Web Application Vulnerability Scan Report

Target: https://www.bugbitething.com/?srsltid=AfmBOooy-PD4B6ZgeJq-VkHGSHSgmTXUXlWcgc_XwJVBkDhyaxESMEHA

Scan Date: 2025-04-18 00:42:07

Scan Mode:

Vulnerability Summary:

High: 12

Medium: 99

Low: 15

Detailed Findings:

SQL Injection

High: Possible SQL Injection detected with payload: '

Recommendation: Use parameterized queries and prepared statements. Implement input validation, escaping, and ORM frameworks. Apply the principle of least privilege for database accounts.

High: Possible SQL Injection detected with payload: ' OR '1'='1

Recommendation: Use parameterized queries and prepared statements. Implement input validation, escaping, and ORM frameworks. Apply the principle of least privilege for database accounts.

High: Possible SQL Injection detected with payload: ' OR '1'='1' --

Recommendation: Use parameterized queries and prepared statements. Implement input validation, escaping, and ORM frameworks. Apply the principle of least privilege for database accounts.

High: Possible SQL Injection detected with payload: ' OR '1'='1' /*

Recommendation: Use parameterized queries and prepared statements. Implement input validation, escaping, and ORM frameworks. Apply the principle of least privilege for database accounts.

Cross-Site Scripting (XSS)

High: High-risk DOM XSS flow detected: location -> location

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

High: High-risk DOM XSS flow detected: location.href -> location

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

High: DOM XSS vulnerable pattern found: innerHTML with location input

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

High: DOM XSS vulnerable pattern found: setTimeout with location input

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

High: DOM XSS vulnerable pattern found: location assignment with user-controlled input

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

High: DOM XSS vulnerable pattern found: href assignment from URL hash

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and document.write (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and innerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and outerHTML (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and eval (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and Function (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and setTimeout (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and location (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and location.href (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.href (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.search (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both location.pathname (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.referrer (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both document.cookie (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both localStorage (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both sessionStorage (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both addEventListener (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Medium: Potential DOM XSS: Both XMLHttpRequest (source) and document.referrer (sink) found in JavaScript

Recommendation: Avoid using vulnerable JavaScript methods like innerHTML, document.write. Use safe DOM methods like textContent instead. Sanitize data before using it in JavaScript contexts.

Low: No reflected XSS detected.

Recommendation: Implement output encoding specific to the context (HTML, JavaScript, CSS, URL). Use Content Security Policy (CSP) headers. Sanitize all user inputs and validate against whitelist.

Low: No stored XSS detected.

Recommendation: Sanitize user input before storing in the database. Implement context-specific output encoding when displaying stored data. Use CSP headers and consider using HTML Sanitizer libraries.

Other

High: Vulnerable to Heartbleed! Memory leak detected (316 bytes).

Recommendation: Upgrade OpenSSL to a non-vulnerable version (1.0.1g or later). Generate new SSL certificates and keys. Revoke and replace compromised certificates.

Medium: OCSP Stapling not detected

Recommendation: Implement all recommended security headers: Content-Security-Policy, X-Content-Type-Options, X-Frame-Options, Strict-Transport-Security, Referrer-Policy, and Permissions-Policy.

Low: Final URL after redirects: https://www.bugbitething.com/?srsltid=AfmBOooy-PD4B6ZgeJq-VkHGSHSgmTXUXlWcgc_XwJVBkDhyaxESMEHA

Recommendation: Review this issue manually for specific remediation steps.

Low: X-Frame-Options is correctly implemented.

Recommendation: Implement all recommended security headers: Content-Security-Policy, X-Content-Type-Options, X-Frame-Options, Strict-Transport-Security, Referrer-Policy, and Permissions-Policy.

Low: Content-Security-Policy is correctly implemented.

Recommendation: Review this issue manually for specific remediation steps.

Low: X-Content-Type-Options is correctly implemented.

Recommendation: Implement all recommended security headers: Content-Security-Policy, X-Content-Type-Options, X-Frame-Options, Strict-Transport-Security, Referrer-Policy, and Permissions-Policy.

Low: Server Information Leakage Detected: Server: cloudflare

Recommendation: Configure your server to hide version info in headers. Remove unnecessary headers that reveal technology information. Use generic error pages in production.

Low: Unable to grab banner for www.bugbitething.com:80

Recommendation: Review this issue manually for specific remediation steps.

Low: Unable to grab banner for www.bugbitething.com:443

Recommendation: Review this issue manually for specific remediation steps.

Low: Unable to grab banner for www.bugbitething.com:8080

Recommendation: Review this issue manually for specific remediation steps.

Low: Unable to grab banner for www.bugbitething.com:8443

Recommendation: Review this issue manually for specific remediation steps.

Network Security

Medium: Open Ports: 80, 443, 8080, 8443

Recommendation: Review this issue manually for specific remediation steps.

Low: Strict-Transport-Security is correctly implemented.

Recommendation: Review this issue manually for specific remediation steps.

Low: Allowed HTTP Methods: GET, HEAD

Recommendation: Disable unnecessary HTTP methods (PUT, DELETE, TRACE) if not required. Implement proper authentication and authorization for dangerous methods. Use method restrictions in web server configuration.

Security Headers & Cookies

Medium: Cookie '_landing_page' has issues: Missing Secure flag, Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/)

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: Cookie '_orig_referrer' has issues: Missing Secure flag, Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/)

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: Cookie '_shopify_s' has issues: Missing Secure flag, Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/)

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: Cookie '_shopify_y' has issues: Missing Secure flag, Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/)

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: Cookie '_tracking_consent' has issues: Missing Secure flag, Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: Cookie 'cart_currency' has issues: Missing Secure flag, Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/)

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: Cookie 'localization' has issues: Missing Secure flag, Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/)

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: Cookie 'secure_customer_sig' has issues: Missing HttpOnly flag, Missing SameSite attribute, Broad cookie scope (path=/)

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

Medium: No CSRF protection token detected in cookies

Recommendation: Set Secure, HttpOnly, and SameSite=Strict attributes for sensitive cookies. Implement proper cookie expiration. Use CSRF tokens for sensitive operations. Limit cookie scope with Path attribute.

SSL/TLS Configuration

High: Perfect Forward Secrecy (PFS) is NOT supported. Server does not prioritize ECDHE or DHE cipher suites.

Recommendation: Configure your server to use DHE or ECDHE cipher suites for Perfect Forward Secrecy. Ensure DH key size is at least 2048 bits. Prioritize ECDHE over DHE for better performance.

Low: SSL Certificate valid until: 2025-06-26 05:05:04

Recommendation: Ensure certificates are valid and not expired. Use certificates from trusted CAs. Implement proper certificate chain. Configure OCSP stapling and certificate transparency.

Low: Secure Protocol in use: TLSv1.3

Recommendation: Review this issue manually for specific remediation steps.

General Remediation Guidance

Input Validation
Implement strict input validation for all user-supplied data using whitelisting approach.

Authentication
Use multi-factor authentication and implement proper session management.

Secure Configuration
Harden web servers, frameworks, and database systems.

Logging & Monitoring
Establish comprehensive logging and monitoring for security events.

Security Headers
Implement all recommended security headers including CSP.

Output Encoding
Apply context-specific output encoding for all data displayed to users.

Authorization
Apply principle of least privilege and role-based access control.

Error Handling
Implement custom error pages that don't reveal sensitive information.

Data Protection
Encrypt sensitive data both in transit and at rest.

Security Testing
Conduct regular security assessments and penetration testing.