

# CS 356 – Lecture 9

## Malicious Code

Spring 2013

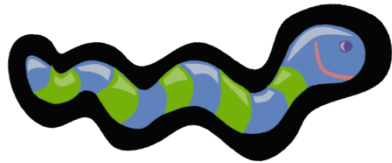
# Review

- Chapter 1: Basic Concepts and Terminology
  - Integrity, Confidentiality, Availability, Authentication, and Accountability
  - Types of threats: active vs. passive, insider/outsider
- Chapter 2: Basic Cryptographic Tools
  - Symmetric key encryption and secure hashing
  - Public key cryptography and Random Numbers
- Chapter 3 – User Authentication
  - Passwords, Checking passwords and Biometrics
- Chapter 4 – Access Control Lists
  - Concepts and Discretionary Access Control
  - Role Based Access Control (RBAC)
- Chapter 5 – Database Security (skipped)
- Chapter 6 – Malicious Software
  - Virus Malware

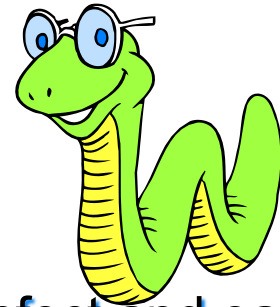


# Chapter 6

## Malicious Software



# Worms




- program that actively seeks out more machines to infect and each infected machine serves as an automated launching pad for attacks on other machines
- exploits software vulnerabilities in client or server programs
- can use network connections to spread from system to system
- spreads through shared media (USB drives, CD, DVD data disks)
- e-mail worms spread in macro or script code included in attachments and instant messenger file transfers
- upon activation the worm may replicate and propagate again
- usually carries some form of payload
- first known implementation was done in Xerox Palo Alto Labs in the early 1980s





# Worm Replication



**electronic mail or instant messenger facility**

- worm e-mails a copy of itself to other systems
- sends itself as an attachment via an instant message service

**file sharing**

- creates a copy of itself or infects a file as a virus on removable media

**remote execution capability**

- 
- worm executes a copy of itself on another system

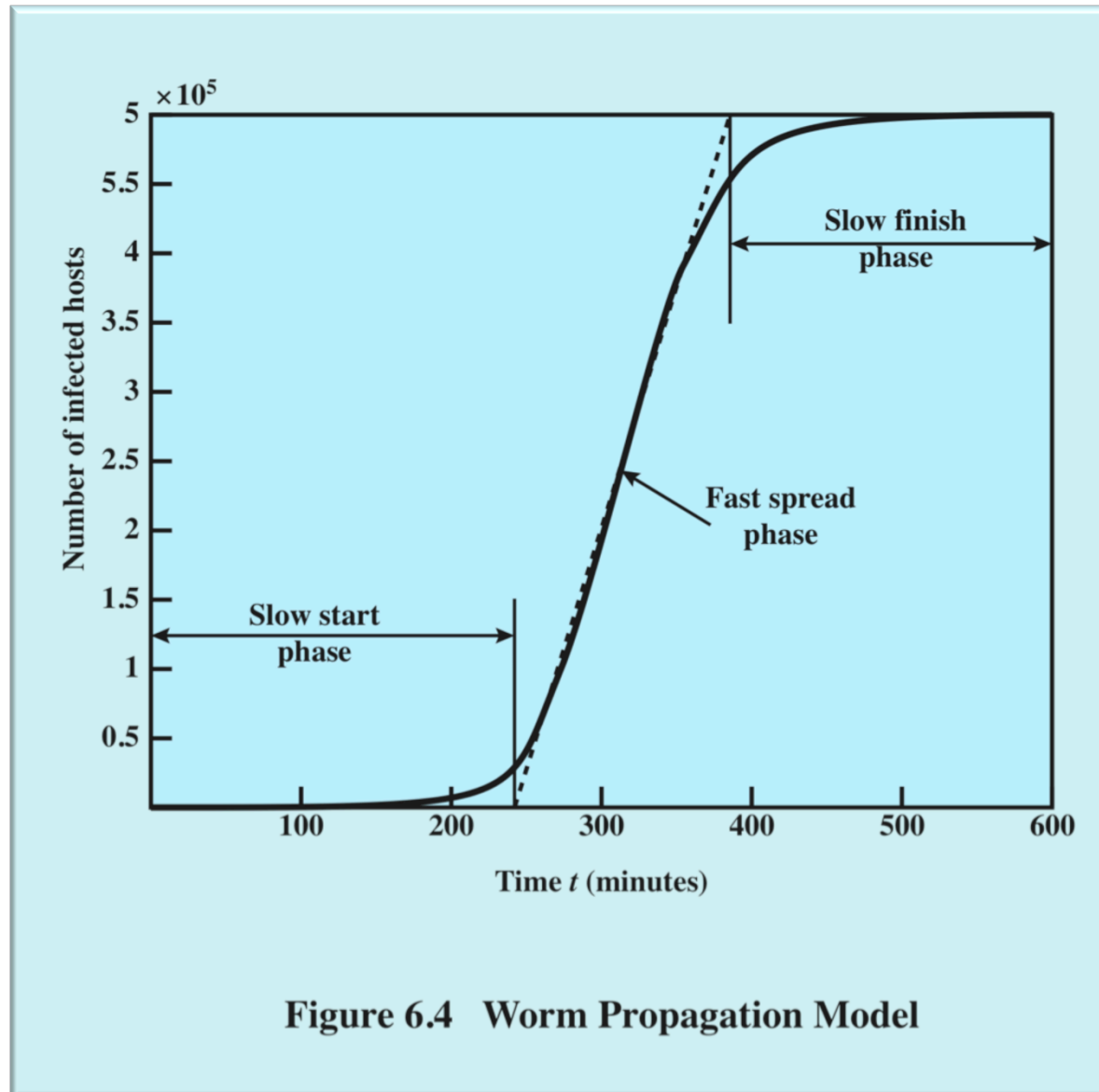
**remote file access or transfer capability**

