Amirkabir University of Technology
(Tehran Polytechnic)

Operating Systems

Security-Part1

Seyyed Ahmad Javadi

sajavadi@aut.ac.ir

Fall 2021

Chapter 16: Security

- The Security Problem
- Program Threats
- System and Network Threats
- Cryptography as a Security Tool
- User Authentication
- Implementing Security Defenses
- Firewalling to Protect Systems and Networks
- Computer-Security Classifications
- An Example: Windows 7

Objectives

- Discuss security threats and attacks
- Explain the fundamentals of encryption, authentication, and hashing
- Examine the uses of cryptography in computing
- Describe the various countermeasures to security attacks

The Security Problem

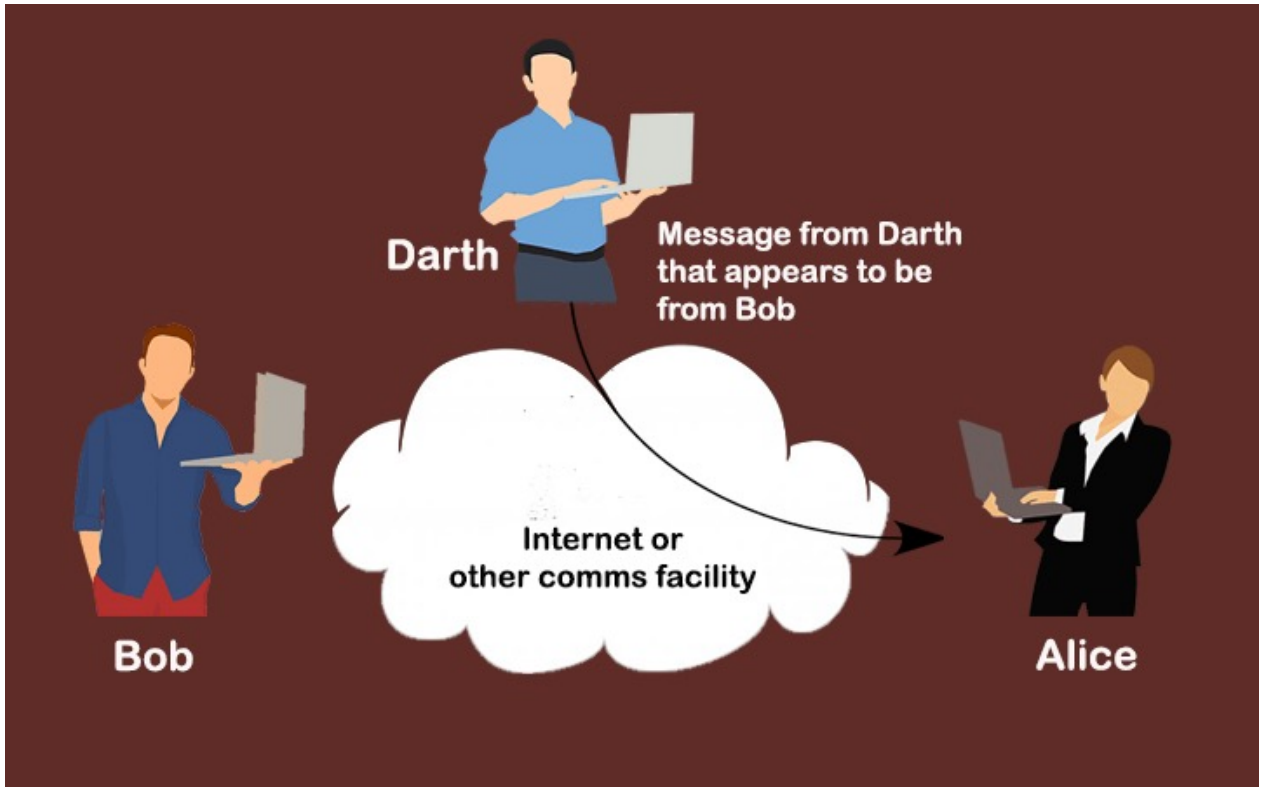
- System secure if resources used and accessed as intended under all circumstances
  - Unachievable
- Intruders (crackers) attempt to breach security
- Threat is potential security violation
- Attack is attempt to breach security
- Attack can be accidental or malicious
- Easier to protect against accidental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

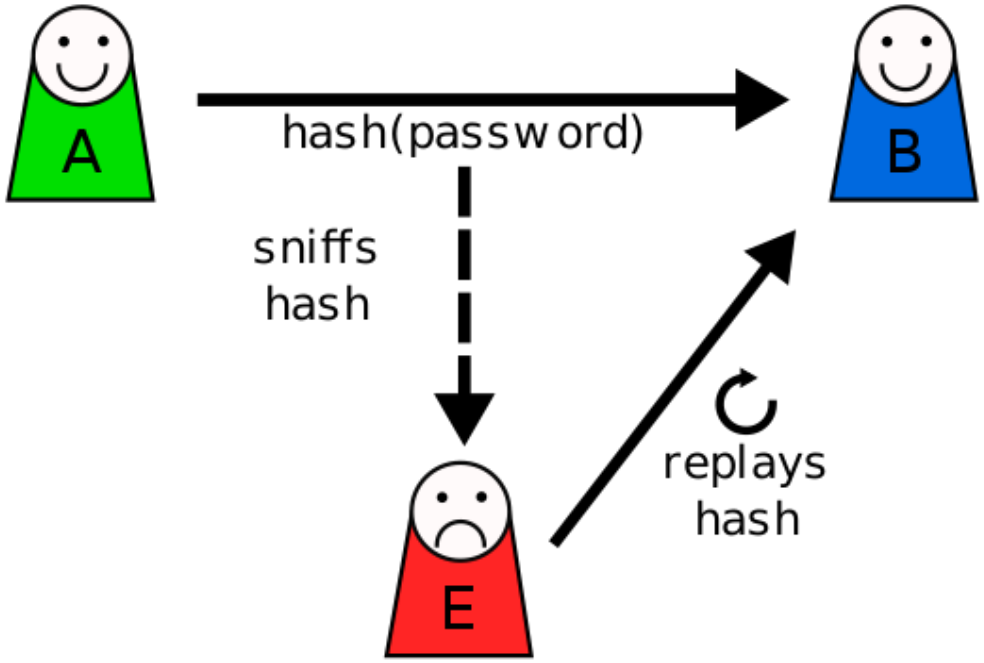
Security Violation Methods

- **Masquerading** (breach **authentication**)
  - Pretending to be an authorized user to escalate privileges



