Chapter 2

NETWORK AND SYSTEM SECURITY

Anoop Singhal

Abstract: This chapter discusses the elements of computer security such as authorization, authentication and integrity. It presents threats against networked applications such as denial of service attacks and protocol attacks. It also presents a brief discussion on firewalls and intrusion detection systems

Key words: computer virus, worms, DOS attacks, firewall, intrusion detection

Computer security is of importance to a wide variety of practical domains ranging from banking industry to multinational corporations, from space exploration to the intelligence community and so on. The following principles are the foundation of a good security solution:

- Authentication: The process of establishing the validity of a claimed identity.
- Authorization: The process of determining whether a validated entity is allowed access to a resource based on attributes, predicates, or context.
- Integrity: The prevention of modification or destruction of an asset by an unauthorized user.
- Availability: The protection of assets from denial-of-service threats that might impact system availability.
- Confidentiality: The property of non-disclosure of information to unauthorized users.
- Auditing: The property of logging all system activities

Computer security attempts to ensure the confidentiality, integrity and availability of computing resources. The principal components of a computer that need to be protected are hardware, software and the communication links. This chapter describes different kind of threats related to computer security and protection mechanisms that have been developed to protect the different components.

1.   VIRUSES AND RELATED THREATS

This section briefly discusses a variety of software threats. We first present information about computer viruses and worms followed by techniques to handle them.

A virus is a program that can "infect" other programs by modifying them and inserting a copy of itself into the program. This copy can then go to infect other programs. Just like its biological counterpart, a computer virus carries in its instructional code the recipe for making perfect copies of itself. A virus attaches itself to another program and then executes secretly when the host program is run.

During it lifetime a typical virus goes through the following stages:

Dormant Phase: In this state the virus is idle waiting for some event to happen before it gets activated. Some examples of these events are date/timestamp, presence of another file or disk usage reaching some capacity.

Propagation Phase: In this stage the virus makes an identical copy of itself and attaches itself to another program. This infected program contains the virus and will in turn enter into a propagation phase to transmit the virus to other programs.

Triggering Phase: In this phase the virus starts performing the function it was intended for. The triggering phase can also be caused by a set of events.

Execution Phase: In this phase the virus performs its function such as damaging programs and data files.

1.1 Types of Viruses

The following categories give the most significant types of viruses.

Parasitic Virus: This is the most common kind of virus. It attaches itself to executable files and replicates when that program is executed.

Memory Resident Virus: This kind of virus resides in main memory. When ever a program is loaded into memory for execution, it attaches itself to that program.

Boot Sector Virus: This kind of virus infects the boot sector and it spreads when the system is booted from the disk.

Stealth Virus: This is a special kind of virus that is designed to evade itself from detection by antivirus software.

Polymorphic virus: This kind of virus that mutates itself as it spreads from one program to the next, making it difficult to detect using the "signature" methods.

1.2 Macro Viruses

In recent years macro viruses have become quite popular. These viruses exploit certain features found in Microsoft Office Applications such as MS Word or MS Excel. These applications have a feature called *macro* that people use to automate repetitive tasks. The macro is written in a programming language such as Basic. The macro can be set up so that it is invoked when a certain function key is pressed. Certain kinds of macros are auto execute, they are automatically executed upon some events such as starting the execution of a program or opening of a file. These auto execution macros are often used to spread the virus. New version of MS Word provides mechanisms to protect itself from macro virus. One example of this tool is a Macro Virus Protection tool that can detect suspicious Word files and alert the customer about a potential risk of opening a file with macros.

1.3 E-mail Viruses

This is a new kind of virus that arrives via email and it uses the email features to propagate itself. The virus propagates itself as soon as it is activated (typically by opening the attachment) and sending an email with the attachment to all e-mail addresses known to this host. As a result these viruses can spread in a few hours and it becomes very hard for anti-virus software to respond before damage is done.

