TECHNICAL SECURITY REPORT

Detailed Vulnerability Analysis
Target: sbsinformatique.com | Generated: 2025-12-17T17:25:32.061Z

VULNERABILITY DETAILS

CRITICAL: SQL Injection Vulnerability

CVE: CVE-2021-44228
The application is vulnerable to SQL injection attacks through user input fields.

HIGH: Cross-Site Scripting (XSS)

Reflected XSS vulnerability found in search parameters.

CRITICAL: Remote Code Execution (Log4j)

CVE: CVE-2021-44228
The application may be vulnerable to Log4Shell remote code execution.

MEDIUM: SSL/TLS Configuration Weakness

Server supports weak cipher suites that could be exploited.

MEDIUM: Cross-Site Request Forgery (CSRF)

Forms lack CSRF tokens, allowing cross-site request forgery attacks.

LOW: Directory Listing Enabled

Web server allows directory listing which can expose sensitive files.

HIGH: Broken Authentication - Session Fixation

Session tokens are not regenerated after authentication.

HIGH: Server-Side Request Forgery (SSRF)

Application allows server-side requests to arbitrary URLs.

HIGH: Insecure Direct Object Reference (IDOR)

API endpoints expose direct references to internal objects without proper authorization checks.

EXPLOITATION EVIDENCE

' EXPLOITED: SQL Injection Payload Injection
```
Successfully extracted database schema
```

' EXPLOITED: XSS Payload Delivery
```
Cookie theft payload executed successfully
```

' EXPLOITED: Log4j JNDI Injection
```
LDAP callback received - RCE confirmed
```

' EXPLOITED: SSL/TLS Downgrade Attack
```
Successfully downgraded connection to TLS 1.0
```

' Not Exploited: CSRF Token Bypass

' EXPLOITED: Directory Traversal
```
Accessed /etc/passwd via path traversal
```

' Not Exploited: Generic Exploit Attempt

' Not Exploited: Generic Exploit Attempt

' Not Exploited: Generic Exploit Attempt

REMEDIATION CODE SNIPPETS

SQL Injection Vulnerability

Language: text | Effort: 2-4 hours

```
Use parameterized queries or prepared statements. Implement input validation and sanitization.
```

Implementation:

Remote Code Execution (Log4j)

Language: text | Effort: 2-4 hours

```
Update Log4j to version 2.17.0 or later. Apply temporary mitigations if update not immediately possible.
```

Implementation:

Cross-Site Scripting (XSS)

Language: text | Effort: 2-4 hours

```
Implement Content-Security-Policy headers and sanitize all user inputs before rendering.
```

Implementation:

Broken Authentication - Session Fixation

Language: text | Effort: 2-4 hours

```
Regenerate session tokens upon authentication. Implement secure session management.
```

Implementation:

Server-Side Request Forgery (SSRF)

Language: text | Effort: 2-4 hours

```
Implement URL allowlisting. Block requests to internal IP ranges and cloud metadata endpoints.
```

Implementation:

Insecure Direct Object Reference (IDOR)

Language: text | Effort: 2-4 hours

```

```

```
Implement proper authorization checks for all object references. Use indirect reference maps.
```

Implementation:

SSL/TLS Configuration Weakness

Language: nginx | Effort: 1-2 hours

```
ssl_protocols TLSv1.2 TLSv1.3;
ssl_ciphers ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256;
ssl_prefer_server_ciphers off;
```

Implementation: Update SSL/TLS configuration to use only modern protocols and cipher suites.

Cross-Site Request Forgery (CSRF)

Language: text | Effort: 2-4 hours

```
# Remediation steps for: Cross-Site Request Forgery (CSRF)
1. Review the vulnerability details
2. Apply vendor patches if available
3. Implement compensating controls
4. Verify fix through re-testing
```

Implementation: Follow vendor recommendations and security best practices for this vulnerability type.

Directory Listing Enabled

Language: text | Effort: 2-4 hours

```
# Remediation steps for: Directory Listing Enabled
1. Review the vulnerability details
2. Apply vendor patches if available
3. Implement compensating controls
4. Verify fix through re-testing
```

Implementation: Follow vendor recommendations and security best practices for this vulnerability type.