

Práctica 6

Análisis Dinámico en Android

Parte II

Tabla de contenido

1.Objetivos	3
2. Configuración del entorno de trabajo	4
2.1 Instalación de las máquinas virtuales Kali Linux y Android	4
2.2 Configuración de la red virtual en Virtualbox	5
3. Captura de tráfico.....	9
3.1 Instalación y ejecución del servidor app.py	9
3.2 Instalación y ejecución del cliente InsecureBank	10
3.3 Captura y análisis de tráfico	11
3.3.1 Wireshark	11
3.3.2 Burp Suite	14
4 Conclusiones	24

1. Objetivos

El análisis dinámico se encarga de la verificación y evaluación de las aplicaciones en tiempo de ejecución. Su principal objetivo es identificar las vulnerabilidades o puntos débiles de la aplicación cuando está en ejecución.

Normalmente, las aplicaciones móviles en ejecución siguen el modelo cliente-servidor, donde la aplicación móvil interactúa con un servidor con el que intercambia información, siguiendo un protocolo específico. Este comportamiento hace que el análisis dinámico deba abarcar tanto la parte implementada en el dispositivo móvil, como la implementada en el servidor con el que interactúa. Para ello deberemos ser capaces de capturar el tráfico intercambiado entre la aplicación bajo análisis y el servidor con el que interactúa. Sin embargo, debemos ser conscientes que la captura y/o modificación del tráfico enviado o recibido por una aplicación/dispositivo únicamente podemos realizarla con fines analíticos y siempre sobre nuestro propio tráfico en una red de nuestra propiedad, dado que la captura de tráfico de terceros es ilegal (penado con cárcel y multa (artículo 197.1 del código penal)).

En esta práctica vamos a aprender el uso de herramientas capaces de llevar a cabo esta tarea. En concreto, vamos a analizar el tráfico generado entre la aplicación ya utilizada en la práctica anterior, InsecureBank y el servidor que le da soporte, en el entorno virtual que nos ofrece Virtualbox.

2. Configuración del entorno de trabajo

2.1 Instalación de las máquinas virtuales Kali Linux y Android

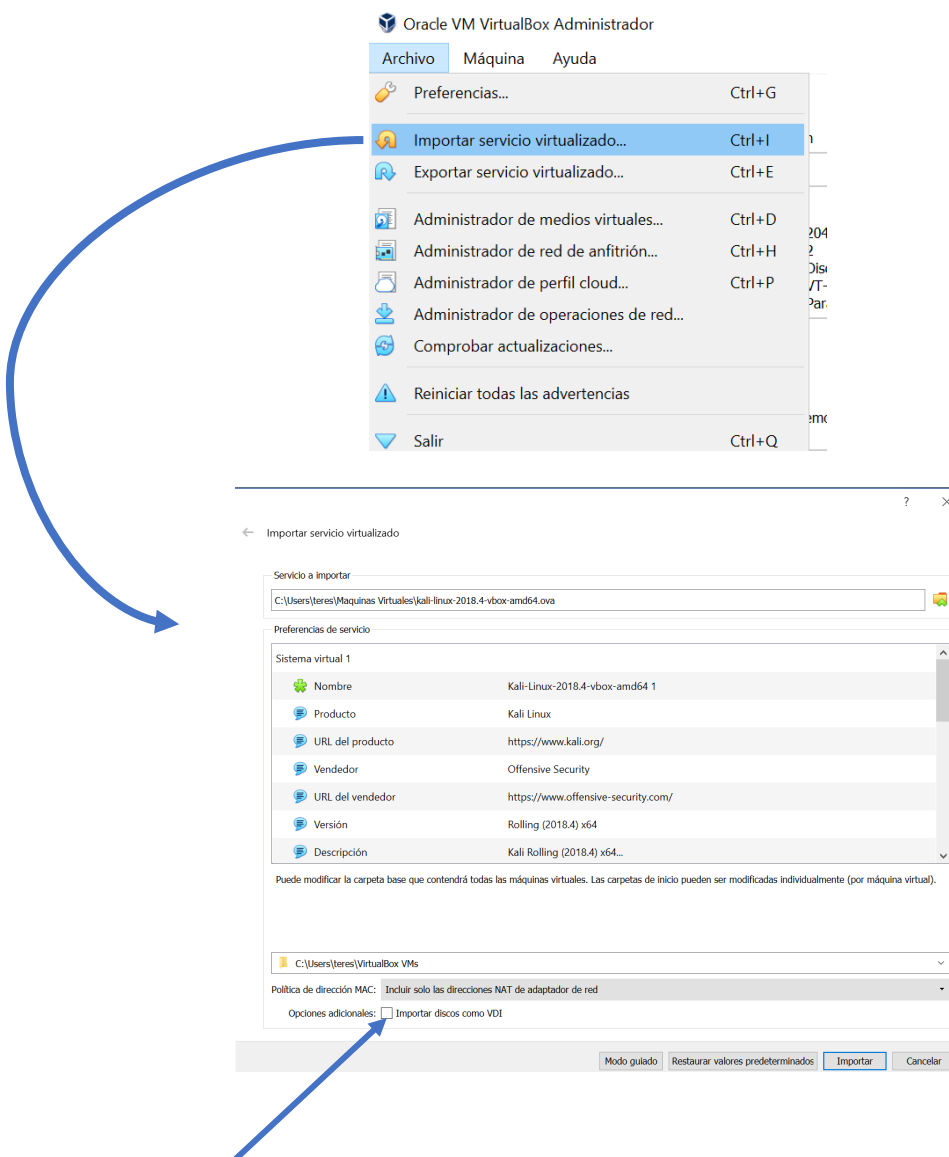
En primer lugar, debemos instalar y ejecutar Virtualbox (<https://www.virtualbox.org/wiki/Downloads>).

En Virtualbox debemos instalar las máquinas virtuales Kali Linux (kali-linux-2018.4-vbox-amd64.ova) y Android (android_7.ova). Los ficheros .OVA de ambas máquinas se encuentran en \\zuria.cc.upv.es\Asignaturas\Alumnos\gii-cdm\Maquinas Virtuales.

Para ello debemos:

1. Conectar nuestro ordenador a la VPN de la UPV. En <https://www.upv.es/contenidos/INFOACCESO/infoweb/infoacceso/dat/934950normalc.html> se describe como establecer esta conexión.
2. Conectar la unidad de red [\\zuria.cc.upv.es\disca](https://www.upv.es/disca) en nuestro ordenador y descargar ambas máquinas virtuales situadas en el directorio es\Asignaturas\Alumnos\gii-cdm\Maquinas Virtuales.

Para su instalación realizaremos la importación del fichero .OVA de cada una de las imágenes de las máquinas virtuales a instalar. Una vez instalada Kali Linux, repetiremos el mismo proceso para instalar la máquina virtual Android.



2.2 Configuración de la red virtual en Virtualbox

Para configurar la red virtual, Virtualbox¹ nos permite escoger entre los siguientes modos de conexión.

No conectado: En este modo, Oracle VM VirtualBox informa al invitado que hay una tarjeta de red, pero que no hay conexión. Esto es como si no hubiera un cable Ethernet conectado a la tarjeta.

NAT (Traducción de direcciones de red): Este modo es adecuado si se la funcionalidad que se requiere es navegar por la Web, descargar archivos y ver el correo electrónico dentro de la máquina virtual. Tiene bastantes limitaciones si tenemos que establecer conexiones con la máquina virtual.

Red NAT: este modo es un tipo de red interna que permite conexiones salientes. Este es el modo que se utilizará en esta práctica.

Adaptador Puente: Esto es para necesidades de red más avanzadas, como simulaciones de red y servidores en ejecución en un invitado. Cuando está habilitado, Oracle VM VirtualBox se conecta a una de sus tarjetas de red instaladas e intercambia paquetes de red directamente, eludiendo la pila de red de su sistema operativo host. Simula una conexión física real a la red, asignando una IP al sistema operativo huésped. Esta IP se puede obtener por DHCP o directamente configurándola en el Sistema Operativo huésped.

Red interna: Esto se puede utilizar para crear un tipo diferente de red basada en software que sea visible para las máquinas virtuales seleccionadas, pero no para las aplicaciones que se ejecutan en el host o en el mundo exterior.

Adaptador sólo- anfitrión: Es una mezcla de Adaptador puente u Red interna. Se puede usar para crear una red que contenga el host y un conjunto de máquinas virtuales, sin la necesidad de la interfaz de red física del host. Se crea una interfaz de red virtual, similar a una interfaz de bucle invertido, en el host, que proporciona conectividad entre las máquinas virtuales y el host.

Controlador genérico: Este modo se utiliza raramente, ya que comparte una misma interfaz de red genérica, al permitir al usuario seleccionar un controlador que puede incluirse con Oracle VM VirtualBox o distribuirse en un paquete de extensión.

En la práctica, vamos a definir una nueva Red NAT, a la que denominaremos NatCDM. Para definirla, seleccionaremos *Preferencias* en Virtualbox e insertaremos una nueva Red NAT como muestran las figuras siguientes.

¹ <https://www.virtualbox.org/manual/UserManual.html>

Oracle VM VirtualBox Administrador

Archivo Máquina Ayuda



Herramientas



Ubuntu
Apagada



Kali-Linux-2018.4-vbox-a...
Apagada



santoku
Apagada



android_7
Apagada



Preferencias



Importar



Exportar



Nueva



Agregar

Preferencias (Ctrl+G)

¡Bienvenido a VirtualBox!

