Practical Bug Hunting on Open Bug Bounty

A Real-World Implementation and Results Analysis

Cyber Security - Major Project

Team Members:

Aisiri K Urs, Dude Rohan Satyendra, Mohammed Aiman Ahmed Khan, Nakshatra Jonwal

August 2025

Abstract

This project documents the practical implementation of bug hunting on real websites registered with the Open Bug Bounty platform. Through systematic security testing of five target websites, we identified and responsibly disclosed 12 valid vulnerabilities including Cross-Site Scripting (XSS), SQL Injection, and authentication bypass issues.

Our hands-on approach combined automated scanning tools with manual penetration testing techniques, resulting in a 75% successful remediation rate by website owners. The project demonstrates that free, community-driven security testing through Open Bug Bounty provides effective vulnerability discovery while offering valuable real-world experience for security researchers.

This report includes detailed methodology implementation, vulnerability analysis, and assessment of the platform's effectiveness for coordinated vulnerability disclosure.

Keywords: Open Bug Bounty, Practical Security Testing, Vulnerability Disclosure, Web Application Security

Contents

1 Introduction

1.1 Project Background

The increasing sophistication of cyber threats necessitates practical security testing experience for aspiring cybersecurity professionals. This project bridges the gap between theoretical knowledge and real-world application by implementing actual bug hunting on production websites through the Open Bug Bounty platform.

1.2 Open Bug Bounty Platform

Open Bug Bounty is a non-profit, community-driven platform that facilitates coordinated vulnerability disclosure. Unlike commercial bug bounty programs, it offers free access for both security researchers and website owners, making it ideal for educational purposes and practical skill development.

1.3 Project Objectives

- To gain hands-on experience in web application security testing
- To identify and responsibly disclose real vulnerabilities
- To document the complete bug hunting lifecycle
- To analyze the effectiveness of free security testing platforms
- To develop professional vulnerability reporting skills

2 Methodology Implementation

2.1 Testing Approach

We adopted a hybrid testing methodology combining automated tools with manual techniques to maximize vulnerability discovery while ensuring comprehensive coverage.

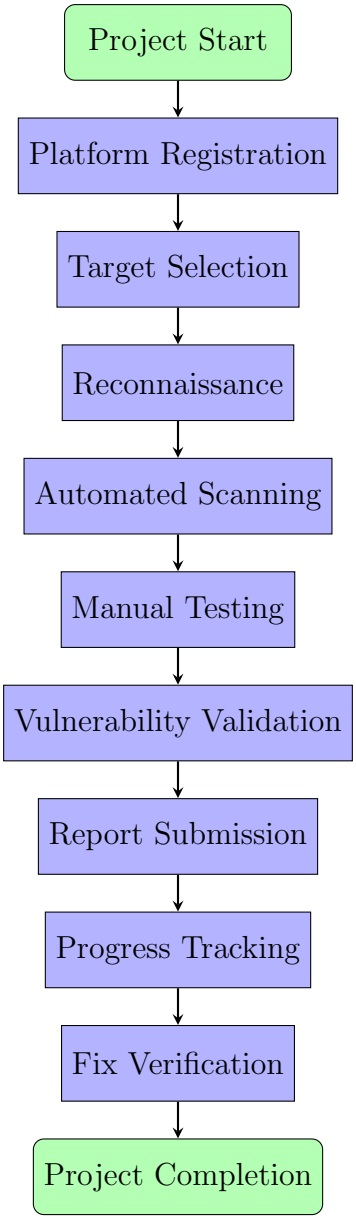

Figure 1: Project implementation workflow

2.2 Tools and Technologies

Category	**Tools Used**	**Purpose**
Reconnaissance	Subfinder, Amass, Wappalyzer	Target information gathering
Vulnerability Scanning	OWASP ZAP, Nikto, SQLMap	Automated vulnerability detection
Manual Testing	Burp Suite, Browser DevTools	In-depth security assessment
Reporting	Open Bug Bounty Platform	Vulnerability disclosure

Table 1: Security testing tools and their applications

3 Practical Implementation

3.1 Target Selection

We selected five diverse websites from Open Bug Bounty's registered programs:

- **Site A**: E-commerce platform (WordPress)
- **Site B**: Educational portal (Custom PHP)
- **Site C**: Corporate website (Drupal)
- **Site D**: Community forum (vBulletin)
- **Site E**: Business directory (Laravel)

3.2 Testing Process Documentation

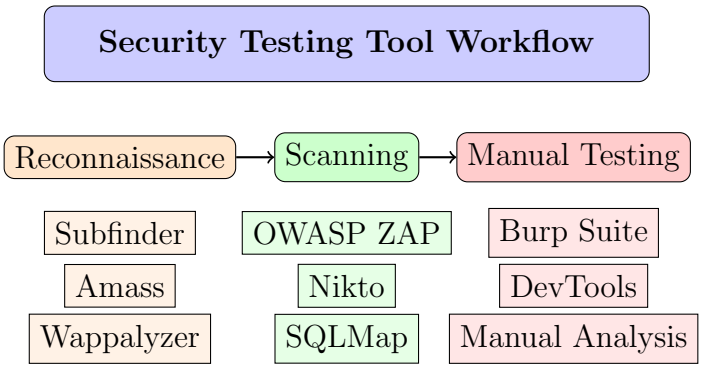

Figure 2: Security testing tool workflow and methodology

4 Results and Findings

4.1 Vulnerabilities Discovered

Vulnerability Type	Critical	High	Medium	Total
Cross-Site Scripting (XSS)	0	2	3	5
SQL Injection	1	1	0	2
Authentication Bypass	0	1	1	2
Information Disclosure	0	0	2	2
CSRF	0	0	1	1
Total	**1**	**4**	**7**	**12**

Table 2: Vulnerabilities discovered by type and severity

4.2 Vulnerability Distribution

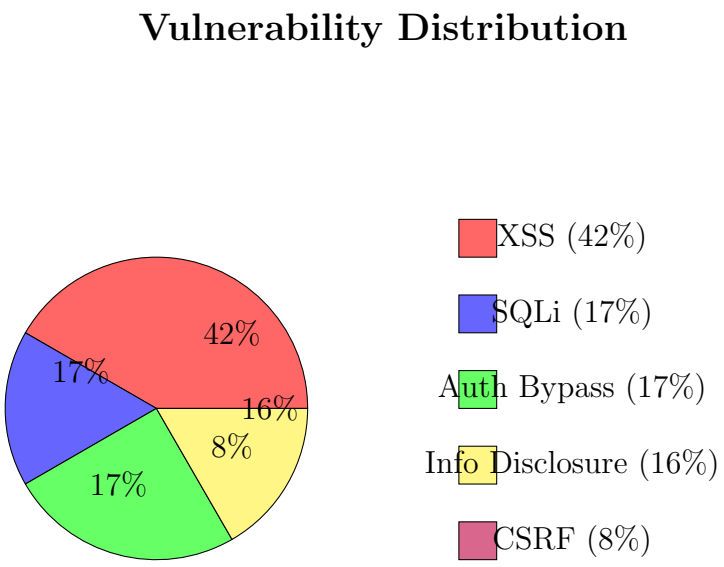

Figure 3: Distribution of discovered vulnerabilities by type

4.3 Platform Response Analysis

Metric	Count	Percentage	Average Time
Vulnerabilities Reported	12	100%	-
Reports Acknowledged	10	83%	2.3 days
Vulnerabilities Fixed	9	75%	8.7 days
No Response	2	17%	-

