Lecture Overview

- Security
  - Overview of security concepts
    - Goals
    - Terminology
  - Cryptography
  - User authentication
  - Attacks from inside system
  - Attacks from outside system
  - Protection mechanisms

Security

- “A computer is secure if it behaves the way that you expect it will”
- Security is concerned with ensuring our computer systems are safe for authorized use and safe from unauthorized use
- Security is often not addressed until last
  - Just like in our lecture...

Security Goals

- *Confidentiality*
  - Authorized access only
- *Integrity*
  - Authorized modification only
- *Availability*
  - Authorized access is possible
- *Authenticity*
  - Knowing someone's true identity

Assets Requiring Security

- The major security assets of a computing system
  - Hardware
    - To some degree, this is the least difficult to protect
    - We won't really talk about this
  - Software
    - More difficult to protect because software is "soft"
    - Deletion, modification, Trojan horse, virus
  - Data
    - The most difficult to protect
    - Contains the most sensitive data
    - Is widely valuable

Security Terminology

- Some terminology
  - *Exposure* - a form of possible loss or harm
  - *Vulnerability* - a weakness in the security system that can be exploited
  - *Threat* - circumstances that have the potential to cause loss
  - *Attack* - exploiting a vulnerability
  - *Control* - a protective measure to reduce vulnerability
  - *Policy* - rules on how a computer system is controlled
  - *Trust* - confidence that a computer systems is secure

Intruders

Common Categories

- Casual prying by non-technical users
- Snooping by insiders
- Determined attempt to make money
- Commercial or military espionage

Accidental Data Loss

Common Causes

- Acts of God
  - Fires, floods, wars
- Hardware or software errors
  - CPU malfunction, bad disk, program bugs
- Human errors
  - Data entry, wrong tape mounted

Security Breaches

- *Interruption*
  - An asset of a system becomes lost or unavailable
- *Interception*
  - An unauthorized party has gained access to an asset
- *Modification*
  - An unauthorized party tampers with an asset
- *Fabrication*
  - An unauthorized party introduces counterfeit objects

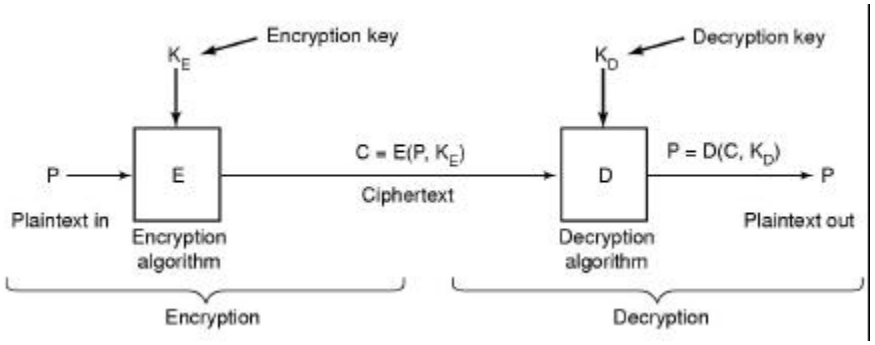
Protective Measures

- *Prevention*
  - Take measures to stop minimize risk of an asset beforehand
- *Detection*
  - Take measures to determine when an asset has become compromised
- *Reaction*
  - Take measures to recover from an asset that has become compromised

Privacy

- Another type of security concern
- Allow personal information to be used for appropriate or desired purposes only
- Much more difficult to maintain that other types of security
  - User gives to personal information access out of necessity
  - Some information is not made available, but becomes available just by using systems and services
    - For example, browsing the Web

Cryptography Overview



Relationship between the plaintext and the ciphertext

Cryptography Overview

- Conventional cryptography is based on secret key or symmetric key encryption, e.g., DES
  - One key is used to encrypt and decrypt
  - Symmetric key encryption is fast
  - Difficult to share secret keys over networks
- Public key cryptography is based on asymmetric key encryption, e.g., RSA and DSA
  - One key to encrypt and one to decrypt
  - Easy to share
  - Very slow

Cryptography Overview

- Digital signatures
  - Enabled by public key encryption
  - A recipient of information can verify authenticity of information's origin using public key
    - Provide authentication and integrity verification
  - Simplistic approach is to just use your private key to encrypt your data, which can then be decrypted with by your public key
  - A better approach is to use a hashing function, like MD5, which calculates a highly unique numeric value (a message digest) for a given input stream, then you only need to encrypt the message digest

