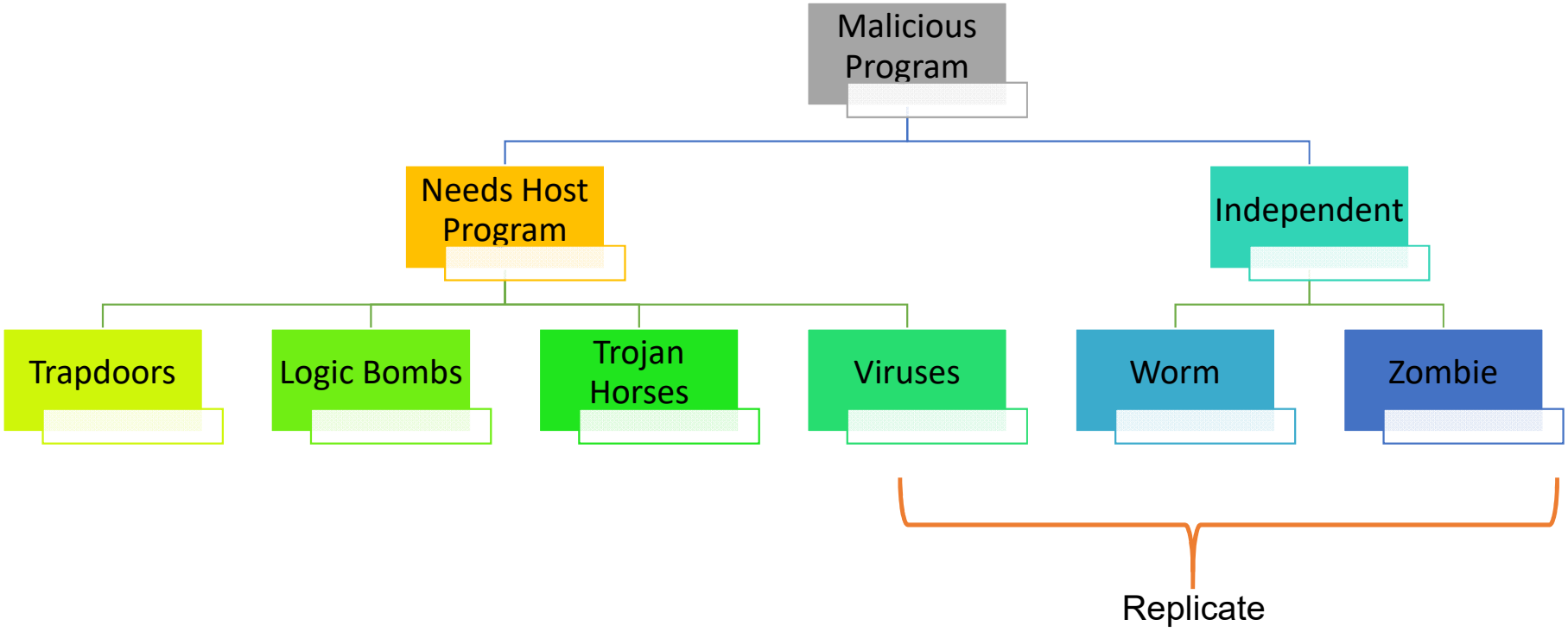
Network Security and Public Key Infrastructure

Unit -6 [6Hours]

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.
-

Malicious Software



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.

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

- 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

- 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

Computer virus has three parts:

1. **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*.
2. **Trigger**: The event or condition that determines when the payload is activated or delivered, sometimes known as a *logic bomb*.
3. **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:

1. **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.
2. **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.
3. **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.
4. **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 Virus

```
program V
1234567;

procedure attach-to-program;
begin
     repeat
          file := get-random-program;
     until first-program-line ≠ 1234567;
     prepend V to file;
end;

procedure execute-payload;
begin
     (* perform payload actions *)
end;

procedure trigger-condition;
begin
     (* return true if trigger condition is true *)
end;

begin (* main action block *)
     attach-to-program;
     if trigger-condition then execute-payload;
     goto original program code;
end;
```

(a) A simple virus

```
program CV
1234567;

procedure attach-to-program;
begin
       repeat
            file := get-random-program;
       until first-program-line ≠ 1234567;
       compress file;     (* t1 *)
       prepend CV to file;    (* t2 *)
end;

begin (* main action block *)
       attach-to-program;
       uncompress rest of this file into tempfile;    (* t3 *)
       execute tempfile;    (* t4 *)
end;
```

(b) A compression virus

Figure: Example Virus Logic

Computer Worm

- 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 Work

- 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
  - Remote Execution capability
  - Remote Login Capability

Rabbit and Bacteria

- ome 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.

