Cryptography and Network Security II

Second Course

Lecture 4: Malicious Software

Malware

“A program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim’s data, applications, or operating system or otherwise annoying or disrupting the victim.”

Malicious software

- Programs exploiting system vulnerabilities
- Known as malicious software or malware
  - 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
- Sophisticated threat to computer systems

Malware Terminology

- Virus: *attaches itself to a program*
- Worm: *propagates copies of itself to other computers*
- Logic bomb: *“explodes” when a condition occurs*
- Trojan horse: *fakes/contains additional functionality*
- Backdoor (trapdoor): *allows unauthorized access to functionality*
- Mobile code: *moves unchanged to heterogeneous platforms*
- Auto-rooter Kit (virus generator): *malicious code (virus) generators*
- Spammer and flooder programs: *large volume of unwanted “pkts”*
- Keyloggers: *capture keystrokes*
- Rootkit: *sophisticated hacker tools to gain root-level access*
- Zombie: *software on infected computers that launch attack on others (aka bot)*

Some terms

- Payload: actions of the malware
- Crimeware: kits for building malware; include propagation and payload mechanisms
  - Zeus, Sakura, Blackhole, Phoenix
- APT (advanced persistent threats)
  - Advanced: sophisticated
  - Persistent: attack over an extended period of time
  - Threat: selected targets (capable, well-funded attackers)

