Vulnerabilities: Malicious Code

Class 10

P&P: Ch 3.3, 3.4

Announcement

- Presence sheet
- Solution to Homework 2 is out
  - Grades will follow …

What Did We Cover Before ?

- Nonmalicious Program Errors
- *Buffer Overflows*
- Incomplete Mediation
- TOCTTOU
- Introduction to Malicious Code

In this lecture

- Viruses
  - How they attach
  - How they gain control
  - Where they are stored
  - Detection …
- Worms
- Web Bugs
- Trapdoors
- …

What is a Virus ?

- *Program that can infect other programs by modifying them to include a, possibly evolved, version of itself*

- Fred Cohen 1983

Qualities of Virus

- Hard to detect
- Not easily destroyed or deactivated
- Spreads widely
- Can re-infect home program/other programs
- Easy to create
- Machine/OS independent

How Viruses Attach

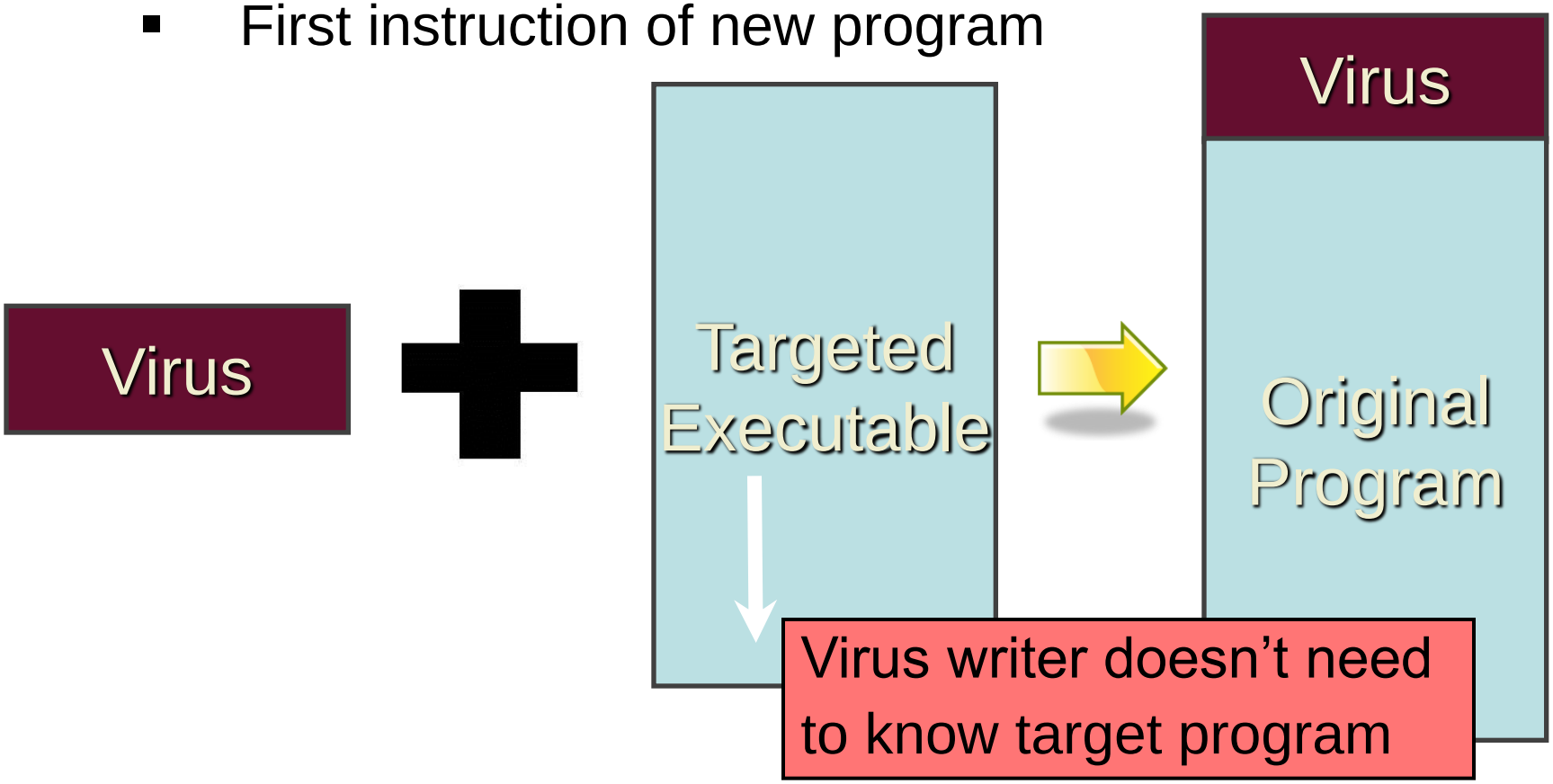
1. Virus is on CD
   - When executed, the virus can
     - Install on hard drive
     - Attach to any executing program in memory
2. E-mail virus
   - The attacker convince victim to open attachment
     - Executable file
     - Graphics, photos …

How Viruses Attach (cont'd)

