SRI SANKANRS DEGREE COLLEGE KURNOOL

Y Prathibha

Long term intership

OWASP Vulnerabilities

OWASP

The Open Web Application Security Project, or OWASP, is an international non-profit organization dedicated to web application security. One of OWASP's core principles is that all of their materials be freely available and easily accessible on their website, making it possible for anyone to improve their own web application security. The materials they offer include documentation, tools, videos, and forums. Perhaps their best-known project is the OWASP Top 10.

OWASP Top10Vulnerabilities Overview

- **Injection**

  Injection attacks happen when untrusted data is sent to a code interpreter through a form input or some other data submission to a web application. For example, an attacker could enter SQL database code into a form that expects a plaintext username. If that form input is not properly secured, this would result in that SQL code being executed. This is known as an SQL injection attack.

- **Broken Authentication**

  Vulnerabilities in authentication (login) systems can give attackers access to user accounts and even the ability to compromise an entire system using an admin account. For example, an attacker can take a list containing thousands of known username/password combinations obtained during a data breach and use a script to try all those combinations on a login system to see if there are any that work.

- **Sensitive Data Exposure**

If web applications don't protect sensitive data such as financial information and passwords, attackers can gain access to that data and sellor utilize it for nefarious purposes. One popular method for stealing sensitive information is using an on-path attack.

- **XML External Entities (XEE)**

This is an attack against a web application that parses XML* input. This input can reference an external entity, attempting to exploit a vulnerability in the parser. An 'external entity' in this context refers to a storage unit, such as a hard drive. An XML parser can be duped into sending data to an unauthorized external entity, which can pass sensitive data directly to an attacker.

- **Broken Access Control**

Access control refers a system that controls access to information or functionality. Broken access controls allow attackers to bypass authorization and perform tasks as though they were privileged users such as administrators. For example a web application could allow a user to change which account they are logged in as simply by changing part of a url, without any other verification.

- **Security Misconfiguration**

Security misconfiguration is the most common vulnerability on the list, and is often the result of using default configurations or displaying excessively verbose errors. For instance, an application could show a user overly-descriptive errors which may reveal vulnerabilities in the application. This can be mitigated by removing any unused features in the code and ensuring that error messages are more general.

- **Cross-Site Scripting**

Cross-site scripting vulnerabilities occur when web applications allow users to add custom code into a url path or onto a website that will be seen by other users. This vulnerability can be exploited to run malicious JavaScript code on a victim’s browser.

- **Insecure Deserialization**

This threat targets the many web applications which frequently serialize and deserialize data. Serialization means taking objects from the application code and converting them into a format that can be used for another purpose, such as storing the data to disk or streaming it.

- **Using Components With Known Vulnerabilities**

Many modern web developers use components such as libraries and frameworks in their web applications. These components are pieces of software that help developers avoid redundant work and provide needed functionality; common example include front-end frameworks like React and smaller libraries that used to add share icons or a/b testing.

- **Insufficient Logging And Monitoring**

Many web applications are not taking enough steps to detect data breaches. The average discovery time for a breach is around 200 days after it has happened. This gives attackers a lot of time to cause damage before there is any response.

OWASP Vulnerability Identification Report for Altro Mutual

1. Overview:

Altro Mutual's website was analyzed for potential vulnerabilities based on the OWASP Top 10 list. The following vulnerabilities were identified along with their potential impact on Altro Mutual's business operations and users.

2. Vulnerability Findings:

a. SQL Injection (Injection Attacks):

Description: The website is vulnerable to SQL injection attacks, which allow attackers to execute arbitrary SQL commands on the database.

Impact: Attackers can gain unauthorized access to sensitive information stored in the database, such as user credentials, financial data, or personal information.

Recommendation: Implement parameterized queries or prepared statements to prevent SQL injection. Conduct regular security assessments and code reviews to identify and mitigate vulnerabilities.

b. Cross-Site Scripting (XSS):

Description: The website is vulnerable to cross-site scripting attacks, enabling attackers to inject malicious scripts into web pages viewed by other users.

Impact: Attackers can steal session cookies, redirect users to malicious websites, or perform actions on behalf of the user without their consent.

Recommendation: Implement input validation and output encoding to sanitize user input and prevent XSS attacks. Utilize content security policy (CSP) headers to mitigate the impact of successful XSS attacks.

c. Broken Authentication:

Description: The website exhibits weaknesses in its authentication mechanisms, allowing attackers to bypass authentication or hijack user sessions.

Impact: Attackers can gain unauthorized access to user accounts, compromise sensitive data, or perform fraudulent transactions.

Recommendation: Implement secure authentication mechanisms such as multi-factor authentication (MFA), strong password policies, and session management controls. Regularly monitor login activity for suspicious behavior.

d. Sensitive Data Exposure:

Description: The website exposes sensitive information such as passwords, credit card numbers, or personal details without adequate protection.

Impact: Attackers can intercept and steal sensitive data transmitted over insecure channels, leading to identity theft, financial fraud, or privacy breaches.

Recommendation: Encrypt sensitive data both at rest and in transit using strong cryptographic algorithms. Implement secure communication protocols such as HTTPS to protect data in transit.

e. XML External Entities (XXE):

Description: The website processes XML input from untrusted sources without proper validation, making it vulnerable to XXE attacks.

Impact: Attackers can exploit XXE vulnerabilities to read sensitive files, perform server-side request forgery (SSRF), or execute arbitrary code on the server.

Recommendation: Disable XML external entity (XXE) processing or use whitelists to restrict the entities allowed in XML input. Update XML parsers to the latest versions that provide protection against XXE attacks.

3. Mitigation Strategy Proposal:

The mitigation strategy should prioritize addressing high-risk vulnerabilities identified in the vulnerability identification report. This includes implementing secure coding practices, conducting regular security assessments, and keeping software dependencies up-to-date to minimize the attack surface.

4. Conclusion:

Addressing the identified vulnerabilities is critical to enhancing the security posture of Altro Mutual's website and protecting sensitive information from unauthorized access or manipulation by malicious actors. By implementing robust security measures and staying vigilant against emerging threats, Altro Mutual can mitigate the risk of security breaches and maintain the trust of its customers.