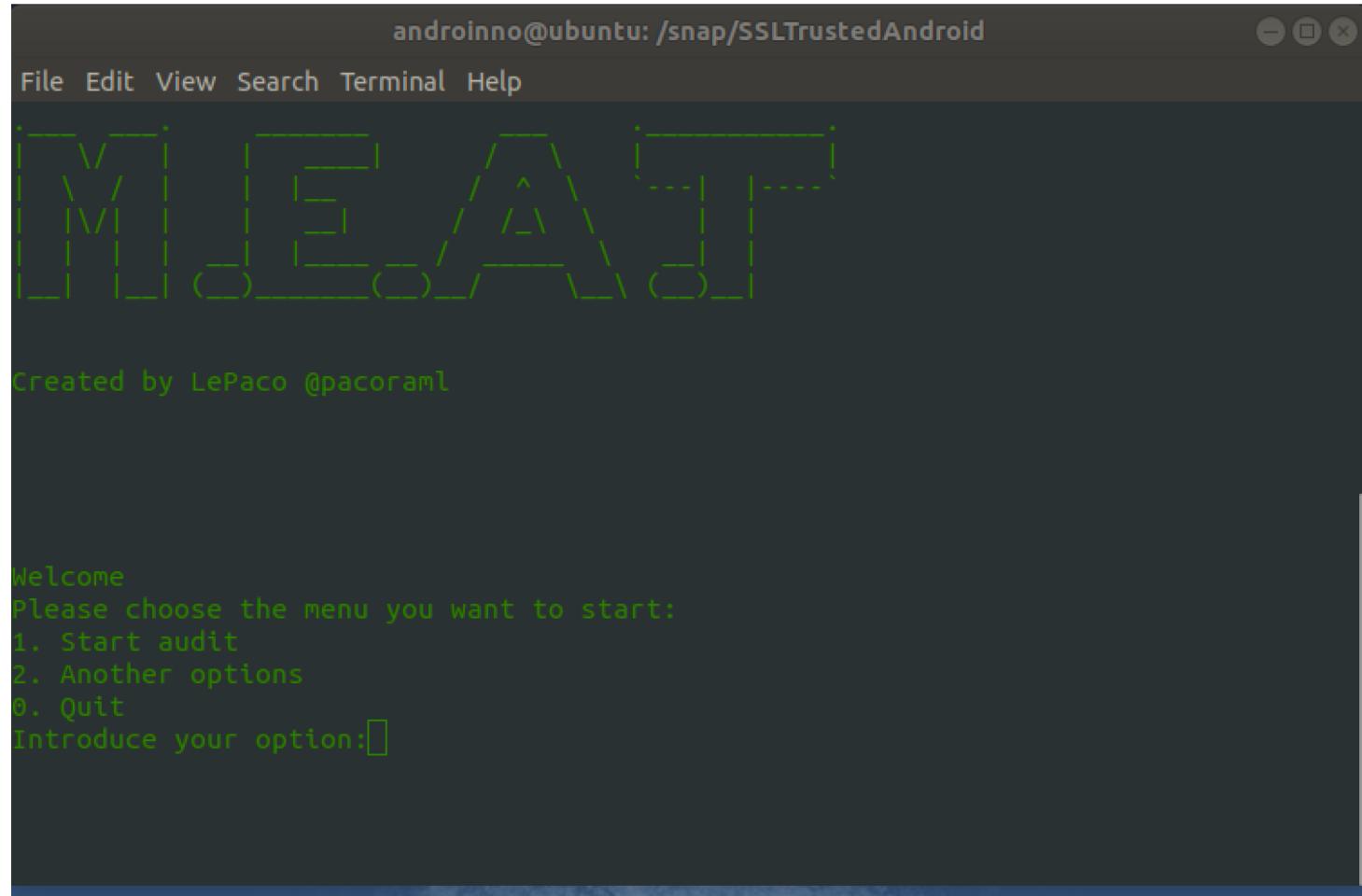
