# 7.1 Malicious logic

The most sophisticated types of threats to computer systems are presented by programs that exploit vulnerabilities in computing systems. Such threats are referred to as malicious software, or malware. Malicious software is software that is intentionally included or inserted in a system for a harmful purpose. Malicious software provides threats to application programs as well as utility programs, such as editors and compilers, and kernel-level programs.

## Types of malicious software

Malicious software can be divided into two categories: those that need a host program, and those that are independent. The former, referred to as parasitic, are essentially fragments of programs that cannot exist independently of some actual application program, utility, or system program. Viruses, logic bombs, and backdoors are examples. Independent malware is a self-contained program that can be scheduled and run by the operating system. Worms and bot programs are examples.

We can also differentiate between those software threats that do not replicate and those that do. The former are programs or fragments of programs that are activated by a trigger. Examples are logic bombs, backdoors, and bot programs. The latter consist of either a program fragment or an independent program that, when executed, may produce one or more copies of itself to be activated later on the same system or some other system. Viruses and worms are examples.

### Virus

A computer virus is a piece of software that can “infect” other programs by modifying them; the modification includes injecting the original program with a routine to make copies of the virus program, which can then go on to infect other programs. Computer viruses first appeared in the early 1980s. Biological viruses are tiny scraps of genetic code—dna or rna—that can take over the machinery of a living cell and trick it into making thousands of flawless replicas of the original virus. Like its biological counterpart, a computer virus carries in its instructional code the recipe for making perfect copies of itself. The typical virus becomes embedded in a program on a computer. Then, whenever the infected computer comes into contact with an uninfected piece of software, a fresh copy of the virus passes into the new program. Thus, the infection can be spread

from computer to computer by unsuspecting users who either swap disks or send programs to one another over a network. In a network environment, the ability to access applications and system services on other computers provides a perfect culture for the spread of a virus.

A virus can do anything that other programs do. The difference is that a virus attaches itself to another program and executes secretly when the host program is run. Once a virus is executing, it can perform any function, such as erasing files and programs that is allowed by the privileges of the current user.

A computer virus has three parts:

• **Infection mechanism**: the means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the infection vector.

• **Trigger**: the event or condition that determines when the payload is activated or delivered.

• **Payload:** what the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity.

During its lifetime, a typical virus goes through the following four phases:

• **Dormant phase**: the virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.

• **Propagation phase**: the virus places a copy of itself into other programs or into certain system areas on the disk. The copy may not be identical to the propagating version; viruses often morph to evade detection. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

• **Triggering phase**: the virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.

• **Execution phase:** the function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

**Viruses' classification**

There has been a continuous arms race between virus writers and writers of antivirus software since viruses first appeared. As effective countermeasures are developed for existing types of viruses, newer types are developed. There is no simple or universally agreed upon classification scheme for viruses. But we can classify viruses along two orthogonal axes: the type of target the virus tries to infect and the method the virus uses to conceal itself from detection by users and antivirus software.

A **virus classification by target** includes the following categories:

• **Boot sector infector**: infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.

• F**ile infector**: infects files that the operating system or shell consider to be executable.

• **Macro virus**: infects files with macro code that is interpreted by an application.

A **virus classification by concealment** strategy includes the following categories:

• E**ncrypted virus:** a typical approach is as follows. A portion of the virus creates a random encryption key and encrypts the remainder of the virus. The key is stored with the virus. When an infected program is invoked, the virus uses the stored random key to decrypt the virus. When the virus replicates, a different random key is selected. Because the bulk of the virus is encrypted with a different key for each instance, there is no constant bit pattern to observe.

• **Stealth virus**: a form of virus explicitly designed to hide itself from detection by antivirus software. Thus, the entire virus, not just a payload is hidden.

• **Polymorphic virus**: a virus that mutates with every infection, making detection by the "signature" of the virus impossible.

• **Metamorphic virus**: as with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses may change their behavior as well as their appearance.

## Worms

A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function. A worm actively seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on other machines.

