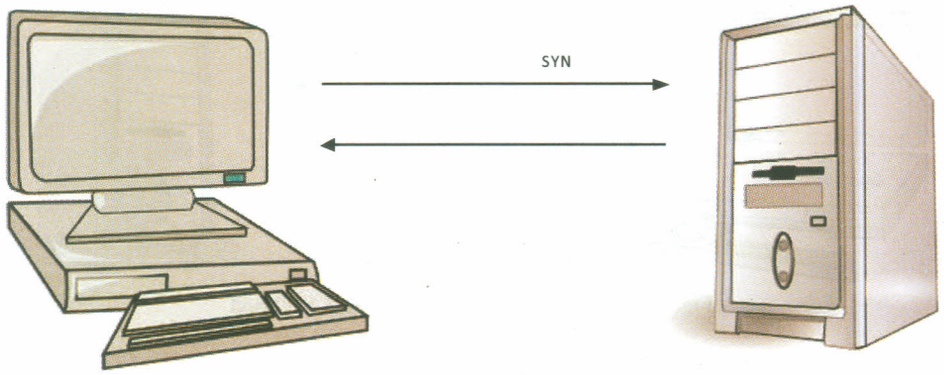


Fig. 10: TCP scan result when a port is closed

Nmap can perform a SYN port scan under both Unix and Windows environments using the -sS flag. Many other Unix half-open port scanners exist, including strobe.

3.2.2.2 Stealth TCP Scan Methods

Stealth scanning methods and techniques aren't effective at accurately mapping the open ports of some operating systems but do provide a degree of stealth and are sometimes not logged.

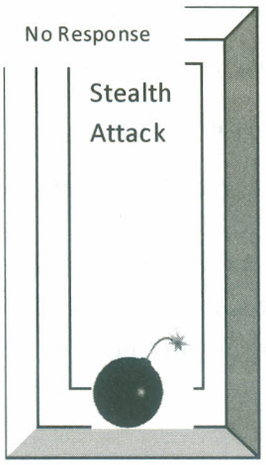


Fig. 11

Inverse TCP flag Scan

Using half-open SYN flags to probe a target is known as an inverted technique because responses are sent back only by closed ports. RFC 793 states that if a port is closed on a host, an RST/ACK packet should be sent to reset the connection. To take advantage of this feature, attackers send TCP probe packets with various TCP flags set.

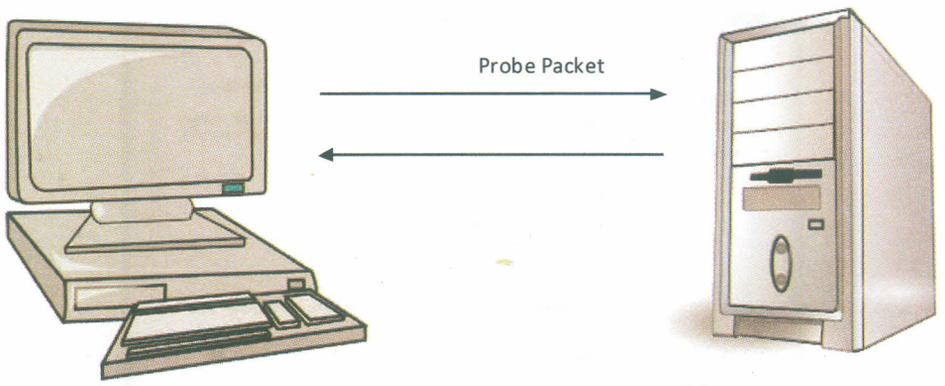


Fig. 12: TCP scan result when a port is open

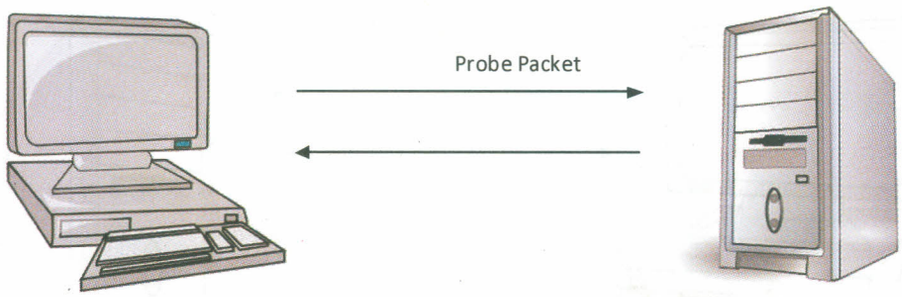


Fig. 13: TCP scan result when a port is closed

Tools that use this technique:

Nmap can perform an inverse TCP flag port scan under both Linux and Windows environments, using the following flags:

-sF For a scan with only the FIN flag set on probe packets
-sN For a NULL scan with no TCP flags set on probe packets
-sX For an Xmas tree scan with all TCP flags set

ACK flag probe scanning

There are two main techniques:

- Analysis of the time-to-live (TTL) field of received packets
- Analysis of the WINDOW field of received packets

How to use ACK flag probe scan:

Nmap supports ACK flag probe scanning, with the -sA and -sW flags to analyze the TTL and WINDOW RST/ACK)ectively.

hping2 can also sample TTL and WINDOW values, but this can prove highly time consuming in most cases. The tool is more useful for analyzing low-level responses, as opposed to port scanning in this fashion. hping2 is available from http://www.hping.org.

TCP Fragmentation Scan

This is not a new scanning method in and of itself, but a modification of other techniques. Instead of just sending the probe packet, you break it into a couple of small IP fragments. You are splitting up the TCP header over several packets to make it harder for packet filters and so forth to detect what you are doing.

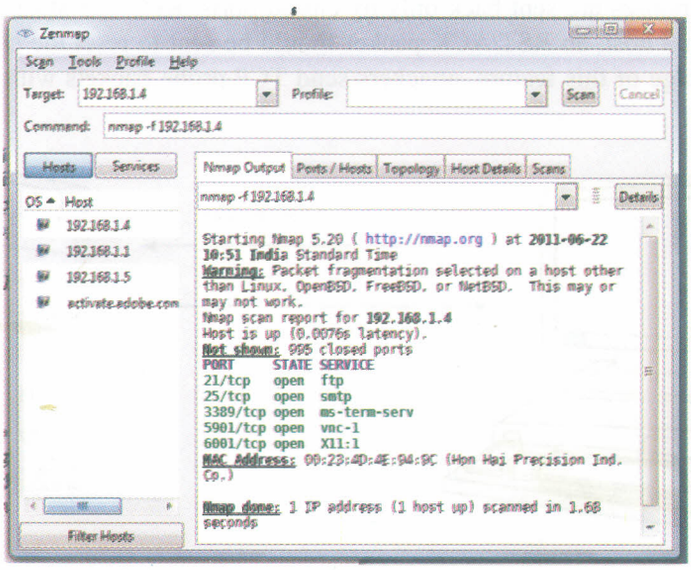


Fig. 14: Nmap with -f option to enable fragmentation

To use TCP fragmentation technique one can use the -f flag with *Nmap*, it instructs the specified SYN or FIN scan to use tiny fragmented packets.

3.2.2.3 Third Party and Spoofed TCP Scan

Third-party port scanning methods allow for probes to be effectively bounced through vulnerable servers to hide the true source of the network scanning. An additional benefit of using a third-party technique in this way is that insight into firewall configuration can be gained by potentially bouncing scans through trusted hosts that are vulnerable.

