Mississippi State UNIVERSITY

J. A. "Drew" Hamilton, Jr., Ph.D.

Director, Center for Cyber Innovation

Professor, Computer Science & Engineering

CCI
Post Office Box 9627
Mississippi State, MS 39762

Voice: (662) 325-2294
Fax: (662) 325-7692
hamilton@cci.msstate.edu

Identity & Access Management

Reference:

Drew Hamilton Lecture Notes
William Lee

Security+ Exam Guide, 5th ed.

Conklin, White, Cothren, Davis and Williams

Domain Outline

- Identity, Access and Accounts
- Identity and Access Services
- Identity and Access Management Controls

Identity, Access and Accounts
Identity & Access Mgt.

Reference:

Drew Hamilton Lecture Notes

William Lee

Security+ Exam Guide, 5th ed.

Conklin, White, Cothren, Davis and Williams

Secure Identities and Identity Management (IdM)

- "Secure identities" have three key aspects: uniqueness, nondescriptness, and issuance.
- Uniqueness – identifier is specific to the individual and no two identifiers may be the same
- Nondescriptness – no piece of credentials should give away who owns the account
- Issuance – identities have been provided by an outside authority

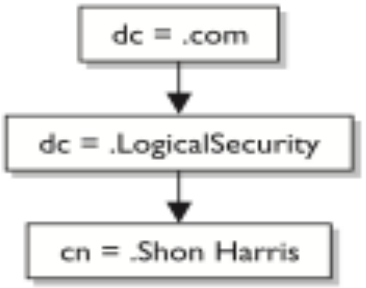
Identity Management

- Identity Management (IdM) technologies help to identify, authenticate, and authorize activities
- High levels of IdM complexity are forcing out traditional IdM manual processes and replacing them with automated ones.
- Many IdM solutions: Directories, Web Access Management, Password Management, Legacy Single Sign – On, Account Management, and Profile Update

Directories

- Generally based on a combination of a database format (X.500, etc.) and a protocol that facilitates user interaction with the directory (LDAP, etc.)
- Objects are managed by a "directory service" that allows administrators to configure and manage their security settings
- How does directory service organize things? → namespaces
- LDAP Method: distinguished names (dn) that are composed of common names (cn) and domain components (dc).
- Example:

```
dn: cn=Shon Harris,dc=LogicalSecurity,dc=com
cn: Shon Harris
```



- Are there problems with directories? Yes → Legacy systems may not support current directory software.

Password Management

- Big Problem – Users forget passwords and require password to be reset. There are three major automated solutions for this that help reduce the need for dedicated human workforce: Password Synchronization, Self – Service Password Reset, and Assisted Password Reset
- Password Synchronization → force user to maintain just one complex password that updates all of his other passwords automatically (has obvious problems and obvious benefits)
- Self – Service Password Reset → password resets are performed using already authenticated external accounts (links sent via e – mail, etc.) or through authorization questions. If the test is passed, the user can reset his password.
- Assisted Password Reset → aid help – desk employees in performing password resets by providing a platform to authenticate users prior to their interaction with the help – desk (usually via personal questions) and forcing the user to change their password after the reset so that the help – desk employee will not know what the password is

Legacy Single Sign – On And Account Management

- Single Sign On (SSO technologies) authenticate one user at a time with no need for re-authentication. SSO technologies are different from password synchronization because a password is sent to ONE authentication system which then communicates with the other authentication systems across the network. In password synchronization, you must login to each different authentication system within the network separately (even though this log-in will be with the same password each time since updating one password updates all of the rest)
- Possible test question → Cons of SSO? Expensive and provides a single point of failure. Shut down the SSO, and everything goes down.

Account Management

- Account management deals with the creation and deletion of user accounts along with the modification of the privileges of those accounts. Often, this is done manually, which is not ideal. Administrators may provide too much access and become bogged down with the workload from changing user accounts across multiple systems. Software helps alleviate both problems by changing user accounts across multiple systems and providing a access request framework.

Provisioning And Profile Updates

- How does everything discussed tie together?

1. Information is pushed from an HR database to a directory (the Identity Repository). Related parties (bosses, etc.) will be notified if necessary.
2. Attributes for different identities will accumulate in the identity repository as the user gains access to more and more information.
3. These attributes will be accessed by IdM solutions in order to test user authorization.

Profile Updates

- Other information about the user may be stored in addition to authorization information. (Date of birth, home address, etc.) When this info is associated with an identity, it is called a Profile. Customer Relationship Management Systems (CRMs) allow a user to modify those parts of the profile that they should be able to view (this is called self - service).

Identity Management

- Today: Centralized Identity Management
  - Overview, Best Practices, and Lessons Learned
  - "Identity 1.0"
- Tomorrow: Federated ID
  - Shibboleth and eduroam
  - "Identity 1.5"
- What's Next: Distributed / User Centric ID
  - Open ID, Cardspace, and Claims
  - "Identity 2.0"

What is Identity Management?

- Lifecycle maintenance of electronic accounts
- Provisioning
  - Account creation
  - Account updates
  - Role maintenance
  - Account removal
- Authentication & Authorization
- Access Control

Why is it Important?

"Your identity is your most valuable possession.

Protect it.

And if anything goes wrong, use your powers!"