- Appending
- Surrounding
- Integrated
- Overwriting

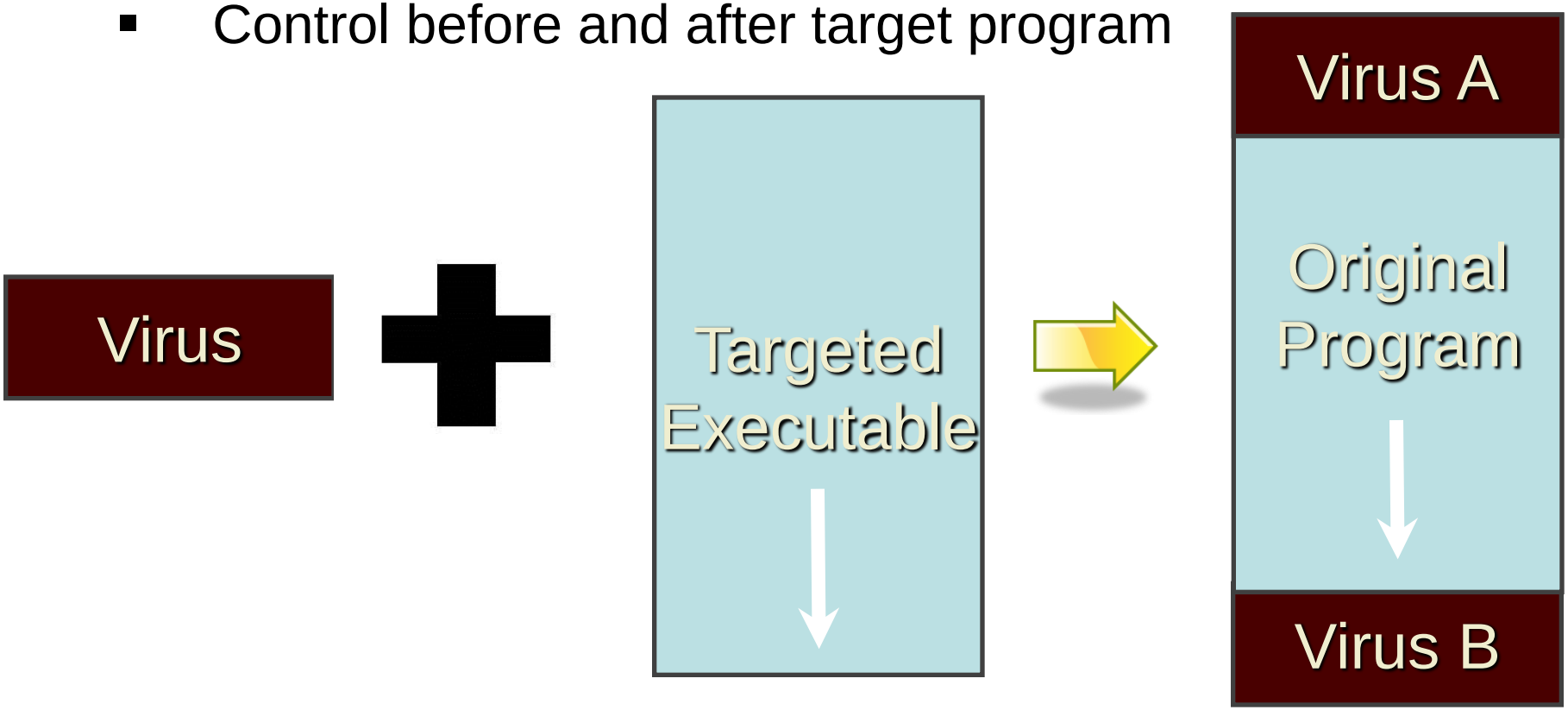
Apending Virus

- Add to beginning of target
  - First instruction of new program



Surrounding Virus

- Add to beginning and end of target
  - Control before and after target program

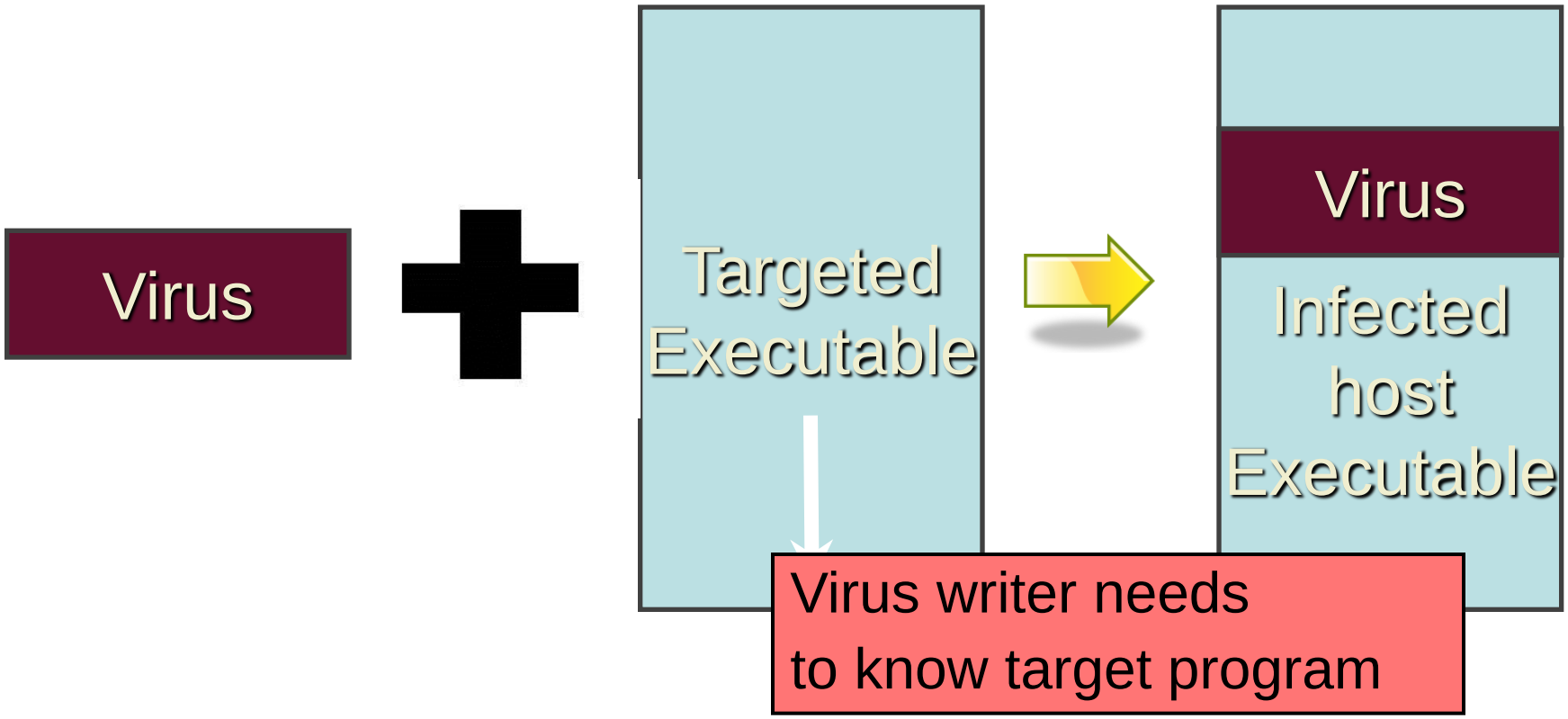


Surrounding Virus - Example

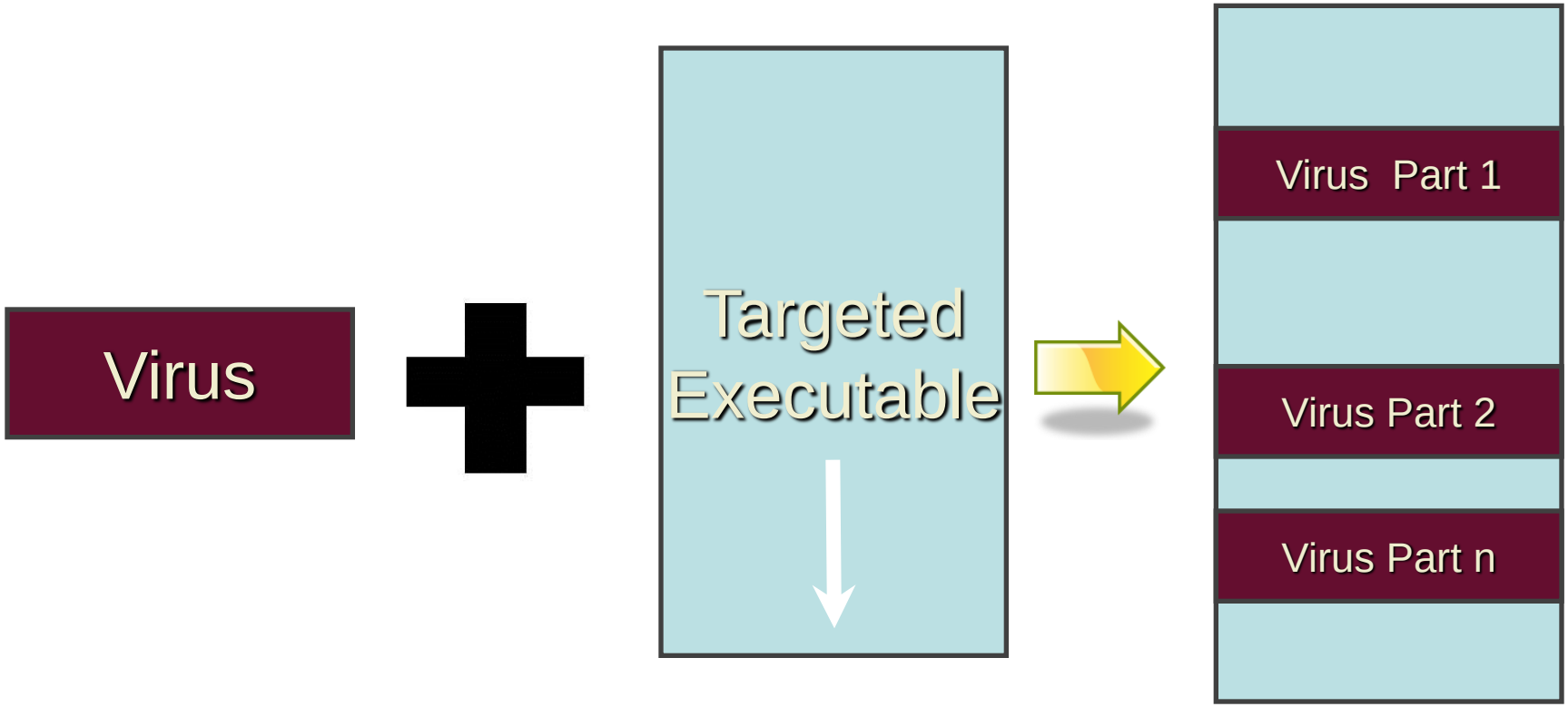
- Prevent user from detecting virus
  - Using file name and size during ls/dir command
- Virus attaches to ls/dir command
- When ls/dir completes, virus takes control
  - Eliminate entry from listing
  - Distribute space among other programs to hide size

