EXPLORING OWASP & WEB SECURITY RISKS

STUDENT'S NAME

YOAGHINI (CI230154)

SUJJASHRI (CI230148)

NUR HUSNINA AINA (CI230076)

NURUL EREYNA NARESHA (CI230127)

NURHUSNA IRDINA (CI230072)

DR ENCIK KHAIRUL AMIN

UNIVERSITY TUN HUSSIEN ONN MALAYSIA

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FAKULTI SAINS KOMPUTER & TEKNOLOGI MAKLUMAT (FSKTM)

LAB 01

SEM II 2024/2025

COURSE NAME	**:**	**WEB SECURITY**
COURSE CODE	**:**	**BIS20303**
LECTURER	**:**	**ENCIK KHAIRUL AMIN BIN MOHAMAD SUKRI**
STUDENT NAME & MATRIC NO	**:**	**YOAGHINI A/P LETCHMANAN(CI230154)**
		SUJJASHRI A/P A SUBRAMANIAM(CI230148)
		NUR HUSNINA AINA BINTI MOHD ISKANDAR(CI230076)
		NURUL EREYNA NARESHA BINTI ROZAK(CI230127)
		NURHUSNA IRDINA BINTI HAMZAH(CI230072)

1. Visit the OWASP official website: https://owasp.org/.

2. Provide a brief introduction to OWASP by answering:

a) What is OWASP, and why is it important?

The nonprofit organization Open Worldwide Application Security Project (OWASP) is well-known around the world and dedicated to enhancing software security. A vital part of the cybersecurity community, OWASP was founded on December 1, 2001, and became a U.S. nonprofit foundation in 2004. Its free, open-source tools, resources, documentation, and community assistance assist enterprises in creating safe applications.

OWASP operates more than 250 local chapters throughout the globe. Its goal is to become the open, worldwide community that supports safe software via cooperation, education, and tools. A major gathering place for those working in software development and cybersecurity, OWASP organizes industry-leading conferences for education and training, funds worthwhile initiatives, and offers thorough security standards.

b) What is the OWASP Top 10, and how does it help organizations?

The Open Web Application Security Project (OWASP) publishes a well-known awareness document called the OWASP Top 10. The ten most important security threats to online applications are highlighted. The 2021 version is the most recent release prior to the 2025 version, and the list is updated on a regular basis. The list classifies and ranks the most prevalent flaws and problems that businesses and developers need to fix in order to increase the security of online applications.

Risks that have been repeatedly shown to jeopardize the security of applications across sectors, including Broken Access Control, Cryptographic Failures, Injection, and Security Misconfiguration, are included in the OWASP Top 10. It explains how these hazards could jeopardize private information or vital systems and provides suggestions for reducing them.

How does it help organizations?

1. **Increasing Awareness:** By outlining the most important security threats, the OWASP Top 10 makes sure that developers and security teams concentrate on the most serious weaknesses.

2. **Effort Prioritization:** The Top 10 risks, which are ranked by severity, assist businesses in concentrating on high-impact threats first, such cryptographic failures and broken access control, in order to successfully enhance security.

c) What could have prevented the attack?

1. **Input Validation:** Injection attacks may be avoided by making sure that all user input is verified.
2. **Output Encoding:** Cross-site scripting (XSS) attacks may be avoided by properly encoding outputs.
3. **Authentication and Password Management:** Stop unwanted access by putting in place robust authentication procedures and safe password storage techniques. This involves hashing passwords using a robust method.

3. Chosen Security Risk From the OWASP Top 10 is <u>Broken Access Control</u>

4. Detailed Analysis Of The Selected Risk:-

a) Risk Description: What does this vulnerability mean?

Broken Access Control is known as a vulnerability that it fails to properly enforce restrictions on what authenticated users should do. By taking advantage of ineffective access control procedures, this vulnerability allows the attackers to obtain the sensitive information or data without authorization. In essence, it happens when an application fails to sufficiently confirm if a user is authorized to access a specific resource or carry out an action.This attack typically lead to information disclosure due to the failure to enforce authentication restriction that affects an organization operation and reputations as well.

b) How It Happens: How do attackers exploit this weakness?

The broken access control attack can be exploited or performed by the attacker by using few ways by enabling the attackers to obtain the unauthorized access towards the restricted data without having the privileges to access to the sensitive informations.The first technique would be by URL Manipulation where the attacker used to modify or alter the URL parameters that gives the privilege to gain access to the highly restricted sensitive resources.

The next would be,Cross Site Request Forgery where an attacker will send a malicious link to make an user falls into their trap by tricking the user do unintended actions.This happens when the users click the malicious link that shared by the attacker while they’re authenticated

it would easily reveal some sensitive data and perform actions such as submitting forms or making online transaction without the user's consent.

c) Prevention: How can developers protect web applications from this risk?

- **Implement Role-Based Access Control (RBAC)**: Implementing RBAC is known as one of the most effective methods that protect from this vulnerability attack where it ensures that only the respective authorized users with the given privileges can access specific data or perform certain action.
- **Deny by Default**: Implementing the deny all approach is one of the easiest way that could perform by all the users.This method ensures that all resources are not given access unless explicitly permitted and grant access only to the respective users, which would be extremely helpful in mitigating the risk of unauthorized access.

5. Research a real-world web security breach that reflects your selected risk.

Real-World Web Security Breach: Facebook – Business Manager Bug

Year :2018

Breach name: Facebook business manager bug

Type of risk: Broken Access

In 2018, Facebook provided a good real-world example of this. There was a serious glitch in Facebook's business manager function, which allows companies to control their pages.

A security researcher discovered that you might send a request to Facebook to assign yourself as an administrator to any Facebook page. You did not require the owner's permission or access, all you had to do was modify the request by modifying the page ID or user ID in the data being sent. Facebook did not properly verify whether the individual making the request was authorized to make those modifications. In simple terms, it trusted the user too much and didn't double-check things on the server.

This is typical broken access control. The software allowed users to perform functions that they had no right to accomplish. If someone had exploited this, they could have taken over famous pages like public figures, brands, and even government pages. That implies they might post anything, fraud followers, or damage reputations.

6. Answer the following in relation to the chosen risk:

a) What happened in the breach?

In 2018, a vulnerability in Facebook's Business Manager tool was uncovered, allowing users to acquire unauthorized admin access to any Facebook Page by simply changing request data such as the Page ID or User ID. Facebook's system failed to authenticate whether the user had the authority to make such changes, allowing attackers to take over high-profile pages. This flaw may have had major implications, such as distributing false information or harming the reputations of public persons and organisations.

b) How does the breach relate to the selected OWASP risk?

This issue is a clear reflection of what OWASP refers to as Broken Access Control, in which the application fails to restrict user actions based on their roles or privileges. In this example, Facebook failed to adequately enforce server-side access checks, allowing unauthorised users to perform administrative functions. The system's failure to validate the legitimacy of the request indicates a breakdown in access control, allowing attackers to exploit the system and do things they were never authorised to undertake.

c) What could have prevented the attack?

- Server-Side Authorization Checks - Always verify instead of relying just on user requests.
- Input Validation - Reject updated request data, such as altered Page IDs or User IDs.
- Role-Based Access Control (RBAC) - Restricts the ability to assign new admins to users with appropriate roles, such as existing administrators.
- The least Privilege Principle - Users should only have minimum access needed for their role.
- Monitoring - Detect or prevent unauthorized admin activities.