CSE 465
Information Assurance

Authentication and Access Control

Professor Stephen S. Yau

Authentication

- Authentication is ***validation of a user's identity***
- Four general ways for authentication:
  - What an entity ***knows*** (passwords, secret information)
  - What an entity ***has*** (badge, card)
  - ***Who*** an entity is (fingerprints, retinal characteristics)
  - ***Where*** an entity is (in front of a particular terminal)

Passwords

- A password is information associated with an entity that confirms the entity’s identity
- Password storage
  - Store in file
  - Store in encrypted file
  - Store with one-way hashes

Password Attacks

- Dictionary attack
- Other types of attacks???
- Countermeasures
  - Random selection of passwords
  - Use strong passwords
  - Disable the account after *n* consecutive attempts to login in to an account fail
  - Other types of countermeasures???

One-Time Passwords

- Password that can be used exactly *once*
- Generation mechanisms
  - ***Time-synchronization***
    - Using a synchronized time between client and server
    - Example

      Let t_x be a current synchronized time,

      $f(t_x)=p_x$ The passwords in the order of use are

      $p_1, p_2 \cdots p_x \cdots$

One-Time Passwords (cont.)

- ***Challenge-response***
  - Using a challenge from server
  - Example: Let c_i be the current challenge from server,

    $f(c_i) = p_{i.}$ The passwords in the order of use are

    $p_1, p_2 \ldots, p_i, \ldots p_n$
- ***Hash chain***
  - Using a chain of hash functions
  - Example: h is the hash function, p is the OTP and an initial seed s

    $h(s)=p_1, h(p_1)=p_2, \ldots, h(p_{n-1})=p_n$

    The passwords in the order of use are

    $p_n, p_{n-1}, \ldots, p_2, p_1$

Biometric Authentication

- ***Fingerprints***: scan fingerprints as graphs
- ***Voices***: speaker verification or recognition
- ***Eyes***: irises
- ***Faces:*** image, or specific characteristics like distance from nose to chin
- ***Keystroke dynamics***: keystroke intervals, pressure, duration of stroke, where key is struck
- ***Combinations of the above***

Effectiveness of Biometrics

- Evaluated on three basic criteria
  - ***False reject rate:*** Rate at which supplicants (authentic users) are denied or prevented from accessing authorized areas due to a failure detected by biometric device ***(Type I error).***
  - ***False accept rate:*** Rate at which supplicants who are not legitimate users are allowed access to systems or areas due to failure detected by biometric device ***(Type II error).***
  - ***Crossover error rate (CER):*** Level at which the number of false rejections equals the number of false acceptances, (equal error rate). This is the most common and important overall measure of the accuracy of biometric systems.

Acceptability of Biometrics

- Balance between how acceptable the security system to users and its effectiveness in maintaining the security
  - Many biometric systems are highly reliable and effective, but are also ***invasive***
  - Many information security professionals do not use them in order to ***avoid confrontation*** and ***possible user boycott*** of biometric control

Ranking of Biometric Effectiveness and Acceptance [1]

Biometrics		Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
	Face	H	L	M	H	L	H	L
	Fingerprint	M	H	H	M	H	M	H
	Keystroke Dynamics	L	L	L	M	L	M	M
	Iris	H	H	H	M	H	L	H
	Retina	H	H	M	L	H	L	H
	Signature	L	L	L	H	L	H	L
	Voice	M	L	L	M	L	H	L
	DNA	H	H	H	L	H	L	L

H=High, M=Medium, L=Low

Access Control Matrix

- Access control matrix is simplest framework for describing rights of users over files in a matrix

	File 1	File 2	File 3	File 4
User 1	R, W, O	R	R, W, X, O	W
User 2	R	R, O	R	R, W, X, O

Access Control List

- A variant of the access control matrix
- Store each column with the object it represents

  ACL(file 1) = {(user 1, RWO), (user 2, R)}

  ACL(file 2) = {(user 1, R), (user 2, RO)}

  ACL(file 3) = {(user 1, RWXO), (user 2, R)}

  ACL(file 4) = {(user 1, W), (user 2, RWXO)}

Capabilities

- Another variant of the access control matrix
- Store each row with the subject it represents

CAP(user 1) = {(file 1, RWO), (file 2, R), (file 3, RWXO), (file 4, W)}

CAP(user 2) = {(file 1, R), (file 2, RO), (file 3, R), (file 4, RWXO)}

ACL vs. Capabilities

- Two different questions
  - Given an object, which subjects can access it, and how?
  - Given a subject, which objects can it access, and how?
- ACL is easy to answer the first question
- Capabilities is easy to answer the second question
- Which question is more important?

ACL vs. Capabilities *(cont.)*

- Authentication
  - Given a process that wishes to perform an operation on an object
  - ACL needs to authenticate the process's identity
  - Capabilities do not require authentication, but require ???
- Least Privilege
  - Capabilities provide finer grained least privilege control
- Revocation
  - ACL can remove a group of users from the list, and those users can no longer gain access to the object
  - Do capabilities have equivalent operations?

