SCS1316 - NETWORK SECURITY

UNIT- V

Malicious Software - Types – Backdoor – Worms - Logic bomb - Trojan Horses – Viruses - Classifications- Virus Kits - Email Viruses-Antivirus Approach-Distributed Denial of Service Attacks-Counter Measures-Intrusion Detection System (IDS),Network Based IDS-Host based IDS- Steps involved in deploying IDS

Malicious Software

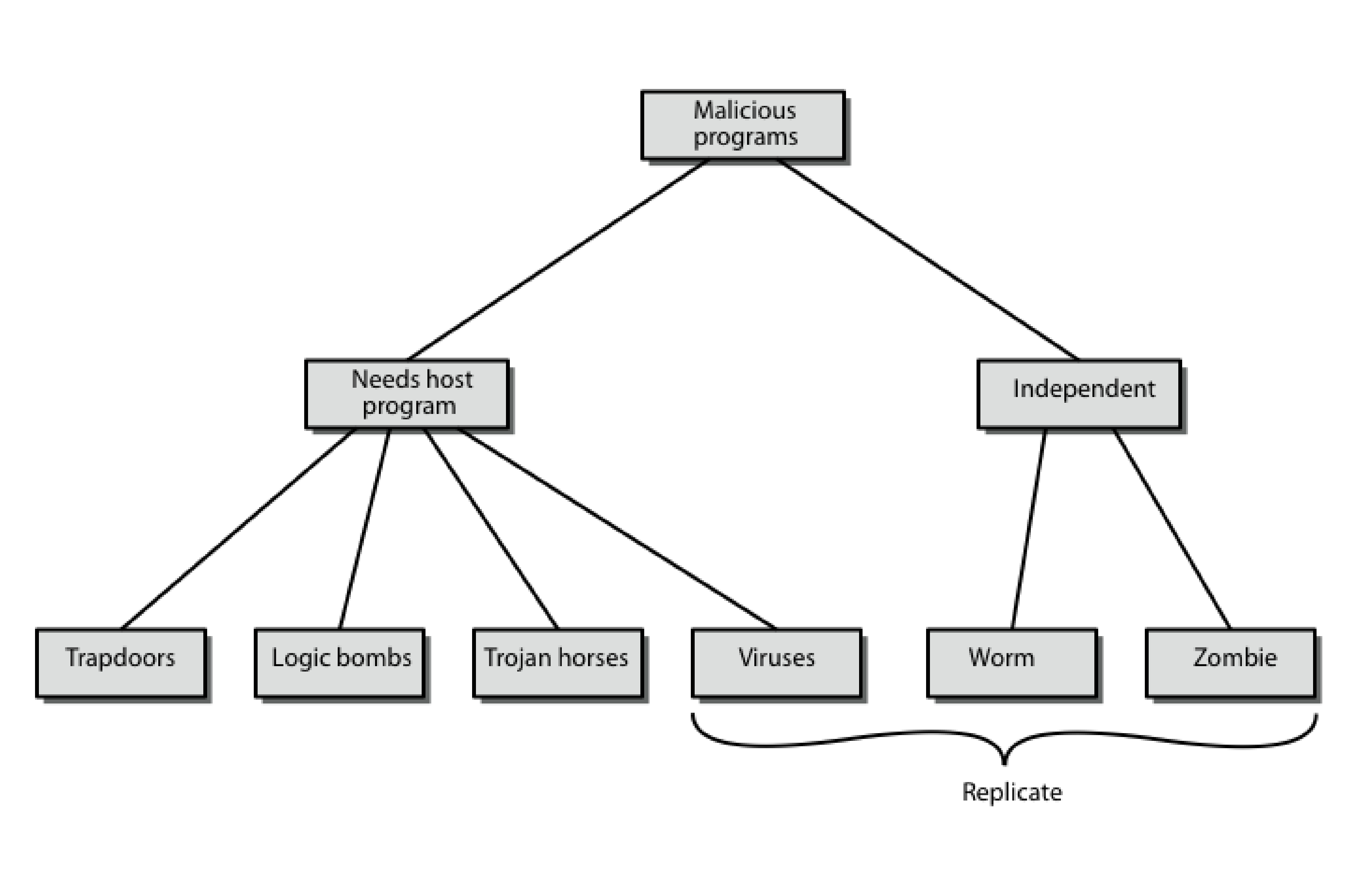
- The words “Malicious Software” coin the word “Malware” and the meaning remains the same
- Malicious Software refers to any malicious program that causes harm to a computer system or network.
- Malicious Software attacks a computer or network in the form of viruses, worms, trojans, spyware, adware or rootkits
- Their mission is often targeted at accomplishing unlawful tasks such as robbing protected data, deleting confidential documents or add software without the user consent.