Zombies

- A zombie is a computer connected to the Internet that has been compromised by a hacker, computer virus or trojan horse program and can be used to perform malicious tasks of one sort or another under remote direction.
- Botnets of zombie computers are often used to spread e-mail spam and launch denial-of-service attacks (DoS attacks).
- 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.
- A bot, short for "robot", is a type of software application or script that performs automated tasks on command. Bad bots perform malicious tasks that allow an attacker to remotely take control over an affected computer. Once infected, these machines may also be referred to as zombies.
- In addition to the worm-like ability to self-propagate, bots can include the ability to log keystrokes, gather passwords, capture and analyze packets, gather financial information, launch Denial of Service (DOS) Attacks, relay spam, and open backdoors on the infected host.

Denial of Service Attacks

- A **denial-of-service (DoS)** attack is an attempt to compromise availability by hindering or blocking completely the provision of some service.
- The attack attempts to exhaust some critical resource associated with the service.
- An example is flooding a Web server with so many spurious requests that it is unable to respond to valid requests from users in a timely manner.

DoS Attack: NIST Definition

- "A **denial of service (DoS)** is an action that prevents or impairs the authorized use of networks, systems, or applications by exhausting resources such as central processing units (CPU), memory, bandwidth, and disk space."
- From this definition, you can see that there are several categories of resources that could be attacked:
  - Network bandwidth
  - System resources
  - Application resources

How can DoS Attacks be made?

- In a DoS attack, the vast majority of traffic directed at the target server (e.g. ISP server) is malicious, generated either directly or indirectly by the attacker. (Flooding Attack). This traffic overwhelms any legitimate traffic, effectively denying legitimate users access to the server. (***cyberslam)***.
- A DoS attack targeting system resources typically aims to overload or crash its network handling software. e.g. ***SYN spoofing*** → It targets the table of TCP connections on the server.
- Another form of system resource attack uses packets whose structure triggers a bug in the system's network handling software, causing it to crash. This means the system can no longer communicate over the network until this software is reloaded, generally by rebooting the target system. This is known as a ***poison packet***.
- Construct a request that triggers a bug in the server program, causing it to crash. This means the server is no longer able to respond to requests until it is restarted.

Distributed Denial-of-Service Attack

- DDoS attacks make computer systems inaccessible by flooding servers,networks, or even end user systems with useless traffic so that legitimate users can no longer gain access to those resources. In a typical DDoS attack, a large number of compromised hosts are amassed to send useless packets.
- In a DDoS attack, an attacker is able to recruit a number of hosts throughout the Internet to simultaneously or in a coordinated fashion launch an attack upon the target.

