Vulnerability Assessment Report

Report Title: Vulnerability Assessment for itsecgames.com (bWAPP Lab) – Reconnaissance Phase
Assessment Date: September 15–19, 2025
Report Date: September 19, 2025
Version: 1.0
Prepared By: Sai Bhavya E – Security Officer Trainee
Prepared For: itsecgames.com Project Stakeholders
Classification: Confidential – Internal Use Only

Table of Contents

---

1. Executive Summary

This vulnerability assessment was conducted on the itsecgames.com bWAPP (Buggy Web Application) lab as part of the reconnaissance phase of a Vulnerability Assessment and Penetration Testing (VAPT) engagement.
Key Highlights:

- **Target Assessed:** itsecgames.com (bWAPP lab, IP: 31.3.96.40)
- **Total Vulnerabilities Identified:** 13 (4 Critical, 4 High, 3 Medium, 2 Low)
- **Risk Posture:** High – exposed services, missing security headers, and directory disclosures expand the attack surface
- **Top Risks:**
  - Open SSH service (Port 22) vulnerable to brute-force attacks
  - Missing HTTP security headers allowing clickjacking and XSS
  - Exposed directories revealing sensitive files
  - Unrestricted File Upload

Overall Risk Rating: High
Estimated Remediation Effort: 3–5 weeks for critical and high issues

---

2. Introduction

Background

This report documents the reconnaissance phase of a Vulnerability Assessment and Penetration Testing (VAPT) engagement on the bWAPP lab hosted at itsecgames.com. The bWAPP environment is deliberately vulnerable for training purposes.

Objectives

- Identify open ports and services
- Enumerate web directories and detect misconfigurations
- Assess initial security posture to guide deeper testing

Limitations

- External reconnaissance only (no internal network access)
- Non-intrusive scans (no denial-of-service or destructive exploitation)
- Exploitation was out-of-scope at this stage

---

3. Scope and Methodology

Scope

- **In-Scope Assets:**
    - IP: 31.3.96.40
    - Domain: itsecgames.com
    - Ports: 22 (SSH), 80 (HTTP), 443 (HTTPS)
    - Web Paths: /bugs.htm, /downloads/, /admin/, /install.php
- **Out-of-Scope:**
    - Social engineering
    - Physical security testing
    - Active exploitation

Methodology

Phase	Tools/Techniques	Description
Reconnaissance	Nmap, Gobuster, Nikto, Curl	Port scanning, directory enumeration, fingerprinting, header checks
Evidence	proofs, Logs	Collected evidence for the vulnerabilities
Analysis	Manual Review	Consolidated findings into vulnerability list

---

4. System Overview

- **Target Environment:** bWAPP app on Apache/2.4.7 with OpenSSH 6.7p1, hosted on Ubuntu 14.04
- **Key Components:**
    - Web server: Apache/2.4.7
    - SSH service: OpenSSH 6.7p1
    - Database: MySQL (internal only)
- **SSL Certificate:** Issued for mmebv.be with SAN = itsecgames.com

---

5. Findings

Vulnerability Summary

ID	Vulnerability	Severity	Evidence Path	Status
V-001	SQL Injection (GET/Search)	Critical	proofs/detected_vulnerabilities/sqli_get_20250917_031059	Confirmed
V-002	SQL Injection (Blind)	High	proofs/detected_vulnerabilities/sqli_blind_20250918_144632	Confirmed
V-003	XSS Reflected (GET/POST)	High	proofs/detected_vulnerabilities/xss_get_20250917_034645, xss_reflected_post_20250918_144632	Confirmed
V-004	XSS Stored (Blog/Change Secret)	High	proofs/detected_vulnerabilities/xss_stored_20250918, xss_change_secret_20250918	Confirmed
V-005	CSRF (Change Secret)	High	proofs/detected_vulnerabilities/csrf_change_secret_20250918	Confirmed
V-006	Unrestricted File Upload	Critical	proofs/detected_vulnerabilities/unrestricted_upload_20250918	Confirmed
V-007	Insecure Direct Object Reference (Change Secret)	High	proofs/detected_vulnerabilities/insecure_dor_change_secret_20250918	Confirmed
V-008	Directory Traversal (/etc/passwd)	Critical	proofs/detected_vulnerabilities/dir_traversal_20250918_153854	Confirmed
V-009	Server-Side Request Forgery (SSRF)	High	proofs/detected_vulnerabilities/ssrf_20250918_144632	Confirmed