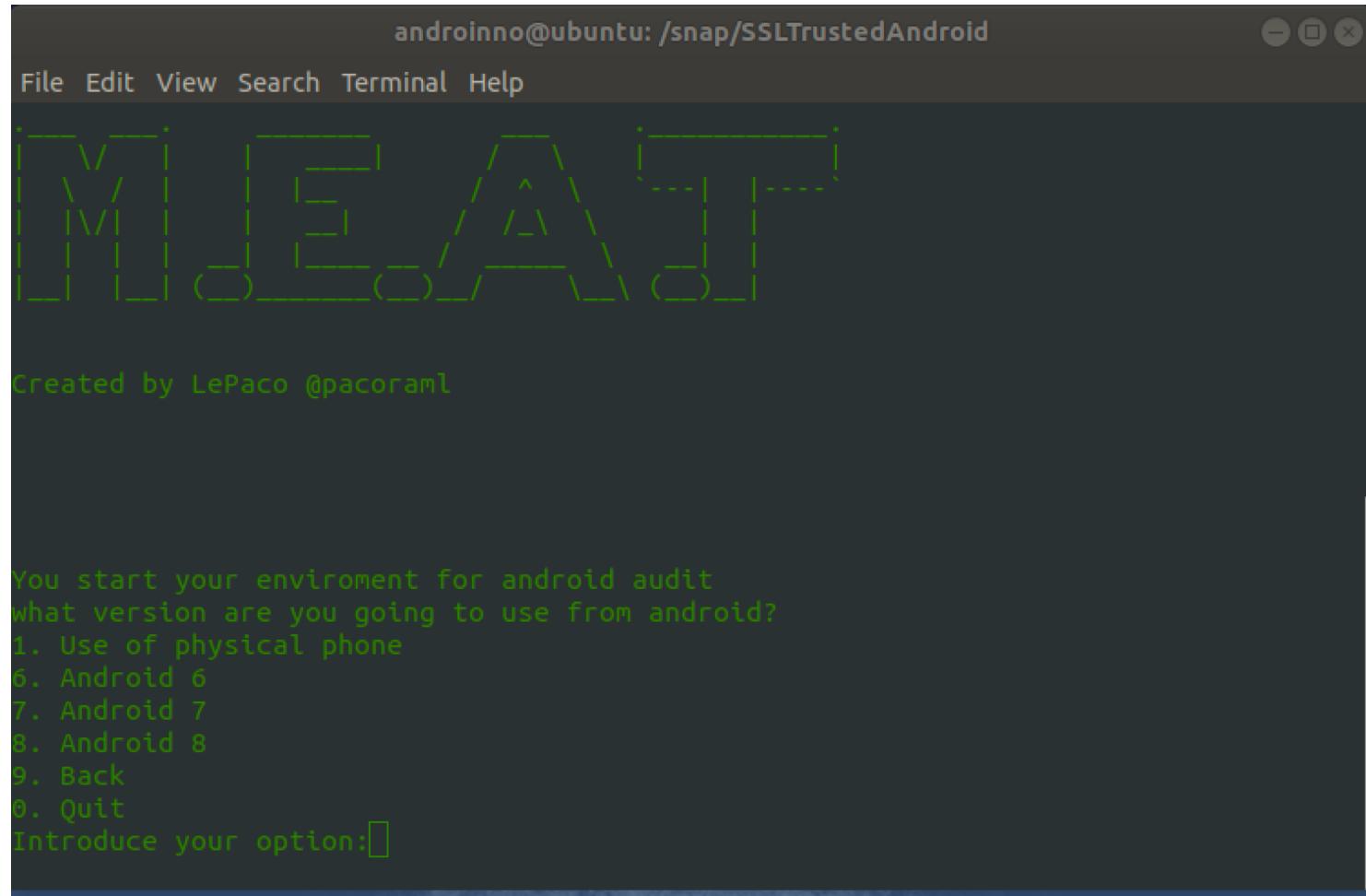