- worm uses a remote file access or transfer service to copy itself from one system to the other

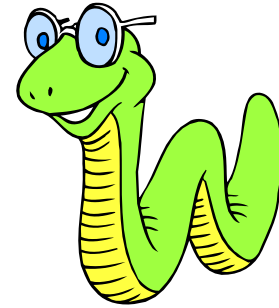
**remote login capability**

- 
- worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other

# Worm Propagation Model



# Morris Worm

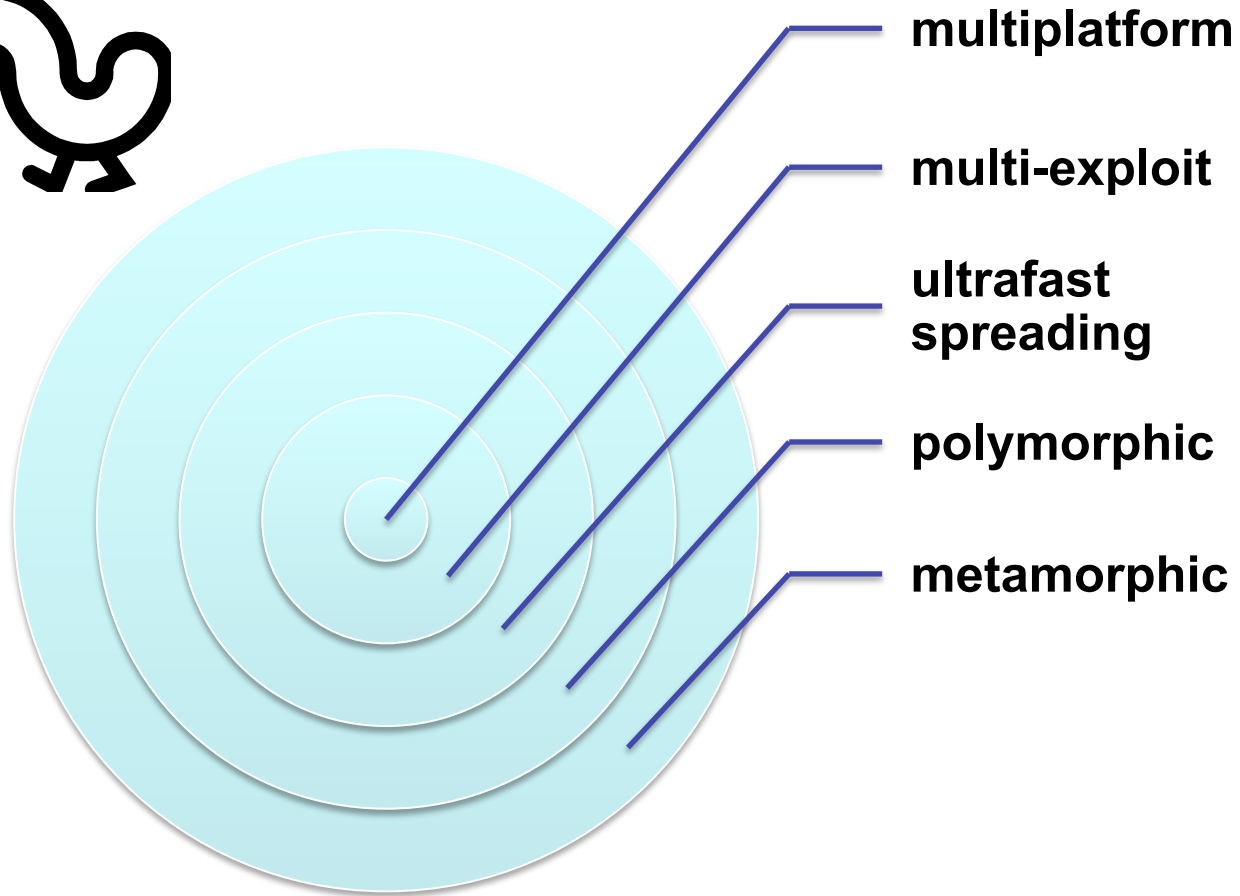
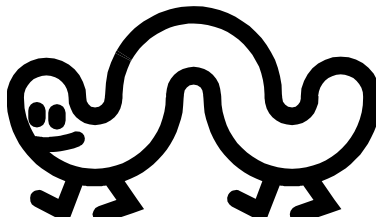


