Malicious Programs and Protection

Unit 6 [LH 4]

Topics

- Computer viruses and worms
- Rabbit and Bacteria Defenses
  - Sandboxing
  - Information flow metrices
  - Reducing the rights
  - Malicious logic altering files
  - Proof carrying code
  - Notion of trust
- Antivirus and features

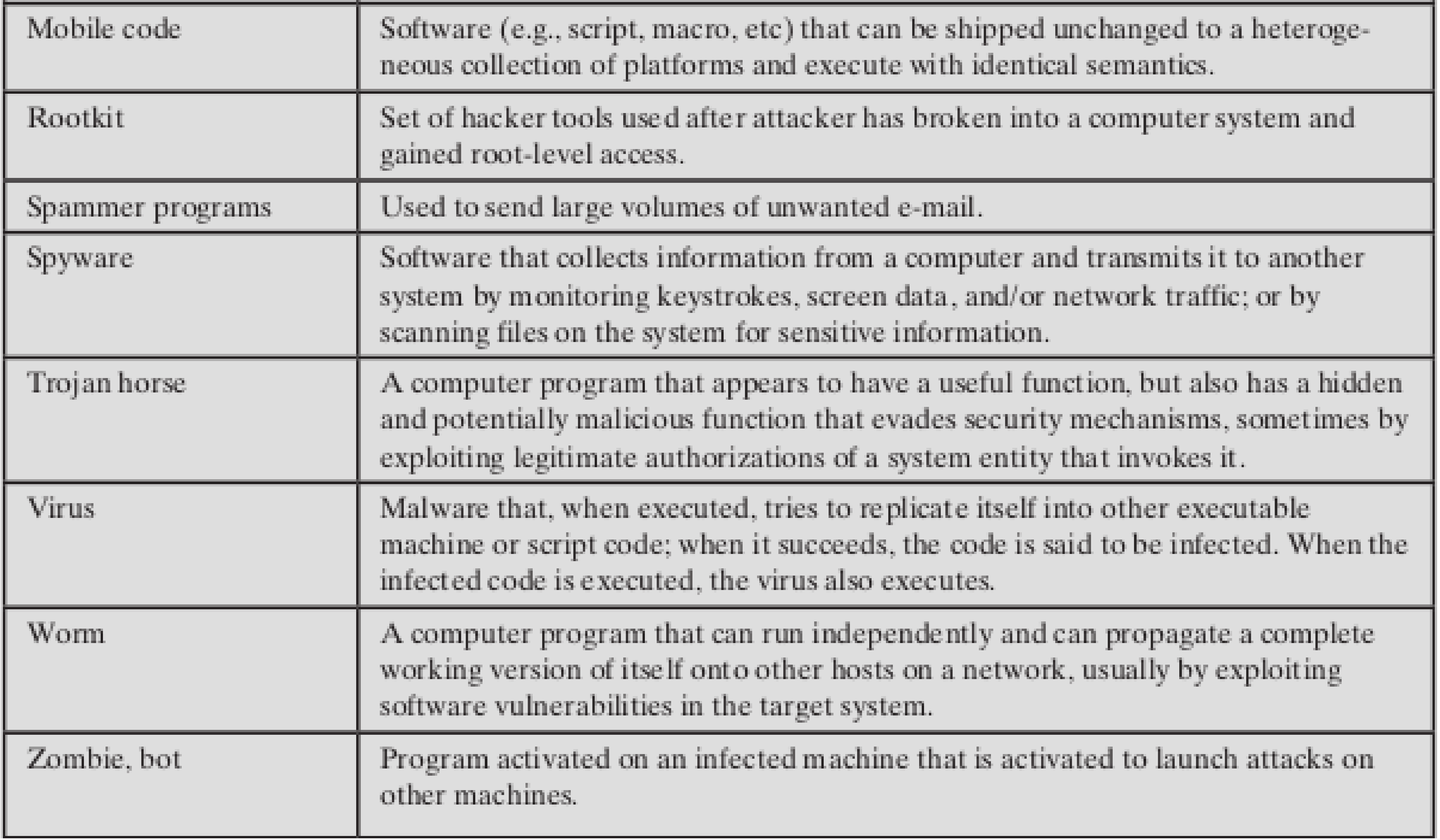
Malicious Logic

- **Malicious logic** is *a set of instructions that causes a site's security policy to be violated*.
- **Malware software**, or **malware** is defined as "*a program that is inserted into a system, usually covertly, with the intent of compromising the CIA traid of the victim's data, applications or operating system or otherwise annoying or disrupting the victim*."
- Malware can be in the form of computer viruses, worms, Trojan horses, spyware, adware and rootkits etc.
- Malware cause harm to a computer and user.

Types of Malicious Software (Malware)

- **Two broad categories**: based first on how it spreads or propagates to reach the desired targets; and then on the actions or payloads it performs once a target is reached.

Name	Description
Advanced Persistent Threat (APT)	Cybercrime directed at business and political targets, using a wide variety of intrusion technologies and malware, applied persistently and effectively to specific targets over an extended period, often attributed to state-sponsored organizations.
Adware	Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.
Attack kit	Set of tools for generating new malware automatically using a variety of supplied propagation and payload mechanisms.
Auto-rooter	Malicious hacker tools used to break into new machines remotely.
Backdoor (trapdoor)	Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality in a program, or onto a compromised system.
Downloaders	Code that installs other items on a machine that is under attack. It is normally included in the malware code first inserted on to a compromised system to then import a larger malware package.
Drive-by-download	An attack using code in a compromised Web site that exploits a browser vulnerability to attack a client system when the site is viewed.
Exploits	Code specific to a single vulnerability or set of vulnerabilities.
Flooders (DoS client)	Used to generate a large volume of data to attack networked computer systems, by carrying out some form of denial-of-service (DoS) attack.
Keyloggers	Captures keystrokes on a compromised system.
Logic bomb	Code inserted into malware by an intruder. A logic bomb lies dormant until a predefined condition is met; the code then triggers an unauthorized act.