Distributed Denial-of-Service Attack

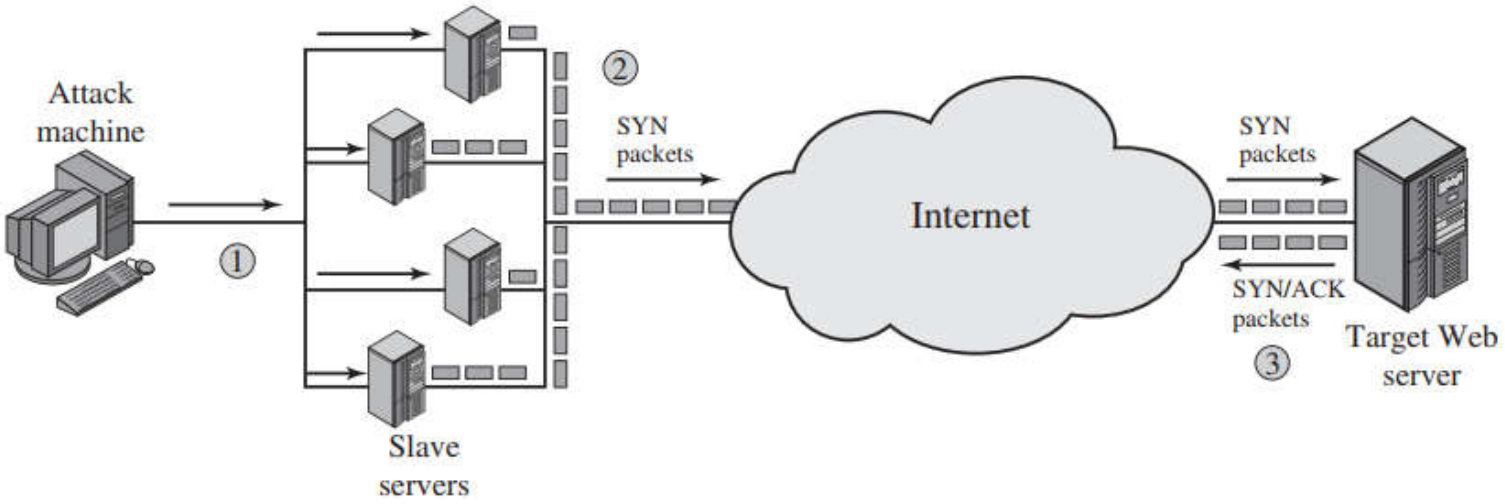


Figure: Example DDoS Attack : Distributed SYN flood attack

1. The attacker takes control of multiple hosts over the Internet, instructing them to contact the target Web server.
2. The slave hosts begin sending TCP/IP SYN (synchronize/initialization) packets, with erroneous return IP address information, to the target.
3. Each SYN packet is a request to open a TCP connection. For each such packet, the Web server responds with a SYN/ACK synchronize/acknowledge) packet, trying to establish a TCP connection with a TCP entity at a spurious IP address. The Web server maintains a data structure for each SYN request waiting for a response back and becomes bogged down as more traffic floods in. The result is that legitimate connections are denied while the victim machine is waiting to complete bogus “half-open” connections.

