

Mobile Security TA-2

Name: Sanjay Kumar

Enrolment No.: 032200300002036

Frida:-

Frida is an open-source dynamic instrumentation framework that allows developers and security researchers to inject and modify code in running applications on various platforms, including mobile devices. It provides a powerful set of tools and APIs that enable deep analysis, debugging, and manipulation of mobile applications at runtime.

Features of Frida:-

Frida offers a wide range of features.

- 1) **Dynamic Code Injection:** Frida allows developers to inject and execute custom code into a running application without requiring source code modifications or recompilation.
- 2) **Cross-Platform Support:** Frida supports multiple platforms, including Android and iOS. It provides a unified API and scripting interface, allowing users to write scripts that can work seamlessly on both platforms.
- 3) **High-Level API:** Frida provides a high-level API that offers a comprehensive set of functions for interacting with the target application. This includes features such as method hooking, data access and modification, network monitoring, file system interception, and more.
- 4) **Dynamic Hooking:** Frida allows developers to intercept and modify method calls in real-time. This feature is particularly useful for analyzing and modifying the behavior of an application, such as bypassing certain checks, tampering with data, or tracing specific functions.
- 5) **Code Obfuscation and Anti-Reversing:** Frida can be used to analyze and bypass code obfuscation and anti-reversing techniques employed by mobile applications. By injecting custom code and manipulating the runtime environment, Frida can help in understanding and reverse engineering complex obfuscated code.

Installation of Frida:-

Python can be used to install frida in your system.

Command:- `pip3 install frida-tools`

To use frida we have to install the frida server in our android mobile.

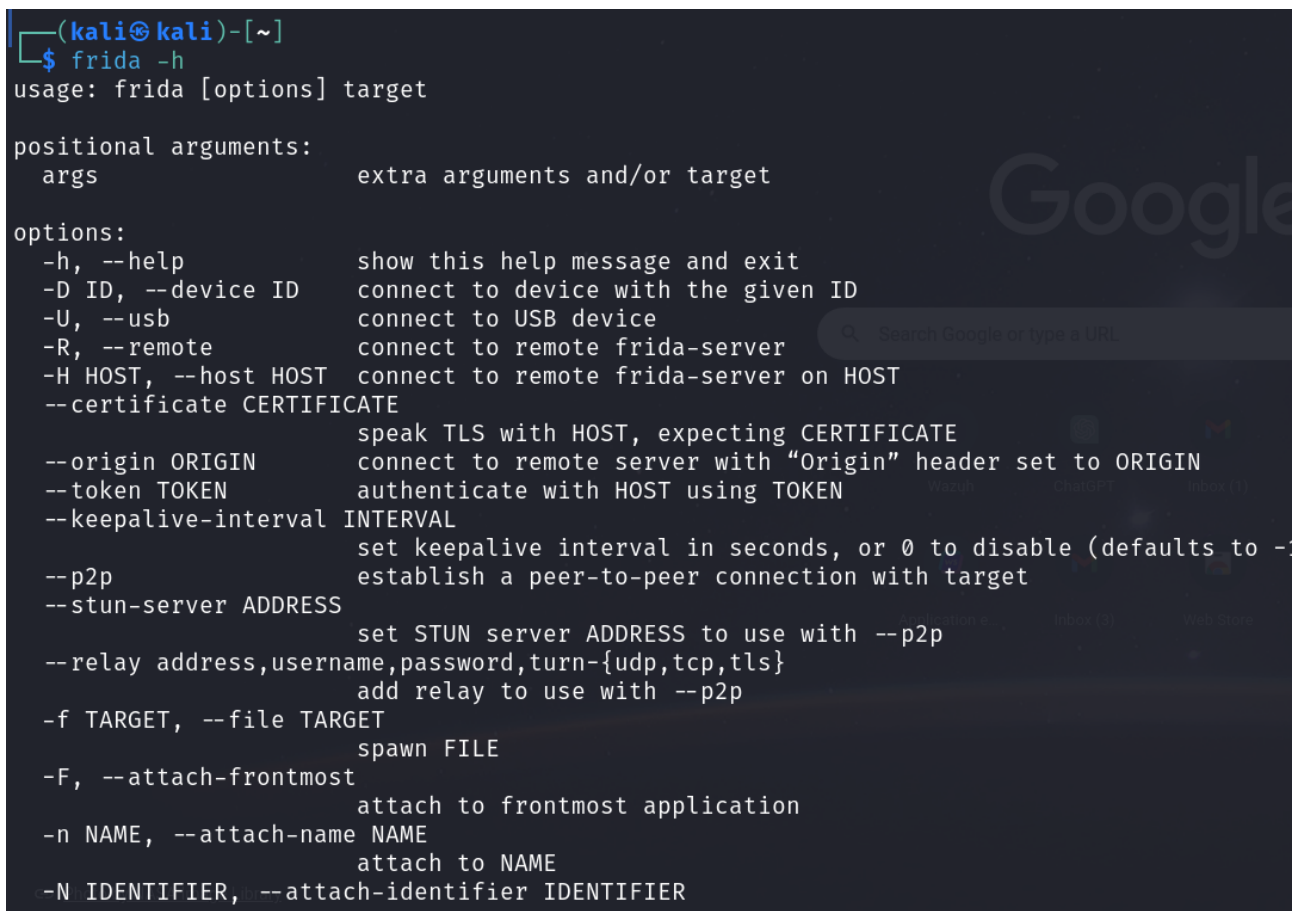
Commands to install frida server in android using adb.

```
adb root
adb push frida-server /data/local/tmp/
adb shell "chmod 755 /data/local/tmp/frida-server"
adb shell "/data/local/tmp/frida-server &"
```