– Elastigirl

Kim Cameron's Identity Weblog

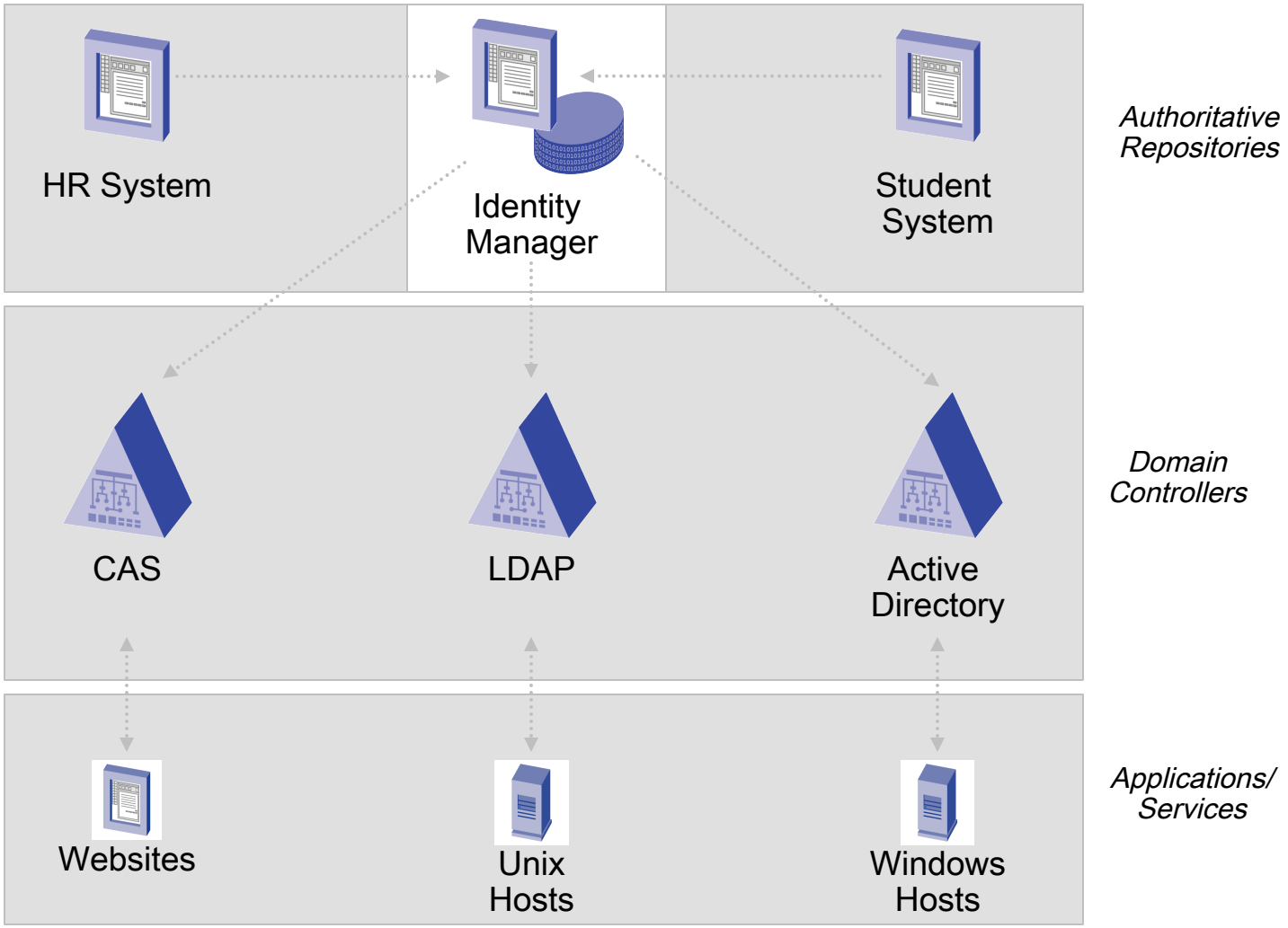
Identify

- **How to identify?**
  - **Usually done once, each device or process is assigned a unique value.**
  - **These values are non-sharable so that activities are traceable to a specific user or process.**
  - **For security these identification values should be non-descriptive.**

The Three A's

- **Authentication**
  - Verify the identify of user or process.
- **Authorization**
  - Restrict user functionality based on identification.
- **Accounting**
  - Tracking resource usage and who is using those resources.

A Provisioning Example



Authentication

- *Something you know*: This may be something you mentally possess. This could be a password, a secret word known by the user and the authenticator.
- *Something you have:* This may be any form of issued or acquired self identification such as:
  - SecurID
  - CryptoCard
  - Activcard
  - SafeWord
  - and many other forms of cards and tags.
- *Something you are:* This being a naturally acquired physical characteristic such as voice, fingerprint, iris pattern and other biometrics.
- In addition to the top three factors, another factor, though indirect, also plays a part in authentication.
  - *Somewhere you are:* This usually is based on either physical or logical location of the user. The use, for example, may be on a terminal that can be used to access certain resources.

Multifactor Authentication

- Five methods of authentication
  - What are you?
  - Where are you?
  - What do you have?
  - What do you know?
  - What do you do?
- Multifactor authentication
  - A combination of two or more of those five methods.
  - The goal is to increase security by having several items that an attack must obtain.
- Identity federation
  - Policies to help manage identifications across organizations.

What Are You?

- **Definition**
  - When someone's anatomy is used for authentication.
- **Positives**
  - Features like eyes and fingerprints will never change.
- **Negatives**
  - Public uneasiness about using biometric scanners.
  - Incontinency caused by having to remove any items that obstructs a particular part of the body, like gloves.
- **Examples**
  - Facial recognition
  - Retinal Scan
  - Fingerprint reader

Where Are You?

- **Definition**
  - When someone's location is used for authentication.
- **Positives**
  - Several methods can be used to find someone's location such as the IP address or a GPS location.
  - Two concurrent connections from different locations can be a sign of something suspicious.
- **Negatives**
  - Someone's location can be spoofed.
- **Examples**
  - Google mail sends an alert to a user if someone is logging in for the first time at a new location.

What Do You Have?

- **Definition**
  - When an item in your possession is used for authentication.
- **Positives**
  - Impersonation is impossible as long as you have the only copy of that item.
- **Negatives**
  - These items are susceptible to being lost, stolen, or duplicated which weakens there reliability.
- **Examples**
  - Identification cards
  - Keys

What Do You Know?

- **Definition**
  - When someone's knowledge is used for authentication.
- **Positives**
  - Most common type of authentication method.
- **Negatives**
  - Knowledge can be forgotten or stolen without the original owner being aware.
- **Examples**
  - Passwords
  - Security questions

What Do You Do?

- **Definition**
  - When someone's physical actions are used for authentication.
- **Positives**
  - Some physical actions can be difficult to reproduce.
- **Negatives**
  - These actions are hard to record without specialized hardware.
- **Examples**
  - Signatures/ signature pads

Single Sign-on

- **Definition**
  - Once a user signs into one service they well gain access to several other connecting services.
  - This requires transitive trust where several services trust the authentication method of the original service.
  - Service can have one-way or two-way trust relationship.
- **Positives**
  - Users do not have to remember several different. passwords instead they can focus on remembering only one.
- **Negatives**
  - All services connected to that login will be compromised if it is stolen.

Single Sign On

(Traditional vs. Federated) (*)

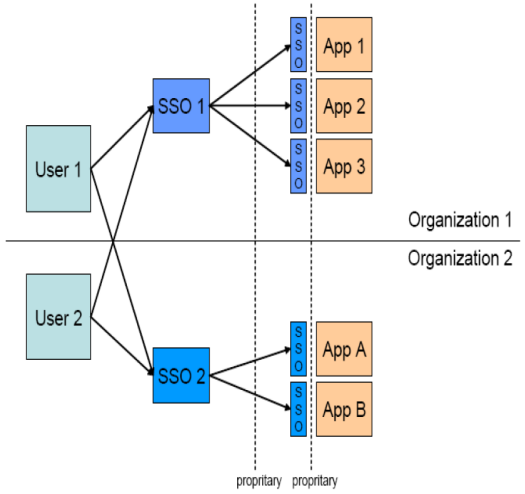


Figure 3.13-1. Traditional SSO: Using Multiple SSOs to Manage Application Access

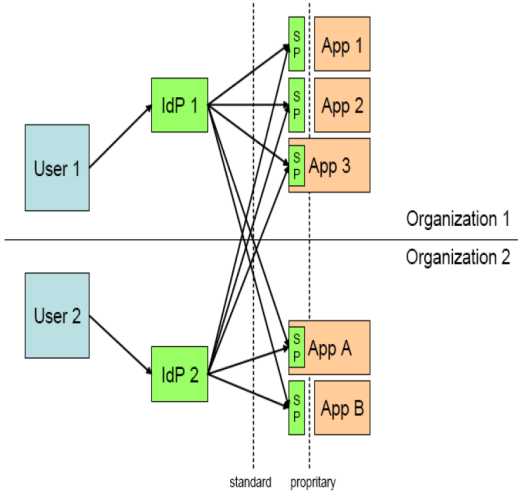


Figure 3.14-1. Typical Federation SSO

(*) “CSC White Paper: “Identity Federation Concepts”
http://assets1.csc.com/cybersecurity/downloads/FIM_White_Paper_Identity_Federation_Concepts.pdf

Account Types

- **Purpose**
  - Rather than managing the rights of each individual user of a system, they can be grouped together. Any changes to a group's rights will automatically propagate to each individual user under that group as well. This grouping system is referred to as an account.
- **Types of accounts**
  - User account
  - Shared/ generic account
  - Service account
  - Privileged account

Account Types

- **User accounts**
  - The lowest level of privilege on a system.
  - Users cannot create their own account so they are made by a privileged user.
  - User identification should be unique for traceability, but simple enough to memorize.
  - Permissions are usually restrictive and assigned by a higher privileged member.
  - Rather than removing members from the system disable their accounts.