La parte izquierda de esta ventana contiene herramientas globales y una lista de todas las máquinas virtuales y grupos de máquinas virtuales en su computadora. Puede importar, añadir y crear nuevas MVs usando los botones correspondientes de la barra de herramientas. Puede abrir un «popup» del elemento seleccionado actualmente usando el botón de elemento correspondiente. Puede presionar la tecla **F1** para obtener ayuda instantánea o visitar www.virtualbox.org para más información y las últimas noticias.



Oracle VM VirtualBox Administrador

VirtualBox - Preferencias



General



Entrada



Actualizar



Idioma



Pantalla



Red



Extensiones



Proxy

Red

Redes NAT

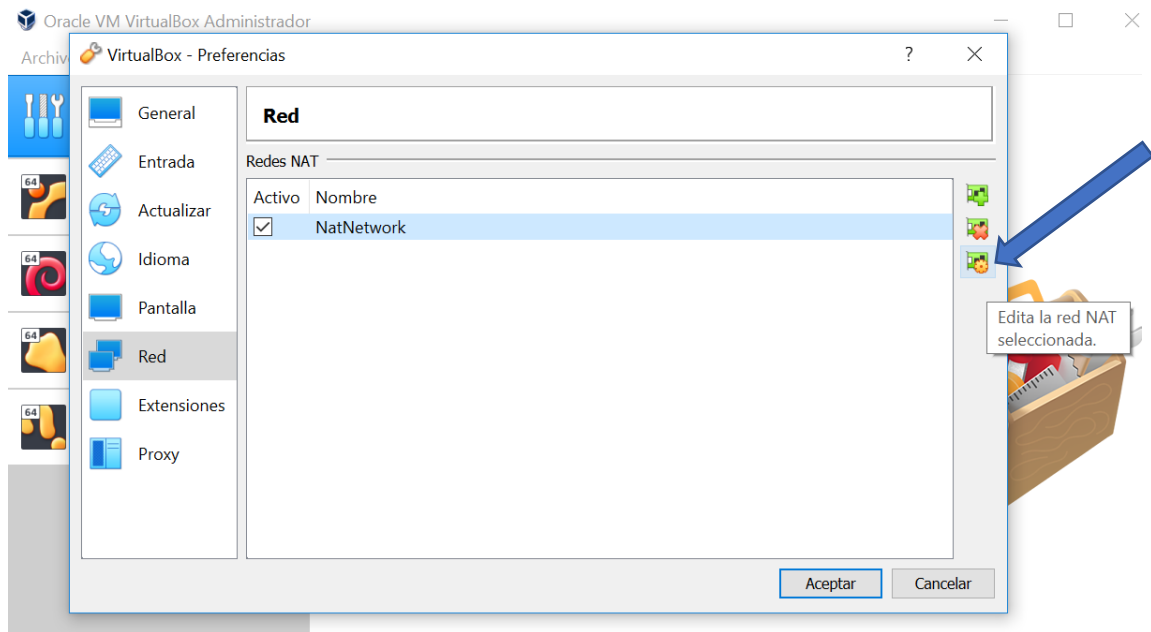
Activo Nombre



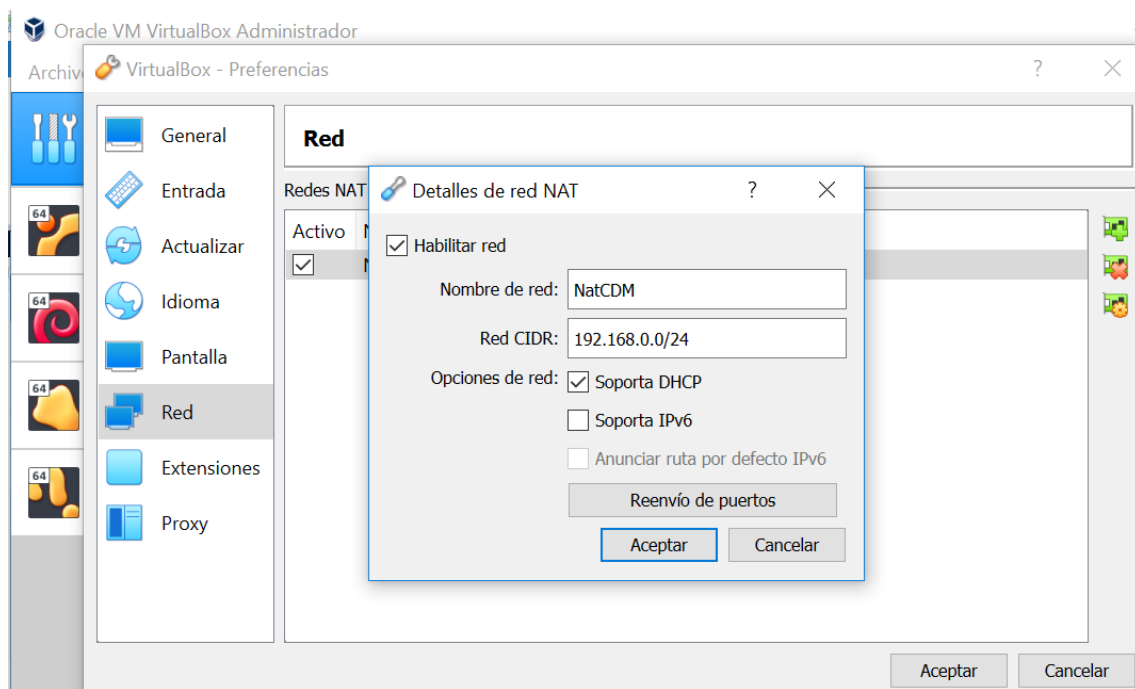
Agrega nueva red NAT.

Aceptar

Cancelar

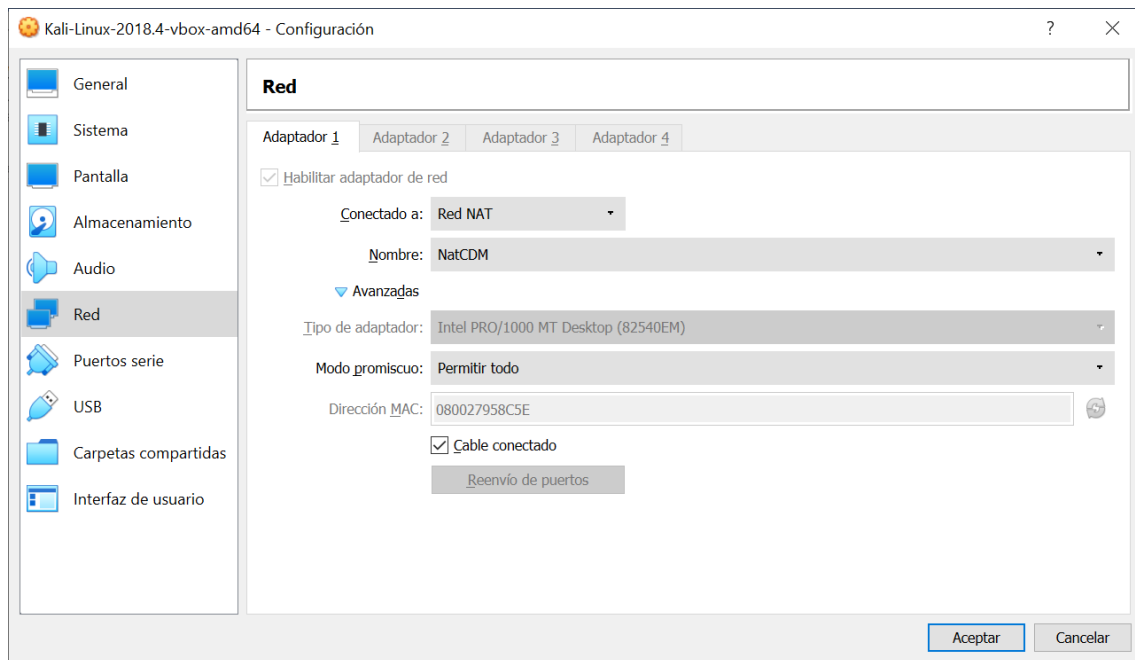


Una vez creada la red NatCDM, la configuraremos según se muestra en la siguiente figura:

