Multi-Factor Authentication

Multi-Factor Authentication (1 of 2)

Oregon State University
College of Engineering

- Use multiple elements/factors to prove identity

- Factors of authentication
  - What entity knows (*e.g.* password, private key)
  - What entity has (*e.g.* badge, smart card)
  - What entity is (*e.g.* fingerprints, retinal characteristics)
  - What entity does (e.g., voice pattern, handwriting, typing rhythm)
  - Where entity is (*e.g*. in front of a particular terminal)

- Example: Chip & PIN credit card transaction
  - Need to 'have' Chip and 'know' PIN

Multi-Factor Authentication (2 of 2)

Oregon State University
College of Engineering

- Provides additional protection as compromising one-factor is not sufficient
  - Ex: Guessing a password is not sufficient
  - Ex: Stealing an ATM card is not sufficient
- Factors used must be commensurate with the value of asset
  - Ex: Using SMS as a second factor was proven to be insufficient for protecting multi-millionaire and billionaire accounts. Adversaries bribed telco employees and cloned SIM cards.
- Two-Factor Authentication is becoming the norm
  - Almost everyone has a connected device that can be leveraged

Two Factor Authentication (TFA)

Oregon State University
College of Engineering

- Use two factors, e.g., password + ?
  - Mobile Phone – SMS
  - Software OTPs
  - Email account
  - OTP Hardware dongles

Two Factor Authentication

Oregon State University
College of Engineering

Log In | Help | Security and Protection Search

PayPal

Enter Security Code

Secure Log In

12172441925

Confirm your phone number and 6-digit code.

Your mobile number: 12178988368

Send SMS

Please wait a moment for the SMS to arrive. The code you receive is valid for one minute from when you receive it. Didn't get the code?

6-digit code:

Submit

I don't have my security key with me

About PayPal | Contact Us | Fees | PayPal Developers | Merchant Services | Worldwide | Site Feedback [+]
Privacy | PayPal Blog | PayPal Labs | Jobs | Legal Agreements | Site Map | eBay

Two Factor Authentication

Oregon State University
College of Engineering

Google

2-step verification

Enter the verification code generated by your mobile application.

Enter code:

Verify

Don't ask for codes again on this computer

Don't have your phone?
Cancel

Two Factor Authentication?

Oregon State University
College of Engineering

Bank of America Sign In

Enter your Passcode

If your SiteKey is correct, enter your Passcode to sign in. If this isn't your SiteKey, do not enter your Passcode.

SiteKey lets you know you're at a Bank of America site and not a fraudulent one.

Your SiteKey

Holy Grail

Passcode

Sign in

Summary

Oregon State University
College of Engineering

- Multi-factor authentication provides better protection than traditional single-factor authentication
- Two-factor authentication (TFA) is becoming the norm
- SMS, Software OTP are commonly used 2nd factors

One-time Passwords (OTP)

Dynamic Password Generator

Oregon State University
College of Engineering

- System periodically creates and displays new password
  - E.g., RSA SecurID, Google Authenticator App,

RSA SecurID

- User enters current password


Google Authenticator


FreeOTP

One-Time Passwords

Oregon State University
College of Engineering

- Password that can be used exactly *once*
  - After use, it is immediately invalidated
- Challenge-response mechanism
  - Challenge is one of a number of authentications; response is password for that particular number
- Problems
  - Synchronization of user (prover), system/server (verifier)
  - Generation of good random passwords
  - Password distribution problem

S/Key

Oregon State University
College of Engineering

- One-time password scheme based on idea of Lamport

- *h* one-way hash function (SHA-256, for example)

- User chooses initial seed *k*
- Server calculates:

$$h(k) = k_1,\ h(k_1) = k_2,\ \ldots,\ h(k_n) = k_{n+1}$$

  - Needs only save k_{n+1}

- Passwords are reverse order:

$$p_1 = k_n,\ p_2 = k_{n-1},\ \ldots,\ p_{n-1} = k_2,\ p_n = k_1$$

S/Key

Central Ideas:

