NC STATE UNIVERSITY Computer Science

CSC 405
Introduction to Computer Security

Topic 3. Program Security -- Part I

Program Security

- What is exactly a secure program?
  - Different people may give different answers
  - In general, a secure program should behave as their designers intended or uses expected
- Program security flaws
  - Unexpected program behavior
  - Two classes
    - Inadvertent human errors
    - Intentionally induced flaws
    - Both types can cause serious damages

Non-malicious Program Errors

- Common non-malicious program errors
  - Buffer overflows
  - Incomplete mediation
  - Time-of-check to time-of-use errors

Buffer Overflows

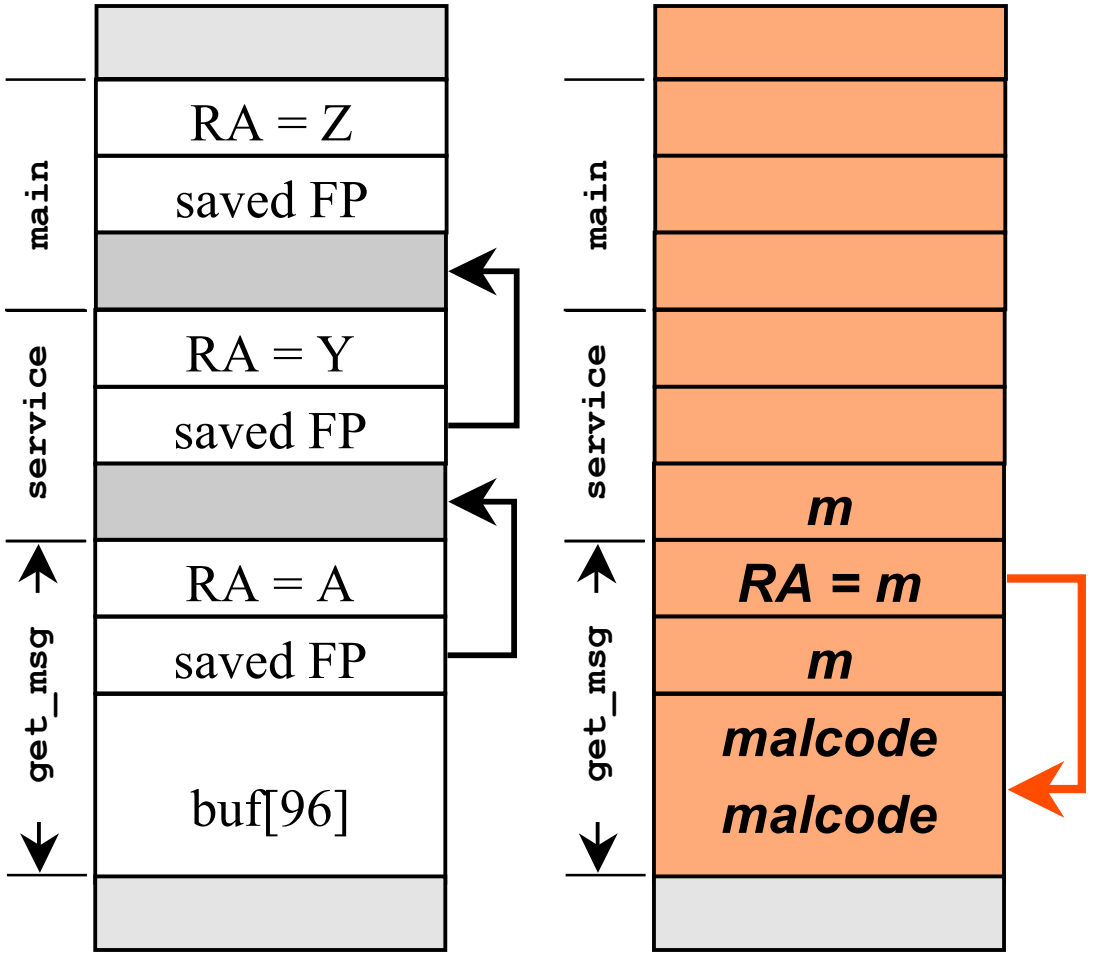
- A buffer is a space in which data can be held
  - A buffer is finite
- Buffer overflow
  - Out of bound use of the buffer
- Example

```
char sample[10];
for (int i=0; i<=9; i++)
  sample[i]='A';
sample[10]='B';
```

An Example of Stack Buffer Overflow

```
service(){
  get_msg();
A:send_msg();
}

get_msg(...){
  char buf[96];
  ...
  gets(buf);
  ...
B:return;
}
```



