MALWARE ANALYSIS & REVERSE ENGINEERING ON ANDROID APPLICATION

Introduction

Malware analysis is a fundamental component of cybersecurity aimed at understanding the behavior, origin, and potential impact of malicious software. This report documents a comprehensive analysis of an Android APK file suspected of containing malware or security vulnerabilities. The analysis involved reverse engineering the APK to uncover any hidden malicious code, exposed sensitive information, or security misconfigurations that could be exploited by attackers. The findings provide insight into the security posture of the application and inform recommendations to mitigate identified risks.

Objectives

The primary objectives of this malware analysis project were as follows:

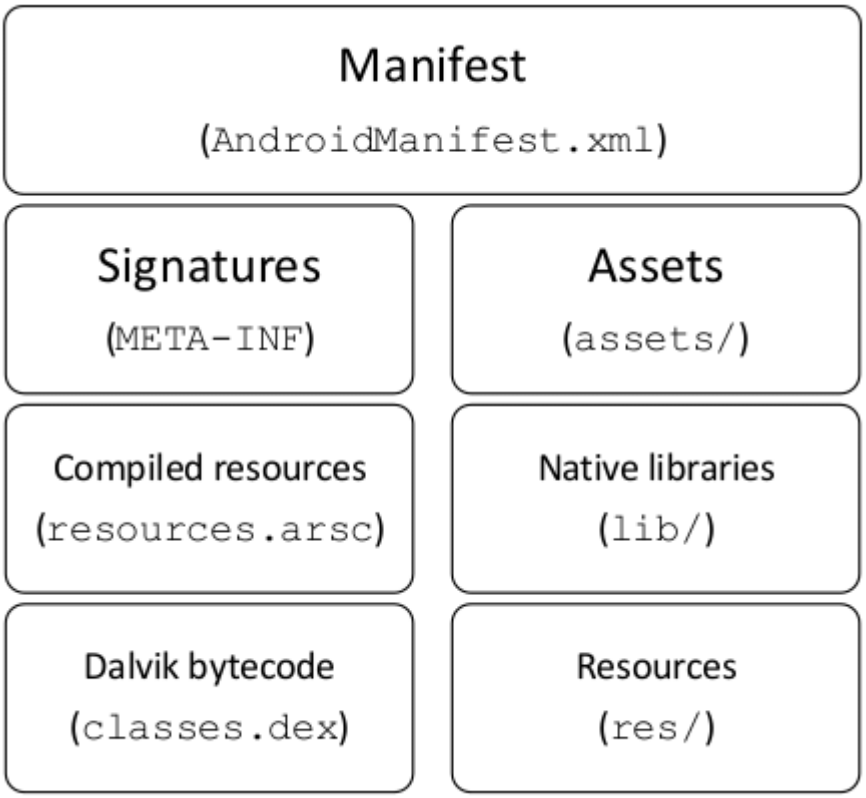
- To establish a robust environment for reverse engineering Android APK files using industry-standard tools.
- To decompile and thoroughly inspect the APK's source code, resources, and manifest files for any malicious or suspicious behavior.
- To identify any exposed sensitive information such as uses permission, credentials, or configuration data embedded within the APK.
- It uses connected to website where APK file listen to connection.
- It is access the internal memory and file creation and delectation without uses permission.
- Checking certifications is having risk or not.

Android Apk File Structure

- **Manifest (AndroidManifest.xml):** Defines app metadata like permissions, components, and system requirements.
- **Signatures (META-INF):** Holds cryptographic signatures to verify the app's integrity and authenticity.
- **Assets (assets/):** Contains raw files the app can access directly, like fonts or media.
- **Compiled Resources (resources.arsc):** Stores precompiled resources such as strings and layouts for efficient access.

- **Native Libraries (lib/):** Includes compiled native code for different CPU architectures.
- **Dalvik Bytecode (classes.dex):** Contains the app's executable code in a format the Android runtime can run.

Resources (res/): Holds compiled UI elements like XML layouts and images referenced by the app.



Tools and Environment

The analysis was conducted within a Kali Linux environment, leveraging the following tools:

JADX: An open source decompiler that converts Android APK bytecode into readable Java source code, facilitating detailed code inspection.

Apktool: A utility for decoding and rebuilding APK resources and manifest files, enabling modification and deeper analysis of application components.

Kali Linux: A penetration testing-focused operating system providing a stable and secure platform for running the analysis tools and executing commands.

Key tool: A certification reader tool it useful for read certificate like public/private airs od associated certificates.

Ghidra :

Ghidra serves as a software reverse engineering (SRE) framework that allows us ers to analyze compiled code without needing access to the original source cod e.

Methodology and Practical Steps

APK Decompilation Using Apktool

The revealing its decompiled Java source code, resource files, and AndroidManifest.xml. The APK's digital signature was verified to confirm that the package had not been tampered with post-signing, ensuring the integrity and authenticity of the application.

