



Android Penetration Test Analysis Report

Customer: Trustee Plus

Date: 04/03/2025



We express our gratitude to the Trustee team for the collaborative engagement that enabled the execution of this Pentest.

Trustee Plus is a platform that provides a full range of services and tools for working with crypto assets. Founded in 2016 by a group of crypto enthusiasts, the company aims to unlock the potential of cryptocurrency technology. Trustee's infrastructure seamlessly integrates digital finance with the traditional financial system and is available globally.

Document

Name	Android Penetration Test Security Analysis Report for Trustee
Audited By	Bogdan Bodisteanu
Approved By	Stephen Ajayi
Website	trusteeglobal.com
Changelog	24/02/2025 - Preliminary Report
Changelog	04/03/2025 - Retest Report
Platform	Android
Language	Java, React Native
Tags	Android
Methodology	https://hackenio.cc/dApp_methodology

Review Scope

Android Application	Shared Privately
---------------------	------------------

Audit Summary

The system users should acknowledge all the risks summed up in the risks section of the report

6	0	6	0
Total Findings	Resolved	Accepted	Mitigated

Findings by Severity

Severity	Count
Critical	0
High	0
Medium	2
Low	3

Vulnerability	Severity
F-2025-8873 - SSL Pinning Bypass	Medium
F-2025-8919 - Original IP Address	Medium
F-2025-8874 - GraphQL Introspection Query Exposure	Low
F-2025-8875 - Insecure FileProvider Paths Configuration	Low
F-2025-8918 - Clear Text Traffic Enabled	Low
F-2025-8917 - Application Support Legacy SDK Versions	Info

Executive Summary

This threat model provides an overview of the security assessment conducted on the Trustee Plus Android application. The goal of this evaluation was to identify vulnerabilities and assess the security posture of the application against common penetration testing techniques, including static and dynamic analysis, reverse engineering, and data leakage risks.

During our assessment, several vulnerabilities were identified, ranging from medium to low severity. The primary areas of concern include SSL Pinning Bypass, Clear Text Traffic Enabled, Insecure FileProvider Paths Configuration, GraphQL Introspection Query Exposure, and exposure of the original IP address. Addressing these vulnerabilities will enhance the application's resilience against potential attacks.

Methodology

The evaluation was performed using a combination of **static analysis** (analyzing the APK structure, permissions, manifest configurations, and code obfuscation) and **dynamic analysis** (monitoring application behavior in real-time and testing against different attack vectors). The security controls were evaluated to determine their effectiveness against tampering, debugging, and sensitive data leakage.

Key Findings

1. Exposure of Original IP Address (Medium, 6.9)

- The application exposes the original IP address of users, which can lead to privacy concerns and tracking risks.
- Attackers may exploit this vulnerability to perform reconnaissance or targeted attacks.
- **Recommendation:** Implement IP masking techniques, use VPN proxies, and restrict sensitive logs.

2. SSL Pinning Bypass (Medium, 6.9)

- The application implements SSL pinning but is vulnerable to bypass techniques.
- Attackers can intercept encrypted traffic, leading to potential data exposure.
- **Recommendation:** Enforce stronger SSL pinning mechanisms, implement certificate validation at multiple layers, and consider hardware-backed security measures.

3. Clear Text Traffic Enabled (Low, 2.3)

- The application allows unencrypted HTTP traffic, which increases the risk of man-in-the-middle (MITM) attacks.
- **Recommendation:** Disable cleartext traffic by enforcing HTTPS for all communications using the `android:usesCleartextTraffic="false"` directive in the manifest.

4. Insecure FileProvider Paths Configuration (Low, 2.3)

- The application has misconfigured FileProvider paths, potentially exposing sensitive files.
- **Recommendation:** Restrict access to FileProvider paths and ensure that only required components have access.

5. GraphQL Introspection Query Exposure (Low, 2.3)

- The application exposes the GraphQL introspection query, allowing attackers to enumerate API endpoints and gain insight into backend structures.
- **Recommendation:** Disable introspection queries in production, implement authentication checks for API access, and restrict excessive data exposure.

