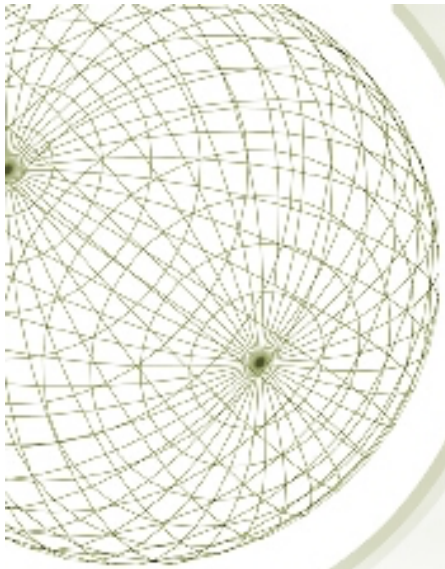
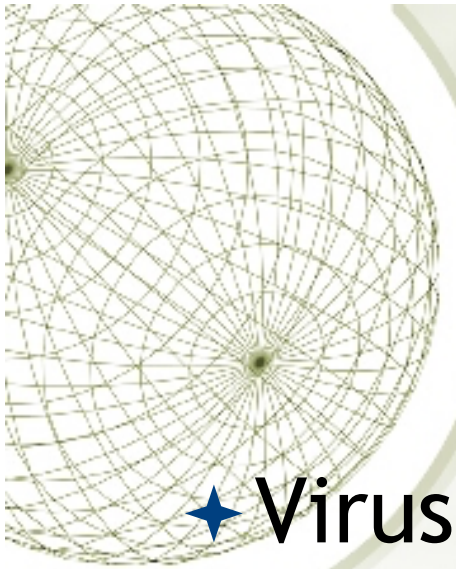


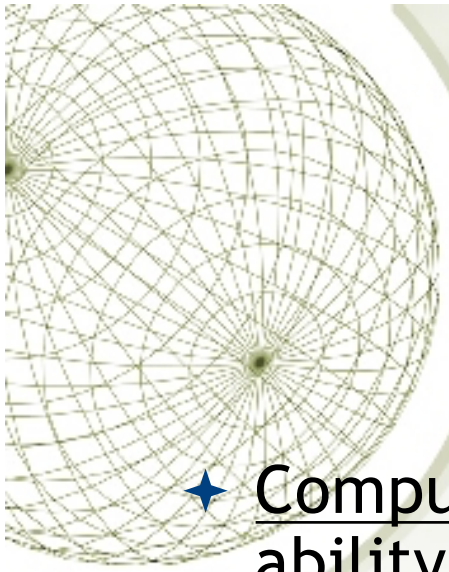
Malicious Software

Ola Flygt
Linnaeus University, Sweden
http://homepage.lnu.se/staff/oflmsi/
Ola.Flygt@lnu.se



Outline

- Viruses and Related Threats
  - Malicious Programs
  - The Nature of Viruses
  - Anti malware Approaches
  - Worm attacks and defences
  - DDoS attacks and countermeasures

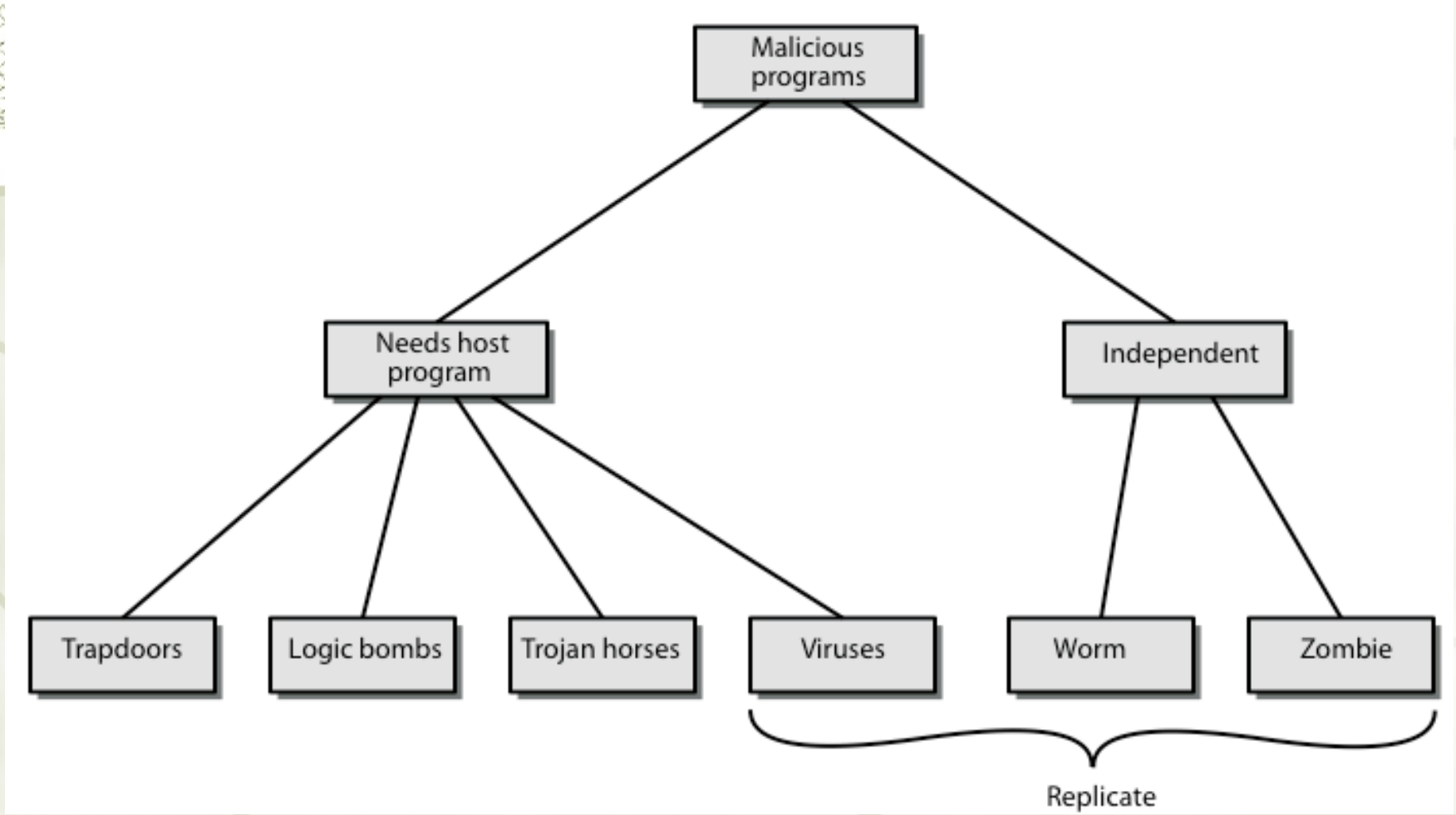


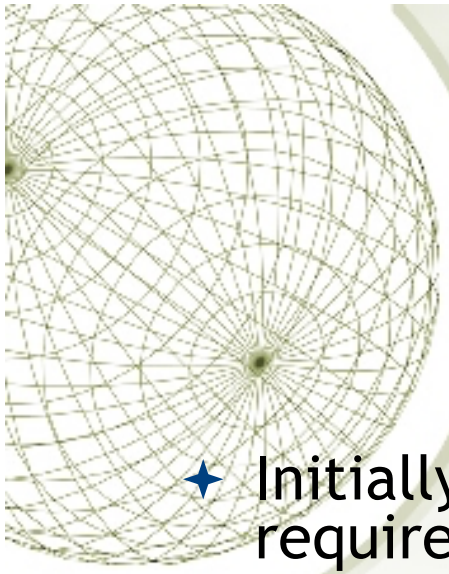
Viruses and "Malicious Programs"

- Computer "Viruses" and related programs have the ability to replicate themselves on an ever increasing number of computers. They originally spread by people sharing floppy disks. Now they spread primarily over the Internet (a "Worm").

- Other "Malicious Programs" may be installed by hand on a single machine. They may also be built into widely distributed commercial software packages. These are very hard to detect before the payload activates (Trojan Horses, Trap Doors, and Logic Bombs).