Network worm programs use network connections to spread from system to system. Once active within a system, a network worm can behave as a computer virus or bacteria, or it could implant Trojan horse programs or perform any number of disruptive or destructive actions.

To replicate itself, a network worm uses some sort of network vehicle. Examples include the following:

• **Electronic mail facility**: a worm mails a copy of itself to other systems, so that its code is run when the e-mail or an attachment is received or viewed.

• **Remote execution capability:** a worm executes a copy of itself on another system, either using an explicit remote execution facility or by exploiting a program flaw in a network service to subvert its operations.

• **Remote login capability:** a worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other, where it then executes.

The new copy of the worm program is then run on the remote system where, in addition to any functions that it performs at that system, it continues to spread in the same fashion.

A network worm exhibits the same characteristics as a computer virus: **a dormant phase, a propagation phase, a triggering phase, and an execution phase.** The propagation phase generally performs the following functions:

1. Search for other systems to infect by examining host tables or similar repositories of remote system addresses.

2. Establish a connection with a remote system.

3. Copy itself to the remote system and cause the copy to be run.

The network worm may also attempt to determine whether a system has previously been infected before copying itself to the system. In a multiprogramming system, it may also disguise its presence by naming itself as a system process or using some other name that may not be noticed by a system operator. As with viruses, network worms are difficult to counter.

## Trojan horse

A Trojan horse, or Trojan, is a type of malicious code or software that looks legitimate but can take control of your computer. A Trojan is designed to damage, disrupt, steal, or in general inflict some other harmful action on your data or network.

Trojans take their name from the hollow wooden horse that the Greeks hid inside of during the Trojan War. The Trojans, thinking the horse was a gift, opened their walled city to accept it, allowing the Greeks to come out of hiding at night to attack the sleeping Trojans.

A Trojan acts like a bona fide application or file to trick you. It seeks to deceive you into loading and executing the malware on your device. Once installed, a Trojan can perform the action it was designed for.

A Trojan is sometimes called a Trojan virus or a Trojan horse virus, but that’s a misnomer. Viruses can execute and replicate themselves. A Trojan cannot. A user has to execute Trojans. Even so, Trojan malware and Trojan virus are often used interchangeably.

Trojans are generally spread by some form of social engineering, for example where a user is fooled into executing an email attachment disguised to appear not suspicious, (e.g., a routine form to be filled in), or by clicking on some fake advertisement on social media or anywhere else. Although their payload can be anything, many modern forms act as a backdoor, contacting a controller which can then have unauthorized access to the affected computer. Trojans may allow an attacker to access users' personal information such as banking information, passwords, or personal identity. It can also delete a user's files or infect other devices connected to the network. Ransom-ware attacks are often carried out using a Trojan.

Some Trojans take advantage of a security flaw in older versions of internet explorer and Google chrome to use the host computer as an anonymizer proxy to effectively hide internet usage,

enabling the controller to use the internet for illegal purposes while all potentially incriminating evidence indicates the infected computer or its IP address. The host's computer may or may not show the internet history of the sites viewed using the computer as a proxy. The first generation of anonymizer Trojan horses tended to leave their tracks in the page view histories of the host computer. Later generations of the Trojan tend to "cover" their tracks more efficiently. Several versions of sub7 have been widely circulated in the us and Europe and became the most widely distributed examples of this type of Trojan.

## Zombies

In computing, a zombie is a computer connected to a network that has been compromised by a hacker, a virus or a Trojan. It can be used remotely for malicious tasks. The term originated in the West Indies, where a zombie is a will-less, automaton-like person who is said to have been revived from the dead and must now do the will of the living

Most owners of zombie computers do not realize that their system is being used in this way, hence the comparison with the living dead. They are also used in ddos attacks in coordination with botnets in a way that resembles the typical zombie attacks of horror films.

*What are they used for?*

Zombies are frequently used in Denial-Of-Service Attacks (DDoS), which refers to the saturation of websites with a multitude of computers accessing at the same time. As so many users are making requests at the same time to the server hosting the web page, the server crashes, denying access to genuine users.