Different Types of Malicious Software

Computer Virus

- A computer virus is a malicious software which self-replicates and attaches itself to other files/programs.
- It is capable of executing secretly when the host program/file is activated.
- The different types of Computer virus are Memory-Resident Virus, Program File Virus, Boot Sector Virus, Stealth Virus, Macro Virus, and Email Virus.

Malicious Software - Types



Backdoor or Trapdoor

- secret entry point into a program
- allows those who know access - bypassing usual security procedures
- have been commonly used by developers
- a threat when left in production programs allowing exploited by attackers
- very hard to block in O/S
- requires good s/w development & update

Logic Bomb

- one of oldest types of malicious software
- code embedded in legitimate program
- activated when specified conditions met
  - eg presence/absence of some file
  - particular date/time
  - particular user
- when triggered typically damage system
  - modify/delete files/disks, halt machine, etc

Trojan Horse

- program with hidden side-effects
- which is usually superficially attractive
  - eg game, s/w upgrade etc
- when run performs some additional tasks
  - allows attacker to indirectly gain access they do not have directly
- often used to propagate a virus/worm or install a backdoor
- or simply to destroy data

Zombie

- program which secretly takes over another networked computer
- then uses it to indirectly launch attacks
- often used to launch distributed denial of service (DDoS) attacks
- exploits known flaws in network systems

Worms

- A worm is a malicious software which is like that of a computer virus is a self-replicating program, however, in the case of worms, it automatically executes itself.
- Worms spread over a network and are capable of launching a cumbersome and destructive attack within a short period.

Logic bomb

- A logic bomb is a malicious piece of code that’s secretly inserted into a computer network, operating system, or software application.
- It lies dormant until a specific condition occurs.
- When this condition is met, the logic bomb is triggered — devastating a system by corrupting data, deleting files, or clearing hard drives.

How does a logic bomb work?

- The conditions that trigger a logic bomb can be categorized as **positive** or **negative**.
- Logic bombs with positive triggers detonate after a condition is met, such as when you open a particular file.
- Negative triggers launch a logic bomb when a condition is not met, such as when the bomb isn’t deactivated in time.
- Either way, when the desired conditions are achieved, the program’s system of logic will order the logic bomb to go off and inflict its damage.

Viruses

- a piece of self-replicating code attached to some other code
  - like biological virus
- both propagates itself & carries a payload
  - carries code to make copies of itself
  - as well as code to perform some covert task

Virus Operation

- virus phases:
  - dormant – waiting on trigger event
  - propagation – replicating to programs/disks
  - triggering – by event to execute payload
  - execution – of payload
- details usually machine/OS specific
  - exploiting features/weaknesses

Virus Structure

```
program V :=
   {goto main;
   1234567;
   subroutine infect-executable :=    {loop:
        file := get-random-executable-file;
        if (first-line-of-file = 1234567) then goto loop
        else prepend V to file; }
   subroutine do-damage :=  {whatever damage is to be done}
   subroutine trigger-pulled := {return true if condition holds}
   main: main-program :=    {infect-executable;
                if trigger-pulled then do-damage;
                goto next;}
   next:
     }
```

