Operating Systems and Security: Security

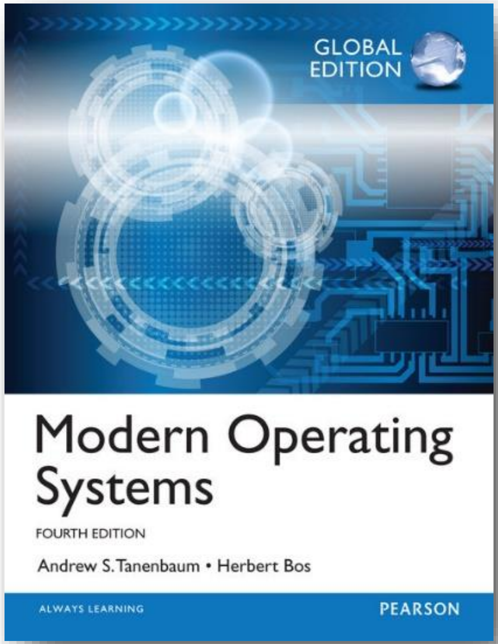
Bruno da Silva

E-mail: bdasilva@etrovub.be

CURSUS MATERIAL

SECURITY

- Protection
- Attacks
- Malware
- Defenses



- Reference Book:
- **A. S. Tanenbaum, "Modern Operating Systems," Pearson, Chapter 9**
  (Cryptography or Authentication part are covered in another course)

ETRO
ELECTRONICS &
INFORMATICS

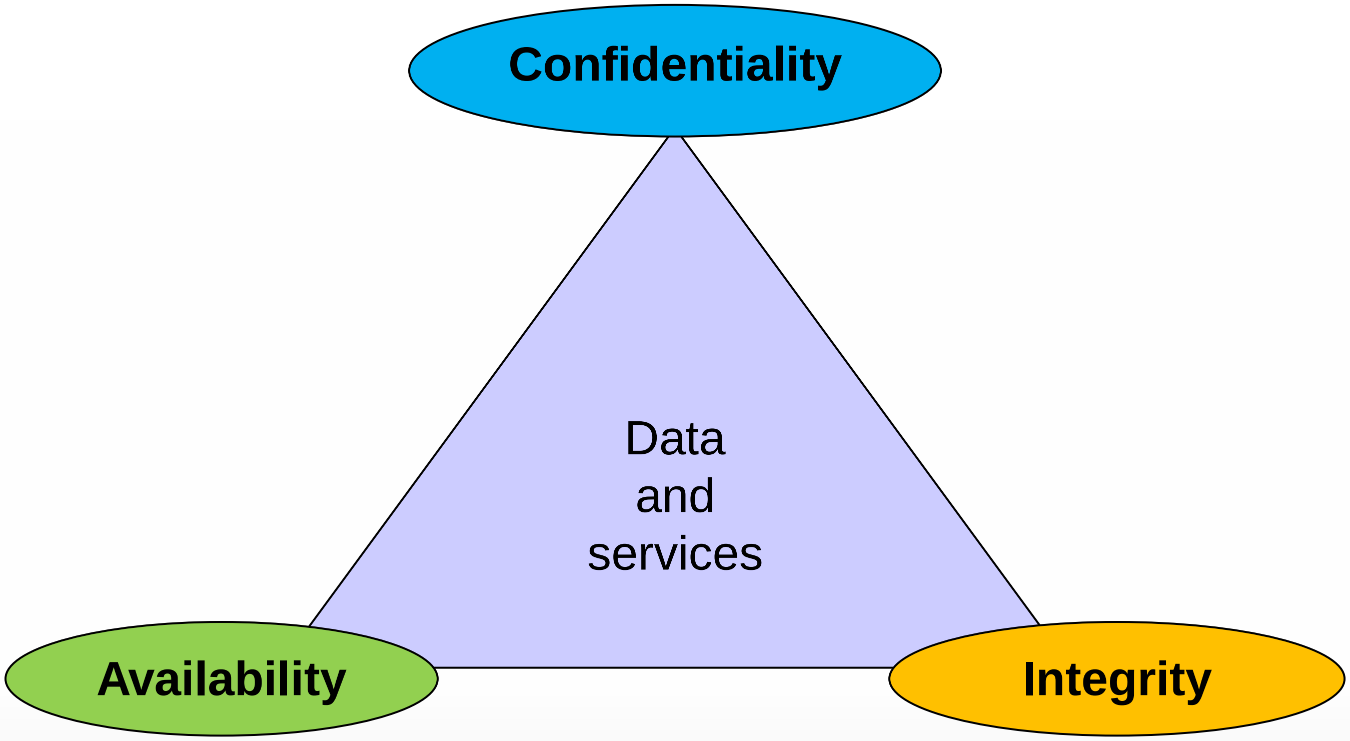
OPERATING SYSTEMS

SECURITY – PART 1: PROTECTION

COMPUTER SECURITY TRIAD

THREADS

Three key objectives are at the heart of computer security



CAN WE BUILD SECURE SYSTEMS?

Two questions concerning security:

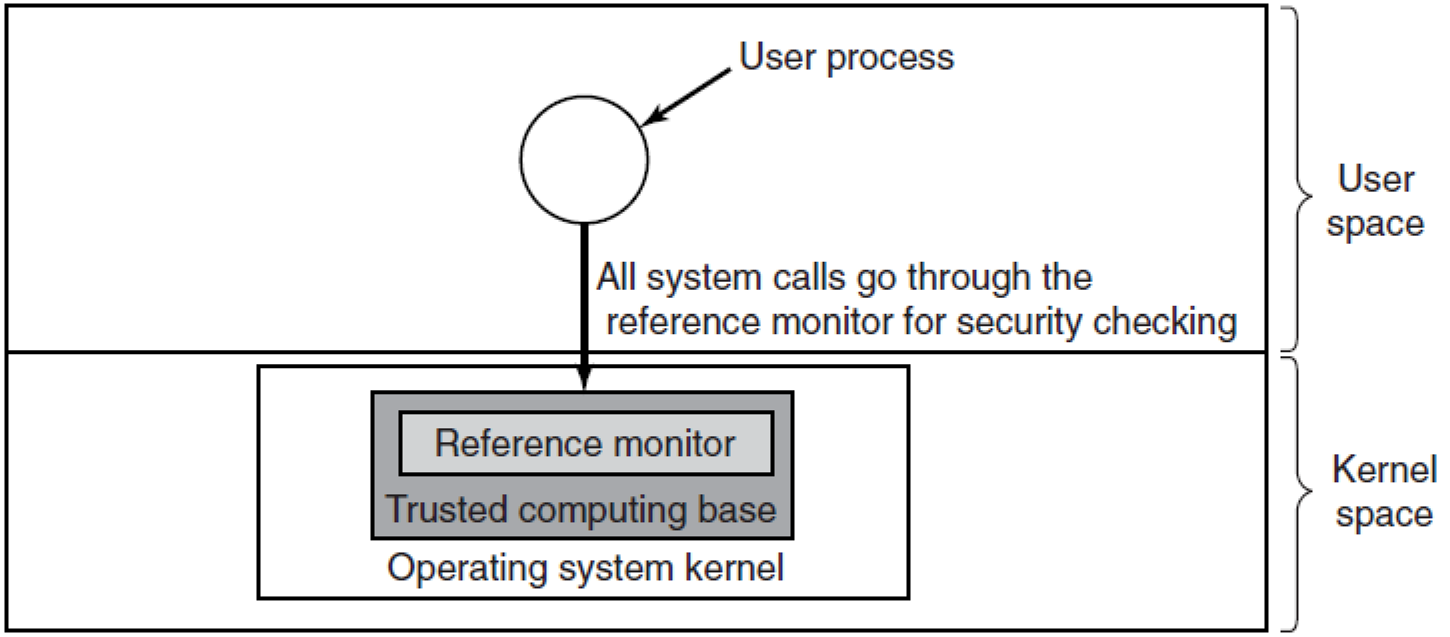
1. Is it possible to build a secure computer system?
2. If so, why is it not done?

TRUSTED COMPUTING BASE

DEFINITION

Reference monitor

Accepts all system call security and makes security decisions



PROTECTION DOMAINS

MODEL

Domain is a collection of access rights

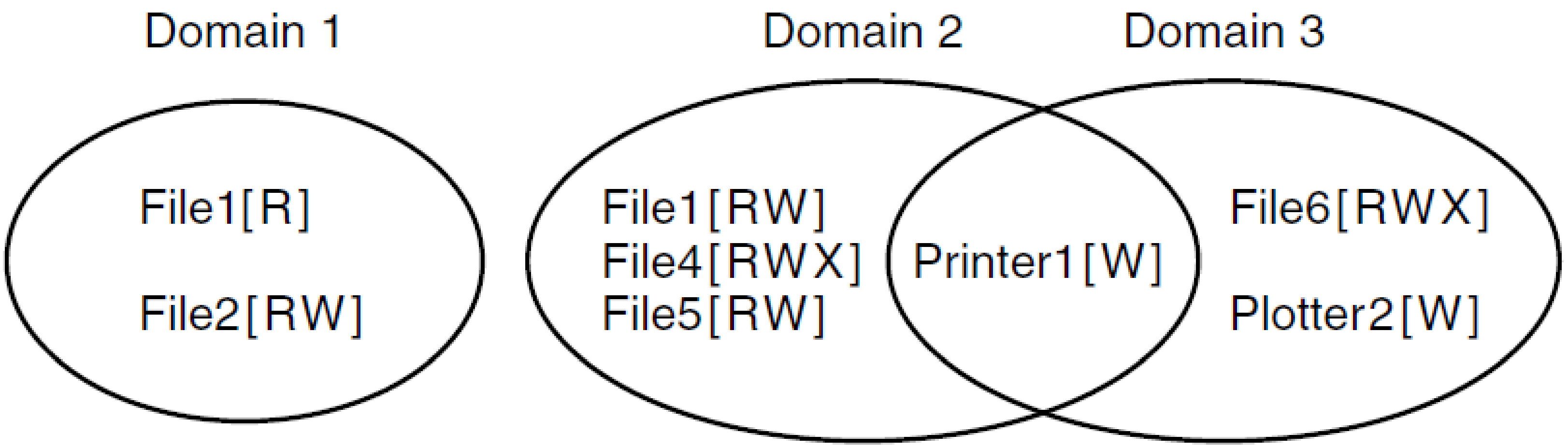
Domain is a set of (object, rights) pairs.

- Each pair specifies an object and some operation that can be preformed on it.
- Right, in this context, means the permission to perform one of the operations.

PROTECTION DOMAINS

EXAMPLE

Three protection domains.



ETRO ELECTRONICS & INFORMATICS

CONTROLLING ACCESS TO RESOURCES

PRINCIPLE OF LEAST AUTHORITY/PRIVILEGE

Principle of Least Authority: Security works best when each domain has the minimum objects and privileges to do its work and no more.

At each abstraction layer, every element should be able to access only the resources necessary to perform its task.

PROTECTION DOMAINS

A PROTECTION MATRIX

Domain	File1	File2	File3	File4	File5	File6	Printer1	Plotter2
1	Read	Read Write						
2			Read	Read Write Execute	Read Write		Write	
3						Read Write Execute	Write	Write

Object

ETRO ELECTRONICS & INFORMATICS

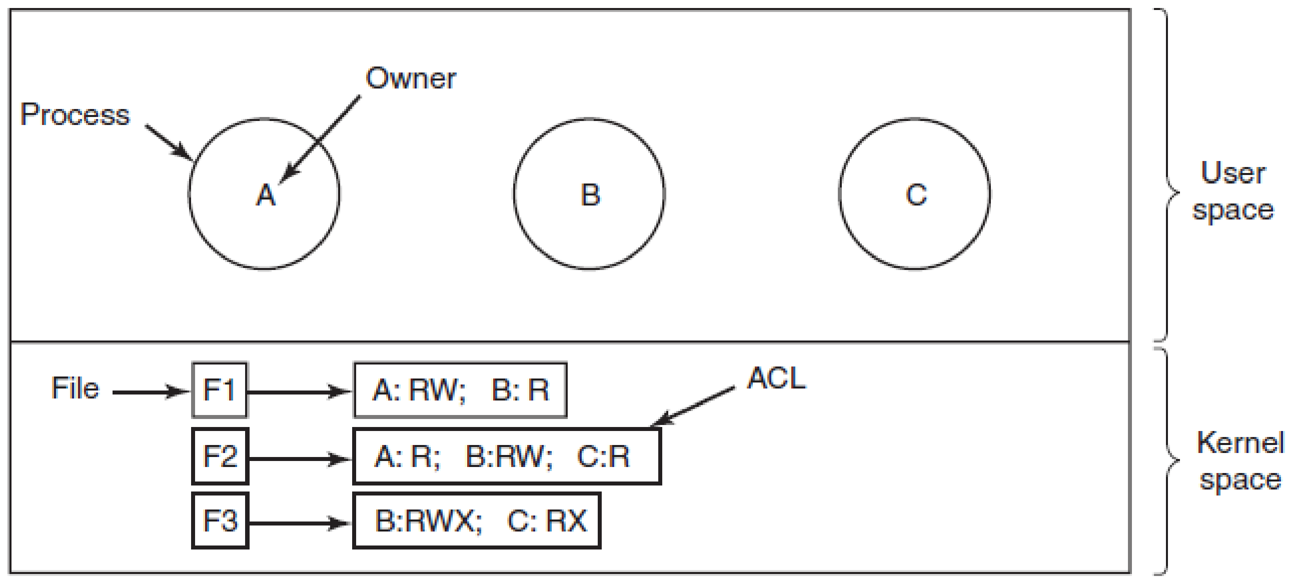
TWO METHODS OF IMPLEMENTATION

- Access Control lists
- Capabilities list

CONTROLLING ACCESS TO RESOURCES

ACCESS CONTROL LISTS

Access control lists associate each object with an ordered list containing all the domains that may access the object and how.



Many systems support the concept of a group of users. Groups have names and can be included in ACLs.