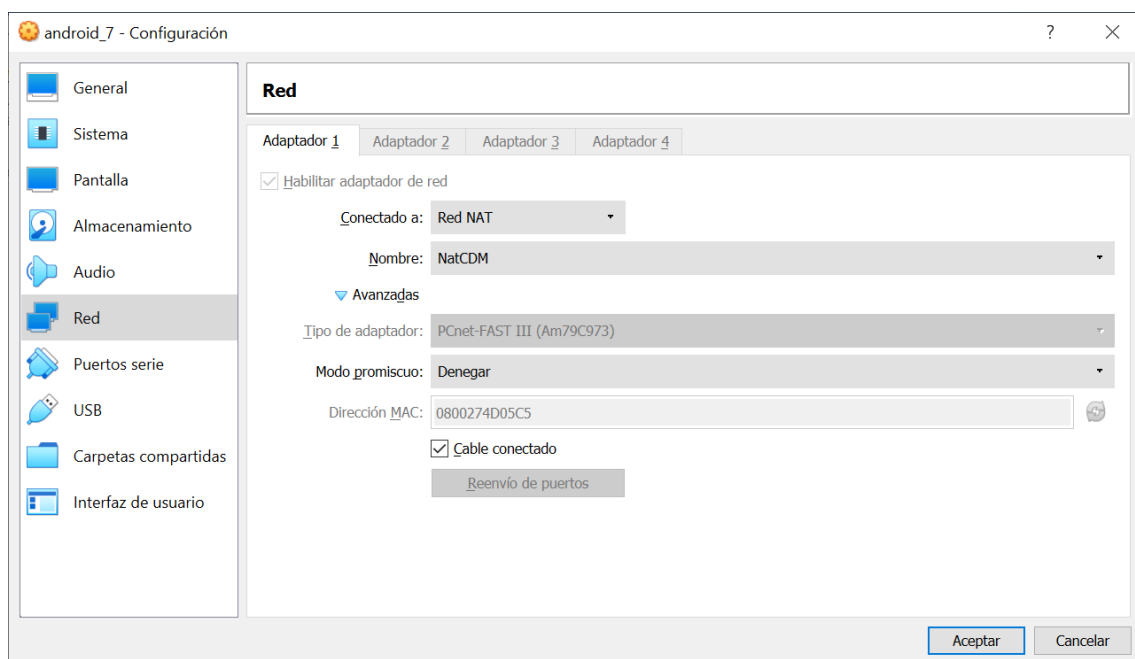


Una vez definida la red NatCDM, debemos configurar la red de cada una de las máquinas virtuales, seleccionando *Configuración*, y posteriormente *Red* en cada una de ellas del siguiente modo:

Kali Linux



Android



3. Captura de tráfico

Para la ejecución de la aplicación InsecureBank se debe de instalar la parte servidora que simula un servidor web, el servidor `app.py`, en Kali Linux y la app cliente InsecureBank en Android. Comenzaremos descargando en Kali el paquete Android-InsecureBankv2-master.zip desde:

<https://github.com/dineshshetty/Android-InsecureBankv2>

En Kali nos validaremos con el usuario: user – contraseña: user

```
root@kali:~/Downloads# unzip Android-InsecureBankv2-master.zip
```

3.1 Instalación y ejecución del servidor `app.py`

En el directorio `/Android-InsecureBankv2-master/AndroLabServer` se encuentra la aplicación servidora `app.py`. Para poder compilarlo con `python3` sin errores deberíamos realizar algunas modificaciones en la sintaxis de `app.py`. Por simplicidad, **sustituiremos este fichero `app.py` por la versión adecuada para `python3` que podemos descargar desde la carpeta de la práctica en PoliformaT.**

A continuación, en este mismo crearemos un script que instalará Python y todas las dependencias que necesita el servidor `app.py`.

- Para crear dicho script debemos **crear un fichero de texto al que llamaremos `EjecutarServidor.sh`** (cualquier otro nombre también sería valido) e incluiremos en él las siguientes líneas:

```
#!/bin/bash

sudo apt-get update

sudo apt-get install python-setuptools

sudo apt install python3-pip

sudo pip3 install flask flask-sqlalchemy simplejson cherrypy

sudo pip3 install web.py

python3 app.py
```

- Después de guardar el fichero debemos darle permisos de ejecución:

```
root@kali:~/Downloads/Android-InsecureBankv2-master/AndroLabServer#chmod +x EjecutarServidor.sh
```

- Finalmente lo ejecutaremos con la orden:

```
root@kali:~/Downloads/Android-InsecureBankv2-master/AndroLabServer#./EjecutaServidor.sh
```

Al finalizar la ejecución del script veremos que el servidor está en ejecución, esperando las peticiones del cliente en el puerto 8888.

3.2 Instalación y ejecución del cliente InsecureBank

En el directorio

```
root@kali:~/Desktop/InsecureBankv2/Android-InsecureBankv2-master#
```

tenemos la aplicación InsecureBankv2.apk que copiaremos en la máquina virtual Android del siguiente modo.

- En primer lugar, en la máquina virtual Android:
 - Entraremos en modo consola tecleando: ALT+F1
 - Abriremos un puerto en modo escucha que nos permitirá conectarnos a Android desde Kali para transferir el fichero InsecureBankv2.apk.
 - Ejecutando los comandos:

```
X86_64:/#su
X86_64:/#setprop service.adb.tcp.port 5555
X86_64:/#stop adbd
X86_64:/#start adbd
```
 - Además, para averiguar la dirección IP asignada a la máquina Android ejecutaremos:

```
X86_64:/# ip a
```

En la respuesta obtendremos las terminales lo y eth0. La IP que necesitamos es la asignada a la interfaz eth0. Para los ejemplos posteriores de configuración asumiremos que la interfaz eth0 tiene asignada la IP 192.168.0.6

- Salir del modo consola y volver a Android: ALT+F7
- A continuación, en la máquina virtual Kali:
 - Instalaremos la aplicación adb ejecutando en un terminal:

```
root@kali:~/$ sudo apt-get install adb
```

- Nos situaremos en el directorio `Android-InsecureBankv2-master`, donde se encuentra el fichero `InsecureBankv2.apk`
- Instalaremos la aplicación `InsecureBankv2.apk` desde Kali en nuestra máquina virtual Android mediante `adb`, ejecutando en un terminal:

```
root@kali:~/ $ adb connect 192.168.0.6:5555
```

```
(user@kali)-[~/Descargas/Android-InsecureBankv2-master]  
$ adb connect 192.168.0.6:5555  
connected to 192.168.0.6:5555
```

```
root@kali:~/ $ adb -s 192.168.0.6:5555 install InsecureBankv2.apk
```

```
(user@kali)-[~/Descargas/Android-InsecureBankv2-master]  
$ adb -s 192.168.0.6:5555 install InsecureBankv2.apk  
Performing Streamed Install  
Success
```

3.3 Captura y análisis de tráfico

La captura y análisis del tráfico puede realizarse de varias formas, dependiendo de la arquitectura utilizada para la ejecución de la aplicación cliente-servidor.

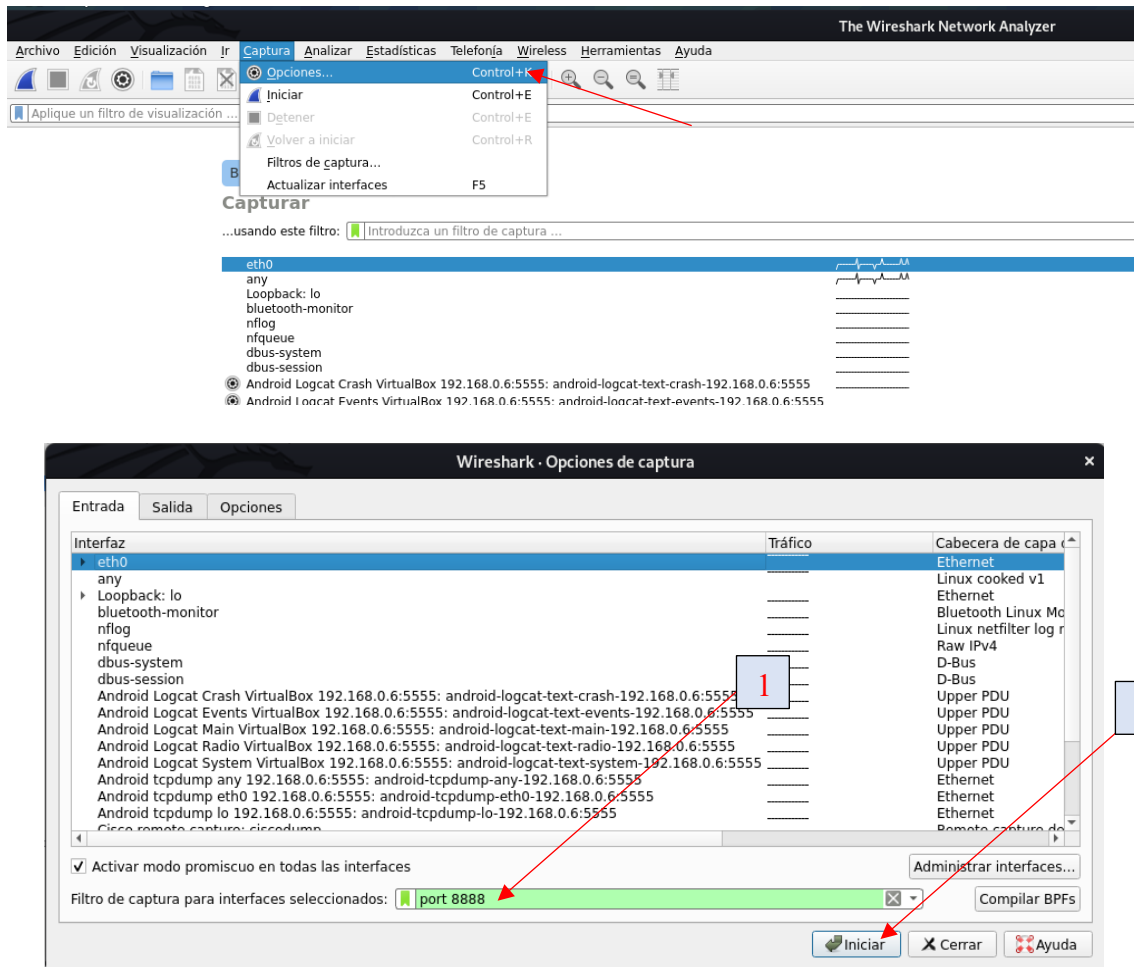
Si tenemos acceso a la máquina donde se está ejecutando el servidor, podemos ejecutar en esa misma máquina cualquier programa analizador de tráfico (también denominados sniffers), como puede ser Wireshark. Sin embargo, si no disponemos de tal acceso, podemos incluir en la arquitectura un proxy-analizador de tráfico, como Burpsuite, entre el cjackjacliente y el servidor, forzando de esa forma a que todo el tráfico intercambiado entre el cliente y el servidor pase por dicho proxy permitiéndonos su captura y posterior análisis.