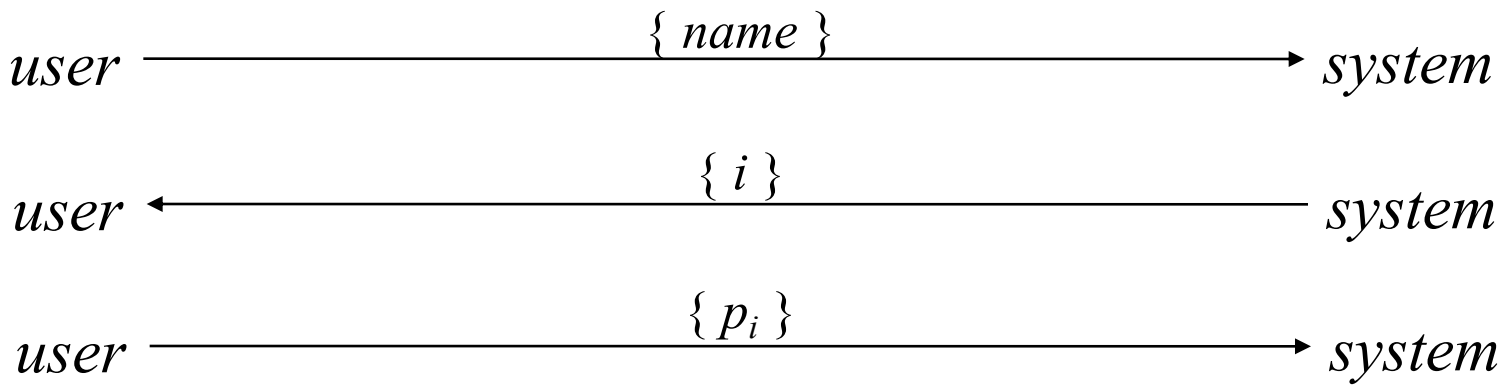
- Given last pwd p, observer cannot predict p’ s.t. h(p’) = p,
  - i.e., cannot predict next password.

- Server remembers last pwd p, and when p’ is offered, validates h(p’) = p

S/Key Protocol

Oregon State University
College of Engineering

System stores maximum number of authentications n, number of next authentication i, last correctly supplied password p_{i-1}.



System computes $h(p_i) = h(k_{n-i+1}) = k_{n-i+2} = p_{i-1}$. If match with what is stored, system replaces p_{i-1} with p_i and increments i.

HOTP

Oregon State University
College of Engineering

- HMAC-based One-Time Password (HOTP) algorithm
  - IETF RFC 4226, December 2005
- Server and user pre-establish a shared secret K, and a beginning counter value C
- HOPT One-time password value is computed as follows
  - HOTP(K,C) = Truncate(HMAC-SHA-1(K,C))
  - HOTP Password = HOTP(K,C) mod 10^d where d is password length (typically 6 - 8 digits)
  - Truncate() extracts 31-bits but NOT the last 31 bits. 31 bits starting at index *i+1* where *i* is last four digits of the MAC value.
- Counter is updated after every successful login

TOTP

Oregon State University
College of Engineering

- Time-based One-Time Password (TOTP) algorithm
  - Extension of HOTP
  - IETF RFC 6238, May 2011

- Server and user pre-establish a shared secret K

- Counter value is obtained using current Unix time
  - C_T = floor([(T – T_0)/Time-step])
  - T is current Unix Time, T_0 defaults to 0. Time-step defaults to 30 seconds

- To account for network delays and synchronization issues, validator will also check with C_T+1 and C_T-1

Summary

Oregon State University
College of Engineering

- Dynamic or One-Time Passwords provide better security against password loss etc.
- S/KEY's use is reducing as a main form of authentication
- OTPs (e.g., HOTP, TOTP) are gaining use as a 2nd factor of authentication

Biometric-based Authentication

Biometric

Oregon State University
College of Engineering

- Measurement of biological, behavioural feature that identifies a person
  - Examples: fingerprints, DNA, palm print, hand geometry, iris patterns, retinal patterns, typing rhythm, gait, voice

- Advantages vs. Passwords
  - No need to remember anything, it is intrinsic to the user
  - Typically considered to have higher entropy than passwords

What makes a good biometric?

Oregon State University
College of Engineering

- Universality
  - All users of the system must possess the trait
- Uniqueness
  - Trait should be different for each individual for distinguishability
- Permanence
  - Trait should remain stable over time
- Collectability/Measurability
  - Trait should be easy to measure and process
- Performance
  - Accuracy, speed and robustness of technology for collecting/measuring trait
- Acceptability
  - User willingness to permit collection of biometric
- Circumvention
  - Ease with which the trait can be imitated

Jain, A. K.; Bolle, R.; Pankanti, S., eds. (1999). *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publications

Some biometrics in use (1 of 2)

Oregon State University
College of Engineering

- Fingerprints: optical or electrical techniques
  - Maps fingerprint into a graph, then compares with database
  - Measurements imprecise, so approximate matching algorithms used

- Voices: speaker verification or recognition
  - Verification: uses statistical techniques to test hypothesis that speaker is who is claimed (speaker dependent)
  - Recognition: checks content of answers (speaker independent)

Some biometrics in use (2 of 2)

Oregon State University
College of Engineering

- Eyes: patterns in irises unique
  - Measure patterns, determine if differences are random; or correlate images using statistical tests
- Faces: image, or specific characteristics like distance from nose to chin
  - Lighting, view of face, other noise can hinder this
- Keystroke dynamics: believed to be unique
  - Keystroke intervals, pressure, duration of stroke, where key is struck
  - Statistical tests used

