SecureThreadVMS

SECURITY ASSESSMENT REPORT

STATIC APPLICATION SECURITY TESTING (SAST)

Target Repository:
vulnerable-node

Assessment Date:
21 December 2025

Report Reference ID:
`SCR-VULNERABLE-NODE-179`

CONFIDENTIAL DOCUMENT

This document contains proprietary and confidential information.
Do not distribute without authorization.

Table of Contents

List of Figures

Executive Summary

Sandbox Security conducted a comprehensive secure code review of **vulnerable-node** using advanced static application security testing (SAST) techniques. This assessment identified **20** security findings across **23** analyzed files.

Severity	Critical	High	Medium	Low
Count	4	13	2	1

Overall Risk Assessment: Medium

0.1 Detailed Findings

0.1.1 Critical Severity Vulnerabilities

VULN-017: XSS - Function Constructor with User Data

Severity: CRITICAL

Category: xss

CWE ID: CWE-95

File: public/js/jquery.js:2

Description:

Detects dangerous Function constructor usage**Vulnerable Code:**

```
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?
    module.exports=a.document?b(a,!0):function(a){if(!a.document)throw
    new Error("jQuery requires a window with a document");return b(a)}:b
    (a)}("undefined"!=typeof window?window:this,function(a,b){var c=[],d
    =c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h.
    hasOwnProperty,k={},l="1.11.1",m=function(a,b){return new m.fn.init(
    a,b)},n=/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,o=/^-ms-/,p=/-([\da-z])
    /gi,q=function(a,b){return
%
```

Recommendation:

“‘json
{
"explanation": "The analysis appears to be based on a misunderstanding. The provided code is a minified version of jQuery v1.11.1, which is a widely-used, legitimate JavaScript library. The reported vulnerabilities (XSS via Function Constructor and innerHTML assignment) are likely false positives from automated scanning tools that misinterpret the minified code patterns. jQuery itself contains security features when used properly, though older versions may have known vulnerabilitie

VULN-018: XSS - Function Constructor with User Data

Severity: CRITICAL

Category: xss

CWE ID: CWE-95

File: public/js/jquery.js:4

Description:

Detects dangerous Function constructor usage**Vulnerable Code:**

```
},cur:function(){var a=Zb.propHooks[this.prop];return a&&a.get?a.get(this):Zb.propHooks._default.get(this)},run:function(a){var b,c=Zb.propHooks[this.prop];return this.pos=b=this.options.duration?m.easing[this.easing](a,this.options.duration*a,0,1,this.options.duration):a,this.now=(this.end-this.start)*b+this.start,this.options.step&&this.options.step.call(this.elem,this.now,this),c&&c.set?c.set(this):Zb.propHooks._default.set(this),this}},Zb.prototype.init.prototype=Zb.prototype,Zb.propHooks={_
%
```

Recommendation:

“‘json
{
"explanation": "The analysis appears to be based on a misunderstanding. The provided code is a minified version of jQuery v1.11.1, which is a widely-used, legitimate JavaScript library. The reported vulnerabilities (XSS via Function Constructor and innerHTML assignment) are likely false positives from automated scanning tools that misinterpret the minified code patterns. jQuery itself contains security features when used properly, though older versions may have known vulnerabilitie

VULN-019: XSS - innerHTML Assignment with User Input

Severity: CRITICAL

Category: xss

CWE ID: CWE-79

File: public/js/jquery.js:2

Description:

Detects dangerous innerHTML assignment with untrusted data**Vulnerable Code:**

```
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?
    module.exports=a.document?b(a,!0):function(a){if(!a.document)throw
    new Error("jQuery requires a window with a document");return b(a)}:b
    (a)}("undefined"!=typeof window?window:this,function(a,b){var c=[],d
    =c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h.
    hasOwnProperty,k={},l="1.11.1",m=function(a,b){return new m.fn.init(
    a,b)},n=/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,o=/^-ms-/,p=/-([\da-z])
    /gi,q=function(a,b){return
%
```

Recommendation:

“‘json
{
"explanation": "The analysis appears to be based on a misunderstanding. The provided code is a minified version of jQuery v1.11.1, which is a widely-used, legitimate JavaScript library. The reported vulnerabilities (XSS via Function Constructor and innerHTML assignment) are likely false positives from automated scanning tools that misinterpret the minified code patterns. jQuery itself contains security features when used properly, though older versions may have known vulnerabilitie

VULN-020: Crypto - Data At Rest Not Encrypted

Severity: CRITICAL

Category: cryptography

CWE ID: CWE-311

File: model/init_db.js:15

Description:

Detects data stored without encryption**Vulnerable Code:**

```
    db.one('CREATE TABLE users(name VARCHAR(100) PRIMARY KEY, password VARCHAR(50));')
%
```

Recommendation:

“‘json
{
"explanation": "The critical vulnerability identified is that user passwords are being stored in plaintext in the database. This is extremely dangerous because:\n\n1. **Data Breach Impact**: If the database is compromised, attackers immediately gain access to all user credentials without any additional effort.\n2. **Credential Reuse Risk**: Many users reuse passwords across multiple services, so plaintext passwords can lead to account takeover on other platforms.\n3. **Regulatory N