Conclusion

The security evaluation of the Trustee Plus Android application identified several vulnerabilities that could be exploited by attackers. While none of the issues were classified as critical, medium-risk vulnerabilities like SSL Pinning Bypass and Original IP Address Exposure pose significant threats to data security and user privacy.

To improve security, it is recommended to:

- Strengthen SSL pinning mechanisms and prevent bypass techniques.
- Disable cleartext traffic to enforce encrypted communication.
- Secure FileProvider paths to prevent unauthorized access to sensitive data.
- Restrict GraphQL introspection queries to minimize API enumeration risks.
- Implement logging best practices to avoid exposing user IP addresses.

By addressing these findings, the Trustee Plus Android application can significantly enhance its security posture and mitigate potential attack vectors.

Table of Contents

System Overview	7
Findings	8
Vulnerability Details	8
F-2025-8873 - SSL Pinning Bypass - Medium	8
F-2025-8919 - Original IP Address - Medium	10
F-2025-8874 - GraphQL Introspection Query Exposure - Low	12
F-2025-8875 - Insecure FileProvider Paths Configuration - Low	15
F-2025-8918 - Clear Text Traffic Enabled - Low	17
F-2025-8917 - Application Support Legacy SDK Versions - Info	19
Disclaimers	20
Hacken Disclaimer	20
Technical Disclaimer	20
Appendix 1. Severity Definitions	21
Appendix 2. Scope	22

System Overview

The audited mobile application is a crypto financial trading platform that provides users with functionalities such as account management, real-time market data, trade execution, and secure transactions. The app is developed for Android platform, utilizing a client-server architecture for communication with backend services.

Findings

Vulnerability Details

[F-2025-8873](#) - SSL Pinning Bypass - Medium

Description:

SSL Pinning is a security mechanism that ensures an application only trusts a specific set of SSL certificates, preventing Man-in-the-Middle (MITM) attacks. However, in the Trustee Plus Android application, it was found that SSL Pinning can be bypassed using dynamic instrumentation. This allows an attacker to intercept and manipulate HTTPS traffic between the application and its backend servers, potentially leading to data leakage, credential theft, or API abuse.

Assets:

- com.trusteeplus

Status:

Accepted

Classification

Vulnerability Type(s):

- Bypass Something
- Gain Information

CVSS 4.0:

6.9 (AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N)

Weakness (CWE):

- [CWE-295 | Improper Certificate Validation](#)

Severity:

Medium

Recommendations

Remediation:

- Utilize a stronger SSL Pinning implementation that includes multiple trusted certificates to prevent easy bypassing.
- Use the latest versions of security libraries such as TrustKit or Conscrypt, which make SSL Pinning harder to bypass.
- Implement detection and prevention techniques against runtime modifications, such as monitoring for hooking frameworks (e.g., Frida, Objection) and validating SSL configurations dynamically.

[F-2025-8919](#) - Original IP Address - Medium

Description: GraphQL introspection is enabled on the Trustee Plus Android application's backend, specifically on `apollo.trusteeglobal.com`. This allows attackers to enumerate all available queries, mutations, and types, potentially revealing sensitive internal API structure. Additionally, an original server IP address (`52.50.167.8`) has been discovered, which could expose the application to direct attacks bypassing cloud-based security protections such as DDoS mitigation or WAF rules.

Assets:

- `com.trusteeplus`

Status: Accepted

Classification

Vulnerability Type(s):

- Bypass Something
- Gain Information

CVSS 4.0: 6.9 (AV:N/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N)

Weakness (CWE):

- [CWE-200 | Exposure of Sensitive Information to an Unauthorized Actor](#)

Severity: Medium

Recommendations

Remediation:

- Mask original server IP addresses behind a proxy or cloud-based security solutions to mitigate direct attacks.
- Restrict access to the backend infrastructure by limiting requests to only trusted sources.

