Computer Security: Principles and Practice

Chapter 6 – Malicious Software

Malicious Software

- programs exploiting system vulnerabilities
- known as malicious software or malware
- Classification criteria
  - Independence level of the program
    - program fragments that need a host program (e.g. viruses, logic bombs, and backdoors)
    - independent self-contained programs
      - e.g. worms, bots
  - Replicating or not

Malicious Software (other classifications)

- Propagation
  - Infected content – e.g., viruses
  - Vulnerability exploit – e.g., worms
  - Social engineering – e.g., spam and trojans
- Payload action
  - System corruption
  - Attack agent – e.g., bots
  - Information theft – e.g., keyloggers and spyware
  - Stealthing – e.g., backdoors and rootkits

Malware Terminology

- Virus: attaches itself to a program and propagates copies of itself to other programs
- Worm: runs independently and propagates copies of itself to other programs
- Logic bomb: triggers action when condition occurs
- Trojan horse: contains unexpected additional functionality
- Backdoor (trapdoor): modification that allows unauthorized access to functionality
- Mobile code: can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics
- Auto-rooter Kit (virus generator): hacker tools used to break into new machines to generate new viruses automatically
- Spammer and Flooder programs: send large volumes of unwanted e-mail, or to attack systems with a large volumes of traffic to carry out a DoS attack
- Keyloggers: captures keystrokes on a compromised system
- Rootkit: set of hacker tools used after attacker has broken into a computer system and gained root-level access
- Zombie, bot: program on infected machine activated to launch attacks on other machines

Virus

Def: a program that can infect other programs by modifying them to include a, possibly evolved, version of itself

Fred Cohen 1983

Viruses

- piece of software that infects programs
  - modifying them to include a copy of the virus
  - so it executes 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

- components:
  - infection mechanism - enables replication
  - trigger - event that makes payload activate
  - payload - what it does, malicious or benign
- prepended / postpended / embedded
- the infected program, when invoked, will first execute the virus code and then execute the original code of the program
- can block initial infection (difficult)
- or propagation (with access controls)

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:

}
```

Propagation

Execution

Triggering

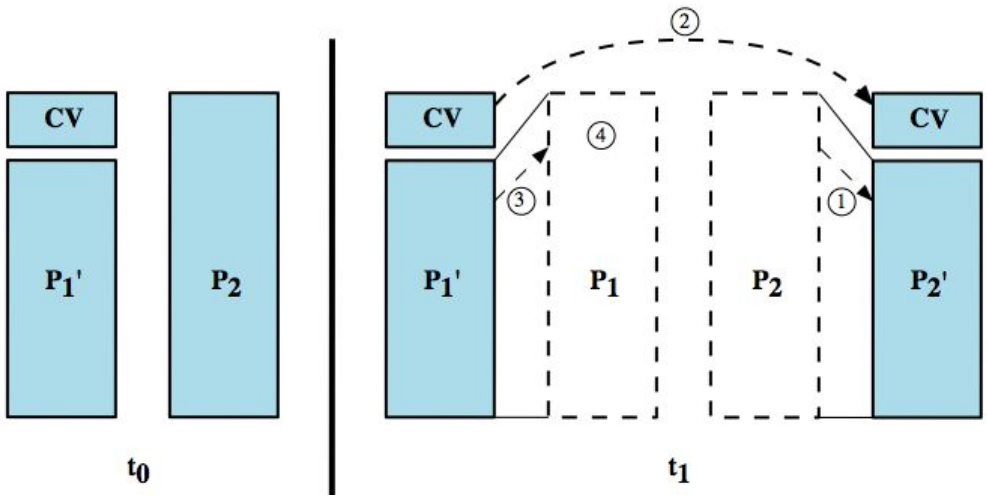
Compression Virus

```
    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
  - file infector
  - macro virus
- By concealing strategy
  - encrypted virus
  - stealth virus
  - polymorphic virus
  - metamorphic virus

Macro Virus

- became very common in mid-1990s since
  - platform independent
  - infect documents
  - easily spread
- exploit macro capability of office apps
  - executable program embedded in office doc
  - often a form of Basic
- more recent releases include protection
- recognized by many anti-virus programs