Integrated Virus

- Replace some of target

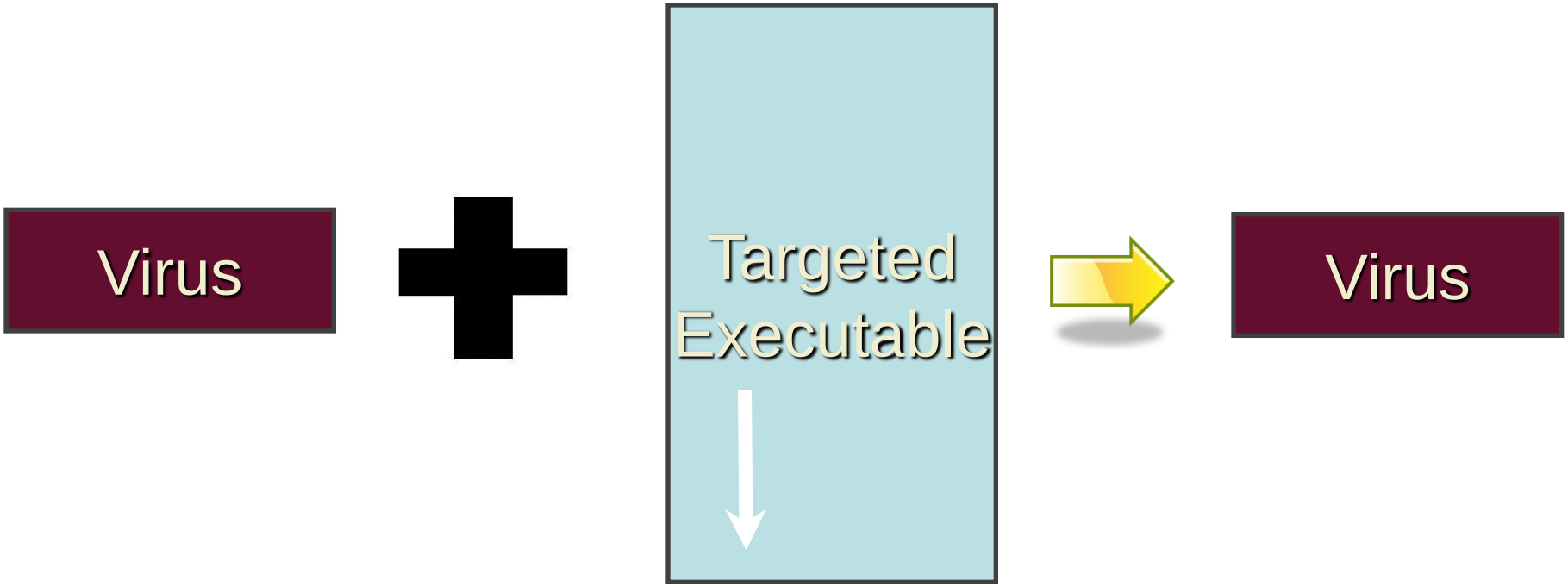


Integrated Virus (cont'd)



Overwriting Virus

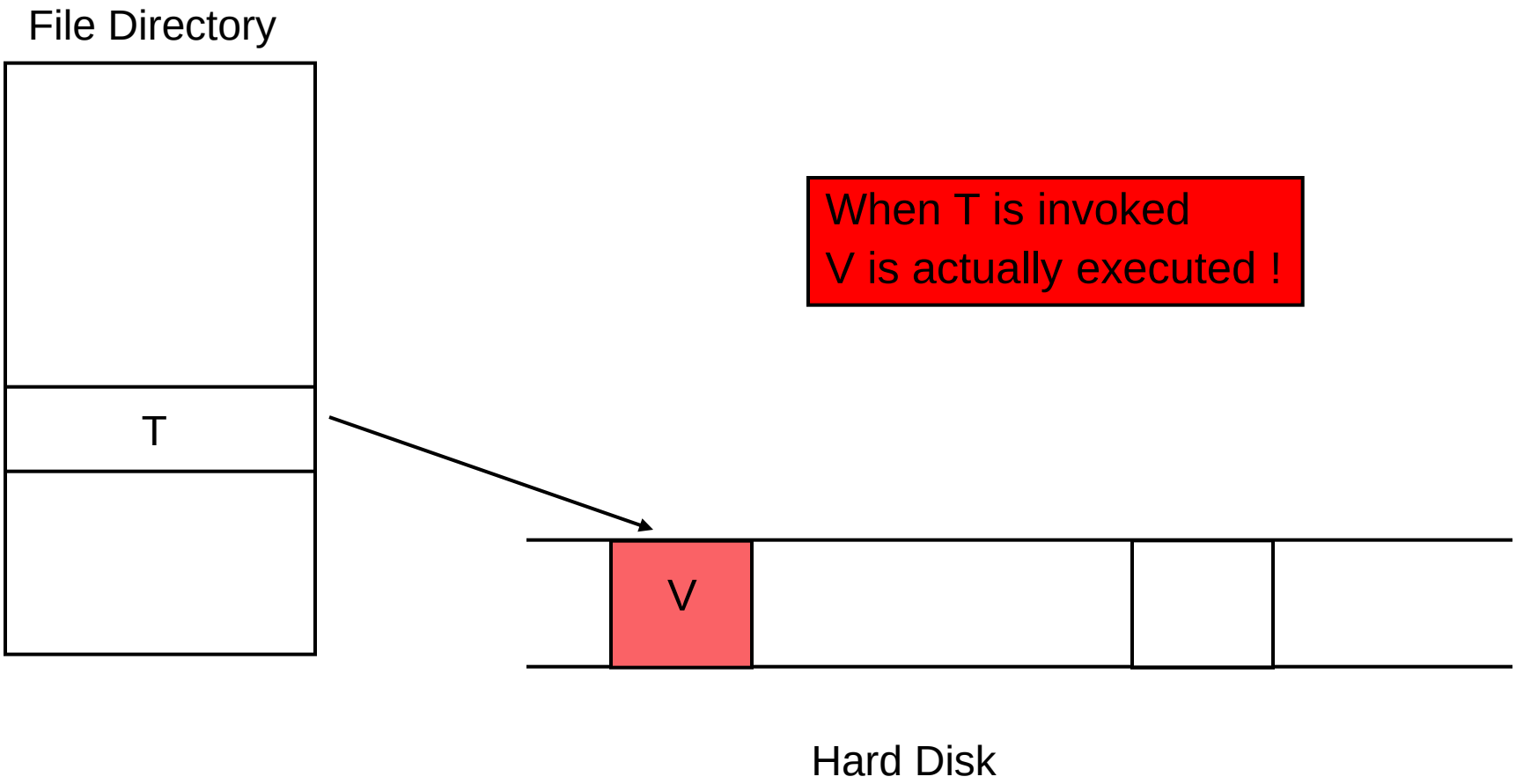
- Replace entire target
  - Mimick effect of target or
  - Not – user likely to perceive virus



How Viruses Gain Control

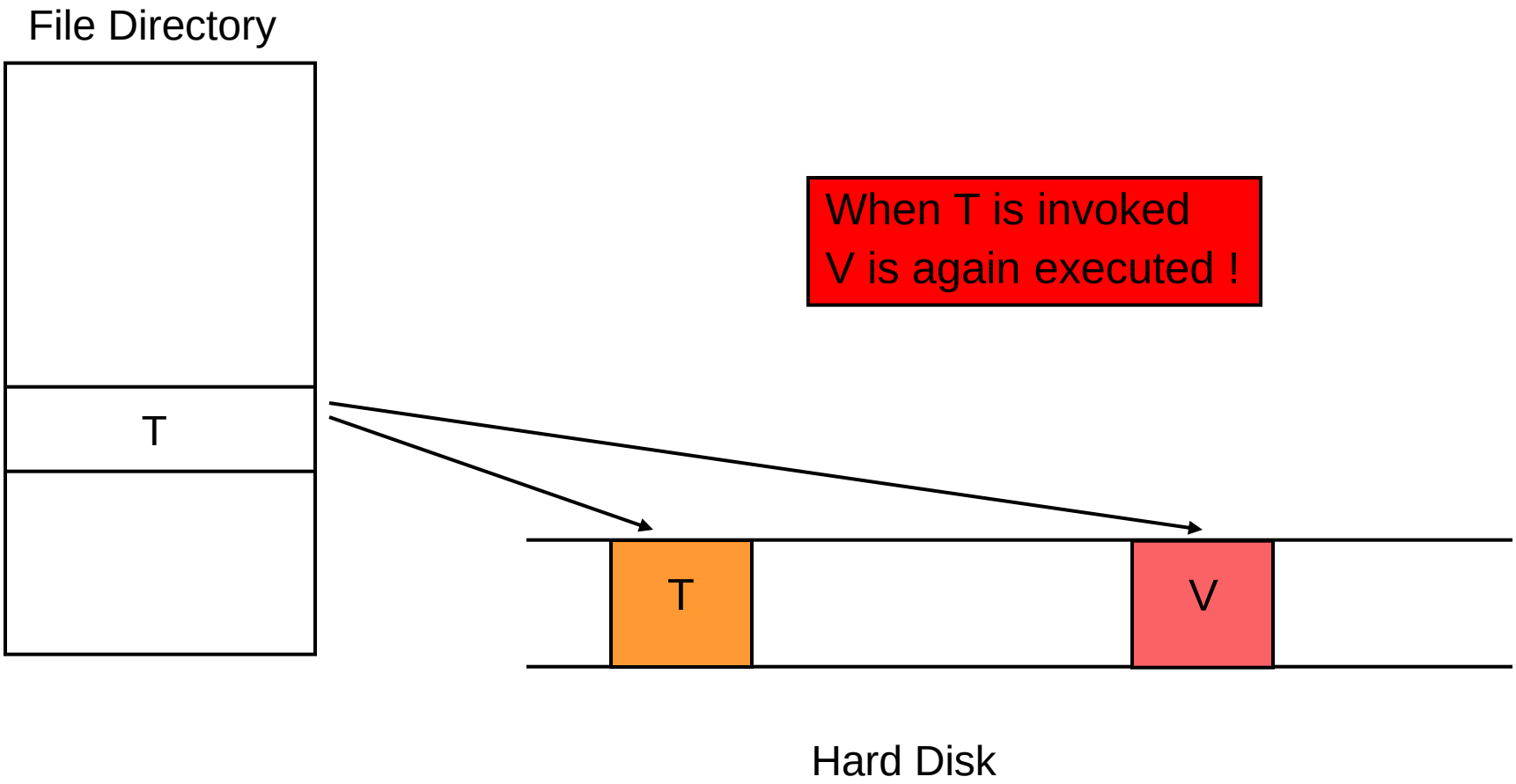
- After attachment virus needs to be invoked
  - Overwriting a target program
  - Changing pointers to programs
- V denotes virus, T is the target program