- earliest significant worm infection
- released by Robert Morris in 1988
- designed to spread on UNIX systems
  - attempted to crack local password file to use login/password to logon to other systems
  - exploited a bug in the finger protocol which reports the whereabouts of a remote user
  - exploited a trapdoor in the debug option of the remote process that receives and sends mail
- successful attacks achieved communication with the operating system command interpreter
  - sent interpreter a bootstrap program to copy worm over

# Recent Worm Attacks

<b>Melissa</b>	<b>1998</b>	<b>e-mail worm first to include virus, worm and Trojan in one package</b>
<b>Code Red</b>	<b>July 2001</b>	<b>exploited Microsoft IIS bug probes random IP addresses consumes significant Internet capacity when active</b>
<b>Code Red II</b>	<b>August 2001</b>	<b>also targeted Microsoft IIS installs a backdoor for access</b>
<b>Nimda</b>	<b>September 2001</b>	<b>had worm, virus and mobile code characteristics spread using e-mail, Windows shares, Web servers, Web clients, backdoors</b>
<b>SQL Slammer</b>	<b>Early 2003</b>	<b>exploited a buffer overflow vulnerability in SQL server compact and spread rapidly</b>
<b>Sobig.F</b>	<b>Late 2003</b>	<b>exploited open proxy servers to turn infected machines into spam engines</b>
<b>Mydoom</b>	<b>2004</b>	<b>mass-mailing e-mail worm installed a backdoor in infected machines</b>
<b>Warezov</b>	<b>2006</b>	<b>creates executables in system directories sends itself as an e-mail attachment can disable security related products</b>
<b>Conficker (Downadup)</b>	<b>November 2008</b>	<b>exploits a Windows buffer overflow vulnerability most widespread infection since SQL Slammer</b>
<b>Stuxnet</b>	<b>2010</b>	<b>restricted rate of spread to reduce chance of detection targeted industrial control systems</b>

# Worm Technology



# Mobile Code

- programs that can be shipped unchanged to a variety of platforms
- transmitted from a remote system to a local system and then executed on the local system
- often acts as a mechanism for a virus, worm, or Trojan horse
- takes advantage of vulnerabilities to perform its own exploits
- popular vehicles include Java applets, ActiveX, JavaScript and VBScript

# Mobile Phone Worms

- first discovery was Cabir worm in 2004
- then Lasco and CommWarrior in 2005
- communicate through Bluetooth wireless connections or MMS
- target is the smartphone
- can completely disable the phone, delete data on the phone, or force the device to send costly messages
- CommWarrior replicates by means of Bluetooth to other phones, sends itself as an MMS file to contacts and as an auto reply to incoming text messages