Macro virus	A type of virus that uses macro or scripting code, typically embedded in a document, and triggered when the document is viewed or edited, to run and replicate itself into other such documents.



Mobile code	Software (e.g., script, macro, etc) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access.
Spammer programs	Used to send large volumes of unwanted e-mail.
Spyware	Software that collects information from a computer and transmits it to another system by monitoring keystrokes, screen data, and/or network traffic; or by scanning files on the system for sensitive information.
Trojan horse	A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes it.
Virus	Malware that, when executed, tries to replicate itself into other executable machine or script code; when it succeeds, the code is said to be infected. When the infected code is executed, the virus also executes.
Worm	A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network, usually by exploiting software vulnerabilities in the target system.
Zombie, bot	Program activated on an infected machine that is activated to launch attacks on other machines.

Trojan Horses

- A Trojan Horse is a program or command procedure containing a hidden code that when invoked, performs some unwanted or harmful function.
- A Trojan Horse is a program with an *overt* (documented or known) effect and a *covert* (undocumented or unexpected) effect.
- Trojan horse programs can be used to accomplish functions indirectly that the attacker could not accomplish directly.
- For example, to gain access to sensitive, personal information stored in the files of a user, an attacker could create a Trojan horse program that, when executed, scans the user's files for the desired sensitive information and sends a copy of it to the attacker via a Web form or e-mail or text message.
- The author could then entice users to run the program by incorporating it into a game or useful utility program, and making it available via a known software distribution site or app store.

Trojan Horses

- Trojan horses fit into one of three models:
  1. Continuing to perform the function of the original program and additionally performing a separate malicious activity.
  2. Continuing to perform the function of the original program but modifying the function to perform malicious activity (e.g., a Trojan horse version of a login program that collects passwords) or to disguise other malicious activity (e.g., a Trojan horse version of a process listing program that does not display certain processes that are malicious).
  3. Performing a malicious function that completely replaces the function of the original program.