Overwriting the Target



- Overwrite T with V

Changing Pointer To Target



- Change pointer to T to point to V

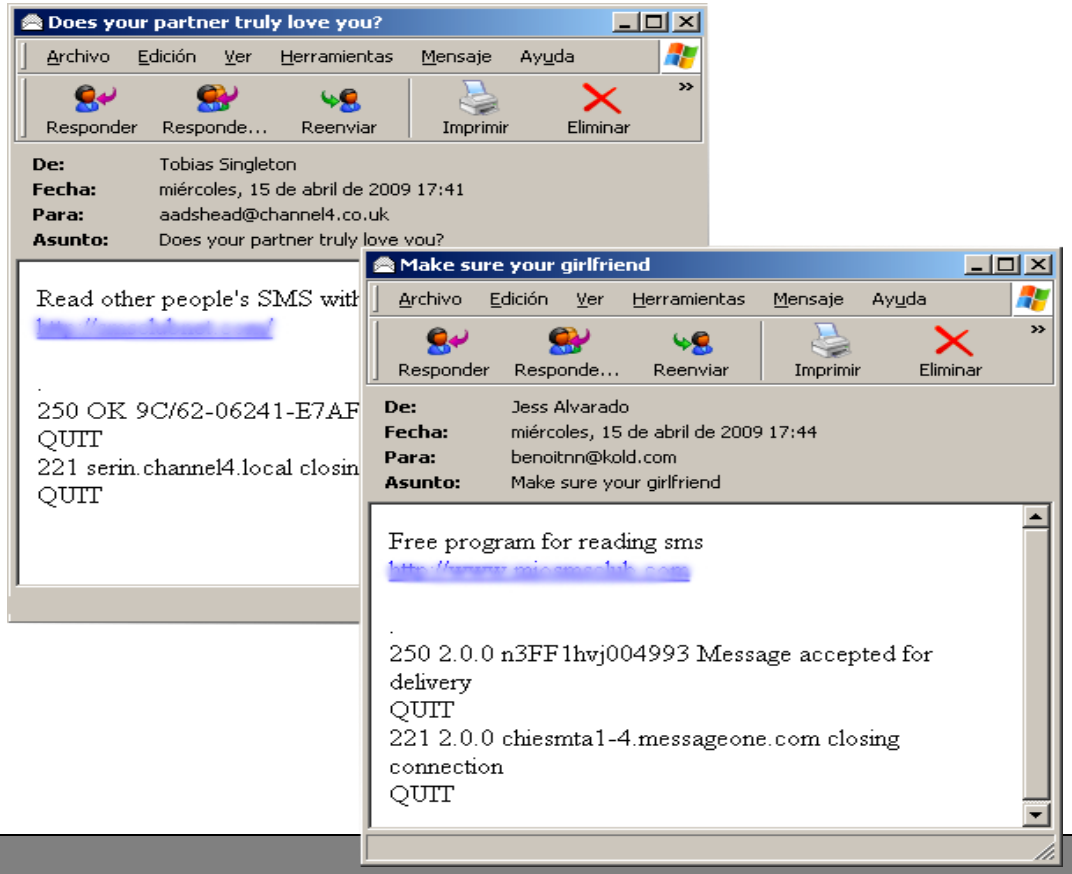
Where Are Viruses Stored

- One-Time execution
- Boot sector
- Memory resident
- …

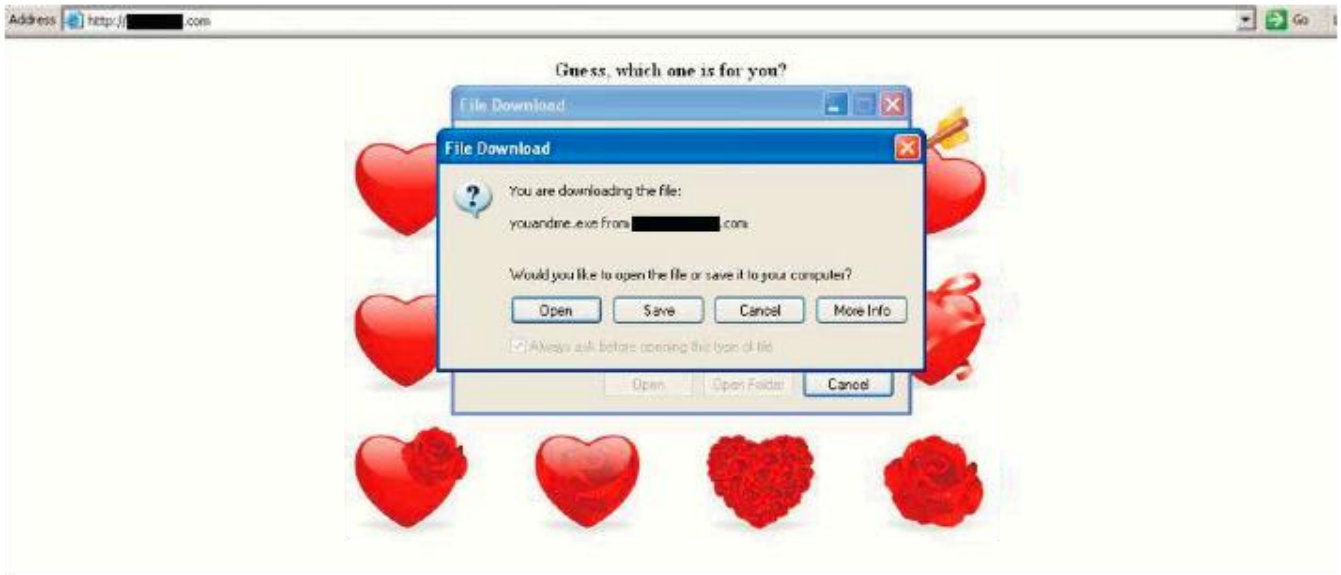
One-Time Execution: E-mail

Virus writer generates e-mail

1. Sends it to all addresses in victim's address book
2. Or leave it to the victim to forward it

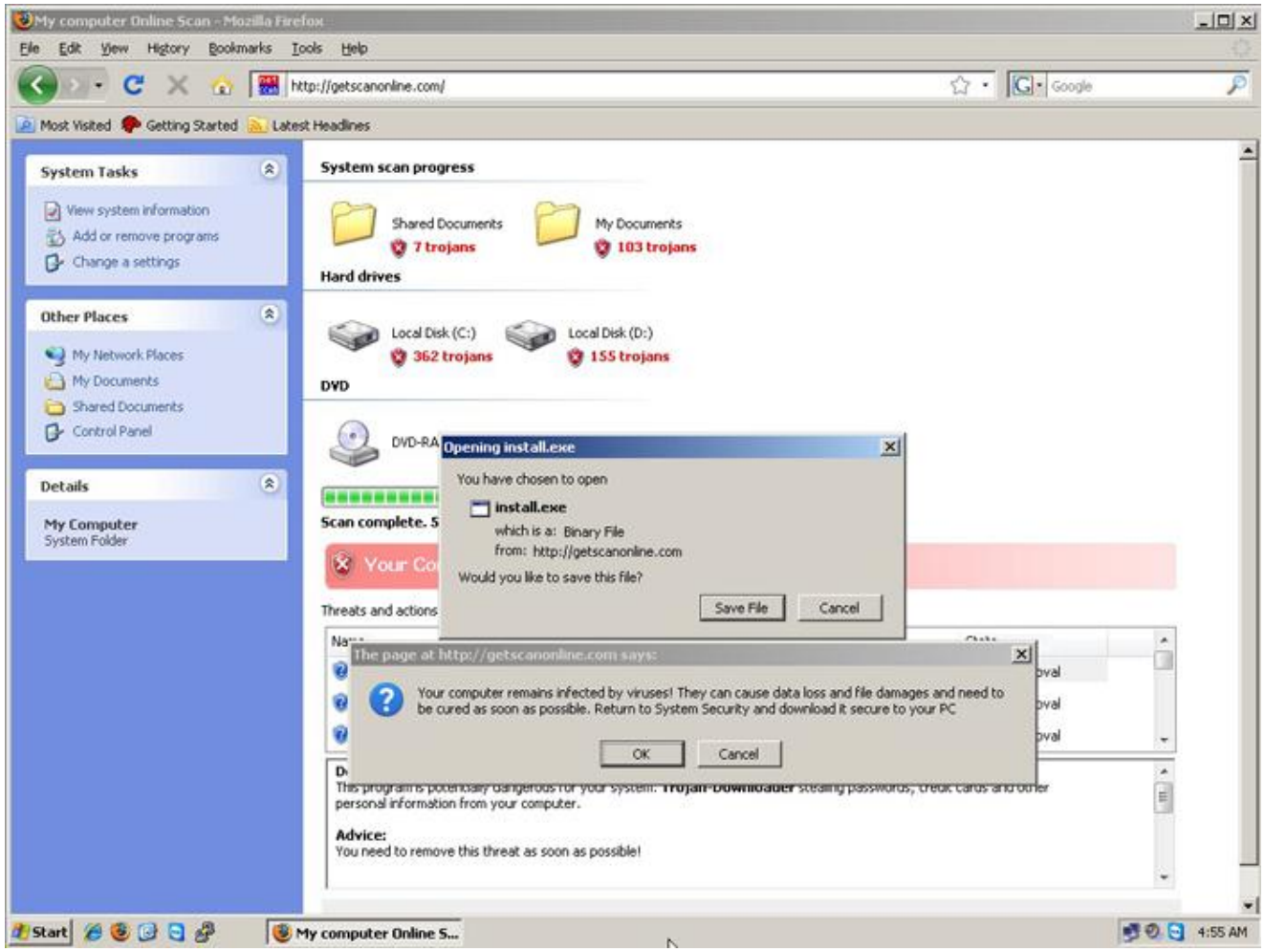


One-Time Execution: Valentine Day



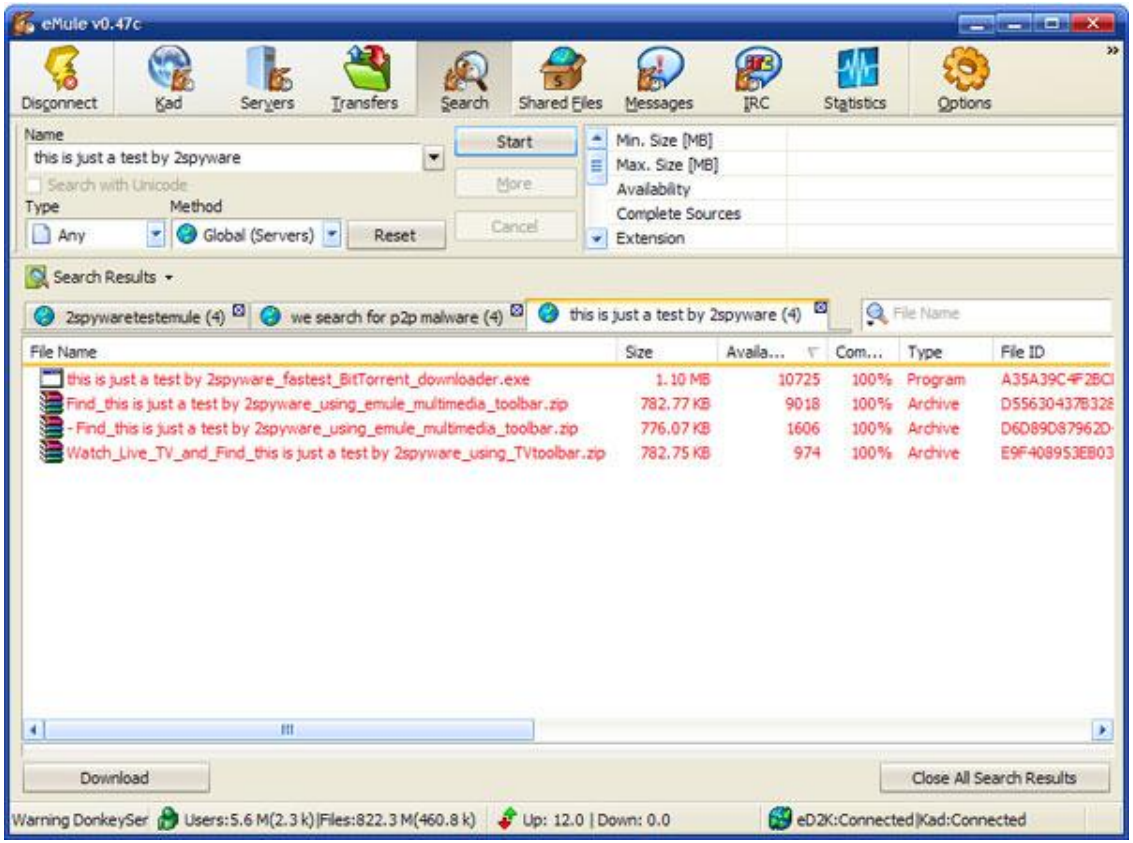