Incomplete Program Mediation

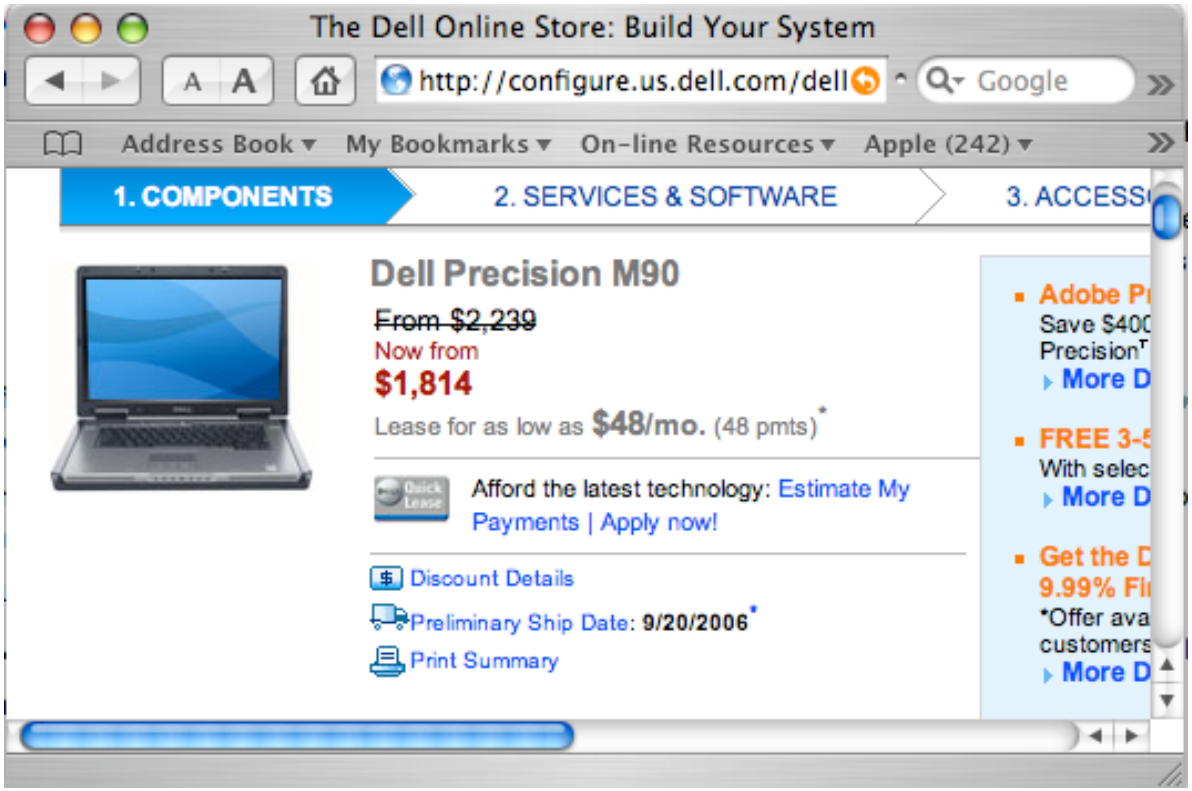


Illustration only. This by no means implies DELL has this problem.

After clicking "order", the system completes the transaction with http://www.xxxx.com/order.php?final=yes&custID=101&product=PreM90&quantity=1&price=~~1814~~ 100

Time-of-Check to Time-of-Use Errors

- Access control is often required, but access privileges are not checked universally

Example

- `binmail` vulnerability
  - On Sun OS 4.1.x
  - Permission of `/var/spool/mail is rwxrwxrwt`
  - Before it opens the mail file, `binmail` does an `lstat()` to check that it is not about to write to a linked file
  - It then use `open()` to access that file
  - If a link is created after `lstat()`, `open()` will then follow the link
  - Exploit
    - Create/append to root's `.rhost` file
    - Note that `binmail` can write to anybody's mailbox