Entornos de prueba

- Hemos creado un menú sencillo donde simplemente tienes la opción directa de comenzar una auditoria o tener otras opciones para ayudar en la auditoria

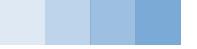


Si decidimos comenzar una auditoria lo primero seria escoger un entorno para probar nuestra aplicación, en nuestro menú

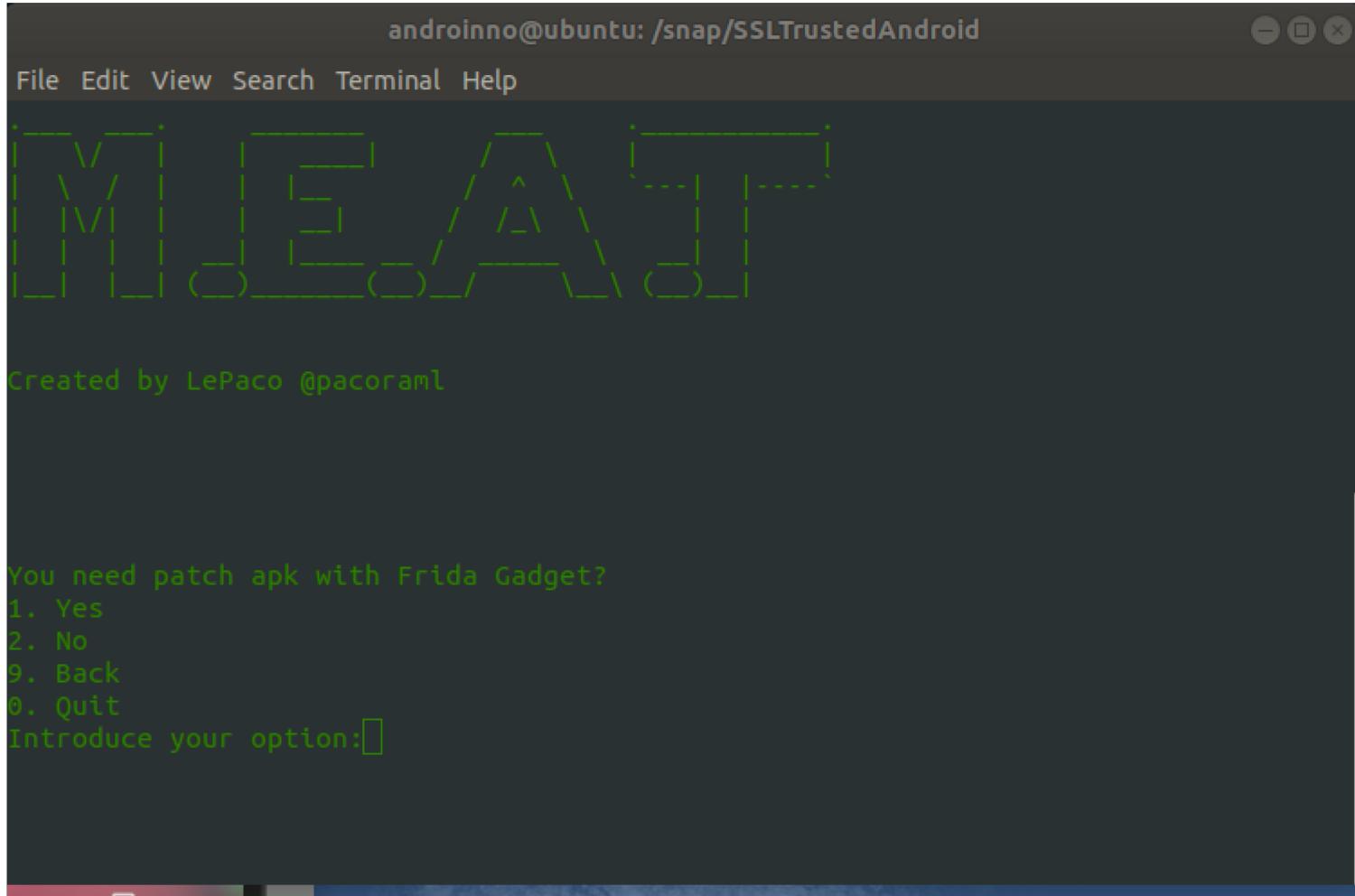


Si decidimos comenzar una auditoria lo primero seria escoger un entorno para probar nuestra aplicación, en nuestro menú