Account Types

- **Shared/ generic accounts**
  - **Only use if traceability is not needed because shared accounts cannot accurately be tracked.**
  - **Provides very limited and specific functionality to reduce the scope of a potential attack.**
- **Example**
  - **Guest accounts**
    - **Used in workplaces to allow visitors access to the computer system without having to register for unique identification.**
    - **Has limited functionality like web browsing, printing, etc.**
    - **Tracking is not needed because of the limited functionality.**

Account Types

- **Service accounts**
  - Set up by a privileged account to run simple automated processes that do not require human interaction.
  - It is important to implement security measures that limit their access to prevent an attacker from exploiting the account.
- **Privileged accounts**
  - Root or administrative users.
  - Have unrestricted access over the system.
  - Should always be monitored especially when accessing the remotely.

General Concepts

- **Least privilege**
  - Only giving users the minimum amount of rights necessary to complete their task.
  - This limits the amount of individuals who have access to critical information, which increases security.
- **Onboarding/ offboarding**
  - As new members join a team
    - Create an account
    - Assign an appropriate role
  - As new member leave a team
    - Remove their rights
    - Disable their account

General Concepts

- **Permission auditing and review**
  - Remove any unnecessary accounts from the system.
  - Remove any invalid users from system accounts.
  - Make sure that users who transition often are not maintaining rights from previous teams.
- **Usage auditing and review**
  - Examine logs to view user activity including privileged user activity.
  - Since privileged users have so much power it is important to ensure that they are not performing malicious activity.

General Concepts

- **Time-of-day restrictions**
  - Limit user access to specific hours like weekdays or shift hours.
  - Important for privileged users since they already have an elevated position.
  - Increases security be reducing the number of potential targets for an attacker.
  - Allow for emergency situations to override normal restrictions.
- **Recertification**
  - Ensures that only users that need accounts have accounts.
  - Could be done in person or electronically.

General Concepts

- **Standard naming convention**
  - Account names follow a particular pattern.
  - Allows users to associate names with particular account levels, but this is also true for attackers.
  - Issues can occur when someone changes accounts and has to change heir name to fit the new convention.
- **Account Maintenance**
  - Determines if previously created accounts are still necessary.
  - Checks that the permissions under each account is appropriately configured for their needs.

General Concepts

- **Group-based access control**
  - Allows someone to change the access level of multiple users who are grouped together.
  - This method is quicker than managing the access levels of each individual user.
  - Anyone added to a group will automatically assume the same access level.
- **Location-based policies**
  - Changing users' access rights depending on their current location.

General Concepts

- **Account policy enforcement**
  - Used to ensure users comply to a password creation standard that enforces security.
  - The policy can pertain to any rules used to help secure a users' account.
- **Credential management**
  - Storing user credentials for multiple sites in order to automate the login process.
- **Group policy**
  - Used in Microsoft Windows Enterprise.
  - Allows a privileged user to change registry settings like security and credential management.

General Concepts

- **Password Complexity**
  - All organizations should enforce rules that help create a secure password.
- **Expiration**
  - Accounts should expire when a user is no longer authorized within the system.
  - Windows allows for temporary accounts that have a predefined expiration date.
- **Recovery**
  - Create a simple recovery method to use when access is lost to an account.

General Concepts

- **Disablement**
  - Disabling an account is preferable to removing it because removals can cause item ownership issues.
  - Disablement can be undone while removals cannot.
  - Usually used as a response to an attack or someone leaving the company.
- **Lockout**
  - Similar to disablement, but is considered temporary.
  - Lockouts can occur from a user supplying an incorrect password too many times.
  - Lockouts can range from a few minutes to any extended amount of time depending on the policy.

General Concepts

- **Password history**
  - Any password previously associated with an account.
  - Many systems do not allow users to reuse passwords.
  - Max age forces users to change their password after a certain number of days, while minimum age prevents them from changing their password for so many days.
- **Password reuse**
  - Passwords should not be reused for a substantial amount of time (at least a year or after 6 changes).
  - Old passwords are not secure.
- **Password length**
  - Longer passwords are harder for attackers to crack but easier to remember.

Identity and Access Services
Identity & Access Mgt.

Reference:

Drew Hamilton Lecture Notes

William Lee

Security+ Exam Guide, 5th ed.

Conklin, White, Cothren, Davis and Williams

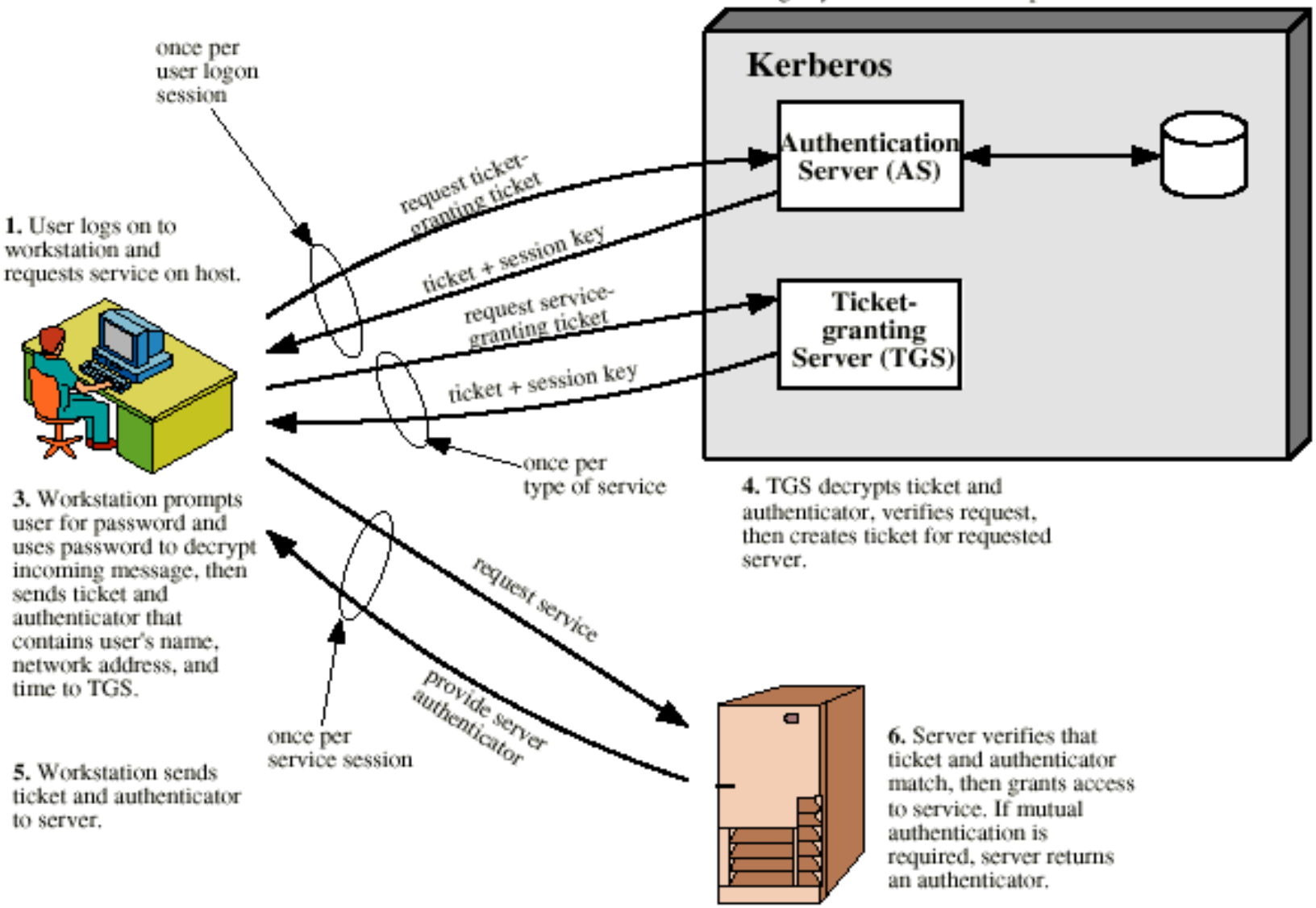
LDAP

- Directory
  - A storage method similar to a database, but has more efficient read times.
  - Directories use a standard known as X.500.
- DAP
  - Directory Access Protocol.
  - Used to access X.500 directories, but is extremely taxing on computer.
- LDAP
  - Lightweight Directory Access Protocols.
  - Only uses essential functions from DAP which allows for less computational resource over TCP connections.
  - Used to handle user authentication and authorization.

