Copperbelt University
Computer Science Department

Computer Security

Compiled by: Prof. Dr. Hastings M. Libati

Introduction

This course provides a broad overview of the threats to the security of information systems, the responsibilities and basic tools for information security and the levels of training and expertise needed in organisations to reach and maintain a state of acceptable security. It covers concepts and applications of system and data security.

This course focuses on the importance of information as a resource, on which the knowledge base of successful organisations is dependent. While the main focus of the course is information management within the organisation, a broader context is important. National and international issues relating to information access will be addressed.

Information Management

Information, as we know it includes both electronic and physical information. In the case of organizations, the organizational structure must be capable of managing this information throughout the information lifecycle regardless of the source or format (data, paper documents, electronic documents, audio, video, etc.) for delivery through multiple channels that may include, among others, cell phones and web interfaces.

Given this definition of information, we can say that the focus of information management is the ability of organizations to capture, manage, preserve, store and deliver the right information to the right people at the right time. Management means the organization of and control over the structure, processing and delivery of information.

For any organization, information management requires the adoption and adherence to guiding principles that may, among others, include:

(a) Information must be made available and shared. It should be clear that though in principle the sharing of information helps the organization in the use and exploitation of corporate knowledge, not all information is open to anyone. Access rights have to be granted.

(b) Information that an organisation needs to keep is managed and retained corporately. This refers to the retention and archiving of information. If, for example, one saves a document today, one should expect that document to be secured and still be available the following day.

Information management therefore, is a corporate responsibility that needs to be addressed and followed from the upper most senior levels of management to the front line worker. Organizations must be held and must hold its employees accountable to capture, manage, store, share, preserve and deliver information appropriately and responsibly. Part of that responsibility lies in training the users (employees) to become familiar with the policies, processes, technologies and best practices in information

management. One important aspect of information management is to ensure that our information is secure.

The meaning of secure

One of the most valuable resources of any organisation today is information. Most of our information today is kept on computers. However, computers can be broken into. If this happens then the information that such computers contain can be stolen and lost. The question is: How do we secure our data and information – or rather how do we secure our computer systems?

In our lessons we shall look at various examples of how computer security affects our lives – directly and indirectly. We shall also endeavour to examine techniques to prevent security breaches or at least to mitigate their effects. We shall also address the concerns of information and communications technology practitioners as well as managers and users whose products, services and well being depend on the proper functioning of computer systems. Consequently, in our lessons, we shall endeavour to do the following:

(a) Examine the *risks* of security in computing
(b) Consider available *countermeasures* or *controls*
(c) Stimulate thought about *uncovered vulnerabilities*
(d) Identify areas where *more work* is needed

Characteristics of Computer Intrusion

Any part of a computing system can be the target of a crime. When we refer to a computing system, we mean a collection of hardware, software, storage media, data and people that an organisation uses to perform computing tasks. Sometimes we assume that parts of a computing system are not valuable to an outsider, but often we are mistaken. For instance, we tend to think that the most valuable property in a bank is the cash, gold or silver that the bank keeps. But in fact the customer information in the bank's computer may be far more valuable to someone. Stored on paper, recorded on computer storage media, resident in the main memory, or transmitted over communications lines or satellite links, this information can be used in myriad ways to make money illicitly. A competing bank can use this information to steal clients or even to disrupt services and discredit the bank. An unscrupulous individual could move money from one account to another without the owner's permission. A group of con artists could contact large depositors and convince them to invest in fraudulent schemes. The variety of targets and attacks makes computer security very difficult.

Any system is most vulnerable at its **weakest point**. A robber intent on stealing something from your house will not attempt to penetrate a 5cm-thick metal door if a window gives easier access. Similarly, a sophisticated perimeter physical security system does not compensate for unguarded access by means of a simple telephone line and a modem. We can modify this idea as one of the principles of computer security as given below:

Principle of Easiest penetration: *An intruder must be expected to use any available means of penetration. The penetration may not necessarily be by the most obvious means, nor is it necessarily the one against which the most solid defence has been installed. And it certainly does not have to be the way we want the attacker to behave.*

This principle implies that computer security specialists must consider all possible means of penetration. Moreover, the penetration analysis must be done repeatedly and especially whenever the system and its security change. People sometimes underestimate the determination or creativity of attackers. Remember that computer security is a game with rules only for the defending team: The attackers can (and will) use

any means they can in order to achieve their goal. Perhaps the hardest issue for people outside the security community to do is to think like the attacker. Strengthening one aspect of a system may simply make another means of penetration into the system more appealing to intruders.

The meaning of computer Security

Any computer-related system has one or two weaknesses if not more. The purpose of computer security is to devise ways to prevent such weaknesses from being exploited.

Security Goals

The term "security" is used in many ways in our daily lives. When we talk about computer security, it means that we are addressing three important aspects of any computer-related system. These aspects are **confidentiality**, **integrity** and **availability**.

(a) *Confidentiality* ensures that computer-related assets are accessed only by authorised parties. That is, only those who should have access to some items should be granted that access. It is important to understand that by the term "access", we mean not only reading but also viewing, printing, or simply knowing that a particular asset exists. Confidentiality is sometimes called secrecy or privacy.

However, ensuring confidentiality can be difficult. For example, who determines which people or systems are authorised to access the current system? By accessing data, do we mean that an authorised party can access a single bit? the whole collection? pieces of data out of context? Can someone who is authorised disclose those data to other parties?

(b) *Integrity* means that assets can be modified only by authorised parties or only in authorised ways. In this context, modification includes writing, changing, deleting and creating.

(c) *Availability* means that assets are accessible to authorised parties at appropriate times. In other words, if some person or system has legitimate access to a particular set of objects, that access should not be prevented. For this reason, availability is sometimes known by its opposite, **denial of service**.

Security in computing addresses these three goals. One of the challenges in building a secure system is finding the right balance among the three goals, which often conflict. For example, it is easy to preserve the confidentiality of a particular object in a secure system simply by preventing everyone from accessing that object. However, this system is not secure, because it does not meet the requirement of availability for proper access. That is, there must be a balance between confidentiality and availability.

However, balance is not all. In fact, these three characteristics can be independent, can overlap (as shown in figure 1), and can even be mutually exclusive. For example, we have seen that strong protection of confidentiality can severely restrict availability. Below is an in depth examination of these three security goals.

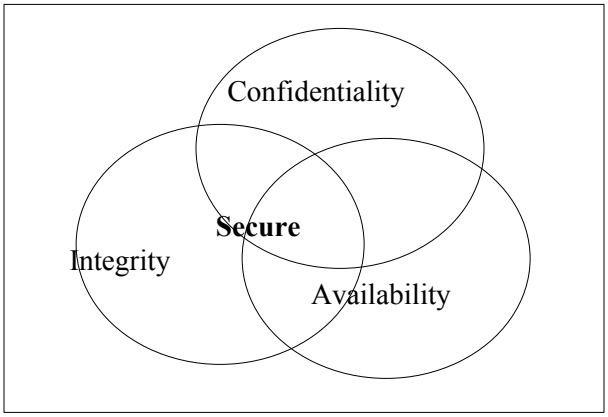

Figure 1: Relationship among confidentiality, Integrity and Availability

Who are the Threats? (Computer Criminals)

There are many threats to computer and network resources. These include intruders such as computer criminals, unintentional actions by innocent people as well as natural disasters, among others.

It is hard to describe computer criminals in such a way that they can be easily identified when one sees them. Some computer criminals wear business suits, have university degrees, and appear to be pillars of their communities. Some are college or university students. Others are middle-aged business executives. Some are mentally deranged, overly hostile or extremely committed to a cause and they attack computers as symbol. Others are ordinary people tempted by personal profit, revenge, challenge, advancement, or job security. No single profile captures the characteristics of a typical computer criminal and many who fit the profile are not criminals at all.

Whatever their characteristics and motivations, computer criminals have access to enormous amounts of hardware, software and data. Computer criminals have the potential to cripple much of effective business and government throughout the world. In a sense, therefore, we can say that the purpose of security is to prevent these criminals from doing damage.

For the purpose of studying computer security, we say that **computer crime** is any crime involving a computer or aided by the use of one. Although this definition is admittedly broad, it allows us to consider ways to protect ourselves, our businesses and our communities against those who use computers maliciously. The estimates of how much is lost through computer crime in monetary terms are quite high. This tells us that it is important for us to pay attention to computer crime and to try to prevent it or at least to moderate its effects.

One approach to prevention or moderation is to understand who commits these crimes and why. Many studies have attempted to determine the characteristics of computer criminals. By studying those who have already used computers to commit crimes, we may be able in the future to spot likely criminals and prevent the crimes from occurring. **Some of these characteristics are discussed below.**

Curious crackers poke around just to see what they can get into and just to satisfy their selfish interests. Vandals on the other hand, inflict various sorts of damage to networks and network resources. They cause system down-times, network outages and steal network bandwidth usage. These incidents can cause tremendous losses to the targeted organisations. Industrial spies target organisations so as to steal trade secrets and such information, which might put the target organisation at a disadvantage regarding its competitors. In some cases, they aim at harming the reputation of the targeted organisation. However, in some cases sensitive information may accidentally be disclosed to the people in whose hands the information is not supposed to land.

What are the Threats? - Attacks

As we have already seen earlier in our text, there are many threats to computer and network resources. Among them are intentional and unintentional actions by people as well as natural disasters.

When you test any computer system, one of your jobs is to imagine how the system could malfunction. Then you improve the system's design so that the system can withstand any of the problems you have identified. In the same way, we analyse a system from a security perspective by thinking about ways in which the system's security can malfunction and diminish the value of its assets.

Vulnerabilities, Threats, Attacks and Controls

A computer-based system has three separate but valuable components. These are the **hardware**, **software** and **data**. Please, be mindful that in most cases when we mention software, we also include data and information in there. However, for the sake of driving our point home, we shall separate software and data in this case. To analyse security, we can brainstorm about the ways in which the system or its information can experience some kind of loss or harm. For example, we can identify data whose format or contents should be protected in some way. We want our security system to ensure that no data are disclosed to unauthorised parties. Neither do we want the data to be modified in illegitimate ways. At the same time, we must ensure that legitimate users have access to the data. In this way, we can identify weaknesses in the system.

A **vulnerability** is a weakness in the security system, for example, in procedures, design or implementation that might be exploited to cause loss or harm. For instance, a particular system may be vulnerable to unauthorised data manipulation because the system does not verify a user's identity before allowing data access.

A **threat** to a computing system is a set of circumstances that has the potential to cause loss or harm. To understand the difference between a threat and vulnerability, let us consider figure 2. In this figure, a wall is holding water back. The water to the left of the wall is a threat to the man on the right of the wall. This is so because the water level could rise and start overflowing onto the man standing on the right side of the wall. The water level could also stay beneath the height of the wall but cause the wall to collapse. So the threat of harm is the potential for the man to get wet, get hurt or be drowned. For now the wall is intact, so the threat to the man does not materialise, it just remains as a threat.

However, we can see a small crack in the wall. This crack is a vulnerability, which threatens the man's security. If the water rises to or beyond the level of the crack, it will exploit the vulnerability and harm the man.

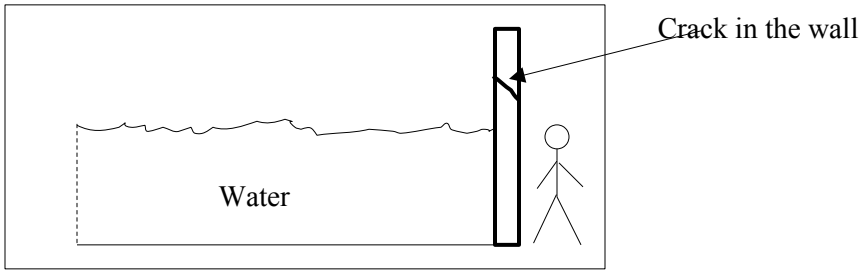

Figure 2: An illustration of a threat and vulnerability

There are many threats to a computer system, including human-initiated and computer-initiated ones. We have all experienced the results of inadvertent human errors, hardware design flaws and software failures. But natural disasters are threats, too. This is because natural disasters can bring a system down when the computer room is flooded or the data centre collapses from an earthquake, for example.

A human who exploits a system vulnerability perpetrates an **attack** on the system. An attack can also be launched by another system, as when one system sends an overwhelming set of messages to another, **virtually shutting down the second system's ability to function**. One such type of attack is the denial-of-service attack where a system floods a server with more messages than the server can handle.

The question is how do we address these problems? The answer is we use a **control** as a protective measure. That is, a control is an action, device, procedure or technique that removes or reduces a vulnerability. In general, we can describe the relationship among threats, controls and vulnerabilities in this way:

A threat is blocked by control of a vulnerability.

Most of our lessons will be devoted to describing a variety of controls and understanding the degree to which they enhance a system's security. To devise controls, we must know as much about threats as possible. We can view any threat as being one of four kinds: interception, interruption, modification and fabrication. Each threat exploits vulnerabilities of the assets in computing systems.

(a) An **interception** means that some unauthorised party has gained access to an asset. The outside party can be a person, a program, or a computing system. Examples of this type of attack are illicit copying of programs or data files or wiretapping to obtain data in a network. Although a loss may be discovered fairly quickly, a silent interceptor may leave no traces by which the interception can readily be detected.

(b) In an **interruption**, an asset of the system becomes lost, unavailable or unusable. An example is malicious destruction of a hardware device, erasure of a program or data file or malfunction of an operating system file manager so that it cannot find a particular disk file.

(c) If an unauthorised party not only accesses but tempers with an asset, the threat is called **modification**. For example, someone might change the values in a database, alter a program so that it performs an additional computation or temper with data that is being transmitted electronically. It is even possible to modify hardware. Some cases of modification can be detected with simple measures, but other, more subtle changes may be almost impossible to detect.

(d) Finally, an unauthorised party may create a **fabrication** of counterfeit objects on a computing system. The intruder may insert spurious transactions into a network communication system or add records to an existing database. Sometimes, these additions can be detected as forgeries, but if skilfully done, they are virtually indistinguishable from the real items.

The four classes of threats outlined above, namely, interception, interruption, modification and fabrication describe the kinds of problems that we may encounter when dealing with security in computing.

Method, Opportunity and Motive

A malicious attacker must have three items, namely:

(a) *Method*: the skills, knowledge, tools and other items with which to be able to pull off the attack

(b) *Opportunity*: the time and access to accomplish the attack

(c) *Motive*: a reason to want to perform this attack against the system

Knowledge of systems is widely available. Mass-market systems (such as the Microsoft, Apple or UNIX operating systems) are readily available, as are common products, such as word processors or database management systems. Sometimes the manufacturers release detailed specifications on how the system was designed or how the system operates. Such specifications or manuals serve as guides for users and integrators who want to implement other complementary products. But even without documentation, attackers can purchase and experiment with many systems. Often, only time and inclination limit an attacker.

Many systems are readily available. Systems available to the public are, by definition, accessible; often their owners take special care to make them fully available so that if one hardware component fails, the owner has spares instantly ready to be pressed into service.

However, it is difficult to determine motive for an attack. Some places are what are called *attractive targets*, meaning that they are very attractive to attackers. These may include such well-protected sites like government sites. Other systems are attacked because they are easy to attack, such as universities. On the other hand, other systems are attacked simply because they are there. Protecting against attacks can be difficult. Anyone can be a victim of an attack perpetuated by an attacker.

To sum up, we can say that the threats to network and computing systems manifest themselves as attacks targeted at resources of a system. These attacks often exploit flaws in either the operating system, network protocols or application programs. The general goal of such attacks is to subvert the traditional security mechanisms on the systems and execute operations in excess of the attacker's authorisation. These operations could include reading protected or private data or simply doing malicious damage to the systems or user files. Threats to network and computing resources could also be due to disasters like floods, fires or even electrical surges and spikes. Confidential information can reside in two states on a network.

- It can reside on physical storage media, such as a hard drive or main memory or
- It can reside in transit across the physical network wire in the form of network packets.

These two information states present multiple opportunities for attacks from internal as well as external attackers of a network. Some general attacks are discussed below:

Man-in-the Middle Attacks

A *man-in-the-middle* attack requires that the attacker have access to network packets that come across the networks. Network packet filters are the tools that attackers use for this type of attacks. Most network protocols distribute network packets in *clear text* and so the network packets can be processed and understood by any application that can pick them off the network and process them. A packet filter is a software application that uses a network adapter card in promiscuous mode to capture all network packets that are sent across a network. A packet filter can provide its user with sensitive information, such as user account names, passwords and the actual data carried in the captured packets. Packet filters can also provide information about the topology of a network, such as what computers run which services, how many computers are on a network and which computers have access to others

If an attacker gains access to a re-usable password for a system-level user account, he can use it to create a new account that can be used at anytime as a *back-door* to get into the network and its resources. Once in the target system, the attacker can modify critical system resources such as the password for the system administrator account, the list of services and permissions on file servers, and the login information for other computers that contain confidential information. In addition, a network packet filter can be modified to interject new information or change existing information in a network packet. By doing so, the attacker can cause network connections to shut down prematurely, as well as change critical information within the network packets. This can cause immense damage to the affected systems and consequently the affected organisations. When an attacker successfully gains access to a resource, he has the same rights as the user whose account has been compromised to gain access to that resource.

IP Address Spoofing

An IP Address Spoofing attack occurs when an attacker outside a network pretends to be a trusted computer either by using an IP address that is within the range of IP addresses for the attacked network or using an authorised external IP address that the attacked network trusts.

An attacker who manages to change the routing tables of the attacked network to point to the spoofed IP address, can receive all the network packets that are addressed to the spoofed address. In such a case, an attacker can monitor all the network's traffic, effectively becoming a *man in the middle*. IP spoofing can be used by both internal and external attackers of a network. In this way, IP spoofing can also yield access to user accounts and passwords. An attacker can also emulate an internal user in ways that can prove embarrassing for an organisation. For example, the attacker could send irresponsible e-mail messages to an organisation's business partners that appear to have originated from someone within the partner organisation.

Denial-of-Service Attacks (DOS)

Denial-of-service attacks focus on making a network service unavailable to legitimate users for normal use, and this is typically accomplished by exhausting some resource limitation on the network or within an operating system or application.

When involving specific network server applications, such as HTTP or FTP, these attacks can focus on acquiring and keeping open all the available connections supported by that server, effectively locking out valid users of the server or service. Most denial-of-service attacks exploit a weakness in the overall architecture of the system being attacked. Some attacks compromise the performance of a network by flooding the network with undesired network packets (e.g., ping floods) and by providing false information about the status of network resources.

Application Layer Attacks

Application layer attacks usually exploit well-known weaknesses in software commonly found on network servers, such as sendmail (e.g., the Internet Worm of 1988). By exploiting these weaknesses, attackers can gain access to a computer system with the permissions of the account running the application, which is usually a privileged system-level account.

Trojan horses are commonly used as application layer attacks. Trojan horses are programs that may provide all the functionality that a normal program offers, but they also include other features that are known only to the attacker, such as monitoring login attempts to capture user account and password information. These programs can capture sensitive information and distribute it back to the attacker. They can also modify application functionality, such as applying a blind carbon copy to all e-mail messages so that the attacker can read all the e-mails of an organisation.

Some application layer attacks exploit the openness of such technologies as the Hypertext Markup Language (HTML) specification, Web browser functionality, and the HTTP protocol. These attacks include Java applets, whereby harmful programs are passed across the network and are then loaded through a user's browser. These are then used as trojan horses to cause malicious damage, e.g. overwriting files or executing other malicious programs.

What are the Defences?

Computer crime is certain to continue for a foreseeable future. For this reason, we must look carefully at controls for preserving confidentiality, integrity and availability. Sometimes these controls can prevent or mitigate attacks. Other less powerful methods of defence can only inform us that security has been compromised, by detecting a breach as it happens or after it has occurred.

Harm occurs when a threat is realised against a vulnerability. To protect against harm, then, we can neutralise the threat, close the vulnerability, or both. The possibility for harm to occur is called a **risk**. We can deal with harm in several ways. We can seek to

(a) *prevent it*, by blocking the attack or closing the vulnerability

(b) *deter it*, by making the attack harder but not impossible

(c) *deflect it*, by making another target more attractive (or this one less so)

(d) *detect it*, either as it happens or sometime after the fact (successful attack)

(e) *recover* from a successful attack and its effects

Of course more than one of these actions can be taken at once. For example, we might try to prevent intrusions. But in case we do not prevent them all, we might install a detection device to warn of an imminent attack. And we should have in place incident response procedures to help in the recovery in case an intrusion does succeed.

Controls

To consider the controls or countermeasures that try to prevent exploiting a computing system's vulnerabilities, we begin by thinking about traditional ways to enhance physical security. In the Middle Ages, castles and fortresses were built to protect the people and valuable property inside. The fortress might have had one or more security characteristics, including

(a) a strong gate or door, to repel invaders
(b) heavy walls to withstand objects thrown or projected against them
(c) a surrounding moat, to control access
(d) arrow slits, to let archers shoot at approaching enemies
(e) areas to allow inhabitants to lean out from the roof and pour hot or vile liquids on the attackers
(f) a drawbridge to limit access to authorised people
(g) gatekeepers to verify that only authorised people and goods could enter

Similarly, today we can use multipronged approach to protect our homes and offices. We may combine strong locks on the doors with a burglar alarm and reinforced windows. In each case, we select one or more ways to deter an intruder or attacker and we base our selection not only on the value of what we protect but also on the effort that we think an attacker or intruder will expend to get inside.

Computer security has the same characteristics. We have many controls at our disposal. Some are easier than the others to use or implement. Some are cheaper than others to use or implement. And some are more difficult than others for intruders to override. When it comes to security controls, we can use one or more controls, according to what we are protecting, how the cost of protection compares with the risk of loss and how hard we think intruders will work to get what they want. Below is an overview of the controls available to us.

Software Controls

Programs are also very important in providing computer security. Programs must be secure enough to prevent outside attack. They must also be developed and maintained so that we can be confident of the program's dependability. Program controls include the following:

(a) *internal program controls*: parts of the program must enforce security restrictions, such as access limitations in database management system

(b) *operating system and network system controls*: limitations imposed by the operating system or network to protect each user from all other users

(c) *independent control programs*: application programs such as password checkers, intrusion detection utilities or virus scanners that protect against certain types of vulnerabilities

(d) *development controls*: quality standards under which a program is designed, coded, tested and maintained to prevent software faults from becoming exploitable vulnerabilities

We can implement software controls by using tools and techniques such as hardware components, encryption or information gathering. Software controls usually affect users directly, such as when the user is interrupted and asked for a password before being given access to a program or data. Because they influence the way that users interact with a computing system, software controls must be carefully designed.

Several methods and technologies have been devised to protect network resources from malicious damage. Most of these defensive mechanisms and technologies are based on:

- protection from attacks
- detection of successful attacks (and also attacks in progress) and
- recovery from successful attacks.

Policies and Procedures

Sometimes, we can rely on agreed-on procedures or policies among users rather than enforcing security through hardware or software means. In fact, some of the simplest controls, such as frequent changes of passwords, can be achieved at essentially no cost but with tremendous effect. Training and administration follow immediately after establishment of policies, to reinforce the importance of security and to ensure their proper use.

The network security policy, for example, is a collection of security rules, conventions, and procedures governing communications into and out of a protected network, allowing only approved services and packets to pass through between internal and external networks. The policy also spells out the access rights of users to information and information resources. Furthermore, a network security policy identifies the vital resources of a network, which require protection. A network security policy is the first step that should be taken as one embarks on securing a network.

Physical Security (Physical Controls)

Some of the easiest, most effective, but least expensive controls are physical controls. Physical controls include locks on doors, guards at entry points, backup copies of important programs and data, and physical site planning that reduces the risk of natural disasters. Often the simple physical controls are overlooked while we seek more sophisticated approaches.

Physical security is paramount in any intention to secure a network and its resources. A network whose computing equipment is physically accessible to anybody can never be well secured regardless of the effort and capital spent on security. Besides, computer equipment is sensitive to many types of environmental conditions such as dust, fire, smoke and humidity and should consequently be protected against them. Dust can shorten the life span of magnetic media, tape and optical drives. Surge suppression devices should be used to prevent computer and communication equipment from surges. Physical access to computing resources should be denied to unauthorised people. By controlling physical access to computers and communication equipment, it becomes difficult for vandals to steal or damage either data or equipment. Access policies for computing facilities should be established and the affected users educated on security.

Backups

A reliable and current backup can be used as a first line of defence against a network disaster. Backups are vital for the following reasons:

- If a site's network is compromised or undergoes any serious damage, the administrator can use the latest backup to restore the data and make the network operational again.
- If a network is compromised and one does not know the extent of the damage, backups can help the administrator to determine what changes were made to the system and so enable the administrator to restore the data with confidence.

- Backups are also useful for archiving seldom used data that must be saved for legal or historical purposes.

Backups should be updated from time to time. There are several types of backups such as:

- **Incremental backups**: where only data that has changed since the last backup is backed up.
- **Differential backup**: where only data that has changed since the last full backup is backed up
- **Full backup**: where the entire specified data is backed up.

Which type of backup to carry out and the frequency at which it is carried out depends on the security policy of the site. To ensure the smooth restoration of data even in cases of catastrophes like floods or fires that befall an operational site, it is advisable to keep a standby backup copy in a water- and fire-proof safe off-site.

The three generation system could also be employed to give a big window for data recovery. In very sensitive organisations where system outages should not be tolerated, hardware backup systems could also be installed. These are operated on standby basis and become operational immediately the main system stops working due to corruption or break-down.

User Education

User-awareness is a very powerful tool in security. Trained and educated users are less likely to fall for scams and social engineering attacks. They are also more likely to participate in security measures without reservations if they understand why such security measures are in place. In security, education should be a continuous process since there are always new tools and new threats, new techniques, and new information to be learned.

Operating Systems Security

Operating systems such as 4.4-BSD UNIX and Linux, provide some security mechanisms that can be used to protect system and network resources from misuse. Such mechanisms include the use of file flags that restrict changes made to files, they also restrict who accesses which file or system resource. This gives control of access to files and system resources. Operating systems also keep system logs about the activities registered on the system. Used carefully, these logs can be a very handy tool in monitoring what goes on in the system. In the case of a successful attack, the logs can be used to ascertain the extent of the damage caused and even to find out who did the damage and at what time.

Integrity mechanisms

We have already discussed techniques that ensure the integrity of data against accidental damage. These are the parity bits, checksums and cyclic redundancy checks (CRCs), among others. A checksum or CRC cannot absolutely guarantee data integrity for two reasons. First, if malfunctioning hardware changes the value of a checksum as well as the value of the data, it is possible for the altered checksum to be valid for the altered data. Second, if data changes result from a planned attack, the attacker can create a valid checksum for the altered data. Several mechanisms have been used to guarantee the integrity of messages against intentional change. In general, the methods encode transmitted data with a *message authentication code (MAC)* that an attacker cannot break or forge. Typical encoding schemes use *cryptographic hashing* mechanisms. For example, one cryptographic hashing scheme uses a secret key known only to the sender and receiver. When a sender encodes a message, the cryptographic hash function uses the *secret key* to scramble the position of bytes within the message as well as to encode the data. Only the receiver can

unscramble the data; an attacker who does not have the secret key, cannot decode the message without introducing an error. Thus, the receiver knows that any message that can be decoded correctly is authentic.

Access control and passwords

Many computer systems use a password mechanism to control access to resources. Each user has a password, which is kept secret. When a user needs to access a protected resource, the user is asked to enter the password. A simple password scheme works well for a conventional computer system because the system does not reveal the password to others. In a network, however, a simple password mechanism is susceptible to eavesdropping. If a user at one location sends a password across a network to a computer at another location, anyone who wiretaps the network can obtain a copy of all traffic. In such situations, additional steps must be taken to prevent passwords from being reused - ONE TIME PASSWORDS.

Authentication Servers

Like firewalls, authentication servers are a network's first line of defence against attackers as they limit access to network resources by selectively permitting or denying access based on specific characteristics such as phone numbers or passwords. This helps to keep attackers (intruders) off an organisation's network. Dial-back modems, for instance, call users back at predefined telephone numbers, thus ensuring that only authorised users are allowed access to networks. In such call-back (dial-back) systems, the user phones the main system and provides a valid username and password. The dial-back system then connects to a centrally held database to find out what telephone number is associated with the password. The call-back system then phones back the caller and allows the user's connection. In this way, even if an attacker manages to compromise a password, the system would not grant access because the call from the dial-back system would not reach him. By validating users before they are allowed to dial in (log on) to the network, authentication servers ensure that only legitimate users are allowed access.

Encryption and privacy

We have already seen that in our endeavours, we seek to protect hardware, software and data. We can make it particularly hard for an intruder to find data useful if we somehow scramble the data so that interception is meaningless without the intruder’s knowing how the scrambling was done. Indeed, the most powerful tool in providing computer security is this scrambling or encoding.

Encryption is the formal name for the scrambling process. We take data in their normal, unscrambled state, called **clear text** and transform them so that they are unintelligible to the outsider observer. The transformed data are called **enciphered text** or **cipher text**. Using encryption, security professionals can virtually nullify the value of an interception and the possibility of effective modification or fabrication.

Encryption clearly addresses the need for confidentiality of data. Additionally, it can be used to ensure integrity; data that cannot be read generally cannot easily be changed in a meaningful manner.

For example, to ensure that the contents of a message remain confidential despite wiretapping, the message must be *encrypted*. In essence, encryption scrambles bits of the message in such a way that only the intended recipient can unscramble them. Someone who intercepts a copy of the encrypted message will not be able to extract information. Several technologies exist for encryption. In some technologies, a sender and receiver must both have a copy of an *encryption key*, which is kept secret. The sender uses the key to produce an encrypted message, which is then sent across a network. The receiver uses the key to decode the encrypted message. That is, the *encrypt* function used by the sender takes two arguments: a

key, K, and a message to be encrypted, M. The function produces an encrypted version of the message, E.

