OPERATING SYSTEMS

Protection & Security

The Security Problem

- Security must consider external environment of the system, and protect the system resources
- 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 Violations

- Categories
  - **Breach of confidentiality**
  - **Breach of integrity**
  - **Breach of availability**
  - **Theft of service**
  - **Denial of service**
- Methods
  - **Masquerading (breach authentication)**
  - **Replay attack**
    - **Message modification**
  - **Man-in-the-middle attack**
  - **Session hijacking**

Security Measure Levels

- Security must occur at four levels to be effective:
  - **Physical**
  - **Human**
  - **Operating System**
  - **Network**
- Security is as weak as the weakest link in the chain

Program Threats

- **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**
- **Trap Door**
  - Specific user identifier or password that circumvents normal security procedures
  - Could be included in a compiler
- **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)

Program Threats (Cont.)

- Viruses
  - Code fragment embedded in legitimate program
  - Very specific to CPU architecture, operating system, applications
  - Usually borne via email or as a macro

Program Threats (Cont.)

- **Virus dropper** inserts virus onto the system
- Many categories of viruses, literally many thousands of viruses
  - File
  - Boot
  - Macro
  - Source code
  - Polymorphic
  - Encrypted
  - Stealth
  - Tunneling
  - Multipartite

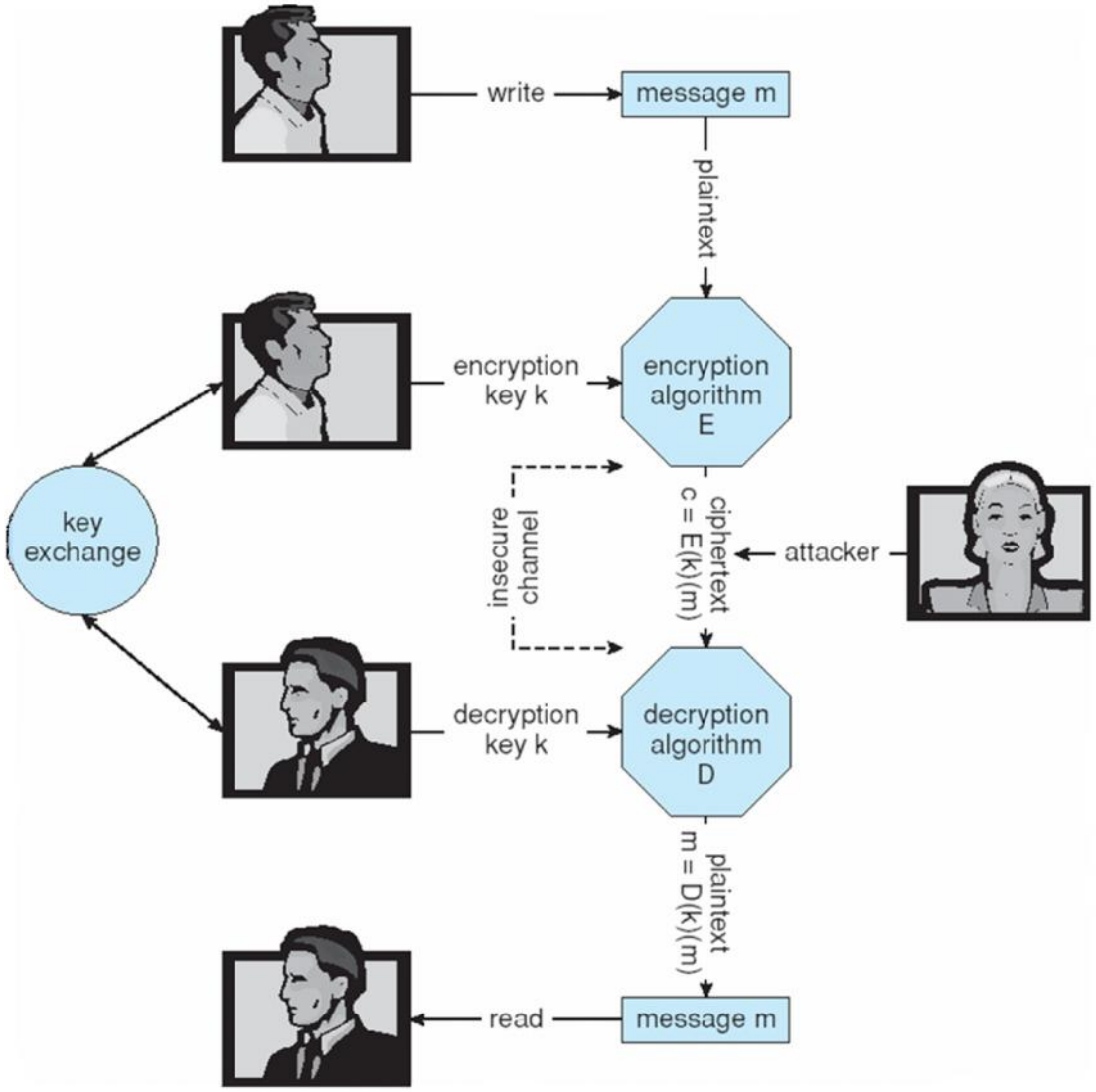
System and Network Threats

- **Worms** – use **spawn** mechanism; standalone program
- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs
  - **Grappling hook** program uploaded main worm program
- **Port scanning**
  - Automated attempt to connect to a range of ports on one or a range of IP addresses
- **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

Cryptography as a Security Tool

- Broadest security tool available
  - Source and destination of messages cannot be trusted without cryptography
  - Means to constrain potential senders (*sources*) and / or receivers (*destinations*) of *messages*
- Based on secrets (**keys**)

Secure Communication over Insecure Medium



Encryption

- **Encryption** algorithm consists of
  - Set of *K* keys
  - Set of *M* Messages
  - Set of *C* ciphertexts (encrypted messages)