V-010	Clickjacking	Medium	[proofs/detected_vulnerabilities/clickjacking_headers_20250918_144632]()	Confirmed
V-011	Information Disclosure – Headers	Medium	[proofs/detected_vulnerabilities/info_disclosure_headers_20250918]()	Confirmed
V-012	Environment Exposure (Debug Info)	Low	[proofs/detected_vulnerabilities/environment_20250917_034654]()	Confirmed
V-013	Missing TLS/SSL (Cleartext HTTP)	Critical	[Proofs/detected_vulnerabilities/ss_tls_cleartext_20250919/]()	Confirmed

Total: 13 vulnerabilities (4 Critical, 4 High, 3 Medium, 2 Low)

Detailed Finding (V-001)

ID: V-001
Title: SQL Injection (GET / Search)
Severity: Critical

Description:
An input used in a GET request (search parameter / query string) is vulnerable to SQL injection. The application fails to properly parameterize or sanitize the input, allowing an attacker to inject SQL payloads that reveal or extract database contents.

Impact:

- Unauthorized disclosure of sensitive data (users, credentials, application configuration).
- Possible full database compromise and pivot to remote code execution depending on database privileges.
- Data integrity loss and privacy breach.

Reproduction Steps:

1. **curl -s "http://127.0.0.1:8080/target_page.php?search=bee" -b cookies.txt -D - | sed -n '1,120p'**
2. **sqlmap -u "http://127.0.0.1:8080/target_page.php?search=bee" -p search --batch –dbs**

Evidence:

[**proofs/detected_vulnerabilities/sqli_get_20250917_031059/**]()

ID: V-002
Title: SQL Injection (Blind)
Severity: High

Description:
A parameter does not return SQL errors but is vulnerable to blind SQL injection (time-based / Boolean). An attacker can extract data by measuring response behaviour/time.

Impact:

- Exfiltration of database contents without visible errors.
- Possibility to enumerate schema, users, hashes and pivot further.

Reproduction Steps:

1.
   ```
   curl -s "http://127.0.0.1:8080/target_page.php?id=1" -b cookies.txt -D - | sed -n '1,120p'
   ```
2. Run sqlmap time-based test:
   ```
   sqlmap -u "http://127.0.0.1:8080/target_page.php?id=1" -p id --batch --risk=3 --level=5 --technique=T
   ```

Evidence:

proofs/detected_vulnerabilities/sqli_blind_20250918_144632/

ID: V-003
Title: Cross-Site Scripting (Reflected — GET / POST)
Severity: High

Description:
User input returned in responses without proper output encoding. Payloads in query/body can execute in victim browsers. Both GET and POST reflected XSS confirmed.

Impact:

- Session theft, CSRF escalation, phishing, account takeover for logged-in users.

Reproduction Steps:

1. Reflected (GET):
   ```
   curl -s "http://127.0.0.1:8080/search.php?q=<script>alert(1)</script>" -b cookies.txt -D - | sed -n '1,120p'
   ```
2. Reflected (POST):
   ```
   curl -s -X POST -b cookies.txt -d "comment=<script>alert(1)</script>" http://127.0.0.1:8080/comment.php -D -
   ```

Evidence:

proofs/detected_vulnerabilities/xss_get_20250917_034645/
proofs/detected_vulnerabilities/xss_reflected_post_20250918_144632/

ID:V-004
Title: Cross-Site Scripting (Stored — Blog / Change Secret)
Severity: High

Description:
User content stored by the application is rendered later without sanitization (stored XSS). Payloads persist and execute in any visitor/admin context.

Impact:

- Persistent site-wide XSS, potential remote code execution in some contexts, user/session compromise.

Reproduction Steps:

1. Submit payload to blog or change-secret endpoint:
   curl -s -b cookies.txt -d "entry=<script>alert('xss')</script>" http://127.0.0.1:8080/blog.php -D -
2. Visit blog page and observe execution.