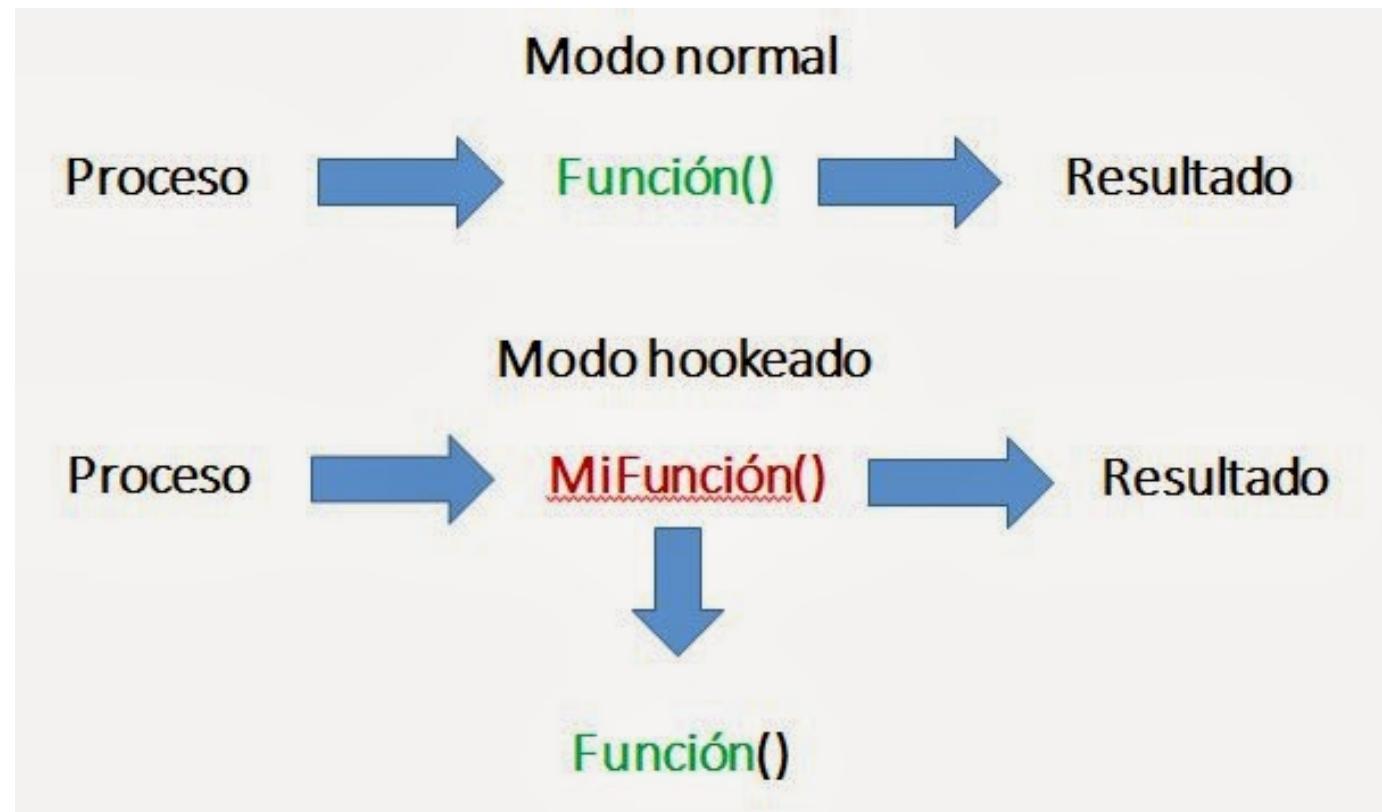
- Debemos tener en cuenta los siguientes puntos, para elegir el adecuado
 - Dispositivos emulados : Pueden ser mas cómodos para realizar la auditoria que un dispositivo físico, pero no tienen arquitectura ARM y es probable que nos falle alguna librería. Lo malo es también que algunos emuladores suelen fallar bastante...
 - Dispositivo físico: No es tan cómodo para hacer las pruebas, ya que lo tenemos que tener conectado a nuestro PC, pero son bastante mas fiable a la hora de hacer pruebas. No es posible hacer pruebas en diferentes versiones.

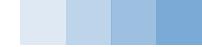


Una de las herramientas mas usadas por auditores de aplicaciones móviles es Frida y su framework vitaminado Objection



Hooking se llama al proceso de depurar una función dinámicamente para poder modificar su resultado, todo esto modificando sus registros en memoria y es todo dinámico.