E = encrypt (K, M)

The *decrypt* function reverses the mapping to produce the original message:

M = decrypt (K, E)

Mathematically, *decrypt* is the inverse of *encrypt*: *M = decrypt (K, encrypt (K, M))*
Although encryption is an important task in any computer security tool kit, we should not overrate its importance. Encryption does not solve all computer security problems, so other tools must complement its use. Furthermore, if encryption is not used properly, it may have no effect on security or could even degrade the performance of the entire system. Weak encryption can actually be worse than no encryption at all, because it gives users an unwarranted sense of protection. Therefore, we must understand those situations in which encryption is most useful as well as ways to use it effectively.

Public key encryption

In many encryption schemes, the key must be kept secret to avoid compromising security. One encryption technique assigns each user a pair of keys. One of the user's keys, called the *private key*, is kept secret, while the other, called the *public key*, is published along with the name of the user, so everyone knows the value of the key. The encryption function has the mathematical property that a message encrypted with the public key cannot be easily decrypted except with the private key, and a message encrypted with the private key can not be decrypted except with the public key. The relationships between encryption and decryption with the two keys can be expressed mathematically. Let *M* denote a message, *pub-u1* denote user1's public key and *prv-u1* denote user1's private key. Then

M = decrypt (pub-1, encrypt (prv-u1, M))

and

M = decrypt (prv-u1, encrypt (pub-u1, M))

Revealing a public key is safe because the functions used for encryption and decryption have a *one-way property*. That is, telling someone the public key does not allow the person to forge a message that appears to be encrypted with the private key. Public key encryption can be used to guarantee confidentiality. A sender who wishes a message to remain private uses the receiver's public key to encrypt the message. Obtaining a copy of the message as it passes across the network does not enable someone to read the contents because decryption requires the receiver's private key. Thus, the scheme ensures that data remains confidential because only the receiver can decrypt the message.

Authentication with digital signatures

An encryption mechanism can also be used to authenticate the sender of a message. The technique is known as a *digital signature*. To sign a message, the sender encrypts the message using a key known only to the sender. The recipient uses the inverse function to decrypt the message. The recipient knows who sent the message because only the sender has the key needed to perform the encryption. To ensure that encrypted messages are not copied and resent later, the original message can contain the time and date that the message was created.

Let us consider how a public key can be used to provide a digital signature. To sign a message, a user encrypts the message using his or her private key. To verify the signature, the recipient looks up the user's public key and uses it to decrypt the message. Because only the user knows the private key, only the user

can encrypt a message that can be decoded with the public key. Two levels of encryption can be used to guarantee that a message is both authentic and private. First, the message is signed by using the sender's private key to encrypt it. Second, the encrypted message is encrypted again using the recipient's public key. Mathematically, double encryption can be expressed as:

X = encrypt (pub-u2, encrypt (prv-u1, M))

Where *M* denotes a message to be sent, *X* denotes the string that results from the double encryption, *prv-u1* denotes the sender's private key, and *pub-u2* denotes the recipient's public key.

At the receiving end, the decryption process is the reverse of the encryption process. First, the recipient uses his or her private key to decrypt the message. The decryption removes one level of encryption, but leaves the message digitally signed. Second, the recipient uses the sender's public key to decrypt the message again. The process can be expressed as

M = decrypt (pub-u1, decrypt (prv-u2, X))

Where *X* denotes the encrypted string that was transferred across the network, *M* denotes the original message, *prv-u2* denotes the recipient's private key and *pub-u1* denotes the sender's public key. If a meaningful message results from the double decryption, it must be true that the message was confidential and authentic. The message must have reached its intended recipient because only the intended recipient has the correct private key needed to remove the outer encryption. The message must have been authentic, because only the sender has the private key needed to encrypt the message so the sender's public key will correctly decrypt it.

Effectiveness of Controls

Merely having controls does no good unless these controls are used properly. Let us consider several aspects that can enhance the effectiveness of controls.

Awareness of Problem

People using controls must be convinced of the need for security. That is, people will willingly cooperate with security requirements only if they understand why security is appropriate in a given situation. However, many users are unaware of the need for security, especially in situations in which a group has recently undertaken a computing task that was previously performed with lax or no apparent security.

Likelihood of Use

Of course, no control is effective unless it is used. The lock on a computer room door does no good if people leave the door open.

> **Principle of Effectiveness**: *Controls must be used – and used properly – to be effective. They must be efficient, easy to use and appropriate.*

This principle implies that computer security controls must be efficient enough, in terms of time, memory space, human activity or other resources used, that using the controls does not seriously affect the task being protected. Controls should be selective so that they do not exclude legitimate accesses.

Overlapping Controls

As with home security, several different controls may apply to address a single vulnerability. For example, we may choose to implement security for a microcomputer application by using a combination of controls on program access to the data, on physical access to the microcomputer and storage media, and even by file locking to control access to the processing programs.

Periodic Review

Few controls are permanently effective. Just when the security specialist finds a way to secure assets against certain kinds of attacks, the opposition doubles its efforts in an attempt to defeat the security mechanisms. Thus judging the effectiveness of a control is an on-going task.

Seldom, if ever, are controls perfectly effective. Controls fail, controls are incomplete, or people circumvent or misuse controls, for example. For this reason, we use **overlapping controls**, sometimes called a **layered defence** or **multitier defence**, in the expectation that one control will compensate for the failure of another. In some cases, controls do nicely complement each other. But two controls are not always better than one and, in some cases, two can even be worse than one. This brings us to another security principle.

> **Principle of Weakest Link**: *Security can be no stronger than its weakest link. Whether it is the power supply that powers the firewall or the operating system under which the security application operates or the human who plans, implements and administers controls, the failure of any control can lead to a security failure.*

Network Security

Let us now turn to the focus of our lessons, which is network security. In the wake of increased network connectivity, computer systems are becoming increasingly vulnerable to attacks. These attacks are aimed at compromising networks and the sensitive and confidential information that they contain. This scenario underscores the importance of network security. Network security sets out to protect information traversing networks and that, which is contained in network devices from attackers by preventing the theft, destruction, corruption, and introduction of unwanted pieces of information. Only authenticated users are granted access to the network and network resources. Integrity and confidentiality, as they apply to messages that transit networks, ensure that the information which arrives is identical to the one which was sent, and that only the intended recipients can access it and so affording legitimate users an uninterruptible availability of network resources.

Secure networks and policies

Although the concept of a secure network is appealing to most users, networks cannot be classified simply as secure or not secure because the term is not absolute. Each organisation defines the level of access that is permitted or denied. For example, some organisations store data that is valuable. Such organisations define a secure network to be a system that prevents outsiders from accessing the organisation's computers. Other organisations need to make information available to outsiders, but prohibit outsiders from changing the data. Such organisations may define a secure network as one that allows arbitrary access to data, but includes mechanisms that prevent unauthorised changes. Still other groups focus on keeping communication private. They define a secure network as one in which no one other than the intended recipient can intercept and read a message. Finally, many organisations need a

complex definition of security that allows access to selected data or services the organisation chooses to make public, while preventing access or modification to sensitive data and services that are kept private.

Because no absolute definition of *secure network* exists, the first step an organisation must take to achieve a secure system is to define the organisation's *security policy*. The policy does not specify how to achieve protection. Instead, it states clearly and unambiguously the items that are to be protected. Defining a network security policy is complex. The primary complexity arises because a network security policy cannot be separated from the security policy for computer systems attached to the network. In particular, defining a policy for data that traverses a network does not guarantee that data will be secure. For example consider data stored in a file that is readable. Network security cannot prevent unauthorised users who have accounts on the computer from obtaining a copy of the data. Thus, to be effective, a security policy must apply all the times. The policy must hold for the data stored on disk, data communicated over a telephone line with a dialup modem, information printed on paper, data transported on portable media such as a floppy disk and data communicated over a computer network.

Assessing the costs and benefits of various security policies also adds complexity. In particular, a security policy cannot be defined unless an organisation understands the value of its information. In many cases, the value of information is difficult to assess. Consider, for example, a simple payroll database that contains a record for each employee, the hours the employee worked and the rate of pay. The easiest aspect of value to assess is the replacement cost. That is, one can compute the man-hour required to recreate or verify the contents of the database (e.g., by restoring the data from an archive or by performing the work needed to collect the information). A second aspect of value arises from the liability an organisation can incur if the information is incorrect. For example, if an unauthorised person increases the pay rate in a payroll database, the company could incur arbitrary costs because employees would be overpaid. A third aspect of value arises from the indirect costs that can be incurred from security violations. If payroll information becomes public, competitors might choose to hire workers, which results in costs for hiring and training replacements as well as increased salaries needed to retain other employees.

Defining a security policy is also complicated because each organisation must decide which aspects of protection are most important, and often must compromise between security and ease of use. For example, an organisation can consider several facets of security:

Facets of Network Security

The primary facets of network security are:

- authentication,
- access control,
- integrity,
- confidentiality,
- non-repudiation
- auditing and
- availability.

An elaboration of each of these items follows below.

Authentication

Authentication is establishing proof of identity. Usually this involves one or a combination of

(a) something the user is,
(b) something the user knows and
(c) something the user has.

What one knows could be an account name and a password. What one has could be a hardware authentication device (e.g. smart cards or token cards which provide one-time passwords). Authentication differentiates one entity or user from another. Without authentication, all users would be treated equally, or each blindly trusted to his own unsupported claim of identity.

Access control

Access control relates to who (or what) may have access to some network resource. In other words, access control strives to regulate access to the network and its resources. Implementing access control based on identity requires some form of authentication. With an accurate authentication mechanism access controls can be applied to provide authorisation, access rights, and privileges.

Integrity

Integrity refers to protection from change: is the data that arrives exactly the same as the data that was sent? So, integrity refers to the current condition of some data as compared to their pure and original state. A message or file that traverses a network is at risk of having data added, removed, or modified along the way. A message that undergoes this experience loses its integrity. To ensure integrity, information should be protected from being deleted or altered in any way without the permission of the owner of that information.

Confidentiality and Privacy

Confidentiality and privacy refer to protection against snooping or wiretapping: So the question to ask here is - is data protected against unauthorised access? Confidentiality is protecting information from being read or copied by anyone who has not been explicitly authorised by the owner of that information. Most of the data that traverses computer networks can claim no pretence whatsoever of confidentiality. Protocols such as Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP) and Hypertext Transfer Protocol (HTTP) send their data in clear text and so anyone that lays hands on the data can read it. However, solutions are now available so that sensitive data can be encrypted into an unintelligible format before transmission, and decrypted after delivery. Encryption provides more confidentiality because only intended recipients can decrypt the messages. Encryption solutions include, among others, programs like Pretty Good Privacy (PGP) and Privacy Enhanced Mail (PEM) for securing e-mails.

Non-repudiation

Non-repudiation is mostly achieved through the use of digital signatures. Non-repudiation means that after one has signed and sent a message, one cannot later claim that one did not sign the original message. One cannot repudiate one's signature, because the message was signed with one's private key, for instance (which, presumably only the owner knows). This is an important factor when exchanging sensitive information over the network.

Auditing

As well as worrying about unauthorised users, authorised users sometimes make mistakes, or even commit malicious acts. In such cases, one needs to determine what was done, by whom, and what was affected. Auditing can be of great help in such cases since it provides some incorruptible record of activity on a system that positively identifies the actors and their actions. Extensive logging is prerequisite to successful auditing.

Availability

Availability refers to protection against disruption of service: The question to ask is does data remain accessible for legitimate uses? Networks aim at making their services available to their authorised users. Hence a top priority when running a network is to ensure that it is well protected and that its services are not degraded or made unavailable (e.g. a system crash) without authorisation. An unavailable system denies its legitimate and authorised users the needed services. This state of affairs can lead to disruption of production.

Responsibility and control

Many organisations discover that they cannot design a security policy because the organisation has not specified how responsibility for information is assigned or controlled. The issue of responsibility for information has several aspects to consider:

- *Accountability*. Accountability refers to how an audit trail is kept: which group is responsible for each item of data? How does the group keep records of access and change?

- *Authorisation*. Authorisation refers to responsibility for each item of information and how such responsibility is delegated to others: who is responsible for where information resides and how does a responsible person approve access and change?

The critical issue underlying both accountability and authorisation is control - an organisation must control access to information analogous to the way the organisation controls access to physical resources such as offices, equipment and supplies.

Firewalls

A firewall is a computer system, which is used to separate two networks from each other. The firewall only allows certain types of network traffic to pass through it from one network to the other. With the boom of the Internet, many organisations connect their local area networks (intranets) to the Internet. This allows them to share the vast amount of information that is available on the Internet. Besides, these organisations can also advertise themselves through the Internet. However, being a public network, the Internet, is full of security risks. An organisation that connects to the Internet exposes its local data and information to all the risks that exist on the Internet. Much as many organisations wish to connect to the Internet, they still want their private data and information to remain untampered with by intruders. To ensure that all access from external networks (such as the Internet) to the local network is controlled, a firewall is placed in between the local network and the Internet. The firewall will then ensure that only particular types of network traffic that satisfy the network security policy of the organisation is allowed through. This controlled access tremendously reduces the risks of unauthorised access to the local network from external networks. Unauthorised access from the Internet could lead to private and sensitive information on the local network being changed, deleted or copied to other areas. In performing its task to control access to the local network, the firewall uses network policy rules that are applied on the network traffic packets, using one of the following two principles:

- Only the type of network traffic that has been specified in the policy rules should be allowed and all other network traffic should be prohibited.
- Only the type of network traffic that has been specified in the policy rules should be prohibited and all other network traffic should be allowed.

The first principle is quite strict and allows the firewall to offer maximum security. However, it requires much work to keep on changing the rules as new services appear which an organisation wishes to use. The second principle offers little security but is much more flexible. The second principle is also powerless in case of new attacks. To achieve tight security, the first principle should be preferred.

Firewall Designs

There are many different designs of firewalls. These include packet filtering routers, Bastion hosts, Dual Homed hosts and combinations of these.

Packet Filtering Routers (Screening Routers)

Screening routers are normally store-and-forward routers, which have been extended to include some packet filtering capabilities. Apart from their usual function of routing packets, they filter these packets according to the network security policy and so function as firewalls. Such filtering is done at the Internet and Transport Layers of the TCP/IP model using IP addresses and sometimes port numbers. Screening

routers are the simplest type of firewalls and hence offer limited security. However, filtering routers are completely transparent to the users on the local network.

To prevent each computer on a network from accessing arbitrary computers or services, many sites use a technique known as *packet filtering*. As figure 3 shows, a packet filter is a program that operates in a router. The filter consists of software that can prevent packets from passing through the router on a path from one network to another. A manager must configure the packet filter to specify which packets are permitted to pass through the router and which should be blocked. *(Please note: Some LAN switches also provide a similar form of filtering that allows a manager to configure which frames are permitted to pass from one computer to another and which should be blocked)*. A packet filter operates by examining fields in the header of each packet. The filter can be configured to specify which header field to examine and how to interpret the value. To control which computers on the network can communicate with computers on another, a manager specifies that a filter should examine the *source* and *destination* fields in each packet header. In figure 1, to prevent a computer with IP address 192.5.48.27 on the right-hand network from communicating with any computer on the left-hand network, a manager specifies that the filter should block all packets with a source address equal to 192.5.48.27. Similarly, to prevent a computer with address 128.10.0.32 on the left-hand network from receiving any packets from the right-hand network, a manager specifies that the filter should block all packets with a destination address equal to 128.10.0.32.

In addition to using the source and destination addresses, a packet filter can examine the protocol in the packet or the high-level service to which the packet corresponds. The ability to selectively block packets for a particular service means that a manager can prevent traffic to one service, while allowing traffic to another. For example, a manager can configure a packet filter to block all packets that carry World Wide Web communication, while allowing packets that carry e-mail traffic. A packet filter mechanism allows a manager to specify complex combinations of source and destination addresses and services. Typically, the packet filter software permits a manager to specify Boolean combinations of source, destination and service type. Thus, a manager can control access to specific services on specific computers. For example, a manager might choose to block all traffic destined for FTP service on computer 128.10.2.14, all WEB traffic leaving computer 192.5.48.33 and all e-mail from computer 192.5.48.34. The filter blocks only the specified combinations - the filter passes traffic destined for other computers and traffic for other services on the specified computers.

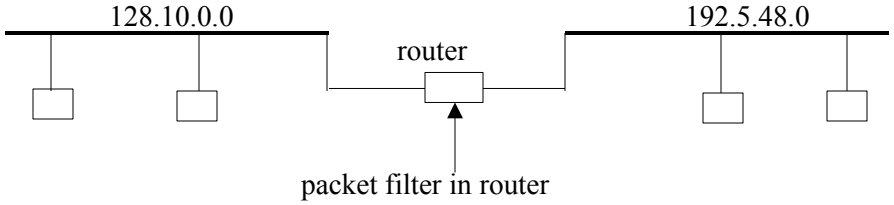

Figure 3: Illustration of the location of a packet filter.

Bastion host

In a Bastion hot set-up, the internal and external networks are separated by a screening router. The internal network has a special and well-secured host - the Bastion host. The screening router only allows packets to and from the Bastion host. This forces all traffic to pass through the firewall (Bastion host) and so making the Bastion host design ideal for proxying. The Bastion host has no routing capability. Any attacker who intends to attack the local network from the external network has to go through the Bastion host. As a result, the more secure the Bastion host, the more secure the local network as well. For internal users, the Bastion host set-up of a firewall is less transparent. The local network hosts also need to be configured specifically to suit the Bastion host set-up.

Dual-Homed host

The Dual Homed host set-up of a firewall uses two routers. One of the routers has a connection to the external network while the other has a connection to the internal network. The area in between the two routers is called the Demilitarised Zone (DMZ). The Demilitarised Zone consists of at least one host, the firewall. The external router (the one connected to the external network) only allows packets through if they are addressed to hosts on the Demilitarised Zone. As a result, all network traffic from the outside is made to go through the firewall where it is analysed for suitability using the firewall rules, before being allowed into the internal network. Traffic from the inside, too, is forwarded to the firewall by the internal router. This subjects all traffic to and from the internal network to be checked against the network policy on the firewall. This set-up ensures that there is no direct connection between the external and the internal network. The internal hosts are totally invisible from the external network. The Dual Homed host is therefore suitable for running proxying software and also for providing Network Address Translation (NAT) so as to hide the internal network structure and addresses. The firewall on the Demilitarised Zone has two interfaces (homes), one for the external and the other for the internal connection, hence the name Dual Homed host.

Firewall Filtering Principles

The filtering of packets depends on whether the firewall is a:

- packet filter,
- circuit-level firewall,
- application-level firewall or
- hybrid firewall.

When filtering, each network packet that passes through the firewall will be evaluated against the following rules:

(a) If no matching rule is found, then the network packet is dropped.
(b) If a matching rule is found that permits the communication, then a peer-to-peer communication is allowed.
(c) If a matching rule that denies the communication is found then the network packet is dropped.

Static Packet Filter Firewalls

A packet filter firewall analyses network traffic at the network and transport protocol layers. Flags within the protocol headers can also be checked, for example, to determine if a packet is part of an established TCP session. Each IP network packet is examined to see if it matches one of a set of rules defining what data flows are allowed, depending on the following:

(a) the physical network interface that the packet arrives on
(b) the address the packet is supposedly coming from (e.g., source IP address)
(c) the address the packet is going to (e.g., destination IP address)
(d) the transport layer protocol (e.g., TCP, UDP)
(e) the source port and
(f) the destination port of the packet.

Packet filter firewalls are stateless. Each packet is examined in isolation from previous packets. Logging of packets is also done without regard to previous packets.

Advantages of packet filtering

(a) It is relatively easy to add new protocols since this requires only modifying filtering rules. As a result new services can be taken advantage of as they become available.
(b) They are generally faster than other firewall technologies because they perform relatively fewer evaluations.
(c) By using network address translation, they can shield internal IP addresses from external users.
(d) They are transparent to users as they require no client computers to be specifically configured.

Disadvantages of packet filtering

(a) They do not necessarily understand application layer protocols. As a result they may allow malicious commands in the payload data to enter the protected network.
(b) They are stateless in that they do not keep information about a session or application-derived information. Thus, they may not be in position to correlate events.
(c) The filtering languages may not usually be easy to implement correctly.
(d) Since they do not provide application-layer awareness, logging may also not be all that extensive. This can lead to ineffective network auditing.

Dynamic Packet Filter Firewalls

A dynamic packet filter is a firewall system that can monitor the state of active connections and use this information to determine which network packets to allow through the firewall. The firewall accomplishes its functional requirements by tracking and matching requests and replies and so it can screen for replies that don't match a request. This makes dynamic packet filtering firewalls ideal for filtering User Datagram Protocol (UDP) traffic. When a request is received, the dynamic packet filter records session information such as IP addresses and port numbers, and then it opens up a virtual connection so that only the expected data reply is let back through this virtual connection. Once the reply is received, the virtual circuit is closed. The state information associated with the virtual connection is typically remembered for a short period of time, and if no response packet is received within a given period of time then the virtual connection times out and is consequently invalidated. The ability to remember the state of the virtual circuit enables dynamic packet filters to have increased security capabilities than static packet filters. Dynamic packet filter firewalls have almost the same advantages and disadvantages associated with static packet filter firewalls, except that they do not allow unsolicited UDP packets on the local network.

Circuit-level Firewalls

A circuit level firewall, validates the fact that a packet is either a Transmission Control Protocol (TCP) connection request or a data packet belonging to a TCP connection between two peer transport layers. To validate a session, a circuit level firewall examines each connection set-up to ensure that it follows a legitimate handshake. Data packets are not forwarded until the handshake is complete. The firewall maintains a table of valid connections that includes complete session state and sequencing information. Data packets matching an entry in the table are allowed to pass through the firewall. When the connection is terminated then the corresponding table entry is also removed from the table and the virtual circuit between the two peer transport layers is closed. When a connection is set up, the circuit level firewall typically stores the following information about the connection:

(a) a unique session identifier for the connection, which is used for tracking purposes

(b) the state of the connection (e.g., handshake, established, or closing)
(c) the sequencing information
(d) the source IP address
(e) the destination IP address
(f) the physical network interface through which the packet arrives and
(g) the physical network interface through which the packet departs.

Advantages of Circuit-level Firewalls

(a) They are generally faster than application layer firewalls because they perform fewer evaluations.
(b) They can be used to prohibit connections between specific external hosts and internal hosts.
(c) By using network address translation, they can shield internal IP addresses from external users.

Disadvantages of Circuit-level Firewalls

(a) They can only restrict access to TCP.
(b) They cannot perform strict security checks on higher-level protocols. As a result, packets with malicious commands in their payloads may be allowed through the protected network.
(c) They offer limited logging and auditing as they are not application-layer aware.

Application-level Firewalls (proxies)

Application-level firewalls (also called proxies) evaluate network packets for valid data through all the layers up to the application layer before allowing a connection through the firewall. Proxies control connection establishment and user authentication based on source/destination addresses and port numbers and they maintain complete connection state and sequencing information. They mediate traffic by paying attention to particular protocol data and commands in the data payload. Since proxies are protocol-specific, they can provide increased access control, detailed checks for valid data, and generate detailed audit records about the traffic that they transfer. Proxy services never allow direct connections between internal and external hosts, hence they force all network packets to be examined and filtered for suitability.

Advantages of Application-level Firewalls

(a) Since they are end-points for communications, they guarantee that untrusted users and services communicate only to the proxy. This secures the addressed services and hosts. Also proxies are usually written with security in mind.
(b) Since they understand the application layer protocols, they can offer flexible security policy controls, authentication, increased access controls, detailed checks for valid data, extensive logging and auditing.
(c) New proxies are usually easy to implement.
(d) They offer Network Address Translation (NAT) and so shielding internal IP addresses from external users.

Disadvantages of Application-level Firewalls

(a) New application layer protocols that need to be added to the firewall will require new proxies.
(b) They may be less transparent, requiring users to have to connect to the firewall, which can expose the firewall to undesirable manipulations. Application-level firewalls may also require modified client software.
(c) Due to the extensive processing that they have to do on network packets, they may have lower performance as compared to firewalls, which operate at lower layers.
(d) Because proxy servers listen on the same port as network servers, one cannot run such network servers on the firewall server.

Hybrid firewalls

Hybrid firewalls combine the characteristics and techniques of two or more other firewall types. A hybrid firewall possesses all the advantages and disadvantages of the firewall types that it combines.

Virtual Private Networks (VPN)

Virtual Private Networks (VPN) may be included into firewalls and so adding value to firewall systems. A virtual private network is a temporary and secure connection over a public network, usually the Internet, for the purpose of exchanging private information. Traffic over public networks is vulnerable to eavesdropping (snooping) attacks. Hence, a virtual private network encrypts data over a connection on a public network to protect the information from being revealed if intercepted. This makes virtual private networks ideal for securing on-line communications over public networks. Since they use public networks for communications, virtual private networks offer significant cost savings, greater flexibility, and easier management as compared to private networks like leased lines and dial-up remote access.

Virtual private networks use strong authentication, encryption and access control. Strong authentication technologies like token cards, smart cards, digital certificates, biometrics (fingerprints and iris scanning) enable the verification of individual identities and hence their activities on the network.

Virtual Private Network Implementations

Although there are many types of VPN implementations, they can be grouped into three main categories:

(a) Intranet VPN: between an organisation's branch offices.
(b) Remote Access VPN: between an organisation's remote or travelling employees.
(c) Extranet VPN: between an organisation and its associations, such as partners, customers, suppliers, and investors.

Intranet VPN

Intranet virtual private networks are temporary LAN-to-LAN connections that link branches of the same organisation, which are geographically separated from each other. This scenario is commonplace in today’s globalisation. Since the sites belong to one organisation and therefore have one and the same security policy, the branches have some trusted relationship to each other. Hence an intranet virtual private network establishes encrypted bi-directional tunnels between trusted local area networks across the Internet. Highly secured intranet virtual private networks can also ensure that only certain individuals at one branch office have access to the resources of the other branch, and that each individual user has a different set of permissions. All data transferred across the Internet is completely encrypted and authenticated all the way to the endpoints.

Remote Access VPN

Remote access virtual private networks are mostly used by mobile and telecommuting employees to enable them to have access to the organisations resources from a remote location. Apart from strong encryption methods, a remote virtual private network also requires strong authentication methods. Once an employee has authenticated to the organisations virtual private network server, a certain level of access is granted depending on the profile of the particular employee. If authentication fails then access is denied. During an authenticated session, all data is encrypted from one endpoint to the other.

Extranet VPN

Extranet virtual private networks are intended to reach networks of varying security policies such as those of an organisation's partners, customers, and suppliers. Hence, extranet virtual private networks provide a hierarchy of security, with access to the most sensitive data being nested under the tightest security control. An extranet virtual private network filters access to network resources based on several parameters including:

(a) source IP address,
(b) destination IP addresses,
(c) application usage,
(d) type of encryption and authentication used, and
(e) individual, group and subnet identity.

To be able to identify individual users, strong authentication methods such as token cards, smart cards, digital certificates, biometrics (fingerprints and iris scanning) are employed.

The Internet firewall concept – The Summary

A packet filter is often used to protect an organisation's computers and networks from unwanted Internet traffic. As figure 4 illustrates, the filter is placed in the router that connects the organisation to the rest of the Internet. A packet filter configured to protect an organisation against traffic from the rest of the Internet is called an Internet firewall. The term is derived from the fireproof physical boundary placed between two structures to prevent fire from moving between them. An Internet firewall is designed to keep problems in the Internet from spreading to an organisation's computers. Firewalls are the most important tools used to handle network connections between two organisations that do not trust each other. By placing a firewall on each external network connection, an organisation can define a secure perimeter that prevents outsiders from interfering with the organisation's computers. In particular, by limiting access to a small set of computers, a firewall can prevent outsiders from probing all computers in an organisation, flooding the organisation's networks with unwanted traffic, or attacking a computer by sending a sequence of IP datagrams that is known to cause the computer system to misbehave (e.g., to crash).

A firewall can lower the cost of providing security. Without a firewall to prevent access, outsiders can send packets to arbitrary computers in an organisation. For example, an outsider can guess the IP address of the computers in an organisation by finding the set of network numbers that the organisation has been assigned and then trying each of the possible hosts on those networks. Consequently, to provide security, an organisation must make all of its computers secure. With a firewall, however, a manager can restrict incoming packets to a small set of computers. In the extreme case, the set can contain a single computer. Although computers in the set must be secure, other computers in the organisation do not need to be. Thus, an organisation can save money because it is less expensive to install a firewall than to make all computer systems secure.

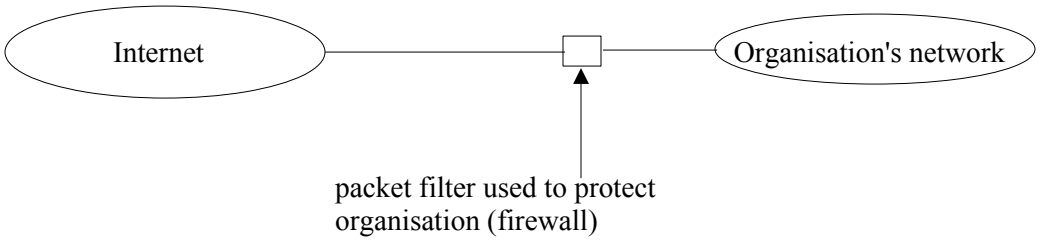

Figure4: Illustration of a packet filter used as the primary part of a firewall that protects an organisation against unwanted traffic from the Internet

CopperbeltUniversity
Computer Science Department

Computer Systems Security

Compiled by: Prof.Dr. Hastings M. Libati

Intrusion Detection Systems (IDS)

In this lesson we shall deal with intrusion detection systems.