DDoS Attack

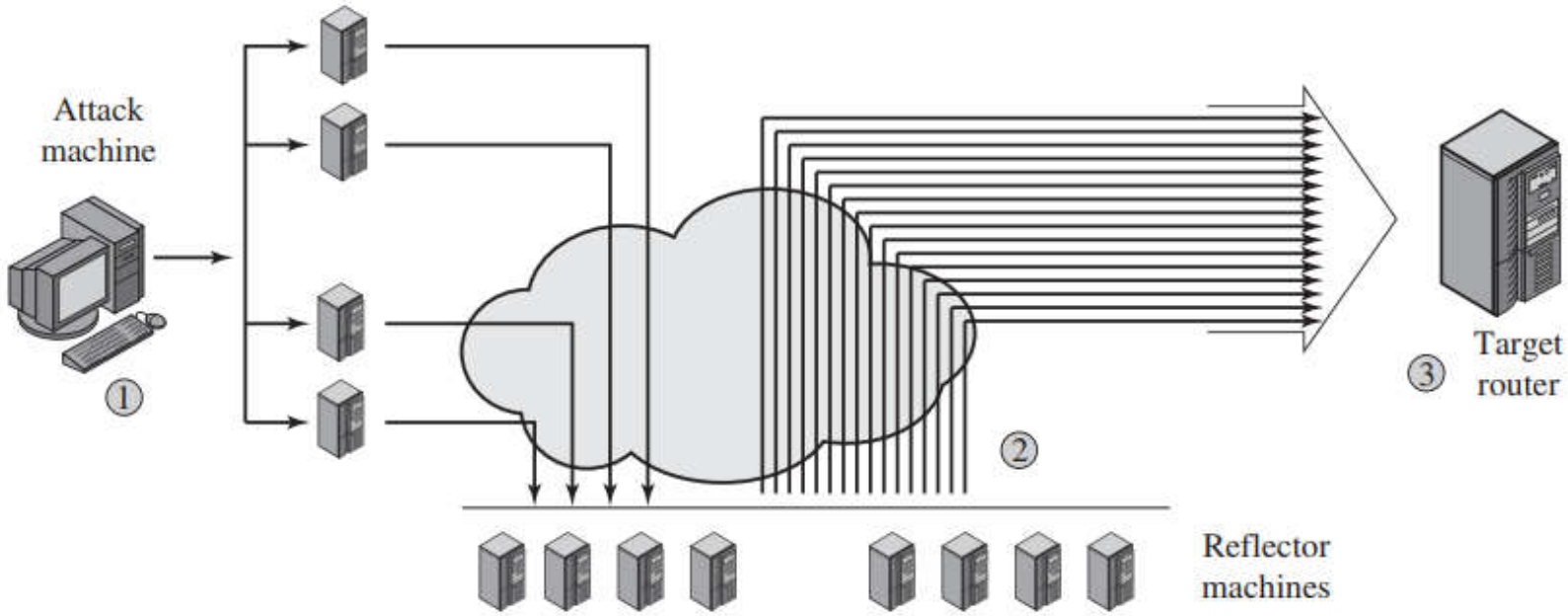