En esta práctica, aunque nos encontramos en la primera situación, al estar ejecutándose el servidor en Kali. A modo demostrativo vamos a implementar ambas soluciones.

3.3.1 Wireshark

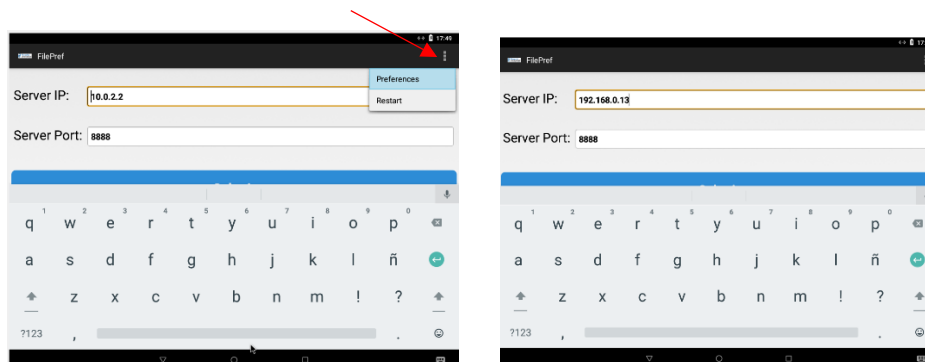
Para realizar la captura de tráfico entre la app `InsecureBank` en Android y su servidor en Kali ejecutaremos Wireshark en esta última. Podemos encontrarlo en la categoría de *Aplicaciones -> Sniffing & Spoofing*.

El filtro que debemos indicar para capturar únicamente el tráfico dirigido al servidor `app.py` es el puerto `8888`.

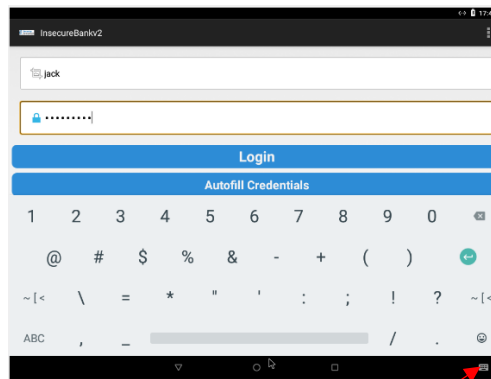


Una vez tenemos Wireshark con la captura iniciada, debemos **dirigirnos a Android** y ejecutar InsecureBank.apk.

- Debemos indicar en las *Preferences* la dirección IP de Kali (192.168.0.13):

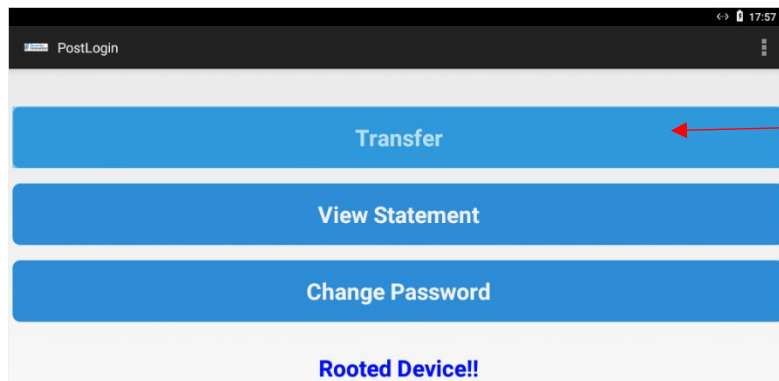


- Realizaremos el login del usuario utilizando como usuario: jack y contraseña: Jack@123\$

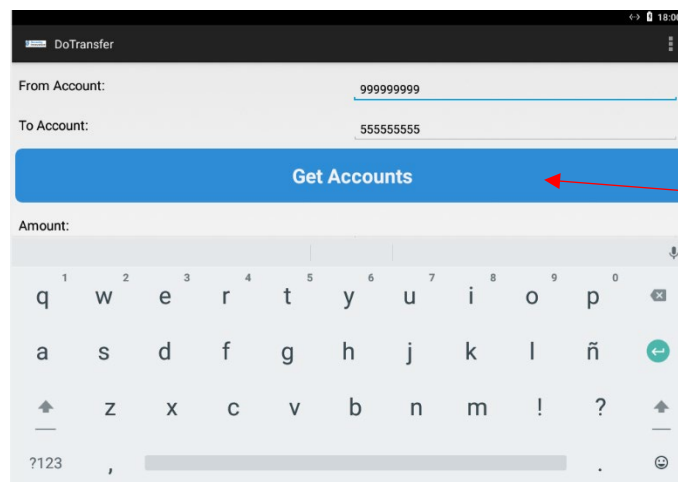


Utiliza el teclado de la máquina Android para los caracteres especiales (@\$)

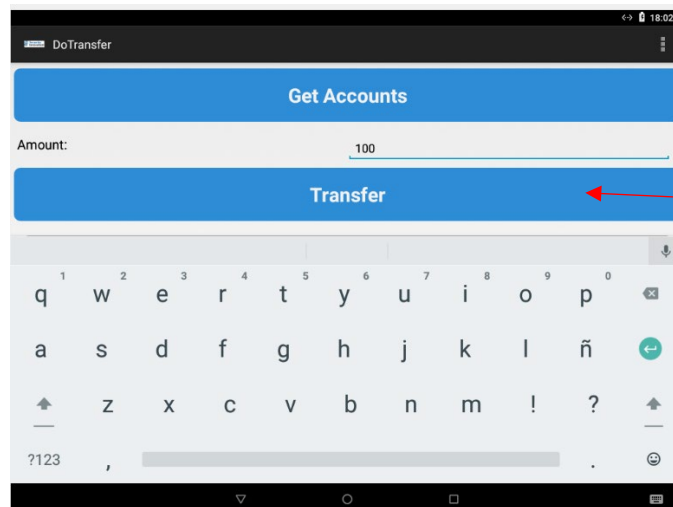
- Una vez hemos accedido a la aplicación, realizaremos una transferencia.



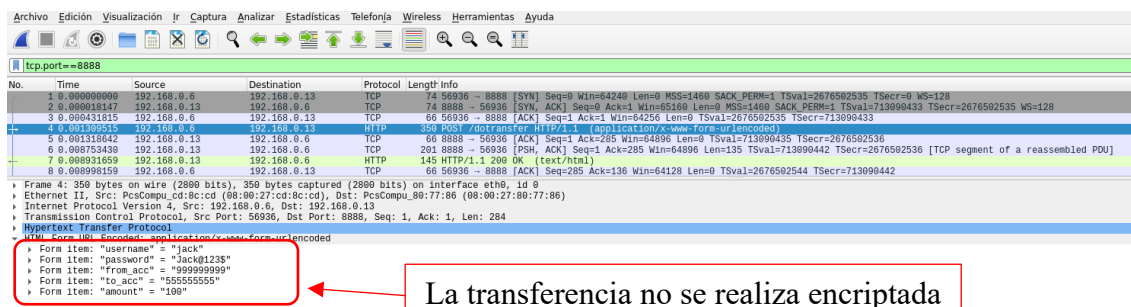
- Como desconocemos las cuentas para realizar la transferencia, seleccionaremos el botón “Get Accounts”.



- Posteriormente indicaremos la cantidad a transferir, por ejemplo 100, y pulsaremos “Transfer”.



En Kali, Wireshark habrá capturado el tráfico intercambiado entre el cliente y el servidor. Para poderlo analizar con mayor claridad, podemos aplicar un filtro de visualización escribiendo *http* en el filtro de visualización de captura, como muestra la siguiente figura.



Al analizar la captura realizada con Wireshark, vemos que todo el tráfico intercambiado entre el cliente y el servidor se ha realizado mediante peticiones y respuestas HTTP sin encriptar, permitiendo que todos los detalles (usuario, contraseña, números de cuenta y cantidad transferida) de las transacciones realizadas sean accesibles en la captura realizada. Esto confirma lo obtenido en el análisis estático.

