OPERATING SYSTEMS

I/O Management, System Protection and Security

Kakoli Bora
Department of Computer Science

OPERATING SYSTEMS

System Security - Program Threats

Kakoli Bora
Department of Computer Science

Slides Credits for all PPTs of this course

- The slides/diagrams in this course are an **adaptation**, **combination**, and **enhancement** of material from the following resources and persons:

1. Slides of Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne - 9th edition 2013 and some slides from 10th edition 2018
2. Some conceptual text and diagram from Operating Systems - Internals and Design Principles, William Stallings, 9th edition 2018
3. Some presentation transcripts from A. Frank – P. Weisberg
4. Some conceptual text from Operating Systems: Three Easy Pieces, Remzi Arpaci-Dusseau, Andrea Arpaci Dusseau

Program Threats

- Many variations, many names
- **Trojan Horse**
  - Code segment that misuses its environment
  - Exploits mechanisms for allowing programs written by users to be executed by other users
  - **Spyware**, **pop-up browser windows**, **covert channels**
    - The goal of spyware is to download ads to display on the user's system, create pop-up browser windows when certain sites are visited, or capture information from the user's system and return it to a central site.
    - Covert channels are new type of spyware attack. It was estimated that 90 percent of spam was being delivered by this method.

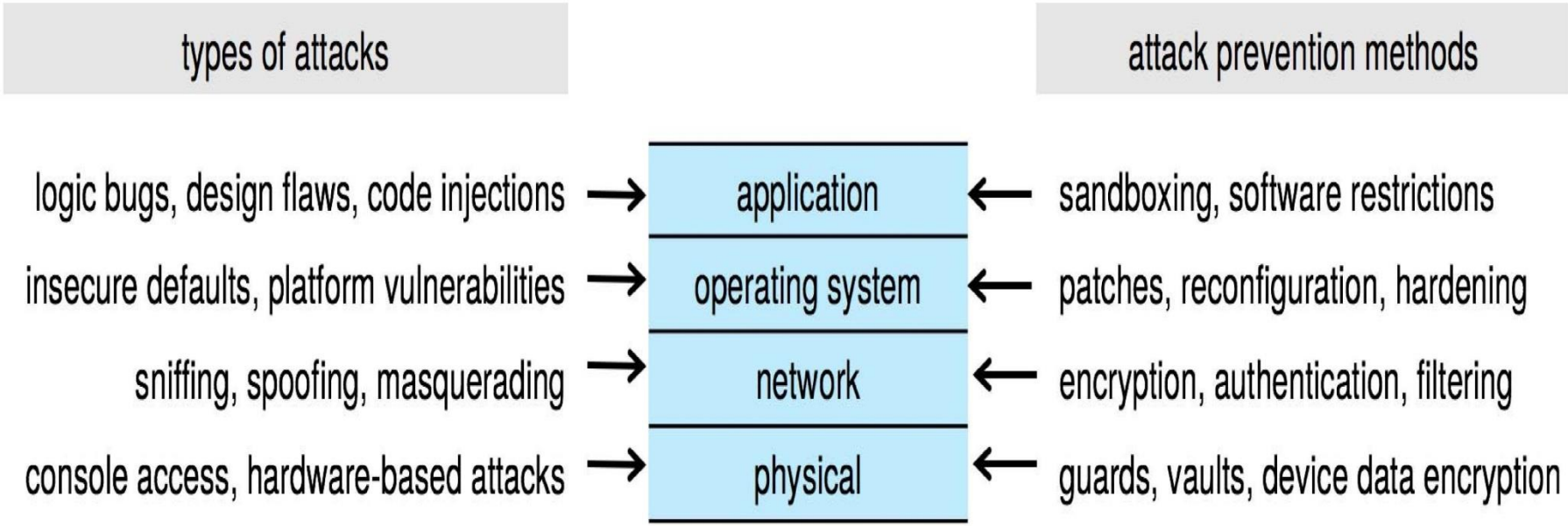
Program Threats

- **Trap Door**
  - The designer of a program or system might leave a hole in the software through which intruders attack
  - Specific user identifier or password that circumvents normal security procedures
  - Could be included in a compiler
  - How to detect them?