Access Control Models

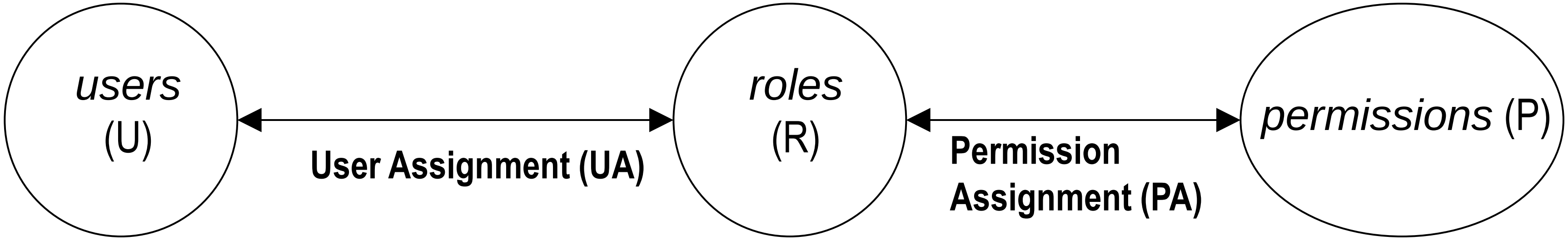
- ***Discretionary Access Control (DAC)***
  - Restricting access to objects ***based on identity of subjects and/or groups*** to which they belong
- ***Mandatory Access Control (MAC)***
  - Restrict access to objects ***based on the sensitivity*** (as represented by a label***) of the information*** contained in the objects and the ***formal authorization*** (i.e. clearance) ***of subjects*** to access information of such sensitivity

Access Control Models *(cont.)*

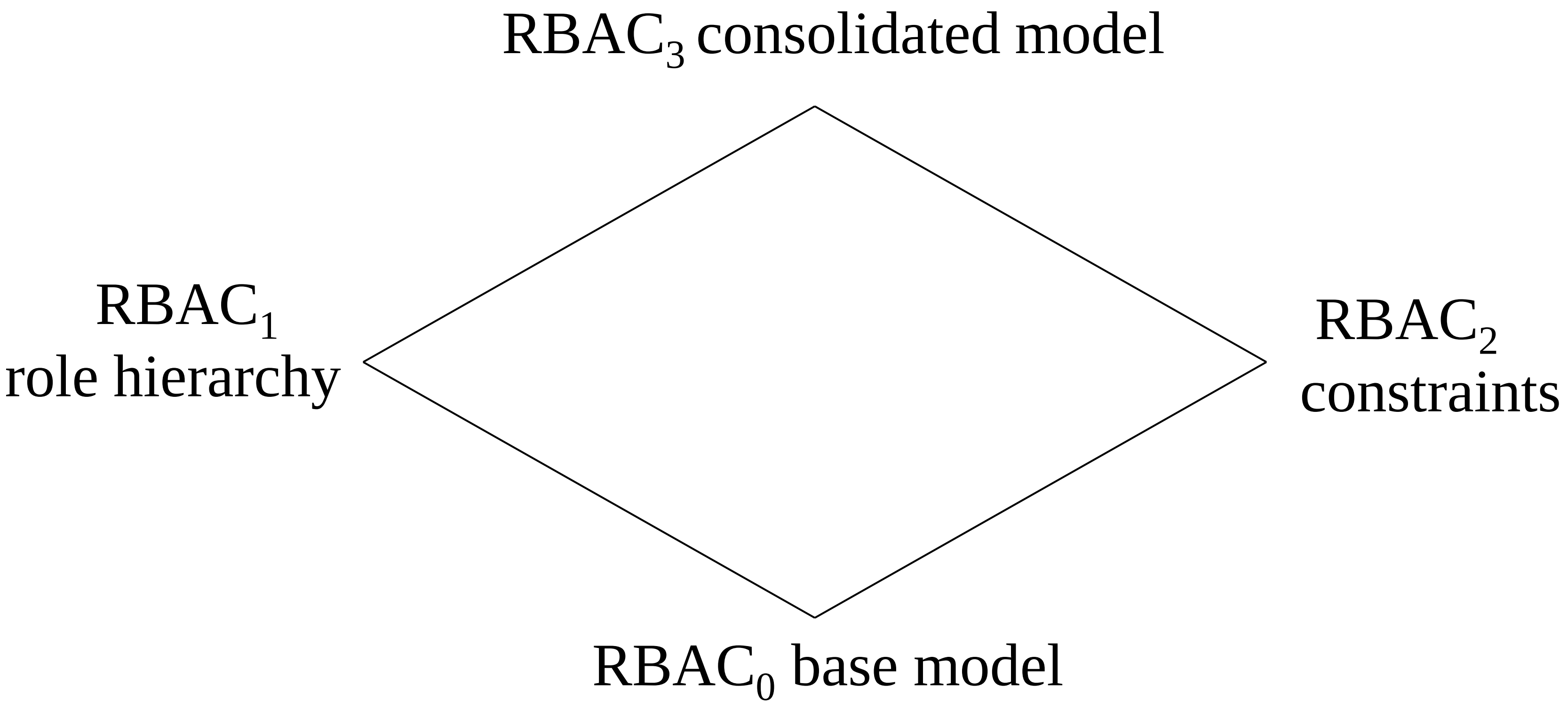
- ***Role based access control (RBAC)***
  - To facilitate the ***security management in multi-user , multi-application systems***
  - Minimum requirements:
    - Associate ***roles*** with ***each individual***.
    - Each ***role*** defines ***a specific set of operations*** that the individual acting in that role may perform.
    - An individual needs to be ***authenticated***, chooses a ***role assigned to the individual***, and accesses information according to ***operations needed for the role.***