Kerberos

- **Definition**
  - **A network authentication protocol for clients and servers.**
  - **Allows for the client to verify itself for the server and the server to verify itself for the client.**
  - **Meant to function in unsecure environments so all data is heavily encrypted.**

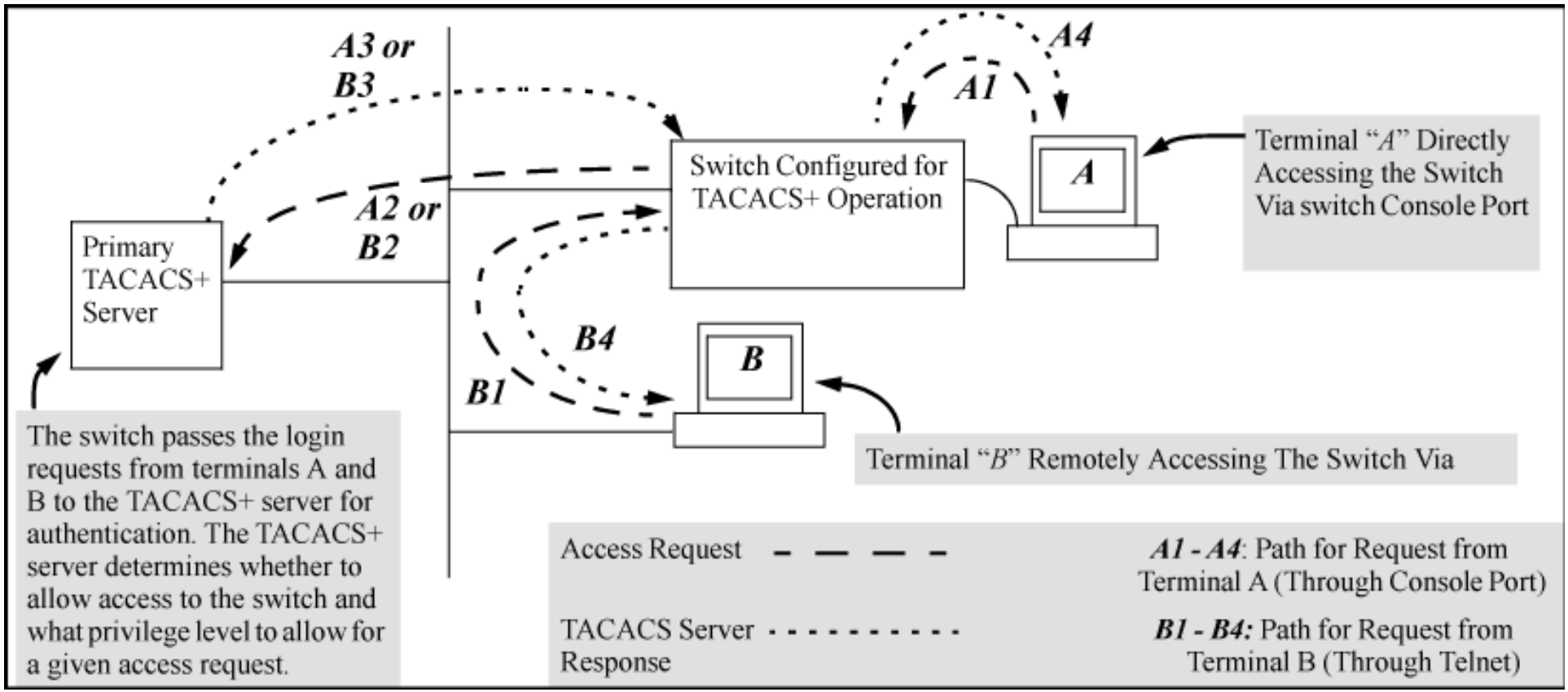
Kerberos


2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
Kerberos
Authentication Server (AS)
Ticket-granting Server (TGS)
once per user logon session
request ticket-granting ticket
ticket + session key
request service-granting ticket
ticket + session key
once per type of service
1. User logs on to workstation and requests service on host.
3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.
request service
provide server authenticator
once per service session
5. Workstation sends ticket and authenticator to server.
4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.
6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

TACAS+

- **Purpose**
  - Client/ server protocol.
  - The client is usually a network access server (NAS) and all communications are encrypted.
  - If the client is a PC then communications are unencrypted, which can allow for potential exploitation.
- **Features**
  - Separates authentication, authorization, and accounting.
  - Usually operates over TCP port 49, but UDP port 49 is also reserved for communication.

TACAS+


A3 or
B3
A4
A1
A2 or
B2
Switch Configured for
TACACS+ Operation
A
Terminal "A" Directly
Accessing the Switch
Via switch Console Port
Primary
TACACS+
Server
B4
B1
B
Terminal "B" Remotely Accessing The Switch Via
The switch passes the login requests from terminals A and B to the TACACS+ server for authentication. The TACACS+ server determines whether to allow access to the switch and what privilege level to allow for a given access request.
Access Request
TACACS Server Response
A1 - A4: Path for Request from Terminal A (Through Console Port)
B1 - B4: Path for Request from Terminal B (Through Telnet)

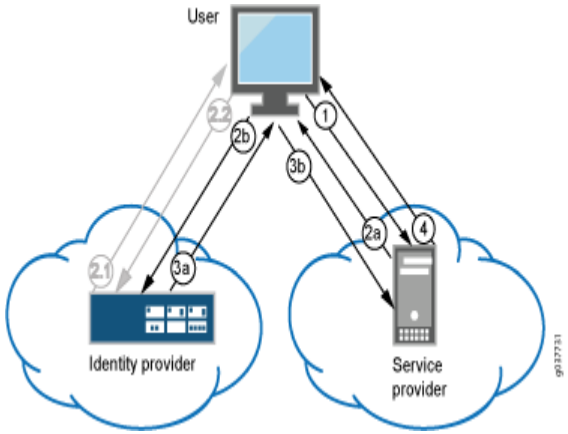
Protocols

- **CHAP**
  - Provides authentication across a point to point link.
- **PAP**
  - Uses a two-way handshake to authenticate the user.
  - Has been depreciated because usernames and passwords are sent in clear text.
- **MSCHAP**
  - Microsoft variant of the CHAP protocol.
- **RADIUS**
  - Client/ server protocol
  - Secure communication with NAS devices
  - Unencrypted communication with PCs

Protocols

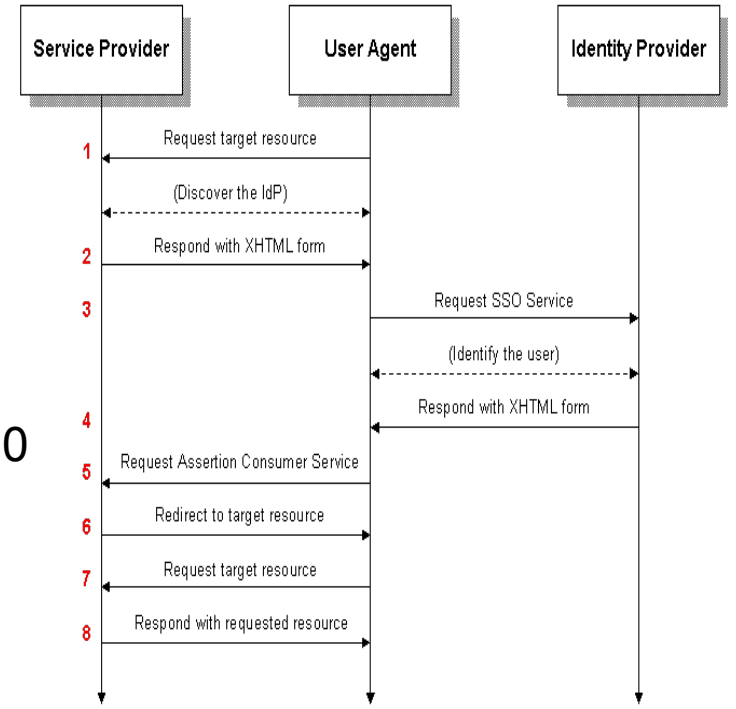
- SAML
  - Used for single sign-on by web applications to ensure identities can be shared and are protected.
- OpenID connect
  - Meant to make authentication easier.
  - Allows third parties to identify users for you by using previously established accounts.
- OAUTH
  - Used with OpenID
  - Through cookies it shares authentication information without sharing login information.