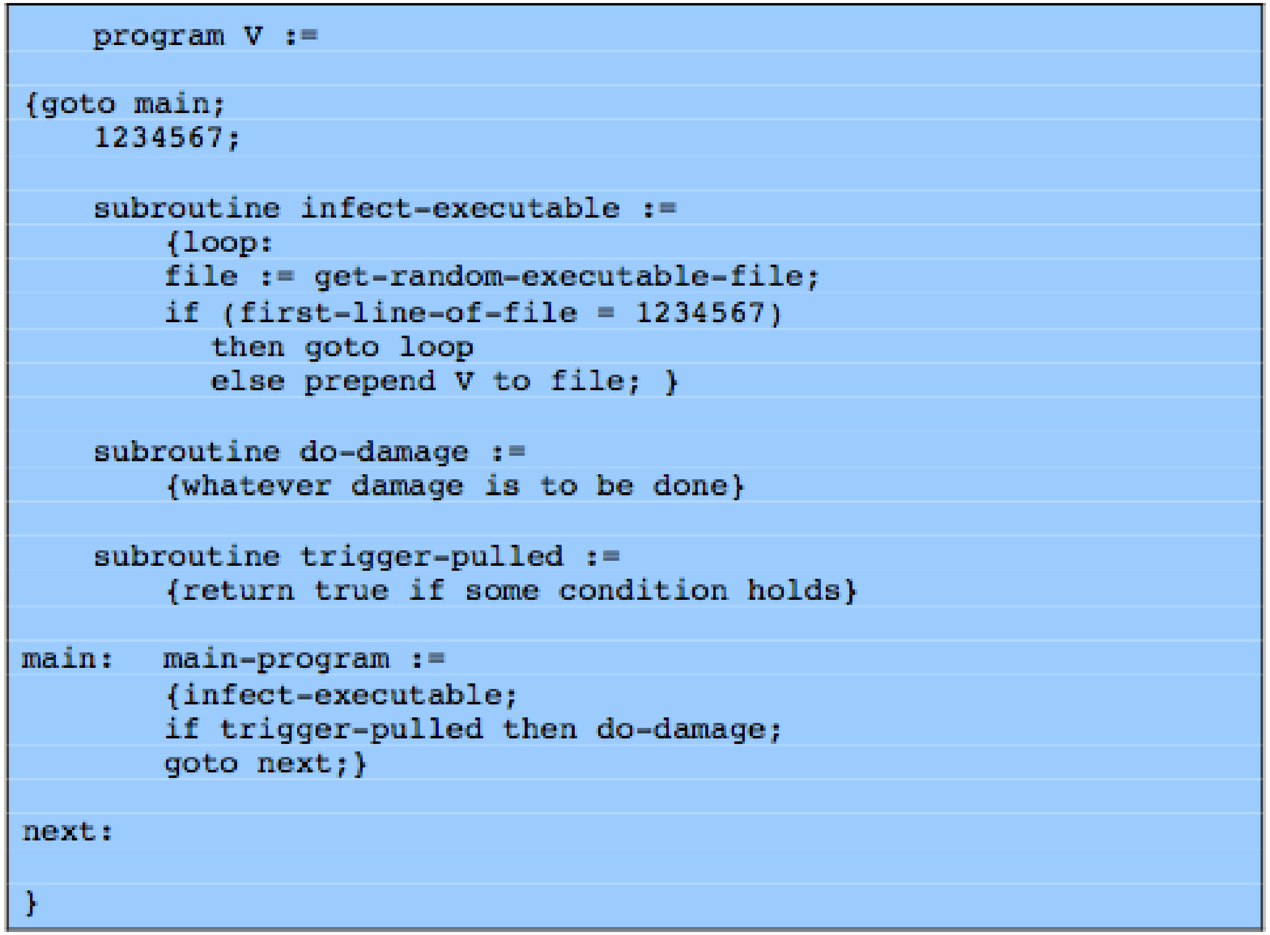
Viruses

- 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: *idle*
  - propagation: *copies itself to other program*
  - triggering: *activated to perform functions*
  - execution: *the function is performed*

Virus structure

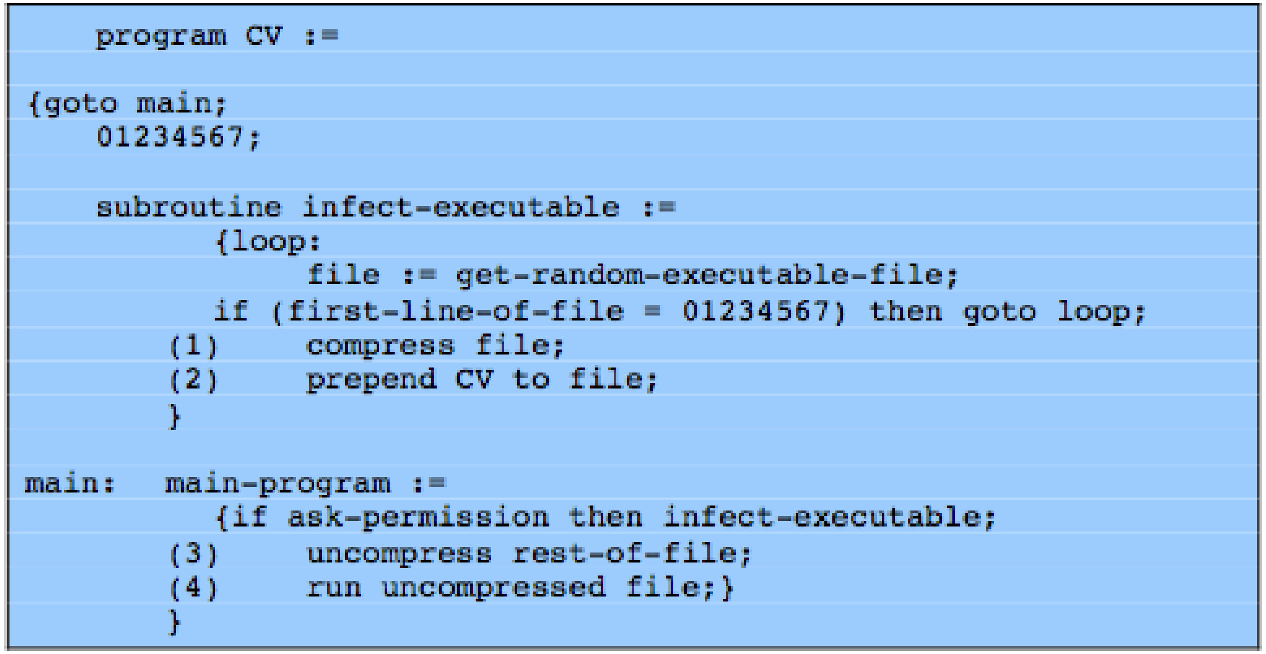
- Components:
  - infection mechanism: enables replication
  - trigger: event that makes payload activate
  - payload: what it does, malicious or benign
- Prepended/postpended/embedded
- When infected program invoked, executes virus code then original program code
- Can block initial infection (difficult) or propagation (with access controls)