- **Logic Bomb**
  - Program that initiates a security incident under certain circumstances

OPERATING SYSTEMS

Four-layered Model of Security



Program Threats (Cont.)

- **Stack** and **Buffer Overflow**
  - Exploits a bug in a program (overflow either the stack or memory buffers)
  - Failure to check bounds on inputs, arguments
  - Write past arguments on the stack into the return address on stack
  - When routine returns from call, returns to hacked address
    - Pointed to code loaded onto stack that executes malicious code
  - Unauthorized user or privilege escalation

OPERATING SYSTEMS

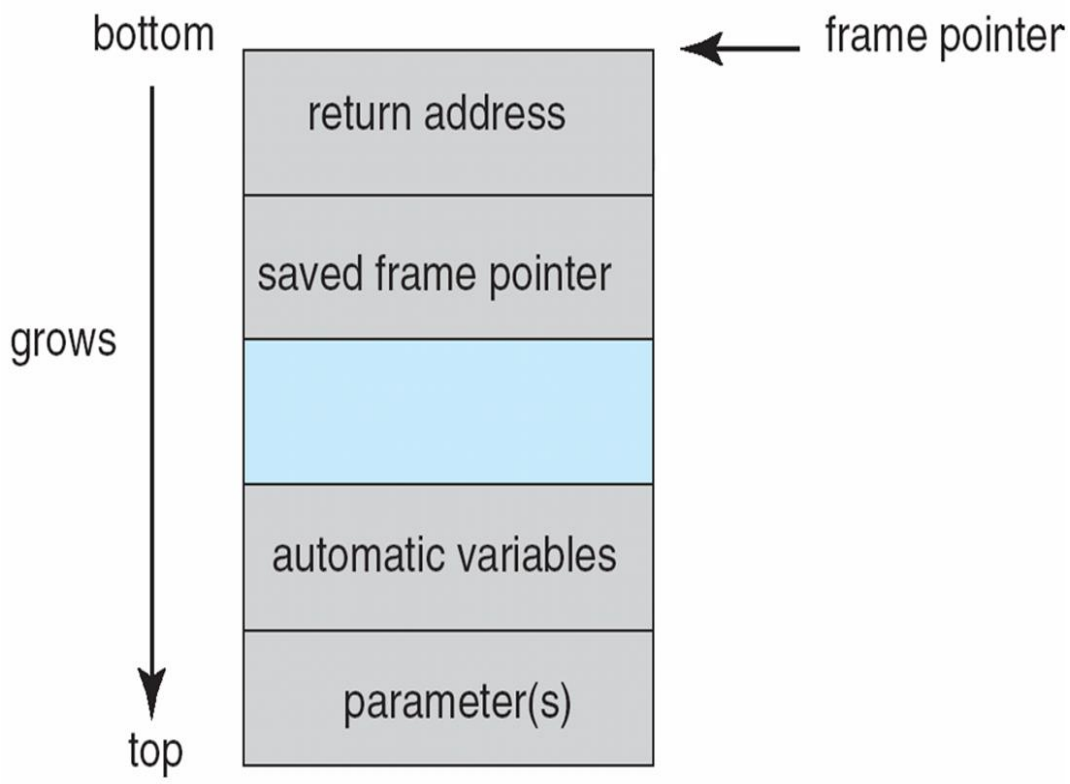
C Program with Buffer-overflow Condition

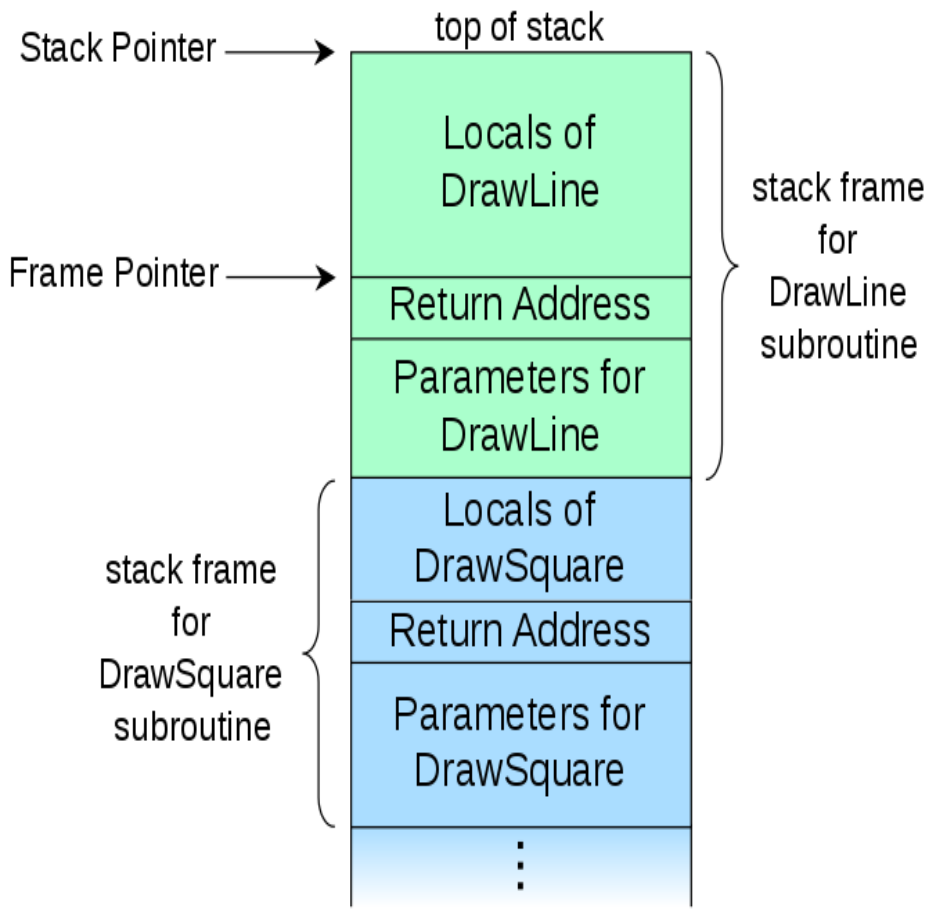
```
#include <stdio.h>
#define BUFFER SIZE 256
int main(int argc, char *argv[])
{
   char buffer[BUFFER SIZE];
   if (argc < 2)
        return -1;
   else {
        strcpy(buffer, argv[1]);
        return 0;
   }
}
```

- **Strncpy() can help**
- **Code review** can help – programmers review each other's code, looking for logic flaws, programming flaws

OPERATING SYSTEMS

Layout of Typical Stack Frame





Frame pointer is the address of the beginning of the stack frame.

It is saved on the stack, as the value of the SP can vary during the function call.