Virus Structure

- The virus code (V) is prepended to infected programs (assuming the entry point is the first line of the program).
- The first line of code jumps to the main virus program.
- The second line is a special marker for infected programs.
- The main virus program first seeks out uninfected executable files and infects them.
- Then it may perform some action, usually detrimental to the system, depending on some trigger.
- Finally, the virus transfers control to the original program.
- If the infection phase of the program is reasonably rapid, a user is unlikely to notice any difference between the execution of an infected and uninfected program.
- This type of virus can be detected because the length of the program changes.
- More sophisticated variants attempt to hide their presence better, by for example, compressing the original program.

Types of Viruses

On basis of how they attack

- parasitic virus
- memory-resident virus
- boot sector virus
- stealth
- polymorphic virus
- metamorphic virus

Types of Viruses

- **Parasitic virus:** traditional and still most common form of virus, it attaches itself to executable files and replicates when the infected program is executed
- **Memory-resident virus:** Lodges in main memory as part of a resident system program, and infects every program that executes
- **Boot sector virus:** Infects a master boot record and spreads when a system is booted from the disk containing the virus
- **Stealth virus:** a virus explicitly designed to hide itself from detection by antivirus software
- **Polymorphic virus:** mutates with every infection, making detection by the “signature”of the virus impossible.
- **Metamorphic virus:** mutates with every infection, rewriting itself completely at each iteration changing behavior and/or appearance, increasing the difficulty of detection.

Macro Virus

- **macro code** attached to some **data file**
- interpreted by program using file
  - eg Word/Excel macros
  - esp. using auto command & command macros
- code is now platform independent
- is a major source of new viral infections
- blur distinction between data and program files
- classic trade-off: "ease of use" vs "security”
- have improving security in Word etc
- are no longer dominant virus threat

Email Virus

- spread using email with attachment containing a macro virus
- triggered when user opens attachment
- or worse even when mail viewed by using scripting features in mail agent
- hence propagate very quickly
- usually targeted at Microsoft Outlook mail agent & Word/Excel documents
- need better O/S & application security

Virus Countermeasures

- best countermeasure is prevention
- but in general - not possible
- hence need to do one or more of:
  - **detection** - of viruses in infected system
  - **identification** - of specific infecting virus
  - **removal** - restoring system to clean state