Virus structure



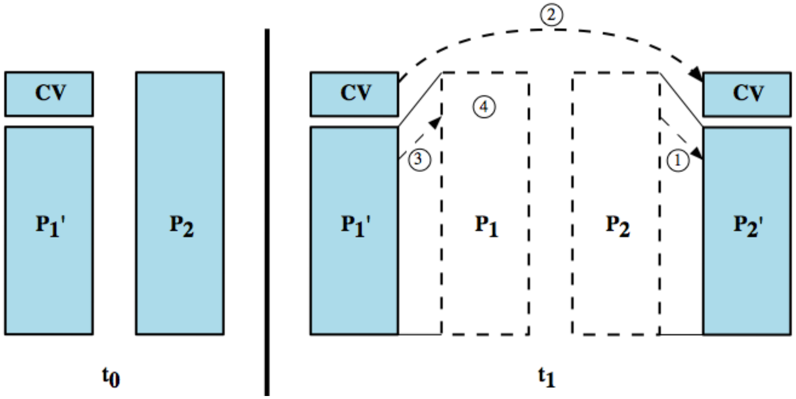
```
    program V :=

{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:

}
```

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;}
        }
```

P1 is infected



Virus classification

- By target
  - boot sector: *infect a master boot record*
  - file infector: *infects executable OS files*
  - macro virus: *infects files to be used by an app*
  - multipartite: infects multiple ways
- By concealment
  - encrypted virus: *encrypted; key stored in virus*
  - stealth virus: *hides itself (e.g., compression)*
  - polymorphic virus: *recreates with diff “signature”*
  - metamorphic virus: *recreates with diff signature and behavior*

Macro and scripting viruses

- 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

- More recent development
- 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

Virus countermeasures

- Prevention: ideal solution but difficult
- Realistically need:
  - detection: determine what occurred
  - identification: identify the specific virus
  - removal: remove all traces
- If detected but can’t identify or remove, must discard and replace infected program

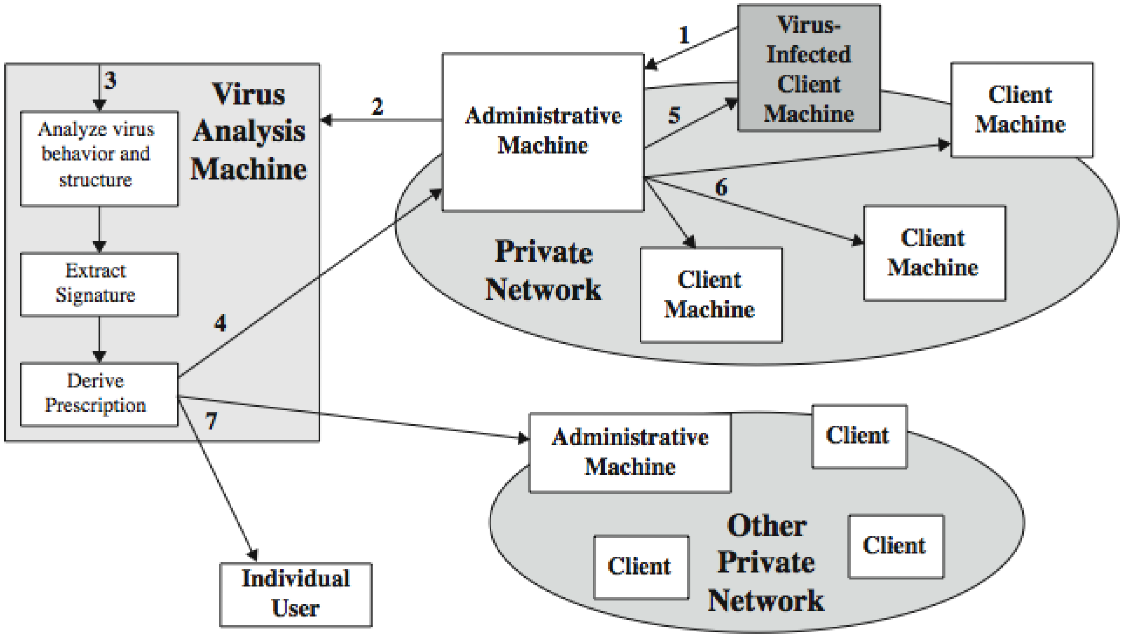
Anti-virus evolution

- Virus & antivirus tech have both evolved
- Early viruses simple code, easily removed
- As viruses become more complex, so did the countermeasures
- Generations
  - first - signature scanners (bit patterns all the same)
  - second – heuristics (integrity checks; checksums)
