threatsys

Your 360° Cyber Security Partner

Threatsys Technologies Pvt. Ltd.

ANDROID APP PENETRATION TESTING

A general guidance on decoding Android applications

Operating Globally

Email: **sales@threatsys.co.in**
Website: **www.threatsys.co.in**

threatsys

Your 360° Cyber Security Partner

Android Application Penetration Testing:

A general guidance on decoding Android applications

Mobile application penetration testing is commonly used for identifying weaknesses and vulnerabilities in mobile applications as part of cybersecurity procedures. Android penetration testing involves critical, high, medium, and low-level debugging, utilising tools like JADX-GUI, Drozer and Android Studio. Decoding the debugging procedures involves acknowledging the application structure and revealing the unknown secrets related to Application vulnerabilities.

Decoding Apps: Reversing Easy APK with JADX-GUI

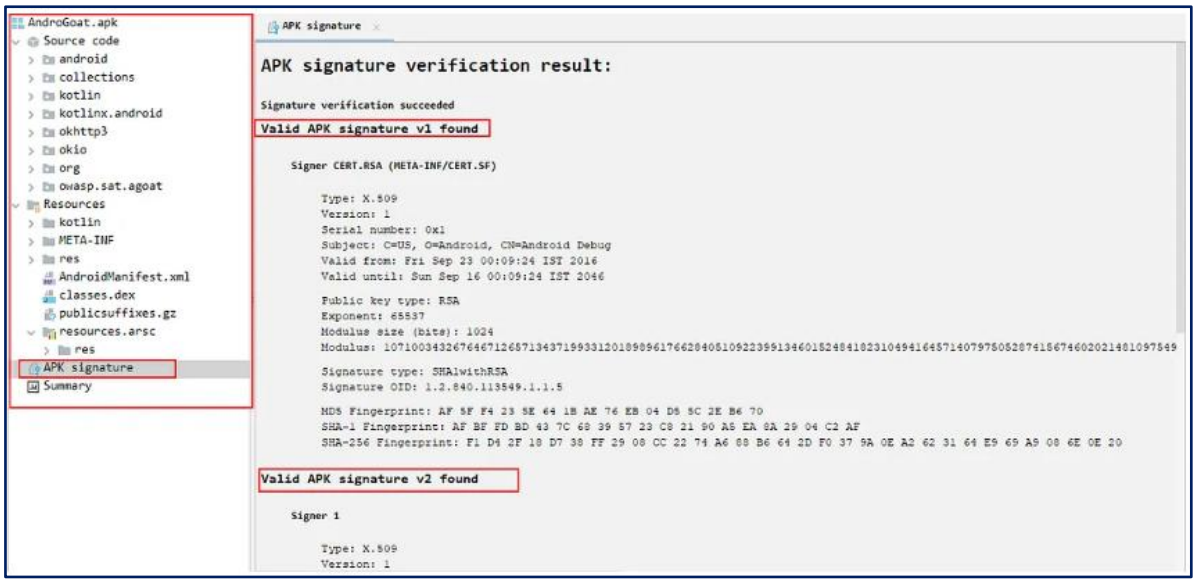
JADX-GUI helps professionals to extract APKs in terms of v1, v2, and v3 according to their APK versions.

v1 (JAR Signature) defines an Android application in terms of originality and legacy signing schemes. This method is efficiently suitable for any Android version due to the utilisation of Java Archive signing procedures.

v2 (APK Signature Scheme v2) is defined as the upgrade of the original signing process established in Android 7.0. This enhancement scheme compiles with a separate APK signature block to increase the efficiency of the verification process and limit the signing-related hazards.

v3 (APK Signature Scheme v3) is also described as an improvement for the signing process for Android 9.0 that includes supportive key rotation similar to v2. This scheme additionally introduces strengthening security features for app signing.

When the application is only signed with v1, it's considered less secure for the users. It has a lot of absence of modern security features and options that are incorporated with the latest Android versions to enhance the potential to acknowledge the potential or certain types of Cyber-attacks.



The professional needs to open the manifest file and recognise the API level, which should be above 21, and also needs to be the minimum SDK version. API level is recognised as a numerical identification tool that is utilised for recognising the Android operating system different versions. This level is recognised by the teachers and the abilities of each Android version, which is provided by the developer while creating the applications. The high level of API represents that developers can access more functionality and create more enhancements in the Android versions.

Analysing and identifying the unwanted permissions

Checking the components is essential to acknowledge the cybersecurity-related errors or any attacks. While utilising the Android export attribute, showcase the result through but it should be false, which is a clear indication of any real-life consequences.

Unauthorised access makes the export attribute through, and it provides access to the other third-party applications or entities to allow access to the sensitive information. This also leads

to creating data leakage and exposing users' sensitive data and personal information that should be restricted. Mini cyber-attacks create manipulation with the application by creating security breaches and lead to unauthorised control through remote code execution.

Allow backup is a procedure that helps the application to allow the data backup in an AndroidManifest.xml file format. It is easier to utilise for users, but it increases the risk of cybersecurity-related attacks and creates potential vulnerabilities. The attackers mostly target these Store backups to expose the sensitive data and violate the data privacy compliance. For this reason, encrypted data storage backup is essential as a security storage practice to minimise the potential risk of third-party entities.

When setting debuggable to true in the AndroidManifest.xml automatically opens the debugging mode on the application. It is easier for development but creates a strong security risk potential in the aspect of the production environment. Debuggable apps can be more sensitive towards exposing the private information of the users by unauthorised access or reverse engineering. During the setting, debuggable needs to be false to reduce the potential of cybersecurity-related risk and exposure of sensitive data.

```
android:debuggable="true" android:allowBackup="true" a
```

Setting the Android: usescleartexttraffic to true in the AndroidManifest.xml file provides access to the unencrypted network traffic. This creates a major security risk or challenges for the users that expose sensitive information during transmission. Professionals are recommended to set the result to false to provide security by utilising secure communication

protocols like HTTPS to provide clear protection and encrypted data to prevent unauthorised access or third-party integration.

```
android:usesCleartextTraffic="true"
```

Manual inspection and automated analysis are essential for identifying obfuscated source code.

Variable and method names are also considered as methods, where the short names for methods and variables can create challenges for the developers or the testers to acknowledge the purpose of the code.

Professionals face difficulties in understanding the code's logic or purpose due to obfuscation tools that remove any documentation from the comments of the specific code.

Professionals commonly identify obfuscation patterns, which are mostly the single-letter names or variables or any other alphanumeric sequence.

The obfuscation tool also helps in identifying the unused code that can create major confusion during the analysis.

Firebase databases provide the ability to the developers to store data and sync the stored data into the NoSQL cloud-hosted database efficiently. A network call can help in identifying a misconfigured firewall dataset.

Professionals copy the URL and paste it into the browser to acknowledge the output as shown in the picture below

Professional checks each code according to the API keys, internal IP address, hard-coded certifications, usernames, passwords and many other things.

WebView in an Android application helps developers in fixing the web browser in their Android application. It allows developers to directly analyse the internal user interface by displaying the web content related to different websites and web pages.

Unlocking Secrets: Exploring/Exploiting Internal Components with Drozer

Drozer defines it as a security analysis framework that provides an assistant to acknowledge the cyber-attacks and their potential impact vulnerabilities, along with testing security controls by third-party applications. After the installation of Drozer, the procedures help in recognising the interaction between different components.

Firstly, the shown button needs to be on in the Drozer agent.

Need the full report with all the bits and bytes?

🔗 Find the download link in the comments 👇