جامـــعـة المـــســتـقـبـل
AL MUSTAQBAL UNIVERSITY

قســــم الامــــــــــن الســـــــــــــــــــــيبرانــــــــــــــــــــــي

DEPARTMENT OF CYBER SECURITY

SUBJECT:

AUTHENTICATION AND ACCESS CONTROL

CLASS:

SECOND

LECTURER:

DR. SUHA ALHUSSIENY

LECTURE: (8)

ACCESS CONTROL – CHALLENGES, SOFTWARE, AND THE ACCESS CONTROL MATRIX

1. The Challenges of Access Control

Access control is one of the foundations of cybersecurity. At first glance, it may seem straightforward: identify a user, verify their identity, and allow them to access what they are supposed to access. In practice, however, implementing effective access control is one of the *most complex* and *error-prone* tasks in computer security. This complexity arises from organizational growth, diverse technology environments, user behavior, and constantly evolving threats.

1.1. Scale and Complexity of Modern Systems

Modern organizations do not operate on a small scale. Instead, they typically employ:

- Thousands of users (employees, contractors, students).
- Many systems and platforms (email servers, databases, cloud applications, internal apps).
- Distributed networks that span across countries or regions.
- Multiple identity sources and authentication methods.

As a result, access control must support a vast and dynamic ecosystem.

For example, a university's environment includes:

- Students with different academic roles.
- Professors with teaching and grading privileges.
- Administrative staff with financial and HR access.
- IT teams with superuser permissions.

Managing who gets what access — and making sure these permissions stay correct over time — is extremely challenging. If permissions are too broad, security is compromised. If they are too restrictive, users cannot do their jobs.

1.2. Constantly Changing Roles and Responsibilities

Users rarely keep the same role forever. They:

- Change departments
- Get promoted
- Switch projects
- Take temporary assignments
- Leave the organization entirely

Each role change requires permissions to be updated immediately. Otherwise, users may retain access to sensitive systems long after they should not, creating a high-risk situation known as privilege creep.

Example:

A financial analyst moves to the marketing department. If their financial database access is not revoked, they still hold the ability to view or manipulate highly sensitive financial information — even though they no longer need it.

Effective access control requires continuous monitoring to ensure permissions reflect a user's current responsibilities, not their history.

1.3. Insider Threats and Human Factors

One of the biggest misconceptions in cybersecurity is that attackers are always external hackers. In reality, many security breaches originate *inside* an organization — either intentionally or accidentally.

There are two forms:

1. Malicious insiders:

   Employees who purposely misuse access, often to steal data, sabotage the system, or commit fraud.

2. Unintentional insiders:

   Users who accidentally expose sensitive data due to weak passwords, misconfigurations, or negligence.

Because insiders already possess legitimate access, traditional defenses like firewalls and intrusion detection systems may not detect their actions. This makes strong access control, auditing, and monitoring essential.

1.4. Enforcing the Principle of Least Privilege

Least privilege means granting users only the exact amount of access needed to perform their tasks — *no more, no less*.

While simple in theory, determining the minimum required access is difficult in practice:

- Job roles may not be clearly defined.
- Permissions may overlap between departments.
- Some tasks may require temporary elevated rights.
- Legacy systems may not support fine-grained control.

If implemented incorrectly, least privilege can either cause:

- Over-permissioning → dangerous access
- Under-permissioning → blocked workflows and frustrated users

Thus, enforcing least privilege requires careful planning, ongoing auditing, and strong communication between IT and department managers.

1.5. Complexity of Cloud and Hybrid Environments

Modern organizations often use a combination of:

- On-premises servers
- Cloud infrastructure (e.g., AWS, Azure, Google Cloud)

- SaaS applications (e.g., Office 365, Salesforce)
- Mobile devices
- Remote access tools

Each environment uses different access models, permissions, and authentication methods.

Cloud platforms introduce:

- Hundreds of granular permission types
- Resource hierarchy structures
- Identity federation
- APIs that require their own authorization

A small misconfiguration — such as granting "public access" to a cloud storage bucket — can expose private data to the world.

1.6. Misconfigurations and Human Error

A large percentage of data breaches occur not due to advanced attacks, but because of simple mistakes, such as:

- Incorrectly configured firewalls
- Overly permissive cloud roles
- APIs left without authentication
- Shared accounts without passwords
- A public-facing database with no access protection

These mistakes are so common that Broken Access Control has ranked #1 in the OWASP Top 10 (2021 & 2023).

2. Access Control Software

Access control software refers to the tools, systems, and technologies that organizations use to define, apply, and monitor access policies. These tools form the backbone of organizational cybersecurity.

Access control software can be divided into two major categories:

Physical access control and **logical (digital) access control**.

2.1. Physical Access Control Software

Physical access systems limit entry to buildings, offices, or rooms. These systems use hardware and software to authenticate people in real-world spaces.