FTP Bounce Scan

The following occurs when performing an FTP bounce scan:

1) The attacker connects to the FTP control port (TCP port 21) of the vulnerable FTP server that she is going to bounce her attack through and enters passive mode, forcing the FTP server to send data using DTP (data transfer process) to a specific port of a specific host:

   QUOTE PASV

   227 Entering Passive Mode (64,12,168,246,56,185).

2) A PORT command is issued, with an argument passed to the FTP service telling it to attempt a connection to a specific TCP port of the target server; for example,

   TCP port 23 of 144.51.17.230:

   PORT 144,51,17,230,0,23

   200 PORT command successful.

3) After issuing the PORT command, a LIST command is sent. The FTP server then attempts to create a connection with the target host defined in the PORT command issued previously:

   LIST

   150 Opening ASCII mode data connection for file list

   226 Transfer complete.

   If a 226 response is seen, then the port on the target host is open. If, however, a 425 response is seen, the connection has been refused:

   LIST

   425 Can't build data connection: Connection refused

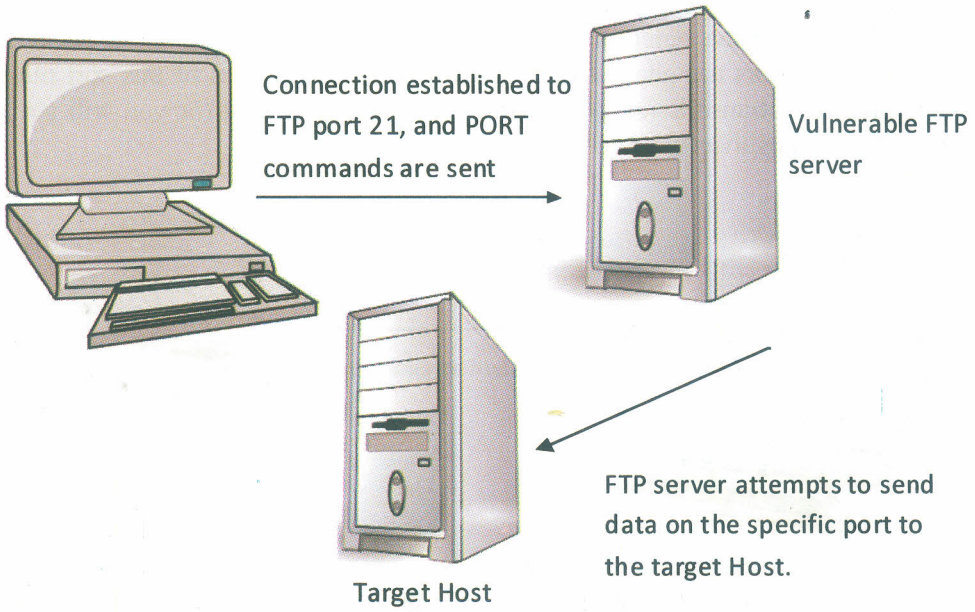


Fig. 15

Nmap can be used to do a FTP bounce scan using the following command

Nmap -P0 -b username:password@ftp-server:port <target host>

Proxy Bounce Scanning

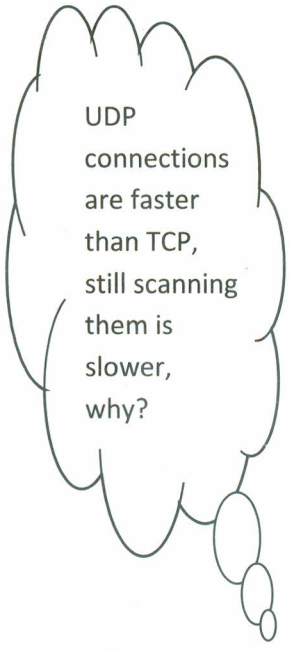
Attackers bounce TCP attacks through open proxy servers. Depending on the level of poor configuration, the server will sometimes allow a full-blown TCP port scan to be relayed. Using proxy servers to perform bounce port scanning in this fashion is often time consuming, so many attackers prefer to abuse open proxy servers more efficiently by bouncing actual attacks through to target networks.

ppscan.c, a publicly available Unix-based tool to bounce port scans, can be found in source form at http://www.dsinet.org/tools/network-scanners/ppscan.c

IP ID header scanning

IP ID header scanning (also known as idle or dumb scanning) is an obscure scanning technique that involves abusing implementation peculiarities within the TCP/IP stack of most operating systems. Three hosts are involved:

- The host, from which the scan is launched
- The target host, which will be scanned
- A zombie or idle host, which is an Internet-based server that is queried with spoofed port scanning against the target host to identify open ports from the perspective of the zombie host



Nmap supports such IP I UDP probe packet th the option:

-sI <zombie host[:probe port]>

3.2.3 UDP Port Scaning

UDP ICMP port unreachable scanning

This scanning method varies from the above in that we are using the UDP protocol instead of TCP. While this protocol is simpler, scanning it is actually significantly more difficult. This is because open ports don't have to send an acknowledgement in response to our probe and closed ports aren't even required to send an error packet. Fortunately, most hosts do send an ICMP_PORT_UNREACH error when you send a packet to a closed UDP port. Thus you can find out if a port is NOT open and by exclusion determine which ports which are. Neither UDP packets, nor the ICMP errors are guaranteed to arrive, so UDP scanners of this sort must also implement retransmission of packets that appear to be lost (or you will get a bunch of false positives).

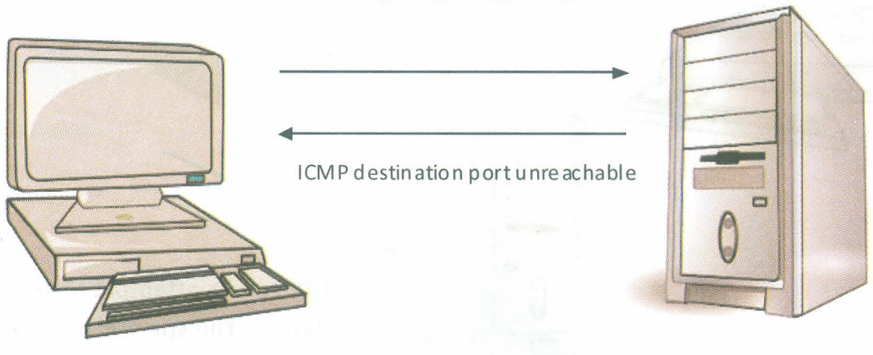


Fig. 16: UDP scan result when the port is closed.

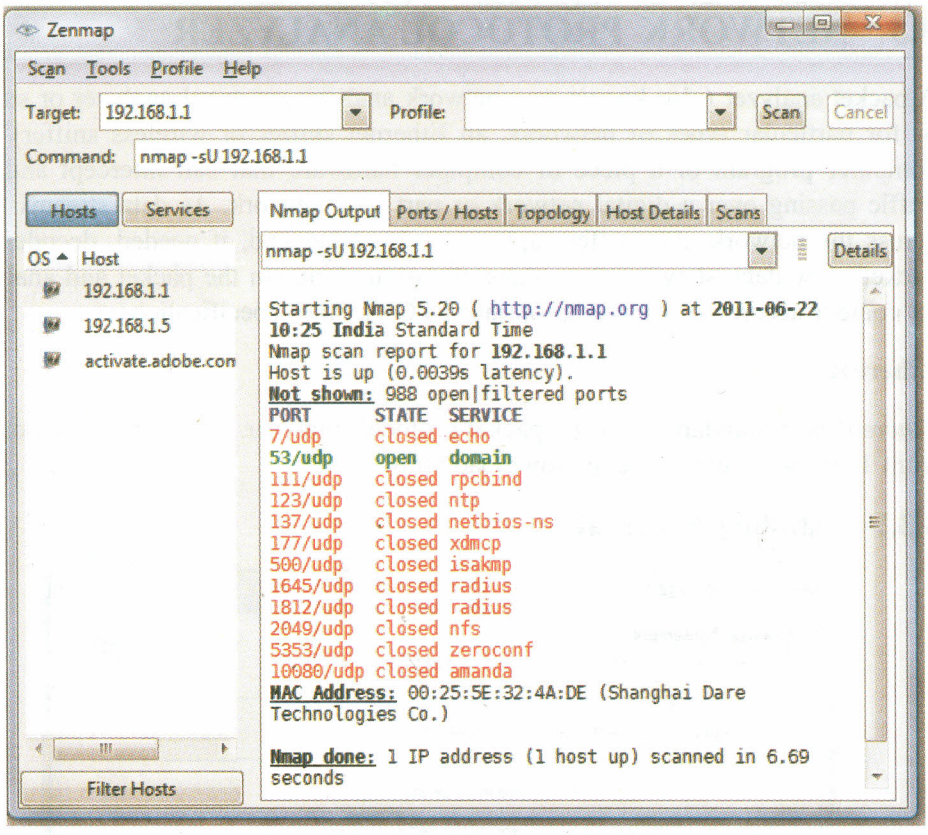


Fig. 17: Scanning target host with nmap -sU option gives a listing of udp ports.

