Cryptography and Network Security

Third Edition

CONTENTS

1

INTRODUCTION TO THE CONCEPTS OF SECURITY

1.1 INTRODUCTION

This is a book on network and Internet security. As such, before we embark on our journey of understanding the various concepts and technical issues related to security (i.e. trying to understand *how* to protect), it is essential to know *what* we are trying to protect. What are the various dangers when we use computers, computer networks, and the biggest network of them all, the Internet? What are the likely pitfalls? What can happen if we do not set up the right security policies, framework and technology implementations? This chapter attempts to provide answers to these basic questions.

We start with a discussion of the basic question: Why is security required in the first place? People sometimes say that security is like statistics: what it reveals is trivial, what it conceals is vital! In other words, the right security infrastructure opens up just enough doors that are mandatory. It protects everything else. We discuss a few real-life incidents that should prove beyond doubt that security cannot simply be compromised. Especially these days, when serious business and other types of transactions are being conducted over the Internet to such a large extent, that inadequate or improper security mechanisms can bring the whole business down, or play havoc with people's lives!

We then discuss the key principles of security. These principles help us identify the various areas, which are crucial while determining the security threats and possible solutions to tackle them. Since electronic documents and messages are now becoming equivalent to the paper documents in terms of their legal validity and binding, we examine the various implications in this regard.

This is followed by a discussion on the types of attacks. There are certain theoretical concepts associated with attacks, and there is a practical side to it as well. We shall discuss all these aspects.

Finally, we discuss the outline and scope of the rest of the book. This will pave the way for further discussions of network and Internet security concepts.

1.2 THE NEED FOR SECURITY

1.2.1 Basic Concepts

Most previous computer applications had *no*, or at best, *very little* security. This continued for a number of years until the importance of data was truly realized. Until then, computer data was considered to be useful, but not something to be protected. When computer applications were developed to handle financial and personal data, the real need for security was felt like never before. People realized that data on computers is an extremely important aspect of modern life. Therefore, various areas in security began to gain prominence. Two typical examples of such security mechanisms were as follows:

- Provide a user identification and password to every user, and use that information to authenticate a user.
- Encode information stored in the databases in some fashion, so that it is not visible to users who do not have the right permission.

Organizations employed their own mechanisms in order to provide for these kinds of basic security mechanisms. As technology improved, the communication infrastructure became extremely mature, and newer applications began to be developed for various user demands and needs. Soon, people realized the basic security measures were not quite enough.

Furthermore, the Internet took the world by storm. There were many examples of what could happen if there was insufficient security built in applications developed for the Internet. Figure 1.1 shows such an example of what can happen when you use your credit card for making purchases over the Internet. From the user's computer, the user details such as user id, order details such as order id and item id,

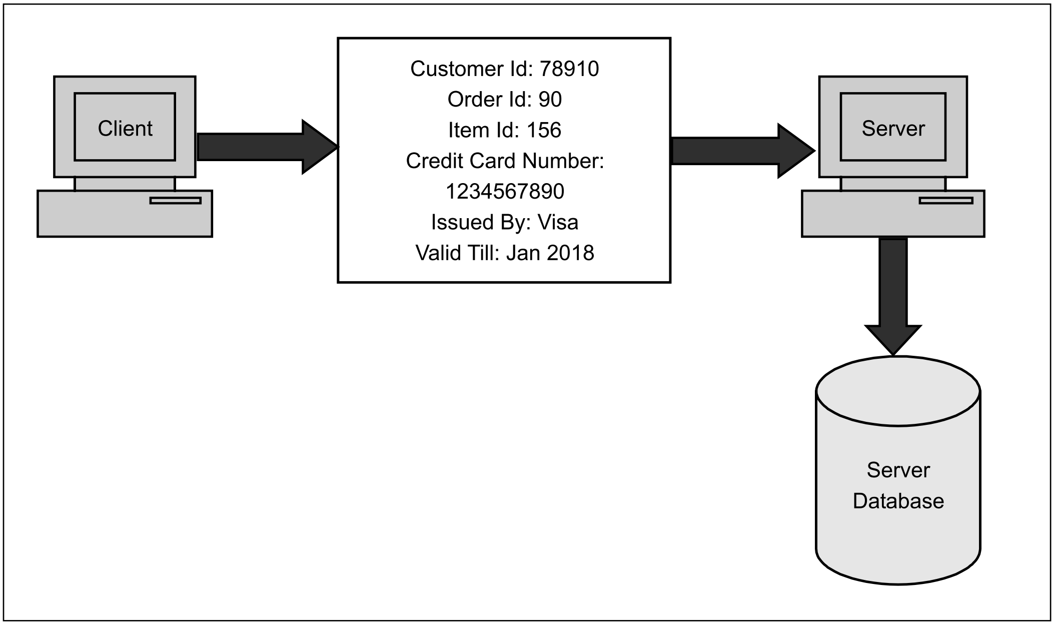


Fig. 1.1 Example of information traveling from a client to a server over the Internet

and payment details such as credit-card information travel across the Internet to the server (i.e. to the merchant's computer). The merchant's server stores these details in its database.

There are various security holes here. First of all, an intruder can capture the credit-card details as they travel from the client to the server. If we somehow protect this transit from an intruder's attack, it still does not solve our problem. Once the merchant computer receives the credit-card details and validates them so as to process the order and later obtain payments, the merchant computer stores the credit-card details into its database. Now, an attacker can simply succeed in accessing this database, and therefore gain access to all the credit-card numbers stored therein! One Russian attacker (called 'Maxim') actually managed to intrude into a merchant Internet site and obtained 300,000 credit-card numbers from its database. He then attempted extortion by demanding protection money ($100,000) from the merchant. The merchant refused to oblige. Following this, the attacker published about 25,000 of the credit-card numbers on the Internet! Some banks reissued all the credit cards at a cost of $20 per card, and others forewarned their customers about unusual entries in their statements.

Such attacks could obviously lead to great losses—both in terms of finance and goodwill. Generally, it takes $20 to replace a credit card. Therefore, if a bank has to replace 3,00,000 such cards, the total cost of such an attack is about $6 million! How helpful would it have been, if the merchant in the example just discussed had employed proper security measures!

Of course, this was just one example. Several such cases have been reported in the last few months, and the need for proper security is being felt increasingly with every such attack. In another example of security attack, in 1999, a Swedish hacker broke into Microsoft's Hotmail Web site, and created a mirror site. This site allowed anyone to enter any Hotmail user's email id, and read his/her emails!

In 1999, two independent surveys were conducted to invite people's opinions about the losses that occur due to successful attacks on security. One survey pegged the losses figuring at an average of $256,296 per incident, and the other one's average was $759,380 per incident. Next year, this figure rose to $972,857!

1.2.2 Modern Nature of Attacks

If we attempt to demystify technology, we would realize that computer-based systems are not all that different from what happens in the real world. Changes in computer-based systems are mainly due to the speed at which things happen and the accuracy that we get, as compared to the traditional world.

We can highlight a few salient features of the modern nature of attacks, as follows:

1. Automating Attacks

The speed of computers make several attacks worthwhile for miscreants. For example, in the real world, let's suppose someone manages to create a machine that can produce counterfeit coins. Would that bother authorities? It certainly would. However, producing so many coins on a mass scale may not be that much economical compared to the return on that investment! How many such coins would the attacker be able to get into the market so rapidly? But, the scenario is quite different with computers. They are quite efficient and happy in doing routine, mundane, repetitive tasks. For example, they would excel in somehow stealing a very low amount (say half a dollar or 20 rupees) from a million bank accounts in a matter of a few minutes. This would give the attacker a half million dollars possibly without any major complaints! This is shown in Fig. 1.2.

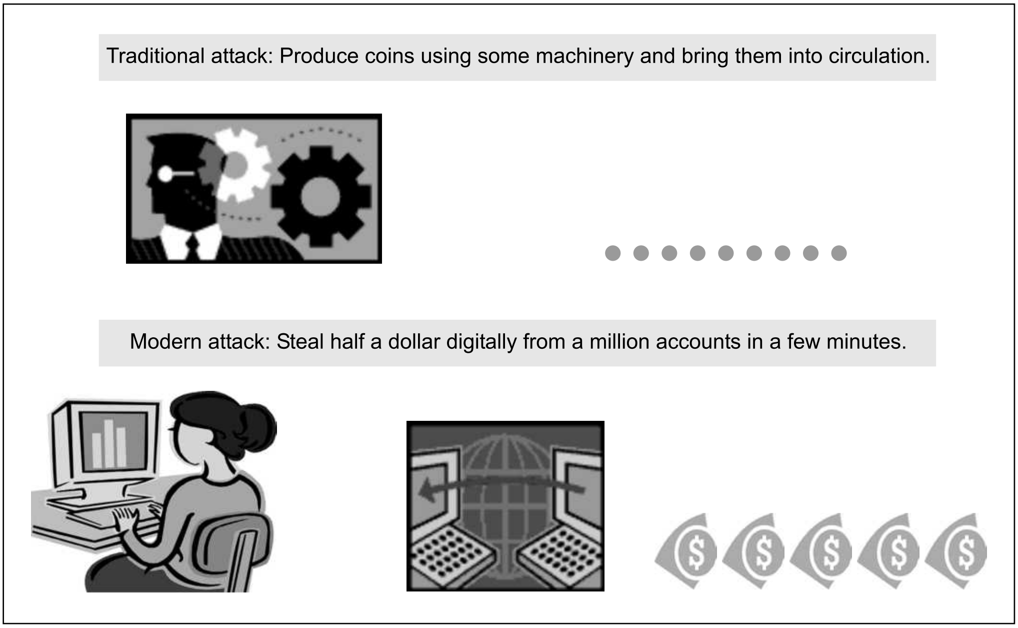

Traditional attack: Produce coins using some machinery and bring them into circulation.
Modern attack: Steal half a dollar digitally from a million accounts in a few minutes.