Typical components include:

- **Access cards / badges** (RFID, smart cards)
- **Biometric scanners** (fingerprint, retina, facial recognition)
- **Door controllers** and **electronic locks**
- **CCTV integration** for monitoring
- **Audit logs** that record all entry attempts

Such systems help organizations keep unauthorized individuals from entering sensitive environments.

Real-world example:

A data center uses:

- Card readers for general entry
- Biometric fingerprint scans to access server rooms
- Cameras to monitor high-security areas
- Logs to track every person who entered or attempted entry

This ensures only authorized personnel can access critical equipment.

2.2. Logical Access Control Software

Logical access controls protect digital systems. They ensure that only authenticated and authorized users can access applications, databases, or networks.

Key components include:

✔ Identity and Access Management (IAM)

IAM platforms manage:

- User identities
- Permissions
- Roles
- Authentication methods
- Access policies

Examples:

IAM systems automate:

- User onboarding
- Role assignments
- Password resets
- Permission reviews
- Offboarding (removing access when someone leaves)

✔ Single Sign-On (SSO)

SSO allows users to log in once and then access multiple systems without re-entering credentials.

This improves usability but requires strong security controls to prevent compromised accounts.

✔ Multi-Factor Authentication (MFA)

MFA significantly improves security by requiring two or more authentication methods, such as:

- Password + fingerprint
- Password + SMS code
- Password + authenticator app

This prevents attackers from entering even if they steal a password.

✔ Privileged Access Management (PAM)

PAM focuses on protecting **admin accounts**, which have high-level access.
Since admin accounts can control entire systems, they are prime targets for attackers.
PAM includes:

- Just-in-time access
- Session recording
- Password vaulting

✔ Directory Services (LDAP, Kerberos)

Directory services store identity information and authenticate users.
Kerberos (e.g., in Windows networks) prevents password exposure by using ticket-based authentication.

✔ Logging and Monitoring Tools

Effective access control requires visibility.

Monitoring tools track:

- Who accessed what
- Failed login attempts
- Suspicious activity
- Privilege misuse

This helps detect breaches early.

3. Access Control Matrix

The Access Control Matrix (ACM) is a foundational model that describes how permissions are structured within a system. It provides a conceptual way to think about access rights before implementing them using ACLs or capability lists.

3.1. What the Access Control Matrix Represents

An ACM is a table-like structure where:

- **Rows represent subjects** (users, processes, roles)
- **Columns represent objects** (files, databases, printers)
- **Cells contain permissions** (read, write, execute, delete)

This matrix provides a full overview of all access relationships within the system.

The ACM is crucial because it helps:

- Visualize who has access to which resources
- Identify security gaps
- Enforce consistent policies
- Translate abstract policies into concrete controls

3.2. Example of an Access Control Matrix

Imagine a simple environment:

Subject / Object	File A	File B	Printer	Database
Alice (Teacher)	Read	Write	Print	Read
Bob (Student)	Read	NoAccess	Print	NoAccess
Admin	All	All	All	All

This matrix clearly shows:

- The exact permissions each user has
- Which resources are protected
- How access varies across roles

This model is very useful during system design, auditing, and troubleshooting.

3.3. Implementations of the Access Control Matrix

Storing a full ACM in memory is inefficient for large systems, so two practical structures are used instead.

✔ Access Control Lists (ACLs)

ACLs store access permissions **by object**.

Example: ACL for File A

- Alice → read
- Bob → read
- Admin → all

Useful when managing permissions per file or resource.

✔ Capability Lists

Capabilities store permissions **by subject**, meaning each user keeps a list of what they are allowed to access.

Example: Alice's capability list

- File A → read
- File B → write
- Printer → print

Useful in distributed systems or object-capability models.

3.4. Why the Access Control Matrix Matters

The ACM helps ensure a system has:

- Clear boundaries
- Proper separation of duties
- No unintended access paths
- Strong auditing capability

It also forms the basis for modern models such as RBAC and ABAC.

H.W/

1. Which of the following best explains why access control is challenging in modern organizations?

2. What is privilege creep?

3. Which of the following is an example of a malicious insider?

4. Least privilege requires:

5. Which of the following environments increases access control complexity?

6. A common cause of access control failures is:

7. Which system restricts entry to buildings or rooms?

8. Which of the following is a component of physical access control?

9. Logical access control protects:

10. Identity and Access Management (IAM) systems manage:

11. Single Sign-On (SSO) enables users to:

12. Multi-Factor Authentication (MFA) improves security by:

13. Privileged Access Management (PAM) focuses on:

14. Which protocol uses ticket-based authentication?

15. Which tool helps detect suspicious access patterns?

16. The Access Control Matrix represents:

17. In an Access Control Matrix, columns represent:

18. Which structure stores permissions by object?

19. Capability lists store permissions based on:

20. Which access model helps identify unintended access paths?

21. Which of the following is an example of IAM automation?

22. A cloud storage bucket accidentally set to public access is an example of:

23. Which component is common in physical access systems?

24. Which access control failure ranked #1 in the OWASP Top 10?

25. Which users typically have the highest access rights?