

## Lab 2: OWASP Top 10 Web Application Testing for GRC Professionals

### 1. Introduction

Welcome back, GRC analysts. In the modern threat landscape, web applications are a primary attack vector. This lab will familiarize you with the **OWASP Top 10**, a powerful awareness document and de facto standard for understanding the most critical web application security risks. You will use Burp Suite Professional to identify these risks and learn how to articulate them in terms of business impact and compliance failure.

### 2. Scenario

Your team has been made aware of a legacy web application server (owaspbwa) in a development environment. Due to an upcoming merger, this system must be assessed for compliance with the company's new application security standard, which is based on the OWASP Top 10. Your task is to perform a targeted assessment, focusing on these specific risk categories.

### 3. Objectives

- Discover and explore vulnerable web applications on the target.
- Configure and use Burp Suite Professional to intercept, analyze, and manipulate web traffic.
- Identify specific vulnerabilities that map to the OWASP Top 10.
- Assess the business impact and compliance implications of each finding.
- Document findings in a structured report for technical and management audiences.

### 4. Lab Setup

- **Your Machine:** Kali Linux (Attacker) - IP: 192.168.x.x
- **Target Machine: OWASP Broken Web Applications (BWA)** - IP: 192.168.x.z (Instructor will provide this)
  - *Note: OWASP BWA is a collection of vulnerable web applications, perfect for this lab.*

### 5. Step-by-Step Instructions

#### Phase 1: Discovery and Reconnaissance

##### Step 1: Target Discovery

- **Task:** Confirm the target is alive and identify its IP address.
- **Command:** ping -c 4 <TARGET\_IP>

##### Step 2: Service Enumeration

- **Task:** Identify what web services and ports are open on the target.
- **Command:** nmap -sV -p 80,8080,9090 <TARGET\_IP> (Common web ports for BWA)
- Action:
  - ✓ Save output in multiple formats for comprehensive documentation and analysis:
  - ✓ nmap -sV -sC -O -p- -oA initial\_scan <TARGET\_IP>

- This generates three files initial\_scan.nmap, initial\_scan.gnmap, and initial\_scan.xml.
- Generate an HTML report for enhanced visualization and reporting:
  - ✓ xsltproc -o initial\_scan\_report.html initial\_scan.xml
  -
- **GRC Context:** Asset discovery is the first step in any risk management lifecycle. You cannot protect what you don't know about.

### **Step 3: Application Discovery**

- **Task:** Explore the target to find the vulnerable applications.
- **Action:** Open `http://<TARGET_IP>` in your **Firefox browser**. You will see a directory listing of available applications (e.g., WebGoat, DVWA, Bodgelt Store).
- **Action:** Click on several links to see what applications are present. For this lab, we will focus on "**Bodgelt Store**" and "**WebGoat**".

## **Phase 2: Tool Configuration & Automated Scanning**

### **Step 4: Configuring Burp Suite Professional**

- **Task:** Set up your environment to route traffic through Burp.
  1. Launch **Burp Suite Professional**.
  2. In Firefox, configure the manual proxy settings as in the previous lab: **HTTP Proxy: 127.0.0.1 Port: 8080**.
  3. Ensure "**Intercept is on**" in the **Proxy > Intercept** tab.
  4. Browse to `http://<TARGET_IP>` and forward the requests in Burp to populate the **Target > Site map**.
  5. **Turn Intercept off** once the site map is populated.

### **Step 5: Automated Passive & Active Scanning**

- **Task:** Let Burp's scanner perform an initial automated assessment to find low-hanging fruit.
- **Action:**
  1. In the **Site map**, right-click on the `http://<TARGET_IP>` folder.
  2. Select "**Scan**" > "**Scan defined URLs**".
  3. Review the scan configuration and launch it.
  4. Monitor the progress from the **Dashboard** tab. This will run in the background while you perform manual testing.

## **Phase 3: Manual Testing Against OWASP Top 10**

## **Step 6: Testing for A01:2021-Broken Access Control (BodgeIt Store)**

- **Task:** Try to access privileged resources without proper authorization.
- **Action:**
  1. In your browser, go to **BodgeIt Store** ([http://<TARGET\\_IP>/bodgeit](http://<TARGET_IP>/bodgeit)).
  2. Log in or register a new user account.
  3. Add a few items to your cart and proceed to the checkout. Note the URL.
  4. **Try to access an administrative page directly:** Enter the URL [http://<TARGET\\_IP>/bodgeit/admin.jsp](http://<TARGET_IP>/bodgeit/admin.jsp). Can you see it?
- **GRC Context:** Broken Access Control is a critical failure in authorization policies, directly violating the principle of least privilege.

## **Step 7: Testing for A02:2021-Cryptographic Failures (WebGoat)**

- **Task:** Identify if data is being transmitted in plaintext.
- **Action:**
  1. Go to **WebGoat** ([http://<TARGET\\_IP>/webgoat](http://<TARGET_IP>/webgoat)).
  2. Log in and navigate to a lesson.
  3. In Burp, go to the **Proxy > HTTP history** tab.
  4. Look for any POST request that contains a password. Look at the raw request in the "Params" or "Raw" sub-tab. Is the password in plain text?
- **GRC Context:** Transmitting passwords in cleartext is a severe violation of virtually every data protection standard, including PCI DSS and GDPR.