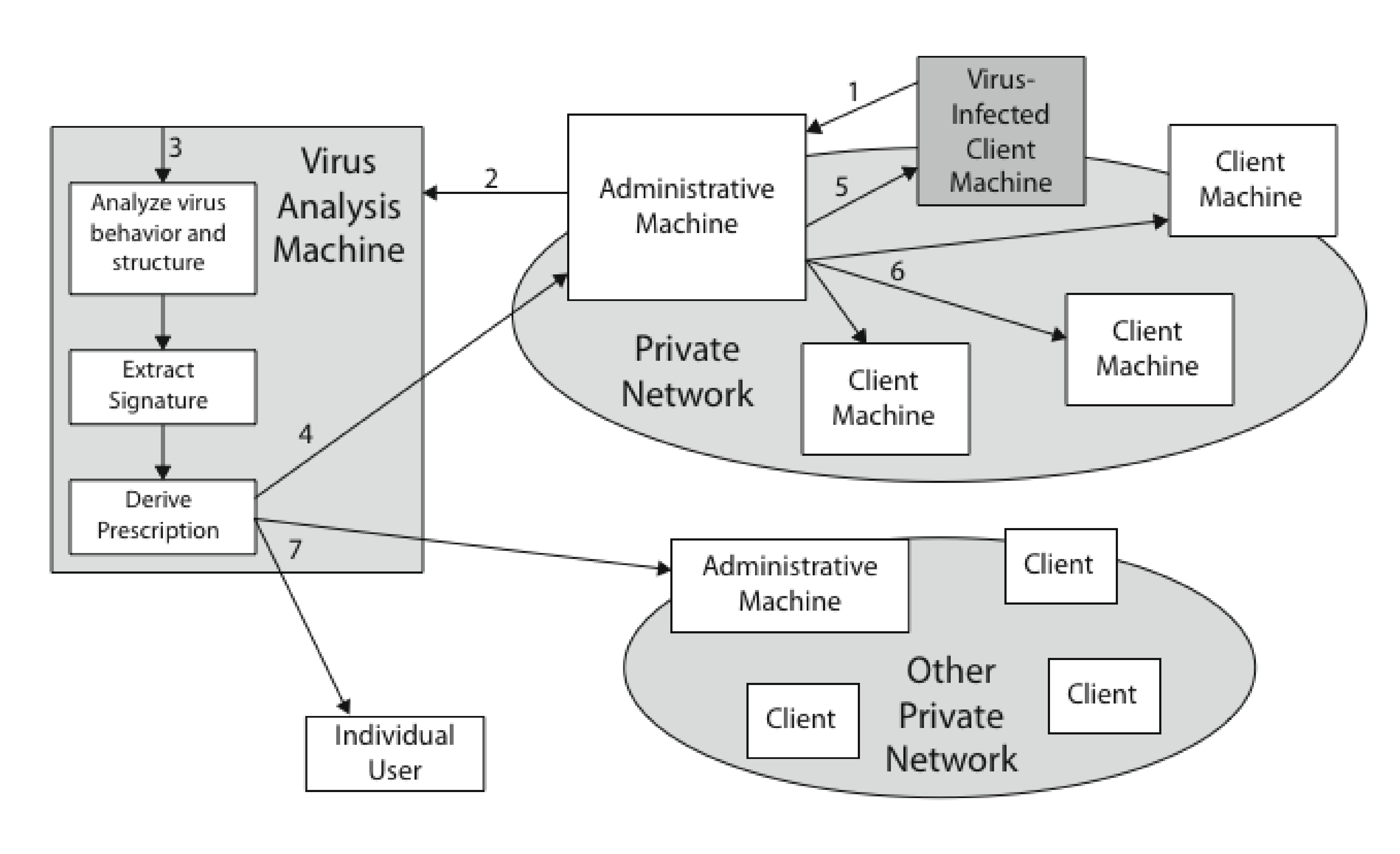
Anti-Virus Software

- **first-generation**
  - scanner uses virus signature to identify virus
  - or change in length of programs
- **second-generation**
  - uses heuristic rules to spot viral infection
  - or uses crypto hash of program to spot changes
- **third-generation**
  - memory-resident programs identify virus by actions
- **fourth-generation**
  - packages with a variety of antivirus techniques
  - eg scanning & activity traps, access-controls

Advanced Anti-Virus Techniques

- generic decryption
  - use CPU simulator to check program signature & behavior before actually running it
- digital immune system (IBM)
  - general purpose emulation & virus detection
  - any virus entering org is captured, analyzed, detection/shielding created for it, removed

Digital Immune System



Digital Immune System

1. A monitoring program on each PC uses a variety of heuristics based on system behavior, suspicious changes to programs, or family signature to infer that a virus may be present, & forwards infected programs to an administrative machine
2. The administrative machine encrypts the sample and sends it to a central virus analysis machine
3. This machine creates an environment in which the infected program can be safely run for analysis to produces a prescription for identifying and removing the virus
4. The resulting prescription is sent back to the administrative machine
5. The administrative machine forwards the prescription to the infected client
6. The prescription is also forwarded to other clients in the organization
7. Subscribers around the world receive regular antivirus updates that protect them from the new virus.

