CIS 610: Advanced Topics in Systems Security

Security Principles

Prof. Kevin Butler

Fall 2011
XSS Problems

- Web application/Media player
  - Failure to identify malicious input
  - Failure to filter malice from input

- Operating system
  - Failure to confine media player (HTTPS backdoor)
  - Failure to limit access to/of root processes (not in TCB?)

- TCB process
  - Failure to filter malicious input

- Failure to prevent malicious function at several stages
Authorization and Authentication

- **Authentication**
  - Def: *Verifying someone or something’s identity*
  - E.g., XSS content

- **Authorization**
  - Def: *Deciding whether a subject can perform a requested operation on an object*
  - Deciding whether the media player can read content

- **Combination**
  - *Authentication is performed for authorization*
Protection System

• Manages the access control policy for a system
  ‣ Security goal

• It represents
  ‣ Protection state
  ‣ Protection state operations

• It describes what operations each subject (via their processes) can perform on each object
The Access Matrix

- An access matrix is one way to represent policy.
  ‣ Frequently used mechanism for describing policy

- Columns are objects, subjects are rows.

- To determine if $S_i$ has right to access object $O_j$, find the appropriate entry.

- Succinct descriptor for $O(|S|*|O|)$ entries

- There is a matrix for each right.
Access Matrix Protection System

• Protection State
  › Current state of matrix

• Can modify the protection state
  › Via protection state operations
  › E.g., can create subjects and objects
  › E.g., owner can add a subject, operation mapping for their objects

• Lampson’s “Protection” paper
  › Can even delegate authority to perform protection state ops
XSS Problems

- Protection system approach is inadequate
- Web application/Media player
  - Can access any web object (no labeling)
  - Like creating a new file in p-state with default rights for that user (modify p-state)
- Operating system
  - Media player runs as user (does anything user can do)
  - Including access to root processes (change passwd?)
  - Root processes are not confined (any can break system)
- What do we need to achieve necessary controls?
Define and Enforce Goals

- Claim: *If we can define and enforce a security policy that ensures security goals, then we can prevent such attacks*

- How do we know the policy expresses effective goals?
  - Will look into this in depth later

- How should such a policy be represented/managed?

- How do we know the enforcement mechanism will enforce policy as expected?
Mandatory Protection System

- Is a protection system that can be modified only by trusted administration that consists of
  - A mandatory protection state where the protection state is defined in terms of a set of labels associated with subjects and objects
  - Label set is defined by trusted administration
  - A labeling state that assigns system subjects and objects to those labels in the mandatory protection state
  - A transition state that determines the legal ways that subjects and objects may be relabeled
- MPS is immutable
Mandatory Protection System

The image shows a diagram of a Mandatory Protection System. The diagram includes labels for Labeling State, Transition State, and Protection State. It also includes a table that outlines permissions for different label types (secret, unclassified, trusted, untrusted) across different processes and files. The table indicates read and write permissions for each combination.
Mandatory Protection State

• Immutable table of
  ‣ Subject labels
  ‣ Object labels
  ‣ Operations authorized for former upon latter

• MPS for OS
  ‣ Allow media player to communicate with browser, exec certain files
  ‣ No network access

• MPS for media player
  ‣ Play only trusted input
Labeling State

• Immutable rules mapping
  ‣ Subjects to labels (in rows)
  ‣ Objects to labels (in columns)

• Labeling State of OS
  ‣ Browser, Media Player have own subject labels
  ‣ Label inputs from network (network connection)
  ‣ Root and TCB program files have labels based on their trust

• Labeling State of Web Application
  ‣ Content – untrusted; Prevent integrity violation
Transition State

- Immutable rules mapping
  - Processes to conditions that change their subject labels
  - IPC to conditions that change their object labels

- Transition State of OS
  - Change label of processes that receive untrusted input
  - Change label of outputs of these processes

- Transition State of Programs
  - Server, Browser, Media Player change labels of their internal objects (threads and variables)
  - Server, Browser, Media Player may be trusted to change their labels (down only?)
Managing MPS

• Challenge
  ‣ Determining how to set and manage an MPS in a complex system involving several parties

• Parties
  ‣ What does programmer know about deploying their program securely?
  ‣ What does an OS distributor know about running a program in the context of their system?
  ‣ What does an administrator know about programs and OS?
  ‣ Users?
Reference Monitor

- **Purpose:** Ensure enforcement of security goals
  - Mandatory protection state defines goals
  - Reference monitor ensures enforcement

- *Every component that you depend upon to enforce your security goals must be a reference monitor*
Reference Monitor

- **Components**
  - Reference monitor interface (e.g., LSM)
  - Authorization module (e.g., SELinux)
  - Policy store (e.g., policy binary)
Reference Monitor Guarantees

- **Complete Mediation**
  - The reference validation mechanism must always be invoked

- **Tamperproof**
  - The reference validation mechanism must be tamperproof

- **Verifiable**
  - The reference validation mechanism must be subject to analysis and tests, the completeness of which must be assured
Complete Mediation

• Every security-sensitive operation must be mediated
  ‣ What’s a “security-sensitive operation”?
  ‣ Operation that enables a subject of one label to access an object that may be a different label

• How do we validate complete mediation?
  ‣ Every such operation must be identified
  ‣ Then we can check for dominance of mediation

• **Mediation**: Does interface mediate correctly?

• **Mediation**: On all resources?

• **Mediation**: Verifiably?
Tamperproof

• Prevent modification by untrusted entities
  ‣ Interface, mechanism, policy of reference monitor
  ‣ Code and policy that can affect reference monitor mods

• How to detect tamperproofing?
  ‣ Transitive closure of operations
  ‣ Challenge: Often some untrusted operations are present

• **Tamperproof**: Is reference monitor protected?

• **Tamperproof**: Is system TCB protected?
Verification

• Test and analyze reference validation mechanism
  ‣ And tamperproof dependencies
  ‣ And what security goals the system enforces

• Determine correctness of code and policy
  ‣ What defines correct code?
  ‣ What defines a correct policy?

• **Verifiable**: Is TCB code base correct?

• **Verifiable**: Does the protection system enforce the system’s security goals?
Evaluation

- **Mediation**: Does interface mediate correctly?
- **Mediation**: On all resources?
- **Mediation**: Verifably?
- **Tamperproof**: Is reference monitor protected?
- **Tamperproof**: Is system TCB protected?
- **Verifiable**: Is TCB code base correct?
- **Verifiable**: Does the protection system enforce the system’s security goals?
Multiple Reference Monitors

• The reference monitor concept approach was designed with a centralized reference validation mechanism in mind
  ‣ What about the case where there are several such mechanisms grouped together?
Take Away

• Mandatory Protection System
  ‣ Means to define security goals that applications cannot impact

• Reference Monitor Concept
  ‣ Requirements for a reference validation mechanism that can correctly enforce an MPS
  ‣ NOTE: This will be a major focus of this course

• Until we come up with coherent approach to defining MPS and validating reference monitor guarantees, we will continue to have insecure systems
  ‣ That is the challenge of systems security research