Cryptography Overview

- Digital certificates
  - Public key systems work only if you know or trust the source of the public key
    - Digital certificates are useful when you don't
  - A certificate is data that functions as a form of credential
  - Information is included with a person's public key
    - Identity and one or more additional digital signatures
  - Certificates are signed by "well known" and "trusted" authorities
    - In the end it comes down to a human being
  - Certificates chains are created by subordinate certificate authorities

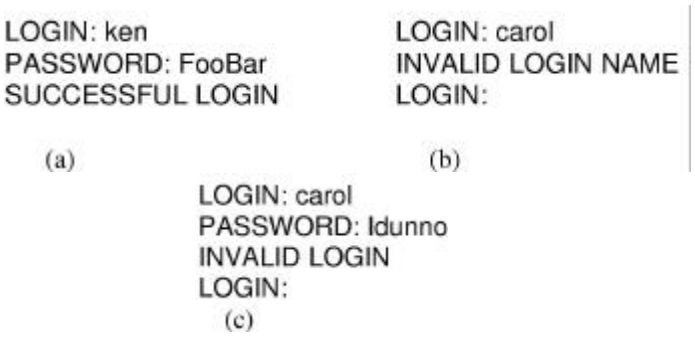
User Authentication

Authentication must identify

- Something the user knows
- Something the user has
- Something the user is

This is done before user can use the system

Authentication Using Passwords



a) A successful login

b) Login rejected after name entered

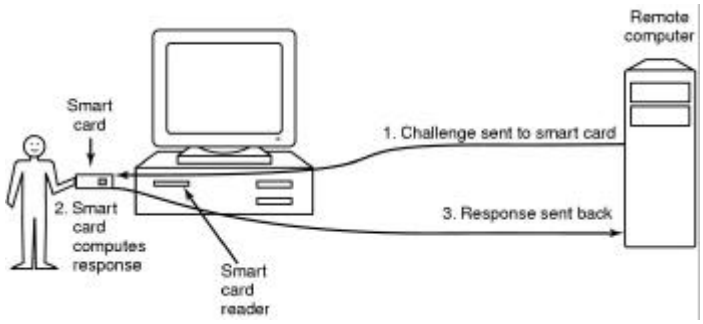
- Exposes information

c) Login rejected after name and password typed

Password Cracking

- Need a valid login identifier
  - Pretty easy to get via email, net news, etc.
- Use a large dictionary of common words and just keep trying them all
  - Can pre-compute encrypted format and just compare it to values in password file
    - Password file is normally read accessible
- Can use *salt* to foil pre-computed encrypted passwords
  - Include a randomly generated number as part of the encrypted password; salt is stored in the password file too

Authentication Using Physical Object



- Magnetic cards
  - Magnetic stripe cards
  - Chip cards: stored value cards, smart cards
- Could also use some form of biometrics
  - Voice recognition, retinal scan, etc.

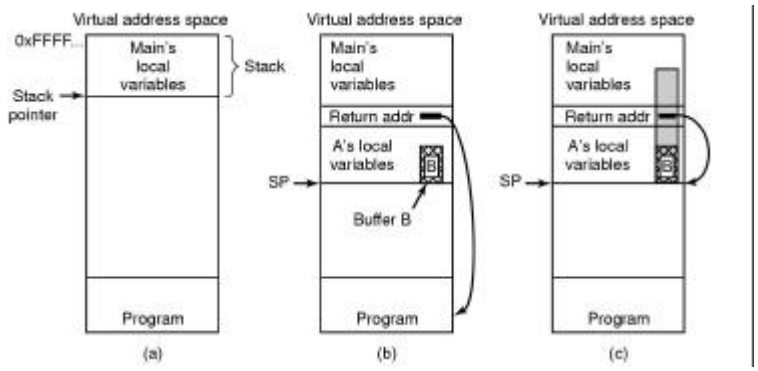
Authentication Countermeasures

- One-time passwords
- Limiting times when someone can log in
- Automatic callback at number prespecified
- Limited number of login tries
- A database of all logins
- Simple login name/password as a trap
  - Security personnel notified when attacker bites

