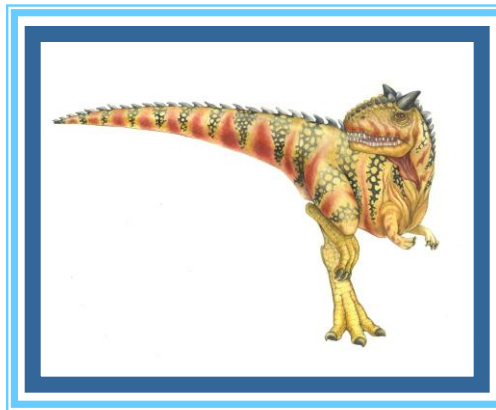
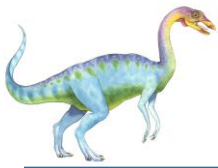


Chapter 15: Security





The Security Problem

- System **secure** if resources used and accessed as intended under all circumstances
 - Unachievable
- **Intruders** (**crackers**) attempt to breach security
- **Threat** is potential security violation
- **Attack** is attempt to breach security
- Attack can be accidental or malicious
- Easier to protect against accidental than malicious misuse

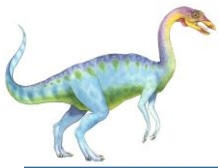




Security Violation Categories

- ❑ **Breach of confidentiality**
 - ❑ Unauthorized reading of data
- ❑ **Breach of integrity**
 - ❑ Unauthorized modification of data
- ❑ **Breach of availability**
 - ❑ Unauthorized destruction of data
- ❑ **Theft of service**
 - ❑ Unauthorized use of resources
- ❑ **Denial of service (DOS)**
 - ❑ Prevention of legitimate use





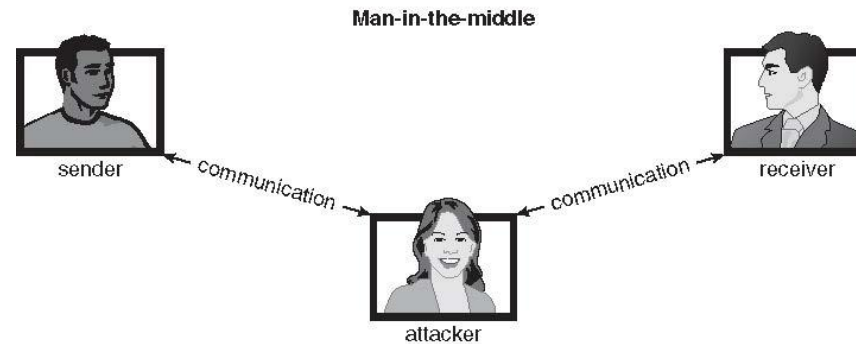
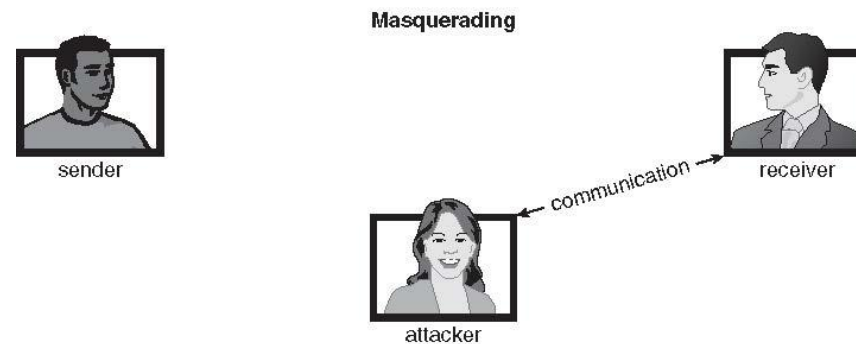
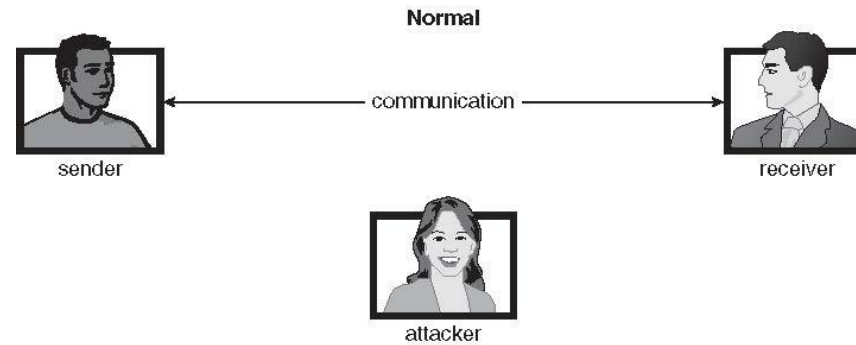
Security Violation Methods

