

Lecture 5, April 8, 2026

State Transitions

- Change the protection state of system
- \vdash represents transition
 - $X_i \vdash_{\tau} X_{i+1}$: command τ moves system from state X_i to X_{i+1}
 - $X_i \vdash^* Y$: a sequence of commands moves system from state X_i to Y
- Commands often called *transformation procedures*

Primitive Operations

- **create subject s ; create object o**
 - Creates new row, column in ACM; creates new column in ACM
- **destroy subject s ; destroy object o**
 - Deletes row, column from ACM; deletes column from ACM
- **enter r into $A[s, o]$**
 - Adds r rights for subject s over object o
- **delete r from $A[s, o]$**
 - Removes r rights from subject s over object o

Creating File

- Process p creates file f with r and w permission

```
command create•file( $p$ ,  $f$ )  
    create object  $f$ ;  
    enter own into  $A[p, f]$ ;  
    enter  $r$  into  $A[p, f]$ ;  
    enter  $w$  into  $A[p, f]$ ;  
end
```

Mono-Operational Commands

- Make process p the owner of file g

command *make* • *owner*(p , g)

enter *own* **into** $A[p, g]$;

end

- Mono-operational command
 - Single primitive operation in this command

Conditional Commands

- Let p give q r rights over f , if p owns f
command $grant \cdot read \cdot file \cdot 1(p, f, q)$
 if own **in** $A[p, f]$
 then
 enter r **into** $A[q, f];$
 end
- Mono-conditional command
 - Single condition in this command

Multiple Conditions

- Let p give q r and w rights over f , if p owns f and p has c rights over q

```
command grant.read.file.2( $p, f, q$ )  
  if own in  $A[p, f]$  and  $c$  in  $A[p, q]$   
  then  
    enter  $r$  into  $A[q, f]$ ;  
    enter  $w$  into  $A[q, f]$ ;  
end
```

Copy Flag and Right

- Allows possessor to give rights to another
- Often attached to a right (called a *flag*), so only applies to that right
 - r is read right that cannot be copied
 - rc is read right that can be copied
- Is copy flag copied when giving r rights?
 - Depends on model, instantiation of model

Own Right

- Usually allows possessor to change entries in ACM column
 - So owner of object can add, delete rights for others
 - May depend on what system allows
 - Can't give rights to specific (set of) users
 - Can't pass copy flag to specific (set of) users

Attenuation of Privilege

- Principle says you can't increase your rights, or give rights you do not possess
 - Restricts addition of rights within a system
 - Usually *ignored* for owner
 - Why? Owner gives herself rights, gives them to others, deletes her rights.

What Is “Secure”?

- Adding a generic right r where there was not one is “leaking”
 - In what follows, a right leaks if it was not present *initially*
 - Alternately: not present *in the previous state* (not discussed here)
- If a system S , beginning in initial state s_0 , cannot leak right r , it is *safe with respect to the right r*
 - Otherwise it is called *unsafe with respect to the right r*

Safety Question and Basic Results

- Is there an algorithm for determining whether a protection system S with initial state s_0 is safe with respect to a generic right r ?
 - Here, “safe” = “secure” for an abstract model
- Mono-operational systems: yes, there is such an algorithm
- General systems: no, there is no such algorithm
 - Proof: reduce the halting problem to the safety question
 - Proved by Harrison, Ruzzo, and Ullman; often called the HRU result
 - Says *nothing* about particular classes of systems; this is a generic result

Security Policy

- Policy partitions system states into:
 - Authorized (secure)
 - These are states the system can enter
 - Unauthorized (nonsecure)
 - If the system enters any of these states, it's a security violation
- Secure system
 - Starts in authorized state
 - Never enters unauthorized state

Confidentiality

- X set of entities, I information
- I has the *confidentiality* property with respect to X if no $x \in X$ can obtain information from I
- I can be disclosed to others
- Example:
 - X set of students
 - I final exam answer key
 - I is confidential with respect to X if students cannot obtain final exam answer key

Integrity

- X set of entities, I information
- I has the *integrity* property with respect to X if all $x \in X$ trust information in I
- Types of integrity:
 - Trust I , its conveyance and protection (data integrity)
 - I information about origin of something or an identity (origin integrity, authentication)
 - I resource: means resource functions as it should (assurance)

Availability

- X set of entities, I resource
- I has the *availability* property with respect to X if all $x \in X$ can access I
- Types of availability:
 - Traditional: x gets access or not
 - Quality of service: promised a level of access (for example, a specific level of bandwidth); x meets it or not, even though some access is achieved

Policy Models

- Abstract description of a policy or class of policies
- Focus on points of interest in policies
 - Security levels in multilevel security models
 - Separation of duty in Clark-Wilson model
 - Conflict of interest in Brewer-Nash model

Mechanisms

- Entity or procedure that enforces some part of the security policy
 - Access controls (like bits to prevent someone from reading a homework file)
 - Disallowing people from bringing CDs and floppy disks into a computer facility to control what is placed on systems

Types of Security Policies

- Military (governmental) security policy
 - Policy primarily protecting confidentiality
- Commercial security policy
 - Policy primarily protecting integrity
- Confidentiality policy
 - Policy protecting only confidentiality
- Integrity policy
 - Policy protecting only integrity

Integrity and Transactions

- Begin in consistent state
 - “Consistent” defined by specification
- Perform series of actions (*transaction*)
 - Actions cannot be interrupted
 - If actions complete, system in consistent state
 - If actions do not complete, system reverts to a consistent state

Trust

Administrator installs patch

1. Trusts patch came from vendor, not tampered with in transit
2. Trusts vendor tested patch thoroughly
3. Trusts vendor's test environment corresponds to local environment
4. Trusts patch is installed correctly