Taxonomy of Malicious Programs





Attack kits

- Initially the development and deployment of malware required considerable technical skill by software authors
- This changed with the development of virus-creation toolkits in the early 1990s and more general attack kits in the 2000s
  - These toolkits are often known as *crimeware*
  - Include a variety of propagation mechanisms and payload modules that even novices can combine, select, and deploy
  - Can easily be customized with the latest discovered vulnerabilities in order to exploit the window of opportunity between the publication of a weakness and the deployment of patches to close it
  - These kits greatly enlarged the population of attackers able to deploy malware

Attack sources

- Another significant malware development over the last couple of decades is the change from attackers being individuals to more organized and dangerous attack sources
  - These include politically motivated attackers, criminals, organized crime, organizations that sell their services to companies and nations, and national government agencies
- This has significantly changed the resources available and motivation behind the rise of malware leading to development of a large underground economy involving the sale of attack kits, access to compromised hosts, and to stolen information

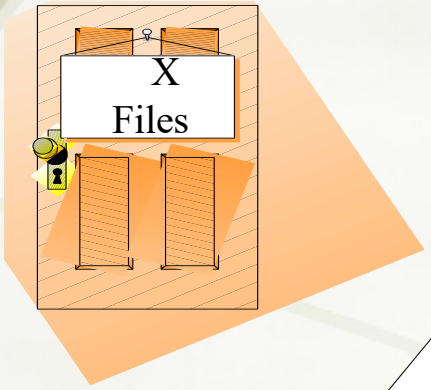
Advanced persistent threat (APT)

- Have risen to prominence in recent years
- A well-resourced, persistent application of a wide variety of intrusion technologies and malware to selected targets, usually business or political
- APTs differ from other types of attack by their careful target selection, and persistent, often stealthy, intrusion efforts over extended periods
  - Aurora, RSA, APT1, and Stuxnet are often cited as examples
- Named as a result of these characteristics:
  - Advanced
    - The individual components may not necessarily be technically advanced, but are carefully selected to suit the chosen target
  - Persistent
    - Determined application of the attacks over an extended period against the chosen target in order to maximize the chance of success
  - Threats
    - Threats to the selected targets as a result of the organized, capable, and well-funded attackers intent to compromise the specifically chosen targets

Kinds of Malicious Code

Trapdoors

Trojan Horses



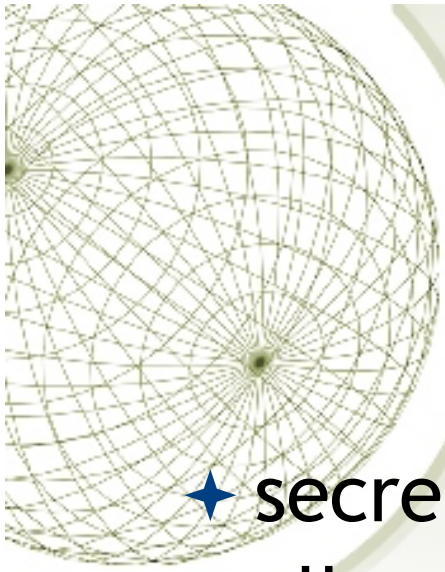
Bacteria

Logic Bombs

Worms

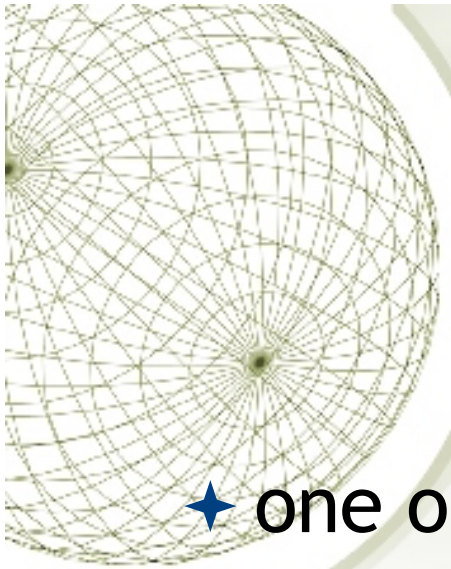
Viruses

[cf. Barbara Edicott-Popovsky and Deborah Frincke, CSSE592/492, U. Washington]



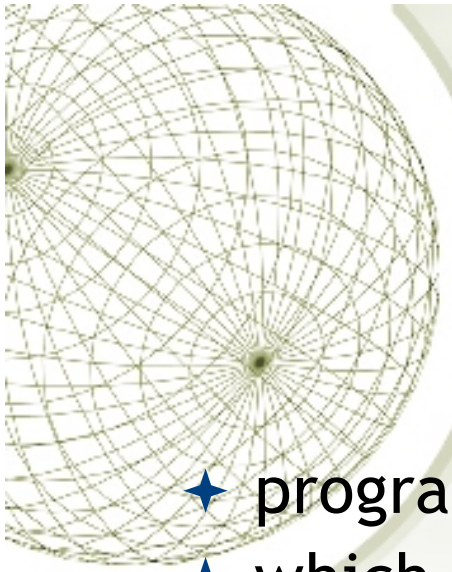
Backdoor or Trapdoor

- 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 OS



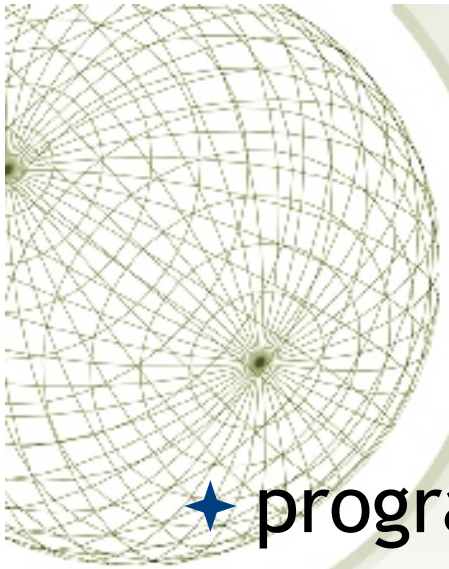
Logic Bomb

- one of oldest types of malicious software
- code embedded in legitimate program
- activated when specified conditions met
  - E.g., 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
  - E.g., 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
- Mail the password file

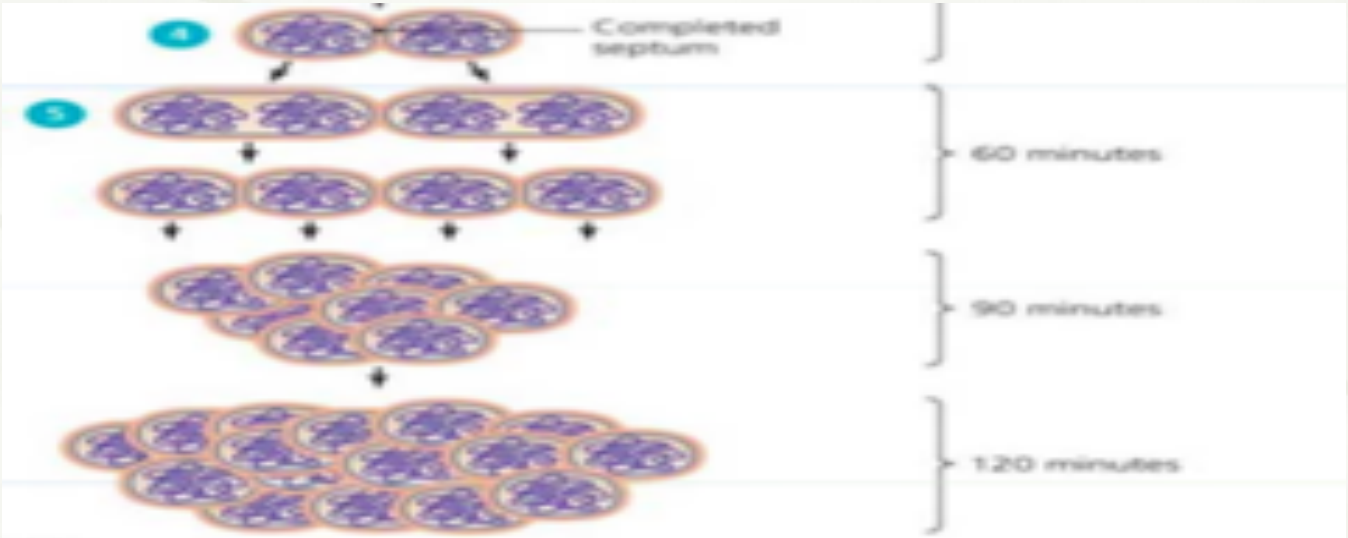


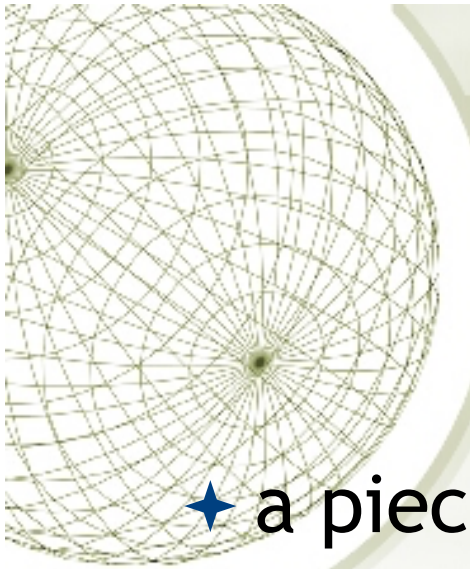
Zombie

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

Bacteria

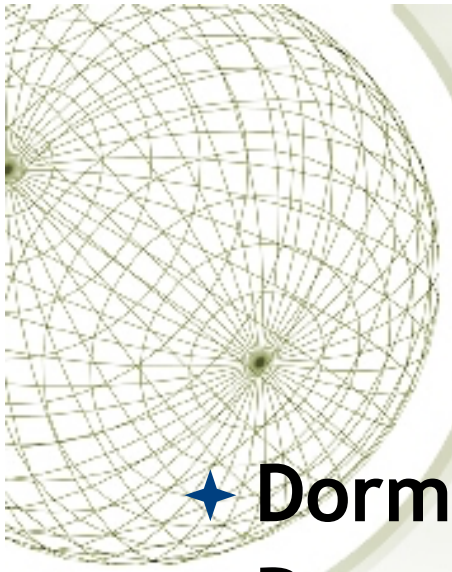
- A “Bacteria” replicates until it fills all disk space, or CPU cycles