  - A function $E : K \rightarrow (M \rightarrow C)$. That is, for each $k \in K$, $E(k)$ is a function for generating ciphertexts from messages
    - Both *E* and *E*(*k*) for any *k* should be efficiently computable functions
  - A function $D : K \rightarrow (C \rightarrow M)$. That is, for each *k* Č *K*, *D*(*k*) is a function for generating messages from ciphertexts
    - Both *D* and *D*(*k*) for any *k* should be efficiently computable functions
- An encryption algorithm must provide this essential property: Given a ciphertext *c* Č *C*, a computer can compute *m* such that *E*(*k*)(*m*) = *c* only if it possesses *D*(*k*).
  - Thus, a computer holding *D*(*k*) can decrypt ciphertexts to the plaintexts used to produce them, but a computer not holding *D*(*k*) cannot decrypt ciphertexts
  - Since ciphertexts are generally exposed (for example, sent on the network), it is important that it be infeasible to derive *D*(*k*) from the ciphertexts

Symmetric Encryption

- Same key used to encrypt and decrypt
  - *E*(*k*) can be derived from *D*(*k*), and vice versa
- Triple-DES considered more secure
  - DES encrypt with K_1, DES *decrypt* with K_2, then DES encrypt with K_3.
  - DES decrypt with K_3, *encrypt* with K_2, then decrypt with K_1.
- Advanced Encryption Standard (**AES**)
- RC4 is most common symmetric stream cipher, but known to have vulnerabilities
  - Encrypts/decrypts a stream of bytes (i.e., wireless transmission) – e.g.., WEP, SSL
  - Key is a input to pseudo-random-bit generator
    - Generates an infinite **keystream**

Asymmetric Encryption

- Public-key encryption based on each user having two keys:
  - public key – published key used to encrypt data
  - private key – key known only to individual user used to decrypt data
- Must be an encryption scheme that can be made public without making it easy to figure out the decryption scheme
  - Most common is RSA block cipher
  - Efficient algorithm for testing whether or not a number is prime
  - No efficient algorithm is known for finding the prime factors of a number

Authentication – Hash Functions

- Basis of authentication
- Creates small, fixed-size block of data (**message digest**, **hash value**) from *m*
- If H(m) ≠ H(m'), therefore m ≠m'
  - The message has been modified
- If H(m) = H(m'), therefore m=m'
  - The message has not been modified
- Common message-digest functions include **MD5**, which produces a 128-bit hash, and **SHA-1**, which outputs a 160-bit hash