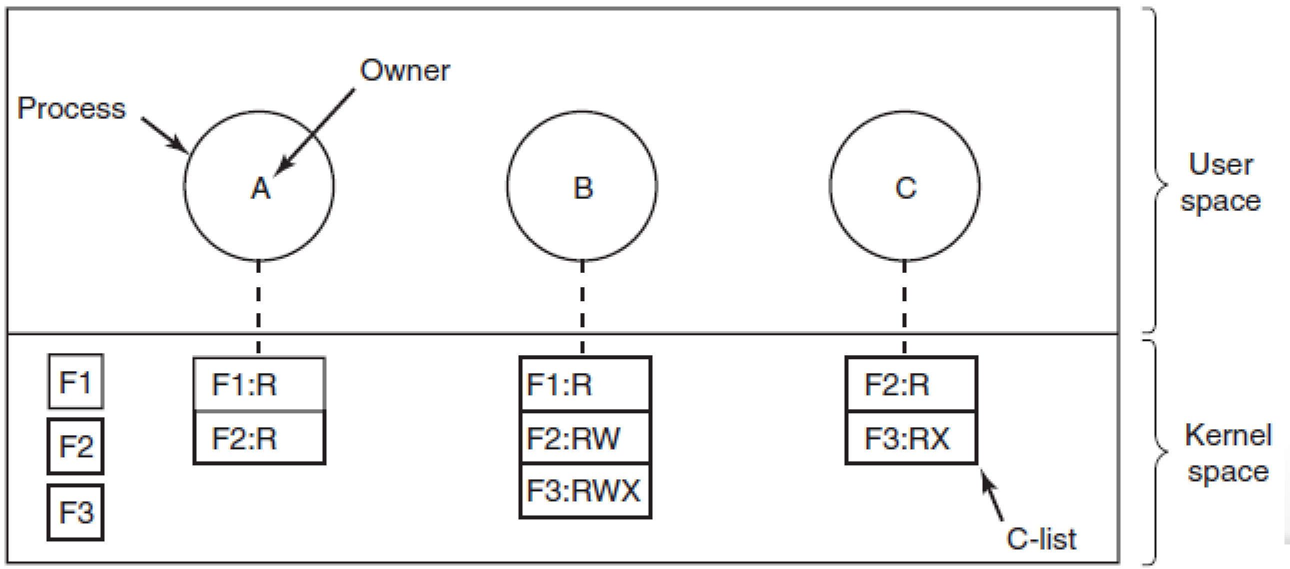
CONTROLLING ACCESS TO RESOURCES

CAPABILITIES

When capabilities are used, each process has a capability list.

Each element of this list is called capability, which grants the owner certain rights on a certain object.

Capability lists are themselves objects and may be pointed to from other capability lists, thus facilitating sharing of subdomains.



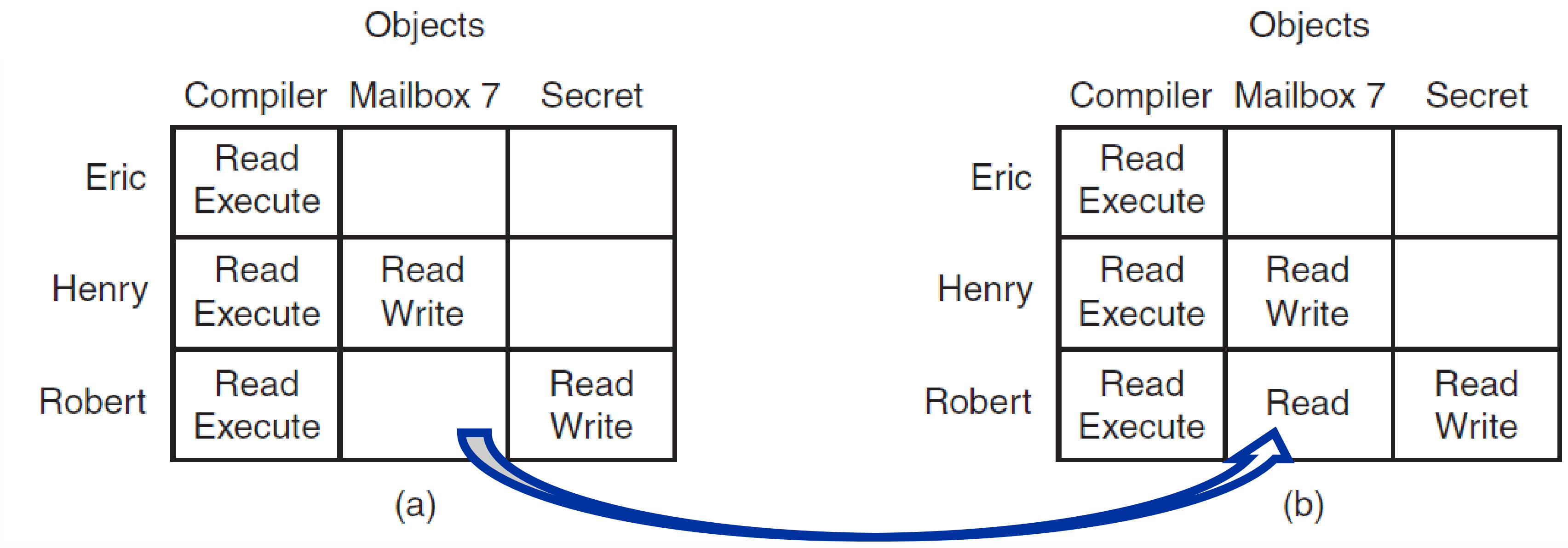
ETRO ELECTRONICS & INFORMATICS

FORMAL MODELS OF SECURE SYSTEMS

PROTECTION MATRICES ARE NOT STATIC

(a) An authorized state.

(b) An unauthorized state.



(a)

	Compiler	Mailbox 7	Secret
Eric	Read Execute		
Henry	Read Execute	Read Write	
Robert	Read Execute		Read Write

(b)

	Compiler	Mailbox 7	Secret
Eric	Read Execute		
Henry	Read Execute	Read Write	
Robert	Read Execute	Read	Read Write

Given an initial authorized state and a set of commands, can it be proven that the system can never reach an unauthorized state?

MULTILEVEL SECURITY

Discretionary Access Control

- Operating systems allow individual to determine who may read and write their files

Mandatory Access Control

- Organizational rules state who can see and modify what.

BELL-LAPADULA MODEL

Bell-LaPadula Model rules for information flow:

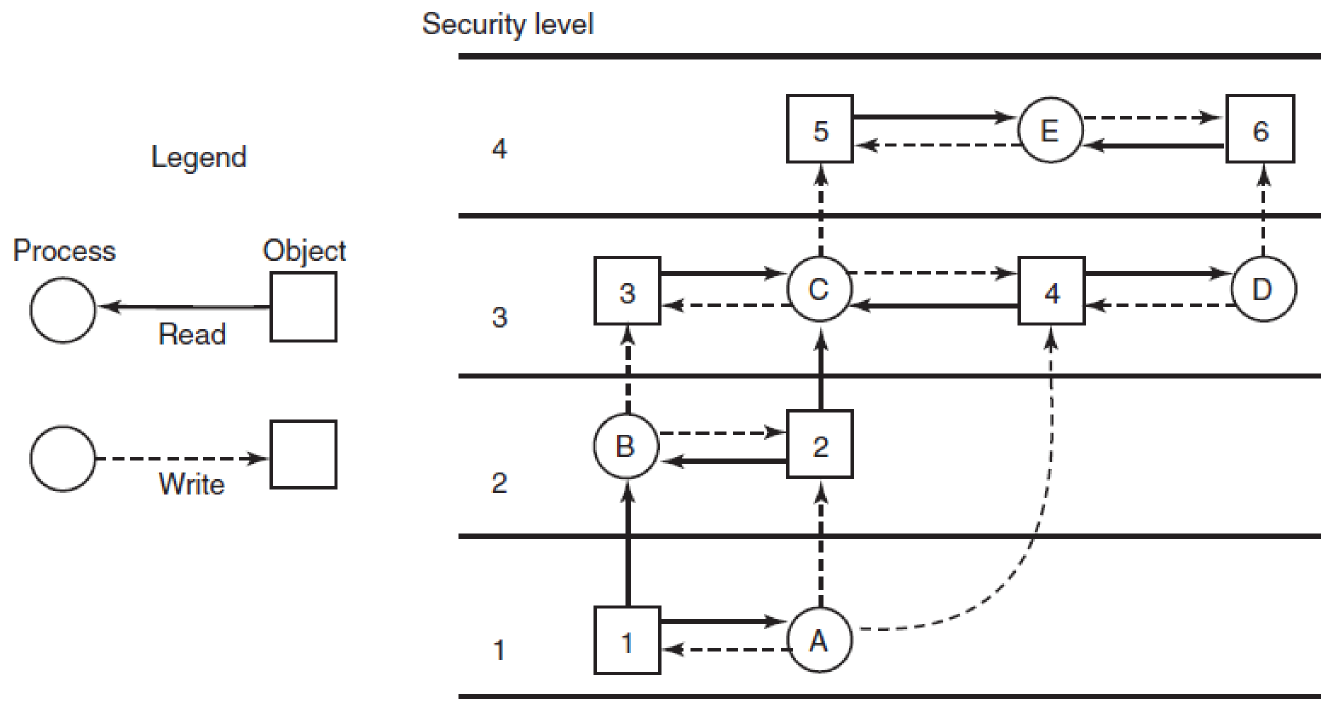
1. The simple security property
   - Process running at security level *k* can read only objects at its level or lower
2. The * property
   - Process running at security level *k* can write only objects at its level or higher

Fun Fact: The * property was so named because in the original report, the authors could not think of a good name for it and used * as a temporary placeholder until they could devise a better name. They never did and the report was printed with the *.

MULTILEVEL SECURITY

BELL-LAPADULA MODEL

The Bell-LaPadula multilevel security model.



The Bell-LaPadula multilevel security model was devised to keep secrets, not guarantee the integrity of data.

THE BIBA MODEL

To guarantee the integrity of the data:

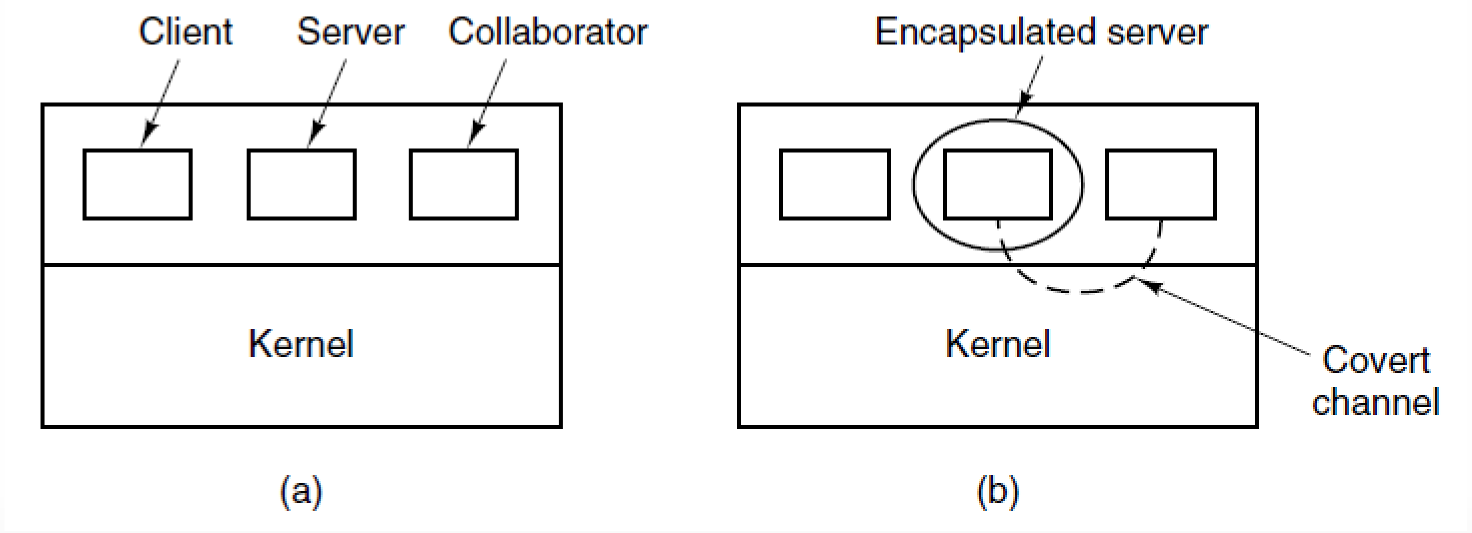
1. The simple integrity principle
   - process running at security level *k* can write only objects at its level or lower (no write up).
2. The integrity * property
   - process running at security level *k* can read only objects at its level or higher (no read down).

BUT… security leaks still can occur

COVERT CHANNELS

Lampson's model, showing how information leaks, was originally formulated in terms of a single timesharing system, but the same ideas can be adapted to applications running in the cloud.

(a) The client, server, and collaborator processes. (b) The encapsulated server can still leak to the collaborator via covert channels.



COVERT CHANNELS (2)

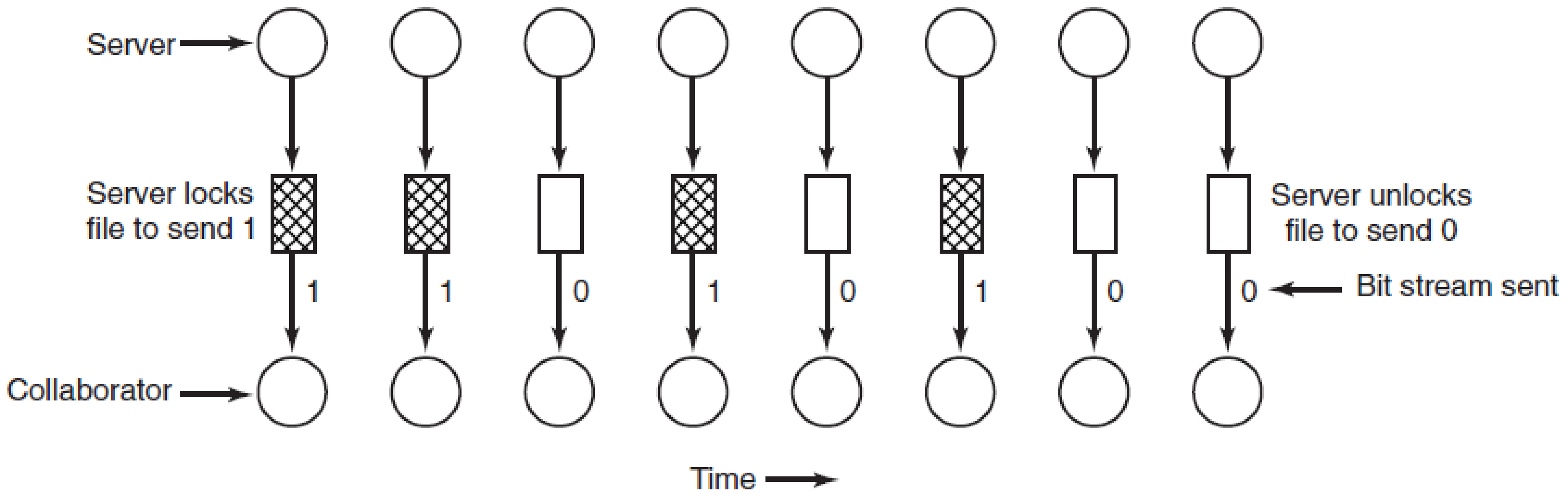


Figure 9-13. A covert channel using file locking.

