International Journal of Business, Management and Visuals (IJBMV), ISSN: 3006-2705
Volume 1, Issue 2, July-December, 2018, Available online at: https://ijbmv.com

Security in Networks

Palak Raina[1], Hitali Shah[2]

[1,2]Institute of Technology, Nirma University, India

ABSTRACT

In this report we describe what makes a network similar to and different from an application program or an operating system. In investigating networks, we have shown how the concepts of confidentiality, integrity, and availability apply in networked settings. At the same time, we have also added that the basic notions of identification and authentication, access control, accountability, and assurance are the basis for network security, just as they have been in other settings. Networking is growing and changing perhaps even faster than other computing disciplines. Consequently, this report is unlikely to present you with the most current technology, the latest attack, or the newest defense mechanism.

Keywords—Network Concepts, Threats in Networks, Types of attacks, Network Security Controls, Network Security Tools

NETWORK CONCEPTS

To study network threats and controls, we first must review some of the relevant networking terms and concepts.

Network

Figure shows a network in its simplest form, as two devices connected across some medium by hardware and software that enable the communication. In some cases, one device is a computer (sometimes called a ”server”) and the other is a simpler device (sometimes called a ”client”) enabled only with some means of input (such as a keyboard) and some means of output (such as a screen).

Fig. 1.Simple View of Network

Mode of Communication

A computer network implements communication between two endpoints. Data are communicated either in digital format (in which data items are expressed as discrete binary values) or analog (in which data items are expressed as points in a continuous range, using a medium like sound or electrical voltage). Computers typically store and process digital data, but some telephone and similar cable communications are in analog form (because telephones were originally designed to transmit voice). When the transmission medium expects to transfer analog data, the digital signals must be converted to analog for transmission and then back to digital for computation at the receiving end. Some mostly analog networks may even have some digital segments, so the analog signals are digitized more than once. These conversions are performed by a modem (the term is derived from modulator-demodulator), which converts a digital data stream to tones and back again.

ISO OSI Reference Model

The International Standards Organization (ISO) Open Systems Interconnection model consists of layers by which a network communication occurs. The OSI reference model contains the seven layers. We can think of the layers as creating an assembly line, in which each layer adds its own service to the communication. In concert, the layers represent the different activities that must be performed for actual transmission of a message. Separately, each layer serves a purpose; equivalent layers perform similar functions for the sender and receiver. For example, the sender's layer four affixes a header to a message, designating the sender, the receiver, and relevant sequence information. On the receiving end, layer four reads the header to verify that the message is for the intended recipient, and then removes this header.

Types of Networks

A network is a collection of communicating hosts.

Local Area Networks: local area network (or LAN) covers a small distance, typically within a single building. Usually a LAN connects several small computers, such as personal computers, as well as printers and perhaps some dedicated file storage devices. Figure shows the arrangement of a typical LAN.

Wide Area Networks: A wide area network, or WAN, differs from a local area network in terms of both size or distance (as its name implies, it covers a wider geographic area than does a LAN) and control or ownership (it is more likely not to be owned or controlled by a single body). Still, there tends to be some unifying principle to a WAN. The hosts

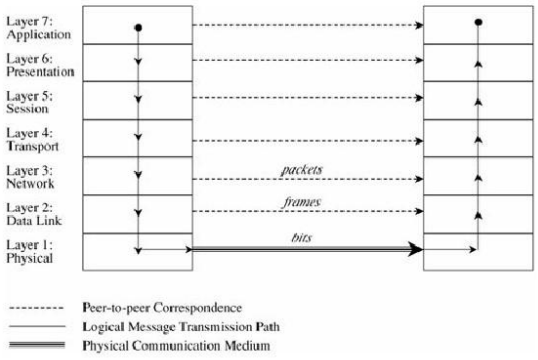

Fig. 2.ISO OSI Network Model

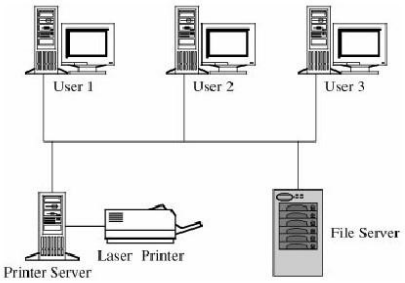

Fig. 3.Typical LAN

on a WAN may all belong to a company with many offices, perhaps even in different cities or countries, or they may be a cluster of independent organizations within a few miles of each other, who share the cost of networking hardware. These examples also show how WANs themselves differ. Some are under close control and maintain a high degree of logical and physical isolation (typically, these are WANs controlled by one organization), while others are only marriages of convenience.

THREATS IN NETWORKS

Threats aimed to compromise confidentiality, integrity, or availability, applied against data, software, and hardware by nature, accidents, non-malicious humans, and malicious attackers.

What Makes a Network Vulnerable?
An isolated home user or a stand-alone office with a few employees is an unlikely target for many attacks. But add a network to the mix and the risk rises sharply. Consider how a network differs from a stand-alone environment:

Anonymity: An attacker can mount an attack from thousands of miles away and never come into direct contact with the system, its administrators, or users. The potential attacker is thus safe behind an electronic shield. The attack can be passed through many other hosts in an effort to disguise the attack's origin.

Many points of attack both targets and origins: A simple computing system is a self-contained unit. Access controls on one machine preserve the confidentiality of data on that processor. However, when a file is stored in a network host remote from the user, the data or the file itself may pass through many hosts to get to the user.

Sharing: Because networks enable resource and workload sharing, more users have the potential to access networked systems than on single computers. Perhaps worse, access is afforded to more systems, so that access controls for single systems may be inadequate in networks.

Complexity of system: Reliable security is difficult, if not impossible, on a large operating system, especially one not designed specifically for security. A network combines two or more possibly dissimilar operating systems. Therefore, a network operating/control system is likely to be more complex than an operating system for a single computing system.

Unknown perimeter: A network's expandability also implies uncertainty about the network boundary. One host may be a node on two different networks, so resources on one network are accessible to the users of the other network as well. Although wide accessibility is an advantage, this unknown or uncontrolled group of possibly malicious users is a security disadvantage. A similar problem occurs when new hosts can be added to the network.

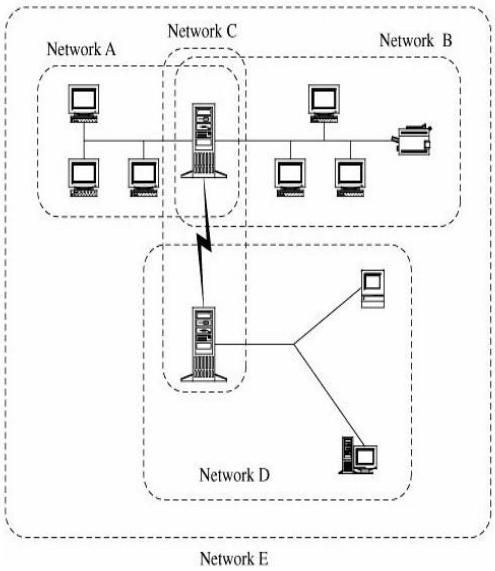

Fig. 4.Unclear Network Boundaries

Unknown path: Figure illustrates that there may be many paths from one host to another. Suppose that a user on host A1 wants to send a message to a user on host B3. That message might be routed through hosts C or D before arriving at host B3. Host C may provide acceptable security, but not D. Network users seldom have control over the routing of their messages.