Evidence:

proofs/detected_vulnerabilities/xss_stored_20250918/
proofs/detected_vulnerabilities/xss_change_secret_20250918/

ID: V-005
Title: Cross-Site Request Forgery (CSRF — Change Secret)
Severity: High

Description:
State-changing form (change secret) lacks anti-CSRF token and can be triggered by third-party sites.

Impact:

- Attackers can change user secrets/settings if victims visit a malicious page while authenticated.

Reproduction Steps:

1. Create a simple HTML page that POSTs to /csrf_3.php with login=bee and action=change and secret=attacker.
2. Host page and have victim visit it while logged in; observe secret changed.

Evidence:

proofs/detected_vulnerabilities/csrf_change_secret_20250918/

ID: V-006
Title: Unrestricted File Upload
Severity: Critical

Description:
File upload accepts arbitrary file types (text allowed) and the app links to uploaded file under webroot. Allows storing non-image content and potentially executable webshells.

Impact:

- Remote code/shell upload if server executes uploaded files, stored XSS via uploaded HTML, data exfiltration.

Reproduction Steps:

1. printf 'test' > /tmp/upload_test.txt
2. curl -s -b cookies.txt -F "file=@/tmp/upload_test.txt" -F "form=Upload" http://127.0.0.1:8080/unrestricted_file_upload.php -D -
3. Visit the returned /images/upload_test.txt URL.

Evidence:

proofs/detected_vulnerabilities/unrestricted_upload_20250918/

ID: V-007
Title: Insecure Direct Object Reference (IDOR / Insecure DOR — Change Secret)
Severity: High

Description:
Application uses direct identifiers (e.g., username/ID) in hidden fields without authorization checks, allowing one user to change another's secret by supplying their login value.

Impact:

- Unauthorized modification of other users' data (privacy breach, account takeover).

Reproduction Steps:

1. Observe form contains <input type="hidden" name="login" value="bee">.
2. Replace login value with another user and submit; if change succeeds, IDOR confirmed.

Evidence:

proofs/detected_vulnerabilities/insecure_dor_change_secret_20250918/

ID: V-008

Title: Directory Traversal (/etc/passwd disclosure)

Severity: Critical

Description:

Application allows path traversal sequences to access files outside webroot (e.g., ../../../../etc/passwd). Note: captured evidence files were initially empty — re-capture recommended.

Impact:

- Exposure of sensitive system files (passwords, configuration), which greatly aid attackers.

Reproduction Steps:

1. curl -s -b cookies.txt --get --data-urlencode 'page=../../../../../../etc/passwd' "http://127.0.0.1:8080/" -D -
2. Check response body for /etc/passwd contents.

Evidence:

proofs/detected_vulnerabilities/dir_traversal_20250918_153854/

ID: V-009

Title: Server-Side Request Forgery (SSRF)

Severity: High

Description:

Application makes server-side HTTP requests using attacker-controlled input (e.g., URL fetch) enabling internal network probing or access to metadata services.

Impact:

- Internal service access, metadata/credential disclosure, pivot to internal network.

Reproduction Steps:

```
curl -s -b cookies.txt -G --data-urlencode "url=http://127.0.0.1:80/admin"
"http://127.0.0.1:8080/ssrf.php" -D –
```

Evidence:

proofs/detected_vulnerabilities/ssrf_20250918_144632/

ID: V-010

Title: Clickjacking (Missing X-Frame-Options / CSP frame-ancestors)

Severity: Medium

Description:

Responses lack anti-framing headers (X-Frame-Options or Content-Security-Policy: frame-ancestors), enabling UI redressing attacks (clickjacking).

Impact:

- Trick users into performing actions in framed interfaces (e.g., change settings).

Reproduction Steps:

1. curl -I http://127.0.0.1:8080/ and observe no X-Frame-Options header.
2. Build a page with <iframe src="http://127.0.0.1:8080/..."> and verify embedding.