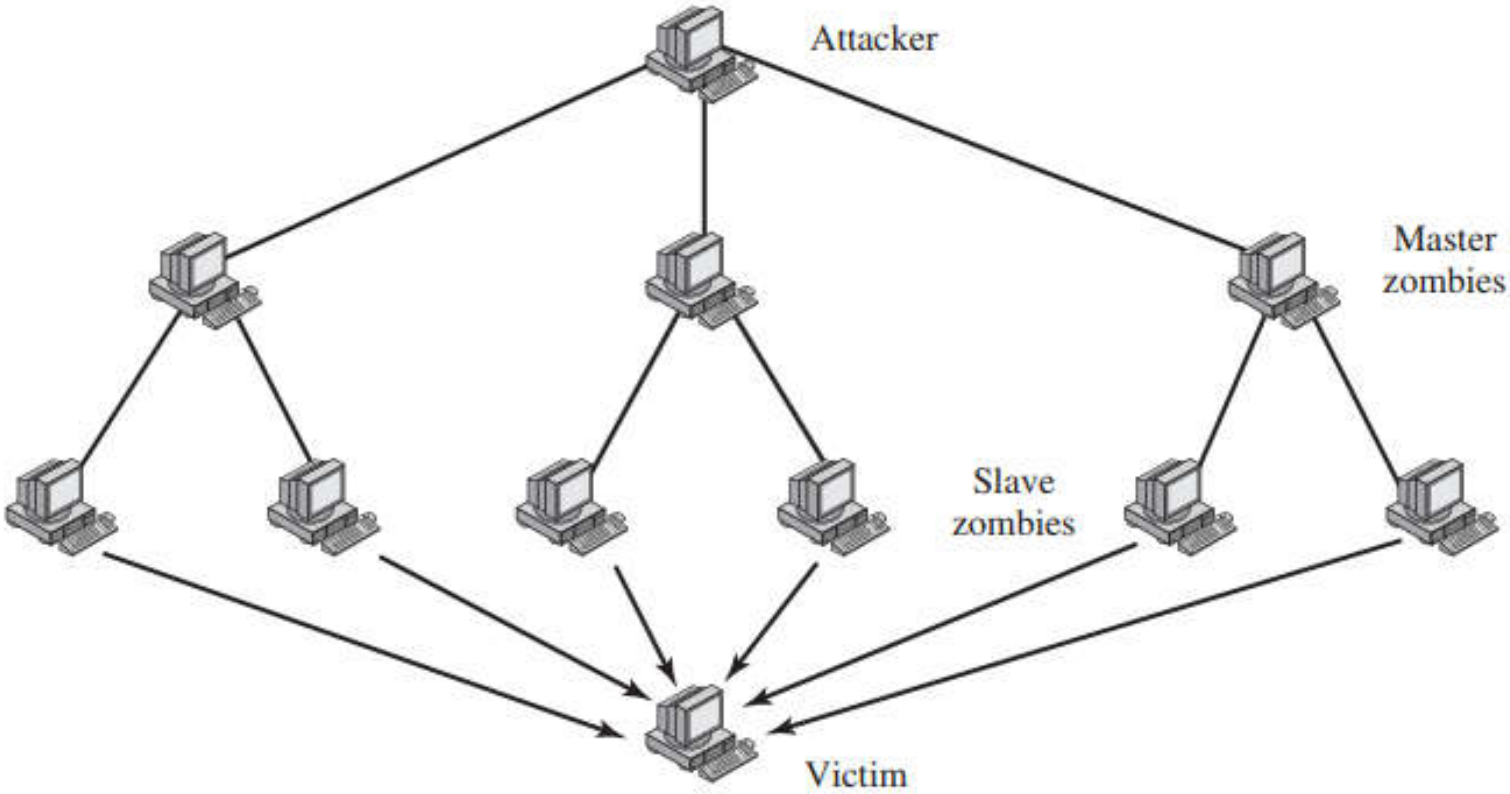
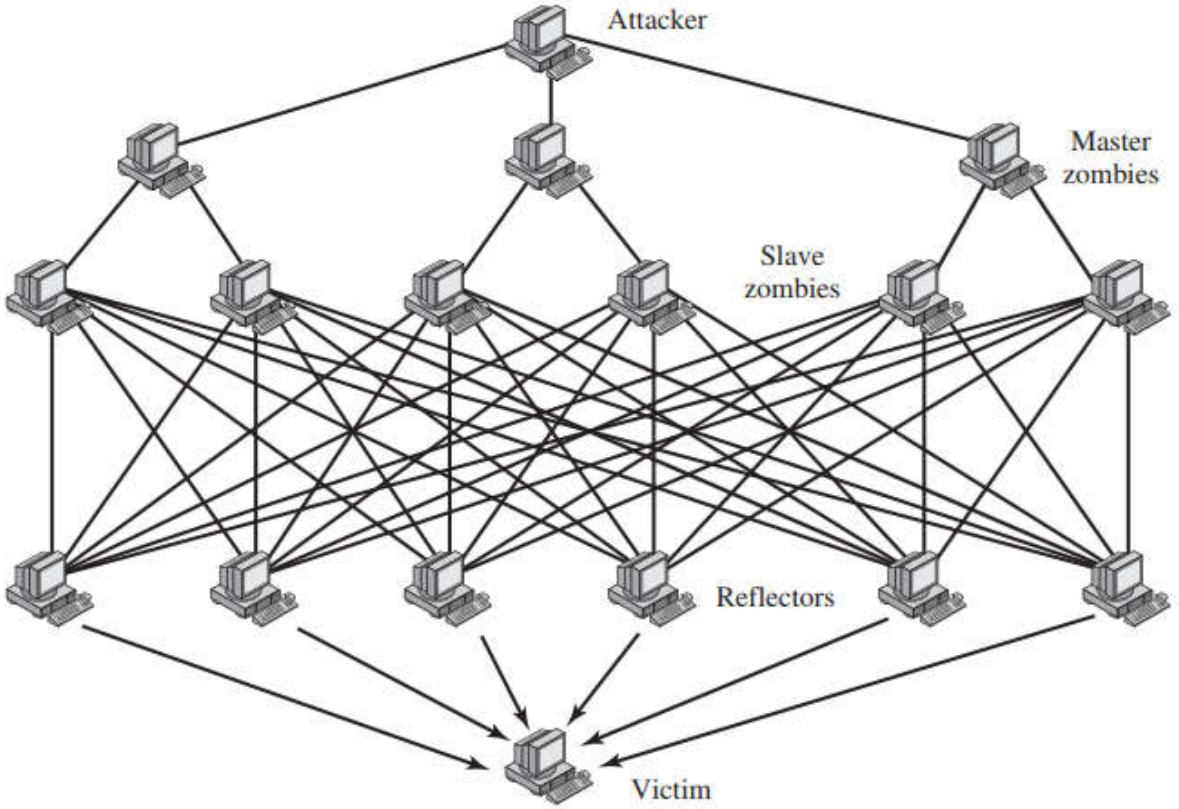