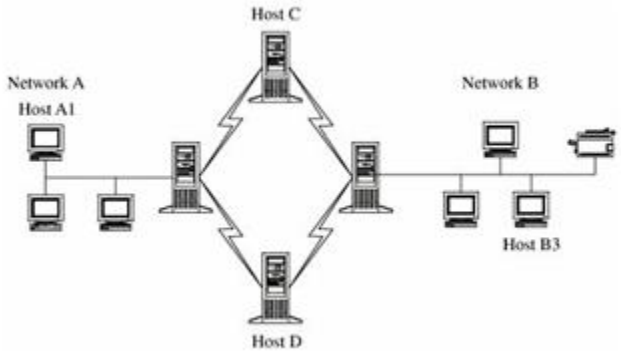

Fig. 5.Uncertain Message Routing in a Network

Who Attacks Networks?
we consider first the motives of attackers. Focusing on motive may give us some idea of who might attack a networked host or user. Four important motives are challenge or power, fame, money, and ideology.

Threat Precursors
How attackers prepare for attacks? - Investigate and plan These are threat precursors. Threat precursors techniques include:

Port scan: An easy way to gather network information is to use a port scan, a program that, for a particular IP address, reports which ports respond to messages and which of several known vulnerabilities seem to be present. Port scanning tells an attacker three things: which standard ports or services are running and responding on the target system, what operating system is installed on the target system, and what applications and versions of applications are present. This information is readily available for the asking from a networked system; it can be obtained quietly, anonymously, without identification or authentication, drawing little or no attention to the scan. Port scanning tools are readily available, and not just to the underground community.

Social engineering: Social engineering involves using social skills and personal interaction to get someone to reveal security-relevant information and perhaps even to do something that permits an attack. The point of social engineering is to persuade the victim to be helpful. The attacker often impersonates someone inside the organization who is in a bind: ”My laptop has just been stolen and I need to change the password I had stored on it,” or ”I have to get out a very important report quickly and I can’t get access to the following thing.” This attack works especially well if the attacker impersonates someone in a high position, such as the division vice president or the head of IT security.

Reconnaissance: we turn to how attackers perpetrate their attacks. Attackers do not ordinarily sit down at a terminal and launch an attack. A clever attacker investigates and plans before acting. Just as you might invest time in learning about a jewelry store before entering to steal from it, a network attacker learns a lot about a potential target before beginning the attack.

OS and application fingerprinting: It is finding out OS/app name, manufacturer and version by using peculiarities in OS/app responses Example: Attackers approach Earlier port scan reveals that port 80 HTTP is running. Attacker uses Telnet to send meaningless message to port 80. Attacker uses response (or a lack of it) to infer which of many possible OS/app it is. Each version of OS/app has its fingerprint (peculiarities) that reveals its identity (manufacturer, name, version)

Threats in Transit a network involves data in transit, we look first at the harm that can occur between a sender and a receiver.

Eavesdropping: The easiest way to attack is simply to listen in. An attacker can pick off the content of a communication passing in the clear. The term eavesdrop implies overhearing without expending any extra effort.

Wiretapping: A more hostile term is wiretap, which means intercepting communications through some effort. Passive wiretapping is just ”listening,” much like eavesdropping. But active wiretapping means injecting something into the communication. For example, Marvin could replace Manny’s communications with his own or create communications purported to be from Manny. Originally derived from listening in on telegraph and telephone communications, the term wiretapping usually conjures up a physical act by which a device extracts information as it flows over a wire. But in fact no

actual contact is necessary. A wiretap can be done covertly so that neither the sender nor the receiver of a communication knows that the contents have been intercepted.

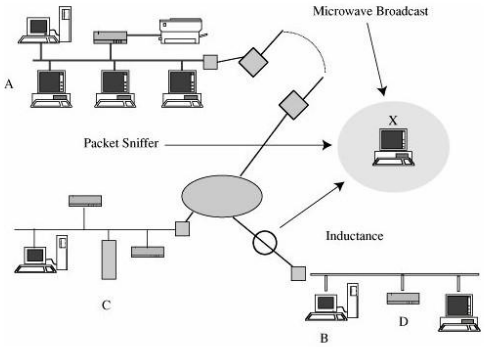

Fig. 6.Wiretap Vulnerabilities

Protocol Flaws

Internet protocols are publicly posted for scrutiny by the entire Internet community. Each accepted protocol is known by its Request for Comment (RFC) number. Many problems with protocols have been identified by sharp reviewers and corrected before the protocol was established as a standard. But protocol definitions are made and reviewed by fallible humans. Likewise, protocols are implemented by fallible humans. For example, TCP connections are established through sequence numbers. The client (initiator) sends a sequence number to open a connection, the server responds with that number and a sequence number of its own, and the client responds with the server's sequence number.

Impersonation

Impersonation is a more significant threat in a wide area network than in a local one. Local individuals often have better ways to obtain access as another user; they can, for example, simply sit at an unattended workstation. Still, impersonation attacks should not be ignored even on local area networks, because local area networks are sometimes attached to wider area networks without anyone's first thinking through the security implications.

Authentication Foiled by Guessing: A second source of password guesses is default passwords. Many systems are initially configured with default accounts having GUEST or ADMIN as login IDs; accompanying these IDs are well-known passwords such as "guest" or "null" or "password" to enable the administrator to set up the system. Administrators often forget to delete or disable these accounts, or at least to change the passwords.

Authentication Thwarted by Eavesdropping or Wiretapping: Because of the rise in distributed and client-server computing, some users have access privileges on several connected machines. To protect against arbitrary outsiders using these accesses, authentication is required between hosts. This access can involve the user directly, or it can be done automatically on behalf of the user through a host-to-host authentication protocol. In either case, the account and authentication details of the subject are passed to the destination host. When these details are passed on the network, they are exposed to anyone observing the communication on the network. These same authentication details can be reused by an impersonator until they are changed.

Authentication Foiled by Avoidance: Obviously, authentication is effective only when it works. A weak or flawed authentication allows access to any system or person who can circumvent the authentication. In a classic operating system flaw, the buffer for typed characters in a password was of fixed size, counting all characters typed, including backspaces for correction. If a user typed more characters than the buffer would hold, the overflow caused the operating system to bypass password comparison and act as if a correct authentication had been supplied. These flaws or weaknesses can be exploited by anyone seeking access.

Trusted Authentication: Finally, authentication can become a problem when identification is delegated to other trusted sources. For instance, a file may indicate who can be trusted on a particular host. Or the authentication mechanism for one system can "vouch for" a user. We noted earlier how the Unix. Rhosts, .rlogin, and /etc/hosts/equiv files indicate hosts or

users that are trusted on other hosts. While these features are useful to users who have accounts on multiple machines or for network management, maintenance, and operation, they must be used very carefully. Each of them represents a potential hole through which a remote user or a remote attacker can achieve access.