Computer Virus

- A computer virus is a program that inserts itself into one or more files and then performs some (possibly null) action.
- When the Trojan horse can propagate freely and insert a copy of itself into another file, it becomes a computer 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 Virus

- 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.

Computer Virus

- Thus, the infection can spread from computer to computer, aided by unsuspecting users, who exchange these programs or carrier files on disk or USB stick; or who send them to one another over a network.
- In a network environment, the ability to access documents, applications, and system services on other computers provides a perfect culture for the spread of such viral code.
- A virus that attaches to an executable program can do anything that the program is permitted to do.
- It executes secretly when the host program is run.
- Once the virus code is executing, it can perform any function, such as erasing files and programs, that is allowed by the privileges of the current user.

Computer Virus

- Computer virus has three parts:
  i. **Infection mechanism**: The means by which a virus spreads or propagates, enabling it to replicate. The mechanism is also referred to as the *infection Vector*.
  ii. **Trigger**: The event or condition that determines when the payload is activated or delivered, sometimes known as a *logic bomb*.
  iii. **Payload**: What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity.

Computer Virus

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

i. **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.

ii. **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.

iii. **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.

iv. **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.

Computer Virus

- Most viruses that infect executable program files carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform.
- Thus, they are designed to take advantage of the details and weaknesses of particular systems.

Computer Worms

- A worm is a program that can replicate itself and send copies from computer to computer across network connections.
- A computer virus infects other programs. A variant of the virus is a program that spreads from computer to computer, spawning copies of itself on each one.
- Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function.

Computer Worm

- An e-mail virus has some of the characteristics of a worm because it propagates itself from system to system.
- However, we can still classify it as a virus because it uses a document modified to contain viral macro content and requires human action.
- 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.

Computer Worm

- 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 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.

Computer Worm

- **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.

Rabbit and Bacteria

- Some malicious logic multiplies so rapidly that resources become exhausted. This create a denial of service attack.
- A bacterium or rabbit is a program that absorbs all of some class of resource.
- A bacterium is not required to use all resources on the system; it uses some specific class of resource such as disk space.
- Bacteria do not explicitly damage any files. Their sole purpose is to replicate themselves.
- Bacteria, or rabbit programs, make copies of themselves to overwhelm a computer system's resources.
- Bacteria reproduce exponentially, eventually taking up all the processor capacity, memory, or disk space, denying the user access to those resources.

Rabbit and Bacteria

- **EXAMPLE** : Dennis Ritchie presented the following shell script as something that would quickly exhaust either disk space or inode tables on a UNIX Version 7 system:

```
while true
 do
     mkdir x
    chdir x
done
```

- He pointed out, however, that the user who caused a crash using this program would be immediately identified when the system was rebooted.

Defense

- The different types of malware share many common characteristics.
- Defenses focus on these characteristics, so the defenses apply to many different types of malware.
- That is part of the reason why security companies that market malware detection and prevention tools refer to them as “antivirus” tools.
- The other reason is that the term “virus” has captured the public’s imagination, and in marketing literature that term encompasses all types of malware.

Defense

Sandboxing

- Sandboxing is a computer security term referring to when a program is set aside from other programs in a separate environment so that if errors or security issues occur, those issues will not spread to other areas on the computer.
- Programs are enables in their own isolated area, where they can be worked on without posing any threat to other programs.
- Sandboxes can look like a regular operating environment
- Virtual machines are often used for what are referred to as run-time sandboxes.

Defense

Information Flow Metrics

- Because a user (unknowingly) executes malicious logic, that code can access and affect objects within the user’s protection domain. So, limiting the objects accessible to a given process run by the user is an obvious protection technique. This draws on the mechanisms for confining information.
- Information flow metrics approach is used to limit the distance of a virus can spread.
- ***Definition***: Define the flow distance metric *fd(x)* for some information x as follows: Initially, all information has fd(x) = 0. Whenever x is shared, fd(x) increases by 1. Whenever x is used as input to a computation, the flow distance of the output is the maximum of the flow distance of the input. • Information is accessible only while its flow distance is less than some particular value.