Check Your Progress 1

Note: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of the Unit.

1) What are the different techniques used for TCP port scanning? How are they classified?

.....

.....

.....

.....

2) What is meant by ICMP probing? What are the different types of ICMP messages that can be used for probing?

.....

.....

.....

.....

3) Why can't the TCP port scanning techniques be used with UDP?

.....

.....

.....

.....

3.3 NETWORK PROTOCOL ANALYZER

A packet analyzer (also known as a network analyzer, protocol analyzer or sniffer or for particular types of networks, an Ethernet sniffer or wireless sniffer) is a computer program or a piece of computer hardware that can intercept and log traffic passing over a digital network or part of a network. As data streams flow across the network, the sniffer captures each packet and, if needed, decodes the packet's raw data, showing the values of various fields in the packet and analyzes its content according to the appropriate RFC or other specifications.

Sniffers are commonly used by Hackers

Ethereal

Ethereal is a standard network packet/protocol analyzer. It can be downloaded from http://www.ethereal.com/download.html

3.3.1 Installing Ethereal

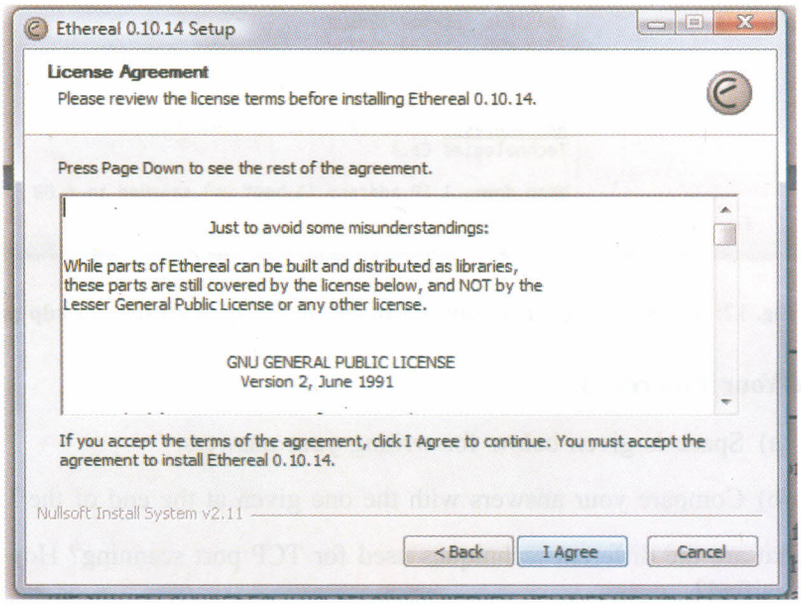


Fig. 18

Click on I Agree

WinPcap is the library used to hook up the packets from Windows TCP stack

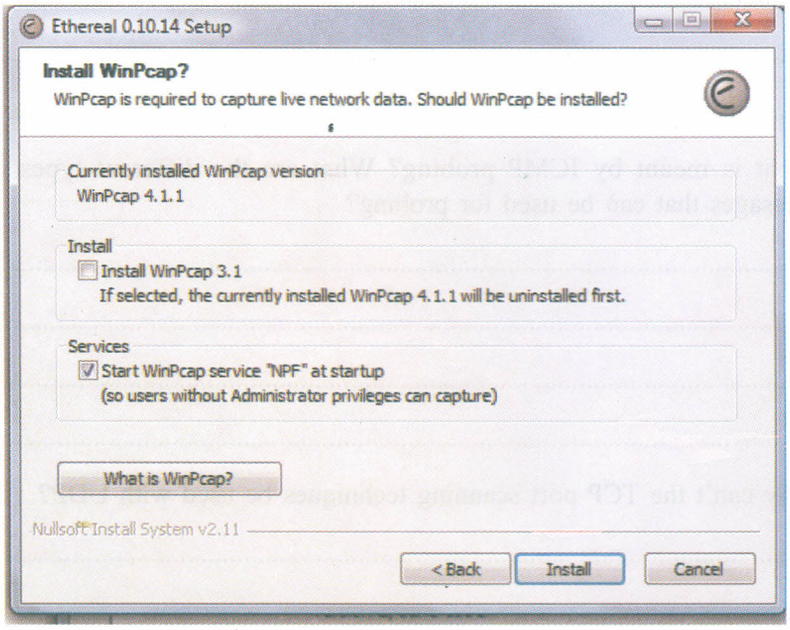


Fig. 19

You need to Install WinPcap along with Ethereal, Check Install WinPcap if not already installed.

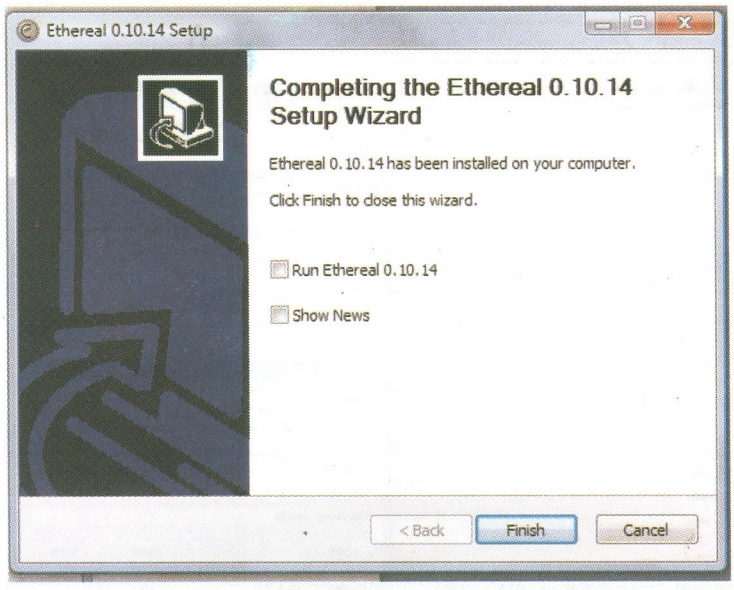


Fig. 20

Click on Finish to finish the installation. And then Run Ethereal.

3.3.2 Using Ethereal

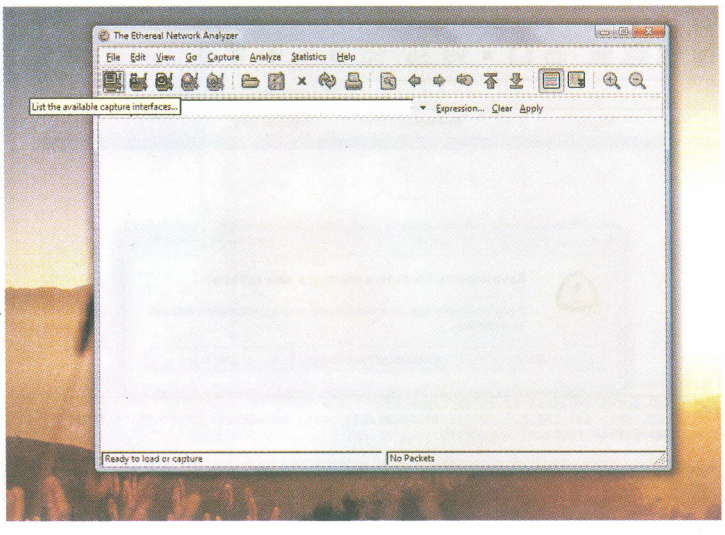


Fig. 21

Click on List the available interfaces to see the detected interfaces.