RBAC [4,5]

- Users: human
- Roles: job function (title)
- Permissions: approval of a mode of access
  - Always positive
  - Abstract representation
  - Can apply to single object or to many



RBAC Family



RBAC$_3$ consolidated model

RBAC$_1$
role hierarchy

RBAC$_2$
constraints

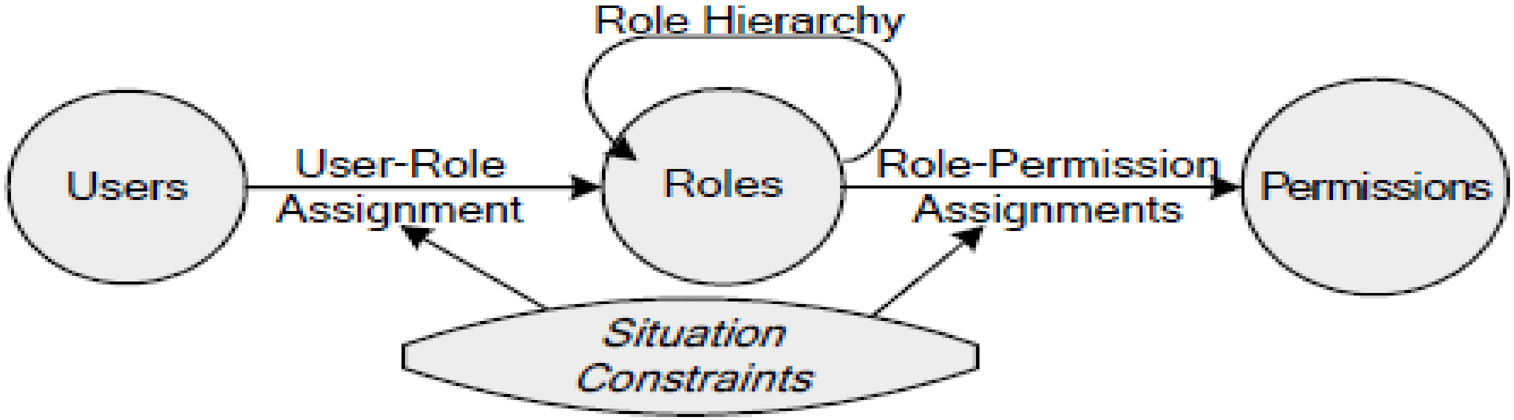
RBAC$_0$ base model

RBAC Family (cont.)

- $RBAC_0$: the base model indicating that it is the minimum requirement for RBAC
- $RBAC_1$: include $RBAC_0$ and support of ***role hierarchy***
  - Inheritance among roles
  - Inheritance of permission from junior role (bottom) to senior role (top)
- $RBAC_2$: include $RBAC_0$ and support of ***constraints***
  - Enforces high-level organizational policies, such as mutually exclusive roles
- $RBAC_3$: combine $RBAC_1$ and $RBAC_2$

Situation-Aware Access Control [6]

- Situation-aware access control model incorporates ***situation-awareness*** into RBAC
  - Example, only when the user with the role of a teacher in the Smart Classroom during the class time, the user can create a group discussion



References

1. M. E. Whitman and H. J. Mattord, *Principles of Information Security*, *Thomson Course Technology*, 5rd edition, 2014.
2. M. Bishop, *Introduction to Computer Security*, Addison-Wesley, 2005, Chapter 11, 14
3. Comparing ACLs and Capabilities, http://www.eros-os.org/essays/ACLSvCaps.html
4. Sandhu, R., Coyne, E.J., Feinstein, H.L. and Youman, C.E. "Role-Based Access Control Models“ *IEEE Computer* (IEEE Press) **29** (2): 38–47, 1996
5. Role Based Access Control and Role Based Security, http://csrc.nist.gov/groups/SNS/rbac/
6. S. S. Yau, Y. Yao, and V. Banga, “Situation-aware access control for service-oriented autonomous decentralized systems”, *Proc. International Symposium on Autonomous Decentralized Systems (ISADS)*, 2005, pp. 17-24