Guia de uso para el sistema de análisis de tráfico de red para la detección de aplicaciones maliciosas en Android

En este documento se encuentra una guía de cómo generar tráfico a partir de los archivos que se encuentran en el repositorio.

Requerimientos:

- Android studio
- Python 2.7
- Sistema operativo Linux

Paso inicial:

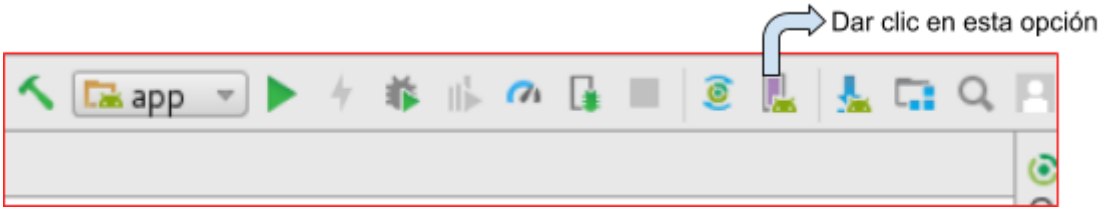
Se debe descargar la carpeta correspondiente a *Traffic- generator* del repositorio en git y pegarla en la carpeta *Android* creada por el sistema cuando se instala Android studio.

Proceso 1: Preparación de aplicaciones

1. Se crea una carpeta (si ya se tiene se omite este paso) y se ponen allí todas las aplicaciones a ser analizadas.
2. Pegar fuera de esa carpeta el archivo llamado *generacionRutas.py* y editarlo.
3. Al cambiar los campos requeridos en el archivo se debe ejecutar una terminal en linux y ejecutar el comando ***phython generacionRutas.py*** Al hacer esto se imprimirá un número correspondiente a la cantidad de aplicaciones cuyas rutas fueron generadas y se generará un txt llamado *datasetMal.txt*. Si la estructura de las rutas escritas en el archivo es correcta, se le debe cambiar el nombre al archivo generado por *datasetPrueba.txt.*
4. Se pega este archivo en la carpeta de generación de tráfico descargada de git.

Proceso 2: Preparación del emulador (si el lector ya sabe como crear un emulador en Android studio, omitir el paso 1 y 2)

1. Se inicia Android studio (para linux se debe buscar la carpeta *android-studio* y abrir una terminal en la carpeta *bin*). En la terminal se ejecuta el comando ***./studio.sh*** para que se ejecute Android studio.
2. Se crea un emulador, para esto nos situamos en la barra de herramientas de Android studio y elegimos *AVD manager.* A continuación se observa donde se encuentra dicha opción.



Importante: no confundir la carpeta *Android* con la carpeta *android-studio* ya que ambas carpetas cumplen funciones muy distintas.

3. Se debe copiar el nombre que se le asignó al emulador o al menos recordarlo debido a que será usado más adelante.
4. Se corre el emulador manualmente y se ejecuta el script *eliminar paquetes.py* el cual se encarga de eliminar todos los paquetes excepto por el paquete com.example.android.apis ya que en mi opinión algunas de las funcionalidades son importantes y pueden quizás facilitarle las cosas a el malware. Al final se imprimirá en consola el número de paquetes eliminados que deberían ser 6.

Proceso 3: Ejecución del algoritmo de generación de tráfico

1. En la carpeta Android se debe crear una carpeta llamada *Capturas*, es aquí donde se guardarán las capturas de tráfico.
2. Se debe editar el archivo instalacionApp.py y cambiar los siguientes campos:
    - Cambiar Apps por el nombre de la carpeta donde se encuentran las aplicaciones. Este campo se encuentra en la siguiente línea:

```
nombreArchivoPartido2=nombreArchivoPartido[0].split("Apps/")
```

    - Cambiar tanto la interfaz de red como la dirección ip en la siguiente línea:

```
wiresharkInicio="tshark -i em1 host 192.168.131.24 -a duration:300 -w ../../Capturas/"+paqueteAUsar+".pcap"
```

Para obtener la dirección ip del emulador se debe ejecutar una terminal en la ruta *Android/Sdk/platform-tools,* escribir el comando ***./adb shell*** y luego el comando ***ip a.***

3. Si no has cerrado el emulador entonces debes ejecutar el script *instalacionApp.py*. En caso contrario se debe editar el archivo *emulador.py*. Una vez editado, se ejecuta este archivo y la ejecución del script *instalacionApp.py* se ejecutará en otra terminal.
4. El proceso de generación de tráfico tarda aproximadamente 8 minutos por aplicación procesada. Al final se obtienen las capturas de tráfico de las aplicaciones cuya instalación fue exitosa.

Anotaciones especiales:

1. Al finalizar todo el proceso, se imprimirá en consola la cantidad de aplicaciones que no pudieron ser instaladas.
2. El log de las aplicaciones que no pudieron ser instaladas se encuentra en la ruta *Android/Sdk/platform-tools* y se llama *instalacionesFallidas.txt*. En este archivo se encuentra el nombre de la aplicación instalada y el error que se produjo.

3. Cuando la aplicación “no responde” monkey se detiene.
4. En algunos casos debido a la cantidad de procesos que se llevan a cabo es posible que el emulador se detenga.
5. Cualquier duda o comentario pueden enviar un correo a belalcazar180@gmail.com