AI with cyber security

Assignment1

TOP 5 OWASP Vulnerabilities

What Is the OWASP Top 10 and How Does It Work?

The OWASP Top 10 is a report, or “awareness document,” that outlines security concerns around web application security. It is regularly updated to ensure it constantly features the 10 most critical risks facing organizations. OWASP recommends all companies to incorporate the document’s findings into their corporate processes to ensure they minimize and mitigate the latest security risks.

The OWASP vulnerabilities report is formed on consensus from security experts all over the world. It ranks risks based on security defect frequency, vulnerability severity, and their potential impact. This provides developers and security professionals with insight into the most prominent risks and enables them to minimize the potential of the risks in their organizations’ security practices.

Top 5 OWASP Vulnerabilities

1. Injection
2. Broken authentication
3. Sensitive data exposure
4. XML external entities (XXE)
5. Broken access control

Injection

Injection attacks occur when untrusted data is injected through a form input or other types of data submission to web applications. A common type of injection attack is a Structured Query Language injection (SQLi), which occurs when cyber criminals inject SQL database code into an online form used for plaintext.

These types of attacks can be prevented by sanitizing and validating data submitted by users. Data validation ensures that suspicious data will be rejected, and data sanitization helps organizations clean data that looks suspicious. Database admins can also set controls that minimize how much information injection attacks can expose.

Broken Authentication

Authentication vulnerabilities can enable attackers to gain access to user accounts, including admin accounts that they could use to compromise and take full control of corporate systems.

Websites commonly suffer broken authentication, which typically occurs as a result of issues in the application's authentication mechanism. This includes bad session management, which can be exploited by attackers using brute-force techniques to guess or confirm user accounts and login credentials.

The OWASP Top 10 provides a list of broken authentication vulnerabilities, which include web applications that:

1. Permit attacks like credential stuffing
2. Permit weak or default passwords
3. Employ ineffective user credential and lost password processes
4. Are missing or use ineffective multi-factor authentication (MFA)
5. Expose session IDs in the Uniform Resource Locator (URL), do not rotate session IDs, and do not properly invalidate session IDs and authentication tokens after a period of inactivity

These vulnerabilities are typically caused by insecure software, which is often a result of inexperienced developers writing them, a lack of security testing, and rushed software releases.

Broken authentication vulnerabilities can be mitigated by deploying MFA methods, which offer greater certainty that a user is who they claim to be and prevent automated and brute-force attacks. These vulnerabilities can also be prevented by ensuring developers apply best practices to website security and are given an appropriate period of time to properly test codes before applications are put into production.

Other tactics include checking for weak passwords, ensuring users protect their accounts with strong, unique passwords, and using secure session managers.

Sensitive Data Exposure

Sensitive data exposure or data leakage is one of the most common forms of cyberattack. Sensitive data, like credit card information, medical details, Social Security numbers, and user passwords, can be exposed if a web application does not protect it effectively. Attackers who are able to access and steal this information can use it as part of wider attacks or sell it to third parties.

Protecting sensitive data is increasingly important given the stringent rules and punishments of data and privacy regulations, such as the European Union's General Data Protection Regulation (GDPR). To do so, organizations must be able to protect data at rest and data in transit between servers and web browsers.

Data on a website can be protected using a secure sockets layer (SSL) certificate, which establishes an encrypted link between a web browser and a server. It also protects the integrity of data when in transit between a server or firewall and the web browser. Sensitive data exposure can also be prevented by encrypting data through secure encryption processes, protecting stored passwords with strong hashing functions, and ensuring that strong, updated algorithms, keys, and protocols are in place.

XML External Entities (XXE)

XXE attacks target web applications that parse the Extensible Markup Language (XML). They occur when an XML input that contains a reference to an external entity, such as a hard drive, is processed by an XML parser with weak configuration. XML parsers are often vulnerable to an XXE by default, which means developers must remove the vulnerability manually.

The OWASP Top 10 states that XXE attacks typically target vulnerable XML processors, vulnerable code, dependencies, and integrations.

XXE attacks can be avoided by ensuring web applications accept less complex forms of data (such as JavaScript Object Notation (JSON) web tokens), patching XML parsers, or disabling the use of external entities. Organizations can also defend themselves against XXE attacks by deploying application programming interface (API) security gateways, virtual patching, and web application firewalls (WAFs).

Broken Access Control

Access control refers to the specific data, websites, databases, networks, or resources that users are allowed to visit or have access to. Broken access controls result in users having access to resources beyond what they require. This enables attackers to bypass access restrictions, gain unauthorized access to systems and sensitive data, and potentially gain access to admin and privileged user accounts.

The risk of broken access control can be reduced by deploying the concept of least privileged access, regularly auditing servers and websites, applying MFA, and removing inactive users and unnecessary services from servers. Organizations can also secure access controls by using authorization tokens when users log in to a web application and invalidating them after logout. Other recommendations include logging and reporting access failures and using rate limiting to minimize the damage caused by automated attacks.