Behavior-Blocking Software

- integrated with host O/S
- monitors program behavior in real-time
  - eg file access, disk format, executable mods, system settings changes, network access
- for possibly malicious actions
  - if detected can block, terminate, or seek ok
- has advantage over scanners
- but malicious code runs before detection

Distributed Denial of Service Attacks (DDoS)

- A denial of service (DoS) attack is an attempt to prevent legitimate users of a service from using that service.
- When this attack comes from a single host or network node, then it is simply referred to as a DoS attack.
- A more serious threat is posed by a DDoS attack.
- 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 Attacks (DDoS)

DDoS attacks form a significant security threat

- making networked systems unavailable
- by flooding with useless traffic
- using large numbers of “zombies”
- growing sophistication of attacks
- defense technologies struggling to cope

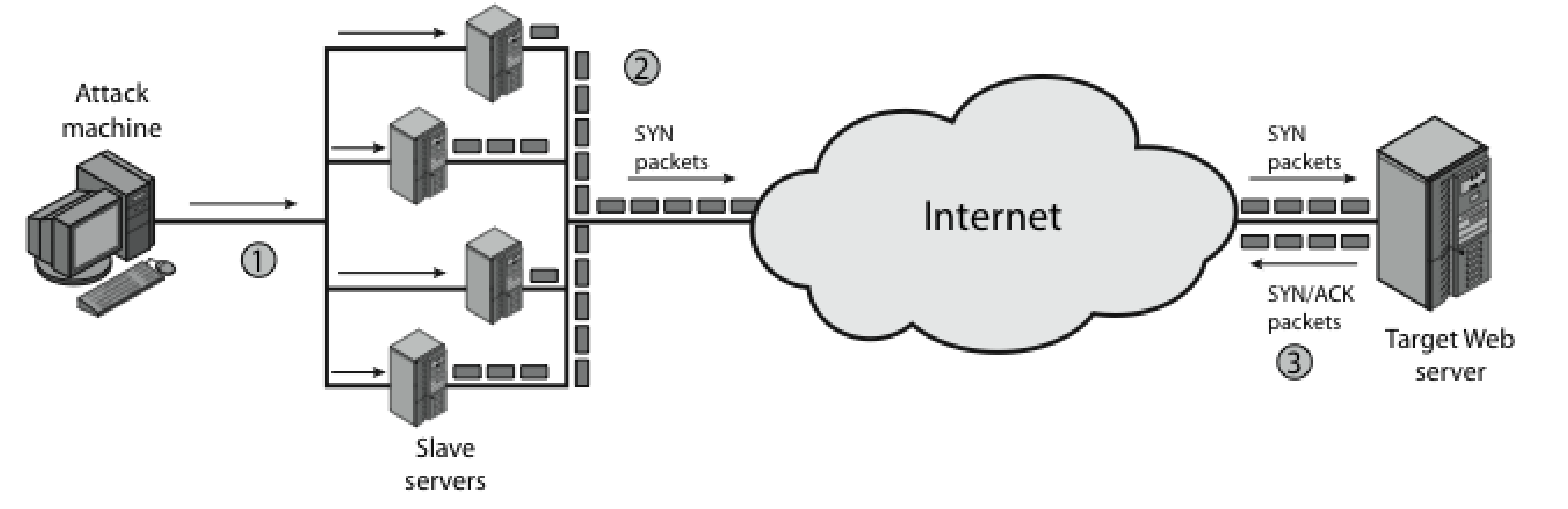
Distributed Denial of Service Attacks (DDoS)

- A DDoS attack **attempts to consume the target's resources** so that it cannot provide service.
- One way to classify DDoS attacks is in terms of the type of resource that is consumed, either an internal host resource on the target system, or data transmission capacity in the target local network.

Figure shows an example of an **internal resource attack** - the **SYN flood attack**.

1. The attacker takes control of multiple hosts over the Internet
2. The slave hosts begin sending TCP/IP SYN (synchronize/initialization) packets, with erroneous return IP address information, to the target
3. For each such packet, the Web server responds with a SYN/ACK (synchronize/acknowledge) packet.
4. 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.

