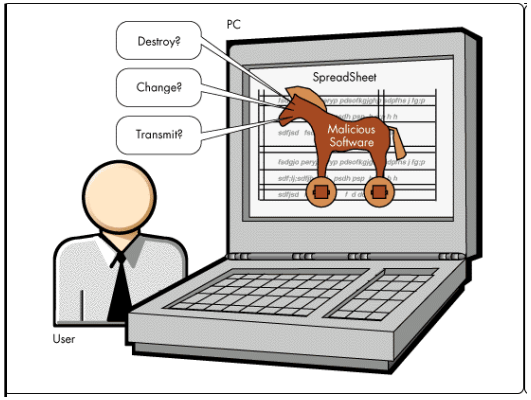
Malicious Software



Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:

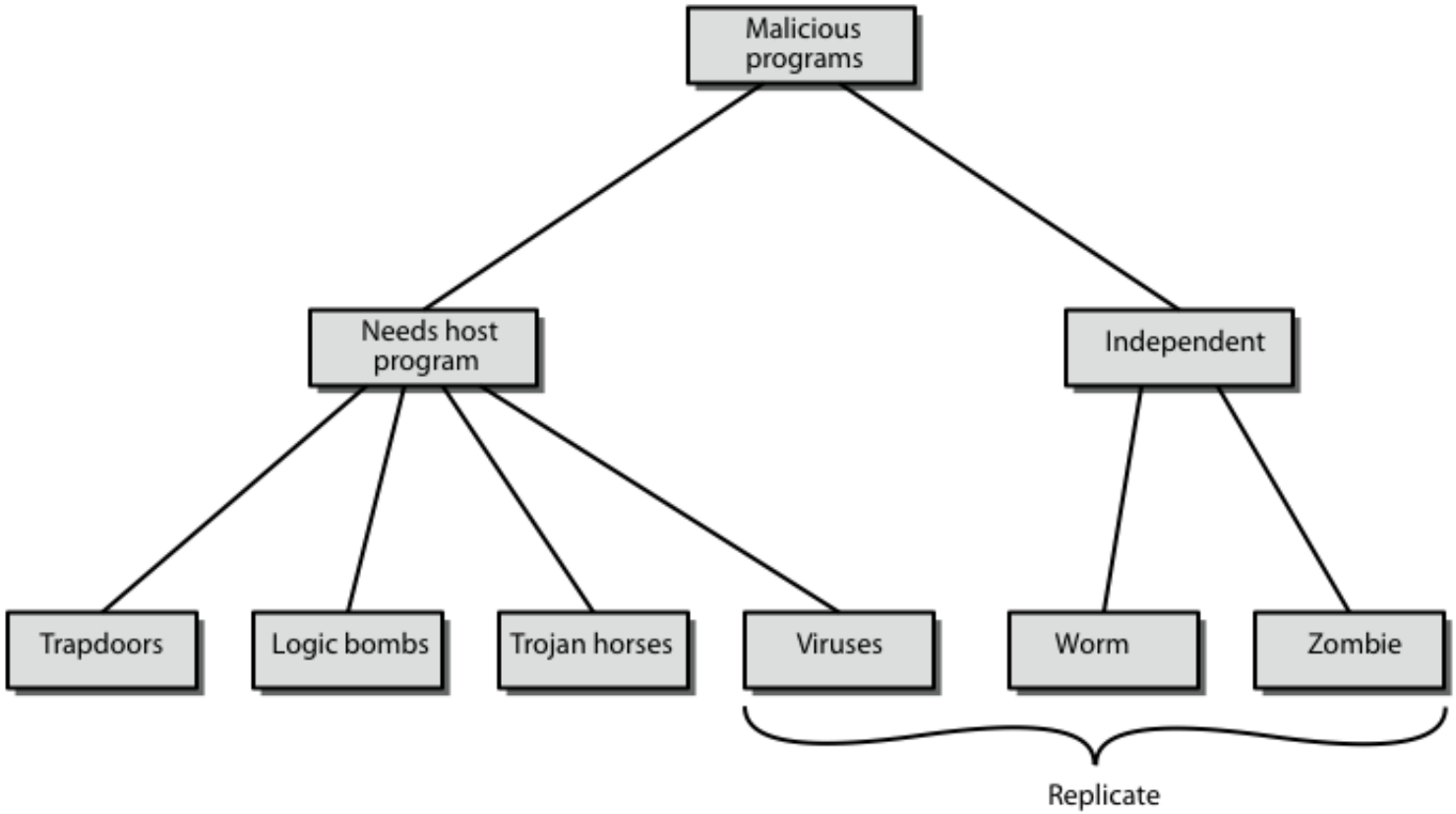
http://www.cse.wustl.edu/~jain/cse571-14/

Overview

1. Types of Malicious Software
2. Viruses
3. Virus Countermeasures
4. Worms
5. Distributed Denial of Service Attacks

These slides are based partly on Lawrie Brown's slides supplied with William Stallings's book "Cryptography and Network Security: Principles and Practice," 6th Ed, 2014.