Proof of Concept:

```
┌──(charan㉿kali-purple)-[~/Downloads]
└─$ apktool d AdobeReader.apk
I: Using Apktool 2.7.0-dirty on AdobeReader.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/charan/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...

┌──(charan㉿kali-purple)-[~/Downloads]
└─$
```

APK file details:

In this we detailing file like APK. In this we observe files sha value, md5 value and uses permissions.

```
┌──(charan㉿kali-purple)-[~/Downloads]
└─$ file  AdobeReader.apk
AdobeReader.apk: Android package (APK), with AndroidManifest.xml

┌──(charan㉿kali-purple)-[~/Downloads]
└─$ md5sum AdobeReader.apk
93de85925c2848a124de4972b3b2304b  AdobeReader.apk

┌──(charan㉿kali-purple)-[~/Downloads]
└─$ shasum AdobeReader.apk
c990e405708e84bde585bda65ddc1ee9e32f01f1  AdobeReader.apk

┌──(charan㉿kali-purple)-[~/Downloads]
└─$ aapt dump badging AdobeReader.apk | grep "uses-permission"
uses-permission: name='android.permission.INTERNET'
uses-permission: name='android.permission.WRITE_EXTERNAL_STORAGE'
uses-permission: name='android.permission.ACCESS_NETWORK_STATE'
uses-permission: name='com.android.vending.BILLING'
uses-permission: name='android.permission.CAMERA'
uses-permission: name='android.permission.READ_EXTERNAL_STORAGE'
```

Here we APK file using the unrequired permissions like camera, external storage etc.

```
┌──(charan㉿kali-purple)-[~/Downloads/AdobeReader]
└─$ cat apktool.yml
!!brut.androlib.meta.MetaInfo
apkFileName: AdobeReader.apk
compressionType: false
doNotCompress:
- resources.arsc
- png
- gif
- jpg
isFrameworkApk: false
packageInfo:
  forcedPackageId: '127'
  renameManifestPackage: null
sdkInfo:
  minSdkVersion: '15'
  targetSdkVersion: '21'
sharedLibrary: false
sparseResources: false
unknownFiles:
  main/AndroidManifest.xml: '8'
  org/apache/http/entity/mime/version.properties: '8'
usesFramework:
  ids:
  - 1
  tag: null
version: 2.7.0-dirty
versionInfo:
  versionCode: '131440'
  versionName: '15.3'
```

In this file we observe more details like non compares, versions etc.

Key tool: It is used for certification reader a=in public/private keys in APK files**.**

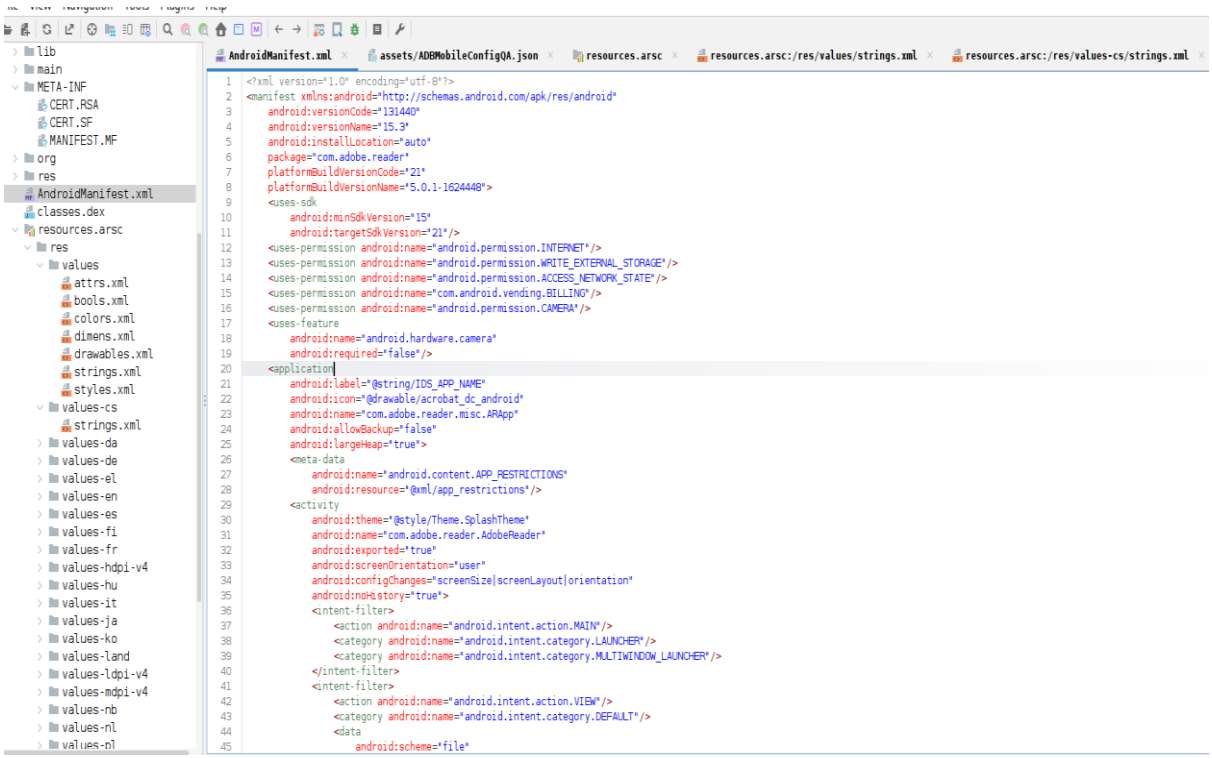
```
┌──(charan㉿kali-purple)-[~/Downloads]
└─$ keytool -printcert -jarfile AdobeReader.apk
Signer #1:

Certificate #1:
Owner: CN=Adobe Systems Incorporated, OU=Adobe Reader, O=Adobe Systems Incorporated, L=San Jose, ST=California, C=U
S
Issuer: CN=Adobe Systems Incorporated, OU=Adobe Reader, O=Adobe Systems Incorporated, L=San Jose, ST=California, C=
US
Serial number: 4be060b7
Valid from: Tue May 04 23:30:23 IST 2010 until: Sat Sep 19 23:30:23 IST 2037
Certificate fingerprints:
         SHA1: C0:7A:0B:5E:C6:F0:1A:57:89:C4:BB:F8:8A:83:03:60:51:4F:02:C5
         SHA256: B6:DD:05:62:25:64:87:FC:D6:C9:8C:DE:13:78:58:EF:50:D9:AD:B9:F9:CD:2F:1C:A5:8C:53:57:EF:DF:0F:AF
Signature algorithm name: SHA1withRSA (weak)
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3


Warning:
The certificate uses the SHA1withRSA signature algorithm which is considered a security risk.
```