Distributed Denial of Service Attacks (DDoS)



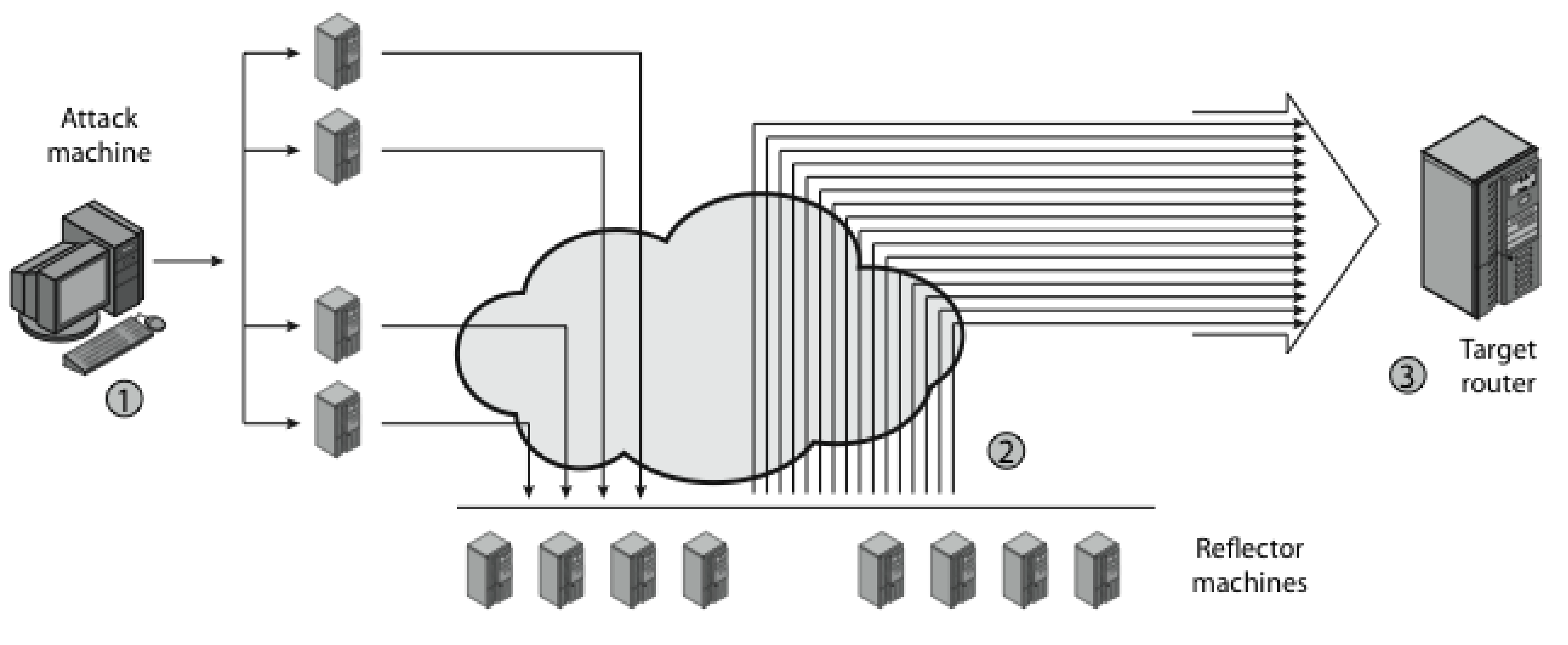
(a) Distributed SYN flood attack

Distributed Denial of Service Attacks (DDoS)

Figure illustrates an example of an attack that consumes **data transmission resources**.

1. The attacker takes control of multiple hosts over the Internet, instructing them to send ICMP ECHO packets with the target's spoofed IP address to a group of hosts that act as reflectors
2. Nodes at the bounce site receive multiple spoofed requests and respond by sending echo reply packets to the target site.
3. The target's router is flooded with packets from the bounce site, leaving no data transmission capacity for legitimate traffic.