Malicious Software

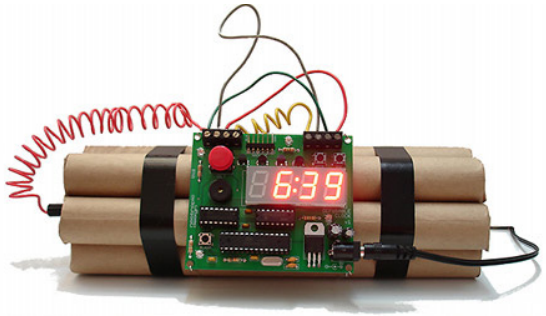


Backdoor or Trapdoor

- Secret entry point into a program
- Allows those, who know, access bypassing usual security procedures
- Commonly used by developers
- A threat when left in production programs
  Allowing exploitation by attackers
- Very hard to block in O/S
- Requires good s/w development & update

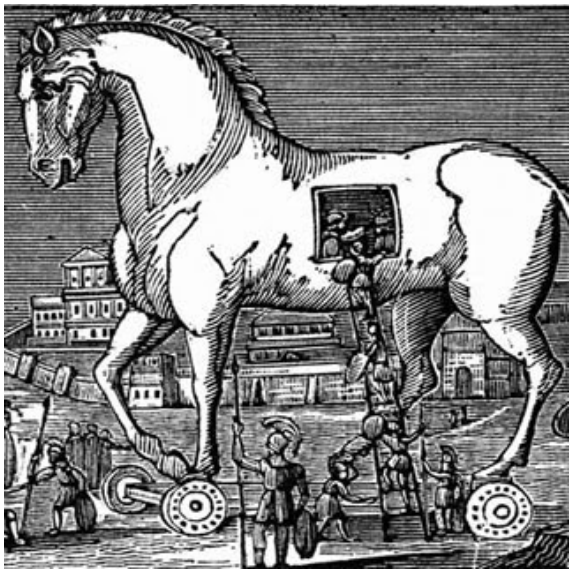
Logic Bomb

- One of oldest types of malicious software
- Code embedded in legitimate program
- Activated when specified conditions met
  - E.g., presence/absence of some file
  - Particular date/time
  - Particular user
- When triggered typically damages the system
  - Modify/delete files/disks, halt machine, etc.



Trojan Horse

- A superficially attractive program with hidden side-effects
  - E.g., game, s/w upgrade, etc.
- When run performs some additional tasks
  - Allows attacker to indirectly gain access
- Often used to propagate a virus/worm or install a backdoor or simply to destroy data



Mobile Code

- Programs/scripts/macros that run unchanged
  - on heterogeneous collection of platforms
  - E.g., java applets
- Transmitted from remote system to local system and then executed on local system
- Inject virus, worm, or Trojan horse
- Perform own exploits:
  unauthorized data access, root compromise

Multiple-Threat Malware

- Malware may operate in multiple ways
- **Multipartite** virus infects in multiple ways
  - E.g., 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
    e.g., Nimda had worm, virus, mobile code
  - Can also use instant messaging and P2P

Ref: http://en.wikipedia.org/wiki/Nimda

Viruses

- Piece of software that infects programs
  - Modifying them to include a copy of the virus
  - The code is executed secretly when host program is run
- Specific to operating system and hardware
  - Taking advantage of their details and weaknesses
- A typical virus goes through phases of:
  - Dormant
  - Propagation
  - Triggering
  - Execution

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 some condition holds}

main:   main-program :=
        {infect-executable;
        if trigger-pulled then do-damage;
        goto next;}

next:

}
```

- Infected program is larger than original

Compression Virus

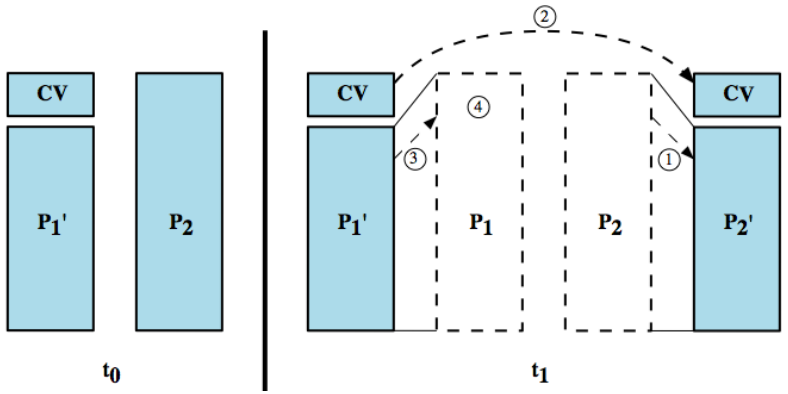
- Infected program has the same size as original