Viruses

- a piece of self-replicating code attached to some other code
- attaches itself to another program and executes secretly when the host program is executed.
- propagates itself & carries a payload
  - carries code to make copies of itself
  - as well as code to perform some covert task

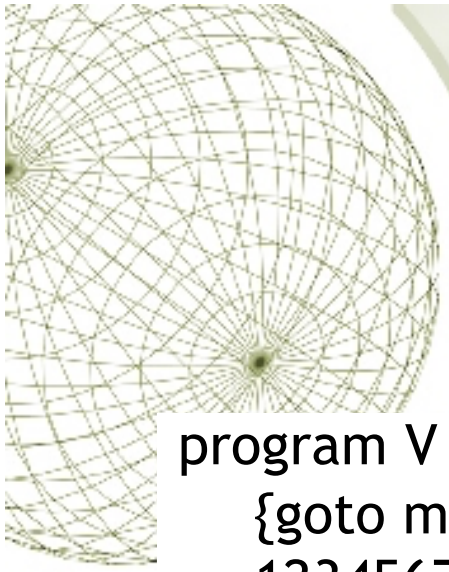


Virus Phases

- **Dormant phase** - the virus is idle
- **Propagation phase** - the virus places an identical copy of itself into other programs
- **Triggering phase -** the virus is activated to perform the function for which it was intended
- **Execution phase** – the function is performed

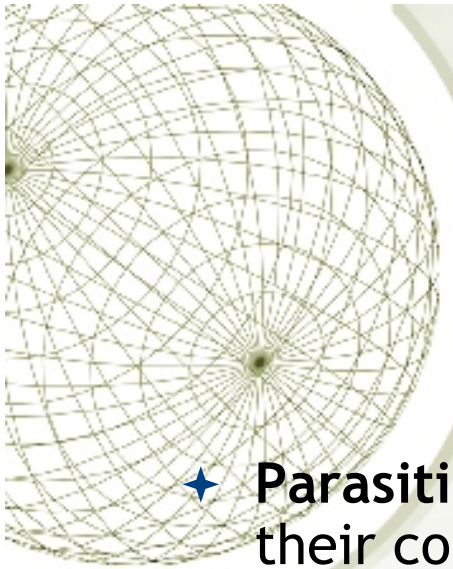
Details usually machine/OS specific

exploiting features/weaknesses



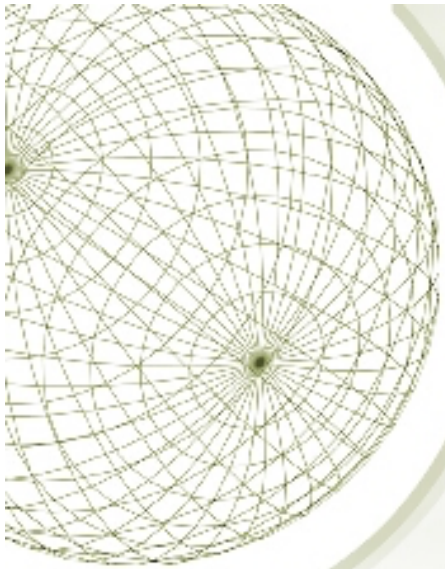
Virus Structure

```
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:
}
```

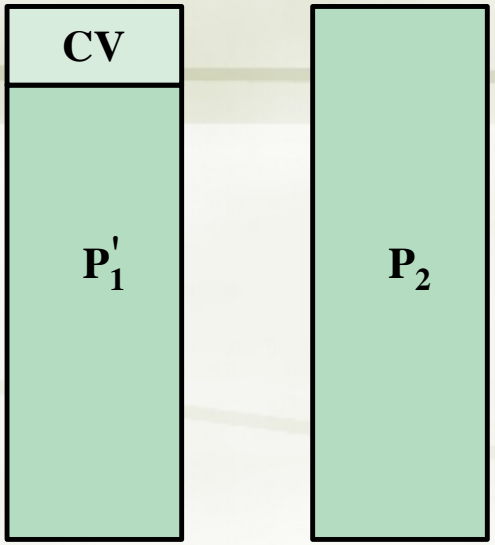


Types of Viruses

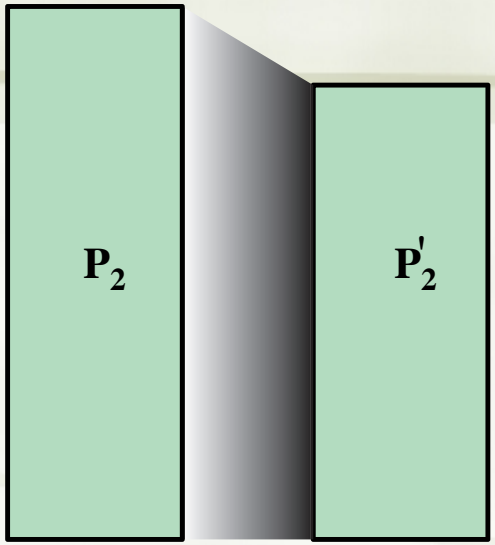
- **Parasitic Virus** - attaches itself to executable files as part of their code. Runs whenever the host program runs.
- **Memory-resident Virus** - Lodges in main memory as part of the residual operating system.
- **Boot Sector Virus** - infects the boot sector of a disk, and spreads when the operating system boots up (original DOS viruses).
- **Stealth Virus** - explicitly designed to hide from Virus Scanning programs.
- **Polymorphic Virus** - mutates with every new host to prevent signature detection.
- **Metamorphic virus** - mutates with every infection, but rewrites itself completely every time. Making it extremely difficult to detect.



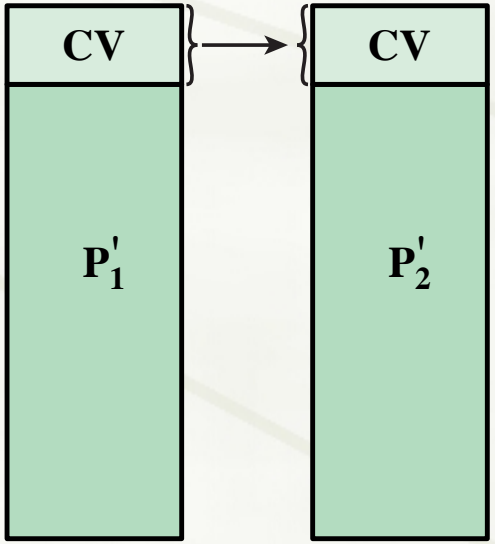
A Compression Virus



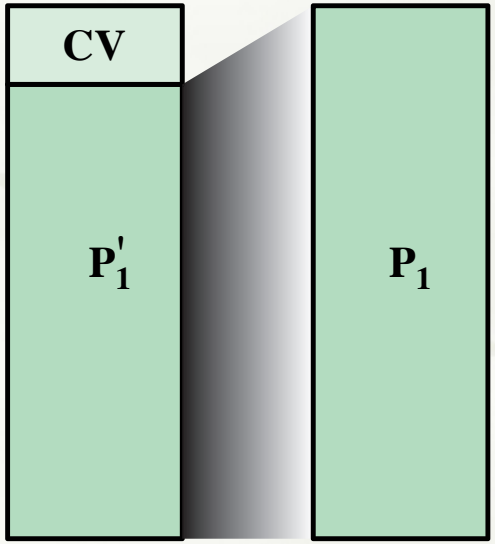
t_0: P_1' is infected version of P_1; P_2 is clean



t_1: P_2 is compressed into P_2'



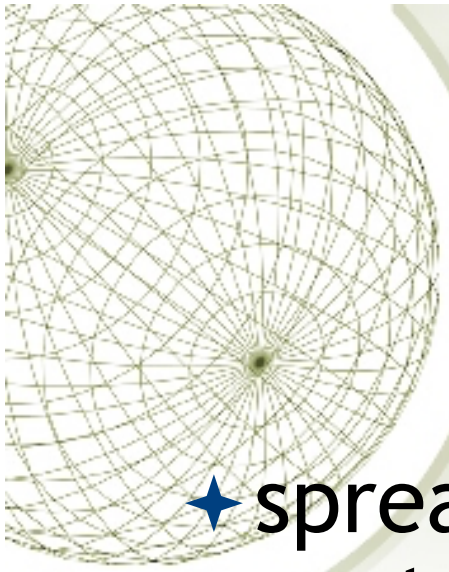
t_2: CV attaches itself to P_2'



t_3: P_1' is decompressed into the original program P_1

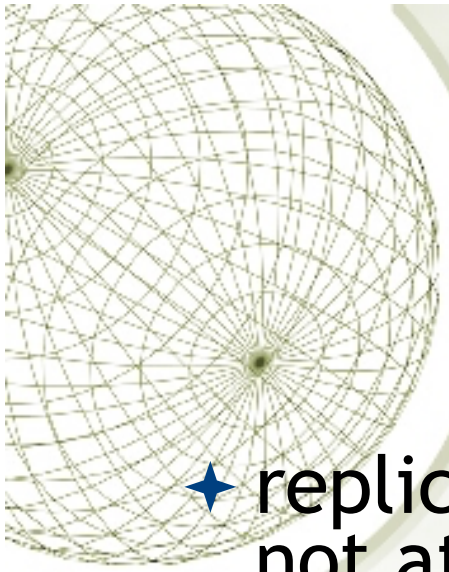
Macro Viruses

- Microsoft Office applications allow “macros” to be part of the document. The macro could run whenever the document is opened, or when a certain command is selected (Save File)
- Platform independent (in theory)
- Infect documents, delete files, generate email and edit letters