Evidences

Proof of Concept (PoC)

Location: `com.trusteeplus`

Request					Response				
Pretty	Raw	Hex	GraphQL	GraphQL (InQL - GraphQL Scanner)	Pretty	Raw	Hex	Render	
<pre> 1 POST /graphql HTTP/1.1 2 Host: 52.58.167.8 3 Content-Type: application/json 4 Content-Length: 2744 5 6 { 7 "query": 8 "query InspectionQuery (\n schema (\n # Typically query is called \"Query\" and Mutation \"Mutation\", but there can be redefine ed.\n # For some reason, spec does not force queryType[name] to b e String, but I don't think it can be null.\n queryType (\n name\n \n)\n 'mutationType' can be null if there ar e no mutations.\n mutationType (\n name\n \n)\n subscriptionType (\n name\n \n)\n types (\n name\n \n)\n 'kind' is enum with values: SCALAR, OBJECT, INTERFACE, UNION, ENUM, INPUT_OBJECT, LIST, NON_NULL\n kind\n \n description\n \n # The follo wing are only present for OBJECT and INTERFACE, otherwise null\n fields(includeDeprecated: true) (\n name\n \n)\n description\n \n ... InputValue\n type\n \n \n \n TypeRef\n defReason\n interfaces (\n .. TypeRef\n)\n # The following is only non-null for INTERFACE and UNION:\n possibleTypes (\n .. TypeRef\n)\n # The following is only non-null for ENUM:\n enumValues(includeDeprecated: true) (\n name\n \n)\n deprecationReason\n)\n } </pre>					<pre> 1 HTTP/1.1 200 OK 2 X-Powered-By: Express 3 Access-Control-Allow-Origin: * 4 cache-control: no-store 5 Content-Type: application/json; charset=utf-8 6 Content-Length: 226756 7 ETag: W/"735c4-vBv9DqICuZlYDMXdsCaZgJxwC" 8 Date: Thu, 20 Feb 2025 19:14:30 GMT 9 Connection: keep-alive 10 Keep-Alive: timeout=5 11 12 { "data": { "_schema": { "queryType": { "name": "Query" }, "mutationType": { "name": "Mutation" }, "types": { "name": "Object", "kind": "SCALAR", "description": null, "fields": null, "interfaces": null, "possibleTypes": null, "enumValues": null, "inputFields": null, </pre>				

Request

Pretty	Raw	Hex	GraphQL	GraphQL (inQL - GraphQL Scanner)
1 POST /graphql HTTP/2				
2 Host: apollo.trusteeglobal.com				
3 Content-Type: application/json				
4 Content-Length: 3014				
5				
6				

```

{
  "query":
    "query IntrospectionQuery {
      __schema {
        # Typically query is
        # called 'Query' and mutation 'Mutation', but those can be redefined
        # For some reason, I specify queryType{name} to be
        # String, but I don't think it can be null.\n
        queryType {
          name\n
          # 'mutationType' can be null if there are
          # no mutations.\n
          mutationType {
            name\n
            # TODO: We're not parsing subscriptions and directives at all right
            now\n
            subscriptionType {
              name\n
              # directives {
              #   name\n
              #   types {
              #     name\n
              #     # 'kind' is enum with values: SCALAR, OBJECT, INTERFACE, UNION, ENUM, INPUT_OBJECT, LIST, NON_NULL
              #     kind\n
              #     # For some reason, I specify description{name} to be
              #     # String, but I don't think it can be null.\n
              #     fields(includeDeprecated: true) {
              #       name\n
              #       description\n
              #       args(includeDeprecated: true) {
              #         name\n
              #         type {
              #           name\n
              #           isDeprecated\n
              #           deprecationReason\n
              #         }\n
              #       }\n
              #     }\n
              #     interfaces {
              #       name\n
              #       typeRef\n
              #     }\n
              #     # The following is only non-null for INTERFACE and UNION.\n
              #     possibleTypes {
              #       name\n
              #       # The following is only non-null for ENUM.\n
              #       enumValues(includeDeprecated: true) {
              #         name\n
              #         description\n
              #         isDeprecated\n
              #         deprecationReason\n
              #       }\n
              #     }\n
              #     # The following is only non-null for INPUT_OBJECT.\n
              #     inputFields(includeDeprecated: true) {
              #       name\n
              #       type\n
              #       # The following is only non-null for LIST and NON_NULL.\n
              #       ofType\n
              #       # Only (optional) non-null for custom scalars
              #       specifiedByURL\n
              #     }\n
              #     nonfragmentInputValue\n
              #     onInputValue {
              #       name\n
              #       description\n
              #       type\n
              #       # ...T
            "

