Operating Systems chapter 6 Security

Linking Deadlock to Security

- **Deadlock** affects **Availability**, which is one of the main goals of **security**.
-
- **Security** is not only about stopping **hackers**; it also means keeping systems **stable, reliable, and usable**.
-
- **Example:**
  - If a malicious or poorly written program causes a deadlock, the system may freeze.
  - Students cannot use the computer.
  - **Availability** is lost.
  - Therefore, **security** **is affected**.

Deadlock and Security (Exam-Ready Version)

- One of the main goals of security is **Availability**.
-
- **Availability** means authorized users can access the system when they need it.
- Deadlock occurs when processes wait forever for resources.
-
- When deadlock happens, the system may freeze or stop responding.
-
- This prevents users (such as students) from using the system.
- Even if no data is stolen, **security is still affected** because **Availability** is lost.

Simple Real-World Analogy (Easy to Remember)

Imagine a **traffic deadlock at an intersection**

- Each car blocks the others.
- No car can move forward.
- The road becomes unusable.

Similarly:

- Processes block each other in a deadlock.
- The system becomes unusable.
- Users are denied access → **Availability is lost** → **Security is affected**.

Security-Related Deadlock Example in Windows

Deadlock Affecting Availability

Deadlock occurs when two or more processes are waiting indefinitely for resources held by each other. In an operating system like Windows, this situation can cause applications or system services to stop responding, making the system partially or fully unavailable to users.

From a **security** perspective, this is important because **Availability** is one of the **three core goals** of system security in the **CIA Triad** (**Confidentiality**, **Integrity**, and **Availability**). If users cannot access system resources when needed, **security** has been negatively affected—even if no data is **stolen** or **modified**.

Practical Windows Example

Consider a printing scenario in Windows:

- A **Word document** holds the file it wants to print and requests access to the printer.
- At the same time, another application (such as **Adobe PDF**) is also waiting for the same printer resource.
- Each program is waiting for the other to release the resource.
- As a result, neither program can continue.

This deadlock prevents the printer from being used, stops the user from completing their work, and leads to a **loss** of **Availability**. Therefore, **system security** is **affected** due to **reduced availability** of **resources**.

Security-Related Deadlock Example in Windows

Deadlock Due to Shared Files

Deadlock can occur when multiple programs attempt to access the same file simultaneously with conflicting access modes, such as reading and writing. If each program holds a lock on part of the file and waits for the other to release its lock, none of them can proceed.

From a security perspective, this situation affects both **Integrity** and **Availability**, which are key elements of the **CIA Triad**. When data cannot be accessed, saved, or updated correctly, the system fails to provide reliable and secure operation.

Practical Windows Example

Consider a shared database file in Windows:

- **Microsoft Excel** opens the database file for editing.
- **Microsoft Access** attempts to open the same file for writing at the same time.
- Each application waits for the other to release the file lock.
- This results in a deadlock situation.

Because of this deadlock:

- The user cannot save or modify the data.
- The system may freeze or become unresponsive.
- Data consistency may be at risk.

As a result, **Availability is reduced**, and **Integrity may be affected**, making this a clear example of how deadlock can **affect** system security.

Security-Related Deadlock Example in Windows

Deadlock as Part of a Malicious Attack (Denial of Service)

Deadlock can also be intentionally caused by malicious software as part of a **Denial of Service (DoS)** attack. In this case, the malware deliberately requests and holds multiple system resources, preventing other processes from accessing them. This can create deadlock situations and cause the operating system to become slow or completely unresponsive.

From a security perspective, this directly affects **Availability**, which is a core component of the **CIA Triad**. Even though no data is stolen or modified, users are unable to access the system or its services, meaning system security is **affected due to loss of availability**. This example highlights that security is not only about preventing data breaches, but also about ensuring that systems continue to operate normally.

Practical Example

Consider the following scenario in Windows:

- An unofficial or malicious program opens many large files at the same time.
- The program consumes excessive system resources and holds file and memory locks.
- Other applications are forced to wait for these resources.
- The system becomes unresponsive or freezes.

As a result, the user cannot use the system, system **availability is degraded**, and overall system security is affected.

Deadlock in Security Services

Deadlock is not limited to user applications; it can also occur within **security services** themselves. In Windows, services such as **Windows Defender** and **Windows Update** may attempt to access the same system files or resources at the same time. If each service holds a resource and waits for the other to release it, a deadlock situation can occur.

From a security perspective, this affects **Availability** and **system reliability**. When security services become stuck or cause the system to freeze, users may be unable to use the system, and essential security operations may fail to complete. This shows that security mechanisms must be carefully designed to avoid deadlock and resource conflicts.

Practical Example

Consider the following Windows scenario:

- **Windows Defender** starts scanning large system files.
- At the same time, **Windows Update** attempts to access or modify the same files.
- Each service waits for access to the resource held by the other.
- The system may slow down or become unresponsive.

As a result, system availability is **degraded**, and the system's protective functions are **affected**. This demonstrates that even security services must be designed with deadlock prevention in mind to maintain overall system security.

Summary: Deadlock & OS Security Link

Deadlock in operating systems can directly impact **security goals** by reducing system **Availability** and, in some cases, **Integrity**. The table below summarizes common scenarios in Windows:

Deadlock Scenario	Security Goal Affected	Example Summary
Printer / Word conflict	Availability	User cannot print because Word and another program (e.g., PDF) wait for the printer.
File sharing	Integrity &	Excel and Access both try to access the same

Deadlock Scenario	Security Goal Affected	Example Summary
conflict	Availability	database file → system freeze, data may not save.
Malicious program	Availability	Malicious program consumes resources → DoS-like deadlock → system becomes unresponsive.
Security service conflict	Availability	Windows Update and Windows Defender try to access the same files → system may freeze → protection and updates are delayed.

Key Takeaways

- **Deadlock affects system security** even if no data is stolen or altered.
- Most commonly, it impacts **Availability**, and sometimes **Integrity**.
- Security mechanisms themselves must be designed to **avoid deadlock**, ensuring the system remains usable and safe.
- Awareness of deadlock scenarios is essential for both **system stability** and **security planning**.

Denial of Service (DoS)

(رفض الخدمة)

Definition:

Denial of Service (DoS) is a type of attack where the goal is to **make a system, service, or network unavailable to its users**.

A **Denial of Service (DoS)** attack occurs when an attacker intentionally overloads or blocks a system so that legitimate users cannot access it.

Denial of Service (DoS) is an attack that prevents legitimate users from accessing a system by exhausting its resources, thereby affecting availability.

How It Relates to Security

- DoS attacks target **Availability**, not data.
- No need to steal or modify information.

- If users cannot use the system, **security is affected**.

Simple Explanation

In a DoS attack:

- The attacker sends too many requests **or**
- Consumes too many system resources (CPU, memory, files, locks)
- The system becomes slow, unresponsive, or crashes.

Example (Windows Context)

- A malicious program opens many files or requests many resources.
- The operating system cannot serve other applications.
- Windows freezes or stops responding.
- Legitimate users are denied access.