Authentication – Digital Signature

- Based on asymmetric keys and digital signature algorithm
- Authenticators produced are **digital signatures**
- A digital signature scheme typically consists of three algorithms:
  - A key generation algorithm that selects a private key from a set of possible private keys. The algorithm outputs the private key and a corresponding public key.
  - A *signing* algorithm that, given a message and a private key, produces a signature.
  - A signature verifying algorithm that, given a message, public key and a signature, either accepts or rejects the message's claim to authenticity.
- Properties:
  - A signature generated from a fixed message and fixed private key should verify the authenticity of that message by using the corresponding public key.
  - It should be computationally infeasible to generate a valid signature for a party.

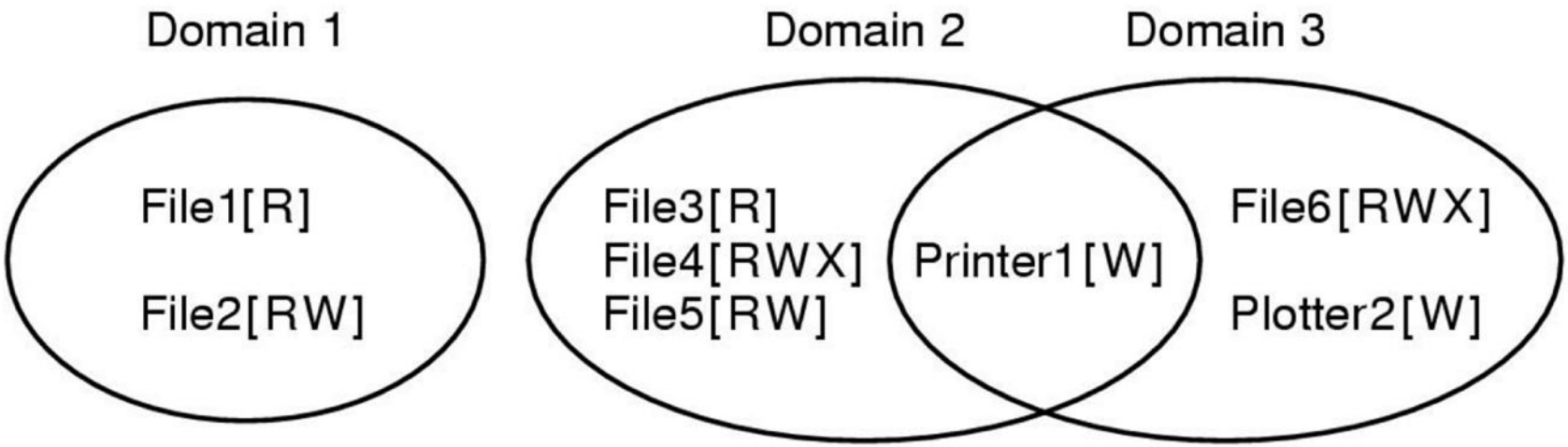
Digital Certificates

- Proof of who or what owns a public key
- Public key digitally signed a trusted party
- Trusted party receives proof of identification from entity and certifies that public key belongs to entity
- Certificate authority are trusted party – their public keys included with web browser distributions
  - They vouch for other authorities via digitally signing their keys, and so on

User Authentication

- Crucial to identify user correctly, as protection systems depend on user ID
- User identity most often established through *passwords*, can be considered a special case of either keys or capabilities
  - Also can include something user has and /or a user attribute
- Passwords must be kept secret
  - Frequent change of passwords
  - Use of “non-guessable” passwords
  - Log all invalid access attempts
- Passwords may also either be encrypted or allowed to be used only once