Malicious Code

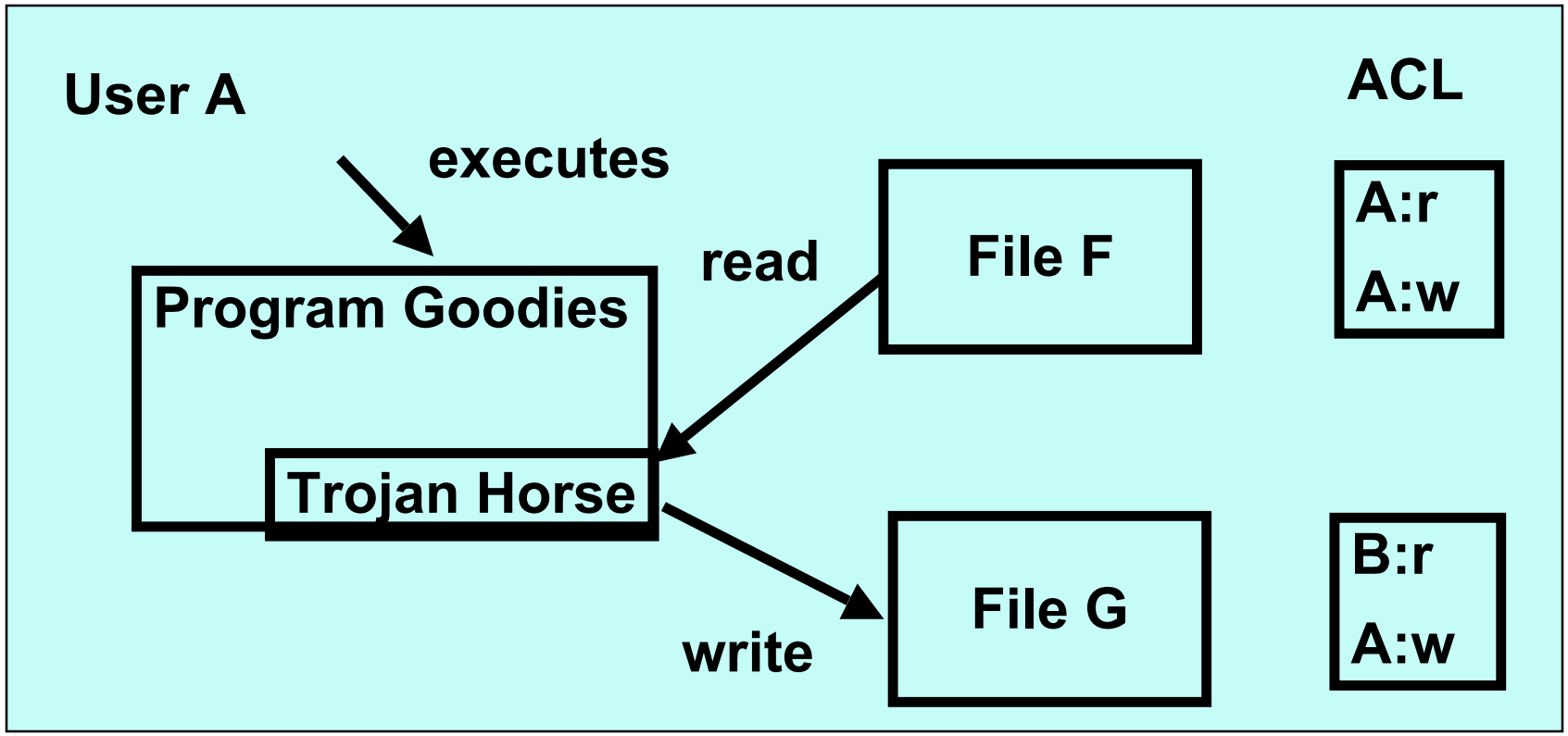
- Kinds of malicious code
  - Viruses
  - Trojan horses
  - Logic bombs
  - Trapdoor (backdoor)
  - Worms
  - Rabbits

Virus

- A virus is a program that attaches itself into one or more files and then performs some (possibly null) action
- Transient virus
  - A transient virus has a life that depends on the life of its host
    - Runs when the attached program runs
    - Terminates when that program terminates
- Resident virus
  - Locates itself in memory
    - Remain active even when the attached program ends

Trojan Horses

- A Trojan horse is malicious code that, in addition to its primary effect, has a second, non-obvious malicious effect



Logic Bombs

- A logic bomb is a program that performs an action that violates the security policy when some external event occurs
  - Example
    - Erase all the employee records when John Smith is no longer an active employee
  - Time bomb: a logic bomb whose trigger is a time or date

Trapdoor (Backdoor)

- A feature in a program by which someone can access the program other than by the obvious way, perhaps with special privilege
  - Example
    - An ATM allows anyone entering 990099 on the keypad to get all the transactions

Worm

- A worm is a program that spreads copies of itself through a network
- Different from viruses
  - Viruses depend on other programs
  - Worms are usually standalone applications
  - Viruses usually trick people into propagating them
  - Worms can hack into vulnerable systems and spread without depending on others

Rabbits (Bacteria)