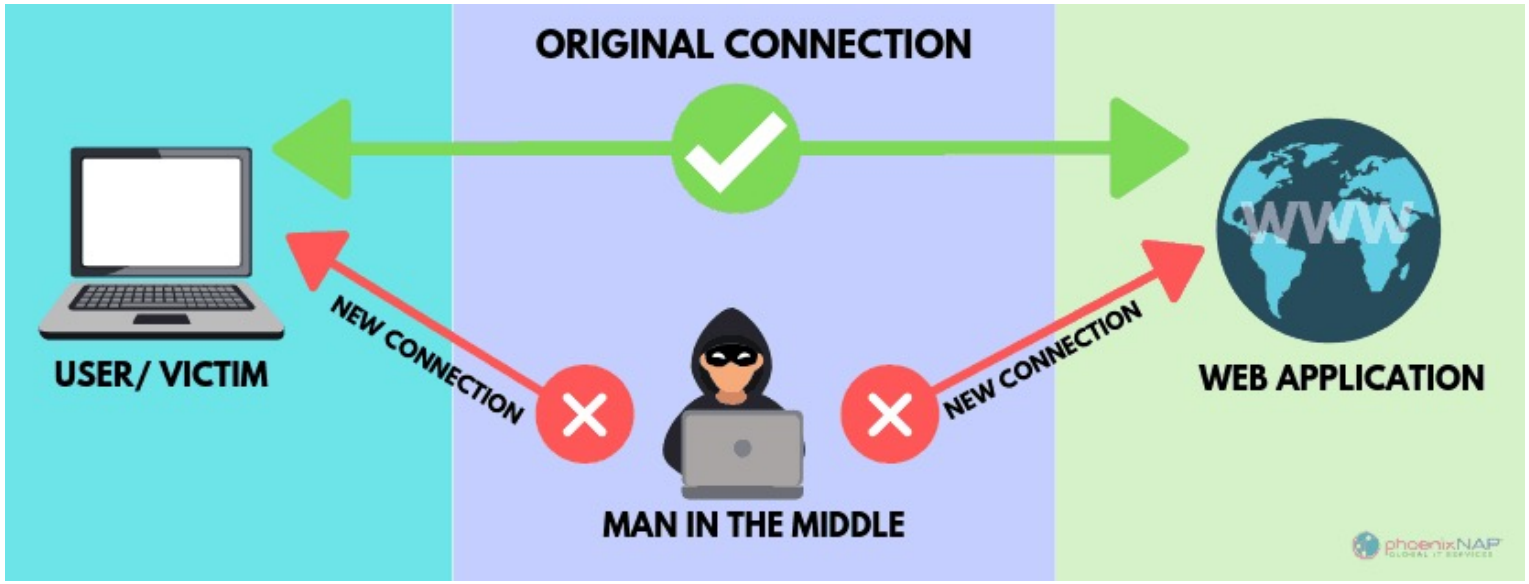
Security Violation Methods

- **Replay attack**
  - As is or with **message modification**



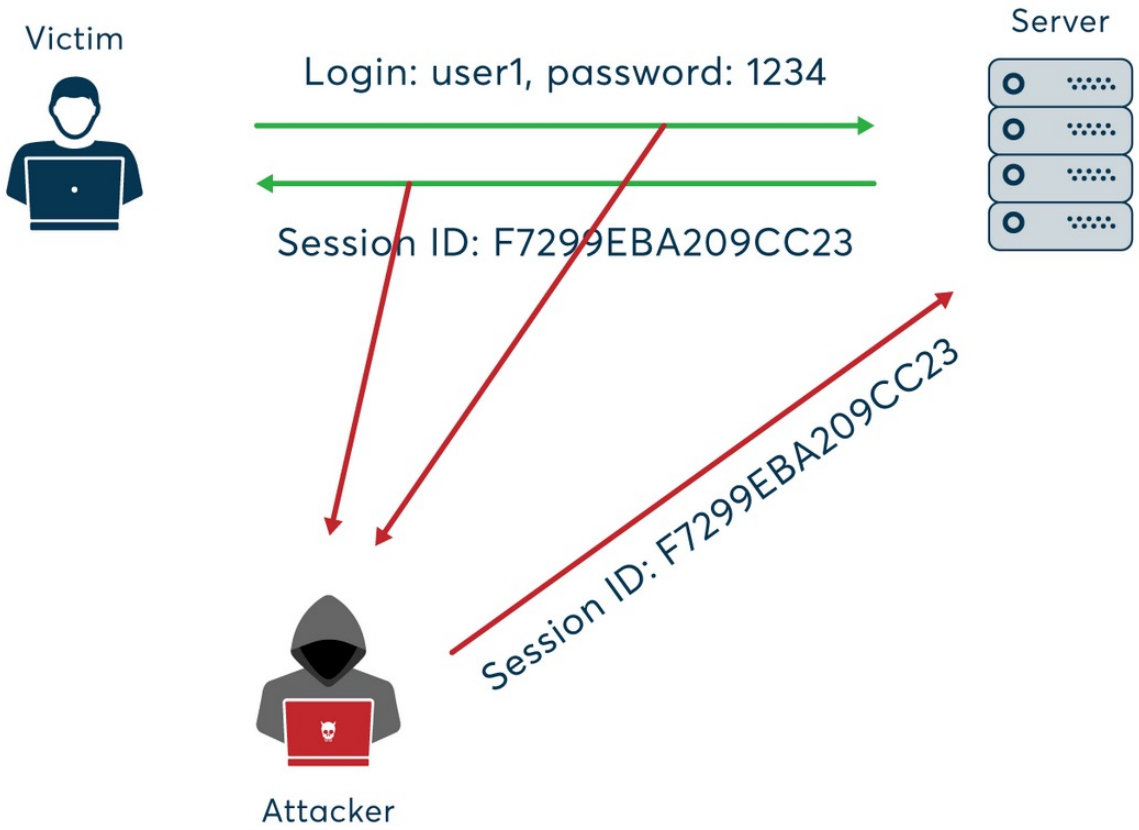
Security Violation Methods

- **Man-in-the-middle attack**
  - Intruder sits in data flow, masquerading as sender to receiver and vice versa



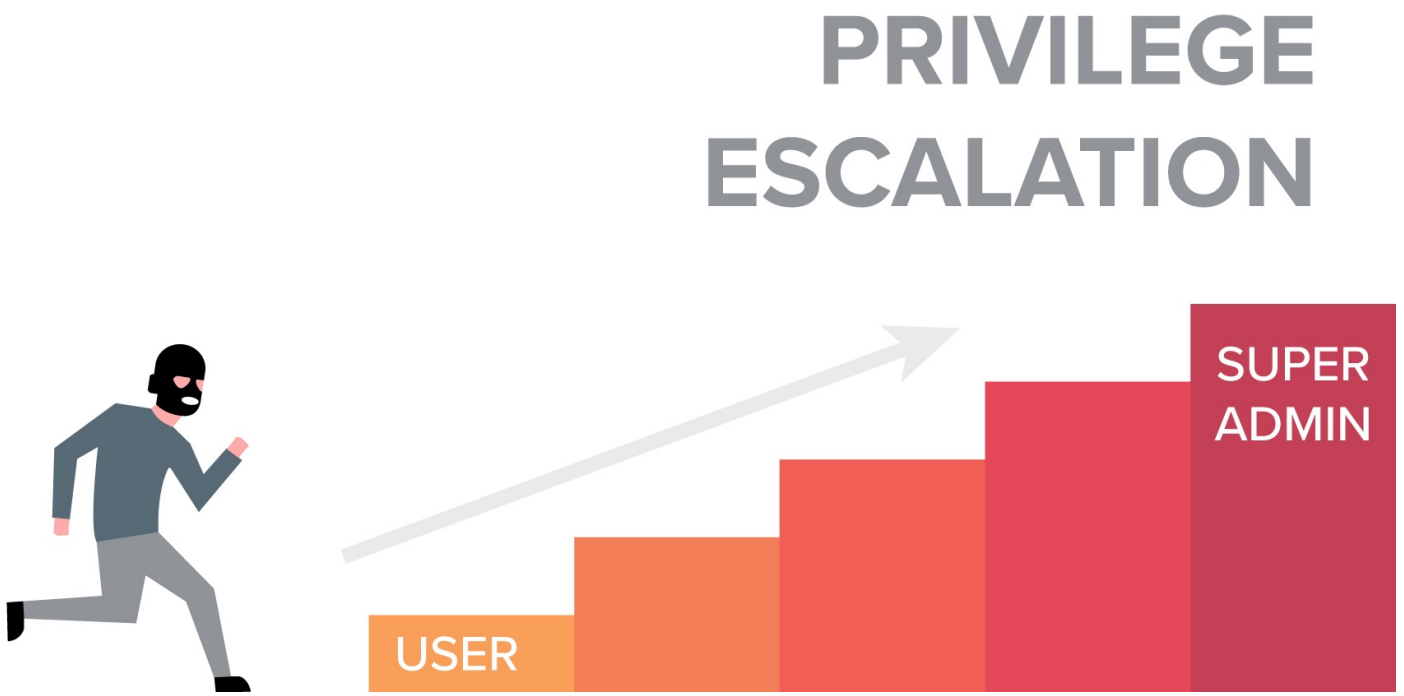
Security Violation Methods

- **Session hijacking**