Biometric-based Authentication

Oregon State University
College of Engineering

- Physical characteristics encoded in a template
  - The C or complement information
- User registers physical information (S)
  - Generally with multiple measurements
- The verification function takes a measurement and tries to line up with template

Performance Measures

Oregon State University
College of Engineering

- Enrollment Failure Rate
  - Rate of failure for creating templates during enrollment
  - Mainly due to low quality inputs (e.g., worn out fingers)
- Failure to Capture/Measure
  - Failure or capturing metric when presented correctly
- False Match or False Positive Rate
  - Rate of incorrectly matching an input to a template
- False Non-match or False Negative Rate
  - Rate of incorrectly rejecting an input that matches a template

Authentication vs Identification

Oregon State University
College of Engineering

- Used for surveillance
  - Subject is motivated to avoid detection
- Used for authentication
  - Subject is motivated to positively identify
  - Perhaps pick up other's characteristics
- False positives vs. false negatives

Biometric Cautions (1 of 2)

Oregon State University
College of Engineering

- These can be fooled!
  - Assumes biometric device accurate *in the environment it is being used in!*
  - Transmission of data to validator is tamperproof, correct
- Physical characteristics change over time
- Some people may not be able to identify via specific characteristics
  - Albinos and iris scans
- Require special hardware to capture the biometric for verification

Biometric Cautions (2 of 2)

Oregon State University
College of Engineering

- Where are the biometric templates stored? What if your biometric template data is stolen?
  - Does it compromise your ability to use the biometric in other systems?
  - Can the system recover and create a new template?
- Could be dangerous to the owner?
  - e.g., thieves may now want your finger instead of keys to steal your car

Summary

Oregon State University
College of Engineering

- Biometrics are not a silver bullet for authentication
- There are both advantages and disadvantages to using biometrics vs. passwords
- Privacy of biometric data is a big concern

Access Control: An Introduction

Access Control

Oregon State University
College of Engineering

- Central to Computer Security!

- Applied at a number of levels
  - Application
  - Middleware
  - Operating Systems
  - Hardware
  - Network

Access Control Context

Oregon State University
College of Engineering

Authorization Database

User Access Request

Authentication

Access Control/ Authorization Function

Request Granted

Resources

Auditing

AAA

Oregon State University
College of Engineering

Access control part of a broader context

- Authentication
  - Bind external entity to system entity
- Authorization
  - Grant a right or permission to the system entity to access a system resource
- Audit
  - Independent review of system actions

Some Access Control Policy Models

Oregon State University
College of Engineering

- Discretionary Access Control (DAC)
  - Decision made based on identity of requestor and access rules
  - Regular users can adjust the policy
- Mandatory Access Control (MAC)
  - Decision made by testing labels associated with processes and resources against system policy rules
  - Regular user cannot adjust the policy
- Role-Based Access Control (RBAC)
  - Access policies defined in terms of roles rather than individual requestors
- Attribute-Based Access Control (ABAC)
  - Access policies defined in terms of attributes of the user, resource, and environment or context for the access

Access Control Considerations

Oregon State University
College of Engineering

- Reliable Input
  - Access control assumes user has been authenticated, that other control inputs (e.g. object names, addresses) are correct
- Fine and course grain specifications
  - Management overhead vs. precision in policy/access control
  - Unix controls at the level of files, not accounts, not bytes
- Open/Closed policies
  - "closed" --- access limited to those explicitly stated
    - E.g., default "deny" on firewall rule
  - "open" --- access limitations are specified, all others allowed

Access Control Considerations

Oregon State University
College of Engineering

- Policy combination and conflict resolution
  - Different entities have different policy and we need to adjudicate with respect to both
  - A given policy may have conflicting rules---how to resolve?
    - Unix example 'deny global read' at account level, 'global read' permission at file level….what's the policy decision?

- Administrative policies – How is change to access control managed?

Access Control – Some Principles

Oregon State University
College of Engineering

- Least Privilege
  - Provide the least set of privileges needed for the task
- Separation-of-Duty
  - More than one entity to complete a task
  - Manages conflict of interest
  - Example : accountant and auditor
- Dual Control
  - Two entities required to implement policy change

Summary

Oregon State University
College of Engineering

- Access Control is central to computer security

- Applied at all levels
  - hardware, operating system, middleware, application, networks etc.
- Comprises – Authentication, Authorization and Audit

- Common Access Control Models
  - DAC, MAC, RBAC, ABAC

- Key Principles: Least Privilege, Separation-of-Duty (SoD)

Access Control Matrix: An Abstraction

Access Control

Oregon State University
College of Engineering

- Central to Computer Security!