E-Mail Viruses/Worms

- more recent development
- e.g. 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 version can be activated by opening an email that contains the virus rather than opening an attachment
- hence much faster propagation

Malware Countermeasures

- prevention - ideal solution but difficult
  - Policy / awareness / vulnerability mitigation
- realistically need:
  - detection
  - identification
  - removal
- if detect but can't identify or remove, must discard and replace infected program
- Where to detect: host / perimeter / distributed

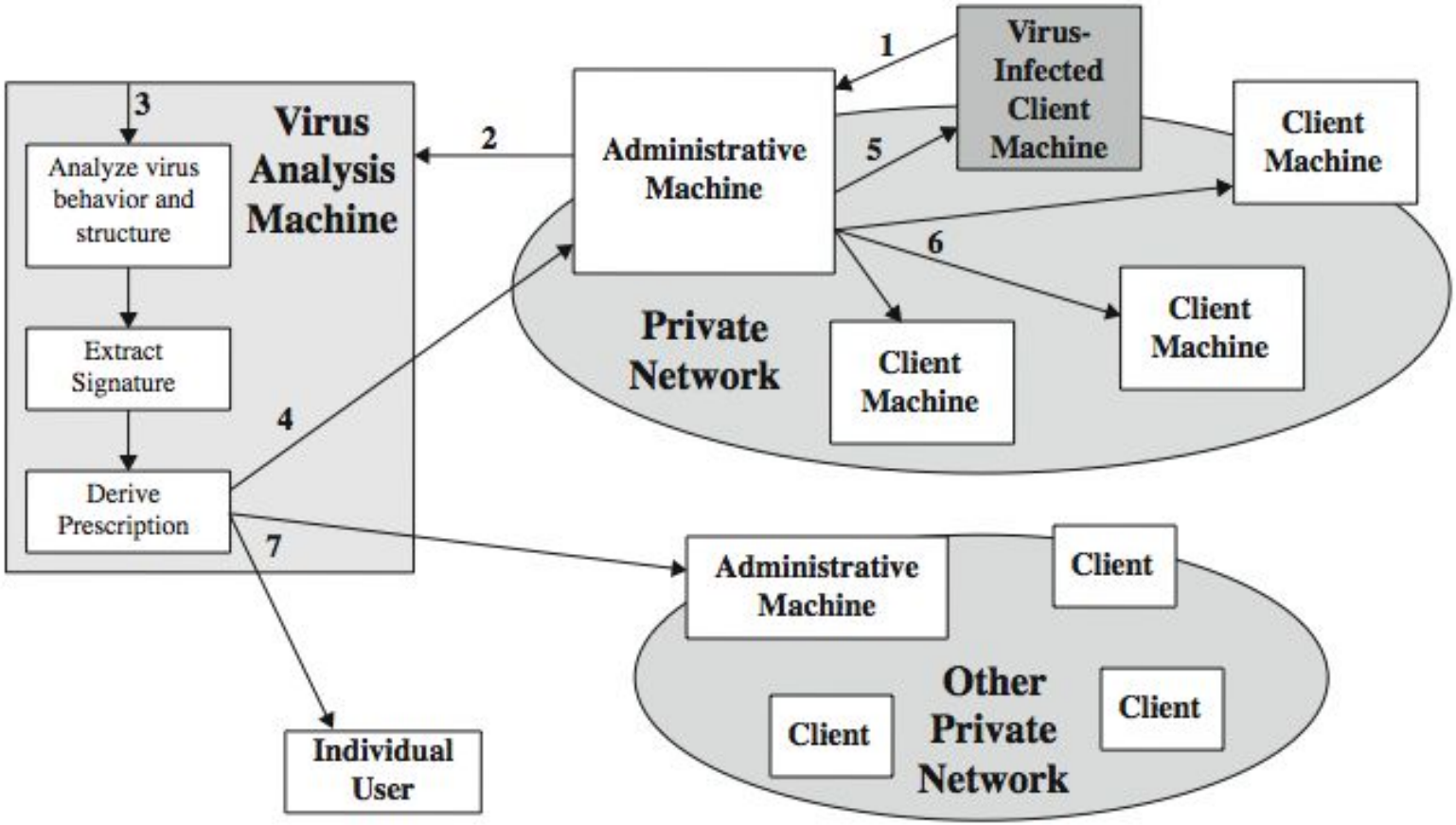
Anti-Virus Evolution

- virus & antivirus tech have both evolved
- early viruses simple code, easily removed
- as become more complex, so must the countermeasures
- generations
  - first - signature scanners
  - second - heuristics
  - third - identify actions
  - fourth - combination packages

