OWASP SECURITY  
REPORT

December 11, 2021

ANA MARIA NITU

Contents

1. What is OWASP? ……………..………………………..….2
2. Injection………………………………...……………………2
3. Authentication……………………….…………………………3
4. Sensitive Data Exposure…………………………………….3
5. XXE…………………..…………………………………..3
6. Broken Access Control.……………………………………3
7. Security Misconfiguration…………………………………4
8. XSS…………………………………………………..……..4
9. Insecure Deserialization…………………………………..…4
10. Using Components with Known Vulnerabilities……4
11. Insufficient Logging/Monitoring……..4

# what is owasp?

“The Open Web Application Security Project, or OWASP, is an international non-profit organization dedicated to web security application 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.”

# A1:2017 - INJECTION

Injection flaws within a software application appear when the attacker can send hostile data to the interpreter

In my project, I use a safe API, making use of the Object Relational Mapping Tools. Whenever the user needs to input data through the website, it is validated.

# A2:2017 – BROKEN AUTHENTICATION

This refers to “brute force” attacks, when the attacker wants to gain access to an existing account on the website by using a huge amount of login credentials combinations.

In the application, this issue is handled by storing a join date (when they created the account) and last login date. If the user tries to log in with the incorrect credentials for more than 5 times, the account is automatically locked and only an admin can unlock it.

# A3:2017 – SENSITIVE DATA EXPOSURE

This refers to exposing key/sensitive information to attackers by not encrypting the important data correctly. In order to prevent this, I use BCryptPasswordEncoder.

# A4:2017 – XML EXTERNAL ENTITIES (XXE)

XXE Injection attacks occur when an XML input containing references to an external entity is processed by a weakly configured XML parser. The attacker takes advantage of it by embedding malicious inline DOCTYPE definition in the XML data.

On my website, I make use of JSON format data (which is less complex) and avoid serialization of sensitive data.

# A5:2017 – BROKEN ACCESS CONTROL

It refers to attacks when the hackers use different tools to exploit unfunctional access control configuration. In my application, there is the log access control for repeated failures and I make sure the end user can not modify the access control.

# A6:2017 – SECURITY MISCONFIGURATION

Security misconfigurations are security controls that are inaccurately configured or left insecure, putting your systems and data at risk. Attackers can try to access unused pages or unprotected files or directories. In my application, I tried to keep a minimal approach, not having unnecessary or unused features, frameworks or components.

# A7:2017 – CROSS-SITE SCRIPTIONG (XSS)

It works by manipulating a vulnerable website so that it returns malicious JavaScript to users. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.

For the individual project, as a frontend framework, I user ReactJS, which automatically escapes XSS by design.

# A8:2017 – INSECURE DESERIALIZATION

When a website deserializes user-controllable data, it is known as insecure deserialization. This gives an attacker the ability to alter serialized objects in order to inject malicious data into the application code.

As long as my application is concerned, I monitor deserialization and alert if a user deserializes constantly.

# A9:2017 – USING COMPONENTS WITH KNOWN VULNERABILITIES

The attackers can effortlessly use automated scan tools or perform manual analysis of the application to find the flaws. If they are able to find that the particular application uses a component which has previously been found vulnerable, they can take advantage and attack the software application.

In my application, all the components, packages, dependencies, libraries are from official sources and I do not have unnecessary components.

# A10:2017 – INSUFFICIENT LOGGING & MONITORING

This occurs when actions are not properly logged, resulting in a lack of monitoring and knowledge of what is going on within the application. In the application, there is a logger in most of the classes with more complex functionality (service, controller).