- Applied at a number of levels
  - Application
  - Middleware
  - Operating Systems
  - Hardware

Access Control Context

Oregon State University
College of Engineering

Authorization Database

User Access Request

Authentication

Access Control/ Authorization Function

Request Granted

Resources

Auditing

Oregon State University
College of Engineering

Representing Authorization or Access Control Policy

Access Control Elements

Oregon State University
College of Engineering

- Subject
  - system entity capable of access objects.
  - Generally a process in an OS context
- Object
  - a resource in a system
  - Often a file; could also be mutex, process, network interface, network port
- Access right
  - way or mode that a subject may access an object in the system
  - Read, Write, Execute, Delete, Create, Search, Change Access, Own

Access Control Matrix (ACM)

- Provides very basic abstraction
- Maps different systems to a common form for comparison
- Enables standard proof techniques
- Not directly used in implementation

Access Control Matrix: File Example

Oregon State University
College of Engineering

	File 1	File 2	File3	File4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Implementing Access Matrix

Oregon State University
College of Engineering

- Slice by column
  - Store identity of entities that are permitted access *with the objects* :Access control list (ACL)
  - Used by Multics and most modern Operating Systems

ACL Example

Oregon State University
College of Engineering

File 1	**User A** Own Read Write	**User B** Read	**User C** Read Write
File 2	**User B** Own Read Write	**User C** Read	
File 3	**User A** Own Read Write	**User B** Write	
File 4	**User B** Read	**User C** Own Read Write	

	File 1	File 2	File3	File4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Slice by Column: Access Control Lists (ACLs)

Ways to make ACL scale up

Oregon State University
College of Engineering

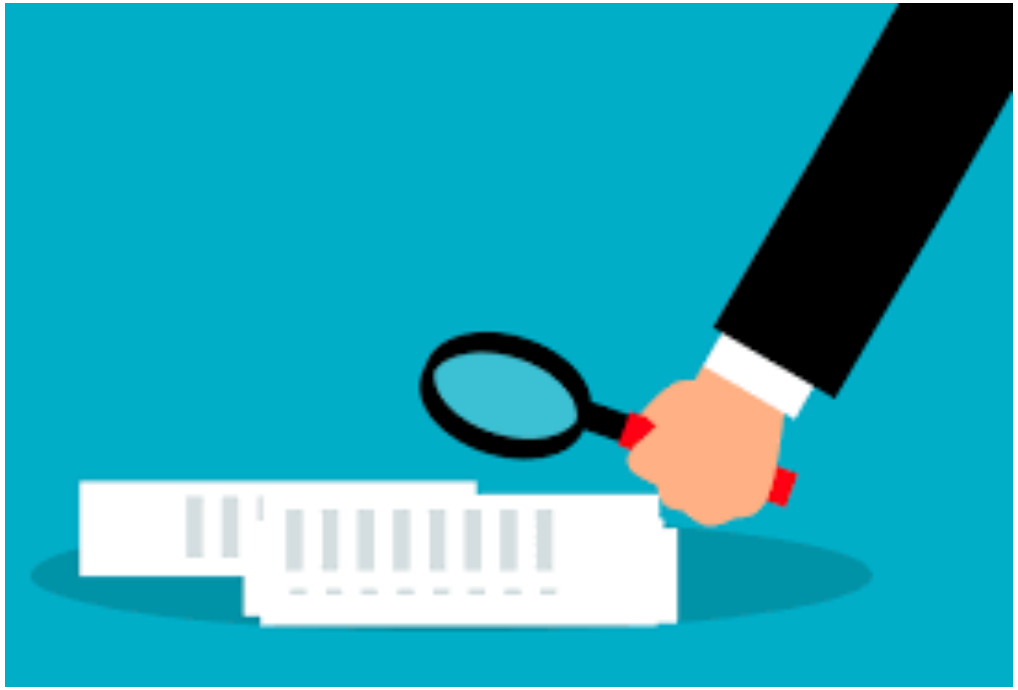
- Groups of users
  - Shortens list of ACEs
- Role-Based Access Control
  - Users can take on role at a time
  - One role may apply to many users
- Directory inheritance
  - Transfer rights to lower objects in directory tree
- Negative rights
  - Infer what is permitted by the negation of what is stated to be denied

Revoking rights with ACLs

- What is revocation?
- Revoking rights for subject *s* to a particular object *o* straightforward
  - Remove *s* from ACL(*o*) OR
  - Make sure *s* has a negative entry in the ACL(*o*)
- Example: Alice removes all of Bob's rights to file f
  - What if Bob had given Carol read rights to f?
  - Should Carol still have those rights?