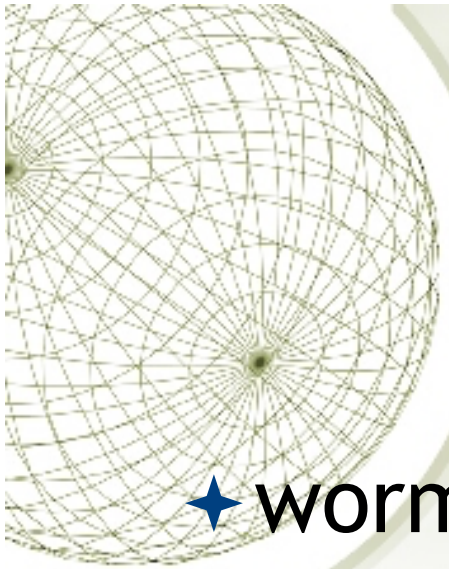
Email Virus

- 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 propagates very quickly
- usually targeted at Microsoft Outlook mail agent & Word/Excel documents



Worms

- replicating but not infecting program (does not attach itself to a program)
- typically spreads over a network
- using users distributed privileges or by exploiting system vulnerabilities
- worms perform unwanted functions
- widely used by hackers to create **zombie PC's**, subsequently used for further attacks, esp. DoS
- major issue is lack of security of permanently connected systems, esp. PC’s



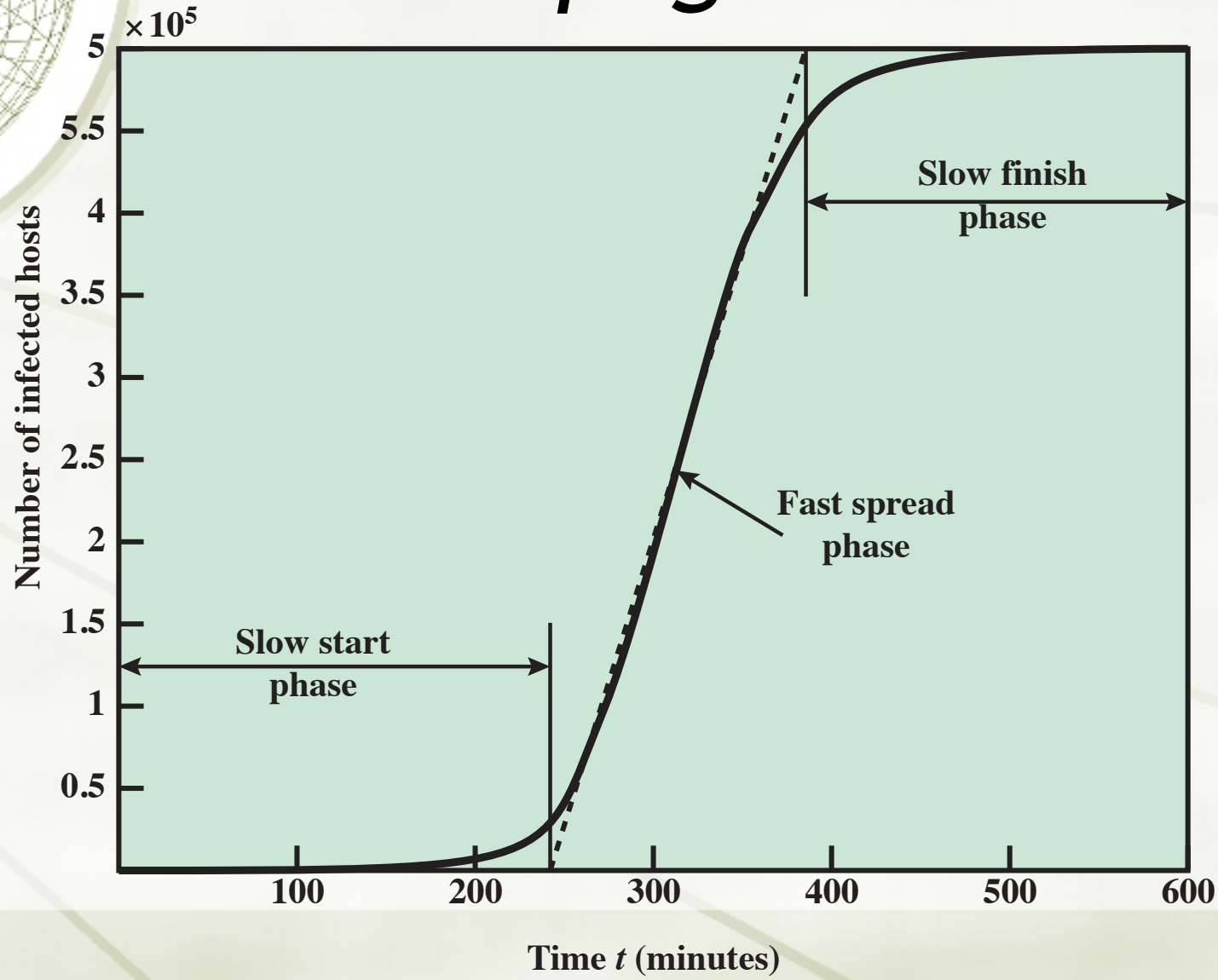
Worm Operation

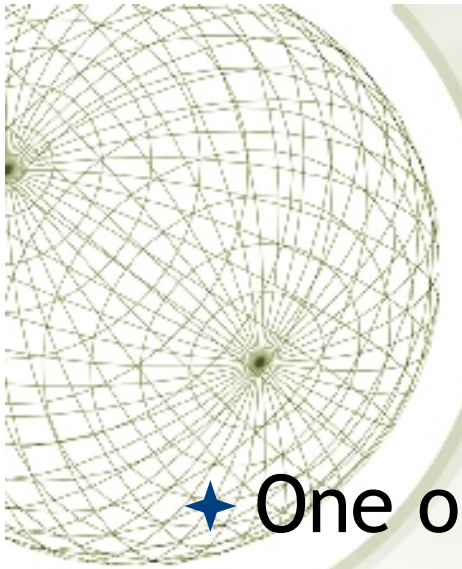
- worm has phases like those of viruses:
  - dormant
  - propagation
    - search for other systems to infect
    - establish connection to target remote system
    - replicate self onto remote system
  - triggering
  - execution

Target discovery

- Scanning/fingerprinting
  - The function in the propagation phase for a network worm to search for other systems to infect
- Worm network scanning strategies:
  - Random
    - Each compromised host probes random addresses in the IP address space, using a different seed
    - Produces a high volume of Internet traffic, which may cause generalized disruption even before the actual attack is lunched
  - Hit list
    - The attacker first compiles a long list of potential vulnerable machines
    - Once the list is compiled, the attacker begins infecting machines on the list
    - Each infected machine is provided with a portion of the list to scan
    - This results in a very short scanning period, which may make it difficult to detect that infection is taking place
  - Topological
    - Uses information contained on an infected victim machine to find more hosts to scan
  - Local subnet
    - If a host is infected behind a firewall, that host then looks for targets in its own local network
    - The host uses the subnet address structure to find other hosts that would otherwise be protected by the firewall

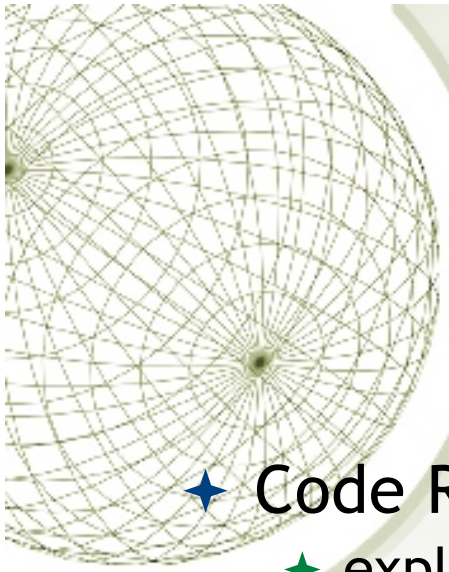
Worm Propagation Model





Morris Worm

- One of the best known classic worms
- released by Robert Morris in 1988
- targeted Unix systems
- using several propagation techniques
  - simple password cracking of local pw file
  - exploit bug in finger daemon
  - exploit debug trapdoor in sendmail daemon
- if any attack succeeds then replicated self

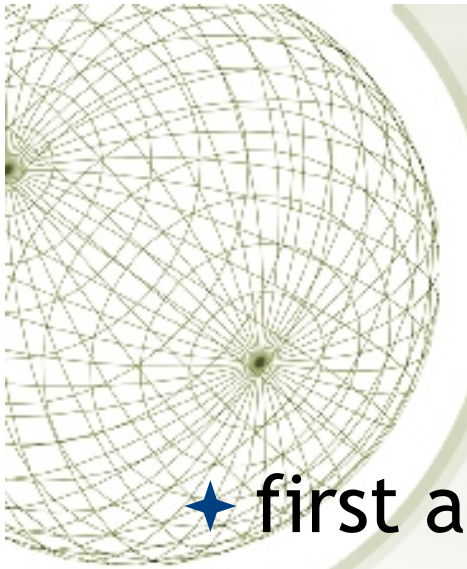


Other Worm Attacks

- Code Red (2001)
  - exploiting MS IIS bug
  - probes random IP address, does DDoS attack
- Code Red II variant includes backdoor
- SQL Slammer (2003)
  - attacks MS SQL Server
- Mydoom (2004)
  - mass-mailing e-mail worm
  - installed remote access backdoor in infected systems
- Warezov family of worms (2006)
  - scan for e-mail addresses, send in attachment