Fig. 1.2 The changing nature of attacks due to automation

The morale of the story is:

> *Humans dislike mundane and repetitive tasks. Automating them can cause financial destruction or a security nuisance quite rapidly.*

2. Privacy Concerns

Collecting information about people and later (mis)using it is turning out to be a huge problem these days. The so-called *data mining* applications gather, process, and tabulate all sorts of details about individuals. People can then illegally sell this information. For example, companies like Experian (formerly TRW), TransUnion, and Equifax maintain credit history of individuals in the USA. Similar trends are seen in the rest of the world. These companies have volumes of information about a majority of citizens of that country. These companies can collect, collate, polish, and format all sorts of information to whosoever is ready to pay for that data! Examples of information that can come out of this are: which store the person buys more from, which restaurant he/she eats in, where he/she goes for vacations frequently, and so on! Every company (e.g. shopkeepers, banks, airlines, insurers) are collecting and processing a mind-boggling amount of information about us, without us realizing when and how it is going to be used.

3. Distance Does not Matter

Thieves would earlier attack banks, because banks had money. Banks do not have money today! Money is in digital form inside computers, and moves around by using computer networks. Therefore, a modern thief would perhaps not like to wear a mask and attempt a robbery! Instead, it is far easier and cheaper to attempt an attack on the computer systems of the bank while sitting at home! It may be far

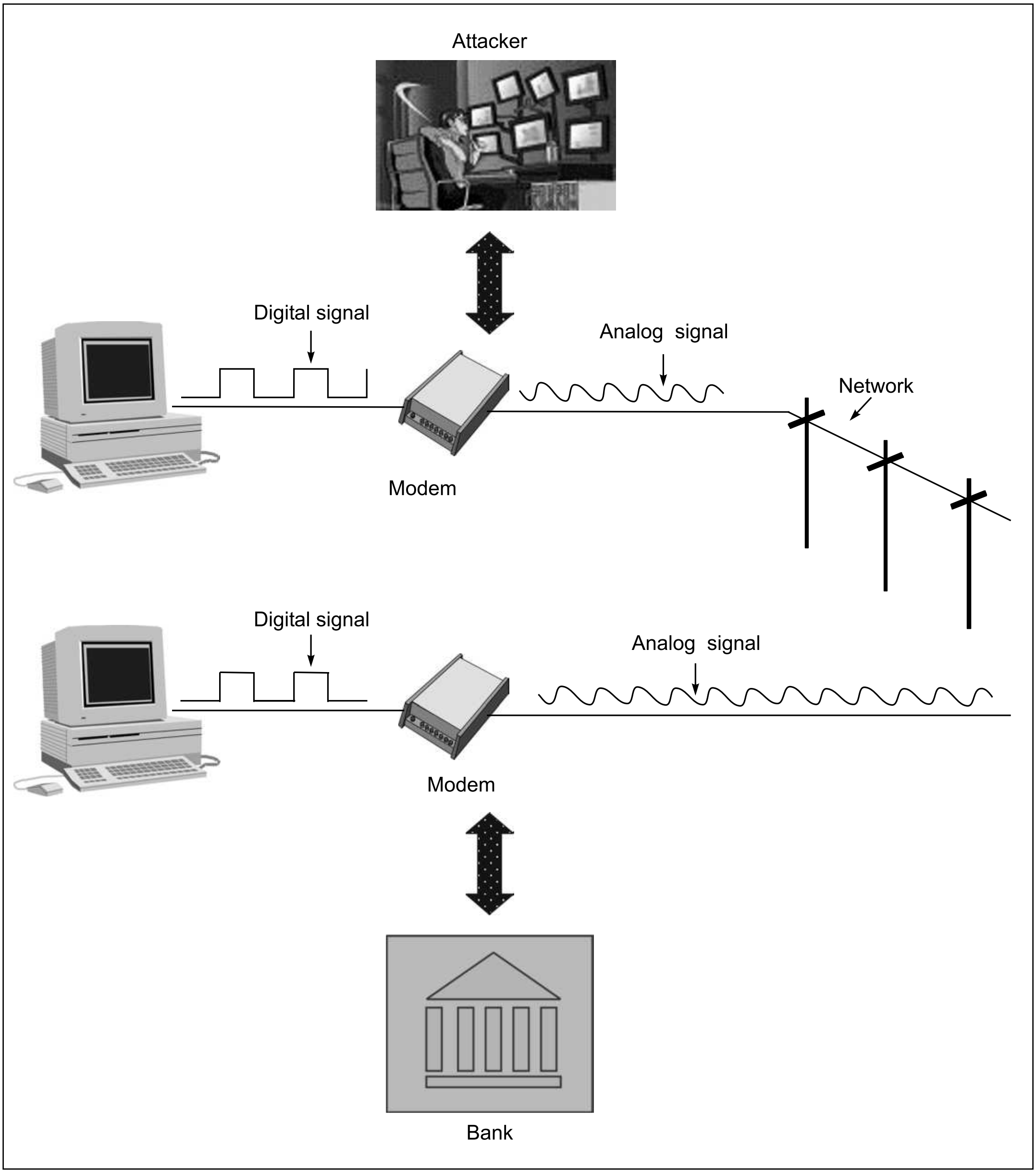


Fig. 1.3 Attacks can now be launched from a distance

more prudent for the attacker to break into the bank's servers, or steal credit card/ATM information from the comforts of his/her home or place of work. This is illustrated in Fig. 1.3.

In 1995, a Russian hacker broke into Citibank's computers remotely, stealing $12 million. Although the attacker was traced, it was very difficult to get him extradited for the court case.

1.3 SECURITY APPROACHES

1.3.1 Trusted Systems

> *A* **trusted system** *is a computer system that can be* trusted *to a specified extent to enforce a specified security policy.*

Trusted systems were initially of primary interest to the military. However, these days, they have spanned across various areas, most prominently in the banking and financial community, but the concept never caught on. Trusted systems often use the term **reference monitor**. This is an entity that is at the logical heart of the computer system. It is mainly responsible for all the decisions related to access controls. Naturally, following are the expectations from the reference monitor:

(a) It should be tamper-proof.

(b) It should always be invoked.

(c) It should be small enough so that it can be tested independently.

In their 1983 *Orange Book* (also called the *Trusted Computer System Evaluation Criteria (TCSEC)*), the National Security Agency (NSA) of the US Government defined a set of *evaluation classes*. These described the features and assurances that the user could expect from a trusted system.

The highest levels of assurance were provided by significant efforts directed towards reduction of the size of the trusted computing base, or TCB. In this context, TCB was defined as a combination of hardware, software, and firmware responsible for enforcing the system's security policy. The lower the TCB, the higher the assurance. However, this raises an inherent problem (quite similar to the decisions related to the designing of operating systems). If we make the TCB as small as possible, the surrounding hardware, software, and firmware are likely to be quite big!

The mathematical foundation for trusted systems was provided by two relatively independent yet interrelated works. In the year 1974, David Bell and Leonard LaPadula of MITRE devised a technique called the **Bell-LaPadula model**. In this model, a highly trustworthy computer system is designed as a collection of objects and subjects. Objects are passive repositories or destinations for data, such as files, disks, printers, etc. Subjects are active entities, such as users, processes, or threads operating on behalf of those users. Subjects cause information to flow among objects.

Around the same time, Dorothy Denning at Purdue University was preparing for her doctorate. It dealt with *lattice-based information flows* in computer systems. A mathematical *lattice* is a partially ordered set, in which the relationship between any two vertices either *dominates*, *is dominated by* or *neither*. She devised a generalized notion of *labels*—similar to the full security markings on classified military documents. Examples of this are TOP SECRET.

Later, Bell and LaPadula integrated Denning's theory into their MITRE technical report, which was titled *Secure Computer System: Unified Exposition and Multics Interpretation*. Here, *labels* attached to *objects* represented the sensitivity of data contained within the *object*. Interestingly, the Bell–LaPadula model talks only about *confidentiality* or *secrecy* of information. It does not talk about the problem of *integrity* of information.

1.3.2 Security Models

An organization can take several approaches to implement its security model. Let us summarize these approaches.

1. No Security

In this simplest case, the approach could be a decision to implement no security at all.

2. 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.