1.4   Worms

A virus typically requires some human intervention (such as opening a file) to propagate itself. A worm on the other hand typically propagates by itself. A worm uses network connections to propagate from one machine to another. Some examples of these connections are:

Electronic mail facility
Remote execution facility
Remote login facility

A worm will typically have similar phases as a virus such as dormant phase, a propagation phase, a triggering phase and an execution phase. The propagation phase for a worm uses the following steps:

Search the host tables to determine other systems that can be infected.
Establish a connection with the remote system
Copy the worm to the remote system and cause it to execute

Just like virus, network worms are also difficult to detect. However, properly designed system security applications can minimize the threat of worms.

1.5   The Morris Worm

This worm was released into the internet by Robert Morris in 1998. It was designed to spread on UNIX systems and it used a number of techniques to propagate. In the beginning of the execution, the worm would discover other hosts that are known to the current host. The worm performed this task by examining various list and tables such as machines that are trusted by this host or user's mail forwarding files. For each discovered host, the worm would try a number of methods to login to the remote host:

Attempt to log on to a remote host as a legitimate user.
Use the finger protocol to report on the whereabouts of a remote user.
Exploit the trapdoor of a remote process that sends and receives email.

1.6   Recent Worm Attacks

One example of a recent worm attack is the Code Red Worm that started in July 2001. It exploited a security hole in the Microsoft Internet

Information Server (IIS) to penetrate and spread itself. The worm probes random IP addresses to spread to other hosts. Also during certain periods of times it issues denial of service attacks against certain web sites by flooding the site with packets from several hosts. Code Red I infected nearly 360,000 servers in 14 hours. Code Red II was a second variant that targeted Microsoft IIS.

In late 2001, another worm called Nimda appeared. The worm spread itself using different mechanisms such as
Client to client via email
From web server to client via browsing of web sites
From client to Web server via exploitation of Microsoft IIS vulnerabilities

The worm modifies Web documents and certain executables files on the infected system.

1.7   Virus Counter Measures

Early viruses were relatively simple code fragments and they could be detected and purged with simple antivirus software. As the viruses got more sophisticated the antivirus software packages have got more complex to detect them.

There are four generations of antivirus software:

First Generation: This kind of scanner requires a specific signature to identify a virus. They can only detect known viruses.
Second Generation: This kind of scanner does not rely on a specific signature. Rather, the scanner uses heuristic rules to search for probable virus infections. Another second generation approach to virus detection is to use integrity checking. For example, a checksum can be appended to every program. If a virus infects the program without changing the checksum, then an integrity check will detect the virus.
Third Generation: These kind of programs are memory resident and they identify a virus by its actions rather than by its structure. The advantage of this approach is that it is not necessary to generate signature or heuristics. This method works by identifying a set of actions that indicate some malicious work is being performed and then to intervene.

Fourth Generation: These kind of packages consist of a variety of antivirus techniques that are used in conjunction. They including scanning, access control capability which limits the ability of a virus to penetrate the system and update the files to propagate the infection.

2.   PRINCIPLES OF NETWORK SECURITY

In the modern world we interact with networks on a daily basis such as when we perform banking transactions, make telephone calls or ride trains and planes. Life without networks would be considerably less convenient and many activities would be impossible. In this chapter, we describe the basics of computer networks and how the concepts of confidentiality, integrity and availability can be applied for networks.

2.1   Types of Networks and Topologies

A network is a collection of communicating hosts. There are several types of networks and they can be connected in different ways. This section provides information on different classes of networks.

a) Local Area Networks: A local area network (or LAN) covers a small distance, typically within a single building. Usually a LAN connects several computers, printers and storage devices. The primary advantage of a LAN to users is that it provides shared access to resources such programs and devices such as printers.
b) Wide Area Networks: A wide are network differs from a local area network in terms of both size and distance. It typically covers a wide geographical area. The hosts on a WAN may belong to a company with many offices in different cities or they may be a cluster of independent organizations within a few miles of each other who would like to share the cost of networking.   Therefore a WAN could be controlled by one organization or it can be controlled by multiple organizations.
c) Internetworks (Internets): Network of networks or internet is a connection of two or more separate networks in that they are separately managed and controlled. The Internet is a collection of networks that is loosely controlled by the Internet Society. The Internet Society enforces certain minimal rules to make sure that all users are treated fairly.