3.3.2 Burp Suite

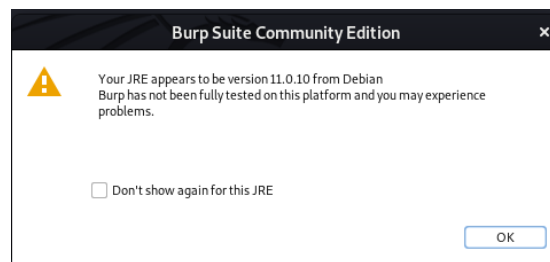
Burp Suite² es una plataforma integrada para realizar pruebas de seguridad de aplicaciones web. Sus diversas herramientas funcionan a la perfección para soportar todo el proceso de prueba, desde el mapeo inicial y el análisis de la superficie de ataque de una aplicación hasta la búsqueda y explotación de vulnerabilidades de seguridad. Burp Suite proporciona un gran control, permitiendo combinar técnicas manuales avanzadas con

² <https://tools.kali.org/web-applications/burpsuite>

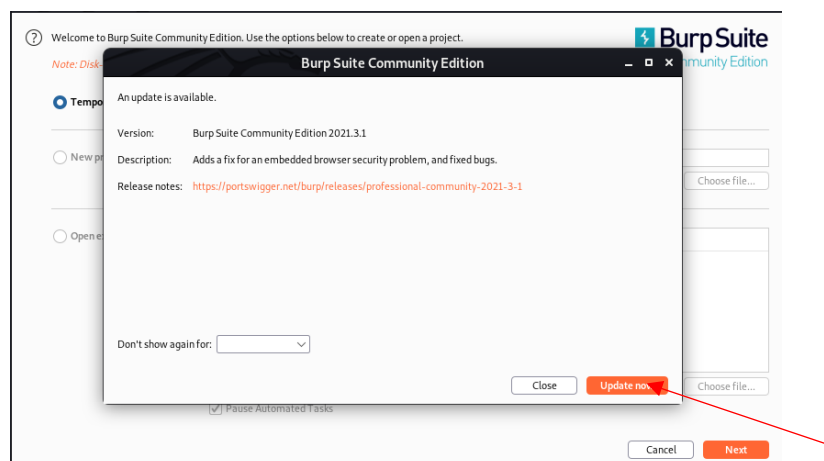
automatización de última generación, para que el trabajo sea más rápido, más efectivo y más divertido.

En esta práctica se va a utilizar Burp Suite para interceptar el tráfico intercambiado entre el cliente y el servidor. Para ello ejecutaremos Burp Suite que se encuentra en la categoría de *Aplicaciones-> Analisis de Aplicaciones Web*.

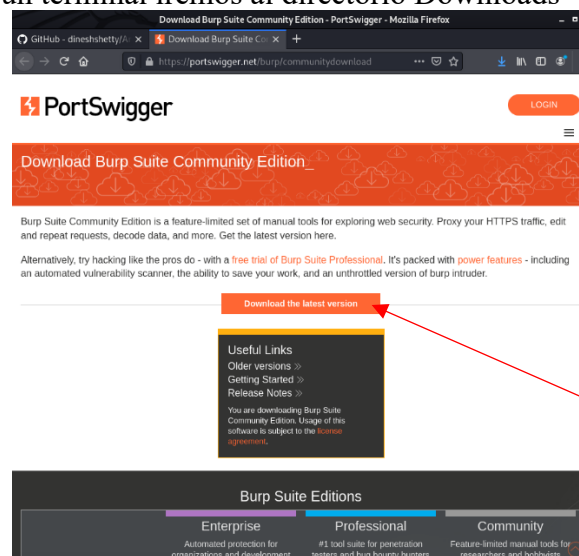
Al ejecutar la versión incluida en Kali, nos indica que puede que algunas opciones no funcionen correctamente debido al jre instalado, además nos indica que existe una nueva versión disponible.



Aceptaremos y iniciaremos Burp Suite, y seleccionaremos realizar la actualización, pinchando en el botón de “update now”.



Esto provocará que Firefox abra la pagina Web desde la que descargar la última versión. Tras descargarla, en un terminal iremos al directorio Downloads



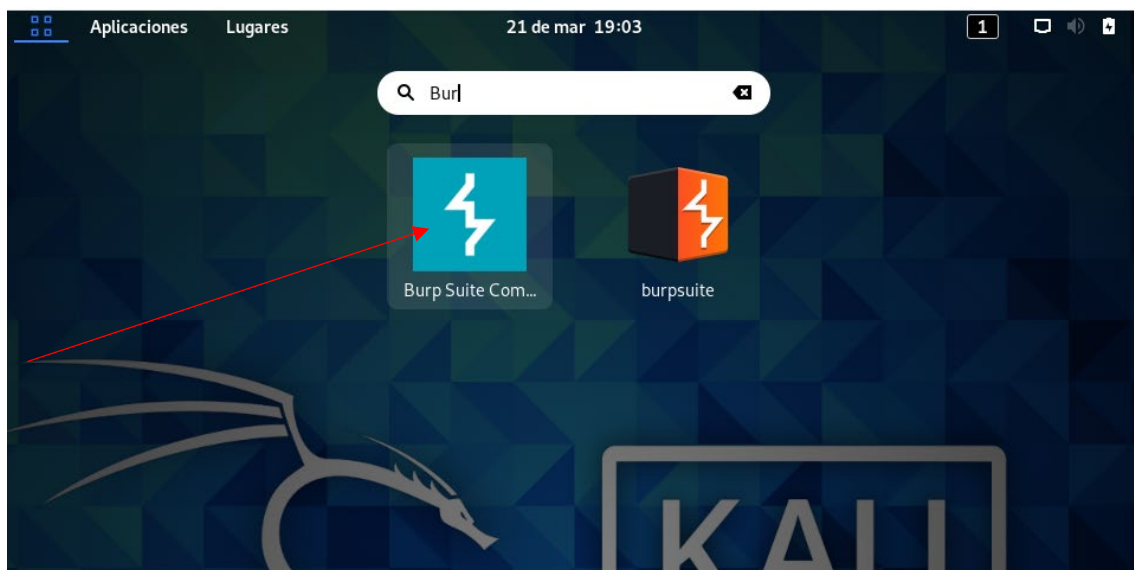
y daremos permisos de ejecución al fichero burpsuite_community_linux_v2021_3_1.sh

```
root@kali:~/Downloads# chmod +x burpsuite_community_linux_v2021_3_1.sh
```

y posteriormente la ejecutaremos:

```
root@kali:~/Downloads# ./burpsuite_community_linux_v2021_3_1.sh
```

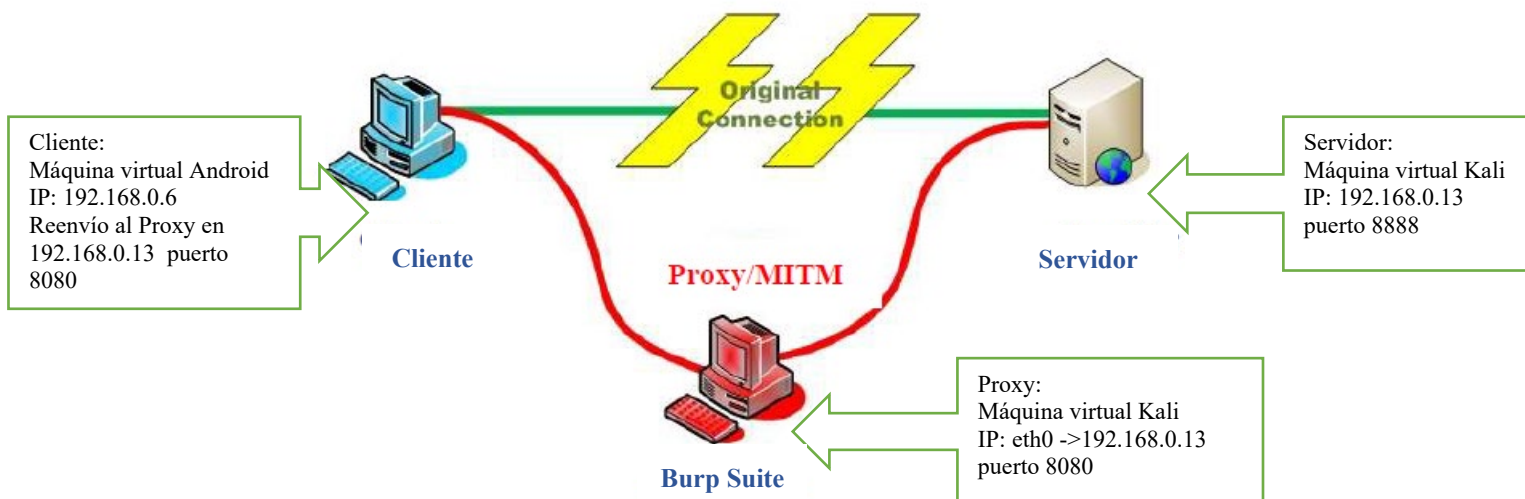
A continuación, ejecutaremos la nueva versión denominada Burp Suite Community Edition.



