Identity and Access Management

Securing Systems in the Digital Age

Instructor Name

School/College Name

March 11, 2025

Welcome to Identity and Access Management: Securing the Digital Front Door

- **Identity and Access Management (IAM)** is the framework for ensuring the right individuals access the right resources at the right times for the right reasons.
- Understanding IAM is essential for protecting systems against unauthorized access and potential security breaches.
- Modern organizations typically manage thousands of digital identities, making systematic approaches necessary.
- IAM encompasses both technical systems and policies that govern how identities are created, verified, and granted permissions.

Key Question

How do we ensure only authorized users can access sensitive information while still making systems convenient to use?

Why IAM Matters: Real-World Security Scenarios

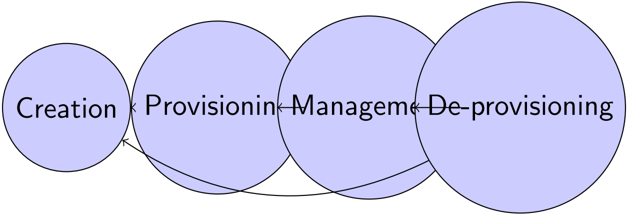
- A hospital must ensure patient records are only accessible to authorized healthcare providers while maintaining ease of access in emergencies.
- Financial institutions need to verify identities before allowing transfers, with higher security requirements for larger transactions.
- Companies must immediately remove access when employees leave to prevent security vulnerabilities from lingering accounts.
- Educational institutions must provide appropriate access levels to students, faculty, and staff while protecting sensitive data.

Example

The 2020 SolarWinds breach occurred partly because attackers gained privileged access credentials, showing how IAM failures can have devastating consequences across thousands of organizations.

The IAM Lifecycle: An Overview

- The **IAM lifecycle** begins with identity creation and verification to establish who a user is.
- Once verified, users receive appropriate access permissions based on their role and needs.
- Throughout the lifecycle, authentication mechanisms verify user identity during each access attempt.
- The cycle concludes with de-provisioning when access is no longer needed or appropriate.



User Account Basics: Creating and Managing Digital Identities

- A **digital identity** is the electronic representation of a person or entity within a system.
- User accounts store essential identifying information such as username, contact details, and authentication credentials.
- Most systems use unique identifiers (like user IDs) that remain consistent even when other account details change.
- Properly structured user accounts enable appropriate access while maintaining security and accountability.

Components of a Digital Identity

- Identifiers (username, email, ID number)
- Authentication data (password hash, biometric templates)
- Profile information (name, department, contact info)
- Access rights and permissions

The Art of Provisioning: Adding Users Securely

- **Provisioning** is the process of creating user accounts and assigning appropriate access rights to resources.
- Automated provisioning reduces human error and ensures consistency in how accounts are created and configured.
- Proper provisioning includes verification of identity before granting access to sensitive systems.
- Organizations typically develop standardized workflows to ensure all necessary approvals are obtained before access is granted.

Provisioning Type	**Best Used For**
Manual	Small organizations, specialized roles
Self-service	Common resources, low-risk assets
Automated	Large organizations, standard onboarding
Just-in-time	Temporary access needs

De-provisioning: Why Removing Access Matters

- **De-provisioning** is the systematic removal of access rights when they are no longer needed or authorized.
- Orphaned accounts (accounts belonging to former employees) represent significant security vulnerabilities if not properly managed.
- Effective de-provisioning should be timely, complete, and documented to maintain security compliance.
- Regular access reviews help identify accounts that should be de-provisioned but might have been overlooked.

Security Risk

The 2020 IBM Cost of a Data Breach Report found that organizations with orphaned accounts experienced higher data breach costs, with abandoned credentials frequently exploited by attackers.

Identity Proofing: Verifying Who's Who

- **Identity proofing** is the process of verifying that a person is who they claim to be before creating their digital identity.
- The strength of identity proofing should match the sensitivity of the resources the user will access.
- Common methods include document verification (ID cards, passports), knowledge-based verification, and biometric matching.
- The National Institute of Standards and Technology (NIST) defines three assurance levels for identity proofing, from basic to highly secure.