Evidence:

proofs/detected_vulnerabilities/clickjacking_headers_20250918_144632/

ID: V-011
Title: Information Disclosure — Server / PHP Headers
Severity: Medium

Description:
HTTP responses reveal server software and PHP version via headers (e.g., Server: Apache/2.4.7, X-Powered-By: PHP/5.5.9), which leaks actionable version info for attackers.

Impact:

- Attackers can look up targeted CVEs for those versions.

Reproduction Steps:

1. curl -I http://127.0.0.1:8080/
2. Observe Server and X-Powered-By headers in the response.

Evidence:

proofs/detected_vulnerabilities/info_disclosure_headers_20250918/

ID: V-012
Title: Environment Exposure (Debug Info / Image / Container metadata)
Severity: Low

Description:
Extra environment data (docker inspect, image info, sha256sums) was captured in environment_20250917_034654/ showing metadata about the environment that isn't needed publicly. May include image fingerprints or container logs.

Impact:

- Information may help fingerprint environment and find relevant exploits; low risk but should not be public.

Reproduction Steps:

List captured files:

```
ls -la proofs/detected_vulnerabilities/environment_20250917_034654/
cat proofs/detected_vulnerabilities/environment_20250917_034654/docker_inspect.json
```

Evidence:

proofs/detected_vulnerabilities/environment_20250917_034654/

ID: V-013
Title: Missing SSL/TLS Certificate & Weak HTTPS Configuration
Severity: Medium

Description:
The application on port 443 does not present a valid SSL/TLS certificate when probed. Our openssl s_client output showed *"no peer certificate available"* and no valid cipher negotiation. This indicates that HTTPS is either misconfigured or entirely absent, leaving the service without encryption in transit.

Impact:

- Users cannot securely connect via HTTPS.

- Risk of man-in-the-middle (MITM) attacks, credential theft, and data exposure.
- Negative trust indicators in browsers (invalid certificate warnings).

Reproduction Steps:

1. Run openssl s_client -connect 127.0.0.1:8080 </dev/null
2. Observe: *"no peer certificate available"*.
3. Nmap SSL scripts (ssl-cert, ssl-enum-ciphers) fail to retrieve certificate details.

Evidence:

proofs/detected_vulnerabilities/ssl_tls_misconfig_20250919/

---

6. Risk Assessment

Risk Matrix

Vuln ID	**Vulnerability**	**Likelihood**	**Impact**	**Risk Level**	**Business Impact**
V-001	SQL Injection (GET/Search)	High	Critical	Critical	Full database compromise; exposure of sensitive data
V-002	SQL Injection (Blind)	Medium	High	High	Data extraction possible with time; increased attacker persistence
V-003	XSS Reflected (GET/POST)	Medium	Medium	Medium	User session hijacking, phishing risk
V-004	XSS Stored (Blog/Change Secret)	High	High	Critical	Persistent session hijacking, privilege escalation
V-005	CSRF (Change Secret)	High	High	Critical	Unauthorized state change; attacker controls victim's account settings
V-006	Unrestricted File Upload	High	Critical	Critical	Remote code execution possible; server takeover
V-007	Insecure Direct Object Reference (Change Secret)	Medium	High	High	Unauthorized access to sensitive objects; data manipulation
V-008	Directory Traversal (/etc/passwd)	Medium	High	High	Disclosure of system files; aid in privilege escalation
V-009	Server-Side Request Forgery (SSRF)	Medium	High	High	Pivot to internal network; possible metadata/API key exposure
V-010	Clickjacking	Medium	Medium	Medium	Trick users into malicious actions; reputational/legal risk

Vuln ID	Vulnerability	Likelihood	Impact	Risk Level	Business Impact
V-011	Information Disclosure – Headers	Low	Medium	Low	Reveals stack versions (Apache, PHP); aids attacker reconnaissance
V-012	Environment Exposure (Debug Info)	Low	Medium	Low	Leakage of configuration/debug details; increases attacker knowledge
V-013	Missing TLS/SSL Encryption (Cleartext HTTP)	Critical	Critical	Critical	Cleartext HTTP → MITM, credential theft.

