Web Security Assessment Report

Target Information

Target URL:	https://example.com
Scan Date:	2025-08-01 06:14:46
Total Alerts:	15
URLs Scanned:	0

Risk Summary

Risk Level	Count
High	1
Medium	1
Low	1
Informational	0

Executive Summary

The security assessment identified 15 potential security issues. Among these, 1 are classified as high risk and require immediate attention. Additionally, 1 medium-risk vulnerabilities were found that should be addressed in the near term. This report provides detailed information about each finding along with recommended remediation steps.

Key Findings

Critical Issues Found:

• SQL Injection

Vulnerability Details

High Risk Vulnerabilities

1. SQL Injection

Description: SQL injection vulnerabilities allow an attacker to interfere with the queries that an applicatio

Risk Level: High

Confidence: High

URL: https://example.com/login

Parameter: username

Recommended Solution:

Use parameterized queries and input validation.

Medium Risk Vulnerabilities

1. Cross Site Scripting (XSS)

Description: Cross-site scripting (XSS) is a type of computer security vulnerability typically found in web

Risk Level: Medium

Confidence: High

URL: https://example.com/search

Parameter: q

Recommended Solution:

Encode all user input and use Content Security Policy.

Low Risk Vulnerabilities

1. Information Disclosure

Description: The web server is configured to expose sensitive information.

Risk Level: Low

Confidence: Medium

URL: https://example.com/

Parameter:

Recommended Solution:

Configure the web server to hide version information.

Security Recommendations

General Security Best Practices:

• Implement proper input validation and sanitization
• Use HTTPS for all communications
• Implement Content Security Policy (CSP) headers
• Regular security updates and patches
• Implement proper authentication and authorization
• Use secure coding practices
• Regular security assessments and penetration testing
• Implement proper logging and monitoring

Specific Recommendations Based on Findings:

• Use parameterized queries and stored procedures
• Remove sensitive information from error messages and headers
• Implement proper input validation and output encoding