3. Host Security

In this scheme, the security for each host is enforced individually. This is a very safe approach, but the trouble is that it cannot scale well. The complexity and diversity of modern sites/organizations makes the task even harder.

4. Network Security

Host security is tough to achieve as organizations grow and become 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.

1.3.3 Security-Management Practices

Good **security-management practices** always talk of a **security policy** being in place. Putting a security policy in place is actually quite tough. A good security policy and its proper implementation go a long way in ensuring adequate security-management practices. A good security policy generally takes care of four key aspects, as follows.

- *Affordability* How much money and effort does this security implementation cost?
- *Functionality* What is the mechanism of providing security?
- *Cultural Issues* Does the policy complement the people's expectations, working style and beliefs?
- *Legality* Does the policy meet the legal requirements?

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

(a) Explanation of the policy to all concerned.

(b) Outline everybody's responsibilities.

(c) Use simple language in all communications.

(d) Accountability should be established.

(e) Provide for exceptions and periodic reviews.

1.4 PRINCIPLES OF SECURITY

Having discussed some of the attacks that have occurred in real life, let us now classify the principles related to security. This will help us understand the attacks better, and also help us in thinking about the possible solutions to tackle them. We shall take an example to understand these concepts.

Let us assume that a person A wants to send a check worth $100 to another person B. Normally, what are the factors that A and B will think of, in such a case? A will write the check for $100, put it inside an envelope, and send it to B.

- A will like to ensure that no one except B gets the envelope, and even if someone else gets it, he/she does not come to know about the details of the check. This is the principle of **confidentiality**.
- A and B will further like to make sure that no one can tamper with the contents of the check (such as its amount, date, signature, name of the payee, etc.). This is the principle of **integrity**.
- B would like to be assured that the check has indeed come from A, and not from someone else posing as A (as it could be a fake check in that case). This is the principle of **authentication**.
- What will happen tomorrow if B deposits the check in his/her account, the money is transferred from A's account to B's account, and then A refuses having written/sent the check? The court of law will use A's signature to disallow A to refute this claim, and settle the dispute. This is the principle of **non-repudiation**.

These are the four chief principles of security. There are two more: **access control** and **availability**, which are not related to a particular message, but are linked to the overall system as a whole.

We shall discuss all these security principles in the next few sections.

1.4.1 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. An example of compromising the confidentiality of a message is shown in Fig. 1.4. Here, the user of computer A sends a message to the user of computer B. (**Actually, from here**

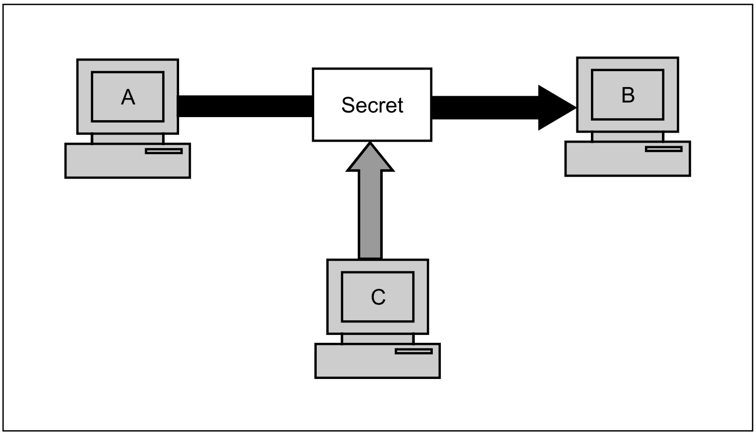


Fig. 1.4 Loss of confidentiality

onwards, we shall use the term A to mean the *user* A, B to mean *user* B, etc., although we shall just show the computers of users A, B, etc.). Another user C gets access to this message, which is not desired, and therefore defeats the purpose of confidentiality. An example of this could be 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 **interception**.

Interception causes loss of message confidentiality.

1.4.2 Authentication

Authentication mechanisms help establish **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. However, the trouble is that user C had posed as user A when he/she sent this document to user B. How would user B know that the message has come from user C, who is posing as user A? A real-life example of this could be the case of a user C, posing as user A, sending a funds transfer request (from A's account to C's account) to bank B. The bank might happily transfer the funds from A's account to C's account—after all, it would think that user A has requested for the funds transfer! This concept is shown in Fig. 1.5. This type of attack is called **fabrication**.

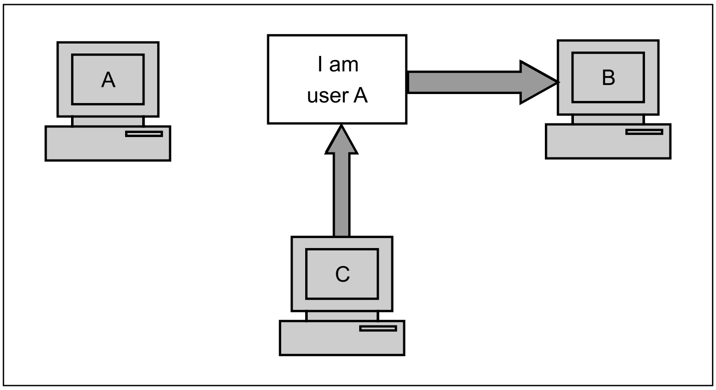


Fig. 1.5 Absence of authentication

Fabrication is possible in absence of proper authentication mechanisms.

1.4.3 Integrity

When the contents of a message are changed after the sender sends it, but before it reaches the intended recipient, we say that the *integrity* of the message is lost. For example, suppose you write a check for $100 to pay for goods bought from the US. However, when you see your next account statement, you are startled to see that the check resulted in a payment of $1000! This is the case for loss of message integrity. Conceptually, this is shown in Fig. 1.6. Here, user C tampers with a message originally sent by user A, which is actually destined for user B. User C somehow manages to access it, change its contents, and send the changed message to user B. User B has no way of knowing that the contents of

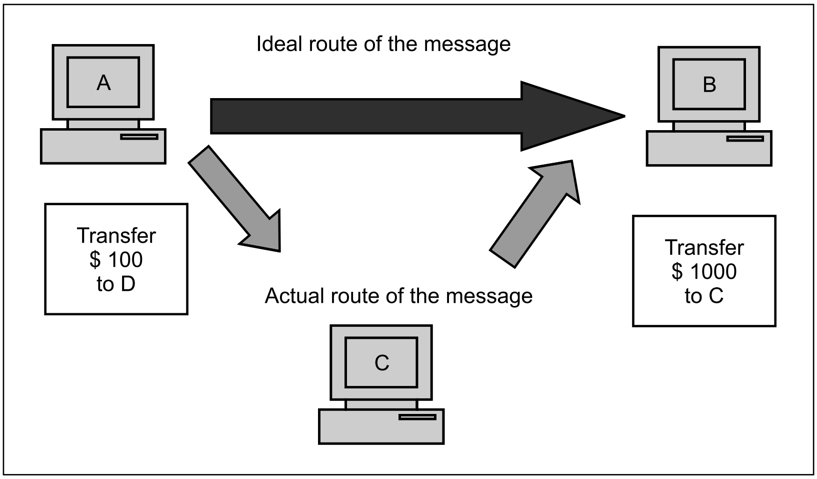


Fig. 1.6 Loss of integrity

the message were changed after user A had sent it. User A also does not know about this change. This type of attack is called **modification**.

Modification causes loss of message integrity.

1.4.4 Non-repudiation

There are situations where a user sends a message, and later on refuses that she had sent that message. For instance, user A could send a funds transfer request to bank B over the Internet. After the bank performs the funds transfer as per A's instructions, A could claim that he/she never sent the funds transfer instruction to the bank! Thus, A repudiates, or denies, his/her funds transfer instruction. The principle of *non-repudiation* defeats such possibilities of denying something after having done it. This is shown in Fig. 1.7.

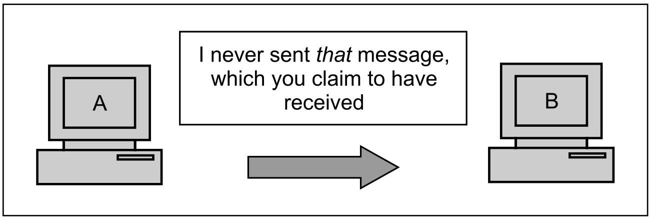

A
I never sent *that* message, which you claim to have received
B

Fig. 1.7 Establishing non-repudiation

Non-repudiation does not allow the sender of a message to refute the claim of not sending that message.

1.4.5 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, 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), whereas 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 (e.g. 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****.*

1.4.6 Availability

The principle of *availability* states that resources (i.e. information) should be available to authorized parties at all times. For 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. 1.8. This would defeat the principle of availability. Such an attack is called **interruption**.