7. Recommendations

Priority	Vuln ID	Vulnerability	Recommendation	Effort	Timeline	Owner
Critical	V-004	Outdated Apache Version	Upgrade Apache to 2.4.62+ (latest stable). Apply vendor patches regularly.	High	2 weeks	SysAdmin
Critical	V-005	Accessible install.php	Remove or restrict /install.php. Use file permissions or delete after installation.	Low	1 week	DevOps
Critical	V-010	Unrestricted File Upload	Enforce MIME/extension whitelisting, scan uploads, store outside web root.	Medium	2 weeks	DevOps
Critical	V-013	Missing TLS/SSL (Cleartext HTTP)	Enable HTTPS, configure TLS 1.2+/1.3, enforce secure cookies, redirect HTTP→HTTPS.	Medium	2 weeks	SysAdmin
High	V-002	Missing Security Headers	Add headers: X-Frame-Options: SAMEORIGIN, X-Content-Type-Options: nosniff, Content-Security-Policy: default-src 'self'.	Low	1 week	DevOps
High	V-001	Open SSH Port	Enable fail2ban, restrict SSH to trusted IPs, enforce key-based auth, disable root login.	Medium	2 weeks	SysAdmin
High	V-006	SQL Injection (GET/POST)	Use parameterized queries (prepared statements), sanitize inputs, enforce least privilege DB user.	High	3 weeks	Dev + DBA

Priority	Vuln ID	Vulnerability	Recommendation	Effort	Timeline	Owner
High	V-007	Cross-Site Scripting (Reflected/Stored)	Apply output encoding, validate input server-side, use Content Security Policy.	Medium	2 weeks	Dev
Medium	V-003	Exposed Directory (/downloads/)	Restrict directory browsing, move sensitive files out of web root, apply access controls.	Low	1 week	DevOps
Medium	V-008	CSRF (Change Secret / Change Password)	Use anti-CSRF tokens, enforce SameSite cookies, validate referrers.	Medium	2 weeks	Dev
Medium	V-011	Directory Traversal	Sanitize user input (../ filtering), use allowlist for file access, run app with least privilege.	Medium	2 weeks	Dev
Low	V-009	Insecure Direct Object Reference (IDOR)	Add proper authorization checks before accessing objects. Use indirect references (mapping IDs).	Low	2 weeks	Dev
Low	V-012	Information Disclosure (Headers/PHP)	Disable server signature and X-Powered-By in Apache/PHP. Configure ServerTokens Prod.	Low	1 week	SysAdmin

General: Schedule quarterly scans, automate with OWASP ZAP, train staff on secure config.

---

8. Conclusion

The reconnaissance and vulnerability detection phases identified **13 confirmed vulnerabilities** in the bWAPP lab, including **4 critical risks** (SQL Injection, Stored XSS/CSRF, Unrestricted File Upload, and Directory Traversal). These findings highlight significant weaknesses in input validation, access control, and server configuration, which could lead to **database compromise, remote code execution, or persistent account hijacking** if exploited.
Immediate remediation of critical and high-severity issues is strongly recommended to reduce the attack surface and mitigate exploitation risk.
Next Steps:

- Address critical vulnerabilities within 1–2 weeks.
- Re-test after remediation to validate fixes.
- Proceed into the **exploitation and post-exploitation phase by October 1, 2025** to further validate security controls under real-world attack scenarios.

---

9. Appendices

A: Glossary

- **CVSS:** Scoring system for vulnerabilities
- **bWAPP:** Buggy Web App for training
- **Nmap:** Network Mapper, used for port scanning/service discovery.
- **Nikto:** Web server vulnerability scanner.
- **Gobuster:** Directory/file brute-forcing tool.
- **Burp Suite:** Proxy/interceptor for manual testing (we used it lightly for CSRF/XSS POC).
- **CSRF/XSS/SQLi:** expand acronyms at least once in glossary for clarity.

B: References

- OWASP Testing Guide v4 – Industry standard methodology for web application security testing.
- NIST SP 800-115 – Technical Guide to Information Security Testing and Assessment.
- CVE Details (https://cve.mitre.org/) – Reference database for Common Vulnerabilities and Exposures.
- OWASP Top Ten 2021 – Most critical web application security risks.
- Penetration Testing Execution Standard (PTES) – Reconnaissance and vulnerability assessment phases used for alignment.