- ❑ **Masquerading** (breach **authentication**)
 - ❑ Pretending to be an authorized user to escalate privileges
- ❑ **Replay attack**
 - ❑ As is or with **message modification**
- ❑ **Man-in-the-middle attack**
 - ❑ Intruder sits in data flow, masquerading as sender to receiver and vice versa
- ❑ **Session hijacking**
 - ❑ Intercept an already-established session to bypass authentication





Standard Security Attacks





Security Measure Levels

- ❑ Impossible to have absolute security, but make cost to perpetrator sufficiently high to deter most intruders
- ❑ Security must occur at four levels to be effective:
 - ❑ **Physical**
 - ▶ Data centers, servers, connected terminals
 - ❑ **Human**
 - ▶ Avoid **social engineering**, **phishing**, **dumpster diving**
 - ❑ **Operating System**
 - ▶ Protection mechanisms, debugging
 - ❑ **Network**
 - ▶ Intercepted communications, interruption, DOS
- ❑ Security is as weak as the weakest link in the chain
- ❑ But can too much security be a problem?





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**
 - Up to 80% of spam delivered by spyware-infected systems
- **Trap Door**
 - Specific user identifier or password that circumvents normal security procedures
 - Could be included in a compiler
 - How to detect them?





Program Threats (Cont.)

❑ Logic Bomb

- ❑ Program that initiates a security incident under certain circumstances

❑ 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





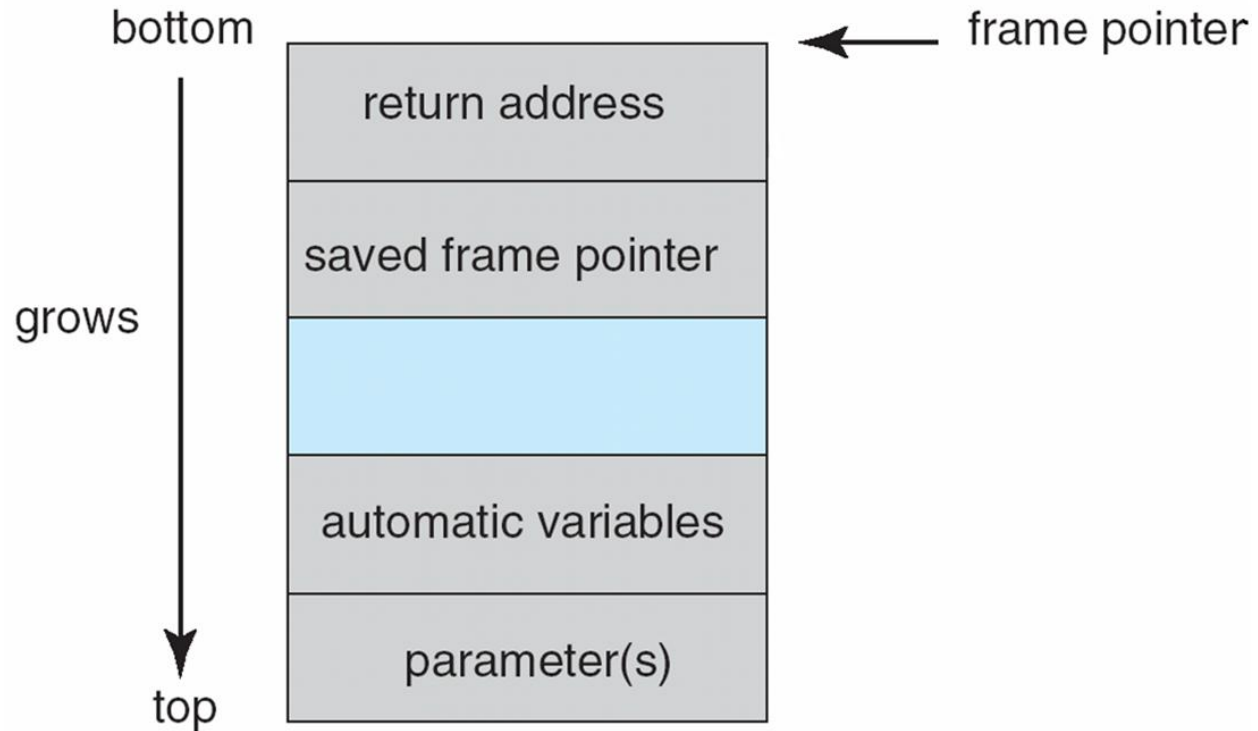
C Program with Buffer-overflow Condition

```
#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;
    }
}
```