```

Response

Pretty	Raw	Hex	Render
1 HTTP/2 200 OK			
2 Date: Tue, 18 Feb 2025 14:15:06 GMT			
3 Content-Type: application/json; charset=utf-8			
4 Content-Length: 226756			
5 X-Powered-By: Express			
6 Access-Control-Allow-Origin: *			
7 Cache-Control: no-store			
8 Etag: W/375c4-bV9p0lqEuIZYDMX5CaZgXwc"			
9 CF-Cache-Status: DYNAMIC			
10 Report-To: { "endpoints": [{ "url": "https://v47s.net.cloudflare.com/report/v47s-JASR5m2Scn uIMN428kmsES95923xglt3j2F0z2F28021v1fIbbjJlCrt6p8YDv5HsRtdovFXjAlE 29428942F2cfyAFwL12b2ygyG2HwE6Rn1I3HrY1842BMH4Xc0ueYvQPQCIBW8Qv43D3d "}, "group": "cf-nel", "max_age": 604800 }], "nel": { "success_fraction": 0, "report_to": "cf-nel", "max_age": 604800 } }, "cf-ray": 913ea9b2df2348e-WAW } Server: cloudflare CF-Ray: 913ea9b2df2348e-WAW Server-Timing: cfI4;desc="3prontCP6rt7t242196mln_rtt239646rtt_var79085cent146rcv280i0 st=86trans=0scent_bytes7816rcv_bytes4314deliverv_rate=604236cwnd=2516m sent_bytes=86cid=9476916e3d3ade36ts=1796x=0" }			
11 { "data": { "schema": { "queryType": { "name": "Query" }, "mutationType": { "name": "Mutation" }, "types": [{ "name": "Object", "kind": "SCALAR", "description": null, "fields": null, }] } } }			

As we can see from the responses, this is the same server, but the original IP address does not have a waf

[F-2025-8874](#) - GraphQL Introspection Query Exposure - Low

Description:

GraphQL introspection has been enabled on the [apollo.trusteeglobal.com](#) API, allowing unauthenticated users to retrieve detailed information about the API schema. This includes query and mutation names, data types, and possible arguments.

Attackers can use this information to map out the API, identify potential vulnerabilities, and craft targeted attacks such as unauthorized access, privilege escalation, or injection-based exploits.

Assets:

- com.trusteepus

Status:

Accepted

Classification

Vulnerability Type(s):

- Gain Information

CVSS 4.0:

2.3 (AV:A/AC:H/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N)

Weakness (CWE):

- [CWE-200 | Exposure of Sensitive Information to an Unauthorized Actor](#)

Severity:

Low

Recommendations

Remediation:

- Disable introspection queries in the production environment to prevent schema enumeration.
- Implement proper authentication and authorization for accessing the GraphQL API.
- Utilize rate limiting and logging mechanisms to detect and mitigate excessive query attempts.

Evidences

Proof of Concept (PoC)

Location: com.trusteeplus

Reproduce:

- Send the following GraphQL introspection query to the API endpoint:

```
POST /graphql HTTP/2
Host: apollo.trusteeglobal.com
Content-Type: application/json

{"query": "query IntrospectionQuery { __schema { queryType { name } mutationType { name } types { name kind description fields { name type { name kind } } } } }"}"
```

- Observe the API response containing the full schema details.
- Identify available queries, mutations, and sensitive data structures.

Results:

The API responded with a complete schema definition, including all available queries, mutations, and object types.

```

  v  InQL
  v  apollo.trusteeglobal.com
    v  2025-02-17_185824
      v  mutations
        activatePhysicalCard.graphql
        addCurrentPricePush.graphql
        addEventBasedNewsAndPush.graphql
        addNewTopUpAddress.graphql
        addNotificationAndPush.graphql
        addTokenToPushNotifications.graphql
        addTokenToPushNotificationsV2.graphql
        addUserToEuAddressWhitelist.graphql
        appAnswerRequest.graphql
        appCheckRequest.graphql
        applyCoupon.graphql
        applyDiscount.graphql
        applyNewYear.graphql
        applyReferralCode.graphql
        approveSourceOfFunds.graphql
        becomeMerchant.graphql
        becomeVerified.graphql
        blockUser.graphql
        blockUserTemporary.graphql
        changeDisabledStatus.graphql
        changeNotificationStatus.graphql
        changePaymentDetailsUsersCustomName.graphql
        changePinTrusteeCard.graphql
        changeTransactionComment.graphql
        changeTrusteeCardSkin.graphql
        changeTrusteeCardTokenStatus.graphql
        checkIban.graphql
        checkMobileNumberForLogin.graphql
        checkMobileNumberForLoginSelectedMethod.graphql

```

[F-2025-8875](#) - Insecure FileProvider Paths Configuration - Low

Description:

Improper configuration of `FileProvider` paths has been identified in the Trustee Plus Android Application. The issue arises due to the use of `.` (dot) and `/` (slash) in multiple `FileProvider` XML path configurations, which could allow an attacker to gain unauthorized access to arbitrary files on the device.

res/xml/file_provider_paths.xml

```
<paths>
<external-path
name="shared"
path="." />
</paths>
```

- The `external-path` is defined with `path="."`, allowing access to all external storage files.

res/xml/imagepicker_provider_paths.xml

```
<paths>
<cache-path
name="cacheDir"
path="." />
</paths>
```

- The `cache-path` provides unrestricted access to the entire cache directory.

res/xml/ivpusic_imagepicker_provider_paths.xml

```
<paths>
<external-path
name="external_files"
path="." />
</paths>
```

- Similar to the first case, this exposes all external storage files.

res/xml/share_download_paths.xml

```
<paths>
<cache-path
name="rnshare2"
path="/" />
<root-path
name="rnshare_sdcard"
path="." />
</paths>
```

- Using `path="/"` in a `cache-path` allows full access to all files in the root directory.

Assets:

- com.trusteeplus

Status:**Accepted**

Classification**Vulnerability****Type(s):**

- Gain Information
- Gain Privilege

CVSS 4.0:

2.3 (AV:A/AC:H/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N)

Weakness (CWE):

- [CWE-552 | Files or Directories Accessible to External Parties](#)

Severity:**Low**

Recommendations**Remediation:**

- Avoid using `.` or `/` in `FileProvider` paths unless absolutely necessary.
- Restrict file access to specific subdirectories instead of allowing access to entire external or cache directories.
- Use explicit file paths instead of generic ones.
- Validate and sanitize inputs before processing file requests.

[F-2025-8918](#) - Clear Text Traffic Enabled - Low

Description:

Clear text traffic has been found to be enabled for the Trustee Plus Android application. This is configured through the `android:usesCleartextTraffic="true"` attribute in the application's manifest file.

Clear text traffic refers to unencrypted HTTP communication, which poses a security risk as it allows potential attackers to intercept and manipulate sensitive data transmitted between the application and the server. This could lead to various attacks such as man-in-the-middle (MITM) attacks, information disclosure, or unauthorized data modification.

If exploited, an attacker within the same network could eavesdrop on user communications, steal authentication tokens, inject malicious content, or compromise user privacy.

