Module 1

Attacks on Computers and Computer Security:

Need for Security, Security Approaches, Principles of Security Types of Attacks

1.1 The Need for Security

1.1.1 Basic Concepts

- Initial computer applications had no, or very little security as the importance of data was not realized then.
- When computer applications were developed to handle financial and personal data, the need for security arose. With this realization, security began to gain prominence and security mechanisms began to evolve.
- Examples of security mechanisms:
  - Provide a user id and password to every user, and use that information to authenticate a user
  - Encode information stored in the databases, so that it is not visible to users who do not have the right permissions.
- Organizations employed their own security mechanisms to provide basic security. As technology improved, newer applications began to be developed and the basic security measures were not sufficient.
- Further with the evolution of the biggest computer network, Internet, the need for right security policy, technology implementations became very important.
- Example of information traveling from Client to server over the internet.

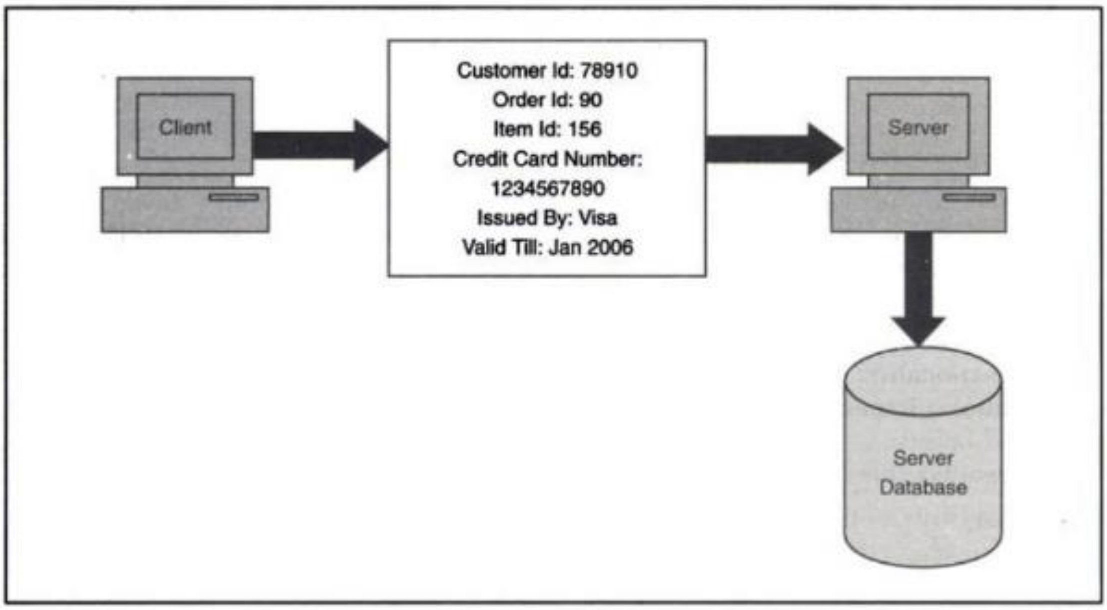


Fig 1.1 Information travelling from a client to a server over the internet

- From the user's computer, the user details such as user id, order details such as order id and item id and payment details such as credit card information travel across the

Internet to the merchant’s server. The merchant’s server stores these details in its database.

- The various security holes in this are:
    - An intruder can capture the credit card details as they travel from the client to the server.
    - Once the merchant receives the credit card details and validates them so as to process the order and later obtain payments, the merchant stores the credit card details into its database. An attacker can succeed in accessing this database and gain access to all the credit card numbers stored there.

1.1.2 Modern Nature of attacks

The salient features of the modern nature of attacks are:

- **Automating attacks:** Humans dislike repetitive and difficult tasks. Automating them can cause destruction more rapidly. Rather than producing fake currency on a mas scale, modern thieves will excel in stealing a very low amount from million bank accounts in a matter of a few minutes.
- **Privacy concerns:** Collecting information about people and later misusing it is turning out to be a huge problem. The data mining applications gather, process and tabulate all sorts of details about individuals. People can illegally sell this information.
- **Distance does not matter:** Thieves would earlier attack banks, as banks had money. These days Money is in digital form and moves around using computer network. It is easier for modern thief to attempt an attack on the computer system of the bank, sitting at home.

1.2 Security Approaches

1.2.1 Trusted systems

- A trusted system is a computer system that can be trusted to a specified extent to enforce a specified security policy.
- Trusted system uses the term reference monitor, an entity at the logical heart of the computer system which is responsible for all decisions across controls.
- The reference monitor should be tamperproof, always be invoked and small enough so that it can be independently tested.
- The mathematical foundation for trusted systems was provided by two independent, yet interrelated works. In 1974, a technique called as Bell-LaPadula model was devised which was a highly trustworthy computer system designed as a collection of objects (files, disks and printers) and subjects (users, processes or threads)

1.2.2 Security Models

An organization can take several approaches to implement its security model. The various approaches are:

a) **No security**: This is the simplest model with no security at all.

b) **Security through obscurity**: In this model, a system is secure simply because nobody knows about its existence and contents. This approach cannot work for too long, as there are many ways an attacker can come to know about it.

c) **Host security**: In this scheme, the security for each host is enforced individually. This is a safe approach, but the complexity and diversity of modern sites/organizations makes the task harder and difficult to scale.

d) **Network security**: Host security is tough to achieve as organization grows and becomes more diverse. In this technique, the focus is to control network access to various hosts and their services, rather than individual host security. This is a very efficient and scalable model.

1.2.3 Security Management Practices

Good security management practices always have a good security policy which takes care of 4 key aspects.

1) **Affordability:** How much money and efforts does this security implementation cost?
2) **Functionality**: What is the mechanism of providing security?
3) **Cultural issues**: Does the policy gel well with people's expectations, working style and beliefs?
4) **Legality**: Does the policy meet the legal requirements?

Once a security policy is in place, the following points should be ensured.

- Explanation of the policy to all concerned.
- Outline everybody's responsibilities.
- Use simple language in all communications.
- Accountability should be established.
- Provide for exceptions and periodic reviews.

1.3 Principles of Security

The four chief principles of security are:

1) Confidentiality

2) Authentication

3) Integrity

4) Non repudiation

Two more principles that are linked to the overall system are:

5) Access control

6) Availability

1) Confidentiality

- The principle of confidentiality specifies that only the sender and the intended recipient(s) should be able to access the contents of a message.
- Confidentiality gets compromised if an unauthorized person is able to access a message. Example of compromising the confidentiality of a message is shown in Fig 1.2.

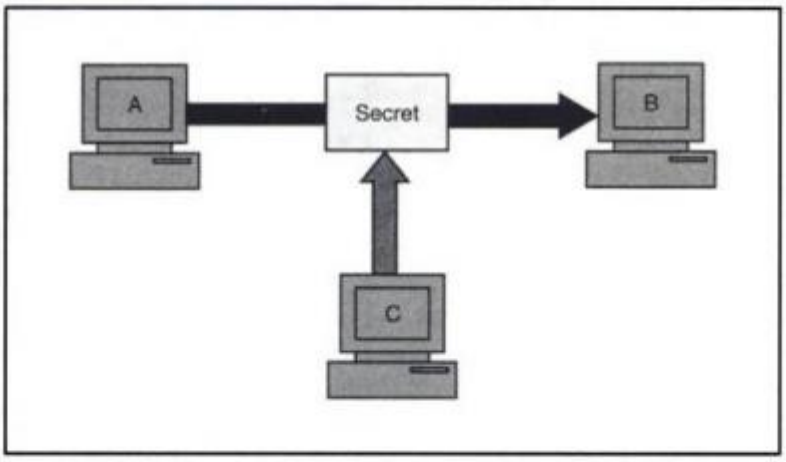


Fig 1.2 Loss of confidentiality

- Here the user of computer A sends a message to the user of computer B. Another user C gets access to this message, which is not desired, and therefore, defeats the purpose of confidentiality.
- **Example:**- A confidential email message sent by A to B, which is accessed by C without the permission or knowledge of A and B. This type of attack is called as **interception.** "Interception causes loss of message confidentiality"

2) Authentication:

- Authentication establishes **proof of identities**.
- The authentication process ensures that the origin of an electronic message or document is correctly identified.
- For instance, suppose that user C sends an electronic document over the Internet to user B, posing as user A. How would user B know that the message has come from user C, who is posing as user A.