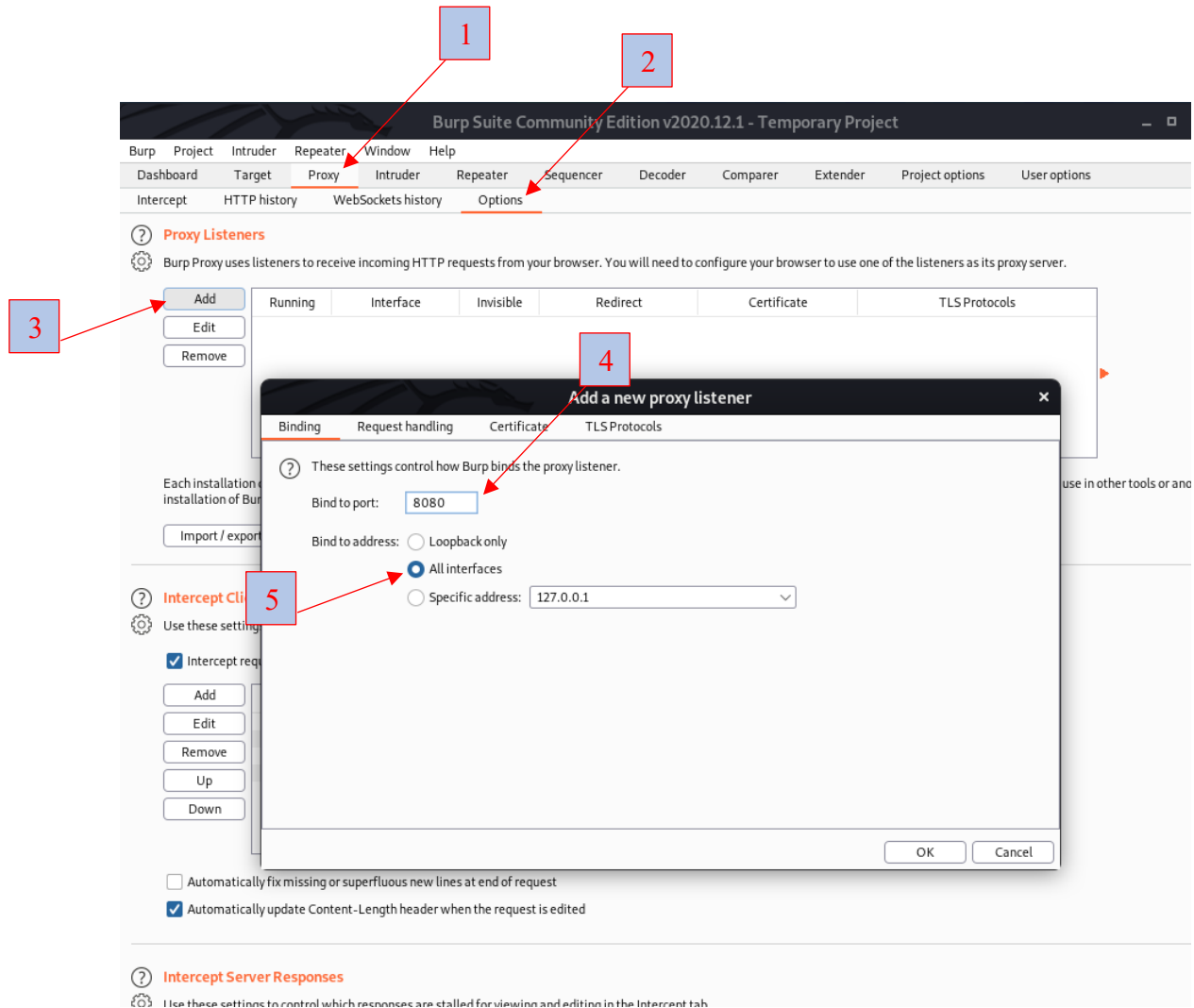
Dejando las opciones por defecto que nos propone en su ejecución: Temporary Project y Use Burp defaults.

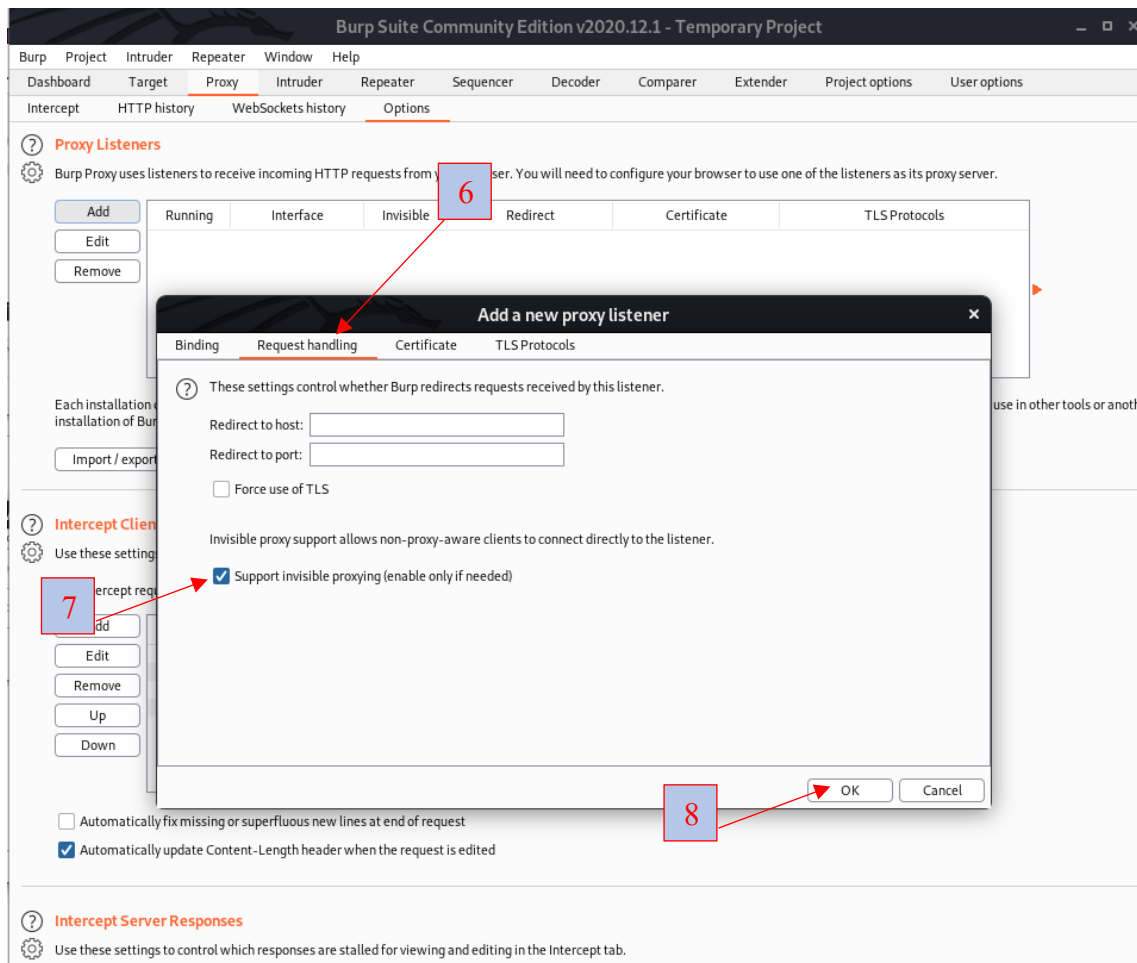
3.3.2.1 Configuración del Proxy

Queremos configurar el Proxy que proporciona Burp Suite para conseguir el esquema de interconexión que muestra la siguiente figura.



En Seleccionaremos la pestaña *Options* del Proxy para indicar que queremos capturar todo el tráfico que se reciba dirigido al puerto 8080. Para ello en el apartado de Proxy Listeners en primer lugar seleccionaremos la entrada existente de la interfaz 127.0.0.1:8080 y la eliminaremos pinchando “Remove”. A continuación realizaremos la configuración que se muestra en las siguientes imágenes.





Además, en el apartado de Intercept Request añadiremos la regla que indique que queremos interceptar todas las peticiones que vengan de la IP de la máquina Android (192.168.0.6)

Burp Suite Community Edition v2020.12.1 - Temporary Project

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extender Project options

Intercept HTTP history WebSockets history **Options**

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners.

	Running	Interface	Invisible	Redirect	Certificate	TLS F
<div><div>Add</div><div>Edit</div><div>Remove</div></div>	<input checked="" type="checkbox"/>	*:8080	<input checked="" type="checkbox"/>		Per-host	Default

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export the installation of Burp.

Import / export CA certificate

Regenerate CA certificate

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☒ Intercept requests based on the following rules:

9

Add

Edit

Remove

Up

Down

Automatically

☒ Automatically

Enabled	Operator	Match type	Relations	Condition
<input checked="" type="checkbox"/>	Or	Domain name	Matches	192.168.0.6

10

11

12

Specify the details of the interception rule.

Boolean operator:

Or

Match type:

Domain name

Match relationship:

Matches

Match condition:

192.168.0.6

OK

Cancel

De este modo la configuración de nuestro Proxy quedará del siguiente modo:

Burp Suite Community Edition v2020.12.1 - Temporary Project

Dashboard Target **Proxy** Intruder Repeater Sequencer Decoder Comparer Extender Project options User options

Intercept HTTP history WebSockets history **Options**

Proxy Listeners

Burp Proxy uses listeners to receive incoming HTTP requests from your browser. You will need to configure your browser to use one of the listeners as its proxy server.

Running	Interface	Invisible	Redirect	Certificate	TLS Protocols
<input checked="" type="checkbox"/>	*:8080	<input checked="" type="checkbox"/>		Per-host	Default

Each installation of Burp generates its own CA certificate that Proxy listeners can use when negotiating TLS connections. You can import or export this certificate for use in other installations of Burp.

Intercept Client Requests

Use these settings to control which requests are stalled for viewing and editing in the Intercept tab.