We can observe that certificate having security risk.

Setting Up JADX

The JADX tool was downloaded as a compressed ZIP archive and extracted within the Kali Linux environment. The graphical user interface (GUI) was launched successfully, providing an interactive platform for exploring the APK's internal structure, including source code, resources, and manifest files.

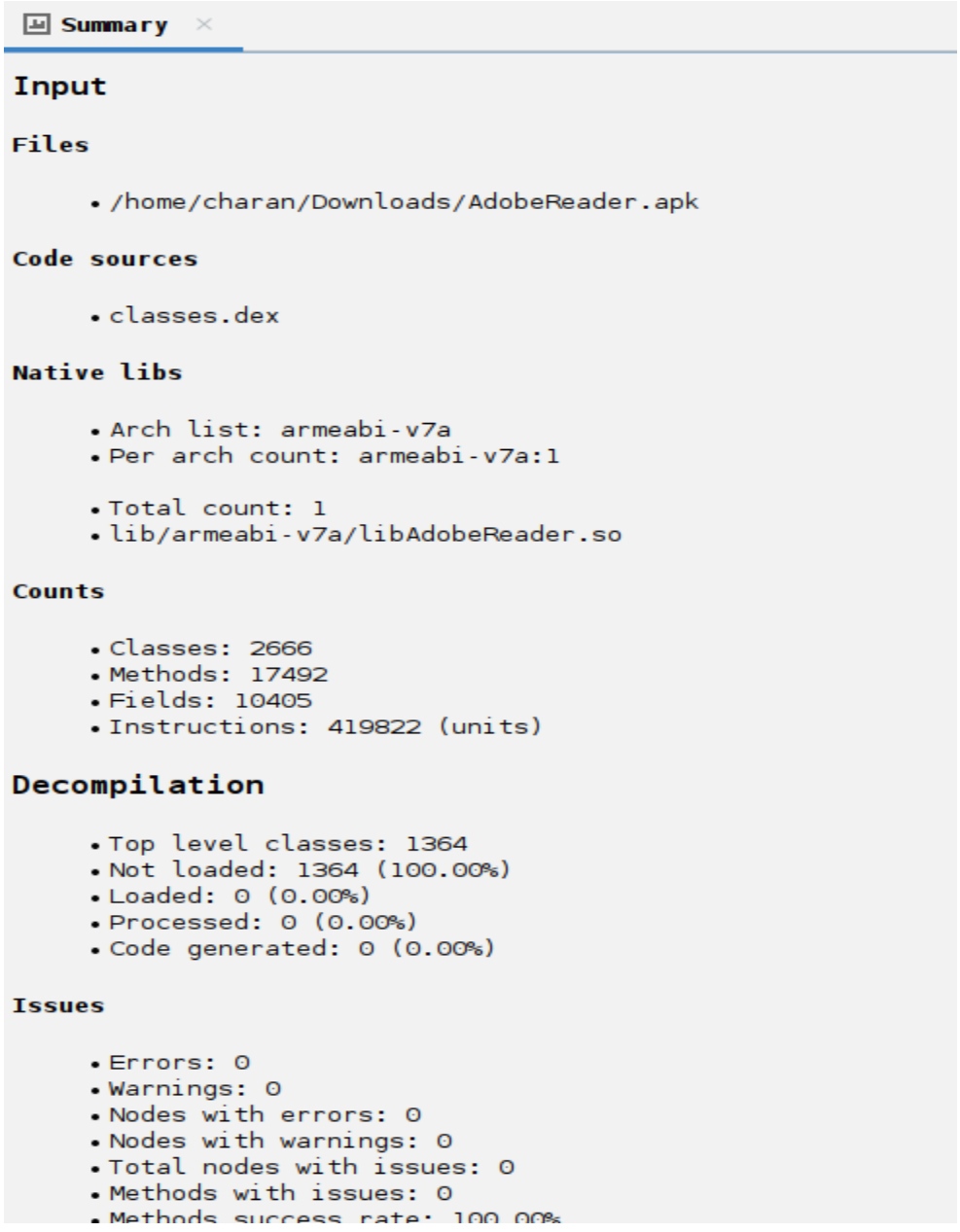
Proof of Concept:



APK Decompilation and Inspection

The target APK was loaded into JADX, revealing its decompiled Java source code, resource files, and AndroidManifest.xml. The APK's digital signature was verified to confirm that the package had not been tampered with post-signing, ensuring the integrity and authenticity of the application.

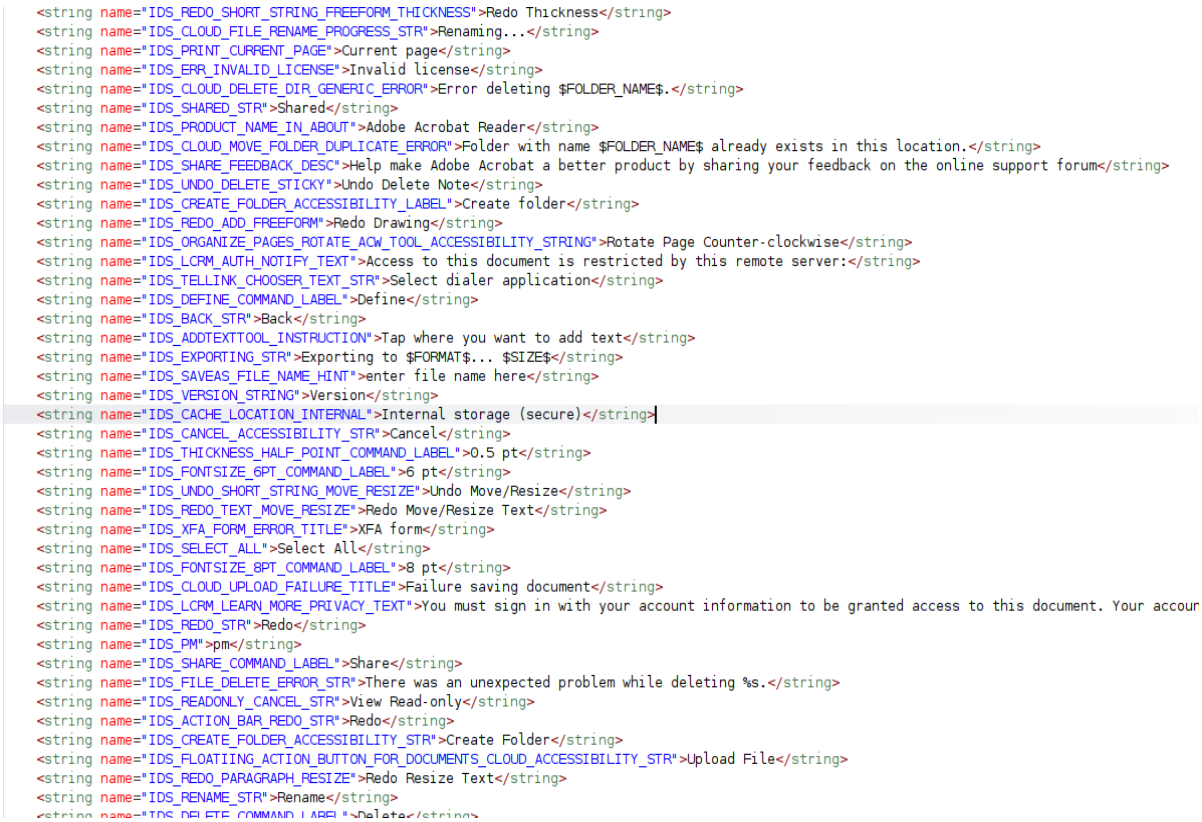
Proof of Concept:



In this we observer how many classes is decompiled and non-decompiled.Any issues in core code.