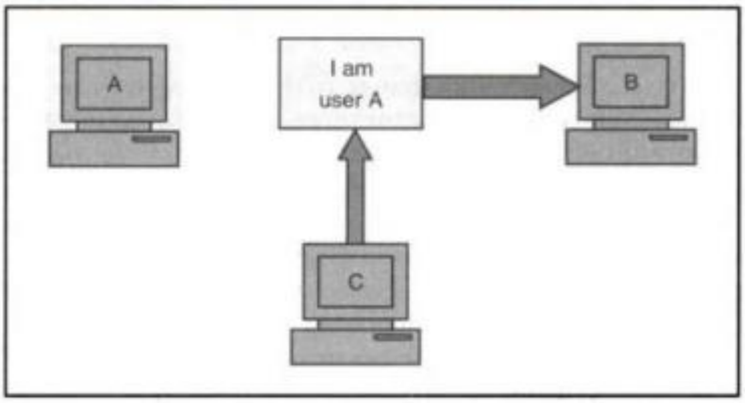


Fig 1.3 Absence of authentication

- **Example:** User C posing as user A, sends a funds transfer request (from A's account to C's account) to bank B. The bank will transfer the funds from A's account to C's account, thinking that user A has requested for the funds transfer. This type of attack is called as **fabrication.**
- "Fabrication is possible in the absence of proper authentication mechanisms"

3) Integrity:

- When the contents of a message are changed after the sender sends it, but before it reaches the intended recipient, the integrity of the message is lost.
- **Example:** Suppose you write a cheque for $100 to pay for the goods bought from the store, but in the account statement it is observed that the cheque resulted in a payment of $1000! This is the case of loss of message integrity.
- Fig 1.4 demonstrates loss of integrity. User C tampers (modifies) a message originally sent by user A, which is actually destined for user B.
- User C somehow manages to access the message, change its contents and send the changed message to user B. Neither A nor B knows that the contents of the message were changed after user A had sent it. This type of attack is called as **modification.**
- "Modification causes loss of message integrity"

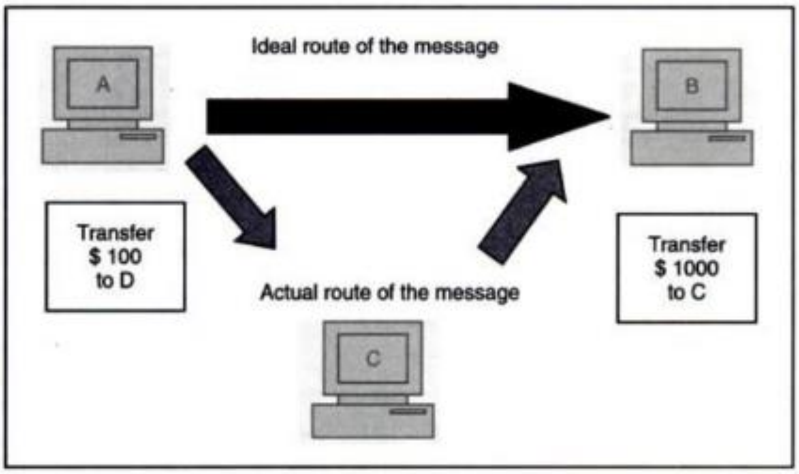


Fig 1.4 Loss of integrity

4) Non repudiation:

- There are situations where a user sends a message and later refuses that the message was sent. This is repudiation (refuse to accept).
- **Example:** User A could send a fund transfer request to bank B over the internet. After the bank performs the funds transfer as per A's request, A could claim that he never sent the fund transfer request to the bank.
- Thus, A repudiates, or denies the fund transfer instruction.
- The principle of non-repudiation defeats such possibilities of denying something, having done it.

"Non repudiation does not allow the sender of a message to refuse the claim of not sending that message".

5) Access control:

- The principle of access control determines who should be able to access what.
- For instance, we should be able to specify that user A can view the records in a database, but cannot update them. However, another user B might be allowed to make updates as well. An access control mechanism can be set up to ensure this.
- Access control is broadly related to two areas: role management and rule management.
- **Role management** concentrates on the user side (which user can do what)
- **Rule management** focuses on the resources side (which resource is accessible, and under what circumstances).
- Based on the decisions taken here, an access control matrix is prepared, which lists the users against a list of items they can access (it can say that user A can write to file X, but can only update files Y and Z).
- An **access control list (ACL)** is a subset of an access control matrix.
- "Access control specifies and controls who can access what."

6) Availability:

- The principle of availability states that resources (information) should be available to authorized parties at all times.

- **Example:** Due to the intentional actions of another unauthorized user C, an authorized user A may not be able to contact a server computer B as shown in fig 1.5.
- This would defeat the principle of availability. Such an attack is called as **interruption.**
- **"**Interruption puts the availability of resources in danger"

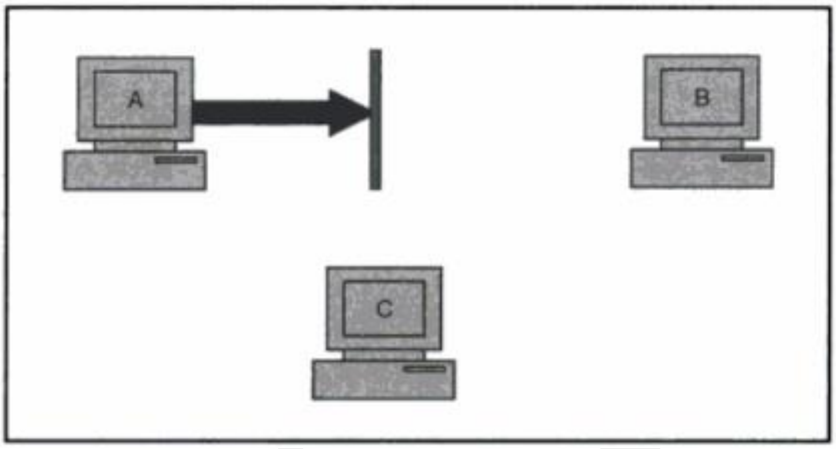


Fig 1.5 Attack on availability

7) Ethical and Legal issues:

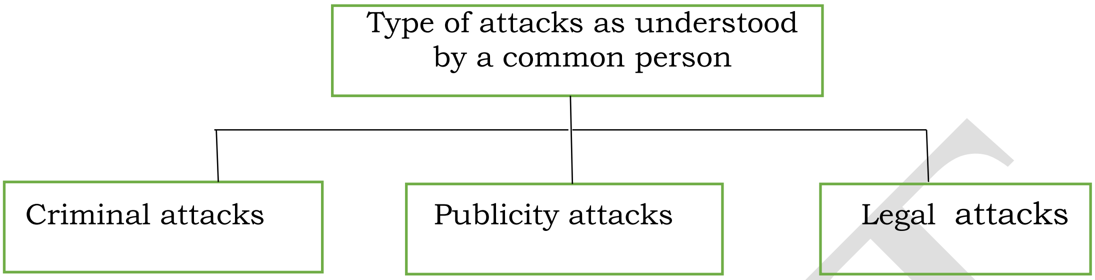
- The ethical issues in security systems are classified into four categories:
  - Privacy – deals with the right of an individual to control personal information
  - Accuracy – deals about the responsibility for the authenticity, fidelity and accuracy of information.
  - Property – talks about the owner of the information and about who controls access.
  - Accessibility – deals with the issue of type of information an organization has the right to collect.
- While dealing with legal issues, there is a hierarchy of regulatory bodies that govern the legality of information security which can be classified as follows:
  - International -e.g International Cybercrime Treaty
  - Federal – e.g. FERPA, Patriot Act
  - State – e.g. UCITA
  - Organization – e.g. Computer use policy

1.4 Types of attacks

Attacks can be classified into two views : A common man's view and technological view.