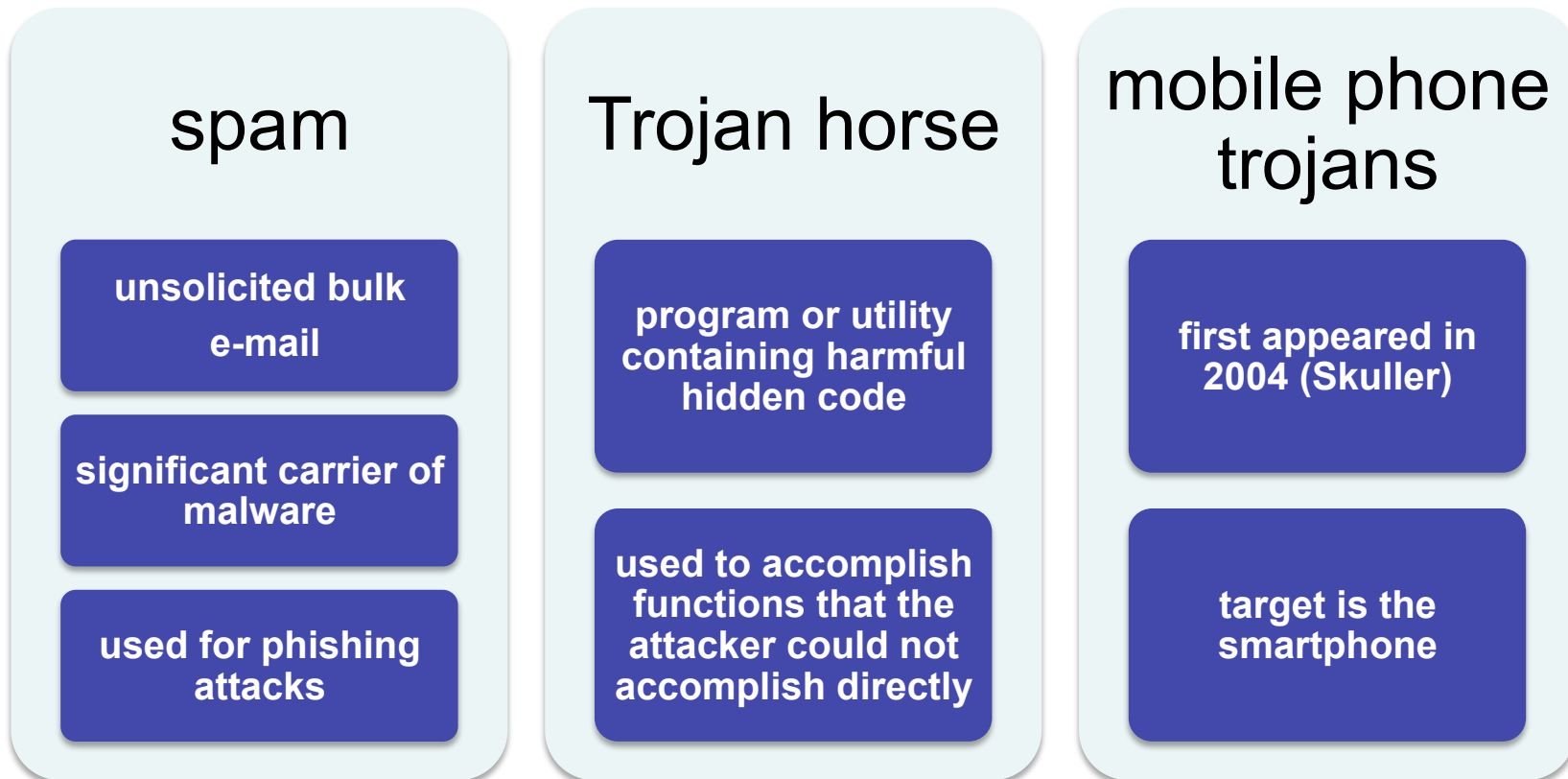
# Drive-By-Downloads

- exploits browser vulnerabilities to download and install malware on the system when the user views a Web page controlled by the attacker
- in most cases does not actively propagate
- spreads when users visit the malicious Web page



# Social Engineering

- “tricking” users to assist in the compromise of their own systems



# Payload System Corruption

- data destruction
  - Chernobyl virus
    - first seen in 1998
    - Windows 95 and 98 virus
    - infects executable files and corrupts the entire file system when a trigger date is reached
  - Klez
    - mass mailing worm infecting Windows 95 to XP systems
    - on trigger date causes files on the hard drive to become empty
  - ransomware
    - encrypts the user's data and demands payment in order to access the key needed to recover the information
    - PC Cyborg Trojan (1989)
    - Gpcode Trojan (2006)