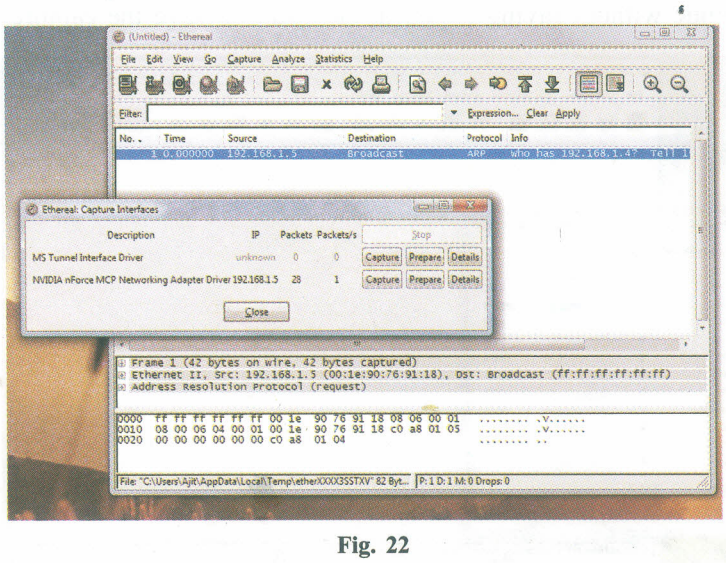


Fig. 22

Choose the desired interface on which to capture, click Prepare

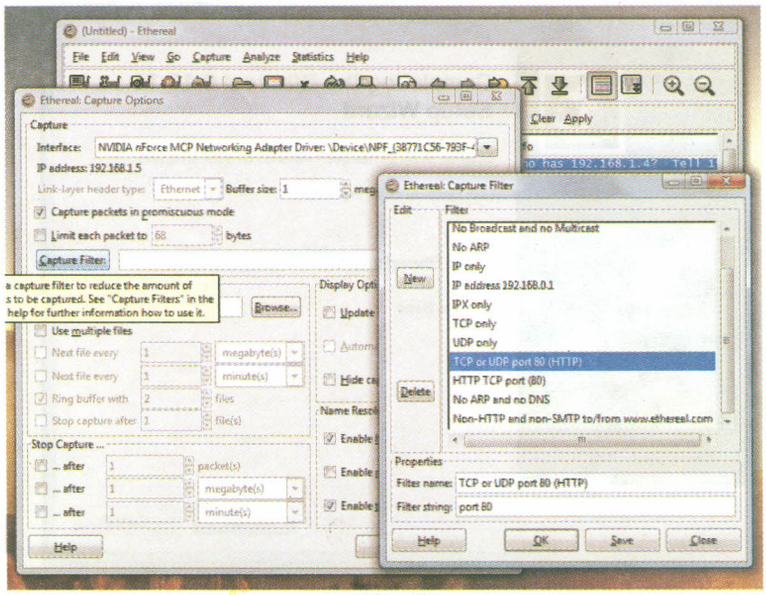


Fig. 23

Click on the capture filter to select a filter(if any), we select TCP or UDP port 80(HTTP) filter. Click on save and then click on Start.

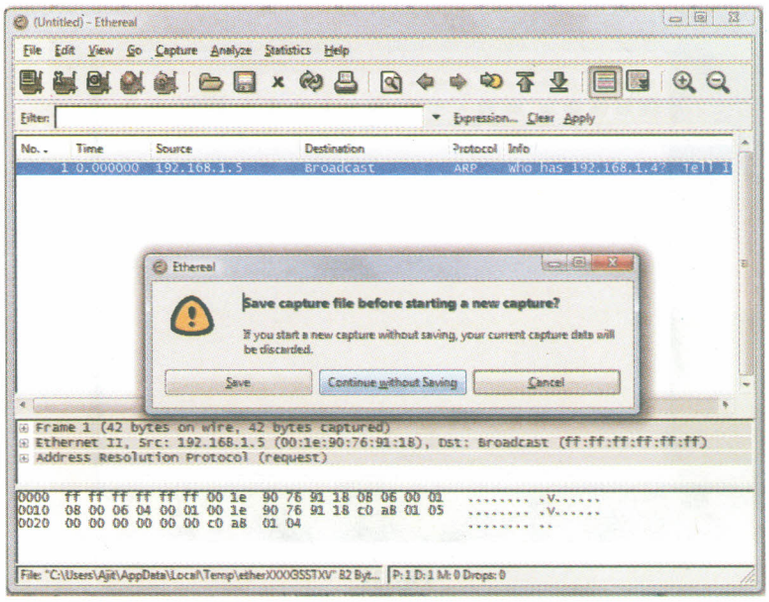


Fig. 24

Click Continue without saving if you do not want to save the capture.

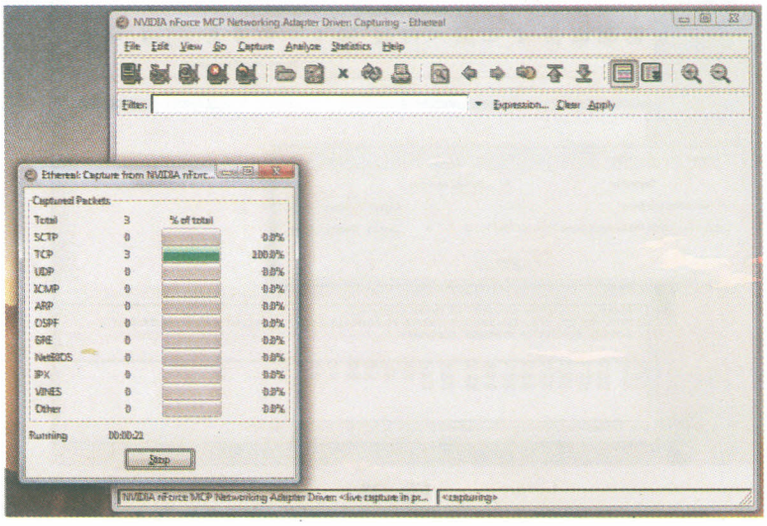


Fig. 25

The Capture is Running and the Dialog will show you what type of packets are being captured. Click Stop to stop Capturing.

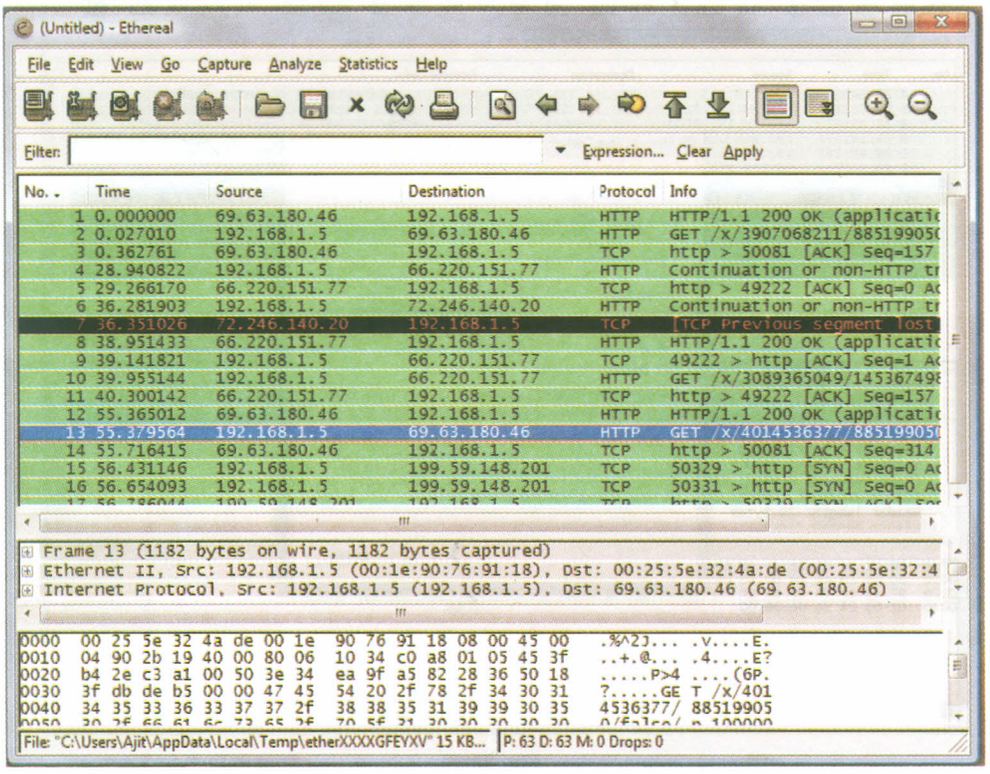


Fig. 26

Captured Packets displayed.