1.4.1 Attacks: A general view

Attacks can be classified into three categories:



- **Criminal attacks –**

In this, the aim of attackers is to maximize financial gain by attacking computer systems. Some of the criminal attacks are listed below:

Attack	**Description**
Fraud	Modern fraud attacks concentrate on manipulating some aspects of electronic currency, credit cards, electronic stock certificates, cheques, letters of credit, purchase order, ATMs etc.
Scams	Some forms of scams are sales of services, auctions, multi- level marketing schemes, general merchandise and business opportunities etc. People are tempted to send money in return of great profits, but end up losing their money.
Destruction	The main motive behind these attacks is some sort of grudge. Example: Some unhappy employees attack their own organization; terrorists strike at bigger levels
Identity theft	An attacker does not steal anything from a legitimate user, instead he becomes that legitimate user! Example, it is much easier to manage to get the password of someone else's bank account or to actually be able to get a credit card on someone else's name. That privilege can be misused until it gets detected.
Intellectual property theft	Intellectual property theft ranges from stealing companies' trade secrets, databases, digital music and videos, electronic documents and books, Identity theft, Intellectual property theft software and etc.

Brand theft	It is quite easy to set up fake Web sites that look like real Web sites. It is difficult for a common user to know if she is visiting the real Bank site or an attacker's site? Innocent users end up providing their secrets and personal details on these fake sites to the attackers. The attackers use these details to then access the real site, causing an identity theft.

- **Publicity Attacks**
  - Occurs because the attackers want to see their names appear on television news channels and newspapers for publicity. These types of attackers are usually not hardcore criminals.
  - They are people such as students in universities or employees in large organizations, who seek publicity by adopting a novel approach of attacking computer systems.

- **Legal attacks**
  - This form of attack is quite novel and unique. The attacker tries to make the judge or jury doubtful about the security of a computer system.
  - The attacker attacks the computer system and the attacked party (Bank or organization) manages to take the attacker to the court. The attacker tries to convince the judge that there is inherent weakness in the computer security system and exploits the weakness of the judge.

1.4.2 Attacks: A Technical view

The types of attacks on computers and network systems can be classified into two categories:
(a) Theoretical concepts behind these attacks
(b) Practical approaches used by the attackers.

a) Theoretical Concepts

The principles of security face threat from various attacks. These attacks are classified into four categories, as mentioned namely:

- **Interception -**
  - This attack results from violating confidentiality.
  - It means that an unauthorized party has gained access to a resource. The party can be a person, program or computer-based system.
  - Example: Copying of data or programs and listening to network traffic.

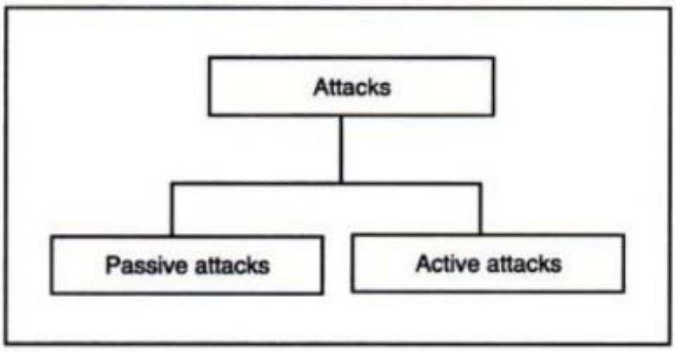
- **Fabrication -**
  - This attack results from violating authentication.
  - This involves creation of illegal objects on a computer system.
  - Example: The attacker may add fake records to a database.

- **Modification –**
  - This attack results from violating Integrity. The attacker may modify the values in a database.

- **Interruption**
  - This attack results from violating availability.
  - The resource becomes unavailable, lost or unusable.
  - Example: Causing problems to a hardware device, erasing program, data or operating system components.

These attacks are further grouped into two types:

- Passive attacks
- Active attacks



Passive attacks

- Passive attacks are those, wherein the attacker indulges in eavesdropping or monitoring of data transmission.
- The attacker aims to obtain information that is in transit.
- The term passive indicates that the attacker does not attempt to perform any modifications to the data.
- Passive attacks are harder to detect.
- The general approach to deal with passive attacks is to think about prevention, rather than detection or corrective actions.

Passive attacks do not involve any modifications to the contents of an original message.

Passive attacks can be further classified into two sub-categories. These categories are:

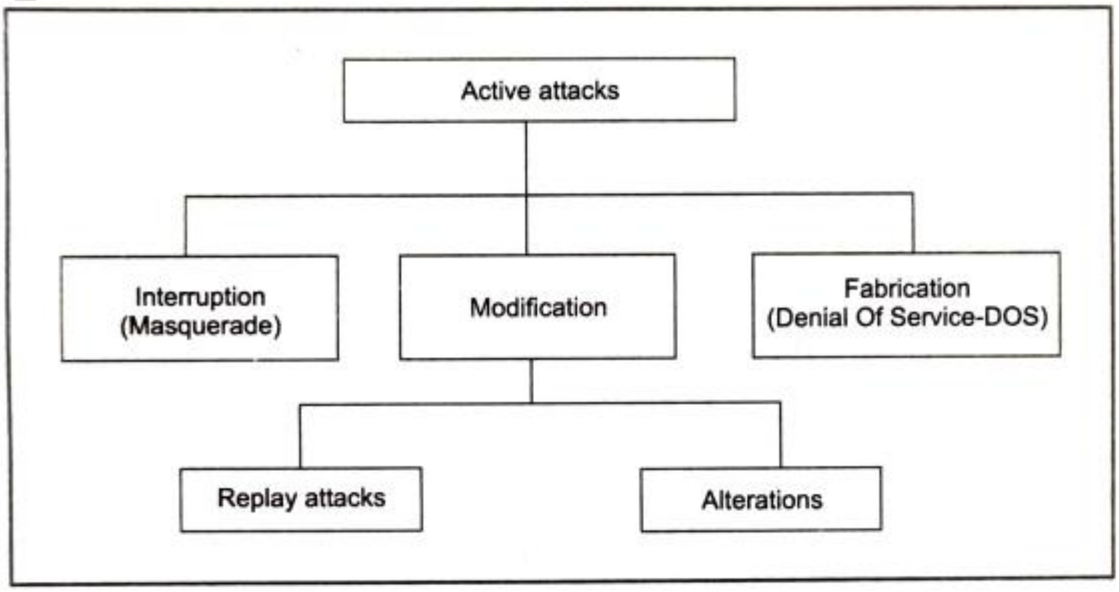
- Release of message contents
- Traffic analysis.

- **Release of message contents:**

- When a confidential email message is sent, it is desired that only the recipient is able to access it. Otherwise, the contents of the message are released against our wishes to someone else.
- Using certain security mechanisms, we can prevent release of message contents. For example, we can encode messages using a code language, so that only the desired parties understand the contents of a message, because only they know the code language.
- However, if many such messages are passing through, a passive attacker could try to figure out similarities between them to come up with some sort of pattern that provides the attacker some clues regarding the communication that is taking place.
- Such attempts of analysing (encoded) messages to come up with likely patterns are the work of the **traffic analysis attack.**

Active attacks

- The active attacks are based on modification of the original message in some manner or the creation of a false message. These attacks cannot be prevented easily.
- They can be detected with some effort and attempts can be made to recover from them. These attacks can be in the form of **interruption, modification and fabrication.**
- In active attacks, the contents of the original message are modified in some way.
- Trying to pose as another entity involves **masquerade (interruption)** attacks.
- Modification attacks can be classified further into **replay attacks** and **alteration of messages**.
- Fabrication causes **Denial Of Service (DOS)** attacks.
- This Classification can be shown as follows:


Active attacks
Interruption (Masquerade)
Modification
Fabrication (Denial Of Service-DOS)
Replay attacks
Alterations

- **Masquerade** is caused when an unauthorized entity pretends to be another entity.
  - Example: User C might pose as user A and send a message to user B. User B might be led to believe that the message indeed came from user A. In masquerade attacks, an entity poses as another entity.
  - Example, the attack may involve capturing the user's authentication sequence (e.g. user ID and password). Later those details can be used to gain illegal access to the computer system.