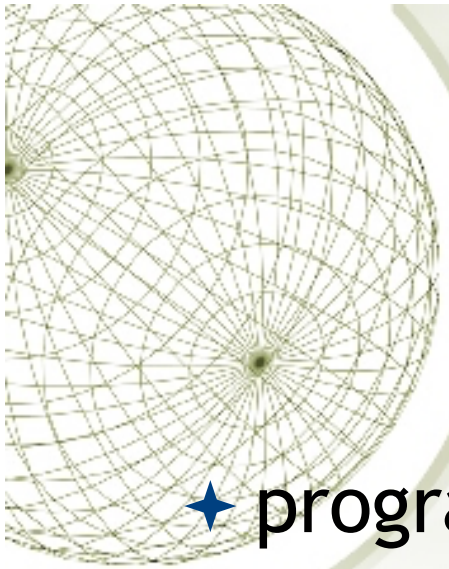
Worm Technology

- multiplatform
- multi-exploit
- ultrafast spreading
- polymorphic
- metamorphic
- transport vehicles
- zero-day exploit



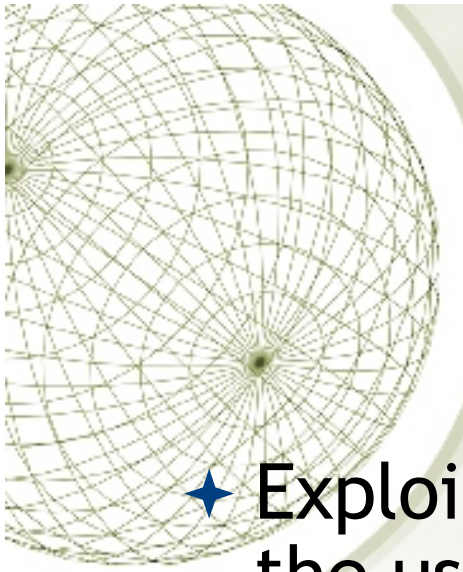
Mobile Phone Worms

- first appeared on mobile phones in 2004
  - target smartphone which can install s/w
- they communicate via Bluetooth or MMS
- to disable phone, delete data on phone, or send premium-priced messages
- CommWarrior, launched in 2005
  - replicates using Bluetooth to nearby phones
  - and via MMS using address-book numbers



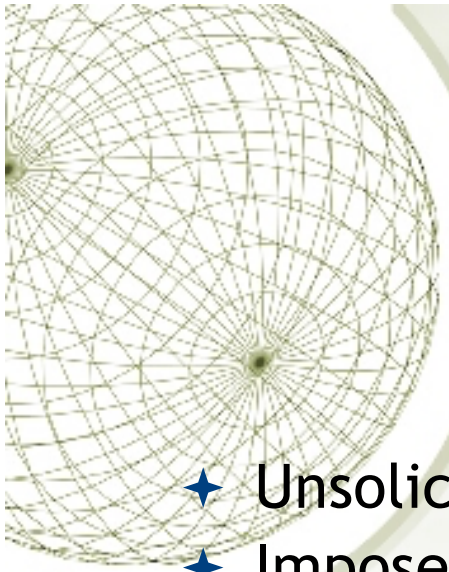
Mobile Code

- program/script/macro that runs unchanged
  - on heterogeneous collection of platforms
  - on large homogeneous collection (Windows)
- transmitted from remote system to local system & then executed on local system
- often to inject virus, worm, or Trojan horse
- or to perform own exploits
  - unauthorized data access, root compromise



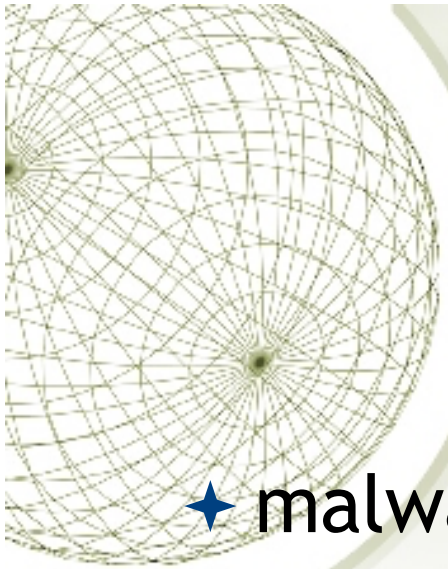
Drive-by-downloads

- Exploits browser vulnerabilities so that when the user views a Web page controlled by the attacker, it contains code that exploits the browser bug to download and install malware on the system without the user's knowledge or consent
- Does not actively propagate as a worm does, but rather waits for unsuspecting users to visit the malicious Web page in order to spread to their systems



SPAM

- Unsolicited bulk e-mail
- Imposes significant costs on both the network infrastructure needed to relay this traffic and on users who need to filter their legitimate e-mails
- Most recent spam is sent by botnets using compromised user systems
- Is a significant carrier of malware
- May be used in a phishing attack
- Although a significant security concern, in many cases it requires the user's active choice to view the e-mail and any attached document or to permit the installation of some program, in order for the compromise to occur

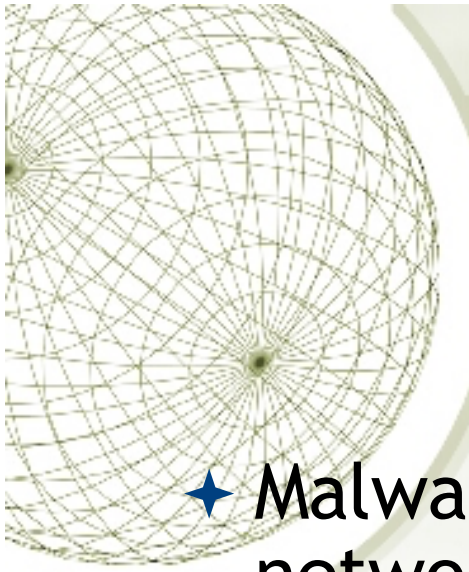


Multiple-Threat Malware

- malware may operate in multiple ways
- multipartite virus infects in multiple ways
  - eg. multiple file types
- blended attack uses multiple methods of infection or transmission
  - to maximize speed of contagion and severity
  - may include multiple types of malware
  - eg. Nimda has worm, virus, mobile code
  - can also use IM & P2P

Payload – system corruption

- Once malware is active on the target system, the next concern is what actions it will take on this system
- Examples:
  - Data destruction on the infected system when certain trigger conditions were met
  - Display unwanted messages or content on the user's system when triggered
  - Encrypt the user's data and demand payment in order to access the key needed to recover this information (ransomware)
  - Inflict real-world damage on the system
    - Attempt to rewrite the BIOS code used to initially boot the computer
    - Target specific industrial control system software
  - Logic bomb
    - Code embedded in the malware that is set to "explode" when certain conditions are met

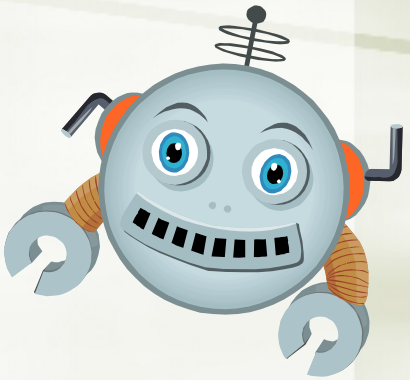


Payload – attack agent

- Malware subverts the computational and network resources of the infected system for use by the attacker
  - Bot (robot), zombie, drone
  - Secretly takes over another Internet-attached computer and then uses that computer to launch or manage attacks that are difficult to trace to the bot's creator
- A *botnet* is a collection of bots often capable of acting in a coordinated manner

Uses of bots

- Distributed denial-of-service (DDoS) attacks
- Spamming
- Sniffing traffic
- Keylogging
- Spreading new malware
- Installing advertisement add-ons and browser helper objects (BHOs)
- Attacking Internet Relay Chat (IRC) networks
- Manipulating online polls/games



Remote control facility

- Distinguishes a bot from a worm
  - A worm propagates itself and activates itself, whereas a bot is controlled from some central facility
- Typical means of implementing is on an IRC server
- More recent botnets use covert communication channels via protocols such as HTTP
- Distributed control mechanisms, using peer-to-peer protocols, are also used, to avoid a single point of failure
- Once a communications path is established between a control module and the bots, the control module can activate the bots
  - Can also issue update commands that instruct the bots to download a file from some Internet location and execute it

Payload – information theft

Keylogger

- Captures keystrokes on the infected machine to allow an attacker to monitor user login and password credentials

