BIS 20303: WEB SECURITY

SEMESTER II 2024/2025

SECTION 1

LAB 1_GROUP 6

LECTURE NAME:

DR KHAIRUL AMIN BIN MOHAMAD SUKRI

GROUP MEMBERS

NAME	MATRIC NUMBER
VAIGISH RAJ A/L JAYASILAN	DI220024
ARULVADIVELAN A/L SUBRAMANIAM	DI220087
MANOJJ KUMARR NAIDU A/L GANESAN	DI220043
ABU YAZID BIN AB KADIR	DI220081
THAMILINIYAN A/L BALA SANTARAN	AI210201

BIS 20303 Web Security

Lab 1: Exploring OWASP & Web Security Risks

Lab Type: Group Research & Online Publication

Objective

- Understand the role of OWASP and its importance in web security.
- Explore the OWASP Top 10 security risks and their impact on web applications.
- Analyze real-world case studies of web security breaches related to OWASP vulnerabilities.
- Develop research and technical writing skills by publishing findings online.
- Promote academic integrity through originality and responsible research.

Group Formation & Submission Format

- Students must form groups of five (5).
- The final work must be published online using any publicly accessible platform (e.g., Blogspot, GitHub Pages, Medium, or a personal website).
- Each group must submit one consolidated document (not separate files) containing:
    1. Publication link to their online report.
    2. PDF copy of their full report (2-3 pages).
    3. Turnitin report, ensuring:
        - Plagiarism score is below 20%.
        - AI detection score is below 10%.
- Submissions not meeting these requirements will be rejected.

Task 1: OWASP Top 10 Risk Analysis & Case Study

Instructions:

1. Visit the OWASP official website: https://owasp.org/.
2. Provide a brief introduction to OWASP by answering:
    - What is OWASP, and why is it important?

      - The Open Web Application Security Project (OWASP) is a non-profit organization created in 2001 to assist website owners and security professionals in protecting web applications from cyber threats. OWASP has 32,000 volunteers worldwide who conduct security evaluations and research. Furthermore, OWASP is important because it offers the best practices, tools, standards, and learning guides that contribute to improving the security of applications and highlights the most critical security concerns that help organizations to safeguard their applications more effectively.

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

      - Top 10 OWASP are broken access control, cryptographic failures, injection, insecure design, security misconfiguration, vulnerable and outdated components, identification and authentication failures, software and data integrity failures, security logging and monitoring failures, and server-side request forgery. Thus, it helps organizations to focus on areas of most critical web application security risks and prioritize security efforts, improve development practices, and make developers and security staff more aware of common vulnerabilities. By addressing these risks, organizations not only reduce the chance of security incidents but also improve trust with users and, furthermore, keep sensitive data secure.

    - How often is the OWASP Top 10 updated, and why does it change?

      - The OWASP Top 10 is updated every three to four years to reflect the changing cybersecurity landscape. It evolves as new threats emerge, attack strategies advance, and security trends move over time. The improvements are based on real-world data, industry research, and community feedback to ensure that organizations prioritize the most essential and relevant security concerns. By updating the list, OWASP helps organizations stay ahead of potential attacks and continuously improve their security processes.

3. Select one security risk from the OWASP Top 10.

   - One of OWASP Top 10 security vulnerabilities is Injection. Following is a general explanation:

   When a malicious user injects malicious data into an app, such as an application or website, with the main goal of interrupting the program's functionality, this is known as the injection method. They might be able to access or steal information, alter the data or even have full control of the system as a result. Moreover, another example is hackers insert malicious code into a form or URL in order to trick the system into executing instructions it shouldn't. In order to avoid this, all inputs need to be checked and sanitised so that only safe information may get into the system.

4. Provide a detailed analysis of the selected risk, including:
   - Risk Description: What does this vulnerability mean?

   - Injection is a security concern where an attacker sends untrusted data to an interpreter (e.g., SQL engine or command shell) as part of a command or query. Malicious data used by the interpreter can lead to unauthorised access, data leakage, loss, and system compromise. SQL Injection is a frequent sort of attack where malicious SQL code is inserted into a query to manipulate the database.

   - How It Happens: How do attackers exploit this weakness?