One-Time Execution: Fake Antivirus

One-Time Execution: Fake Page



One-Time Execution: P2P Files

- Popular query
- 35.5% are malwares
  (Kalafut 2006)

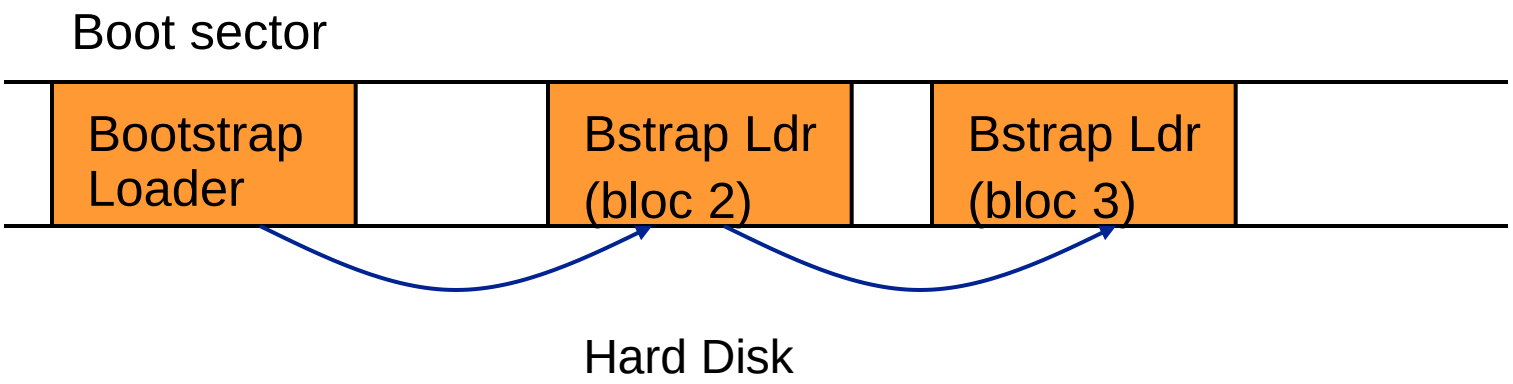


Boot Sector Viruses

- When computer starts
  - Firmware determines hardware components
  - Transfer control to OS
- OS stored on disk
- Bootstrap process:
  - Firmware reads *boot sector* to a fixed address in mem
  - Jump to that address
  - Boot sector contains the bootloader
  - Bootloader pulls the rest of the OS from disk