# Payload System Corruption

- **real-world damage**
  - **causes damage to physical equipment**
    - Chernobyl virus rewrites BIOS code
  - **Stuxnet worm**
    - targets specific industrial control system software
  - **there are concerns about using sophisticated targeted malware for industrial sabotage**
- **logic bomb**
  - **code embedded in the malware that is set to “explode” when certain conditions are met**

# Payload – Attack Agents Bots

- takes over another Internet attached computer and uses that computer to launch or manage attacks
- *botnet* - collection of bots capable of acting in a coordinated manner
- uses:
  - distributed denial-of-service (DDoS) attacks
  - spamming
  - sniffing traffic
  - keylogging
  - spreading new malware
  - installing advertisement add-ons and browser helper objects (BHOs)
  - attacking IRC chat networks
  - manipulating online polls/games



# Remote Control Facility

- distinguishes a bot from a worm
  - worm propagates itself and activates itself
  - bot is initially controlled from some central facility
- typical means of implementing the remote control facility is on an IRC server
  - bots join a specific channel on this server and treat incoming messages as commands
  - more recent botnets use covert communication channels via protocols such as HTTP
  - distributed control mechanisms use peer-to-peer protocols to avoid a single point of failure



# Payload – Information Theft

## Keyloggers and Spyware

### keylogger

- captures keystrokes to allow attacker to monitor sensitive information
- typically uses some form of filtering mechanism that only returns information close to keywords (“login”, “password”)

### spyware

- subverts the compromised machine to allow monitoring of a wide range of activity on the system
  - monitoring history and content of browsing activity
  - redirecting certain Web page requests to fake sites
  - dynamically modifying data exchanged between the browser and certain Web sites of interest

# Payload – Information Theft Phishing



- exploits social engineering to leverage the user's trust by masquerading as communication from a trusted source
  - include a URL in a spam e-mail that links to a fake Web site that mimics the login page of a banking, gaming, or similar site
  - suggests that urgent action is required by the user to authenticate their account
  - attacker exploits the account using the captured credentials
- spear-phishing
  - recipients are carefully researched by the attacker
  - e-mail is crafted to specifically suit its recipient, often quoting a range of information to convince them of its authenticity



# Payload – Stealthing Backdoor

- also known as a *trapdoor*
- secret entry point into a program allowing the attacker to gain access and bypass the security access procedures
- *maintenance hook* is a backdoor used by programmers to debug and test programs
- difficult to implement operating system controls for backdoors in applications



# Payload – Stealthing Rootkit

- set of hidden programs installed on a system to maintain covert access to that system
- hides by subverting the mechanisms that monitor and report on the processes, files, and registries on a computer
- gives administrator (or root) privileges to attacker
  - can add or change programs and files, monitor processes, send and receive network traffic, and get backdoor access on demand