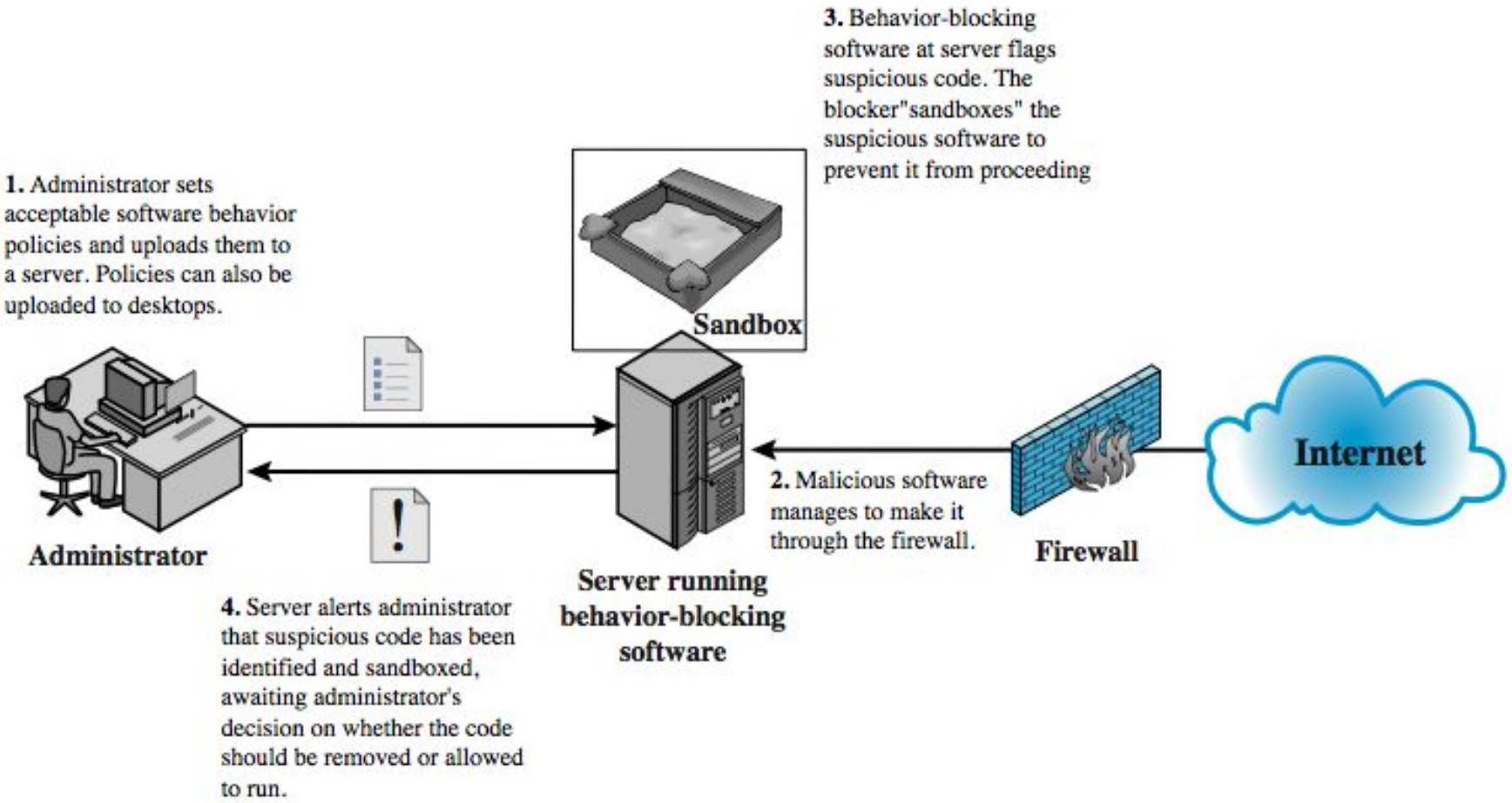
Generic Decryption

- 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



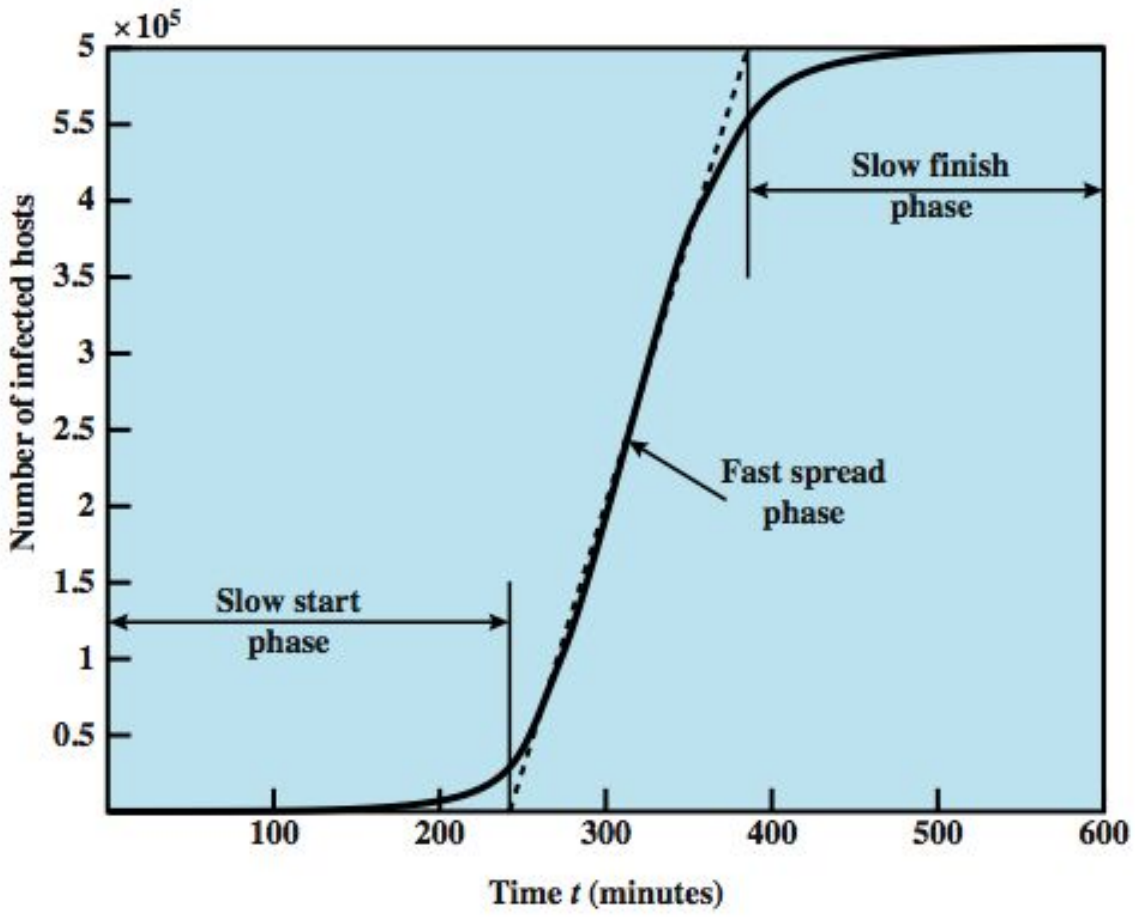
Worms

- 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 best know 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



Recent Worm Attacks

- Code Red
  - July 2001 exploiting MS IIS bug
  - probes random IP address, does DDoS attack
  - consumes significant net capacity when active
- Code Red II variant includes backdoor
- SQL Slammer
  - early 2003, attacks MS SQL Server
  - compact and very rapid spread
- Mydoom
  - mass-mailing e-mail worm that appeared in 2004
  - installed remote access backdoor in infected systems
- Stuxnet – target (Iranian/nuclear) industrial control system