Layout of Typical Stack Frame





Modified Shell Code

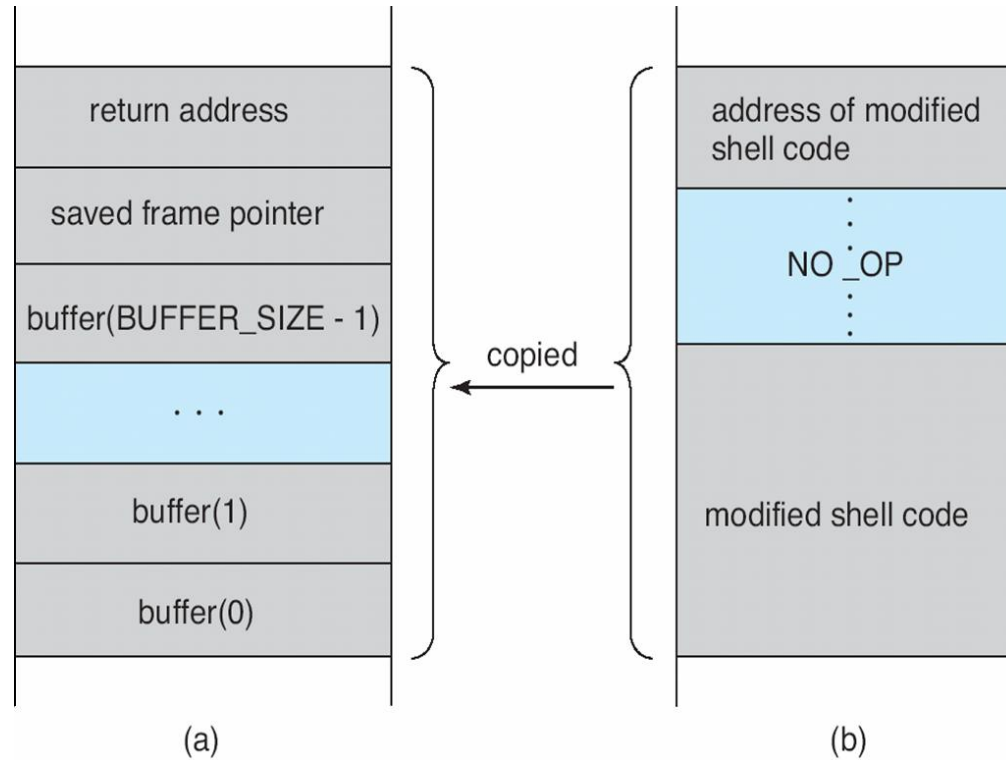
```
#include <stdio.h>

int main(int argc, char *argv[])
{
    execvp(“\bin\sh”, “\bin \sh”, NULL);
    return 0;
}
```





Hypothetical Stack Frame



Before attack

After attack





Program Threats (Cont.)