Defenses

Information Flow Metrics

- **Example**: Anne, Bill, and Cathy work on the same computer. The system uses the flow distance metric to limit the flow of information. Anne can access information with a flow distance less than 3, and Bill and Cathy can access information with a flow distance less than 2. Anne creates a program *dovirus* containing a computer virus. Bill executes it. Because the contents of the program have a flow distance of 0, when the virus infects Bill's file *safefile*, the flow distance of the virus is 1, and so Bill can access it. Hence, the copying succeeds. Now, if Cathy executes *safefile*, when the virus tries to spread to her files, its flow distance increases to 2. Hence, the infection is not permitted (because Cathy can only access information with a flow distance of 0 or 1).
- The limitation of this approach is disallowance for sharing.

Defenses

Reducing the rights

- The user can reduce the associated protection domain when running a suspect program.
- This follows from the principle of least privilege.

Defenses

Malicious Logic Altering Files

- Some mechanisms use *manipulation detection codes* (MDCs) to apply some function to a file to obtain a set of bits called the signature block and then protect that block.
- If, after recomputing the signature block, the result differs from the stored signature block, the file has changed, possibly as a result of malicious logic altering the file.
- This mechanism relies on selection of good cryptographic checksums.
- An assumption is that the signed file does not contain malicious logic before it is signed.

Defense

Proof-Carrying Code

- Necula proposed a technique that combines specification and integrity checking.
- His method, called *proof-carrying code (PCC)*, requires a “code consumer” (user) to specify a safety requirement.
- The “code producer” (author) generates a proof that the code meets the desired safety property and integrates that proof with the executable code.
- This produces a PCC binary.
- The binary is delivered (through the network or other means) to the consumer.
- The consumer then validates the safety proof and, if it is correct, can execute the code knowing that it honors that policy.
- The key idea is that the proof consists of elements drawn from the native code.
- If the native code is changed in a way that violates the safety policy, the proof is invalidated and will be rejected.

Defense

Notion of Trust

- The effectiveness of any security mechanism depends on its correct implementation and the base with which is it is implemented.
- If we cannot trust the security base of mechanism and its implementation, the mechanism will not be "secure".
- Thus, "secure", like "trust", is a relative notion.
- The design of any mechanism for enhancing computer security must attempt to balance the cost of the mechanism against the level of security desired and the degree of trust in the base of mechanism.
- Research dealing with malicious logic assumes that the interface, software, and/or hardware used to implement the proposed scheme will perform exactly as desired, meaning that the trust is in the underlying computing base, the implementation, and (if done) the verification.

Antivirus

- Antivirus software, or antivirus software (abbreviated to AV software), also known as anti-malware, is a *computer program used to prevent, detect, and remove malware*.
- Antivirus software was originally developed to detect and remove computer viruses, hence the name.
- However, with the proliferation of other malware, antivirus software started to protect from other computer threats.
- In particular, modern antivirus software can protect users from malicious **browser helper objects (BHOs), browser hijackers, ransomware, keyloggers, backdoors, rootkits, trojan horses, worms, malicious LSPs, dialers, fraud tools, adware, and spyware**.
- Some products also include protection from other computer threats, such as infected and **malicious URLs, spam, scam and phishing attacks, online identity (privacy), online banking attacks, social engineering techniques, advanced persistent threat (APT), and botnet DDoS attacks**.

