

Practical Assignment #3

1. Goals

- Explore the WSTG (Web Security Testing Guide)¹ web security testing guidelines.
- Configure and explore the usage of ModSecurity reverse proxy as a WAF (Web Application Firewall)

2. General description

The main goals of this assignment are to explore **web application security** and to implement a **web application firewall** to secure a web application against application-layer attacks. The web application to be used in this assignment is the OWASP JuiceShop^{2,3}. This assignment is split in two phases: the first phase is dedicated to exploring the security of the JuiceShop web application, and the second phase aims at monitor, filter, and block HTTP traffic to the JuiceShop, through the implementation of a WAF using ModSecurity, with the aim to address the security issues identified in the first phase. Figure 1 illustrates the two phases of the assignment, depicting the JuiceShop web server, the penetration testing client and the WAF.

Scenario 1:

Web security testing



Scenario 2:

Web application firewall

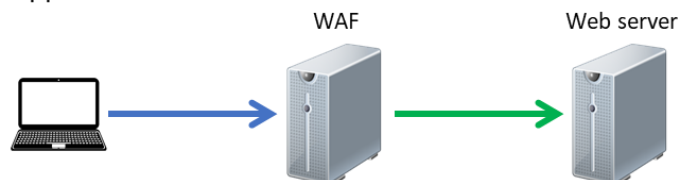


Figure 1 – Security testing and WAF phases of the Assignment

¹ WSTG with v42 is available at: <https://owasp.org/www-project-web-security-testing-guide/>

² OWASP JuiceShop: <https://owasp.org/www-project-juice-shop/>

³ For this assignment, it is recommended to use the most recent version of the JuiceShop. At the time of writing this document it is v17.2.0



3. Phase 1 - Web application security testing

In this phase the goal is to explore web application security using the JuiceShop website following the relevant and applicable WSTG web security testing guidelines, and for this purpose any security tools can be used. In this context, OWASP ZAP and security tools already available in Kali Linux are particularly relevant. This web security testing phase is described in Figure 1, where the client has direct communications to the web server. As part of your tests, the OWASP ZAP penetration tests must, at least:

- a. Perform an automated scan to the website.
- b. Perform an active scan to the website (explore the most effective policies).
- c. Manage add-on required to improve the test and maximize threats identification.
- d. Perform a Fuzz attack to the login form.
- e. Perform a manual penetration test to explore logged in threats.
- f. Configure OWASP ZAP active scan to explore authenticated area.

The JuiceShop application can be installed via source code or using docker, as follows:

- Installation through [source code](#) in one of the virtual machines
- Using a [docker approach](#) (requires Docker Desktop)

As a result of your tests, you should prepare a web application security report, structured along the WSTG guidelines. The report must document the identified vulnerabilities and on how these can be exploited (e.g., weak passwords, insecure configurations).

4. Phase 2 - Setup and testing of a WAF (web application firewall)

Based on the web application security report produced in the first phase of the assignment, deploy an WAF between the client and the web server, as depicted in Figure 1. The goals of this WAF are to monitor, filter, and block HTTP traffic to the Juice Shop. This WAF server should be composed of an Apache 2 service with ModSecurity, and the WAF configuration should be optimized to prevent all possible attacks.

As a result of this phase of the Assignment, you should repeat all penetration tests performed in the previous task and assess the performance of the WAF in detecting and blocking the attacks. You should update the web application security report accordingly, by including the configurations, description of the tests and performance results in a separate section.

5. Delivery of the Practical Assignment



- The deadline for the delivery of the assignment (configuration files and report, via Inforestudante) is 31/5/2026.
- Notes:
 - Assignments without PGP will be accepted, although with a discount of 5% in the final grade.
 - Submissions via Inforestudante.

The delivery of the practical assignment can document aspects regarding the methodology of testing, the analysis of results and can be structured as follows:

- 1) Introduction
- 2) Architecture considered for the PA#3 (for both scenarios 1 and 2)
 - Network structure
 - Servers
 - Services
- 3) Web application security testing
 - 1 Information Gathering
 - 2 Configuration and Deployment Management Testing
 - 3 Identity Management Testing
 - 4 Authentication Testing
 - 5 Authorization Testing
 - 6 Session Management Testing
 - 7 Input Validation Testing
 - 8 Testing for Error Handling
 - 9 Testing for Weak Cryptography
 - 10 Business Logic Testing
 - 11 Client Side Testing
- 4) Web application security firewall
 - 1 Information Gathering
 - 2 Configuration and Deployment Management Testing
 - 3 Identity Management Testing
 - 4 Authentication Testing
 - 5 Authorization Testing
 - 6 Session Management Testing
 - 7 Input Validation Testing
 - 8 Testing for Error Handling
 - 9 Testing for Weak Cryptography
 - 10 Business Logic Testing
 - 11 Client Side Testing
- 5) Conclusions



6. Important/relevant aspects

The [Web Application Security Testing document](#) includes several sections, providing guidelines for testing. The guideline applicable to this assignment is mainly in Section 4, which must be analysed carefully, since testing tools may be suggested in each section.

The practical assignment targets black-box testing, which is according to OWASP in WSTG “the art of testing a system or application remotely to find security vulnerabilities, without knowing the inner workings of the target itself”. Thus, in this type of testing strategy we focus on tools such as web application security scanners, vulnerability scanners and penetration testing software.

The following aspects are relevant in what respects the WSTG guidelines and structure:

1. Section 4.7 should be considered as a whole, which can be tested with OWASP ZAP or a similar tool. There are some subsections, that do not apply in this assignment. For instance, the Juice Shop does not include any support for LDAP, so subsection 4.7.6 – “Testing for LDAP injection” does not require any action/testing.
2. Section 4.11 should be considered as a whole, which can be tested with OWASP ZAP or a similar tool.
3. Section 4.9 should not be considered since communications with Juice Shop are not over HTTPS.
4. Section 4.10 should not be considered as well, as it is out of scope of this assignment.
5. Other subsections are out of scope of this assignment, and students should identify these in the report (and explain why). For instance. subsection 4.2.9, 4.2.10 and 4.2.11 are not applicable in this assignment.

Regarding the second phase of the work, with the Web Application Firewall, the following aspects should be considered:

1. The main goal of the project in the second phase is to enable detection and prevention of the issues identified in the first phase. Nonetheless, all the detection and prevention actions must be possible using Apache and ModSecurity (with OWASP CRS), no other tools should be considered for this purpose.
2. The issues identified Section 4.3 of WSTG cannot be detected and solved with ModSecurity (with OWASP CRS), so no action is required. ModSecurity is a WAF that operates at the HTTP level – it analyzes HTTP requests and responses and blocks traffic based on patterns (malicious payloads, suspicious headers, etc.). Identity Management issues are application logic flaws, not attacks with detectable patterns in HTTP traffic.