SAML 2.0 – Web SSO Protocol

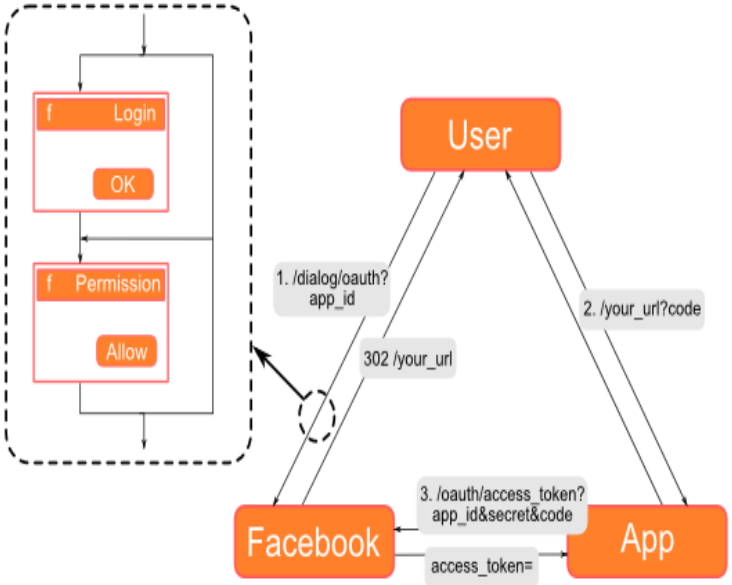


Implementing SAML 2.0 Web Browser SSO for Google Apps
http://www.juniper.net/techpubs/en_US/sa8.0/topics/example/example-simple/secure-access-saml-cloud-googleapps.html

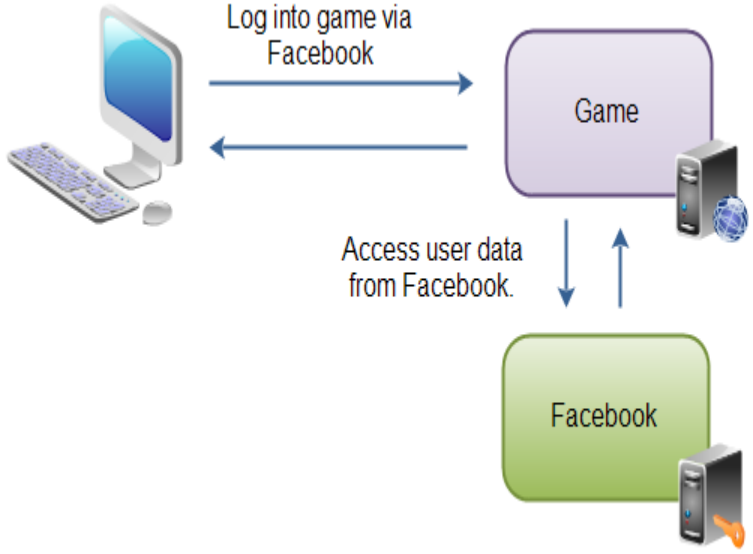
https://en.wikipedia.org/wiki/SAML_2.0



OAuth Flow



http://tutorials.jenkov.com/oauth2/index.html



http://tungwaiyip.info/blog/2011/02/19/
facebook_oauth_authentication_flow

Protocols

- **Shibboleth**
  - Build using SAML.
  - Not widespread.
  - Supports single sign-on across networks.
- **Secure token service**
  - Issues, validates, renews, and cancels secure tokens.
  - A secure token can be used by any service that follows the WS-Trust standard.
  - Solves the problem of authentication in stateless platforms.
- **NTLM**
  - Depreciated Microsoft security protocol for authentication on Window OS.

What is?

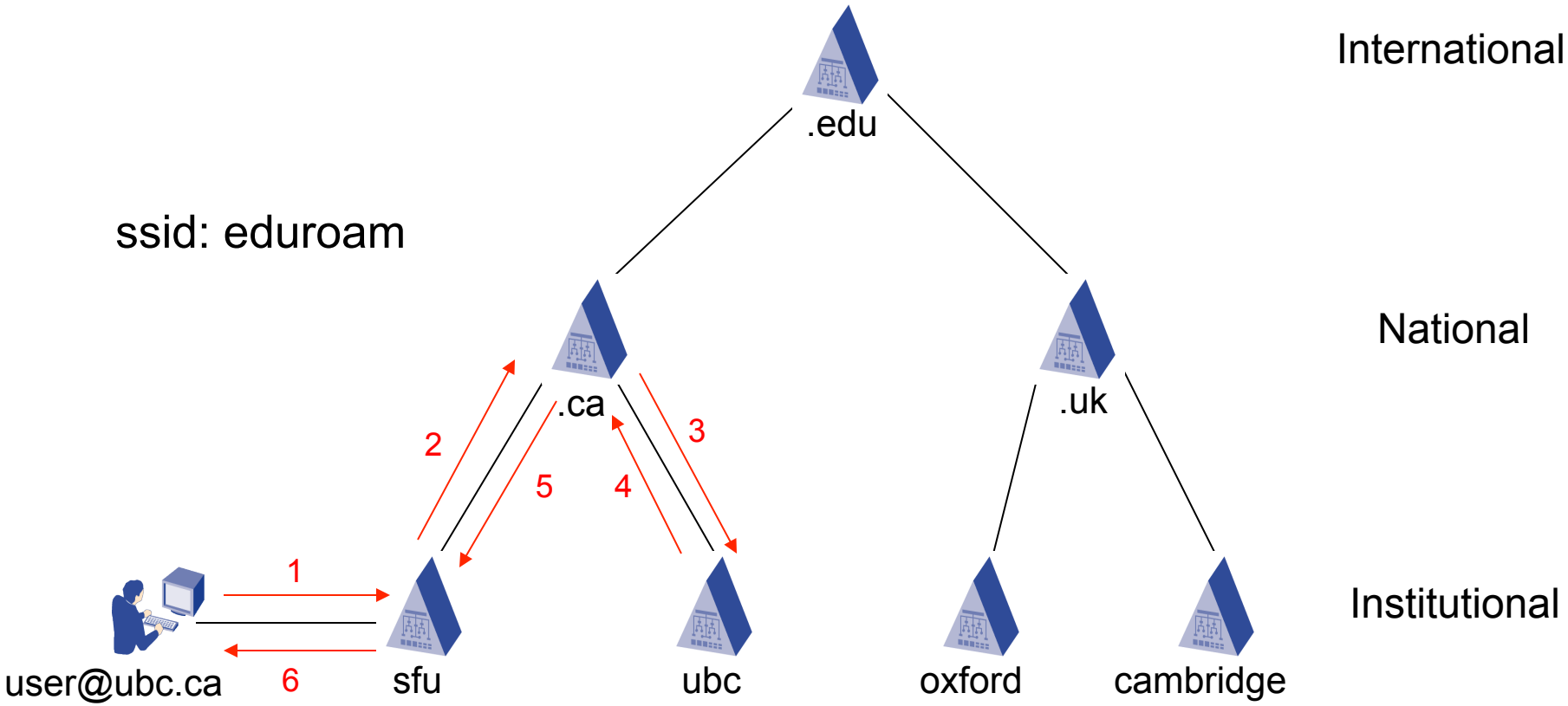
Shibboleth®

- An open source project supporting inter-institutional sharing of web resources subject to access controls.
- Streamlines sharing secured online services
- Leverages campus identity and access management infrastructures
  - sends information about users to resource site
  - enables resource provider to make authorization decisions
- Ideal for lightweight web authentication
  - digital libraries
  - learning object repositories

What is?