0.1.2 High Severity Vulnerabilities

VULN-004: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: dummy.js:20

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
        "price": parseInt(Math.random() * 100),
%
```

Recommendation:

Review and fix the cryptography issue in dummy.js

VULN-005: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: dummy.js:26

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
      "price": parseInt(Math.random() * 100),
%
```

Recommendation:

Review and fix the cryptography issue in dummy.js

VULN-006: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: public/js/bootstrap.js:1668

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
    do prefix += ~~(Math.random() * 1000000)
%
```

Recommendation:

“‘json
{
"explanation": "The vulnerability at line 1668 involves the use of Math.random() for cryptographic purposes, which is insecure. Math.random() generates pseudo-random numbers using a deterministic algorithm that is not cryptographically secure. Attackers can potentially predict the generated values, compromising security in scenarios requiring true randomness (such as generating tokens, session IDs, or cryptographic keys). This vulnerability is rated HIGH because predictable random

VULN-007: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: public/js/jquery.js:2

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?
    module.exports=a.document?b(a,!0):function(a){if(!a.document)throw
    new Error("jQuery requires a window with a document");return b(a)}:b
    (a)}("undefined"!=typeof window?window:this,function(a,b){var c=[],d
    =c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h.
    hasOwnProperty,k={},l="1.11.1",m=function(a,b){return new m.fn.init(
    a,b)},n=/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,o=/^-ms-/,p=/-([\da-z])
    /gi,q=function(a,b){return
%
```

Recommendation:

Review and fix the cryptography issue in public/js/jquery.js

VULN-008: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: public/js/jquery.js:2

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.document)throw new Error("jQuery requires a window with a document");return b(a)}:b(a)}("undefined"!=typeof window?window:this,function(a,b){var c=[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h.hasOwnProperty,k={},l="1.11.1",m=function(a,b){return new m.fn.init(a,b)},n=/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,o=/^-ms-/,p=/-([\da-z])/gi,q=function(a,b){return
%
```

Recommendation:

Review and fix the cryptography issue in public/js/jquery.js

VULN-009: XSS - React createElement with User Props

Severity: HIGH

Category: xss

CWE ID: CWE-79

File: public/js/jquery.js:3

Description:

Detects React.createElement with unsanitized props**Vulnerable Code:**

```
if(k&&j[k]&&(e||j[k].data)||void 0!==d||"string"!=typeof b)return k||(k=i?a[h]=c.pop()||m.guid++:h),j[k]||(j[k]=i?{}:{toJSON:m.noop}),("object"==typeof b||"function"==typeof b)&&(e?j[k]=m.extend(j[k],b):j[k].data=m.extend(j[k].data,b)),g=j[k],e||(g.data||(g.data={}),g=g.data),void 0!==d&&(g[m.camelCase(b)]=d),"string"==typeof b?(f=g[b],null==f&&(f=g[m.camelCase(b)])):f=g,f}}function R(a,b,c){if(m.acceptData(a)){var d,e,f=a.nodeType,g=f?m.cache:a,h=f?a[m.expando]:m.expando;if(g[h]){if(b&&(d=c?g[h
%
```

Recommendation:

Review and fix the xss issue in public/js/jquery.js

VULN-010: GraphQL - Sensitive Data in Error Messages

Severity: HIGH

Category: graphql_security

CWE ID: CWE-209

File: public/js/jquery.js:2

Description:

Detects sensitive data exposure in GraphQL errors**Vulnerable Code:**

```
!function(a,b){"object"==typeof module&&"object"==typeof module.exports?module.exports=a.document?b(a,!0):function(a){if(!a.document)throw new Error("jQuery requires a window with a document");return b(a)}:b(a)}("undefined"!=typeof window?window:this,function(a,b){var c=[],d=c.slice,e=c.concat,f=c.push,g=c.indexOf,h={},i=h.toString,j=h.hasOwnProperty,k={},l="1.11.1",m=function(a,b){return new m.fn.init(a,b)},n=/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,o=/^-ms-/,p=/-([\da-z])/gi,q=function(a,b){return
%
```

Recommendation:

Review and fix the graphql_security issue in public/js/jquery.js

VULN-011: Vertical Access Control - Test Account in Production

Severity: HIGH

Category: access_control

CWE ID: CWE-489

File: dummy.js:7

Description:

Detects test or demo accounts in production code**Vulnerable Code:**

```
      "username": "admin",
%
```

Recommendation:

Review and fix the access_control issue in dummy.js

VULN-012: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: dummy.js:50

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
        "price": parseInt(Math.random() * 100),
%
```

Recommendation:

“‘json
{
"explanation": "The code contains multiple instances of insecure random number generation using Math.random(), which is cryptographically weak and predictable. Math.random() generates pseudo-random numbers using a deterministic algorithm that is not suitable for security-sensitive applications. Attackers can potentially predict the generated values, leading to vulnerabilities in scenarios where randomness is critical for security (e.g., session tokens, cryptographic keys, or sensit

VULN-013: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: dummy.js:56

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
        "price": parseInt(Math.random() * 100),
%
```

Recommendation:

"'json
{
"explanation": "The code contains multiple instances of insecure random number generation using Math.random(), which is cryptographically weak and predictable. Math.random() generates pseudo-random numbers using a deterministic algorithm that is not suitable for security-sensitive applications. Attackers can potentially predict the generated values, leading to vulnerabilities in scenarios where randomness is critical for security (e.g., session tokens, cryptographic keys, or sensit

VULN-014: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: dummy.js:62

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
        "price": parseInt(Math.random() * 100),
%
```

Recommendation:

“‘json
{
"explanation": "The code contains multiple instances of insecure random number generation using Math.random(), which is cryptographically weak and predictable. Math.random() generates pseudo-random numbers using a deterministic algorithm that is not suitable for security-sensitive applications. Attackers can potentially predict the generated values, leading to vulnerabilities in scenarios where randomness is critical for security (e.g., session tokens, cryptographic keys, or sensit

VULN-015: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: public/js/bootstrap.min.js:6

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
if("undefined"==typeof jQuery)throw new Error("Bootstrap’s JavaScript requires jQuery");+function(a){"use strict";var b=a.fn.jquery.split(" ")[0].split(".");if(b[0]<2&&b[1]<9||1==b[0]&&9==b[1]&&b[2]<1||b[0]>2)throw new Error("Bootstrap’s JavaScript requires jQuery version 1.9.1 or higher, but lower than version 3")}(jQuery),+function(a){"use strict";function b(){var a=document.createElement("bootstrap"),b={WebkitTransition:"webkitTransitionEnd",MozTransition:"transitionend",OTransition:"oTransit
%
```

Recommendation:

“‘json
{
"explanation": "The vulnerability report incorrectly identifies a high-severity crypto vulnerability in Bootstrap v3.3.6’s minified JavaScript file. After thorough analysis, I can confirm this is a false positive detection. The minified Bootstrap code shown does not contain any cryptographic functions, random number generation, or security-sensitive operations. Bootstrap 3.3.6 is a front-end framework for building responsive web interfaces and does not implement cryptographic funct

VULN-016: Crypto - Insecure Random Number Generator

Severity: HIGH

Category: cryptography

CWE ID: CWE-338

File: dummy.js:32

Description:

Detects usage of insecure random number generators**Vulnerable Code:**

```
        "price": parseInt(Math.random() * 100),
%
```

Recommendation:

“‘json
{
"explanation": "The code contains multiple instances of insecure random number generation using Math.random(), which is cryptographically weak and predictable. Math.random() generates pseudo-random numbers using a deterministic algorithm that is not suitable for security-sensitive applications. Attackers can potentially predict the generated values, leading to vulnerabilities in scenarios where randomness is critical for security (e.g., session tokens, cryptographic keys, or sensit

0.1.3 Medium Severity Vulnerabilities

VULN-001: React - Uncontrolled Component Input

Severity: MEDIUM

Category: react_security

CWE ID: CWE-20

File: public/js/jquery.js:3

Description:

Detects uncontrolled form inputs with defaultValue from props**Vulnerable Code:**

```
if(k&&j[k]&&(e||j[k].data)||void 0!==d||"string"!=typeof b)return k||(k=
    i?a[h]=c.pop()||m.guid++:h),j[k]||(j[k]=i?{}:{toJSON:m.noop}),("
    object"==typeof b||"function"==typeof b)&&(e?j[k]=m.extend(j[k],b):j
    [k].data=m.extend(j[k].data,b)),g=j[k],e||(g.data||(g.data={}),g=g.
    data),void 0!==d&&(g[m.camelCase(b)]=d),"string"==typeof b?(f=g[b],
    null==f&&(f=g[m.camelCase(b)])):f=g,f}}function R(a,b,c){if(m.
    acceptData(a)){var d,e,f=a.nodeType,g=f?m.cache:a,h=f?a[m.expando]:m
    .expando;if(g[h]){if(b&&(d=c?g[h
%
```

Recommendation:

Review and fix the react_security issue in public/js/jquery.js

VULN-002: Data Exposure - Internal IP in Response

Severity: MEDIUM

Category: data_exposure

CWE ID: CWE-200

File: config.js:12

Description:

Detects internal IP addresses exposed in responses**Vulnerable Code:**

```
        "server": "postgres://postgres:postgres@10.211.55.70",
%
```

Recommendation:

Review and fix the data_exposure issue in config.js

0.1.4 Low Severity Vulnerabilities

VULN-003: Docker - ADD Instead of COPY

Severity: LOW

Category: docker_security

CWE ID: CWE-710

File: services/postgresql/Dockerfile:5

Description:

Detects use of ADD instead of COPY**Vulnerable Code:**

```
ADD init.sql /docker-entrypoint-initdb.d/
%
```

Recommendation:

Review and fix the docker_security issue in services/postgresql/Dockerfile