# Rootkit Classification

## Characteristics

**persistent**

**memory  
based**

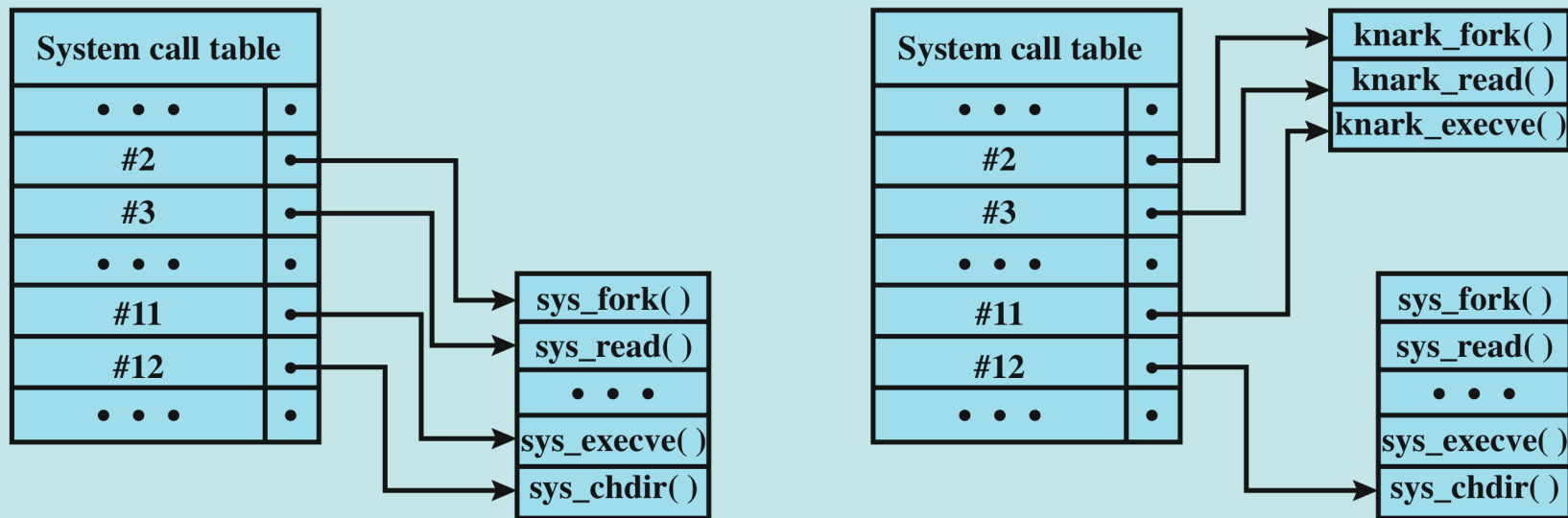
**user mode**

**kernel mode**

**virtual  
machine  
based**

**external  
mode**

# System Call Table Modification



(a) Normal kernel memory layout

(b) After nkark install

Figure 6.5 System Call Table Modification by Rootkit (based on [LEVI06])

# Malware Countermeasure Approaches

- ideal solution to the threat of malware is prevention

four main elements of prevention:

- policy
- awareness
- vulnerability mitigation
- threat mitigation

- if prevention fails, technical mechanisms can be used to support the following threat mitigation options:
  - detection
  - identification
  - removal

# Generations of Anti-Virus Software

## first generation: simple scanners

- requires a malware signature to identify the malware
- limited to the detection of known malware

## second generation: heuristic scanners

- uses heuristic rules to search for probable malware instances
- another approach is integrity checking

## third generation: activity traps

- memory-resident programs that identify malware by its actions rather than its structure in an infected program

## fourth generation: full-featured protection

- packages consisting of a variety of anti-virus techniques used in conjunction
- include scanning and activity trap components and access control capability

# Generic Decryption (GD)

- enables the anti-virus program to easily detect complex polymorphic viruses and other malware while maintaining fast scanning speeds
- executable files are run through a GD scanner which contains the following elements:
  - CPU emulator
  - virus signature scanner
  - emulation control module
- the most difficult design issue with a GD scanner is to determine how long to run each interpretation

# Host-Based Behavior-Blocking Software

- integrates with the operating system of a host computer and monitors program behavior in real time for malicious action
  - blocks potentially malicious actions before they have a chance to affect the system
  - blocks software in real time so it has an advantage over anti-virus detection techniques such as fingerprinting or heuristics

## limitations

- because 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

- anti-virus software typically included in e-mail and Web proxy services running on an organization's firewall and IDS
- may also be included in the traffic analysis component of an IDS
- may include intrusion prevention measures, blocking the flow of any suspicious traffic
- approach is limited to scanning malware

located at the border between the enterprise network and the Internet

one technique is to look for incoming traffic to unused local IP addresses

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

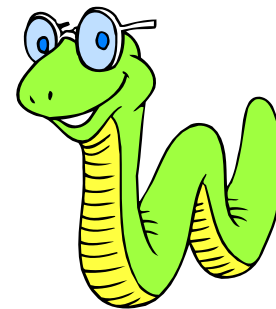
monitors outgoing traffic for signs of scanning or other suspicious behavior

- two types of monitoring software



# Worm Countermeasures

- considerable overlap in techniques for dealing with viruses and worms
- once a worm is resident on a machine anti-virus software can be used to detect and possibly remove it
- perimeter network activity and usage monitoring can form the basis of a worm defense
- worm defense approaches include:
  - signature-based worm scan filtering
  - filter-based worm containment
  - payload-classification-based worm containment
  - threshold random walk (TRW) scan detection
  - rate limiting
  - rate halting



