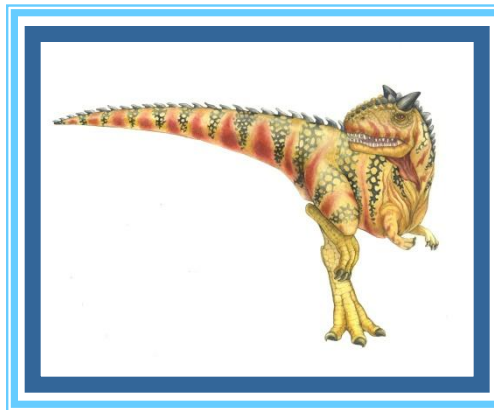


Protection & Security



Chapter 14: Protection

- Goals of Protection
- Principles of Protection
- Domain of Protection

Goals of Protection

- In one protection model, computer consists of a collection of objects, hardware or software
- Each object has a unique name and can be accessed through a well-defined set of operations
- Protection problem - ensure that each object is accessed correctly and only by those processes that are allowed to do so

Principles of Protection

- Guiding principle – **principle of least privilege**
 - Programs, users and systems should be given just enough **privileges** to perform their tasks
 - Limits damage if entity has a bug, gets abused
 - Can be static (during life of system, during life of process)
 - Or dynamic (changed by process as needed) – **domain switching, privilege escalation**
 - “Need to know” a similar concept regarding access to data

Principles of Protection (Cont.)

- Must consider “grain” aspect
 - Rough-grained privilege management easier, simpler, but least privilege now done in large chunks
 - For example, traditional Unix processes either have abilities of the associated user, or of root
 - Fine-grained management more complex, more overhead, but more protective
 - File ACL lists, RBAC
- Domain can be user, process, procedure

Chapter 15: Security

- The Security Problem
- Program Threats

Objectives

- To discuss security threats and attacks
- To explain the fundamentals of encryption, authentication, and hashing
- To examine the uses of cryptography in computing
- To 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

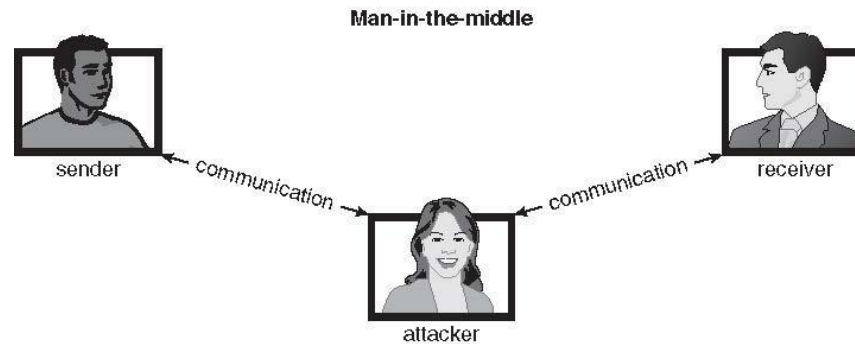
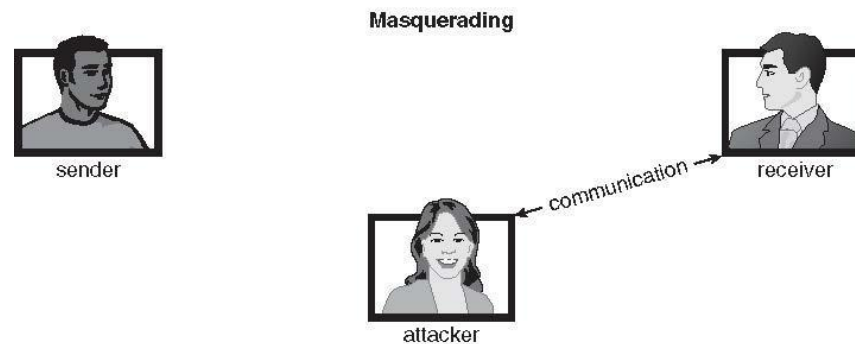
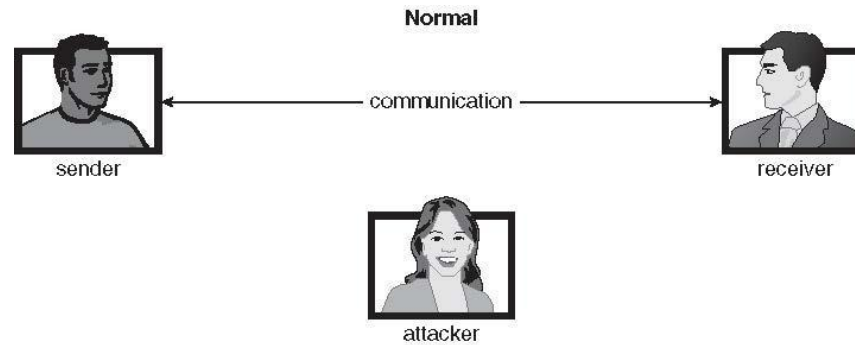
Security Violation Categories

- **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
- **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



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