Sensitive Information Discovery

Using JADX's search functionality, an exposed string labeled internal memory access was identified within the source code. This string was cross-referenced against Firebase configurations embedded in the APK to assess potential misuse or unauthorized access.



```
<string name="IDS_REDO_SHORT_STRING_FREEFORM_THICKNESS">Redo Thickness</string>
<string name="IDS_CLOUD_FILE_RENAME_PROGRESS_STR">Renaming...</string>
<string name="IDS_PRINT_CURRENT_PAGE">Current page</string>
<string name="IDS_ERR_INVALID_LICENSE">Invalid license</string>
<string name="IDS_CLOUD_DELETE_DIR_GENERIC_ERROR">Error deleting $FOLDER_NAME$.</string>
<string name="IDS_SHARED_STR">Shared</string>
<string name="IDS_PRODUCT_NAME_IN_ABOUT">Adobe Acrobat Reader</string>
<string name="IDS_CLOUD_MOVE_FOLDER_DUPLICATE_ERROR">Folder with name $FOLDER_NAME$ already exists in this location.</string>
<string name="IDS_SHARE_FEEDBACK_DESC">Help make Adobe Acrobat a better product by sharing your feedback on the online support forum</string>
<string name="IDS_UNDO_DELETE_STICKY">Undo Delete Note</string>
<string name="IDS_CREATE_FOLDER_ACCESSIBILITY_LABEL">Create folder</string>
<string name="IDS_REDO_ADD_FREEFORM">Redo Drawing</string>
<string name="IDS_ORGANIZE_PAGES_ROTATE_ACW_TOOL_ACCESSIBILITY_STRING">Rotate Page Counter-clockwise</string>
<string name="IDS_LCRM_AUTH_NOTIFY_TEXT">Access to this document is restricted by this remote server:</string>
<string name="IDS_TELLINK_CHOOSER_TEXT_STR">Select dialer application</string>
<string name="IDS_DEFINE_COMMAND_LABEL">Define</string>
<string name="IDS_BACK_STR">Back</string>
<string name="IDS_ADDTEXTTOOL_INSTRUCTION">Tap where you want to add text</string>
<string name="IDS_EXPORTING_STR">Exporting to $FORMAT$... $SIZE$</string>
<string name="IDS_SAVEAS_FILE_NAME_HINT">enter file name here</string>
<string name="IDS_VERSION_STRING">Version</string>
<string name="IDS_CACHE_LOCATION_INTERNAL">Internal storage (secure)</string>
<string name="IDS_CANCEL_ACCESSIBILITY_STR">Cancel</string>
<string name="IDS_THICKNESS_HALF_POINT_COMMAND_LABEL">0.5 pt</string>
<string name="IDS_FONTSIZE_6PT_COMMAND_LABEL">6 pt</string>
<string name="IDS_UNDO_SHORT_STRING_MOVE_RESIZE">Undo Move/Resize</string>
<string name="IDS_REDO_TEXT_MOVE_RESIZE">Redo Move/Resize Text</string>
<string name="IDS_XFA_FORM_ERROR_TITLE">XFA form</string>
<string name="IDS_SELECT_ALL">Select All</string>
<string name="IDS_FONTSIZE_8PT_COMMAND_LABEL">8 pt</string>
<string name="IDS_CLOUD_UPLOAD_FAILURE_TITLE">Failure saving document</string>
<string name="IDS_LCRM_LEARN_MORE_PRIVACY_TEXT">You must sign in with your account information to be granted access to this document. Your accour
<string name="IDS_REDO_STR">Redo</string>
<string name="IDS_PM">pm</string>
<string name="IDS_SHARE_COMMAND_LABEL">Share</string>
<string name="IDS_FILE_DELETE_ERROR_STR">There was an unexpected problem while deleting %s.</string>
<string name="IDS_READONLY_CANCEL_STR">View Read-only</string>
<string name="IDS_ACTION_BAR_REDO_STR">Redo</string>
<string name="IDS_CREATE_FOLDER_ACCESSIBILITY_STR">Create Folder</string>
<string name="IDS_FLOATIING_ACTION_BUTTON_FOR_DOCUMENTS_CLOUD_ACCESSIBILITY_STR">Upload File</string>
<string name="IDS_REDO_PARAGRAPH_RESIZE">Redo Resize Text</string>
<string name="IDS_RENAME_STR">Rename</string>
<string name="IDS_DELETE_COMMAND_LABEL">Delete</string>
```

APK Decompilation Using Unzip tool

To gain deeper insight into the APK's internal structure and code, the **unzip** utility was employed. Unzip is a powerful tool that decodes Android APK resources and disassembles the Dalvik bytecode into smali files, which are human-readable representations of the app's compiled code. This process enables detailed static analysis of the application's behavior at the bytecode level.