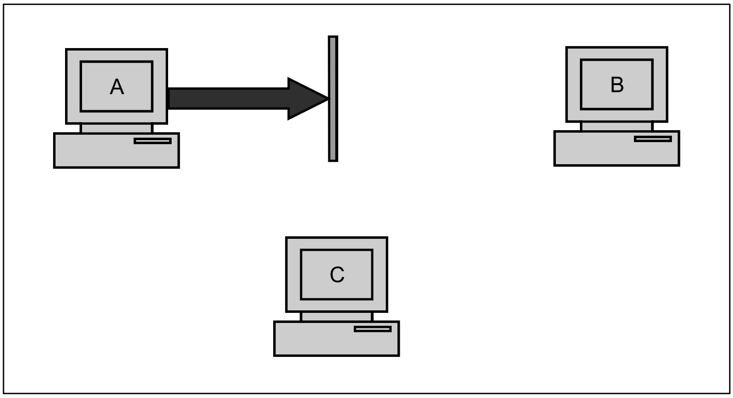


Fig. 1.8 Attack on availability

Interruption puts the availability of resources in danger.

We may be aware of the traditional OSI standard for Network Model (titled OSI Network Model 7498-1), which describes the seven layers of the networking technology (application, presentation, session, transport, network, data link, and physical). A very less known standard on similar lines is the **OSI standard for Security Model** (titled OSI Security Model 7498-2). This also defines seven layers of security in the form of

- Authentication
- Access control

- Non-repudiation
- Data integrity
- Confidentiality
- Assurance or availability
- Notarization or signature

We shall be touching upon most of these topics in this book.

Having discussed the various principles of security, let us now discuss the various types of attacks that are possible, from a technical perspective.

1.4.7 Ethical and Legal Issues

Many ethical issues (and legal issues) in computer security systems seem to be in the area of the individual's right to privacy versus the greater good of a larger entity (e.g. a company, society, etc.) Some examples are tracking how employees use computers for crowd surveillance, managing customer profiles, tracking a person's travel with a passport, so as to spam their cell phone with text-message advertisements), and so on. A key concept in resolving this issue is to find out a person's expectation of privacy.

Classically, the ethical issues in security systems are classified into the following four categories:

Privacy This deals with the right of an individual to control personal information.

Accuracy This talks about the responsibility for the authenticity, fidelity, and accuracy of information.

Property Here, we find out the owner of the information. We also talk about who controls access.

Accessibility This deals with the issue of what information does an organization have the right to collect? And in that situation, it also expects to know what the measures are, which will safeguard against any unforeseen eventualities.

Privacy is the protection of personal or sensitive information. Individual privacy is the desire to be *left alone* as an extension of our *personal space* and may or may not be supported by local regulations or laws. Privacy is subjective. Different people have different ideas of what privacy is and how much privacy they will trade for safety or convenience.

When dealing with legal issues, we need to remember that there is a hierarchy of regulatory bodies that govern the legality of information security. We can roughly classify them as follows.

- **International**, e.g. International Cybercrime Treaty
- **Federal**, e.g. FERPA, GLB, HIPAA, DMCA, Teach Act, Patriot Act, Sarbanes-Oxley Act, etc.
- **State**, e.g. UCITA, SB 1386, etc.
- **Organization**, e.g. computer use policy

■ 1.5 TYPES OF ATTACKS ■

We shall classify attacks with respect to two views: the common person's view and a technologist's view.

1.5.1 Attacks: A General View

From a common person's point of view, we can classify attacks into three categories, as shown in Fig. 1.9.

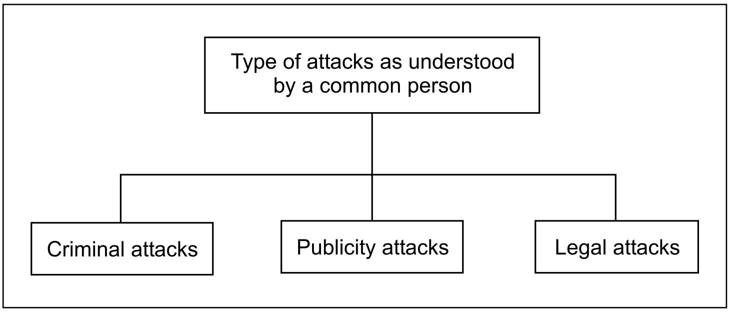


Fig. 1.9 Classification of attacks as understood in general terms

Let us now discuss these attacks.

1. Criminal Attacks

Criminal attacks are the simplest to understand. Here, the sole aim of the attackers is to maximize financial gain by attacking computer systems. Table 1.1 lists some forms of criminal attacks.

2. Publicity Attacks

Publicity attacks occur because the attackers want to see their names appear on television news channels and newspapers. History suggests that 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.

One form of publicity attacks is to damage (or deface) the Web pages of a site by attacking it. One of the most famous of such attacks occurred on the *US Department of Justice*'s Web site in 1996. The *New York Times* home page was also infamously defaced two years later.

3. Legal Attacks

This form of attack is quite novel and unique. Here, the attacker tries to make the judge or the jury doubtful about the security of a computer system. This works as follows. The attacker attacks the computer system, and the attacked party (say a bank or an organization) manages to take the attacker to the court. While the case is being fought, the attacker tries to convince the judge and the jury that there is inherent weakness in the computer system and that she has done nothing wrongful. The aim of the attacker is to exploit the weakness of the judge and the jury in technological matters.

For example, an attacker may sue a bank for performing an online transaction, which he/she never wanted to perform. In court, the attacker could innocently say something like: *The bank's Web site asked me to enter a password and that is all that I provided; I do not know what happened thereafter*. A judge is unwittingly likely to sympathize with the attacker!

Table 1.1 Types of criminal attacks

Attack	Description
Fraud	Modern fraud attacks concentrate on manipulating some aspects of electronic currency, credit cards, electronic stock certificates, checks, letters of credit, purchase orders, ATMs, etc.
Scams	Scams come in various forms, some of the most common ones being sale of services, auctions, multilevel marketing schemes, general merchandise, and business opportunities, etc. People are enticed to send money in return of great returns, but end up losing their money. A very common example is the *Nigeria scam*, where an email from Nigeria (and other African countries) entices people to deposit money into a bank account with a promise of hefty gains. Whosoever gets caught in this scam loses money heavily.
Destruction	Some sort of grudge is the motive behind such attacks. For example, unhappy employees attack their own organization, whereas terrorists strike at much bigger levels. For example, in the year 2000, there was an attack against popular Internet sites such as Yahoo!, CNN, eBay, Buy.com, Amazon.com, and e*Trade where authorized users of these sites failed to log in or access these sites.
Identity theft	This is best understood with a quote from Bruce Schneier: *Why steal from someone when you can just become that person?* In other words, an attacker does not steal anything from a legitimate user—he/she becomes that legitimate user! For example, it is much easier 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. Then 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, software, and so on.
Brand theft	It is quite easy to set up fake Web sites that look like real Web sites. How would a common user know if he/she is visiting the HDFC 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*.

1.5.2 Attacks: A Technical View

From a technical point of view, we can classify the types of attacks on computers and network systems into two categories for better understanding: (a) Theoretical concepts behind these attacks, and (b) Practical approaches used by the attackers. Let us discuss these one by one.

1. Theoretical Concepts

As we discussed earlier, the principles of security face threat from various attacks. These attacks are generally classified into four categories, as mentioned earlier. These are the following:

Interception It has been discussed in the context of *confidentiality* earlier. It means that an unauthorized party has gained access to a resource. The party can be a person, program, or computer-based system. Examples of interception are copying of data or programs, and listening to network traffic.

Fabrication It has been discussed in the context of *authentication* earlier. This involves the creation of illegal objects on a computer system. For example, the attacker may add fake records to a database.

Modification It has been discussed in the context of *integrity* earlier. Here, the attacker may modify the values in a database.

Interruption It has been discussed in the context of *availability* earlier. Here, the resource becomes unavailable, lost, or unusable. Examples of interruption are causing problems to a hardware device, erasing program, data, or operating-system components.

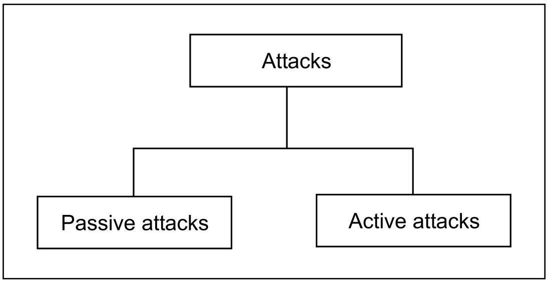


Fig. 1.10 Types of attacks

These attacks are further grouped into two types: **passive attacks** and **active attacks**, as shown in Fig. 1.10.

Let us discuss these two types of attacks now.

(a) Passive Attacks *Passive attacks* are those wherein the attacker indulges in eavesdropping or monitoring of data transmission. In other words, 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. In fact, this is also why passive attacks are harder to detect. Thus, 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.

Figure 1.11 shows further classification of passive attacks into two sub-categories. These categories are, namely **release of message contents** and **traffic analysis**.