Steps to use frida:-

- 1) Set up Frida Server on the Target Device:- To interact with a mobile application on a target device we need to install the Frida server on it.
- 2) Choose a Scripting Language:- Decide on the scripting language you want to use with Frida. Frida primarily supports JavaScript, but it also offers bindings for other languages like Python and Swift.
- 3) Connect your development machine to the target device or emulator running the Frida server. You can establish a connection using various methods, such as USB, Wi-Fi, or remote connections.
- 4) Use the Frida client to inject the Frida script into the target application. The script injection process varies slightly between platforms.
- 5) For Android: Use the `frida-ps -U` command to list the running processes on the device. Identify the process ID (PID) of the target application. Then, use the **`frida -U -l <script.js> -f <package_name>`** command to inject the script into the target application.

Figures:-



```
(kali㉿kali)-[~]
$ frida -h
usage: frida [options] target

positional arguments:
  args                extra arguments and/or target

options:
  -h, --help            show this help message and exit
  -D ID, --device ID    connect to device with the given ID
  -U, --usb             connect to USB device
  -R, --remote          connect to remote frida-server
  -H HOST, --host HOST  connect to remote frida-server on HOST
  --certificate CERTIFICATE
                        speak TLS with HOST, expecting CERTIFICATE
  --origin ORIGIN       connect to remote server with "Origin" header set to ORIGIN
  --token TOKEN         authenticate with HOST using TOKEN
  --keepalive-interval INTERVAL
                        set keepalive interval in seconds, or 0 to disable (defaults to 30)
  --p2p                establish a peer-to-peer connection with target
  --stun-server ADDRESS
                        set STUN server ADDRESS to use with --p2p
  --relay address,username,password,turn-{udp,tcp,tls}
                        add relay to use with --p2p
  -f TARGET, --file TARGET
                        spawn FILE
  -F, --attach-frontmost
                        attach to frontmost application
  -n NAME, --attach-name NAME
                        attach to NAME
  -N IDENTIFIER, --attach-identifier IDENTIFIER
```

