Stateful Firewall

• Keeps track of the state of each connection

• Similar to a proxy firewall
  • No proxies involved, but this can examine contents of connections
  • Analyzes each packet, keeps track of state
  • When state indicates an attack, connection blocked or some other appropriate action taken
Network Organization: DMZ

• DMZ is portion of network separating a purely internal network from external network
• Usually put systems that need to connect to the Internet here
• Firewall separates DMZ from purely internal network
• Firewall controls what information is allowed to flow through it
  • Control is bidirectional; it control flow in both directions
One Setup of DMZ

One dual-homed firewall that routes messages to internal network or DMZ as appropriate.
Another Setup of DMZ

Two firewalls, one (outer firewall) connected to the Internet, the other (inner firewall) connected to internal network, and the DMZ is between the firewalls.
Key Points

• Both amount of information, direction of flow important
  • Flows can be explicit or implicit

• Analysis assumes lattice model
  • Non-lattices can be embedded in lattices

• Compiler-based checks flows at compile time

• Execution-based checks flows at run time

• Analysis can be for confidentiality, integrity, or both
Principles of Secure Design

• These are from Saltzer, Schroeder, Kaashoek
• There are others
  • Cybersecurity Curricular Guidance
  • NICE framework
  • and more!
• All boil down to the same basic ideas
Basis of Design Principles

• Simplicity
  • Less to go wrong
  • Fewer possible inconsistencies
  • Easy to understand

• Restriction
  • Minimize access
  • Inhibit communication
Least Privilege

• A subject should be given only those privileges necessary to complete its task
  • Function, not identity, controls
  • Rights added as needed, discarded after use
  • Minimal protection domain
Related: Least Authority

• Principle of Least Authority (POLA)
  • Often considered the same as Principle of Least Privilege
  • Some make distinction:
    • Permissions control what subject can do to an object directly
    • Authority controls what influence a subject has over an object (directly or indirectly, through other subjects)
Fail-Safe Defaults

• Default action is to deny access
• If action fails, system as secure as when action began
Economy of Mechanism

• Keep it as simple as possible
  • KISS Principle
• Simpler means less can go wrong
  • And when errors occur, they are easier to understand and fix
• Interfaces and interactions
Complete Mediation

• Check every access
• Usually done once, on first action
  • UNIX: access checked on open, not checked thereafter
• If permissions change after, may get unauthorized access
Open Design

• Security should not depend on secrecy of design or implementation
  • Popularly misunderstood to mean that source code should be public
  • “Security through obscurity”
  • Does not apply to information such as passwords or cryptographic keys
Separation of Privilege

• Require multiple conditions to grant privilege
  • Separation of duty
  • Defense in depth
Least Common Mechanism

- Mechanisms should not be shared
  - Information can flow along shared channels
  - Covert channels

- Isolation
  - Virtual machines
  - Sandboxes
Least Astonishment

• Security mechanisms should be designed so users understand why the mechanism works the way it does, and using mechanism is simple
  • Hide complexity introduced by security mechanisms
  • Ease of installation, configuration, use
  • Human factors critical here
Related: Psychological Acceptability

- Security mechanisms should not add to difficulty of accessing resource
  - Idealistic, as most mechanisms add *some* difficulty
    - Even if only remembering a password
  - Principle of Least Astonishment accepts this
    - Asks whether the difficulty is unexpected or too much for relevant population of users
The Confinement Problem

- Isolating entities
  - Virtual machines
  - Sandboxes

- Covert channels
  - Detecting them
  - Analyzing them
  - Mitigating them
Example Problem

• Server balances bank accounts for clients
• Server security issues:
  • Record correctly who used it
  • Send *only* balancing info to client
• Client security issues:
  • Log use correctly
  • Do not save or retransmit data client sends
Generalization

• Client sends request, data to server
• Server performs some function on data
• Server returns result to client
• Access controls:
  • Server must ensure the resources it accesses on behalf of client include only resources client is authorized to access
  • Server must ensure it does not reveal client’s data to any entity not authorized to see the client’s data
Confinement Problem

• Problem of preventing a server from leaking information that the user of the service considers confidential
Total Isolation

• Process cannot communicate with any other process
• Process cannot be observed

Impossible for this process to leak information
  • Not practical as process uses observable resources such as CPU, secondary storage, networks, etc.
Example

• Processes $p$, $q$ not allowed to communicate
  • But they share a file system

• Communications protocol:
  • $p$ sends a bit by creating a file called $0$ or $1$, then a second file called $send$
    • $p$ waits until $send$ is deleted before repeating to send another bit
  • $q$ waits until file $send$ exists, then looks for file $0$ or $1$; whichever exists is the bit
    • $q$ then deletes $0$, $1$, and $send$ and waits until $send$ is recreated before repeating to read another bit
Covert Channel

• A path of communication not designed to be used for communication
• In example, file system is a (storage) covert channel
Rule of Transitive Confinement

• If $p$ is confined to prevent leaking, and it invokes $q$, then $q$ must be similarly confined to prevent leaking

• Rule: if a confined process invokes a second process, the second process must be as confined as the first
Lipner’s Notes

• All processes can obtain rough idea of time
  • Read system clock or wall clock time
  • Determine number of instructions executed

• All processes can manipulate time
  • Wait some interval of wall clock time
  • Execute a set number of instructions, then block
Isolation

• Constrain process execution in such a way it can only interact with other entities in a manner preserving isolation
  • Hardware isolation
  • Virtual machines
  • Library operating systems
  • Sandboxes

• Modify program or process so that its actions will preserve isolation
  • Program rewriting
  • Compiling
  • Loading
Hardware Isolation

• Ensure the hardware is disconnected from any other system
  • This includes networking, including wireless

• Example: SCADA systems
  • 1st generation: serial protocols, not connected to other systems or networks; no security defenses needed, focus being on malfunctions
  • 2nd generation: serial networks connected to computers not connected to Internet
  • 3rd generation: TCP/IP protocol running on networks connected to Internet; need security defenses for attackers coming in over Internet

• Example: electronic voting systems
  • Physical isolation protects systems from attackers changing votes remotely
  • Required in many U.S. states, such as California: never connect them to any network