The following steps were performed:

- The APK file was decompiled the uncompiled files using the command: Copy

unzip -l file name .APK

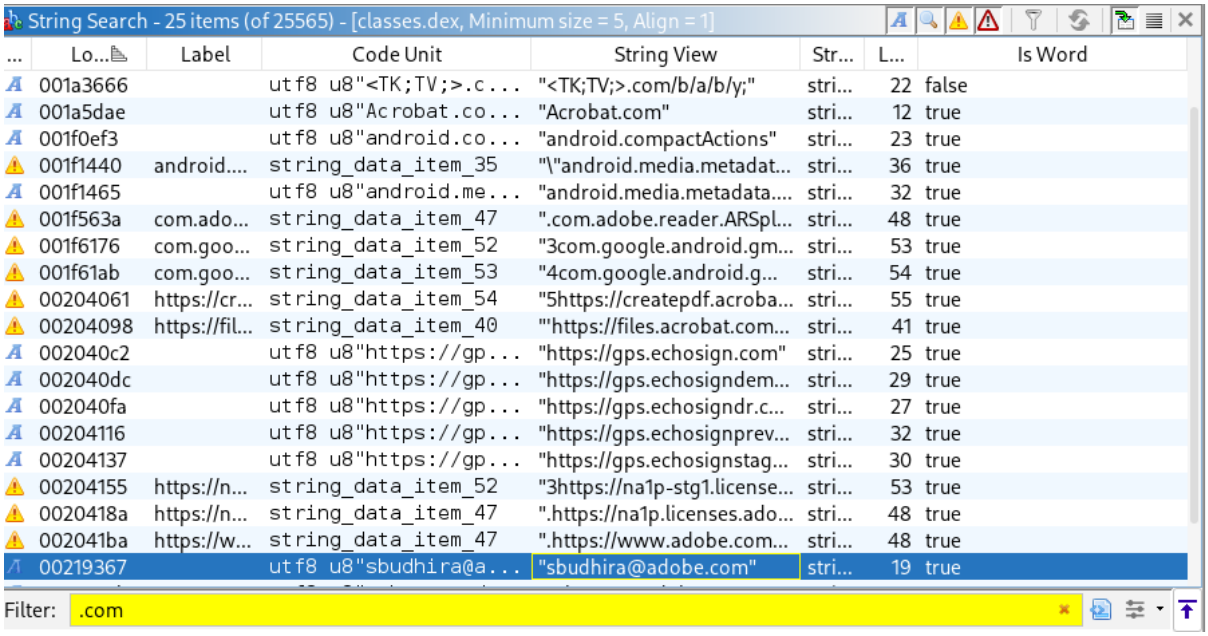
unzip file name .APK

```
Session  Actions  Edit  View  Help
└─$ unzip -l AdobeReader.apk
Archive:  AdobeReader.apk
  Length      Date    Time    Name
---------  ---------- -----   ----
    20348  2015-11-08 23:59   AndroidManifest.xml
      948  2015-11-08 23:59   assets/ADBMobileConfigProd.json
      947  2015-11-08 23:59   assets/ADBMobileConfigQA.json
    28024  2015-11-08 23:59   assets/fonts/AdobeSansF2-Regular.otf
    57137  2015-11-08 23:59   assets/getting_started.pdf
    48687  2015-11-08 23:59   assets/javascript/AForm.js
     3488  2015-11-08 23:59   assets/javascript/App.js
     8256  2015-11-08 23:59   assets/javascript/Doc.js
     1206  2015-11-08 23:59   assets/javascript/EScriptString.js
     7675  2015-11-08 23:59   assets/javascript/Event.js
     7129  2015-11-08 23:59   assets/javascript/Field.js
    19504  2015-11-08 23:59   assets/javascript/Util.js
      758  2015-11-08 23:59   assets/javascript/index.html
    16503  2015-11-08 23:59   assets/javascript/sprintf.js
     2013  2015-11-08 23:59   assets/javascript/utils.js
      352  2015-11-08 23:59   res/anim/context_menu_fade_in.xml
      352  2015-11-08 23:59   res/anim/context_menu_fade_out.xml
      284  2015-11-08 23:59   res/anim/cycles.xml
      352  2015-11-08 23:59   res/anim/fade_out.xml
      400  2015-11-08 23:59   res/anim/shake.xml
      640  2015-11-08 23:59   res/anim/toolbar_slide_in_bottom.xml
      640  2015-11-08 23:59   res/anim/toolbar_slide_in_top.xml
      640  2015-11-08 23:59   res/anim/toolbar_slide_out_bottom.xml
      640  2015-11-08 23:59   res/anim/toolbar_slide_out_top.xml