Fig 1: various options available in frida

```
frida-ps -U

#Basic frida hooking
frida -l disableRoot.js -f owasp.mstg.uncrackable1

#Hooking before starting the app
frida -U --no-pause -l disableRoot.js -f owasp.mstg.uncrackable1
#The --no-pause and -f options allow the app to be spawned automatically,
#frozen so that the instrumentation can occur, and the automatically
#continue execution with our modified code.
```

Fig 2: using frida from command-line

QARK:-

QARK (Quick Android Review Kit) is an easy to use tool capable of finding common security vulnerabilities in Android applications. Unlike commercial products, it is 100% free to use. QARK features educational information allowing security reviewers to locate precise, in-depth explanations of the vulnerabilities. QARK automates the use of multiple decompilers, leveraging their combined outputs, to produce superior results, when decompiling APKs. Finally, the major advantage QARK has over traditional tools, that just point you to possible vulnerabilities, is that it can produce ADB commands, or even fully functional APKs, that turn hypothetical vulnerabilities into working "POC" exploits.

Features of QARK:-

- 1) Automated Security Checks: QARK automates various security checks to identify potential vulnerabilities and security issues in Android applications
- 2) Comprehensive Vulnerability Detection: QARK scans the application's source code and resources to identify a wide range of vulnerabilities, including insecure data storage, insecure use of cryptographic functions, insecure network communication, improper use of WebView, intent vulnerabilities, exported components, and more. It covers a broad spectrum of security issues commonly found in Android applications.
- 3) Customizable Security Rules: QARK allows users to define custom security rules based on their specific requirements.
- 4) Detailed Security Reports: QARK generates detailed security reports for the analyzed Android application. The reports provide a comprehensive overview of the identified vulnerabilities, including a description of each issue, severity level, affected code snippets, and recommendations for remediation.

5) Integration with IDEs: QARK provides plugins for popular Integrated Development Environments (IDEs), such as Android Studio. These plugins allow developers to seamlessly integrate QARK into their development workflow, making it easier to perform static code analysis directly within the IDE environment.

6) Open-Source Community Support: QARK is an open-source project with an active community of developers. It benefits from community contributions, enhancements, and bug fixes, ensuring ongoing development and support.

Types of security Vulnerabilities qark can find:-

- Inadvertently exported components
- Improperly protected exported components
- Intents which are vulnerable to interception or eavesdropping
- Improper x.509 certificate validation
- Creation of world-readable or world-writeable files
- Activities which may leak data
- The use of Sticky Intents
- Insecurely created Pending Intents
- Sending of insecure Broadcast Intents
- Private keys embedded in the source
- Weak or improper cryptography use
- Potentially exploitable WebView configurations
- Exported Preference Activities
- Tapjacking
- Apps which enable backups
- Apps which are debuggable
- Apps supporting outdated API versions, with known vulnerabilities

Installation of QARK:-

1) Using python

```
pip3 install --user qark
```

2) Using github

```
git clone https://github.com/linkedin/qark.git
cd qark
pip install -r requirements.txt
pip install . --user
qark --help
```

Running Qark on Android application:-

Running qark on your Android apps is quick and easy. Simply navigate to the directory where your app is installed and use the command "python qark.py <path_to_apk_file>" to start the analysis. qark will then produce a detailed report of potential vulnerabilities and security risks within your app.

Output And Analysis:-

qark produces an HTML report that highlights each potential vulnerability and explains the steps to remediate it. This report includes a detailed analysis of the security risks present in your app and includes a scorecard indicating the level of risk your app faces.

qark analysis can be tailored to your specific needs, whether you want to focus on one specific issue or analyze all potential vulnerabilities comprehensively. In either case, you'll be left with a detailed understanding of your app's security posture.

Limitation:-

qark is a powerful tool, but it isn't without its limitations and challenges. One limitation is that it is limited to Android applications, so it cannot be used to analyze other types of software. Additionally, it may not always be able to detect very specific vulnerabilities or issues with a particular application.