  - Intercept an already-established session to bypass authentication



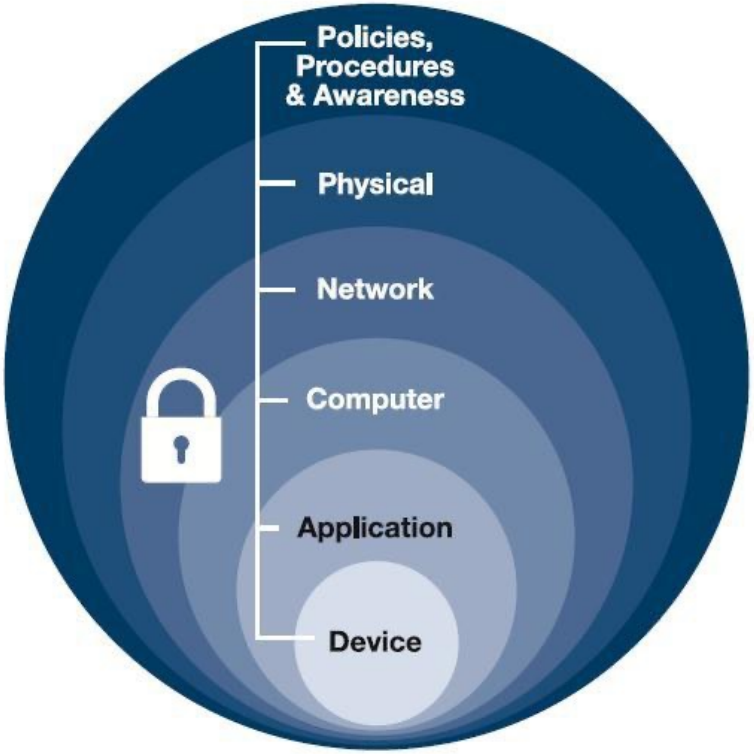
Security Violation Methods

- **Privilege escalation**
  - Common attack type with access beyond what a user or resource is supposed to have



Security Measure Levels

- Impossible to have absolute security, but make cost to perpetrator sufficiently high to deter most intruders.