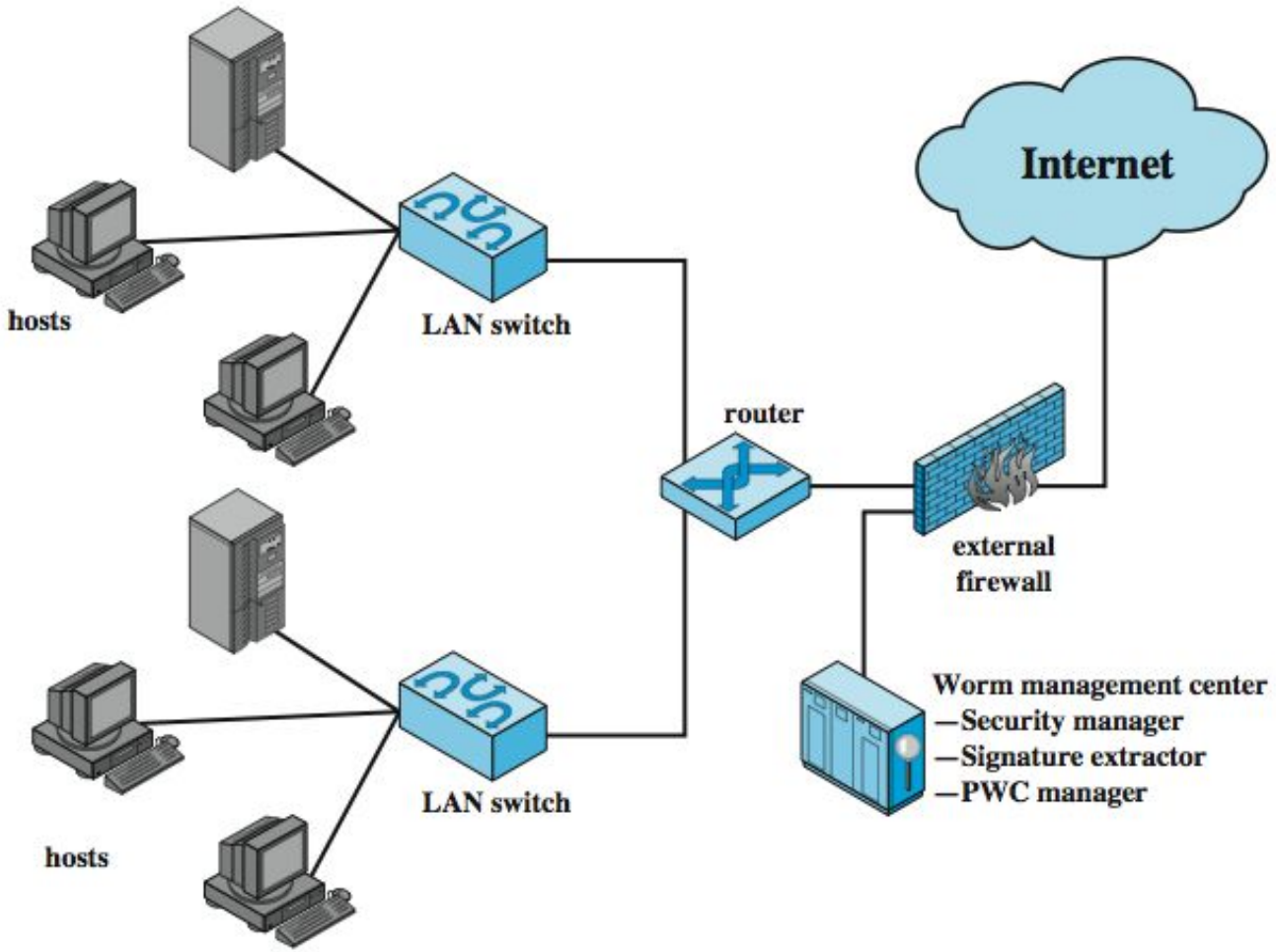
Worm Technology

- Multiplatform: not limited to Windows machines
- Multi-exploit: penetrate systems in a variety of ways
- ultrafast spreading: conduct a prior Internet scan to accumulate Internet addresses of vulnerable machines
- Polymorphic: Each copy of the worm has new code generated on the fly
- Metamorphic: have a repertoire of behavior patterns that are unleashed at different stages of propagation
- transport vehicles: ideal for spreading other distributed attack tools
- zero-day exploit: exploit an unknown vulnerability