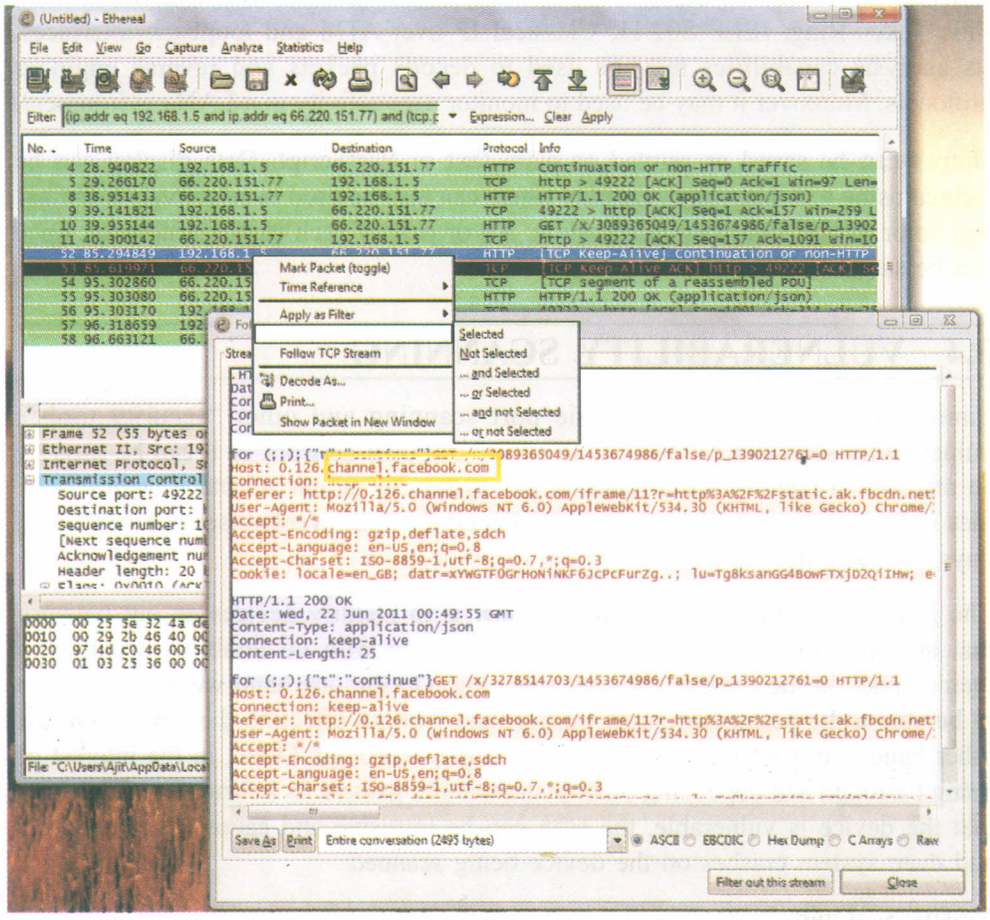


Fig. 27

Rt. Click on any packet and then Follow TCP stream to see the communications, as we can see here the user was communicating with facebook.com

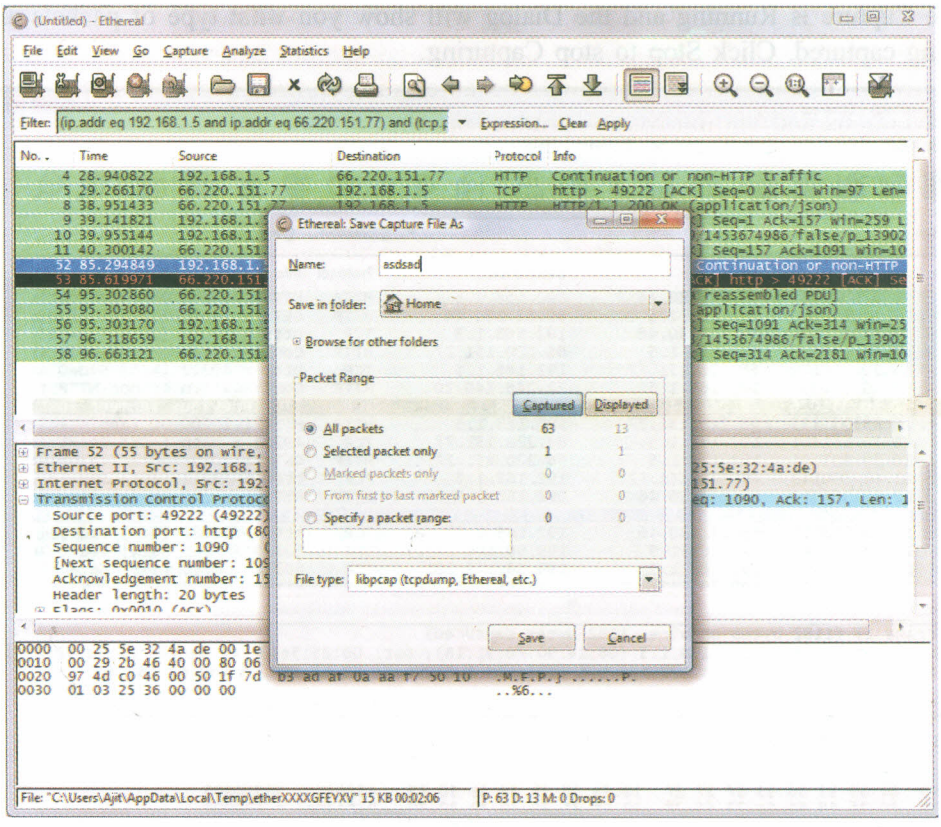


Fig. 28

You can also Save the packet or the full capture to a file on the disk.

Ethereal can be used to capture Network traffic and live data from Ethernet, FDDI, PPP, Token-Ring, IEEE 802.11, Classical IP over ATM and loopback interfaces. Its analysis capabilities may be used to decode packets and reverse engineer protocols. Moreover it may be used to monitor user activity in a LAN environment.

Output can be saved or printed as plain text or PostScript. Data display can be refined using a display filter. Display filters can also be used to selectively highlight and color packet summary information. All or part of each captured network trace can be saved to disk.

3.4 VULNERABILITY SCANNING

Nesus scanner was an Open Source tool until 2005 after which the makers closed the source

Similar to protocol analysis, basic port scanning and other "security tools", vulnerability scanning can help us to secure our own network or it can be used by an attacker to identify weaknesses in our system to mount an attack against. The idea is for us to use these tools to identify and fix these weaknesses before the bad guys use them against us.

The goal of running a vulnerability scanner is to identify devices on our network that are open to known vulnerabilities. One issue with vulnerability scanners is their impact on the devices they are scanning. On the one hand we want the scan to be able to be performed in the background without affecting the device. On the other hand, we want to be sure that the scan is thorough. Often, in the interest of being thorough and depending on how the scanner gathers its information or verifies that the device is vulnerable, the scan can be intrusive and cause adverse affects and even system crashes on the device being scanned.

Nessus Scanner

Nessus is a product by and is also freely available. While there is a Windows graphical front-end available, the core Nessus product requires Linux/Unix to run. The up side to that is that Linux can be obtained for free and many versions of

Linux have relatively low system requirements so it would not be too difficult to take an old PC and set it up as a Linux server. For administrators used to operating in the Microsoft world there will be a learning curve to get used to Linux conventions and get the Nessus product installed.

After performing an initial vulnerability scan you will need to implement a process for addressing the identified vulnerabilities. In most cases there will be patches or updates available to cure the problem. Sometimes though there may be operational or business reasons why you can't apply the patch in your environment or the vendor of your product may not yet have released an update or patch. In those cases you will need to consider alternative means to mitigate the threat.

3.4.1 Installing Nessus Scanner

Download Nessus Scanner from the website of Tenable Network Security http://www.tenable.com/products/nessus/nessus-download-agreement.

You may download the Home Feed for free personal use. Accept the License agreement and register with Nessus. You will receive an e-mail containing information about your serial key.

We will use this key during our installation of Nessus Scanner. Now begin the installation by executing the binary downloaded from the internet.