      720  2015-11-08 23:59   res/color/account_entry_subscribe_text_color_selector.xml
      844  2015-11-08 23:59   res/drawable-hdpi-v4/a12_addfile.png
      771  2015-11-08 23:59   res/drawable-hdpi-v4/a12_createpdfpack.png
      663  2015-11-08 23:59   res/drawable-hdpi-v4/a12_createpdfpack_select.png
      469  2015-11-08 23:59   res/drawable-hdpi-v4/a12_deletetextdyn.png
      473  2015-11-08 23:59   res/drawable-hdpi-v4/a12_deletetextdyn_selected.png
     1454  2015-11-08 23:59   res/drawable-hdpi-v4/a12_documentcloud.png
     1114  2015-11-08 23:59   res/drawable-hdpi-v4/a12_documentcloud_select.png
      454  2015-11-08 23:59   res/drawable-hdpi-v4/a12_editpdfdc.png
      412  2015-11-08 23:59   res/drawable-hdpi-v4/a12_editpdfdc_select.png
     1192  2015-11-08 23:59   res/drawable-hdpi-v4/a12_exportpdf.png
     1127  2015-11-08 23:59   res/drawable-hdpi-v4/a12_exportpdf_disabled.png
      957  2015-11-08 23:59   res/drawable-hdpi-v4/a12_exportpdf_select.png
      281  2015-11-08 23:59   res/drawable-hdpi-v4/a12_folder_back.png
     1010  2015-11-08 23:59   res/drawable-hdpi-v4/a12_handbook.png
      865  2015-11-08 23:59   res/drawable-hdpi-v4/a12_handbook_select.png
      793  2015-11-08 23:59   res/drawable-hdpi-v4/a12_menucheckmark.png
      240  2015-11-08 23:59   res/drawable-hdpi-v4/a12_menucheckmark_disabled.png
      691  2015-11-08 23:59   res/drawable-hdpi-v4/a12_menucheckmarkdisabled.png
     1036  2015-11-08 23:59   res/drawable-hdpi-v4/a12_mobilelinkoff.png
     1016  2015-11-08 23:59   res/drawable-hdpi-v4/a12_mobilelinkoffline.png
      909  2015-11-08 23:59   res/drawable-hdpi-v4/a12_mobilelinkon.png
      856  2015-11-08 23:59   res/drawable-hdpi-v4/a12_nextfield.png
      827  2015-11-08 23:59   res/drawable-hdpi-v4/a12_nextfield_disabled.png
```