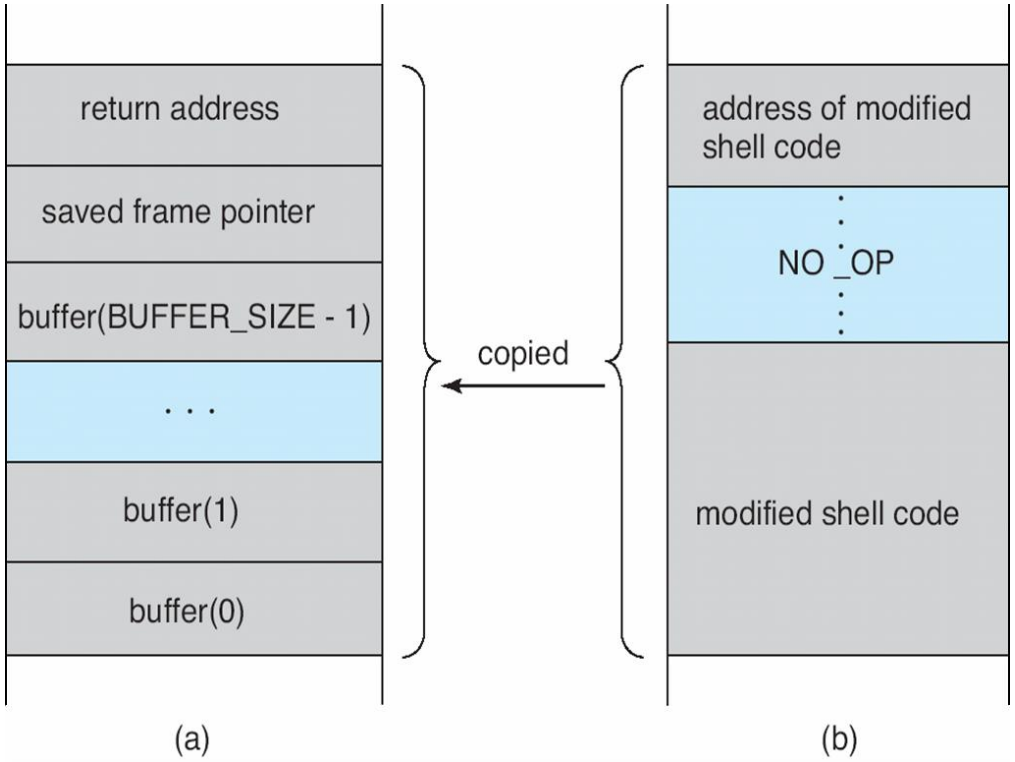
Modified Shell Code

- Using the execvp() system call, this code segment creates a shell process.
- If the program being attacked runs with system-wide permissions, this newly created shell will gain complete access to the system.

```
#include <stdio.h>
int main(int argc, char *argv[])
{
   execvp(‘‘\bin\sh’’,‘‘\bin \sh’’, NULL);
   return 0;
}
```

OPERATING SYSTEMS

Hypothetical Stack Frame



Before attack

After attack

Great Programming Required?

- For the first step of determining the bug, and second step of writing exploit code, yes
- **Script kiddies** can run pre-written exploit code to attack a given system
- Attack code can get a shell with the processes' owner's permissions
  - Or open a network port, delete files, download a program, etc
- Depending on bug, attack can be executed across a network using allowed connections, bypassing firewalls
- Buffer overflow can be disabled by disabling stack execution or adding bit to page table to indicate "non-executable" state
  - Available in SPARC and x86
  - But still have security exploits

Program Threats (Cont.)

- **Viruses**
  - Virus is a fragment of code embedded in a legitimate program
  - Self-replicating, designed to infect other computers
  - Very specific to CPU architecture, operating system, applications
  - Usually borne via email or as a macro
  - Visual Basic Macro to reformat hard drive of a Windows computer as soon as the macro file is opened

```
Sub AutoOpen()
Dim oFS
  Set oFS = CreateObject(''Scripting.FileSystemObject'')
  vs = Shell(''c:command.com /k format c:'',vbHide)
End Sub
```

OPERATING SYSTEMS

Program Threats (Cont.)

- **Virus dropper** inserts/launches virus onto the system
- Many categories of viruses, literally many thousands of viruses
  - File / parasitic virus – appends itself to a file, changes execution start or order of the program
  - Boot / memory virus – infects boot sector, infects bootable media, do not appear in the file system
  - Macro – written in high language such as Visual Basic
  - Source code – modifies source code to include a call to some malicious code, spreads virus
  - Polymorphic – changes **virus signature (identity pattern)** each time it is installed to avoid detection by antivirus software