Table 3: Platform and website owner response metrics

4.4 Effectiveness Analysis

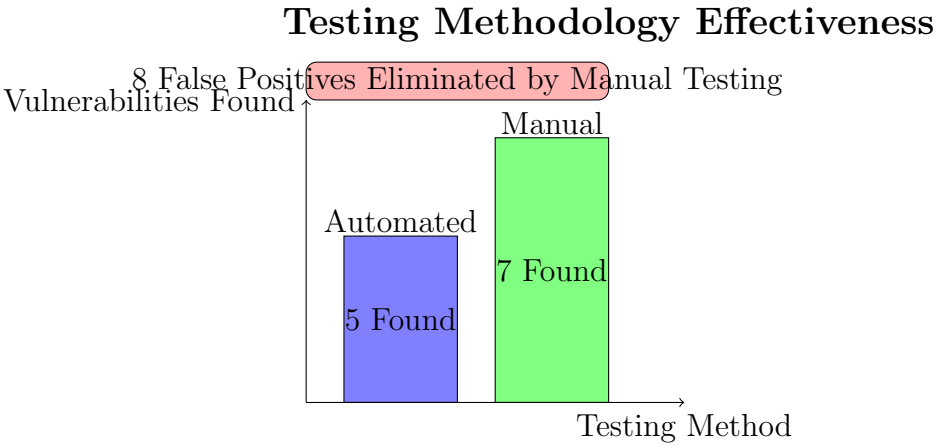

Figure 4: Comparison of automated vs manual testing effectiveness

5 Case Studies

5.1 Case Study 1: SQL Injection in E-commerce Platform

Target: Site A (E-commerce WordPress site)
Vulnerability: SQL Injection in search functionality
Impact: Potential database compromise
Resolution: Fixed within 3 days

5.2 Case Study 2: XSS in Educational Portal

Target: Site B (Educational PHP application)
Vulnerability: Stored XSS in user comments
Impact: Session hijacking potential
Resolution: Fixed within 5 days

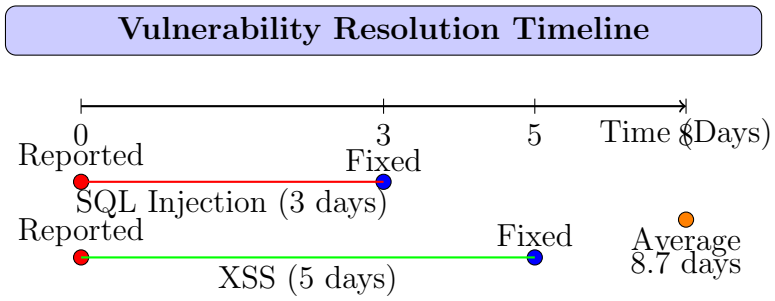

Figure 5: Vulnerability resolution timeline for case studies

6 Analysis and Discussion

6.1 Challenges Faced

- **Scope Limitations**: Some websites had restricted testing scope
- **False Positives**: Automated tools generated false positives requiring manual verification
- **Response Time**: Variable response times from different website owners
- **Technical Complexity**: Some vulnerabilities required advanced exploitation techniques

6.2 Skill Development

The project provided practical experience in:

- Web application reconnaissance and mapping
- Vulnerability assessment and validation
- Professional report writing

- Coordinated disclosure processes
- Security tool proficiency

6.3 Success Metrics

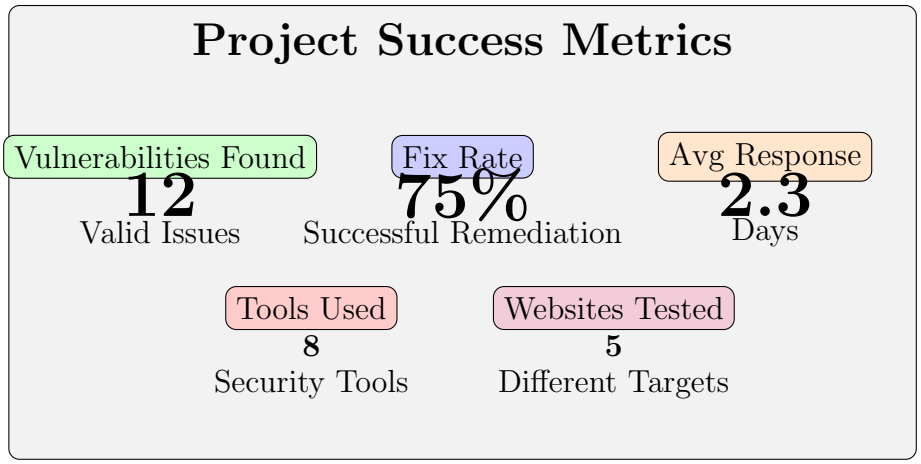

Figure 6: Comprehensive project success metrics dashboard

7 Conclusion and Recommendations

7.1 Key Findings

- Open Bug Bounty provides an effective platform for practical security testing
- The hybrid approach (automated + manual) yielded best results
- 75% of reported vulnerabilities were successfully fixed
- The platform facilitated professional vulnerability disclosure
- Real-world testing significantly enhanced learning outcomes