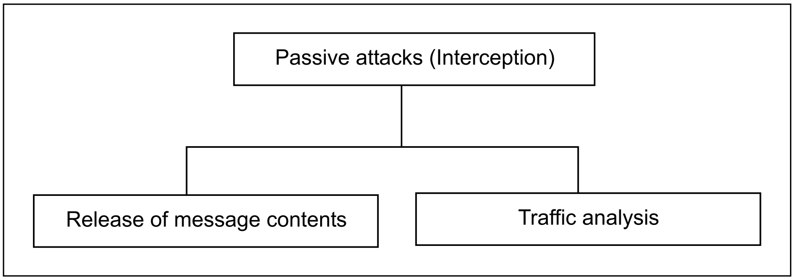


Fig. 1.11 Passive attacks

Release of message contents is quite simple to understand. When you send a confidential email message to your friend, you desire that only he/she be 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 the *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 her some clues regarding the communication that is taking place. Such attempts of analyzing (encoded) messages to come up with likely patterns are the work of the *traffic-analysis* attack.

(b) Active Attacks Unlike *passive attacks*, the *active attacks* are based on the modification of the original message in some manner, or in the creation of a false message. These attacks cannot be prevented easily. However, 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** attacks.
- Modification attacks can be classified further into **replay attacks** and **alteration of messages**.
- Fabrication causes **Denial Of Service (DOS)** attacks.

This classification is shown in Fig. 1.12.

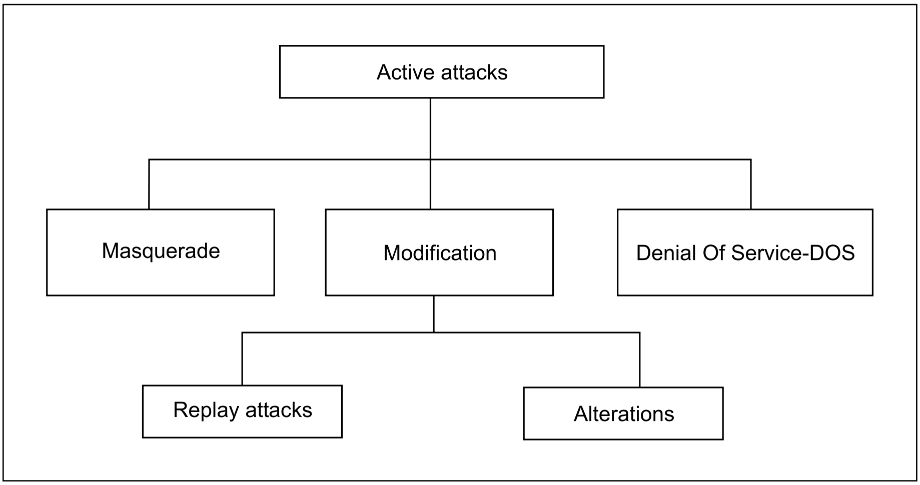


Fig. 1.12 Active attacks

Masquerade is caused when an unauthorized entity pretends to be another entity. As we have seen, 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. In masquerade attacks, usually some other forms of active attacks are also embedded. As an instance, the attack may involve capturing the user's authentication sequence (e.g. user ID and password). Later, those details can be replayed to gain illegal access to the computer system.

In a *replay attack*, 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 $1000 to D's account* to bank B. User C might capture this, and change it to

Transfer $10000 to C's account. Note that 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 in quick succession, so as to flood the network and deny other legitimate users to use the network facilities.

1.5.3 The Practical Side of Attacks

The attacks discussed earlier can come in a number of forms in real life. They can be classified into two broad categories: application-level attacks and network-level attacks, as shown in Fig. 1.13.

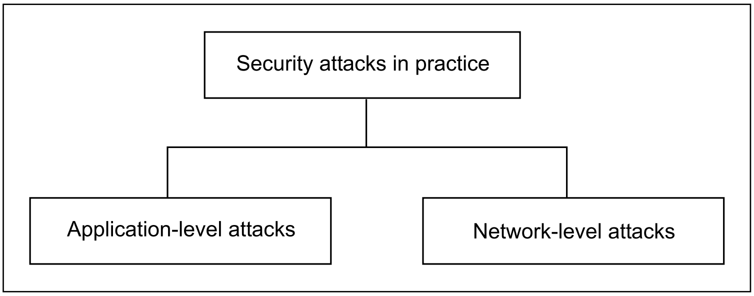


Fig. 1.13 Practical side of attacks

Let us discuss these now.

1. Application-level Attacks

These attacks happen at an application level in the sense that the attacker attempts to access, modify, or prevent access to information of a particular application, or the application itself. Examples of this are 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.

2. 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. Note that this automatically can lead to application-level attacks, because once someone is able to gain access to a network, usually he/she is able to access/modify at least some sensitive information, causing havoc.

These two types of attacks can be attempted by using various mechanisms, as discussed next. We will not classify these attacks into the above two categories, since they can span across application as well as network levels.

Security attacks can happen at the application level or the network level.

1.5.4 Programs that Attack

Let us now discuss a few programs that attack computer systems to cause some damage or to create confusion.

1. Virus

One can launch an application-level attack or a network level attack using a **virus**. In simple terms, a virus is a piece of program code that attaches itself to legitimate program code, and runs when the legitimate program runs. It can then infect other programs in that computer, or programs that are in other computers but on the same network. This is shown in Fig. 1.14. 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.

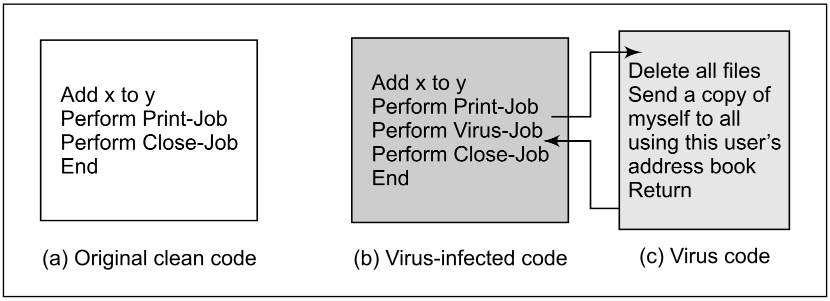


Fig. 1.14 Virus

Viruses can also be triggered by specific events (e.g. a virus could automatically execute at 12 p.m. every day). Usually viruses cause damage to computer and network systems to the extent that they can be repaired, assuming that the organization deploys good backup and recovery procedures.

A virus is a computer program that attaches itself to another legitimate program, and causes damage to the computer system or to the network.

During its lifetime, a virus goes through four phases:

(a) Dormant Phase Here, the virus is idle. It gets activated based on a certain action or event (e.g. the user typing a certain key or a 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 itself, 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 virus. 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.

There is another popular category of viruses, called the **macro virus**. This virus affects specific application software, such as Microsoft Word or Microsoft Excel. They affect the documents created by users, and spread quite easily since such documents are very commonly exchanged over email. There is a feature called *macro* in these application-software programs, which allows users to write small, useful, utility programs within the documents. Viruses attack these macros, and hence the name *macro virus*.

2. Worm

Similar in concept to a virus, a **worm** is actually different in implementation. A virus modifies a program (i.e. it attaches itself to the program under attack). A worm, however, does not modify a program. Instead, it replicates itself again and again. This is shown in Fig. 1.15. The replication grows so much that ultimately the computer or the network on which the worm resides, becomes very slow, ultimately coming to a halt. Thus, 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.

> *A worm does not perform any destructive actions, and instead, only consumes system resources to bring it down.*

3. Trojan Horse

A Trojan horse is a hidden piece of code, like a virus. However, the purpose of a Trojan horse is different. Whereas the main purpose of a virus is to make some sort of modifications to the target computer or network, a Trojan horse attempts to reveal confidential information to an attacker. The name (Trojan horse) comes from the epic poem *Iliad*. The story says that Greek soldiers hid inside a large hollow horse, which was pulled into the city of Troy by its citizens, unaware of its *contents*. Once the Greek soldiers entered the city of Troy, they opened the gates for the rest of the Greek soldiers.

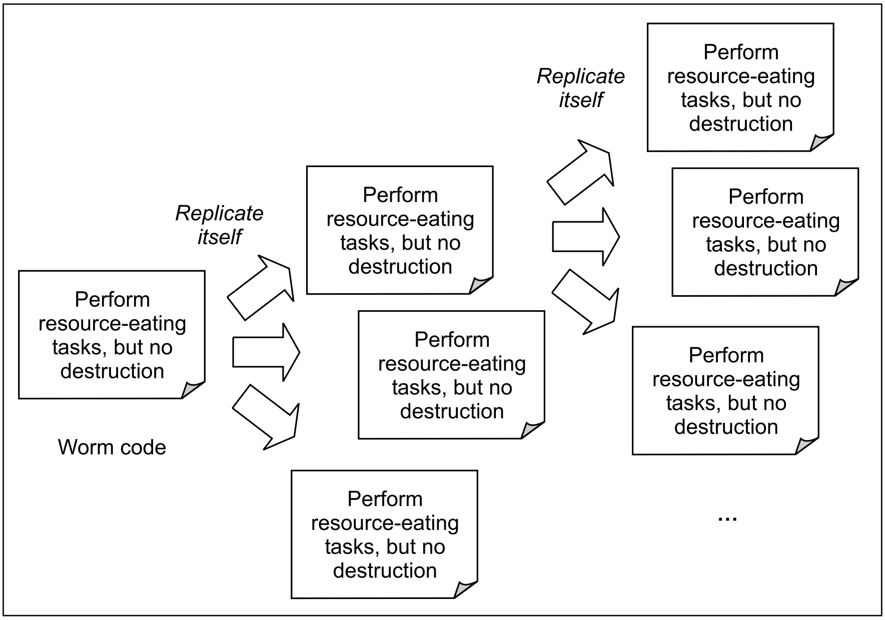

Perform resource-eating tasks, but no destruction
Worm code
Replicate itself
Perform resource-eating tasks, but no destruction
Perform resource-eating tasks, but no destruction
Perform resource-eating tasks, but no destruction
Replicate itself
Perform resource-eating tasks, but no destruction
Perform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

Fig. 1.15 Worm

In a similar fashion, 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. The attacker can then merrily misuse the user id and password to gain access to the system. This is shown in Fig. 1.16.

