



# CS 3530 Operating Systems

## Computing Security Part 2

CS 3530 Operating Systems. Dr. Ken Hoganson, Copyright © 2008

# Security Policy

- Safeguard
- Defeat
- Detect
  
- Define Policies that govern access and process.

- **password**  
Access to data or files governed by password.
  
- **User ID**  
User IDs can have specified (limited) access and privileges:
  - Limit to read only or allow write, etc.
  
- **User ID within Groups**
  - Specify privileges and capabilities based on user IDs.
  - Define groups, and assign user IDs to groups
  - Easy to ensure that proper access is granted/restricted when creating new user IDs: assign to existing group based on their work function.

- Owner
- Group
- Public
- System Admin/Database Admin
  
- Controls:
  - Read
  - Write
  - Modify
  - Delete

# File Protection

- Protection assigned to files
- Stored in the file header information
- Limits capabilities

Read	Write	Execute	
0	0	0	No access allowed
0	0	1	Execute Only
0	1	0	Write Only – make sense?
0	1	1	Write-Execute – useful?
1	0	0	Read Only
1	0	1	Read-Execute
1	1	0	Read-Write – <i>data file</i>
1	1	1	Read-Write-Execute – <i>no protection</i>

# File Protection

- Protection assigned to files
- Stored in the file header information
- Limits capabilities

RWE

<b>Owner</b>	<b>Group</b>	<b>Public</b>	<b>Filename</b>
111	110	100	fileXYZ
110	100	000	fileABC

- Defines the set of objects that a process can access and how they can be accessed
- Object Granularity
  - Computer
  - File
  - Record in a file
  - Field in a record

- **File Attributes (previously mentioned)**
  - Each file specify attributes for Owner, Group, Everyone
- **Access Control Lists**
  - Associate a list with each protected resource
  - Specifies the access rights for IDs and/or groups

## **MS Windows**

- NTFS File System
  - Access Control Lists
- FAT file systems
  - no support for file protection

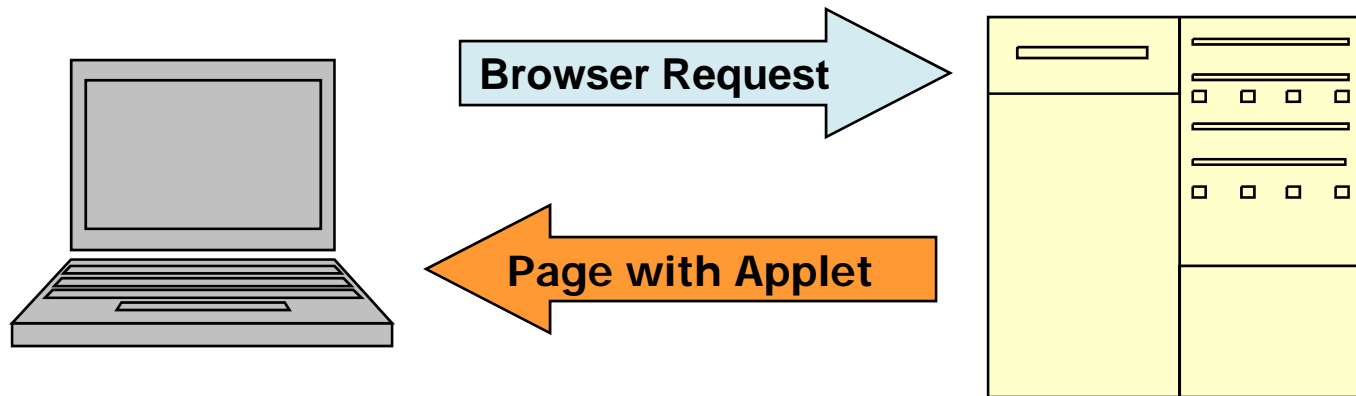
## **Linux/Unix/Mainframe/Minicomputer**

- File/object attribute protection, based on owner, group, public groups.

## **Database Systems**

- Control down to record, some support control down to fields

# Web Browsers



- Web Browsers can execute programs (Applets) that are downloaded from other machines/internet
  - Java Applets
  - ActiveX Components (Internet Explorer)
- Potential Vulnerability

- Run applets within the limited environment defined by the Java Virtual Machine (JVM)
  - JVM restricts applet capabilities
  - Unsigned applets cannot access files or other resources on the browser's machine: only on the web server that hosts the applet
  - A Signed Applet can access files/resources
    - Process for creating a signed applet
    - Users can still limit even signed applets

- Execute native code
  - Does not run in a restricted/protected environment
  - Should only allow SIGNED ActiveX components.
- A signed component includes a Digital Signature by the author
- Digital Signatures can be checked/verified – browsers have the capability to do that.

- Similar to Viruses
- Purpose is to return info to originator
  - Can be data on user: password, IDs
  - Can be data on search history – perhaps marketing use
  - Can be data on files/system configuration/installed program

- **Social Engineering**
- Take advantage of human behavior to trick users to reveal sensitive info
- Fraudulently masquerade as a know and trusted system or vendor (banks, paypal, etc)

- **Anti-Virus Software: identify, neutralize, eliminate**
- **Now protect against malware:**
  - Virus, worm, trojan horse
- **Check program code/access areas for known vulnerabilities attempting to exploit**
- **Scans files, memory, the OS, registry, etc.**
- **Look for suspicious behavior**
- **Identify known viruses (and types) by a “signature” – requires frequent updates**
- **Virus writers starting to be able to trick the signature detection mechanisms by modifying**

- A Firewall prevents unauthorized access and communications
- Hardware firewall sits between the internet and internal resources. All comms go through firewall for checking.
- Software firewall can be installed on PCs to provide a level of protection
- Often used to protect resources by doing a network address translation: public address must be translated to a valid (and hidden)

# Encryption

- Hide information by algorithm to replace characters/bits to make message unintelligible
- Requires a program to translate and encrypt a message or file:  
     $\text{Encrypt}(\text{plaintext}, \text{key1}) \rightarrow \text{ciphertext}$
- Requires a program to decrypt  
     $\text{Decrypt}(\text{ciphertext}, \text{key2}) \rightarrow \text{plaintext}$

- A single key used to both encrypt and decrypt
- Obviously both sender and recipient must have the key
- Requires a secure channel to distribute the key
- Examples:
  - DES (Data Encryption Standard)
    - 56 bit keys
    - 30 years old
  - AES (Advanced Encryption Standard)
    - Keys can be 128, 192, or 256 bits
    - New standard
- More bits/better security in general.

The original key must be distributed to both

- Two different keys, one for sender and recipient.
- Keys are mathematically related, and “undo” each other
- One key is public – used to encrypt
- The other is private, used to decrypt
- Also known as Public Key encryption

- Public/Private keys used in pairs:
- Mike needs to send a message to Ron:
  - Mike encrypts the message with Ron's public key
  - Only Ron can decrypt the message using his secret private key
  - Others with public key cannot decrypt – only the private key
- Ron needs to reply:
  - Ron encrypts message with Mike's public key
  - Only Mike can decrypt the message

- Can be used to “sign” a document to verify the originator
- Allows anyone to verify the originator, but only recipient can decrypt
- Compute a hash of the document
  - Creates a fixed-length string, unique to the document, but cannot create doc from hash
  - Encrypts hash with private key
  - Transmit encrypted message with hash
- Recipient
  - Decrypt the hash using public key
  - Decrypt the message, compute hash

- **Examples**
  - **RSA-1 (Rabin, Shamir, Adelman)**
    - Based on the difficulty of factoring very large numbers
  - **ECC (Elliptic Curve Cryptography)**
    - Based on the difficulty of calculating the coefficients of an ellipse
- **Weaknesses in both have been detected mathematically.**
  - New RSA-3 is being developed.

- A Digital Certificate identifies the author of a Signed Component
  - Contains name of the author
  - Public key of the author
  - The Certificate Authority (CA) that validated the author's public key
  - Signed by Certificate Authority
- Can be verified

- Get name of Certificate Authority
- Lookup the authority's public key
- Use public key to decrypt certificate hash
- Then verify the hash
- Proves digital certificate is valid

- Private keys must be kept secret and guarded
- A break in the other security components can allow unauthorized access to the secret key
- Often, the creation of public/private keys is done on a separate secure machines.
- Sometimes, encryption and decryption are don and separate secure machines.
- I have a certificate authority machine in my office for creating our cluster's grid certificate.
  - Machine is not networked, is turned-off, sits in a corner, no keyboard, power, etc.

- Rather than trying to detect malware:
- Instead the system only executes approved and known code
  
- A more rigorous approach
- Requires that all SW use a Digital Signature and cryptography
- Implementation is difficult – standards at OS and application levels – many vendors

**End**

**Of**

**Today's**

**Lecture.**



This slide intentionally left blank