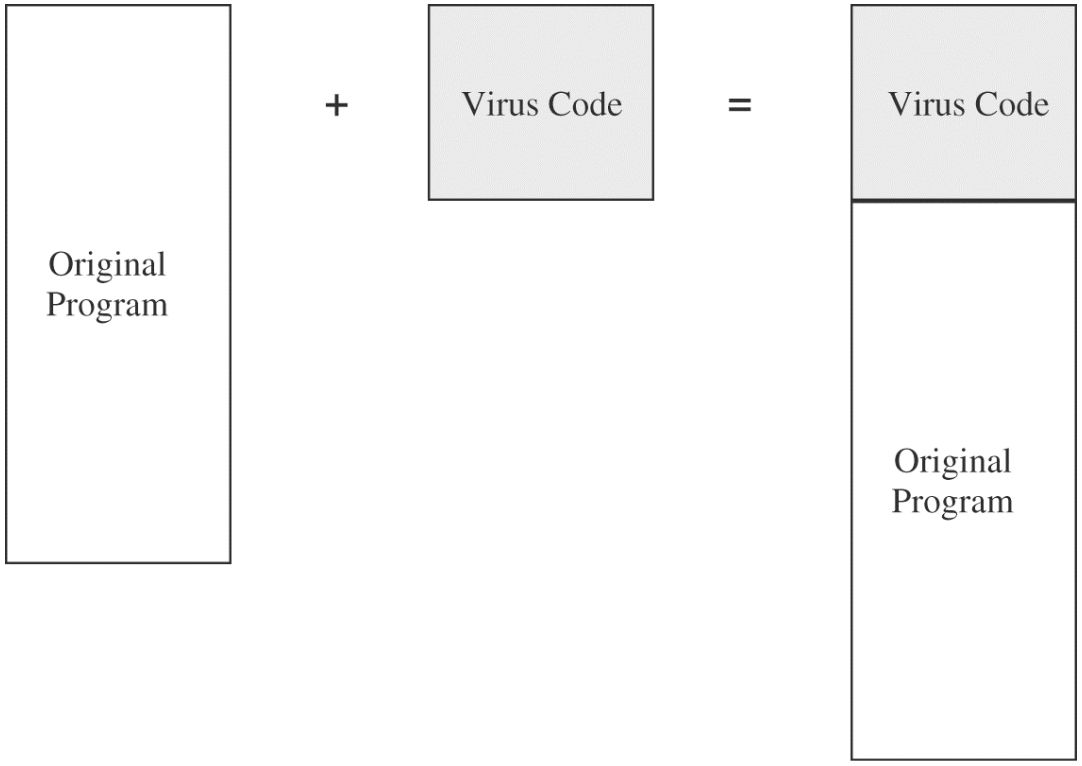
- A bacterium or a rabbit is a program that absorbs all of some class of resource
  - Could be a virus or worm
- Example
  - Exhaust disk space
  - Exhaust inode tables

How does Virus Work

- Two phases
  - Insertion phase
    - The virus inserts itself into a file (or files)
  - Execution phase
    - The virus executes
- Usually trick human users to execute the virus
  - This is necessary for the virus to take control
  - Examples
    - Email attachments
    - Hide in boot sector of bootable medium