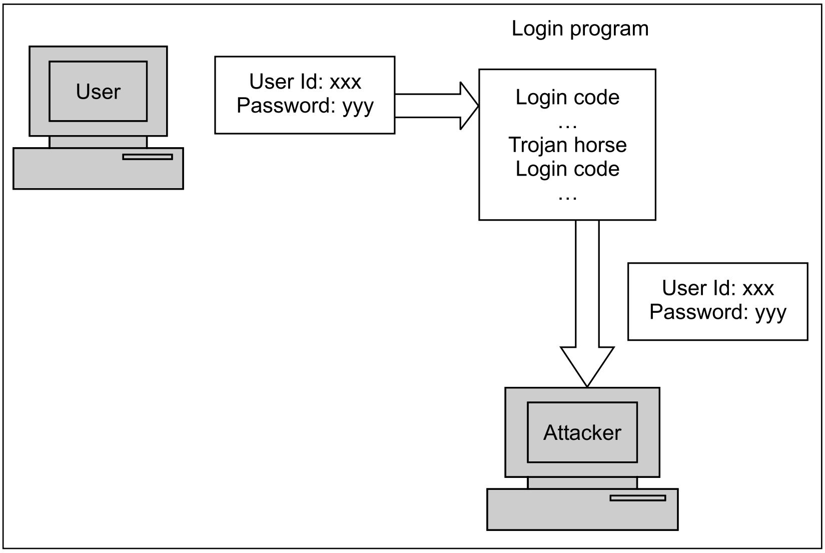


Fig. 1.16 Trojan horse

A Trojan horse allows an attacker to obtain some confidential information about a computer or a network.

1.5.5 Dealing with Viruses

Preventing viruses is the best option. However, in today's world, it is almost impossible to achieve cent per cent security given that the world is connected to the Internet all the time. We have to accept that viruses will attack, and we would need to find ways to deal with them. Hence, we can attempt to detect, identify, and remove viruses. This is shown in Fig. 1.17.

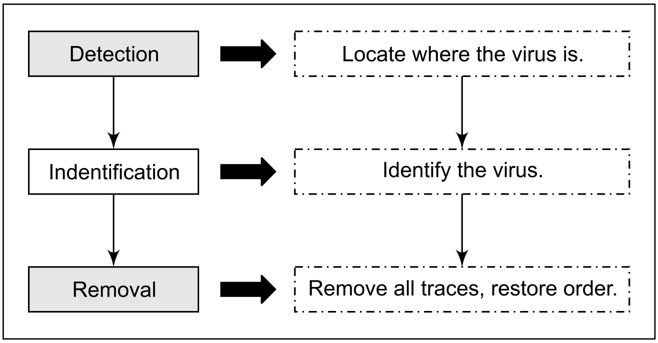


Fig. 1.17 Virus-elimination steps

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 depicted in Fig. 1.18.

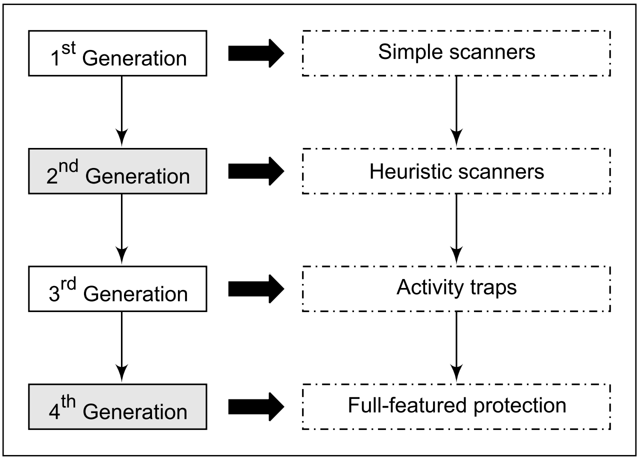


Fig. 1.18 Generations of Anti-virus software

Let us summarize the key characteristics of the four generations of anti-virus software.

1. First Generation

These anti-virus software programs were called *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.

2. Second 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. For example, such a program could look for an encryption key used by a virus, find it, decrypt and remove the virus, and clean the code. Another variation of these anti-virus programs used to store some identification about the file (e.g. a message digest, which we shall study later) are also notorious for detecting changes in the contents of the file.

3. Third 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.

4. Fourth Generation

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

There is a category of software called **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
- Attempts 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.

1.5.6 Specific Attacks

1. Sniffing and Spoofing

On the Internet, computers exchange messages with each other in the form of small groups of data, called packets. A packet, like a postal envelope 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 **snooping**), and (b) **Packet spoofing**. Since the protocol used in this communication is called Internet Protocol (IP), other names for these two attacks are (a) **IP sniffing**, and (b) **IP spoofing**. The meaning remains the same.

Let us discuss these two attacks.

(a) Packet Sniffing Packet sniffing is a passive attack on an ongoing conversation. An attacker need not *hijack* a conversation, but instead, can simply observe (i.e. *sniff*) packets as they pass by. Clearly, 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 ways, or (ii) The transmission link itself can be encoded. To read a packet, the attacker somehow needs to access it in the first place. The simplest way to do this is to control a computer via which the traffic goes through. 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 an incorrect source address. When this happens, the receiver (i.e. the party who receives these packets containing false addresses) would inadvertently send replies back to this forged address (called **spoofed address**), and not to the attacker. 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.

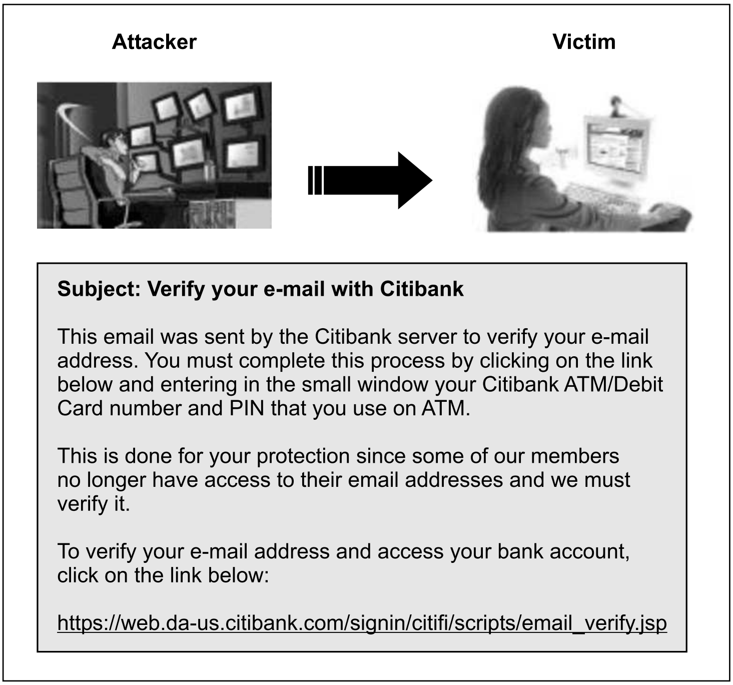
2. Phishing

Phishing has become a big problem in recent times. In 2004, the estimated losses due to phishing were to the tune of USD 137 million, according to Tower Group. Attackers set up fake Web sites, which look like real Web sites. It is quite simple to do so, since creating Web pages involves relatively simple technologies such as HTML, JavaScript, CSS (Cascading Style Sheets), etc. Learning and using these technologies is quite simple. The attacker's modus operandi works as follows.

- The attacker decides to create his/her 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 the human eye will not be able to distinguish between the real (Citibank's) and fake (attacker's) site.

- The attacker can use many techniques to attack the bank's customers. We illustrate the most common one below.

The attacker sends an email to the legitimate customers of the bank. The email itself appears to have 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.com). This fake email warns the user that there has been some sort of attack on 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. This is conceptually shown in Fig. 1.19.



Subject: Verify your e-mail with Citibank

This email was sent by the Citibank server to verify your e-mail address. You must complete this process by clicking on the link below and entering in the small window your Citibank ATM/Debit Card number and PIN that you use on ATM.