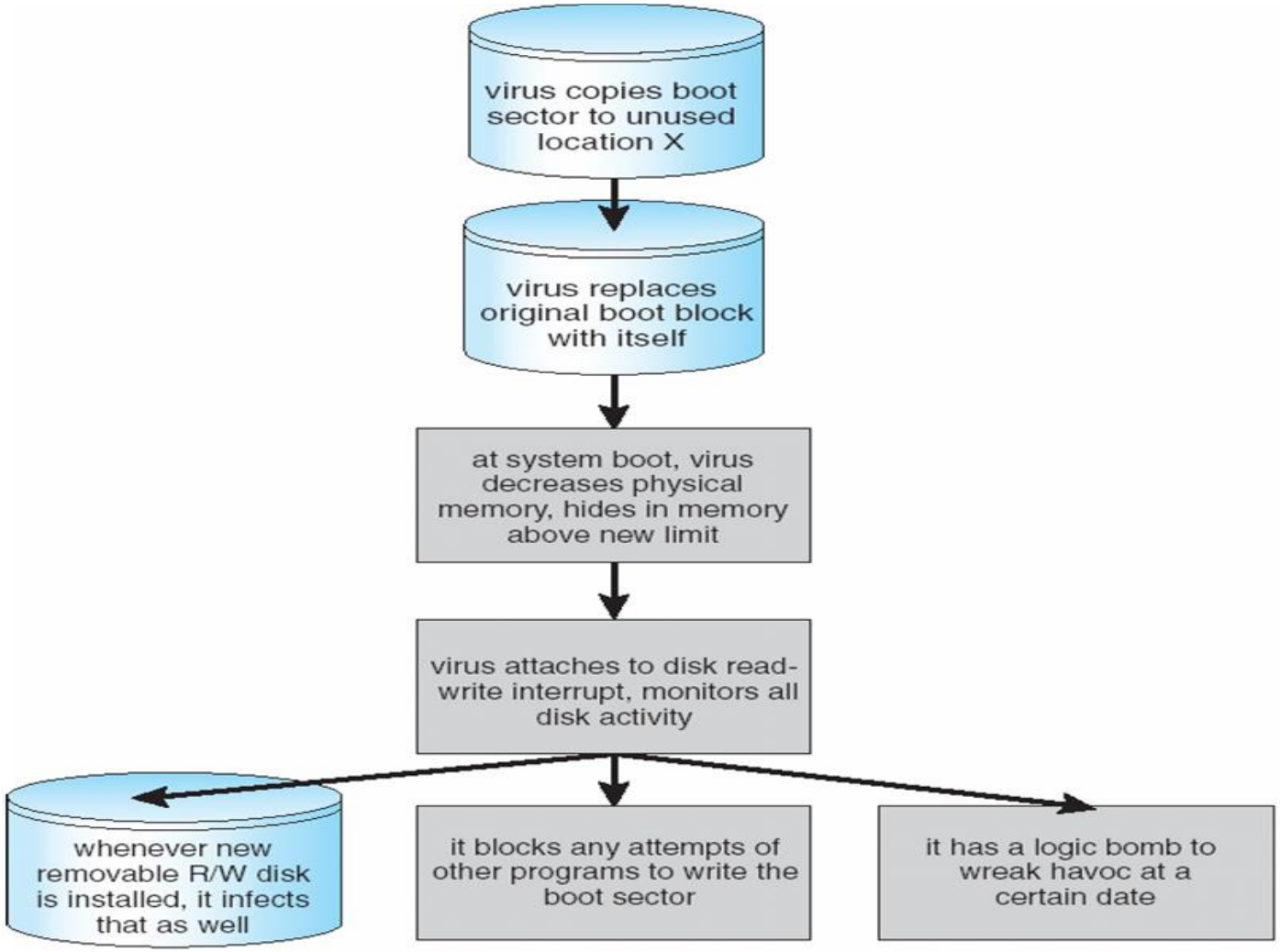
A **polymorphic virus** decrypts its code, runs that code, and then when propagating itself encrypts the decrypted code with a different key.
A **metamorphic virus** simply runs its code and then when propagating itself mutates its code into different but functionally identical code.

Program Threats (Cont.)

- Encrypted – includes encrypted virus and decryption code
- Stealth –hidden computer virus that attacks processes and averts typical anti-virus or anti-malware scans,
  - hides in files, partitions and boot sectors
  - adept at deliberately avoiding detection.
- Tunneling – installs itself in interrupt-handler chain and device drivers
- Multipartite – infects multiple parts of a system such as boot sectors, memory and files
- Armored – fools antivirus software from disclosing its actual location, adds complicated/confusing code

OPERATING SYSTEMS

A Boot-sector Computer Virus



The Threat Continues

- Attacks still common, still occurring
- Attacks moved over time from science experiments to tools of organized crime
  - Targeting specific companies
  - Creating botnets to use as tool for spam and DDOS delivery
  - **Keystroke logger** to grab passwords, credit card numbers
- Why is Windows the target for most attacks?
  - Most common
  - Everyone is an administrator
    - Licensing required?
  - **Monoculture** considered harmful

THANK YOU

Kakoli Bora
Department of Computer Science Engineering

k_bora@pes.edu