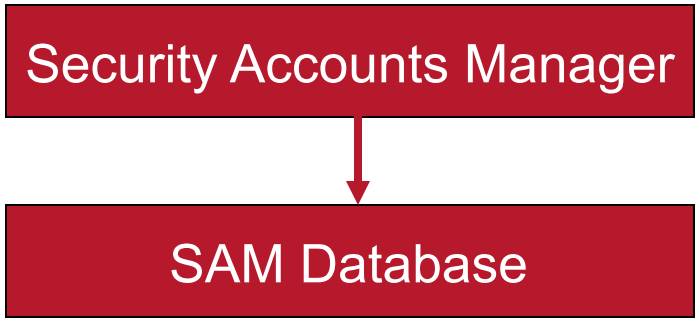
- eduroam stands for Education Roaming
- Originally a European initiative
- Launched in 2003 to deal with the “Roaming Scholar problem”
- RADIUS-based infrastructure
- Uses 802.1X to allow inter-institutional roaming
- Allows users visiting other eduroam institutions to access WLAN using home credentials

How Does it Work?

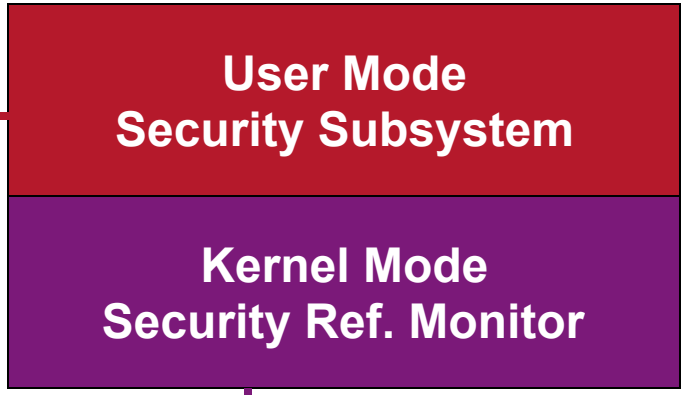


Windows 2000 Passwords

- **LM**
  - an encrypted, fixed, hex no.
- **NT Password Hash**
  - 3 rounds of MD4 hashing algorithm



1. 2 password entries for each account.
2. Format:
   ID:LM representation :NT Hash



1. Checks user and program permissions before allowing access to objects
2. Defines how audit settings translate into the actual capture of events by the Event Log

LM (LanManager) Password Representation

1. Adjust password length to 14 characters by either truncation or padding.
2. Divide string into 2 parts, add one bit of parity to each part.
   - Parity required for using DES
   - Each part used as a key for DES encryption of a hexadecimal number
   - Splitting the string into two parts allows an attacker to attack each half independently
- LM representation is neither a hash nor an encrypted password, it is an encrypted, fixed hex number is which the password is used as the key.

NT Password Representation

1. Adjust password length to 14 characters
2. Use MD-4 hashing algorithm three times to produce a hash of the password.

- NT Password is not salted
- NT password cracking programs only need to access a dictionary.

Conclusion

- Identity practice undergoing dramatic changes
- Users will expect to engage with us in new ways
  - Bring identity information when they join
  - Gradual migration to claim based access
- Prepare by continuing to strengthen and consolidate internal Identity Management
- Target low hanging fruit for Federation
- Keep abreast of user-centric identity management

Identity and Access Management Controls
Identity & Access Mgt.

Reference:

Drew Hamilton Lecture Notes

William Lee

Security+ Exam Guide, 5th ed.

Conklin, White, Cothren, Davis and Williams

A Multi-Layered Privilege Model

- Issues relating to access apply not only to the web application itself but also to the other infrastructure ties which lie beneath it
- In this case, these access controls could be a good alternative:

1. Programmatic Control
2. Discretionary Access Control (DAC)
3. Role-Based Access Control (RBAC)
4. Declarative Control

Programmatic Control

- The matrix of individual database privileges is stored in a table within the database, and applied programmatically to enforce access control decisions.

- The classification of user roles provides a shortcut for applying certain access control checks, and this is also applied programmatically

- Advantages:

  - It can be extremely fine-grained

  - It can build in arbitrarily complex logic into the process of carrying out access control decisions within the application

Access Control Model

- **Definition**
  - A variety of protection schemes to prevent unauthorized access to a computer system or network.
- **MAC**
  - Mandatory access control.
  - Restricts access to objects based on its sensitivity and the users clearance level.
  - High, medium, low, confidential, private, and public.
- **DAC**
  - Discretionary access control.
  - The owner of an object decides who else should have access.

Access Control Model

- ABAC
  - Attribute-based access control.
  - Allows for Boolean logic in access decision.
  - Access depends on particular attributes of the object or environment.
- Role-based access control
  - Each user is defined a set of roles.
  - This role defines their access privileges.
- Rule-based access control
  - Sets of rules contained in an ACL determines if a user has access.

Discretionary Access Control (DAC)

Various appication users have privileges to create user accounts

Closed DAC Model

Access denied unless explicitly granted

Open DAC Model

Access is permitted unless explicitly with-drawn

Role-Based Access Control (RBAC)

- Named roles which contain different sets of specific privileges. Each user is assigned to one of these roles.

- Enables many unauthorized requests to be quickly rejected with a minimum amount of processing being performed

- Number of roles should be balanced

  Too many roles ➔ Difficult to manage accurately

  Too few roles ➔ Resulting roles will be assigned privileges that are not strictly necessary for performance of their function

Declarative Control

- Uses restricted database accounts when accessing the database
- Employs different accounts for different groups of users with each account having the least level of privilege necessary for carrying out the actions which that groupis permitted to perform
- Advantage: Even if a user finds a means of breaching the access controls implemented within the application tier, so as to perform a sensitive action such as adding a new user, they will be prevented from doing so because the database account that they are using does not have the required privileges within the database

Attacking Access Controls

Finding a break in access controls is almost trivial

- ➔ **Request a common administrative URL and gain direct access to the functionality.**
- ➔ **In other cases, it may be very hard, and subtle defects may lurk deep within application logic, particularly in complex, high-security applications.**
- ➔ **The most important lesson when attacking access controls is to look everywhere. If you are struggling to make progress, be patient and test every single step of every application function. A bug that allows you to own the entire application may be just around the corner.**

Physical Access Control

- **Definition**
  - Identifying and enforcing who can physically have access to a system.
- **Types of physical access**
  - Proximity cards and smart cards
  - Biometric factors
    - Fingerprint scanner
    - Retinal scanner
    - Facial recognition
- **False positives and false negatives**
  - When someone is authenticated into a system that they should have access to.

False Positives/ Negatives

- False positives
  - When someone is authenticated into a system that they should not have access.
- False negative
  - When someone cannot be authenticated into a system even thought they should have access.
- False acceptance rate
  - How many false positives that are allowed in a system.
- False rejection rate
  - How many false negatives that are allowed in a system.
- Crossover error rate
  - Rate where both accept and reject error rates are equal.