Protection Mechanisms
Protection Domains



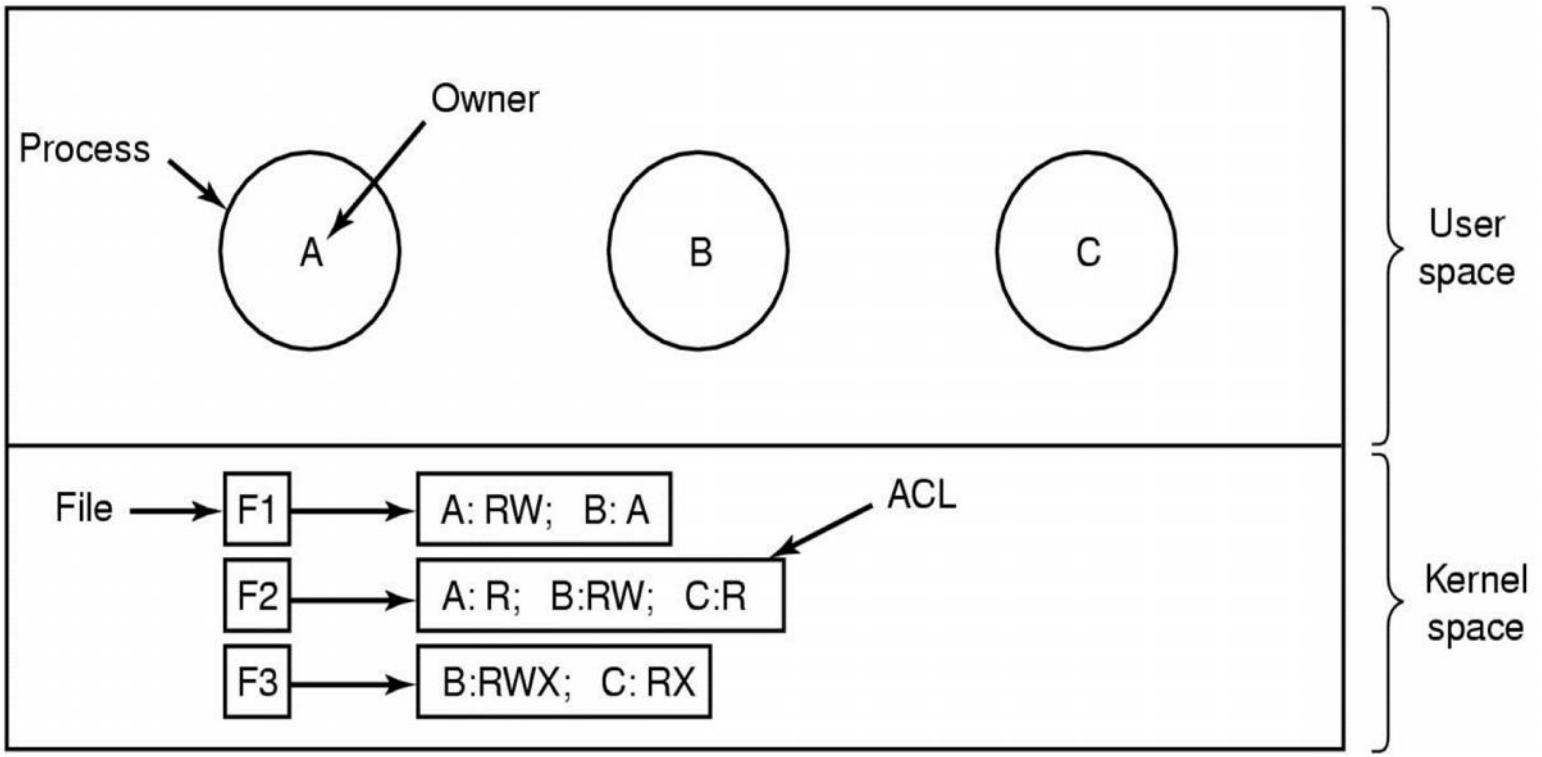
Examples of three protection domains

Protection Domains (Cont.)