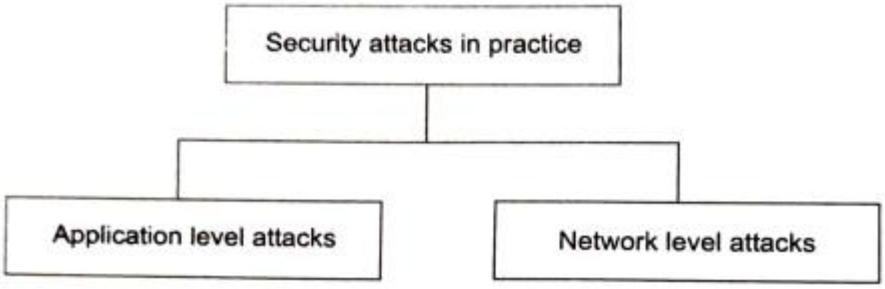
- **Replay attack** is caused when a user captures a sequence of events or some data units and re-sends them.
  - For instance, suppose user A wants to transfer some amount to user C's bank account.
  - Both users A and C have accounts with bank B. User A might send an electronic message to bank B, requesting for the funds transfer.
  - User C could capture this message and send a second copy of the same to bank B. Bank B would have no idea that this is an unauthorized message and would treat this as a second and different, funds transfer request from user A.
  - Therefore, user C would get the benefit of the funds transfer twice: once authorized, once through a replay attack.

- **Alteration of messages** involves some change to the original message.
  - For instance, suppose user A sends an electronic message Transfer $100 to D's account to bank B. User C might capture this and change it to Transfer $1000 to C's account.
  - Both the beneficiary and the amount have been changed - instead, only one of these could have also caused alteration of the message.

- **Denial Of Service (DOS)** attacks make an attempt to prevent legitimate users from accessing some services, which they are eligible for.
  - For instance, an unauthorized user might send too many login requests to a server using random user ids one after the other in quick succession, so as to flood the network and deny other legitimate users from using the network facilities.

1.4.3 The Practical Side of Attacks

- The security attacks can happen at the application level or the network level and can be classified into broad categories : **Application-level attacks** and **Network-level attacks**



- **Application-level attacks**:

These attacks happen at an application level, i.e. the attacker the attempts to access, modify or prevent access to information of a particular application or the application itself.

Example: Trying to obtain someone's credit card information on the Internet or changing the contents of a message to change the amount in a transaction, etc.

- **Network-level attacks**:

These attacks generally aim at reducing the capabilities of a network by a number of possible means. These attacks generally make an attempt to either slow down or completely bring to halt, a computer network.

This can lead to application-level attacks, because once someone is able to gain access to a network, they can access/modify at least some sensitive information, causing havoc.

1.4.4 PROGRAMS that ATTACK

Programs that attack computer systems to cause some damage or to create confusion are:

- Virus
- Worm
- Trojan Horse
- Applets & ActiveX Controls
- Cookies
- JavaScript VBScript & Jscript

1) Virus

- Virus can be used to launch an application-level attack or a network level attack virus.
- *A virus is a computer program that attaches itself to the legitimate program code and runs when the legitimate program runs causing damage to the computer system or to the network.*
- It can infect other programs in that computer or programs that are in other computers but on the same network as shown in figure 1.6.

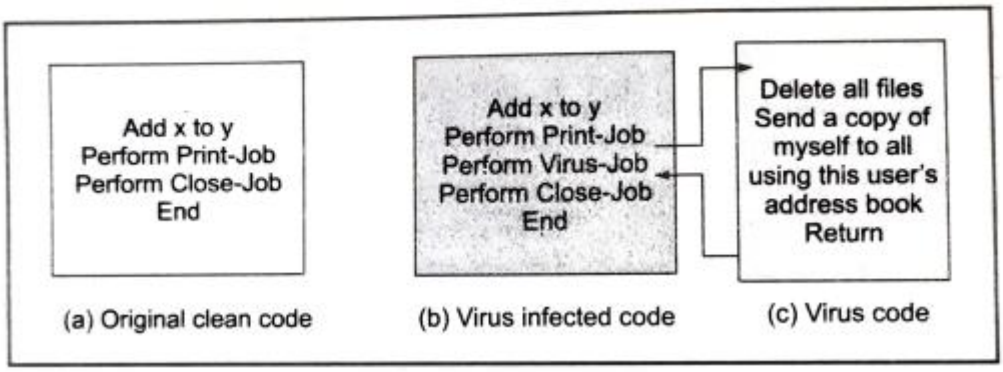


Figure 1.6 Virus

- In this example, after deleting all the files from the current user's computer, the virus self-propagates by sending its code to all users whose email addresses are stored in the current user's address book.
- Viruses can also be triggered by specific events (e.g. a virus could automatically execute at 12 PM every day). Usually, viruses cause damage to computer and network systems to the extent that it can be repaired, assuming that the organization deploys good backup and recovery procedures.

During its lifetime, a virus goes through four phases:

(a) **Dormant phase:** Here, the virus is idle. It gets activated based on certain action or event (e.g. the user typing a certain key or certain date or time is reached, etc). This is an optional phase.

(b) **Propagation phase:** In this phase, a virus copies itself and each copy starts creating more copies of self, thus propagating the virus.

(c) **Triggering phase:** A dormant virus moves into this phase when the action/event for which it was waiting is initiated.

(d) **Execution phase:** This is the actual work of the virus, which could be harmless (display some message on the screen) or destructive (delete a file on the disk).

Viruses can be classified into the following categories:

(a) **Parasitic virus:** This is the most common form of viruses. Such a virus attaches itself to executable files and keeps replicating. Whenever the infected file is executed, the virus looks for other executable files to attach itself and spread.

(b) **Memory-resident virus:** This type of virus first attaches itself to an area of the main memory and then infects every executable program that is executed.

(c) **Boot sector virus:** This type of virus infects the master boot record of the disk and spreads on the disk when the operating system starts booting the computer.

(d) **Stealth virus:** This virus has intelligence built in, which prevents anti-virus software programs from detecting it.

(e) **Polymorphic virus:** A virus that keeps changing its signature (i.e. identity) on every execution, making it very difficult to detect.

(f) **Metamorphic virus:** In addition to changing its signature like a polymorphic virus, this type of virus keeps rewriting itself every time, making its detection even harder.

(g) **Macro virus**. This virus affects specific application software, such as Microsoft Word or Microsoft Excel. These viruses affect the documents created by users and spread easily since such documents are very commonly exchanged over email. There is a feature called as *macro* these application software programs work, which allows the users to write small useful utility programs within the documents. Viruses attack these macros and hence the name **macro virus**.

2) Worm:

- A worm is similar to a virus, but different in implementation. A virus modifies a program (i.e. it attaches itself to the program under attack) whereas a worm does not modify a program. Instead, it **replicates itself again and again.**
- The replication grows so much that the computer or the network on which the worm resides, becomes very slow, finally coming to a halt.
- The basic purpose of a worm attack is different from that of a virus. A worm attack attempts to make the computer or the network under attack unusable by eating all its resources. This is illustrated in figure 1.7
- A worm does not perform down any destructive actions and instead, only consumes system resources to bring it down.

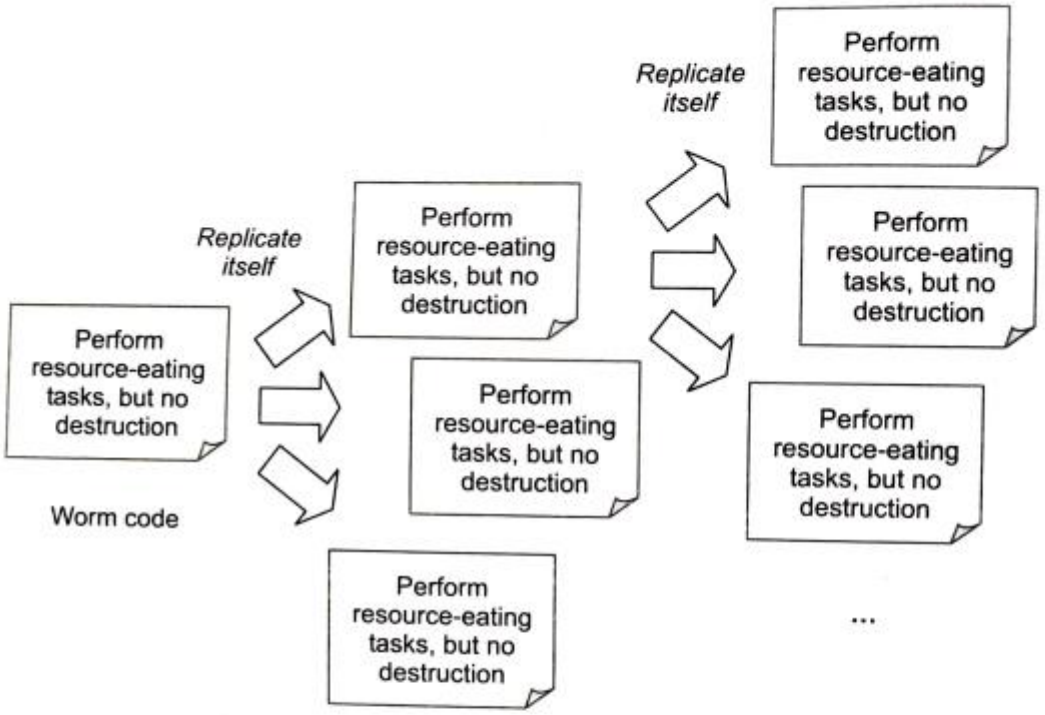

Perform resource-eating tasks, but no destruction
Worm code
Replicate itself
Perform resource-eating tasks, but no destruction
Perform resource-eating tasks, but no destruction
Perform resource-eating tasks, but no destruction
Replicate itself
Perform resource-eating tasks, but no destruction
Perform resource-eating tasks, but no destruction
Perform resource-eating tasks, but no destruction
...