Spoofing

Guessing or otherwise obtaining the network authentication credentials of an entity (a user, an account, a process, a node, a device) permits an attacker to create a full communication under the entity's identity. Impersonation falsely represents a valid entity in a communication. Closely related is spoofing, when an attacker falsely carries on one end of a networked interchange. Examples of spoofing are masquerading, session hijacking, and man-in-the-middle attacks.

Masquerade: In a masquerade one host pretends to be another. A common example is URL confusion. Domain names can easily be confused, or someone can easily mistype certain names. Thus xyz.com, xyz.org, and xyz.net might be three different organizations, or one bona fide organization (for example, xyz.com) and two masquerade attempts from someone who registered the similar domain names. Names with or without hyphens (coca-cola.com versus cocacola.com) and easily mistyped names (l0pht.com versus lopht.com, or citibank.com versus citybank.com) are candidates for masquerading.

Session Hijacking: Session hijacking is intercepting and carrying on a session begun by another entity. Suppose two entities have entered into a session but then a third entity intercepts the traffic and carries on the session in the name of the other. Our example of Books-R-Us could be an instance of this technique. If Books Depot used a wiretap to intercept packets between you and Books-R-Us, Books Depot could simply monitor the information flow, letting Books-R-Us do the hard part of displaying titles for sale and convincing the user to buy. Then, when the user has completed the order, Books Depot intercepts the "I'm ready to check out" packet, and finishes the order with the user, obtaining shipping address, credit card details, and so forth. To Books-R-Us, the transaction would look like any other incomplete transaction: The user was browsing but for some reason decided to go elsewhere before purchasing. We would say that Books Depot had hijacked the session.

Man-in-the-Middle Attack: Man-in-the-middle attacks are frequently described in protocols. To see how an attack works, suppose you want to exchange encrypted information with your friend. You contact the key server and ask for a secret key with which to communicate with your friend. The key server responds by sending a key to you and your friend. One man-in-the-middle attack assumes someone can see and enter into all parts of this protocol. A malicious middleman intercepts the response key and can then eavesdrop on, or even decrypt, modify, and reencrypt any subsequent communications between you and your friend. This attack is depicted in Figure.

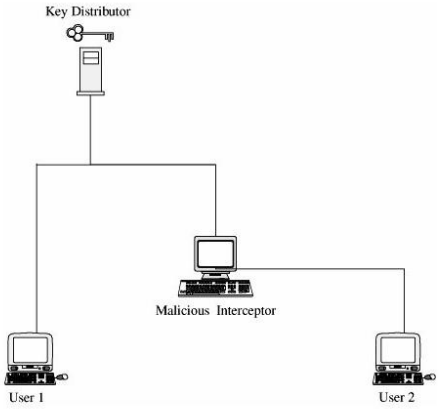

Fig. 7.Key Interception by a Man-in-the-Middle Attack

Message Confidentiality Threats

An attacker can easily violate message confidentiality (and perhaps integrity) because of the public nature of networks. Eavesdropping and impersonation attacks can lead to a confidentiality or integrity failure. Here we consider several other vulnerabilities that can affect confidentiality.

Misdelivery: Sometimes messages are misdelivered because of some flaw in the network hardware or software. Most frequently, messages are lost entirely, which is an integrity or availability issue. Occasionally, however, a destination

address is modified or some handler malfunctions, causing a message to be delivered to someone other than the intended recipient. All of these ”random” events are quite uncommon. More frequent than network flaws are human errors. It is far too easy to mistype an address such as 100064,30652 as 10064,30652 or 100065,30642, or to type ”idw” or ”iw” instead of ”diw” for David Ian Walker, who is called Ian by his friends. There is simply no justification for a computer network administrator to identify people by meaningless long numbers or cryptic initials when ”iwalker” would be far less prone to human error.

Exposure: To protect the confidentiality of a message, we must track it all the way from its creation to its disposal. Along the way, the content of a message may be exposed in temporary buffers; at switches, routers, gateways, and intermediate hosts throughout the network; and in the workspaces of processes that build, format, and present the message. In earlier chapters, we considered confidentiality exposures in programs and operating systems. All of these exposures apply to networked environments as well. Furthermore, a malicious attacker can use any of these exposures as part of a general or focused attack on message confidentiality.

Traffic Flow Analysis: Sometimes not only is the message itself sensitive but the fact that a message exists is also sensitive. For example, if the enemy during wartime sees a large amount of network traffic between headquarters and a particular unit, the enemy may be able to infer that significant action is being planned involving that unit. In a commercial setting, messages sent from the president of one company to the president of a competitor could lead to speculation about a takeover or conspiracy to fix prices. Or communications from the prime minister of one country to another with whom diplomatic relations were suspended could lead to inferences about a rapprochement between the countries. In these cases, we need to protect both the content of messages and the header information that identifies sender and receiver.

Message Integrity Threats

In many cases, the integrity or correctness of a communication is at least as important as its confidentiality. In fact for some situations, such as passing authentication data, the integrity of the communication is paramount. In other cases, the need for integrity is less obvious. Next we consider threats based on failures of integrity in communication.

Falsification of Messages: Increasingly, people depend on electronic messages to justify and direct actions. For example, if you receive a message from a good friend asking you to meet at the pub for a drink next Tuesday evening, you will probably be there at the appointed time. Likewise, you will comply with a message from your supervisor telling you to stop work on project A and devote your energy instead to project B. As long as it is reasonable, we tend to act on an electronic message just as we would on a signed letter, a telephone call, or a face-to-face communication.

Noise: Signals sent over communications media are subject to interference from other traffic on the same media, as well as from natural sources, such as lightning, electric motors, and animals. Such unintentional interference is called noise. These forms of noise are inevitable, and they can threaten the integrity of data in a message.

Format Failures: Network communications work because of well-designed protocols that define how two computers communicate with a minimum of human intervention. The format of a message, size of a data unit, sequence of interactions, even the meaning of a single bit is precisely described in a standard. The whole network works only because everyone obeys these rules. Almost everyone, that is. Attackers purposely break the rules to see what will happen. Or the attacker may seek to exploit an undefined condition in the standard. Software may detect the violation of structure and raise an error indicator. Sometimes, however, the malformation causes a software failure, which can lead to a security compromise, just what the attacker wants.

J. Web Site Vulnerabilities

A web site is especially vulnerable because it is almost completely exposed to the user. If you use an application program, you do not usually get to view the program’s code. With a web site, the attacker can download the site’s code for offline study over time. With a program, you have little ability to control in what order you access parts of the program, but a web attacker gets to control in what order pages are accessed, perhaps even accessing page 5 without first having run pages 1 through 4. The attacker can also choose what data to supply and can run experiments with different data values to see how the site will react. In short, the attacker has some advantages that can be challenging to control.

Buffer Overflows: Buffer overflow is alive and well on web pages, too.The attacker simply feeds a program far more data than it expects to receive. A buffer size is exceeded, and the excess data spill over into adjoining code and data locations. Perhaps the best-known web server buffer overflow is the file name problem known as iishack.To execute the procedure, an attacker supplies as parameters the site to be attacked and the URL of a program the attacker wants that server to execute.