□ Viruses

- Code fragment embedded in 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

```
Sub AutoOpen()
```

```
Dim oFS
```

```
Set oFS = CreateObject('Scripting.FileSystemObject')
```

```
vs = Shell('c:command.com /k format c:',vbHide)
```

```
End Sub
```





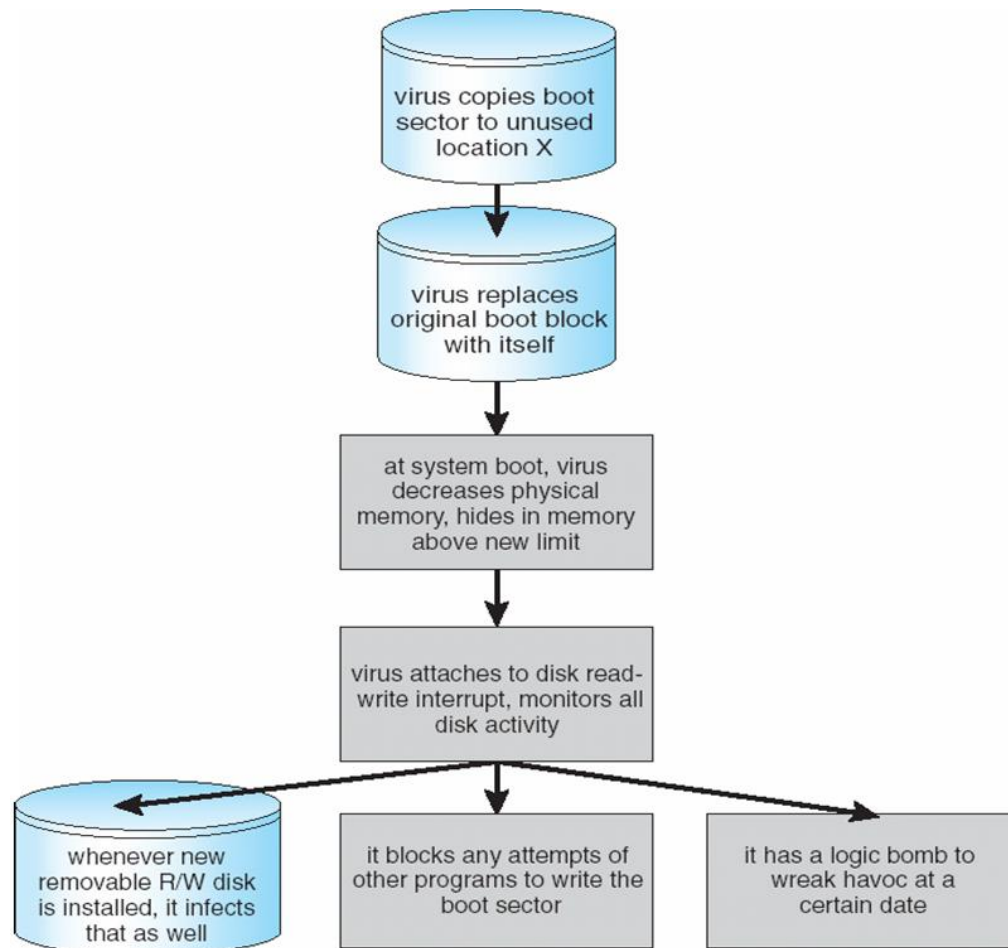
Program Threats (Cont.)

- ❑ **Virus dropper** inserts virus onto the system
- ❑ Many categories of viruses, literally many thousands of viruses
 - ❑ File / parasitic
 - ❑ Boot / memory
 - ❑ Macro
 - ❑ Source code
 - ❑ Polymorphic to avoid having a **virus signature**
 - ❑ Encrypted
 - ❑ Stealth
 - ❑ Tunneling
 - ❑ Multipartite
 - ❑ Armored





A Boot-sector Computer Virus





System and Network Threats (Cont.)

❑ Denial of Service

- ❑ Overload the targeted computer preventing it from doing any useful work
- ❑ **Distributed denial-of-service (DDOS)** come from multiple sites at once
- ❑ Consider the start of the IP-connection handshake (SYN)
 - ▶ How many started-connections can the OS handle?
- ❑ Consider traffic to a web site
 - ▶ How can you tell the difference between being a target and being really popular?
- ❑ Accidental – CS students writing bad `fork()` code
- ❑ Purposeful – extortion, punishment