What is an intrusion detection system?

A network intrusion is any set of actions that attempt to compromise the integrity, confidentiality or availability of a network resource. This can be achieved through unauthorised use, misuse or abuse of computer systems. Individuals who perpetrate intrusions in computer and network systems are referred to as attackers or intruders. Examples of intrusions include:

1. Unauthorised use of computing resources and modifications of system files so as to permit unauthorised access to either system or user information.
2. Unauthorised modifications of tables or other system information in network components (e.g., modifications of router tables in an internet to deny use of the network to legitimate network users).

Intrusion Detection (ID) is the process of identifying individuals who are using (or intend to use) a computer system or service without authorisation and those who have legitimate access to the system but are abusing their privileges. An intrusion Detection System (IDS), therefore, is a computer system that monitors other systems for the purpose of detecting intrusions.

Classification of Intrusion Detection Systems

Intrusion detection systems take either a network- or a host-based approach to recognising and deflecting attacks. Figure 5 shows a classification of intrusion detection systems. Intrusion detection systems look for specific patterns (attack signatures) that usually indicate malicious or suspicious intent. When an intrusion detection system looks for these patterns in network traffic by monitoring the traffic, it is network-based.

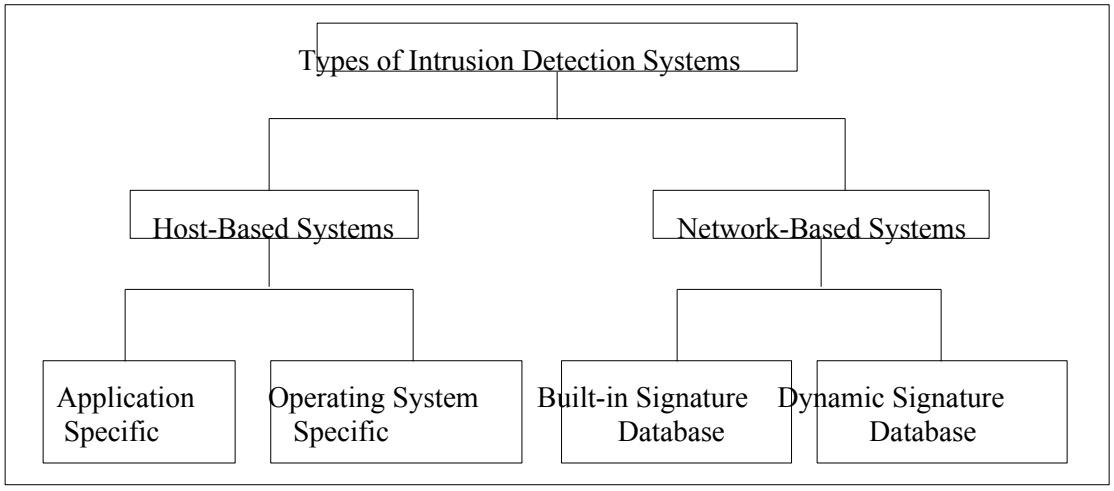

Figure 5: Classification of Intrusion Detection Systems

When an intrusion detection system looks for attack signatures in log files (audit trails), it is host-based. Furthermore, an intrusion detection system either employs anomaly or the misuse detection model.

The anomaly detection model operates on the assumption that users (or group of users) and networks always exhibit a predictable pattern of behaviour and do not depart from this pattern over short periods of time. An intrusion detection system based on anomaly detection must first be trained to know the expected behaviour (normalcy profile) of users. This normalcy profile is built using statistical analysis of each user's use of the system and logical rules that define likely behaviour for various types of users. Once a normalcy profile has been established, the intrusion detection system monitors the system by comparing each user's activity with the particular user's normalcy profile. If some activity deviates markedly from the profile then the intrusion detection system flags it as anomalous and, therefore, a possible intrusion. Because this type of intrusion detection system looks for sessions, which are not normal, it is referred to as an *anomaly detection model*. A profile is a description of a subject's normal (expected) behaviour with respect to a set of intrusion detection measures.

The anomaly detection model, however, is vulnerable to defeat by internal users who know that their behaviour is being compared with their previously established behaviour patterns and slowly vary their behaviour overtime, until they have established a new behaviour pattern within which they can safely mount an attack. Trend analysis on user behaviour patterns that observes how fast user behaviour changes overtime, may be useful in detecting such internal attacks.

Using the anomaly detection model, a legitimate user can be flagged as an intruder (a false positive) because abnormal behaviour is not necessarily an attack. For example, a legitimate user may become more proficient in using a program and thus employ commands not previously invoked. Also, false negatives can occur when an intruder's actions closely resemble the normal behaviour of a legitimate user whose login has been illegally obtained. Establishing the right time period over which to analyse the user's behaviour and how often to retrain the intrusion detection system can affect its performance.

The misuse detection model, on the other hand, bases its detection upon a comparison of parameters of a user's session and the user's commands to a rule-base of techniques (attack signatures) used by attackers to penetrate computers and network systems. Since this model looks for patterns known to cause security problems (i.e., known attack methods), it is called a *misuse detection model*. This type of intrusion detection system may use an expert system which contains rules that describe attack signatures based on

the knowledge of past intrusions, known system vulnerabilities, or the site-specific security policy. However, the misuse detection model of intrusion detection systems is vulnerable to intrusion scenarios that use attack signatures, which are not described in the knowledge base.

Host-based intrusion detection systems

Host-based intrusion detection systems are designed to monitor a single host on which the intrusion detection system agent resides. Host-based intrusion detection systems are used to protect critical network servers or other individual systems containing sensitive information, see figure 6. The host-based intrusion detection system monitors different aspects of the server security such as operating system log files or application system log files. Host-based systems require the use of a host server's system resources such as disk space, Random Access Memory (RAM) and Central Processing Unit time. This can have a negative impact on the system's performance. In client-server communication scenarios, for example, the server logs the activities of a client accessing an application server and the intrusion detection system agent polling these log files extracts and analyses them. Intrusions are detected by analysing the audit trails and other system activities which; occur locally on the monitored host, such as file permission modifications and user accounts; and tries to match that data against a statistically known user's normal behaviour profile and attack patterns. When a deviation from a user's normal profile is observed or a suspicious pattern is observed, the intrusion detection system flags the activity as suspicious and reports it. Host-based intrusion detection systems are ideal if a limited number of critical servers must be protected, but they do not scale well if a large number of hosts needs to be protected.

Host-based intrusion detection systems make extensive use of auditing. This requires audit events such as user accesses to protected file and changes in access privileges to be logged (recorded). A log file is a collection of audit events (audit trails). The audit trails represent the history of the system and each logged event (audit trail) represents any change in the state of the system that is related to its security. Auditing is the process of analysing the audit trails. Through careful use of auditing and audit trails, several categories of attackers can be identified:

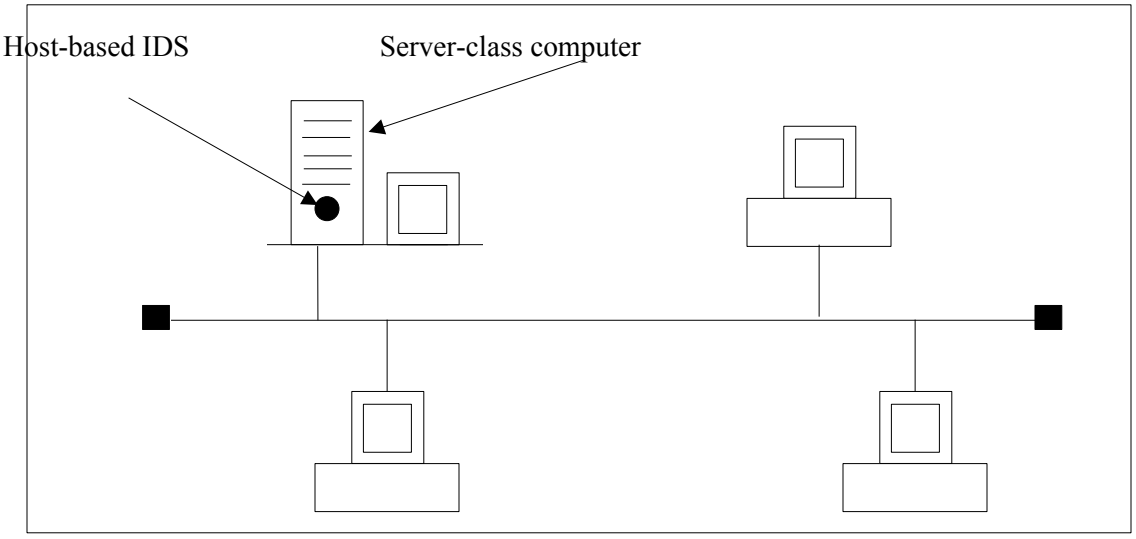

Figure 6: A host-based intrusion detection system

1. External attackers: These are attackers who are not authorised to use the computer resources. They can be detected by auditing failed log-in attempts.

2. Internal attackers: These are attackers who are authorised to use the computer system but are not authorised to use certain data, programs or other resources, which they have accessed. These can be detected by observing failed access attempts to data files, programs and other resources.

3. Masquerades: These are people who operate under another user's identity and password. Masquerades can be detected by observing departures from established patterns of use for given individual users.

Some host-based intrusion detection systems detect intrusions by digitally signing key system files and executables via checksums and then checking them at regular intervals for unexpected changes. Other host-based intrusion detection systems listen to port activities and alert administrators when specific ports are accessed. This type of detection brings an elementary level of network-based intrusion into host-based environment.

Network-based intrusion detection systems

The main limitation of host-based intrusion detection systems is that the access to audit trails is available only at the operating system level or at the application level. The evolution of large networks requires monitoring data at all levels of communication. This shortcoming is overcome by network-based intrusion detection systems.

The network-based intrusion detection system has access to data at all layers of communication. Therefore, this type of intrusion detection system can do extensive analysis for attack detection. Since the network-based intrusion detection system runs on a computer other than the host being monitored (see figure 7), there is no performance impact on the monitored computers at all. A network-based intrusion detection system typically utilises a network adapter running in promiscuous mode to monitor and analyse all traffic preferably in real-time as it travels across the network.

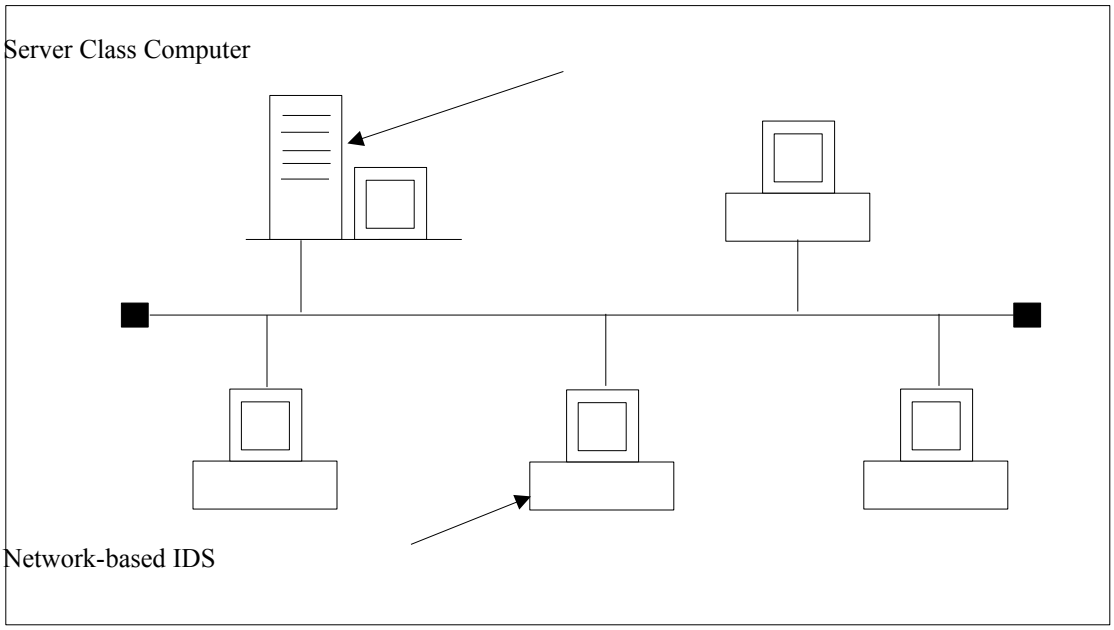

Figure 7: A Network-based Intrusion Detection System

Network-based intrusion detection systems are used to monitor activities on a specific network segment. The systems analyse packet headers and packet payload data and form their attack detection upon comparison of the packet details to a rule-base of attack techniques (attack signatures) that attackers use to penetrate a computer or network system. Network-based intrusion detection systems use either context or content oriented attack signature methodology or both. Context-oriented attack signatures consist of known network service vulnerabilities that can be detected by inspecting packet headers. These include TCP hijacking, source routing and IP spoofing attack profiles. Content-oriented signatures require the inspection of the payload or datafields within a packet to determine if an attack or policy violation has occurred at the application level (e.g., attacks through Java applets and Active X).

The use of attack signatures to detect attacks requires constant updates of the signature database so that the intrusion detection system has current knowledge of any latest security vulnerabilities.

Network-based intrusion detection systems scale well to network protection because the number of hosts being monitored is not a critical factor. It is only the amount of traffic that matters. In addition, several intrusion detection systems placed in different subnets can be configured to report back to a central site so enabling central monitoring of a large network.

Most network-based intrusion detection systems utilise a built-in attack signature database (rule-base) and are therefore called *static signature intrusion detection systems*. In static signature intrusion detection systems, each attack signature is processed using a set of functions to detect that specific signature. The actual static signature (built-in signature) database is a collection of such functions, one for each attack signature, which is built into the system. This approach has its own weaknesses. A new processing function has to be added every time a new signature is added to the database. Static signature intrusion detection systems:

1. offer no real-time extensibility of new attack signatures,

2. there is also a large overhead on the intrusion detection system's performance because of the sequential execution of processing functions, and

3. the performance of the intrusion detection system degrades even further as more signatures are built-in.

To overcome the limitations of extensibility and performance of the built-in signature intrusion detection systems, some intrusion detection systems use a technique *called Stateful Dynamic Signature Inspection (SDSI)* (e.g., Axenttechnologies'sNetProwler). In this design, each attack signature is a set of instructions, which the SDSI virtual processor executes using a cache entry describing the current user session state and the current packet received from the network. As we have seen, NetProwler has an advanced method of detection called Stateful Dynamic Signature Inspection (SDSI) to detect network-based attacks.

NetProwler is stateful because it can remember the contents of the active sessions that it monitors on the network. Therefore, rather than simply comparing an attack signature with a single packet, NetProwler builds a context around a network session. This allows NetProwler to monitor and prevent much more sophisticated attacks than the simple exploits that a single packet of data may contain. For example, NetProwler can detect attacks that occur in separate actions or steps. Furthermore, NetProwler is dynamic in the sense that it can also create new attack signatures and have them activated in real time without having to take the system offline. In addition, this technology allows one to customize NetProwler to the needs of one's organization. In this way, NetProwlercanbe made to respond smoothly to the threats that an organization faces. Signature Inspection is the method of detection that NetProwler uses and works by comparing an attack signature, which is a set of rules that describe an attack, to a communication packet.

Network-based IDS Placement

A network-based intrusion detection system can be placed behind a firewall (i.e., between the internal protected network and the firewall) or in front of a firewall (i.e., between the firewall and the external network) as shown in figure 8. Placing an intrusion detection system in front of a firewall enables it to monitor out-bout traffic as well as in-bound traffic that would otherwise be blocked by the firewall. Monitoring traffic that is targeting the protected network has the advantage of providing administrators with information that enables them to adjust security policies and deploy security tools to best protect the network.

However, network-based intrusion detection systems may be installed on any network segment that is critical to the operation of an organisation.

Advantages of Host-Based IDS

Host-based intrusion detection systems do offer advantages, which include stronger forensic analysis. They can enforce a user based reaction policy when an attack is detected, such as disabling a user account or terminating a user process.

Other advantages include:

1. *Verifying success or failure of an attack*: Since host-based intrusion detection systems use logs containing events that have actually occurred, they can measure whether an attack was successful or not with greater accuracy and fewer false positives. In this respect, host-based intrusion detection systems make an excellent complement to network-based intrusion detection, with the network component providing early warning and the host component providing verification of whether an attack was successful. This can also help to verify the effectiveness of the existing system protection and access control mechanisms.

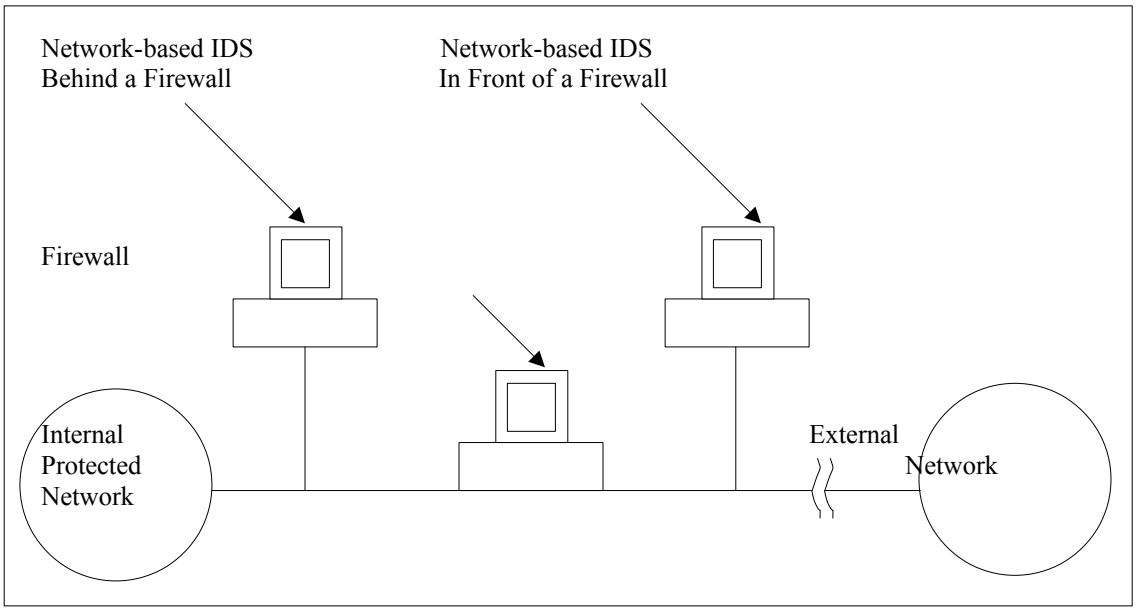

Figure 8: Network-based Intrusion Detection System Placement

2. *Monitoring specific system activities*: Host-based intrusion detection systems monitor user and file access activity, including changes to file permissions, attempts to install new executables and / or attempts to access privileged services. For example, a host-based intrusion detection system can monitor all user log-in and log-off activities, as well as what each user does while connected to the network. Host-based intrusion detection systems can also monitor activities that are normally executed only by an administrator. Operating systems log any event where user accounts are added, deleted or modified. The host based intrusion detection system can detect an improper change as soon as it is executed. Host-based intrusion detection can also audit policy changes that affect what systems track in their logs. Host-based intrusion detection systems can monitor changes to key system files and executables. Attempts to overwrite vital systems or to install Trojan horses or back-doors can be detected, reported or even stopped. Users with inappropriate access privileges can also be detected.

3. *Detecting attacks that network-based systems miss*: Host-based intrusion detection systems can detect attacks that cannot be recognised by network-based intrusion detection systems. For example, attacks from the keyboard of a critical server-class computer do not cross the network, and so cannot be detected by a network-based intrusion detection system.

4. *Well-suited for encryption and switched environments*: Since host-based intrusion detection systems can reside on various hosts throughout an enterprise, they can overcome some of the

deployment challenges faced by network-based intrusion detection systems in switched and encrypted environments. Host-based intrusion detection systems provide greater visibility in a switched environment by residing on as many critical hosts as needed. Depending on where the encryption resides within a protocol stack, it may leave a network-based intrusion detection system blind to certain attacks. Host-based intrusion detection systems do not have this limitation. By the time an operating system, and therefore the host-based intrusion detection system, handles incoming traffic, the data stream has already been decrypted.

5. *Near real-time detection and response*: Although host-based intrusion detection systems do not offer true real-time response, they can come extremely close if implemented correctly. Some host-based intrusion detection systems receive an interrupt from the operating system when there is a new log file entry (e.g., AxentTechnologies's Intruder Alert for Netware). This new entry can be processed immediately, significantly reducing the time between attack recognition and response. There remains a delay between when the operating system records the event and when the host-based intrusion detection system recognises it, but an intruder can be detected and stopped before much damage is done.

6. *Require no additional hardware*: Host-based intrusion detection systems reside on existing network infrastructure, including file servers, web servers and other shared resources. However, this efficiency can make host-based intrusion detection systems very cost effective.

Limitations of Host-based intrusion detection systems

1. Since they reside on the host which they monitor and therefore share the same computing resources, this can negatively impact the performance of the monitored hosts.
2. Network-based attacks, which exploit protocol and network vulnerabilities may be hard to detect.
3. Since they analyse data from the audit trails, reaction to an attempted intrusion may not be in real-time.
4. The complexity of deployment and administration increases with the number of hosts that have to be monitored.

Most limitations of host-based intrusion detection systems are cared for by network-based intrusion detection systems.

Advantages of Network-based Intrusion Detection Systems

Since network-based intrusion detection systems can monitor network traffic at all the 7 layers of the ISO/OSI model, they provide good overall network security. Other advantages of network-based intrusion detection systems include:

1. *Lower cost of ownership:* Network-based intrusion detection systems allow strategic deployment at critical access points for viewing network traffic destined to multiple systems. Since fewer detection points are required, the cost of ownership is lower for an organisation.

2. *Detect attacks that host-based intrusion detection systems miss:* The increase in internetworking also means that many more attacks can be launched over the network. The network itself, being

an important component of a computing environment can be the object of attack. Since network-based intrusion detection systems examine all packet headers and packet data fields for signs of malicious and suspicious activity, it can monitor attacks launched against the network itself, an attack that host-based audit trail analysers would probably miss. For example, a denial of service through the Christmas-Tree attack can only be detected by analysing the network traffic packets.

3. *More difficult for an attacker to remove evidence*: Network-based intrusion detection systems use live network traffic for real-time attack detection. Therefore, an attacker cannot remove the evidence. Captured data includes not only the method of attack, but information that may help lead to attacker identification and prosecution. In comparison, host-based intrusion detection systems depend on audit trails for evidence of attack. However, an audit trail is also typically the first item intruders will modify once they successfully penetrate a computer or application system.

4. *Real-time detection and response*: Network-based intrusion detection systems detect malicious and suspicious activities as they occur, and so provide faster notification and response. Real-time notification allows rapid reaction according to predefined parameters.

5. *Detect unsuccessful attacks and malicious intent*: Network-based intrusion detection systems add valuable data for determining malicious intent. A network-based intrusion detection system placed in front of a firewall can detect attacks intended for resources behind the firewall, even though the firewall may be rejecting these attempts. This information can be critical in evaluating and refining security policies.

6. *Operating system independence*: Network-based intrusion detection systems are not dependent on host operating systems as detection sources and since they use standard network protocols like TCP/IP, which most major operating systems use, one intrusion detection system can monitor a heterogeneous set of hosts and operating systems simultaneously.

7. *No performance impact on monitored hosts*: Network-based intrusion detection systems do not degrade the performance of the hosts being monitored. The monitored hosts are not even aware of the intrusion detection systems.

Limitations of network-based intrusion detection systems

1. The performance of a network-based intrusion detection system can vary due to the speed of the network being monitored, the amount of traffic, the number of nodes being protected, the number of attack signatures employed and the power of the platform on which the intrusion detection system resides. In general, intrusion detection systems may not perform well on busy networks. However, multiple intrusion detection systems can be placed on a given segment to sub-divide host protection, therefore increasing performance and overall protection.

2. High-speed networks, such as Asynchronous Transfer Mode (ATM), which use packet fragmentation to improve bandwidth usage, can pose problems to intrusion detection systems in terms of performance and response.

3. Switched networks do not provide the static IP range needed for network intrusion detection systems to operate effectively. One could also mirror switched ports through a switched management port.

Most of the limitations of network-based intrusion detection systems are catered for by host-based intrusion detection systems.

Desirable Intrusion detection systems characteristics

Intrusion Detection Systems (IDS) must meet a number of requirements to provide truly effective protecti on against attacks. The major requirements include:

1. *Real-time operation*: The attack and recognition software must be capable of detecting, reporting and reacting to suspected attacks in real-time. Software that merely logs events and provides audit logs for examination after-the-fact is ineffective. Attackers erase logs during break-ins, so their intrusion may not be detected by merely scanning an event log.

2. *Capable of update*: Attackers continually find new ways to break into computer systems. As a result, intrusion detection systems must be capable of continually adding to their knowledge base of known break-in patterns and unauthorised activities.

3. *Run on popular operating systems*: Intrusion detection systems should be able to support existing network operating systems such as UNIX and Microsoft Windows-based operating systems.

4. *Easy to configure*: Configuration should be easy without sacrificing effectiveness.

5. *Easy to manage*: Intrusion detection systems must be easy to manage. In addition, the software should be easy to integrate with existing network management infrastructure. This requires compliance with network management standards such as the Simple Network Management Protocol (SMNP). It must run continuously with minimal human supervision and should be able to scale so as to monitor a large number of hosts while still providing results in a timely and accurate manner.

6. *Adaptable to changing security policies*: To remain effective, intrusion detection systems should be easy to adapt to changing security policies.

7. *Non-obtrusive*: The software should not degrade network performance and should be transparent to authorised users so that it does not hamper productivity. In addition, it should not alert the intruder to its presence.

8. *Comprehensive recognition of intrusion*: It should recognise all or most intrusions and should have a low number of false intrusion alarms (false positives) and false negatives. In identifying attacks, it should give sufficient characterisation data to support an effective response and it should require a minimum of apriori information about potential attackers and their methods.

9. *Interoperability*: No single security tool can provide comprehensive security, but through interoperating with other security tools, a much higher degree of security can be achieved. Consequently, it is desirable for an intrusion detection system to be able to interoperate with other security tools through such standards as Open Platform for Security Enterprise Connectivity (OPSEC) and the Common Intrusion Detection Framework (CIDF).

Intrusion Detection Systems Response to an Intrusion

An intrusion detection system can be configured to react automatically to an attack in a variety of ways as given below:

1. *Log the event*: The intrusion detection system can log the event with associated information. This can later help to determine the responsible attacker and the logged information can be used as evidence if legal action is taken against the attacker. However, determining whom to hold accountable can be very difficult in the case of network-based intrusion detection systems, since most network protocols do not associate users with connections and the attack might be laundered through multiple compromised accounts and might cross multiple administrative domains.

2. *Block the connection*: The intrusion detection system can call a user-defined script or program which can then send commands directly to network equipment such as routers and firewalls, reconfiguring them to block further connections from the attacker's site.

3. *Turn defence into offence*: The system can be configured to turn defence into offence by countering the attacker's activities with misleading or incorrect information or even luring the attacker to a system designed specifically to monitor intruders.

4. *Administrator notification*: The system can automatically log the attacker off and alert the administrator through console messages, e-mail or pagers.

5. Combine *actions*: The intrusion detection system can be configured to perform a combination of the above actions.

Once a violation has been detected, the attacked system needs to be analysed to determine the immediate cause of the system's vulnerability and the extent of the damage. Knowing the vulnerabilities exploited by the attacker can often help to stop on-going attacks and future ones. If the vulnerability cannot be fixed, knowing its causes helps determine what to monitor.

Limitations of existing intrusion detection systems

Many of the existing network- and host-based intrusion detection systems perform data collection and analysis centrally using an architecture whereby the data is collected by a single host, either from audit trails or by monitoring network packets, and the data is then analysed by a single machine. Other intrusion detection systems perform distributed data collection (and some processing) by using modules distributed in the hosts that are being monitored, but the collected data is still sent to a central location where it is analysed by a single machine. There are a number of problems with these architectures:

1. *Single point of failure*: The central analyser is a single point of failure. If an intruder manages to prevent it from working (for example, by crushing or slowing down the host where it runs), the whole network is without protection.

2. *Scalability*: Processing all the information at a single host implies a limit on the size of the network that can be monitored. After that limit the central analyser becomes unable to keep up with the flow of information. Distributed data collection can cause problems with excessive data traffic in the network.

3. *Not easy to reconfigure*: It is difficult to reconfigure or add capabilities to the intrusion detection systems. Changes and capabilities are usually done by editing a configuration file, adding an

entry to a table or installing a new module. The intrusion detection system usually has to be restarted to enable the changes to take effect. During the time the central intrusion detection system is being restarted, the entire network is vulnerable to attack.

4. *Analysis of network data can be flawed*: Collection of network data in a host other than the one to which the data is destined can provide the attacker the possibility of manipulating the packets.

The fundamental problems in intrusion detection systems design still remain how to recognise the behaviour associated with intrusions. A determined attacker effects his intrusion through a sequence of activities to achieve a desired result. Generally, each of these actions viewed by itself is a normal legitimate activity. It is only when the sequence of activities of an attack is assembled that the intruder's hostile objectives become clear. Since most intrusion detection systems depend on known attack signatures to do their work, hitherto unknown attacks, anomalies and also highly co-ordinated attacks still pose a challenge to intrusion detection systems design and applications. Furthermore, in an environment in which communications are encrypted, intrusion detection systems are rendered ineffective. Switched networks also make the deployment of network-based intrusion detection systems ineffective by reducing the visibility scope.