Hay herramientas que nos ayudan a modificar estos procesos en memoria y sacar otro resultado, la mas conocida y que vamos a usar en este taller es Frida.

FRIDA

A continuación vamos a mostrar un video de como se puede hacer un bypass de una función de detección de root en un dispositivo Android para que veáis como funciona Frida.



Para usar Frida le tenemos que pasar un fichero en JavaScript que tenga la siguiente estructura y decir que función queremos hacer hooking y cual queremos que sea su resultado.

```
jscode = """"
Java.perform(function () {
    // Function to hook is defined here
    var MainActivity = Java.use('com.example.secccon2015.rock_paper_scissors');

    // Whenever button is clicked
    MainActivity.onClick.implementation = function (v) {
        // Show a message to know that the function got called
        send('onClick');

        // Call the original onClick handler
        this.onClick(v);
    };
});
```

```
// Set our values after running the original onClick handler
this.m.value = 0;
this.n.value = 1;
this.cnt.value = 999;

// Log to the console that it's done, and we should have the final value
console.log('Done:' + JSON.stringify(this.cnt));
};
```



Computer



Labs



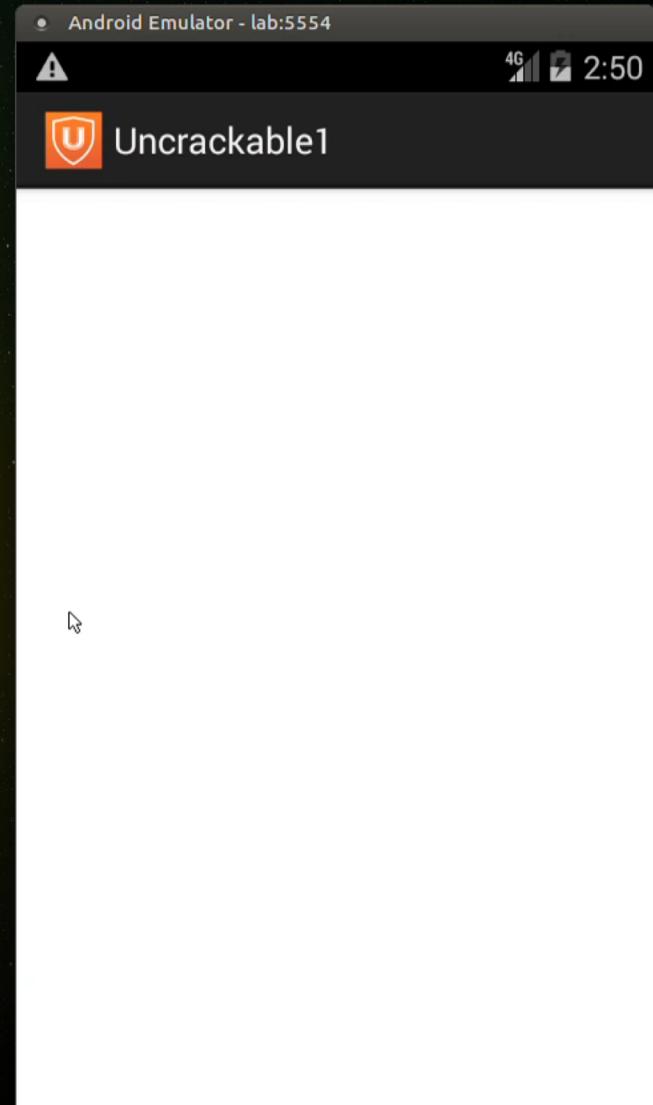
Tools

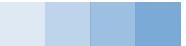


Emulator



Curso_Auditorias_Movil





- La manera mas fácil de usar Frida es con el framework objection que nos ofrece un nuevo terminal donde podemos hacer hooking de funciones de una manera fácil, como por ejemplo evadir SSL Pinning.

```
Terminal
File Edit View Search Terminal Help
Using USB device `Android Emulator 5554`
Agent injected and responds ok!

[object]inject([ion]) v1.5.3

Runtime Mobile Exploration
by: @leonjza from @sensepost

[tab] for command suggestions
com.mediafire.android on (Android: 6.0) [usb] #
```