STEGANOGRAPHY

Concealing secrets in plain site

- Hiding secret information in paintings, jpegs, MP3 files… or messages… or the network (e.g. Watermarking)
- Generic description of the pieces of the steganographic process:
  - cover_medium + hidden_data + stego_key = stego_medium

STEGANOGRAPHY

EXAMPLE

The German Embassy in Washington, DC, sent these messages in telegrams to their headquarters in Berlin during World War I (Kahn 1996).

PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY.

Do you see hidden the message?

STEGANOGRAPHY

EXAMPLE

The German Embassy in Washington, DC, sent these messages in telegrams to their headquarters in Berlin during World War I (Kahn 1996).

PRESIDENT'S EMBARGO RULING SHOULD HAVE IMMEDIATE NOTICE. GRAVE SITUATION AFFECTING INTERNATIONAL LAW. STATEMENT FORESHADOWS RUIN OF MANY NEUTRALS. YELLOW JOURNALS UNIFYING NATIONAL EXCITEMENT IMMENSELY.

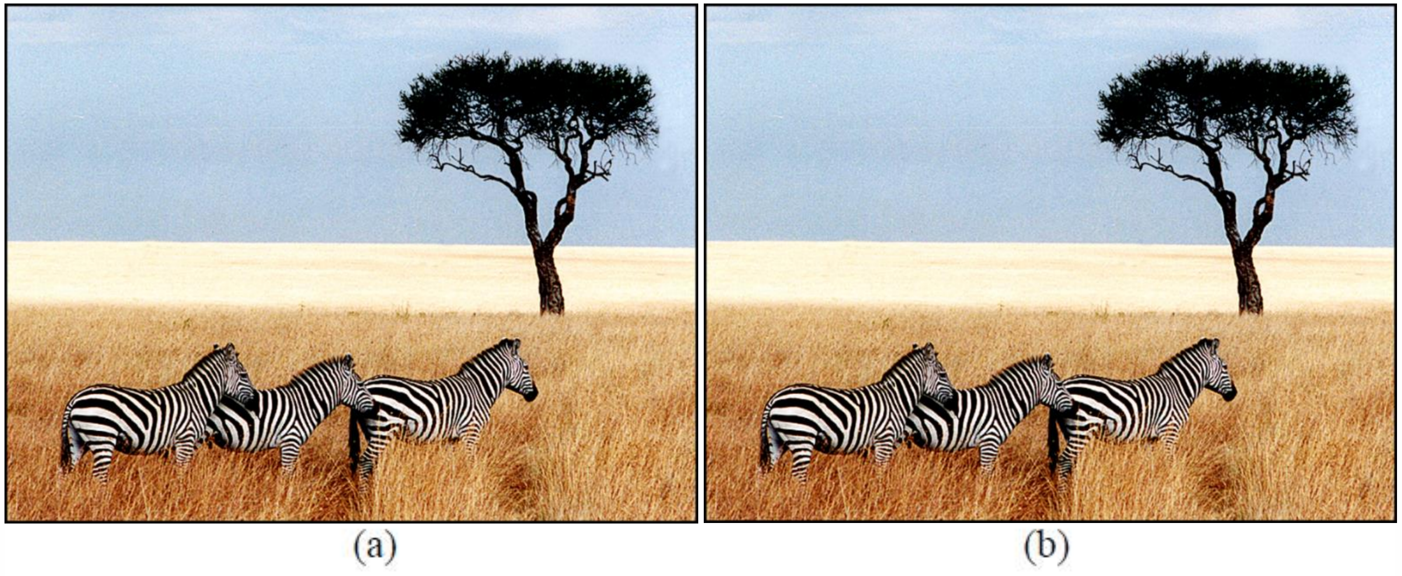
PERSHING SAILS FROM N.Y. JUNE 1

More examples: https://www.garykessler.net/library/steganography.html

STEGANOGRAPHY

EXAMPLE

- Pictures appear the same
- Picture on right has text of 5 Shakespeare plays
  - encrypted, inserted into low order bits of color values



a) Three zebras and a tree.

b) Three zebras, a tree, and the complete text of five plays by William Shakespeare.

Demo: http://www.cs.vu.nl/~ast/books/mos2/zebras.html

ETRO
ELECTRONICS &
INFORMATICS

OPERATING SYSTEMS

SECURITY – PART 2: ATTACKS

WHAT WE ARE COVERING

- Exploiting Software
- Insider Attacks

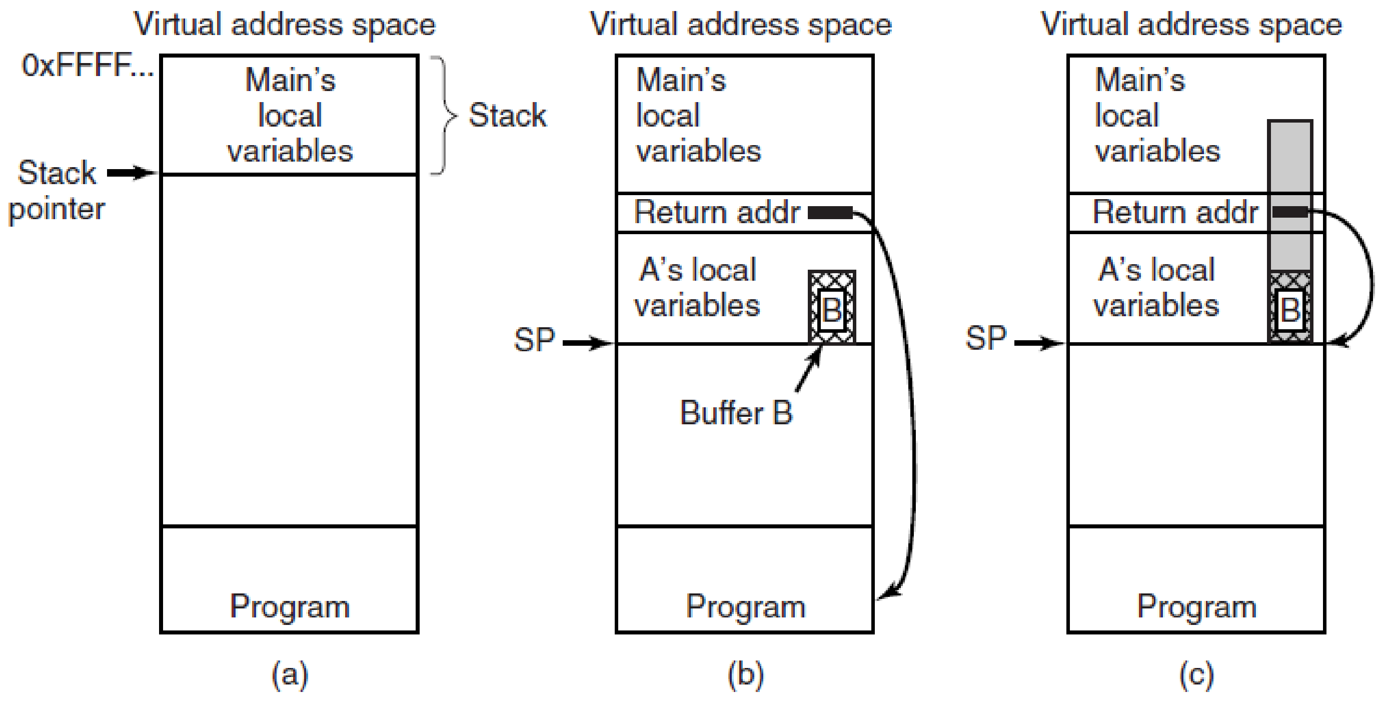
EXPLOITING SOFTWARE

TYPES

- Buffer overflow attacks
  - Control Flow Attacks
  - Non-Control Flow Attacks
- Memory Corruption Attacks
  - Format String Attack
  - Dangling pointers
- Null Pointer Dereference Attacks
- Integer Overflow attacks
- Command Injection Attacks
- Time of Check to Time of Use Attacks

BUFFER OVERFLOW ATTACKS

NO C OR C++ COMPILER DOES ARRAY BOUNDS CHECKING



(a)Situation when the main program is running.

(b) After the procedure A has been called.

(c) Buffer overflow shown in gray.

DEFENSE: STACK CANARIES

Modern computers use digital ‘canaries’ as an early warning system, detecting possible buffer overflow attack.

In the code, at places where the program makes a function call, the compiler inserts code to save a random ‘canary’ value on the stack, just below the return address.

The compiler inserts code at the return to check the canary value.

If the canary has changed… trouble!

DEFENSE: DATA EXECUTION PREVENTION

AGAINST CODE INJECTION ATTACKS

Redefine the ‘real’ problem: The fact that the attacker can inject code and have it executed in the heap or stack!

Defense: prevent the bytes provided by the attacker from being executed as legitimate code.

Modern CPUs have a feature **NX bit** (No-eXecute bit):

- It distinguishes between data segments and the code segments.
- This ensures data segments are writable, but not executable.

A generic name for this security measure is **DEP** (Data Execution Prevention).

CODE REUSE ATTACKS

Attacker constructs the necessary functions out of existing functions and instructions in the existing binaries and libraries

Return to libc

Almost all c (C++) programs are linked with a shared library containing the function *system.*

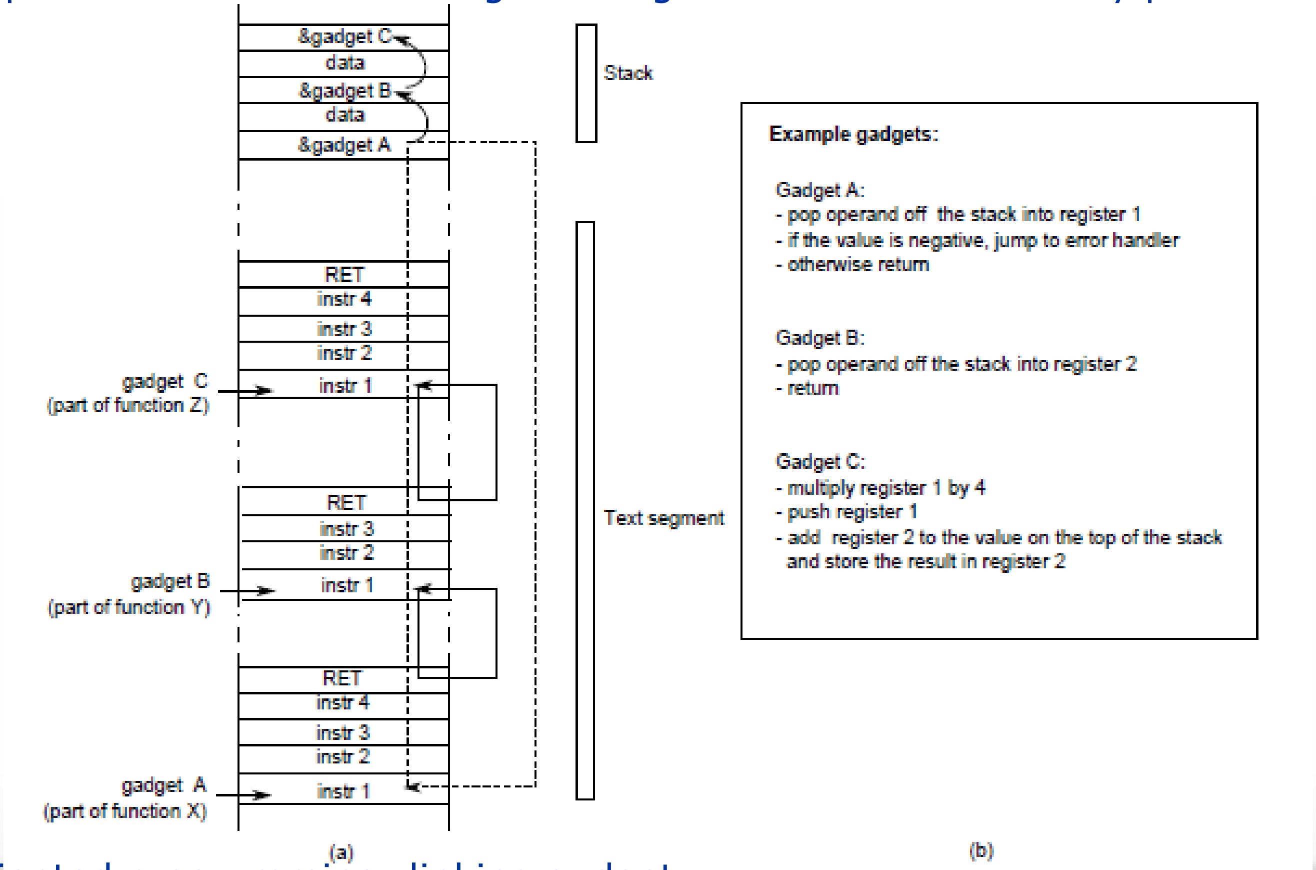
System takes a string containing a command and passes it to the shell for execution.

Attack: Place a string containing commands to be executed and divert control the *system* function via the return address.

CODE REUSE ATTACKS

RETURN-ORIENTED PROGRAMMING (ROP)

The concept of Return-Oriented Programming is to return to the entry points of library functions.



Example gadgets:

Gadget A:
- pop operand off the stack into register 1
- if the value is negative, jump to error handler
- otherwise return

Gadget B:
- pop operand off the stack into register 2
- return

Gadget C:
- multiply register 1 by 4
- push register 1
- add register 2 to the value on the top of the stack and store the result in register 2

Return-oriented programming: linking gadgets

ETRO ELECTRONICS & INFORMATICS

Tanenbaum & Bo, Modern Operating Systems:4th ed., (c) 2013 Prentice-Hall, Inc. All rights reserved.

DEFENSE: ADDRESS SPACE LAYOUT RANDOMIZATION (ASLR)

- Randomize the address of functions and data between every run of the program
- Supported with varying granularity.
- Few apply it to the system kernel

NON-CONTROL FLOW ATTACKS

- Change the data instead of the return addresses!!
- Possible use: change the security credentials granting more or less protection on objects.

```
void A( ) {
    int authorized;
    char name [128];
    authorized = check_credentials (...);  /* the attacker is not authorized, so returns 0 */
    printf ("What is your name?\n");
    gets (name);
    if (authorized != 0) {
        printf ("Welcome %s, here is all our secret data\n", name)
        /* ... show secret data ... */
```

ETRO
ELECTRONICS &
INFORMATICS

FORMAT STRING ATTACKS

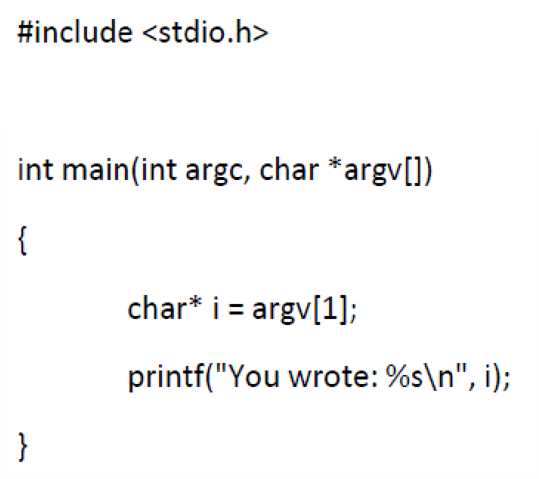
In a C functions that perform formatting, (printf), **when the application doesn't properly validate the input**, the attacker can causes the submitted data to be compromised.