Access Review with ACLs

Oregon State University
College of Engineering



- Per-object review
  - Who has access to a given object?
  - Easy – look at the ACL
- Per-subject review
  - What object can a subject access?
  - Hard – access and search the ACLs of all object in the system

Implementing Access Matrix

Oregon State University
College of Engineering

- Slice by row
  - Store identity of objects a subject may access *with the subjects*: Capability list, or "capability tickets"
    - Subjects can explicitly share or transfer capabilities with others
  - Many implementations in the '80's
  - Often associated with object-oriented systems

Capabilities Example

Oregon State University
College of Engineering

User A	**File 1** Own Read Write	**File 3** Own Read Write		
User B	**File 1** Read	**File 2** Own Read Write	**File 3** Write	**File 4** Read
User C	**File 1** Read Write	**File 2** Read	**File 4** Own Read Write	

	File 1	File 2	File3	File4
User A	Own Read Write		Own Read Write	
User B	Read	Own Read Write	Write	Read
User C	Read Write	Read		Own Read Write

Slice by Row: Capabilities

Capabilities

Oregon State University
College of Engineering

- Slice Access matrix by row
- Capabilities associated with the subjects
- Challenges?

Capability Integrity

Oregon State University
College of Engineering

- Subject presents "capability" in order to access object
  - Capability data structure encapsulates object ID with allowed rights.
- Unlike ACLs, capabilities are not completely contained by the OS
  - Can be transmitted, e.g., tokens in Kerberos
- Capability integrity is a big concern
  - Tagged memory
  - Segmented memory
  - Cryptographic hashes

Revoking capabilities

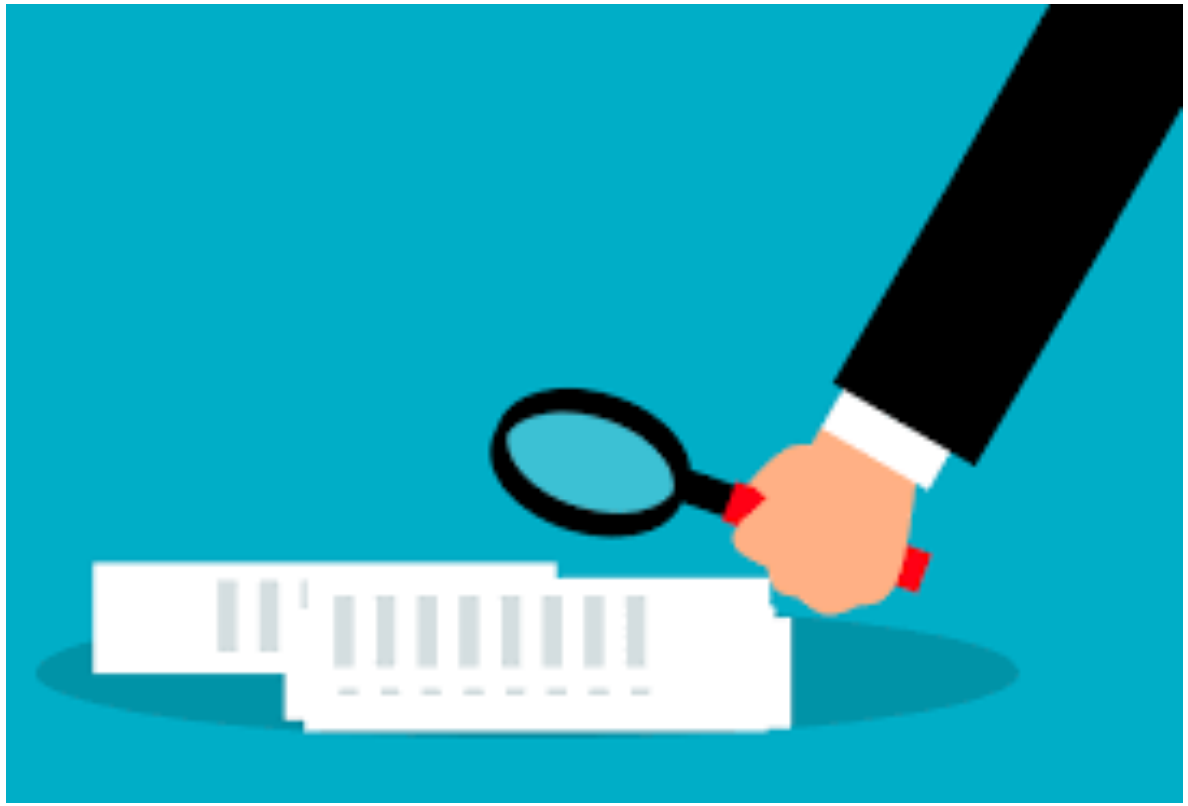
Oregon State University
College of Engineering

- Easy to revoke all rights to a given subject
- What about revoking everyone's rights to a particular object?