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

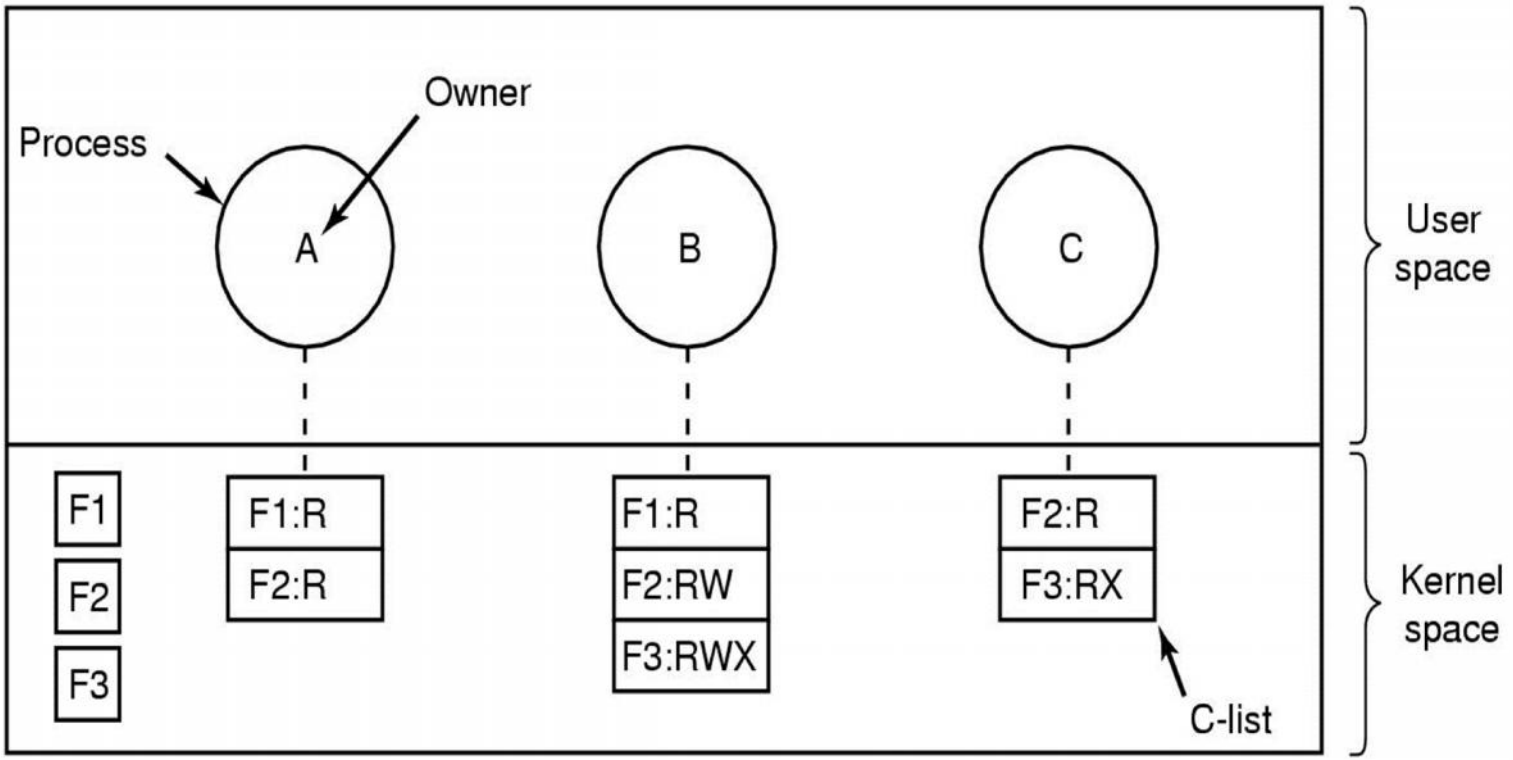
A protection matrix