Other web servers are vulnerable to extremely long parameter fields, such as passwords of length 10,000 or a long URL padded with space or null characters.

Dot-Dot-Slash: Web server code should always run in a constrained environment. Ideally, the web server should never have editors, xterm and Telnet programs, or even most system utilities loaded. By constraining the environment in this way, even if an attacker escapes from the web server application, no other executable programs will help the attacker use the web server's computer and operating system to extend the attack. The code and data for web applications can be transferred manually to a web server or pushed as a raw image. But many web applications programmers are nave. They expect to need to edit a web application in place, so they install editors and system utilities on the server to give them a complete environment in which to program. A second, less desirable, condition for preventing an attack is to create a fence confining the web server application. With such a fence, the server application cannot escape from its area and access other potentially dangerous system areas (such as editors and utilities). The server begins in a particular directory subtree, and everything the server needs is in that same subtree. Enter the dot-dot. In both Unix and Windows, '..' is the directory indicator for "predecessor." And '../..' is the grandparent of the current location. So someone who can enter file names can travel back up the directory tree one .. at a time. Cerberus Information Security analysts found just that vulnerability in the webhits.dll extension for the Microsoft Index Server.

Application Code Errors: A user's browser carries on an intricate, undocumented protocol interchange with applications on the web server. To make its job easier, the web server passes context strings to the user, making the user's browser reply with full context. A problem arises when the user can modify that context. To see why, consider our fictitious shopping site called CDs-R-Us, selling compact discs. At any given time, a server at that site may have a thousand or more transactions in various states of completion. The site displays a page of goods to order, the user selects one, the site displays more items, the user selects another, the site displays more items, the user selects two more, and so on until the user is finished selecting. Many people go on to complete the order by specifying payment and shipping information. But other people use web sites like this one as an online catalog or guide, with no real intention of ordering. For instance, they can use this site to find out the price of the latest CD from Cherish the Ladies; they can use an online book service to determine how many books by Iris Murdoch are in print. And even if the user is a bonafide customer, sometimes web connections fail, leaving the transaction incomplete. For these reasons, the web server often keeps track of the status of an incomplete order in parameter fields appended to the URL. These fields travel from the server to the browser and back to the server with each user selection or page request.

Server-Side Include: A potentially more serious problem is called a server-side include. The problem takes advantage of the fact that web pages can be organized to invoke a particular function automatically. For example, many pages use web commands to send an e-mail message in the "contact us" part of the displayed page. The commands, such as e-mail, if, goto, and include, are placed in a field that is interpreted in HTML.

K. Denial of Service applications. Availability attacks, sometimes called denial of-service or DOS attacks, are much more significant in networks than in other contexts. There are many accidental and malicious threats to availability or continued service.

Transmission Failure: Communications fail for many reasons. For instance, a line is cut. Or network noise makes a packet unrecognizable or undeliverable. A machine along the transmission path fails for hardware or software reasons. A device is removed from service for repair or testing. A device is saturated and rejects incoming data until it can clear its overload. Many of these problems are temporary or automatically fixed (circumvented) in major networks, including the Internet. However, some failures cannot be easily repaired.

A break in the single communications line to your computer (for example, from the network to your network interface card or the telephone line to your modem) can be fixed only by establishment of an alternative link or repair of the damaged one. The network administrator will say "service to the rest of the network was unaffected," but that is of little consolation to you.

Connection Flooding: The most primitive denial-of service attack is flooding a connection. If an attacker sends you as much data as your communications system can handle, you are prevented from receiving any other data.

Even if an occasional packet reaches you from someone else, communication to you will be seriously degraded. More sophisticated attacks use elements of Internet protocols. In addition to TCP and UDP, there is a third class of protocols,

called ICMP or Internet Control Message Protocols. Normally used for system diagnostics, these protocols do not have associated user applications.

ICMP Protocols Include:

- ping, which requests a destination to return a reply, intended to show that the destination system is reachable and functioning
- echo, which requests a destination to return the data sent to it, intended to show that the connection link is reliable (ping is actually a version of echo)
- destination unreachable, which indicates that a destination address cannot be accessed
- source quench, which means that the destination is becoming saturated and the source should suspend sending packets for a while These protocols have important uses for network management. But they can also be used to attack a system. The protocols are handled within the network stack, so the attacks may be difficult to detect or block on the receiving host. We examine how these protocols can be used to attack a victim. -Echo-Chargen This attack works between two hosts. Chargen is a protocol that generates a stream of packets; it is used to test the network's capacity. The attacker sets up a chargen process on host A that generates its packets as echo packets with a destination of host B. Then, host A produces a stream of packets to which host B replies by echoing them back to host A. This series puts the network infrastructures of A and B into an endless loop. If the attacker makes B both the source and destination address of the first packet, B hangs in a loop, constantly creating and replying to its own messages.
- Ping of Death A ping of death is a simple attack. Since ping requires the recipient to respond to the ping request, all the attacker needs to do is send a flood of pings to the intended victim. The attack is limited by the smallest bandwidth on the attack route. If the attacker is on a 10- megabyte (MB) connection and the path to the victim is 100 MB or more, the attacker cannot mathematically flood the victim alone. But the attack succeeds if the numbers are reversed: The attacker on a 100-MB connection can easily flood a 10-MB victim. The ping packets will saturate the victim's bandwidth.

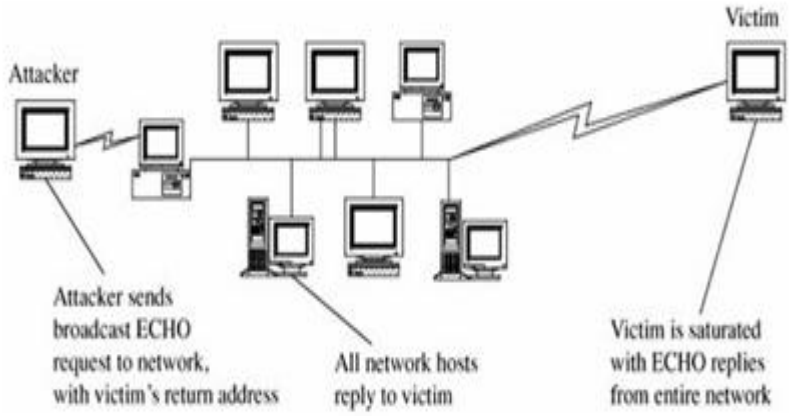

Fig. 8.Connection Flood

Syn Flood: Another popular denial-of-service attack is the syn flood. This attack uses the TCP protocol suite, making the session-oriented nature of these protocols work against the victim. For a protocol such as Telnet, the protocol peers establish a virtual connection, called a session, to synchronize the back-and-forth, command-response nature of the Telnet terminal emulation. A session is established with a three-way TCP handshake. Each TCP packet has flag bits, two of which are denoted SYN and ACK. To initiate a TCP connection, the originator sends a packet with the SYN bit on. If the recipient is ready to establish a connection, it replies with a packet with both the SYN and ACK bits on. The first party then completes the exchange to demonstrate a clear and complete communication channel by sending a packet with the ACK bit on, as shown in Figure.

