CodeShuriken

Security Assessment Report

Repository: https://github.com/example/test-repo.git

Generated: 2025-08-07T06:06:47.366Z

Report ID: test_20250807_direct_api

Table of Contents

Preface

About This Report

This automated security assessment report has been generated to provide comprehensive insights into the security posture of your software repository. The analysis includes dependency vulnerabilities, code quality metrics, and actionable security recommendations.

Methodology

Our security assessment employs multiple scanning techniques including:

- *Static code analysis for vulnerability detection*
- *Dependency vulnerability scanning against known CVE databases*
- *Software Bill of Materials (SBOM) generation and analysis*
- *Code quality and security best practices evaluation*

Scope and Limitations

This report covers static analysis findings and known vulnerabilities in dependencies. It does not include dynamic testing, manual code review findings, or infrastructure security assessments. The analysis is based on the repository state at the time of scanning.

How to Use This Report

Review the Executive Summary for high-level findings, then proceed to detailed sections. Prioritize critical and high-severity vulnerabilities for immediate remediation. Use the recommendations section to improve your overall security posture.

Executive Summary

This report was prepared by automated security scanning to review aspects of the security and integrity of the repository **https://github.com/example/test-repo.git**.

This report identifies potential security weaknesses and vulnerabilities found through static code review and searches of public vulnerability sources. The analysis focused particularly on dependency vulnerabilities that could be exploited to alter system behavior, access critical data, or conduct denial of service attacks.

Scan Type:	complete	**Status:**	completed
Repository:	https://github.com/example/test-repo.git		
Branch:	main	**Completed:**	2025-08-07T06:06:47.366Z

Key Findings Summary:

- Repository contains 10 files with 10 successfully processed
- Scan completion rate: 100%
- **Total vulnerabilities found:**
    - Critical:
    - High:
    - Medium:
    - Low:
- Primary languages identified: JavaScript (70%), HTML (30%)

Last update:	2025-08-07T06:06:47.366Z	**Status:**	completed
Version:	1.0	**Repository:**	https://github.com/example/test-repo.git

1 Repository Information

1.1 Language Distribution

Language	Percentage
JavaScript	70%
HTML	30%

1.2 File Types Analysis

File Extension	Count
.js	7
.html	3

2 Security Analysis & Vulnerability Assessment

2.0 test.js Analysis

File: src/test.js

Type: CODE

Test vulnerability analysis for demonstration

3 Code Metrics

Total Lines of Code:	1000
Files Processed:	10/10
Completion:	100%

4 Recommendations

Immediate Actions Required:

- Review and address all critical and high severity vulnerabilities
- Update all outdated dependencies to their latest stable versions
- Implement proper dependency management practices

Long-term Security Improvements:

- Integrate automated security scanning into CI/CD pipeline
- Regular security code reviews
- Monitor security advisories for used dependencies
- Implement secure coding practices

5 Summary

Assessment Results Overview

Scan Type: complete

Status: completed

Files: 10/10

Progress: 100%

Total Vulnerabilities:

Vulnerability Breakdown

Critical:

High:

Medium:

Low:

Key Statistics

Repository URL:	https://github.com/example/test-repo.git
Branch Analyzed:	main
Scan Started:	2025-08-07T06:06:47.366Z
Scan Completed:	2025-08-07T06:06:47.366Z
Total Lines of Code:	1000

Primary Languages

JavaScript: 70%

HTML: 30%

Next Steps

1. **Immediate:** Address any critical or high-severity vulnerabilities identified
2. **Short-term:** Review and update outdated dependencies
3. **Long-term:** Implement continuous security monitoring and automated scanning
4. **Process:** Integrate security practices into development workflow

Report generated on 2025-08-07T06:06:47.366Z

Automated Security Assessment Tool