OS Security Mechanisms

- Memory protection
- Access control
- User authentication
- Process security
- Boot sequence
- Miscellaneous issues

Key Issues in OS security

- Systems are layered
  - Hardware, processor, OS kernel, processes
  - Each layer needs to provide the security mechanisms needed by the higher layer
    - Each higher layer may attempt to bypass the mechanisms provided by ALL the layers below (not just the one immediately below)
- Physical security to protect hardware
- Hardware protects kernel
- OS kernel provides access control mechanisms, and ensures processes can't access each other’s memory
  - Check every use Vs check first access
  - Checking acceptable use
- Privilege escalation through trusted programs
  - “gates” or “interrupt handlers”
  - setuid programs

Memory protection

- Basis for all security
  - How? Think of a simple file access
- How to protect memory of one process from another?
  - Use hardware mechanisms
  - Segmentation
  - Paging

Memory protection

- System has at least 2 modes of operation, supervisor state and user state.
- Some instructions can only be run in supervisor state.
  - Operations to change virtual memory table can be executed in the supervisor state
- Use special instructions to transfer control to the privileged mode
  - Software interrupt
UNIX Processes and Security

- Processes have
  - Userid
    - Effective userid (euid)
    - Real userid
    - Saved userid
  - Groupid
    - Primary groupid Vs other groupids
    - Group passwords
    - Effective, real and saved gids
  - Changing userid/gid

Interprocess Communication

- Parent-to-child communication
- Signals and process groups
- Debugging and tracing
- Network connection
  - Bind
  - Connect
  - Accept
- Socket connections

Boot Sequence

- Loader loads the kernel
  - Init process (pid 0) starts execution
- Kernel modules get loaded
  - Some immediately, others when boot scripts run
- Boot scripts typically stored in /etc/init.d
  - Different run levels: 1, 2, 3 etc
  - Scripts to be run in a runlevel N are stored in the directory /etc/rcN.d/
    - Symlinks to scripts in /etc/init.d
    - Links names Sxx (for startup) or Kxx (for termination), run in ascending order of the number xx.
  - Usually firewalls, network servers, and so on are all started up from here

Other UNIX Security Issues

- Devices
  - Hard disk
  - /dev/mem, /dev/kmem
  - /dev/tty
- Mounting file systems
  - NFS files
- Search path
  - For commands
  - Libraries
- Network access
  - Firewalls
  - Network address translation (NAT)
  - Server configuration
    - Hosts.allow and hosts.deny
  - /etc/exports, ...
- Capabilities
Database Security

- **Main issue:** finer granularity protection
  - Protecting certain rows, columns, or a complex combination
  - Don’t show column X when column Y has a certain value
- **SQL security**
  - Based on users, actions and objects
  - Objects associated with owners, privileges with users
  - Privileges are of the form (grantor, grante, object, action, grantable)
- **Complex access control can be supported using views**
  - Rather than exposing a table, define a query that extracts only the information that can be exposed
  - Example: a table that represents a user’s schedule
    - May contain sensitive info (who is being met)
    - Define a view to expose only availability (not the other details) of private appointments
  - Issues: updating a view may be difficult

Statistical Security and the Inference Problem

- **Individual records in a DB may be sensitive, but we still want to permit aggregate queries**
  - Number of native americans in a state
- **Inference Problem:**
  - Ability to infer sensitive info from aggregate query results
- **Attacks**
  - Query over small aggregates
  - Evasion measures to defeat obvious defenses
- **Inference problem is very hard to solve fully**
  - Limit sizes of query sets
  - Anonymization and randomization techniques
  - Limit queries (compute worst case leak from a series of queries)