CSE 127: Computer Security

Security Concepts (cont)

Deian Stefan

Slides adopted from Kirill Levchenko and Stefan Savage
Incentives and Deterrents

• Attacker’s equation:
  (expected gain) > (cost of attack)

• Defender’s equation:
  (cost of protection) < (expected loss)
Incentives and Deterrents

• Attacker’s equation:
  
  \[(\text{expected gain}) > (\text{cost of attack}) + (\text{expected punishment})\]

• Defender’s equation:
  
  \[(\text{cost of protection}) < (\text{expected loss})\]
Security Model

• **Subjects**: Individuals or processes acting on their behalf

• **Objects**: Protected information or function
  ➤ Objects often also include subjects

• **Subjects operate on objects**
  ➤ System mediates and facilitates subject-object interaction
Security Policy

• What action is subject allowed to do with object
• Is this enough?
Security Policy

• What action is subject allowed to do with object

• Is this enough?
  ➢ And who can introduce new subjects and objects into system?
# Access Control Matrix

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{allowed actions}</td>
</tr>
</tbody>
</table>
## Access Control Matrix

<table>
<thead>
<tr>
<th></th>
<th>Broccoli</th>
<th>Fruit from Tree of Life</th>
<th>Fruit from Tree of Knowledge</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam</td>
<td>{see, eat}</td>
<td>{see, eat}</td>
<td>{see}</td>
</tr>
<tr>
<td>Eve</td>
<td>{see, eat}</td>
<td>{see, eat}</td>
<td>{see}</td>
</tr>
</tbody>
</table>
Access Control Lists (ACLs)

• What are ACLs?
  ➤

• How are ACLs enforced?
  ➤

• Real world examples?
Access Control Lists (ACLs)

• An access control list of an object identifies which subjects can access the object and what they are allowed to do

• ACLs are object-centric: access control is associated with objects in the system

• Each access to object is checked against object’s ACL

• Example: guest list at a night club
Capabilities

• What is a capability?
  ➤

• How are capabilities enforced?
  ➤

• Real world example of capabilities?
Capabilities

• A capability grants a subject permission to perform a certain action
  ➤ Unforgeable
  ➤ Usually transferrable

• Capabilities are subject-centric: access control is associated with subjects in the system

• Example: car key
Unix File System Security Model

• Subjects:

• Objects:

• Actions:
Unix File System Sec. Model

• Subjects: Users

• Objects: Files and directories

• Actions: read, write, execute
  ➤ Execute a file means can call `exec()` on file
  ➤ Directory “execute” means user can traverse it

• Unix is a simplified ACL system
  ➤ Arbitrary ACLs not possible in traditional Unix
  ➤ Modern Unix operating systems allow arbitrary
Permissions

• Each file has an owner and a group
  ➢ **Group**: named set of users

• File permissions specify what owner, group, and other (neither owner nor group) is allowed (read, write, exec)
Permissions

• Each file has an owner and a group
  ➤ **Group**: named set of users

• File permissions specify what owner, group, and other (neither owner nor group) is allowed (read, write, exec)
Permissions

• User’s allowed actions on file are:
  ➢ Owner’s permissions if the user is the owner,
  ➢ Group’s permissions if the user is in the group,
  ➢ Other’s permissions otherwise
Permissions

• Users interact with system via processes acting on their behalf
• When you interact with system via terminal, command shell acts on your behalf
• Each process is associated with a user
Permissions

• Who can change permissions?
  ➢ Only owner and superuser can change permissions

• Who can change owner?
  ➢ Only superuser can change owner

• Who can change group?
  ➢ Owner can only change to group she belongs to
Permissions

• Can you change group to arbitrary group?
  ➤ A: yes, B: no
Permissions

• Only owner and superuser can change permissions

• Only superuser can change owner

• Only owner and superuser can change group
  ➤ Owner can only change to group she belongs to

• User’s allowed actions on file are:
  ➤ Owner’s permissions if the user is the owner,
  ➤ Group’s permissions if the user is in the group,
Login

• When user connects to system via physical terminal, system runs login process as root to start session
  ➤ Authenticates user using username and password
  ➤ Changes its user id and group id to that of user
  ➤Executes user's shell

• sshd performs similar actions
Login

- When user connects to system via physical terminal, system runs login process as `root` to start session
  - Authenticates user using username and password
  - Changes its user id and group id to that of user
  - Executes user’s shell
- `sshd` performs similar actions
Superuser can drop privilege to become regular user
Changing Privilege

• Superuser can drop privilege to become regular user
• Want way to elevate privilege in controlled manner
Changing Privilege

• Superuser can drop privilege to become regular user

• Want way to elevate privilege in controlled manner

• How?
Elevating Privilege

• Executable files have a setuid and setgid bit

• If setuid is set, files is executed with privilege of owner
  ➤ ruid is that of executing user, euid and suid that of owner

• The setgid bit does same for group
  ➤ But supplementary groups remain that of executing user
Unix Security Model

• What do you like about the Unix security model?
• What do you dislike about it?
• Is it a good model?