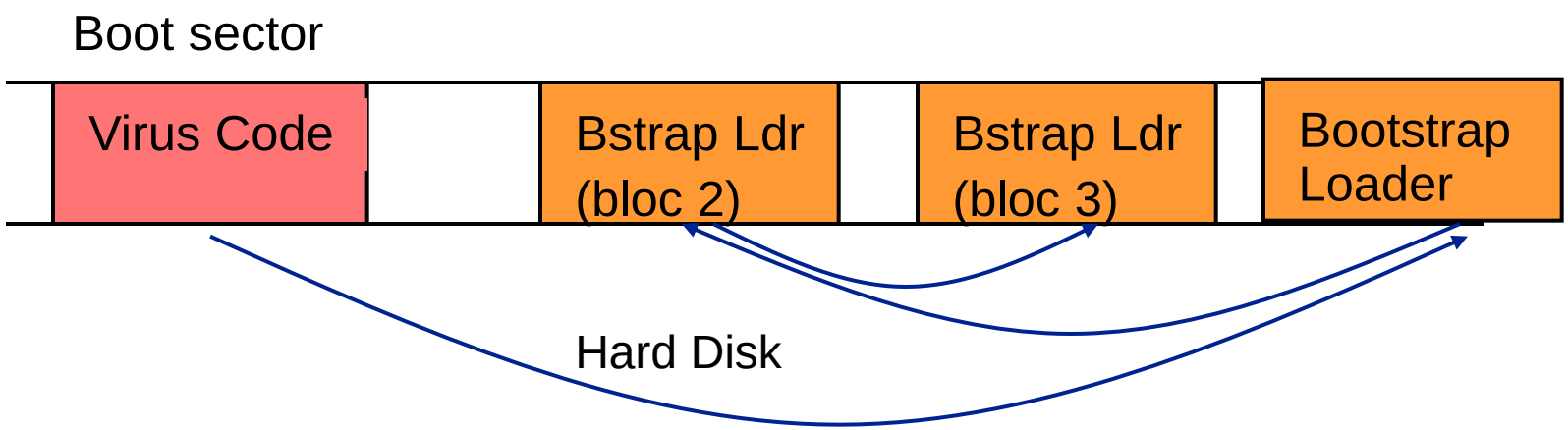
Boot Sector

- Boot sector has 512 bytes
- Bootstrap loader size > 512 bytes
- Use chaining



Placing Virus in Boot Sector

- The virus could be placed in any bootstrap sector
- But … boot sector particularly appealing
  - Virus gains control right at the beginning
  - Protection tools are not yet active



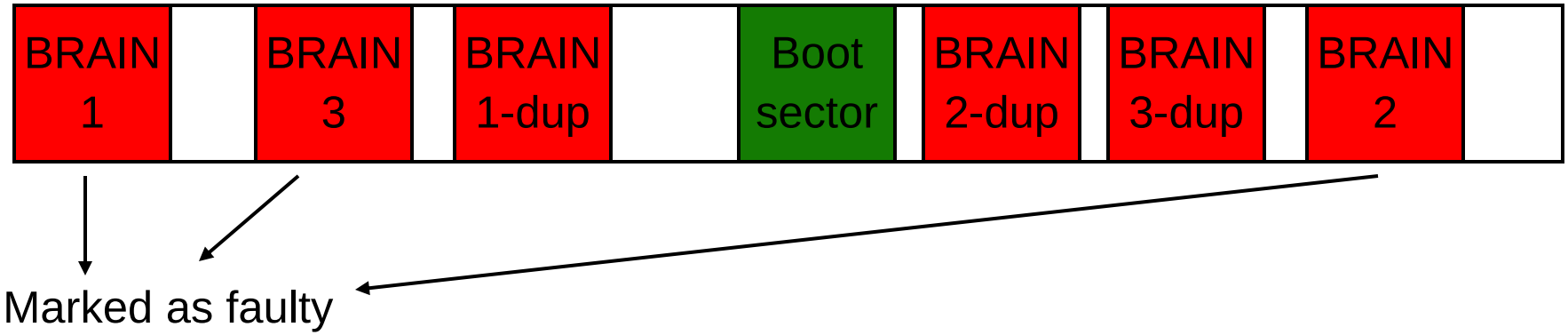
Example: The BRAIN Virus

- Changes label of infected disks to BRAIN ☺
- From Pakistan (Believed)
- Sole purpose: to pass the infection
  - Traps disk read interrupts
  - Only interested in reads in the boot sector
- Believed to be *proof-of-concept*
- Many other variants, more *efficient* …

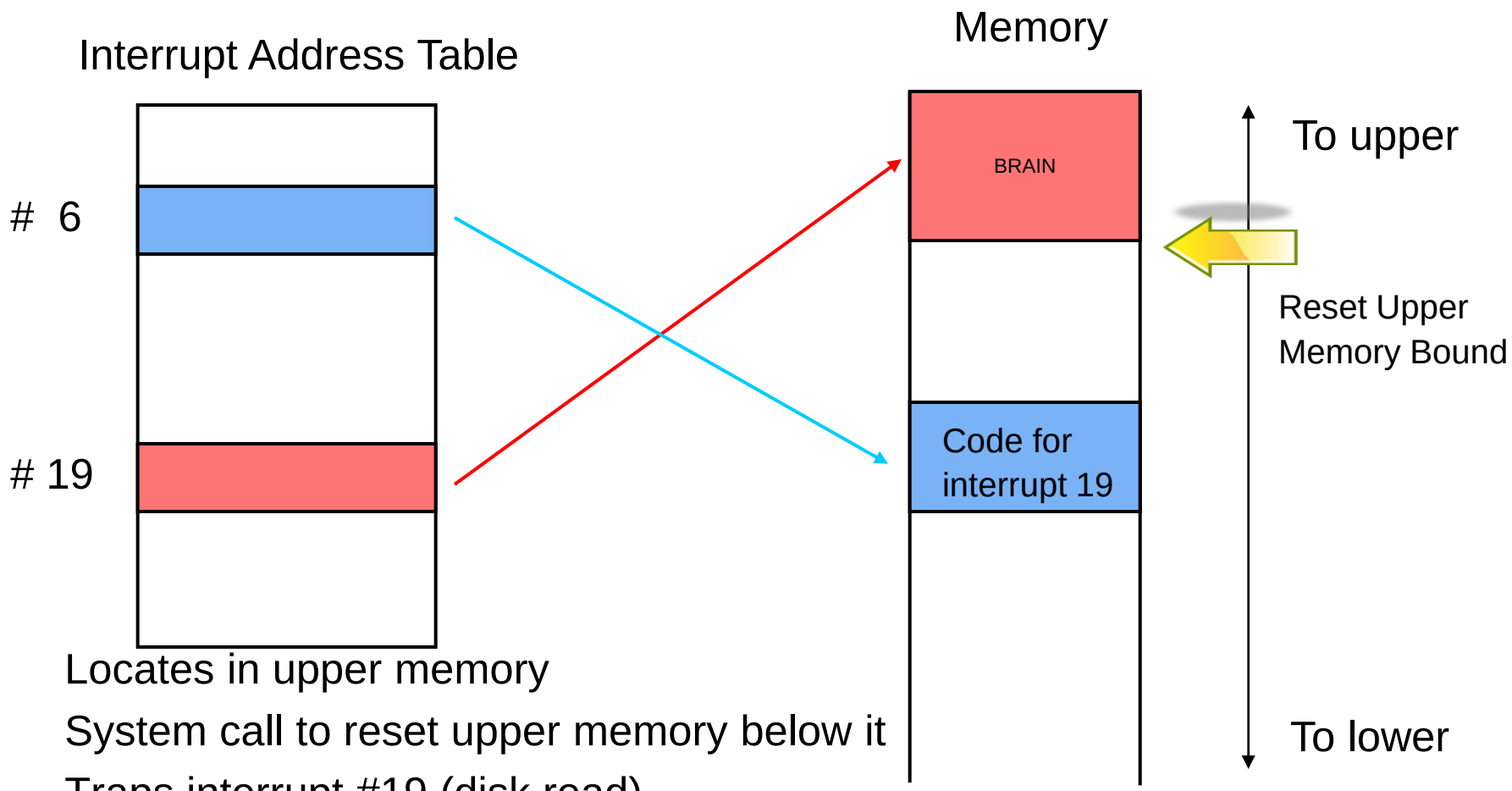
The BRAIN Virus Location

Before

After …



The BRAIN Virus Infection



1. Locates in upper memory
2. System call to reset upper memory below it
3. Traps interrupt #19 (disk read)
4. Any disk read for boot sector returns content of hijacked sector

In this lecture

- Viruses
  - How they attach
  - How they gain control
  - Where they are stored
  - Detection …
- Worms
- Web Bugs
- Trapdoors
- …