Access Control Lists



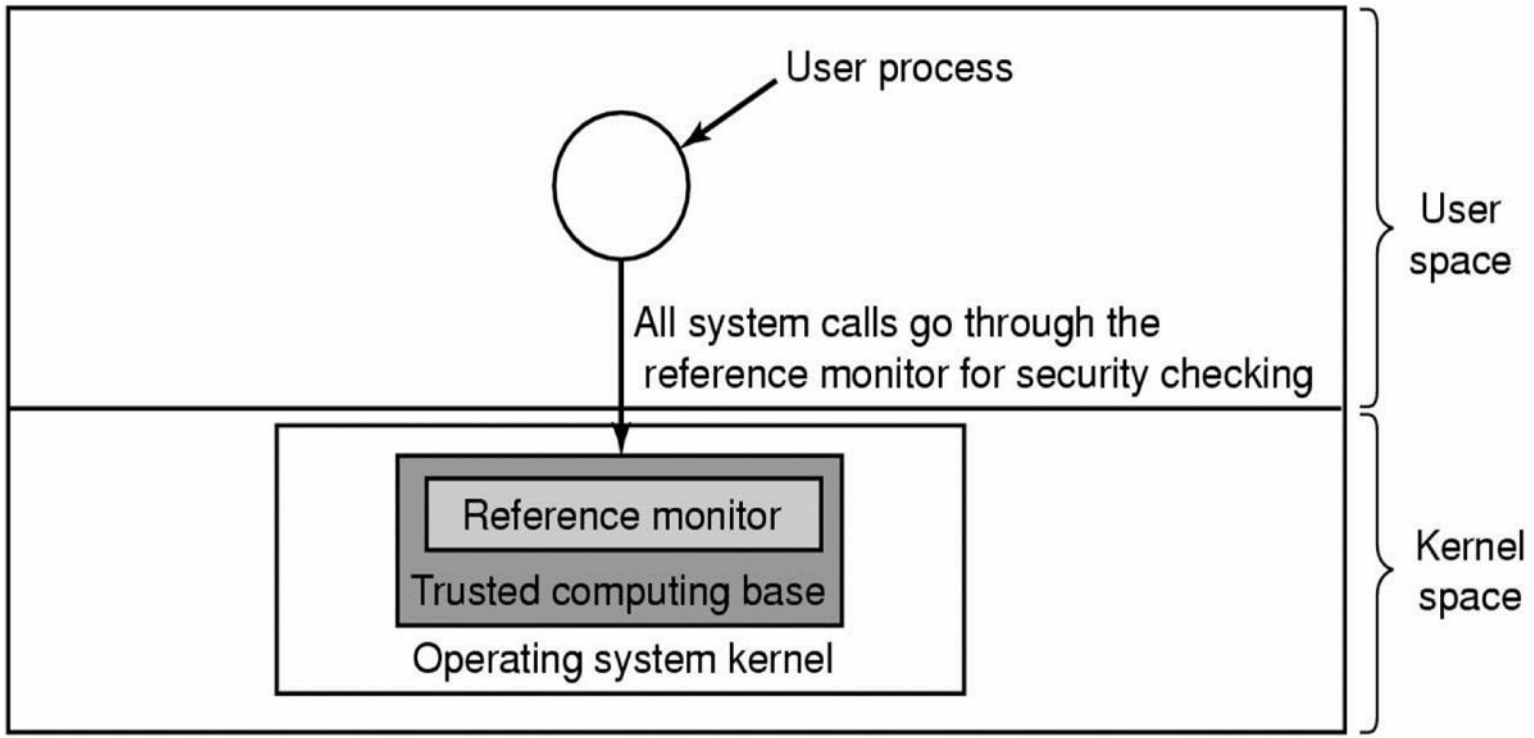
Use of access control lists of manage file access