Despite all these setbacks, intrusion detection systems technology still remains a good second line of defence to backup perimeter security defences provided by authentication servers, firewalls and virtual private networks.

CopperbeltUniversity
Computer Science Department

Computer Systems Security

Compiled by: Prof.Dr. Hastings M. Libati

Security Vulnerability

The goal of security is to provide confidentiality, integrity and availability of system resources. Therefore, this aims at the preservation of confidentiality(ensuring that information is accessible only to those authorised to have access), integrity(safeguarding the accuracy and completeness of information and processing methods) and availability (ensuring that authorised users have access to information and associated assets when required).

What is a security vulnerability?

A security vulnerability is a flaw in a product that makes it infeasible – even when using the product properly— to prevent an attacker from usurping privileges on the user's system, regulating its operation, compromising data on it, or assuming ungranted trust.

To make this definition clearer, in the text that follows, the critical phrases and words from the definition are discussed so as to show exactly what is meant by each. Furthermore, examples are also given to illustrate how the definition could be applied to real-life situations.

…a flaw in a product…

- *Flaw***:** Security vulnerabilities involve inadvertent weaknesses. By-design weaknesses may sometimes occur in a product, but these may not necessarily be security vulnerabilities.

  Example: The choice to implement a 40-bit cipher in a product would not constitute a security vulnerability, even though the protection it provides would be inadequate for some purposes. In contrast, an implementation error that inadvertently caused a 128-bit cipher to discard half the bits in the key would be a security vulnerability.

- *Product:*Security vulnerabilities are a result of a problem in a product. Problems that result from adhering to imperfect but widely accepted standards are not security vulnerabilities.

  Example: A browser that, when connecting to an FTP site, conducts the session in plaintext would not be considered to have a security vulnerability, since the FTP specification calls for plaintext sessions. However, if the browser conducted Secure Socket Layer (SSL) sessions in plaintext, it would constitute a security vulnerability since the SSL specification calls for encrypted sessions.

...that makes it infeasible… to prevent…

- *Prevention:*Security vulnerabilities involve a loss of control. That is, in order for a flaw to constitute a security vulnerability, it must be possible for an attacker to compel the victim to submit to the attack despite reasonable efforts to avoid it.

  Example: Suppose a flaw in a web browser could be misused by a website to "hang" the browser of any user who visited the site. If the user were able to resume normal operation by stopping the browser, restarting it, and avoiding the attacker's website in the future, the flaw would not constitute a security vulnerability. However, if the flaw enabled the website to force the user to return to it and submit to a new attack each time the browser was restarted, it would constitute a security vulnerability. (Of course, much depends on the specific scenario. If the flaw allowed, for instance, data to be compromised, it would not matter whether the user could be forced to return to the site—stumbling onto the site one time would be sufficient to allow the attacker to exploit the flaw).

...even when using the product properly…

- *Proper usage:*Every product has documentation that describes how it is intended to be used. In addition, best practice guidelines discuss reasonable and customary ways of using products. In order for a flaw to be a security vulnerability, it must occur when the product is used in the expected, required, or reasonable way.

  Example: Suppose a Web server contained a security flaw that was only exposed if all website visitors were given administrative privileges on the server. The flaw would not constitute a security vulnerability, because widely-accepted best practices warn against ever giving website visitors administrative privileges. However, if the flaw were exposed even when visitors to the site were given only the normal and expected privileges, it would constitute a security vulnerability.

...usurping privileges on the user's system…

- *Usurping:*Privilege elevation vulnerabilities involve assuming *unauthorised* capabilities.

  Example: A flaw that allows an administrator to change the permissions on any file on the computer would not be a security vulnerability, because an administrator already has this capability. In contrast, if a flaw allowed an unprivileged user to achieve the same goal as an administrator, it would constitute a security vulnerability.

...regulating its operation…

- *Regulating:*Regulating a system's operation is actually a special case of usurping privileges.

  Examples: A flaw that enables an attacker to cause a server to fail would constitute a security vulnerability, since the attacker would be able to regulate whether the server provided service or not. However, the fact that an attacker could send a huge number of legitimate requests to a server and monopolize its resources would not constitute a security vulnerability, as long as the server operator still could control the computer.

...compromising data on it…

- *Compromising:*The ability to access data contrary to the owner's or the administrator's efforts constitutes a security vulnerability. Depending on the scenario, this could involve reading, adding, or modifying data.

  Example: Suppose an operating system provides file-by-file access control. A flaw that allows one user to read another user's data, regardless of the permissions on the file, would constitute a security vulnerability. On the other hand, if the default permissions on a newly-created file provided global read access, this would not constitute a security vulnerability. Likewise, if the operating system didn't provide file-by-file access control, and this fact was documented, it wouldn't constitute a security vulnerability.

- *On:*There is a subtle but important difference between data *on* a system and data *about* a system. Data *on* a system includes information whose compromise poses a danger as a primary effect. Examples of data *on* a system include user data files, cryptographic key stored thereon, etc. Data *about* a system includes information whose compromise, if it poses a danger at all, poses it as a secondary effect. Examples include information about the logical structure of the system such as network topology, file locations, and so on.

  Example: A flaw in a Web server that enables a visitor to read a file he should not be able to read would constitute a security vulnerability. However, a flaw that revealed the physical location of a file would not constitute a vulnerability—although such a flaw could be useful for reconnaissance purposes, and could be used in conjunction with a another vulnerability to compromise files, it would not by itself enable an attacker to compromise data, and thus it would not constitute a security vulnerability.

...gaining ungranted trust…

- *Ungranted trust:* Many products enable the user to specify people or organizations that they trust, and regulate their actions accordingly. Flaws that enable an attacker to gain a level of trust the user didn't grant constitute security vulnerabilities.

  Example: A flaw that enables a website to engage in an SSL session with a browser in the guise of another, trusted site would be a security vulnerability. However, spoofing based on social engineering—for instance, giving a phony name on Internet Relay Chat (IRC) channel as a means of persuading someone to run Trojan horse software—would not constitute a security vulnerability.

There are many definitions of security vulnerability that one may find. Following is one of the many other definitions. In computer security, a **vulnerability** is a weakness which allows an attacker to reduce a system's information assurance.

It is important to note that **vulnerability** is the intersection of three elements. These elements are the **system susceptibility** or **flaw**, **attacker access to the flaw**, and **attacker capability to exploit the flaw**. To be vulnerable, an attacker must have at least one applicable tool or technique that can connect to a system weakness. In this context, vulnerability is also sometimes referred to as the **attack surface.**

Vulnerability management is the cyclical practice of identifying, classifying, removing, and mitigating vulnerabilities. This practice generally refers to software vulnerabilities in computing systems. *In other words, Vulnerability management*is all about keeping computer infrastructure safe. Regardless of the size

of an organization, if left unchecked, the weaknesses in the organisation's computer infrastructure can become one of the biggest liabilities. This could quickly lead to lost productivity and even lost corporate reputation. The never ending effort to find new vulnerabilities and to fix them is called **computer insecurity**.

A security risk may also be classified as a vulnerability. The usage of vulnerability with the same meaning of risk can lead to confusion. The term risk is tied to the potential of a significant loss. However, there are vulnerabilities without risk, for example, when the affected asset has no value. Avulnerability with one or more known instances of working and fully-implemented attacks is classified as an **exploitable vulnerability** - a vulnerability for which an exploit exists. The **window of vulnerability** is the time from when the security hole was introduced or manifested in deployed software, for example, to the time when access was removed, a security fix was made available or deployed, or the attacker was disabled.

A **security software bug** is a narrower concept as it only affects software. This is so because there are vulnerabilities that are not related to software. Examples of vulnerabilities that are not security software bugs include hardware, site and personnel vulnerabilities. Experience shows that constructs in programming languages that are difficult to use properly could be a large source of system vulnerabilities.

A resource (either physical or logical) may have one or more vulnerabilities that can be exploited by a threat agent (attacker) in a threat action (attack). The result of such a threat action can potentially compromise the confidentiality, integrity or availability of resources (not necessarily the vulnerable one) belonging to an organization. Such a threat action or attack can be **active**, in which case it attempts to alter system resources or affect the operation of such system resources. This means that the attack compromises the integrity or availability of the system resources. A **passive attack**, on the other hand, attempts to learn or make use of information from the system resources but does not affect the system resources. This means that the attack just compromises confidentiality of the system resources.

Classification of Vulnerabilities

Vulnerabilities are classified according to the asset class that they relate to. In this regard, one can identify the following classes:

Hardware vulnerabilities

- susceptibility to humidity
- susceptibility to dust
- susceptibility to soiling
- susceptibility to unprotected storage

Software vulnerabilities

- insufficient testing
- lack of audit trail

Network vulnerabilities

- unprotected communication lines
- insecure network architecture

Personnel vulnerabilities

- inadequate recruiting process
- inadequate security awareness

Site vulnerabilities

- area subject to flood
- unreliable power source

Organizational vulnerabilities

- lack of regular audits
- lack of continuity plans

Causes of Vulnerabilities

A number of causes of vulnerabilities can be identified as given below:

- **Complexity**: Large, complex systems increase the probability of flaws and unintended access points.

- **Familiarity**: Using common, well-known code, software, operating systems, and/or hardware increases the probability that an attacker has or can find the knowledge and tools to exploit a flaw.

- **Connectivity**: More physical connections, privileges, ports, protocols, and services and time each of those are accessible increase vulnerability.

- **Password management flaws**: The computer user uses weak passwords that could be discovered by brute force. The computer user stores the password on the computer where a program can access it. Users re-use passwords between many programs and websites.

- **Fundamental operating system design flaws**: The operating system designer chooses to enforce suboptimal policies on user or program management. For example operating systems with policies such as default permit – grant every program and every user full access to the entire computer. This operating system flaw could easily allow viruses and malware to execute commands on behalf of the administrator.
- **Internet Website Browsing**: Some internet websites may contain harmful spyware or adware that can be installed automatically on the unsuspecting users' computer systems. After visiting those websites, the computer systems become infected and personal information could be collected and passed on to third party individuals.

- **Software bugs**: The programmer leaves an exploitable bug in a software program. The software bug may allow an attacker to misuse an application.

- **Unchecked user input**: The program assumes that all user input is safe. Programs that do not check user input can allow unintended direct execution of commands (non-validated inputs).

- **Not learning from past mistakes**: Once a mistake is discovered, it is important to learn from it and to avoid a repeat of such a mistake in future. An inventory of such mistakes could help their avoidance. New members of staff could also be assisted in such a way.

Experience has shown that the most vulnerable point in most information systems is the human user, operator, designer, or other humans. Therefore, human beings should be considered in their different roles such as assets, threats or information resources. In this respect, it has been found that social engineering is an increasing security concern or trap to which humans are ever falling prey.

Vulnerability consequences

The impact of a security breach can be very high. In some cases, computer personnel are to blame for such breaches. It has been established that in some cases, IT managers may be aware that some IT systems have vulnerabilities but they choose not to rectify the situation in order to manage the known risks. This could actually be classified as misconduct. Laws are required to force managers to act in order to reduce the impact or likelihood of such security risks. **Information technology security audit** is a way to let other independent people certify that an IT environment is managed properly and so lessen the risks. A **Penetration test** is a form of verification of the weakness and countermeasures adopted by an organization. For this purpose, a **white hat hacker** could be hired to try to attack an organization's information technology assets, in order to find out how easy or difficult it is to compromise the IT security. The proper way to professionally manage the IT risk is to adopt an Information Security Management System, such as ISO/IEC 27002. The ISO / IEC is an information security standard developed by the International Standards Organisation (ISO) and the International Electrotechnical Commission (IEC). Standards organizations like these prescribe some set of criteria to be satisfied by a computer, its operating system and applications in order to meet a good security level.

One of the key concepts of information security is the principle of **defence in depth**. That is, to set up a multilayer defence system that can:

- prevent the exploit
- detect and intercept the attack
- find out the threat agents and persecute them

Many systems could be used for the purpose of defence. Intrusion detection systems (IDSs) are an example of a class of systems used to detect attacks. Firewalls are also another line of defence. Physical security is a set of measures to protect physically the information asset. It is believed that if somebody can get physical access to an information resource then it should be quite easy to make such a resource unavailable to its legitimate users.

Vulnerability disclosure

Responsible disclosure of vulnerabilities is a topic of great importance. There are guidelines that have been formulated on how to go about making a vulnerability disclosure. These guidelines constitute what is called a responsible disclosure. In this case the affected vendor is confidentially informed about the vulnerability. Later the Computer Emergency Response Team (CERT), if it exists, is informed. A full disclosure is then effected when all the details of the vulnerability are available. Usually, vulnerability information is discussed on a mailing list or published on a security web site and results in a security advisory afterward.

The **time of disclosure** is the first date that asecurity vulnerability is described on a channel where the disclosed information on the vulnerability has to fulfil the following requirements:

- The information is freely available to the public.
- The vulnerability information is published by a trusted and independent channel or source.
- The vulnerability has undergone analysis by experts such that risk rating information is included upon disclosure.

It is important to ensure that a vulnerability inventory is taken. This entails maintaining a list of disclosed vulnerabilities in a system called Common Vulnerabilities and Exposures. Here, vulnerabilities are classified (scored) using Common Vulnerability Scoring System (CVSS).

Vulnerability assessment

A vulnerability assessment is the process of identifying, quantifying, and prioritizing (or ranking) the vulnerabilities in a system. Vulnerability assessment has many factors in common with risk assessment. Assessments are typically performed according to the following steps:

1. Cataloguing assets and capabilities (resources) in a system.
2. Assigning quantifiable value (or at least rank order) and importance to those resources
3. Identifying the vulnerabilities or potential threats to each resource
4. Mitigating or eliminating the most serious vulnerabilities for the most valuable resources

Vulnerability analysis, on the other hand, focuses both on the consequences for the object itself and on primary and secondary consequences for the surrounding environment. It also concerns itself with the possibilities of reducing such consequences and of improving the capacity to manage future incidents.

Consequently, vulnerability analysis strives to reduce the security risks posed by software vulnerabilities, by trying to address both the number of vulnerabilities in software that is being developed and the number of vulnerabilities in software that is already deployed. Vulnerability analysis work therefore deals with identifying and reducing the number of new vulnerabilities before software is deployed (vulnerability discovery), and also vulnerability remediation work that deals with existing vulnerabilities in deployed software.

With vulnerability discovery, the idea is to enable software developers to understand how vulnerabilities are created and discovered. The hope is that with this education, software developers will learn how to detect and eliminate vulnerabilities and then eventually avoid vulnerabilities in software products before the products are offered to users. The concept of security programming is very important at this stage. However, the reality is that many software products today are still being rolled out with vulnerabilities that attackers may be able to exploit.

Identifying and removing vulnerabilities

Many software tools exist that can aid in the discovery (and sometimes removal) of vulnerabilities in a computer system. Though these tools can provide an information technology auditor with a good overview of possible vulnerabilities present, they cannot replace human judgment. So relying solely on scanners may yield false positives and a limited-scope view of the problems present in a system.

Vulnerabilities have been found in every major operating system. One major way to reduce the chance of a vulnerability being used against a system is through constant vigilance, including careful system

maintenance (e.g. applying software patches), best practices in deployment (e.g. the use of firewalls, intrusion detection systems and access controls) and auditing (both during development and throughout the product deployment lifecycle).

Examples of tools that can help in network management include Wireshark, Tcpdump,Snort and SATAN (Security Administrator Tool for Analyzing Networks), among others.

CopperbeltUniversity
Computer Science Department

Computer Systems Security

Compiled by: Prof.Dr. Hastings M. Libati

Some Common Terminologies Used in Network Security

In this text we shall endeavour to discuss some of the common terminologies that are used in every day speak in the area of network security.

Password protection. Password protection is the technique of restricting access to data files, programs and even terminals by the use of passwords. A password is a secret code that is formed by combining letters, numbers or special symbols. A password-protected program, for example, will require a valid password to be entered before it can be run. If an invalid password is entered, the program will not allow access and cannot be run. In this way, control can be exercised as to who should be allowed to access a data file, program, terminal or even a network. Users choose their own passwords. Since a password is a secret code, it should be known only to one particular user. If user A's password is exposed to other users, then these other users may use the password for user A to do some damage to the system. A follow up will show that userAdid the damage, when actually the damage was done by other users who know user A's password. Each user is responsible for keeping his or her password a real secret. It is advisable to change the password frequently. However, forgetting a password is not advisable.

Bank customers also use Personal Identification Numbers (PIN) to have access to their bank accounts through Automatic Teller Machines (ATMs). The PIN is also a way of ensuring that only the owner of an account can have access to that account. Only the user should know the PIN code.

Computer hacker. Computer hackers are people who are mainly self-taught computer specialists or hobbyists. The general understanding is that hackers are people who carry out unauthorised computer activities, like trying to access computer networks without authority. On the other hand, **hacking** is when an unauthorised user gains access to a computer system or files to which the unauthorised user is not entitled to have access. People who try to gain entry to computer systems but are unsuccessful can also be classified as carrying out hacking activities. Though, generally we refer to hackers as people who break into computer systems and steal or corrupt data and cause other malicious damage to computer systems and users, this is not quite true about hackers. As stated earlier, computer hackers are just computer enthusiasts, that is, people who enjoy learning programming languages and computer systems. We can also go ahead to say that computer hackers are people who gain unauthorised access to computer systems or networks, often just for the challenge of it. Remember that computer hackers are mainly self-taught computer specialists or hobbyists who enjoy pursuing the question: *what if ……..?* As a result, a hacker can even be called to assist in solving problems that organisations face with their computer systems. We can therefore, say that hackers do more or less positive work (also referred to as white-hat hackers or ethical hackers). They hack to gain knowledge and we can even utilise that knowledge positively. On the other hand, if a hacker utilises his knowledge for malicious activities then he is no longer a hacker but a **cracker (**unethical hacker**)**. Such malicious activities could include breaking into computer systems to obtain information for financial gains, shut down computer systems and so causing denial of service attacks, pirate software, steal people's credit card information, or even alter or destroy data.

Based on the foregoing discussion, we can identify two types of hackers and four types of crackers. The two types of hackers are the thrill-seeker hackers and the white-hat hackers. Thrill-seeker hackers are those hackers who gain unauthorised access to computer systems simply for the challenge of it. However, after gaining this illegal access, the thrill-seeker hackers do not do any damage or steal anything from the computer system that they have broken into. They are content with the fact that they have proved a point – they have managed to break into the computer system without permission. This illegal act is a very big achievement for them. On the other hand, white-hat hackers are usually computer professionals who break into computer systems and networks with the full knowledge of the owners of such computer systems and networks. The whole idea here is to test if the system is robust enough to detect break-ins or to deter intended break-ins. This is what is called security vulnerability testing. Any security vulnerabilities that are exposed during the testing can then be fixed.

The four types of crackers are the script kiddies, hacktivists, black-hat hackers and cyber terrorists. **Script kiddies** are also called script bunnies and are mostly teenagers without much technical expertise in computing. Script kiddies mostly use downloadable software or source code to perform malicious break-ins. It should be noted that there are websites on the Internet where malicious software can be downloaded. In most cases, documentation for such software is also available. Script kiddies then use such software for their various malicious activities. Script kiddies are potential hackers and crackers. **Hacktivists** are hacker activists that is, people who break into computer systems for a political or socially motivated reason. A hacktivist can break into a website and post a message that expresses a viewpoint that is contrary to the one held by the owners of the website that has been broken into. Unlike white-hat hackers who are professionals and hack for a good cause, **black-hat hackers** are crackers who are also mostly professionals but hack for malicious purposes. Black-hat hackers are an example of crackers who break into computer systems and networks so as to enrich themselves. They steal or even destroy information or sell the information stolen from one company to its competitors so as to gain some money in their pockets. Black-hat hackers can be blamed for the increase in cyberattacks on corporate networks. Another type of cracker is the **cyberterrorist**. Cyberterrorism can be likened to the physical terrorism that is experienced in the real world. The only difference is that cyberterrorism takes place in the cyberworld. Consequently, we can say that cyber terrorists are politically motivated people that use the cyberworld to attack computer systems and computer networks so as to bring physical or financial harm to a lot of people or destroy a lot of information belonging to another group of people on whom they would like to exercise their terror for political reasons. Among the commonly affected installations attacked by cyber terrorists are power plants, water systems, traffic control centres, and military installations.

Example: In Zambia, we have seen some motorists lock their car keys inside their cars unintentionally while they are outside their cars. The result is that they can no longer get into their cars again after such an incident. In such cases, we have witnessed people being called to come and hack the car mostly through the car windows and then the car doors are opened. When the car doors open, the car owner can then enter his car and have access to the keys, which were locked inside the car. The car owner in this case used experts in car break-ins. But, we also see that these experts were used for a good purpose. So we can liken these experts to hackers. On the other hand, if these experts use their knowledge to illegally open people's cars and steal items from the cars or steal the cars themselves, then they now turn into crackers. We could use this analogy to strengthen our understanding of the difference between hackers and crackers.

Phishing. In computer security, *phishing* is the criminal and fraudulent process of attempting to acquire sensitive information such as usernames, passwords and credit card details of other people by masquerading as a trustworthy entity in an electronic communication. Communications purporting to be from popular social web sites, or information technology (IT) administrators, etc., are most often used to lure the unsuspecting victims. Most phishing incidents are carried out by using e-mail or instant

messaging systems. Typically, such phishing incidents entice the unsuspecting users to enter details of their usernames, passwords etc. Such user details will then be captured by the perpetrators of such vices and then used for undesirable activities. When users are lured to some fake websites, it is normally observed that the fake websites have the look and feel of the legitimate websites being imitated and so making the victims not to suspect any foul play. Phishing is an example of social engineering techniques used to entice users to disclose confidential information and also to exploit known vulnerabilities of the World Wide Web or operating systems.

When phishers target their phishing activities towards a particular organisation then this type of phishing is called **spear phishing**. When such phishing activities are targeted specifically at senior executives of a particular organisation or company then this is called **whaling** (targeting a big fish). Phone phishing also called **vishing**(voice phishing) is a phishing method where the perpetrators use phones to lure their victims into giving away their confidential data.

Wi-Fi phishing.Wi-Fi phishing is a method that attackers (hackers or crackers) use to steal data in wireless networks such as Wi-Fi networks. This attack, which is also called Evil Twin attack can be said to be a variation of the conventional phishing attack. In Wi-Fi phishing, an attacker sets up a Wi-Fi hot spot or access point to trap users and steal the information that they communicate over the bogus network. An unsuspecting user will see that his computer (laptop) has sensed a hot spot and unsuspectingly the user then uses this hot spot to access services like his bank, home or company network. On the other hand, the attacker is busy monitoring all the data being communicated over this network and is free to steal such data that does not have the right security measures, for instance, not encrypted.

Spyware. Spyware is a type of malware that is installed on computers that are normally connected to a network, usually, the Internet. In most cases, spyware is secretly installed on the unsuspecting user's personal computer. Spyware then collects pieces of information about the user of the computer that spyware is installed on and also the network to which that computer is connected. All this occurs without the knowledge of the affected users. It might be quite difficult to detect the presence of spyware.

From the term *spyware* one can see that the software actually secretly monitors a user's computing pattern and activities. In other words we can say that this is software that spies. Unfortunately, the functions of spyware may go beyond simple monitoring of the activities of a user. Spyware may be able to collect various types of personal information about a user. Such information may include the Internet surfing habits and also a history of the web sites that such a user has visited. Spyware can also interfere with the user's control of his / her computer in many ways. For instance, spayware can install additional software on the user's computer, which will be redirecting web browser activities like visiting websites that the user did not type or request for. It is also possible for Spyware to change the user's computer settings, resulting in the user's computer having different home pages that the user did not set, slow connection speeds (slow transmission rates) and intermittent loss of Internet facilities. The user may also realise that some of the programs that were installed on the computer no longer function, yet they used to function properly before.

The term adware is normally taken to refer to any software, which displays advertisements on a computer connected to the Internet. However, most adware is also spyware because it displays advertisements that are as a result of what the spyware spied on the users. Consequently, the user may receive a myriad of pop-up and pop-under advertisements. However, unlike viruses and worms, spyware does not usually self-replicate; instead, spyware gets on a system through deception of the user or through exploitation of software vulnerabilities on the target computer. Such vulnerable software could be the Web browser or the operating system of the user's computer. It is very possible for some spyware programs to masquerade as security software and so tricking the unsuspecting users into installing them on their computers.

Malware. The term malwareistheshort form for malicious software and is a generic term used to mean a variety of forms of hostile, intrusive, or annoying software or program code. Malware is any software that has been designed to gain access to computer systems without the owners' permission. To this effect, software is considered malware based on the perceived intent of the creator of the software rather than any particular features. As a result, malware includes computer viruses, worms, trojan horses, most rootkits, spyware, dishonest adware, crimeware and other malicious and unwanted software.

Grayware. Also spellt as **greyware,** grayware is a generic term that is sometimes used to classify applications that behave in a manner that is annoying or undesirable to users. However, grayware is generally considered to be less serious or less troublesome as compared to malware. Consequently, grayware includes such programs as spyware, adware, joke programs, remote access tools, and any other unwelcome files and programs. These programs negatively affect the performance of the computers on a network and may introduce significant security risks to an organization. Grayware may perform a variety of undesired actions such as irritating users with pop-up or pop-under windows and tracking the habits of users on the Internet. Some grayware may even unnecessarily expose the vulnerabilities of operating systems, web browsers and other programs running on the target computer or network. Once exposed, such vulnerabilities may then be exploited through malicious attacks to the detriment of the user.It should be noted, however, that grayware does not include viruses or trojan horse programs because these are designed to harm the performance of users' networked computers and therefore, are directly harmful.

Pharming. The term **pharming** is a general word used to describe the practice in which malicious code is installed on a computer maliciously and such malicious programs then begin misdirecting users' web requests to fraudulent web sites without their knowledge or consent. Sometimes pharming is referred to as *phishing without a lure*.

In pharming attacks, a large number of computers that are connected to the Internet can be victims because in this type of attacks it is not necessary to target individual computers or users one by one and also no conscious action is required on the part of the users, unlike in phishing attacks. For example, a program code could be sent in an e-mail message to modify local host files on a connected computer. A local host file is a file on a computer connected to the Internet that helps convert Uniform Resource Locators (URLs) into the Internet Protocol (IP) addresses that the computer uses to access web sites. Once a computer's host file is compromised, one possible consequence is that if you click on a web site link that you well know, your request will possibly be directed to a different website. If you click on a bookmark that has been affected by the pharming attack, again your request may be directed to a different website where the attacker has laid a trap. Such traps could be downloading malicious programs to your computer. Such illegally downloaded programs could then be used to collect credit card numbers, bank account numbers or even passwords that can be used by the attackers to impersonate the victim.

Another type of pharming attack is called Domain Name System (DNS) poisoning. In this type of attack, the attackers modify the domain name system table in a DNS server so that users who think they are accessing legitimate websites using their computers end up finding that their requests are actually directed to fraudulent websites. This redirection is done at the DNS server side on the domain of the users. In this method of pharming attacks, the individual host files on the connected computers may not be corrupted at all. As it is seen clearly, the problem occurs in the DNS server. For a large and busy network, the DNS server handles quite a lot of user requests for URLs. As is the case in such attacks, the unsuspecting users will end up at websites that they did not request for. Again personal information may be stolen from such unsuspecting victims, which the attackers can then use maliciously.

The Domain Name System (DNS) is the software that translates computer names into equivalent Internet Protocol addresses on the Internet. An example of a computer name is www.mazusa.com. This name cannot be used to send data on the Internet and it requires to be translated into the equivalent Internet

Protocol (IP) address before data can be sent. The software that does this translation is the Domain Name System (DNS). Computer names were invented to make it easy for human beings to use the Internet. So, since computer names were invented for human convenience, computers do not understand them. That is why computer names must be translated into equivalent Internet Protocol (IP) addresses, which computers can understand. This translation is the task of the Domain Name System (DNS).

Crimeware. This is a class of malware whose design intent is specifically to automate cybercrime. In other words, crimeware is any computer program or set of programs designed to facilitate illegal activities online. Many spyware programs such as browser hijackers, and keyloggers can be considered crimeware, although only those that are used illicitly. One common type of crimeware is the phishing kit. A phishing kit is a collection of tools that are assembled in such a way as to make it easier for people with little technical skill to launch a phishing attack. A phishing kit normally includes such development software that can be used even by a novice to commit cybercrime. Phishing kits and other types of crimeware are, unfortunately, readily available on the Internet for anyone to download and use.

Crimeware, unlike spyware, adware, and malware is designed through social engineering or technical stealth so as to perpetrate identity theft or to carry out unauthorized transactions that benefit the person who is controlling the crimeware. Crimeware is one of the growing problems in network security because many malicious programs are designed to break into systems so as to steal confidential information from users' computers.

To steal confidential data through crimeware, attackers use a variety of techniques. For example, crimeware can be used to secretly install keystroke loggers to collect sensitive data such as login information and then report these back to the perpetrator of the crime. In some cases, acrimeware program can be used to redirect a user's web browser to a fake website controlled by the attacker even though the user typed in the correct website address. Using crimeware, attackers can have access to remote applications and thereby enabling the attackers to break into remote networks for malicious purposes.