Security Measure Levels

- **Physical**
  - Data centers, servers, connected terminals
- **Application**
  - Benign or malicious apps can cause security problems
- **Operating System**
  - Protection mechanisms, debugging
- **Network**
  - Intercepted communications, interruption, DOS

Security Measure Levels (cont.)

- Security is as weak as the weakest link in the chain
- Humans a risk too via **phishing** and **social-engineering** attacks
- But can too much security be a problem?

Program Threats

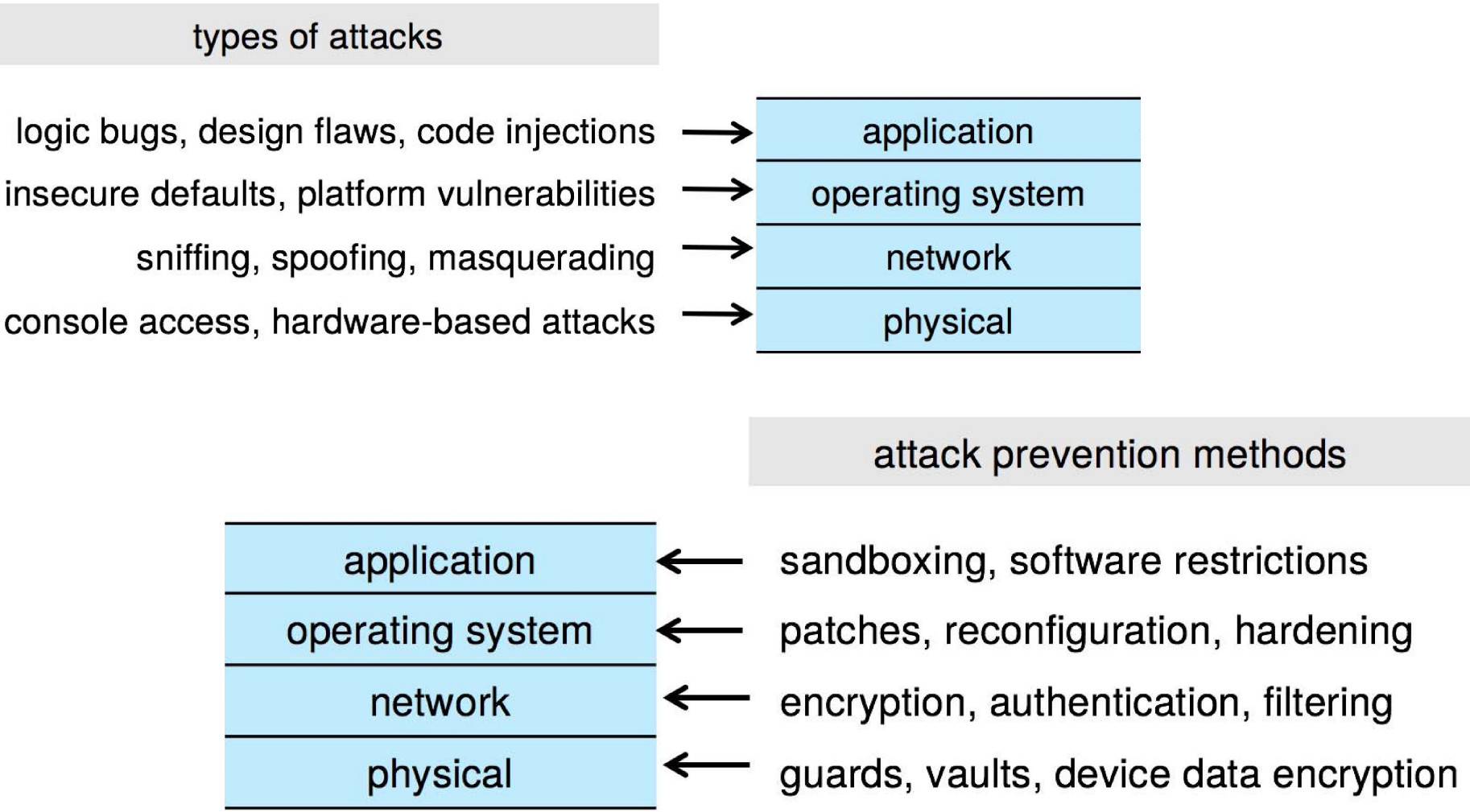
- Many variations, many names
- **Trojan Horse**
  - Code segment that misuses its environment
  - Exploits mechanisms for allowing programs written by users to be executed by other users
  - Spyware, pop-up browser windows, covert channels
  - Up to 80% of spam delivered by spyware-infected systems

https://www.kaspersky.com/resource-center/threats/trojans

Program Threats (cont.)

- Many variations, many names
- **Trap Door**
  - Specific user identifier or password that circumvents normal security procedures
  - Could be included in a compiler
  - How to detect them?

Four-layered Model of Security

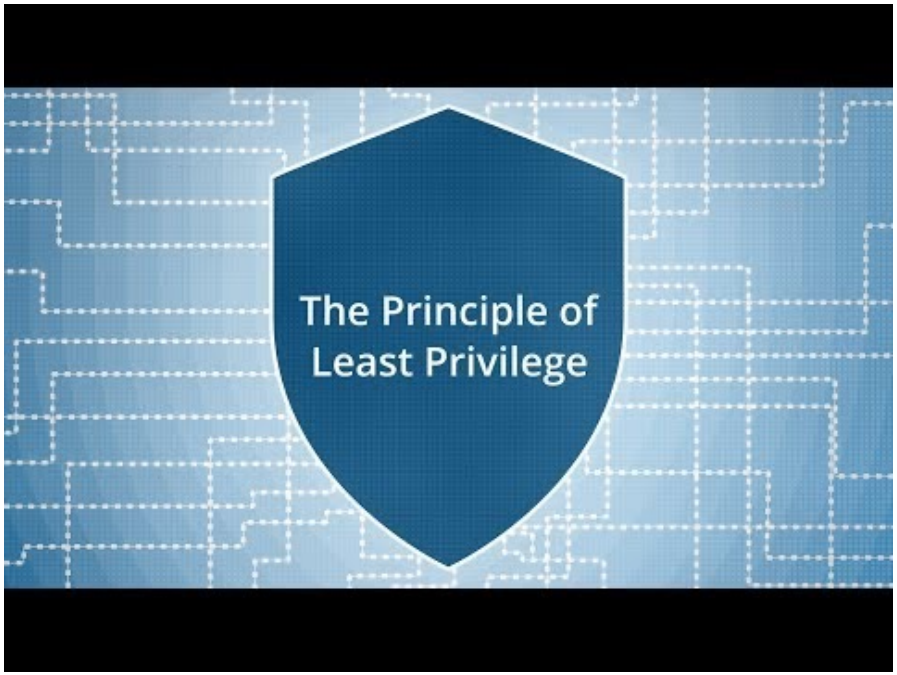


Program Threats (cont.)

- **Malware**-Software designed to exploit, disable, or damage computer
- **Trojan Horse** – Program that acts in a clandestine manner
  - **Spyware** – Program frequently installed with legitimate software to display adds, capture user data
  - **Ransomware** – locks up data via encryption, demanding payment to unlock it

Program Threats (cont.)

- Others include trap doors, logic boms
- All try to violate the Principle of Least Privilege
- Goal frequently is to leave behind Remote Access Tool (RAT) for repeated access.



Program Threats (cont.)

THE PRINCIPLE OF LEAST PRIVILEGE

"The principle of least privilege. Every program and every privileged user of the system should operate using the least amount of privilege necessary to complete the job. The purpose of this principle is to reduce the number of potential interactions among privileged programs to the minimum necessary to operate correctly, so that one may develop confidence that unintentional, unwanted, or improper uses of privilege do not occur."—Jerome H. Saltzer, describing a design principle of the Multics operating system in 1974: https://pdfs.semanticscholar.org/1c8d/06510ad449ad24fbdd164f8008cc730cab47.pdf.

C Program with Buffer-overflow Condition

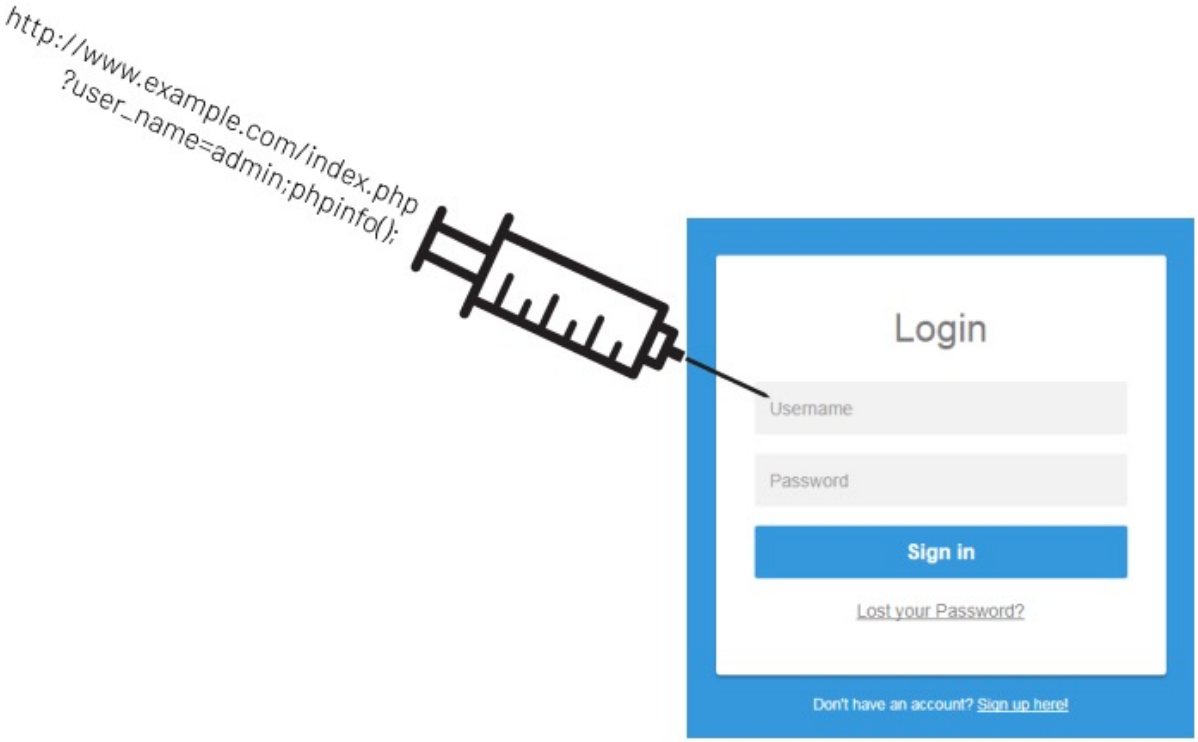
```
#include <stdio.h>
#define BUFFER SIZE 256
int main(int argc, char *argv[])
{
   char buffer[BUFFER SIZE];
   if (argc < 2)
      return -1;
   else {
      strcpy(buffer,argv[1]);
      return 0;
   }
}
```

C Program with Buffer-overflow Condition

- **Code review** can help – programmers review each other's code, looking for logic flows, programming flaws.