Access Review with Capabilities

Oregon State University
College of Engineering

- Per-object review
  - Who has access to a given object?
  - hard – access and search all capabilities



- Per-subject review
  - What object can a subject access?
  - easy – look up subject's capability

Authorization Table

Oregon State University
College of Engineering

- Each row describes one access right to one object by one user
- Sort by subject groups like capability list
- Sort by object groups like access control table
- Easily implemented with relational database
- Not sparse like ACM

Subject	Access Right	Object
A	Own	File 2
A	Read	File 2
A	Write	File 2
B	Read	File 1
B	Read	File 3
C	Own	File 3
C	Read	File 3

Sorted by Subject

Subject	Access Right	Object
B	Read	File 1
A	Own	File 2
A	Read	File 2
A	Write	File 2
B	Read	File 3
C	Own	File 3
C	Read	File 3

Sorted by Object

Summary

01

Access Control Matrix (ACM) is a useful abstraction of access policy/models

02

It is not directly implemented but slices of ACM are implemented as Access Control Lists (ACLs) or Capabilities

03

ACLs are the commonly used mechanism for implementing DAC

04

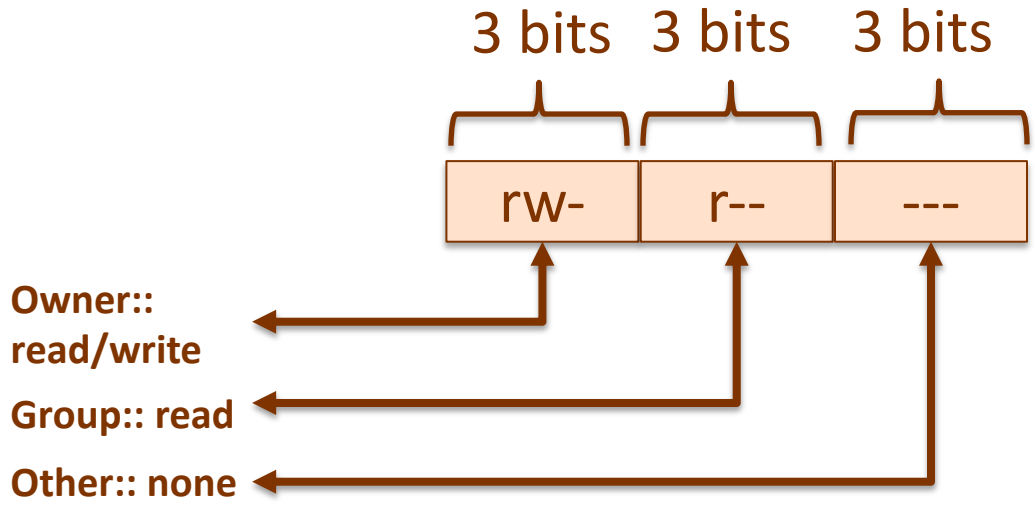
Access review is easier with authorization table compared with ACLs or Capabilities

Discretionary Access Control in Practice

UNIX File Access Control

Oregon State University
College of Engineering

- Unique user identification number (user ID)
- Member of a primary group identified by a group ID
- Three permission octets associated with each file and directory
  - Specify read, write, and execute permission for the owner of the file, members of the group and all other users
- The owner ID, group ID, and protection bits are part of the file's inode



Traditional Unix File Protection

UNIX File Access Control

Oregon State University
College of Engineering

- 3 additional bits
  - "Set user ID"(SetUID)
  - "Set group ID"(SetGID)
  - "Sticky Bit"

- "Set user ID"(SetUID), "Set group ID"(SetGID)
  - System temporarily uses rights of the file owner/group in addition to the real user's rights when making access control decisions
  - Enables privileged programs to access files/resources not generally accessible

- Sticky bit
  - When applied to a directory it specifies that only the owner of any file in the directory can rename, move, or delete that file

UNIX File Access Control

Oregon State University
College of Engineering

- Superuser
  - Is exempt from usual access control restrictions
  - Has system-wide access