Attacks From Inside System

- Trojan Horses
  - Malicious program masquerading as something benign
- Login spoofing
  - Using a fake login screen to collect passwords
- Logic bombs
  - Code in a program that is causes problems if/when certain conditions are met
- Trap doors
  - Code to bypass security mechanisms
- Covert channels
  - Using some obscure mechanism to send data, like response time or file locking

Buffer Overflow



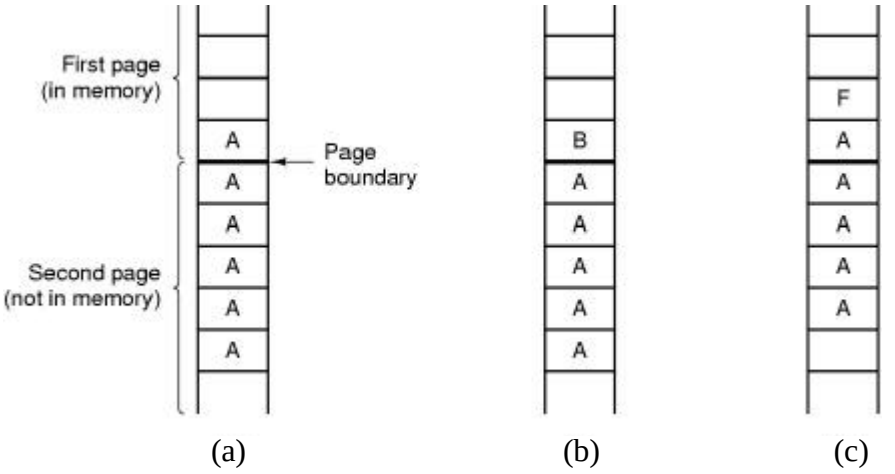
a) Situation when main program is running
b) After program *A* called
c) Buffer overflow shown in gray
   – return address is overwritten with new address

General Security Attacks

Typical attacks

- Request memory, disk space, tapes and just read
- Try illegal system calls
- Start a login and hit DEL, RUBOUT, or BREAK
- Try modifying complex OS structures
- Try to do specifically what you are told not to do

Famous Security Flaw



The TENEX – password problem

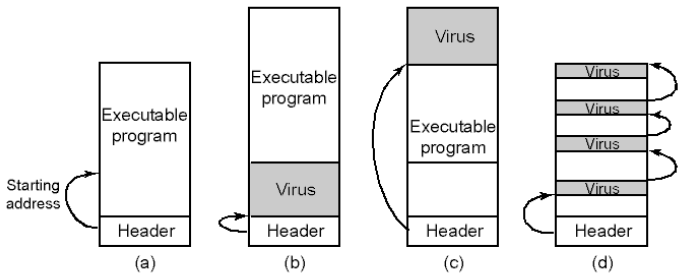
Design Principles for Security

- System design should be public
- Default should be: *no access*
- Check for current authority
- Give each process least privilege possible
- Protection mechanism should be
  - Simple
  - Uniform
  - In lowest layers of system
- Scheme should be psychologically acceptable
- Keep it simple

Attacks From Outside System

- Mobile code
- Password cracking
- Denial of service
- Viruses
  - Goals
    - Spread quickly virus
    - Difficult to detect
    - Hard to get rid of
  - Virus = program can reproduce itself
    - Attach its code to another program
    - Additionally, do harm

How a Virus Infects

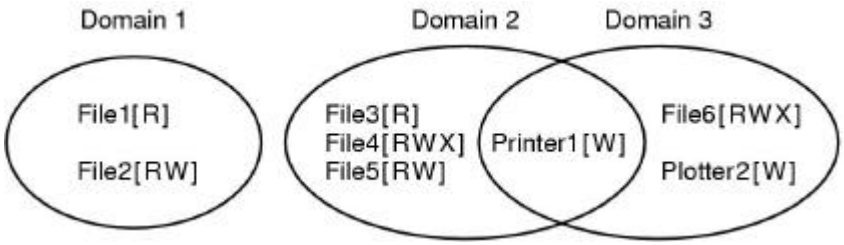


a) An executable program
b) With a virus at the front
c) With the virus at the end
d) With a virus spread over free space within program

Protection Mechanism

- Policy versus mechanism
  - *Policy* = whose data are protected from whom
  - *Mechanism* = how the policy is enforced
- Protection domains
  - A computer system has many "objects" that must be protected; this includes hardware and software
  - Each object has a unique name and a finite set of operations
  - A domain is a set of *(object, rights)* pairs
    - A right is a permission to perform an operation on an object