Figure 1.7 Worm

3) Trojan Horse

- A Trojan horse is a hidden piece of code, like a virus. The main purpose of a virus is to make some sort of modifications to the target computer or network whereas a trojan horse attempts to reveal confidential information to an attacker.
- The name (Trojan horse) is due to the Greek soldiers, who hid inside a large hollow horse, which was pulled by troy citizens, unaware of its contents. Once the Greek soldiers entered the city of Troy, they opened the gates for the rest of Greek soldiers.
- In the same way, a Trojan horse could silently sit in the code for a Login screen by attaching itself to it. When the user enters the user id and password. the Trojan horse could capture these details and send this information to the attacker without the knowledge of the user who had entered the id and password.
- A Trojan horse allows an attacker to obtain some confidential information about a computer or a network. The attacker can then use the user id and password to gain access to the system. This is shown in figure

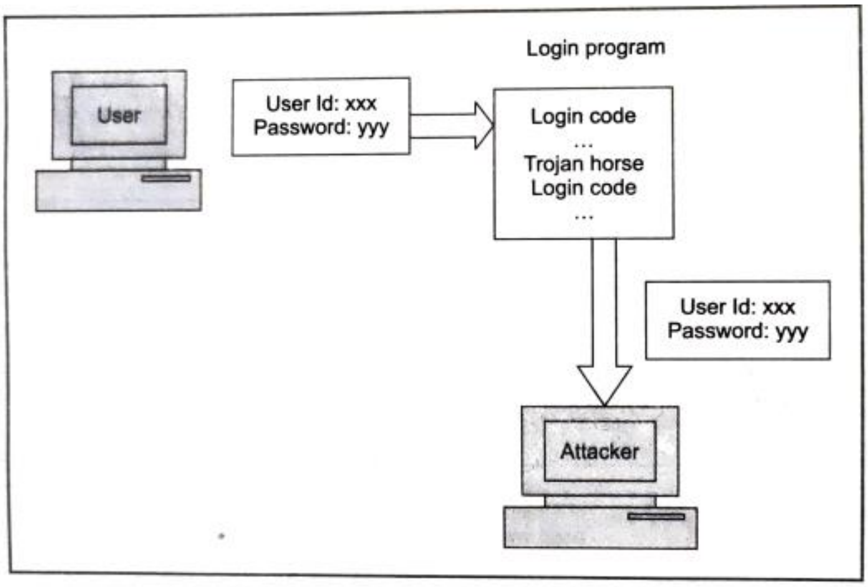


Figure 1.8 Trojan Horse

4) Applets and ActiveX Controls:

- Applets and ActiveX controls were born due to the technological development of the World Wide Web (www) application of the Internet.
- The Web consists of communication between client and server computers using a communications protocol called as Hyper Text Transfer Protocol (HTTP).
- The client uses a piece of software called as Web browser. The server runs a program called as Web server.
- In its simplest form, a browser sends a HTTP request for a Web page to a Web server. The Web server locates this Web page (actually a computer file) and sends it back to the Web browser, again using HTTP.
- The Web browser interprets the contents of that file and shows the results on the screen to the user. This is shown in Fig. 1.9.

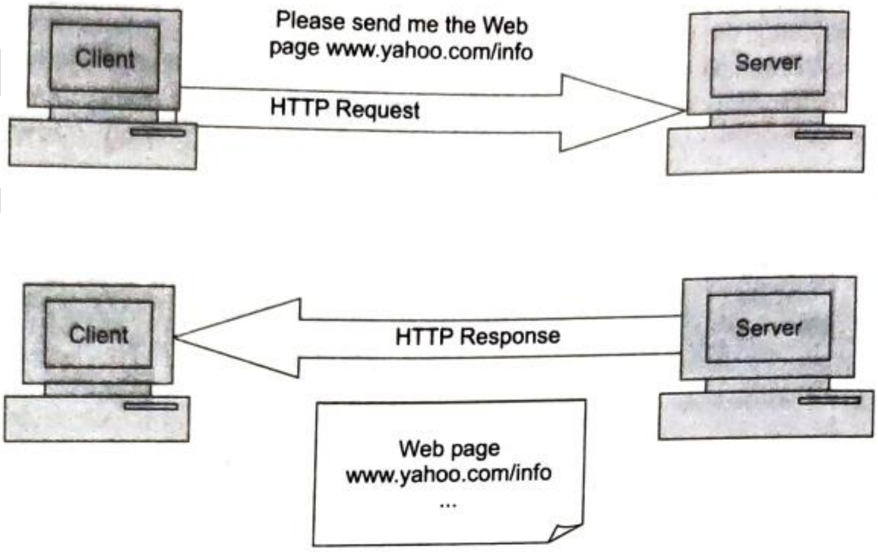


Figure 1.9 HTTP connection between client and server

- Here, the client sends a request for a Web page called as www.yahoo.com/info, which the server sends back to the client.
- Many Web pages contain small programs that get downloaded onto the client along with the Web page itself. These programs then execute inside the browser.
- Sun Microsystems provides Java applets for this purpose and Microsoft's technology makes use of ActiveX controls for the same purpose.
- Both are small programs that get downloaded along with a Web page and then execute on the client. This is shown in Fig. 1.10.

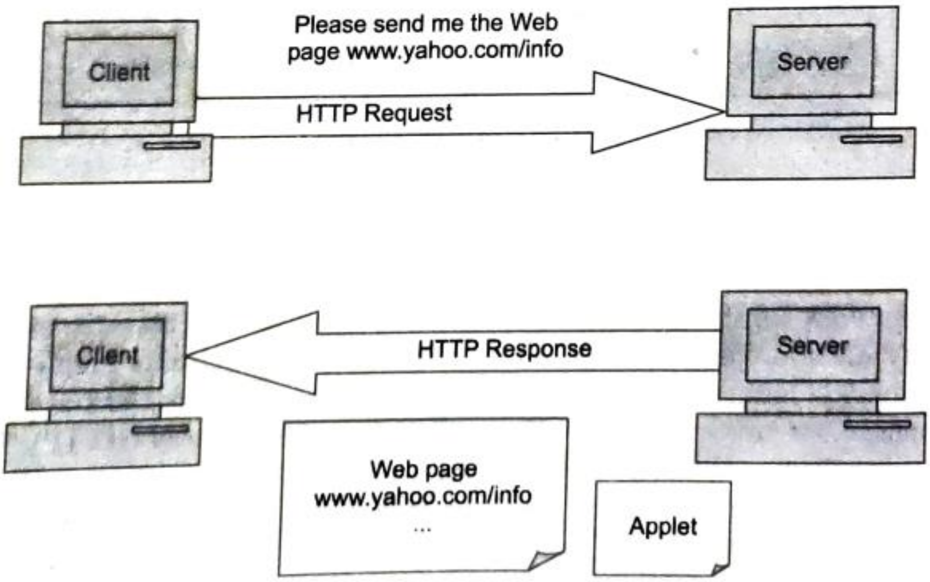

Client
Please send me the Web page www.yahoo.com/info
HTTP Request
Server
Client
HTTP Response
Server
Web page www.yahoo.com/info ...
Applet

Figure 1.10 Applet sent back along with a Web page

Here, the server sends an applet along with the Web page to the client.