Tokens

- Hardware tokens
  - The value of the physical token constantly changes.
  - When logging in the value of the token must be entered.
  - Even if an attacker has a username and password they still won't have the unique token.
- Software token
  - Does not require for the user to have a physical separate device.
  - Two way authentication can be enforced through a pin or a symmetric key.
- HOTP/ TOTP
  - Method of getting a one time password through a hashed message or current timestamp.

Certificate-Based Authentication

- **Definition**
  - Being authenticated by providing a certificate.
- **PIV/CAC/Smart Card**
  - Methods of carrying a users credentials on a card that can be read by a computer for authentication.
- **IEEE 802.1x**
  - An authentication standard that supports port based authentication between devices and users.

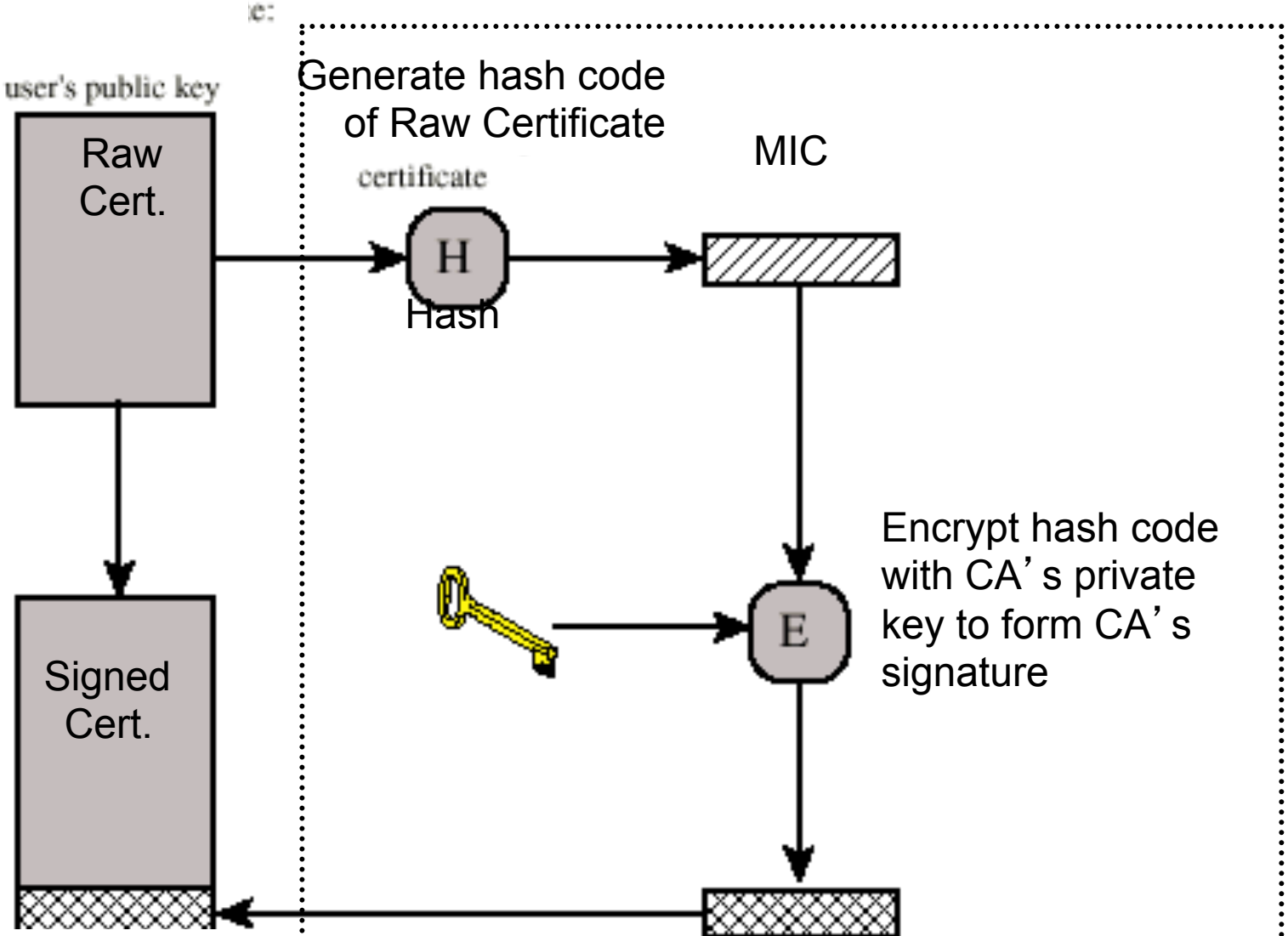
X.509 Authentication Service

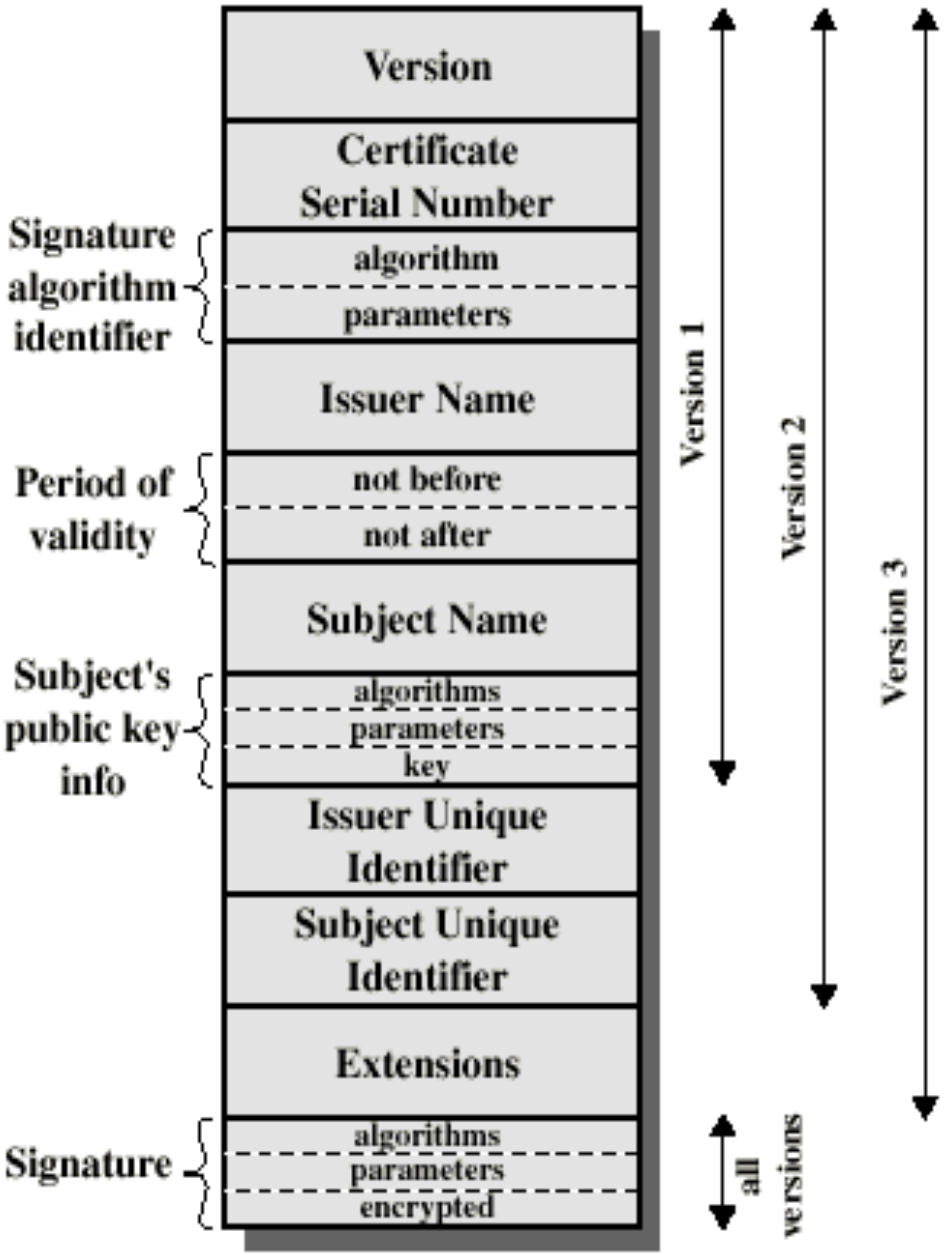
- An International Telecommunications Union (ITU) recommendation (versus "standard") for allowing computer host or users to securely identify themselves over a network.
- An X.509 certificate purchased from a "Certificate Authority" (trusted third party) allows a merchant to give you his public key in a way that your Browser can generate a session key for a transaction, and securely send that to the merchant for use during the transaction (padlock icon on screen closes to indicate transmissions are encrypted).

