Assignment 03
Reconnaissance Report

December 05 - December 09

Prepared by: Danil Vilmont & Jason Mou
Course: ITAS 268 - Assignment 03
Date: December 09, 2025

Table of Contents

Next Page →

Team Information:

- **Team Member 1:** Jason Mou
    - **Role:** Hunter
- **Team Member 2:** Danil Vilmont
    - **Role:** Scribe
- **Date Range of Reconnaissance:** 12/05/2025 - 12/08/2025
- **bWAPP Version:** 2.2

Executive Summary:

- **Total Vulnerability Categories Identified:** 10 / 8 (minimum)
- **Brief Overview:** Our reconnaissance of the bWAPP instance revealed multiple high-impact vulnerability classes across authentication, input validation, and client-side injection. Initial inspection suggests several areas can lead to privilege escalation or data compromise. Manual inspection confirmed exploitable parameters in multiple areas.

Application Mapping:

Page/Function	URL Path	Authentication Required?
Login Page	/bWAPP/login.php	No
Home Page	/bWAPP/home.php	Yes
SQL Injection (Search)	/bWAPP/sqli_1.php	Yes
SQL Injection (Login)	/bWAPP/sqli_2.php	No
XSS – Reflected	/bWAPP/xss_reflected.php	Yes
XSS – Stored (Blog)	/bWAPP/xss_stored_1.php	Yes
OS Command Injection	/bWAPP/os_cmd.php	Yes
Local File Inclusion	/bWAPP/rlfi.php	Yes
Directory Traversal	/bWAPP/directory_traversal_1.php	Yes
IDOR – Change Secret	/bWAPP/idor_1.php	Yes
Broken Authentication (CAPTCHA)	/bWAPP/ba_captcha_bypass.php	Yes
Base64 / Sensitive Data	/bWAPP/base64.php	Yes

Next Page →

FINDING #1:

- **Vulnerability Category:** SQL Injection (GET/Search)
- **OWASP Top 10 Classification:** A05:2025 - Injection
- **Locations:**
    - /bWapp/sqli_1.php (parameter: search)
    - **How Discovered:** Manual Testing
    - **Initial Severity:** High
    - **Severity Justification:** Allows direct execution of SQL and data extraction.
    - **Brief Description:** The GET parameter is concatenated into the SQL query without sanitization, enabling data retrieval.
    - **Plan:** Exploit in Phase 2

FINDING #2:

- **Vulnerability Category:** SQL Injection (Login Form - POST)
- **OWASP Top 10 Classification:** A05:2025 - Injection
- **Locations:**
    - /bWapp/login.php (login, password)
    - **How Discovered:** Manual Testing
    - **Initial Severity:** Critical
    - **Severity Justification:** A login bypass is possible using SQL injection payloads.
    - **Brief Description:** The Login query directly uses unvalidated input, enabling authentication bypass.
    - **Plan:** Exploit in Phase 2

FINDING #3:

- **Vulnerability Category:** XSS - Reflected
- **OWASP Top 10 Classification:** A05:2025 - Injection
- **Locations:**
    - /bWapp/xss_reflected.php (input)
    - **How Discovered:** Manual Testing
    - **Initial Severity:** High
    - **Severity Justification:** Attacker-controlled JavaScript executed in the victim's browser.
    - **Brief Description:** Input is reflected unescaped in the HTML response, allowing arbitrary JS.
    - **Plan:** Exploit in Phase 2

Next Page →

FINDING #4:

- **Vulnerability Category:** XSS - Stored (Blog)
- **OWASP Top 10 Classification:** A05:2025 - Injection
- **Locations:**
    - /bWapp/xss_stored_1.php (comment fields)
    - **How Discovered:** Manual Testing
    - **Initial Severity:** Critical
    - **Severity Justification:** Payload persists and executes for all visitors.
    - **Brief Description:** Application stores user input unsanitized and re-renders it to all users.
    - **Plan:** Exploit in Phase 2

FINDING #5:

- **Vulnerability Category:** OS Command Injection
- **OWASP Top 10 Classification:** A05:2025 - Injection
- **Locations:**
    - /bWapp/os_cmd.php (target)
    - **How Discovered:** Manual Testing
    - **Initial Severity:** Critical
    - **Severity Justification:** Direct shell execution is possible.
    - **Brief Description:** User input is passed directly to the OS shell command, resulting in remote command execution.
    - **Plan:** Exploit in Phase 2

FINDING #6:

- **Vulnerability Category:** Local File Inclusion
- **OWASP Top 10 Classification:** A02:2025 - Security Misconfiguration
- **Locations:**
    - /bWapp/rlfi.php (page)
    - **How Discovered:** Manual Testing
    - **Initial Severity:** High
    - **Severity Justification:** Arbitrary local files read from the server.
    - **Brief Description:** The File path parameter is used directly in include(), allowing LFI.
    - **Plan:** Exploit in Phase 2

Next Page →

FINDING #7:

- **Vulnerability Category:** Broken Authentication (CAPTCHA)
- **OWASP Top 10 Classification:** A07:2025 - Identification & Authentication Failure
- **Locations:**
  - /bWapp/ba_captcha_bypass.php
  - **How Discovered:** Manual Testing
  - **Initial Severity:** Medium
  - **Severity Justification:** CAPTCHA can be easily bypassed and is not rate-limiting.
  - **Brief Description:** Weak CAPTCHA allows brute-force login attempts.
  - **Plan:** Exploit in Phase 2

FINDING #8:

- **Vulnerability Category:** Insecure Direct Object Reference (Change Secret)
- **OWASP Top 10 Classification:** A01:2025 - Broken Access Control
- **Locations:**
  - /bWapp/idor_1.php (id)
  - **How Discovered:** Manual Testing
  - **Initial Severity:** High
  - **Severity Justification:** The User can modify other users' information by changing identifiers.
  - **Brief Description:** Access control is not enforced on object IDs, enabling unauthorized modification.
  - **Plan:** Exploit in Phase 2

FINDING #9:

- **Vulnerability Category:** Directory Traversal
- **OWASP Top 10 Classification:** A02:2025 - Security Misconfiguration
- **Locations:**
  - /bWapp/directory_traversal_1.php (parameter: page)
  - **How Discovered:** Manual Testing
  - **Initial Severity:** High
  - **Severity Justification:** The application allows arbitrary file retrieval on the server via path manipulation.
  - **Brief Description:** The file parameter accepts relative paths such as "../..", which can lead to directory traversal and expose system or sensitive application files.
  - **Plan:** Exploit in Phase 2

Next Page →

FINDING #10:

- **Vulnerability Category:** Sensitive Data Exposure (Clear-Text HTTP)
- **OWASP Top 10 Classification:** A04:2025 - Cryptographic Failures
- **Locations:**
  - Entire application (HTTP communication)
  - **How Discovered:** Manual Testing
  - **Initial Severity:** Medium
  - **Severity Justification:** The application uses unencrypted HTTP, which may expose credentials or session identifiers in transit across non-trusted networks.
  - **Brief Description:** The application does not enforce HTTPS and transmits authentication and session data in clear text over HTTP, enabling interception on untrusted networks.
  - **Plan:** Exploit in Phase 2 (kind of unnecessary though, as the environment is localhost)

Tools Used During Reconnaissance:

- **[x] Browser Developer Tools**
  - **Notes:** Used to list input fields, form parameters, request/response structure, and client-side behavior.
- **[x] Burp Suite Community Edition**
  - **Notes:** Intercepted requests and manually inspected parameters for potential manipulation during reconnaissance.
- **[x] OWASP ZAP**
  - **Notes:** Passive scanning and spidering are used to identify exposed attack surface and inputs.
- **[ ] Nmap**
  - **Notes:** Not Used
- **[ ] Nikto**
  - **Notes:** Not Used
- **[ ] Manual Code Review**
  - **Notes:** Not Used
- **[ ] Other**
  - **Notes:** None

Next Page →

Attack Surface Summary:

- **[x] Form fields (**Count: ~20**)**
- **[x] URL parameters (**Count: ~25**)**
- **[x] Cookies (**Count: 2**)**
- **[x] HTTP Headers (**Count: A lot (default browser + host headers)**)**
- **[x] File Uploads (**Count: ~10**)**
- **[x] Hidden fields (**Count: ~15**)**
- **[x] Other (**Count: N/A**)**

Authentication Mechanisms Observed:

- Simple username/password login form
- No MFA
- CAPTCHA only on specific authentication flows
- Session cookie used to track authenticated state
- No enforced lockout or rate-limiting

Session Management Notes:

- Session identifiers are stored in PHP session cookies
- Cookies are without HttpOnly or secure flags
- Logout does not clear the session cookie from the browser
- Session relies on only cookie validity
- Session not bound to client (IP/Other), allowing reuse if grabbed

---

Next Page →

Phase 2: Exploitation plan

Priority	Vulnerability Category	Assigned To	Rationale
1 Critical	**F#2:** SQL Injection (Login Form - POST)	Jason Mou	The login page trusts whatever we type and sends it straight to the database.
2 Critical	**F#4:** XSS - Stored (Blog)	Danil Vilmont	Payload persists and executes for all users, enabling session theft and privilege escalation.
3 High	**F#9:** Directory Traversal	Danil Vilmont	Enables reading of arbitrary local files. This can include config files.
4 High	**F#1:** SQL Injection (GET/Search)	Jason Mou	The movie search field puts user input directly into the SQL query without validation.
5 Medium	**F#7:** Broken Authentication (CAPTCHA)	Danil Vilmont	Weak CAPTCHA allows brute-force login attempts.
Backup	**F#5:** OS Command Injection	Jason Mou	The application accepts user input and passes it directly to a system command without validation.

Initial Obervations & Notes:

- Many vulnerable modules are accessible directly from the menu.
- Several issues persist even with the security level increased.
- Error messages often reveal internal workings.
- Sessions rely solely on cookies, with no other protections.

Next Page →

Team Member Signatures:

- By signing below, we confirm this reconnaissance was performed ethically on our own local bWAPP installation and represents our original work.
    - **Team Member 1:** Jason Mou
        - **Date:** 12/09/2025
    - **Team Member 2:** Danil Vilmont
        - **Date:** 12/09/2025