Exercise 3

1. **Broken Access Control (A01):**

Inadequate access controls that allow unauthorized users to access sensitive data or perform actions they shouldn't have permission for.

Prevention:

- Implement proper authentication mechanisms.
- Enforce strict access controls based on roles and permissions.
- Regularly audit and review access policies.
- Use session management best practices.

In the Equifax breach, attackers exploited a web application vulnerability to gain unauthorized access to sensitive personal information of 147 million individuals. Weak access controls allowed the attackers to navigate through the system and extract valuable data.

https://www.csoonline.com/article/567833/equifax-data-breach-faq-what-happened-who-was-affected-what-was-the-impact.html

Cryptographic Failures (A02):

Failures in cryptographic processes leading to the exposure of sensitive data or compromise of the entire system.

Prevention:

- Up-to-date cryptographic algorithms.
- Regularly updating cryptographic libraries.
- Employing proper key management practices.
- Conducting regular security audits of cryptographic implementations.

Heartbleed was a cryptographic vulnerability in the OpenSSL library. Attackers could exploit it to retrieve sensitive data from the memory of millions of web servers. This highlighted the critical importance of maintaining and updating cryptographic libraries.

https://heartbleed.com/

Injection (A03):

Improper handling of user inputs that allows attackers to inject malicious code, often leading to unauthorized access or data breaches.

Prevention:

- Implement parameterized queries to avoid SQL injection.
- Use input validation and sanitization.
- Employ web application firewalls (WAFs) to filter out malicious inputs.
- Educate developers on secure coding practices.

The SQL Slammer worm exploited a vulnerability in Microsoft SQL Server, using a SQL injection attack to spread rapidly across the internet. It caused widespread disruption by overloading servers and slowing down network traffic.

https://www.netscout.com/blog/asert/remembering-sql-slammer

Insecure Design (A04):

Design flaws in applications that expose vulnerabilities, emphasizing the need for threat modeling, secure design patterns, and reference architectures.

Prevention:

- Conduct thorough threat modeling during the design phase.
- Adopt secure design patterns.
- Follow industry best practices for secure architecture.
- Continuously assess and refine the design based on evolving threats.

Volkswagen used software designed to cheat emissions tests, leading to a scandal when it was discovered. This incident showcased the impact of insecure design choices in the automotive industry and the need for ethical considerations in software design.

https://www.bbc.com/news/business-34324772

Security Misconfiguration (A05):

Misconfigured settings that expose security weaknesses, making systems vulnerable to exploitation.

Prevention:

- Implement secure default settings.
- Regularly review and update configuration settings.
- Employ automated tools to check for misconfigurations.
- Follow security best practices for the specific technologies in use.

Numerous incidents have occurred where organizations misconfigured their AWS S3 buckets, exposing sensitive data to the public internet. These misconfigurations allowed unauthorized access to data, leading to data breaches.

https://www.securityhq.com/blog/security-101-compromised-aws-s3-buckets/

Vulnerable and Outdated Components (A06):

associated with using outdated or vulnerable third-party components, which can introduce security vulnerabilities into the application.

Prevention:

- Regularly update and patch third-party components.
- Monitor for security advisories related to used components.

- Use dependency scanning tools to identify vulnerabilities.
- Using only well-maintained and actively supported components is a good idea.

WannaCry exploited a vulnerability in the Windows operating system, leveraging a component with a known vulnerability (EternalBlue). The attack spread rapidly, affecting organizations worldwide that had not applied a critical security patch.

https://www.cloudflare.com/learning/security/ransomware/wannacry-ransomware/

Identification and Authentication Failures (A07):

Failures in user identification and authentication processes, potentially leading to unauthorized access.

Prevention:

- Implement multi-factor authentication (MFA).
- Enforce strong password policies.
- Regularly review and revoke unnecessary privileges.
- Use standardized authentication frameworks.

In the LinkedIn breach, attackers exploited weak password hashing practices. The breach exposed millions of user passwords, highlighting the importance of strong authentication mechanisms and secure password storage.

https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/2012-linkedin-breach-117-million-emails-and-passwords-stolen-not-6-5m

Software and Data Integrity Failures (A08):

Assumptions related to software updates, critical data, and CI/CD pipelines without proper verification of integrity.

Prevention:

- Verify software updates using digital signatures.
- Implement integrity checks for critical data.
- Ensure secure CI/CD pipeline practices.
- Regularly test and validate data integrity.

In the SolarWinds incident, attackers compromised the software supply chain, injecting malicious code into software updates. This compromised the integrity of the updates and allowed for a widespread espionage campaign.

https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

Security Logging and Monitoring Failures (A09):

Failures in logging and monitoring practices, impacting visibility, incident alerting, and forensics.

Prevention:

- Implement robust logging mechanisms.
- Regularly review and analyze logs.
- Set up alerts for suspicious activities.
- Conduct regular drills for incident response.

In the Target breach, security monitoring systems failed to detect and alert on suspicious activities. Attackers successfully exfiltrated customer data, emphasizing the importance of robust logging and monitoring practices.

https://redriver.com/security/target-data-breach#:~:text=What%20Happened%20During%20the%20Target,was%20one%20of%20the%20largest.

Server-Side Request Forgery (A10):

Manipulating a server to make requests to other resources on behalf of the attacker.

Prevention:

- Validate and sanitize user inputs.
- Employ proper input validation on URLs and parameters.
- Implement whitelisting of allowed resources.
- Restrict server-side access to external resources.

In the Capital One breach, a server-side request forgery vulnerability allowed an attacker to access and steal sensitive information from the bank's systems. The vulnerability enabled the attacker to make requests to internal resources.

https://www.capitalone.com/digital/facts2019/