AI-Powered Security Policy Report

Generated: 2025-11-06 22:49:56

Total Vulnerabilities Scanned: 8
• SAST: 8
• SCA: 0
• DAST: 0

LLM Models Used:

• LLaMA 3.3 70B (Groq) - SAST/SCA
• LLaMA 3.1 8B Instant (Groq) - DAST

Policy #1: SAST

Title: Unquoted Attribute Var
Severity: MEDIUM
LLM: LLaMA 3.3 70B

POLICY IDENTIFIER
SP-2024-001: Unquoted Attribute Variable Security Policy

RISK STATEMENT
The presence of unquoted attribute variables in our web application poses a significant risk to our organization's security posture. If exploited, this vulnerability could allow malicious actors to inject custom JavaScript handlers, potentially leading to cross-site scripting (XSS) attacks. This could result in unauthorized access to sensitive data, compromise of user sessions, and damage to our reputation. The affected systems include our web application, and the impacted users are our customers and employees who interact with the application.

COMPLIANCE MAPPING
This policy aligns with the following compliance requirements:
- NIST CSF: PR.DS-5 (Data Protection), DE.CM-8 (Vulnerability Management)
- ISO 27001: A.8.8 (Management of technical vulnerabilities), A.5.10 (Acceptable use of information and other associated assets)
- Industry standards: OWASP Top 10 (A07:2021 - Cross-Site Scripting), CWE-79 (Improper Neutralization of Input During Web Page Generation)

POLICY REQUIREMENTS
To mitigate this risk, the following security controls must be implemented:
- All template variables used as attributes must be properly quoted to prevent injection of malicious code.

- Developers must follow secure coding practices, including input validation and sanitization.
- Success criteria: All template variables are properly quoted, and no XSS vulnerabilities are detected during code reviews and security scans.
- Validation methods: Regular code reviews, security scans, and penetration testing will be conducted to ensure compliance with this policy.

REMEDIATION PLAN
To remediate this vulnerability, the following technical actions are required:
- Review and update the affected code to properly quote template variables.
- Conduct a code review to ensure all template variables are properly quoted.
- Responsible party: Dev Lead
- Timeline: 2 weeks
- Verification steps: Code review, re-scan, and penetration test to ensure no XSS vulnerabilities are present.

MONITORING AND DETECTION
To detect similar vulnerabilities in the future, the following measures will be implemented:
- Regular security scans and code reviews will be conducted to identify potential vulnerabilities.
- Logging and alerting requirements: All security-related events will be logged and alerts will be sent to the security team for review and action.
- Continuous monitoring strategies: Our web application will be continuously monitored for signs of XSS attacks, and our security team will respond promptly to any detected incidents.

Policy #2: SAST

Title: Unquoted Attribute Var
Severity: MEDIUM
LLM: LLaMA 3.3 70B

POLICY IDENTIFIER
Policy ID: SP-2024-001
Policy Title: Prevention of Cross-Site Scripting (XSS) through Unquoted Attribute Variables

RISK STATEMENT
The presence of unquoted attribute variables in our web application poses a significant risk to our organization. If exploited, this vulnerability could allow malicious actors to inject custom JavaScript handlers, potentially leading to unauthorized access, data theft, or disruption of service. The affected systems include our web application, specifically the purchase-basket component, and the data/users at risk are our customers' sensitive information and our organization's reputation.

COMPLIANCE MAPPING
This policy aligns with the following compliance requirements:
- NIST CSF: PR.DS-5 (Data Protection), DE.CM-8 (Vulnerability Management)
- ISO 27001 Annex A controls: A.8.8 (Management of technical vulnerabilities), A.5.10 (Acceptable use of information and other associated assets)
- Industry standards: OWASP A07:2021 (Cross-Site Scripting), CWE-79 (Improper Neutralization of Input During Web Page Generation)

POLICY REQUIREMENTS
To prevent Cross-Site Scripting (XSS) attacks through unquoted attribute variables, the following security controls must be implemented:
- All template expressions must be enclosed in quotes, following the format "{{ expr }}".
- Code reviews must be conducted regularly to identify and remediate similar vulnerabilities.
- Success criteria include the absence of unquoted attribute variables in code reviews and vulnerability scans.
- Validation methods include automated code analysis tools and manual code reviews.