ACLs in UNIX

Oregon State University
College of Engineering

Modern UNIX systems support ACLs

- FreeBSD, OpenBSD, Linux, Solaris

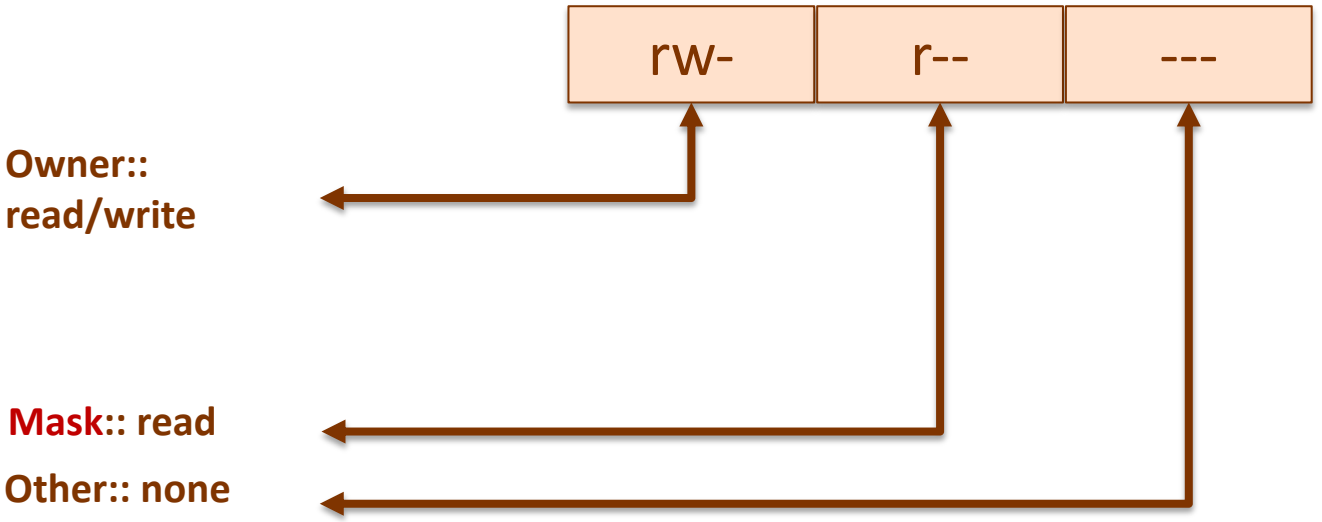
FreeBSD

- Setfacl command assigns a list of UNIX user IDs and groups
- Any number of users and groups can be associated with a file
- Read, write, execute protection bits
- A file does not need to have an ACL
- Includes an additional protection bit that indicates whether the file has an extended ACL

When a process requests access to a file system object two steps are performed:

- Step 1 selects the most appropriate ACL
- Step 2 checks if the matching entry contains sufficient permissions

