# Android VA

**BY**

# CyberSapiens



DAST Report on

Android application

CyberSapiens

THE CYBER SECURITY EXPERTS

## Table of Contents

# 1. Document Attributes

| | |
|---|---|
| Date | DATE |
| Version | 1.0 |
| Prepared by | Name |
| Reviewed by | Name |
| Approved by | Name |
| Submitted to | Name |

## 2. Executive Summary

CyberSapiens was contracted by **Client** to conduct an Android Application Vulnerability Assessment activity to determine its exposure to the targeted attacks and ensure that **Client** application is secure from advanced attack techniques.
This activity was conducted in a manner that malicious attacker is engaged to assess the provided scope of **SmartCoin**. The goals of the vulnerability assessment and Penetration Testing scan were:

- Identifying the threats or vulnerabilities that might be present on the Android application
- Confidentiality of the **Client data** that are stored on the company storage/servers



## 3. VAPT Test Graph

Non-Vulnerable assets: The assets that we tested has no vulnerabilities and is secure
Vulnerable assets: The asset which we have tested and has vulnerability present.

| Type | Count |
|---|---|
| Non-Vulnerable assets | 0 |
| Vulnerable assets | 1 |

## 4. AUDITING SCOPE

Detailed list of assets is given in the below table.

| SL NO | URL/Name | Type of Asset |
|---|---|---|
| 1 | Android application | Android |

## 5. Methodologies and Standards

The following standards were referred during the project:

- OWASP Top 10 Mobile testing guide - The Open Web Application Security Project® (OWASP) is a nonprofit foundation that works to improve the security of software. This is a group of security experts from around the world are a part of this. The risks are ranked on the frequency of discovered security defects, severity, and impact of the vulnerability
- PTES - Penetration Testing Execution Standard (PTES)

The following methodology was followed during the testing:

- Discovery requires the pen tester to collect information that is essential in understanding events that lead to the successful exploitation of mobile applications.
- Assessment or analysis involves the penetration tester going through the mobile application source code and identifying potential entry points and weaknesses that can be exploited.
- Exploitation involves the penetration tester leveraging the discovered vulnerabilities to take advantage of the mobile application in a manner not intended by the programmer initially did not intend.
- Reporting is the final stage of the methodology and it involves recording and presenting the discovered issues in a manner that makes sense to management. This is also the stage that differentiates a penetration test from an attack. A more detailed discussion of the four stages follows.

## 6. VAPT Project Timeframe

The VAPT activity was conducted between **dd-mm-yyyy to dd-mm-yyyy**

## 7. Risk Ratings and Threat Level

| Severity | Description |
|---|---|
| **Critical** | Loss of business / Breach of internal data / non-bearable financial and reputational loss / Breakdown of assets / Access and modification of critical data |
| **High** | Loss of customer / Exposure of internal data / Noncompliance to regulations / Unavailability of the services / Access to configurational changes / High financial and reputational loss / Access and modification of internal data |
| **Medium** | Customer service affected for one day / Noncompliance with internal requirements / Bearable financial and reputational loss / Disclosure of non-public data |
| **Low** | Internal services affected / Minor inconvenience to customers / Very minimal financial and reputation loss |

## 8. Vulnerability Summary

A summary of vulnerabilities that have been discovered while performing android application security assessment are given bellow:



| | | |
|---|---|---|
| **Critical** 1 | **High** 3 | **Medium** 1 |
| **Low** 2 | **Info** 0 | **Total** 7 |

## 9. Observations

| | |
|---|---|
| **Vulnerability #1** | **Parameter Tampering** |
| **IP Address/URL** | http://xxxxxx/savings/xxxxx/xxxxx/2242325/xxxxxxxx |
| **Risk** | **Critical** |
| **Description** | Parameter Tampering attack is based on the manipulation of parameters exchanged between client and server in order to modify application data, such as user credentials and permissions, price and quantity of products, etc. |
| **Impact** | Attacker can tamper the amount of the gold and continue buying the gold with tampered amount. This will lead to huge financial loos to the company |
| **Proof of Concept (POC)** | Step 1:<br>Step 2:<br>Step 3:<br>Step 4: |
| **Suggested Remediation** | [redacted] |

| | |
|---|---|
| **Vulnerability #2** | **BFLA – Unauthorized access to application status** |
| **URL / APK Path** | http://xxxxxxx/xxxxx/224234235/xxxxxx/12345 |
| **Risk** | **High** |
| **Description** | Broken Function Level Authorization arising due to improper validation of the authorization level of the user of an API and the function that it is intended to perform. |
| **Impact** | Attacker is able to make the application status to “VERIFIED” |
| **Proof of Concept (POC)** | Step 1:<br>Step 2:<br>Step 3: |
| **Suggested Remediation** | [redacted] |