REMEDIATION PLAN
To remediate the identified vulnerability:
- Technical action: Review and fix the unquoted attribute variable in the purchase-basket.component.html file.
- Responsible party: Dev Lead
- Timeline: 2 weeks
- Verification steps: Conduct a code review and re-scan the application using a vulnerability scanner to ensure the fix is effective.

MONITORING AND DETECTION
To detect similar vulnerabilities in the future:
- Implement logging and alerting mechanisms to monitor web application activity for potential XSS attacks.
- Conduct regular code reviews and vulnerability scans to identify unquoted attribute variables and other security weaknesses.
- Utilize continuous monitoring strategies, including automated code analysis tools and manual reviews, to ensure the security and integrity of our web application.

Policy #3: SAST

Title: Unquoted Attribute Var
Severity: MEDIUM
LLM: LLaMA 3.3 70B

POLICY IDENTIFIER
Policy ID: SP-2024-001
Policy Title: Prevention of Cross-Site Scripting (XSS) through Unquoted Attribute Variables

RISK STATEMENT
The presence of unquoted attribute variables in our web application poses a significant risk to our organization. If exploited, an attacker could inject malicious JavaScript code, potentially leading to unauthorized access, data theft, or disruption of service. This vulnerability affects our web application, specifically the search result component, and could impact all users interacting with the application. The potential consequences include compromised user data, reputational damage, and financial loss.

COMPLIANCE MAPPING
This policy aligns with the following compliance requirements:
- NIST CSF: PR.DS-5 (Data Protection), DE.CM-8 (Vulnerability Management)
- ISO 27001: A.8.8 (Management of technical vulnerabilities), A.5.10 (Acceptable use of information and other associated assets), A.8.23 (Web filtering), A.8.3 (Information access restriction)
- Industry standards: OWASP A07:2021 (Cross-Site Scripting), CWE-79 (Improper Neutralization of Input During Web Page Generation)

POLICY REQUIREMENTS
To mitigate the risk of XSS attacks through unquoted attribute variables, the following security controls must be implemented:
- All template variables used as attributes must be properly quoted.
- Regular code reviews must be conducted to identify and fix similar vulnerabilities.
- The development team must adhere to secure coding practices, including input validation and sanitization.
Success criteria include:
- All template variables are properly quoted.
- No XSS vulnerabilities are detected during code reviews or security scans.
Validation methods include regular code reviews, automated security scans, and penetration testing.

REMEDIATION PLAN
To remediate the identified vulnerability:
- Technical action: Quote the template variable in the search-result.component.html file (line 40).
- Responsible party: Dev Lead
- Timeline: 2 weeks
- Verification steps: Code review, automated security scan, and penetration test to ensure the vulnerability is resolved and no similar issues exist.

MONITORING AND DETECTION
To detect similar vulnerabilities in the future:
- Implement regular automated security scans and code reviews.
- Monitor application logs for suspicious activity.
- Conduct periodic penetration testing to identify vulnerabilities.
- Maintain up-to-date knowledge of common web application vulnerabilities, including XSS attacks, and apply appropriate security patches and mitigations.

Policy #4: SAST

Title: Hardcoded Jwt Secret
Severity: MEDIUM
LLM: LLaMA 3.3 70B

POLICY IDENTIFIER
SP-2024-001: Hardcoded Credential Policy

RISK STATEMENT
The use of hardcoded credentials, such as the detected hardcoded JWT secret, poses a significant risk to our organization's security posture. If exploited, this vulnerability could lead to unauthorized access to sensitive data, systems, and applications, resulting in potential data breaches, financial losses, and reputational damage. The affected systems include our web applications, APIs, and backend services, which could compromise the confidentiality, integrity, and availability of our data and services.