  - third - identify actions (find by actions they do)
  - 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
- *Let virus do the work for an antivirus program by exposing it in a controlled environment*

Digital immune system

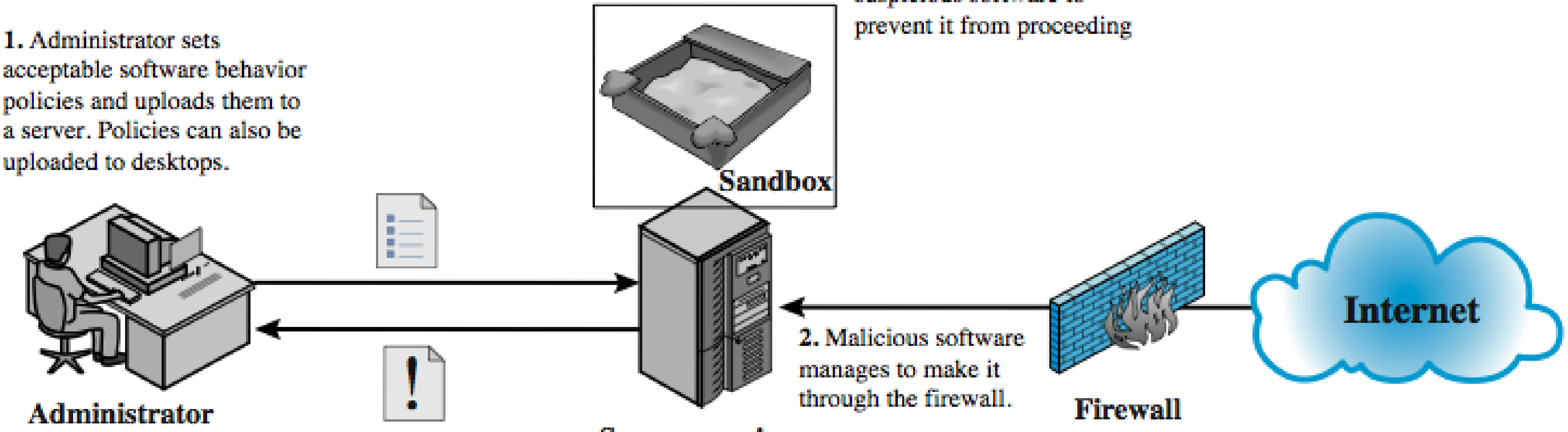


1. A monitoring pgm infers a virus, sends a copy to an adm machine
2. Adm encrypts, sends to a central analysis machine
3. Central analysis: Safe exec of virus, analyze, give a prescription
4. Prescription sent back to the adm machines
5. Adm machine forwards to all clients
6. Prescription forwarded to other organizations
7. Subscribers worldwide receive regular updates

IBM/Symantec Project

Behavior-blocking software

Integrates with the OS; looks for bad behavior



3. Behavior-blocking software at server flags suspicious code. The blocker"sandboxes" the suspicious software to prevent it from proceeding

1. Administrator sets acceptable software behavior policies and uploads them to a server. Policies can also be uploaded to desktops.

2. Malicious software manages to make it through the firewall.

4. Server alerts administrator that suspicious code has been identified and sandboxed, awaiting administrator's decision on whether the code should be removed or allowed to run.

Server running behavior-blocking software

Monitored behaviors:

-Attempts to open, view, delete, modify files

-Attempts to format drives

-Modifications to the logic of executables

-Modifications to critical system settings

-Scripting of emails to send exec contents

