1.- Instalación de entorno de análisis Android/IOS

Instalaremos el entorno de análisis Mobile Security Framework en MacOS, para ello tendremos que satisfacer unos requisitos previos antes de empezar con la instalación:

- **GIT**
- **Python 3.8-3.9**
- **JDK +8**
- **xcode-select**
- **wkhtmltopdf**

- **virtualbox como HyperVisor**
- **adb para comunicarse con el dispoitivo virtual por consola**

Nuestro sistema tenia instalado la versión python 3.11.3:

```
rajgon@RajKit.local ~
  python --version
Python 3.11.3
```

Instalamos la versión 3.9.16 con pyenv para funcionar con diferentes versiones:

- **pyenv install 3.9.16**

```
rajgon@RajKit.local ~/Mobile-Security-Framework-MobSF  <master>
  pyenv versions
  system
  2.7.18
  3.9.16
  3.10.11
* 3.11.3 (set by /Users/rajgon/.pyenv/version)
```

Descargamos el repositorio MobSF:

- **git clone** https://github.com/MobSF/Mobile-Security-Framework-MobSF.git

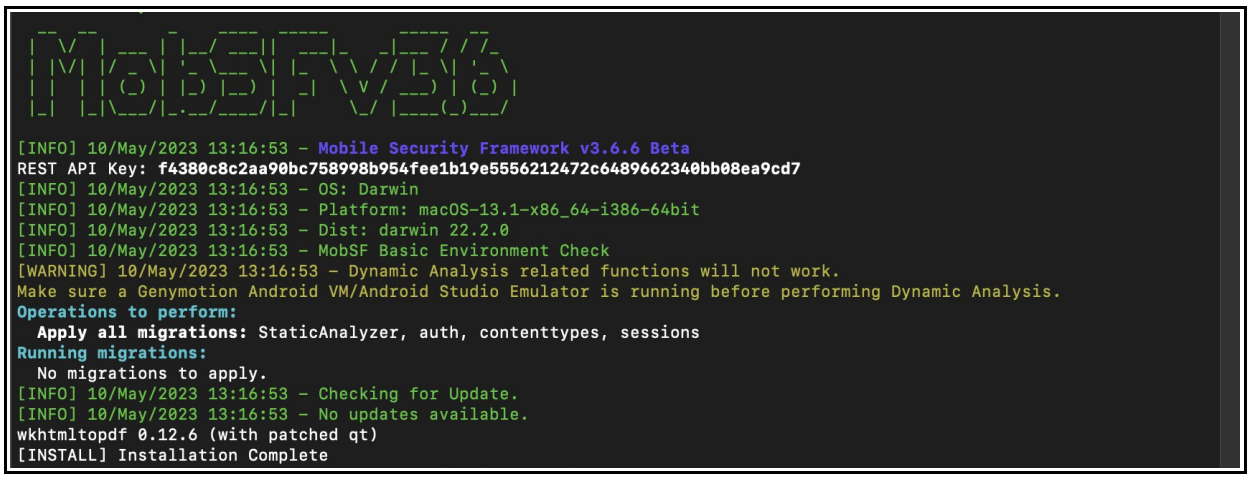
Accedemos al directorio y elegimos localmente la versión de python 3.9.16, de tal forma que en ese directorio se use esa versión de Python sin afectar al resto del equipo

- **cd Mobile-Security-Framework-MobSF**
- **pyenv local 3.9.16**

Después creamos un entorno virtual para cuidar las dependencias con virtualenv y ejecutamos el script de instalación:

- **Virtualenv mobile-reversing**
- **source mobile-reversing/bin/activate**
- **./setup.sh**