Crimeware programs can be installed on a user's computer through a variety of ways. For example, cyber criminals could use vulnerabilities in the user's applications such as web applications or the operating system. In some cases, a user could be socially engineered using an e-mail that appears to be very valid. Such an e-mail may have an attachment and could purport to have come from a particular company or person that may be well known to the unsuspecting user. The malicious e-mail could now use social engineering techniques to manipulate the user to open the attachment and execute the payload. When this attachment is opened the crimeware now loads and configures itself on the user's computer. Once this software is configured on the user's computer, it will be used to gather information illegally through a secretly installed keylogger. The keylogger will then record everything that is entered at the keyboard, including passwords and other privileged information. Periodically, an associated trojan horse program also installed on the user's computer without his / her knowledge will then start sending this collected information to the crimeware originator.

Keystroke logging. **Keystroke logging** (also known as **keylogging**) is the practice of tracking (or logging) the keys struck on a keyboard by a user. This is done so secretly that even the person who is using the keyboard is unaware that his / her actions are being monitored. Several methods can be used for keylogging and these include hardware and software-based keyloggers, among others.

Remote access software keyloggersare local software keyloggers programmed with an added feature to enable transmission of recorded data from the target computer to a monitor at a remote location. Periodical e-mailing of the maliciously collected data to a pre-defined e-mail address can easily facilitate such remote communication. The collected data can also be wirelessly transmitted to the monitor by

means of an attached hardware system. However, it is also possible for the remote monitor computer to log into the target user's computer via the Internet and access the files stored on the target computer.

Some software keyloggers may not necessarily require the keyboard keys to be pressed for them to do their job. Such software keyloggers may record anything that has been copied to the clipboard, instead. Other applications use screen logging. Applications with screen logging abilities may take periodic screenshots of the whole screen or just part of the screen, especially the screen area around the cursor.Hardware-based keyloggers, on the other hand, do not depend upon any software being installed on the target computer as they exist at a hardware level in a computer system. In this case, the cybercriminal may have to physically find his way to the target computer so as to insert the hardware component that will assist him to accomplish his malicious activities.

Stories are abound where criminals have placed their keypads exactly over the keyboard of the ATM machine. These keypads are designed to look like an integral part of the ATM machine and are so deceiving to the users. These are what are called keyboard overlays. In this way, each keypress of the unsuspecting user of the ATM keyboard will be registered by the criminal's keypad as well as accomplishing the user's goal. In the end the criminals succeed in capturing the PIN codes of innocent ATM users which will now be used for criminal purposes.

Acoustic keyloggers are used by criminals to monitor the sound created by someone typing on a computer's keyboard. This feature utilises the fact that each character on the keyboard makes a totally different acoustic signature when pressed. Consequently, it is possible to identify which keystroke signature relates to which keyboard character using statistical methods such as frequency analysis. Some criminals use electromagnetic emissions as well. This makes it possible to capture the electromagnetic emissions of a wired keyboard from up to about 20 metres away, without being physically wired to that keyboard.

Optical surveillance can also be used. Though this method is really not a keylogger as such, it can nonetheless be used to capture passwords or PIN codes. In this case, cameras are placed strategically in such a way that they are hidden at an ATM and cannot be seen. This can easily enable a criminal to watch a PIN or password being entered.

Social Engineering. The term **social engineering** refers to the act of manipulating people into performing actions or divulging confidential information which they would not have divulged in normal circumstances. It is therefore, a technique of acquiring confidential information without breaking into something or using technical hacking techniques. Social engineering typically applies to trickery or deception for the purpose of information gathering, fraud, or computer system access. It sometimes happens that the attacker can get all the information he requires through social engineering without even coming face-to-face with the victim. Social engineering techniques are based on specific attributes of the human decision-making process. This can be exploited in various combinations to create attack techniques by criminals. Several techniques can be used to implement social engineering.

One of the attack techniques for social engineering is pretexting. Pretexting is the act of creating and using an invented scenario (the pretext) so as to engage a targeted victim in a way that increases the chance that the victim will release information or perform actions that would be unlikely in ordinary circumstances. Pretexting normally requires that the attacker does his share of research and then puts up such a lie that is easy to believe and so enabling the attacker to establish some form of legitimacy in the mind of the targeted user. The technique can be used to trick a business into disclosing customer information, for example.

One other technique of social engineering is called baiting. In this case a bait that relies on the curiosity or greed of the victim is used. For example, the attacker could leave a malware infected floppy disk, CD ROM, or flash disk in a location where he is sure that someone will find it and so the attacker simply waits for the victim to use the device. For example, an attacker could have a flash disk which has a logo of a particular company on the cover. On the disk, there could also be some words like "*The revised salaries for this year*". The attacker could leave such a disk in the corridor, elevator or lobby of the targeted company. An unsuspecting employee of the company would pick it up to satisfy his curiosity and then insert it in his computer that could be connected to the company network. Without knowing, the employee has installed malware on his computer. This could definitely give the attacker access to the victim's computer and perhaps also to the targeted company's internal computer network (intranet). This could prove disastrous for the targeted company.

Rootkit. A rootkit is a software program or coordinated set of programs designed to gain control over a computer system or network of computing systems without being detected. The purpose of all this is almost always malicious. The term *rootkit* is a compound word derived from the superuser's (root's) toolkit. In Unix systems, the network administrator's account is referred to as the root. The rootkit therefore, was supposed to be the kit that the root uses to do his job. In most cases, one part of the rootkit may initiate the actual entry into the target computer system and another part of the rootkit ensures that the host computer's operating system is modified to enable easy entry at a later stage (create a backdoor channel for the attacker). This also usually includes the erasure of the system event logging capacity of an operating system in an attempt to hide evidence of any perpetrated attacks. This will enable the attacker to access the compromised system easily and unnoticed thereafter. Successful attacks may have the capacity to siphon and transmit sensitive data, such as PIN codes, passwords or even credit card particulars from the compromised system to the attacker's system. Rootkits are also known to enable attackers to gain access to computer systems while evading detection.

Most rootkits typically hide files, processes, network connections and blocks of memory from other programs used by system administrators to detect specially privileged accesses to computer system resources. Many other utility tools used for abuse can be hidden using rootkits. These include tools for further attacks against computer systems with which the compromised system communicates, such as sniffers and keyloggers. A possible form of abuse is using a compromised computer as the ground for staging further attacks. Rootkits may be used for both productive and destructive purposes.

Event logging is the process of recording all the events or some prescribed events that occur at a given computer or network. Such events are recorded on storage media like the hard disk. Such logged events can enable the administrator to know, for example that MrHaamwiibaHaakintuntu logged onto computer H on network T at 12:00 hours. He further opened file X and deleted record B from file X at 12:17 hours. Therefore, when an attacker gains illegal access to a computer, his first interest is to delete the file in which the events taking place at that computer are logged (recorded). When the event logging system is disturbed then the attacker's actions may not be easily followed and therefore there may not be any evidence that the attacker illegally accessed a particularcomputersystem event logs are a very good tool for litigation cases.

Joke Programs. Joke programs mostly intend to entertain or frustrate their recipients. Joke programs inflict no damage to their target systems. Joke programs do not contain malicious codes or viruses. However, they often trick users into believing that their files will be deleted because of an automated reformatting process of the hard disk, for example. A dialog box will normally display a countdown on the screen and thereby preventing the affected user from clicking the cancel button. When the countdown ends the joke program will normally reveal the joke. The joke program might even go ahead to ridicule the affected user. Though joke programs may not necessarily be considered a threat as such; they often

place fear and panic into a user after reading the messages written on the screen and imagining the consequences of having a formatted disk, for example. Such panic situations could cause the users to inflict damage to their systems by improperly shutting down their systems to prevent their hard disks from being formatted, for instance. The risk here is that in this panic the user might end up damaging important files or causing other harm to computer systems.

Spam. Spam is the term used to describe the flooding of the Internet with many copies of the same message; the aim of which is to force the message on unsuspecting network users who would otherwise not have willingly decided to receive it. With spam, innocent users are overwhelmed with a bombardment of advertising or other irrelevant messages. Most spam messages are commercial advertisements of questionable products. For instance, some spam advertising messages contain such information as "*obtain a PhD. degree online in 6 months*". E-mail spam, also known as junk e-mail, is a type of spam whereby spammers send nearly identical messages to numerous recipients by e-mail.

Spam is costly. Most of the costs for spam messages are paid for by the recipient or the carriers rather than by the sender and so the sender achieves the joy of making others receive disturbing and unsolicited for messages and also even pay for the unwanted messages. Some Internet users use the telephone service to access their e-mails and therefore, they pay for the amount of time they spend to access and send, receive or read their e-mails. Any e-mail spam that they receive will definitely cost them extra money. The Internet Service Providers (ISPs) who provide the Internet services incur costs for transmitting spam. It is just natural that the ISPs also transmit such costs directly to subscribers of their services. The end result is that the poor users are the losers.

Pop-up advertisements. Pop-up advertisementsalso known as **Pop-up ads** or **pop-ups** are a commonly seen form of online advertising on the World Wide Web. Pop-up advertisements are intended to provide the user with some information as per the motive of the originator. They are also intended to attract web traffic or even capture email addresses from unsuspecting users. Pop-up advertisements are displayed in pop-up windows. For example, a user may have opened a web site page and could be busy reading the contents of that particular page when suddenly another window (usually small and superimposed on the active window) opens up to provide some information to the user or to request for information from the user. The user could be requested to enter his e-mail address or password, etc. In most cases, the pop-up window may have nothing to do with what the user is currently interested in, thus causing some irritation in the user.

Pop-under advertisements.Pop-under advertisementsalso known as **pop-under ads** or **pop-unders** are similar in nature and function to pop-up ads. However, pop-unders open a new browser window that is hidden under the active window whereas pop-up ads open a new window that is superimposed in front or on top of the active one. Pop-under ads do not immediately disturb the user's attention as he reads or views the contents of the current active page or window as they remain unnoticed under the active window until the active browser window is closed. This allows the user to focus his attention on his desired goal free of the advertisements. Consequently, pop-under ads do not interrupt the user's attention immediately and are therefore less of a disturbance than pop-up ads.

Zombie computers. Networks like the Internet are full of people working in the *underground* and determined to compromise computers and operations of organisations and people that use the Internet. Such people are collectively referred to as belonging to the *underground community*. The underground community is actively working around the clock to find computers that they can compromise. Consequently, a zombie (also called a drone) is a computer that has been taken over stealthily and has been fraudulently programmed to respond to instructions sent remotely from an attacker in the underground community. In most cases, attacks sent remotely to zombie computers utilise the instant-messaging channels. Attackers that take over computers in this manner may actually compromise several

computers scattered around the Internet. Collectively, this set of compromised computers is known as a *robot network* or simply *botnet* for short. Computers in a botnet, are usually compromised by means of a trojan horse that plants instructions within each computer to wait for commands from the person that has taken over those computers and is hence controlling the robot network. Once in full control, the attacker can now use the zombie computers and the botnet to launch malicious attacks such as phishing attacks, denial of service attacks or even send spam messages, among others.

Brute force technique. Brute force technique is a technique that is used for solving a complex problem by taking advantage of the computer's fast processing capability to repeat a simple procedure many times. Brute force technique is the technique that is used by spell-checkers in word processing programs, for example. A spell-checker in a word processor doesn't really check the spelling of words, but instead, it compares all the words that a user types from the keyboard into a document with a dictionary of correctly spelt words that is stored by the word processing program. The same goes for complex games like the game of chess. A chess-playing computer program calculates all the possible moves that can apply to a given situation and then chooses the best move. Consequently, chess-playing computer programs never analyse or strategise in the manner that we human beings do when we play chess. Attackers that guess user passwords also use the brute force technique to try out several if not all the possible combinations of characters to see if the computer will accept any one of them as the correct password.

Trap doors. A trap door is a secret entry point into a program that allows someone that is aware of the trap door to gain access without going through the usual security access procedures. Trap doors have been used legitimately for many years by programmers to debug and test programs. Trap doors become threats when they are used by unscrupulous programmers to gain unauthorized access. It is difficult to implement operating system controls for trap doors.

Viruses. The computer virus is a computer program, which invades a computer system by attaching itself to other programs also called host programs (since in this way they host the virus). When the host program is loaded into the computer's main memory for execution, the virus permanently installs itself on the invaded computer system. Once it has installed itself, the virus can display unwanted and disturbing messages, erase data or programs, or even promote activities that can be quite destructive. Viruses and other destructive computer programs have many different modes of operations.

There are many different types of viruses, which include boot-sector viruses, file viruses, multipartite viruses and macro viruses. **Boot-sector viruses**. The boot sector is that part of the system software that contains most of the instructions for booting (that is powering up and starting the computer system). Before the computer can start operating, the systems programs (part of the operating system) on the boot sector are first loaded into the main memory. This is the boot process. Boot sector viruses have the tendency of corrupting these boot sector programs that are used for booting. Once the boot sector programs are infected and the system is turned on, the virus is loaded into the main memory before the operating system (boot sector programs). From this point on, the virus can have a field day and is in position to perform its destructive acts. Any storage medium that is used in the drives of the computer may definitely become infected and when such storage media are used on other computers, the infections just spread. In some cases, the boot sector virus simply deletes the boot sector programs and the computer system simply fails to start up. **File viruses**. File viruses are those viruses that attach themselves to executable files (***.EXE**), for example. Executable files are those files that if their names are clicked on twice then the program saved in such files begins execution. In the Microsoft Disk Operating System (MS-DOS), these files have the extensions .com and .exe. When such an executable file (program) is run, the virus that attached itself to it starts working. Since a program can only be executed when it is loaded into the main memory, the attached virus is also automatically loaded into the main memory. When it gets into the main memory the virus then begins its destructive phase. **Multipartite virus**. Multipartite viruses are actually a hybrid of the file and boot-sector viruses. Consequently, the multipartite viruses infect both

files and boot sectors. This enables multipartite viruses to spread easily and quickly but also makes it more difficult for such viruses to be detected. There is a type of multipartite virus called the polymorphic virus, which is capable of mutating and changing its form and thereby changing its profile. This makes it difficult even for antiviral programs to detect and remove such viruses. **Macro virus**. These are the types of viruses that take advantage of a procedure in which miniature programs, which are known as macros, are embedded inside common data files, such as those created by e-mail or spreadsheets and are sent over computer networks.

Worms. Unlike computer viruses, worms are computer programs that are self-contained, that is, worms do not need to attach themselves to other programs in order for them to move from one computer to another. Once a worm gets into the memory of a computer it will keep on duplicating itself until it has filled all the space. As a result, the whole processing activities may come to a halt. This self-duplicating nature can enable worms to invade an entire network. It is important to avoid foreign portable storage media being used on your computer. If foreign portable storage media (diskettes, CD-ROMs, flash disks, etc.) must be used, then they should undergo extensive scanning with the help of appropriate anti-virus software before they can be used.

Bombs. Bombs come in two types, *time bombs* and *logic bombs*. Like trojan horses, bombs hide in usual files and do not duplicate themselves. Time bombs are preset such that they will only start their destructive phase on a particular date and time. Before the preset date and time, the time bombs will remain dormant. When activated, they may delete vital files in a system and so cause destruction to processing activities. On the other hand, logic bombs remain dormant until a particular logic event occurs. For instance, the number of times (e.g., 100 times) your computer is cold booted. When activated, the logic bomb may open particular data files, look for particular records and then edit them or even delete them and they can do many more harmful activities.

Trojan horses. A trojan horse is not self-replicating. Normally, a trojan horse will hide or disguise itself as something else. It could disguise itself as part of a game or graphics demonstration program. When such a demonstration program is run, the user will be busy concentrating on the demo program, while the trojan horse is also busy deleting some files from the hard disk or carrying out other irritating activities. Since a trojan horse hides in other programs without replicating itself, deleting the hideout also clears off the trojan horse.

Bacteria. Bacteria are programs that do not explicitly damage any files. Their sole purpose is to replicate themselves. Bacteria reproduce exponentially, eventually taking up all the processor capacity, main memory, or disk space.

HTTP cookie. A cookie is a small piece of text that is stored on a user's computer by a web browser. A cookie consists of one or more name-value pairs containing bits and tits of information. The server can set a cookie by using the following command: Set-Cookie: BROWID = 89543278dreg7689. In this case, BROWID is the name of the cookie and the string 89543278dreg7689 is the value of the cookie. Cookies are also known by various other names such as tracking cookies, browser cookies, and also HyperText Transfer Protocol (HTTP) cookies.

When a user visits a particular website, a cookie containing the unique session identifier for the session may be sent as part of the HTTP header by the web server that is being visited to the user’s web browser and then sent back unchanged by the user’s browser each time that user accesses that web server. In this way, a cookie can be used for authentication, session tracking (keeping the state of a session), storing the preferences of a particular site, shopping basket contents or identifying a particular server-based session. A shopping basket is a virtual device into which a user can store items he wants to purchase as he

navigates a particular website on the Internet's World Wide Web. One can say that the cookie can be used for any purpose that can be accomplished through storing textual data.

Since cookies are mere text, they are not executable. Consequently, they may not pose much danger and therefore do not qualify as spyware or viruses. However, cookies may be used to track users on the Internet and in this way they can violate user privacy.

Cookies come in different varieties. *Tracking cookies* may be used to track Internet users' web browsing habits. Consider the following example. If a user requests for a web page of a particular website, but the request contains no cookie then the server will presume that the requested page is the first page visited by that user on that website. This is so because the browser's request came without a cookie. Consequently, the server on the visited website will create a random text and send this so created text as a cookie back to the requesting user's browser together with the user's requested web page. After this, the cookie will be automatically sent by the user's browser to that particular server each time the user requests for a new page from the same site. In response, the server will send the requested page as usual. However, the server will also store the Uniform Resource Locator (URL) of the requested page. Furthermore, the server will also store the date, the time of the request and the cookie in a log file. As can be seen here, a *tracking cookie* can potentially be used to infringe upon a user's privacy. This is so because if the log file at the server is analysed, a great deal of detail about the user's browsing habits may be clearly seen. Yet, how the user browses and what the user browses is really none of anybody's business but the user.

When viewing a web page, some objects contained within that page may reside on other servers elsewhere connected through the World Wide Web (WWW). A user can easily click on any of the objects to request the server to load these objects from these other servers. While the information requested for from these other servers is being retrieved, some of these other servers may set cookies in the user's browser. *First-party cookies* are cookies that are set by the same server that is shown on the address bar of the user's browser. *Third-party cookies* are cookies being set by one or more of the servers that the user downloads data from while he / she is still logged on to the current server that is showing on the address bar of the user's browser. When a cookie is being set, the cookie setter can specify a deletion date, that is, the date when the cookie will expire and be deleted. If the cookie setter does not specify a deletion date then the cookie will be deleted immediately the user quits his / her browser at the end of the browsing session. Therefore, specifying a date is one way of making a cookie survive across sessions. Consequently, cookies that bear an expiration date are also called *persistent cookies*.

Watering Hole Exploit or Attack. The term *watering hole* refers to initiating an attack against targeted businesses and organizations. In a wateringhole attack scenario, threat actors compromise a carefully selected website by inserting an exploit resulting in malware infection. A watering hole attack is typically executed in such a way that first, attackers gather strategic information that they can use to gain entry into their targeted organization. This is the reconnaissance (information gathering) step. The information gathered may include insights on trusted websites often visited by employees or members of their targeted entity. After gathering information, attackers may insert an exploit into the selected sites. When the targeted victims visit the compromised site, the exploit takes advantage of software vulnerabilities, either old or new, to drop malware. The dropped malware may be in the form of a remote access Trojan (RAT), which allows attackers to access sensitive data and take control of the vulnerable system. Attackers incorporate strategies to circumvent the targeted organization's defences in order for watering hole attacks to be effective. These may come in the form of outdated systems or simply human error.In watering hole attacks, the goal is not to serve malware to as many systems as

possible. Instead, the attackers run exploits on well-known and trusted sites likely to be visited by their targeted victims. This makes the watering hole technique effective in delivering its intended payload. Aside from carefully choosing sites to compromise, watering hole attacks are known to incorporate zero-day exploits that target unpatched vulnerabilities. Thus, the targeted entities are left with little or no defences against these exploits.

This doesn't mean that attackers don't target patched system vulnerabilities. Because of patch management difficulties in an enterprise setting, ICT administrators may delay deploying critical updates. This window of exposure may lead to attacks.The stolen information, in turn, may be used to initiate more damaging attacks against the affected organization.Timely software updating can be a remedy for avoiding watering hole attacks. Users are simply advised to utilise the latest software patches available. Monitoring for suspicious traffic could also help prevent watering hole exploits. This is what is also termed as *virtual patching* or vulnerability shielding.

Zero-Day Attack or Exploit. A **zero-day** (or **zero-hour** or **day zero**) **attack** or threat is an attack that exploits a previously unknown vulnerability in a computer application. Such vulnerability could be one that developers have not had time to address and patch. It is called a *zero-day* because the programmer has had zero days to fix the flaw, in other words, a patch is not available. So, a zero-day exploit is one that takes advantage of a security vulnerability on the same day that the vulnerability becomes generally known. Therefore, there are zero days between the time the vulnerability is discovered and the first attack. Once a patch is available, this attack is no longer called a *zero-day exploit*. Zero-day attacks occur during the vulnerability window that exists in the time between when the vulnerability is first exploited and when software developers start to develop and publish a counter to that threat. There are many uses of zero-day attacks. These may include but not limited to malware, spyware or allowing unwanted access to user information, among others.

Drive-by Exploits or Downloads. Drive-by download refers to two issues, each of which concerns the unintended download of computer software from the Internet. That is:

(a) Downloads which a person authorised but without understanding the consequences (e.g. downloads which install an unknown or counterfeit executable program such as ActiveX component, or Java applet.

(b) Any download that happens without a person's knowledge, often a computer virus, spyware,malware, or crimeware.

It so happens that the World Wide Web is the number one source of malware and the majority of these malware programs or threats come from what is called a drive-by download. The term drive-by download describes how malware can infect a computer simply by visiting a website that is running malicious code. Most of the time, these are legitimate websites that have been compromised to redirect a request from a user to another site controlled by hackers. Cybercriminals in this modern day use sophisticated malware packaged in an exploit kit that can find a vulnerability in software among thousands of possibilities. When a browser is redirected to the site hosting an exploit kit, the exploit kit, in turn, probes the operating system, web browser and other software such as the PDF reader or video player that are on the user's

computer. This is in order to find a security vulnerability that the exploit kit can attack. It is worth noting that if the user's site is not applying security updates to their operating system and other pieces software, then the user's site is unprotected against these exploits. Once the exploit kit has identified a vulnerability, then that is where the infection process begins. At this point, the exploit kit downloads what is known as a *payload*, which is the malware that installs itself on the user's computer. Thereafter, the malware executes and causes the havoc that its programmer designed it to bring about. The malware known as Zbot can access the user's emails or bank accounts. Another type of payload called ransomware can hold a user's files hostage until the user pays to have the files released.

A **drive-by install** (or **installation**) is a similar event as a drive-by download. However, it refers to installation rather than download (though sometimes the two terms are used interchangeably).

CopperbeltUniversity
Computer Science Department

Information Management and Network Security

Compiled by: Prof.Dr. Hastings M. Libati

Security Policies and Procedures

Keeping a network secure is a time-consuming process that requires a lot of attention to the detail. Maintaining network security is also a time consuming practice that may involve the following aspects:

(a) Detection of a security breach or network intrusion

(b) Finding the cause of a security breach

(c) Finding the method of an intrusion

(d) Educating users about security on a continual basis

Policies and procedures

To have security practices that make sense, one must first define a security policy that spells out exactly what can and cannot be done on the network. In other words, we could also say that a security policy is a statement of the security that we expect a particular system to enforce. Intruders who might penetrate the network and compromise data or programs do so in many ways. One of those ways is to exploit "friendly users" of the network. Referred to as social engineering, this is perhaps one of the most overlooked but most often used method for getting access to a network. Most employees who simply use a desktop computer for word processing and other office activities are especially prone to this kind of security breach or attack.

A good security policy that is enforced – in some cases through means of technological enforcement – can go a logway toward keeping naive users from disclosing information to those who might do harm to the network.

As can be seen, a security policy defines the security that will be implemented in an organisation, including physical security, document security and network security. Security policies must be implemented completely because random implementation will not bring good results. Before a network can be truly secure, the network staff must implement a total network security policy that includes such aspects as clean desks, security audits, recording and consequences of not complying with the security policy, etc. A short elaboration of some of these aspects is given below:

Security Audit

A security audit is a review of a network to identify components that are not secure. Although a company can carry out an internal security audit, it can also contract an audit with third party. This is a good idea if the security level of the company is to be certified – as is the case in other countries. However, a consultant's audit is a good follow up to an internal audit.

Clean Desk Policy

A clean desk policy means that all important documents, such as books, confidential letters and the like, are removed from a desk and preferably locked away when employees leave their workstations. For a clean desk policy to be effective, users must clean up their desks every time they walk away from them, without exception.

Recording equipment

In a highly security sensitive area, a good security policy should prohibit the unauthorised presence and use of such devices as recording equipment such as tape recorders, video cameras, still cameras and smart phones for obvious reasons.

Procedures

Apart from having policies in place, it is also important to establish procedures to follow for routine tasks that are performed on a periodic basis, such as backups, restores and creating user accounts. When a task is described by a procedure that must be followed, there is less a chance that something out of the ordinary will be done that can compromise security.

Depending on the site in question, there are several policy and procedure documents that one can use to make users be aware of the policies that are in place for computer and network security. One of the best times to make users read policy documents is at the time they get employed. It can be a policy that every new employee reads and signs that he / she has read the documents. Some of these documents could be:

(a) Network connection policy

(b) Acceptable use statement or policy

(c) Usage guidelines

(d) Escalation procedures

Network Connection Policy

This type of document should define the type of systems that can be connected to the network. It should set forth the security requirements, such as operating system features to be used, and a person responsible for approving the attachment of new devices to the network. When configuring a new computer, a switch or even a router, there should be guidelines as to what is permissible and what is not. For firewalls, there should be a separate network connection policy that dictates what type of network traffic is allowed through the firewall, in both directions. If users are allowed to connect to the network using Virtual Private Networks (VPNs), then there should also be specific documents which detail how the computers that such users use are configured. For example, allowing someone to work from home using his or her own computer is about the worst decision one can make. If such a computer is used for personal as well as business work then be rest assured that your network is being opened up to all sorts of programs that might end up infiltrating the computer and even attempt or manage to compromise your network. In such cases, a VPN link must be used.

If for example, a company decides that certain remote work is confidential, then a policy governing this decision should be put in place that requires a separate computer (for mobile users) to be used. The network can be made more secure by ensuring that a computer (most likely laptop) used by company

mobile users is also company-configured. Furthermore, the users should not be allowed to make use of such a computer for personal access to the Internet and also no configuration changes on the computer should be allowed. It is important to bear in mind that if a remote user (company employee) connects to the company network using his / her own computer, the administrator will probably have little say over what may be downloaded from the network. However, by giving the user a company computer and preventing (through a company policy) the use of the computer for personal usage, another step in protecting the network can be achieved.

The use of security programs such as virus monitoring software should always be required in today's Internet-centric environment. Any procedures that must be used to obtain a computer account, along with the types of rights and privileges that can be granted to an account, also should be documented here, as well as what network addresses can be used and how they are controlled. It should be emphasised in this document that no connections are to be made to the network without following procedures laid down in this document and also without notifications made to the proper persons.

It should be clear that there should be strict guidelines on how computers are configured and that users must obtain permission through a written request for any deviances from the established policy. Any programs that are not supported by the organisation should not be allowed on your company computers. When such programs become necessary then the fact should be documented and added to the allowable network connection documents and users informed about the new changes. In no situation should users be allowed to download software from the Internet and install it on their work computers, company computers that are used in a mobile environment, or on home computers that are used to connect to the corporate network environment.

Acceptable Use Statement (Policy) and Usage Guidelines

A computer is a flexible device and therefore can be used for many activities beyond the tasks that are needed by the ordinary worker during a normal workday. It is for this reason; among others that an acceptable use statement is vital.

An acceptable use statement should state that all computer programs used in the organisation are to be supplied by the organisation and that unauthorised programs, such as those brought from home, are not to be used on company computers or network. Users need to be reminded that software piracy is not a victimless crime; instead, it is a crime that is punishable by stiff fines and even jail sentences. It is important that as an administrator you make your users understand this matter and that you protect your company from possible litigation by showing that you have made an effort to prevent unauthorised programs from being placed on computers at your site.

When it comes to unauthorised programs, piracy is just half the issue. Computer viruses can easily make their way from one computer to another exchangeable media like flash disks or by being downloaded from the Internet. Unfortunately, it is usually after more than one system has become infected that a virus is found or reported. If all software that is used on a company network is first examined, approved and distributed by a central source, you will have better control over this problem.

A statement ought to be included in the document that states that users are not allowed to make copies of software or data that is owned by the company and take it home or otherwise use it in an unauthorised manner.

In this document, it will be good to point out to the users that they are required to report any suspicious activity or misuse of network resources. Users should also be made responsible for taking necessary

measures for protecting data and programs within their scope. This includes not leaving a workstation logged in when they are away from it for extended periods. In such cases, users should use a password-locked screen server when they are away from the computer. Another avenue of infiltration is leaving reports or other outputs containing sensitive information lying around, for instance. It should be realised that trusting one employee does not mean trusting all employees. Consequently, matters of this nature require to be put in a policy statement so as to make employees realise how important the issues are.

If dial-up access is granted to users then the users should certainly understand that they are not allowed to give information used for this access to anyone else, either inside or outside the organisation. All access to the network should be done through a VPN or a dial-up mechanism that uses a firewall.