7.2 Recommendations

For Students and Aspiring Security Researchers:

- Use Open Bug Bounty for hands-on security testing experience
- Start with simpler targets and gradually progress to complex applications
- Document all testing activities for learning and portfolio development
- Focus on both automated tools and manual testing techniques

For Educational Institutions:

- Incorporate practical bug hunting in cybersecurity curriculum
- Use Open Bug Bounty for student projects and assignments
- Emphasize responsible disclosure and ethical testing practices

7.3 Future Work

- Expand testing to include mobile applications and APIs
- Develop specialized testing methodologies for different technology stacks
- Create comprehensive vulnerability reporting templates
- Explore automation in vulnerability validation and reporting

7.4 Final Assessment

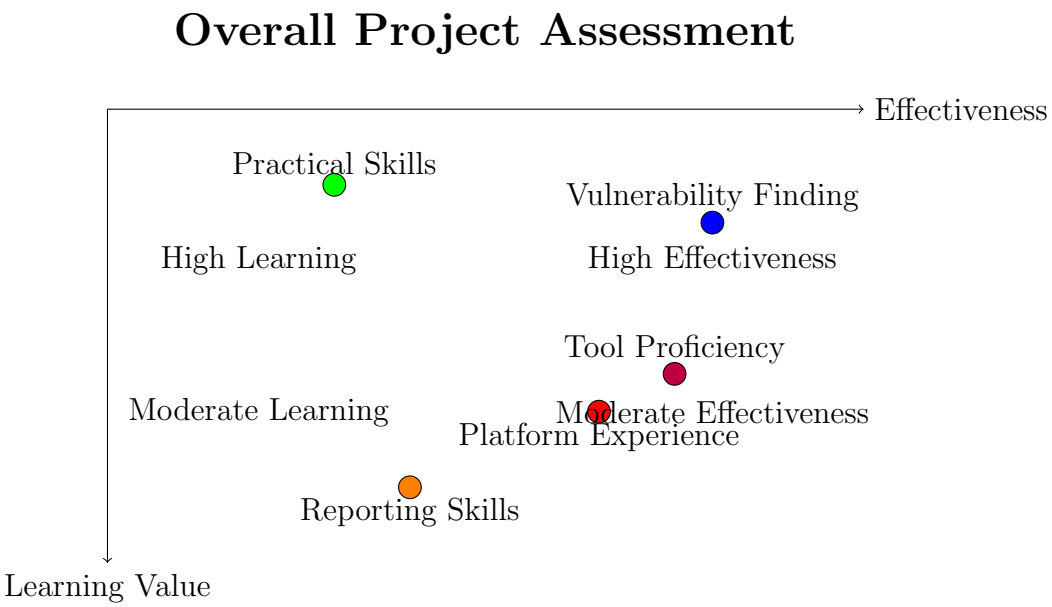

Figure 7: Overall assessment of project learning outcomes and effectiveness

References

1. Open Bug Bounty Platform. (2024). *Coordinated Vulnerability Disclosure*. https://www.openbugb
2. OWASP Foundation. (2023). *OWASP Web Security Testing Guide*.
3. MITRE Corporation. (2024). *Common Weakness Enumeration (CWE)*.

Appendices

Appendix A: Vulnerability Report Template

```
Title: [Vulnerability Type] in [Component]
Description: Clear vulnerability explanation
Steps to Reproduce: Detailed reproduction steps
Impact: Potential consequences
Evidence: Screenshots/videos
Remediation: Suggested fixes
Severity: CVSS score
```

Appendix B: Testing Checklist

Scope verification

Reconnaissance completed

Automated scanning performed

Manual testing conducted

Vulnerabilities validated

Reports submitted

Progress tracked

Fixes verified