2.2   Network Topologies

The security of a network is dependent on its topology. The three different topologies are as follows.

a) Common Bus: Conceptually, a common bus is a single wire to which each node of a LAN is connected. In a common bus, the information is broadcast and nodes must continually monitor the bus to get the information addressed to it.

b) Star or Hub: In this topology each node is connected to a central "traffic controller" node. All communication flows from the source node to the traffic controller node and from the traffic controller node to the other nodes.

c) Ring: In this architecture, each node receives many messages, scans each and removes the one designated for itself. In this topology, there is no central node. However, there is one drawback with this architecture. If a node fails to pass a message that it has received, the other nodes will not be able to receive that information.

3.   THREATS IN NETWORKS

Network security has become important due to the inter-connection of computers and the rise of the internet. This section describes some of the popular network threats.

a) **Spoofing:** By obtaining the network authentication credentials of an entity (such as a user or a process) permits an attacker to create a full communication under the entity's identity. Examples of spoofing are masquerading and man-in-the-middle attack.

b) **Masquerade:** In a masquerade a user who is not authorized to use a computer pretends to be a legitimate user. A common example is URL confusion. Thus abc.com, abc.org or abc.net might be three different organizations or one legitimate organization and two masquerade attempts from some one who registered similar names.

c) **Phishing Attacks:** These attacks are becoming quite popular due to the proliferation of Web sites. In *phishing scams,* an attacker sets up a web site that masquerades as a legitimate site. By tricking a user, the phishing site obtains the user's cleartext password for the legitimate site. Phishing has proven to be quite effective in stealing user passwords.

d) **Session Hijacking:**   It is intercepting and carrying out a session begun by another entity. Suppose two people have entered into a session but then a third person intercepts the traffic and carries out a session in the name of the other person then this will be called session hijacking. For example, if an Online merchant used a wiretap to intercept packets between you and Amazon.com, the Online merchant can monitor the flow of packets. When the user has completed the order, Online merchant can intercept when the "Ready to check out" packet is sent and finishes the order with the user obtaining shipping address, credit card detail and other information. In this case we say the Online merchant has hijacked the session.

e) **Man-in-the-Middle Attack:** In this type of attack also one entity intrudes between two others. The difference between man-in-the-middle and hijacking is that a man-in-the-middle usually participates from the start of the session, whereas a session hijacking occurs after a session has been established. This kind of attack is frequently described in protocols. For example, suppose two parties want to exchange encrypted information. One party contacts the key server to get a secret key that will be used in the communication. The key server responds by sending the private key to both the parties. A malicious middleman intercepts the response key and then eavesdrop on the communication between the two parties.

f) **Web Site Defacement:** One of the most widely known attacks is the web site defacement attack. Since this can have a wide impact they are often reported in the popular press. Web sites are designed so that their code can be easily downloaded enabling an attacker to obtain the full hypertext document. One of the popular attacks against a web site is *buffer overflow.* In this kind of attack the attacker feeds a program more data than what is expected. A buffer size is exceeded and the excess data spills over adjoining code and data locations.

g) Message Confidentiality Threats:

- **Misdelivery:** Sometimes messages are misdelivered because of some flaw in the network hardware or software. We need to design mechanisms to prevent this.
- **Exposure:** To protect the confidentiality of a message, we must track it all the way from its creation to its disposal.
- **Traffic Flow Analysis:** Consider the case during wartime, if the enemy sees a large amount of traffic between the headquarters and a particular unit, the enemy will be able to infer that a significant action is being planned at that unit. In these situations there is a need to protect the contents of the message as well as how the messages are flowing in the network.

4.   DENIAL OF SERVICE ATTACKS

So far we have presented attacks that lead to failures of confidentiality or integrity. Availability attacks in network context are called denial of service attacks and they can cause a significant impact. The following are some sample denial of service attacks.

Connection Flooding: This is the most primitive denial-of-service attack. If an attacker sends so much data that the communication system cannot handle it then you are prevented from receiving any other data.

Ping of Death: Since ping requires the recipient to respond to the ping request, all that the recipient needs to do it to send a flood of pings to the intended victim.

Smurf: This is a variation of a ping attack. It uses the same vehicle, a ping packet with two extra twists. First, the attacker chooses a network of victims. The attacker spoofs the source address in the ping packet so that it appears to come from the victim. Then, the attacker sends this request to the network in broadcast mode by setting the last byte of the address to all 1s; broadcast mode packets are distributed to all the hosts.