EX: By using exactly the right number of **%08x**, the attacker can use the first four characters of the format string as an address.

FORMAT STRING ATTACKS

EXAMPLE

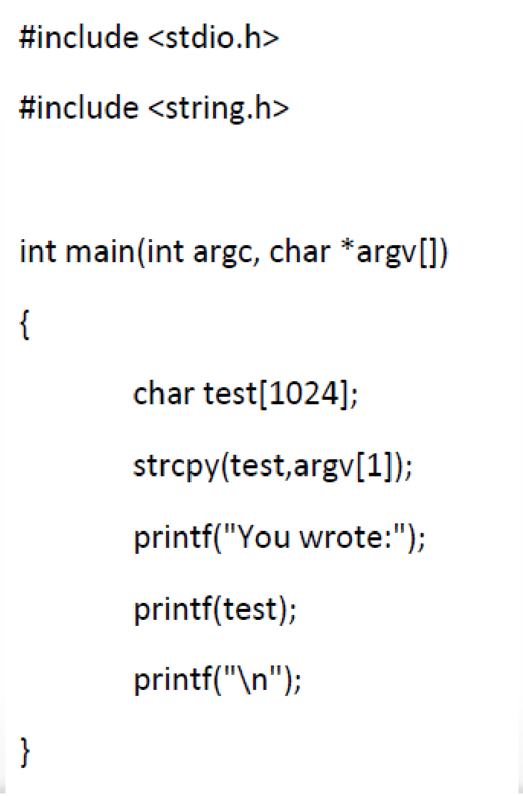
Do you see the problem? Are the codes no equivalent?



```
#include <stdio.h>


int main(int argc, char *argv[])
{
          char* i = argv[1];
          printf("You wrote: %s\n", i);
}
```

Vs



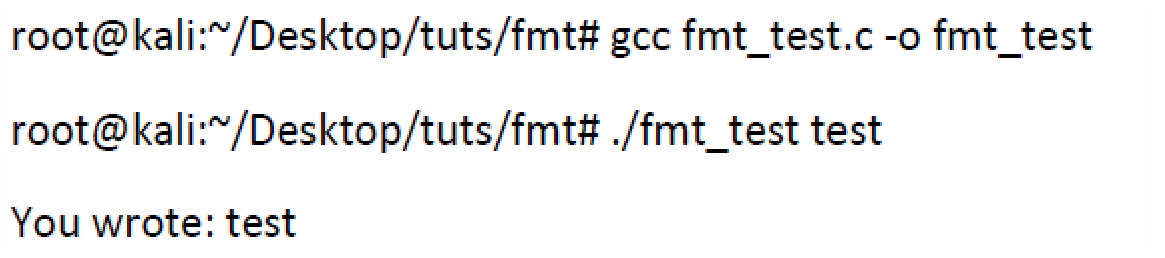
```
#include <stdio.h>
#include <string.h>


int main(int argc, char *argv[])
{
          char test[1024];
          strcpy(test,argv[1]);
          printf("You wrote:");
          printf(test);
          printf("\n");
}
```

FORMAT STRING ATTACKS

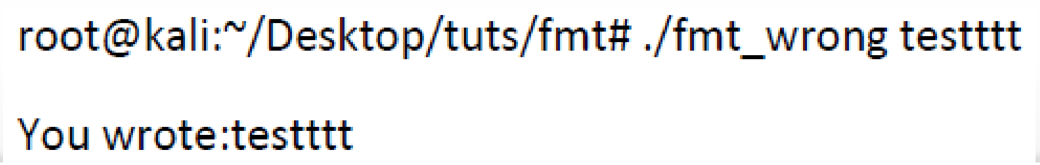
EXAMPLE

So, what happens to the stack when a format string is specified with no corresponding variable on stack?



```
root@kali:~/Desktop/tuts/fmt# gcc fmt_test.c -o fmt_test
root@kali:~/Desktop/tuts/fmt# ./fmt_test test
You wrote: test
```

Vs



```
root@kali:~/Desktop/tuts/fmt# ./fmt_wrong testttt
You wrote:testttt
```

Source: https://www.exploit-db.com/docs/english/28476-linux-format-string-exploitation.pdf

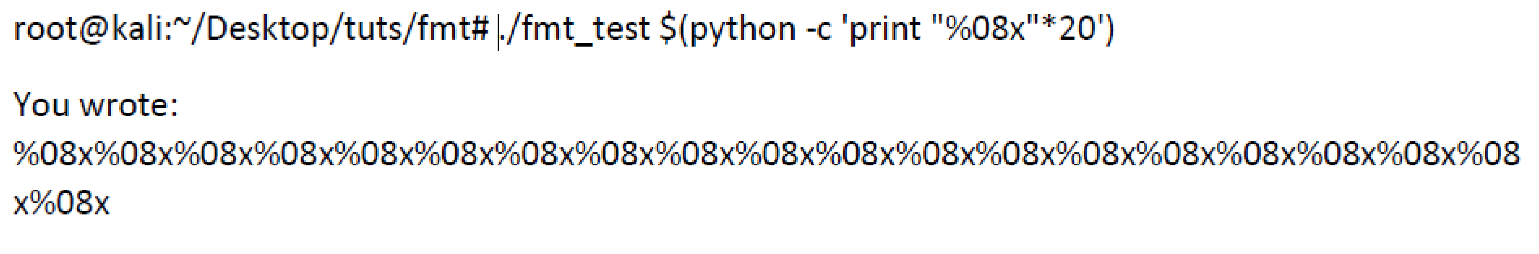
VUB ETRO ELECTRONICS & INFORMATICS

FORMAT STRING ATTACKS

EXAMPLE

So, what happens to the stack when a format string is specified with no corresponding variable on stack?

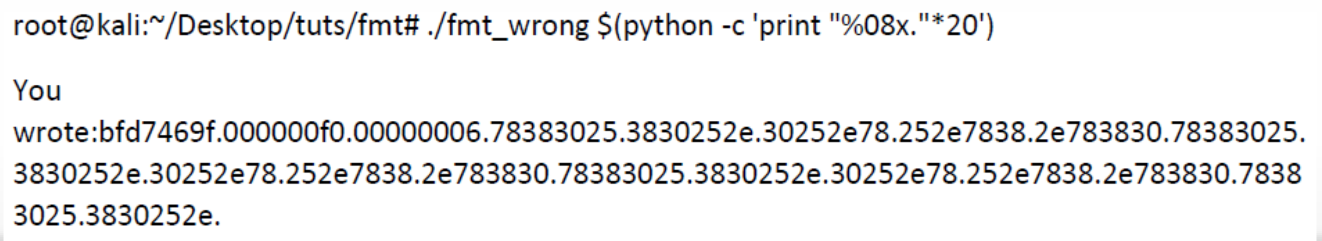
It will start to pop data off the stack from where the variables should have been located.



```
root@kali:~/Desktop/tuts/fmt# ./fmt_test $(python -c 'print "%08x"*20')

You wrote:
%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08x%08
x%08x
```

Vs



```
root@kali:~/Desktop/tuts/fmt# ./fmt_wrong $(python -c 'print "%08x."*20')

You
wrote:bfd7469f.000000f0.00000006.78383025.3830252e.30252e78.252e7838.2e783830.78383025.
3830252e.30252e78.252e7838.2e783830.78383025.3830252e.30252e78.252e7838.2e783830.7838
3025.3830252e.
```

Source: https://www.exploit-db.com/docs/english/28476-linux-format-string-exploitation.pdf

ETRO ELECTRONICS & INFORMATICS

DANGLING POINTERS ATTACK

- User frees memory. But later tries to access it with the pointer.
- Attacker places a specific heap object in the memory location the user frees and re-uses
- Techniques like **heap feng shui** help attackers pull this off.

NULL POINTER DEREFERENCE ATTACKS

In linux, the kernel space is mapped to every process' address space… and whenever the kernel executes it runs in a process's address space.

If a buggy kernel dereferences a NULL pointer, it usually leads to a crash. Or attacker triggers a NULL pointer dereference from the user process.

Crash occurs because there is no code at page 0.

Using a tool like mmap (Posix function that maps files into memory) attacker can map bad code at that location.

Defense: mmap no longer makes it possible to map to page 0.

INTEGER OVERFLOW ATTACK

Integer errors can happen when mathematical operations, and external input lead to a result that is too large to fit within the range of values that can be stored in variables of a given data type.

Can be exploited to corrupt applications.

Integer overflow attacks are possible largely due to incorrectly defining numerical data.

COMMAND INJECTION ATTACKS

- The execution of arbitrary commands on the host operating system via a vulnerable application.

- Command injection attacks are possible largely **due to insufficient input validation** (a complete part of the Defensive Programming approach).

COMMAND INJECTION ATTACKS

EXAMPLE

```
int main(int argc, char *argv[])
{
  char src[100], dst[100], cmd[205] = "cp ";          /* declare 3 strings */
  printf("Please enter name of source file: ");       /* ask for source file */
  gets(src);                                           /* get input from the keyboard */
  strcat(cmd, src);                                    /* concatenate src after cp */
  strcat(cmd, " ");                                    /* add a space to the end of cmd */
  printf("Please enter name of destination file: ");  /* ask for output file name */
  gets(dst);                                           /* get input from the keyboard */
  strcat(cmd, dst);                                    /* complete the commands string */
  system(cmd);                                         /* execute the cp command */
}
```

Suppose user enters for the destination file: ‘file2.txt; rm –rf

Command executed is then: system(“cp file1 file 2; rm –rf”)

Command injection attacks are possible largely due to insufficient input validation.

ETRO ELECTRONICS & INFORMATICS

TIME OF CHECK TO TIME OF USE ATTACK

TOCTOU

Exploits a race condition

Example:

```
int fd;
if (access ("./my_document", W_OK) != 0)
    exit (1);
fd = open ("./my_document", O_WRONLY)
write (fd, user_input, sizeof (user_input));
```

Between the statements, attacker creates a symbolic link with the same file name to the password file

Then user writes password information to attackers file.

INSIDER ATTACKS

- Executed by programmers from within the company.
- Attackers have specialized knowledge
- Insider Attacks
  - Logic Bomb
  - Back Doors
  - Login Spoofing

LOGIC BOMB

A **logic bomb** is a piece of code inserted into an operating system or software application that implements a malicious function after a certain time limit or specific conditions are met.

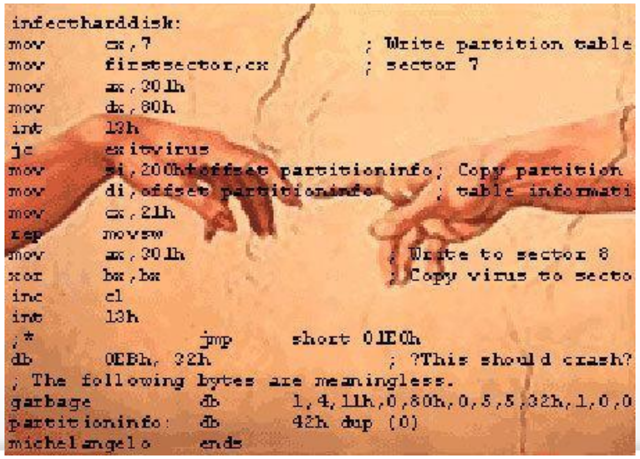
Company programmer writes program:

- potential to do harm
- OK as long as he/she enters password daily
- if programmer fired, no password and bomb explodes

LOGIC BOMB

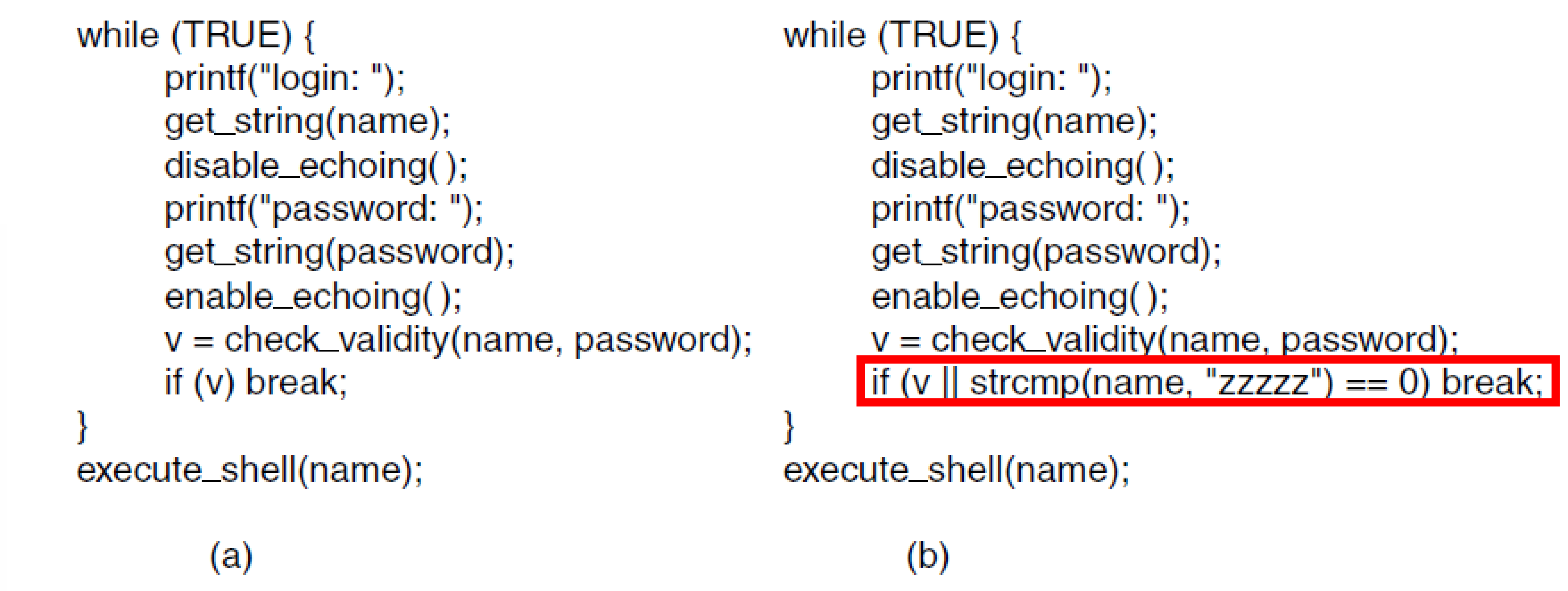
EXAMPLE

- Michelangelo (logic bomb)
  - discovered on 4 February 1991
  - designed to infect DOS systems
  - the virus overwrites the first one hundred sectors of the hard disk with nulls.
  - each year, the virus remained dormant until March 6, the birthday of Renaissance artist Michelangelo.
  - The name was chosen by researchers who noticed the coincidence of the activation date



```
infectharddisk:
mov      cx,7                    ; Write partition table
mov      firstsector,cx          ; sector 7
mov      ax,301h
mov      dx,80h
int      13h
jc       exitvirus
mov      si,200h+offset partitioninfo; Copy partition
mov      di,offset partitioninfo     ; table informati
mov      cx,21h
rep      movsw
mov      ax,301h                     ; Write to sector 8
xor      bx,bx                       ; Copy virus to secto
inc      cl
int      13h
;*            jmp      short 01E0h
db       0EBh, 32h                   ; ?This should crash?
; The following bytes are meaningless.
garbage            db      1,4,11h,0,80h,0,5,5,32h,1,0,0
partitioninfo:     db      42h dup (0)
michelangelo       ends
```