IAL1	**IAL2**	**IAL3**
Self-assertion No validation Minimal assurance	ID verification Remote or in-person Moderate assurance	In-person proofing Physical biometrics High assurance

Understanding Permissions: The Building Blocks of Access

- **Permissions** are specific authorizations that allow users to perform particular actions on resources.
- Common permission types include read, write, execute, modify, and delete capabilities.
- Permissions can be assigned directly to users or indirectly through groups, roles, or attributes.
- Well-designed permission structures balance security needs with usability concerns.

CRUD Permissions Model

Most systems organize permissions around four basic operations:

- **C**reate: Ability to generate new data or resources
- **R**ead: Ability to view existing data
- **U**pdate: Ability to modify existing data
- **D**elete: Ability to remove data or resources

Permission Assignment: Who Gets What Access and Why

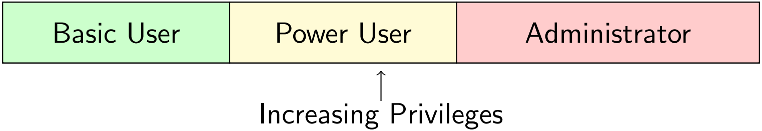
- Permission assignment should be based on legitimate business needs rather than convenience or hierarchy.
- **Segregation of duties** ensures that critical functions are divided among different individuals to prevent fraud.
- Permissions can be assigned through static methods (manual assignment) or dynamic methods (calculated at access time).
- Regular permission audits help identify and correct inappropriate access rights before they cause security incidents.

Example

A financial system might require two different employees to create and approve payment transactions, preventing any single individual from both creating and authorizing fraudulent payments.

The Principle of Least Privilege: Need-to-Know Access

- The **principle of least privilege** states that users should be given only the minimum access rights needed to perform their job functions.
- Implementing least privilege reduces the potential damage from compromised accounts or insider threats.
- This principle applies to both human users and system processes or applications.
- Temporary privilege elevation can be used when higher-level access is occasionally needed but not justified permanently.



Basic User

Power User

Administrator

Increasing Privileges

Access Control Models: Different Approaches to Security

- **Access control models** provide structured frameworks for determining who can access what resources.
- Different models address varying security needs, organizational structures, and compliance requirements.
- Most modern systems implement hybrid approaches that combine elements from multiple access control models.
- The choice of access control model significantly impacts both security posture and administrative complexity.

Access Control Model	**Key Characteristic**
Mandatory (MAC)	System-enforced based on sensitivity labels
Discretionary (DAC)	Owner-determined access permissions
Role-based (RBAC)	Access based on job functions/roles
Rule-based	Access based on predefined rules
Attribute-based (ABAC)	Dynamic access based on attributes

Mandatory vs. Discretionary Access Control: Understanding the Differences

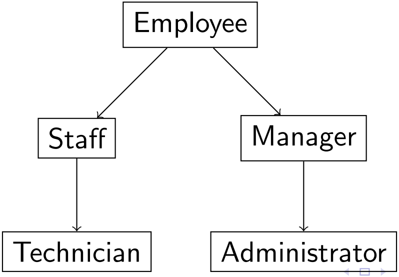
- **Mandatory Access Control (MAC)** uses system-enforced security labels that cannot be altered by users.
- MAC assigns sensitivity labels to resources and clearance levels to users, with access granted only when clearance meets or exceeds sensitivity.
- **Discretionary Access Control (DAC)** allows resource owners to determine who can access their resources.
- DAC is more flexible but potentially less secure, as permissions are at the discretion of individual users.

When to Use Each Model

- **MAC**: Military systems, government classified information, highly regulated industries
- **DAC**: Collaborative environments, file sharing systems, situations requiring user autonomy

Role-Based Access Control: Organizing Permissions by Job Function

- **Role-Based Access Control (RBAC)** assigns permissions to roles, and roles to users based on their job responsibilities.
- RBAC simplifies administration by managing permissions at the role level rather than individually for each user.
- When employees change positions, administrators need only assign them to different roles rather than reconfiguring all permissions.
- Roles can be hierarchical, allowing permissions to be inherited from more general to more specific job functions.