X.509 Authentication Service (2)

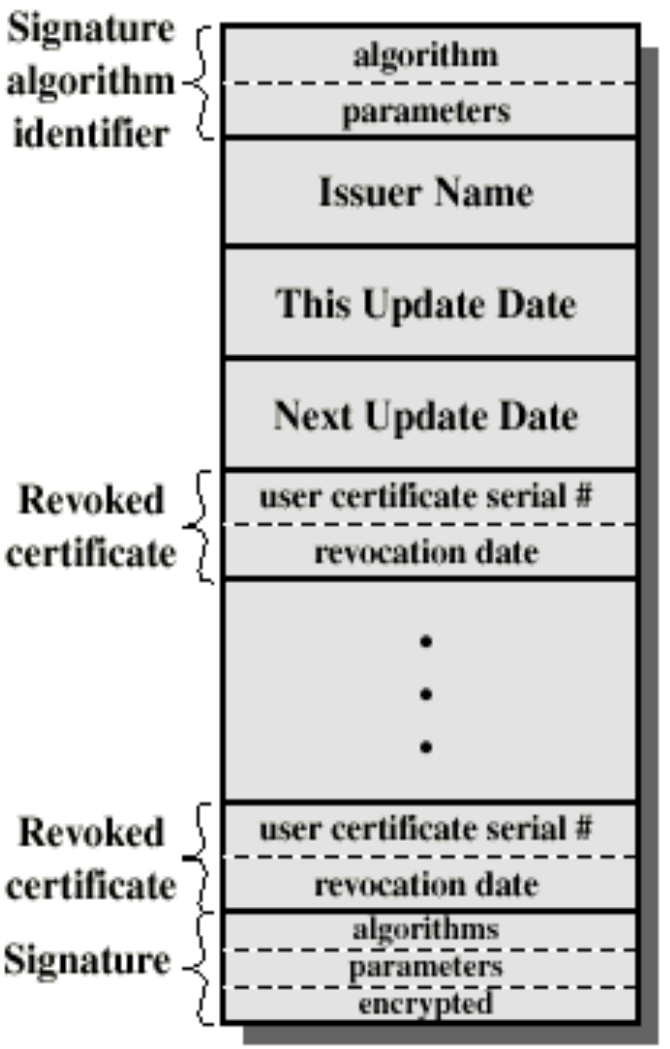
- Once a session key is established, no one can "high jack" the session (for example, after your enter your credit card information, an intruder can not change the order and delivery address).
- User only needs a Browser that can encrypt/ decrypt with the appropriate algorithm, and generate session keys from truly random numbers.
- Merchant's Certificate is available to the public, only the secret key must be protected. Certificates can be cancelled if secret key is compromised.

Raw “Certificate” has user name, public key, expiration date, ...





(a) X.509 Certificate



(b) Certificate Revocation List

File System and Database Security

- **File system security**
  - Methods and processes to prevent unauthorized access and alterations to a file system.
  - Done through user level access differentiation and access control models.
- **Database security**
  - Used to prevent unauthorized users from retrieving information from the database.
  - Access is managed by defined permissions for specific users.
  - Encryption is used to protect the data even if it is copied.

Common Categories of Vulnerabilities

1. **Broken Authentication**
   - Encompasses various defects within the application's login mechanism
2. **Broken Access Controls**
   - Application fails to properly protect access to data and its functionality
3. **SQL Injection**
   - Enables an attacker to submit crafted input to interfere with the application's interaction with back-end databases.
4. **Cross-Site Scripting**
   - Enables an attacker to target other users of the application
5. **Information Leakage**
   - An application divulges sensitive information that is of use to an attacker in developing an assault against the application, through defective error handling or other behavior

Vertical vs Horizontal Access Controls

- **Vertical Access Controls:**
  Allow different types of users to access different parts of the application's functionality

  ➔ **Division between ordinary users and administrators**

- **Horizontal Access Controls:**
  Allow users to access a certain subset of a wider range of resources of the same type

  ➔ **Web mail application may allow you to read your email but no one else's; you can only see your own details**

Access Control Vulnerabilities

- Access controls are broken if any user is able to access functionality or resources for which he is not authorized
- Among the mot commonly encountered categories of web application vulnerabilities
- Two main types of attack against access controls

1. Vertical privilege escalation

   When a user can perform functions that their assigned role does not permit them to do

2. Horizontal privilege escalation

   When a user can view or modify resources to which he is not entitled

Access Control Security and its Weaknesses

1. Completely Unprotected Functionality
2. Identifier-Based Functions
3. Multistage Functions
4. Static Files

Completely Unprotected Functionality

In many cases of broken access controls, sensitive functionality and resources can be accessed by anyone who knows the relevant URL

➔ E.g. when https://wahh-app.com/admin/ allows user to enter certain user interface.

➔ Weaknesses:

1. URL can be guessed (especially by insider)
2. Link appears in browser histories and the logs of web servers and proxy servers
3. Users may write them down, bookmark them or email them around
4. They are not normally changed periodically, as passwords should be
5. When users change job roles, and their access to administrative functionality needs to be withdrawn, there is no way to delete their knowledge of a particular URL.

Identifier-Based Functions

When a function of an application is used to gain access to a specific resource, it is very common to see an identifier for the requested resource being passed to the server in a request parameter, either within the URL query string or the body of a post request

➔ When the user who owns the document is logged in, a link to this URL is displayed on the user's My Documents page. Other users do not see this link. In order to be able to open the link/application an attacker needs to know the name of the application page and the identifier of the document he wishes to view.

➔ Weaknesses:

1. Passwords often easy to guess
2. Lots of people write down resources identifiers or save them on their computer, so easy to find

Multistage Functions

Involves capturing different items of data from the user at each stage. This data is strictly checked when first submitted and then is usally passed to each subsequent stage, using hidden fields in an HTML form.

Main Weaknesses:

1. **Often assumed by the developers is that any user who reaches the later stages of the process must have the relevant privileges because this was verified at the earlier stages**
2. **Also often assumed is that people will access application pages in the intended sequence; by taking "other path" people could avoid user identification**

Static Files

In some cases, requests for protected resources are made directly to the static resources themselves, which are located within the web root of the server.

➔ e.g. an online publisher may allow users to browse its book catalog and purchase ebooks for download. Once the payment has been made, the user is directed to a download URL.

As this is a completely static resource, it does not execute on the server, and its contents are simply returned directly by the web server. Hence, the resource itself cannot implement any logic to verify that the requisting user has the privileges.

When static resources are accessed in this way, it is highly likely that there are no effective access controls protecting them and that anyone who knows the URL naming scheme can exploit this to access ay resources they desire.

Summary

- Identity, Access and Accounts
- Identity and Access Services
- Identity and Access Management Controls