Distributed Denial of Service Attacks (DDoS)



(a) Distributed ICMP attack

Contructing the DDoS Attack Network

- The first step in a DDoS attack is for the attacker to infect large number of machines with zombies
- The essential ingredients are:

1. Software that can carry out the DDoS attack, runnable on a large number of machines, concealed, communicating with attacker or time-triggered, and can launch intended attack toward the target
2. A vulnerability in a large number of systems, that many system admins/users have failed to patch
3. A strategy for locating vulnerable machines, known as scanning, such as:
   - Random: probe random IP addresses in the IP address space
   - Hit-list: use a long list of potential vulnerable machines
   - Topological: use info on infected victim machine to find more hosts
   - Local subnet: look for targets in own local network

DDoS Countermeasures

- huge range of attack possibilities hence evolving countermeasures
- three broad lines of defense against DDoS attacks:

1. attack prevention & preemption (before)
2. attack detection & filtering (during)
3. attack source traceback & identifying (after)

Intruders

- One of the two most publicized threats to security is the intruder (the other is viruses), often referred to as a hacker or cracker.
- In an important early study of intrusion, it had been identified three classes of intruders:
- **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

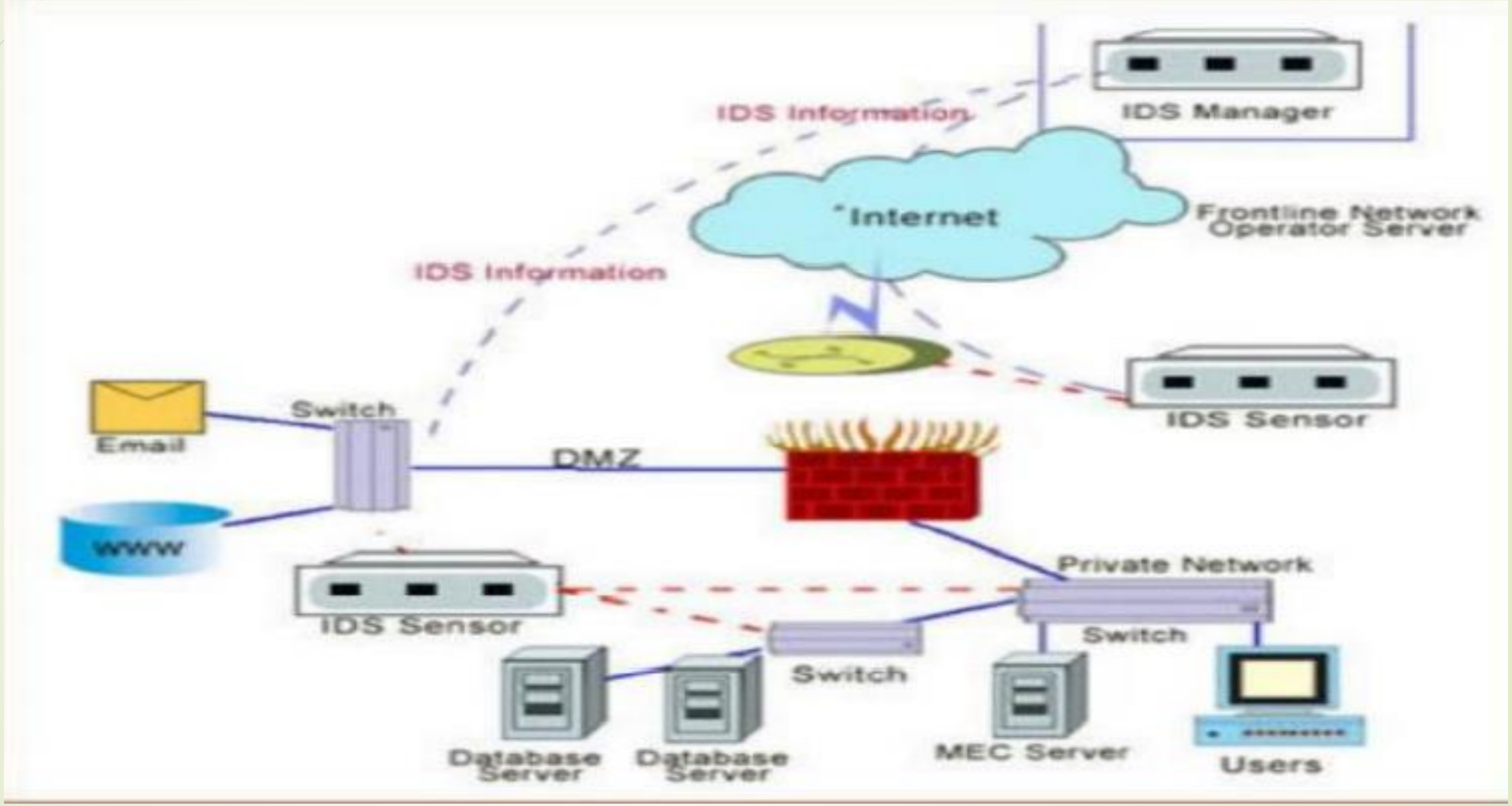
Intruders

- 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.
- Some of the examples for intrusion are,
  - 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 password

Intrusion Detection Systems (IDS)

- Intrusion detection systems are designed to analyse network traffic for potentially malicious behaviour and to report possible “intrusions” to a centralized management node.
- Some IDSs are designed to take action to prevent these attempts from being successful; however, stopping malicious attacks is not a required component of an IDS.
- Many times, an organization will install an IDS to help document existing threats to company networks, to identify existing issues with violations of security policy, or to deter end-users from consistently violating company or organization security policies.