BACK DOORS



```
while (TRUE) {
        printf("login: ");
        get_string(name);
        disable_echoing( );
        printf("password: ");
        get_string(password);
        enable_echoing( );
        v = check_validity(name, password);
        if (v) break;
}
execute_shell(name);
```

(a)

```
while (TRUE) {
        printf("login: ");
        get_string(name);
        disable_echoing( );
        printf("password: ");
        get_string(password);
        enable_echoing( );
        v = check_validity(name, password);
        if (v || strcmp(name, "zzzzz") == 0) break;
}
execute_shell(name);
```

(b)

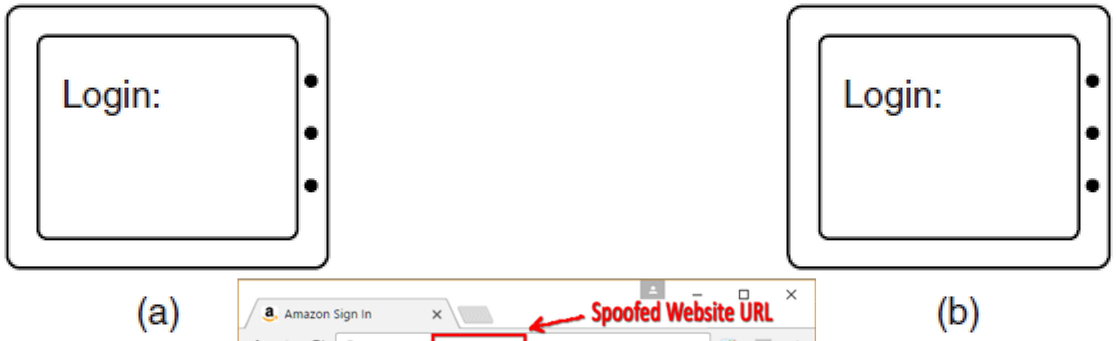
(a) Normal code.

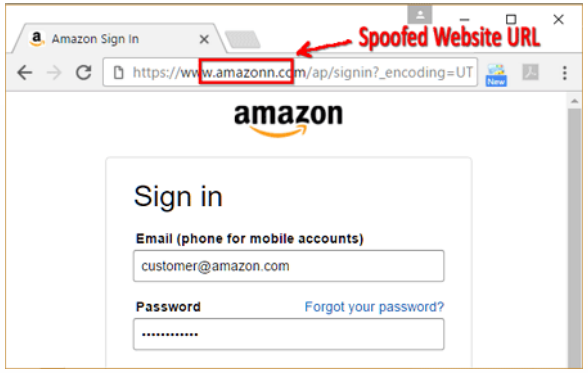
(b) Code with a back door inserted.

Defense: Code reviews

LOGIN SPOOFING

Legitimate user attempting to collect other people's passwords.





(a) Correct login screen.

(b) Phony login screen.

Defense: have a login sequence start with a key combination the user programs can't catch: CTLR-ALT-DEL in Windows cause the current user to log out and system login program started.

ETRO
ELECTRONICS &
INFORMATICS

OPERATING SYSTEMS

SECURITY – PART 3: MALWARE

MALWARE

- Malicious software
  - Trojan horses, virus, worms,.. Etc
- Today’s malware is all about stealth
- Infected machines report back to attacker, its address, information…??
- Attacker uses backdoor to control the infected machine….
  - Make it a *zombie.*
  - A collection of zombies is called a *botnet*

MALWARE

- Criminals can rent out botnets
- Keyloggers
- Identity theft
- Malware can lay in wait for something interesting
- Malware can interfere with competition’s production process
- Malware could target another person in the company to discredit that person

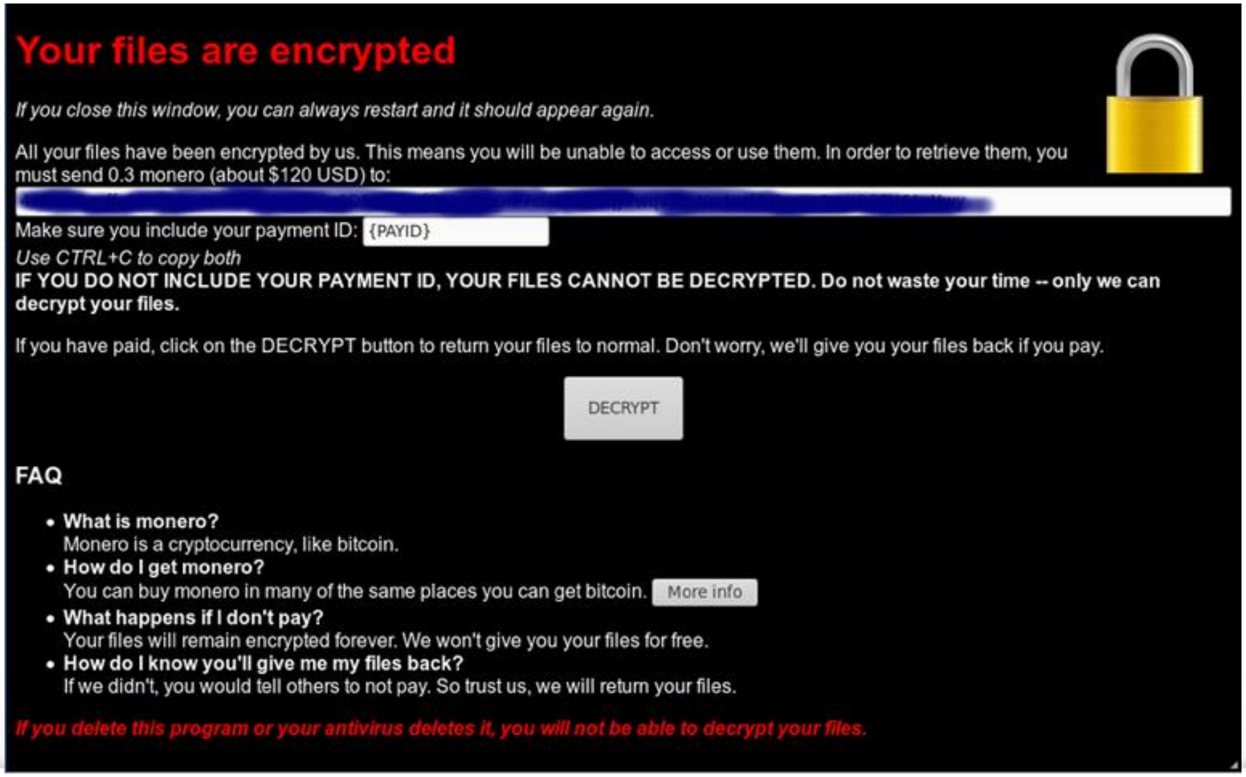
TYPES OF MALWARE

- Ransomware
- Trojan Horse
- Virus
- Worm
- Spyware
- RootKits

RANSOMWARE

Malware used for a form of blackmail. Some famous ones: WannaCry, SamSam…

Example: Encrypts files on victim disk (ransomware), then displays message …



Your files are encrypted

If you close this window, you can always restart and it should appear again.

All your files have been encrypted by us. This means you will be unable to access or use them. In order to retrieve them, you must send 0.3 monero (about $120 USD) to:

[redacted]

Make sure you include your payment ID: {PAYID}

Use CTRL+C to copy both

IF YOU DO NOT INCLUDE YOUR PAYMENT ID, YOUR FILES CANNOT BE DECRYPTED. Do not waste your time -- only we can decrypt your files.

If you have paid, click on the DECRYPT button to return your files to normal. Don't worry, we'll give you your files back if you pay.

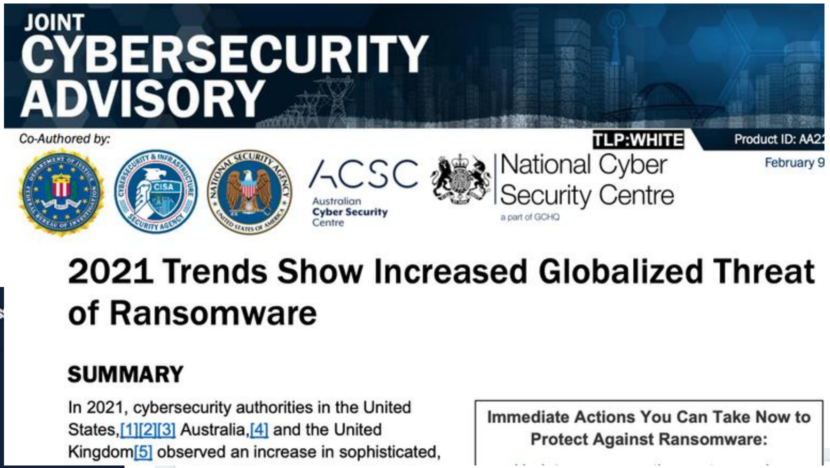
FAQ

- **What is monero?**
  Monero is a cryptocurrency, like bitcoin.
- **How do I get monero?**
  You can buy monero in many of the same places you can get bitcoin. More info
- **What happens if I don't pay?**
  Your files will remain encrypted forever. We won't give you your files for free.
- **How do I know you'll give me my files back?**
  If we didn't, you would tell others to not pay. So trust us, we will return your files.

If you delete this program or your antivirus deletes it, you will not be able to decrypt your files.

RANSOMWARE

EXAMPLES


JOINT CYBERSECURITY ADVISORY

Co-Authored by:

TLP:WHITE

Product ID: AA2

February 9

ACSC Australian Cyber Security Centre

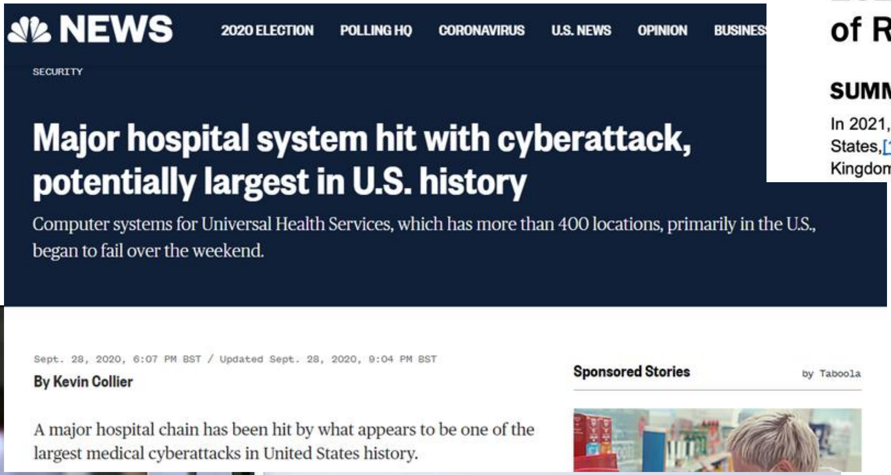
National Cyber Security Centre a part of GCHQ

2021 Trends Show Increased Globalized Threat of Ransomware

SUMMARY

In 2021, cybersecurity authorities in the United States,[1][2][3] Australia,[4] and the United Kingdom[5] observed an increase in sophisticated,

Immediate Actions You Can Take Now to Protect Against Ransomware:


NEWS 2020 ELECTION POLLING HQ CORONAVIRUS U.S. NEWS OPINION BUSINES

SECURITY

Major hospital system hit with cyberattack, potentially largest in U.S. history

Computer systems for Universal Health Services, which has more than 400 locations, primarily in the U.S., began to fail over the weekend.

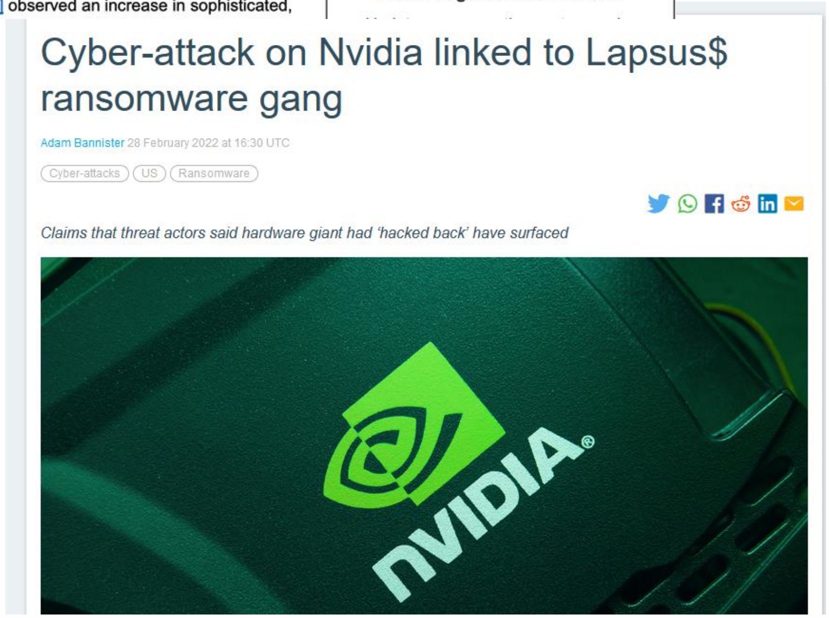
Sept. 28, 2020, 6:07 PM BST / Updated Sept. 28, 2020, 9:04 PM BST

By Kevin Collier

A major hospital chain has been hit by what appears to be one of the largest medical cyberattacks in United States history.