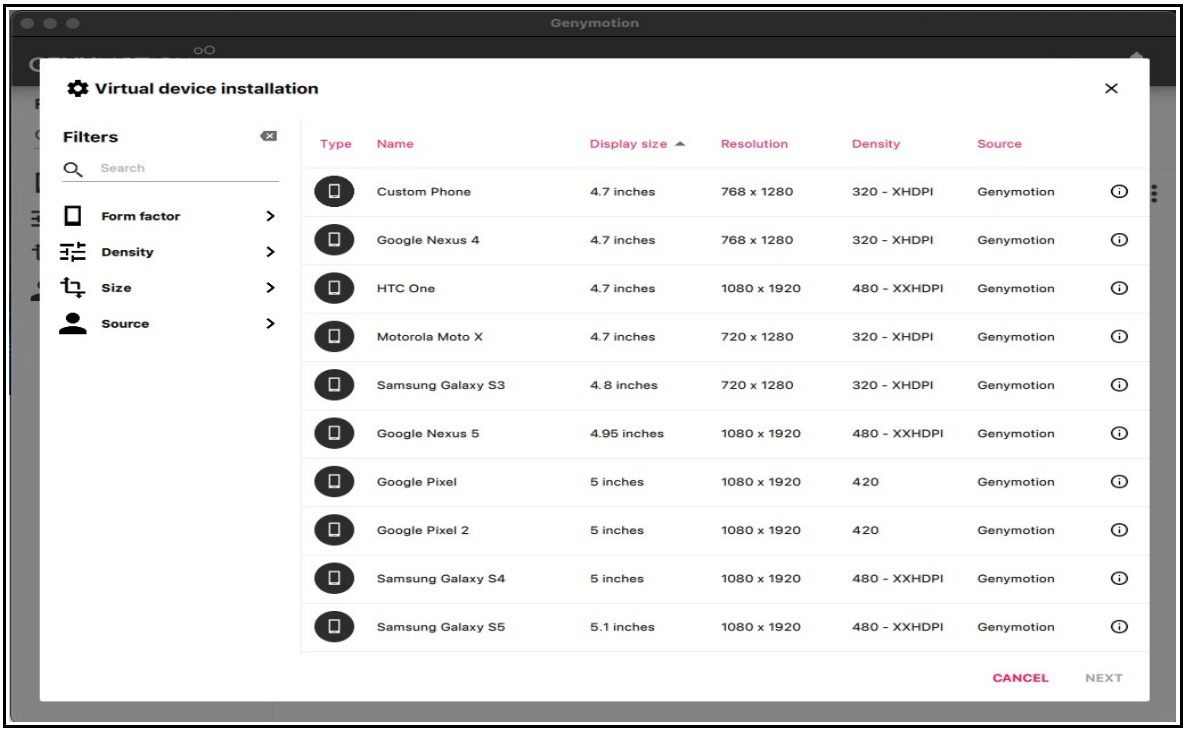
```
rajgon@RajKit.local ~/Mobile-Security-Framework-MobSF  <master*>
  virtualenv mobile-reversing                                                    127 ↵
created virtual environment CPython2.7.18.final.0-64 in 476ms
  creator CPython2macOsFramework(dest=/Users/rajgon/Mobile-Security-Framework-MobSF/mobile-reversing, clear=False, no_v
cs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, wheel=bundle, setuptools=bundle, via=copy, app_data_dir=/Users/rajgon/
Library/Application Support/virtualenv)
    added seed packages: pip==20.3.4, setuptools==44.1.1, wheel==0.37.1
  activators NushellActivator,PythonActivator,FishActivator,CShellActivator,PowerShellActivator,BashActivator
rajgon@RajKit.local ~/Mobile-Security-Framework-MobSF  <master*>
  source mobile-reversing/bin/activate
(mobile-reversing) rajgon@RajKit.local ~/Mobile-Security-Framework-MobSF  <master*>
  ./setup.sh
```