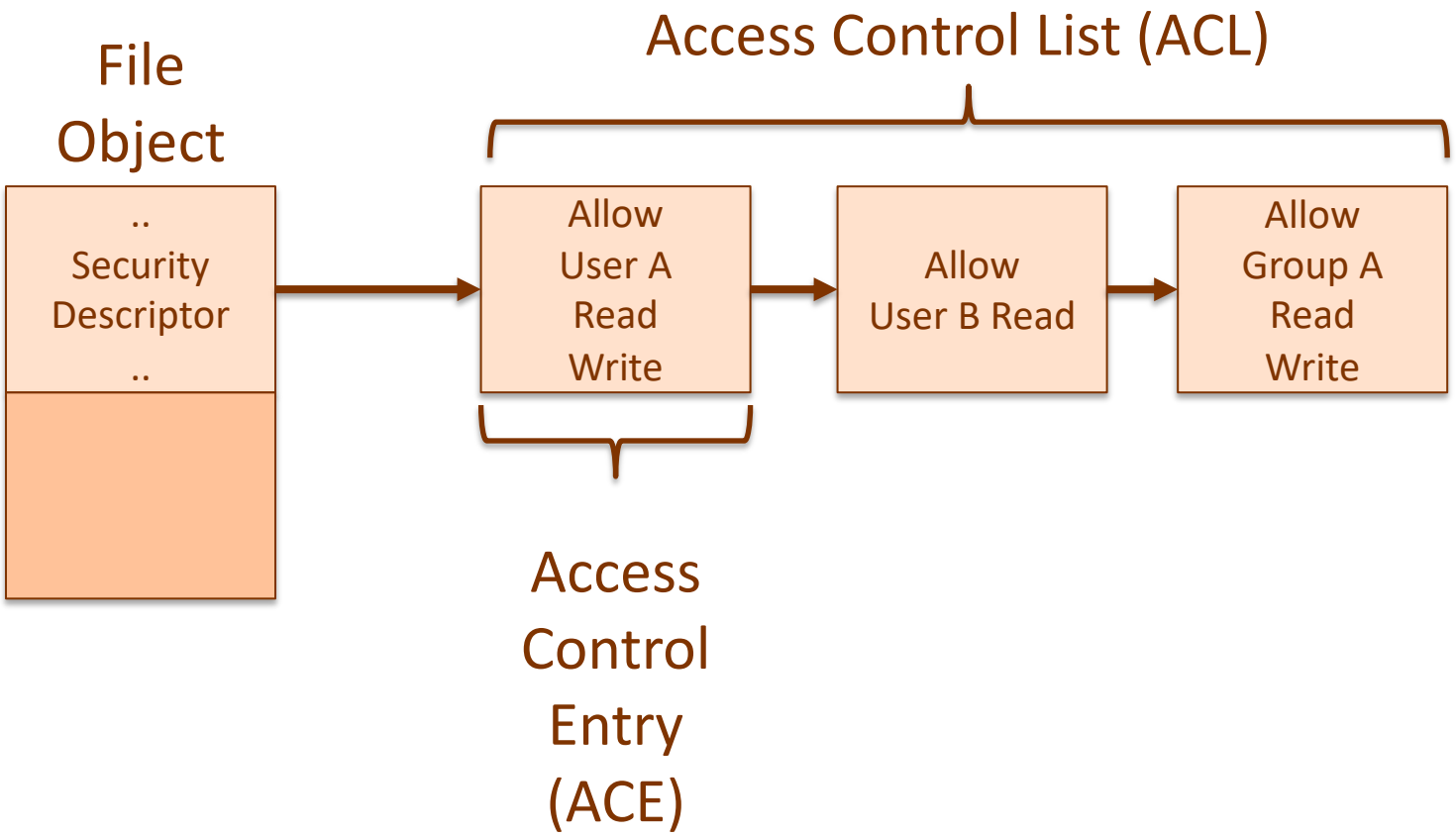
Extended Unix ACL

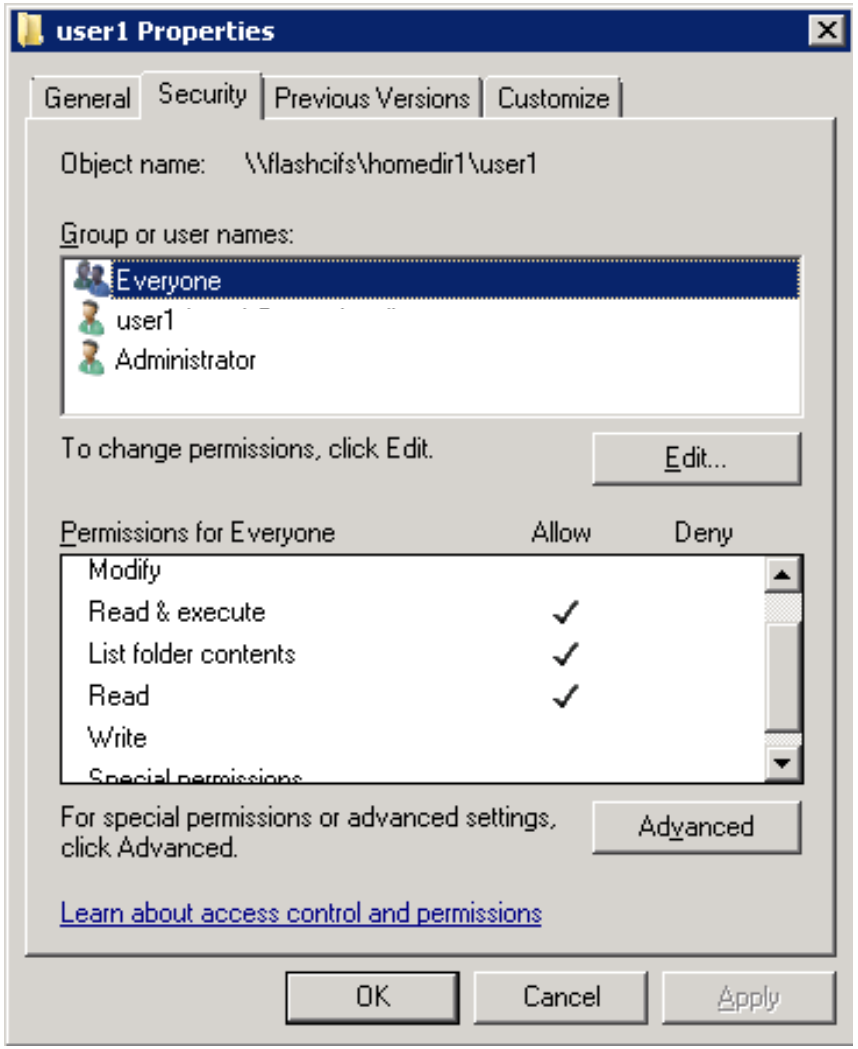


Extended Unix ACL

Windows ACL (1 of 2)

Oregon State University
College of Engineering





Windows ACL (2 of 2)

Oregon State University
College of Engineering

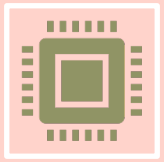
- Actually two ACL's per file
  - System ACL (SACL) – controls auditing and now integrity controls
  - Discretionary ACL (DACL) – controls object access

- Windows ACLs apply to all named objects
  - Files
  - Pipes
  - Events

Summary

Oregon State University
College of Engineering

Both Unix and Windows primarily use DAC, specifically ACLs



Mandatory Access Controls have slowly found their way into mainstream OSes for integrity protection

e.g., SACL in Windows