- Pero antes debemos conocer varios opciones que nos permite objection y una de ella es parchear el fichero apk, para poder usar sin necesidad de tener root en el dispositivo móvil.
- Opciones:
 - objection patchapk –s <apkfile>
 - N para añadir los ficheros de network security config (> Android 6.0)
 - d para activar la depuración en el apk



En nuestra herramienta añadimos la opción por si quieras añadir la configuración del network security config

The terminal window has a dark background and a light gray title bar. The title bar displays the terminal session information: 'androinno@ubuntu: /snap/SSLTrustedAndroid'. Below the title bar is a standard Linux-style menu bar with options: File, Edit, View, Search, Terminal, and Help. The main area of the terminal shows a stylized green logo composed of brackets and arrows. Below the logo, the text 'Created by LePaco @pacoraml' is displayed in green. At the bottom of the terminal window, there is a command-line interface with the following text:
You need patch apk with add NetworkSecurityConfig? (Android > 6.0)
1. Yes
2. No
3. Back
0. Quit
Introduce your option:

- La nueva versión de Android cambiaba su configuración de comunicaciones...



Android 7.0 Nougat



- La nueva versión de Android cambiaba su configuración de comunicaciones...

Configuración de seguridad de la red



La configuración de seguridad de la red permite a las apps personalizar los ajustes de seguridad de la red mediante un archivo de configuración declarativo seguro, sin necesidad de modificar el código de estas. Estos ajustes se pueden configurar para dominios específicos y para una app específica. Las capacidades claves de esta función son las siguientes:

- **Anclajes de confianza personalizados:** establece de manera personalizada las autoridades de certificado (CA) de confianza para las conexiones de seguridad de una app. Puedes hacerlo, por ejemplo, otorgando confianza a certificados autofirmados particulares o restringiendo el conjunto de CA públicas de confianza para la app.
- **Anulaciones de solo depuración:** depura conexiones seguras en una app sin generar riesgos adicionales para la base instalada.
- **Desactivación del tráfico de Cleartext:** Protege las apps contra el uso accidental del tráfico de Cleartext.
- **Fijación de certificados:** limita la conexión segura de una app a certificados específicos.

- En nuestro fichero de configuración de la aplicación o conocido como AndroidManifest.xml tenemos que añadir lo siguiente :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
    <application android:networkSecurityConfig="@xml/network_security_config"
        ...
    ...
    </application>
</manifest>
```

- La entrada anterior hace referencia a un fichero que debemos añadir en la siguiente ruta res/xml/network_security_config.xml y que tiene la siguiente estructura:

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">example.com</domain>
        <trust-anchors>
            <certificates src="@raw/my_ca" />
        </trust-anchors>
    </domain-config>
</network-security-config>
```