Protection Domains



Examples of three protection domains

Protection Domains

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

A protection matrix

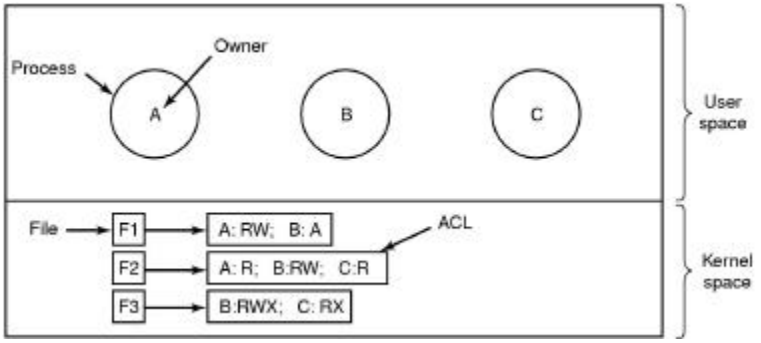
Protection Domains

Domain	Object: File1	File2	File3	File4	File5	File6	Printer1	Plotter2	Domain1	Domain2	Domain3
1	Read	Read Write								Enter	
2			Read	Read Write Execute	Read Write		Write				
3						Read Write Execute	Write	Write			

A protection matrix with domains as objects

- This makes it possible to model operations on domains themselves
  - Such as domain switching using “enter” operation

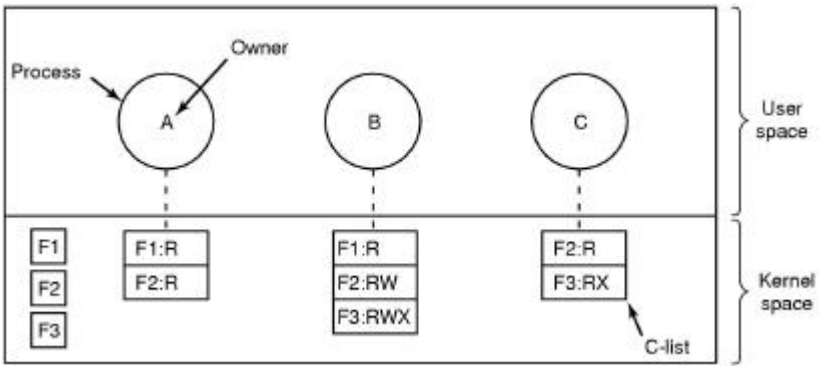
Access Control Lists



Use of access control lists to manage file access

- Lists all processes/users who have access
- ACLs are a column-wise view of the protection matrix
- Centralized

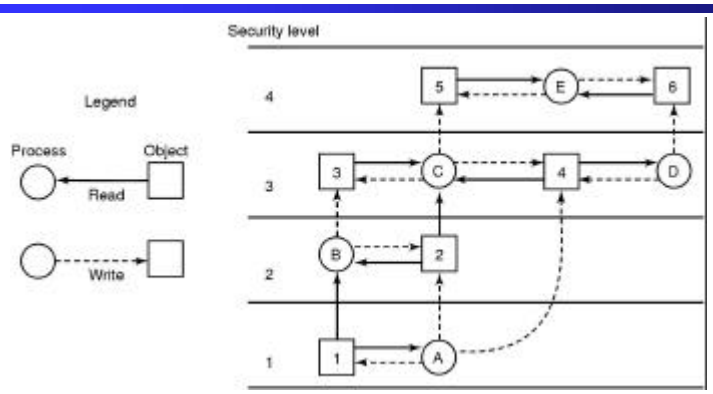
Capabilities



Each process has a capability list

- A capability is a right to access an object
- Capabilities are a row-wise view of protection matrix
- Decentralized

Multilevel Security



The *Bell-La Padula* multilevel security model

- Can only read from equal or lower levels
- Can only write to equal or higher levels
- Intended to keep secrets

Multilevel Security

The *Biba* multilevel security model

- Reverses Bell-La Padula model
- Can only read from equal or higher levels
- Can only write to equal or lower levels
- Intended to guarantee integrity

Security Conclusion

- Security is largely an afterthought
  - It must be considered from the initial design of an OS
- There is no specific rule that you can follow to create a secure OS
  - Best method is to keep it simple