Code Injection

- **Code-injection attack** occurs when system code is not malicious but has bugs allowing executable code to be added or modified.



Code Injection (cont.)

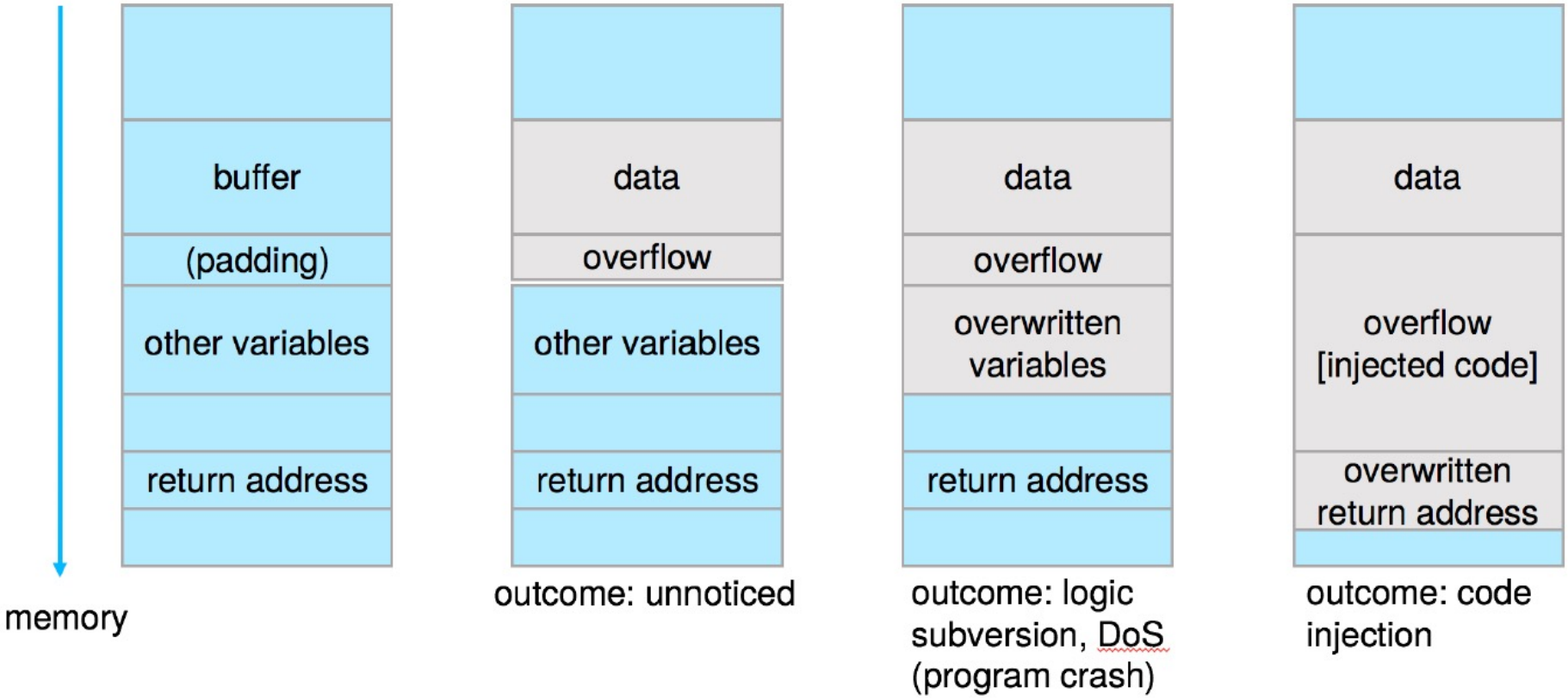
- Results from poor or insecure programming paradigms, commonly in low level languages like C or C++ which allow for direct memory access through pointers.

- Goal is a buffer overflow in which code is placed in a buffer and execution caused by the attack.

- Can be run by script kiddies – use tools written but exploit identifiers.

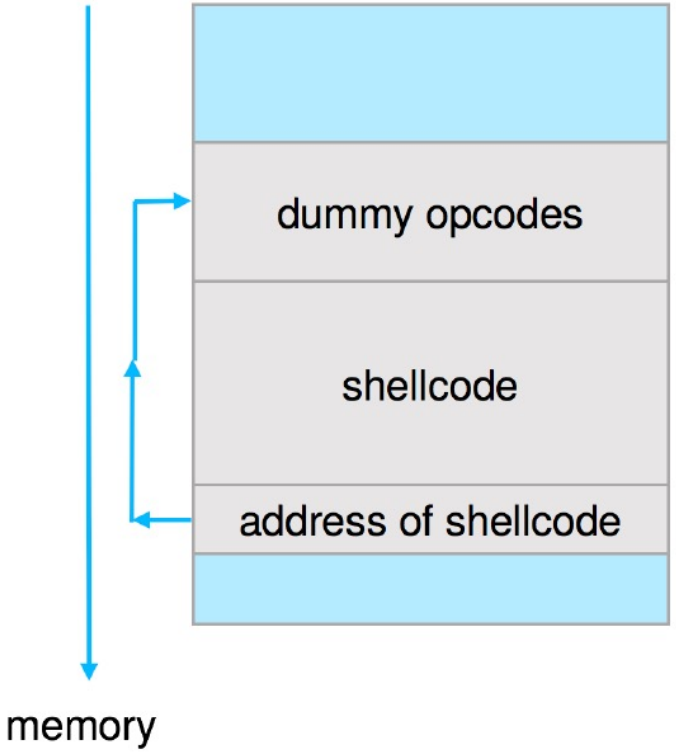
Code Injection (cont.)