- Usually, these programs (applets or ActiveX controls) are used to either perform some processing on the client side or to automatically and periodically request for information from the web server using a technology called as client pull.
- For instance, a program can get downloaded on to the client along with the Web page showing the latest stock prices on a stock exchange and then periodically issue HTTP requests for pulling the updated prices to the Web server.
- To prevent these attacks, Java applets have strong security checks as to what they can do and what they cannot. ActiveX controls have no such restrictions.
- A number of checks have been in place to ensure that neither applets nor ActiveX controls can do a lot of damage and even if they somehow manage to do it, it can be detected.
- Java applets (from Sun Microsystems) and ActiveX controls (from Microsoft Corporation) are small client-side programs that might cause security problems, if used by attackers with a malicious intention.

5) Cookies:

- Cookies were born as a result of a specific characteristic of the Internet. The Internet uses HTTP protocol, which is stateless.
- Suppose that the client sends an HTTP request for a Web page to the server. The Web server locates that page on its disk, sends it back to the client and completely forgets about this interaction!
- If the client wants to continue this interaction, it must identify itself to the server in the next HTTP request. Otherwise, the server would not know that this same client had sent a HTTP request earlier.
- Since a typical application is likely to involve a number of interactions between the client and the server, there must be some mechanism for the client to identify itself to the server each time it sends an HTTP request to the server.
- For this, cookies are used. Cookies are the most popular mechanism of maintaining the state information (i.e. identifying a client to a server). A cookie is just one or more pieces of information stored as text strings in a text file on the disk of the client computer (i.e. the Web browser).
- Actually, a Web server sends the Web browser a cookie and the browser stores it on the hard disk of the client computer. The browser then sends a copy of the cookie to the server during the next HTTP request.
- This is used for identification purposes as shown in Figs 1.11 (a) and 1.11 (b).

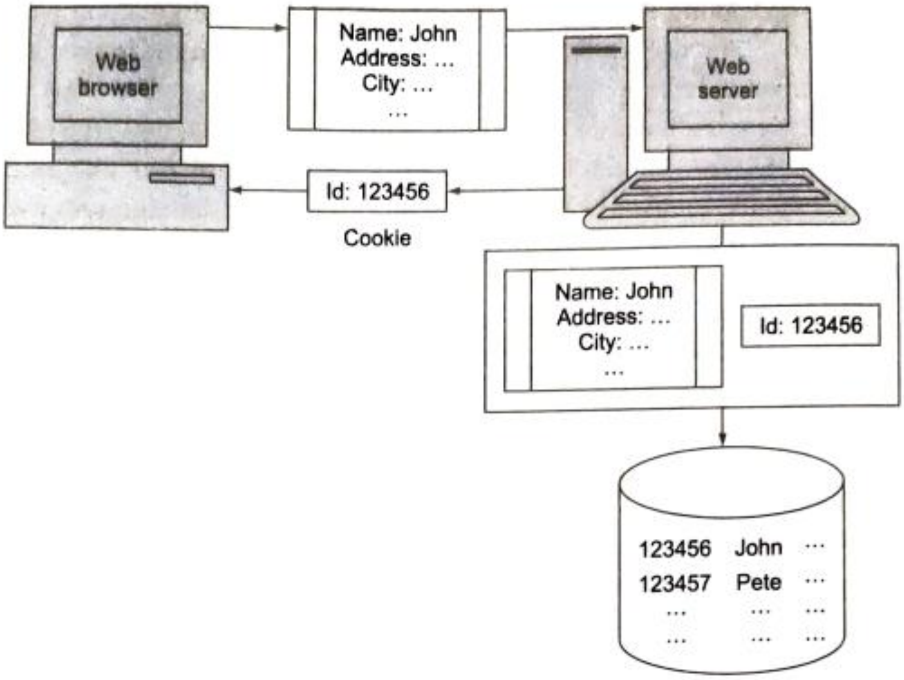


Figure 1.11a Creation of cookies

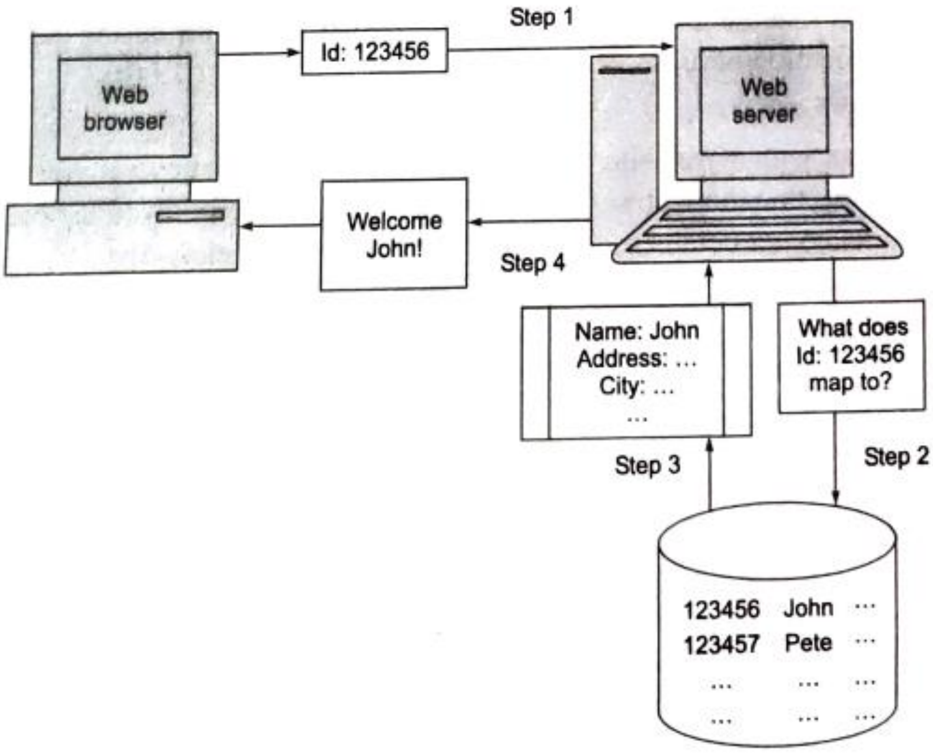


Figure 1.11b Usage of cookies

This works as follows:

(a) When you interact with a Web site for the first time, the site might want you to register yourself. Usually, this means that the Web server sends a page to you wherein you have a form to enter your name, address and other details such as date of birth, interests etc.

(b) When you complete this form and send it to the server with the help of your browser, the server stores this information into its database. Additionally, it also creates a unique id for you. It stores this id along with your information in the database (as shown in Fig. 1.11(b)) and also sends the id back to you in the form of a cookie.

(c) The next time you interact with the server, you do not have to enter any information such as your name and address. Your browser would automatically send your id (i.e. the cookie) along with the HTTP request for a particular page to the server (as shown in Fig. 1.11b)).

(d) The server now takes this id, tries to find a match in its database and having found it, knows that you are a registered user. Accordingly, it sends you the next page.

6) JavaScript, VBScript and JScript

- A Web page is constructed using a special language called as Hyper Text Markup Language (HTML). It is a tag-based language. A tag begins with the symbol <> and it ends with </>.
- Between these boundaries of the tags, the actual information to be displayed on the user's computer is mentioned. As an example, let us consider how the tag pair <B> and </B> can be used to change the text font to boldface.
- When a browser comes across this portion of a HTML document, it realizes that the portion of the text embedded within the <b> and </b> tags need to be displayed in boldface. Therefore, it displays this text in boldface.
- In addition to HTML tags, a Web page can contain client-side scripts. These are small programs written in scripting languages like JavaScript, VBScript or Jscript, which are executed inside the Web browser on the client computer.
- For instance, let us assume that a user visits the Web site of an online bookshop. Suppose that the Web site mandates that the user must place an order for at least three books. Then, the web page can contain a small JavaScript program, which can ensure that this condition is met before the user can place the order. Otherwise, the JavaScript program would not allow the user to proceed. Note that HTML cannot be used for this purpose, as its sole purpose is to display text on the client computer in a pre-specified format. To perform dynamic actions, scripts are needed.
- These scripts can be dangerous at times. Since these scripts are small programs, they can perform a lot of actions on the client's computer. There are restrictions on the actions of a scripting program. Incidents of security breaches have been reported, blaming the scripting languages.