```
    program CV :=

{goto main;
    01234567;

    subroutine infect-executable :=
          {loop:
               file := get-random-executable-file;
          if (first-line-of-file = 01234567) then goto loop;
        (1)    compress file;
        (2)    prepend CV to file;
        }

main:   main-program :=
          {if ask-permission then infect-executable;
        (3)    uncompress rest-of-file;
        (4)    run uncompressed file;}
        }
```



Virus Classification

- By Target:
  - Boot Sector: Spreads when booted with infected disks
  - File Infector: Infects executables
  - Macro Virus: Macro code
- By Concealment Strategy:
  - Encrypted Virus: virus encrypted with a random key stored w the virus
  - Stealth Virus: Explicitly designed to avoid detection
  - Polymorphic Virus: Mutates with every infection
    $\Rightarrow$ Signature varies
  - Metamorphic Virus: Rewrites itself completely with every infection

Macro Virus

- Common in mid-1990s since
  - Platform independent
  - Infect documents
  - Easily spread
- Exploit macro capability of office apps
  - Executable program embedded in office docs
- More recent MS office releases include protection
- Recognized by many anti-virus programs

E-Mail Viruses

- Recent development
- Example: Melissa
  - Exploits MS Word macro in attached doc
  - If attachment opened, macro activates
  - Sends email to all on users address list and does local damage
- Newer versions triggered by just opening email (rather than attachment) ⇒ Much faster propagation

Ref: http://en.wikipedia.org/wiki/Melissa_%28computer_virus%29

Virus Countermeasures

- Prevention: Avoid freewares - ideal solution but difficult
- Realistically need detection, identification, removal
- If detected but can't identify or remove, must discard and replace infected program

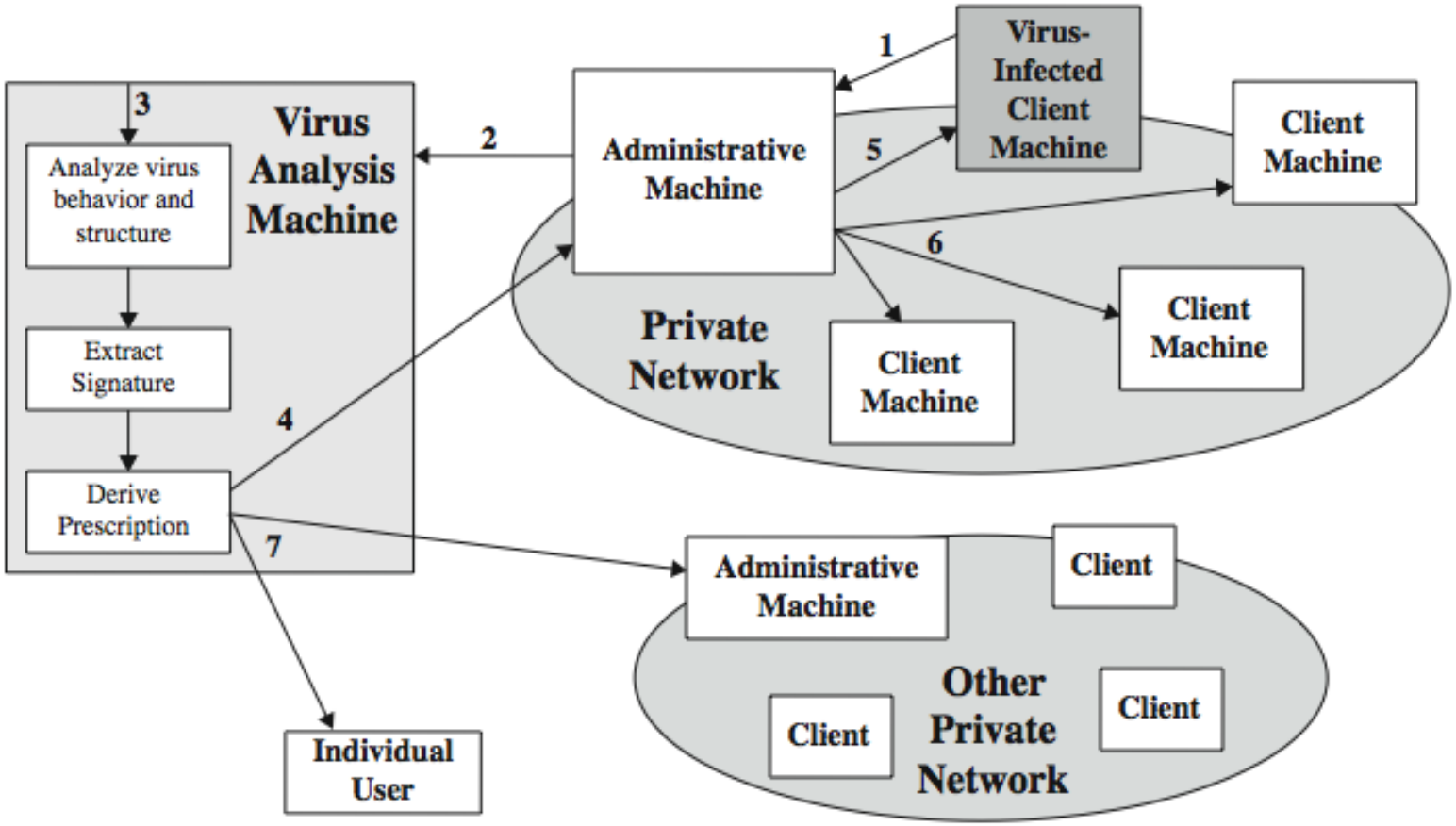
Anti-Virus Evolution

- Virus and antivirus tech have both evolved