Capability List



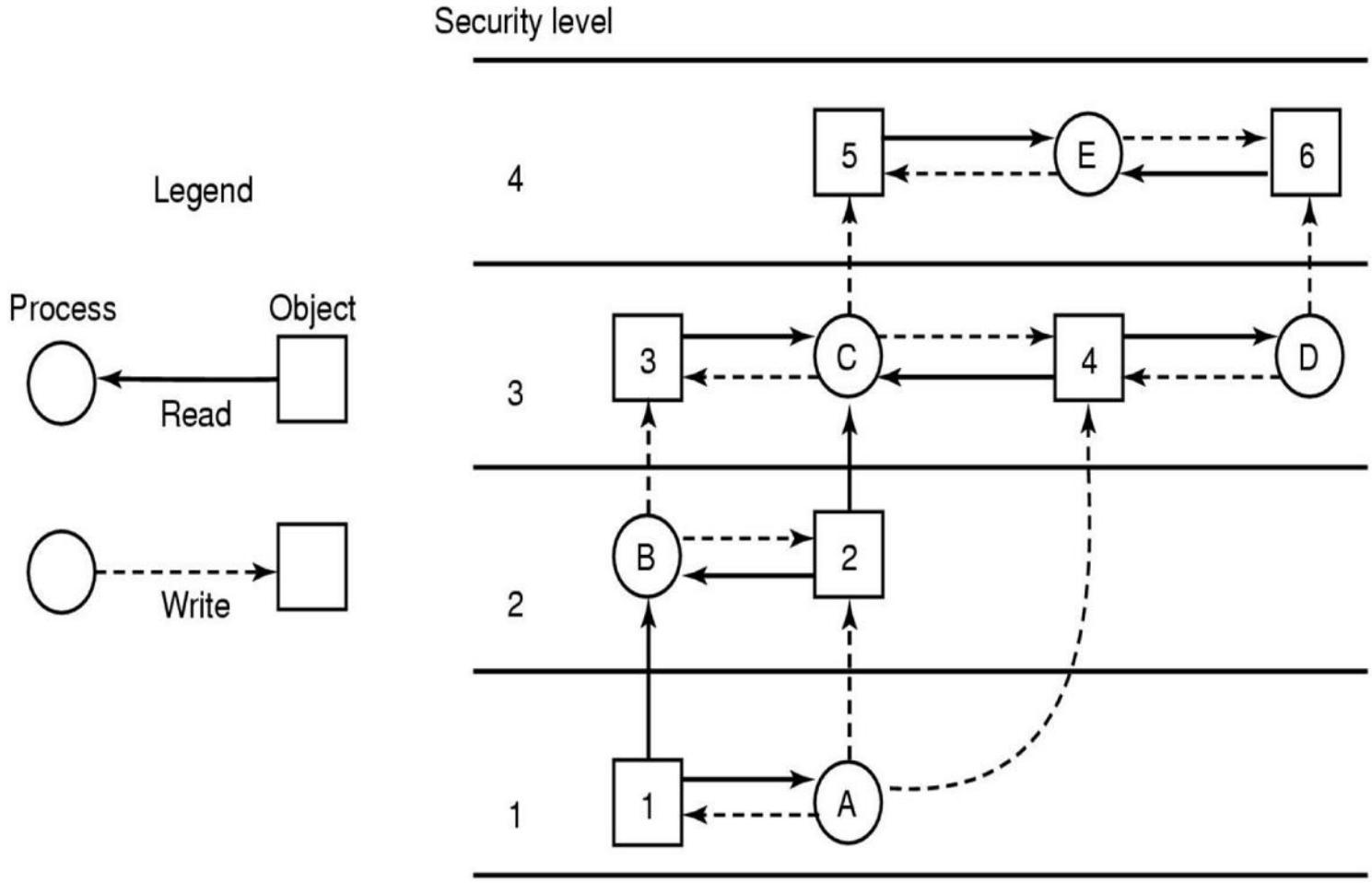
Each process has a capability list

Trusted Systems
Trusted Computing Base



A reference monitor

Multilevel Security



The Bell-La Padula multilevel security model

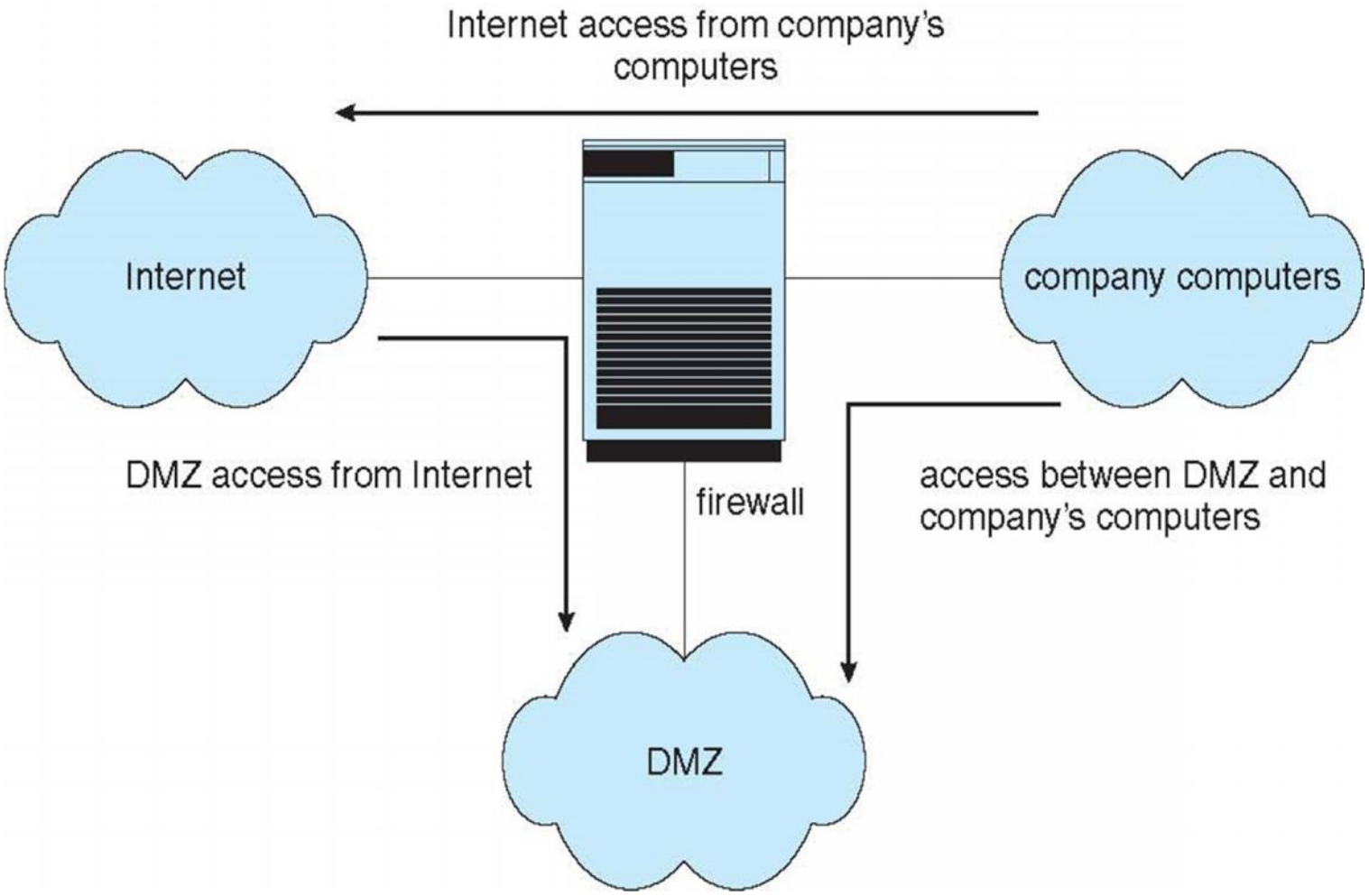
Implementing Security Defenses

- **Defense in depth** is most common security theory – multiple layers of security
- Security policy describes what is being secured
- Vulnerability assessment compares real state of system / network compared to security policy
- Intrusion detection endeavors to detect attempted or successful intrusions
  - **Signature-based** detection spots known bad patterns
  - **Anomaly detection** spots differences from normal behavior
    - Can detect **zero-day** attacks
  - **False-positives** and **false-negatives** a problem
- Virus protection
- Auditing, accounting, and logging of all or specific system or network activities

Firewalling to Protect Systems and Networks

- A network firewall is placed between trusted and untrusted hosts
  - The firewall limits network access between these two security domains
- Can be tunneled or spoofed
  - Tunneling allows disallowed protocol to travel within allowed protocol (i.e., telnet inside of HTTP)
  - Firewall rules typically based on host name or IP address which can be spoofed
- **Personal firewall** is software layer on given host
  - Can monitor / limit traffic to and from the host
- **Application proxy firewall** understands application protocol and can control them (i.e., SMTP)
- **System-call firewall** monitors all important system calls and apply rules to them (i.e., this program can execute that system call)

Network Security Through Domain Separation Via Firewall



Computer Security Classifications

- U.S. Department of Defense outlines four divisions of computer security: **A**, **B**, **C**, and **D**
- **D** – Minimal security
- **C** – Provides discretionary protection through auditing
  - Divided into **C1** and **C2**
    - **C1** identifies cooperating users with the same level of protection
    - **C2** allows user-level access control
- **B** – All the properties of **C**, however each object may have unique sensitivity labels
- **A** – Uses formal design and verification techniques to ensure security

Questions