Spyware

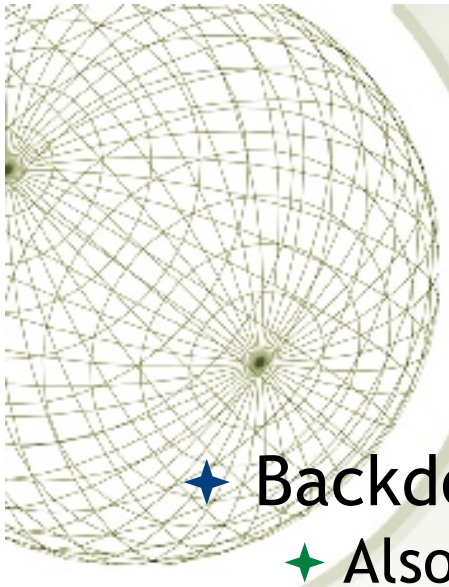
- Developed in response to efforts to try and stop keylogging
- Subvert the compromised machine to allow monitoring of a wide range of activity on the system which can result in significantly compromising the user's personal information

Phishing

- Exploits social engineering to leverage the user's trust by masquerading as communication from a trusted source

Spear-phishing

- An e-mail claiming to be from a trusted source, however, the recipients are carefully researched by the attacker, and each e-mail is carefully crafted to suit its recipient specifically



Payload - stealthing

- Backdoor
  - Also know as a *trapdoor*
  - Is a secret entry point into a program that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures
  - Code that recognizes some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events
  - Usually implemented as a network service listening on some nonstandard port that the attacker can connect to and issue commands through to be run on the compromised system

Payload – stealthing

- Rootkit
  - A set of programs installed on a system to maintain covert access to that system with administrator (or root) privileges, while hiding evidence of its presence to the greatest extent possible
  - Alters the host's standard functionality in a malicious and stealthy way
  - An attacker has complete control of the system and can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand
  - Hides by subverting the mechanisms that monitor and report on the processes, files, and registries on a computer

Rootkits

- Can be classified using the following characteristics:

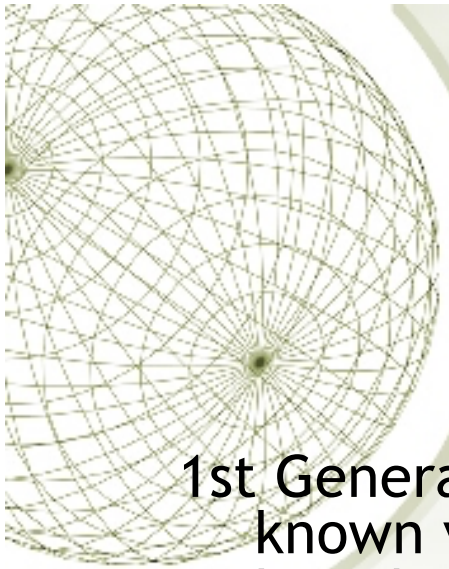
Persistent	• Activates each time the system boots
Memory based	• Has no persistent code and therefore cannot survive a reboot
User mode	• Intercepts calls to application program interfaces (APIs) and modifies returned results
Kernel mode	• Can intercept calls to native APIs in kernel mode
Virtual machine based	• Installs a lightweight virtual machine monitor and then runs the operating system in a virtual machine above it
External mode	• Malware is located outside the normal operation mode of the targeted system, in BIOS or system management mode, where it can directly access hardware

Malicious Software Protection

- Have well-known virus protection and anti spybot programs etc., configured to scan disks and downloads automatically for known viruses.
- Do not execute programs (or "macro's") from unknown sources (e.g., PS files, HyperCard files, MS Office documents.
- Avoid the most common operating systems and email programs, if possible.

Malicious Software Protection

- Best countermeasure is prevention (do not allow a virus to get into the system in the first place.)
- 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



Host-Based Scanners

1st Generation, Scanners: searched files for any of a library of known virus “signatures.” Checked executable files for length changes.

2nd Generation, Heuristic Scanners: looks for more general signs than specific signatures (code segments common to many viruses). Checked files for checksum or hash changes.

3rd Generation, Activity Traps: stay resident in memory and look for certain patterns of software behaviour (e.g., scanning files).

4th Generation, Full Featured: combine the best of the techniques above. Scanning & activity traps, access controls etc.

Host-based behavior-blocking software

- Integrates with the operating system of a host computer and monitors program behavior in real time for malicious actions
- The software then blocks potentially malicious actions before they have a chance to affect the system
- Can block suspicious software in real time so it has an advantage over antivirus detection techniques such as fingerprinting or heuristics
- Limitations:
  - Because the malicious code must run on the target machine before all its behaviors can be identified, it can cause harm before it has been detected and blocked

Perimeter scanning approaches

- Antivirus software is used on an organization's firewall and IDS
  - Typically included in e-mail and Web proxy services running on these systems
  - May also be included in the traffic analysis component of an IDS

Two types of monitoring software may be used:

Ingress monitors

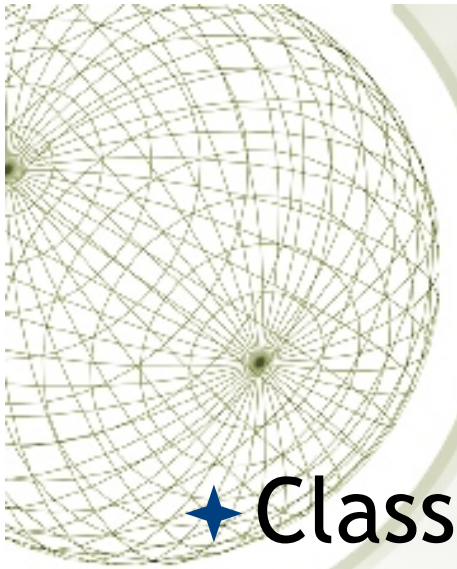
Located at the border between the enterprise network and the Internet

They can be part of the ingress-filtering software of a border router or external firewall or a separate passive monitor

Egress monitors

These can be located at the egress point of individual LANs on the enterprise network as well as at the border between the enterprise network and the Internet

Designed to catch the source of a malware attack by monitoring outgoing traffic for signs of scanning or other suspicious behavior



Perimeter worm countermeasures

- Classes of worm defense:

(Class A) Signature-based worm scan filtering

- This type of approach generates a worm signature, which is then used to prevent worm scans from entering/leaving a network/host

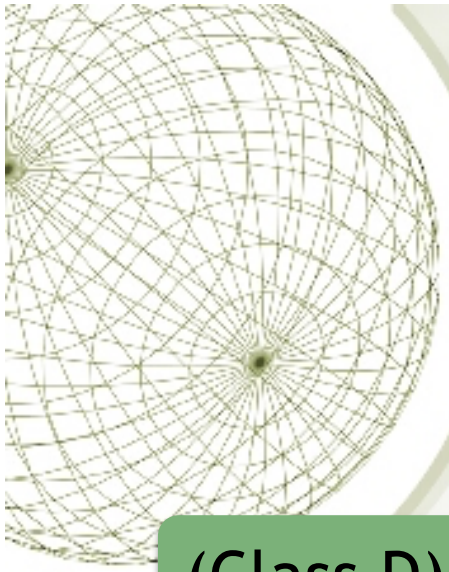
(Class B) Filter-based worm containment

- This approach is similar to class A but focuses on worm content rather than a scan signature

(Class C) Payload-classification-based worm containment

- These network-based techniques examine packets to see if they contain a worm

Continued . . .



Perimeter worm countermeasures

(Class D) Threshold random walk (TRW) scan detection

- Exploits randomness in picking designations to connect to as a way of detecting if a scanner is in operation

