**1. Introduction**

● Project Title:  
OWASP Sentinel – A Simple Web Vulnerability Scanner

● Team Name:  
Bit BY Bit

● Problem Statement:  
Web applications are vulnerable to a wide range of attacks, many of which stem from commonly known issues like Broken Access Control, Injections, and insecure configurations. Developers often miss these vulnerabilities, leading to severe security breaches.

● Proposed Solution:  
OWASP Sentinel is a lightweight web vulnerability scanner that identifies OWASP Top 10 security risks by analyzing HTML and JavaScript code snippets. It uses open-source libraries and custom logic to help developers proactively address vulnerabilities.

● Expected Impact:  
The tool educates and empowers developers to write secure code by identifying and resolving vulnerabilities early in the development process. This contributes to safer applications and a more secure digital environment.

**2. Abstract**

OWASP Sentinel is a lightweight, student-friendly web vulnerability scanner designed to detect OWASP Top 10 security flaws such as Cross-Site Scripting (XSS), SQL Injection, Broken Access Control, and insecure CORS policies. It focuses on analyzing HTML and JavaScript code snippets using open-source tools like Bandit, js-xss, and DOMPurify. This solution empowers developers to identify and fix vulnerabilities at the code level, promoting secure development practices. By offering real-time feedback and educational suggestions, OWASP Sentinel not only helps secure applications but also fosters cybersecurity awareness in early-stage developers. The tool is scalable, simple to use, and geared towards both education and prevention.

**3. Solution Details**

**Technical Architecture:**

* Frontend: JavaScript + HTML interface for code input and feedback display.
* Backend: Python (Flask) API to handle code analysis.
* Security Tools: Bandit for Python scanning; DOMPurify and js-xss for HTML/JS sanitization analysis.
* Detection Engine: Custom rules and pattern matching for identifying OWASP Top 10 vulnerabilities.

**Key Features:**

* Detects XSS, SQL Injection, and Command Injection.
* Flags broken access control and insecure CORS settings.
* Highlights insecure deserialization and header misconfigurations.
* Offers secure coding suggestions alongside findings.
* Simple UI for fast copy-paste scanning and results.

**Implementation Details:**

* Designed static analysis parsers for code input.
* Integrated Bandit and JavaScript sanitization libraries.
* Developed vulnerability flagging logic using regex and keyword detection.
* Built a simple responsive frontend for real-time interaction.

**Testing and Validation:**

* Ran tests using known insecure code samples.
* Validated detection accuracy with manual review and output logs.
* Included screenshots/mockups of detected issues and fix suggestions.

**Scalability and Future Enhancements:**

* Extend to scan URLs and uploaded files.
* Add support for additional languages (PHP, Node.js).
* Enable integration with GitHub and CI/CD pipelines.
* Plan to offer issue severity grading and exportable reports.

**Review of Existing Solutions**

Several tools exist in the web security domain, including:

* OWASP ZAP (Zed Attack Proxy): A powerful dynamic scanner used for penetration testing web apps. While comprehensive, it can be complex for beginners.
* Burp Suite: A widely-used tool in professional penetration testing, but has a steep learning curve and is not fully open-source.
* SonarQube + Plugins: Effective for code quality analysis with some security rules but lacks focus on OWASP Top 10.
* Bandit: Focuses on Python security analysis, which OWASP Sentinel integrates.
* DOMPurify / js-xss: Open-source tools to sanitize JavaScript/HTML input to prevent XSS attacks.

**Novelty:**

* Student-focused
* Works on code snippets
* Educational with secure coding suggestions
* Simple and lightweight

1. **Team**

**Team Members:**

Aarna Tomar – Developer, Analyst, and Designer

**Team Experience:**

* Pursuing B.Sc. Computer Science at Sharda University.
* Skilled in Python programming, web development, and secure coding practices.
* Passionate about cybersecurity, with experience in academic security research and tool development.

**Mentorship:**

Chetan – Mentor and guide throughout the project development phase.

1. **Conclusion**

**Summary:**  
OWASP Sentinel is a simple, powerful tool that detects common vulnerabilities as defined by OWASP. It helps developers understand security risks and fix them before deployment.

**Call to Action:**  
Adopt OWASP Sentinel in your development cycle and be a part of the movement toward a more secure web!

**Thank You:**  
Thank you for your time and support. Special thanks to my mentor, Chetan, for his guidance throughout this project.

**5. Research References:**

* OWASP Top 10 Project: <https://owasp.org/www-project-top-ten/>
* ZAP: <https://www.zaproxy.org/>
* Burp Suite: <https://portswigger.net/burp>
* Bandit GitHub: <https://github.com/PyCQA/bandit>
* DOMPurify GitHub: https://github.com/cure53/DOMPurify