# Digital Immune System

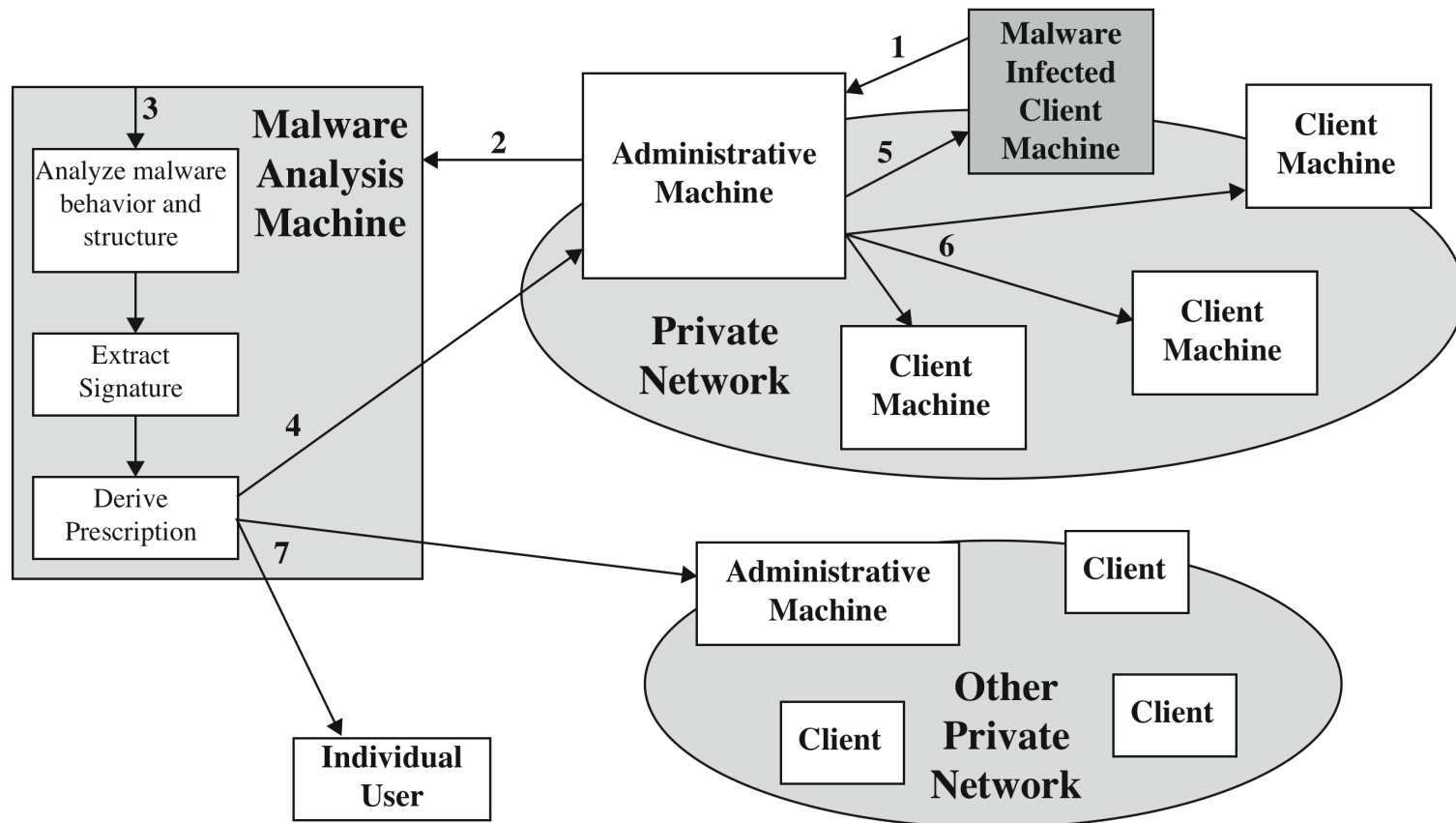


Figure 6.6 Digital Immune System

# Worm Countermeasure Architecture

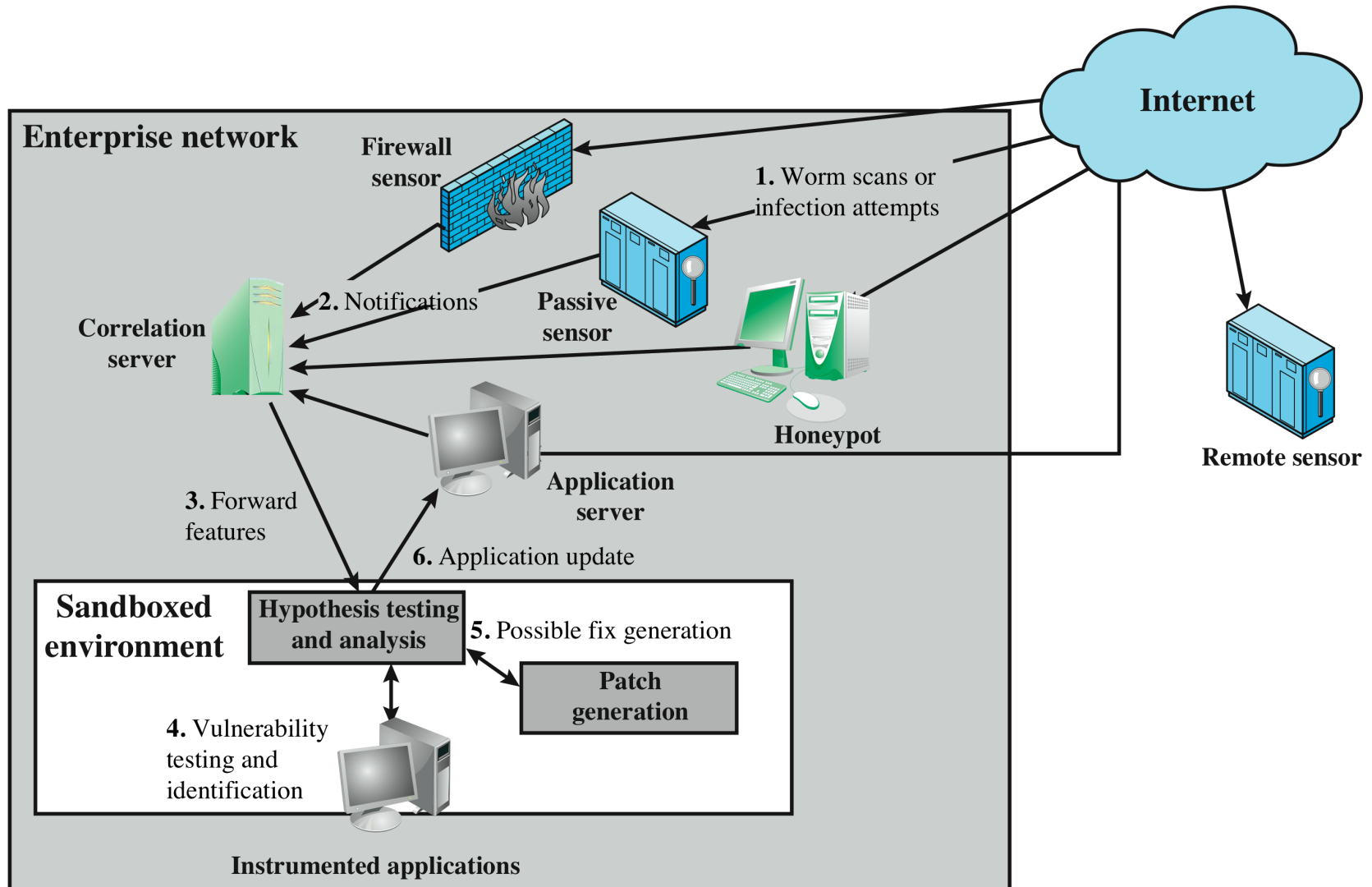


Figure 6.7 Placement of Worm Monitors (based on [SIDI05])



# Summary

- types of malicious software (malware)
- terminology for malicious software
- viruses – infected content
  - infection mechanism, trigger, payload
  - dormant, propagation, triggering, and execution phases
  - boot sector infector, file infector, macro virus, and multipartite virus
  - encrypted, stealth, polymorphic, and metamorphic viruses
- worms – vulnerability exploit
  - replicates via remote systems
  - e-mail, file sharing, remote execution, remote file access, remote login capability
  - scanning/fingerprinting
- spam e-mail/trojans – social engineering
- payload – system corruption
  - data destruction, real world damage
  - ransomware, logic bomb
- payload – attack agent
  - bots
  - remote control facility
- payload – information theft
  - credential theft, keyloggers, spyware
  - phishing, identity theft
- payload – stealthing
  - backdoor/trapdoor
  - rootkit
  - kernel mode rootkits
  - virtual machine/external rootkits
- countermeasures
  - prevention
  - detection, identification, removal
  - host based scanners/behavior blocking software
  - digital immune system



# What's Next

- Read Chapter 1, 2, 3, 4, (skip 5), and 6
  - Chap 1: Focus on big picture and recurring concepts
  - Chap 2: Identify cryptographic tools and properties
  - Chap 3: How can you authenticate a user?
  - Chap 4: Access Control
  - Chap 6: Intrusion Detection
- Homework Posted on Course Website
  - Due Tuesday
- Project 1 Due Thursday
- Next Lecture Topics From Chapter 6
  - Worms, Bots, and Malware