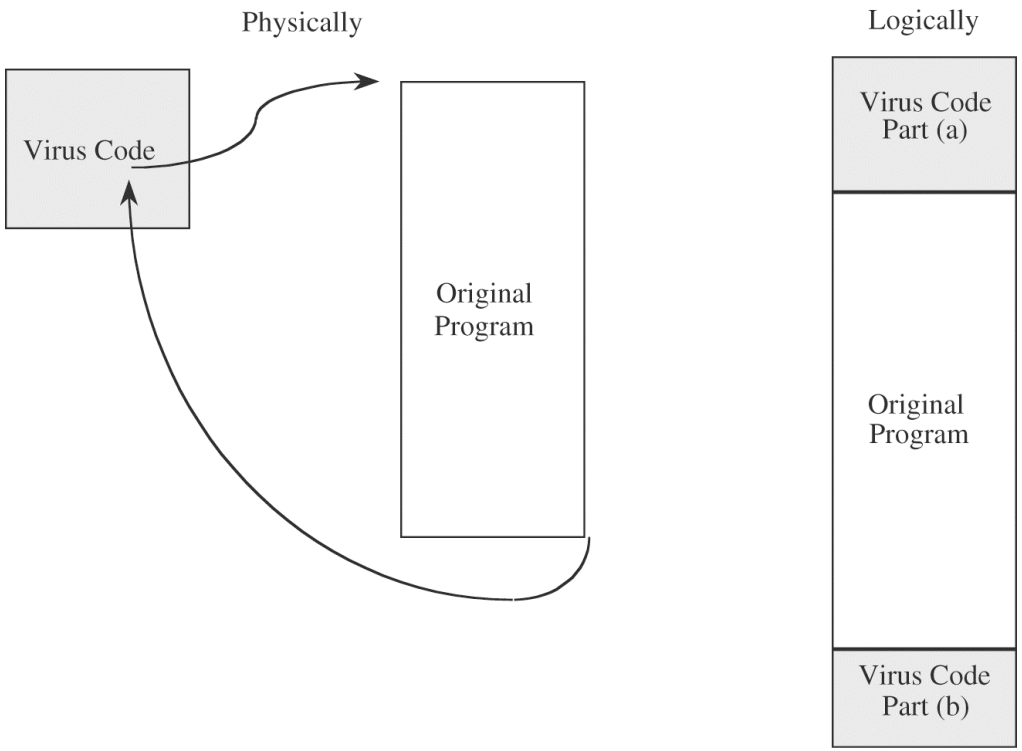
Places to Insert Virus Code

- Virus appended to a program
  - Virus instruction first executed
  - Original program executed after the last virus instruction



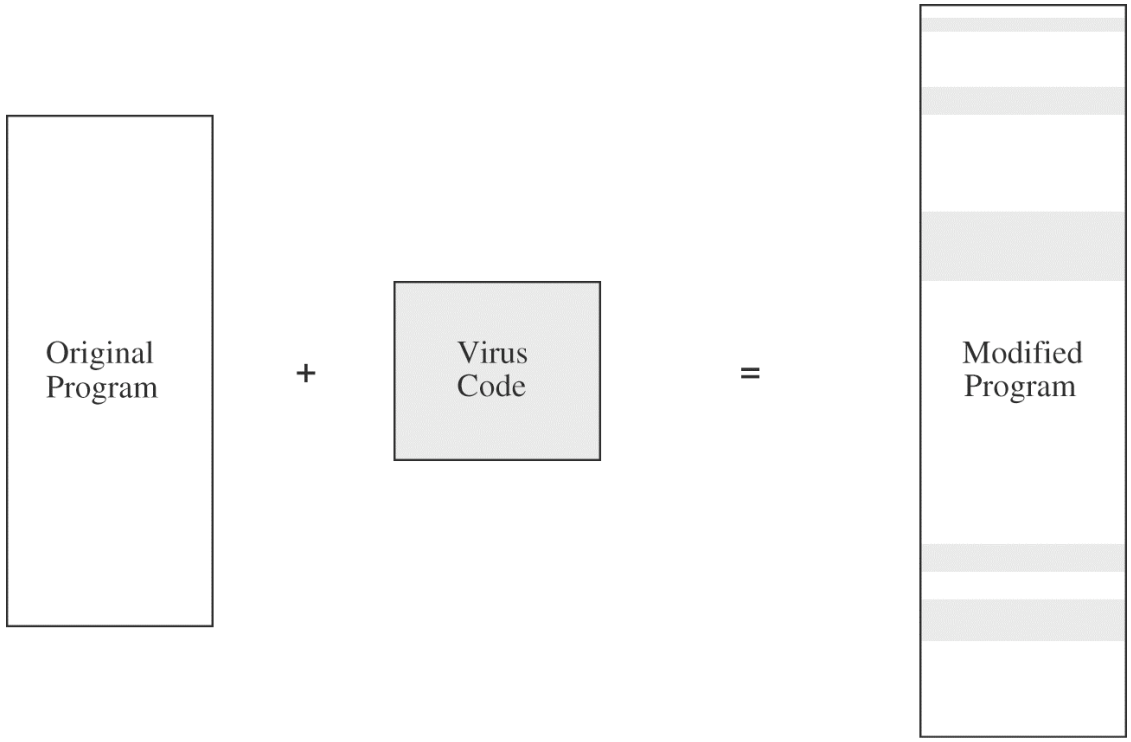
Places to Insert Virus Code (Cont'd)

- Virus that surround a program
  - Has control before and after the virus execution
  - Example: modify the output of the original program



Places to Insert Virus Code (Cont'd)

- Virus integrated in the original program
  - The virus writer has to know the exact structure of the original program
  - Targeted infection; Rare