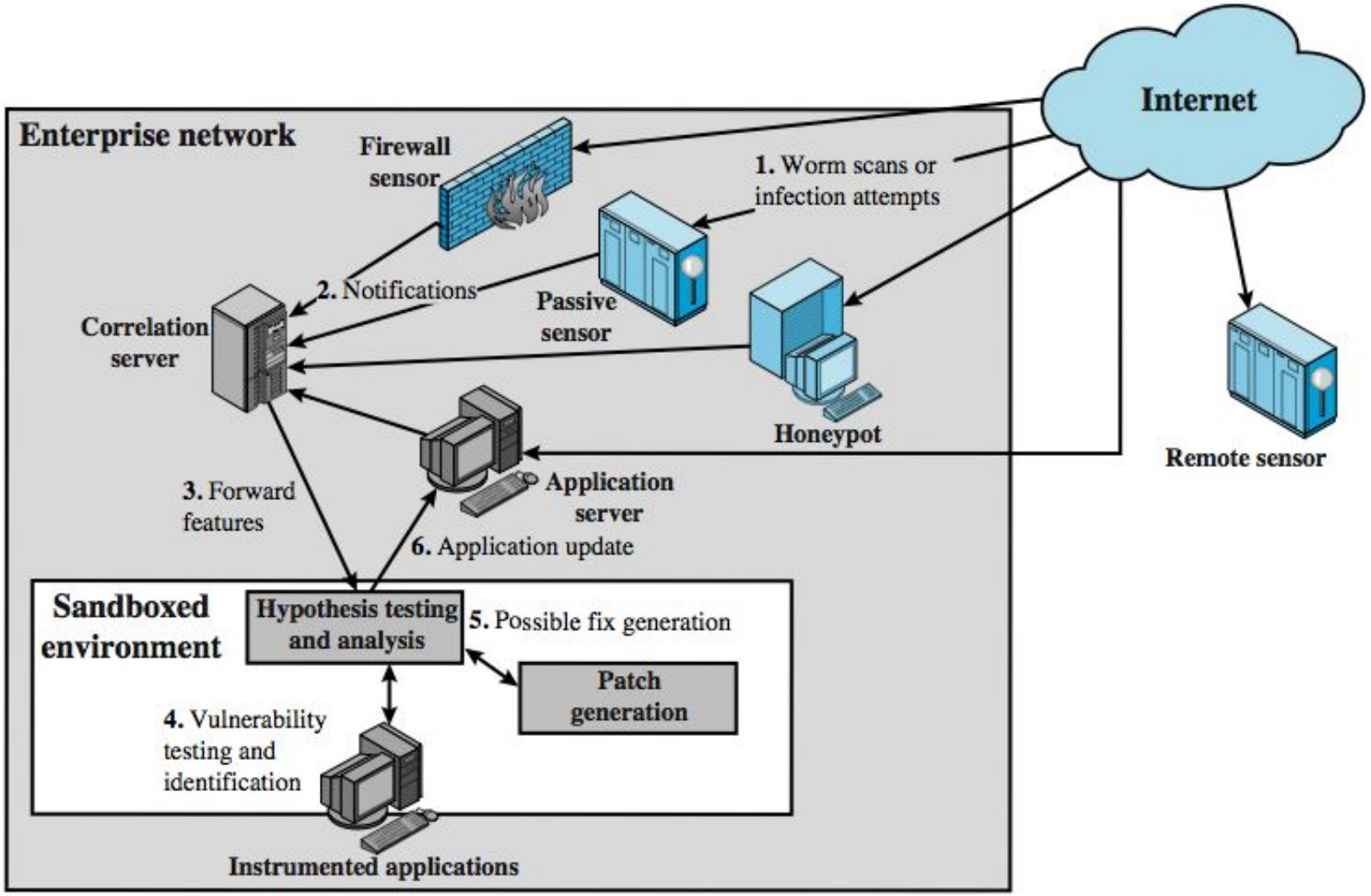
Worm Countermeasures

- 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



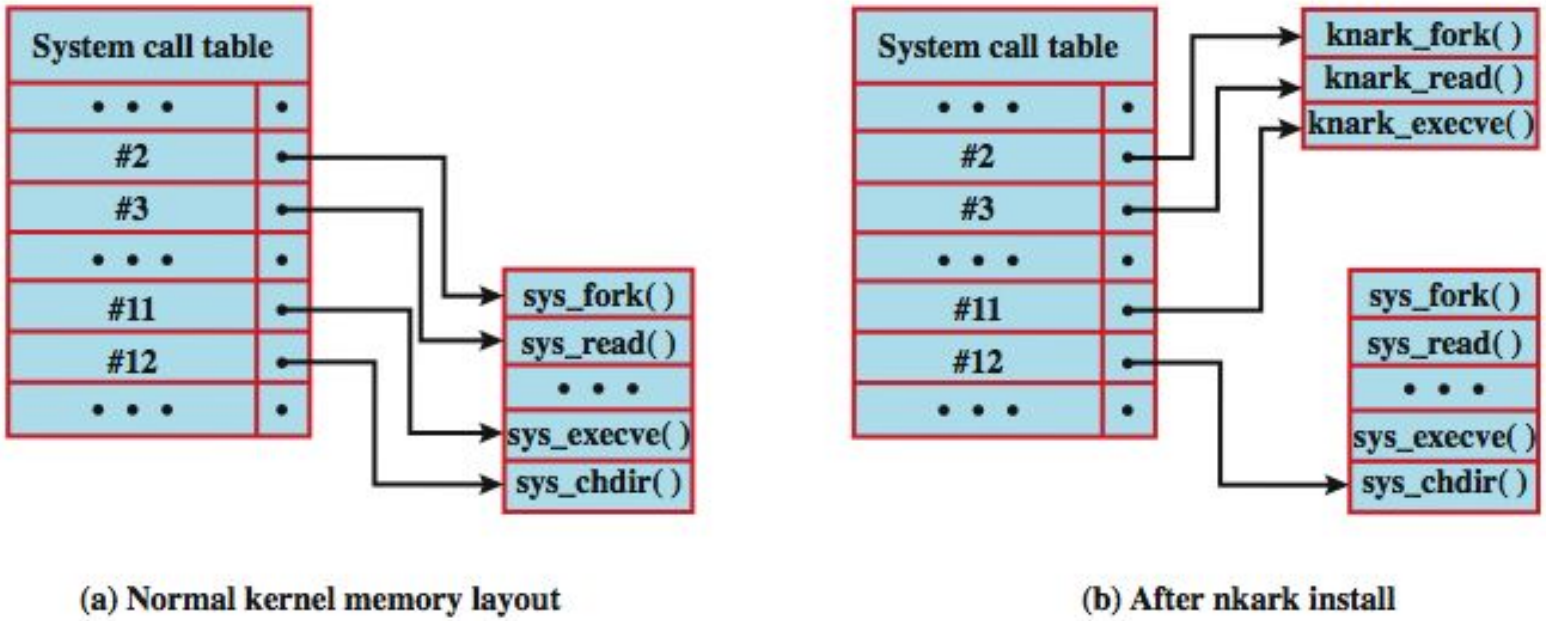
Bots

- program taking over other computers
- to launch hard to trace attacks
- if coordinated form a botnet
- characteristics:
  - remote control facility
    - via IRC/HTTP etc
  - spreading mechanism
    - attack software, vulnerability, scanning strategy
- various counter-measures applicable

Rootkits

- set of programs installed for admin access
- malicious and stealthy changes to host O/S
- may hide its existence
  - subverting report mechanisms on processes, files, registry entries etc
- may be:
  - persistent or memory-based
  - user or kernel mode
- installed by user via trojan or intruder on system
- range of countermeasures needed

Rootkit System Table Mods



(a) Normal kernel memory layout

(b) After nkark install