Virus Detection

- Based on
  - Virus Signatures
  - Storage Patterns
  - Execution Patterns
  - Transmission Patterns
- Virus scanner uses such patterns to
  - Detect
  - And even remove viruses

Virus signatures

- Virus cannot be completely invisible
  - Code must be stored somewhere
  - Code must be in memory to execute
  - Executes according to a pattern
  - Spreads using certain mechanisms
- Example: Code Red

Signature

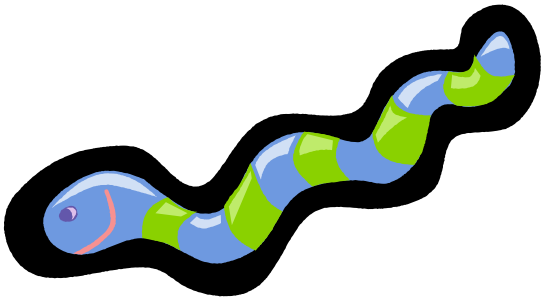
```
GET /default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 %u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801
 %u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3
 %u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a HTTP/1.0
```

In this lecture

- Viruses
- Worms
- Web Bugs
- Trapdoors
- …

Reminder: What is a Worm

- Reproducing programs that run independently and travel across network connections.
- Reproduction Differences:
  - A virus is dependent upon a host file or boot sector, and the transfer of files between machines to spread.
  - A worm can run completely independently and spread of its own will through network connections.



Example: The Internet Worm

- November 2nd 1988
- Internet Worm released
  - Infected many computers
  - Many more severed network connection
- Robert T. Morris Jr.
  - $10,000 fine
  - 3 year suspended jail sentence
  - 400 hours community service
  - Now with MIT

Intent of Internet Worm

1. Determine where it could spread
2. Spread to new target
3. Remain undiscovered and undiscoverable

Determine Targets

- Exploited three known vulnerabilities

1. Find user accounts to invade on target system
   - Remember password vulnerabilities
   - 432 common passwords + dictionary file
2. Fingerd: daemon which responds to queries about users
   - Known buffer overflow vulnerability
   - Give worm a remote shell
3. Sendmail trapdoor
   - In debug mode, sendmail can execute input string

Spread Infection

- Send a bootstrap loader to target machine
  - 99 lines of C code
  - Compile and execute on target machine
  - Fetch rest of worm code from the sending system
- Element of good security ☺
  - Bootstrap loader required to provide password to sending system
  - If fail, sending system breaks connection

Remain Undiscovered

1. If transmission error occurs during worm fetch
   - Bootstrap loader removes code and exits
2. Bring all worm code in memory
   - Encrypt copy in memory
   - Delete copy from disk
   - Thus, the worm cannot easily be discovered
3. Periodic change of name and process id
   - Avoid single process running a long time

Effect of Internet Worm

1. Resource exhaustion
   - If target was already infected, propagate one copy
   - Bug in code (☺): many copies did not terminate !
   - Thus, serious performance degradation
2. Disconnection of machines from Internet
   - To prevent copies from trying to propagate
   - … or to prevent infection
3. Isolation and inability to perform work
   - Estimated cost $100,000 - $97 million
   - Thousands of systems were disconnected

What do we cover

- Viruses
- Worms
- Web Bugs
- Trapdoors
- Salami Attack
- Rootkits
- Privilege Escalation
- Keystroke Logging
- Covert Channels

Web Bugs

- Pixel tag, clear gif/one-by-one/invisible gif
- Part of a web page
  - Invisible to user
  - Track activities of the user
  - Plants a *cookie* in your computer

Cookies

- Set by web sites
  - To push storage from web sites to user platform
- Have 6 fields
  - (name, value, expiration, path to server, server domain, SSL-req?)
- Used to remember values for subsequent usage
  - (“visa credit card”, 1234 1234 1234 1234, …)
  - (“user id”, carbunar, …)
  - (“password”, ****, …)
- Used to build browsing profile
  - (“visits for www.abc.com”, 10, …)

Web Bugs (cont'd)

- Can build a profile for the user containing
  - Surfing habits
  - Personal data: name, DOB, address, IP address, etc
- Can be used for good or bad purposes
- How ?

What do we cover

- Viruses
- Worms
- Web Bugs
- Trapdoors
- Salami Attack
- Rootkits
- Privilege Escalation
- Keystroke Logging
- Covert Channels

Trapdoors

- Undocumented entry point to a software module
  - For testing purposes
  - For future updates
  - For access in case of future failures

Trapdoor: Example 1

- Hidden trap door in Linux, Nov 2003
  - Allows attacker to take over a computer
  - Practically undetectable change
  - Uncovered by anomaly in CVS usage
- Inserted line in wait4()

```
if ((options == (__WCLONE|__WALL)) && (current->uid = 0))
            retval = -EINVAL;
```

- Looks like a standard error check
- Anyone see the problem?

See: http://lwn.net/Articles/57135/

Trapdoor: Example 2

- Rob Harris case - slot machines
  - Insider: worked for Gaming Control Board
- Malicious code in testing unit
  - When testers checked slot machines
    - Downloaded malicious code to slot machine
  - Was never detected
  - *Special sequence of coins activated "winning mode"*
- Caught when greed sparked investigation
  - $100,000 jackpot

Trapdoor: Example 3

- Breeder's cup race
  - Upgrade of software to phone betting system
  - Insider, Christopher Harn, rigged software
- Allowed him and accomplices to call in
  - *Change the bets that were placed*
  - Undetectable
- Caught when got greedy
  - Won $3 million

http://horseracing.about.com/library/weekly/aa110102a.htm

Trapdoors (cont'd)

- Reason for persistence: developers
  - Forget to remove them
  - Leave them for testing
  - Leave them for maintenance
  - Leave them as covert means of access to component

What do we cover

- Viruses
- Worms
- Web Bugs
- Trapdoors
- Salami Attack
- Rootkits
- Privilege Escalation
- Keystroke Logging
- Covert Channels

Salami Attack

- Perform many inconsequential operations with powerful final results
- Example 1 – bank interest
  - Account has $102.87
  - Interest rate is 6.5% per year
  - After one month interest = 31/365 * 0.065 * 102.87 = $0.5495726
  - Round to 0.54 instead of 0.55: *nobody notices !*
  - $0.0095726 goes into attacker's account
- Example 2 – steal a few cents from each account
  - Who checks balance ?

What do we cover

- Viruses
- Worms
- Web Bugs
- Trapdoors
- Salami Attack
- Rootkits
- Privilege Escalation
- Keystroke Logging
- Covert Channels

Rootkits

- Virus variation
  - Attempts to operate as *root*
  - While staying undiscovered
  - And attempting to reinstall itself if removed
- How to go undiscovered ?
  - If the system call is ls or dir

  1. Intercept system call result
  2. Remove itself from the list
  3. Adjust sizes of other files so free space seems legitimate

Sony XCP Rootkit

- Prevents users from copying music CDs
  - Allows them to play music
  - Has its own music player
  - Garbles the results of any other access to CD
- Installs with first insertion of CD
  - Due to *autorun* feature of Microsoft
- To conceal existence, XCP
  - Blocks display of any program starting with sys
  - Including a virus called sysvirus-1 …

What do we cover

- Viruses
- Worms
- Web Bugs
- Trapdoors
- Salami Attack
- Rootkits
- Privilege Escalation
- Keystroke Logging
- Covert Channels