   - Injection vulnerabilities happens when an application fails to sanitise or validate the user input before submitting it to an interpreter such as SQL database, command shell, or XML processor. This will easily allow attackers to inject malicious code into the input fields that affects the system's original behaviour. For like example, in SQL injection, an attacker are able to use special characters such as 'OR '1'='1 to overcome login authentication, which always evaluates to true. Similar attacks can be carried out against systems that use NoSQL, LDAP, or XML, all of which result from the application's inability to distinguish between user input and executable code.

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

   - To defend against injection attacks, developers should use a layered security approach that includes input validation, sanitization, and the use of parameterized queries or prepared statements. All user inputs must be treated as potentially harmful and validated on the server side using whitelists. Access privileges should follow the least privilege principle. Additional protections include web application firewalls, content security policies, regular software updates, security audits, secure coding, proper error handling, and ongoing developer training.

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

- In 2014, a cyber group called the "Guardians of Peace" hacked Sony Pictures and retrieved their employees' personal data, confidential internal conversations for Sony Pictures, and unreleased movie files. Their methods consisted of using SQL injection attacks on the company's web applications. The consequences unveiled severe redundancy in results, finances, and reputation, as well as a major loss of confidential information like personal details of employees alongside sensitive emails. This show centers on SQL injection attacks and web application focus, alongside the importance of having protective measures such as sanitized commands to avoid these situations.

6. Answer the following in relation to the chosen risk:
    - What happened in the breach?

      - The 2014 breach of Sony Pictures Entertainment was accompanied by an attack from the Guardians of Peace who managed to breach the company's internal network. During the course of the attack, the company's web applications were targeted due to an open SQL injection vulnerability. The attackers used the gap to run malicious SQL statements that bypassed authentication checks, allowing them to access backend databases and internal systems. With this access, they were able to obtain enormous amounts of sensitive material, such as employee records, private communications, and unreleased films. The hack was widely disclosed, and Sony suffered significant financial losses and brand harm as a result of the breach of confidential data intended for private inter-corporate discussions.

    - How does the breach relate to the selected OWASP risk?

      - The breach at Sony Pictures Entertainment directly maps to the Injection risk of the OWASP Top 10 more precisely SQL Injection. The Sony-attacks hackers gained access to Sony's sensitive data by tampering with their web applications. They added login bypass to SQL commands which gave them access to a variety of sensitive information including staff records and confidential emails. The breach underscores the failure to sanitize user input and also demonstrates that SQL logic needs to be separated from data, a fundamental requirement for eliminating injection attacks as listed in OWASP 10.

    - What could have prevented the attack?

      - Basic safety measures may have been put in place to avoid the Sony Pictures Entertainment breach. The risk of SQL Injection attacks may have been considerably lowered by treating user input as data and use prepared statements or parameterised searches. Data validation and sanitisation would ensure that the application only accepted legitimate and intended inputs. The SQL Injection vulnerability would have been found before it could be exploited thanks to routine penetration testing and security audits.

References

1. Veracode. (2025, April 2). OWASP Top 10 vulnerabilities | VeraCode. https://www.veracode.com/security/owasp-top-10/

2. What is the OWASP Top 10 and how does it work? | Black Duck. (n.d.). https://www.blackduck.com/glossary/what-is-owasp-top-10.html

3. Radware. (n.d.). SQL Injection: Examples, real life attacks & 9 defensive measures | Radware. https://www.radware.com/cyberpedia/application-security/sql-injection/#:~:text=7%2DEleven%20breach%20(2007)%3A,130%20million%20credit%20card%20numbers.

Expected Output:

- A group research report (2-3 pages total, including the OWASP risk analysis and case study).
- The report must be well-structured and include references to credible sources.

Publication & Submission Instructions

1. Publish the final report online using any public platform (e.g., Blogspot, Medium, GitHub Pages, or any personal website).
2. The format of publication is :

   Suitable Title

   Name of group members, Lecturer name without title and "bin" or "binti"

   Example is

3. Each group must submit one document (PDF format) containing:
   - The publication link to their online report.
   - A full PDF copy of their report.
   - Turnitin report ensuring:
     - Plagiarism score is below 20%.
     - AI detection score is below 10%.
4. Submission must be made via Author.
5. Late submissions will not be entertained. Due date is 12 April 2025.

Publication link (GitHub): https://github.com/MANOJJKUMARR/Exploring-OWASP-Web-Security-Risks_LAB1_SECTION1_GROUP6.git