A variant of this type of saturation is known as **degradation-of-service** attack and uses 'pulsing zombies': degradation of the service by periodically saturating the websites at a low intensity, with the intention of slowing down, instead of blocking, the targeted website. Such attacks are difficult to detect, as the slow service may go undetected for months or even years or is simply assumed to be due to other problems.

Zombies have also been used for sending spam. In 2005, it was estimated that between 50% and 80% of all spam in circulation had been sent by zombie computers. This technique is useful for

criminals as it helps them avoid detection and at the same time reduce bandwidth costs (as the owners of the zombies will bear the cost).

This type of spam is also used for spreading Trojans, as this type of malware is not self-replicating but relies on circulation via email in order to spread, unlike worms that spread via other means. For similar reasons, zombies are also used for fraud against sites with pay-per-click contextual ads, artificially increasing the number of hits.

## Denial of service

A "Denial of Service" or **DoS** attack is used to tie up a website's resources so that users who need to access the site cannot do so. Many major companies have been the focus of dos attacks. Because a dos attack can be easily engineered from nearly any location, finding those responsible can be extremely difficult.

A bit of history: the first dos attack was done by 13-year-old David Dennis in 1974. Dennis wrote a program using the "external" or "ext" command that forced some computers at a nearby university research lab to power off.

DoS attacks have evolved into the more complex and sophisticated "Distributed Denial of Service" (DDoS) attacks. The biggest attack ever recorded — at that time — targeted code-hosting-service github in 2018. Attackers include hacktivists (hackers whose activity is aimed at promoting a social or political cause), profit-motivated cybercriminals, and nation states.

DoS attacks generally take one of two forms. They either flood web services or crash them.

### Flooding attacks

Flooding is the more common form dos attack. It occurs when the attacked system is overwhelmed by large amounts of traffic that the server is unable to handle. The system eventually stops.

An ICMP flood — also known as a ping flood — is a type of DoS attack that sends spoofed packets of information that hit every computer in a targeted network, taking advantage of misconfigured network devices.

A SYN flood is a variation that exploits a vulnerability in the TCP connection sequence. This is often referred to as the three-way handshake connection with the host and the server. Here's how it works:

The targeted server receives a request to begin the handshake. But, in a syn flood, the handshake is never completed. That leaves the connected port as occupied and unavailable to process further requests. Meanwhile, the cybercriminal continues to send more and more requests, overwhelming all open ports and shutting down the server.

**Crash attacks**

Crash attacks occur less often, when cybercriminals transmit bugs that exploit flaws in the targeted system. The result? The system crashes. Crash attacks — and flooding attacks — prevent legitimate users from accessing online services such as websites, gaming sites, email, and bank accounts.

*How a DoS attack works*

Unlike a virus or malware, a dos attack doesn't depend on a special program to run. Instead, it takes advantage of an inherent vulnerability in the way computer networks communicate.

Here's an example. Suppose you wish to visit an e-commerce site in order to shop for a gift. Your computer sends a small packet of information to the website. The packet works as a "hello" – basically, your computer says, "Hi, I'd like to visit you, please let me in."

When the server receives your computer's message, it sends a short one back, saying in a sense, "ok, are you real?" Your computer responds — "yes!" — And communication is established.

The website's homepage then pops up on your screen, and you can explore the site. Your computer and the server continue communicating as you click links, place orders, and carry out other business.

In a DoS attack, a computer is rigged to send not just one "introduction" to a server, but hundreds or thousands. The server — which cannot tell that the introductions are fake — sends back its usual response, waiting up to a minute in each case to hear a reply. When it gets no reply, the server shuts down the connection, and the computer executing the attack repeats, sending a new batch of fake requests.

DoS attacks mostly affect organizations and how they run in a connected world. For consumers, the attacks hinder their ability to access services and information.