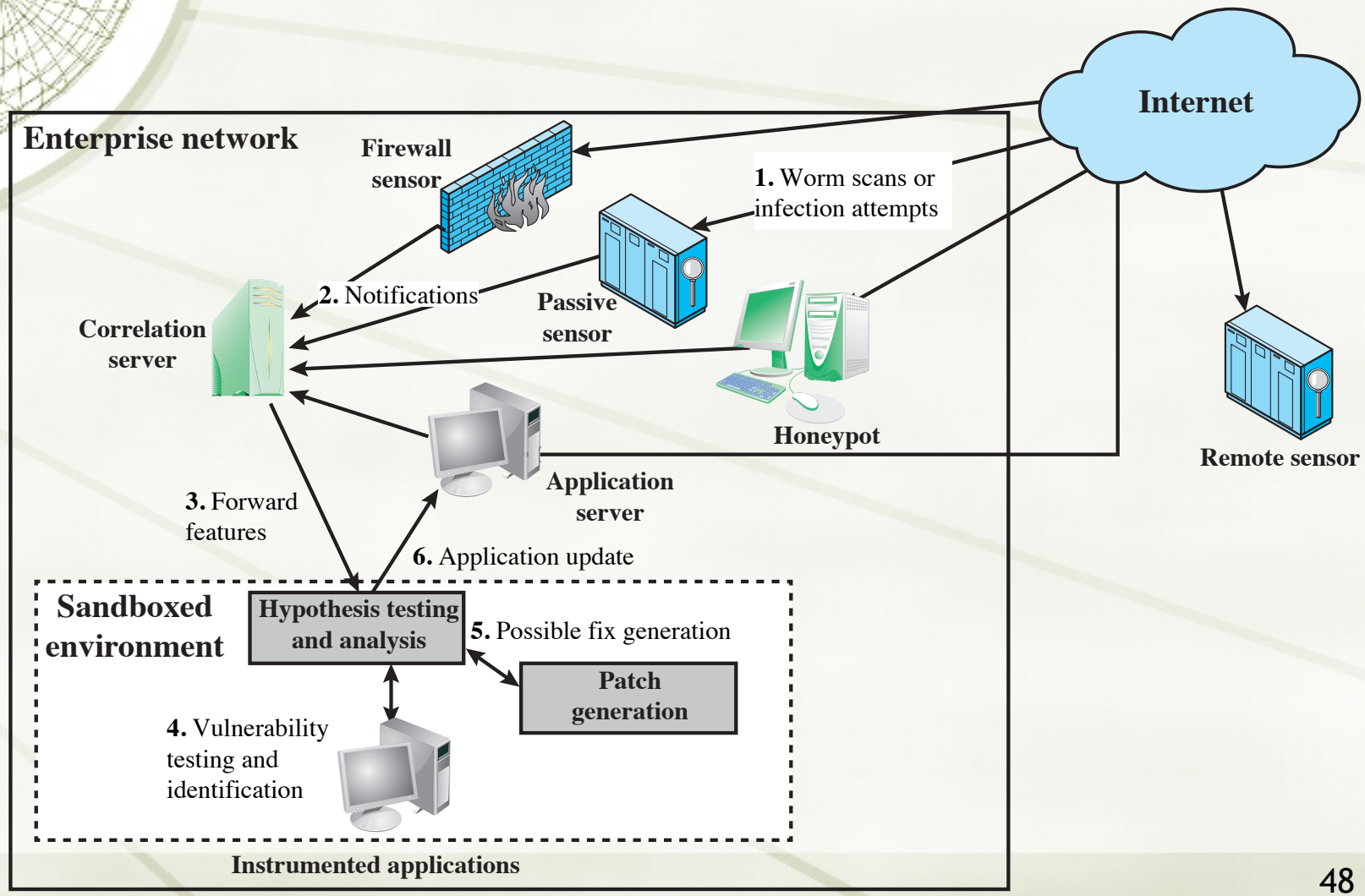
(Class E) Rate limiting

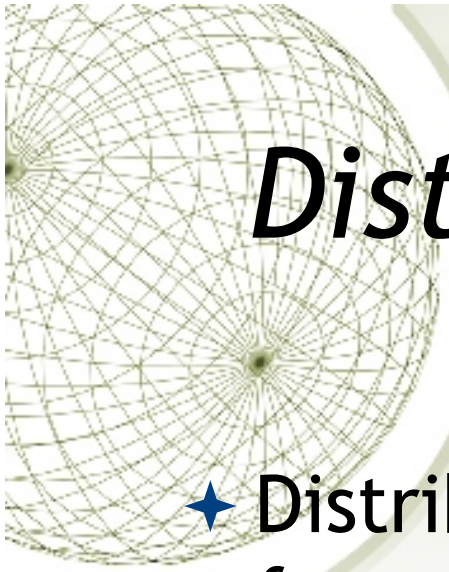
- This class limits the rate of scan-like traffic from an infected host

(Class F) Rate halting

- This approach immediately blocks outgoing traffic when a threshold is exceeded either in outgoing connection rate or in diversity of connection attempts

Placement of worm monitors

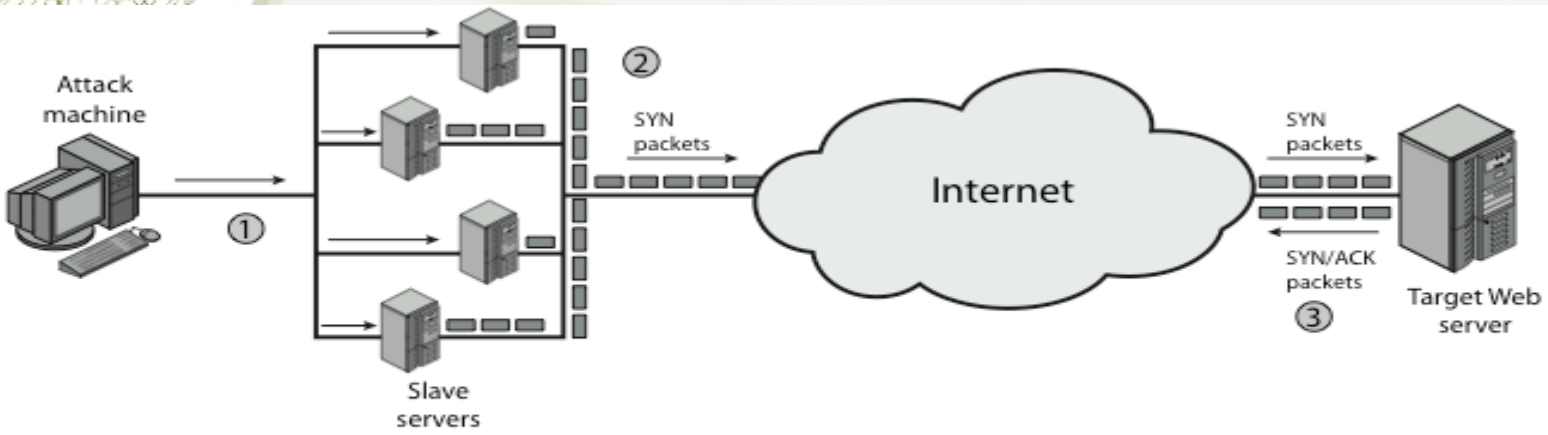




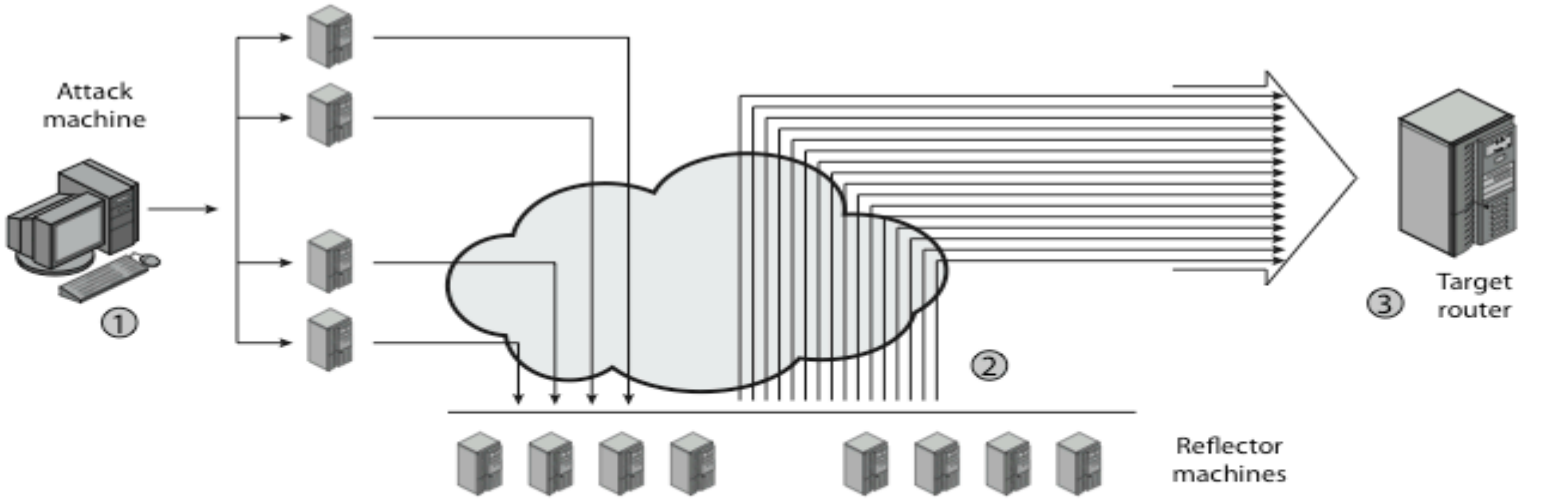
Distributed Denial of Service Attacks (DDoS)

- Distributed Denial of Service (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)