Distributed Denial of Service

To perpetrate a distributed denial-of-service (or DDoS) attack, an attacker does two things, as illustrated in Figure.

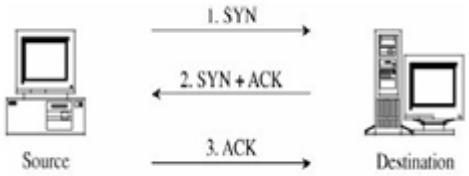

Fig. 9. SYN Flood

In the first stage, the attacker uses any convenient attack (such as exploiting a buffer overflow or tricking the victim to open and install unknown code from an e-mail attachment) to plant a Trojan horse on a target machine. That Trojan horse does not necessarily cause any harm to the target machine, so it may not be noticed. The Trojan horse file may be named for a popular editor or utility, bound to a standard operating system service, or entered into the list of processes (daemons) activated at startup. No matter how it is situated within the system, it will probably not attract any attention.

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's trying to defend against one denial-of-service attack from one malicious host, the victim must try to counter n attacks from the n zombies all acting at once. Not all of the zombies need to use the same attack; for instance, some could use smurf attacks and others, could use syn floods to address different potential weaknesses. In addition to their tremendous multiplying effect, distributed denial-of-service attacks are a serious problem because they are easily launched from scripts. Given a collection of denial-of-service attacks and a Trojan horse propagation method, one can easily write a procedure to plant a Trojan horse that can launch any or all of the denial of-service attacks.

M. Threats in Active or Mobile Code

Active code or mobile code is a general name for code that is pushed to the client for execution. Why should the web server waste its precious cycles and bandwidth doing simple work that the client's workstation can do? For example, suppose you want your web site to have bears dancing across the top of the page. To download the dancing bears, you could download a new image for each movement the bears take: one bit forward, two bits forward, and so forth. However, this approach uses far too much server time and bandwidth to compute the positions and download new images. A more efficient use of (server) resources is to download a program that runs on the client's machine and implements the movement of the bears. Since you

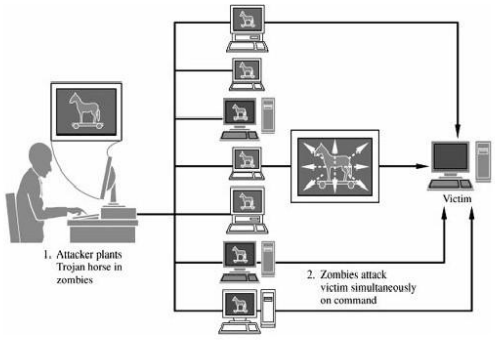

Fig. 10.Distributed Denial-of-Service Attack

have been studying security and are aware of vulnerabilities, you probably are saying to yourself, "You mean a site I don't control, which could easily be hacked by teenagers, is going to push code to my machine that will execute without my knowledge, permission, or oversight?" Welcome to the world of (potentially malicious) mobile code.

Cookies: A cookie is a data object that can be held in memory (a per-session cookie) or stored on disk for future access (a persistent cookie). Cookies can store anything about a client that the browser can determine: keystrokes the user types, the machine name, connection details (such as IP address), date and type, and so forth. On command a browser will send to a server the cookies saved for it. Per-session cookies are deleted when the browser is closed, but persistent cookies are retained until a set expiration date, which can be years in the future. Cookies provide context to a server. Using cookies, certain web pages can greet you with ”Welcome back, James Bond” or reflect your preferences, as in ”Shall I ship this order to you at 135 Elm Street?” But as these two examples demonstrate, anyone possessing someone’s cookie becomes that person in some contexts. Thus, anyone intercepting or retrieving a cookie can impersonate the cookie’s owner. What information about you does a cookie contain? Even though it is your information, most of the time you cannot tell what is in a cookie, because the cookie’s contents are encrypted under a key from the server. So a cookie is something that takes up space on your disk, holding information about you that you cannot see, forwarded to servers you do not know whenever the server wants it, without informing you. The philosophy behind cookies seems to be ”Trust us, it’s good for you.”

Script: Clients can invoke services by executing scripts on servers. Typically, a web browser displays a page. As the user interacts with the web site via the browser, the browser organizes user inputs into parameters to a defined script; it then sends the script and parameters to a server to be executed. But all communication is done through HTML. The server cannot distinguish between commands generated from a user at a browser completing a web page and a user’s handcrafting a set of orders. The malicious user can monitor the communication between a browser and a server to see how changing a web page entry affects what the browser sends and then how the server reacts. With this knowledge, the malicious user can manipulate the server’s actions. To see how easily this manipulation is done, remember that programmers do not often anticipate malicious behavior; instead, programmers assume that users will be benign and will use a program in the way it was intended to be used. For this reason, programmers neglect to filter script parameters to ensure that they are reasonable for the operation and safe to execute. Some scripts allow arbitrary files to be included or arbitrary commands to be executed. An attacker can see the files or commands in a string and experiment with changing them. A well-known attack against web servers is the escape-character attack.

Network Security Controls

There are several strategies for addressing security concerns, such as encryption for confidentiality and integrity, reference monitors for access control, and overlapping controls for defense. These strategies are also useful in protecting networks. This section presents many excellent defenses available to the network security engineer. Subsequent sections provide detailed explanations for three particularly important controls firewalls, intrusion detection systems, and encrypted e-mail.

Security Threat Analysis

First, we scrutinize all the parts of a system so that we know what each part does and how it interacts with other parts. Next, we consider possible damage to confidentiality, integrity, and availability. Finally, we hypothesize the kinds of attacks that could cause this damage. We can take the same steps with a network. We begin by looking at the individual parts of a network:

local nodes connected via local communications links to a local area network, which also has local
data storage local processes local devices

Impact of network architecture

Planning can be the strongest control. In particular, when we build or modify computer based systems, we can give some thought to their overall architecture and plan to ”build in” security as one of the key constructs. Similarly, the architecture or design of a network can have a significant effect on its security.

Segmentation: Just as segmentation was a powerful security control in operating systems, it can limit the potential for harm in a network in two important ways: Segmentation reduces the number of threats, and it limits the amount of damage a single vulnerability can allow. Assume your network implements electronic commerce for users of the Internet. The fundamental parts of your network may be

A web server, to handle users’ HTTP sessions
Application code, to present your goods and services for
Purchase
Database of goods, and perhaps an accompanying inventory to the count of stock on hand and being requested from
Suppliers
Database of orders taken

If all these activities were to run on one machine, your network would be in trouble: Any compromise or failure of that machine would destroy your entire commerce capability. A more secure design uses multiple segments, as shown in Figure. Suppose one piece of hardware is to be a web server box exposed to access by the general public. To reduce the risk of attack from outside the system, that box should not also have other, more sensitive, functions on it, such as user authentication or access to a sensitive data repository. Separate segments and servers corresponding to the principles of least privilege and encapsulation reduce the potential harm should any subsystem be compromised.