Rule-Based and Attribute-Based Access: Dynamic Security Controls

- **Rule-Based Access Control** uses predefined rules to determine access permissions based on specific conditions.
- Rules can incorporate factors such as time of day, network location, or previous access patterns.
- **Attribute-Based Access Control (ABAC)** makes access decisions based on attributes of users, resources, actions, and environment.
- ABAC offers more granular control than RBAC but requires more complex policy definition and evaluation.

ABAC Policy Example

IF user.department = "Finance" AND resource.type = "Financial Report" AND action = "view" AND environment.time BETWEEN "9:00" AND "17:00" THEN permit

Time-Based Restrictions: When Access Matters

- **Time-based access restrictions** limit when users can access resources, regardless of their identities or roles.
- Time restrictions help prevent unauthorized access outside normal business hours when legitimate use is unlikely.
- These controls can be used to enforce maintenance windows, scheduled system upgrades, or compliance with labor regulations.
- Effective time-based controls must account for different time zones, holidays, and emergency access procedures.

Time Restriction Type	**Use Case**
Hours of operation	Limiting access to business applications to normal working hours (8am-6pm)
Day of week	Restricting system maintenance tasks to weekends only
Date range	Allowing temporary contractors access only during their contract period
Seasonal	Enabling tax filing systems only during tax season

Authentication 101: Proving Identity in the Digital World

- **Authentication** is the process of verifying that a user is who they claim to be when accessing a system.
- Authentication is distinct from authorization, which determines what an authenticated user is allowed to do.
- Strong authentication typically relies on multiple factors rather than a single piece of evidence.
- Authentication strength should be proportional to the sensitivity of the information or systems being protected.

Authentication vs. Authorization

- **Authentication** answers: "Are you who you say you are?"
- **Authorization** answers: "What are you allowed to do?"
- Both are required for a complete access control system

Password Best Practices: Length, Complexity, and Management

- **Passwords** remain the most common authentication method despite their known security limitations.
- Password strength is primarily determined by length, with longer passwords being exponentially harder to crack.
- Modern guidance emphasizes memorable passphrases (longer but simpler to remember) over complex but short passwords.
- Organizations should implement password policies that balance security requirements with usability considerations.

Password Characteristic	**Recommendation**	**Rationale**
Length	Minimum 12 characters	Increases attack complexity
Complexity	Mix of character types	Increases possible combinations
Uniqueness	Different for each service	Prevents credential stuffing
Expiration	Only if compromise suspected	Reduces password fatigue

Password Managers: Simplifying Secure Password Usage

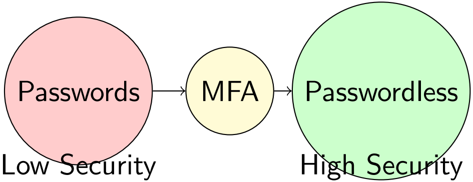
- A **password manager** is a tool that securely stores, generates, and autofills complex unique passwords.
- Using a password manager allows implementation of best practices without requiring users to memorize dozens of complex passwords.
- Password managers typically encrypt their databases with a single master password, creating a secure but convenient system.
- Enterprise password managers offer additional features like shared credentials, access logs, and emergency access protocols.

Security Consideration

While password managers create a single point of failure, the security benefits of using unique, complex passwords for each service far outweigh this risk when proper precautions are taken.

The Future is Passwordless: Modern Authentication Trends

- **Passwordless authentication** eliminates passwords in favor of more secure and convenient methods.
- Common passwordless methods include biometrics, hardware security keys, and cryptographic certificates.
- Standards like FIDO2 and WebAuthn are enabling widespread adoption of passwordless authentication across platforms.
- Passwordless approaches improve security by eliminating password-related vulnerabilities like phishing and credential stuffing.



Multifactor Authentication: Beyond the Password

- **Multifactor authentication (MFA)** requires users to provide two or more verification factors to gain access to a resource.
- MFA significantly reduces the risk of unauthorized access even if one authentication factor is compromised.
- The security benefit of MFA comes from requiring attackers to compromise multiple independent verification methods.
- Organizations can implement MFA with varying levels of strictness depending on risk tolerance and usability requirements.