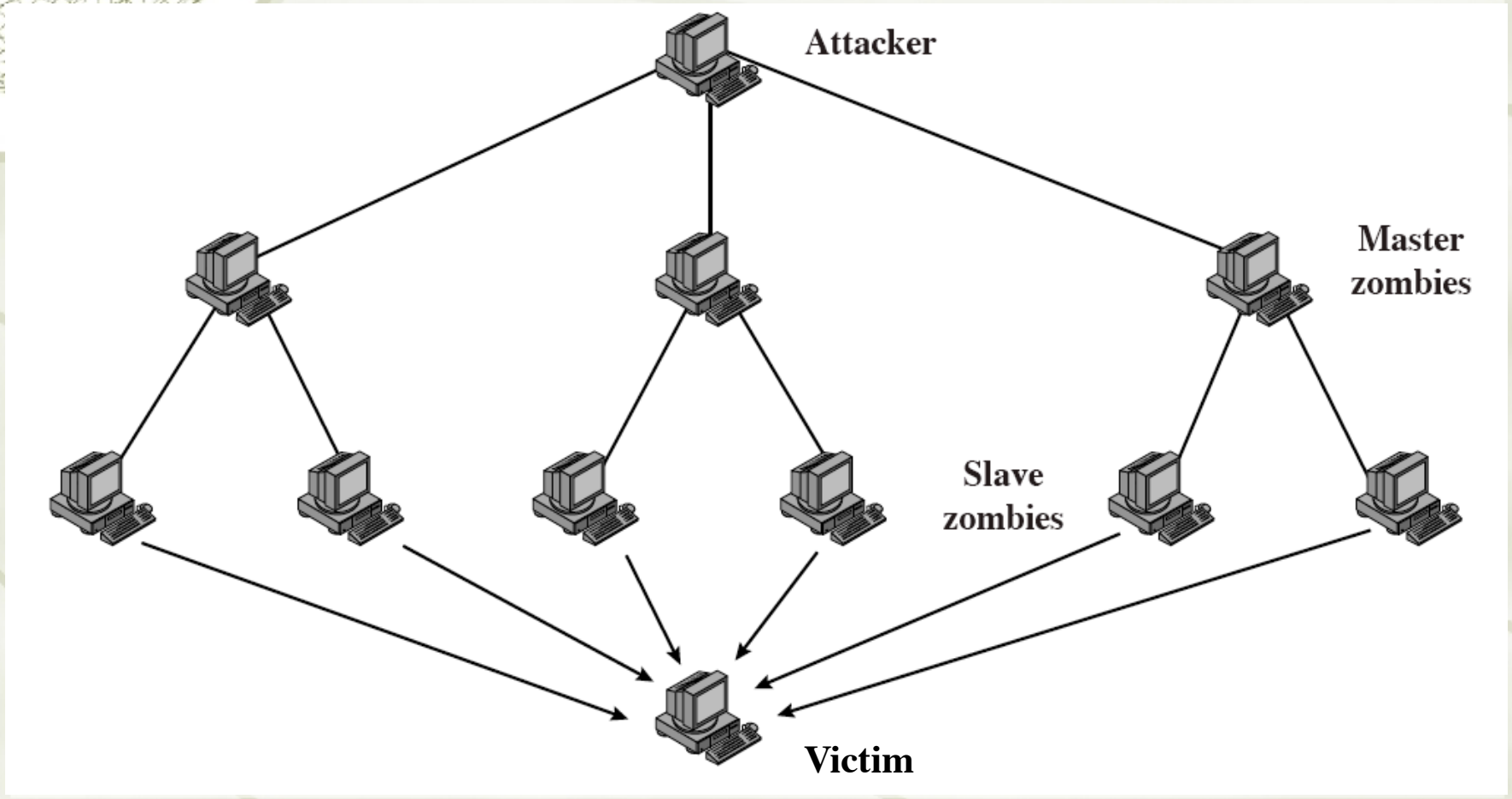


(a) Distributed SYN flood attack

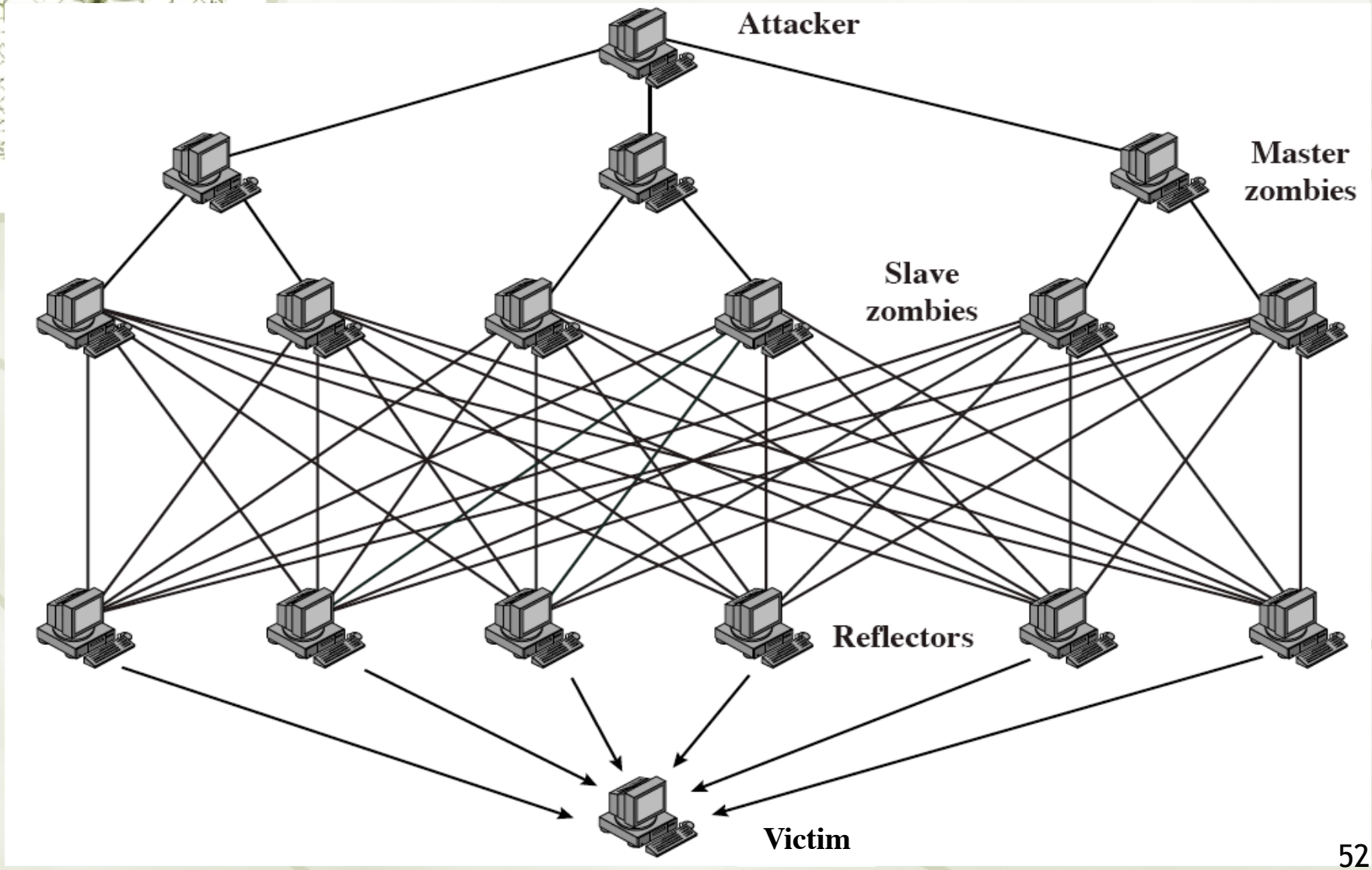


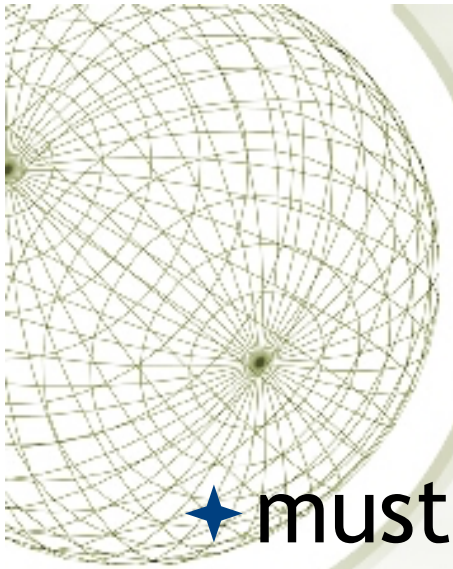
(a) Distributed ICMP attack

Direct DDoS attack



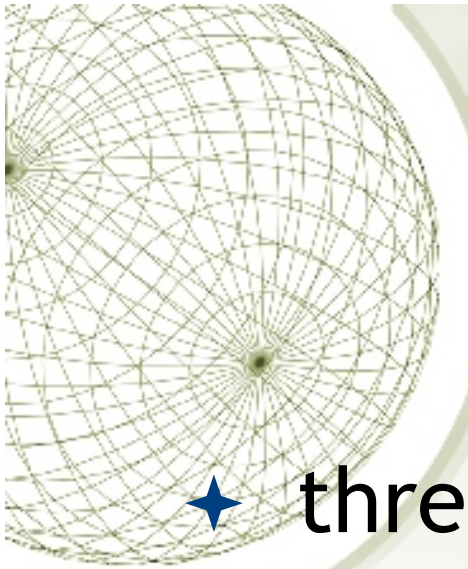
Reflector DDoS Attack





Constructing an Attack Network

- must infect large number of zombies
- needs:
  1. software to implement the DDoS attack
  2. an unpatched vulnerability on many systems
  3. scanning strategy to find vulnerable systems
     - random, hit-list, topological, local subnet



DDoS Countermeasures

- three broad lines of defence:
  1. Before - attack prevention & preemption
  2. During - attack detection & filtering

     Off site filtering can cost around $ 5000/month
  3. After - attack source trace back & identification
- huge range of attack possibilities
- hence evolving countermeasures

DDoS Countermeasures

- Correctly Dimension Central Firewall
- Segmented networks with separate firewalls for public and internal services
- Segmented networks and cloud services
- Functions which prevents or helps during DoS and DDoS attacks (some examples)
  - Limiting the number of connections
  - Bandwidth Management
  - Server Farming and Load balancing (SLB)
  - Load balancing - Redundant Internet connections
  - Protocol validation and consistency checks

Reference: Clavister, " How to protect against DoS and DDoS attacks using Clavister", 2014