Syn Flood: The attacker can deny service to the target by sending many SYN requests and never responding with ACKs. This fills up the victim's SYN_RECV queue. Typically, the SYN_RECV queue is quite small (about 10 to 20 entries). Attackers using this approach do one more thing, they spoof the nonexistent return address in the initial SYN packet.

4.1   Distributed Denial of Service Attacks

In order to perpetrate a distributed denial of service attack, an attacker does two things. In the first step, the attacker uses any convenient step (such as exploiting a buffer overflow) to plant a Trojan horse on a target machine. The installation of the Trojan horse as a file or a process does not attract any attention. The attacker repeats this process with many targets. Each of these targets then become what is known as a *zombie*. The target system carry out their work , unaware of the resident zombie.

At some point, the attacker chooses a victim and sends a signal to all the zombies to launch the attack. Then, instead of the victim trying to defend against one denial-of-service attack from one malicious host, the victim must try to counter n attacks from n zombies all acting at one.

4.2   Denial of Service Defense Mechanisms

The increased frequency of Denial of Service attacks has led to the development of numerous defense mechanisms. This section gives a summary of the taxonomy of defense mechanisms based on this paper.

Classification by Activity Level

Based on the activity level defense mechanisms can be classified into *preventive* and *reactive* mechanisms.

Preventive Mechanisms
The goal of these mechanisms is to either eliminate the possibility of DOS attacks or to endure the attack without denying services to legitimate clients.

Attack Prevention Mechanisms
These mechanisms modify the system configuration to eliminate the possibility of a DOS attack. System security mechanisms increase the overall security by guarding against illegitimate access from other machines. Examples of system security mechanisms include monitored access to the machine, install security patches, and firewall systems.

Protocol security mechanisms address the problem of bad protocol design which can be misused to exhaust the resources of a server by initiating a large number of such transactions. Classic misuse examples are the TCP

SYN attacks and the fragmented packet attack. An example of a protocol security mechanism is to have a design in which resources are committed to the client only after sufficient authentication is done.

Reactive Mechanisms

Reactive mechanisms alleviate the impact of an attack by *detecting* an attack and *responding* to it. Reactive mechanisms can be classified based on the mechanisms that they use *pattern detection, anomaly detection and hybrid detection.*

Mechanism with Pattern Attack Detection

In this method, signatures of known attacks are stored in a database. Each communication is monitored and compared with the database entries to discover the occurrence of an attack. Occasionally, the database is updated with new attack signatures. The obvious drawback of this detection mechanism is that it can only detect known attacks. On the other hand the main advantage is that known attacks are reliably detected and no false positives are encountered.

Mechanism with Anomaly Attack Detection

Mechanisms that deploy anomaly detection have a model of normal system behavior such as traffic or system performance. The current state of the system is periodically compared with the models to detect anomalies. The advantage of these techniques as compared to pattern detection is that unknown attacks can be discovered. However, they have to solve the following problems

Threshold setting: Anomalies are detected based on known settings. The setting of a low threshold leads to many false positives, while a high threshold reduces the sensitivity of the detection mechanism.

Model Update: Systems and communication patterns evolve with time and models need to be updated to reflect this change.

Mechanisms with Hybrid Attack Detection

These techniques combine the pattern based and anomaly-based detection, using data about attacks discovered through an anomaly detection mechanism to devise new attack signatures and update the database. Many intrusion detection systems use this technique but they have to be carefully designed.

Attack Response

The goal of the attack response is to mitigate the impact of attack on a victim machine so as to minimize the collateral damage to clients of the victim. Reactive mechanisms can be classified based on the response strategy into *agent identification, filtering and reconfiguration* approaches.

Agent Identification Mechanisms

These mechanisms provide the victim with information about the identity of the machines that are responsible to perform the attacks. This information can be combined with other response approaches to reduce the impact of attacks.

Filtering Mechanism

These techniques use the information provided by a detection mechanism to filter out the attack stream completely. A dynamically deployed firewall is an example of such a system.

Reconfiguration System

These mechanisms change the connectivity of the victim or the intermediate network topology to isolate the attack machines. One example of such a system is a reconfigurable overlay network.