This is done for your protection since some of our members no longer have access to their email addresses and we must verify it.

To verify your e-mail address and access your bank account, click on the link below:

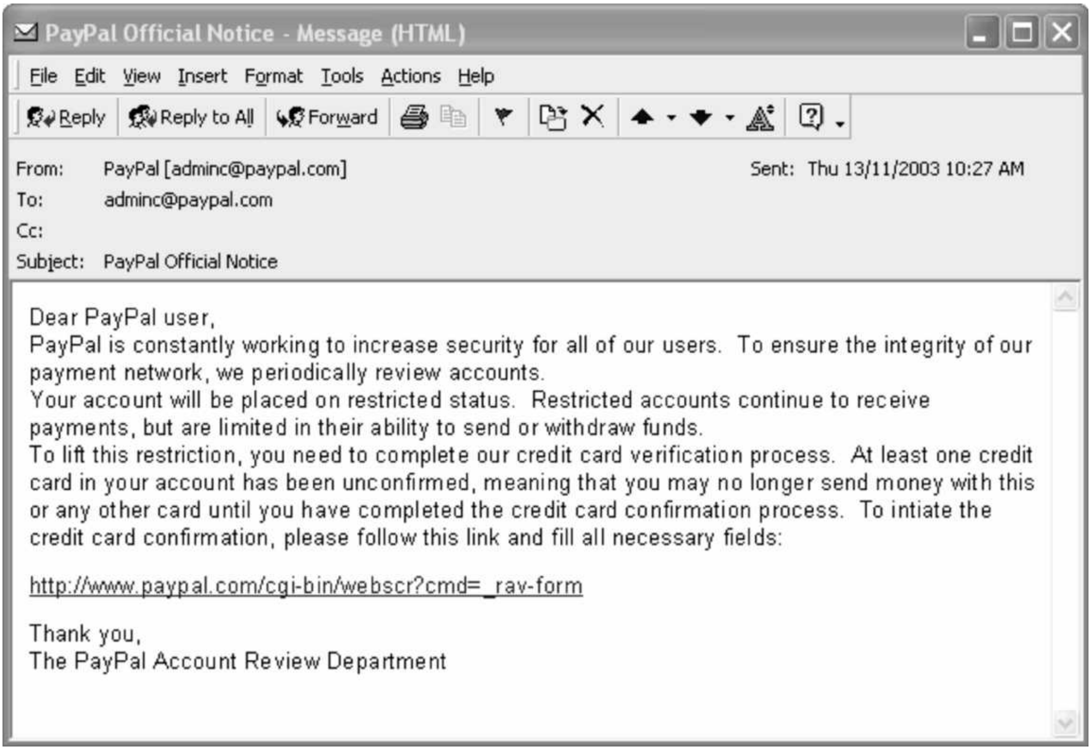
https://web.da-us.citibank.com/signin/citifi/scripts/email_verify.jsp

Fig. 1.19 Attacker sends a forged email to the innocent victim (customer)

- When the customer (i.e. the victim) innocently clicks on the URL specified in the email, he/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 his/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!

A real-life example of this kind of attack is reproduced below from the site http://www.fraudwatchinternational.com.

Figure 1.20 shows a fake email sent by an attacker to an authorized PayPal user.


PayPal Official Notice - Message (HTML)
File Edit View Insert Format Tools Actions Help
Reply Reply to All Forward

From: PayPal [adminc@paypal.com] Sent: Thu 13/11/2003 10:27 AM
To: adminc@paypal.com
Cc:
Subject: PayPal Official Notice

Dear PayPal user,
PayPal is constantly working to increase security for all of our users. To ensure the integrity of our payment network, we periodically review accounts.
Your account will be placed on restricted status. Restricted accounts continue to receive payments, but are limited in their ability to send or withdraw funds.
To lift this restriction, you need to complete our credit card verification process. At least one credit card in your account has been unconfirmed, meaning that you may no longer send money with this or any other card until you have completed the credit card confirmation process. To intiate the credit card confirmation, please follow this link and fill all necessary fields:

http://www.paypal.com/cgi-bin/webscr?cmd=_rav-form

Thank you,
The PayPal Account Review Department

Fig. 1.20 Fake email from the attacker to a PayPal user

As we can see, the attacker is trying to fool the PayPal customer to verify his/her credit-card details. Quite clearly, the aim of the attacker is to access the credit-card information of the customer and then misuse it. Figure 1.21 shows the screen that appears when the user clicks on the URL specified in the fake email.

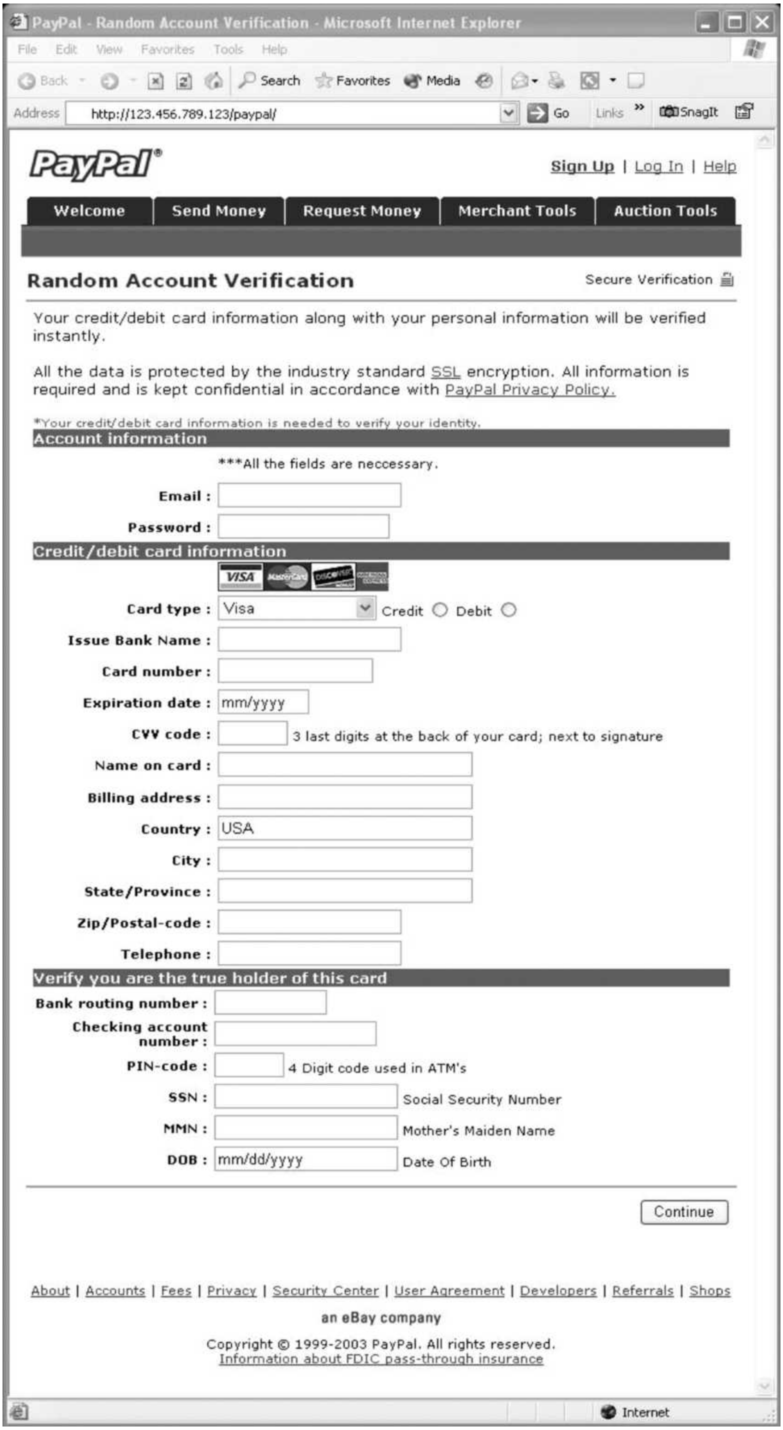
Once the user provides these details, the attacker's job is easy! He/she simply uses these credit-card details to make purchases on behalf of the cheated card holder!

3. Pharming (DNS Spoofing)

Another attack, known earlier as **DNS spoofing** or **DNS poisoning**, is now called **pharming** attack. As we know, using 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 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. With this background, 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


PayPal - Random Account Verification - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://123.456.789.123/paypal/ Go Links SnagIt
PayPal®
Sign Up | Log In | Help
Welcome
Send Money
Request Money
Merchant Tools
Auction Tools
Internet

Random Account Verification Secure Verification

Your credit/debit card information along with your personal information will be verified instantly.

All the data is protected by the industry standard SSL encryption. All information is required and is kept confidential in accordance with PayPal Privacy Policy.

*Your credit/debit card information is needed to verify your identity.