1.4.5 Dealing with viruses

- The detection, identification and removal of viruses' steps is shown in figure 1.12

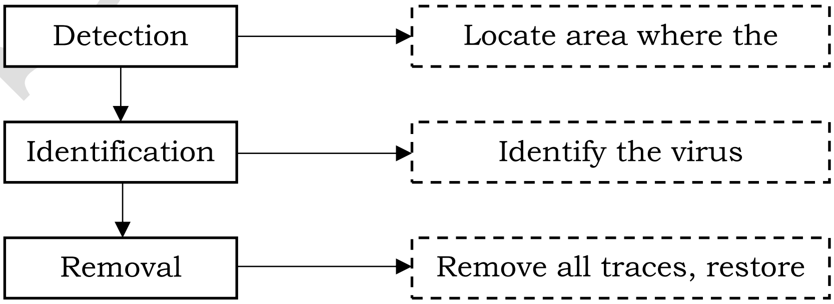


Figure 1.12 Virus elimination steps

- Detection of viruses involves locating the virus, having known that a virus has attacked. Then we need to identify the specific virus that has attacked. Finally, we need to remove it. For this we need to remove all traces of the virus and restore the affected programs/files to their original states. This is done by anti-virus software.
- Anti-virus software is classified into four generations as shown in figure 1.13

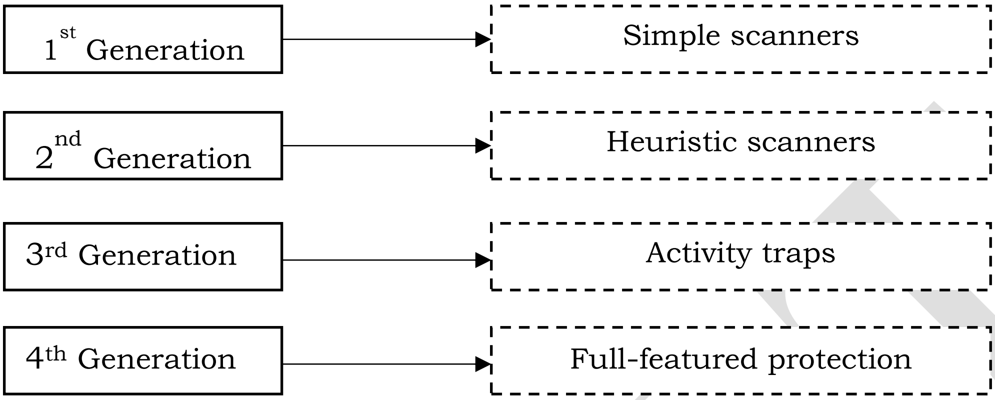


Figure 1.13 Generations of anti-virus software

The key characteristics of the four generations of anti-virus software.

1st generation

- These anti-virus software programs were called as **simple scanners.** They needed a virus signature to identify a virus. A variation of such programs kept a watch on the length of programs and looked for changes so as to possibly identify a virus attack.

2nd generation

- These anti-virus software programs did not rely on simple virus signatures. Rather, they used heuristic rules to look for possible virus attacks. The idea was to look for code blocks that were commonly associated with viruses. Another variation of these anti-virus programs used to store some identification about the file (e.g. a message digest) to detect changes in the contents of the file.

3rd generation

- These anti-virus software programs were memory resident. They watched for viruses based on actions, rather than their structure. Thus, it is not necessary to maintain a large database of virus signatures. Instead, the focus is to keep watch on a small number of suspect actions.

4th generation

- These anti-virus software programs package many anti-virus techniques together (e.g. scanners, activity monitoring). They also contain access control features, thus attempts of viruses to infect files.

- There is a category of software called as behavior-blocking software, which integrates with the operating system of the computer and keeps a watch on virus-like behavior in real time.
- Whenever such an action is detected, this software blocks it, preventing damages. The actions under watch can be:
    - Opening , viewing , modifying , deleting files
    - Network communications
    - Modification of settings such as start-up scripts
    - Attempt to format disks
    - Modification of executable files
    - Scripting of email and instant messaging to send executable content to others

The main advantage of such software programs is that they are more into virus prevention than virus detection. In other words, they stop viruses before they can do any damage, rather than detecting them after an attack.

1.4.6 JAVA Security

- Java was designed in such a way that Java programs are considered safe as they cannot install, execute or propagate viruses and because the program itself cannot perform any action that is harmful to the user's computer.
- One of the key attributes of Java is the ability to download Java programs over a network and execute these programs on a different computer within the context of a Java-enabled browser.
- Different developers were attracted to Java with different expectations. As a result , they brought different ideas about Java security. If we put Java to be free from introducing viruses, any release of Java should satisfy our requirements.
- However, if functionalities such as digital signatures, authentication and encryption are required in the least release 1.1 of Java must be used.

➢ The Java Sandbox

- Java’s security model is closely associated with the idea of a sandbox model. A sandbox model allows a program to be hosted and executed, but there are some restrictions in place.
- The developer/end user may decide to give the program access to certain resources. However, in general, they want to make sure that the program is confined to its sandbox. The overall execution of a java program on the Internet is as shown in Fig 1.14.

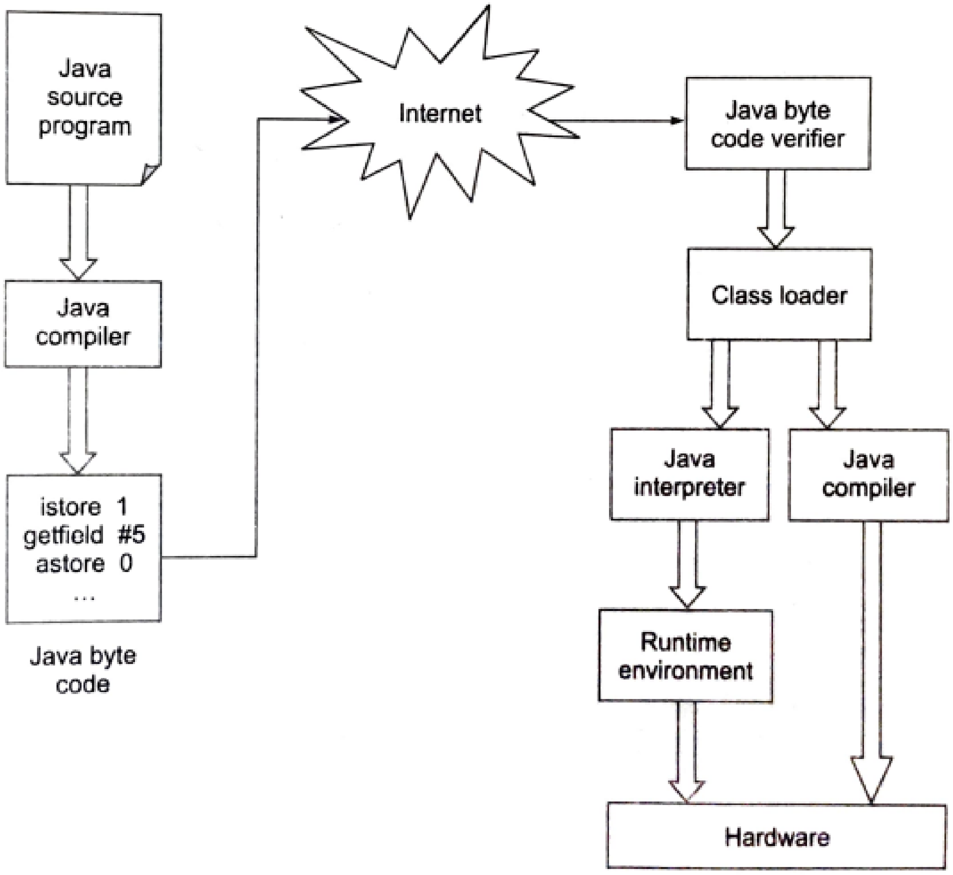


Figure 1.14 Steps in execution of a Java program on the Internet

- The chief job of the Java sandbox is to protect a number of resources and it performs this task so at a number of levels.
  - ✓ A sandbox in which program can access the CPU, the screen, the keyboard and mouse and its own memory. This is the basic sandbox. It contains just enough resources for a program to execute.
  - ✓ A sandbox in which a program can access the CPU and its memory as well as access the Web server from which it was downloaded. This is often considered as the default state for the sandbox.
  - ✓ A sandbox in which program can access the CPU, its memory, its Web server and to a set of resources (files, computers, etc.) that are local.
  - ✓ An open sandbox, in which the program can access whatever resources the host machine can.