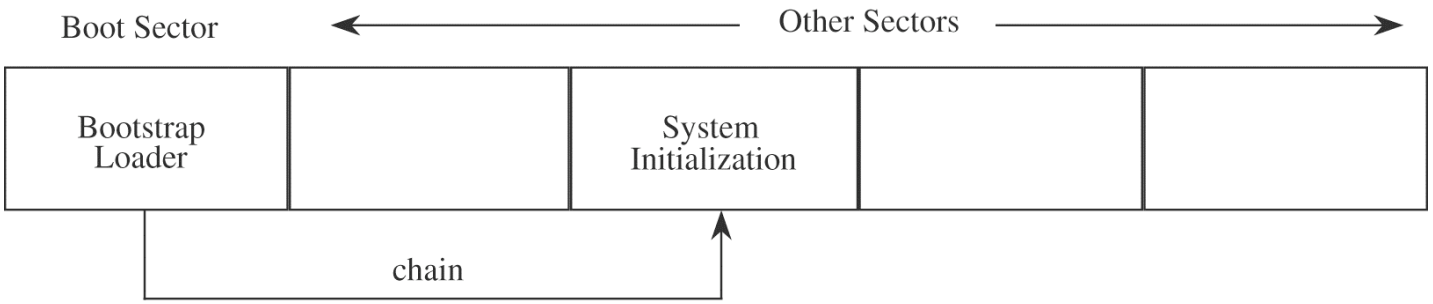
How Virus Gain Control

- Boot sector infectors
  - The boot sector is the part of a disk used to bootstrap the system.
  - Code in a boot sector is executed when the system "sees" the disk for the first time.

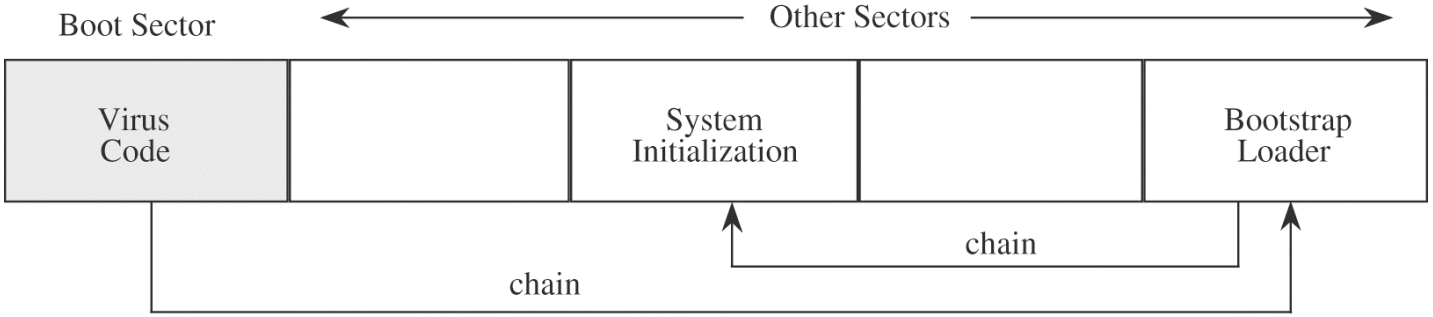
Brian Virus

1. Move the disk interrupt vector 13H to 6DH
2. Set 13H to invoke Brian virus
3. Load the original boot sector

Boot Sector Infector (Cont'd)