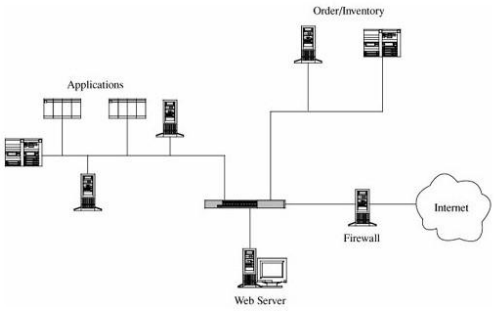

Fig. 11. Segmented Architecture

Redundancy: Another key architectural control is redundancy: allowing a function to be performed on more than one node, to avoid ”putting all the eggs in one basket.” For example, the design of Figure has only one web server; lose it and all connectivity is lost. A better design would have two servers, using what is called failover mode. In failover mode the servers communicate with each other periodically, each determining if the other is still active. If one fails, the other takes over processing for both of them. Although performance is cut approximately in half when a failure occurs, at least some processing is being done.

Single Points of Failure: Ideally, the architecture should make the network immune to failure. In fact, the architecture should at least make sure that the system tolerates failure in an acceptable way (such as slowing down but not stopping processing, or recovering and restarting incomplete transactions). One way to evaluate the network architecture’s tolerance of failure is to look for single points of failure. That is, we should ask if there is a single point in the network that, if it were to fail, could deny access to all or a significant part of the network. So, for example, a single database in one location is vulnerable to all the failures that could affect that location. Good network design eliminates single points of failure. Distributing the database placing copies of it on different network segments, perhaps even in different physical locationscan reduce the risk of serious harm from a failure at any one point. There is often substantial overhead in implementing such a design; for example, the independent databases must be synchronized. But usually we can deal with the failure-tolerant features more easily than with the harm caused by a failed single link. Architecture plays a role in implementing many other controls.

Encryption

Encryption is probably the most important and versatile tool for a network security expert. We have seen in earlier chapters that encryption is powerful for providing privacy, authenticity, integrity, and limited access to data. Because networks often involve even greater risks, they often secure data with encryption, perhaps in combination with other controls.

Link Encryption: In link encryption, data are encrypted just before the system places them on the physical communications link. In this case, encryption occurs at layer 1 or 2 in the OSI model. (A similar situation occurs with TCP/IP protocols.) Similarly, decryption occurs just as the communication arrives at and enters the receiving computer. A model of link encryption is shown in Figure. Link encryption is invisible to the user.

The encryption becomes a transmission service performed by a low-level network protocol layer, just like message routing or transmission error detection. Figure shows a typical link encrypted message, with the shaded fields encrypted. Because some of the data link header and trailer is applied before the block is encrypted, part of each of those blocks is shaded. As the message M is handled at each layer, header and control information is added on the sending side and removed on the

receiving side. Hardware encryption devices operate quickly and reliably; in this case, link encryption is invisible to the operating system as well as to the operator.

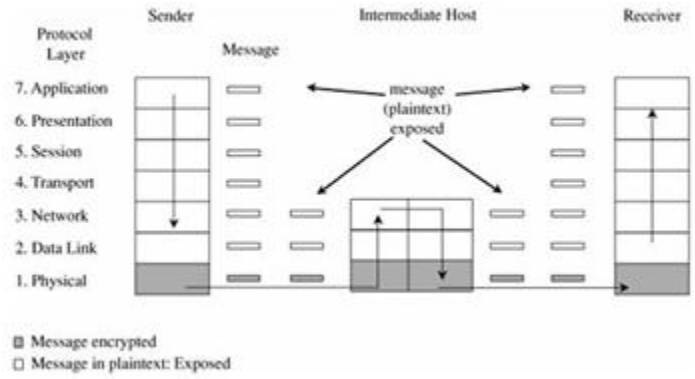

Fig. 12. Link Encryption

End-to-End Encryption: As its name implies, end-to-end encryption provides security from one end of a transmission to the other. The encryption can be applied by a hardware device between the user and the host. Alternatively, the encryption can be done by software running on the host computer. In either case, the encryption is performed at the highest levels (layer 7, application, or perhaps at layer 6, presentation) of the OSI model. A model of end-to-end encryption is shown in Figure.

Since the encryption precedes all the routing and transmission processing of the layer, the message is transmitted in encrypted form throughout the network. The encryption addresses potential flaws in lower layers in the transfer model.

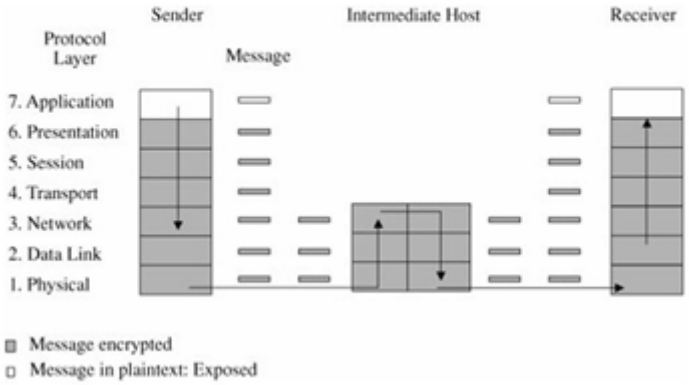

Fig. 13.End-to-End Encryption

SSH Encryption

SSH provides a secure channel over an unsecured network in a client-server architecture that connects SSH client application with an SSH server. Allows remote login and other network services to operate securely over an unsecured network. It was Originally defined for Unix. It Replaced insecure utilities for remote access such as Telnet / rlogin / rsh. Those protocols send information, notably passwords, in plaintext, making them susceptible to interception. So SSH protects against spoofing attacks (falsifying one end of communication, incl. masquerading, sesssion hijacking, MITM) message modification / falsification.

SSL protocol

The SSL (Secure Sockets Layer) protocol was originally designed by Netscape to protect communication between a web browser and server. It is also Known as TLS- transport layer security. To use SSL, the client requests an SSL session, then server responds with its public key certificate, the client returns part of a symmetric session key encrypted under the server's public key. Finally both the server and client compute the session key, and then they switch to encrypted communication, using the shared session key.

Signed Code

Someone can place malicious active code on a web site to be downloaded by unsuspecting users So for that partial solution is code signed by TTP (trusted third party) TTP appends digital signature to piece of code and then PKI can be used by prospective code users to validate signature.

Strong authentication

Networked environments as well as both ends of communication need authentication Strong authentication controls include: One-time passwords, Challenge-response systems, Digital distributed authentication, Kerberos authentication system.

One-time passwords: One time Passwords controls wiretapping and spoofing, prevents reuse of pwd captured by wiretapper, Strong authentication prevents spoofing (incl. masquerading, session hijacking, MITM).Each password is used only once or User has password token (PT)that is a device to randomly generate new pwd (e.g.) every minute. Challenge response systems solve problem of stolen/lost password tokens (PTs).The Solution is PT requires a PIN ,even if stolen by attacker, PIN protects ”responses. It also Solves problem of window of vulnerability by having new challenge for each use.