- Outcomes from code injection include:



Code Injection (cont.)

- Frequently use trampoline to code execution to exploit buffer overflow:



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.

Great Programming Required?

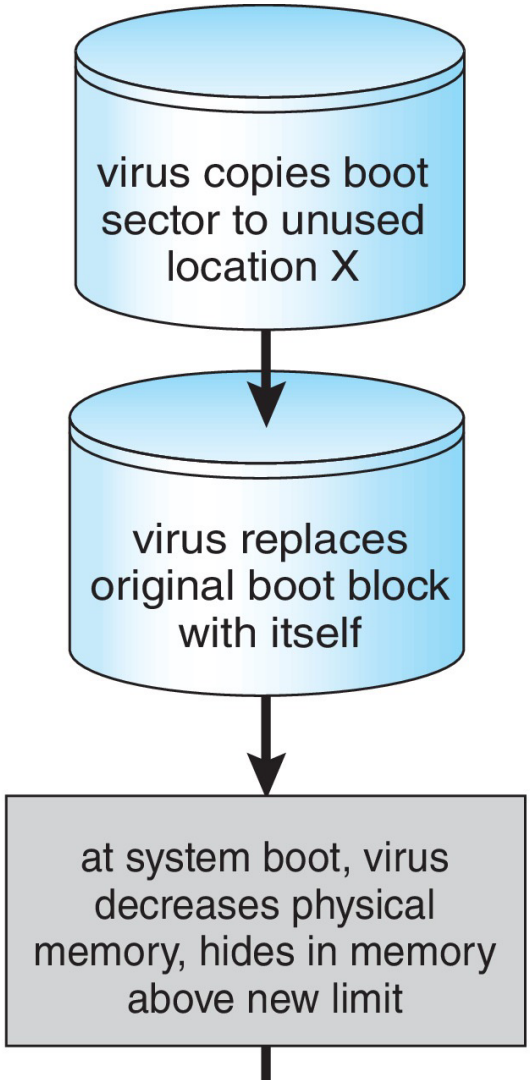
- Depending on bug, attack can be executed across a network using allowed connections, bypassing firewalls.

- 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

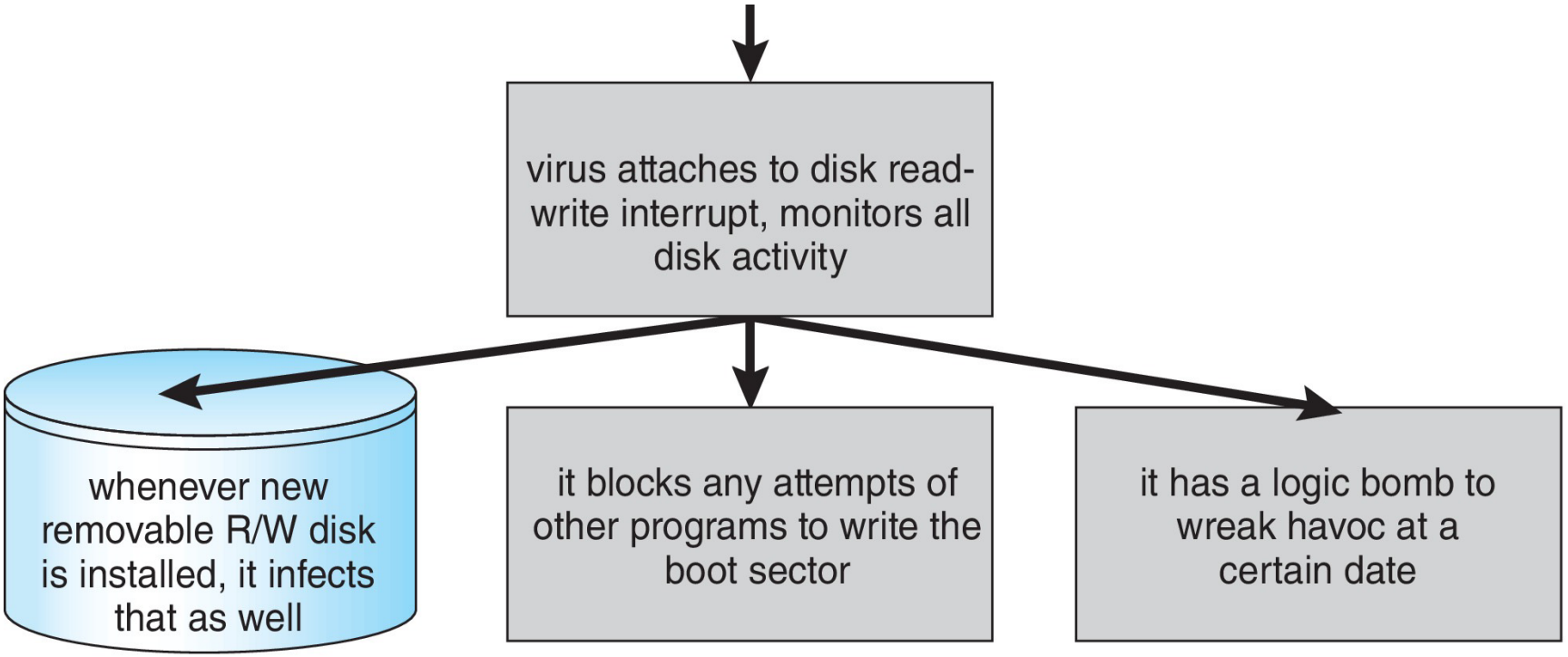
Program Threats (cont.)