- Early viruses simple code, easily removed
- As viruses become more complex, so must the countermeasures
- Generations:
  - First - signature scanners
  - Second - heuristics
  - Third - identify actions
  - Fourth - combination packages

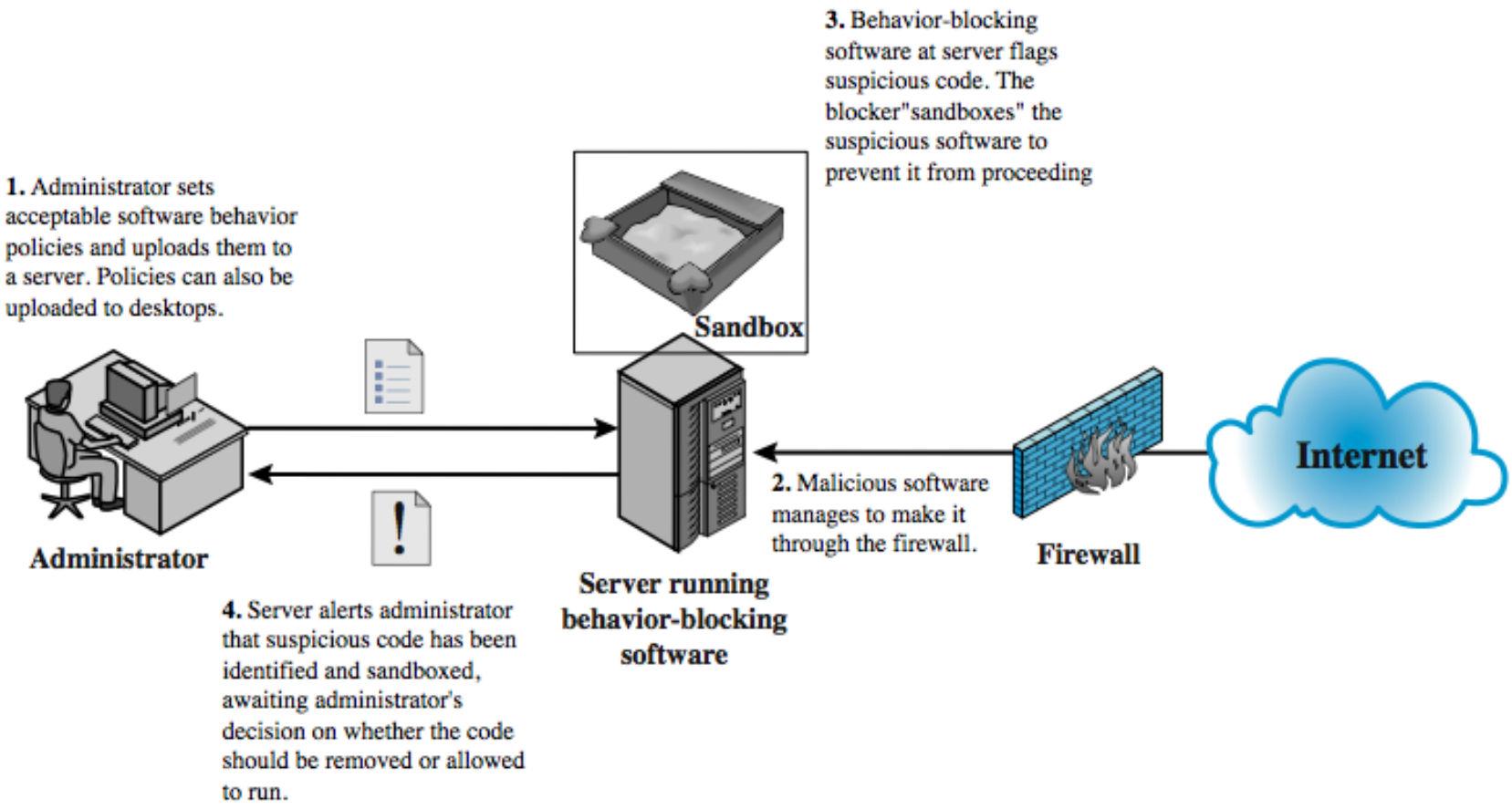
Generic Decryption

- Runs executable files through GD scanner:
  - CPU emulator to interpret instructions
  - Virus scanner to check known virus signatures
  - Emulation control module to manage process
- Lets virus decrypt itself in interpreter
- Periodically scan for virus signatures
- Issue is long to interpret and scan
  - Tradeoff chance of detection vs time delay

Digital Immune System



Behavior-Blocking Software



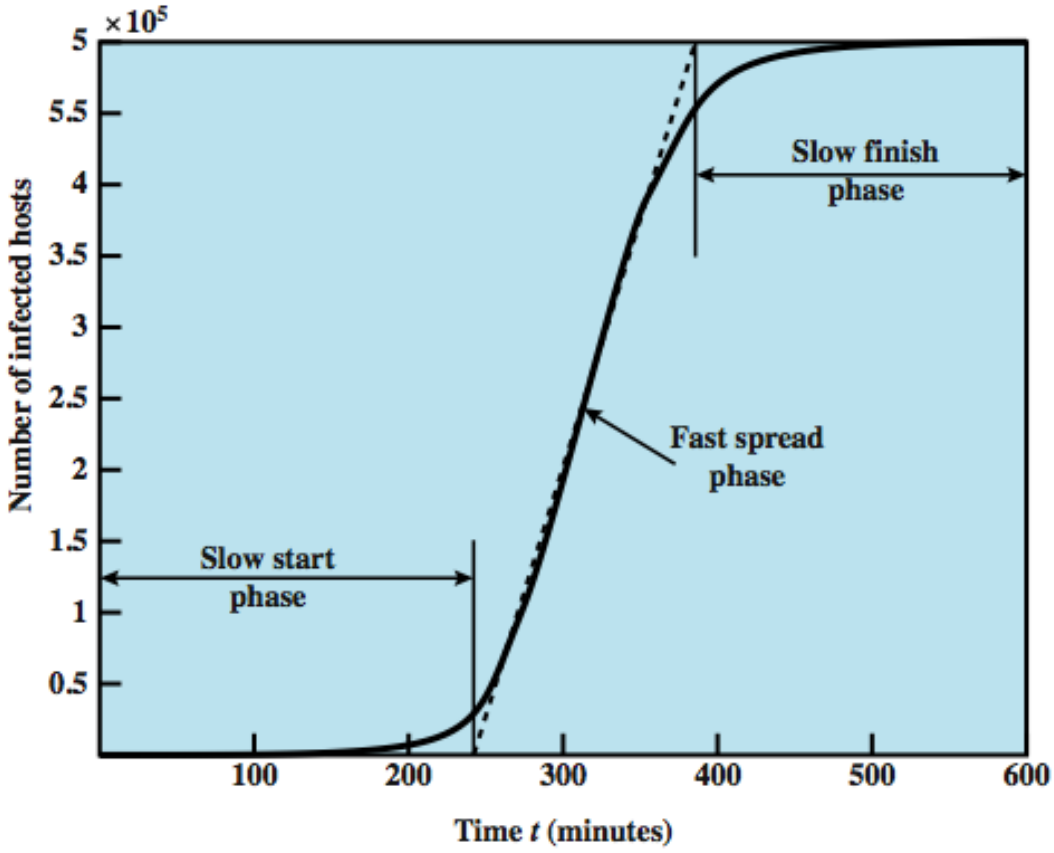
Worms

- Replicating program that propagates over net
  - Using email, remote exec, remote login
- Has phases like a virus:
  - Dormant, propagation, triggering, execution