Kerberos: Kerberos is a system that supports authentication in distributed systems. It Enable systems to withstand attacks in distributed systems. Basic idea of Kerberos: Central server provides tickets to requesting app and Ticket is authenticated, non-forgeable, non-replayable token. it is an encrypted data structure naming a user and a service that user is allowed to obtain. It also contains a time value and some control information. The first step in using Kerberos is to establish a session with the Kerberos server.

A user’s workstation sends the user’s identity to the Kerberos server when a user logs in. The Kerberos server verifies that the user is authorized. The Kerberos server sends two messages: i) to the user’s workstation, a session key SG for use in communication with the ticket-granting server (G) and a ticket TG for the ticket-granting server; SG is encrypted under the user’s password: E(SG + TG, pw). ii) to the ticket granting server, a copy of the session key SG and the identity of the user (encrypted under a key shared between the Kerberos server and the ticket-granting server). If the workstation can decrypt E(SG + TG, pw) by using pw, the password typed by the user, then the user has succeeded in an authentication with the workstation

T he user will want to exercise accessing a file. Using the key SG provided by the Kerberos server, the user U requests a ticket to access file F from the ticket-granting server. After the ticket granting server verifies U’s access permission, it returns a ticket and a session key. The ticket contains U’s authenticated identity (in the ticket U obtained from the Kerberos server), an identification of F (the file to be accessed), the access rights (for example, to read), a session key SF for the file server to use while communicating this file to U, and an expiration date for the ticket. The ticket is encrypted under a key shared exclusively between the ticket-granting server and

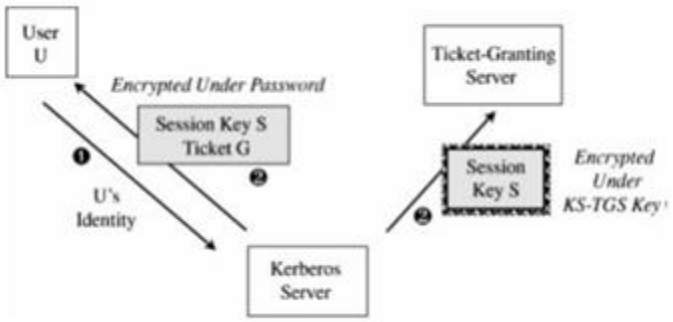

Fig. 14.Initiating a Kerbeross Session

the file server. This ticket cannot be read, modified, or forged by the user U (or anyone else). The ticket-granting server must, therefore, also provide U with a copy of SF, the session key for the file server.

Who needs access = authentication.

What and how will be accessed = access controls.

Access controls include: ACLs (Access Control Lists) on router and Firewalls . Routers perform the major task of directing network traffic either to sub networks they control or to other routers for subsequent delivery to other sub networks. Routers

convert external IP addresses into internal MAC addresses of hosts on a local subnetwork. Suppose a host is being flooded with packets from a malicious rogue host. Routers can be configured with access control lists to deny access to particular hosts from particular hosts. So, a router could delete all packets with a source address of the rogue host and a destination address of the target host.

Honeypots

Honeypot system built as a bait attracting attackers Reasons: a)They are observed to learn how they behave/operate
b)They are traced to catch them or scare them off
c)They are diverted from really valuable attack targets

A honeypot has no special features. Its a computer system or a network segment, loaded with servers and devices and data. It may beprotected with a firewall, although you want the attackers to have some access. There may be some monitoring capability, done carefully so that the monitoring is not evident to the attacker.

Traffic Flow Security

Threat: attacker infering occurrence/location of some event / structure from intensity of encrypted network traffic.
Solution 1: Masking by steady traffic volume
a)X and Y always send the same volume of encrypted traffic
between the
b)If X has nothing to communicate to Y, X sends meaning-

Fig.15: Obtaining a Ticket to Accessa File

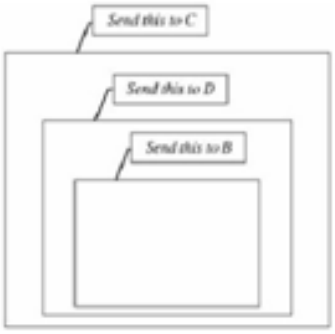

Fig. 16. Masking by onion routing

K. Firewall

a)Small / simple enough for rigorous analysis : Firewall designers strongly recommend keeping the functionality of the firewall simple.

b)always invoked: By carefully positioning a firewall within a network, we can ensure that all network accesses that we want to control must pass through it.

c)tamperproof: A firewall is typically wellisolated, making it highly immune to modification. Usually a firewall is implemented on a separate computer, with direct connections only to the outside and inside networks.

Types of Firewalls:

1) **Packet filtering gateways or screening routers:** A packet filter in gateway controls access to packets on the basis of packet address (source or destination) or specific transport protocol type (such as HTTP web traffic). For Example: Figure shows a packet filter that allows HTTP traffic but blocks traffic using the Telnet protocol.

2) **Stateful Inspection Firewall:** A stateful inspection firewall maintains state information from one packet to another in the input stream. These attributes are collectively known as the state of the connection, and may include such details as the IP addresses and ports involved in the connection and the sequence numbers of the packets traversing the connection.

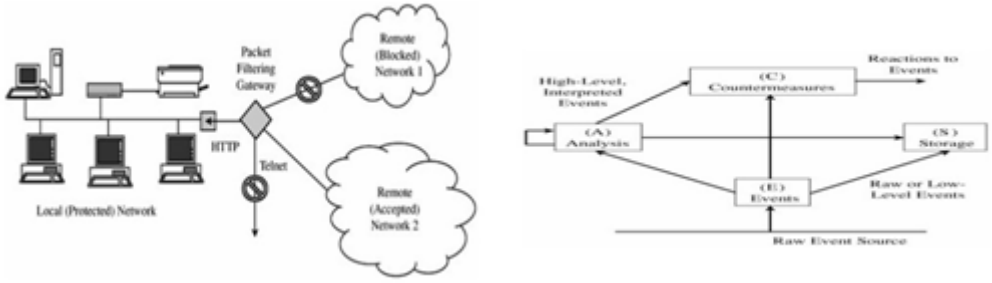

Fig: 17 Packet filtering gateway sorscreening routers

3) **Application Proxy:** Application proxy firewalls fix basic problem with packet filtering firewalls because they: See all pkt data (not just IP addresses and port s).

For Example, a company wants to set up an online price list so that outsiders can see the products and prices offered.

It wants to be sure that

(a)no outsider can change the prices or product list and

(b) outsiders can access only the price list, not any of the more sensitive files stored inside.

The proxy would monitor the file transfer protocol data to ensure that only the price list file was accessed, and that file could only be read, not modified.

4) **Guard:** A guard is a sophisticated firewall. Like a proxy 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 the user's behalf in accordance with its available knowledge, such as whatever it can reliably know of the (outside) user's identity, previous interactions, and so forth. The degree of control a guard can provide is limited only by what is computable.

5) **Personal Firewalls:** Personal firewall screens traffic on a single workstation. The personal firewall is configured to enforce some policy. For example, the user may decide that certain sites, such as computers on the company network, are highly trustworthy, but most other sites are not. The user defines a policy permitting download of code, unrestricted data sharing, and management access from the corporate segment, but not from other sites. Combine it with anti-virus software for more effective protection .