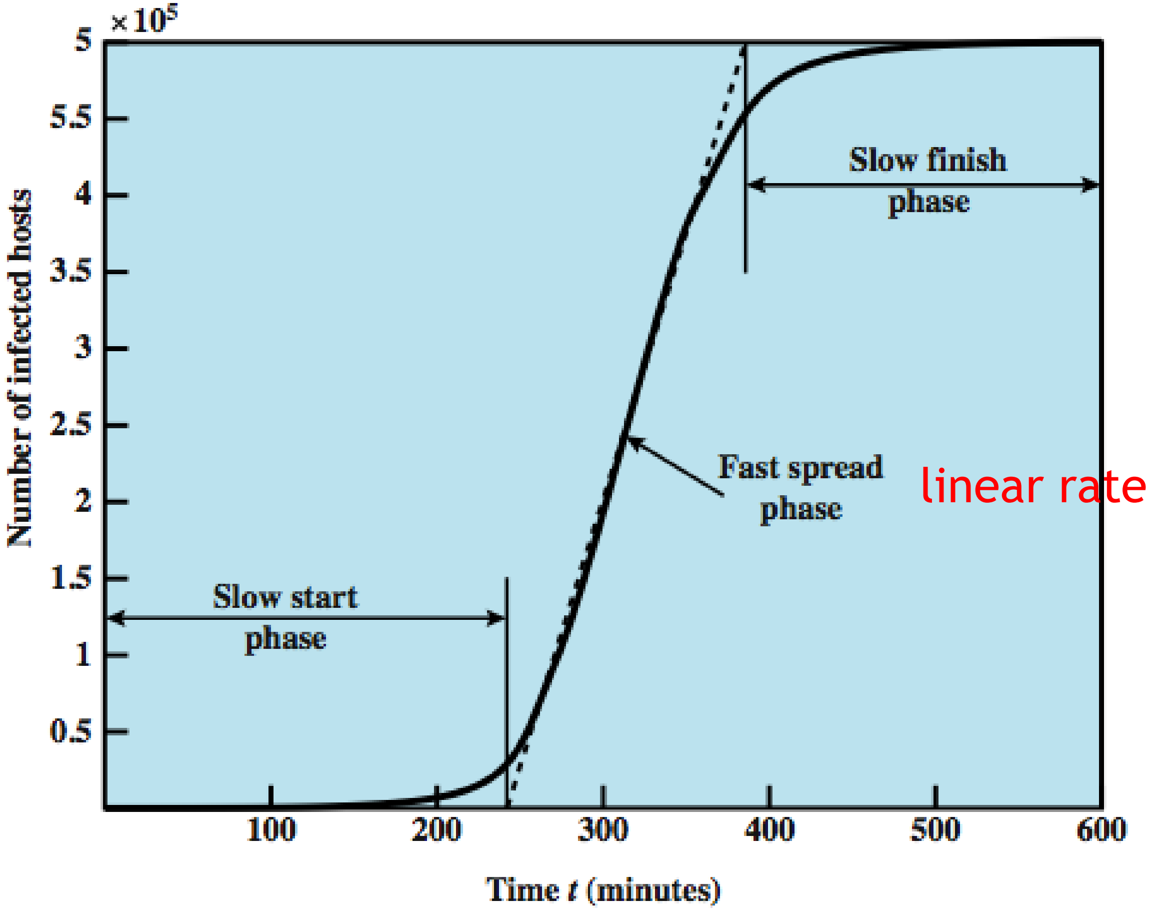
Worms

- 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
  - Affected 6,000 computers; cost $10-$100 M
- 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 (based on recent attacks)



linear rate of infection

exponential rate of infection

More recent worm attacks

- Code Red
  - July 2001 exploiting MS IIS bug
  - probes random IP address, does DDoS attack
  - consumes significant net capacity when active
  - 360,000 servers in 14 hours
- Code Red II variant includes backdoor: hacker controls the worm
- SQL Slammer *(exploited buffer-overflow vulnerability)*
  - early 2003, attacks MS SQL Server
  - compact and very rapid spread
- Mydoom *(100 M infected messages in 36 hours)*
  - mass-mailing e-mail worm that appeared in 2004
  - installed remote access backdoor in infected systems

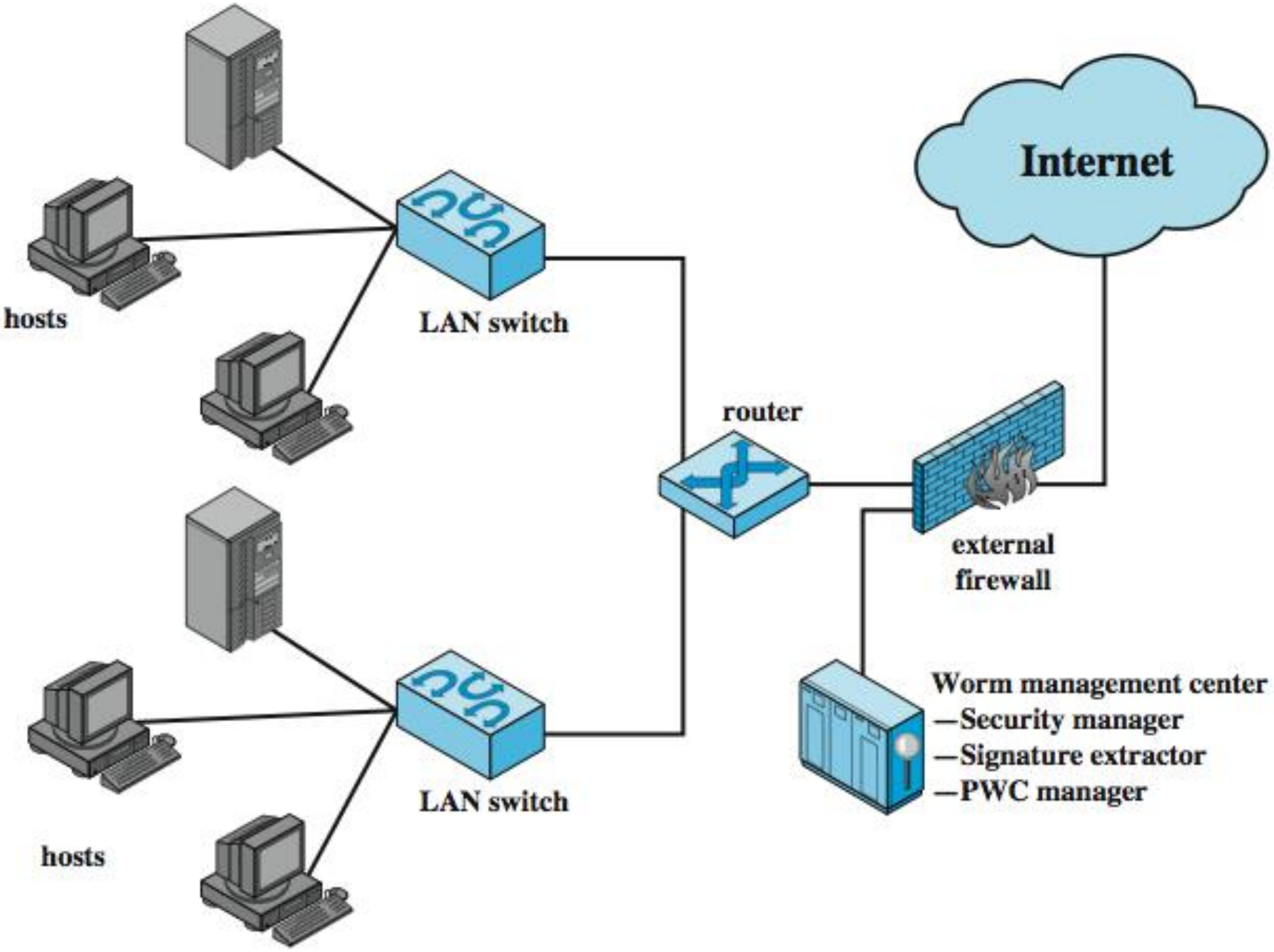
State of worm technology

- Multiplatform: not limited to Windows
- Multi-exploit: Web servers, emails, file sharing …
- Ultrafast spreading: do a scan to find vulnerable hosts
- Polymorphic: each copy has a new code
- Metamorphic: change appearance/behavior
- Transport vehicles (e.g., for DDoS)
- Zero-day exploit of unknown vulnerability (to achieve max surprise/distribution)

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: define signatures
  - filter-based worm containment (focus on contents)
  - payload-classification-based worm containment (examine packets for anomalies)
  - threshold random walk scan detection (limit the rate of scan-like traffic)