- Since IDSs were first introduced, they have become a critical component to most major organization’s security infrastructures.

Intrusion Detection Expert system



Types of IDS

There are three types of intrusion detection systems

- Network Intrusion Detection Systems (NIDS),
- Host-based Intrusion Detection Systems (HIDS),
- and Stack based Intrusion Detection Systems (SIDS).

NIDS

Network intrusion detection system:

- A network intrusion detection system analyses network traffic and hosts to locate potential intrusions.
- The NIDS system connects to a network hub, network tap, or network switch that is configured to allow monitoring of network traffic.
- When setting up a network intrusion detection system, the monitoring points are setup at high-traffic areas on the network to examine the network data packets for potentially malicious actions.

HIDS

Host-based Intrusion Detection System:

- Host-based intrusion detection systems are designed to have one network host agent that uses application logs, file-system modifications, and system call analysis to locate intrusions to the network.
- The sensors in a host-based intrusion detection system normally consist of software agent(s).
- A common example of a HIDS are OSSEC and Tripwire.

SIDS

Stack-based intrusion detection systems:

- Stack-based intrusion detection systems were developed as a succeeding technology to HIDS.
- SIDS examine network packets as they travel through the network stack (TCP/IP).
- As a result, the SIDS technology does not incur the overhead of having to communicate with the network interface in promiscuous mode.

Limitations of IDS

Limitations of Intrusion Detection Systems:

- Intrusion detection systems are not perfect.
- Depending on the design of the system, a number of false-positive results can be generated.
- These “false alarms” can originate from bad software, corrupt domain name server information, or local network traffic.
- As a result, a real network attack can be missed if the IDS is not properly configured for the defended network.
- Another vulnerability of IDSs that rely on signature files is updating the signature library to include the latest threats.
- When left undone, the network can be open to attack from the most current threats.

Free IDSs

- Snort
- File System Saint (FSS)
- Advanced Intrusion Detection Environment (AIDE)