5.   NETWORK SECURITY CONTROLS

Encryption

Encryption is the most important and versatile tool for network security experts. It can provide privacy, authenticity, integrity and limited access to data. Encryption can be applied wither between two hosts (link encryption) or between two applications (called end-to-end encryption).

Link Encryption

Link encryption protects the message in transit between two computers, however the message is in clear text inside the host. In this method, the data is encrypted before it is placed on the physical communication link. The encryption occurs at the lowest layer 1 or 2 in the OSI model. Similarly, decryption occurs when the data arrives at the receiving computer. This mechanism is really useful when the transmission point is of greatest vulnerability.

End-to-end Encryption

This mechanism provides security from one end of transmission to the other. In this case encryption is performed at the highest levels (layer 7 or layer 6).

Virtual Private Networks

Link encryption can be used to give the same protection to a user as of they are on a private network, even when their communication links are part of a public network.

Firewalls can be used to implement a Virtual Private Network (VPN). When a user first requests communication with a firewall, the user can request a VPN session with the firewall. The user and the firewall can agree on a session encryption key and the user can use that key for all subsequent communication. With a VPN all communication passes through an *encrypted tunnel.*

PKI and Certificates

A **public key infrastructure (PKI)** is a process created to enable users to implement public key cryptography usually in a distributed environment. PKI usually offers the following services

Create certificates that associates a user's identity to s cryptographic key
Distribute certificates from its database
Sign certificates to provide authenticity
Confirm a certificate if it is valid

PKI is really a set of policies, products and procedures with some flexibility for interpretation. The policies define a set of rules under which the system operates, it defines procedures on how to handle keys and how to manage the risks.

SSH Encryption

SSH is a protocol that is available under Unix and Windows 2000 that provides an authenticated and encrypted path to the shell or the OS command interpreter. SSH protects against spoofing attacks and modification of during in communication.

SSL Encryption

The Secure Sockets Layer (SSL) protocol was originally designed by Netscape to protect communication between a web browser and a server. It is also known as transport layer security (TLS). SSL interfaces between the applications (e.g. a browser) and the TCP/IP protocols to provide server authentication, client authentication and an encrypted communications channel between the client and server.

IPSec

The address space for Internet is running out as more machines and domain names are being added to the Internet. A new structure called **IPv6** solves this problem by providing a 64 bit address space to IP addresses. As part of IPv6, the Internet Engineering Task Force (IETF) adopted an **IP Security Protocol (IPSec) Suite** that addresses problems such as spoofing, eavesdropping and session hijacking. IPSec is implemented at the IP layer so it affects all layers above it. IPSec is somewhat similar to SSL, in that it supports authentication and confidentiality that does not necessitate significant changes either above it (in applications) or below it (in the TCP protocols). Just like SSL, it was designed to be independent of the cryptographic protocols and to allow the two communicating parties to agree on a mutually supported set of protocols.

The basis of IPSec is called a security association which is basically a set of security parameters that are required to establish a secured communication. Some examples of these parameters are:

Encryption algorithm and mode
Encryption Key
Authentication protocol and key
Lifespan of the association to permit long running sessions to select a new key
Address of the opposite end of an association

6.   FIREWALLS

6.1   What they are

A firewall is a device that filters all traffic between a "protected" network and the "outside" network. Generally, a firewall runs on a dedicated machine

which is a single point through which all the traffic is channeled. The purpose of a firewall is to keep "malicious" things outside a protected environment. For example, a firewall may impose a policy that will permit traffic coming from only certain IP addresses or users.

6.2 How do they work

There are different kind of firewalls.

Packet Filtering Firewall
It is the simplest form of firewall and in some situations it is most effective. It is based on certain packet address (source or destination) or transportation protocol (HTTP Web traffic).

Stateful Inspection Firewall
Filtering firewalls work on a packet at a time. They have no concept of "state" or "context" from one packet to next. A stateful inspection firewall is more sophisticated and it maintains state information to provide better filtering

Personal Firewall
A personal firewall is an application program that runs on a workstation or a PC to block unwanted traffic on a single workstation. A personal firewall can be configured to enforce a certain policy. For example, a user may decide that certain sites (for example a computer on a company network) is trustworthy and the firewall should allow traffic from only those sites. It is useful to combine a virus scanner with a personal firewall. For example, a firewall can direct all incoming email to a virus scanner, which examines every attachment the moment it reaches a particular host.