- Pero la manera anterior es mas complicada ya que necesita de mas pasos para llegar a lo mismo que si creamos esta estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
- <network-security-config>
  - <base-config>
    - <trust-anchors>
      <certificates src="system"/>
      <certificates src="user"/>
    </trust-anchors>
  </base-config>
</network-security-config>
```

Con esto
decimos que
use los
certificados de
confianza del
sistema

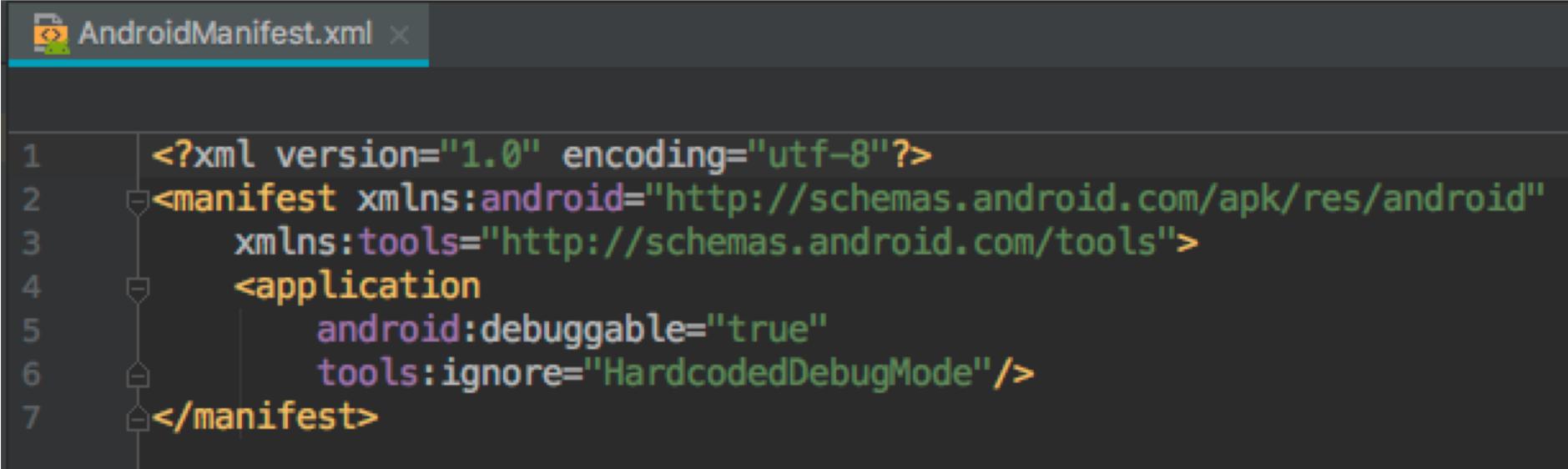
- Otra de las opciones que nos preguntaran en la herramienta es si queremos activar la depuración

The screenshot shows a terminal window titled "androinno@ubuntu: /snap/SSLTrustedAndroid". The window has a dark theme with light-colored text. At the top, there's a file tree icon. Below it, the text "Created by LePaco @pacoraml" is displayed. The main part of the terminal shows the following interaction:

```
You need patch apk with active debug mode?
1. Yes
2. No
3. Back
0. Quit
Introduce your option:1
Introduce apk path: app.apk
```



- Para activar la depuración es necesario modificar el fichero AndroidManifest en la aplicación



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:debuggable="true"
        tools:ignore="HardcodedDebugMode"/>
</manifest>
```



- Después nos preguntara si queremos instalar el apk que hemos parcheado en el dispositivo y este se lancara automáticamente
- Tambien nos preguntara si queremos lanzar la terminal de objection

```
androinno@ubuntu: /snap/SSLTrustedAndroid
File Edit View Search Terminal Help
Created by LePaco @pacoraml

You need install apk in device?
1. Yes
2. No
9. Back
0. Quit
Introduce your option:[]

androinno@ubuntu: /snap/SSLTrustedAndroid
File Edit View Search Terminal Help
Created by LePaco @pacoraml

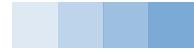
You need launch objection terminal?
1. Yes
2. No
9. Back
0. Quit
Introduce your option:[]
```

- Desde el terminal de objection es posible desactivar el control del SSL Pinning

The screenshot shows a terminal window titled "Terminal". The terminal output is as follows:

```
File Edit View Search Terminal Help
Using USB device `Android Emulator 5554`
Agent injected and responds ok!
[object]inject([ion) v1.5.3
Runtime Mobile Exploration
by: @leonjza from @sensepost

[tab] for command suggestions
com.mediafire.android on (Android: 6.0) [usb] # android sslpinning disable
(agent) Custom TrustManager ready, overriding SSLContext.init()
(agent) Found com.android.org.conscrypt.TrustManagerImpl, overriding TrustManagerImpl.verifyChain()
(agent) Found com.android.org.conscrypt.TrustManagerImpl, overriding TrustManagerImpl.checkTrustedRecursive()
(agent) Registering job yp9ri4ttsto. Type: android-sslpinning-disable
com.mediafire.android on (Android: 6.0) [usb] #
```



- Por ultimo el menú te preguntara si quieres que levantes un proxy para poder ver las comunicaciones que pasan por el dispositivo