AndroidManifest.xml

```
<application
  android:theme="@style/AppTheme"
  android:label="@string/app_name"
  android:icon="@mipmap/original"
  android:name="com.trusteenative.MainApplication"
  android:allowBackup="false"
  android:supportsRtl="true"
  android:extractNativeLibs="false"
  android:fullBackupContent="@xml/appsflyer_backup_rules"
  android:usesCleartextTraffic="true"
```

Captured unencrypted HTTP requests exposing sensitive data, making it vulnerable to eavesdropping and manipulation.

Assets:

- com.trusteeplus

Status:

Accepted

Classification

Vulnerability Type(s):

- Gain Information

CVSS 4.0:

2.3 (AV:A/AC:H/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N)

Weakness (CWE):

- [CWE-319 | Cleartext Transmission of Sensitive Information](#)

Severity:

Low

Recommendations

Remediation:

To mitigate this risk, clear text traffic should be disabled unless explicitly required. This can be done by setting

`android:usesCleartextTraffic="false"` in the application manifest file.

[F-2025-8917](#) - Application Support Legacy SDK Versions - Info

Description:

The `AndroidManifest.xml` file of the Trustee Plus Android application defines the `minSdkVersion` as `24` and `targetSdkVersion` as `34`. While setting appropriate SDK versions is crucial for compatibility, certain misconfigurations can introduce security risks. If an application targets an outdated SDK or does not enforce proper security mechanisms, it may expose itself to known vulnerabilities that have been patched in later Android versions.

AndroidManifest

```
<uses-sdk  
  android:minSdkVersion="24"  
  android:targetSdkVersion="34" />
```

The manifest configuration shows that the application supports devices starting from API level 24, which may introduce security risks if certain security patches are missing on lower API levels.

Assets:

- `com.trusteeplus`

Status:

Accepted

Classification

Vulnerability

Type(s):

- Gain Information

Weakness (CWE):

- [CWE-115 | Misinterpretation of Input](#)

Severity:

Info

Recommendations

Remediation:

- Ensure that the `targetSdkVersion` is always set to the latest stable Android version to benefit from the latest security enhancements.
- Perform security testing on various Android versions to identify potential misconfigurations.
- Restrict access to sensitive application components through proper permissions and exported settings.

Disclaimers

Hacken Disclaimer

The application given for audit has been analyzed based on best industry practices at the time of the writing of this report, with cybersecurity vulnerabilities and issues in the application's source code, its deployment, and functionality (performing the intended functions) being the focus of our analysis.

The report contains no statements or warranties regarding the identification of all vulnerabilities or the absolute security of the code. The report covers only the code that was submitted and reviewed, and therefore may not remain relevant after any modifications have been made. This report should not be considered a definitive or exhaustive assessment of the utility, safety, or bug-free status of the application, nor should it be taken as a guarantee of the absence of other potential issues.

While we have exerted our best efforts in conducting the analysis and producing this report, it is crucial to understand that this report should not be the sole source of reliance for ensuring the security of the application. We strongly recommend undertaking multiple independent audits and establishing a public bug bounty program to enhance the security posture of the application.

English is the original language of this report. The Consultant is not liable for any errors or omissions in any translations of this report.

Technical Disclaimer

Applications, whether decentralized apps (DApps) or other types of applications, are deployed and run within specific environments that may include various platforms, programming languages, and other related software components. These environments and components can have inherent vulnerabilities that might lead to security breaches. Consequently, the Consultant cannot guarantee the absolute security of the audited application.

Appendix 1. Severity Definitions

Severity	Description
Critical	These issues present a major security vulnerability that poses a severe risk to the system. They require immediate attention and must be resolved to prevent a potential security breach or other significant harm.
High	These issues present a significant risk to the system, but may not require immediate attention. They should be addressed in a timely manner to reduce the risk of the potential security breach.
Medium	These issues present a moderate risk to the system and cannot have a great impact on its function. They should be addressed in a reasonable time frame, but may not require immediate attention.
Low	These issues present no risk to the system and typically relate to the code quality problems or general recommendations. They do not require immediate attention and should be viewed as a minor recommendation.

Appendix 2. Scope

Scope Details	
Android Application	Shared Privately
Whitepaper	https://hackenio.cc/hacken-methodologies