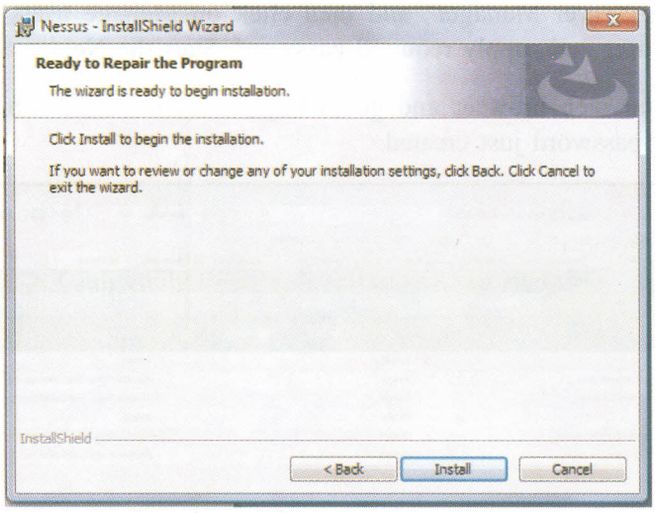


Fig. 29

Click on 'Install' to begin installation

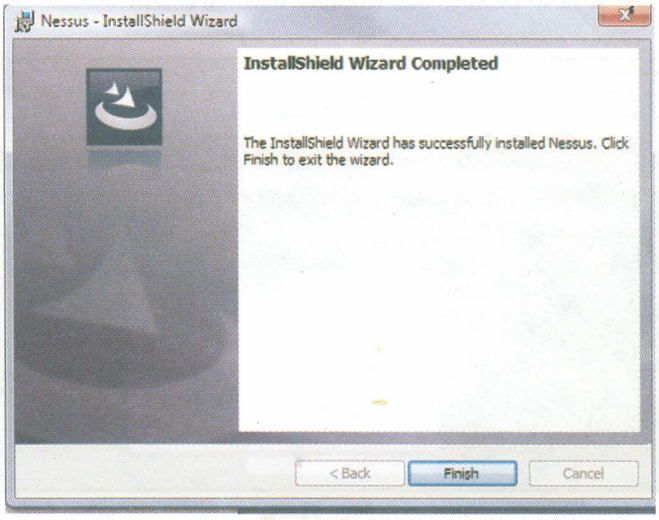


Fig. 30

Click on "Finish" to Complete the installation.

3.4.2 Using Nessus Scanner

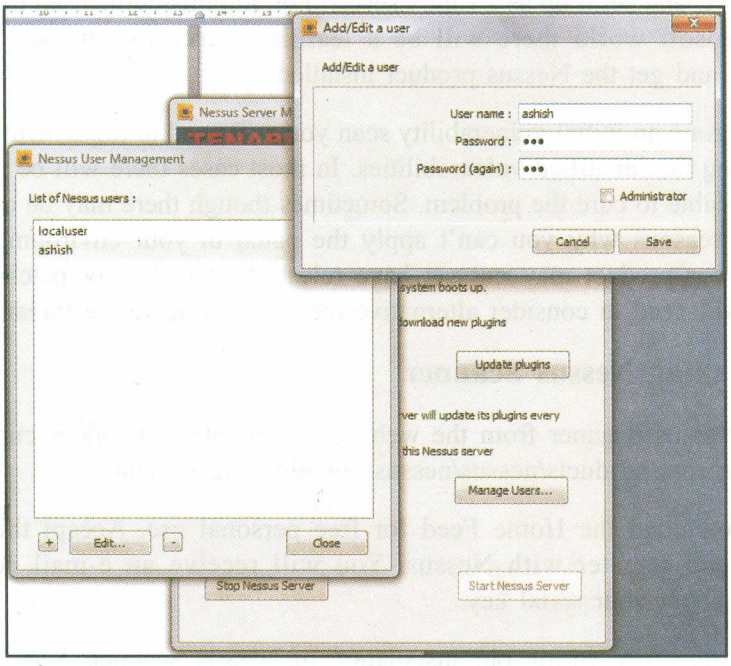


Fig. 31

Open "Nessus Server Manager" and then click on Manage users, now add an administrator user and supply required password. Start the Nessus Server.

Now open your web browser and go to http://localhost:8834. Login using the username and password just created.

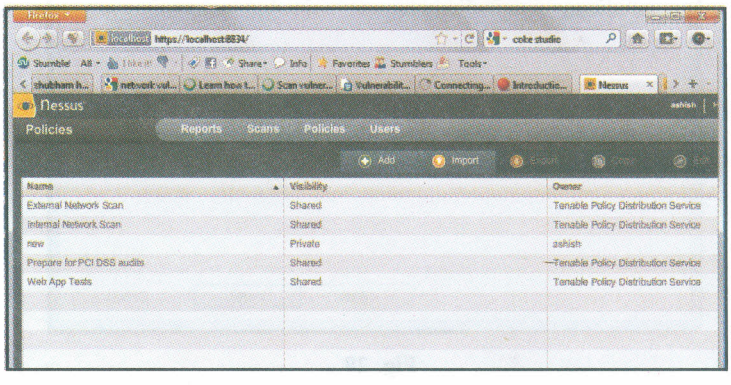


Fig. 32

We can now add a new policy, but to start with, we will select 'Scan' and do a scan.

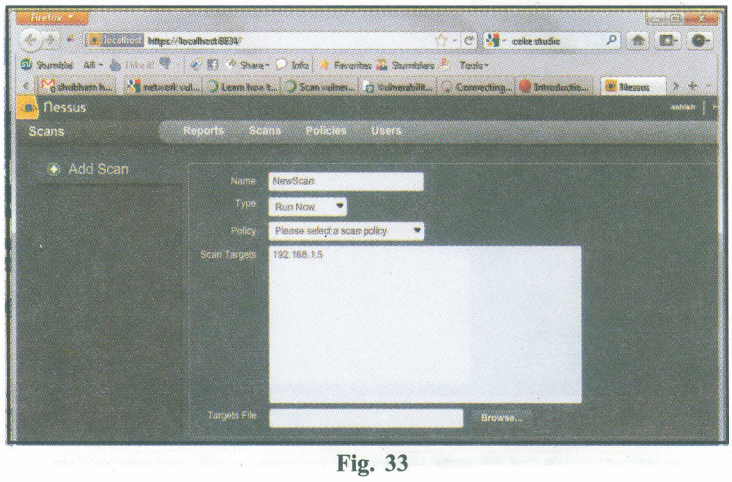


Fig. 33

Click on Add and then fill in the details about the network, we will select a built in policy for our first scan.

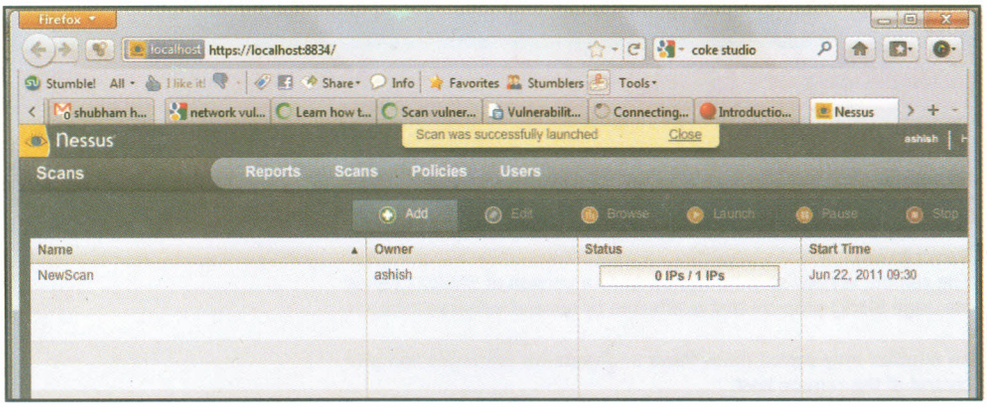


Fig. 34

Scanning in Progress...

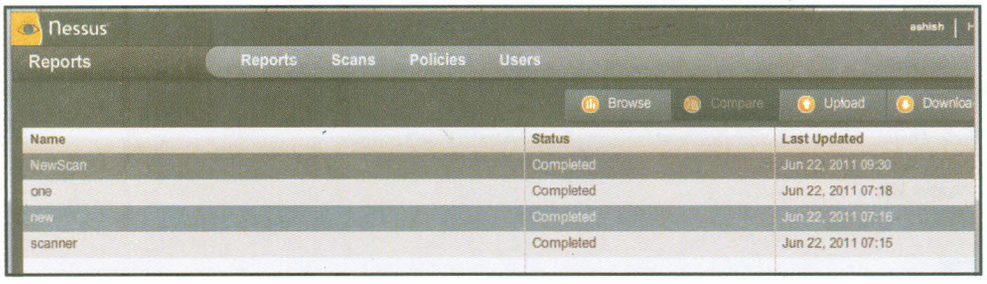


Fig. 35

Click on Reports once the scan is complete to see the report, then Click on Download.

