CyberSecuriosity

PROFESSIONAL CYBER SECURITY SERVICES

Penetration Testing Report

Prepared for

May 22, 2024

CONTENT

1. EXECUTIVE SUMMARY....3

2. PROJECT INFORMATION....4

3. RESULTS CLASSIFICATION....5

4. VULNERABILITY DETAILS....8

5. ASSESSMENT APPROACH....10

END OF THE DOCUMENT....13

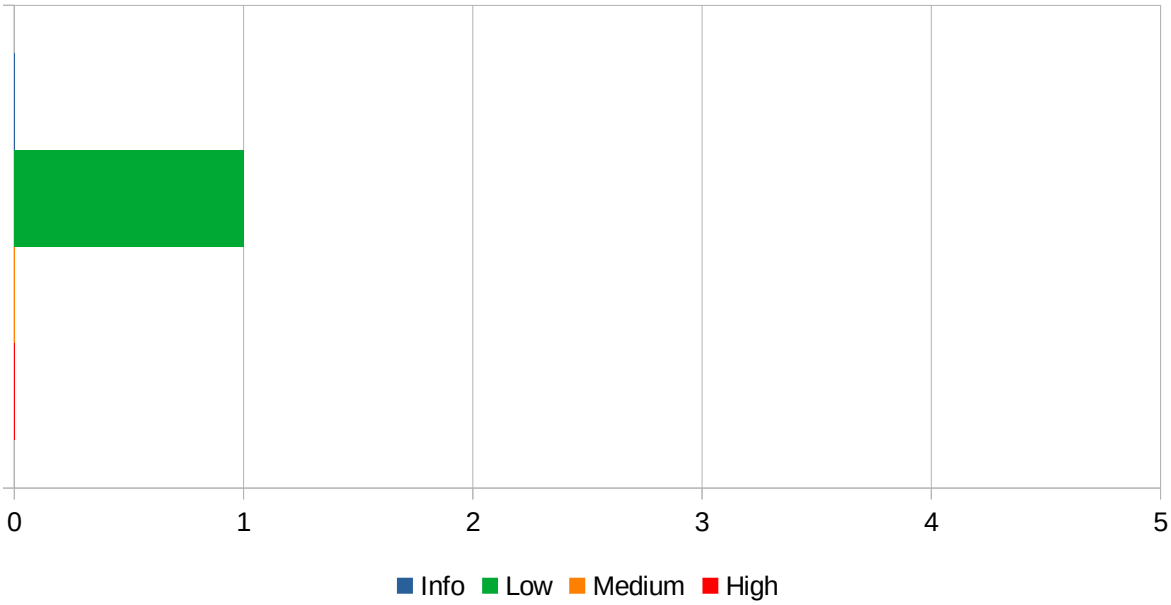
1. Executive Summary

1.1. Project Description

Evaluation of web application, mobile application and network security and vulnerabilities identification by simulating the actions of the attacker and unauthorized access gaining to information or other resources of the Forghetti. Activities have been performed from May, 1st to May 7th , 2024.

1.2. Summary of Findings

1.2.1. Results Summary Table



1.2.2. Brief Results Description

The Penetration Testing was conducted according to the **PTES** standard, **Black Box** or **Gray Box** offender model based on the **OWASP Web Security Testing Guide**, **OWASP Mobile Security Testing Guide** and **NIST SP 800-115 methodology**. Threat classification is based on commonly used standards: **OWASP TOP10**, **OWASP TOP10 Mobile** and **NIST CVSS**. Security check was performed having access to the Internet and producing technical attack methods without the use of social engineering.

Unauthorized access to the IT infrastructure **was NOT gained** during the Penetration Testing activities. However, the system has non critical vulnerabilities, which could be potentially exploited by an attacker. Therefore, the level of security was assessed as “high”.

Overall Security Level

High

2. Project Information

2.1. Overview

CyberSecuriosity specialists conducted an external penetration test against “Forghetti” web application, mobile application and external network perimeter. A penetration test was performed to assess “Forghetti” defensive capabilities and provide security assistance through proactively identifying vulnerabilities, validation of their severity, and providing remediation steps.

The findings in this report are the result of CyberSecuriosity attempts of vulnerabilities discovery, validation, and further exploitation during penetration test execution within the project’s scope and duration.

2.2. Scope of Work

- www.forghetti.com
- app.forghetti.com
- api.forghetti.app
- static.forghetti.app
- Mobile application: Android platform, iOS platform, forghettiapp.firebaseio.com

2.3. Offender Model

Security check was performed according to the **Black Box** model of the offender, having access to the public facing web application(s) and IP address(es) and producing technical attack methods without the use of social engineering.

2.4. Offender Model Definition

Black Box

In a black-box testing assignment, the penetration tester is placed in the role of the average hacker, with no internal knowledge of the target system. Testers are not provided with any architecture diagrams or source code that is not publicly available. A black-box penetration test determines the vulnerabilities in a system that are exploitable from outside the network.

Gray Box

If a Black Box tester is examining a system from an outsider’s perspective, a gray-box tester has the access and knowledge levels of a user, potentially with elevated privileges on a system. Gray Box penetration testers typically have some knowledge of a network’s or app’s internals, potentially including design and architecture documentation and an account internal to the network or application.

3. Results Classification

3.1. Classification Approach

During the test, each finding is classified according to its severity, reflecting the risk each such vulnerability may pose on the business processes implemented by the system or application, based on the following criteria:

- **Severity: key assessment point**
  - **High.** The vulnerability allows unauthorized access, and potentially gain complete control of the affected system.
  - **Medium.** The vulnerability allows to obtain partial control of the system (increase benefits), or take complete control of the system when the additional conditions are met.
  - **Low.** The vulnerability allows obtaining non-critical information that does not lead to the full or partial system compromise.
  - **Informational.** The lowest severity level can be applied to misconfigurations, some kind of typical errors or information disclosure, which is not critical for system and/or business. Such kind of issue cannot lead to vulnerability exploitation in most cases. Not taken into account in the overall security level.
- **Likelihood: point to describe exploitation probability**
  - **High.** Wide public availability of a relatively simple script or exploit. Low level of skill required for reproduction.
  - **Medium.** An average attacker would have to do some research into the vulnerability but would eventually be successful in exploitation.
  - **Low.** Lack of publicly available exploit code, the high difficulty of reproduction.
  - **Informational.** The lowest likelihood level applied in case just to inform about the potential low probability of exploitation and are for informational purposes only. Not taken into account in the overall security level.

In addition to classification by basic parameters, each problem is also assigned a characteristic based on OWASP and NIST CVSS standards.

OWASP Open Web Application Security Project

NIST

3.2. Results Checklists

The following table contains the checklist for OWASP TOP10 2021 declared security flaws.

Security Risk	Result	Corrective Action
A1 Broken Access Control	Passed	
A2 Cryptographic Failures	Passed	
A3 Injection	Passed	
A4 Insecure Design	Passed	
A5 Security Misconfiguration	Passed	
A6 Vulnerable & Outdated Components	Passed	
A7 Identification & Authentication Failures	Passed	
A8 Software & Data Integrity Failures	Passed	
A9 Security Logging & Monitoring Failures	Passed	
A10 Server-Side Request Forgery	Passed	

The following table contains the checklist for OWASP TOP10 Mobile declared security flaws.

Security Risk	Result	Corrective Action
M1 Improper Platform Usage	Passed	
M2 Insecure Data Storage	Passed	
M3 Insecure Communication	Passed	
M4 Insecure Authentication	Passed	
M5 Insufficient Cryptography	Failed	4.1;
M6 Insecure Authorization	Passed	
M7 Client Code Quality	Passed	
M8 Code Tampering	Passed	
M9 Reverse Engineering	Passed	
M10 Extraneous Functionality	Passed	

The following table contains the checklist for OWASP TOP10 API declared security flaws.

Security Risk	Result	Corrective Action
API1 Broken Object Level Authorization	Passed	
API2 Broken User Authentication	Passed	
API3 Excessive Data Exposure	Passed	
API4 Lack of Resources & Rate Limiting	Passed	
API5 Broken Function Level Authorization	Passed	
API6 Mass Assignment	Passed	
API7 Security Misconfiguration	Passed	
API8 Injection	Passed	
API9 Improper Assets Management	Passed	
API10 Insufficient Logging & Monitoring	Passed	

The following table contains the NIST CVSS Base Score 2.0 classified security flaws.

Security Risk	Classification	Result	Corrective Action
High	7.0-10.0	Passed	
Medium	4.0-6.9	Passed	
Low	0.0-3.9	Failed	4.1;

4. Vulnerability Details

Summary	4.1. The code uses AES encryption in CBC mode with PKCS5Padding ("AES/CBC/PKCS5Padding")	OWASP: M5
Severity	Low	CVSS: 1.4
Likelihood	Low	
Target	com/cordova/plugin/android/fingerprintauth/FingerprintAuth.java com/forghetti/plugin/encryption/CryptLib.java	

Description

he code uses AES encryption in CBC mode with PKCS5Padding ("AES/CBC/PKCS5Padding"). This padding mechanism is susceptible to padding oracle attacks if implemented incorrectly.

```
import javax.crypto.spec.SecretKeySpec;
/* loaded from: classes.dex */
public class CryptLib {
    private Cipher _cx = Cipher.getInstance("AES/CBC/PKCS5Padding");
    private byte[]  key = new byte[32];

try {
    mKeyGenerator = KeyGenerator.getInstance("AES", ANDROID_KEY_STORE);
    mKeyStore = KeyStore.getInstance(ANDROID_KEY_STORE);
    try {
        mCipher = Cipher.getInstance("AES/CBC/PKCS7Padding");
```

CBC mode with PKCS5Padding is vulnerable to padding oracle attacks if an attacker can distinguish between different padding error messages. This can lead to leakage of sensitive information, such as plaintext data.

Remediation

OWASP MSTG recommends using Optimal Asymmetric Encryption Padding (OAEP) instead of PKCS5Padding for asymmetric encryption. For symmetric encryption, using PKCS7Padding with AES-GCM mode or another authenticated encryption mode is recommended.

Based on the above analysis, it's advisable to consider the following actions to address potential vulnerabilities:

- Switch to a more secure padding mechanism such as OAEP (Optimal Asymmetric Encryption Padding) for asymmetric encryption or PKCS7Padding with AES-GCM mode for symmetric encryption.
- Implement proper error handling to avoid leaking information that could aid padding oracle attacks.

5. Assessment Approach

5.1. Assessment Methodology

The testing methodology is mostly based on a manual penetration test, which is enhanced through the use of various automated tools. This process begins with detailed scanning and research into the architecture and environment, with the performance of automated testing for known vulnerabilities.

Manual validation and exploitation of vulnerabilities follow, for the purpose of detecting security weaknesses in the applications and networks in scope. The tools used throughout the process vary based on the scope and offender model.

The next step is the exploitation of vulnerabilities to determine the possibility of further infrastructure movement possibilities, privilege escalation, any additional information gathering, or the possibility of vulnerabilities combination and kill-chain creation.

At the last stage, we assess vulnerabilities and assign them an objective threat level, as well as the likelihood that they can be exploited and all results put into the final report which can be used for a better understanding of the current security state and remediation process.

The customer has 2 to 3 weeks to fix vulnerabilities. During the remediation process, our specialists are available to answer all questions regarding the report and its results. As a result of the remediation process, all problems that affect the overall level of security should be fixed.

Another round of re-testing can be started right after confirmation from the customer side that all issues were fixed. The main objective of the re-testing phase is to verify if identified security issues were fixed properly and applied fixes can’t be bypassed. At the end of the retest, the report is updated based on the current security level.

5.2. Penetration Testing Process Overview

Only confirmed results that have passed manual verification are included in our reports, which guarantees the complete exclusion of false-positive results in the report. Final verification is performed both for the results of automated scanning tools and manual testing results.

5.3. List of Used Tools

In the process of performing penetration testing and vulnerability assessment, the team used a mixed set of programs, from specialized open source solutions to popular paid solutions, taking into account the specifics and needs of this project. Below you can find the list of used software:

- Nessus
- Acunetix
- Dirb
- BurpSuite
- jadx
- apktool
- jwt_tool
- ghidra
- objection
- Custom built scripts.

End of the document