Bug Hunting on OpenBugBounty: Project Documentation

Abstract

This project focuses on identifying and reporting vulnerabilities on websites listed on OpenBugBounty. Using systematic security testing methodologies, we aim to enhance the security of these websites by uncovering vulnerabilities such as Cross-Site Scripting (XSS), SQL Injection, and other potential weaknesses. Our findings contribute to a safer online environment and provide learning opportunities for security researchers.

Introduction

The rise in cyber threats demands a proactive approach to online security. This project leverages the OpenBugBounty platform to identify vulnerabilities on selected websites and coordinate responsible disclosure with website owners. By doing so, we enhance cybersecurity awareness, promote safer online services, and gain hands-on experience in penetration testing and vulnerability reporting.

Methodology

1. Target Selection

• Selected a target website from OpenBugBounty based on scope and permissions.

2. Reconnaissance

• Gathered target information such as domain name, hosting environment, and potential entry points using tools like Nmap and Burp Suite.

3. Scanning & Manual Testing

• Performed automated scans using tools like OWASP ZAP for detecting common vulnerabilities.

• Conducted manual tests to uncover issues like XSS, CSRF, and input validation errors.

4. Exploitation

• Verified vulnerabilities’ severity by testing their exploitability. Ensured no destructive exploitation was performed.

5. Reporting

• Documented vulnerabilities in detail (including screenshots) and reported them via OpenBugBounty’s reporting system.

6. Validation & Retesting

• Verified remediation efforts post-report to ensure vulnerabilities were resolved.

Findings

Example Vulnerability: Reflected Cross-Site Scripting (XSS)

• Target: securelogin.example

• Severity: Medium

• Description: Found a reflected XSS vulnerability on the login page, allowing malicious script injection.

• Impact: Could lead to session hijacking, phishing, or data theft.

• Reproduction Steps:

1. Navigate to securelogin.example/login.

2. Inject <script>alert('XSS')</script> into the username field.

3. Observe the alert popup upon form submission.

• Evidence: Screenshots of script execution in the browser.

• Recommendations: Implement input sanitization, use CSP headers, and validate all user input.

Challenges

• Identifying targets within the defined scope without breaching ethical guidelines.

• Time constraints in reporting and verifying fixes.

Conclusion

This project showcased the importance of ethical hacking in cybersecurity. Through the collaboration of security researchers and OpenBugBounty, vulnerabilities were responsibly disclosed, leading to stronger online defenses.

References

• OpenBugBounty Platform: [openbugbounty.org](https://www.openbugbounty.org)

• OWASP Testing Guide: [owasp.org](https://owasp.org)