(a) Before infection

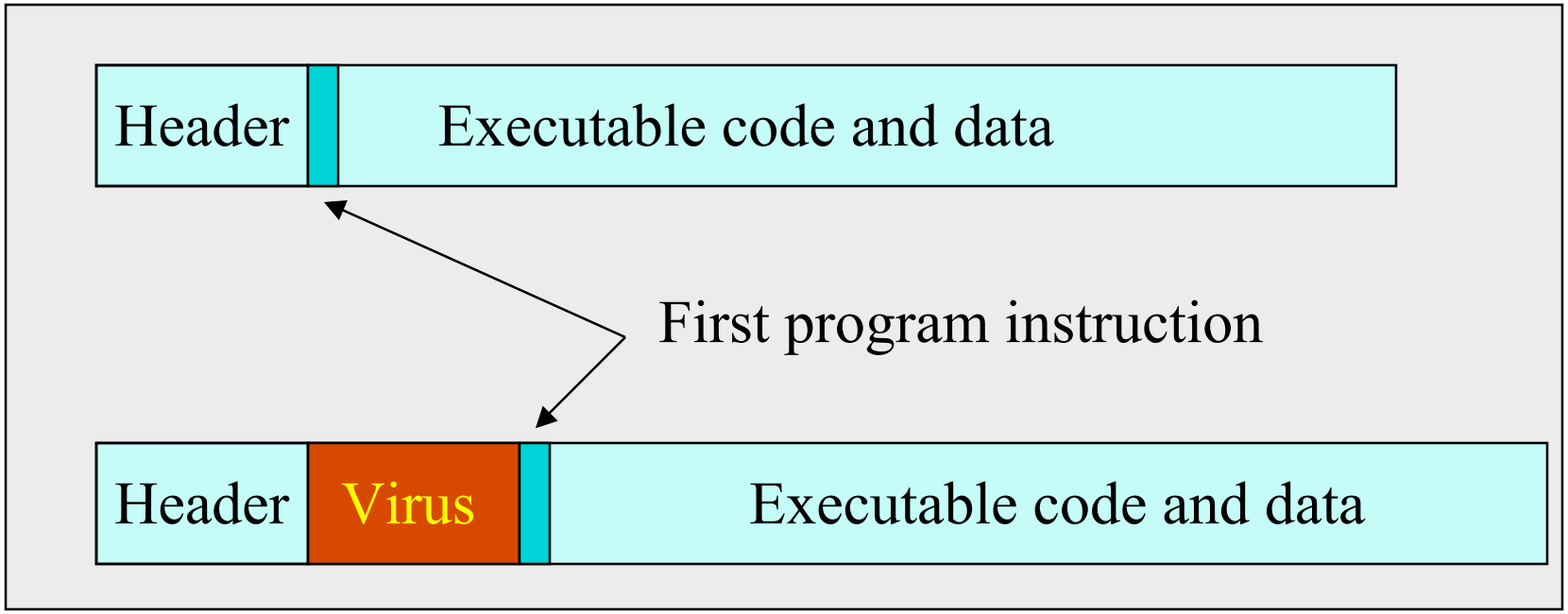


(b) After infection

1. Copy the old boot sector to alternative place;
2. Insert itself into the boot sector.

How Virus Gain Control (Cont'd)

- Executable infectors
  - Triggered if an infected program is executed
  - Infect executables
    - COM and EXE



Terminate and Stay Resident (TSR) Virus

- TSR virus
  - Stays active in memory after the application (or bootstrapping) has terminated.

Brian Virus

1. Move the disk interrupt vector 13H to 6DH
2. Set 13H to invoke Brian virus
3. Load the original boot sector

New disks will be infected as long as the virus is in memory.

Viruses (Cont'd)

- Stealth viruses
  - Conceal the infection of files
  - Make itself difficult to detect
- Polymorphic viruses
  - Encrypt itself with a random key
  - Avoid detection by anti-virus programs, which search for patterns of viruses.
- Metamorphic viruses
  - Change its form each time it inserts itself into another program.

Viruses (Cont'd)

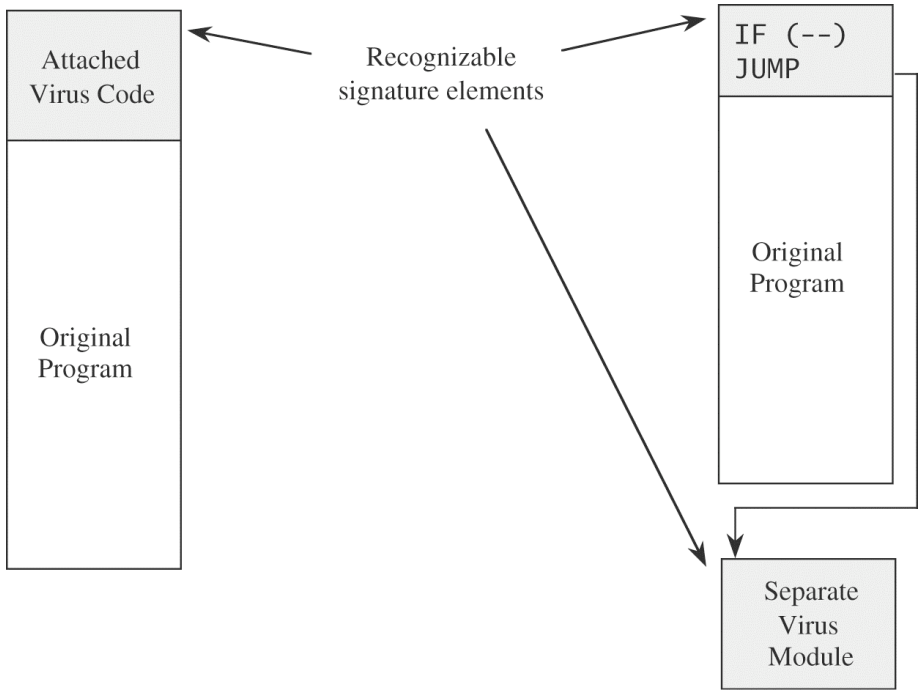
- Document (Macro) viruses
  - Viruses composed of instructions that are interpreted, rather than executed.
  - Examples
    - Word viruses
    - Email viruses
  - MS Office suite is the most popular target.