Figure: Example DDoS Attack : Distributed ICMP Attack

Direct DDoS Attack



Reflector DDoS Attack



Intrusion Detection

Background

- A significant security problem for networked systems is hostile, or at least unwanted, trespass by users or software.
- User trespass can take the form of unauthorized login to a machine or, in the case of an authorized user, acquisition of privileges or performance of actions beyond those that have been authorized.
- Software trespass can take the form of a virus, worm, or Trojan horse.

- One of the key threats to security is the use of some form of hacking by an ***intruder***, often referred to as a ***hacker*** or ***cracker, or interceptor.***
- **Intrusion** is a phenomenon that performs an activity that compromises a computer system by breaking the security or causing it to enter into an insecure state by an intruder.
- A set of attempts to compromise a computer or a computer network resource security is regarded as an intrusion.

Types of Intruders

- Anderson identified three classes of intruders:

1. **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. The masquerader is likely to be an outsider.
2. **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 . The misfeasor generally is an insider.
3. **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 clandestine user can be either an outsider or an insider.

Intruder Attacks

- Intruder attacks range from the benign to the serious.
- The following are some examples of intrusions:
  - Performing a remote root compromise of an e-mail server.
  - Defacing a Web server.
  - Guessing and cracking passwords.
  - Copying a database containing credit card numbers.
  - Viewing sensitive data, including payroll records and medical information, without authorization.
  - Running a packet sniffer on a workstation to capture usernames and passwords.
  - Using a permission error on an anonymous FTP server to distribute pirated software and music files.
  - Dialing into an unsecured modem and gaining internal network access.
  - Posing as an executive, calling the help desk, resetting the executive's e-mail password, and learning the new password.
  - Using an unattended, logged-in workstation without permission.

Intrusion Detection

- **Security Intrusion**: A security event, or a combination of multiple security events, that constitutes a security incident in which an intruder gains, or attempts to gain, access to a system (or system resource) without having authorization to do so.
- **Intrusion Detection**: A security service that monitors and analyzes system events for the purpose of finding, and providing real-time or near real-time warning of, attempts to access system resources in an unauthorized manner.
- 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

Goals of Instruction Detection

- **Detect a wide variety of intrusions**.

  - Intrusions from within the site, as well as those from outside the site, are of interest.

- **Detect intrusions in a timely fashion**.

  - "Timely" here need not be in real time. Often, it suffices to discover an intrusion within a short period of time.

- **Present the analysis in a simple, easy-to-understand format.**

- **Be accurate**.

  A **false positive** occurs when an intrusion detection system reports an attack, but no attack is underway. False positives reduce confidence in the correctness of the results as well as increase the amount of work involved. However, **false negatives** (occurring when an intrusion detection system fails to report an ongoing attack) are worse, because the purpose of an intrusion detection system is to report attacks

Classification of IDS

- Host Intrusion Detection System (HIDS)
- Network Intrusion Detection System (NIDS)
- Distributed or hybrid Intrusion Detection System

Classification of IDS

- **Host-based IDS (HIDS)**: Monitors the characteristics of a single host and the events occurring within that host, such as process identifiers and the system calls they make, for evidence of suspicious activity.

- **Network-based IDS (NIDS)**: Monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity.

- **Distributed or hybrid IDS**: Combines information from a number of sensors, often both host and network-based, in a central analyzer that is able to better identify and respond to intrusion activity.

Approaches to Anomaly Detection

- There are two general approaches to intrusion detection:
  I. Statistical Anomaly Detection, and
  II. Rule (Knowledge/Missuse) Based Anomaly detection
  III. Machine Learning Anomaly Detection

Statistical Anomaly Detection

- Statistical approaches use the captured sensor data to develop a statistical profile of the observed metrics.
- Analysis of the observed behavior using univariate, multivariate, or time-series models of observed metrics.
- simple and low computation cost
- difficulty in selecting suitable metrics

Rule Based (Misuse) Anomaly Detection

- classify the observed data using a set of rules.
- ***determines whether a sequence of instructions being executed is known to violate the site security policy being executed***. If so, it reports a potential intrusion.
- Modeling of misuse requires a knowledge of system vulnerabilities or potential vulnerabilities that attackers attempt to exploit. The intrusion detection system incorporates this knowledge into a rule set.
- When data is passed to the intrusion detection system, it applies the rule set to the data to determine if any sequences of data match any of the rules. If so, it reports that a possible intrusion is underway.
- robust and flexible.

Rule Based (Misuse) Anomaly Detection

- Misuse-based intrusion detection systems often use expert systems to analyze the data and apply the rule set. (difficult and time requiring)
- **These systems cannot detect attacks that are unknown to the developers of the rule set**.
- Previously unknown attacks, or even variations of known attacks, can be difficult to detect.
- Later intrusion detection systems used adaptive methods involving neural networks and Petri nets to improve their detection abilities.

Machine Learning Anomaly Detection

- use data mining techniques to automatically develop a model using the labeled normal training data.
- then classify subsequent observed data as either normal or anomalous.
- requires significant time and computational resources.
- but once model is generated, subsequent analysis is generally fairly efficient.