Sponsored Stories by Taboola

MedStar Georgetown University Hospital
HACKERS PARALYZED HOSPITAL CHAIN
NBC NEWS
This is a Tobacco-Free Campus


Cyber-attack on Nvidia linked to Lapsus$ ransomware gang

Adam Bannister 28 February 2022 at 16:30 UTC

Cyber-attacks US Ransomware

Claims that threat actors said hardware giant had 'hacked back' have surfaced

Attacks 2021: https://illinois.touro.edu/news/the-10-biggest-ransomware-attacks-of-2021.php
https://www.nbcnews.com/tech/security/big-paydays-force-hospitals-prepare-ransomware-attacks-n557176

TROJAN HORSE

Transport means…Getting victims to download virus without attacker's intervention.

Free program made available to unsuspecting user

- Actually contains code to do harm

Place altered version of utility program on victim's computer

- trick user into running that program

Example:

- With hacking, music can take control of your car

https://geeknizer.com/hack-car-with-music/

https://web.archive.org/web/20110316080851/http://www.itworld.c can-take-control-your-car

VIRUSES

Virus is a program that can reproduce itself by attaching its code to another program.

Often written in assembler or C.

Attacker infects a program on his own machine, then gets that program distributed.

Once installed on victim's machine, it remains dormant until executed.

Once *activated*…

- Executes it **payload**
- Often waits for a specific date or time
- …. We want to make sure the virus is well distributed before people start noticing it.

DIFFERENT KINDS OF VIRUSES

7 TYPES

1. Companion
2. Executable Program
3. Memory
4. Boot sector
5. Device Driver
6. Macro
7. Source code

1. COMPANION VIRUS

Old virus type

Runs with the program is supposed to run

Ex: in old MS-DOS

- We install a program named *prog.com*
- When user enters *prog*, instead of *prog.exe,* our infected program is executed.
- We'll call *prog.exe* after our malicious activity and no one will be the wiser

Can also be done with symbolic links

2. EXECUTABLE PROGRAM VIRUS

- Overwrites the executable program with itself.
- Virus written in assembly language
- Inserted into another program
  - use tool called a “dropper”
- Virus dormant until program executed
  - then infects other programs
  - eventually executes its “payload”

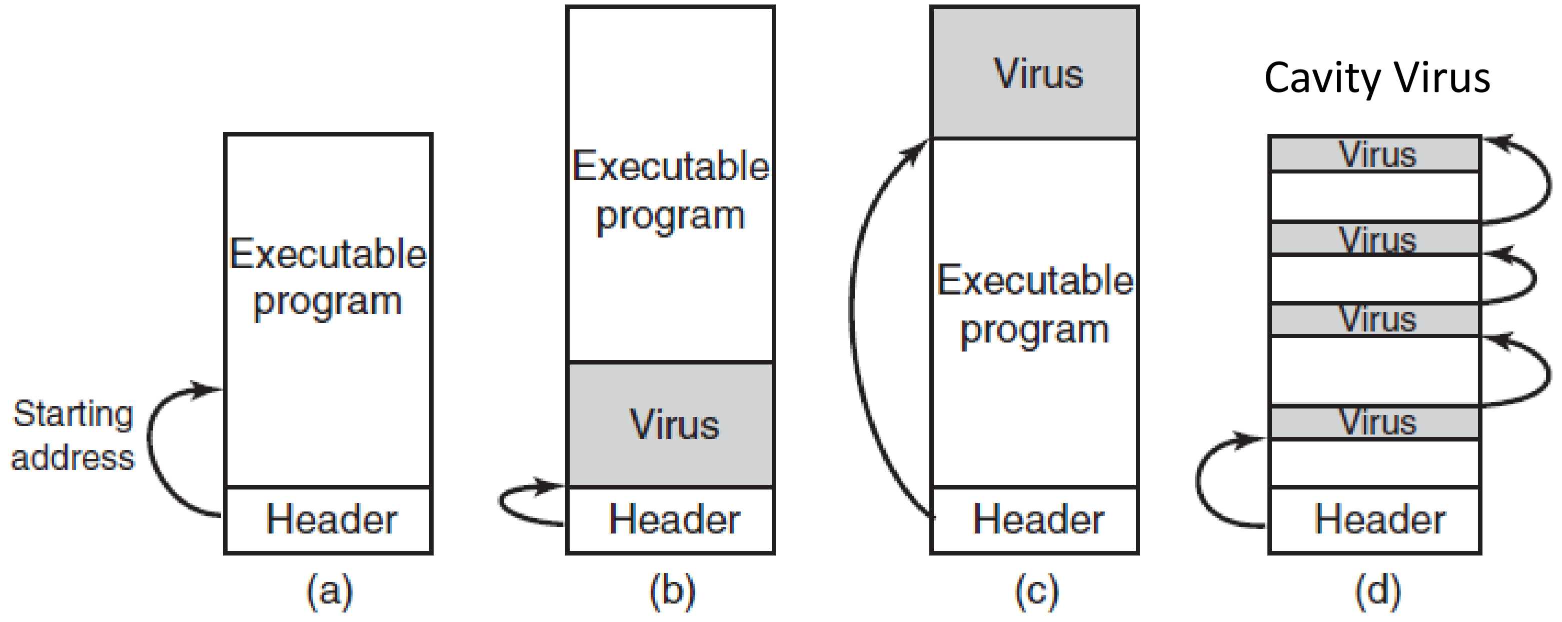
EXECUTABLE VIRUS

Overwriting virus is easy to detect….

Parasitic virus: this virus attaches itself to the program to do the bad thing, but allows the program to function normally afterward.

EXECUTABLE PROGRAM VIRUSES (3)

EXAMPLES



(a) An executable program.

(b) With a virus at the front.

(c) With a virus at the end.

(d) With a virus spread over free space within the program.

3. MEMORY-RESIDENT VIRUSES

Stays in RAM, either hiding at the top of memory or down among the interrupt vectors (the last few hundred bytes are generally unused)

Capture one of the interrupt vectors

- Putting it’s own address there
- Call the interrupt after it does what it does
- Benefit.. It can run in system mode

4. BOOT SECTOR VIRUS

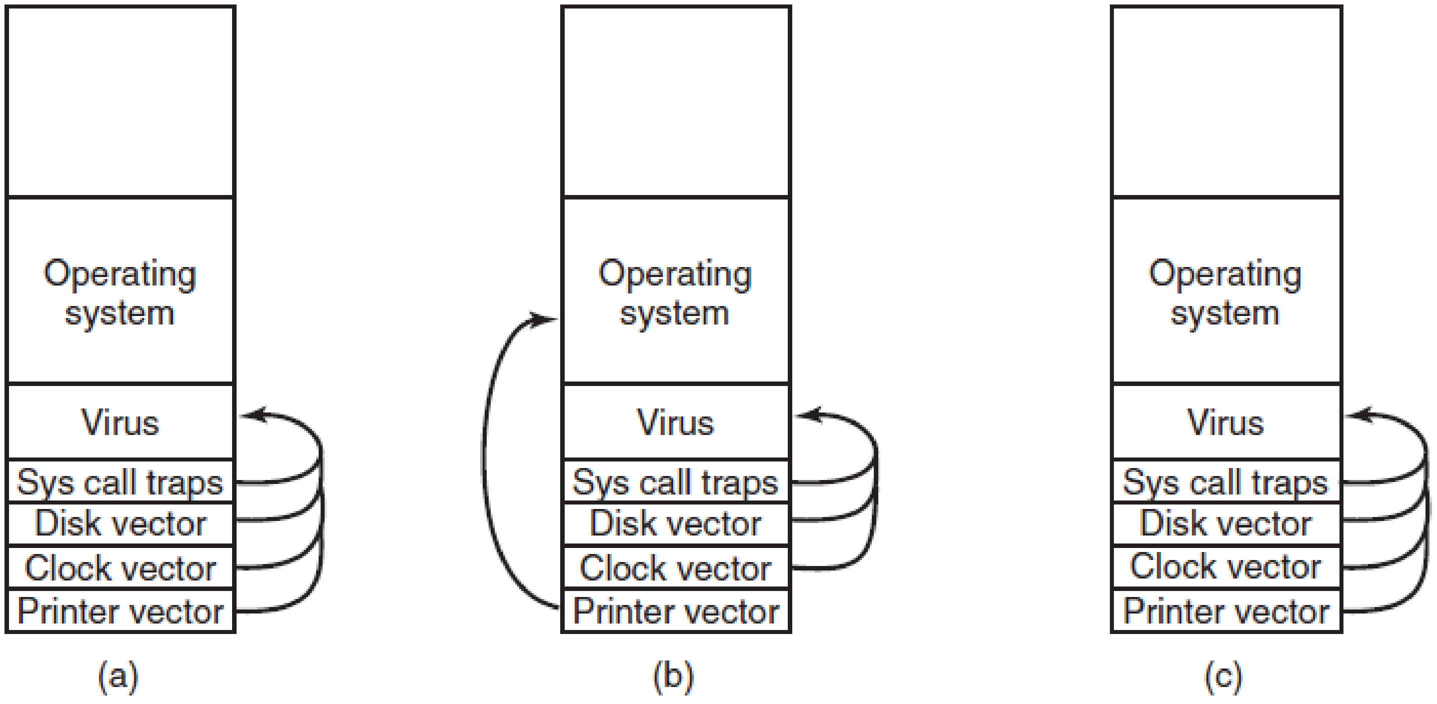
Virus that overwrites the master boot record or boot sector.

Requires intimate knowledge of the operating system’s internal data structure

Copies the first sector of the boot sector to a safe place so it can call it later.

At start-up, it copies the virus to RAM

BOOT SECTOR VIRUSES



(a) After the virus has captured all the interrupt and trap vectors.

(b) After the operating system has retaken the printer interrupt vector.

(c) After the virus has noticed the loss of the printer interrupt vector and recaptured it.

5. DEVICE DRIVER VIRUSES

- Infect the device driver – it’s just a executable programs that live on disk

- Device drivers are always loaded at boot time and **may run kernel mode**.

6. MACRO VIRUSES

- Virus attached to macros in Microsoft Office (e.g. Excel files)
- Send the infected word document to someone.

7. SOURCE CODE VIRUSES

- Very portable
- Looks for C code and changes it to call the virus.

WORMS

- Self-replicating program
- Moves itself through the network and system without the victims help.
- Consisted of two programs
  - bootstrap to upload worm
  - the worm itself
- Worm first hid its existence
- Next replicated itself on new machines
- Famous Robert Morris Internet worm of 1988
  - Computer Emergency Response Team (CERT)
  - https://www.cert.be/en

SPYWARE

- Description: Runs on the victims machine with victim knowing, doing things behind victim’s back
  - Surreptitiously loaded onto a PC without the owner’s knowledge
  - Runs in the background doing things behind the owner’s back

- Characteristics:
  - Hides, victim cannot easily find
  - Collects data about the user
  - Communicates the collected information back to its distant master
  - Tries to survive determined attempts to remove it

- 3 Broad categories
  - Marketing
  - Surveillance
  - Zombie army

HOW SPYWARE SPREADS

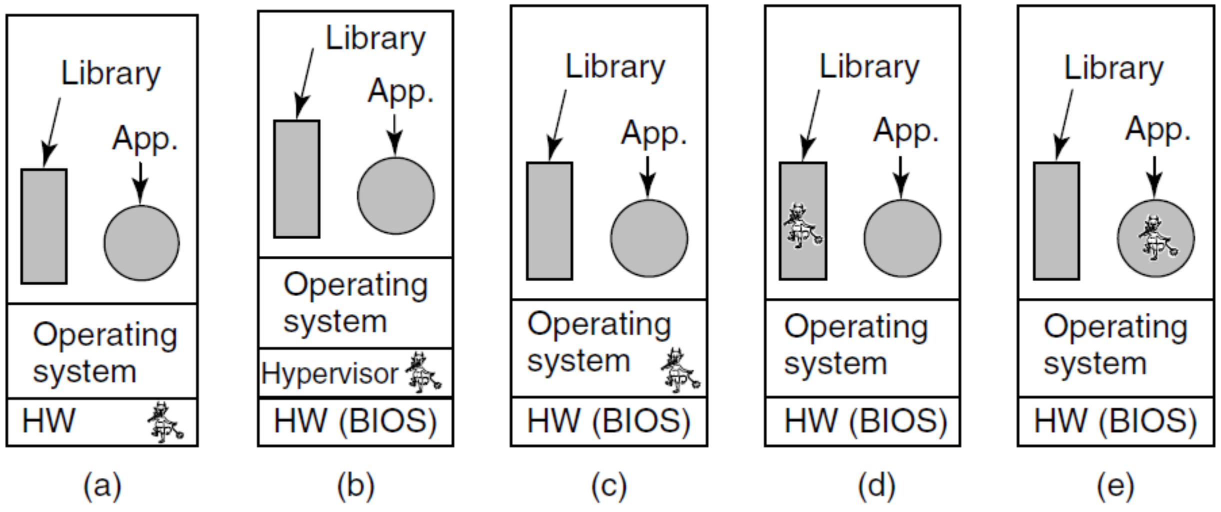
Possible ways:

- Same as malware, Trojan horse
- Drive-by download, visit an infected web site
  - Web pages tries to run an .exe file
  - Unsuspecting user installs an infected toolbar, add-ons
  - Malicious activeX controls get installed

ROOTKITS

- Rootkit: *A rootkit is a collection of computer software, typically malicious, designed to enable access to a computer or an area of its software that is not otherwise allowed (for example, to an unauthorized user) and often masks its existence or the existence of other software**

- Types of Rootkits:
  - (a) Firmware rootkits
  - (b) Hypervisor rootkits
  - (c) Kernel rootkits
  - (d) Library rootkits
  - (e) Application rootkits



**"Rootkits, Part 1 of 3: The Growing Threat" (PDF). McAfee. 2006-04-17. Archived from the original (PDF) on 2006-08-23.*

