AI-Powered Security Policy Report

Generated: 2025-11-07 23:48:37

Total Vulnerabilities Scanned: 10
• SAST: 10
• 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: Explicit Unescape
Severity: HIGH
LLM: LLaMA 3.3 70B

## Security Policy: Explicit Unescape

### Executive Summary
The Explicit Unescape vulnerability poses a significant risk to our web application, allowing attackers to inject malicious scripts and potentially steal sensitive data or take control of user sessions. This policy outlines the necessary steps to mitigate this vulnerability, ensuring compliance with ISO 27001 and NIST Cybersecurity Framework standards. The estimated cost of remediation is $10,000, and the potential cost of a breach is $100,000.

### Risk Analysis

#### CVSS v3.1 Score Breakdown:
- **Base Score**: 8.8
- **Attack Vector**: Network
- **Attack Complexity**: Low
- **Privileges Required**: None
- **User Interaction**: Required
- **Scope**: Unchanged
- **Confidentiality Impact**: High
- **Integrity Impact**: High
- **Availability Impact**: Low

#### Threat Intelligence:

- **Known Exploits**: Yes, public exploits are available
- **MITRE ATT&CK; Techniques**: T1055 (Social Engineering), T1061 (Web Session Cookie Manipulation)
- **Attack Chain**: An attacker sends a malicious link to a user, who clicks on it, allowing the attacker to inject a script and steal sensitive data

### Compliance Framework Mapping

#### NIST Cybersecurity Framework:
- **Control ID**: PR.DS-5
- **Function**: Protect
- **Category**: Data Security
- **Implementation Requirements**: Implement secure coding practices, input validation, and output encoding
- **Evidence Required**: Code reviews, testing results, and configuration documents
- **Control ID**: PR.DS-6
- **Function**: Protect
- **Category**: Data Security
- **Implementation Requirements**: Implement data masking and encryption
- **Evidence Required**: Data masking policies, encryption protocols, and key management documents

#### ISO/IEC 27001:2022 Annex A:
- **Control**: A.8.23
- **Objective**: Manage access to external websites to reduce exposure to malicious content
- **Implementation Guidance**: Implement web filtering, monitor user activity, and provide user awareness training
- **Audit Trail**: Web filtering logs, user activity logs, and training records
- **Control**: A.8.8
- **Objective**: Manage technical vulnerabilities of information systems
- **Implementation Guidance**: Implement vulnerability scanning, patch management, and secure coding practices
- **Audit Trail**: Vulnerability scan reports, patch management records, and code review documents

#### Additional Frameworks:
- **OWASP ASVS**: Verification requirements for secure coding practices, input validation, and output encoding
- **PCI DSS**: Requirements for data encryption, key management, and access control

### Defense-in-Depth Strategy

#### Layer 1: Prevention
1. **Code Level**: Implement secure coding practices, input validation, and output encoding
2. **Framework Level**: Enable security features in the web framework, such as CSRF protection and XSS filtering
3. **Infrastructure Level**: Implement web application firewall (WAF) rules to block malicious traffic

#### Layer 2: Detection
1. **SIEM Correlation Rules**:
```sql
rule ExplicitUnescapeDetection {
description = "Detect Explicit Unescape attacks"
rule = "http.request.uri.query =~ //"<br/> alert = "Explicit Unescape attack
```

detected"<br/>}<br/>```<br/>2. **WAF Signatures**:<br/>```xml<br/><rule><br/> <name>ExplicitUnescape</name><br/> <pattern><script></pattern><br/> <action>block</action><br/></rule><br/>```<br/>3. **IDS/IPS Patterns**: Snort signature for detecting Explicit Unescape attacks<br/>```sql<br/>alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"Explicit Unescape attack"; content:"<script>"; sid:100001;)<br/>```<br/>4. **Log Monitoring**: Monitor web server logs for suspicious activity, such as unusual query parameters or user agent strings<br/><br/>#### Layer 3: Response<br/><br/>##### If Exploitation Detected:<br/>1. **Immediate Actions** (0-15 minutes):<br/> - Contain the attack by blocking the malicious IP address<br/> - Preserve evidence, such as web server logs and network captures<br/>2. **Short-term Response** (15 min - 4 hours):<br/> - Investigate the attack to determine the root cause and scope<br/> - Assess the impact of the attack and identify affected systems and data<br/>3. **Long-term Response** (4+ hours):<br/> - Perform a root cause analysis to identify vulnerabilities and weaknesses<br/> - Deploy remediation measures, such as patching or reconfiguring systems<br/> - Conduct a post-incident review to identify lessons learned and areas for improvement<br/><br/>### Remediation Strategies<br/><br/>#### Strategy A: Secure Framework Migration<br/>- **Approach**: Migrate to a secure web framework that provides built-in security features, such as input validation and output encoding<br/>- **Security**: ■■■■■<br/>- **Performance**: ■■■■<br/>- **Complexity**: ■■■<br/>- **Cost**: $5,000<br/>- **Timeline**: 2 weeks<br/><br/>#### Strategy B: Manual Validation Layer<br/>- **Approach**: Implement a custom validation layer to validate user input and prevent Explicit Unescape attacks<br/>- **Security**: ■■■■<br/>- **Performance**: ■■■<br/>- **Complexity**: ■■<br/>- **Cost**: $3,000<br/>- **Timeline**: 1 week<br/><br/>#### Strategy C: Web Application Firewall<br/>- **Approach**: Implement a WAF to block malicious traffic and prevent Explicit Unescape attacks<br/>- **Security**: ■■■<br/>- **Performance**: ■■■■■<br/>- **Complexity**: ■<br/>- **Cost**: $2,000<br/>- **Timeline**: Immediate<br/>- **Note**: Temporary solution, implement A or B for permanent fix<br/><br/>#### Recommended Approach: Strategy A with justification<br/>Strategy A provides the highest level of security and is a long-term solution. While it may require more resources and time, it is the most effective way to prevent Explicit Unescape attacks.<br/><br/>### Security Tool Integration<br/><br/>#### SAST Configuration:<br/>```yaml<br/>rules:<br/> - id: ExplicitUnescape<br/> pattern: /<script>/<br/> severity: high<br/>```<br/>#### DAST Testing:<br/>```python<br/>import requests<br/><br/>def test_explicit_unescape():<br/> url = "https://example.com/profile"<br/> payload = {"name": "<script>alert('XSS')"}
response = requests.post(url, data=payload)
assert "XSS" not in response.text
```

#### SCA Monitoring:
Monitor package versions and vulnerability reports for dependencies used in the web application

### Threat Modeling

#### Attack Scenarios:
1. **Scenario 1**: An attacker sends a malicious link to a user, who clicks on it, allowing the attacker to inject a script and steal sensitive data
- **Likelihood**: Medium
- **Impact**: High
- **Detection**: SIEM correlation rules and WAF signatures
2. **Scenario 2**: An attacker exploits a vulnerability in the web application to inject a
```