The items that one can put into an acceptable use policy are extensive. You must examine the specific types of resources that you are trying to protect and think up ways to include them in the statement. Some other items that are worth considering for inclusion in the document are given below:

(a) **Harassment of other users**. What might seem like harmless play or jokes in a typical office environment could constitute harassment when it is done over a long period of time. If this is included in the document, users could be forewarned.

(b) **Threats**. Statements that could be construed, as an intention to perform some kind of harmful act should always be treated with the utmost importance and severity.

(c) **Removal of hardware (or software) from the premises without written authorisation**. This includes such items as authorisation codes used to activate copies of software that is downloaded from the Internet, as well as copied software. Furthermore, ensure that no CD burners are provided for employees who do not have an absolute need for them.

(d) **Using the company e-mail system for personal use**. It is a well-known fact today that opening an e-mail attachment can be enough to launch a serious attack on your systems. E-mails have many risks today. These risks should not enter your organisation through personal affairs of your employees. One very burning issue here is coined in the following question: Do you want to pay employees to spend an hour or so each day reviewing their own personal e-mails?

(e) **Bringing hardware into the premises without authorisation, such as laptop computers**. The policy should restrict this categorically. In some case this policy applies specially to vendors and contractors. If this group of people wish to perform functions such as software installation or troubleshooting, then please, if possible provide them with the computer access they need, but then be careful to supervise their access.

(f) **Attempting to access data that is not relevant to the user's job, sometimes referred to as "probing" the network**. This is actually an offence that should be considered serious enough to make the reason for an employee to be fired. There is never a need to go exploring a network. If a particular user wishes to know where data or applications are stored, they should discuss this with management or the right technical staff.

Employees

Any document that outlines guidelines for using the network should point out to employees that they are to behave ethically on the network. For example, network support staff often must access other data owned by other employees when helping them out with some problems. Disclosing information that is obtained during this type of work to a third party is unethical. Administrators and operations personnel

often have elevated rights and privileges on the workstations and server-class computers that are distributed throughout the network. These employees should be made to understand that these privileges include a responsibility to professionally carry out their work without causing problems.

Vendors and Outside Connections

Another area often overlooked when preparing such documents is when outside persons are allowed access to the network. If there are contractors that have been brought in to do work that cannot be done by in-house personnel, then it is better to have a usage guidelines document for them to review and sign. Such a document should specifically include the fact that information on the network is of proprietary nature and should not be disclosed to any outside party, or to any employee in the company who does not have the need to about such information. Additionally, the policy document should state that the contractor cannot discuss with others the type of information to which they have access. It is important to note that information can spread very fast when it lands into wrong hands.

Usually, when hardware repairs need to be done, they are sometimes done by a third party maintenance organisation, or perhaps by the vendor who manufactures the equipment. Diagnosing some equipment problems may require that the repairman have access to a logon account. If you maintain a user account just for this purpose, it is better to be sure that it is one that can be enabled and disabled so that it is available only when it is needed.

Escalation Procedures

Having a plan of action that should be followed in response to a specific event is a good idea. There should be a specific person or persons in the organisation who are designated to be responsible for and to investigate matters relating to security. A document that sets forth the procedures to be followed for particular security violations will also help to show users that security is important for the network to function well and that actions will be taken against those that breach rules and procedures. A document covering escalation procedures should indicate the kinds of issues that are considered a security breach. Such issues could include the following:

(a) Theft of hardware or software

(b) Password discovery or disclosure of usernames and passwords

(c) Improper disposal of storage media such as tapes, floppies, flash disks and printed reports

(d) Sharing of logon accounts

(e) Probing the network to look for where one is not authorised

(f) Interfering with another user's data or account

(g) Suspected network break-in from outside sources

(h) Computer viruses

(i) Physical access violations

Some of these issues may probably seem so obvious. However, to think that a security manager will know how to handle these problems without written procedures is overstated. For example, it is very common

for users to allow others to use their account. It is a lot simpler for one employee to allow another employee to use his / her workstation when the workstation for the other employee is out of service, than it is to get the appropriate permissions from upper-level management. As security manager what do you do when you discover these problems. Users need to know that one can give his / her password to another user to use on one occasion, but this could mean that the password may get used at another occasion, and this time without the knowledge of the owner of the password.

Suppose that there is suspicion that there is break-in from an outside source. What should be the reaction? Change all passwords? Shut down the routers? Panic should not set in! It is better to think about these issues ahead of time and then document a list of steps to follow in such events. The steps should include methods to be used in order to determine the source of the break-in, as well as procedures to follow in order to punish the intruder and to reassert ownership of the pilfered information. For example, if information that is confidential has been compromised, what are the steps that should be taken in order to notify the person to whom the information relates? Are there any legal matters that the security manager needs to know about that pertain to the data that resides on the network?

There are situations when an employee voluntarily resigns from the company or simply retires. It is also possible to have the termination of employment for an employee for actions that caused deliberate damage to the network. In this case, the question is: how can one determine whether the out-going employee has planted any other *time bomb*? What steps should the security manager take to isolate resources that were available to the dismissed employee until further analysis can be done?

It is clear from these examples and scenarios that network security has far-reaching implications. Knowing what to do in the event of a specific security breach will make matters easier for the security manager when such events begin unfolding.

What a Security Policy Should Include

When writing a security policy, one should first perform an inventory of the resources that should be protected. The next step is to identify the users who need to access each resource, and then to determine the most likely places where threats to each resource might come from. If other pieces of information are required, these may be collected. With this information, the construction of a security policy that all users will follow can begin.

To remind users about the importance of security, one might want to post copies of the policy around the offices so that they will be able to see it on regular basis.

A good policy will be composed of several elements, including the following:

(a) **Risk assessment**. What are you trying to protect and from whom? Identify your network assets and possible sources of problems.

(b) **Responsibilities**. Specify who in the organisation is responsible for handling specific matters relating to security. This can include who is authorised to approve a new user account up to items such as who will conduct investigations into security breaches.

(c) **Proper use of network resources**. It is very important to state in the policy that users are not to misuse information, use the network for personal matters, or intentionally cause damage to the network or information that resides on it.

(d) **Legal ramifications**. Be sure to get advice from proper sources about any legal matters that apply to the information that you store or generate on your network. Include statements to this effect in the security policy documents.

(e) **Procedures to remedy security problems**. State what procedures will be followed when a security event occurs and what actions will be taken against those who perpetrate them.

Before designing a security policy it is very important that one makes oneself familiar with the resources found on the network that are vulnerable to potential security threats. The five classes of vulnerability items include:

(a) **Hardware**. This class includes workstations and server-class computers, printers, storage media, network wiring. This class also includes network connection devices such as repeaters, hubs, bridges, switches and routers.

(b) **Software**. Every piece of software that runs on the network is a potential security problem. This includes programs that are purchased from outside vendors and software created in-house by the organisation's programming team. Operating systems frequently have to be patched as new bugs are discovered that give an intruder an easy way to intrude.

(c) **Data**. The most important asset of a network is probably the data that is generated or used by the organisation. Software in terms of application programs and operating systems can be replaced. However, when important data such as customer lists, sales information or proprietary trade secrets, is compromised, this can have a significant impact on the operations of the organisation.

(d) **People**. Users, operators and anyone else who interacts with the network or any device attached to it is a potential security risk.

(e) **Paperwork**. Often overlooked by many, this is a very valuable resource to hackers. Passwords are usually written down. Reports are generated that have confidential information contained in them. Often this vital resource is thrown in bins when it is no longer needed. A better approach is to shred or otherwise make it unusable before getting rid of it.

A good security policy that can be understood by users will go a long way toward preventing some of the problems that can be potentially encountered in running a network. It is important to review the policy with users periodically and to be sure that users understand the responsibilities that go along with having access to the organisation's network.

Passwords

Though at first glance, it does not seem like an important issue, it is very cardinal to enforce a policy that makes users choose good passwords. Simply put, a good password is one that is hard to guess. When one considers that a standard password-cracking technique used by hackers is to simply try every word in a dictionary, then one can begin to understand that luck doesn't have a lot to do with penetrating a network. The success of hackers in such cases comes mostly from lax security that allows doors that are easy to open.

Enforcing Good Passwords

When deciding how passwords are to be constructed, there are a few guidelines that can be followed:

(a) **Use more than one word**. Multiple words that are *glued* together make a pattern of characters that is much harder for a simple password-cracking program to guess. Users have to be educated about using the right words for passwords. For example, users have to be educated about the fact that they should not use the names of celebrities or popular institutions as passwords.

(b) **Use nonalphabetic characters somewhere in the password**. This can be numeric characters or punctuation characters, provided that the operating system being used will permit them.

(c) **Case sensitive passwords are better**. A password that consists of both uppercase and lowercase characters can confound many password-guessing applications. Users are however, warned not to substitute numeric characters that resemble alphabetic characters. Most password-guessing applications find it easy, for example, to substitute the letter "O" for zero. Attackers are too aware of such tricks.

(d) **Passwords should not be too difficult to memorise**. Passwords that are difficult to memorise can lead to frustrated users who begin to write down their passwords so that they will be able to remember them. When this begins to occur then just know that it is time to re-educate such users.

(e) **Use password history restrictions if the operating system permits it**. This means that the operating system keeps track of a limited number of passwords that the user has previously used and will not allow them to be re-used within a certain time frame. A common practice is to change a password when forced to do so and then change it to a value that you like and can easily be remembered. This can be dangerous.

As an administrator, ensure that you do not create user accounts and assign them a password that never gets changed by the user. Most operating systems will allow such a password to be expired on its first use so that when a new user logs in the first time, he / she will be required to change his password.

Sometimes it is important to have a password that makes no sense whatsoever. In a highly secure environment this can make sense, in that the user wants a password that is hard to guess. However, it is important to remember that when a password is difficult to remember it usually gets written down somewhere, which can defeat the purpose of a password altogether. Some operating systems have commands that can assist users to create computer-generated passwords from which they can then choose one – that is if the user has difficulties in thinking up one. The only problem with this method is in getting the user to memorise the password as the computer-generated passwords are generally hard to remember.

Password Policies

No user account, including the one used by the administrator should ever be allowed to keep the same password for an extended period. A good idea for passwords is to require that they be changed every 30 – 60 days, depending on the level of security needed at the site in question. The minimum length of the password should also be enforced. Most operating systems allow administrators to specify the minimum password lengths. This is quite helpful as it ensures that users cannot change their

passwords to one that is shorter than the size that has been specified.Depending on the particular operating system being used, one can enforce many restrictions on passwords or user accounts to enhance security on the network. Some of the capabilities that one might find include:

(a) **Password expirations**. A password should not be used indefinitely.

(b) **Password history lists**. This feature prevents a password from being re-used within a specified period.

(c) **Account lockouts**. When an attacker tries to use the brute-force method to guess a password for an account, it should be possible to lock the account automatically after a specified number of attempts within a specific time frame.

Other Common Security Policies

Security policies can cover hundreds of items, however, below we shall give a summary of some of the more common ones:

Notification. What good is a security policy if no one knows about it? Give users a copy of the security policy when you give then their usernames and passwords. It is also advisable that computers also display a shortened version of the policy when a user attempts connect. For example, "Unauthorised access is prohibited and will be prosecuted to the fullest extent of the law."

Equipment Access. Disable all unused network ports so that non-employees who happen to be on the premises cannot connect their computers to unused ports and gain access to the network. Also, place all network equipment under lock and key.

Door Locks / Swipe Mechanisms. It has to be ensured that only a few, key people know the combination to the cipher lock on protected buildings or rooms; or that only the appropriate people have badges that will allow access (through swiping) to protected buildings or rooms.

Wiring. Network wires or cables should run along the floor where they can be easily accessed. Routers, switches and other network connection devices should not just be hooked up in open office space. They should be in locked closets or rooms, with access to those closets or rooms controlled by badge-swiping systems, for instance.

Badges. Require everybody to wear an ID badge, including contractors and visitors. Furthermore, appropriate access levels to contractors, visitors, and employees ned to be assigned.

Tracking. Require badge access to all entrances to buildings and internal computer rooms. Track and record all entry and exit to these buildings and rooms.

Passwords. Reset passwords at least every one or two months (better every month). Train everyone how to create strong passwords. Set BIOS passwords on every client and server-class computer to prevent BIOS changes.

Monitor Viewing. Block computer monitors so that visitors or people looking through windows can't see them. Be sure that unauthorised users / persons cannot see security guard stations and server-class computer monitors.

Accounts. Each user should have their own, unique user account and employees should not share user accounts. Even temporary employees should have their own accounts. Otherwise, it will be hard to isolate a security breach.

Testing. Let the network security be audited and reviewed at least once a year.

Background checks. Let background checks be done on all network support staff. This may include calling their previous employers, verifying their qualifications, requiring a drug test and checking for any criminal background.

Firewalls. Use a firewall to protect all Internet connections, and use the appropriate proxies and dynamic packet filtering equipment to control access to the network. The firewall should provide as much security as the affected organisation requires and the budget allows.

Intrusion Detection. Let intrusion detection and logging software be used to determine security breaches. Also ensure that the events that need to be monitored are also logged.

Cameras (e.g. CCTV). Cameras should cover all entrances to the company building(s) and the entire parking lot (area). In this regard, it is vital to ensure that cameras are in weather-proof and tamper-proof housings, and review the output at a security monitoring office. Please, record everything of large enough storage devices.

Mail Servers. Provide each person with his or her own e-mail mailbox, and attach an individual network account to each mailbox. If several people need to access a mailbox, please, be sure not to give all of them the password to a single network account. Instead, assign privileges to each person's network account. In this way, one can then track activity to a single person, even with a generic address such as academicoffice@cbu.ac.zm.

DMZ. Use a demilitarised zone for all publicly viewable servers, including web servers, FTP servers and even e-mail relay servers. Ensure not to put such servers outside the firewall. Remember that servers outside the firewall defeat the purpose of the firewall.

Patches. Make sure that the latest security updates are installed after being properly tested on a non-production computer.

Backups. Store backup storage media securely and out of easy reach. Backup storage media should be locked in a waterproof, fireproof safe, and also keep some of the backup materials offsite.

Breaking Policy

A security policy is not effective unless it is enforced, and enforced consistently. No individual should be exempted from policies or consequences of breaking them. Network users need to have a clearly written document that identifies and explains what users are and are not allowed to do. Additionally, it is important to state that breaking the policy will result in a punishment, as well as which types of policy breaks result in which kind of punishment. Punishment may vary, depending on the severity of the incident. If a polity is broken, the appropriate punishment should be administered immediately.

CopperbeltUniversity
Computer Science Department

Information Management and Network Security

Compiled by: Prof.Dr. Hastings M. Libati

Introduction to Cryptography

Cryptography is the strongest tool for controlling against many kinds of security threats. Well-disguised data cannot be read, modified or fabricated easily. Cryptography is rooted in higher mathematics. Cryptography, which can be regarded as secret writing, involves encryption and decryption.

Terminology

Some terminologies are better understood when used in an example. Let us consider the steps involved in sending a message from a **sender**, S, to a **recipient**, R. If S entrusts the message to T, who then delivers it to R, then T becomes the **transmissionmedium**. If an outsider, O, wants to access the message (to read, change, or even destroy it), then we can call O an **interceptor** or **intruder**. Any time after S transmits the message via T, the message is vulnerable to exploitation, and O might try to access the message in any of the following ways:

(a) **Block** it, by preventing it from reaching R, thereby affecting the availability of the message.
(b) **Intercept** it, by reading or listening to the message, thereby affecting the confidentiality of the message.
(c) **Modify** it, by seizing the message and changing it in some way, affecting the message's integrity.
(d) **Fabricate** an authentic-looking message, arranging for it to be delivered as if it came from S, thereby also affecting the integrity of the message.

Encryption could be one of the answers to these problems. Encryption can be said to be one of the most fundamental building blocks of secure computing and it is a means of maintaining secure data in an insecure environment. Encryption is a security technique that is used in protecting programs, databases, networks and electronic communications. **Encryption** could be defined as the process of encoding a message so that its meaning is not obvious. **Decryption**, on the hand is the reverse process that transforms an encrypted message back into its normal and original form. Alternatively, the terms **encode** and **decode** or **encipher** and **decipher** are used instead of encrypt and decrypt. A system for encryption and decryption is called a **cryptosystem**.

The original form of a message is known as **plaintext**, and the encrypted form is called **ciphertext**. Plaintext message (P) is denoted as a sequence of individual characters $P = (p_1, p_2, \ldots p_n)$. Similarly, ciphertext is written as $C = (c_1, c_2, \ldots c_n)$. To describe the transformations between plaintext and ciphertext formal notations are used. For example, C = E(P) and P = D(C), where C represents the ciphertext, E is the encryption rule, P is the plaintext, and D is the decryption rule. What is required is a cryptosystem for which P = D(E(P)).

Encryption Algorithms

A cryptosystem involves a set of rules for how to encrypt the plaintext and how to decrypt the ciphertext. The encryption and decryption rules, called **algorithms**, often use a device called a **key**, denoted by **K**, so that the resulting ciphertext depends on the original plaintext message, the algorithm, and the key value. This dependency is written as C = E(K,P). E is a set of encryption algorithms and K is the encryption key.

Sometimes the encryption and decryption keys are the same, so P = D(K, E(K,P)). This form is called **symmetric** encryption because D and E are mirror-image processes. In some cases, encryption and decryption keys come in pairs. Then, a decryption key K_D, inverts the encryption of K_E so that P = D(K_D, E(K_E,P)). Encryption algorithms of this form are are called **asymmetric** because converting C back to P involves a series of steps and a key that are different from the steps and key of E.

It should be noted that a key gives some flexibility in using an encryption scheme. We can create different encryptions of one plaintext message just by changing the key. Moreover, using a key provides additional security. For instance, if the encryption algorithm should fall into an interceptor's hands, future messages can still be kept secret because the interceptor will not know the new key value after changing the key. An encryption scheme that does not require the use of a key is called a **keyless cipher**.

The word **cryptography** means hidden writing, and it refers to the practice of using encryption to conceal text and decryption to do the opposite of encryption. A **cryptanalyst** studies encryption and encrypted messages, hoping to find the hidden meanings of the encrypted text.

Both a **cryptographer** and cryptanalyst attempt to translate coded material back to its original form. Normally a cryptographer works on behalf of a legitimate sender or receiver, whereas a cryptanalyst works on behalf of an unauthorised interceptor. **Cryptology** is the research into and the study of encryption and decryption; it includes both cryptography and cryptanalysis.

Cryptanalysis

A cryptanalyst's chore is to **break** an encryption. That is, the cryptanalyst attempts to deduce the original meaning of a ciphertext message. In other words, he or she hopes to determine which decrypting algorithm matches the encrypting algorithm so that other messages encoded in the same way can be broken, too. Therefore, the cryptanalyst will attempt to do the following activities:

(a) Break a single message
(b) Recognise patterns in encrypted messages, to be able to break subsequent ones by applying a straightforward decryption algorithm
(c) Infer some meaning without even breaking the encryption, such as noticing an unusual frequency of communication or determining something by whether the communication was short or long
(d) Deduce the key, to break subsequent messages easily
(e) Find weaknesses in the implementation or environment of use of encryption
(f) Find general weaknesses in an encryption algorithm, without necessarily having intercepted any message.

A cryptanalyst works with a variety of pieces of information. These include encrypted messages, known encryption algorithms, intercepted plaintext, data items known or suspected to be in a ciphertext message, mathematical or statistical tools and techniques, properties of languages, creativity and luck. Each piece of evidence can provide a clue and then the analyst puts the clues together to try to form a larger picture of a message's meaning in the context of how the encryption is done. Please, remember that there are no

rules in cryptanalysis. An interceptor can use any means available to extract the message from its encryption.

An encryption algorithm is called **breakable** when, given enough time and data, an analyst can determine the algorithm. However, an algorithm that is theoretically breakable may in fact be impractical to break. Let us consider a 25-character message that is expressed in uppercase letters. A given cipher scheme may have 26^{25} possible decipherments, so the task is to select the right one out of 26^{25}. If a computer can perform on the order of 10^{10} operations per second, finding this decipherment would require on the order of 10^{25} seconds or roughly 10^{17} years. In this case, although we know that theoretically we could generate the solution, determining the deciphering algorithm by examining all possibilities can be ignored as infeasible with current technology.

Two other important issues must be considered when considering the breakability of encryption algorithms. First, the cryptanalyst cannot be expected to try only the hard and long way. Some creativity could be used here which could actually help reduce the number of times that need be tried. Some tricks here and there could help reveal some words or phrases from the encrypted message and some educated guesses could finally have most of or even the entire message decrypted.

Second, estimates of breakability are based on current technology. We know that advances in computing technology are continuous. Moore's Law states that the speed of processors (rate of processing) doubles every 1.5 years. We are all living witnesses to this fact. Therefore, it is risky to pronounce an algorithm secure just because it cannot be broken using current technology, or worse, that it has not been broken before.

Representing Characters

Let us consider the illustration below:

Letter	A	B	C	D	E	F	G	H	I	J	K
Code	0	1	2	3	4	5	6	7	8	9	10
Letter	L	M	N	O	P	Q	R	S	T	U	V
Code	11	12	13	14	15	16	17	18	19	20	21
Letter	W	X	Y	Z							
Code	22	23	24	25							

Since most encryption algorithms are based on mathematical transformations, they can be studied better in mathematical form. Thus, the letter A is represented by a zero, B by one and so on. This representation allows us to consider performing arithmetic on the "letters" of a message. In this case, arithmetic is performed as if the alphabetic table were circular. In other words, addition wraps around from one end of the table to the other so that Y + 3 = B. Thus, every result of an arithmetic operation is between 0 and 25.

The two earliest forms of encryption are **substitutions** and **transpositions**. In substitution, one letter (character) is exchanged for another and in transpositions; the order of the letters (characters) is rearranged.

Substitution Ciphers

One of the most commonly cited substitution cipher is the Caesar Cipher. The Caesar Cipher has an important place in history. Julius Caesar is said to have been the first to use this scheme, in which each letter is translated to the letter a fixed number of places after it in the alphabet. Caesar used a shift of 3, so plaintext letter $\boldsymbol{p_i}$was enciphered as ciphertext letter c_i by the rule

$$c_i = \mathrm{E}(p_i) = p_i + 3$$

A full translation chart of the Caesar cipher is given below:

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c

Let us consider the result of applying Caesar's encryption technique to "TREATY IMPOSSIBLE". If we did not know the plaintext and we were trying to guess it, we would have many clues from the ciphertext. For example, the break between the two words is preserved in the ciphertext and double letters are preserved. The **SS** is translated to **vv**. We might also notice that when a letter is repeated, it maps again to the same ciphertext letter as it did previously. So the letters **T**, **I** and **E** always translate to **w**, **l** and **h**. These clues make this cipher easy to break.

CDPELA LV XQGRXEWHGOB DPRQJ WKRVH FRXQWULHV WKH ZRUOG RYHU

WKDW JRG KDV UHDOOB EOHVVHG LQ DEXQGDQFH

Assignment: *As an exercise, please, decipher the following ciphertext using Caesar's Cipher:*

wklvphvvdjh lv qrwwrrkdugwreuhdn

The Caesar's Cipher is not the only substitution method. There are many other substitution methods. We have already seen that in substitutions, the alphabet is scrambled and each plaintext letter maps to a unique ciphertext letter. This technique can be described in a more mathematical way. Formally, we say that a **permutation** is a reordering of the elements of a sequence. For instance, we can permute the numbers 1 to 10 in many ways, including the permutation π_1 = 1, 3, 5, 7, 9, 10, 8, 6, 4, 2; and π_2 = 10, 9, 8, 7, 6, 5, 4, 3, 2, 1. A permutation is a function, so we can write expressions such as $\pi_1(3) = 5$meaning that the letter in position 3 is to be replaced by the fifth letter. If the set is the first ten letters of the alphabet then $\pi_1(3) = 5$ means that c is transformed into E.

One way to scramble an alphabet is to use a **key**. In this case, a key is a word that controls the permutation. For instance, if the key is the word **word** then the sender or receiver first writes the alphabet and then writes the key under the first few letters of the alphabet as shown below:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
w	o	r	d	a	b	c	e	f	g	h	i	j	k	l	m	n	p	q	s	t	u	v	x	y	z

In this example, the key is short, so most plaintext letters are only one or two positions off from their ciphertext equivalents. With a longer keyword, the distance is greater and less predictable. Because π

must map one plaintext letter to exactly one ciphertext letter, duplicate letters in a keyword, such as the second **s** and **o** in **professional**, are dropped.

It is important to note that an important issue in using any cryptosystem is the time it takes to turn plaintext into ciphertext and vice versa. For instance, transforming a single character can be done in constant amount of time, so we can express the complexity on an algorithm by saying that the time to encrypt a message of **n** characters is proportional to **n**. One way of thinking of this expression is that if one message is twice as long as another, it will take twice as long to encrypt.

Transpositions (Permutations)

The goal of substitution is **confusion**; the encryption method is an attempt to make it difficult for a cryptanalyst or intruder to determine how a message and the key were transformed into ciphertext. A **transposition** is an encryption in which the characters of the message are re-arranged. With transposition, cryptography aims for **diffusion**, widely spreading the information from the message or key across the ciphertext. Transpositions try to break established patterns. Because a transposition is a re-arrangement of the symbols of a message, it is also known as a **permutation**. There are many forms of transpostion encryption. One such is the **columnar transposition**.

Columnar Transposition

The columnar transposition is a re-arrangement of the characters in the plaintext into columns. The following set of characters is a five-column transposition. The plaintext characters are written in rows of five and arranged one row after another, as shown below:

The resulting ciphertext is formed by reading down the columns. For instance, suppose we want to write the plaintext message: **THIS IS A MESSAGE TO SHOW HOW A COLUMNAR TRANSPOSITION WORKS.** We arrange the letters in five columns as shown:

T H I S I
S A M E S
S A G E T
O S H O W
H O W A C
O L U M N
A R T R A
N S P O S
I T I O N
W O R K S

The resulting ciphertext would then be read down the columns as

tssohoaniwhaasolrstoimghw
utpirseeoamrookistwcnasns

In this example, the length of this message happens to be a multiple of five; so all columns are the same length. However, if the message length is not a multiple of the length of a row, the last columns will be

one or more letters short. When this happens, we sometimes use an infrequent letter, such as **x** , to fill in any short columns.

Encipherment / Decipherment Complexity

This cipher involves no additional work beyond arranging the letters and reading them off again. Therefore, the algorithm requires a constant amount of work per character, and the time needed to apply the algorithm is proportional to the length of the message. However, we must also consider the amount of space needed to record or store the ciphertext. The columnar transformation requires storage for all characters of the message, so the space required is not constant like in the algorithms that we have seen so far. In this case, the space required is directly dependent on the length of the message.

Furthermore, we cannot produce output characters until all the message's characters have been read. This restriction occurs because all characters must be entered in the first column before output of the second column can begin, but the first column is not complete until all characters of the message have been read. Thus the delay associated with this algorithm also depends on the length of the message.

Because of the storage space needed and the delay involved in decrypting the ciphertext, this algorithm is not especially appropriate for long messages when time is of the essence.

Digrams and Trigrams

Just as there are characteristic letter frequencies, there are also characteristic patterns of pairs of adjacent letters, called **digrams**. Letter pairs such as **re**, **th**, **en**, and **ed** appear very frequently. The table in figure 7 shows the ten most common digrams and **trigrams** (groups of three letters) in English. The figure shows these in descending order of frequency.

Digrams	Trigrams
EN	ENT
RE	ION
ER	AND
NT	ING
TH	IVE
ON	TIO
IN	FOR
TF	OUR
AN	THI
OR	ONE

Figure 7: Most common English Digrams and Trigrams

It is also useful to know which pairs and triples do not occur often in English because that information helps eliminate possibilities when decrypting a message. For instance, digram combinations such as **vk** and **qp** occur very infrequently. However, the infrequent combinations can occur in acronyms, in foreign words or names or across word boundaries.

Cryptanalysis by Digram Analysis

Suppose we want to decrypt a message that has used a columnar transposition for its encryption algorithm. Transpositions leave the plaintext letters intact, so the work for the cryptanalyst is more exhausting; more relies on a human's judgement of what "l**ooks right**".

The first step in analysing the transposition is computing the letter frequencies. If we find that in fact all letters appear with their normal frequencies, we can infer that a transposition has been performed. Given a string of text, the trick then is to break it into columns.

Two different strings of letters from a transposition ciphertext can represent pairs of adjacent letters from the plaintext. The problem is to find where in the ciphertext a pair of adjacent columns lies and where the ends of the columns are.

Combinations of Approaches

Substitutions and transpositions can be considered as building blocks for encryption. Other techniques can be based on each of them, both of them or a combination with yet another approach. When dealing with encryption, it is important to keep in mind that each technique is only one piece of the large puzzle. For example, we all know that we can have a locked car inside a locked garage that is itself inside a locked courtyard. In the same way, we could also combine various approaches to encryption so as to strengthen the overall security of a system.

A combination of two ciphers is called a **product cipher**. Product ciphers are typically performed one after another, as in $E_2(E_1(P,k_1), k_2)$. Just because we apply two ciphers does not necessarily mean that the result is any stronger than, or even as strong as, either individual cipher.

Making Good Encryption Algorithms

What does it mean for a cipher to be good? There are many kinds of encryption techniques. The meaning of good depends on the intended use of the cipher. For instance, a cipher to be used by military personnel in the field has different requirements from one to be used in a secure installation with substantial computer support.

In 1949, Claude Shannon proposed several characteristics that identify a good cipher. The proposed characteristics are given below:

1. *The amount of secrecy needed should determine the amount of labour appropriate for the encryption and decryption*. This means that even a simple cipher may be strong enough to deter the casual interceptor or to hold off any interceptor for a short time.