  - Propagation phase: searches for other systems, connects to it, copies self to it and runs
- May disguise itself as a system process
- Concept seen in Brunner's "Shockwave Rider"
- Implemented by Xerox Palo Alto labs in 1980's

Morris Worm

- One of the best known worms
- Released by Robert Morris in 1988
- Various attacks on UNIX systems
  - Cracking password file to use login/password to logon to other systems
  - Exploiting a bug in the finger protocol
  - Exploiting a bug in sendmail
- If succeed have remote shell access
  - Sent bootstrap program to copy worm over

Worm Propagation Model



Sample Worm Attacks

- Code Red:
  - July 2001 exploiting MS IIS bug
  - Probes random IP address, does DDoS attack
- Code Red II variant includes backdoor
- SQL Slammer
  - Early 2003, attacks MS SQL Server
- Mydoom
  - Mass-mailing e-mail worm that appeared in 2004
  - Installed remote access backdoor in infected systems
- Warezov family of worms
  - Scan for e-mail addresses, send in attachment

Worm Technology

- Multiplatform: Windows, MAC, Linux, …
- Multi-exploit: Browsers, emails, file sharing, …
- Ultrafast spreading: Prior scans to accumulate IP addresses of vulnerable machines
- Polymorphic: Each copy has a new code generated on the fly
- Metamorphic: Different behavior at different stages of propagation
- Transport Vehicles: Worms used to spread DDoS bots
- Zero-day exploit: Vulnerability detected only when the worm is launched.

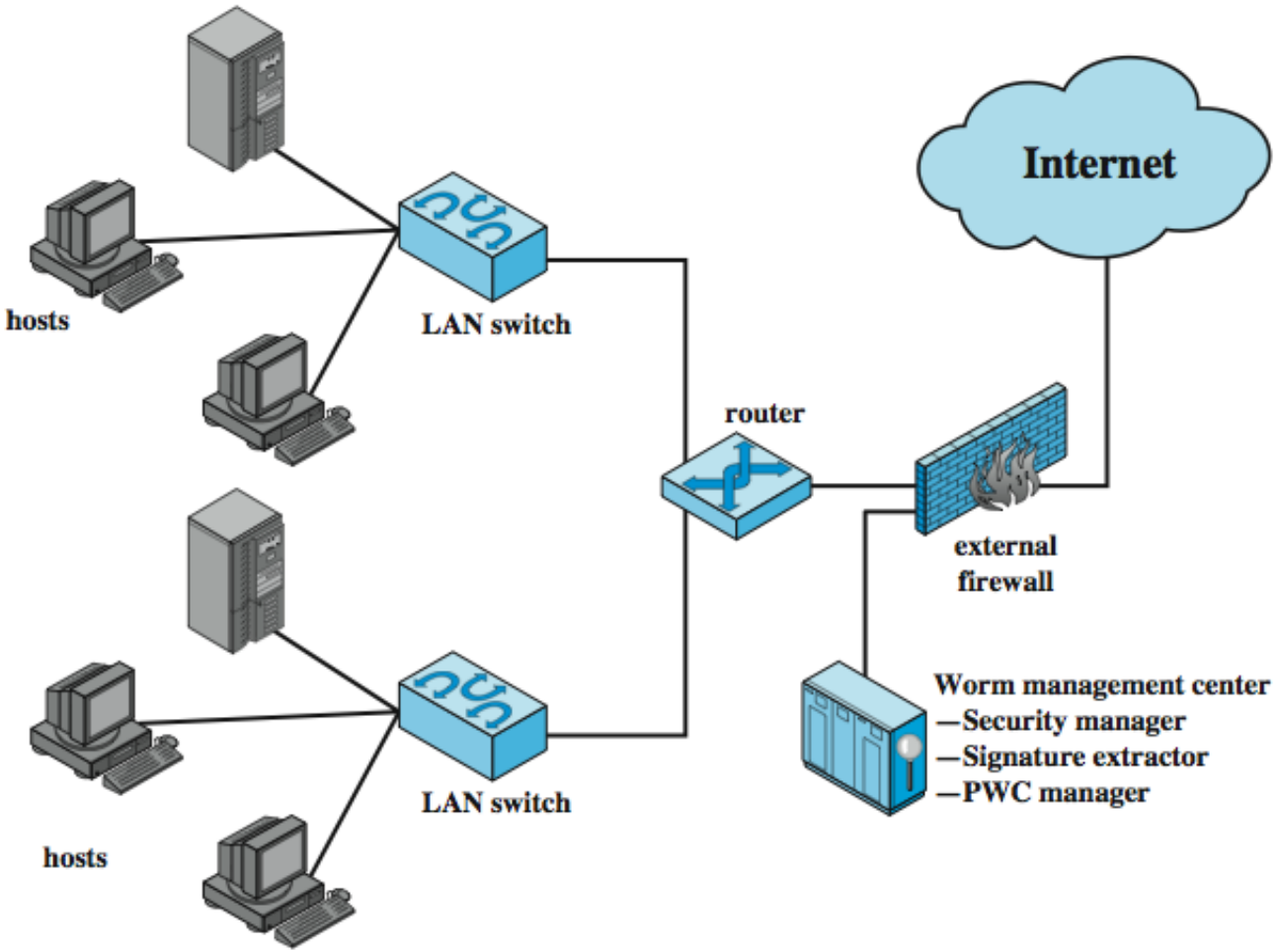
Mobile Phone Worms

- 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

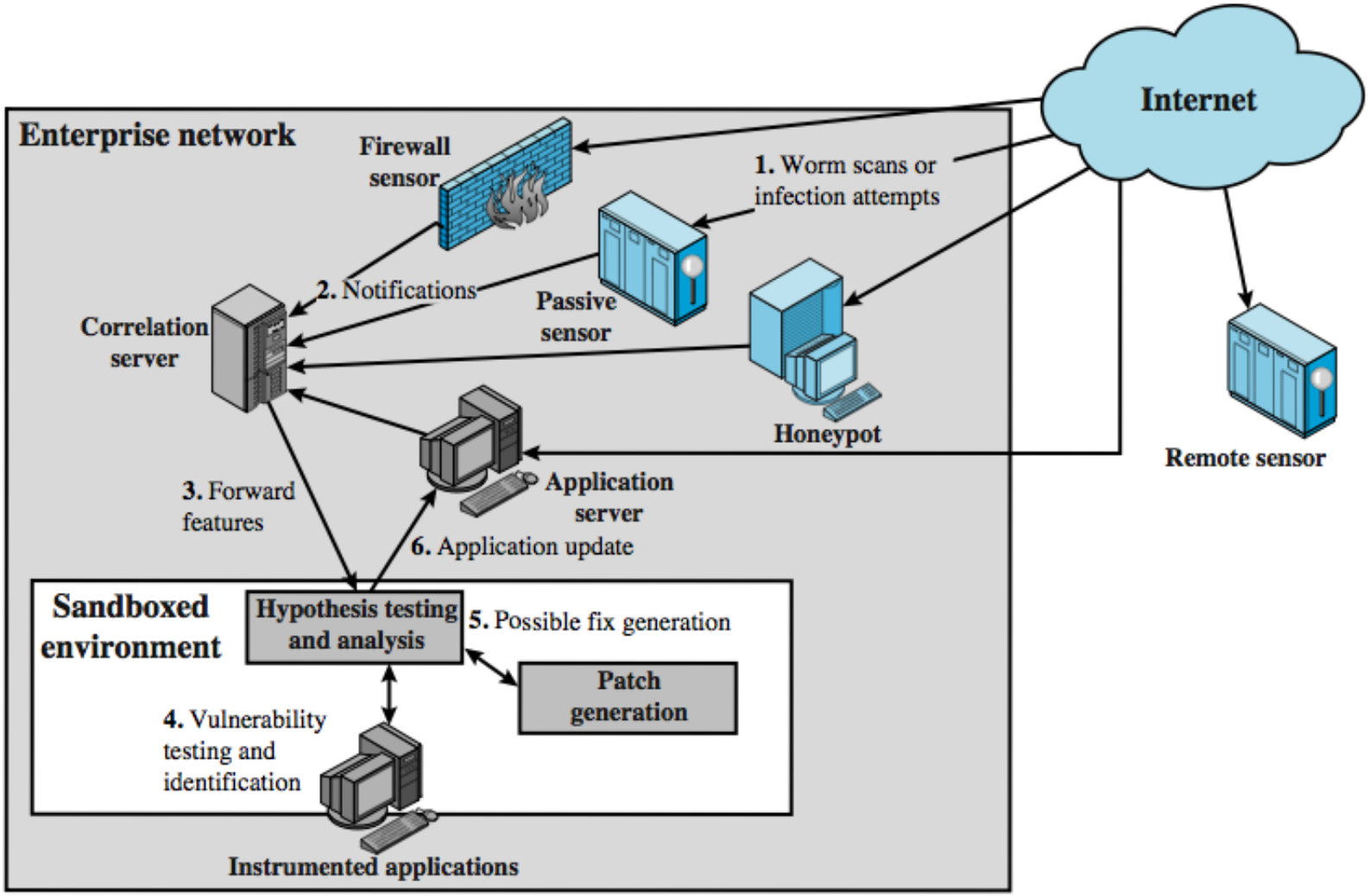
Worm Countermeasures

- Overlaps with anti-virus techniques
- Once worm on system A/V can detect
- Worms also cause significant net activity
- Worm defense approaches include:
  - Signature-based worm scan filtering
  - Filter-based worm containment
  - Payload-classification-based worm containment
  - Threshold random walk scan detection
  - Rate limiting and rate halting