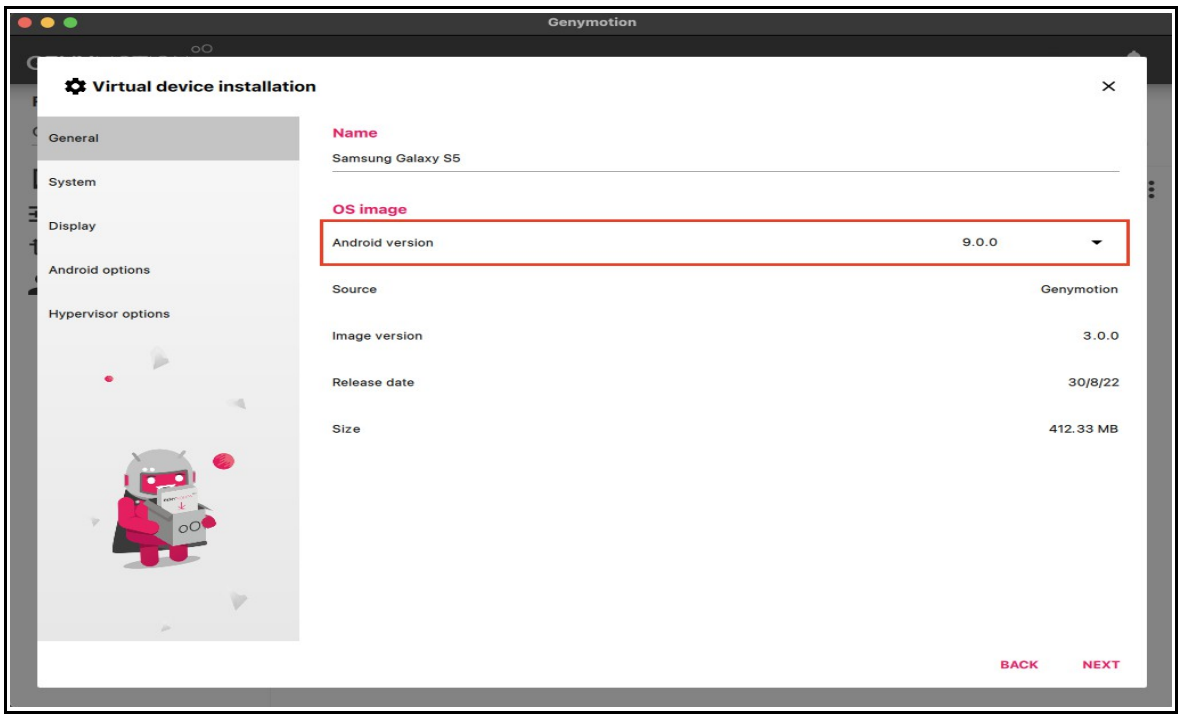
Como emulador usaremos GenyMotion que podemos descargarlo desde su pagina y utilizar previo registro:

Una vez instalado tenemos que crear y configurar un dispositivo virtual:

- Elegimos el dispositivo:

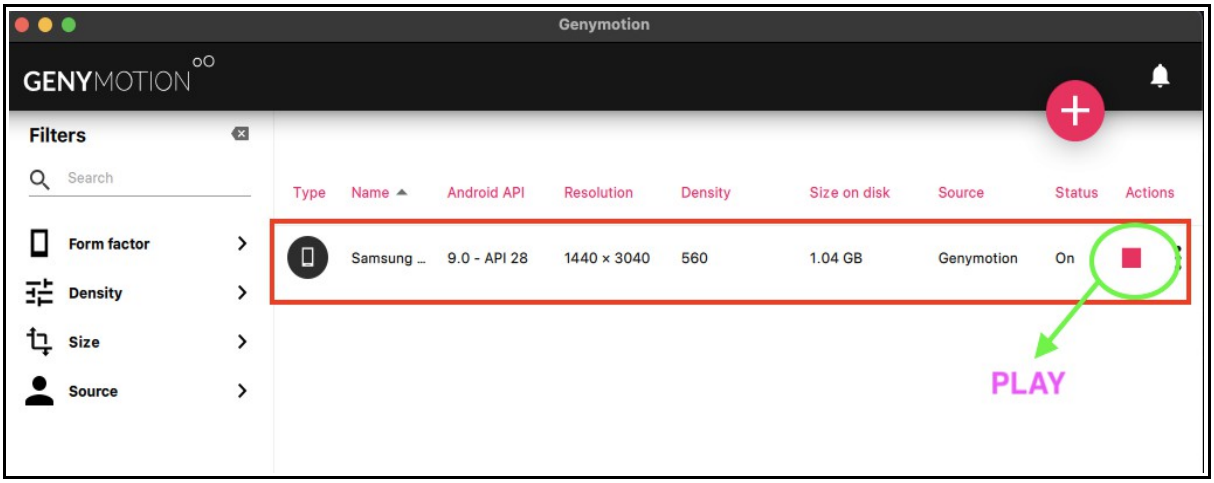


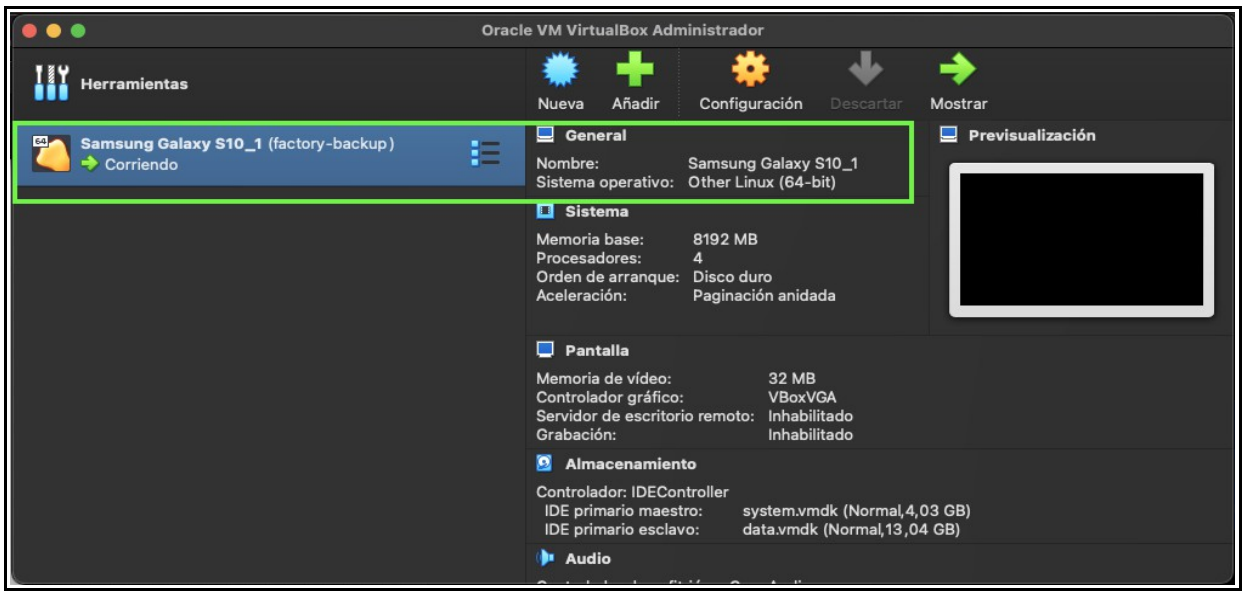
- La imagen del sistema en este caso android y para que sea apto para el análisis dinámico hasta API LEVEL 29, lo que implica una versión de Android no superior a la 10:



- Continuamos los siguientes pasos hasta que cree el dispositivo virtual y finalmente lo arrancamos.

GenyMotion utiliza VirtualBox para virtualizar el dispositivo utilizando su HyperV:



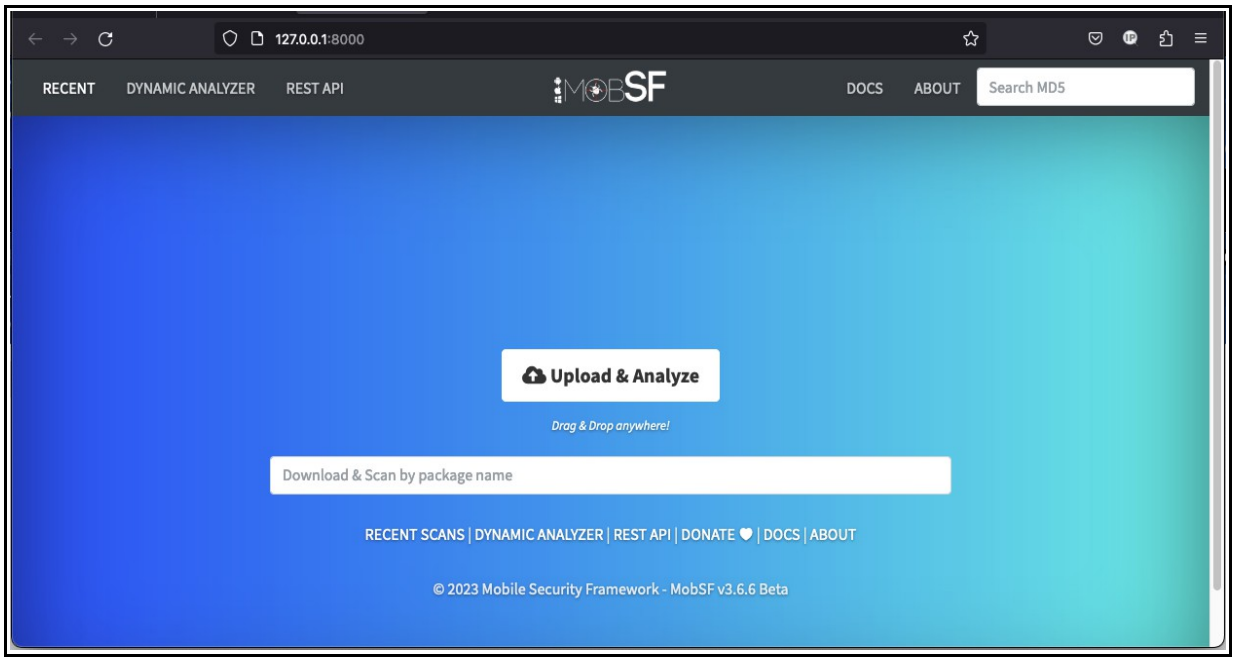


Una vez instalado MobSF y arrancado en dispositivo virtual android, volvemos al entorno virtual mobile-reversing y arrancamos MobSF que nos creara un servidor local con la interfaz gráfica web para el análisis tanto dinámico como estático:

- **/run.sh**

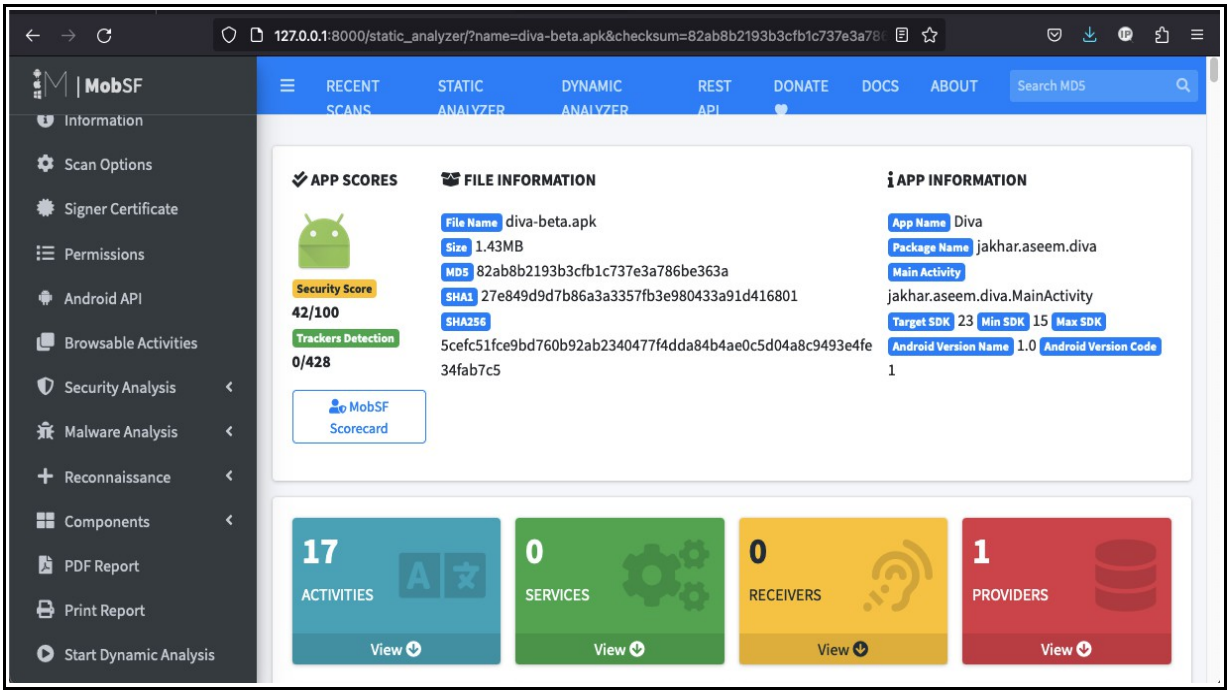
```
(mobile-reversing) ┌─rajgon@RajKit.local ~/Mobile-Security-Framework-MobSF  ‹master*›
└─➤  ./run.sh
[2023-05-10 15:22:25 +0200] [43668] [INFO] Starting gunicorn 20.1.0
[2023-05-10 15:22:25 +0200] [43668] [INFO] Listening at: http://[::]:8000 (43668)
[2023-05-10 15:22:25 +0200] [43668] [INFO] Using worker: gthread
[2023-05-10 15:22:25 +0200] [43672] [INFO] Booting worker with pid: 43672
[INFO] 10/May/2023 13:22:42 -
  __  __       _     ____  _____        _____  __
 |  \/  | ___ | |__/ ___||  ___|_   _|___ / / /_
 | |\/| |/ _ \| '_ \___ \| |_  \ \ / / |_ \| '_ \
 | |  | | (_) | |_) |__) |  _|  \ V / ___) | (_) |
 |_|  |_|\___/|_.__/____/|_|     \_/ |____(_)___/

[INFO] 10/May/2023 13:22:42 - Mobile Security Framework v3.6.6 Beta
REST API Key: f4380c8c2aa90bc758998b954fee1b19e5556212472c6489662340bb08ea9cd7
[INFO] 10/May/2023 13:22:42 - OS: Darwin
[INFO] 10/May/2023 13:22:42 - Platform: macOS-13.1-x86_64-i386-64bit
[INFO] 10/May/2023 13:22:42 - Dist: darwin 22.2.0
[INFO] 10/May/2023 13:22:42 - MobSF Basic Environment Check
[INFO] 10/May/2023 13:22:42 - Checking for Update.
[INFO] 10/May/2023 13:22:42 - No updates available.
[INFO] 10/May/2023 13:22:58 - MobSFying Android instance
[INFO] 10/May/2023 13:23:01 - ADB Restarted
[INFO] 10/May/2023 13:23:01 - Waiting for 2 seconds...
[INFO] 10/May/2023 13:23:03 - Connecting to Android 192.168.56.103:5555
[INFO] 10/May/2023 13:23:03 - Waiting for 2 seconds...
[INFO] 10/May/2023 13:23:05 - Restarting ADB Daemon as root
[INFO] 10/May/2023 13:23:05 - Waiting for 2 seconds...
[INFO] 10/May/2023 13:23:07 - Reconnecting to Android Device
[INFO] 10/May/2023 13:23:07 - Waiting for 2 seconds...
[INFO] 10/May/2023 13:23:09 - Found Genymotion x86 Android VM
[INFO] 10/May/2023 13:23:09 - Remounting
[INFO] 10/May/2023 13:23:09 - Performing System check
[INFO] 10/May/2023 13:23:09 - Android API Level identified as 28
[INFO] 10/May/2023 13:23:09 - Android Version identified as 9.0
[INFO] 10/May/2023 13:23:09 - Android OS architecture identified as x86
[INFO] 10/May/2023 13:23:10 - Downloading binary frida-server-15.2.2-android-x86
[INFO] 10/May/2023 13:23:13 - Copying frida server for x86
[INFO] 10/May/2023 13:23:14 - Installing MobSF RootCA
[INFO] 10/May/2023 13:23:14 - Installing MobSF Clipboard Dumper
[INFO] 10/May/2023 13:23:16 - MobSFying Completed!
```

El servidor por defecto arranca en el puerto 8000, accedemos a la interfaz web:



Antes de realizar un análisis, descargamos el código de al APP vulnerable DIVA y la compilamos en Android Studio.

Para realizar el análisis arrastramos la .APK al escaner y nos devolverá un análisis estático:



Si queremos realizar un análisis estático y usar FRIDA, arrancaremos Dynamic Anlysis: