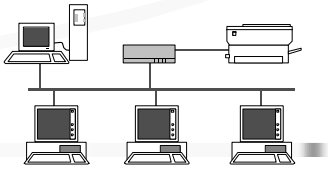


Operating Systems
600.418

Protection and Security

Department of Computer Science
The Johns Hopkins University

Protection and Security

Lecture 9

Reading:
Silberschatz & Galvin
chapter 19, 20
Additional Reading:
Stallings
chapter 15

Protection

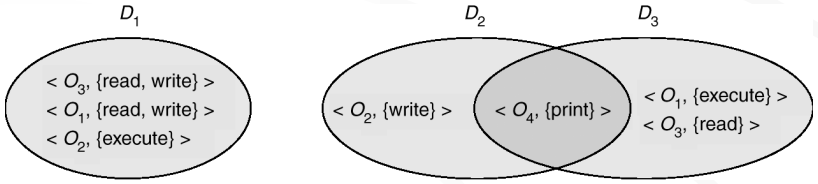
- A mechanism for controlling the access of programs, processes, or users to the resources defined by the computer system.
- The protection mechanism ensures the enforcement of a certain policy.
- Policy is established in several ways:
  - **System design.**
  - **System management.**
  - **User defined.**

Domain of Protection

- A Computer System is a collection of:
  - **Processes**
  - **Objects, each of which has a unique name, and can be accessed through a well-defined set of operations.**
    - Hardware objects: CPU, memory segments, printers, disks.
    - Software objects: files, programs, semaphores.
- Protection domain:
  - **An abstract notion.**
  - **A process operates within a protection domain.**
  - **The domain specifies the resources that the process may access.**

Domain Structure

- Access right:
  - **The ability to execute an operation on an object.**
  - **Defined as <object-name, right-set>**
- Domain is a set of access rights.
- Example:
  - **a process executing in D1 can read and write Object 3.**

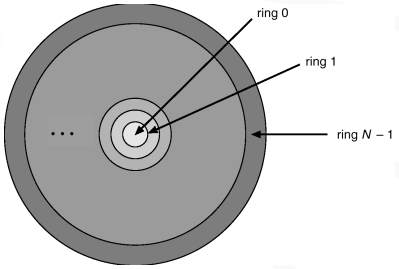


Domain Implementation in Unix

- Domain is associated with users.
- Processes are generally executed in the domain of their users.
- Each file has permission bits for owner, group and world (rwxrwxrwx).
- A switch from a domain to another domain can be accomplished via the file system:
  - If *setuid* bit on a program file is set, the program is executed in the domain of its owner, rather than in the domain of the process' user.
  - When would that be useful ?

Domain Implementation in Multics

- Protection domains are organized hierarchically.
- If i < j, then Dj is a subset of Di.
- D_0 is the most privileged domain.
- What is the limitation of this method?



Access Matrix

- An abstract, general representation of the protection domains model.
  - **Generally, the users determine the content of the column for objects they create.**

object / domain	F_1	F_2	F_3	printer
D_1	read		read	
D_2				print
D_3		read	execute	
D_4	read write		read write	

Access Matrix (cont.)

- A process may switch from a domain to another domain while it executes.
  - **Domains can be viewed as objects.**

domain \ object	F_1	F_2	F_3	laser printer	D_1	D_2	D_3	D_4
D_1	read		read			switch		
D_2				print			switch	switch
D_3		read	execute					
D_4	read write		read write		switch			

The Access Matrix is Dynamic

- Operations to add, delete rights
- Special access rights:
  - Copy: the ability to copy an access right from one domain to another
  - Owner: A process executing in a domain with owner right can add or delete rights for this object.
  - Control: applicable to Domain objects. The ability to modify the access rights for one domain while executing in another.

Copy

domain \ object	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute		

(a)

domain \ object	F_1	F_2	F_3
D_1	execute		write*
D_2	execute	read*	execute
D_3	execute	read	

(b)

Owner

domain \ object	F_1	F_2	F_3
D_1	owner execute		write
D_2		read* owner	read* owner write*
D_3	execute		

(a)

domain \ object	F_1	F_2	F_3
D_1	owner execute		
D_2		owner read* write*	read* owner write*
D_3		write	write

(b)

Implementations of Access Matix

- Global table:
  - **A set of ordered triples <domain, object, right-set>. (a cell in the martix).**
  - **Difficult to take advantage of grouping of objects and/or domains (if a particular object can be read by everyone it must have a separate entry in every domain).**
  - **Table is usually large (storage).**
- Access control list:
  - **A list of sets (one per object). Each set has ordered pairs <domain, right-set>. (a column in the matrix).**
  - **Domains with no access to this objects can be discarded.**
  - **Can be improved by having a default set that applies to all objects.**

Implementations of Access Matix (cont.)

- Capability list:
  - **A list of sets (one per domain). Each set has a capability list of the type <object, right-set>. (a row in the matrix).**
  - **Do not correspond directly to the need of a user (users work with objects).**
  - **Useful to localize access information for a process (because it executes in one domain).**
- A lock-key mechanism:
  - **A compromise between access lists and capability lists.**
  - **Each object has a list of unique *locks*.**
  - **Each domain has a list *keys*.**
  - **A process executing in a domain can access the object if the domain has a *key* to one of the object *locks*.**

Revocation of Access Rights

- In a dynamic system we may need to revoke access rights to objects.
- Several issues are important:
  - **Revocation is immediate or delayed.**
  - **Revocation is general (affects all users) or selective (affects only certain users).**
  - **Revocation is total (affects all access rights) or partial (affects subset of the rights).**
  - **Revocation is temporary or permanent.**
- Easy to implement with access lists.
- Harder to implement with capability lists.

Security in Computer Systems

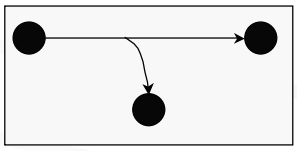
- Protection is an internal operating systems problem - providing controlled access to programs and data stored in the computer.
- Security is mainly focused on malicious behavior of entities external to the system, affecting computer assets:
  - **data.**
  - **Software.**
  - **Hardware.**

Attacks from Intruders

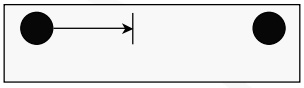
- Real and growing problem.
- Magnified by globalization and the Internet:
  - Today almost every system has to be connected in order to function.
  - Everybody is connected.
- Systems are distributed.
- Hackers (in the bad context of the word) can share information easily.

Security Threats in Computer Systems

- Interception
  - **An unauthorized party gains access to an asset of the system.**



- Interruption / destruction
  - **An asset of the system is destroyed or becomes unavailable or unusable.**



Security Threats in Computer Systems

- Tampering / modification / fabrication
  - **An unauthorized party gains access and tampers with an asset of the system.**

- Resource stealing
  - **An unauthorized party uses system resources.**

Authentication

- User identity most often established through passwords.
- Passwords can be considered as a special. case of either keys or capabilities.
- Passwords must be kept secret.

Techniques for Obtaining Passwords

- Try default password used with standard accounts shipped with computer.
- Exhaustively try all short passwords.
- Try words in dictionary or a list of likely passwords.
- Collect information about users and use these items as passwords: phone numbers, social security numbers, etc.
- Use a Trojan horse to bypass restrictions on access.
- Tap the line between a remote user and the host system.

Password Selection Strategies

- Computer generated passwords
  - **users have difficulty remembering them**
  - **need to write it down**
  - **have history of poor acceptance**
- Eliminate guessable passwords while allowing the user to select a password that is memorable

Password Selection Strategies

- Reactive password checking:
  - **The system periodically runs its own password cracker to find guessable passwords.**
  - **The system cancels passwords that are guessed and notifies user.**
  - **Consumes resources.**
  - **Hackers can use this on their own machine with a copy of the password file.** Can they get the password file?
- Proactive password checking:
  - **The system checks at the time of selection if the password is allowable.**
  - **With guidance from the system, users can select memorable passwords that are difficult to guess.**

One Time Passwords

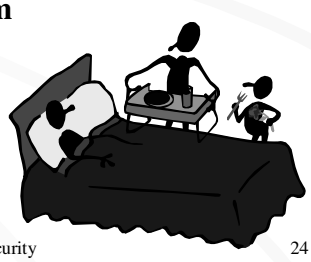
- Challenge and response.
- Challenge can be the time, so only response.

Intrusion Detection

- Assumes that the behavior of an intruder differs from that of a legitimate user.
- Statistical anomaly detection:
  - **Collect data related to the behavior of legitimate users over a period of time.**
  - **Statistical tests are used to determine if the behavior is not legitimate.**
  - **Attempt to define normal, or expected behavior.**
- Rule-based detection:
  - **Rules are developed to detect deviation from previous usage patterns.**
  - **Expert system searches for suspicious behavior.**
  - **Attempt to define proper behavior.**

Viruses and Other Threats

- Those that need a host program.
  - **Fragments of programs that cannot exist independently of some application program, utility, or system program.**
- Independent.
  - **Self-contained programs that can be scheduled and run by the operating system**

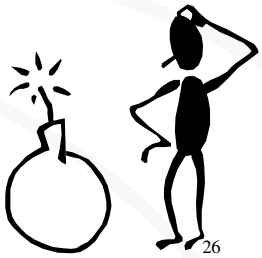


Trapdoor

- Entry point into a program that allows someone who is aware of the trapdoor to gain access.
- Used by programmers to debug and test programs.
  - **Avoids necessary setup and authentication.**
  - **A method to activate a program in case something is wrong with the authentication procedure.**

Logic Bomb

- Code embedded in a legitimate program that is set to "explode" when certain conditions are met.
  - **Presence or absence of certain files.**
  - **Particular date has arrived.**
  - **Particular user is running the application.**



Trojan Horse

- A program that contains hidden code that when invoked performs some unwanted or harmful function.
- Can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly
  - **The user may set the file permissions so that everyone can access.**
- Example: fake login screen.

Bacteria

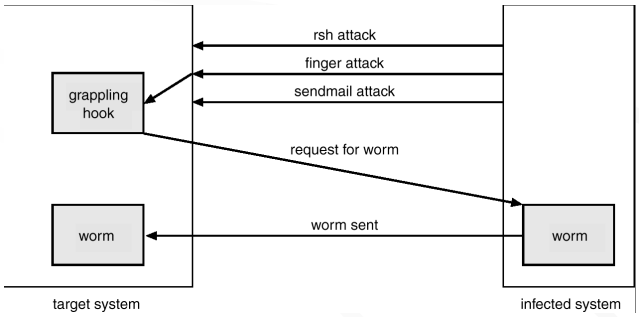
- Their purpose is to replicate themselves.
- Reproduce exponentially.
  - **Take up all the processor capacity.**
  - **Take up memory.**
  - **Take up disk space.**
  - **Eventually deny users access to resources.**

Worm

- Uses network connections to spread from system to system.
- Electronic mail facility.
  - A worm mails a copy of itself to other systems.
- Remote execution capability.
  - A worm executes a copy of itself on other systems.
- Remote log-in capability.
  - A worm logs on to a remote system as a user and then uses commands to copy itself from one system to the other.

The Internet Worm (1988)

- Internet worm
  - Exploited UNIX remote access and bugs in *finger* and *sendmail* programs.
  - Grappling hook program uploaded main worm program.



Virus

- A Program that can “infect” other programs by modifying them
  - **The modification includes a copy of the virus program.**
  - **The infected program can infect other programs.**

Virus Stages

- Dormant phase.
  - The virus is idle.
- Propagation phase.
  - The virus places an identical copy of itself into other programs or into certain system areas on the disk.
- Triggering phase.
  - The Virus is activated to perform the function for which it was intended.
  - Caused by a variety of system events.
- Execution phase.
  - The virus function is performed.

Types of Viruses

- Parasitic.
  - Attaches itself to executable files and replicates.
  - When the infected program is executed, it looks for other executables to infect.
- Memory-resident.
  - Resides in main memory as part of a resident system program.
  - Once in memory, it infects every program that executes.
- Stealth
  - Designed to hide itself from detection by antivirus software.

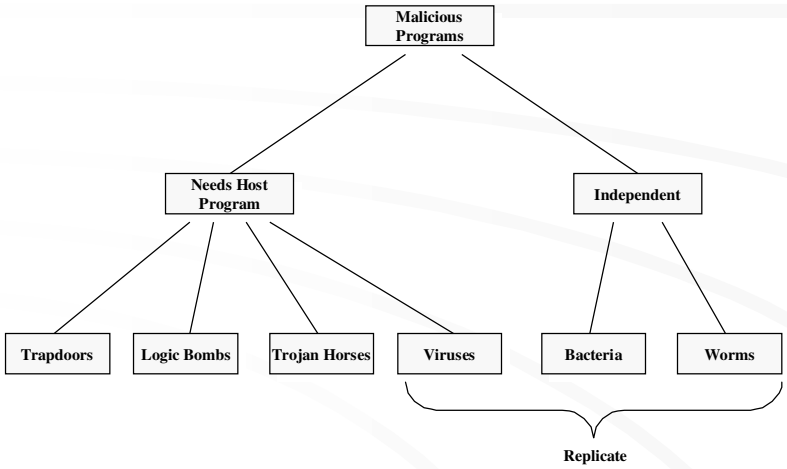
Types of Viruses

- Boot sector.
  - Infects the boot record.
  - Spreads when system is booted from a disk containing the virus.
- Polymorphic
  - Mutates with every infection, making detection by the "signature" of the virus harder.
  - Creates copies of itself that are functionally equivalent but have distinctly different bit patterns.

Antivirus Approaches

- First-generation
  - A scanner identifies a virus by its bit signature.
  - Assumes that a virus has the same structure and bit pattern in all copies.
  - Maintains a record of the length of the programs and looks for changes in length.
- Second-generation
  - Uses heuristic rules to search for probable virus infection.
  - Looks for fragments of code that are often associated with viruses.

Antivirus Approaches

- Third-generation
  - Memory-resident programs that identify a virus by its actions rather than its structure.
  - Intervene when such actions take place.
- Fourth-generation
  - Consists of a variety of antivirus techniques used in conjunction.

Taxonomy of Malicious Programs