ROOTKIT DETECTION

Read the files in the directory

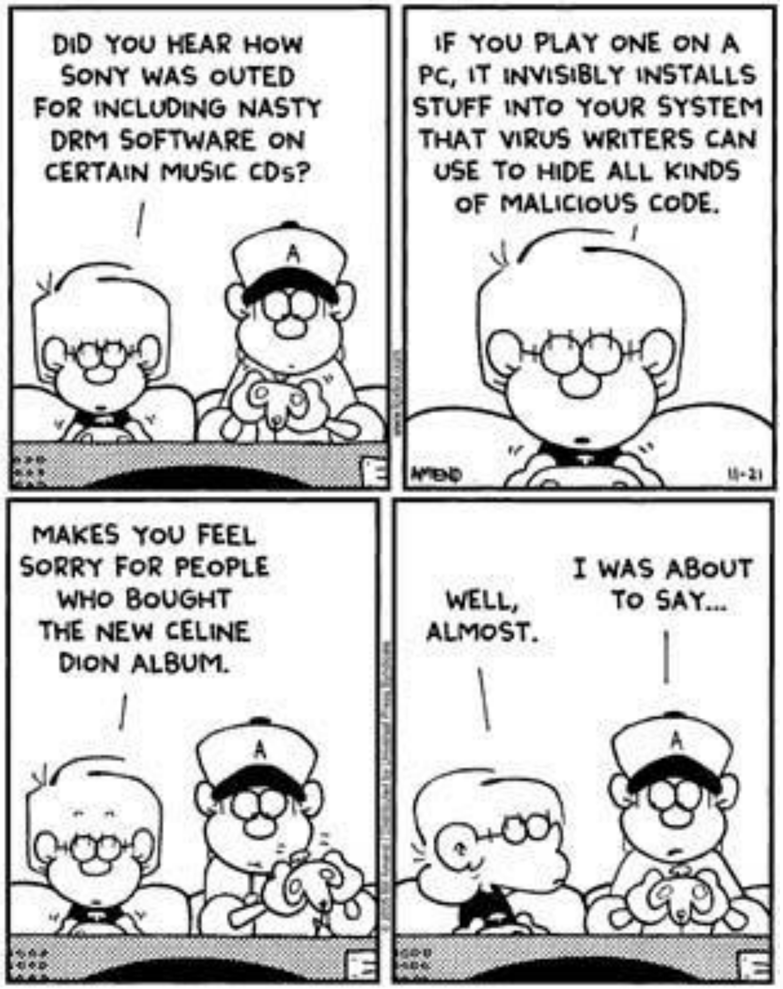
- Unless the dir system call is infected

Timing related

- Does something take longer than it should

Sony Rootkit Scandal (2005)

https://en.wikipedia.org/wiki/Sony_BMG_copy_protection_rootkit_scandal


DID YOU HEAR HOW SONY WAS OUTED FOR INCLUDING NASTY DRM SOFTWARE ON CERTAIN MUSIC CDs?
IF YOU PLAY ONE ON A PC, IT INVISIBLY INSTALLS STUFF INTO YOUR SYSTEM THAT VIRUS WRITERS CAN USE TO HIDE ALL KINDS OF MALICIOUS CODE.
MAKES YOU FEEL SORRY FOR PEOPLE WHO BOUGHT THE NEW CELINE DION ALBUM.
WELL, ALMOST.
I WAS ABOUT TO SAY...

Source: https://www.wired.com/2006/12/sony-settles-bm/

DESIGN PRINCIPLES FOR SECURITY

1. System design should be public
2. Default should be no access
3. Check for current authority
4. Give each process least privilege possible
5. Protection mechanism should be
   - simple
   - uniform
   - in lowest layers of system
6. Scheme should be psychologically acceptable

And … <u>keep it simple</u>

ETRO
ELECTRONICS &
INFORMATICS

OPERATING SYSTEMS

SECURITY – PART 4: DEFENSES

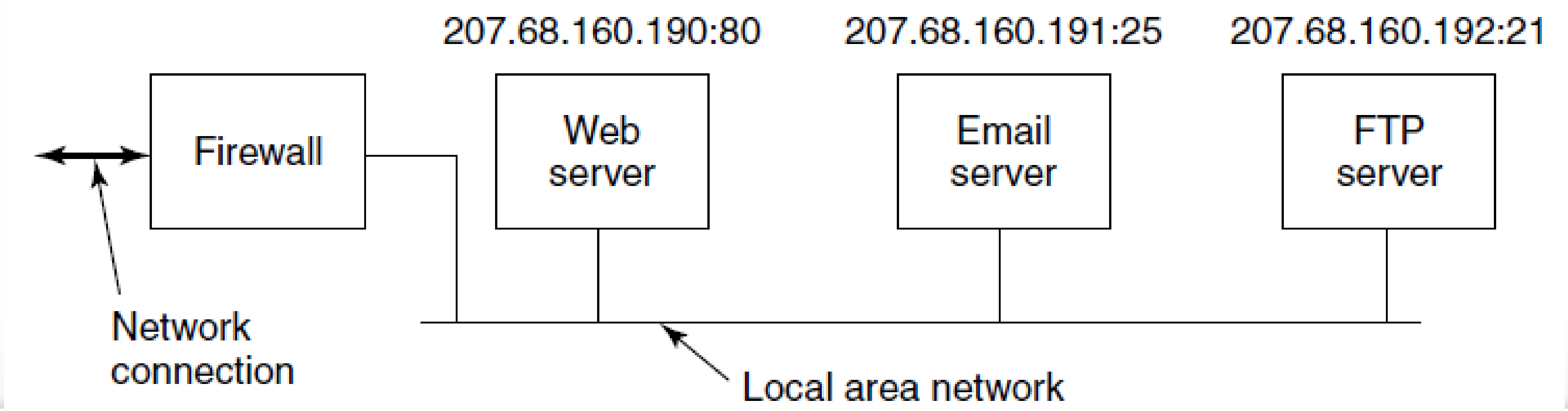
DEFENSES

1. Firewall
2. Antivirus
3. Code Signing
4. Jailing
5. Model-Based Intrusion Detection
6. Encapsulating Mobile Code
7. Java Security

DEFENSE 1: FIREWALLS

A simplified view of a hardware firewall protecting a LAN with three computers

- No packets can enter or exit the LAN without approval from Firewall
- Stateless Firewall – Packet header information is used in approval
- Stateful Firewall – Firewall tracks connections … may inspect packets.



DEFENSE 2: ANTIVIRUS

Some techniques

- Virus Scanner
  - Goat file to attract a virus
    - After analysis of virus, add to database known viruses
  - Store file lengths
    - If they change…. Potential problem
  - Hunt for decryption procedure
    - If virus compresses to fit in pgm size..

DEFENSE 2: VIRUS SCANNERS

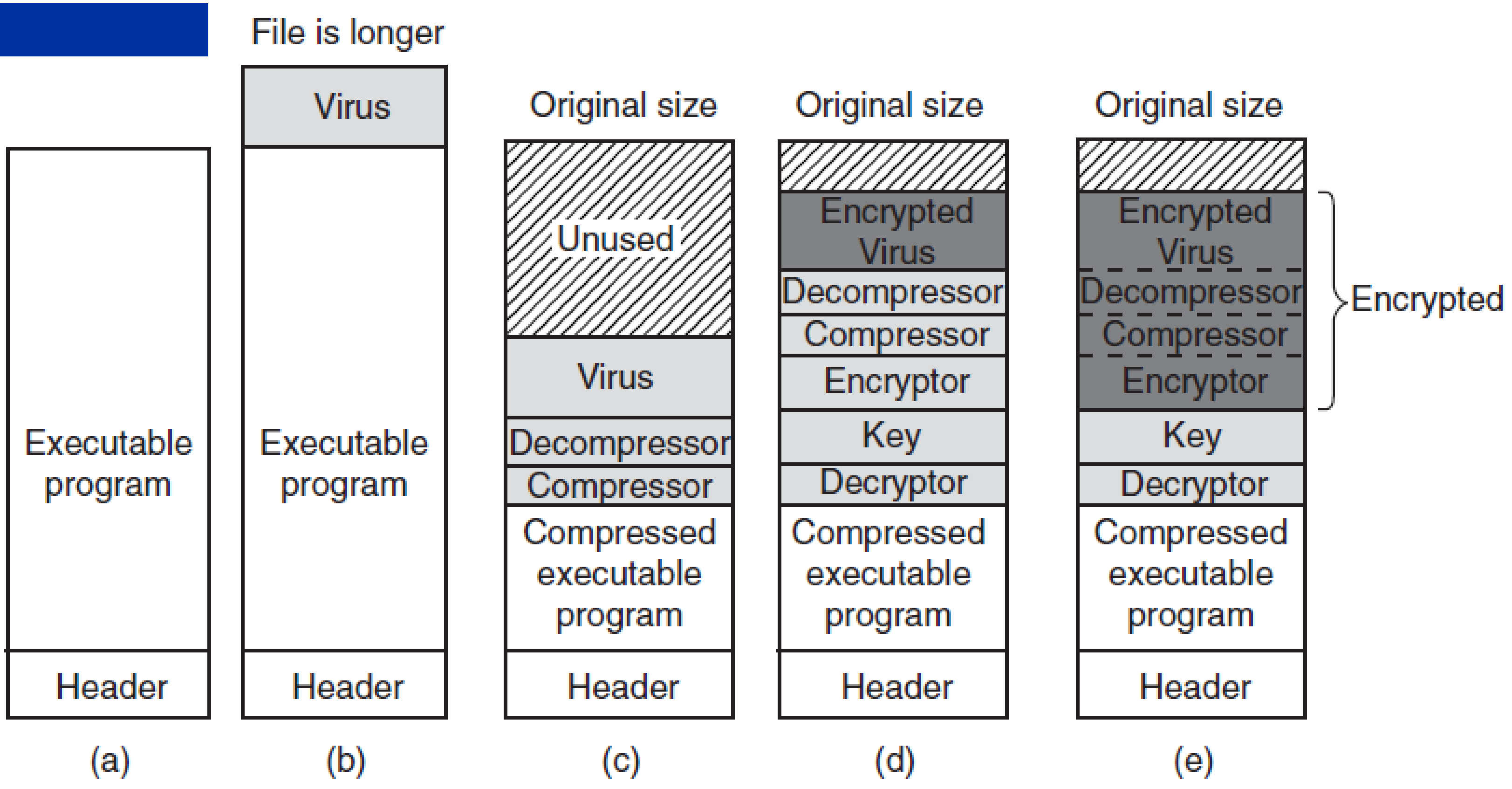
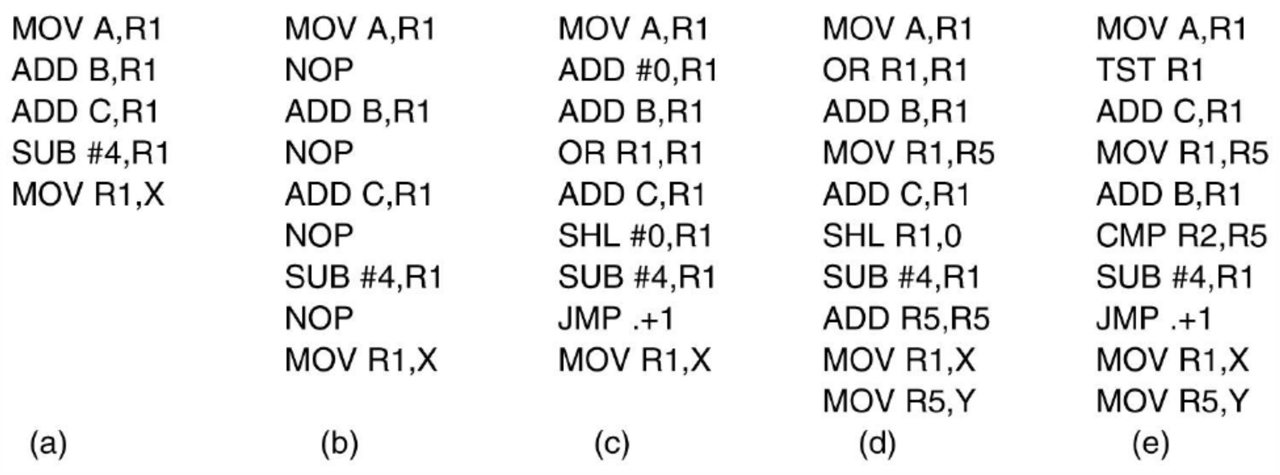


Figure 9-33. (a) A program. (b) An infected program. (c) A compressed infected program. (d) An encrypted virus. (e) A compressed virus with encrypted compression code.

DEFENSE 2: VIRUS SCANNERS

POLYMORPHIC VIRUS

All of these examples do the same thing



```
MOV A,R1      MOV A,R1      MOV A,R1      MOV A,R1      MOV A,R1
ADD B,R1      NOP           ADD #0,R1     OR R1,R1      TST R1
ADD C,R1      ADD B,R1      ADD B,R1      ADD B,R1      ADD C,R1
SUB #4,R1     NOP           OR R1,R1      MOV R1,R5     MOV R1,R5
MOV R1,X      ADD C,R1      ADD C,R1      ADD C,R1      ADD B,R1
              NOP           SHL #0,R1     SHL R1,0      CMP R2,R5
              SUB #4,R1     SUB #4,R1     SUB #4,R1     SUB #4,R1
              NOP           JMP .+1       ADD R5,R5     JMP .+1
              MOV R1,X      MOV R1,X      MOV R1,X      MOV R1,X
                                          MOV R5,Y      MOV R5,Y
  (a)           (b)           (c)           (d)           (e)
```

DEFENSE 2: ANTIVIRUS (II)

INTEGRITY CHECKERS

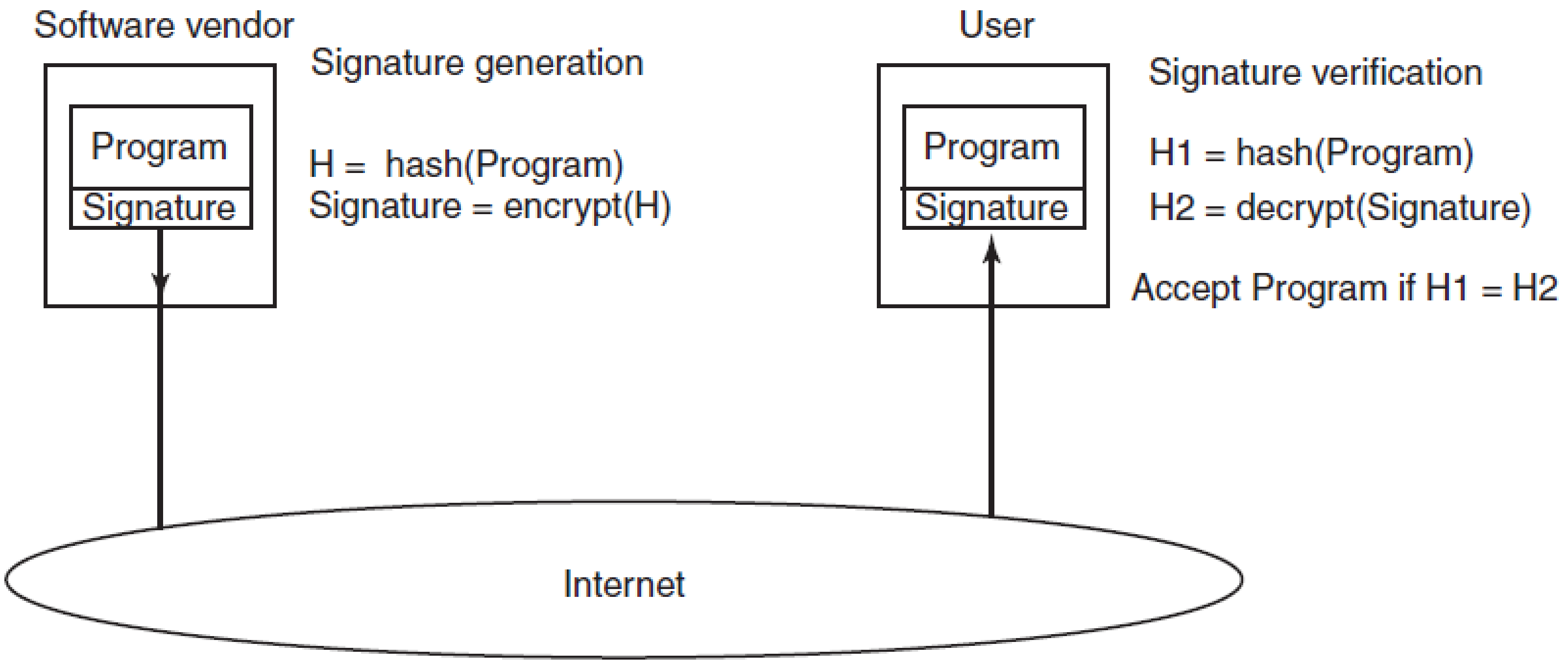
Some techniques

- Integrity Checkers
  - Compute checksum for clean files

- Behavioral Checkers
  - Monitor all activity
  - Word shouldn't overwrite a file