Virus Signature and Virus Scanner

- Virus code must be stored somewhere
- Virus signature
  - Characteristics of a virus
- Virus scanner
  - Program that looks for virus signatures
  - A virus scanner is effective only if it is kept up-to-date with the current virus signatures
  - Examples
    - Symantec AntiVirus
    - Norton AntiVirus

Virus Signatures

- Storage patterns
  - A virus needs to take control of the program
  - Attach to a file ==> increased file size
    - Remove a part of the original file ==> program function impaired



Truths and Misconceptions about Virus

- Virus can only infect Microsoft Windows system
- Viruses can modify “hidden” or “read only” files
- Viruses can appear only in data files, or only in Word documents, or only in programs
- Viruses spread only on disks or only in email
- Viruses cannot remain in memory after a complete power off/power on reboot
- Viruses cannot infect hardware
- Viruses can be malevolent, benign, or benevolent

The Internet Worm

- Morris Worm, Nov 2nd, 1988
  - The first worm
  - Robert T. Morris, Jr.
    - 23 years old
    - Cornell grad student
  - Wrote a self-propagating program as a "test concept"
    - Exploited Unix vulnerabilities in sendmail and fingerd
    - Released at MIT
    - Bug in the worm caused it to go wild
      - Probably wouldn't have caused much damage otherwise!

The Internet Worm (Cont'd)

- Targeted at Sun 3 and VAX Workstations running BSD based Unix operating systems
- Infected about 6,000 Unix hosts
  - About 10% of the 60,000 hosts on the Internet
- Reactions
  - People didn't know what to do, so they panicked
    - Disconnected from net
    - Unable to receive patches!
  - Morris fined $10k, 3 yrs probation, 400 hrs community service
  - CERT was created

The Internet Worm (Cont'd)

- Code accomplishes three objectives
  1. Determine to where it could spread
     - Offline password guessing (use the dictionary for the spell checker)
     - Buffer overflow vulnerability in fingerd ==> remote shell
     - Sendmail vulnerability (debug mode) to execute arbitrary commands
  2. Spread its infection
     - First a bootstrap loader to the target machine
     - Bootstrap loader fetch the rest of the worm
       - Use a one-time password to authenticate the bootstrap loader

The Internet Worm (Cont'd)

- Code accomplishes three objectives (Cont'd)
  - Remain undiscovered and undiscoverable
    - If worm fetching runs into a transmission error, the bootstrap loader deletes all the code already transferred
    - Once worm is received, it loads the code into memory, encrypt it, and delete all the original copies
    - Periodically change its name and process id
    - Definitely discoverable: The huge traffic resulting from the spread!!!

Code Red

- Appeared in July and August in 2001
- Exploit a buffer overflow vulnerability in Microsoft's Internet Information Service (IIS)
  - Buffer in the dynamic link library idq.dll
- Three versions
  - Code Red I version 1
  - Code Red I version 2
  - Code Red II
    - Substantial rewrite

Code Red I Version 1

- Easy to spot
  - Deface the website

    ```
    HELLO!
    Welcome to
    http://www.worm.com !
    Hacked by Chinese!
    ```
- Activities determined by date
  - Day 1 to 19 of the month
    - Scan and compromise vulnerable computers, starting at the same IP
  - Day 20 to 27
    - Distributed denial of service (DDoS) attacks against www.whitehouse.gov
  - Day 28 to end of month
    - Do nothing

Code Red I Version 2

- Discovered near the end of July 2001
  - Did not deface the website
  - Propagation is randomized and optimized to infect servers more quickly

Code Red II

- Discovered on August 4, 2002
- Inject a Trojan hose in the target
- Modify software to ensure a remote attacker can execute any command on the server
  - Copy `cmd.exe` to four places
    - `C(D):\inetpub\scripts\root.ext`
    - `C(D):\progra~1\common~1\system\MSADC\root.exe`
  - Its own copy of explorer.exe
  - Modify the registry to disable certain file protection
  - Reset the registry every 10 minutes
- Automatically stop propagating in October 2002
- Reboot the server after 24 or 48 hours, wiping itself from memory but leaving the Trojan horse