In this we observe un decompiled files like resource.rsrc, AndroidMainfest.xml etc.
All are un decompiled files it gives the death understanding.

Setting Up Ghidra

The Ghidra tool was downloaded as a compressed ZIP archive and extracted within the Kali Linux environment. The graphical user interface (GUI) was launched successfully, providing an interactive platform for exploring the APK's internal structure, classes.dev.



In this ghidra tool we find this APK file is connect to some websites and gmail it showing in fig. The website are locations, gmail, android meta etc.

We observe that mail is valid where I check in real time.

Findings

The analysis yielded the following key findings:

The APK was properly signed, confirming that it had not been altered after the signing process.

- An exposed **internal storage access** was discovered within the APK's source code, representing a potential security risk if misused.
- It connecting to malicious websites and sending gps to mail.
- No additional critical vulnerabilities, suspicious permissions, or malicious code were identified during the inspection.

```
<string name="IDS_REDO_SHORT_STRING_FREEFORM_THICKNESS">Redo Thickness</string>
<string name="IDS_CLOUD_FILE_RENAME_PROGRESS_STR">Renaming...</string>
<string name="IDS_PRINT_CURRENT_PAGE">Current page</string>
<string name="IDS_ERR_INVALID_LICENSE">Invalid license</string>
<string name="IDS_CLOUD_DELETE_DIR_GENERIC_ERROR">Error deleting $FOLDER_NAME$.</string>
<string name="IDS_SHARED_STR">Shared</string>
<string name="IDS_PRODUCT_NAME_IN_ABOUT">Adobe Acrobat Reader</string>
<string name="IDS_CLOUD_MOVE_FOLDER_DUPLICATE_ERROR">Folder with name $FOLDER_NAME$ already exists in this location.</string>
<string name="IDS_SHARE_FEEDBACK_DESC">Help make Adobe Acrobat a better product by sharing your feedback on the online support forum</string>
<string name="IDS_UNDO_DELETE_STICKY">Undo Delete Note</string>
<string name="IDS_CREATE_FOLDER_ACCESSIBILITY_LABEL">Create folder</string>
<string name="IDS_REDO_ADD_FREEFORM">Redo Drawing</string>
<string name="IDS_ORGANIZE_PAGES_ROTATE_ACW_TOOL_ACCESSIBILITY_STRING">Rotate Page Counter-clockwise</string>
<string name="IDS_LCRM_AUTH_NOTIFY_TEXT">Access to this document is restricted by this remote server:</string>
<string name="IDS_TELLINK_CHOOSER_TEXT_STR">Select dialer application</string>
<string name="IDS_DEFINE_COMMAND_LABEL">Define</string>
<string name="IDS_BACK_STR">Back</string>
<string name="IDS_ADDTEXTTOOL_INSTRUCTION">Tap where you want to add text</string>
<string name="IDS_EXPORTING_STR">Exporting to $FORMAT$... $SIZE$</string>
<string name="IDS_SAVEAS_FILE_NAME_HINT">enter file name here</string>
<string name="IDS_VERSION_STRING">Version</string>
<string name="IDS_CACHE_LOCATION_INTERNAL">Internal storage (secure)</string>
<string name="IDS_CANCEL_ACCESSIBILITY_STR">Cancel</string>
<string name="IDS_THICKNESS_HALF_POINT_COMMAND_LABEL">0.5 pt</string>
<string name="IDS_FONTSIZE_6PT_COMMAND_LABEL">6 pt</string>
<string name="IDS_UNDO_SHORT_STRING_MOVE_RESIZE">Undo Move/Resize</string>
<string name="IDS_REDO_TEXT_MOVE_RESIZE">Redo Move/Resize Text</string>
<string name="IDS_XFA_FORM_ERROR_TITLE">XFA form</string>
<string name="IDS_SELECT_ALL">Select All</string>
<string name="IDS_FONTSIZE_8PT_COMMAND_LABEL">8 pt</string>
<string name="IDS_CLOUD_UPLOAD_FAILURE_TITLE">Failure saving document</string>
<string name="IDS_LCRM_LEARN_MORE_PRIVACY_TEXT">You must sign in with your account information to be granted access to this document. Your accour
<string name="IDS_REDO_STR">Redo</string>
<string name="IDS_PM">pm</string>
<string name="IDS_SHARE_COMMAND_LABEL">Share</string>
<string name="IDS_FILE_DELETE_ERROR_STR">There was an unexpected problem while deleting %s.</string>
<string name="IDS_READONLY_CANCEL_STR">View Read-only</string>
<string name="IDS_ACTION_BAR_REDO_STR">Redo</string>
<string name="IDS_CREATE_FOLDER_ACCESSIBILITY_STR">Create Folder</string>
<string name="IDS_FLOATIING_ACTION_BUTTON_FOR_DOCUMENTS_CLOUD_ACCESSIBILITY_STR">Upload File</string>
<string name="IDS_REDO_PARAGRAPH_RESIZE">Redo Resize Text</string>
<string name="IDS_RENAME_STR">Rename</string>
<string name="IDS_DELETE_COMMAND_LABEL">Delete</string>
```

Impact Assessment

Exposing internal storage within an APK increases the attack surface by potentially allowing unauthorized parties to access backend services if security controls are insufficient. Although Firebase denied access in this case, the presence of such keys in client-side code is a recognized security risk. If backend access rules are misconfigured or keys are reused across services, attackers could exploit these weaknesses to compromise data integrity, privacy, or service availability.

Mitigation Strategies

To mitigate the risks associated with exposed sensitive information and improve the overall security posture, the following strategies are recommended:

- Avoid embedding internal storage access directly within APKs. Instead, use secure storage mechanisms or environment variables that are not exposed in the client application.
- Harden Firebase and other backend service access rules by implementing strict authentication and authorization policies to prevent unauthorized use of API keys.
- Employ code obfuscation techniques to complicate reverse engineering efforts and reduce the likelihood of sensitive information disclosure.

- Conduct regular security audits of APKs and backend configurations to identify and remediate vulnerabilities proactively.

Conclusion

This project successfully demonstrated the process of setting up reverse engineering tools and conducting a detailed analysis of an Android APK for malware and security vulnerabilities. The discovery of an exposed API key underscores the critical importance of secure key management and robust backend access controls. By implementing the recommended mitigation strategies, developers and security teams can significantly reduce the risk of exploitation and enhance the security of their mobile applications.

References

- JADX GitHub Repository: https://github.com/skylot/jadx
- Apktool Official Website: https://ibotpeaches.github.io/Apktool/
- Firebase Security Rules Documentation: https://firebase.google.com/docs/rules
- Android Security Best Practices: https://developer.android.com/topic/security/bestpractices.
- Android pen testing: https://www.youtube.com/watch?v=6-M_7O3A8AI&t=849s