## **Step 8: Testing for A03:2021-Injection (BodgeIt Store Search)**

- **Task:** Test a search field for SQL Injection (SQLi).
- **Action:**
  1. In the **BodgeIt Store**, find the search bar.
  2. **Turn Intercept on** in Burp.
  3. In the search bar, type a single quote ' and click search.
  4. The request will be caught in Burp. **Send this request to Repeater** (Right-click > Send to Repeater).
  5. In the **Repeater** tab, modify the search parameter and try different payloads:
    - ' OR '1'='1 -- (Should return all items)

- ' UNION SELECT 1,2,3,4,5,6 -- (See if you can uncover database structure)
6. Observe the HTTP response for errors or unexpected data, which indicate a potential SQLi flaw.
- **GRC Context:** SQL Injection is a classic high-impact vulnerability that can lead to full data breach, violating data integrity and confidentiality policies.

#### **Step 9: Testing for A07:2021-Identification and Authentication Failures (BodgeIt Login)**

- **Task:** Test for weak authentication mechanisms.
- **Action:**
  1. In the **BodgeIt Store**, go to the Login page.
  2. Try common weak credentials like admin:admin, test:test.
  3. **Use Burp Intruder for a brute-force test (Carefully!):**
    - Intercept a login request and send it to **Intruder**.
    - Clear all payload positions and set a new one only on the password parameter.
    - Go to the **Payloads** tab and add a small list of common passwords (e.g., password, admin, 123456, letmein).
    - Start the attack. Look for a response with a different length or status code, which might indicate a successful login.
- **GRC Context:** Weak authentication controls directly violate password policy and account management requirements in frameworks like CIS Controls and NIST CSF.

#### **Phase 4: Analysis and Reporting (The GRC Focus)**

##### **Step 10: Triage and Risk Assessment**

Create a findings table focused on the OWASP Top 10.

OWASP Top 10 Category	Vulnerability Example	Affected Application	Inherent Risk (L/M/H)	Compliance Violation (e.g., PCI DSS 6.5.1)	Business Impact
A01: Broken Access Control	Unprotected Admin Page	BodgeIt Store	H	PCI DSS 7.2.1	Unauthorized data access, privilege escalation
A02: Cryptographic Failures	Cleartext Password Transmission	WebGoat	H	PCI DSS 4.1	Credential theft, account takeover

A03: Injection	SQL Injection in Search	BodgeIt Store	H	PCI DSS 6.5.1	Full database compromise, data loss
A07: Identification Failures	Weak Password Policy	BodgeIt Store	M/H	NIST CSF <a href="#">PR.AC-1</a>	Unauthorized access, initial breach vector

### Step 11: Drafting the Executive Summary

**To:** CISO & Application Development Management

**From:** GRC Audit Team

**Date:** [Date]

**Subject:** Critical OWASP Top 10 Findings in Legacy Web Applications

**1. Executive Summary:** An assessment of the owaspbwa server revealed multiple severe web application vulnerabilities that align with the OWASP Top 10. These findings indicate a systemic failure in our Secure Development Lifecycle (SDLC) and pose a direct threat to customer and company data.

### 2. Key Findings & Risks:

- **Critical Risk - Data Breach Vector:** Several applications were found to be vulnerable to SQL Injection (A03:2021). This could allow an attacker to steal, modify, or delete the entire application database.
- **Critical Risk - Broken Access Controls (A01:2021):** Administrative interfaces were accessible without proper authentication, allowing any user to gain privileged access to application functions and data.
- **High Risk - Lack of Encryption (A02:2021):** User credentials were observed being transmitted without encryption, making them susceptible to interception.

### 3. Recommended Actions:

1. **Immediate Mitigation:** Isolate the affected systems from any network reachable by untrusted users.
2. **Remediation:** The development team must prioritize fixing these vulnerabilities using OWASP Proactive Controls and Cheat Sheets.
3. **Process Improvement:** Mandate formal security training for developers focused on the OWASP Top 10. Integrate SAST/DAST tools (like Burp Suite) into the CI/CD pipeline.
4. **Penetration Testing:** Commission a full third-party penetration test before any similar applications are deployed in production.

### 6. Lab Conclusion

You have now used advanced manual and automated techniques to test for specific, high-impact web application vulnerabilities. For a GRC professional, understanding the mechanics of the OWASP Top 10 is essential for validating application security controls, assessing vendor risk questionnaires, and ensuring the organization's software development practices are secure and compliant.

## **7. Deliverables**

Please submit the following:

1. A screenshot of your Burp Suite **Dashboard** showing the scan results.
2. A screenshot of a successful vulnerability test from the **Repeater** tab (e.g., your SQL Injection payload and response).
3. Your completed **OWASP Top 10 Risk Assessment Table** (from Step 10).
4. Your **Executive Summary** (from Step 11).