  - rate limiting and rate halting (limit outgoing traffic when a threshold is met)

Proactive worm containment



1. PWC agent monitors outgoing traffic for increased activity

2. When an agent notices high traffic, it informs the PWC manager; mgr propagates to other hosts

3. Hosts receive alert and decide if to ignore (based on time of last incoming pkt)

4. Relaxation period (based on threshold)

Mobile code

- Scripts, macros or other portable instructions
- Popular ones: JavaScript, ActiveX, VBScript
- Heterogeneous platforms
- From a remote system to a local system
- Can act as an agent for viruses, works, and Trojan horses
- Mobile phone works: communicate the Bluetooth connections (e.g., CommWarrior on Symbian but attempts on Android and iPhone)

Client-side vulnerabilities

- Drive-by-downloads: common in recent attacks
- Exploits browser vulnerabilities (when a user visits a website controlled by the attacker or a compromised website)
- Clickjacking

Social engineering, spam, email, Trojans

- Spam (much better protection now)
- Trojan horse: looks like a useful tool but contains hidden code

Payload

- Data destruction, theft
- Data encryption (ransomware)
- Real-world damage
  - Stuxnet: caused physical damage also (targeted to Siemens industrial control software)
- Logic bomb

Payload attack agents: bots (zombie/drone)

- Program taking over other computers and launch attacks
  - hard to trace attacks
- If coordinated form a ***botnet***
- Characteristics:
  - **remote control facility** (distinguishing factor)
    - via IRC/HTTP etc
  - spreading mechanism
    - attack software, vulnerability, scanning strategy
- Various counter-measures applicable (IDS, honeypots, …)

Uses of bots

- DDoS
- Spamming
- Sniffing traffic
- Keylogging
- Spreading malware
- Installing advertisement
- Manipulating games and polls

Payload: information theft

- Credential theft, key loggers, spyware
- Phishing identify theft
- Spear phishing (act as a trusted source for a specific target)

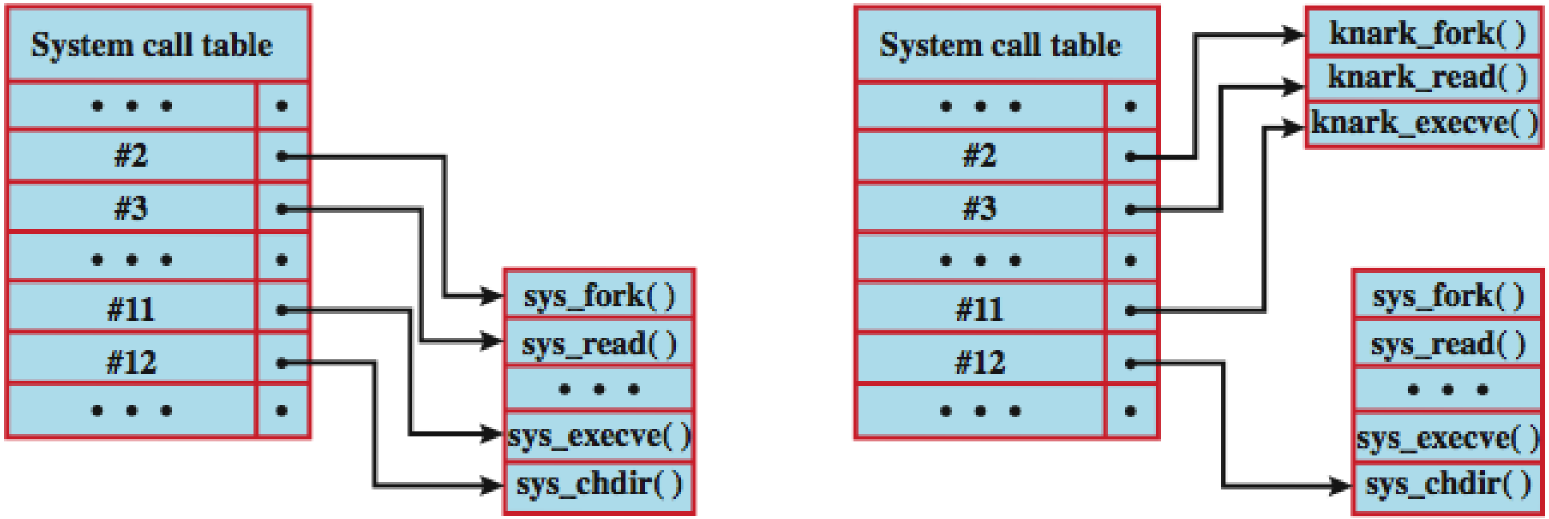
Payload: rootkits and backdoor

- 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 (survives reboot) or memory-based
- Do not rely on vulnerabilities
  - installed via Trojan
  - installed via hackers
- Backdoor: often by programmers

Rootkit System Table Mods

A Unix Example

User API calls refer to a number; the system
maintains a system call table with one entry per number;
each number is used to index to a corresponding system routine



(a) Normal kernel memory layout

(b) After nkark install

rootkit modifies the table and the calls go to the hackers
replacements

Countermeasures

- Prevention
- Detection, identification, removal
- Requirement
  - generality
  - Timeliness
  - Resiliency
  - Minimal DoS costs
  - Transparency
  - Global/local coverage (inside and outside attackers)

Summary

- introduced types of malicous software
  - incl backdoor, logic bomb, trojan horse, mobile
- virus types and countermeasures
- worm types and countermeasures
- bots
- rootkits