Fig. 36

We can now see the report, it tells us about the host information and also the Number of vulnerabilities.

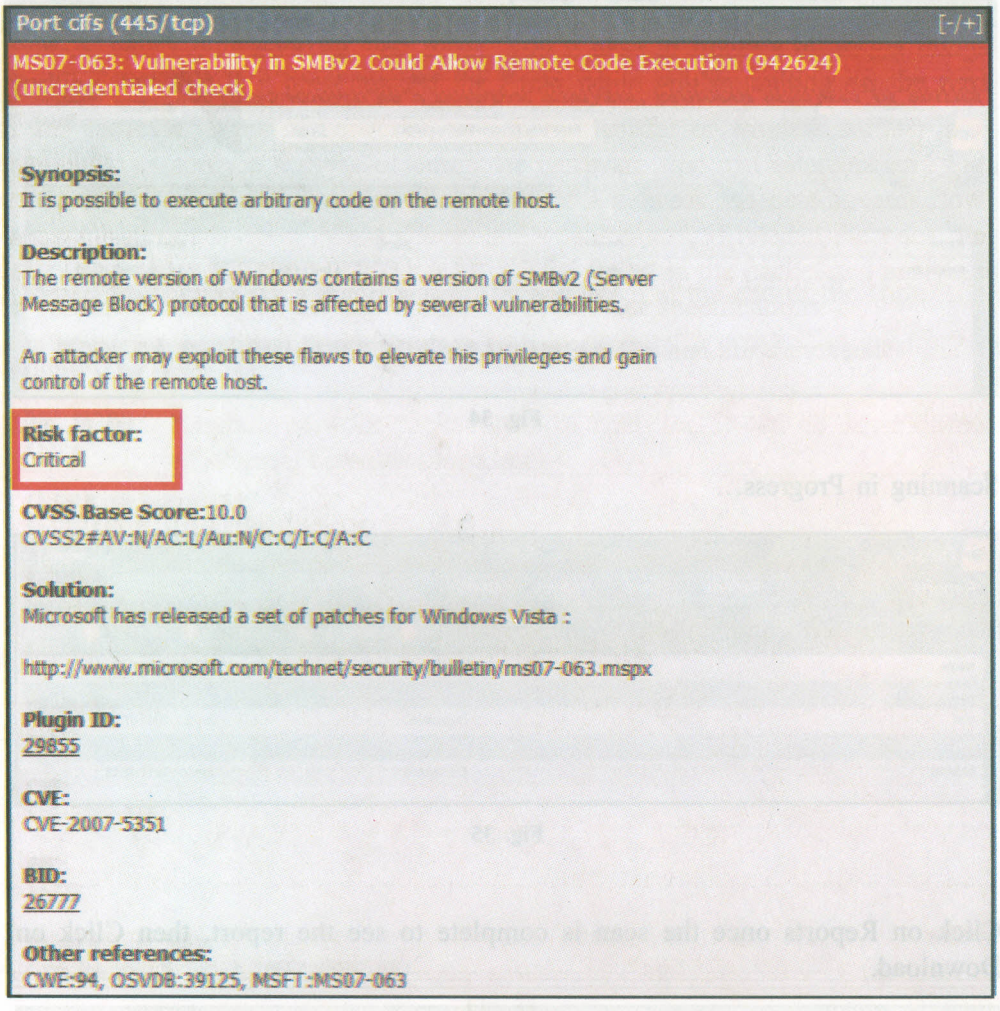


Fig. 37

We can scroll to see the Critical vulnerability found with this host and then resolve it accordingly.

We tried with "Internal Scan" policy. We can add a custom policy based on our requirements, plenty of tutorials and documentation is available online in this regard.

3.5 FORMAL VULNERABILITY ASSESSMENT

- An Enterprise LAN consists of set of Network zones inter-connected through access switches (L3 switches/Routers)
- Global security policy of an enterprise LAN is defined as a set of rules to "permit"/"deny" access to network services between various zones.
  - Policy specification may be incomplete; may not explicitly cover all the service access paths in the network.
- Access control based security mechanisms implement the global policy through a set of device specific access control rules-ACL (e.g. Cisco standard ACL in a distributed manner.
  - Implementation must be complete

For e.g. consider the following scenario:

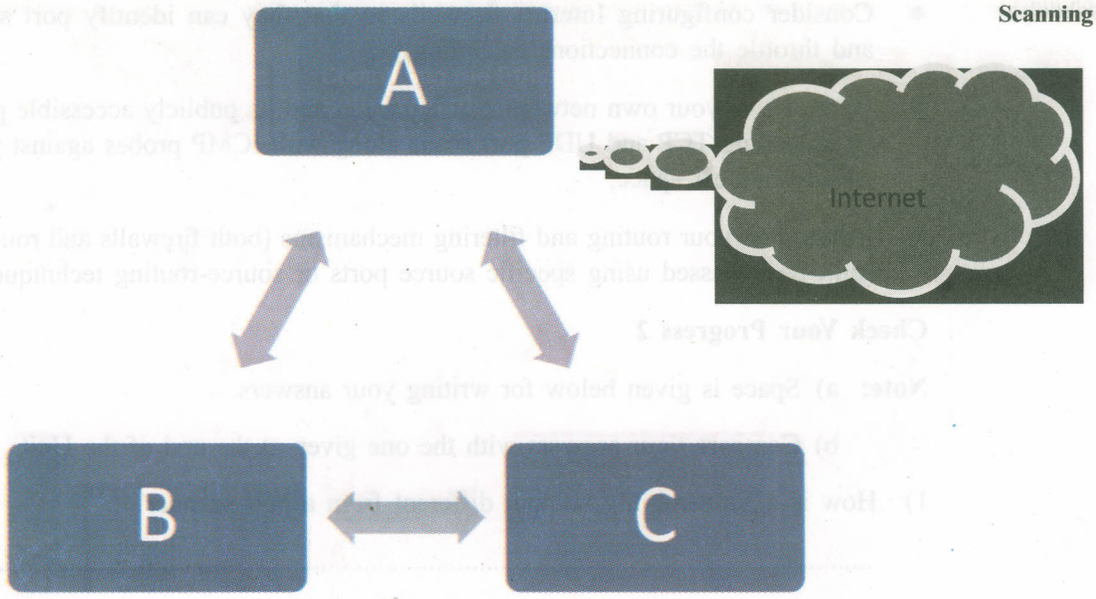


Fig. 38

- Three network zones A, B and C.
- A is connected to the Internet.

Security Policy : http access from B is NOT allowed.

ACL Implementation: Apply Rule ~http(B), i.e. outgoing http not allowed from B.

Now, in order to make sure that the security policy is adhered to we have a few challenges

Challenge 1: To verify the security implementations (distribution of ACLs in the switches) conforming to the Global Policy

- One might be able to access Internet from B via ssh/telnet to C and thereby accessing http.

Challenge 2: To analyze whether a correct security implementation conforms to the global policy under dynamic changes in network topology.

- If any link is down at a particular point of time; it will unable to pass the "allowed/permit" service packets through it. Fine for "restricted/deny" packets .

Challenge 3: Integrating verification and fault analysis module in a common framework and along with a query analysis module.

Thus to meet the above challenges formal methods of verification and analysis of networks have been develop. An interesting way to solve this problem is via a SAT solver based approach, as it reduces the verification problem to a Boolean formula f and checks its satisfiability using appropriate SAT or QBF-SAT solvers.

finSAT: Formal Integrated Network Security Analysis Tool, is one such step in this direction. It is a research project being carried out at School of Information Technology, IIT Kharagpur.

Although Satisfiability analysis is a NP complete problem,

but still is very useful due to tremendous time-tradeoffs

of modern SAT solvers.

3.6 NETWORK SCANNING COUNTERMEASURES

- Filter inbound ICMP message types at border routers and firewalls. This forces attackers to use full-blown TCP port scans against all of your IP addresses to map your network correctly.

- Consider configuring Internet firewalls so that they can identify port scans and throttle the connections accordingly.
- Be aware of your own network configuration and its publicly accessible ports by launching TCP and UDP port scans along with ICMP probes against your own IP address space.
- Ensure that your routing and filtering mechanisms (both firewalls and routers) can't be bypassed using specific source ports or source-routing techniques.

Check Your Progress 2

Note: a) Space is given below for writing your answers.

b) Compare your answers with the one given at the end of the Unit.

1) How is a Vulnerability scanner different from a port scanner?

.....

.....

.....

.....

.....

.....

.....

2) What Network Scanning Countermeasures are necessary?

.....

.....

.....

.....

.....

.....

.....

3.7 LET US SUM UP

Different IP network scanning methods allow you to test and effectively identify vulnerable network components. These scanning methods may be classified into:

- Port scanners (TCP/UDP)
- Vulnerability Scanners (Nesus Scanner)
- Protocol Analyzers/Sniffers (Ethereal)
- Formal assessment tools (finSAT)

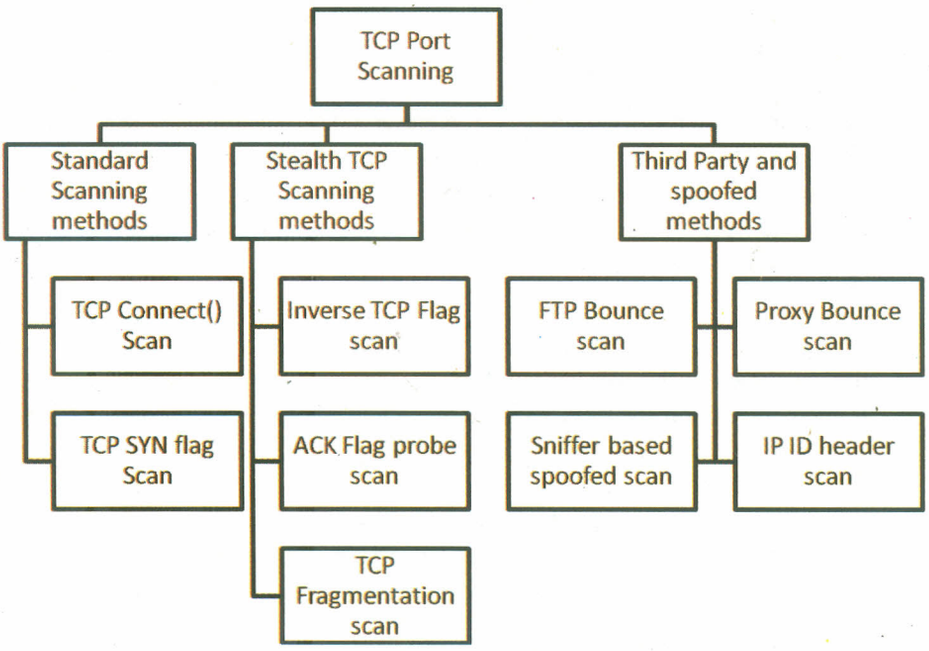
These may be used to remove prospective vulnerabilities and hence strengthen the security of any network, but at the same time can be also sued by an attacker to attack and cause disruption. Thus, it is also necessary to take preventive counter measures to restrict these tools also.

3.8 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) TCP Port Scanning

Accessible TCP ports can be identified by port scanning target IP addresses.



2) ICMP Probing

The Internet Control Message Protocol (ICMP) is one of the core protocols of the Internet Protocol Suite. It is chiefly used by the operating systems of networked computers to send error messages indicating, for example, that a requested service is not available or that a host or router could not be reached. ICMP can also be used to relay query messages. It can be used to identify potentially weak and poorly connected networks.

The following types of ICMP messages are useful for performing network scans:

Type 8 (echo request)

Echo request messages are also known as ping packets. You can use a scanning tool such as nmap to perform ping sweeping and easily identify hosts that are accessible.

Type 13 (timestamp request)

A timestamp request message requests system time information from the target host. The response is in a decimal format and is the number of milliseconds elapsed since midnight GMT.

Type 15 (information request)

The ICMP information request message was intended to support self-configuring systems such as diskless workstations at boot time, to allow them to discover their network address. Protocols such as RARP, BOOTP or DHCP do so more robustly, so type 15 messages are rarely used.

Type 17 (subnet address mask request)

An address mask request message reveals the subnet mask used by the target host. This information is useful when mapping networks and identifying the size of subnets and network spaces used by organizations.

3) We are using the UDP protocol instead of TCP. While this protocol is simpler, scanning it is actually significantly more difficult. This is because open ports don't have to send an acknowledgement in response to our probe and closed ports aren't even required to send an error packet. Fortunately, most hosts do send an ICMP_PORT_UNREACH error when you send a packet to a closed UDP port. Thus you can find out if a port is NOT open and by exclusion determine which ports which are. Neither UDP packets, nor the ICMP errors are guaranteed to arrive, so UDP scanners of this sort must also implement retransmission of packets that appear to be lost (or you will get a bunch of false positives).

Check Your Progress 2

1) The goal of running a vulnerability scanner is to identify devices on our network that are open to known vulnerabilities. One issue with vulnerability scanners is their impact on the devices they are scanning. On the one hand we want the scan to be able to be performed in the background without affecting the device. On the other hand, we want to be sure that the scan is thorough. Often, in the interest of being thorough and depending on how the scanner gathers its information or verifies that the device is vulnerable, the scan can be intrusive and cause adverse affects and even system crashes on the device being scanned.

2) **Network Scanning Countermeasures**

- Filter inbound ICMP message types at border routers and firewalls. This forces attackers to use full-blown TCP port scans against all of your IP addresses to map your network correctly.
- Consider configuring Internet firewalls so that they can identify port scans and throttle the connections accordingly.
- Be aware of your own network configuration and its publicly accessible ports by launching TCP and UDP port scans along with ICMP probes against your own IP address space.
- Ensure that your routing and filtering mechanisms (both firewalls and routers) can't be bypassed using specific source ports or source-routing techniques.

Student Satisfaction Survey

Student Satisfaction Survey of IGNOU Students

ignou THE PEOPLE'S UNIVERSITY

Enrollment No.	
Mobile No.	
Name	
Programme of Study	
Year of Enrolment	
Age Group	☐ Below 30 ☐ 31-40 ☐ 41-50 ☐ 51 and above
Gender	☐ Male ☐ Female
Regional Centre	
States	
Study Center Code	

Please indicate how much you are satisfied or dissatisfied with the following statements

Sl. No.	Questions	Very Satisfied	Satisfied	Average	Dissati-sfied	Very Dissati-sfied
1.	Concepts are clearly explained in the printed learning material	☐	☐	☐	☐	☐
2.	The learning materials were received in time	☐	☐	☐	☐	☐
3.	Supplementary study materials (like video/audio) available	☐	☐	☐	☐	☐
4.	Academic counselors explain the concepts clearly	☐	☐	☐	☐	☐
5.	The counseling sessions were interactive	☐	☐	☐	☐	☐
6.	Changes in the counseling schedule were communicated to you on time	☐	☐	☐	☐	☐
7.	Examination procedures were clearly given to you	☐	☐	☐	☐	☐
8.	Personnel in the study centers are helpful	☐	☐	☐	☐	☐
9.	Academic counseling sessions are well organized	☐	☐	☐	☐	☐
10.	Studying the programme/course provide the knowledge of the subject	☐	☐	☐	☐	☐
11.	Assignments are returned in time	☐	☐	☐	☐	☐
12.	Feedbacks on the assignments helped in clarifying the concepts	☐	☐	☐	☐	☐
13.	Project proposals are clearly marked and discussed	☐	☐	☐	☐	☐
14.	Results and grade card of the examination were provided on time	☐	☐	☐	☐	☐
15.	Overall, I am satisfied with the programme	☐	☐	☐	☐	☐
16.	Guidance from the programme coordinator and teachers from the school	☐	☐	☐	☐	☐

After filling this questionnaire send it to:
Programme Coordinator, School of Vocational Education and Training,
Room no. 19, Block no. 1, IGNOU, Maidangarhi, New Delhi- 110068

MPDD-IGNOU/P.O.2T/August,2011

ISBN-978-81-266-5515-8