COMPLIANCE MAPPING
This policy aligns with the following compliance requirements:
- NIST CSF: PR.DS-5 (Data Protection), PR.IP-1 (Identity and Access Control)
- ISO 27001 Annex A controls: A.8.28 (Secure coding), A.8.1 (User endpoint devices), A.8.5 (Secure authentication)
- Industry standards: OWASP A3:2017 (Sensitive Data Exposure), CWE-798 (Use of Hard-coded Credentials)

POLICY REQUIREMENTS
To mitigate the risk of hardcoded credentials, the following security controls must be implemented:
- All credentials must be stored securely using environment variables or a secure vault/HSM.
- Code reviews must be performed regularly to detect and remove hardcoded credentials.
- Secure coding principles and guidelines must be followed during software development.
- Success criteria: No hardcoded credentials are detected in code reviews and vulnerability scans.
- Validation methods: Regular code reviews, vulnerability scans, and penetration testing.

REMEDIATION PLAN
To remediate the detected hardcoded JWT secret, the following actions must be taken:
- Technical action: Remove the hardcoded JWT secret from the code and store it securely using an environment variable or a secure vault/HSM.
- Responsible party: Dev Lead
- Timeline: 2 weeks
- Verification steps: Code review, re-scan, and penetration testing to ensure the vulnerability is resolved.

MONITORING AND DETECTION
To detect similar vulnerabilities in the future, the following monitoring and detection strategies must be implemented:
- Regular code reviews and vulnerability scans must be performed to detect hardcoded credentials.
- Logging and alerting requirements: All code changes and updates must be logged and reviewed regularly.
- Continuous monitoring strategies: Implement a continuous integration/continuous deployment (CI/CD) pipeline to automate code reviews, vulnerability scans, and penetration testing.

Policy #5: SAST

Title: Unknown Value In Redirect
Severity: MEDIUM
LLM: LLaMA 3.3 70B

POLICY IDENTIFIER
SP-2024-001: Open Redirect Protection Policy

RISK STATEMENT
The Open Redirect vulnerability poses a significant risk to our organization's security posture. If exploited, an attacker could redirect users to malicious websites, potentially leading to phishing attacks, malware infections, or unauthorized access to sensitive information. This vulnerability affects our web applications, putting our users' data and systems at risk. The potential consequences include reputational damage, financial loss, and compromised user trust.

COMPLIANCE MAPPING
This policy aligns with the following compliance requirements:
- NIST CSF: PR.DS-5 (Data Protection), DE.AE-2 (Anomaly Detection)
- ISO 27001 Annex A controls: A.8.23 (Web filtering), A.7.2 (Physical entry), A.5.34 (Privacy and protection of personal identifiable information), A.8.33 (Test information), A.8.3 (Information access restriction)
- Industry standards: OWASP, CWE-601 (URL Redirection to Untrusted Site)

POLICY REQUIREMENTS
To mitigate the Open Redirect vulnerability, the following security controls must be implemented:
- Validate and sanitize all user-inputted URLs to prevent unauthorized redirects
- Implement a whitelist of allowed redirect URLs
- Use a secure redirect mechanism, such as a redirect token or a secure token-based system
- Conduct regular security testing and code reviews to identify and address potential vulnerabilities
Success criteria include:
- No unauthorized redirects are allowed
- All user-inputted URLs are properly validated and sanitized
- The whitelist of allowed redirect URLs is regularly reviewed and updated

REMEDIATION PLAN
To remediate the Open Redirect vulnerability, the following technical actions are required:
- Review and update the `redirect.ts` file to validate and sanitize user-inputted URLs
- Implement a whitelist of allowed redirect URLs
- Conduct a code review to ensure the secure redirect mechanism is properly implemented
The responsible party for this remediation is the Development Lead. The timeline for completion is 2 weeks. Verification steps include a code review, re-scan, and penetration test to ensure the vulnerability is fully addressed.

MONITORING AND DETECTION
To detect similar vulnerabilities in the future, the following monitoring and detection strategies will be implemented:
- Regular security testing and code reviews
- Logging and alerting for suspicious redirect activity
- Continuous monitoring of web application traffic for potential security threats
- Regular updates to the whitelist of allowed redirect URLs to ensure it remains current and effective.