| | |
|---|---|
| **Vulnerability #3** | **OTP Bypass Via response manipulation** |
| **URL / APK Path** | http://xxxxx/xxxx/2243435235/xxx/xxx/?xxxxxxx |
| **Risk** | **High** |
| **Description** | Response manipulation is a technique where attacker try to analyze request using some proxy tool. Attacker can change value of response without entering correct OTP. This kind of attack is most successfully when server-side validation is missing and application trust client-side generated API call which are called upon previous API response code or status |
| **Impact** | Attacker can create account with any mobile number without verifying the OTP. 2FA are consider to additional security layer but failing of can lead to greater impact on application. |
| **Proof of Concept (POC)** | Step 1:<br>Step 2:<br>Step 3: |
| **Suggested Remediation** | [redacted] |

| | |
|---|---|
| **Vulnerability #4** | **Host Header Injection** |
| **URL / APK Path** | https://xxxxxxx/xxxx/2242325/xxxx/ |
| **Risk** | **High** |
| **Description** | During testing of the apk backend was vulnerable to host header injection In many cases, developers are trusting the HTTP Host header value and using it to generate links, import scripts and even generate password resets links with its value. This is a very bad idea, because the HTTP Host header can be controlled by an attacker. This can be exploited using web-cache poisoning and by abusing alternative channels like password reset emails. |
| **Impact** | Without proper validation of the header value, the attacker can supply invalid input to cause the web server to: - |

| | |
|---|---|
| | • dispatch requests to the first virtual host on the list<br>• cause a redirect to an attacker-controlled domain<br>• perform web cache poisoning<br>• manipulate password reset functionality |
| **Proof of Concept (POC)** | Step 1: -<br>Step 2: -<br>Step 3:- |
| **Suggested Remediation** | [redacted] |

| | |
|---|---|
| **Vulnerability #5** | **Missing security headers** |
| **URL / APK Path** | https://xxxxx/xxxx/2242325/xxxx/xxx/xxx/xxxx/faq?type=MAIN |
| **Risk** | **Medium** |
| **Description** | During analysis it was observed that following Security headers were not set.<br>1. X-XSS-Protection header s<br>2. X-content type header<br>3. X-frame options header<br>4. Content security policy Header<br>5. Strict-transport-security header |
| **Impact** | Missing security headers results in various attack vectors<br>1. X-XSS-Protection header: If this security header is missing, website could be at risk of a Cross-site Scripting (XSS) attacks.<br>2. X-frame options header: If this security header is missing, the website could be at risk of a clickjacking attack.<br>3. Content security policy Header: There is no direct impact of not implementing CSP on your website. However, if your website is vulnerable to a Cross-site Scripting attack CSP can prevent successful exploitation of that vulnerability. By not implementing CSP you'll be missing out this extra layer of security<br>4. Strict-transport-security header: If this security header is missing, |

| | |
|---|---|
| | website is being served from not only HTTPS but also HTTP and it lacks of HSTS policy implementation. |
| **Proof of Concept (POC)** | Step 1: - Below POC can be seen that security headers are missing |
| **Suggested Remediation** | [redacted] |

| | |
|---|---|
| **Vulnerability #6** | **Nginx server disclosure via 403 error page** |
| **URL / APK Path** | https://xxxxxx.xxxx.in/xx |
| **Risk** | **Low** |
| **Description** | The HTTP header may disclose server name in response header or improper response handling or any forbidden page. Accurately discovering the type of web server that an application runs on can enable security testers to determine if the application is vulnerable to attack. |
| **Impact** | In particular, servers running older versions of software without up-to-date security patches can be susceptible to known version-specific exploits. |
| **Proof of Concept (POC)** | Step 1:- Below POC can be referred which shows nginx server |
| **Suggested Remediation** | [redacted] |

| | |
|---|---|
| **Vulnerability #7** | **Private email ID disclosed** |
| **URL / APK Path** | https://qxxxxxx/xxxx/224152325/xxxx?xxx_xxx_xxx=16519846&app_version=xxxx |
| **Risk** | **Low** |
| **Description** | During analysis it was observed that, one of the private e-mail id was disclosed. |
| **Impact** | An attacker may launch further attacks using the private email id which was disclosed, also enumeration can be done if the account exists in the DB. |
| **Proof of Concept (POC)** | Step 1: - Below endpoint disclosed private mail id |
| **Suggested Remediation** | [redacted] |



## 10. Tools used for the Assessment

The VAPT activities utilizes many automated tools and manual exploitation methodologies to identify security vulnerabilities. A detailed list of tools used is given below.

| Tool Name | Description |
|---|---|
| Burp Suite | Tool used for scanning and confirming vulnerabilities. Have various different features. |
| JADX | APK Decompiler |
| MobSF | Automated security assessment framework |
| Genymotion | Android emulator |

## 11. Conclusion

With the overall testing performed, the Android application seems to be working perfectly. However, we have noted a few observations/vulnerabilities that needs to be taken into consideration on priority for fixing them.

Overall, the features of the product are very useful considering the current real time threats, which provides a corrective solution during an attack.