Proactive Worm Containment



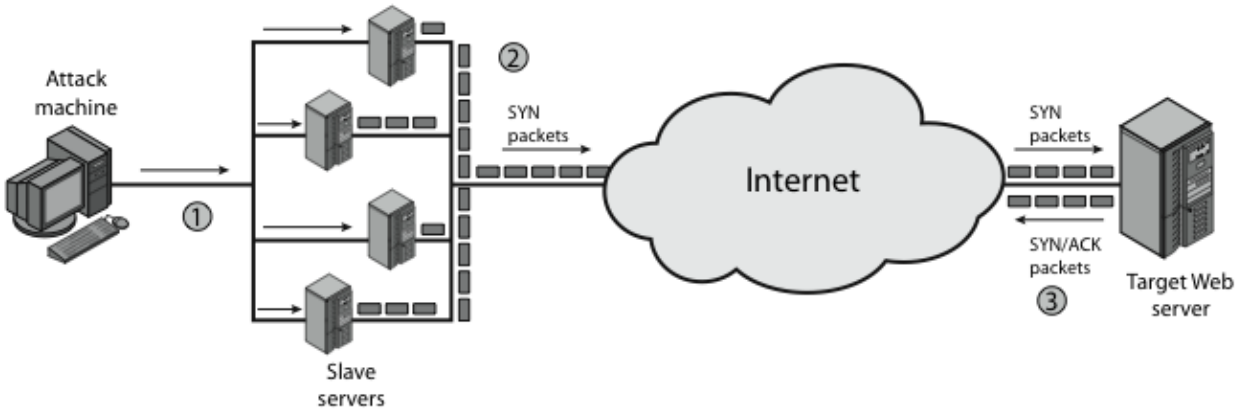
Network Based Worm Defense



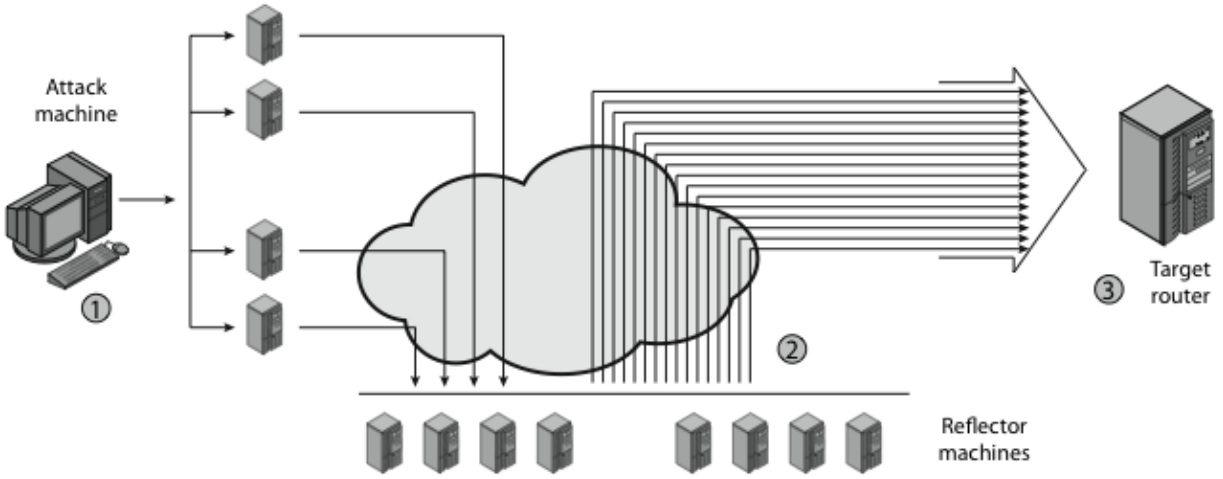
Distributed Denial of Service Attacks (DDoS)

- 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

DDoS (Cont)

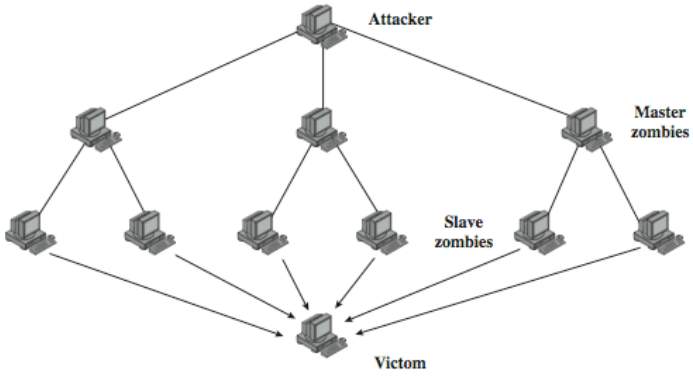


(a) Distributed SYN flood attack

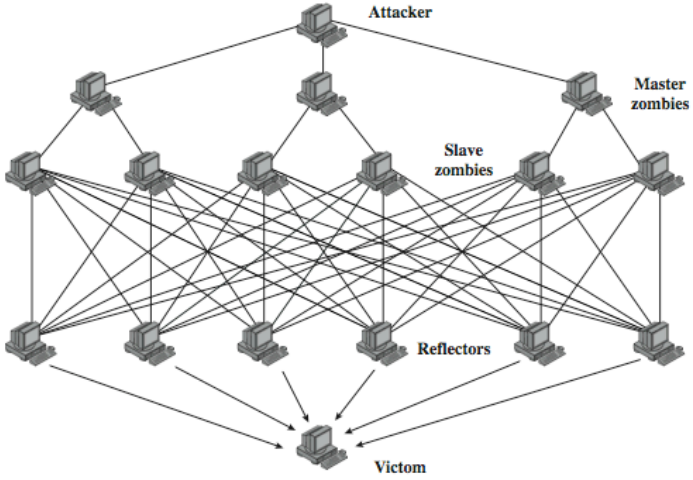


(a) Distributed ICMP attack

DDoS Flood Types



(a) Direct DDoS Attack



(b) Reflector DDoS Attack

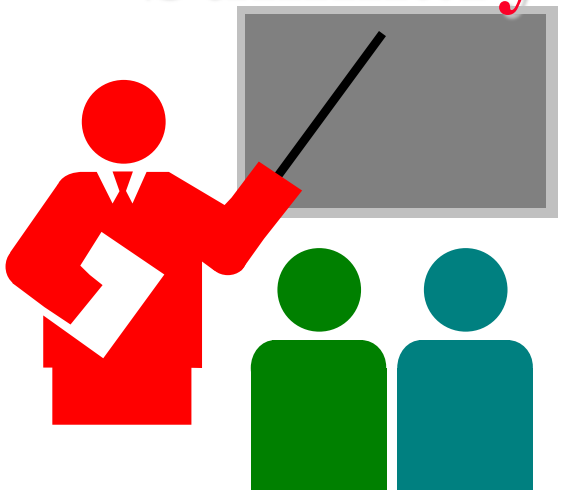
Constructing an Attack Network

- 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 defense:
  1. Attack prevention & preemption (before)
  2. Attack detection & filtering (during)
  3. Attack source traceback & ident (after)
- Huge range of attack possibilities
- Hence evolving countermeasures

Summary



- Malicious programs: trapdoor, logic bomb, trojan horse, zombie
- Viruses
- Worms
- Distributed denial of service attacks