

Secure Shop

Cybersecurity

The Problem

E-commerce web applications are frequent targets for cyberattacks because they handle sensitive user information and financial transactions. Security weaknesses such as missing security controls, weak session protection, and improper input validation can increase the risk of unauthorized access, data exposure, and financial loss. Secure Shop requires a structured cybersecurity assessment to identify vulnerabilities, evaluate risks, and improve the overall security posture of the application.

- E-commerce applications store valuable user and transaction data, making them attractive targets.
- Missing security protections can increase an application's attack surface.
- Regular vulnerability assessments help identify and reduce security risks before they are exploited.
- Security testing provides organizations with actionable recommendations to strengthen defenses.

Our Solution

Secure Shop is a cybersecurity risk assessment project that evaluates the security posture of a web-based e-commerce application using OWASP methodologies and OWASP ZAP. The solution identifies vulnerabilities through passive and active security testing, analyzes associated risks, and provides recommendations to strengthen application security. By applying industry-standard security practices, Secure Shop helps detect weaknesses, reduce potential attack risks, and improve the overall protection of user data and application functionality.

OWASP-Based Security Assessment
Automated Vulnerability Scanning with OWASP ZAP
Risk Analysis and Security Improvement
Recommendations

Architecture & How It Works

Product Catalog



Wireless Mouse

Ergonomic 2.4G wireless mouse

Price: \$19.99

Add to Cart



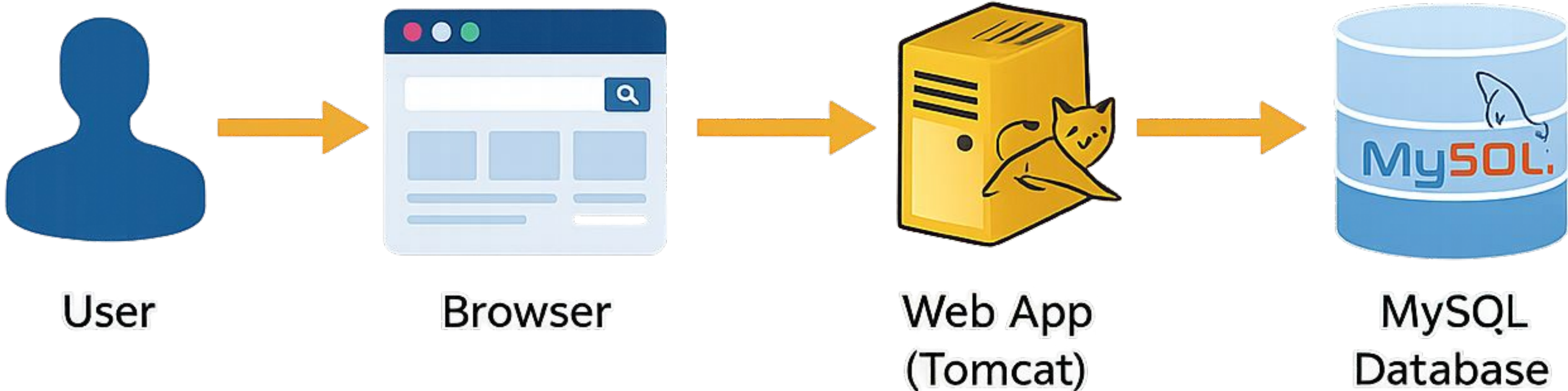
Mechanical Keyboard

RGB backlit, blue switches

Price: \$59.99

Add to Cart

Secure Shop is a web-based e-commerce app where users can register, log in, browse products, and manage a cart. It uses a client-server model, with a web interface for users and a backend database for storing credentials and transactions.



Key Features & Results

High-Risk Finding: SQL Injection

ZAP detected possible SQL query manipulation in the cart action parameter.

ZAP RISK: HIGH

Finding Snapshot

- Affected URL: /cart
- Parameter: action
- Boolean SQL logic payload was accepted
- Threatens database confidentiality and integrity

Primary Mitigations

- Use PreparedStatement / parameterized queries
- Validate action values with a server-side allowlist
- Apply least-privilege database permissions
- Log suspicious parameter behavior

Presentation Takeaway

- This is the highest-priority database finding and should be remediated first.

SQL Injection

URL: http://localhost:8080/cart

Risk: **High**

Confidence: Medium

Parameter: action

Attack: http://localhost:8080/cart?action=1;--

Evidence: CVE ID: 85

Request: http://localhost:8080/cart?action=1;--

Input (Order Form Query):

Description: SQL Injection may be possible.

Solution:

The error results were successfully manipulated using the boolean condition 'and(1=1)'. The original data was returned for the original parameter.

Details:

Do not trust client side input, even if there is client side validation in place. In general, type check all data on the server side.

Pre application issues: JDBC, use PreparedStatement or CallableStatement, with parameters passed by "I".

Reference: https://owasp.org/www-community/attacks/sql_injection

High-Risk Finding: Reflected XSS

ZAP detected reflected script input in the checkout name parameter.

ZAP RISK: HIGH

Finding Snapshot

- Affected URL: /checkout
- Parameter: name
- Script payload reflected in response
- Could allow browser-side code execution

Primary Mitigations

- Encode output before rendering user input
- Validate and sanitize checkout form fields
- Add Content-Security-Policy as defense-in-depth
- Avoid reflecting raw request parameters

Cross Site Scripting (Reflected)

URL: http://localhost:8080/checkout

Risk: **High**

Confidence: Medium

Parameter: name

Attack: http://localhost:8080/checkout?name=<script>alert(1)</script>

Evidence: CVE ID: 79

Request: http://localhost:8080/checkout?name=<script>alert(1)</script>

Input (Order Form Query):

Source: ActiveJS(12) - Cross Site Scripting (Reflected)

Description: Cross site scripting (XSS) is an attack technique that involves injecting malicious code into a user's browser. The code is then executed in the user's browser, potentially leading to stolen data or compromised user sessions. This vulnerability can be prevented by validating and sanitizing user input, encoding output, and implementing browser security protections such as a Content Security Policy (CSP).

Solution:

Pre application issues: CSP, use Content-Security-Policy, Add Content-Security-Policy as defense-in-depth.

Reference: <https://owasp.org/www-community/attacks/xss/>

The security scan found a high-risk Reflected Cross-Site Scripting (XSS) vulnerability in the checkout page. This means malicious scripts could be injected through user input and executed in another user's browser, potentially leading to stolen data or compromised user sessions. This vulnerability can be prevented by validating and sanitizing user input, encoding output, and implementing browser security protections such as a Content Security Policy (CSP).

The Team

Clevan Davermann — Environment & Testing Lead
<https://www.linkedin.com/in/clevan-davermann-2483b826b/>

Yahir Arballo — Risk Assessment Lead
<https://www.linkedin.com/in/yahir-arballo-920884356/>

Joel Bonilla — Vulnerability Analysis Lead
https://www.linkedin.com/in/bonillajoel?utm_source=share_via&utm_content=profile&utm_medium=member_ios

Matias Cavanzo — Security Controls & Mitigation Lead
www.linkedin.com/in/matias-c-355235284

Marcus Testa — Documentation & Presentation Lead
www.linkedin.com/in/marcus-testa-506a8a345

Technologies

- HTML, Java, C#, SQL
- MySQL
- Apache TomCat

Acknowledgments

- Prof. Mustafa Ocal — Beginner/Intermediate Java
- Prof. Rehan Akbar — Websiteconmgmt
- Prof. Abdul Rehman — Mobile and IoT Security
- Prof. Masoud Sadjadi — Knight Foundation School of Computing and Information Sciences