***7.1 Findings and Reports:***

Vulnerability Report: Swift Incidence Response

**1. SSL Certificate Cannot Be Trusted**

* **Severity: High**
* **CWE: CWE-295**
* **OWASP: A6: Security Misconfiguration**
* **Plugin: SSL Certificate Validation Plugin**
* **Port: 443 (HTTPS)**
* **Description:  
  This vulnerability occurs when the SSL certificate presented by the server is not trusted by the client. This can be due to an expired, self-signed, or improperly issued certificate.**
* **Solution:** 
  + **Ensure the SSL certificate is issued by a trusted Certificate Authority (CA).**
  + **Regularly update and renew SSL certificates before expiration.**
  + **Use HTTPS Strict Transport Security (HSTS) to enforce secure connections.**
* **Business Impact:** 
  + **Can lead to Man-in-the-Middle (MitM) attacks, compromising data confidentiality.**
  + **Loss of user trust due to browser warnings about insecure connections.**
  + **Potential non-compliance with industry security standards (PCI-DSS, GDPR).**

**2. Node.js Version Related Multiple Issues**

* **Severity: Critical**
* **CWE: CWE-1104**
* **OWASP: A9: Using Components with Known Vulnerabilities**
* **Plugin: Node.js Security Scanner**
* **Port: Varies (application-dependent)**
* **Description:  
  Multiple security vulnerabilities exist in outdated Node.js versions, including arbitrary code execution, denial of service (DoS), and security bypass mechanisms.**
* **Solution:** 
  + **Upgrade Node.js to the latest LTS (Long-Term Support) version.**
  + **Regularly check Node.js security advisories and apply patches.**
  + **Remove deprecated or unsupported Node.js versions from production environments.**
* **Business Impact:** 
  + **Attackers may execute malicious code remotely.**
  + **Can lead to service disruptions and downtime.**
  + **May expose applications to data breaches and compliance violations.**

**3. OpenJDK 8<=8U432 / 17.0.0<=17.0<=21.0.5 / 23.0.1 Vulnerability**

* **Severity: High**
* **CWE: CWE-21502**
* **OWASP: A9: Using Components with Known Vulnerabilities**
* **Plugin: OpenJDK Security Audit Plugin**
* **Port: Varies (application-dependent)**
* **Description:  
  Older versions of OpenJDK contain multiple security vulnerabilities, which may result in unauthorized access, information disclosure, and privilege escalation.**
* **Solution:** 
  + **Upgrade to the latest stable version of OpenJDK.**
  + **Apply security patches as soon as they are released.**
  + **Configure Java applications to limit permissions and reduce attack surfaces.**
* **Business Impact:** 
  + **May lead to data leaks, compromised authentication, and unauthorized access.**
  + **Poses regulatory risks, affecting compliance with ISO 27001, HIPAA, and GDPR.**
  + **Can be exploited in zero-day attacks, increasing operational risks.**

**4. Apache HTTP Server Site Enumeration**

* **Severity: Medium**
* **CWE: CWE-200**
* **OWASP: A6: Security Misconfiguration**
* **Plugin: Apache Server Security Scanner**
* **Port: 80 (HTTP), 443 (HTTPS)**
* **Description:  
  This vulnerability allows attackers to enumerate all sites hosted on an Apache HTTP server, potentially revealing sensitive information about the server configuration, directory structure, and hosted applications.**
* **Solution:** 
  + **Disable directory listing and other unnecessary HTTP responses.**
  + **Restrict access to server configuration files.**
  + **Implement mod\_security or WAF (Web Application Firewall) to block enumeration attempts.**
* **Business Impact:** 
  + **Attackers can map the server environment for future targeted attacks.**
  + **Information leakage may expose internal applications and services.**
  + **Increases the attack surface, making brute-force and reconnaissance easier.**

**5. Node.js 18.x < 18.20.1 / 20.x < 20.12.1 / <21.7.2 Multiple Vulnerabilities**

* **Severity: Critical**
* **CWE: CWE-400, CWE-352, CWE-20**
* **OWASP: A9: Using Components with Known Vulnerabilities**
* **Plugin: Node.js Vulnerability Scanner**
* **Port: Varies (application-dependent)**
* **Description:  
  Several vulnerabilities in outdated Node.js versions allow uncontrolled resource consumption (CWE-400), cross-site request forgery (CSRF) (CWE-352), and improper input validation (CWE-20). These weaknesses could lead to DoS attacks, unauthorized actions, and data integrity issues.**
* **Solution:** 
  + **Update to the latest stable version of Node.js.**
  + **Implement input validation and sanitization to prevent injection attacks.**
  + **Apply security headers and CSRF protection in web applications.**
* **Business Impact:** 
  + **Risk of Denial of Service (DoS) attacks, leading to downtime and financial loss.**
  + **Can enable cross-site scripting (XSS) or CSRF exploits, compromising user data.**
  + **Regulatory non-compliance, increasing legal and reputational risks.**