2. *The set of keys and the enciphering algorithm should be free from comp*lexity. This principle implies that we should restrict neither the choice of the keys nor the types of the plaintext on which the algorithm can work. For instance, an algorithm that works only on plaintext that has an equal number of A's and E's may be rendered useless. Similarly, it would be difficult to select keys such that the sum of the values of the letters of the key is a prime number. Restrictions such as these make the use of the encipherment prohibitively complex. If the process is too complex, it will not be used. Furthermore, the key must be transmitted, stored and remembered, so it must be short.

3. *The implementation of the process should be as simple as possible*. A complicated algorithm is more prone to error or likely to be forgotten. With the development of digital computers, algorithms far too complex for hand implementation became feasible. Still, the issue of

complexity is important. People will avoid an encryption algorithm whose implementation process severely hinders message transmission, thereby undermining security. Besides, a complex algorithm is more likely to be programmed incorrectly.

4. *Errors in ciphering should not propagate and cause corruption of further information in the message*. For example, dropping one letter in a columnar transposition throws off the entire remaining encipherment.

5. *The size of the enciphered text should be no larger than the text of the original message*. A longer ciphertext means more storage space and more time to communicate (more network bandwidth required).

Properties of *Trustwothy* Encryption Systems

Commercial users have several requirements that must be satisfied when selecting an encryption algorithm. Thus, when saying that encryption is *commercial grade* or *trustworthy*, it means that the encryption method meets the following constraints:

1. *It is based on sound mathematics*. Good cryptographic algorithms are not just invented; they are derived from solid principles.

2. *It has been analysed by competent experts and found to be sound*. Even the best cryptographic experts can think of only so many possible attacks, and the developers may become too convinced of the strength of their algorithm. Thus, a review by critical outside experts is essential.

3. *It has stood the test of time*. As a new algorithm gains popularity, people continue to review both its mathematical foundations and the way it builds on those foundations. Although a long period of successful use and analysis is not a guarantee of a good algorithm, the flaws in many algorithms are discovered relatively soon after their release.

Three algorithms are popular in the commercial world. These are the Data Encryption Standard (DES), Rivest-Shamir-Adelman (RSA – named after the inventors) and the Advanced Encryption Standard (AES). These algorithms meet the criteria for commercial-grade encryption.

Symmetric and Asymmetric Encryption Systems

The symmetric system provides a two-way channel to their users: A and B share a secret key, and they can both encrypt information to send to the other as well as decrypt information from the other. As long as the key remains secret, the system also provides **authentication** – proof that the message was not sent by someone other than the declared sender. Authenticity is ensured because only the legitimate sender can produce a message that will decrypt properly with the shared key.

The symmetry of this situation is a major advantage of this type of encryption, but it also leads to a problem: key distribution. The question is how do A and B obtain their shared secret key? If A wants to share encrypted communication with another user C then A and C need a different shared key. Key distribution is a major difficulty in using symmetric encryption. In general, **n** users who want to communicate in pairs need n * (n – 1) / 2 keys. In other words, the number of keys needed increases at a rate proportional to the square of the number of users. So a property of symmetric encryption systems is that they require a means of **key distribution**.

Public key systems on the other hand excel at key management. By the nature of the public key approach, a public key can be sent in an e-mail message or even posted in a public directory. Only the corresponding private key, which presumably is kept private, can decrypt what has been encrypted with the public key.

But for both kinds of encryption, a key must be kept well secured. So, for all encryption algorithms, key management is a major issue. It involves storing, safeguarding and activating keys.

Stream and Block Ciphers

Stream ciphers are those encryption algorithms that convert one symbol of plaintext immediately into a symbol of ciphertext. The transformation depends only on the symbol, the key and the control information of the encipherment algorithm. Some kind of errors, such as omitting a character during encryption, affects the encryption of all future characters. A **block cipher**, on the other hand, encrypts a group of plaintext symbols as one block. The columnar transposition is an example of a block cipher algorithm. In the columnar transposition, the entire message is translated as one block. Therefore, block ciphers work on blocks of plaintext and produce blocks of ciphertext.

The Data Encryption Standard

The Data Encryption Standard started as the **data encryption algorithm (DEA)**developed by IBM. It later became known as the Data Encryption Standard (DES), although its proper name is **Data Encryption Algorithm (DEA)** in the United States and **Data Encryption Algorithm-1 (DEA-1)** in other countries. DES has been accepted as an international standard by the International Standards Organisation (ISO). Many hardware and software systems have been designed with the DES as the encryption system.

The DES algorithm is a careful and complex combination of two fundamental building blocks: substitution and transposition. The algorithm derives its strength from repeated application of these two techniques, on top of the other, for a total of 16 cycles.

The algorithm begins by encrypting the plaintext as blocks of 64 bits. The key is 64 bits long, but infact it can be any 56-bit number. The extra 8 bits are often used as check digits and so do not affect encryption in normal implementations. The user can change the key at will any time there is uncertainty about the old key.

DES leverages the two techniques of confusion and diffusion to conceal information. That is, the algorithm accomplishes two issues: it ensures that the output bits have no obvious relationship to the input bits and spreading the effect of one plaintext bit to other bits in the ciphertext. Substitution provides the confusion and transposition provides the diffusion. In general, the plaintext is affected by a series of cycles of a substitution then a permutation.

DES uses only standard arithmetic and logical operations on numbers up to 64 bits long, so it is suitable for implementation in software on most computers. Although complex, the algorithm is repetitive, making it suitable for implementation on a single-purpose chip. Several such chips are available on the market for use as basic components in devices that use DES encryption in an application.

DES was designed for a 56-bit key. With the power of computers growing rapidly, most people became uncomfortable with the security provided by DES. This concern let to the development of double DES and triple DES.

Ordinary DES has a key space of 56 bits, double DES with an equivalent of 57 bits key is scarcely better, but two-key triple DES gives an effective length of 80 bits, and three-key triple DES gives a strength of 112 bits.

The Advanced Encryption Algorithm (AES)

The Advanced Encryption Algorithm (AES) is a commercial-grade symmetric algorithm. The AES, sometimes also referred to as Rijndael (after its Belgian inventors), is a fast algorithm that can be implemented easily even on simple processors. Although it has a strong mathematical foundation, it primarily uses substitution, transposition, the shift, exclusive OR and addition operations. Like DES, AES uses repeat cycles. There are 10, 12 or 14 cycles for keys of 128, 192 and 256 bits, respectively. In the AES speak, the *cycles* are also called *rounds*. Each cycle consists of four steps as given below:

1. **Byte substitution**: This step uses a substitution box structure that substitutes each byte of a 128-bit block according to a substitution table. This is a straight diffusion operation.

2. **Shift row**: Each column is multiplied by a polynomial. This is a straight confusion operation.

3. **Mix column**: This step involves shifting left and exclusive-ORing bits with themselves. These operations provide both confusion and diffusion.

4. **Add subkey**:A key is derived and added to each column. The cycle results are also exclusive ORed with the key. This operation provides confusion and incorporates the key.

Please, note that these four steps perform both diffusion and confusion on the input data. Bits from the key are combined with intermediate result bits frequently; so that the key bits are well diffused throughout the result. Furthermore, these four steps are performed extremely fast. Although the steps of a cycle are simple to describe and seem to be rather random transformations of bits, these transformations have a sound mathematical origin.

Public Key Encryption

We have looked at encryption from the point of view of making the scrambling easy to do (so that the sender can easily encrypt a message) and the decryption easy for the receiver but not for the intruder. However, this functional view of transforming plaintext to ciphertext is only part of the picture. It is important also to examine the role of the keys in encryption. We have noted how useful keys can be in deterring an intruder, but we have assumed that the key must remain secret for it to be effective. It is worth noting that the key can also remain public but still protect the message. This is the concept of the public key encryption system. The basis of the public key encryption system is to allow the key to be divulged but to keep the decryption technique secret. Public key cryptosystems accomplish this goal by using two keys: one to encrypt and the other to decrypt.

One might wonder why making the key public should be desirable at all! Here we need to remember that with the conventional symmetric key system, each pair of users needs a separate key. However, with public key systems, anyone using a single public key can send a secret message to a user and the message remains adequately protected from being read by an interceptor. We have seen that an n-user system requires n * (n –1) / 2 keys and each user must track and remember a key for each other user with which he or she wants to communicate. As the number of users grows, the number of keys increases very rapidly. Determining and distributing these keys is a problem. More serious is maintaining security for the keys already distributed, because it is difficult to expect users to memorise so many keys.

In a **public key system** or **asymmetric encryption system**, each user has two keys: a public and a private key. The user may publish the public key freely because each key does only half of the encryption and decryption process. The keys operate as inverses, meaning that one key undoes the encryption provided by the other key.

With public key systems, a user can encrypt a message with a **private key**, and the message can be revealed only with the corresponding public key. These two properties tell us that public and private keys can be applied in either order.

Rivest-Shamir-Adelman (RSA) Encryption

One of the popular commercial-grade public key cryptosystem is the Rivest-Shamir-Adelman (RSA) Cryptosystem. Named after its inventors and introduced in 1978, the RSA algorithm still remains secure to this date. RSA relies on an area of mathematics known as number theory in which mathematicians study the properties of numbers such as their prime factors. The RSA encryption algorithm combines results from number theory with the degree of difficulty in determining the prime factors of a given number. The RSA algorithm also operates with arithmetic mod **n**.

The two keys used in RSA, **e** (encrypt)and **d** (decrypt) are used for encryption and decryption. These keys are actually interchangeable. Either of them can be chosen as the public key, but one having been chosen, then the other one must be kept private. Because of the nature of the RSA algorithm, the keys can be applied in either order.

$P = E(D(P)) = D(E(P))$

Any plaitext block P is encrypted as P^e mod **n**. Because the exponentiation is performed mod **n**, factoring P^e to uncover the encrypted plaintext is difficult. However, the decrypting key **d** is carefully chosen so that $(P^e)^d$ mod **n** = P. Thus the legitimate receiver who knows **d** simply computes $(P^e)^d$ mod **n** = P and recovers P without having to factor P^e.

The Uses of Encryption

Encryption algorithms alone are not an answer to everyone's encryption needs. Although encryption implements protected communications channels, it can also be used for other duties. In fact, combining symmetric and asymmetric encryption often capitalises on the best features of each.

Public key algorithms are useful only for specialised tasks because they are very slow. A public encryption can take many times as long to perform as a symmetric encryption because the underlying modular exponentiation depends on multiplication and division, which are inherently slower than the bit operations (addition, exclusive OR, substitution and shifting) on which symmetric algorithms are based. There are four main applications of encryption and these are *cryptographic hash functions*, *key exchange*, *digital signatures* and *certificates*.

Cryptographic Hash Functions

Encryption is most commonly used for secrecy. In some cases, however, integrity is a more important concern than secrecy. For example, in a document retrieval system that contains legal records, it may be important to know that the copy retrieved is exactly what was stored. In the same way, in a secure communications system, the need for the correct transmission of messages may override secrecy concerns.

It so happens that in most files, the elements or the components of the file are not bound together in any way. That is, each byte or character is independent of every other one in the file. This lack of binding means that changing one value affects the integrity of the file, but that one change can easily go undetected.

To ensure integrity, we would really wish to put a *seal* or *shield* around the file so that we can detect when the seal has been broken and thus know that something has been changed. This notion is similar to the use of wax seals on envelopes in the olden days. If the wax was broken then the recipient would know that someone had broken the seal and read the message inside. In the same way, cryptography can be used to **seal** a file, encasing it so that any change becomes apparent. One technique for providing the seal is to compute a cryptographic function, sometimes called a **hash** or **checksum** or **message digest** of the file.

The hash function has special characteristics. For example, some encryptions depend on a function that is easy to understand but difficult to compute. For a simple example, consider the cube function, $y = x^3$. It is relatively easy to compute x^3 even by hand. But the inverse function $\sqrt[3]{y}$ is much more difficult to compute. And the function $y = x^2$ has no inverse function since there are two possibilities $\sqrt[2]{y}$: +x and –x. Functions like these, which are much easier to compute than their inverses, are called **one-way functions**.

A one-way function can be useful in an encryption algorithm. The function must depend on all bits of the file being sealed, so that any change to even a single bit will alter the checksum result. The so calculated checksum value is stored with the file. Then, each time the file is accessed or used, the checksum is recomputed. If the computed checksum matches the stored value then it is likely that the file has not been changed.

A cryptographic function such as DES or AES is especially appropriate for sealing values, since an outsider will know the key and thus will not be able to modify the stored value to match with the data being modified. In block encryption, schemes, **chaining** means linking each block to the previous block's value (and therefore all previous blocks), for example, by using an exclusive OR to combine the encrypted previous block with the encryption of the current one. A file's cryptographic checksum could be the one obtained from the last block of the chained encryption of a file since that block will depend on all other blocks.

The most widely used cryptographic hash functions are the **Message Digest 4 (MD4)**, **Message Digest 5 (MD5)**, **Secure Hash Algorithm (SHA)** and **Secure Hash Standard (SHS)**.

Key Exchange

One major application of encryption is in the area of key exchange. Suppose we need to send protected message to someone we do not know and who does not know us. For instance, we may want to use a web browser to connect to connect with a shopping website, exchange private (encrypted) e-mails, or arrange for two hosts to establish a protected channel. Such situations depend on being able to exchange an encryption key in such a way that nobody else can intercept it. It should be noted that in order to establish an encrypted session, in such cases, one needs an encrypted means to exchange keys.

Public key encryption can help here. Suppose that S (sender) and R (receiver) want to derive a symmetric key. Suppose also that S and R both have public key system keys for a common encryption algorithm. The keys are $k_{PRIV\text{-}S}$, $k_{PUB\text{-}S}$, $k_{PRIV\text{-}R}$ and $k_{PUB\text{-}R}$, for the private and public keys for S and R, respectively. The simplest solution is for S to choose any symmetric key K and send $E(k_{PRIV\text{-}S}, K)$ to R. Then R takes S's public key, removes the encryption and obtains K. However, any eavesdropper who can get S's public key can also obtain K. Instead, let S send $E(k_{PUB\text{-}R}, K)$ to R. Then only R can decrypt K. Unfortunately, R has no assurance that K came from S. The solution, however, is for S to send to R: $E(k_{BUP\text{-}R}, E(k_{PRIV\text{-}S},K))$.

Another approach not requiring pre-shared public keys is the **Diffie-Hellman key exchange protocol**. In this protocol, S and R use some simple arithmetic to exchange a secret. They agree on a field size **n** and a starting number **g**; they can communicate these numbers in the clear. Each thinks up a secret number, say, **s** and **r**. S then sends to R g^s and R sends to S g^r. Then S computes $(g^r)^s$ and R computes $(g^s)^r$, which are the same, so $g^{rs} = g^{sr}$ becomes the shared secret key. However, these computations are done over a field of integers mod **n**.

Digital Signatures

Cryptosystems are also very important when dealing with digital signatures.Let us imagine that one wishes to request one's bank to transfer a particular amount of money to a friend's account. Using the conventional paper mode, this would proceed as follows:

1. A cheque which is a *tangible object* authorising a financial transaction is issued.

2. The signature on the cheque *confirms authenticity* because (presumably) only the legitimate signer can produce that signature.

3. In the case of an alleged forgery, a third party can be called in to *judge authenticity*.

4. Once a cheque is cashed, it is cancelled so that it *cannot be reused*.

5. The paper cheque is *not alterable*. Or, most forms of alteration are easily detected.

Therefore, transacting business by cheque depends on *tangible objects* in a *prescribed form*. But tangible objects do not exist for transactions on computers. Therefore, authorising payments by computer requires a different model.

Digital Signatures: On his part, the customer wants to be certain that the bank cannot forge such messages. We can see that both parties want to be sure that the message is new, not a reuse of a previous message; and that the message has not been altered during transmission. Using electronic signals instead of paper complicates this process. However, this complication can be overcome with digital signatures. A **digital signature** is a protocol that produces the same effect as a conventional signature. It is a mark that only the sender can make, but other people can easily recognise as belonging to the sender. Just like a conventional signature, a digital signature is used to confirm agreement to a message. We know that if one encrypts a message with one's private key then one has digitally signed that message. The message can only be descripted with the public key of the one who signed the message. At this point it becomes clear as to who signed the message.

A digital signature must meet the following conditions:

1. *It must be unforgeable*. If person P signs message M with signature S -S(P,M), it is impossible for anyone else to produce the pair [M, S(P,M)].

2. *It must be authentic*. If a person R receives the pair [M, S(P,M)] purportedly from P, R can check that the signature is really from P. Only P could have created this signature and the signature is firmly attached to M.

3. *It is not alterable*. After being transmitted, M cannot be changed by S, R or an interceptor. Any such alterations should be easily detectable.

4. *It is not reusable*. A previous message presented again will be instantly detected by R.

If a user encrypts a message with his orher private key then the user is said to have digitally signed the message.

Digital Certificates

Yet another application for cryptosystems is in the use of digital certificates. Human beings establish trust all the time in their daily interactions with people. We identify people we know by recognising their voices, faces or even handwriting. At other times we use affiliation to convey trust. For instance if a stranger telephones us and we hear "I am calling from the school … about your sister …". We may decide to trust the caller even if we do not know him or her.

A human being has what we call a "**trust threshold**", that is a degree to which we are willing to believe an unidentified individual. This threshold exists in commercial interactions as well. In the case of human beings, trust is based on appearance of authenticity (such as a printed, signed form) and outside information (such as a credit report). For electronic communication to succeed, we must develop similar ways for two parties to establish trust without having met.

This trust can also be established through a **common respected individual**. If a student has forgotten or does not have an examination slip in his possession then he will not be allowed to write his examinations. However, a senior member of staff could be called to identify such a student. This can help to establish trust and the student can be allowed to write the examination.

A **digital certificate** can be referred to as an electronic passport that allows a person, computer or an organisation to exchange information securely over the Internet using the public key infrastructure (KPI). Usually, a digital certificate is also referred to as a public key certificate. A digital certificate provides identifying information is forgery resistant and can be verified because it was issued by an official, trusted agency. A digital certificate contains the name of the certificate holder, a serial number, expiration dates, a copy of the certificate holder's public key and the digital signature of the certificate-issuing authority, usually called **certificate authority (CA)** so that a recipient can verify that the certificate is genuine and valid. To provide evidence that a certificate is genuine and valid, it is digitally signed by a root certificate belonging to a trusted certificate authority. Operating systems and browsers contain lists of trusted CA root certificate authorities so that they can easily verify certificates that the CAs have issued and signed. However, when PKI is deployed internally, digital certificates can be self-signed. The widely used standard for digital certificates is the X.509 standard.

Certificates to Authenticate an Identity

Public key infrastructure (PKI) is a system for establishing the identity of people who hold cryptographic keys. Therefore, identifying and authenticating entities in the digital world can be accomplished with the use of a public key infrastructure (PKI). Further, this can be achieved by employing digital signatures and hash functions. For instance, a user's public key and other identities such as the name can be bound together into a **digital certificate**, which can then be signed by some trusted third party called a **certificate authority**, certifying the accuracy of the binding. In this way, people can now use this public key, trusting that they are using an authentic key.

In an organisation, the managerial structure can be used for certificate signing in order to authenticate users. However, this is not necessarily necessary. Anyone who is considered acceptable as an authority can sign a certificate. For example, if one wants to determine whether a person received a degree from a university, one would not contact the Chancellor or the Vice Chancellor but would instead go to the Registrar's officeor the Academic Office.

Sometimes, a particular person is designated to attest to the authenticity or validity of a person or document. For example, a notary publicly attests to the validity of a (written) signature on a document. Some companies have a security officer to verify that an employee has appropriate security clearances to read a document or to attend a meeting. Many companies have a separate human resources office for each site or each plant location; the human resources manager vouches for the employment status of the employees at that site. So any of such officers or heads of offices could credibly sign certificates for people under their purview. This shows that natural hierarchies already exist in society and these same hierarchies can be used to validate certificates.

The only problem with hierarchy is the need for trust at the top level. The entire chain of authenticity is secure because each certificate contains the key that decrypts the next certificate, except for the top. Within a company, it is reasonable to trust the person at the top. But if certificates are to be widely used in electronic commerce, people must be able to exchange certificates securely across companies, organisations and countries.

The Internet is a large federation of networks for intercompany, interorganisational and international (as well as intracompany, intraorganisational and intranational) communication. It is not a part of any government or any company. The Internet is governed by a board called the Internet Society. The Internet Society has power only because its members, the governments and companies that together make up the Internet, agree to work together. Different companies, such as SecureNet, Verisign, Baltimore Technologies, Deutche Telecomm and many more are root certification authorities.This means that each of these certification authorities is a highest authority that signs digital certificates. So instead of one root and one top there are many roots, largely structured around national boundaries.

One might believe that public and secret keys contain little information other than the actual values that are needed for public key encryption and decryption. There is more information stored with each public key. In addition to the encryption information, we may wish to store the user's name or some other kind of identifying information. Otherwise if we had keys for three people, there would be no easy way to tell them apart. In the same way, we need to store more information with each secret key, so that we have a way of telling which secret key belongs to which public key.We have already seen that third parties that certify the information on the key before it is signed are called **certification authorities**.

Digital certificates give people, organisations and businesses on the Internet simple ways to verify each other's identity. For customers some of the advantages of certificates include:

1. A simple way to verify the authenticity of an organisation before providing that organisation with confidential information.

2. The knowledge that if the worse came to the worst, customers can obtain the organisation's physical address and legally registered name, so as to pursue legal action against the company.

For businesses, the advantages of certificates include:

1. A simple way to verify an individual's e-mail address without having to verify it by sending a piece of e-mail. This cuts transaction time, lowering cost. It can also prevent the abuse of e-mail – for example, if an organisation only allows people to sign up for a mailing list by presenting a digital certificate it is not possible for an attacker to maliciously subscribe people to that mailing list without permission.

2. A simple and widely used way for verifying an individual's identity without using usernames and passwords, which are easily forgotten and shared between users.

Steganography

Cryptography is not the only way to hide data. Steganography is another way.This technique is designed to hide data within data or image, for instance. The use of steganography includes watermarking, which, for instance hides copyright information within a watermark by overlaying files not easily detected by the naked eye. This prevents destructive actions and gives copyright protected media extra protection. Using steganography, data can also be hidden within an image.

Copperbelt University
Computer Science Department

Information Management and Network Security

Compiled by: Prof.Dr. Hastings M. Libati

Introduction to Intellectual Property (IP)

Intellectual Property (IP) refers to creations of the mind such as inventions, literary and artistic works, and symbols, names, images, and designs used in commerce.Intellectual Property (IP) is divided into two categories: **Industrial property**, which includes inventions (patents), trademarks, industrial designs, and geographic indications of source; and **Copyright**, which includes literary and artistic works such as books, poems and plays, films, musical works, artistic works such as drawings, paintings, photographs and sculptures, and architectural designs. Rights related to copyright (neighbouring rights) include those of performing artists in their performances, producers of phonograms (sound recordings) in their recordings, and those of broadcasters in their radio and television programs. The innovations and creative expressions of indigenous and local communities are also considered as Intellectual Property and are called **indigenous knowledge**. **Intellectual Property Rights (IPR)**are the rights given to persons over the creations of their minds. They usually give the creator an exclusive right over the use of his or her creation for a certain period of time. The United Nations (UN) established the **World Intellectual Property Organisation (WIPO)** to look into the issues related to Intellectual Property Rights (IPR) in the world. A discussion of the different Intellectual Property Rights (IPR) that can be granted follows below:

Patents

A **patent** is an exclusive right granted for an **invention**, which is a **product** or a **process** that provides, in general, a new way of doing something, or offers a new technical solution to a problem. A patent provides **protection** for the invention to the **owner** of the patent. The protection is granted for a limited period of generally 20 years.Patent protection means that the invention cannot be commercially **made, used, distributed or sold** without the patent owner's **consent.**

A patent owner has the right to decide who may or may not use the patented invention for the period in which the invention is protected. The patent owner **may give permission** to, or **license**, other parties to use the invention on mutually agreed terms. The owner may also **sell** the right to the invention to someone else, who will then become the new owner of the patent. Once a patent expires, the protection ends, and an invention enters the **public domain**, that is, the owner no longer holds exclusive rights to the invention, which becomes available to commercial exploitation by others.

Patents provide **incentives** to individuals by offering them **recognition** for their **creativity** and **material reward** for their marketable inventions. These incentives encourage **innovation,** which could lead to improved living standards.All patent owners are obliged, in return for patent protection, to **publicly disclose information** on their invention in order to **enrich the total body of technical knowledge** in the

world. Such an ever-increasing body of public knowledge promotes **further creativity** and **innovation** in others. In this way, patents provide not only protection for the owner but valuable **information** and **inspiration** for **future generations** of researchers and inventors.

The first step in securing a patent is the filing of a **patent application**. The patent application generally contains the title of the invention, as well as an indication of **its technical field**; it must include the **background** and a **description** of the invention, in clear language and enough detail that an individual with an average understanding of the field could use or reproduce the invention. Such descriptions are usually accompanied by **visual materials** such as drawings, plans, or diagrams to better describe the invention. An invention must, in general, fulfill the following conditions in order for it to be protected by a patent.

1. It must be of **practicaluse**.

2. It must show an element of **novelty**, that is, some **new characteristic** which is not known in the **body of existing knowledge** in its technical field. The invention must show an **inventive step** which could not be deduced by a person with average knowledge of the technical field.

3. Its subject matter must be accepted as **patentable** under the laws of that particular country.

A patent is granted by a **national patent office**like the Patents and Company Registration Office (PACRO) in Zambia or by a **regionaloffice** that does the work for a number of countries such as the African Regional Intellectual Property Organization (ARIPO) in Southern Africa. For anyone wishing to have an invention patented, it is very important to file a patent application before publicly disclosing the details of that particular invention.

It ought to be noted also that computer programs are not patented but they are protected under copyright. However,copyright protection extends only to expressions, not to ideas, procedures, and methods of operation or mathematical concepts.

Copyright and Related Rights (Neighbouring Rights)

Copyright is a legal term describing rights given to creators for their literary and artistic works.The kinds of works covered by copyright include: literary works such as books, poems, plays, reference works, newspapers and computer programs; databases; films, musical compositions, and choreography (sequence of steps, dance routine); artistic works such as paintings, drawings, photographs and sculpture; architecture; and advertisements, maps and technical drawings.The original creators of works protected by copyright, and their heirs, have certain basic rights. They hold the exclusive right to use or authorize others to use the work on agreed terms. The creator of a piece of work can prohibit or authorise:

- its reproduction in various forms, such as printed publication or sound recording;
- its public performance, as in a play or musical work;
- recordings of it, for example, in the form of compact discs, cassettes or videotapes;
- its broadcasting, by radio, cable or satellite;
- its translation into other languages, or its adaptation, such as a novel into a screenplay.

Many creative works protected by copyright require mass distribution, communication and financial investment for their dissemination (for example, publications, sound recordings and films); hence,

creators often sell the rights to their works to individuals or companies best able to market the works in return for payment. These payments are often made dependent on the actual use of the work, and are then referred to as **royalties**.

These economic rights have a time limit of 50 to 70 years after the creator's death. National law may establish longer time-limits. This limit enables both creators and their heirs to benefit financially for a reasonable period of time. Copyright protection also includes moral rights, which involve the right to claim authorship of a work, and the right to oppose changes to it that could harm the creator's reputation, for example.

Rights related to copyright, also called neighbouring rights, do exist and they provide similar, although often more limited and of shorter duration, rights to:

- performing artists (such as actors and musicians) in their performances;
- producers of sound recordings (for example, cassette recordings and compact discs) in their recordings;
- broadcasting organizations in their radio and television programs. Broadcasting organizations are protected as holders of related rights. Broadcast content as such, as opposed to broadcast signals, can also be protected by copyright and related rights.

Copyright and its related rights are essential to human creativity because they give creators incentives in the form of recognition and fair economic rewards. Under this system of rights, creators are assured that their works can be disseminated without fear of unauthorized copying or piracy. This in turn helps increase access to and enhances the enjoyment of culture, knowledge, and entertainment all over the world.Copyright does not depend on official procedures. A created work is considered protected by copyright as soon as it exists. If a person wishes to get permission to use somebody else's copyrighted work then all that should be done is contact the copyright owner and go into an agreement.

As noted earlier, computer programs are protected by copyright, whereas apparatus using computer software or software-related inventions are protected by patent. Furthermore, Copyright law and patent law provide different types of protection. We have already seen that Copyright protection extends only to expressions, and not to ideas, procedures, methods of operation or mathematical concepts, whereas a patent is an exclusive right granted for an invention, which is a product or a process that provides a new way of doing something, or offers a new technical solution to a problem.

Trademark

A **trademark** is a **distinctive sign** which identifies certain goods or services as those produced or provided by a specific person or enterprise. The system of trademarks helps consumers identify and purchase a product or service because its nature and quality, indicated by its **uniquetrademark**, meets their needs.

A trademark provides **protection** to the owner of the mark by ensuring the **exclusive right** to use it to identify goods or services, or to authorize another to use it in return for payment. The period of protection varies from country to country or region to region, but a trademark can be renewed indefinitely beyond the time limit on payment of additional fees.

In a larger sense, trademarks promote initiative and enterprise worldwide by rewarding the owners of trademarks with recognition and financial profit. Trademark protection also hinders the efforts of unfair

competitors, such as counterfeiters, to use similar distinctive signs to market inferior or different products or services. The system enables people with skill and enterprise to produce and market goods and services in the fairest possible conditions, thereby facilitating international trade.

Trademarks may be one or a combination of words, letters, and numerals. They may consist of drawings, symbols, three- dimensional signs such as the shape and packaging of goods, audible signs such as music or vocal sounds, fragrances, or colours used as distinguishing features.