Application Proxy Gateway
An application proxy gateway is also called a *bastion host.* It is a firewall that simulates the proper effects of an application so that the application will receive only requests to act properly. The proxies on a firewall can be tailored to specific requirements such as logging details about the access. A proxy can demand strong authentication such as name, password and challenge-response.

Guard
A guard is another form of a sophisticated firewall. It receives protocol data units, interprets them and passes through the same or different protocol

data units that achieve either the same result or a modified result. The guard decides what services to perform on user's behalf in accordance with its available knowledge. The following example illustrates the use of a guard. A university wants all students to restrict the size of email messages to a certain number of words or characters. Although, this rule can be implemented by modifying email handlers, it is more easily done by monitoring the common point through which all email flows.

6.3   Limitations of Firewalls

Firewalls do not offer complete solutions to all computer security problems. A firewall can only protect the perimeter of its environment against attacks from outsiders. The following are some of the important points about firewall based protection

Firewalls can only protect if they control the entire perimeter. Even if one inside host connects to an outside address by a modem, the entire inside net can be vulnerable through the modem and its host.

Firewalls are the most visible parts of a network and therefore they are the most attractive target for attacks..

Firewalls exercise only minor control over the content of the packets that are admitted inside the network. Therefore inaccurate data or malicious code must be controlled by other means inside the parameter.

7.   BASICS OF INTRUSION DETECTION SYSTEMS

Perimeter controls such as a firewall, or authentication and access control act as the first line of defense. However, prevention is not a complete security solution. Intrusion Detection systems complement these preventive controls by acting as the next line of defense. An IDS is a sensor, like a smoke detector that raises an alarm if specific things occur. Intrusion Detection is the process of identifying and responding to malicious activities targeted at computing and network resources. It involves technology, people and tools. An Intrusion Detection System basically monitors and collects data from a target system that should be protected, processes and correlates the gathered information and initiate responses when an intrusion is detected.

8. CONCLUSIONS

Computer security ensures the confidentiality, integrity and availability of computing resources: hardware, software, network and data. These components have vulnerabilities and people exploit these vulnerabilities to stage attacks against these resources.

In this chapter we have discussed some of the salient features of security in networks and distributed applications. Since the world is becoming connected by computers the significance of network security will continue to grow. When a network and its components are designed and architectured well, the resulting system is quite resilient to attacks.

A lot of work is being done to enhance computer security. Products from vendor companies will lead to more secure boxes. There is a lot of research interests in the area of authentication, access control and authorizations. Another challenge for security is that networks are pervasive: cell phones, personal digital assistants and other consumer appliances are being connected. New applications lead to a new protocol development. There is a need to make sure that these protocols are tested for security flaws and that security measures are incorporated as needed. Intrusion Detection Systems and Firewalls have become popular products to secure networks. In the future, security of mobile code and web services will become an important issue as remote updates and patches become popular.

References

1. Pfleeger C.P. and Pfleeger S.L, "Security in Computing", Third Edition, Published by Prentice Hall, 2003
2. Bishop Matt, "Computer Security Art and Science", Addison-Wesley 2003, ISBN 0-201-44099-7
3. M. D. Abrams, S. Jajodia, and H. J. Podell, eds., Information Security: An Integrated Collection of Essays. IEEE Computer Society Press, 1995
4. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996
5. E. Amoroso, Fundamentals of Computer Security Technology, Prentice Hall, 1994
6. C.Kaufman, R.Perlman, and M.Speciner. Network Security: Private Communication in a Public World. 2nd ed. Prentice Hall, 2002
7. R.Anderson, Security Engineering, John Wiley and Sons 2001

8. Stallings W, "Network Security Essentials", Prentice Hall 2003, ISBN 0-13-035128-8
9. W. Cheswick, S.M. Bellovin, A. Rubin, "Firewalls and Internet Security", Addison Wesley, ISBN 0-201-63466-X, 2003