Privilege Escalation

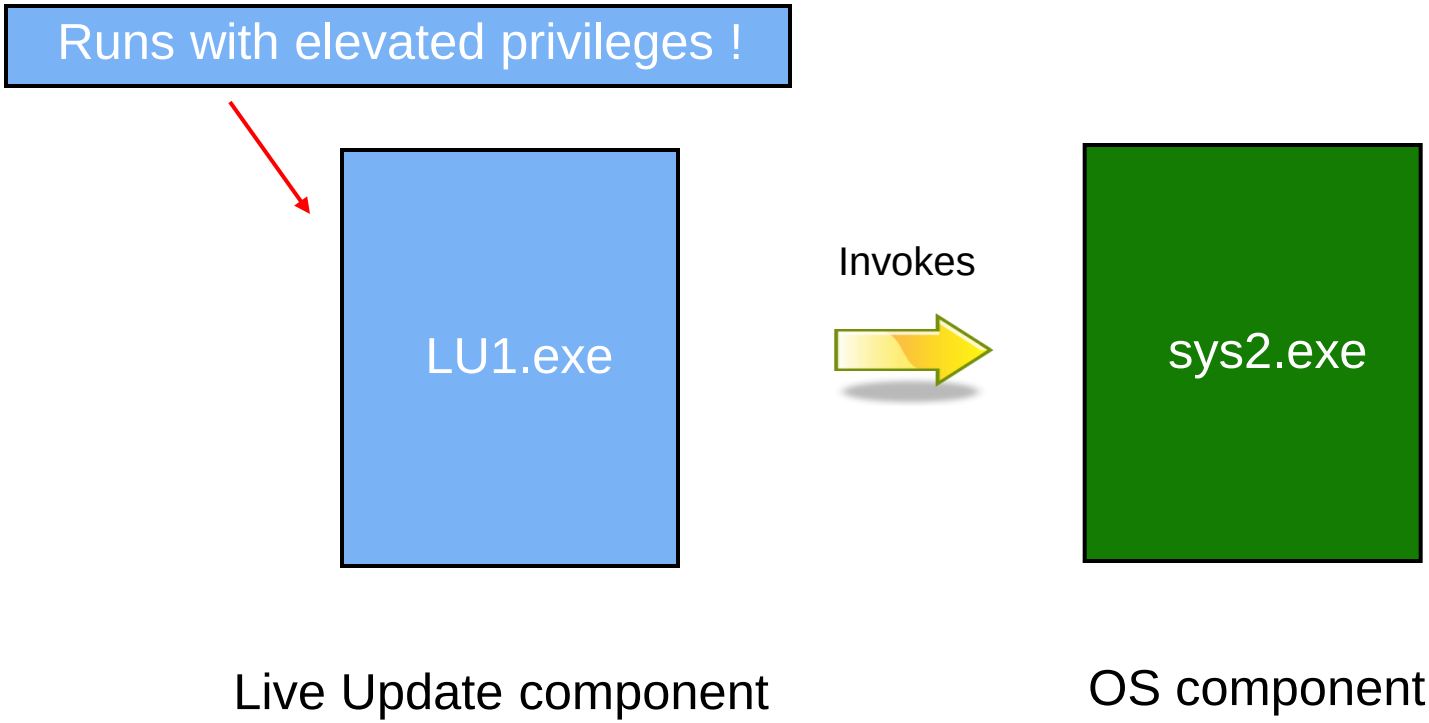
- Assume malicious code has installed on your account
  - Can run processes with your capabilities
  - Can access all your resources
  - … But not other resources
- Attacker wants to run with superuser capabilities
  - To access system resources
  - Or other user's resources

Privilege Escalation Example

- Symantec: software security company
  - Virus scanners, spam filters, system integrity tools …
- Has a Live Update feature
  - Ensure you are running the latest version
  - Periodic or manual invocation
  - Runs with elevated privileges
  - Installs programs in system directory

Example (cont'd)

- Assume Live Update consists of two components



Search Path

- Search Path:
  - Mechanism allowing OS to find program to execute
  - E.g., Path = C:\program files\symantec (LU1.exe)
  - *OS uses first instance found in path*
- User can specify the search path (add/remove)

Privilege Escalation Attack

1. Attacker has infected user account
2. Attacker creates its own sys2.exe version
   - D:\Documents\sys2.exe
   - Path = D:\Documents\::C:\program files\symantec
3. Launch Live Update
   - When LU1 invokes sys2.exe
   - The OS uses first sys2.exe instance in search path
   - Which runs with elevated privileges

What do we cover

- Viruses
- Web Bugs
- Trapdoors
- Salami Attack
- Rootkits
- Privilege Escalation
- Keystroke Logging
- Covert Channels

Keystroke Logging

Key press

Keyboard

Signal

Device driver

Pass along

Word processor

Log keys

Keylogger

- Types of keyloggers
  - Independent – log of all keystrokes
  - Tied to a certain app – log only keystrokes for banking app

What do we cover

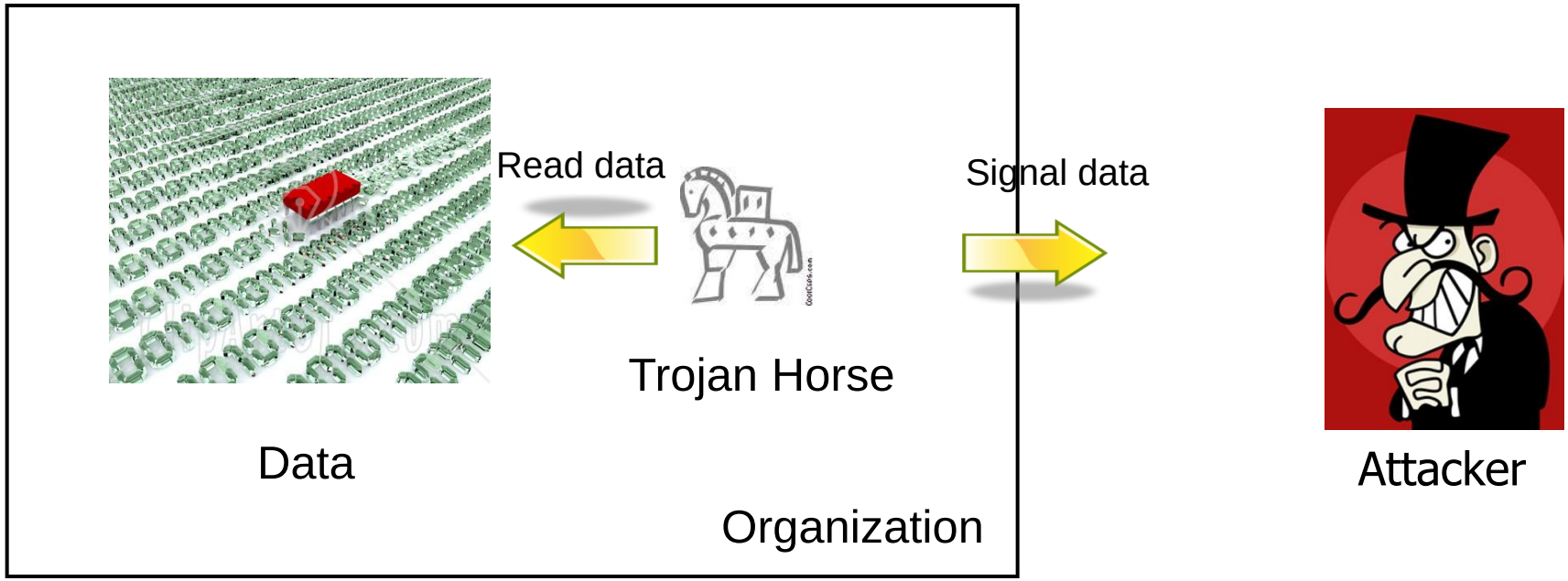
- Viruses
- Worms
- Web Bugs
- Trapdoors
- Salami Attack
- Rootkits
- Privilege Escalation
- Keystroke Logging
- Covert Channels

Covert Channels: Intro

- Communicate information surreptitiously
- How ?
  - Use existing communication channels to hide information
- Example: cheating students ☺
  - One student knows the material
  - Four types of answers: a, b, c, d
  - Cough for a, yawn for b, …
- Problem !
  - If student looses track, it may get the answer for wrong question

Covert Channels: The Problem

- Attacker needs access to data



Attacker cannot simply send the data (even encrypted)

Why ?

Covert Channels: How To

- Modify existing communication in slight ways
  - Assumes attacker and trojan horse share a code
- Example:
  - Change header of file:
    - Word TOTAL implies bit=0
    - Word TOTALS implies bit=1
  - Adding spaces
  - Modify last digit in insignificant field
  - Use of . instead of :

Storage Channels

- Pass information using presence of absence of objects in storage
- Assumes attacker and trojan horse
  - Divide time into intervals
  - Are time synchronized

Storage Channels: Example

- File lock channel
  - Used to provide atomic operations on file
  - Trojan and attacker share a file
  - In each interval
    - If bit=1 then trojan locks the file
    - Attacker tries to access file. If fail, bit=1
- Disk quota
  - If bit=1, trojan creates large file, otherwise does nothing
  - Attacker tries to create file; if not able, bit=1

Timing Channel

- Trojan
  - bit = 1 $\Rightarrow$ enter computation intensive loop
  - bit = 0 $\Rightarrow$ go to sleep
- Attacker
  - perform a task with known computational requirements
  - if completed quickly then bit = 0 otherwise bit = 1

Summary

- Malicious code – Malware
  - What attackers can do with vulnerabilities
- Many other attacks- the list is by far incomplete
- Have created significant damage
- Why a problem
  - Good code is hard to write
  - Patching vulnerabilities is error prone
  - Not everyone does it
  - Cat-and-mouse game between attackers and defense