Antivirus

- The ideal solution to the threat of viruses is prevention: Do not allow a virus to get into the system in the first place, or block the ability of a virus to modify any files containing executable code or macros.
- This goal is, in general, impossible to achieve, although prevention can reduce the number of successful viral attacks.
- The next best approach is to be able to do the following:
  - **Detection**: Once the infection has occurred, determine that it has occurred and locate the virus.
  - **Identification**: Once detection has been achieved, identify the specific virus that has
  - infected a program.
  - **Removal**: Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state.
  - Remove the virus from all infected systems so that the virus cannot spread further.

Antivirus

- If detection succeeds but either identification or removal is not possible, then the alternative is to discard the infected file and reload a clean backup version.
- Advances in virus and antivirus technology go hand in hand.
- Early viruses were relatively simple code fragments and could be identified and purged with relatively simple antivirus software packages.
- As the virus arms race has evolved, both viruses and, necessarily, antivirus software have grown more complex and Sophisticated.
- There are four generations of antivirus software:

1. **First Generation**: simple scanners (record of program lengths)
2. **Second Generation**: heuristic scanners (integrity checking with checksums)
3. **Third Generation**: activity traps (memory resident, detect infected actions)
4. **Fourth Generation**: full-featured protection (suite of antivirus techniques, access control capability)

Features of Antivirus

- Features that make a good antivirus program are mentioned below:
  - Malware Detection and Removal
  - Firewall
  - Auto SandBoxing Technique
  - Virus Scan
  - Identify protection
  - Backup
  - Email Protection
  - Social Media Protection

Features of Antivirus

Malware Detection & Removal:

- Any good antivirus program will be able to detect different types of viruses and malware in compressed or uncompressed form.
- This should be the first and foremost deciding factor for your anti-virus program.
- Besides that, see that your robust antivirus program doesn’t consume a lot of the system resources.
- If these qualities match, then the program is a real winner.

Features of Antivirus

Firewall:

- Few of the free antivirus programs like Comodo free antivirus, after removing all the threats from a device ensures the continual safety with the help of a powerful firewall.
- The powerful firewall helps in keeping away all the incoming threats

Features of Antivirus

Auto SandBoxing Technique:

- An effective anti-virus program provides a secure environment to run files in real time to check for foul play.
- The Comodo free antivirus is a perfect example of one such platform that provides virtual space to run and analyze untrusted, unknown and malicious applications.
- Sometimes, this feature is also known by the names '*virus cleanup*' mode, or '*virtual sandbox*'. The best part of the feature is that it provides safe removal of the virus from the computer

Features of Antivirus

Virus Scan:

- A good anti-virus program automatically runs virus scan at regular intervals to make sure the system is safe from all dangers.
- The virus scan will help spot on all the new threats that sneaked into your computer bypassing the usual authorizations.

Features of Antivirus

Identity protection:

- Identity theft has become common these days.
- With more and more paper facts getting converted into digital forms, it has paved the easy access for online thieves to steal the identity of known and unknown personnel. The online criminals make use of this information for their personal gains.
- A good antivirus program will safeguard your personal information in the best possible ways.
- For instance, it will make sure your system is safe and secure by verifying every time the user inputs the credit card or banking information online

Features of Antivirus

__Backup__:

- During bad times, especially when your computer is under attack, it is better to run the backup without taking any chances.
- Any good antivirus program will sport this feature by default and will help you in restoring that backup when all the issues have settled down

Features of Antivirus

Email Protection:

- Emails also carry viruses and malware in the attachments.
- Even though you might be extra cautious, an impersonator email can cause the damage to your computer.
- Sometimes, all it requires to do the harm is opening without clicking on anything.
- In such worse case scenarios, a good antivirus program protects your computer from being the victim of scans or phishing schemes.

Features of Antivirus

Social Media Protection

- More and more people use social media accounts rigorously on a daily basis. Multiple number of times they open and close their accounts on a single day.
- These actions of a user provide an advantageous platform for hackers to implant viruses and malware on the computers.
- The intentions of hackers may vary but they never do any good to the computer user.
- A good antivirus software will send alerts to the user when a Facebook phishing scam or a Twitter malicious link has been detected