In addition to trademarks identifying the commercial source of goods or services, several other categories of marks exist. **Collective marks** are owned by an **association** whose members use them to identify themselves with a level of quality and other requirements set by the association. Examples of such associations would be those representing accountants (CIMA or ACCA), engineers (IEEE or EIZ), or architects. **Certification marks** are given for compliance with defined standards, but are not confined to any membership. They may be granted to anyone who can certify that the products involved **meet certain established standards.** The internationally accepted **"ISO 9000"** quality standards are an example of such widely-recognized certifications.

In order to register a trademark, an application for registration of a trademark must be filed with the appropriate **national or regional trademark office**. The application must contain a clear reproduction of the sign filed for registration, including any colours, forms, or three-dimensional features. The application must also contain a list of goods or services to which the sign will apply. The sign must fulfill certain conditions in order to be protected as a trademark or other type of mark. It must be distinctive, so that consumers can **distinguish** it as identifying a particular product, as well as from other trademarks identifying other products. It must neither mislead nor deceive customers or violate public order or morality.

Industrial Designs

Industrial designs are what make a product **attractive** and **appealing**and sothey add to the **commercial value** of a product and **increase its marketability**.When an industrial design is protected, this helps to ensure a fair return on investment. An effective system of protection also benefits **consumers and the public at large**, by promoting fair competition and honest trade practices.Protecting industrial designs helps **economic development**, by encouraging creativity in the industrial and manufacturing sectors and contributes to the expansion of commercial activities and the export of national products.

In most countries, an industrial design must be registered in order to be protected. As a general rule, to be registrable, the design must be "**new**" or "**original**". Once a design is registered, the term of protection is generally five years, with the possibility of further periods of renewal up to, in most cases, 15 years.

Depending on the particular national law and the kind of design, an industrial design may also be **protected as a work of art** under **copyrightlaw**. In some countries, industrial design and copyright protection can exist concurrently. In other countries, they are mutually exclusive: once the owner chooses one kind of protection, he can no longer invoke the other.

When an industrial design is registered, the holder receives the right to prevent unauthorized copying or imitation by third parties. This includes the right to prevent all unauthorized parties from making, selling or importing any product in which the design is incorporated or to which it is applied. Because industrial design rights are territorial in nature, this right is limited to the territory for which the design is registered.

Utility Models

In general terms, a utility model is an invention that does not meet all the requirements of patentability but has an industrial use. The inclusion of utility models into the intellectual property system in some African countries has the primary objective of nurturing the rapid evolution of indigenous innovativeness, particularly in small and medium-scale enterprises and among private persons.

Trade Secrets

Trade secrets consist of confidential data, information or compilations used in research, business, commerce or industry. Universities and research and development (R&D) institutions, government agencies, business entities and individuals may own and use trade secrets. The owner of a trade secret must take reasonable measures to maintain its secrecy. The information may include confidential scientific and technical data and business, commercial or financial information not publicly known, that is useful to an enterprise and confers competitive advantage on one having a right to use it. Trade Secrets are also used to protect proprietary portions of technology: formulae, manufacturing processes, business strategies, business management information, customer lists, design concepts etc. Trade secret information may be disclosed or shared under the terms of a confidentiality agreement. Confidential information may be created in sponsored research projects. In that case, the sponsor of the research will generally require the contracted researcher to preserve the secrecy of the information. Trade secrets in the form of know-how may be vital to the working of patented inventions and other innovations. Trade secret information may have considerable value by itself or in conjunction with other forms of intellectual property. A familiar example of a trade secret is the formula for Coca-Cola. If the formula had been patented, it would no longer be a secret, as patent law requires public disclosure of the invention. Anyone who independently and legitimately discovers the secret of the Coca-Cola formula can use that discovery, and the Coca-Cola Company would have no legal means of stopping them.

Academic institutions, however, may have reservations regarding trade secrets protection, arguing that it is hard to reconcile with openness in knowledge-sharing, which is part of the academic mission.

New Plant Varieties

Most universities and research institutions in African countries are involved in research in areas such as crop production, livestock and animal health, forestry, fisheries and crop storage. Research efforts in these areas have led to a number of specific achievements e.g. varieties of many crops, which are capable of producing high yields, more adapted to specific farming systems, resistant or tolerant to certain diseases and pests etc. These varieties are made available to farmers through existing seed services. For each variety, descriptive data are also available which give a brief description of the variety such as the origin (group, common name etc.), agricultural characteristics (farming system, vegetative cycle, adaptability to climatic conditions, yield, grain quality) etc. These data facilitate the choice of a specific variety for a relevant type of farming system. Under the International Convention for the Protection of New Varieties of Plants (UPOV Convention), an intellectual property right, namely **Plant Breeder's Right**, can be granted to a breeder, if the obtained plant variety is considered to be new, distinct, uniform, stable and has a suitable denomination. The plant breeder's right entails that the authorisation of the plant breeder is required before accomplishing some acts in respect of propagating material of the protected variety.

A plant variety is regarded to be distinct if it is clearly distinguishable from other varieties. A variety is stable if its relevant characteristics remain unchanged after repeated propagation. A plant variety is said to be uniform if, subject to the variation that may be expected from the particular features of its propagation, it is sufficiently uniform in its relevant characteristics.A trademark, trade name or other similar indication may be used for the denomination for the purposes of marketing or selling, but the denomination must be easily recognisable.

It should be noted, however, that the UPOV Convention contains important exceptions to the plant breeder's right: The use of protected varieties in subsistence farming does not require the breeder'sauthorisation. Protected varieties are also available without the breeder's authorization for research and plant breeding and Contracting Parties to the Convention may, within certain limits, permit farmers (other than subsistence farmers) to use for propagating purposes, the product of the harvest which they have obtained from the protected variety. By granting a plant breeder's right, the development of new varieties of plants is encouraged in order to contribute to the enhancement of agricultural, horticultural and forestry productivity and, therefore, improvement of incomes and overall development.

The plant breeder's right is granted for a period of not less than 20 years from the date of grant of the right or, in the case of trees, for not less than 25 years.

Folklore and Traditional Knowledge

The emergence of a **global information society** in our world, characterised by the rise of modern information technologies, has also given rise to increasing awareness of the values of traditional knowledge and folklore. The concept of "traditional knowledge" is important for: Environmental conservation; Agriculture and food security; Traditional medicine as a source of primary health care; Indigenous knowledge, in the context of preserving cultural diversity and protecting minority cultures, especially those of indigenous people; the preservation of cultural heritage and sustainable development.

Traditional knowledge and folklore are creations of past societies by people we do not know now and therefore they cannot be assigned to anyone living individual as his or her intellectual property (IP). They need to be protected as national heritage. However, Zambia currently does not have a Law that protects traditional knowledge and folklore.

Integrated Circuits

The layout-designs of integrated circuits are creations of the human mind. They are usually the result of an enormous investment, both in terms of the time of highly qualified experts and financially. There is a continuing need for the creation of new layout-designs, which reduce the dimensions of existing integrated circuits and simultaneously increase their functions. The smaller an integrated circuit, the less the material needed for its manufacture, and the smaller the space needed to accommodate it. Integrated circuits are utilised in a large range of products, including articles of everyday use, such as watches, television sets, washing machines, automobiles etc, as well as sophisticated data processing equipment such as computers.

Geographical Indications

A **geographical indication** is a sign used on goods that have a specific geographical origin and possess qualities, reputation or characteristics that are essentially attributable to that place of origin. Most commonly, a geographical indication includes the name of the place of origin of the goods. Agricultural products typically have qualities that derive from their place of production and are influenced by specific

local factors, such as climate and soil. Geographical indications may be used for a wide variety of products, whether natural, agricultural or manufactured.

An **appellation of origin** is a special kind of geographical indication. It generally consists of a geographical name or a traditional designation used on products which have a specific quality or characteristics that are essentially due to the geographical environment in which they are produced. The concept of a geographical indication encompasses appellations of origin.

The use of geographical indications is not limited to agricultural products. They may also highlight qualities of a product which are due to human factors associated with the place of origin of the products, such as specific manufacturing skills and traditions. That place of origin may be a village or town, a region or a country. For example, “Meissner” is recognized as a geographical indication in many countries for very good quality porcelain ware products made in the German town of Meissner near the city of Dresden.

A geographical indication points to a specific place, or region of production, that determines the characteristic qualities of the product which originates from that place. It is important that the product derives its qualities and reputation from that place. Since those qualities depend on the place of production, a specific "link" exists between the products and their original place of production.

Geographical indications are understood by consumers to denote the origin and the quality of products. Many of them have acquired valuable reputations which, if not adequately protected, may be misrepresented by dishonest commercial operators. False use of geographical indications by unauthorized parties is detrimental to consumers and legitimate producers. Consumers are deceived into believing that they are buying a genuine product with specific qualities and characteristics, when they are in fact getting an imitation. Legitimate producers are deprived of valuable business and the established reputation of their products is damaged.

There is a difference between a trademark and a geographical indication. A trademark is a sign used by an enterprise to distinguish its goods and services from those of other enterprises. It gives its owner the right to exclude others from using the trademark. A trademark will often consist of a fanciful or arbitrary name or symbol. A geographical indication tells consumers that a product is produced in a certain place and has certain characteristics that are due to that place of production. It may be used by all producers who make their products in the place designated by a geographical indication and whose products share specified qualities. Unlike a trademark, the name used as a geographical indication will usually be predetermined by the name of the place of production.

If a geographical term is used as the common designation of a kind of product, rather than an indication of the place of origin of that product, then the term no longer functions as a geographical indication, but instead it becomes a **generic geographical indication**. Where this has occurred in a certain country, then that country may refuse to recognize or protect that term as a geographical indication. For example, the term “cologne” now denotes a certain kind of perfumed toilet water, regardless of whether or not it was produced in the region of Cologne.

Conclusion

We have seen that intellectual property rights (IPR) are like any other property rights. They allow creators, or owners, of patents, trademarks or copyrighted works to benefit from their own work or investment in a creation. One might say why should we promote and protect intellectual property? From

our text, we can deduce that there are several reasons why we should promote and protect intellectual property:

1. The progress and well-being of humanity rest on its capacity to create and invent new works in the areas of technology and culture.

2. The legal protection of new creations encourages the commitment of additional resources for further innovation.

3. The promotion and protection of intellectual property brings about economic growth, creates new jobs and industries, and enhances the quality and enjoyment of life. An efficient and equitable intellectual property system can help all countries to realize intellectual property's potential as a catalyst for economic development and social and cultural well-being. The intellectual property system helps strike a balance between the interests of innovators and the public interest and thereby providing an environment in which creativity and invention can flourish, for the benefit of all mankind.

   One could still ask to say then how does the average person benefit from intellectual property? But we have seen that intellectual property rights reward creativity and human endeavour, which also fuel the progress of humankind. Examples that can be given here are plenty. For example, the lucrative software industry, publishing, music and film recording industries bring immense pleasure to people across our world. These industries would not exist without copyright protection. Furthermore, without the rewards provided by the patent system, researchers and inventors would have little incentives to continue producing better and more efficient products for consumers. Consumers themselves would have no means to confidently buy products or services without reliable, international trademark protection and enforcement mechanisms to discourage counterfeiting and piracy. We see here that the average person actually benefits heavily from intellectual property.

CopperbeltUniversity
Computer Science Department

Information Management and Network Security

Compiled by: Prof.Dr. Hastings M. Libati

Fault Tolerance and Disaster Recovery

Computers are not perfect. They can and do have problems. These problems range from small errors to total system failure. Errors and failures can be the result of environmental problems, hardware and software problems, hacking (malicious, unauthorised use of a computer or a network), as well as natural disasters.

In all cases, one can take measures to minimise the impact of computer and network problems. These measures fall into two major categories: *fault tolerance* and *disaster recovery*. Fault tolerance is the ability of a computer or a network system to respond to a condition automatically, usually resolving the problem, and thus reducing the impact of such a condition on the system. If fault-tolerant measures have been implemented, it is unlikely that a user would even know that a problem existed. Disaster recovery is the ability to get a system functional after a total system failure in the least amount of time. Strictly speaking, if enough fault tolerance measures are in place, one shouldn't need disaster recovery. However, both these methods (measures) are important and are implemented on most, if not all networks.

Assessing Fault Tolerance and Disaster Recovery Needs

Before implementing fault tolerance or disaster recovery, one should determine how critical one's systems are to daily business operations. Additionally, one should determine how long each system could afford to be non-functional (down). Making these determinations will dictate which, fault tolerance and disaster recovery methods one implements and to what extent. The more vital the system, the greater the lengths (and, thus, the greater the expense) one should go in order to protect it from downtimes. Lesser critical systems may call for simpler measures. In terms of how fault tolerance and disaster recovery are implemented, sites can be described as hot, warm or cold.

Hot Sites

In a hot site, every computer system and piece of information has a redundant copy (possibly multiple redundancies). This level of fault tolerance is used when systems must be up 100 percent of the time. Hot sites are strictly fault-tolerant implementations, not disaster recovery implementations (as no downtime is allowed). Budgets for this type of fault-tolerant implementations are typically large.

In a system that has 100 percent redundancy, the redundant system (s) will take over the failed system without any downtime. The technology used to implement hot sites is called *clustering*, which is the process of grouping multiple computers in order to provide increased performance and fault tolerance.

Clustering Technologies

Although server-class computers are commonly clustered, workstations are normally not clustered because they are simple and cheap to replace. Each computer in the cluster is connected to the other computers in the cluster by high-speed, redundant links. Such high-speed links are usually implemented using fibre-optic cables. Each computer in the cluster runs special clustering software that makes the cluster of computers appear as a single entity to clients. There are two levels of cluster service. These are called failover clustering and true clustering.

Failover Clustering

A failover cluster includes two entities (usually server class-machines). The first is the active device (the device that responds to network requests), and the second is the failover device. The failover device is an exact duplicate of the active device, but it is inactive and connected to the active device by a high-speed link. The failover device monitors the active device and its condition by using what is known as a heartbeat. A heartbeat is a signal that comes from the active device at a specified interval. If the failover device does not receive a heartbeat from the active device in the specified interval, then the failover device considers the active device inactive, and the failover device comes on-line (becomes active) and becomes the active device.

When the previously active device comes back on-line, it starts sending out the heartbeat. The failover device, which currently is responding to requests as the active device, "hears" the heartbeat and detects that the active device is now back on-line. At this point, the failover device then goes back into standby mode and starts listening to the heartbeat of the active device again. In a failover cluster, both server-class computers must be running failover-clustering software. The failover-clustering software package provides the required failover functionality. Some of the advantages of the failover clustering approach to fault tolerance are given below:

(a) Resources are almost always available. This approach ensures that the network services that should be provided are actually available as much as 99 percent of the time. Each network service and all data are exactly duplicated on each device, and when one device experiences problems, the other device takes over for virtually uninterrupted service.

(b) It is relatively inexpensive when compared with true clustering.

Below are some disadvantages of the failover clustering approach to fault tolerance:

(a) There is only one level of fault tolerance. This technology works great if the active device fails, but if the failover device remains working. However, if both of them fail at the same time, then the situation becomes really desperate.

(b) There is no load balancing. Server-class computers in a failover-clustering configuration are in either active or standby mode. There is no balancing of network service load across both server machines in the cluster. The active server computer responds to network requests, and the failover server computer simply monitors the active server machine, wasting its processor resources.

(c) During cutover time, the server computer can't respond to requests. Failover clusters take anywhere from a few seconds to a few minutes to detect and recover from a failed server computer. This is called cutover time. During cutover time, the server computer can't respond to

network client requests, so the server computer is effectively down. This time is indeed short, but, nevertheless, clients can't get access to their services during cutover time.

(d) Hardware and software must be exactly duplicated. In most failover configurations, the hardware for both the active and the failover devices must be identical. If it is not, then the transition of the failover device to active device may be hindered. These differences may even cause the failover system to fail. This is a disadvantage because it involves checking all aspects of the hardware. For instance, for server machines, this means that disk types and sizes, NICs, processor speed and type and RAM must be identical.

True Clustering

True clustering differs from failover clustering in two major ways:

(a) It supports multiple devices.

(b) It provides load balancing.

In true clustering (also known as multiple server machine clustering), multiple server computers (or any network devices) act together as a kind of super server devices. True clusters must provide load balancing. For example, 20 server-class computers can act as one big server-class computer. All network services are duplicated across all server-class computers. Each server-class computer is connected to the other server computers through high-speed, dedicated links. If one server-class computer in the cluster malfunctions, the other server computers automatically take over the burden of the failed server-class computer. When the failed server machine comes back online, it resumes responding to requests as part of the cluster.

With the right software available, this technology can provide greater than 99 percent availability for network services hosted by the cluster. The advantages that can accrue with true clustering are:

(a) There is more than 99 percent availability for network services. With multiple server-class computers, the impact of a single server computer, or even more than one server computer, in the cluster going down is minimised because other server computers take over the functionality.

(b) It offers increased performance. Because each server computer is taking part of the load of the cluster; much higher total performance is possible.

(c) There is no cutover time. Because multiple server computers are always responding to network requests, true clusters don't suffer from cutover time even when a server computer goes down. The remaining server computers do receive an increased load, and clients may see a *Server busy* or *Not found* error messages if they should, by some chance, try to communicate with the server that is going down. But if such a user tries the operation again, then one of the remaining server computers will respond to the request.

(d) It provides for replication. If the software in use supports replication then a few server computers can be located off site, just in case the main site is destroyed by fire, floods or other disasters. Because there is a replica (copy) of all data in a different location, this technology is also known as *replication*.

The disadvantages of true clustering are given below:

(a) The more server-class computers, the more complex the cluster. As more server computers are added to the cluster to increase performance, the more the complexity is increased. For this reason, most clustering software is limited to a maximum of 64 server computers. As technology develops, this limit will increase. The minimum number of server machines in a cluster is 12.

(b) It is much more expensive. Because of the hardware involved and the complexity of the clustering software, true clustering requires a serious financial commitment.

Warm Site

In a warm site (also called nearline site), the network services and data are available most of the time (more than 85 percent of the time). The data and services are less critical than those in a hot site. With hot-site technologies, all fault tolerance procedures are automatic and are controlled by the network operating system. Warm site technologies require a little more administrator intervention, but they aren't as expensive.

The most commonly used warm-site technology is a duplicate server computer. A duplicate server computer is a machine that is currently not being used and is available to replace any server computer that fails. When a server computer fails, the administrator installs the new server computer and restores the data; in this case, the network services are available to users with a minimum of downtime. The administrator could then send the failed machine out for repairs. Once the repaired computer comes back, it becomes the spare server computer and is available whenever another server machine fails.

Using a duplicate machine is a disaster recovery method because the entire server machine is replicated, but in a shorter time than if all the components had to be ordered and configured at the same time of the system failure. The major advantage of using duplicate server machines rather than clustering is that it is less expensive. A single duplicate server machine costs much less than a comparable clustering solution. The disadvantages of using duplicate server computers are given below:

(a) There is need to keep current backups. Because the duplicate server machine relies on current backup, backups have to be made every day, which is time-consuming. To stay as current as possible, some organisations run continuous backups.

(b) Data can easily be lost. If a server machine fails in mid-afternoon and the backup was run the evening before, then any data that was placed on the server machine since the last backup will be lost.

Cold Site

A cold site cannot guarantee server uptime. Generally speaking, cold sites have little or no fault tolerance and rely completely on efficient disaster recovery methods to ensure data integrity. If a server machine fails, the ICTs personnel will do their best to recover and fix the problem. If a major component needs to be replaced, the server machine stays down until the component is replaced. Errors and failures are handled as they occur. Apart from regular system backups, no fault tolerance or disaster recovery methods are implemented. This type of site has one major advantage: It is the cheapest way to deal with errors and system failures. No extra hardware is required, except the hardware required for backing up. However, downtimes may be huge in this method.

Power Management

A key element of any fault tolerance plan is a power management strategy. Electricity powers the network, switches, hubs, client computers and server-class computers. Variations in power can cause problems ranging from a reboot after a short loss of service to damaged equipment and data. There are devices that can be used to help protect sensitive systems from the dangers of lightning strikes, dirty (uneven) power, and accidental power cable disconnections. These devices include surge protectors, standby power supplies (SPS), uninterruptible power supplies (UPS) and line conditioners. What one uses depends on how critical one's system is (whether one decides that it is a hot, warm or cold site). At a minimum, one should connect individual workstations to surge protectors, and network hardware and server-class machines should use uninterruptible power supplies or line conditioners. Critical operations could even go a step further by acquiring a backup generator to provide long-term supplemental power to all systems.

Disk System Fault Tolerance

A hard disk is a temporary storage device, and every hard disk will eventually fail. The most common problem is a complete hard-disk failure (also known as a hard-disk crash). When this happens, all stored data may be irretrievable. Therefore, if we want our data to be accessible 90 to 100 percent of the time (as with hot and warm sites), then we need to use some method of disk fault tolerance. Typically, disk fault tolerance is achieved through disk management technologies such as mirroring, striping, and duplexing drives, and provides some level of data protection. As with other methods of fault tolerance, disk fault tolerance means that a disk system is able to recover from an error condition of some kind. The methods that provide fault tolerance for hard disk systems include: mirroring, duplexing, data striping and redundant array of independent (or inexpensive) disks (RAID).

Disk Mirroring

Mirroring a drive means designating a hard-disk drive in the computer as a mirror or duplicate to another, specified drive. The two drives are attached to a single disk controller. This disk fault tolerance feature is provided by most network operating systems (NOSs). When the network operating system writes data to the specified drive, the same data is also written to the drive designated as the mirror. If the first drive fails, the mirror drive is brought on-line into operation and because it has a duplicate of the information contained on the specified drive, the users won't know that a disk drive in the server machine has failed. The network operating system notifies the administrator that a failure has occurred. The downside of this method is that if the disk controller fails, neither drive will be available. The disks used in a mirrored system do not necessarily need to be identical, but it helps if they are. Both disks must have the same amount of free space in order to allow a mirror to be formed. For example, if we have two 4GB hard disks; where one has 3GB free space and the other has 2GB free space, then we can create one 2GB mirrored system. Please, note that mirroring is also regarded as an implementation of RAID level 1.

Disk Duplexing

As with mirroring, duplexing also saves data to a mirror drive. In fact, the only major difference between duplexing and mirroring is that duplexing uses two separate disk controllers (one for each disk). Thus duplexing provides not only a redundant disk, but also a redundant disk controller. Duplexing provides fault tolerance even if one of the controllers fails. Please, note that mirroring is also regarded as an implementation of RAID level 1.

Disk Striping

From a performance point of view, writing data to a single drive is slow. When three drives are configured as single volume, information must fill the first drive before it can go to the second and fill the second before filling the third. If one configures that volume to use disk striping, one will see a definite performance gain. Disk striping breaks up the data to be saved to disk into small portions and sequentially writes the portions to all disks simultaneously in all areas called stripes. These stripes maximise performance all of the read/write heads are working constantly. Once again please, note that data is broken down into sections and that each section is sequentially written to a separate disk.

Redundant Array of Independent (or Inexpensive) Disks (RAID)

RAID is a technology that uses an array of less expensive hard disks instead of one enormous hard disk and provides several methods for writing data to those disks to ensure redundancy. Those methods are described as levels, and each level is designated for a specific purpose.

The discussion of the different levels of RAID will be carried out by our lovely students.

Backup Considerations

Although we can never be completely prepared for a natural disaster or human error that can bring down our network, we can at least make sure that we have a solid backup plan in place to minimise the impact of lost data. If even the worst happens, we don't have to lose days or weeks of work, provided that we have a solid backup plan in place. A backup plan is a set of guidelines and schedules that determine which data should be backed up and how often. A backup plan includes such information as:
Where to backup, where the data should be backed up (on what type of storage media), when to backup, how often to backup, who should be responsible for backups, where storage media containing backups should be stored, how often to test the backups, the procedure to follow in the case of data loss.

Backup Media Options

When we backup our network's data, we must have something (storage medium) on which to store that data, which is called the backup medium. There are several options here, which include:

- Small-capacity removable disks
- Large-capacity removable disks
- Removable optical disks
- Magnetic tape (The oldest of them all.)

Each of these options has advantages and disadvantages and the lovely student is humbly requested to give this a thorough thought.

Backup Utilities

A backup utility is a software program that can archive the data on a hard disk to a removable medium, for instance. Backup utilities can compress data before they store it, making it more efficient to use a backup program to archive data than to simply copy the backup medium.

Most operating systems include backup utilities, but these are usually simple programs that may lack the advanced features offered by fully-fledged, third-party programs such as Seagate Backup Exec, Symantec Backup Exec (enterprise wide back), etc. So, though operating system backup utilities can do the job to

some extent, one is better advised to purchase a third-party product for backup purposes – especially if one needs a complete set of backup management features such as scheduling, compression, etc.

Backup Types

After one has chosen the backup medium and also the backup utility to use, one must then decide what type of backup to run. The backup types vary depending on how much data needs to be backed up each time a backup is run and on how many storage media it takes to restore data after a complete system crash. There are three types of backups and these are the:

- Full
- Differential and the
- Incremental backup.

Full Backup

In a full backup, all the network data is backed up. This type of backup is straightforward because one simply instructs the software, which server-class machines to backup and where to backup the data and then the backup process starts. If a restore has to be done, for instance after a system crash, one has only one set of backup media to restore from (the number of storage media that it took to backup the network data). In this case, one simply inserts the most recent full backup into the drive and the restore process can begin. Depending on the size of an organisation, a full backup could take a good amount of time to complete. In any case, one will usually be backing up the same data each day, for example.

Differential Backup

In a differential backup strategy, a single, full backup is done typically once a week. Every night for the next six nights, the backup utility backs up all files that have changed since the last full backup (the actual differential backup). After a week's worth of differential backups, another full backup is done, starting the cycle all over again. With differential backups, one uses a maximum of two backup sessions to restore a file or group of files.

This is so because the backup utility keeps track of which files have been backed up through the use of the archive bit, which is simply an attribute that indicates the file's status with respect to the current backup type. The archive bit is cleared for each file backed up during the full backup. After that, anytime a program opens and changes a file, the network operating system sets the archive bit, indicating that the file has changed and needs to be backed up. Then each night, in a differential backup, the backup program backs up every file that has its archive bit set, indicating that the file has changed since the last full backup. The archive bit is not touched during each differential backup.

When restoring the data after a complete system failure, one must restore two sets of backup storage media. These are the last full backup and the most current differential backup. A full restoration may take longer, but each differential backup takes much less time than a full backup. This type of backup is used when the amount of time that is available to perform a system backup each day is smaller during the working week but larger over the weekend. The backup window is a term that is used to refer to the period when time to perform a backup is available.

Incremental Backup

In an incremental backup, a full backup is used in conjunction with daily partial backups to backup the entire data on a backup server-class machine, thus reducing the amount of time it takes for a daily backup. With an incremental backup, the weekly full backup takes place as it does during a differential backup and the archive bit is cleared during the full backup. The incremental daily backups, back up only the data that has changed since the last backup (not the last full backup). The archive bit is cleared each time a backup occurs. With this method, only the files that have changed since the previous day's backup are backed up. Each day's backup is a different size because a different number of files are modified each day.

This method provides the fastest daily backups for networks whose daily backup window is extremely small. The restores made with this method after a system failure take the longest of the three methods. The full backup set is restored plus every backup from the day of the failure back to the preceding full backup.

Rotating Removable Backup Storage Media

When using removable storage media for backups then rotating the backup storage media is the most practical way to manage the backups. These removable backup storage media could be rotated on weekly, monthly or even yearly basis. The most famous scheme in this case is the so-called Grandfather-Father-Son system of backups.

In addition to daily, weekly and yearly backups, some organisations, for archival purposes, do an end-of-year backup, which is then kept offsite in long-term storage. This is done in order to keep a record of the year's data.

Virus Protection

A virus is a program that causes malicious change in a computer system and makes copies of itself. Sophisticated viruses encrypt and hide themselves to avoid detection. There are many different viruses that a computer can catch. Known viruses are referred to as being "in the wild." Research laboratories and universities study viruses for commercial and academic purposes. These viruses are known as being "in the zoo" or not out in the wild. The number of viruses that are in the wild keeps on increasing. Viruses can cause a lot of problems and destruction. The types of viruses vary, but the approach to handling them does not change. One needs to install virus protection software on all computer equipment. This is similar to vaccinating an entire family. In this regard, client computers, server-class computers and firewall machines etc, must all have virus protection software (anti-virus software), even if they never connect to the network. This is because such computers may still get viruses from movable storage devices like flash disks or Internet downloads (e.g., via modems). Always keep your anti-virus software up-to-date. Viruses are not the only malicious types of software that exist. Please, ensure that you are prepared for these other malicious pieces of software as well. These include Trojan horses, bacteria, etc.

Software Patches

Patches, fixes, service packs and updates are all the same thing – free software revisions. These are intermediary solutions until a new version of the product is released. A software patch may solve a particular problem, as does a security patch, or change the way a system works, as does an update. One can apply a so-called hot patch without rebooting a computer; in other cases, applying a patch requires that server-class computer goes down.

Because patches are designed to fix problems, it would seem that one would want to download the most current patches and apply them immediately. That may not always be the best way to go. Patches can sometimes cause problems with existing, older software. Different philosophies exist regarding the application of newer patches. The philosophy is to keep the systems only as up-to-date as is necessary to keep them running. Our colleagues in the industry refer to this as the "if it isn't broken, don't fix it" approach. After all, the point of a patch is to fix a piece of software. So why fix it if it isn't broken? The other philosophy or approach is to keep the software as up-to-date as possible because of the additional features that a patch will sometimes provide. Please, always ensure that you obtain your patches from the manufacturer.