- **Viruses**
  - 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
- **Virus dropper** inserts virus onto the system

A Boot-sector Computer Virus



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

The Threat Continues

- Why is Windows the target for most attacks?
  - Most common
  - Everyone is an administrator
  - **Monoculture** considered harmful
    - Many systems run the same hardware, operating system, and application software.

System and Network Threats

- Network threats harder to detect, prevent
  - Protection systems weaker
  - More difficult to have a shared secret on which to base access
  - No physical limits once system attached to internet
    - Or on network with system attached to internet
  - Even determining location of connecting system difficult
    - IP address is only knowledge

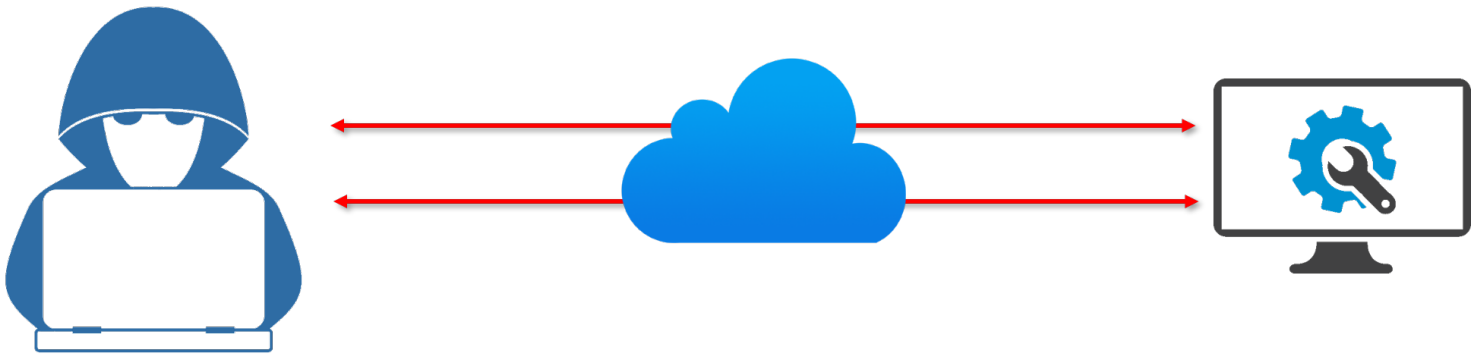
System and Network Threats (cont.)

- **Worms** – use **spawn** mechanism; standalone program

- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs
  - Exploited trust-relationship mechanism used by *rsh* to access friendly systems without use of password

System and Network Threats (cont.)

- **Port scanning**
  - Automated attempt to connect to a range of ports on one or a range of IP addresses
  - Detection of answering service protocol
  - Detection of OS and version running on system
  - …

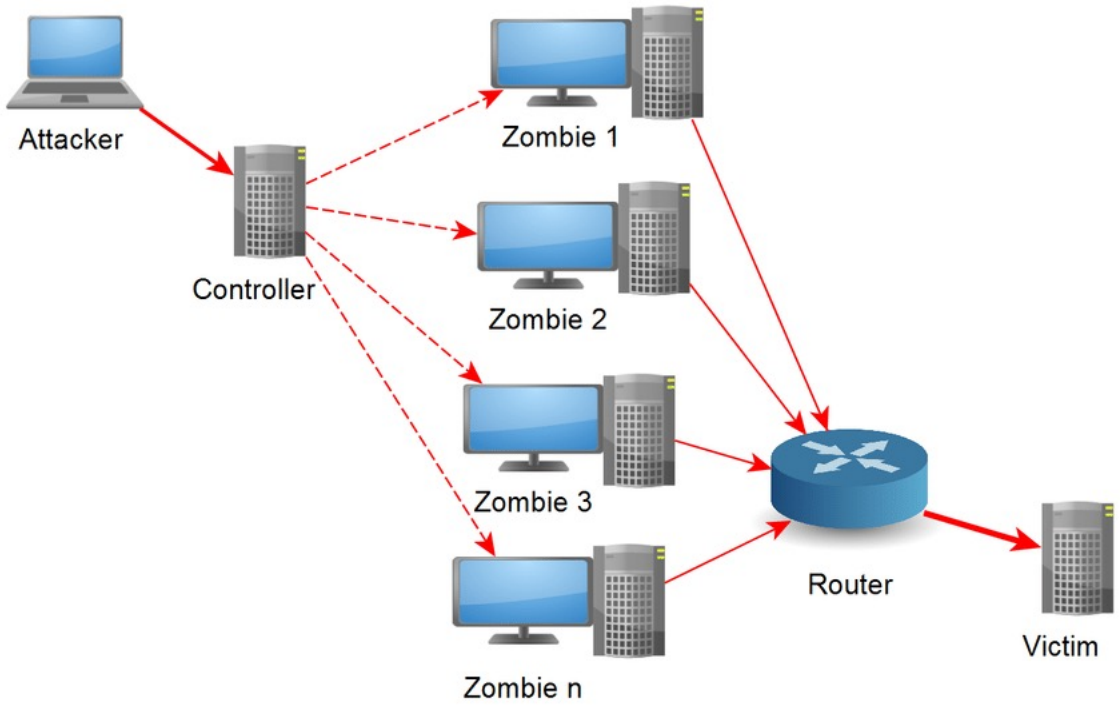


System and Network Threats (cont.)

- **Port scanning**
  - ....
  - `nmap` scans all ports in a given IP range for a response
  - `nessus` has a database of protocols and bugs (and exploits) to apply against a system
  - Frequently launched from **zombie systems**
    - To decrease trace-ability

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
  - …



System and Network Threats (cont.)

- **Denial of Service**
  - 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

Standard Security Attacks