**Other types of attacks: DDoS**

Distributed denial of service (DDoS) attacks represent the next step in the evolution of dos attacks as a way of disrupting the internet. Cybercriminals began using DDoS attacks around 2000.

The attacks use large numbers of compromised computers, as well as other electronic devices — such as webcams and smart televisions that make up the ever-increasing internet of things — to force the shutdown of the targeted website, server or network.

Security vulnerabilities in internet-of-things devices can make them accessible to cybercriminals seeking to anonymously and easily launch DDoS attacks. In contrast, a DoS attack generally uses a single computer and a single ip address to attack its target, making it easier to defend against.

## 7.2 Intrusion

An intruder is a person who attempts to gain unauthorized access to a system, to damage that system, or to disturb data on that system. In summary, this person attempts to violate security by interfering with system availability, data integrity or data confidentiality.

One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker. Intruders can divided into three classes:

• **Masquerader**: an individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account

• **Misfeasor**: a legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges

• **Clandestine user**: an individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection

The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider.

### Intrusion techniques

The objective of the intruder is to gain access to a system or to increase the range of privileges accessible on a system. Most initial attacks use system or software vulnerabilities that allow a user to execute code that opens a back door into the system.

Alternatively, the intruder attempts to acquire information that should have been protected. In some cases, this information is in the form of a user password. With knowledge of some other user's password, an intruder can log in to a system and exercise all the privileges accorded to the legitimate user.

Typically, a system must maintain a file that associates a password with each authorized user. If such a file is stored with no protection, then it is an easy matter to gain access to it and learn passwords. The password file can be protected in one of two ways:

• **One-way function:** the system stores only the value of a function based on the user's password. When the user presents a password, the system transforms that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the one-way function and in which a fixed-length output is produced.

• **Access control**: access to the password file is limited to one or a very few accounts.

## Intrusion detection

Inevitably, the best intrusion prevention system will fail. A system's second line of defense is intrusion detection, and this has been the focus of much research in recent years. This interest is motivated by a number of considerations, including the following:

1. If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised. Even if the detection is not sufficiently timely to preempt the intruder, the sooner that the intrusion is detected, the less the amount of damage and the more quickly that recovery can be achieved.

2. An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.

3. Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified. Of course, we cannot expect that there will be a crisp, exact distinction between an attack by an intruder and the normal use of resources by an authorized user. Rather, we must expect that there will be some overlap.

Although the typical behavior of an intruder differs from the typical behavior of an authorized user, there is an overlap in these behaviors. In Anderson's study, it was postulated that one could, with reasonable confidence, distinguish between a masquerader and a legitimate user. Patterns of legitimate user behavior can be established by observing past history, and significant deviation from such patterns can be detected. Anderson suggests that the task of detecting a misfeasor (legitimate user performing in an unauthorized fashion) is more difficult, in that the distinction between abnormal and normal behavior may be small. Anderson concluded that such violations

would be undetectable solely through the search for anomalous behavior. However, misfeasor behavior might nevertheless be detectable by intelligent definition of the class of conditions that suggest unauthorized use. Finally, the detection of the clandestine user was felt to be beyond the scope of purely automated techniques. These observations, which were made in 1980, remain true today.

**1. Statistical anomaly detection:** involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

- **Threshold detection:** this approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.
- **Profile based:** a profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

**2. Rule-based detection**: involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

- **Anomaly detection**: rules are developed to detect deviation from previous usage patterns.
- **Penetration identification**: an expert system approach that searches for suspicious behavior.

In a nutshell, statistical approaches attempt to define normal, or expected, behavior, whereas rule-based approaches attempt to define proper behavior.

In terms of the types of attackers listed earlier, statistical anomaly detection is effective against masqueraders, who are unlikely to mimic the behavior patterns of the accounts they appropriate. On the other hand, such techniques may be unable to deal with misfeasors. For such attacks, rule-based approaches may be able to recognize events and sequences that, in context, reveal penetration. In practice, a system may exhibit a combination of both approaches to be effective against a broad range of attacks.