Vulnerability Assessment Report

Introduction

This is an article about a website security demonstration site. It discusses how to use Acunetix Web Vulnerability Scanner. The article also details how to find potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF). The website is not a real shop, but is an example PHP application that is intentionally vulnerable to web attacks. This allows users to test Acunetix and understand how to protect their websites.The assessment was conducted using a combination of automated and manual techniques.

Methodology

- **Automated scanning:** Acunetix Web Vulnerability Scanner was used to scan the web application for known vulnerabilities.
- **Manual testing:** Manual testing was conducted to verify the results of the automated scan and to identify any additional vulnerabilities.

Objective

The objective of this test was to determine security vulnerabilities in the web server configuration and website running on the server. The tests were carried out assuming the identity of an attacker or with malicious intent. At the same time due care was taken not to harm the web server. This is the vulnerability assessment and penetration testing (VAPT)

Scope

HTML Injection & XSS:

HTML injection is a technique used to modify a web page using non-validated input, while Cross-site Scripting (XSS) is a client-side code injection attack.

The blog explains how the web application was vulnerable to these attacks and provides examples of the impact.

Directory Listing & Improper Error Handling:

The blog describes instances of directory listing and improper error handling flaws, along with the potential risks associated with these vulnerabilities.

Broken Authentication & SQL Injection:

Details of broken authentication attacks and SQL injection vulnerabilities are outlined, including methods used to exploit these vulnerabilities and their implications.

Directory Traversal & Local File Inclusion:

Explanation of directory traversal and local file inclusion vulnerabilities, along with successful execution and outcomes of these attacks.

Business Price Change & Information Disclosure within a Cookie:

Details provided on capturing requests, altering product prices, and gaining access to sensitive information within a cookie, showcasing potential security gaps.

potential impact

Security Professionals: Security professionals can utilize this website to test and demonstrate the effectiveness of security tools, such as web vulnerability scanners, by exploiting the intentionally included vulnerabilities.

Developers: Developers can gain insights into common web application vulnerabilities, including SQL Injections, Cross-site Scripting (XSS), Cross-site Request Forgery (CSRF), and more. They can use this knowledge to improve their understanding of secure coding practices and how to mitigate such vulnerabilities.

Ethical Hackers and Penetration Testers: Ethical hackers and penetration testers can leverage this website for hands-on practice and skill improvement. It provides a safe environment for them to test their manual hacking techniques and strategies.

Security Researchers and Students: Security researchers and students can use this website as a learning resource to study and understand the impact of various web vulnerabilities in a controlled environment.

Acunetix Web Vulnerability Scanner Users: Users of the Acunetix Web Vulnerability Scanner can use this website to test the effectiveness of the scanner and validate its ability to identify and exploit vulnerabilities.

Vulnerability

SQL Injection

Severity: High

Description: The web application is susceptible to SQL injection attacks, allowing an attacker to inject malicious SQL code and execute arbitrary commands on the database.

Potential Impact: An attacker could leverage this vulnerability to steal sensitive data, modify data, or delete data.

Cross-site Scripting (XSS)

Severity: Medium

Description: The web application is vulnerable to XSS attacks, enabling an attacker to inject malicious JavaScript code that gets executed when another user visits the page.

Potential Impact: An attacker could exploit this vulnerability to steal cookies, redirect users to malicious websites, or execute arbitrary commands on the user's computer.

Cross-site Request Forgery (CSRF)

Severity: Medium

Description: The web application is susceptible to CSRF attacks, allowing an attacker to trick a user into clicking on a malicious link, causing the user's browser to submit a request to the web application that the user did not intend.

Potential Impact: An attacker could exploit this vulnerability to change a user's password, transfer money from a user's account, or perform other unauthorized actions.

Recommendations

- Patch all known vulnerabilities. This can be done by applying the latest security patches to the web application and its underlying software.
- Implement input validation. Input validation is a technique for ensuring that user-supplied data is safe to use before it is processed by the application.
- Use a web application firewall (WAF). A WAF is a security device that can help to protect web applications from a variety of attacks, including SQL injection, XSS, and CSRF.

Conclusion

The web application is vulnerable to a number of security vulnerabilities. The identified vulnerabilities could allow an attacker to steal sensitive data, modify data, delete data, or perform other unauthorized actions. It is important to remediate the identified vulnerabilities as soon as possible to protect the web application from attack.