☒ Intercept requests based on the following rules:

Enabled	Operator	Match type	Relationship	Condition
<input checked="" type="checkbox"/>	Or	File extension	Does not match	(^gif\$ ^jpg\$ ^png\$ ^css\$ ^js\$ ^ico\$...
<input type="checkbox"/>		Request	Contains parameters	
<input type="checkbox"/>	Or	HTTP method	Does not match	(get post)
<input type="checkbox"/>	And	URL	Is in target scope	
<input checked="" type="checkbox"/>	Or	Domain name	Matches	192.168.0.6

☐ Automatically fix missing or superfluous new lines at end of request

☒ Automatically update Content-Length header when the request is edited

Ahora necesitamos configurar la máquina Android para que redirija su tráfico al Proxy que hemos definido en Kali. Aunque podríamos realizar la configuración del Proxy de varios modos, algunos tan sencillos como indicar la IP del Proxy en los ajustes de Android, resulta más interesante realizarlo utilizando las reglas de iptables.

iptables gestiona, mantiene e inspecciona las reglas de filtrado de paquetes IPv4 a través de tablas. Estas tablas clasifican y organizan las reglas de acuerdo al tipo de decisiones que se deben tomar sobre los paquetes. Por ejemplo, si una regla se encarga de implementar traducción de direcciones, como es nuestro caso, será puesta en la tabla "nat". En cambio, si una regla decide cuándo o no dejar pasar un paquete hacia su destino, probablemente será agregada en la tabla "filter".

Para configurar el Proxy **en Android** mediante iptables procederemos del siguiente modo:

- Entraremos en modo consola tecleando: ALT+F1
- Insertaremos una regla en la tabla nat de iptables donde indicaremos que todo el tráfico de salida del protocolo tcp debe de reenviarse a la dirección IP

192.168.0.13 (IP de Kali), y puerto 8080 (puerto donde está escuchando el Proxy en Kali).

```
X86_64:/#iptables -t nat -A OUTPUT -p tcp -j DNAT --to-destination  
192.168.0.13:8080
```

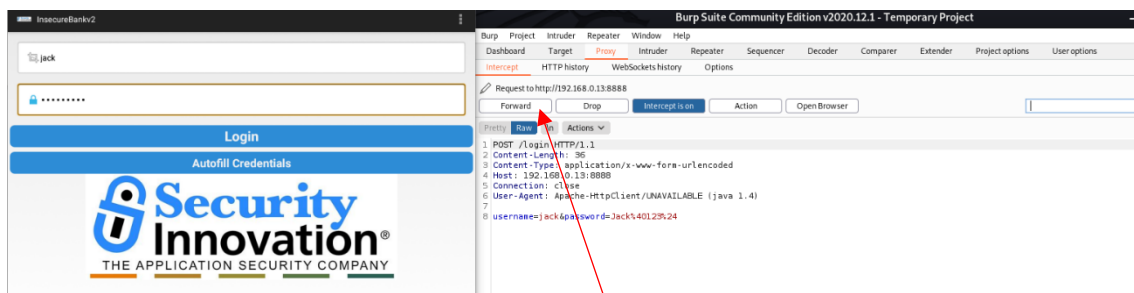
- Para ver las reglas en la tabla nat ejecutaremos:

```
X86_64:/#iptables -t nat -L
```

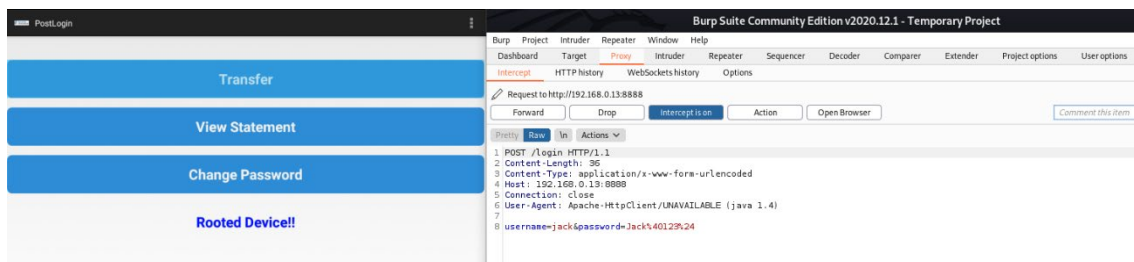
- Saldremos de modo consola tecleando: ALT+F7

Ahora ya estamos en situación de interceptar el tráfico entre la parte cliente y la servidora de la aplicación InsecureBankv2, por lo que ejecutaremos de nuevo la aplicación InsecureBankv2.apk en la máquina Android, como hicimos en la sección 3.3.1.

En Kali podremos ver que la petición de login ha sido interceptada:



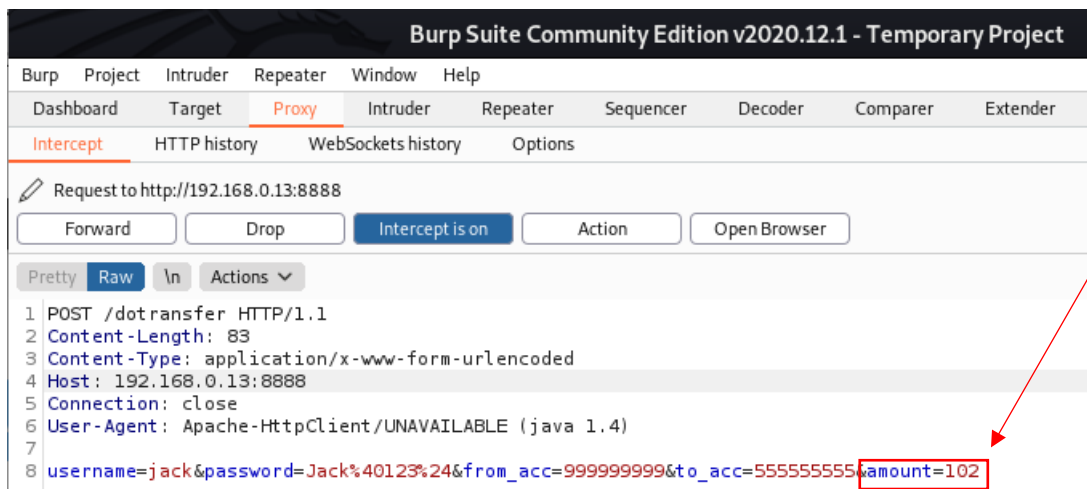
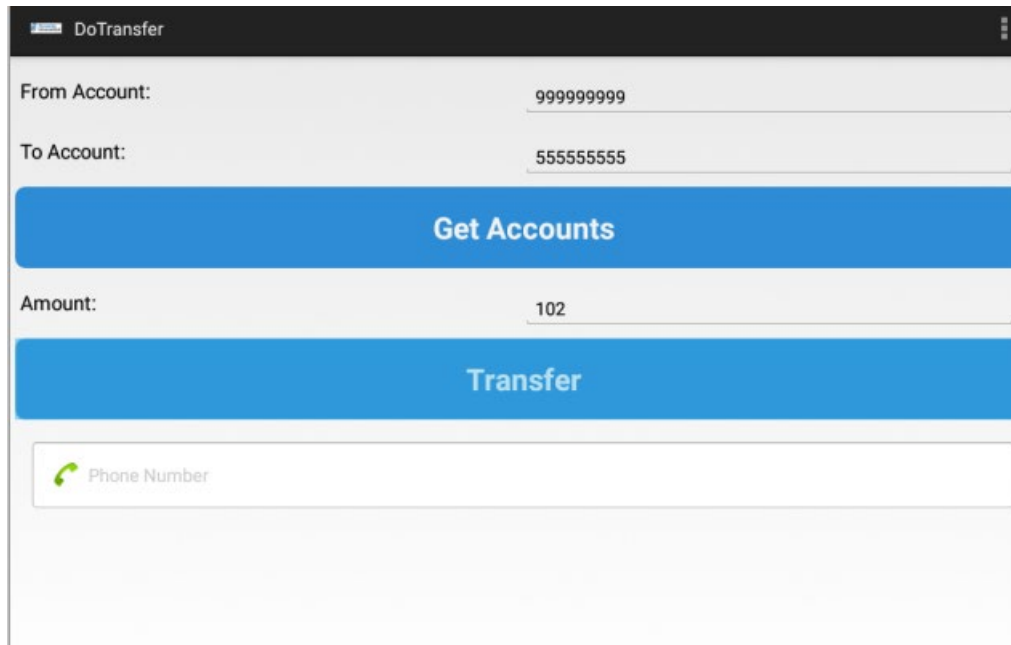
Únicamente cuando pulsemos el botón “Forward” de Burp Suite será transmitida al servidor. Pulsémoslo y veamos como en ese momento el login se completa en la aplicación de Android. Además, podemos observar los parámetros de la petición, al igual que anteriormente en Wireshark.



Sin embargo, hay que destacar que Burp Suite, no está capturando el tráfico para mostrárnoslo, sino que lo ha interceptado, de forma que nos permitiría cambiar el valor de los parámetros que está enviando la aplicación antes de que estos lleguen al servidor. Para ver como afecta estas modificaciones de los parámetros en Burp Suite es interesante realizar también la captura del tráfico con Wireshark.