MFA Implementation Options

- Required for all users and all access
- Required for sensitive operations only
- Required based on risk factors (new device, unusual location)
- Required for specific user roles or resource types

Something You Know, Have, Are, or Where You Are: The Four Factors

- Authentication factors are categorized by the type of verification they provide, with each category offering different security properties.
- **Something you know** includes passwords, PINs, and security questions that rely on secret knowledge.
- **Something you have** includes physical devices like phones, smart cards, or security keys that must be in the user's possession.
- **Something you are** includes biometric characteristics like fingerprints or facial features that are unique to the individual.
- **Somewhere you are** uses location data to verify that access attempts come from expected or approved locations.

Example

Withdrawing money from an ATM typically uses two-factor authentication: something you have (the bank card) and something you know (the PIN).

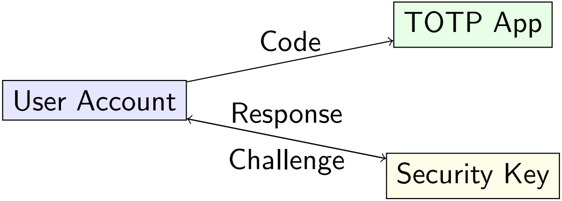
Biometrics in Action: Using Physical Traits for Authentication

- **Biometric authentication** uses unique physical or behavioral characteristics to verify a person's identity.
- Common biometric methods include fingerprint scanning, facial recognition, iris scanning, and voice recognition.
- Biometrics offer convenience because they don't need to be remembered and are difficult to transfer between individuals.
- Unlike passwords, biometric characteristics cannot be changed if compromised, creating unique security challenges.

Biometric Type	**Advantages**	**Limitations**
Fingerprint	Fast, accurate, widely accepted	Can be affected by injuries
Facial recognition	Non-intrusive, improving rapidly	Sensitive to lighting, aging
Voice recognition	Works remotely by phone	Background noise, illness affects
Iris scanning	Extremely accurate, stable	Specialized equipment needed

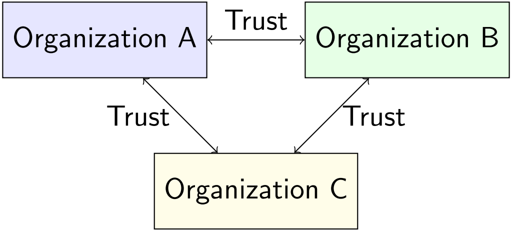
Authentication Tokens and Security Keys: Physical Security Tools

- **Authentication tokens** generate temporary codes or cryptographic responses that prove the user possesses the token.
- **Hard tokens** are physical devices dedicated to authentication, while **soft tokens** are software implementations on general-purpose devices.
- Time-based One-Time Password (TOTP) tokens generate codes that change periodically and become invalid after a short time.
- **Security keys** like FIDO U2F devices use cryptographic challenges and responses that protect against phishing attacks.



Federation: Extending Trust Across Organizations

- **Identity federation** allows organizations to recognize and accept identity credentials issued by trusted external parties.
- Federation establishes trust relationships that enable secure authentication across organizational boundaries without duplicate accounts.
- In federated systems, users authenticate with their home organization (identity provider) to access resources at partner organizations (service providers).
- Federation reduces administrative overhead while improving security by centralizing identity management.



Single Sign-On: One Key for Many Doors

- **Single Sign-On (SSO)** allows users to authenticate once and gain access to multiple systems without re-entering credentials.
- SSO improves user experience by eliminating the need to remember and manage multiple sets of credentials.
- From a security perspective, SSO reduces password fatigue and encourages stronger authentication for the single login point.
- SSO can be implemented within a single organization (enterprise SSO) or across multiple organizations (federated SSO).

Security Consideration

While SSO is generally more secure, it creates a single point of failure - if the SSO account is compromised, all connected applications are potentially vulnerable.

LDAP, OAuth, and SAML: Understanding Authentication Protocols

- **Lightweight Directory Access Protocol (LDAP)** is a protocol for accessing and maintaining directory information services.
- LDAP servers store user accounts and authentication information in a hierarchical directory structure for organizational use.
- **Security Assertion Markup Language (SAML)** is an XML-based standard for exchanging authentication and authorization data between parties.
- **Open Authorization (OAuth)** enables third-party applications to obtain limited access to user accounts without sharing credentials.

Protocol	**Primary Use Case**
LDAP	Directory services within organizations
SAML	Enterprise SSO and cross-domain federation
OAuth	Delegated authorization for third-party applications
OpenID Connect	User authentication based on OAuth 2.0

Interoperability: Making Different Systems Work Together

- **Interoperability** refers to the ability of different IAM systems to work together seamlessly despite differences in design and implementation.
- Standards-based approaches ensure consistent interpretation of identity information across heterogeneous systems.
- **Attestation** provides verified claims about identity attributes that can be trusted across organizational boundaries.
- Modern IAM systems must balance proprietary features with compatibility with widely-adopted industry standards.

Key Interoperability Standards

- SCIM (System for Cross-domain Identity Management) for user provisioning
- JWT (JSON Web Tokens) for securely transmitting claims between parties
- X.509 certificates for public key infrastructure
- FIDO (Fast Identity Online) for passwordless authentication

Privileged Access: Managing the Keys to the Kingdom

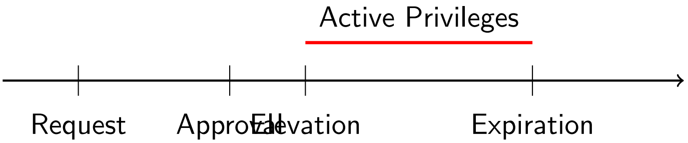
- **Privileged access** refers to elevated permissions that provide extensive control over critical systems and sensitive data.
- Privileged accounts represent the highest security risk because they can bypass normal security controls and make widespread changes.
- **Privileged Access Management (PAM)** includes special controls to secure, monitor, and audit privileged account usage.
- Effective PAM requires both technical solutions and operational practices like separation of duties and regular access reviews.

Example

Examples of privileged accounts include domain administrators, database administrators, root accounts on servers, emergency access accounts, and service accounts that run critical system processes.

Just-in-Time Permissions: Access When Needed

- **Just-in-Time (JIT) permissions** provide elevated access only when needed and only for the duration required.
- JIT permissions reduce the risk of privilege abuse by limiting the window of opportunity for malicious actions.
- Implementation typically involves a workflow where users request temporary privileges with justification and receive automatic expiration.
- This approach follows the principle of zero standing privileges, where no user permanently holds administrative rights.



Password Vaulting and Ephemeral Credentials: Temporary Access Solutions

- A **password vault** securely stores privileged account credentials and controls their usage through check-out procedures and automatic rotation.
- Password vaults eliminate the need for users to know actual passwords while still allowing controlled access to privileged accounts.
- **Ephemeral credentials** are temporary authentication secrets generated for a single session and discarded afterward.
- Cloud environments increasingly use ephemeral credentials to minimize the risk of long-lived access keys being compromised.

Security Benefit

With properly implemented password vaulting, even administrators cannot access privileged credentials directly, reducing insider threat risks and preventing password sharing among team members.

IAM Best Practices: Putting It All Together

- Implement the principle of least privilege by providing minimal access required for each job function.
- Use multifactor authentication for all accounts, especially those with privileged access.
- Automate provisioning and de-provisioning to ensure consistency and timeliness.
- Conduct regular access reviews to identify and correct inappropriate permissions.

Balancing Security and Usability

The most effective IAM implementations find the right balance between:

- Strong security controls without excessive user friction
- Centralized governance with appropriate delegation
- Standardized policies with flexibility for special cases
- Automated processes with human oversight

The Future of Identity and Access Management: Trends and Challenges

- The movement toward **Zero Trust Architecture** emphasizes continuous verification rather than implicit trust based on network location.
- **Artificial intelligence** is increasingly used to detect abnormal access patterns and provide risk-based authentication.
- **Decentralized identity** approaches using blockchain technology promise user control over personal data and credentials.
- Cloud and mobile computing continue to challenge traditional perimeter-based security models and drive IAM innovation.

Traditional IAM	**Future IAM**
Static permissions	Dynamic, contextual access
Password-centric	Passwordless authentication
Organization-controlled	User-controlled identity
Perimeter-focused	Zero Trust model