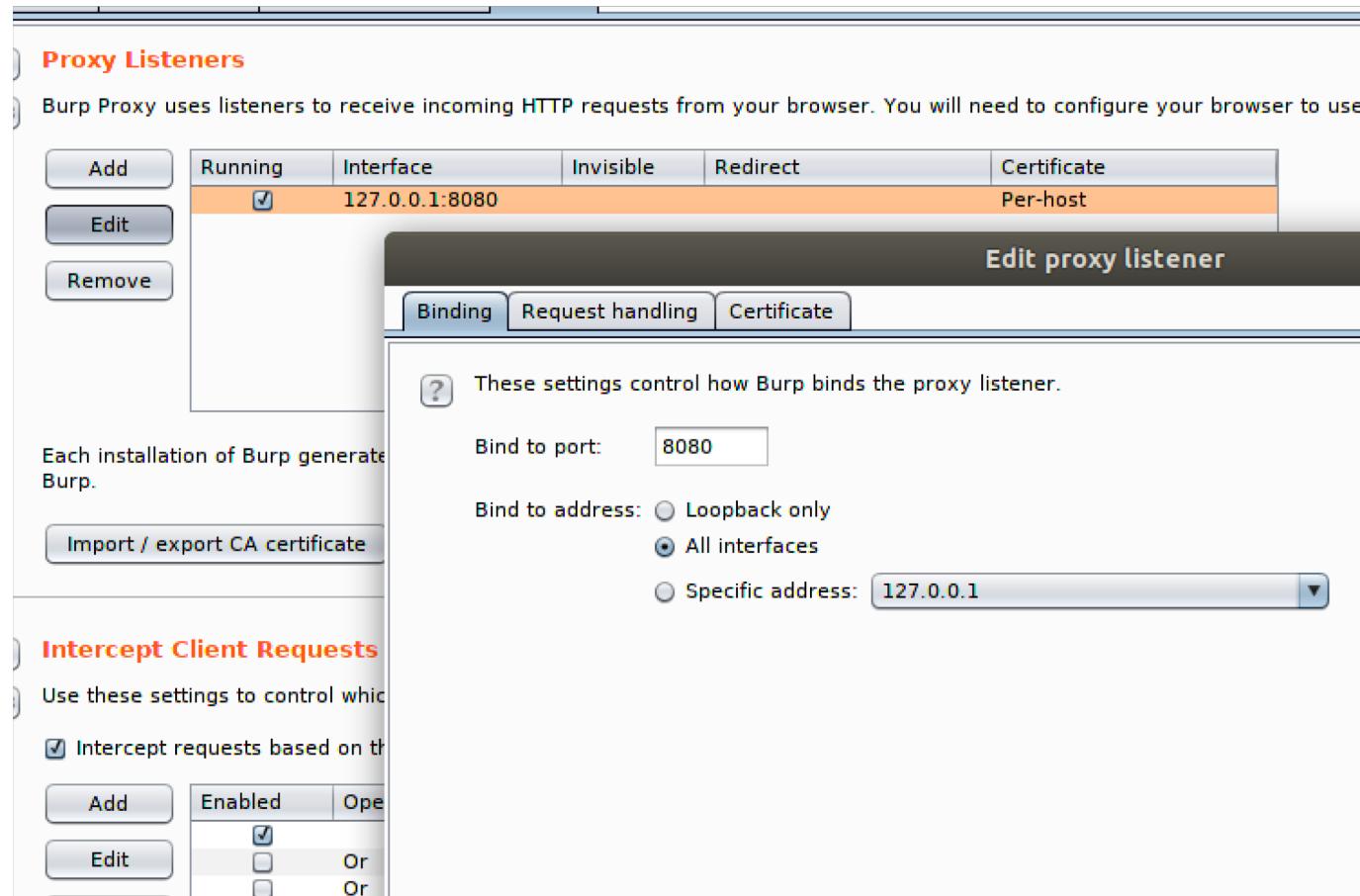
The screenshot shows a terminal window with the following content:

```
androinno@ubuntu: /snap/SSLTrustedAndroid
File Edit View Search Terminal Help
[Decorative icons representing file, edit, view, search, terminal, and help functions]

Created by LePaco @pacoraml

You need launch proxy?
1. Yes
2. No
9. Back
0. Quit
Introduce your option: [Input field]
```

- Deberemos activar para escuchar en todos las interfaces de la maquina



Además incluye algunos extras para las auditorias

- **Otras opciones**

- Se ha incluido un análisis estático con la herramienta MARA Framework
- Se puede rootear el dispositivo y abrir un terminal con acceso de root

DEMO TIME