Account information

***All the fields are neccessary.

Email :

Password :

Credit/debit card information

VISA MasterCard DISCOVER AMERICAN EXPRESS

Card type : Visa Credit ○ Debit ○

Issue Bank Name :

Card number :

Expiration date : mm/yyyy

CVV code : 3 last digits at the back of your card; next to signature

Name on card :

Billing address :

Country : USA

City :

State/Province :

Zip/Postal-code :

Telephone :

Verify you are the true holder of this card

Bank routing number :

Checking account number :

PIN-code : 4 Digit code used in ATM's

SSN : Social Security Number

MMN : Mother's Maiden Name

DOB : mm/dd/yyyy Date Of Birth

Continue

About | Accounts | Fees | Privacy | Security Center | User Agreement | Developers | Referrals | Shops

an eBay company

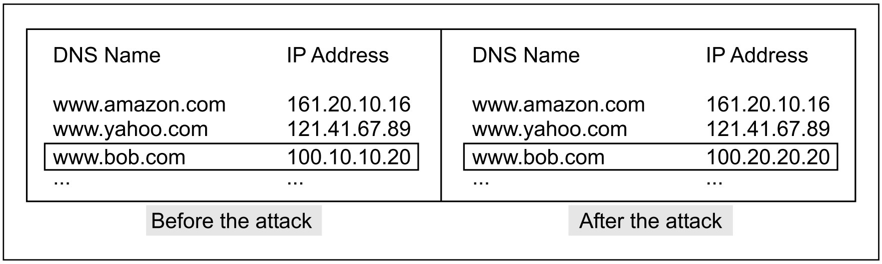
Copyright © 1999-2003 PayPal. All rights reserved.
Information about FDIC pass-through insurance

Fig. 1.21 Fake PayPal site asking for user's credit-card details

- The attacker (say, Trudy) manages to hack and replace the IP address of Bob with her own (say 100.20.20.20) in the DSN 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 Fig. 1.22.



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

Fig. 1.22 Effect of the DNS 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. Even worse, the attacker (Trudy) does not have to listen to the conversation on the wire! She has to simply be able to hack the DNS server of the ISP and replace a single IP address with her own!

A protocol called **DNSSec (Secure DNS)** is being used to thwart such attacks. Unfortunately, it is not widely used.

Summary

- Network and Internet security has gained immense prominence in the last few years, as conducting business using these technologies have become very crucial.
- Automation of attacks, privacy concerns, and distance becoming immaterial are some of the key characteristics of modern attacks.
- The principles of any security mechanism are confidentiality, authentication, integrity, non-repudiation, access control, and availability.
- Confidentiality specifies that only the sender and the intended recipients should be able to access the contents of a message.
- Authentication identifies the user of a computer system, and builds a trust with the recipient of a message.

- Integrity of a message should be preserved as it travels from the sender to the recipient. It is compromised if the message is modified during transit.
- Non-repudiation ensures that the sender of a message cannot refute the fact of sending that message in case of disputes.
- Access control specifies what users can do with a network or Internet system.
- Availability ensures that computer and network resources are always available to the legitimate users.
- Attacks on a system can be classified into interception, fabrication, modification, and interruption.
- Common way of classifying attacks is to categorize them into criminal, publicity, and legal attacks.
- Attacks can also be classified into passive and active attacks.
- In passive attacks, the attacker does not modify the contents of a message.
- Active attacks involve modification of the contents of a message.
- Release of message contents and traffic analysis are types of passive attacks.
- Masquerade, replay attacks, alteration of messages and Denial Of Service (DOS) are types of active attacks.
- Another way to classify attacks is application-level attacks and network-level attacks.
- Viruses, worms, Trojan horses and Java applets, ActiveX controls can practically cause attacks on a computer system.
- Java offers a high amount of security in programming, if implemented correctly.
- Sniffing and spoofing cause packet-level attacks.
- Phishing is a new attack which attempts to fool legitimate users to provide their confidential information to fake sites.
- DNS spoofing or pharming attack involves changing the DNS entries so that users are redirected to an invalid site, while they keep thinking that they have connected to the right site.

Key Terms and Concepts

- Access Control List (ACL)
- ActiveX control
- Application-level attack
- Authentication
- Behavior-blocking software
- Denial Of Service (DOS) attack
- Identity theft
- Interception
- Java applet
- Modification
- Active attack
- Alteration of message
- Attacker
- Availability
- Confidentiality
- Fabrication
- Integrity
- Interruption
- Masquerade
- Network-level attack

- Non-repudiation
- Phishing
- Release of message contents
- Signed Java applet
- Trojan horse
- Worm
- Passive attack
- Pharming
- Replay attack
- Traffic analysis
- Virus

PRACTICE SET

Multiple-Choice Questions

1. The principle of ____________ ensures that only the sender and the intended recipients have access to the contents of a message.
   (a) confidentiality
   (b) authentication
   (c) integrity
   (d) access control
2. If the recipient of a message has to be satisfied with the identify of the sender, the principle of _____________ comes into picture.
   (a) confidentiality
   (b) authentication
   (c) integrity
   (d) access control
3. If we want to ensure the principle of ___________, the contents of a message must not be modified while in transit.
   (a) confidentiality
   (b) authentication
   (c) integrity
   (d) access control
4. The principle of ___________ ensures that the sender of a message cannot later claim that the message was never sent.
   (a) access control
   (b) authentication
   (c) availability
   (d) non-repudiation
5. The __________ attack is related to confidentiality.
   (a) interception
   (b) fabrication
   (c) modification
   (d) interruption
6. The __________ attack is related to authentication.
   (a) interception
   (b) fabrication
   (c) modification
   (d) interruption
7. The __________ attack is related to integrity.
   (a) interception
   (b) fabrication
   (c) modification
   (d) interruption
8. The __________ attack is related to availability.
   (a) interception
   (b) fabrication
   (c) modification
   (d) interruption
9. In _________ attacks, there is no modification to message contents.
   (a) passive
   (b) active
   (c) both of the above
   (d) none of the above

10. In _________ attacks, the message contents are modified.
    (a) passive
    (b) active
    (c) both of the above
    (d) none of the above
11. Interruption attacks are also called ___________ attacks.
    (a) masquerade
    (b) alteration
    (c) denial of service
    (d) replay attacks
12. DOS attacks are caused by _________.
    (a) authentication
    (b) alteration
    (c) fabrication
    (d) replay attacks
13. Virus is a computer _________.
    (a) file
    (b) program
    (c) database
    (d) network
14. A worm _____ modify a program.
    (a) does not
    (b) does
    (c) may or may not
    (d) may
15. A _____ replicates itself by creating its own copies, in order to bring the network to a halt.
    (a) virus
    (b) worm
    (c) Trojan horse
    (d) bomb

■ Exercises

1. Find out more examples of security attacks reported in the last few years.
2. What are the key principles of security?
3. Why is confidentiality an important principle of security? Think about ways of achieving the same. (*Hint:* Think about the ways in which children use a secret language).
4. Discuss the reasons behind the significance of authentication. Find out the simple mechanisms of authentication. (*Hint:* What information do you provide when you use a free email service such as Yahoo or Hotmail?)
5. In real life, how is message integrity ensured? (*Hint*: On what basis is a check honored or dishonored?)
6. What is repudiation? How can it be prevented in real life? (*Hint:* Think what happens if you issue a check, and after the bank debits your account with the amount therein, you complain to the bank that you never issued that check).
7. What is access control? How different is it from availability?
8. Why are some attacks called passive? Why are other attacks called active?
9. Discuss any one passive attack.
10. What is 'masquerade'? Which principle of security is breached because of that?
11. What are 'replay attacks'? Give an example of replay attacks.
12. What is 'denial of service' attack?
13. What is a 'worm'? What is the significant difference between a 'worm' and a 'virus'?
14. Discuss the concepts of 'phishing' and 'pharming'.
15. Would message integrity on its own ensure that the contents of a message are not changed during transit? Does something more needs to be done?

Design/Programming Exercises

1. Write a C program that contains a string (char pointer) with a value ‘Hello World’. The program should XOR each character in this string with 0 and display the result. Repeat the exercise by an XOR operation with 1.
2. Write a C program that contains a string (char pointer) with a value ‘Hello World’. The program should AND, OR and XOR each character in this string with 127 and display the result. Why are these results different?
3. Study ‘phishing’ in more detail. Find out which popular bank sites have been phished and how.
4. Think about offering phishing-prevention techniques. Which ones of them would be most effective, and why?
5. Why is it easier to fall prey to ‘pharming’ than ‘phishing’? Explain in technical terms.
6. Often, it is said that a technology called SSL can prevent ‘phishing’ and ‘pharming’. Is it always true? Why?
7. Write a small viruslike program in plain English language that accepts a file name and changes every character in the file to an asterisk.
8. How is DNS secured? Are standard protocols available?
9. Study what is meant by Nigerian Fraud and how it can be prevented.
10. What is the online lottery scam? How does it work?
11. What tricks do attackers use to hack into online banking accounts?
12. Study what is meant by social engineering and how it works.
13. Who is Kevin Mitnick? Why is he well known?
14. What threats do attacks on social networking sites pose? How can those be prevented?
15. Which tools are popularly used by attackers to attack Web sites?