➢ Java Application Security

- The broad level aspects of Java security and their relation to each other.
- ***The bytecode verifier***: The bytecode verifier ensures that Java class files obey the rules of the Java programming language. The bytecode verifier ensures memory protection for all Java programs. However, not all files are required to go through byte code verification.

- ***The class loader:*** Class loaders load classes that are located in Java's default path (called as CLASSPATH). In Java 1.2, the class loaders also take up the job of loading classes that are not found in the CLASSPATH.
- ***The access controller:*** In Java 1.2, the access controller allows (or prevents) access from the core JAVA API to the operating system.
- ***The security manager:*** The security manager is the chief interface between the core Java API and the operating system. It has the ultimate responsibility for allowing or disallowing access to all the operating system resources. The security manager uses the access controller for many of these decisions.
- ***The security package:*** The security package (that is, classes in the java.security package) helps in authenticating signed Java classes.
- ***The key database***: The key database is a set of keys used by the security manager and access Controller to validate the digital signature that comes along with a signed class file.

➢ Built-in Java Application Security

- From version 1.2, the Java platform itself comes with a Security model built for the applications it runs. Here, the classes that are found in the CLASSPATH may have to go through a security check. This allows running of the application code in a sandbox defined by a user or an administrator. The following points are important:
  - ✓ Access methods are strictly adhered to
  - ✓ A program cannot access arbitrary memory location
  - ✓ Entities that are declared as final must not be changed
  - ✓ Variables may not be used before they are initialized
  - ✓ Array bounds must be checked during all array accesses
  - ✓ Objects cannot arbitrarily cast into other object type

1.4.7 Specific Attacks

1) Sniffing and Spoofing :

- On the Internet, computers exchange messages with each other in the form of small blocks of data, called as packets. A packet, like a postal envelope that contains the actual data to be sent and the addressing information.
- Attackers target these packets, as they travel from the source computer to the destination computer over the Internet. These attacks take two main forms:

  (a) Packet sniffing (also called as snooping) and **(b) Packet spoofing.**
- Since the protocol used in this communication is called as Internet Protocol (IP), other names for these two attacks are: (a) IP sniffing and (b) IP spoofing.

a) Packet sniffing:

- Packet sniffing is a passive attack on an ongoing conversation. An attacker need not hijack a conversation, but instead, can just observe (i.e. snif) packets as they pass by.
- To prevent an attacker from sniffing packets, the information that is passing needs to be protected in some ways. This can be done at two levels:

  (i) The data that is traveling can be encoded in some way or

  (ii) The transmission link itself can be encoded.
- To read a packet, the attacker needs to access it. The simplest way to do this is to control a computer through which the traffic goes. Usually, this is a router. However, routers are highly protected resources. Therefore, an attacker might not be able to attack it and instead, attack a less protected computer on the same path.

b) Packet spoofing:

- In this technique, an attacker sends packets with a false source address. When this happens, the receiver (i.e. the party who receives these packets containing false address) would inadvertently send replies back to this forged address (called as spoofed address). This can lead to three possible cases:

  i) **The attacker can intercept the reply** - If the attacker is between the destination and the forged source, the attacker can see the reply and use that information for hijacking attacks.

  ii) **The attacker need not see the reply** - If the attacker's intention was a Denial Of Service (DOS) attack, the attacker need not bother about the reply.

  iii) **The attacker does not want the reply**- The attacker could simply be angry with the host, so it may put that host's address as the forged source address and send the packet to the destination. The attacker does not want a reply from the destination, as it wants the host with the forged address to receive it and get confused.

2) Phishing :

- In Phishing , attackers set up fake Web sites, which look like real Web sites. It is simple to create Web pages as it involves simple technologies such as HTML, JavaScript, CSS (Cascading Style Sheets), etc. Learning and using these technologies is quite simple. Phishing works as follows.
- The attacker decides to create his own Web site, which looks very identical to a real Web site. For example, the attacker can clone Citibank's Web site. The cloning is so clever that human eye will not be able to distinguish between the real (Citibank's) and fake (attacker's) sites now.
- The attacker sends an email to the legitimate customers of the bank. The email itself appears to come from the bank. For ensuring this, the attacker exploits the email system to suggest that the sender of the email is some bank official (e.g. accountmanager@citibank.comn).
- This fake email warns the user that there has been some sort of attack on the Citibank's computer systems and that the bank wants to issue new passwords to all its customers or verify their existing PINs, etc. For this purpose, the customer is asked to visit a URL mentioned in the same email.
- When the customer (i.e. the victim) innocently clicks on the URL specified in the email, she is taken to the attacker's site and not the bank's original site. There, the customer is prompted to enter confidential information, such as her password or PIN.
- Since the attacker's fake site looks exactly like the original bank site, the customer provides this information. The attacker gladly accepts this information and displays a Thank you to the unsuspecting victim. In the meanwhile, the attacker now uses the victim's password or PIN to access the bank's real site and can perform any transaction as if he/she is the victim!

3) Pharming (DNS Spoofing):

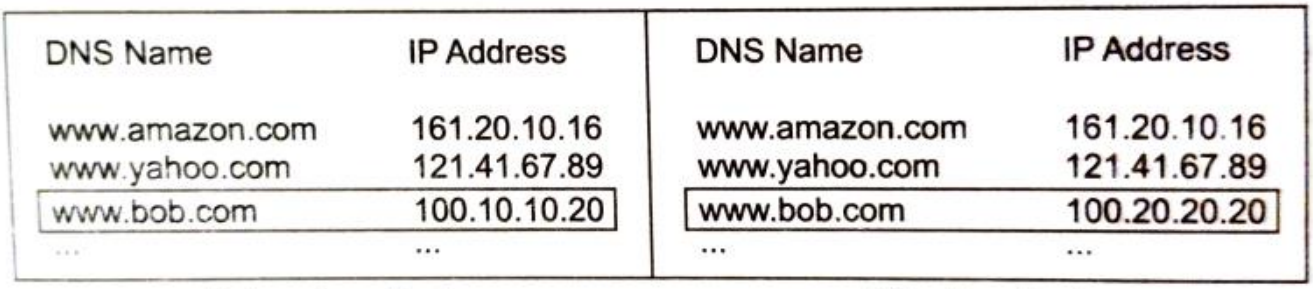
- This attack was earlier known as **DNS spoofing** or DNS poisoning is now called as **pharming attack**.
- With the Domain Name System (DNS), people can identify Web sites with human-readable names (such as www.yahoo.com) and computers can continue to treat them as IP addresses (such as 120.10.81.67).
- For this, a special server computer called as a DNS server maintains the mappings between domain names and the corresponding IP addresses. The DNS server could be located anywhere. Usually, it is with the Internet Service Provider (ISP) of the users.

Example: The DNS spoofing attack works as follows.

- Suppose that there is a merchant (Bob), whose site's domain name is www.bob.com and the IP address is 100.10.10.20. Therefore, the DNS entry for Bob in all the DNS servers is maintained as follows: www.bob.com 100.10.10.20

- The attacker (Trudy) manages to hack and replace the IP address of Bob with his own (say 100.20.20.20) in the DNS server maintained by the ISP of a user(say Alice). Therefore, the DNS server maintained by the ISP of Alice now has the following entry:

  www.bob.com 100.20.20.20
- Thus, the contents of the hypothetical DNS table maintained by the ISP would be changed. A hypothetical portion of this table (before and after the attack) is shown in Figure below.



DNS Name	IP Address
www.amazon.com	161.20.10.16
www.yahoo.com	121.41.67.89
www.bob.com	100.10.10.20
...	...

Before the attack

DNS Name	IP Address
www.amazon.com	161.20.10.16
www.yahoo.com	121.41.67.89
www.bob.com	100.20.20.20
...	...

After the attack

- When Alice wants to communicate with Bob's site, her Web browser queries the DNS server maintained by her ISP for Bob's IP address, providing it the domain name (i.e. www.bob.com). Alice gets the replaced (i.e. Trudy's) IP address, which is 100.20.20.20.
- Now, Alice starts communicating with Trudy, believing that she is communicating with Bob! Such attacks of DNS spoofing are quite common and cause a lot of havoc.