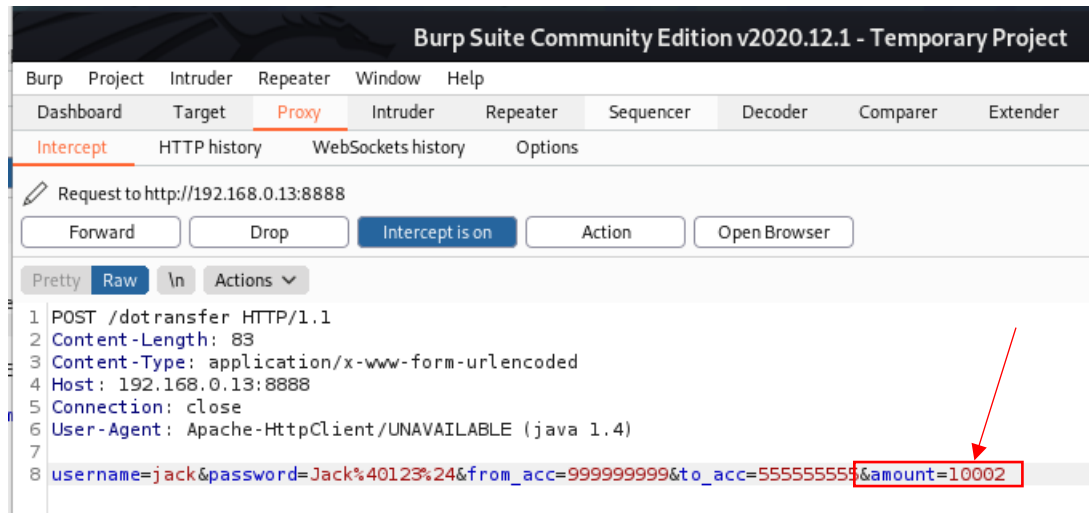
Ejecutaremos Wireshark indicando el puerto de filtrado 8080.

- **En Android**, con InsecureBankv2.apk realizaremos una transferencia. Como desconocemos las cuentas para realizar la transferencia, seleccionaremos el botón “*Get Accounts*”. Posteriormente indicaremos la cantidad a transferir, por ejemplo 102, y pulsaremos “*Transfer*”.
- **En Kali**, con Burp Suite al interceptar el tráfico referente a la transferencia:

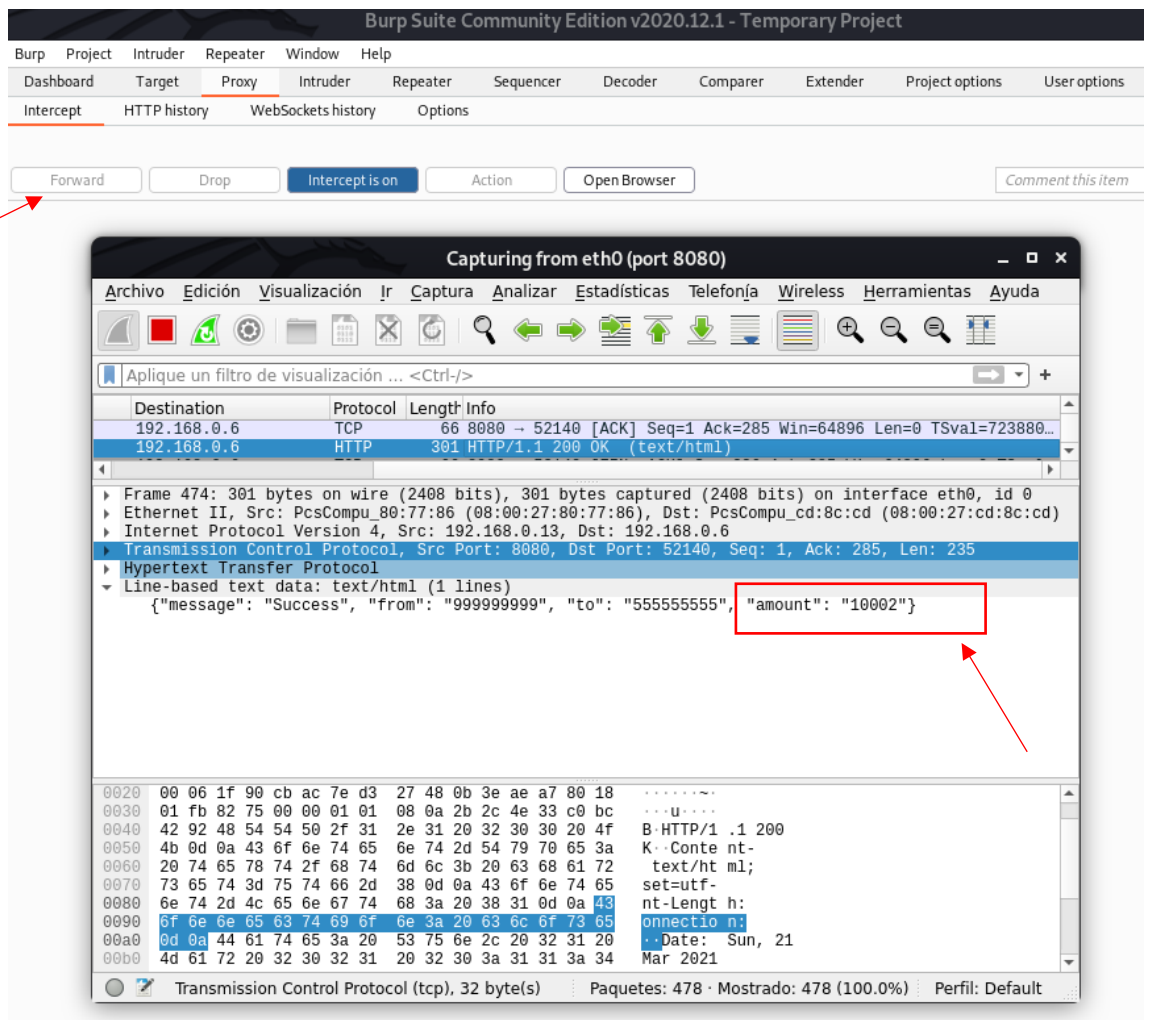


```
1 POST /dotransfer HTTP/1.1
2 Content-Length: 83
3 Content-Type: application/x-www-form-urlencoded
4 Host: 192.168.0.13:8888
5 Connection: close
6 User-Agent: Apache-HttpClient/UNAVAILABLE (java 1.4)
7
8 username=jack&password=Jack%40123%24&from_acc=999999999&to_acc=555555555&amount=102
```

- Modificamos la cantidad a transferir:

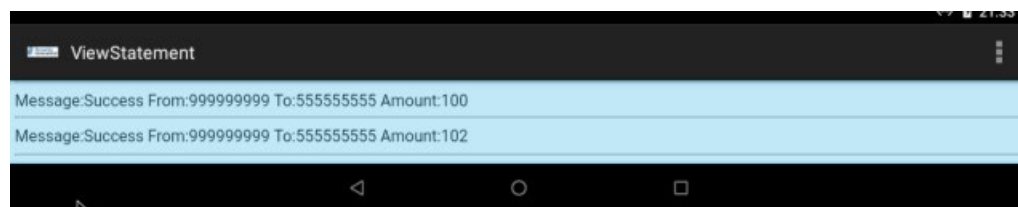
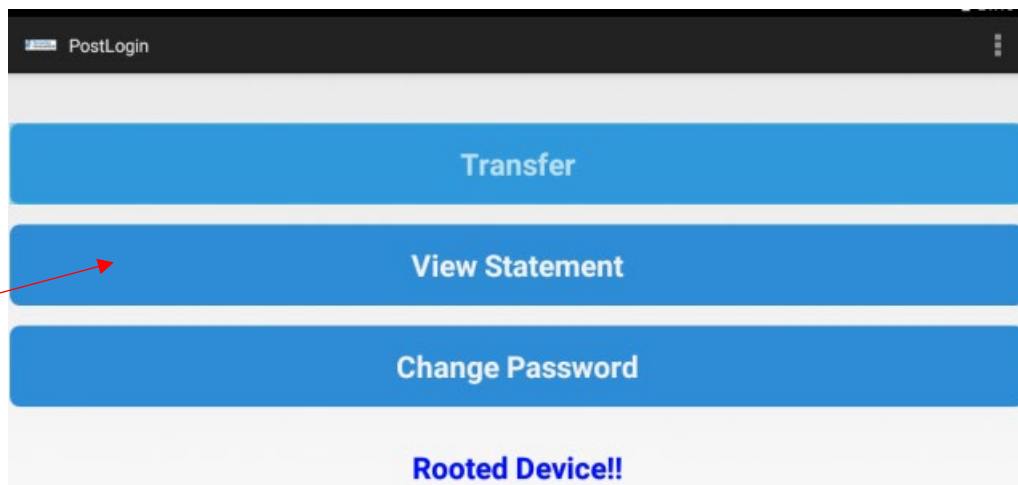


- A continuaci3n, pulsaremos el bot3n *Forward*.



Podemos ver como en la respuesta del servidor se confirma la transferencia de 1020 y no de 102 como indico el usuario en la app de Android.

Comprueba si InsecureBank.apk es consciente de estos cambios, accediendo a “View Statement”.



4 Conclusiones

Al finalizar el análisis dinámico realizado a la aplicación InsecureBankv2 y por extensión a su servidor, hemos podido detectar varias vulnerabilidades.

Como ejercicio redacta un breve informe donde se indiquen dichas vulnerabilidades con el fin de tenerlas identificadas para su posterior solución en la práctica siguiente. Sube este informe a tu espacio compartido para su evaluación por parte del profesor.