**Ethical Hacking Technical Report**

**Client: Acme Corporation**

**Date: May 09, 2024**

**Prepared by: Reymond Canonce and Kenneth Togores**

**Introduction**

**This report presents the findings of a comprehensive ethical hacking assessment conducted on Acme Corporation's network infrastructure and web applications. The assessment aimed to identify potential vulnerabilities and security loopholes that could be exploited by malicious actors.**

**Scope**

**The assessment focused on the following areas:**

**Network Infrastructure**

**Web Applications**

**Methodology**

**The assessment was conducted using a combination of automated tools and manual techniques to simulate real-world attack scenarios. The following vulnerabilities were identified:**

**Outdated Software Versions: Several systems were found to be running outdated software versions, making them vulnerable to known exploits.**

**Weak Password Policies: Weak password policies were observed, including the use of default or easily guessable passwords.**

**Missing Security Patches: Critical security patches were missing on various systems, leaving them susceptible to known vulnerabilities.**

**Insecure Network Configuration: Network devices were found to have insecure configurations, such as default credentials or unnecessary open ports.**

**SQL Injection: Web applications were susceptible to SQL injection attacks due to improper input validation.**

**Cross-Site Scripting (XSS): XSS vulnerabilities were found in web applications, allowing attackers to inject malicious scripts.**

**Sensitive Data Exposure: Some web applications exposed sensitive data, such as user credentials or personal information, without proper encryption.**

**Insecure File Uploads: Web applications allowed users to upload files without proper validation, potentially leading to the execution of malicious code.**

**Lack of Rate Limiting: APIs lacked rate limiting controls, making them vulnerable to brute force and DoS attacks.**

**Broken Authentication: Authentication mechanisms in web applications were found to be vulnerable to session fixation and session hijacking attacks.**

**Insecure Direct Object References (IDOR): Some web applications exposed internal implementation details, allowing attackers to access unauthorized resources.**

**Security Misconfigurations: Improperly configured security settings were found on web servers and network devices, leading to potential exploitation.**

**Cross-Site Request Forgery (CSRF): Lack of CSRF protection in web applications could allow attackers to perform unauthorized actions on behalf of authenticated users.**

**Insecure Deserialization: Web applications were vulnerable to insecure deserialization attacks, potentially leading to remote code execution.**

**Sensitive Information Disclosure in Error Messages: Error messages from web applications sometimes exposed sensitive information, aiding attackers in crafting targeted attacks.**

**Recommendations for Remediation**

**Based on the findings of the assessment, the following recommendations are proposed to mitigate the identified vulnerabilities:**

**Regular Software Updates: Implement a robust patch management process to ensure all software and systems are kept up-to-date with the latest security patches.**

**Enforce Strong Password Policies: Implement password complexity requirements and enforce regular password changes to prevent the use of weak passwords.**

**Vulnerability Scanning and Remediation: Conduct regular vulnerability scans and promptly remediate any identified vulnerabilities.**

**Secure Network Configurations: Review and update network device configurations to ensure they follow security best practices, including changing default credentials and closing unnecessary ports.**

**Input Validation and Sanitization: Implement proper input validation and sanitization techniques to prevent SQL injection and XSS attacks in web applications.**

**Data Encryption: Encrypt sensitive data both at rest and in transit to protect it from unauthorized access.**

**File Upload Validation: Implement strict file upload validation checks to prevent the execution of malicious code via file uploads.**

**API Rate Limiting: Implement rate limiting controls on APIs to mitigate brute force and DoS attacks.**

**Session Management: Implement secure session management practices, including session rotation and proper session token handling, to prevent session-related attacks.**

**Access Controls: Implement proper access controls and authorization mechanisms to prevent unauthorized access to resources.**

**Security Configuration Reviews: Regularly review and update security configurations on web servers and network devices to ensure they adhere to security best practices.**

**CSRF Protection: Implement CSRF protection mechanisms, such as anti-CSRF tokens, to prevent CSRF attacks.**

**Secure Deserialization: Implement secure deserialization practices, such as input validation and integrity checks, to prevent insecure deserialization attacks.**

**Error Handling: Ensure error messages do not reveal sensitive information and provide generic error messages to users.**

**Conclusion**

**Addressing the identified vulnerabilities and implementing the recommended remediation measures is essential to strengthen Acme Corporation's security posture and mitigate the risk of cyber attacks. Regular security assessments and proactive measures are crucial in maintaining a robust and resilient security environment.**