DEFENSE 3: CODE SIGNING

Using digital signatures to sign code

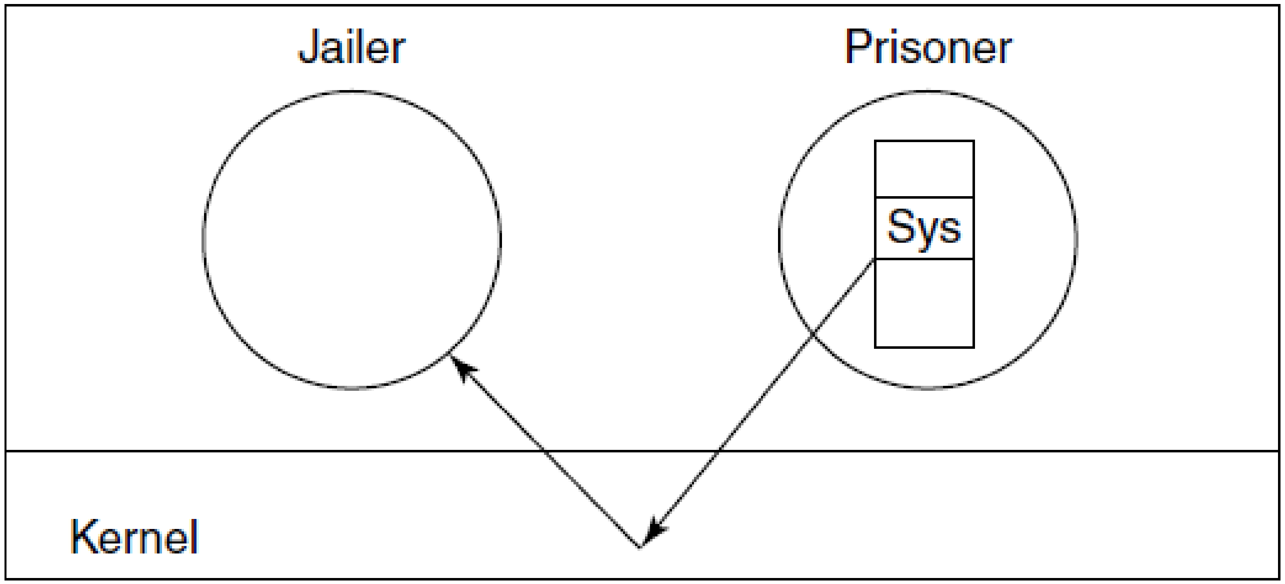


DEFENSE 4: JAILING

The new program’s execution is monitored in a jail.

System call is transferred to jailer who makes the decision if it is allowed.

Like running in a debugger.



DEFENSE 5: MODEL-BASED INTRUSION DETECTION

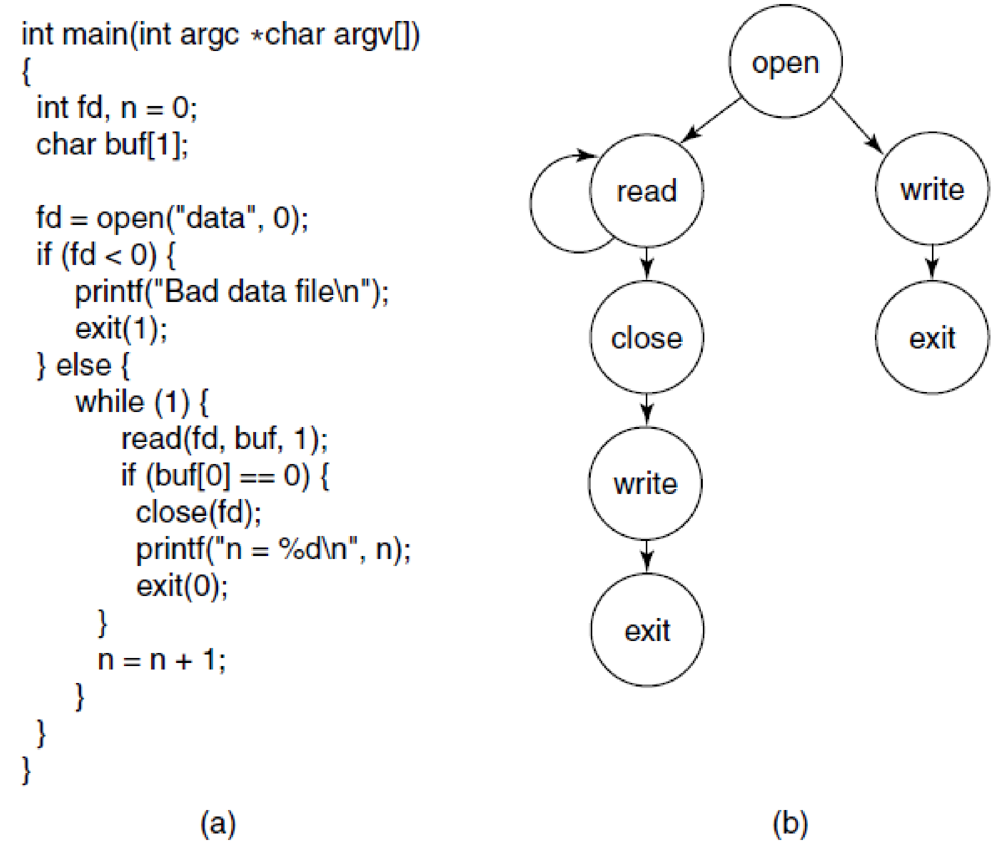
Intrusion Detection System (IDS)

1. Network-Based IDS
   - Focused on incoming packets
2. Host based IDS
   - Static model-based intrusion detection

MODEL-BASED INTRUSION DETECTION

Static model-based intrusion detection

- Implemented using jailing technique
- Learn the 'good' behavior of a program from program model.
  - Compiler can generate it and the author certifies it



```
int main(int argc *char argv[])
{
 int fd, n = 0;
 char buf[1];

 fd = open("data", 0);
 if (fd < 0) {
     printf("Bad data file\n");
     exit(1);
 } else {
     while (1) {
         read(fd, buf, 1);
         if (buf[0] == 0) {
          close(fd);
          printf("n = %d\n", n);
          exit(0);
         }
         n = n + 1;
     }
 }
}
```

(a)

(a) A program.

(b) System call graph for (a).

DEFENSE 6: ENCAPSULATING MOBILE CODE

Problem:

- Javascript, applets, agents…
- Things that want to execute on our machines
- Things we may want to let execute on our machines

Defensive methods

- Sandboxing
- Interpretation

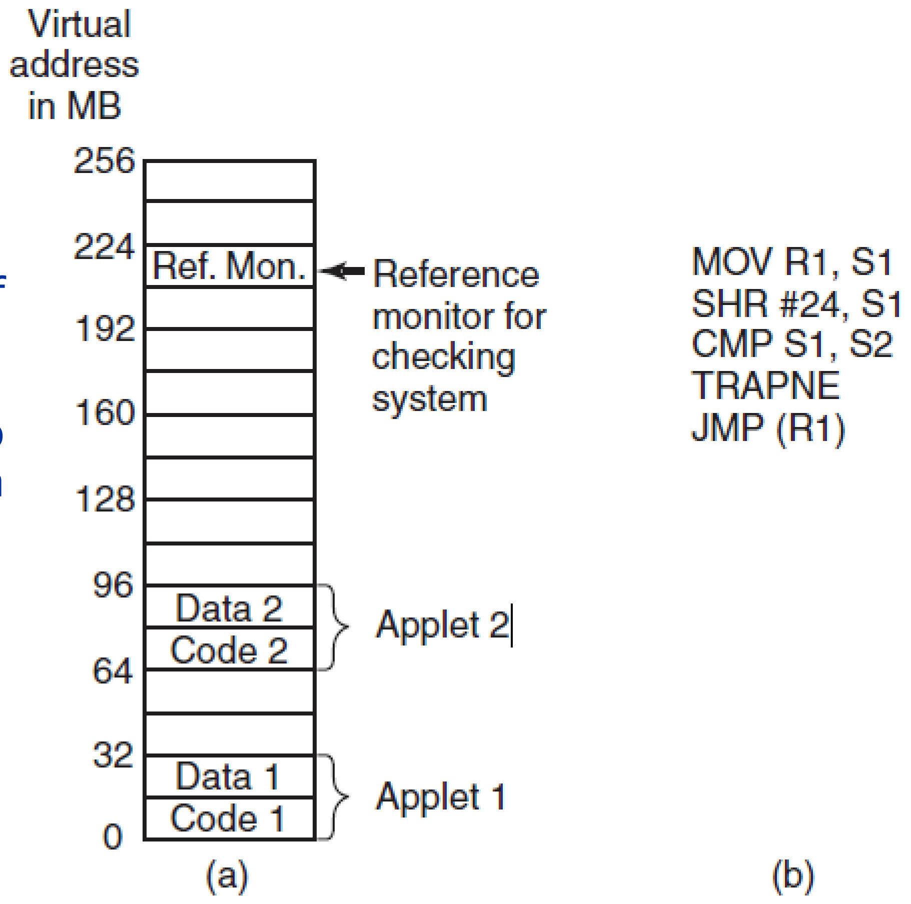
SANDBOXING

Divides virtual address into 2 regions:

- One for data
- One for code

Confines applet to a limited range of virtual addresses enforced at runtime

Guarantees the applet cannot jump to code outside its code or reference data outside data sandbox.



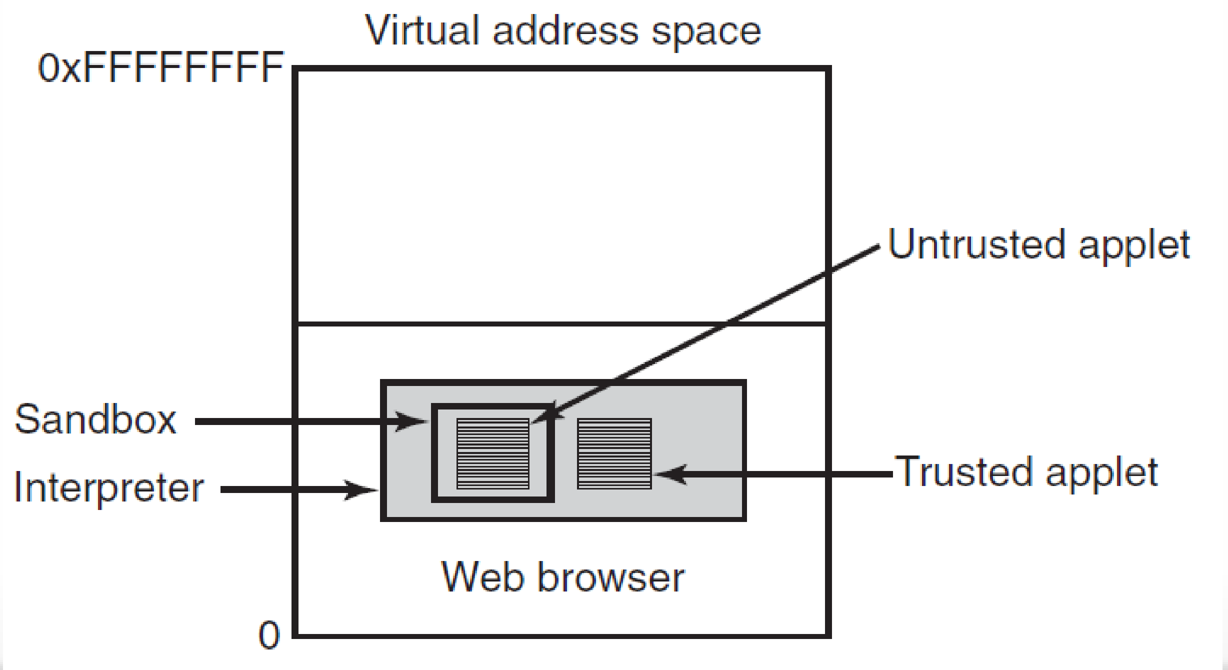
```
MOV R1, S1
SHR #24, S1
CMP S1, S2
TRAPNE
JMP (R1)
```

(a) Memory divided into 16-MB sandboxes.
(b) One way of checking an instruction for validity.

INTERPRETATION

Run applets interpretively.

Every instruction can be examined by interpreter.



DEFENSE 7: JAVA SECURITY

- A type safe language
  - compiler rejects attempts to misuse variable

Checks on applets include:

1. Does applet attempt to forge pointers?
2. Does it violate access restrictions on private-class members?
3. Does it try to use variable of one type as another?
4. Does it generate stack overflows or underflows?
5. Does it illegally convert variables of one type to another?

SUMMARY

- Operating systems support mechanisms to satisfy the computer security triad.
- Trusted Computing Base consists of the hardware and software necessary for enforcing all the security rules.
- Protection domains define the operations that can be performed on each object.
- There are different types of attacks by:
  - Exploiting Software
  - Insider Attacks
- Another type of attacks involve malicious software.
  - Trojan horses, viruses, and worms are collectively called malware.
- There are different strategies for defense.
  - From the use of Firewalls, antivirus software to Intrusion Detection Systems.

Viruses
Intruders
Spyware
Trojans and
Adware

"We'll make some piracy improvements this year."

-Steve Ballmer
CEO, Microsoft