L. Intrusion Detection Systems

Functions:

i)monitoring users and system activity auditing system con-
figuration for vulnerabilities and misconfigurations

ii) assessing the integrity of critical system and data files iii) recognizing known attack patterns in system activity iv) identifying abnormal activity through statistical analysis v) installing and operating traps to record information about intruders

Types of IDSs:

Signature-based: A simple signature for a known attack type might describe a series of TCP SYN packets sent to many different ports in succession and at times close to one another, as would be the case for a port scan. An intrusion detection system would probably find nothing unusual in the first SYN, say, to port 80, and then another (from the same source address) to port 25. But as more and more ports receive SYN packets, especially ports that are not open, this pattern reflects

a possible port scan. The problem with signature-based detection is the signatures themselves For example, the attacker may convert lowercase to uppercase letters or convert a symbol such as ”blank space” to its character code equivalent

Heuristic Intrusion Detection: Heuristic intrusion detection looks for behavior that is out of the ordinary. For example, one user might always start the day by reading e-mail, write many documents using a word processor, and occasionally back up files. These actions would be normal. This user does not seem to use many administrator utilities. If that person tried to access sensitive system management utilities, this new behavior might be a clue that someone else was acting under the user’s identity.

Stealth Mode: Whereby an IDS has two network interfaces: one for the network (or network segment) being monitored and the other to generate alerts and perhaps other administrative needs. The IDS uses the monitored interface as input only; it never sends packets out through that interface. Often, the interface is configured so that the device has no published address through the monitored interface; that is, a router cannot route anything to that address directly, because the router does not know such a device exists. It is the perfect passive wiretap. If the IDS needs to generate an alert, it uses only the alarm interface on a completely separate control network.

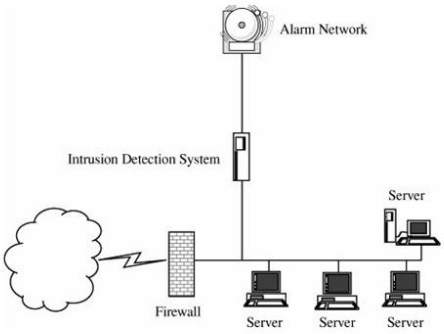

Fig. 18.Stealth Mode

4) Goals for Intrusion Detection Systems: Goals are: a.Filter on packet headers
b.Filter on packet content
c.Maintain connection state
d.Hide its presence
e.Responding to Alarms
f.False Results 5) IDS Strengths: Strengths are: a.Becoming more effective
b.Becoming cheaper
c.Becoming easier to administer 6) IDS limitations: Limitations are:
a.Attackers use avoidance strategies to avoid detection by

IDS
b.IDS sensitivity is difficult to measure and adjust
c.Must strike a balance false alarms and missing attacks
d.Someone has to monitor its track record

M. Secure E-Mail

E-mail is vital for today’s commerce, as well a convenient medium for communications among ordinary users. But, as we noted earlier, email is very public, exposed at every point from the sender’s workstation to the recipient’s screen. 1) Threats to E-mail: Threats are:
-message interception (confidentiality)
-message interception (blocked delivery)
-message interception and subsequent replay
-message content modification
-message origin modification

-message content forgery by outsider -message origin forgery by outsider 2) Secure e-mail requirements: They are:
-Msg confidentiality (protection from disclosure)
-Msg integrity (protection from modification)
-Sender authentication
-Non-repudiation (preventing denial by sender)

PGP (Pretty Good Privacy): PGP addresses the key distribution problem with what is called a ”ring of trust” or a user’s ”keyring.” One user directly gives a public key to another, or the second user fetches the first’s public key from a server. Some people include their PGP public keys at the bottom of email messages. And one person can give a second person’s key to a third (and a fourth, and so on). Thus, the key association problem becomes one of caveat emptor: ”Let the buyer beware.” If I am reasonably confident that an e-mail message really comes from you and has not been tampered with, I will use your attached public key.

If I trust you, I may also trust the keys you give me for other people. The model breaks down intellectually when you give me all the keys you received from people, who in turn gave you all the keys they got from still other people, who gave them all their keys, and so forth. You sign each key you give me. The keys you give me may also have been signed by other people. I decide to trust the veracity of a keyand- identity combination, based on who signed the key. PGP does not mandate a policy for establishing trust. Rather, each user is free to decide how much to trust each key received. The PGP processing performs some or all of the following actions, depending on whether confidentiality, integrity, authenticity, or some combination of these is selected:

- Create a random session key for a symmetric algorithm.
- Encrypt the message, using the session key (for message confidentiality).
- Encrypt the session key under the recipient’s public key.
- Generate a message digest or hash of the message; sign the hash by encrypting it with the sender’s private key (for message integrity and authenticity).
- Attach the encrypted session key to the encrypted message and digest.
- Transmit the message to the recipient.

The recipient reverses these steps to retrieve and validate the message content.

S/MIME: An Internet standard governs how e-mail is sent and received. The general MIME specification defines the format and handling of e-mail attachments. S/MIME (Secure Multipurpose Internet Mail Extensions) is the Internet standard for secure e-mail attachments. S/MIME is very much like PGP and its predecessors, PEM (Privacy-Enhanced Mail) and RIPEM.

The principal difference between S/MIME and PGP is the method of key exchange. Basic PGP depends on each user’s exchanging keys with all potential recipients and establishing a ring of trusted recipients; it also requires establishing a degree of trust in the authenticity of the keys for those recipients. S/MIME uses hierarchically validated certificates, usually represented in X.509 format, for key exchange. Thus, with S/MIME, the sender and recipient do not need to have exchanged keys in advance as long as they have a common certifier they both trust.

S/MIME works with a variety of cryptographic algorithms, such as DES, AES, and RC2 for symmetric encryption. S/MIME performs security transformations very similar to those for PGP. PGP was originally designed for plaintext messages, but

S/MIME handles (secures) all sorts of attachments, such as data files (for example, spreadsheets, graphics, presentations, movies, and sound). Because it is integrated into many commercial e-mail packages, S/MIME is likely to dominate the secure e-mail market.

ACKNOWLEDGMENT

We are grateful to the Nirma University for providing access to IEEE and ACM digital library which helped us to access different research papers. Special thanks to Prof.Parita Oza who helped us throughout the entire process of writing the report paper.

REFERENCES

[1]. Security in Computing, Fourth Edition By Charles P. Pfleeger
[2]. Bellovin, Steven M., and William R. Cheswick. ”Network firewalls.” Communications Magazine, IEEE 32.9 (1994): 50-57.
[3]. Biswas, Kamanshis, and Md Liaqat Ali. ”Security threats in Mobile AdHoc Network.” Department of Interaction and System Design School of Engineering, march2007 (2007): 9-26.
[4]. Staniford-Chen, Stuart, et al. ”GrIDS-a graph based intrusion detection system for large networks.” Proceedings of the 19th national information systems security conference. Vol. 1. 1996.