DOCUMENTACIÓN TFM

# Abreviaturas

**MiTM:** Man in The Middle.

**HTTP:** Hypertext Transfer Protocol.

**HTTPS:** Hypertext Transfer Protocol Secure.

**SSL/TLS:** Secure Socket Layer/Transport Layer Security.

**CA:** Certification Authority.

**PKI:** Public Key Infrastructure.

**APK**

# Definiciones de conceptos.

**Modelo cliente-servidor:** este tipo de arquitectura se basa en que los servicios y recursos de un determinado software son proporcionados por un servidor y, aquellas entidades que demandan estos servicios y recursos son los clientes.

**HTTP:** es un protocolo de comunicación que permite la transferencia de datos e información comúnmente utilizado en la World Wide Web (www). En la capa de transporte es soportado por el protocolo TCP y en el nivel de red por el IP. Este protocolo sigue un modelo cliente-servidor. El cliente realiza peticiones al servidor (request) y el servidor proporciona recursos y servicios contestando mediante una respuesta (response). Este protocolo tiene definidos varios métodos para poder realizar las peticiones y en las respuestas se incluyen códigos para verificar el estado de la comunicación. El puerto estándar para este protocolo es el 80.

**HTTPS:** protocolo HTTP pero que utiliza el cifrado del protocolo SSL/TLS en las comunicaciones.El protocolo funciona igual que HTTP, pero l diferencia es que al cifrar las comunicaciones ya no se pueden realizar ataques estilo MiTM ya que al interceptar las tramas HTTPS estarían cifradas.

**SSL/TLS:** SSL es un protocolo predecesor del protocolo TLS. Estos protocolos criptográficos consisten en proporcionar confidencialidad e integridad en las comunicaciones en una red de comunicación. Tratan de garantizar que la información que es transmitida por un canal no sea modificada ni interceptada y, que los integrantes de la comunicación puedan acceder a ella. Tal y como se muestra en [1], estos protocolos son muy utilizados para la seguridad de las comunicaciones web, pero para la situación de este proyecto, hay que tener en cuenta que para dispositivos Android se utilizan estos protocolos para la comunicación cifrada.

**Certificados digitales:** los certificados en este contexto, hacen referencia a mecanismos o utilidades a partir de las cuales se puede confirmar la autenticidad de la información. Además, también se consigue mantener la integridad del mensaje y el no repudio. Este mecanismo utiliza un par de claves, la clave privada conocida únicamente por el propio usuario y, la clave pública, que es conocida por todos.

**Autoridad de certificación:** son autoridades reconocidas las cuales se encargan de generar y mantener los certificados digitales ya que de alguna manera se tienen que comprobar que sean válidos. Para saber si los certificados son válidos, se emplean cadenas de confianza de tal manera que atravesando certificados se llega al certificado raíz, y este certificado es el que ha sido generado directamente por una autoridad de certificación.

**Certificate pinning (fijación de certificados):** según[2], certificate pinning es un mecanismo que consiste en que los certificados se incrustan directamente en las aplicaciones del dispositivo Android, de esta manera, aunque si se intercepta la comunicación no se podrá obtener el certificado.

**Ataques Man in The Middle:** este ataque se basa en que el atacante se sitúa en medio de la comunicación entre los integrantes, de manera que todo el tráfico de la comunicación pasa por el atacante. De esta manera, puede obtener, observar y modificar la información y reenviarla de nuevo o manipularla de cualquier forma.

**Integridad:** este principio básico de la seguridad de la información versa sobre que la información que es transmitida no sea modificada durante el trayecto.

**Confidencialidad:** principio de la seguridad de la información que consiste en que si se intercepta la información o el mensaje sea ininteligible, de manera que únicamente los verdaderos integrantes de la comunicación puedan comprender el contenido del mensaje.

**Autenticidad:** este principio consiste al recibir información de alguna de las partes de la comunicación, sea enviada realmente por ese integrante y no por otra entidad que no pertenece a la comunicación.

# Definiciones de herramientas utilizadas.

**Máquina virtual:**

**Oracle virtual box:**

**Androidx86:**

**Tsurugi:**

**Docker:**

**Python:**

**IDA:**

**ADB: Android Debug Bridge 🡪 COMANDOS (solo para mí, luego eliminar de aquí) 🡪 adb shell pm list packages (para ver las apps instaladas en el móvil); adb pull /data/app/package-name-1.apk (para hacer un pull de la app en el móvil).**

**Apktool:**

**¿?Smali source: desensamblado**

**Dex2jsr: convierte DEX (Dalvik bytecode) to JAR (java bytecode).**

**Java decompilers: Jd-gui, JAD, Dare, Procyon, etc. Se utilizan para transformer el JAR en código Java y poder de esta manera obtener el código del apk.**

# Estado del arte.

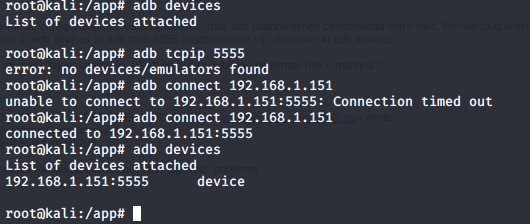
# Instalación y configuración del entorno.

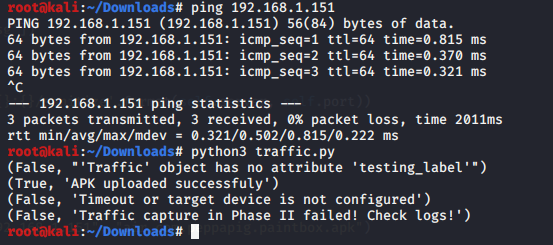
El entorno con el que se trabaja y con el que se desarrollará todo el estudio de la manera en la que las aplicaciones hacen uso de los mecanismos de seguridad de Certificate Pinning es mediante dos máquinas virtuales. En este caso, se ha decidido utilizar el entorno Virtual Box para poder realizar la instalación y configuración de todas las máquinas que se necesitan para realizar el estudio. Se ha escogido Virtual Box en vez de otros sistemas de virtualización por comodidad y por familiaridad.

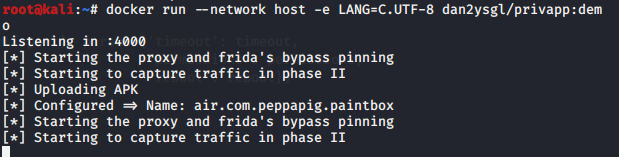
La primera máquina virtual se utilizará para instalar la imagen de Docker que es la que contiene la plataforma desarrollada por otros trabajos posteriores. El sistema operativo que se utilizará en esta primera máquina virtual es Kali Linux, se ha escogido esta distribución ya que proporciona múltiples herramientas que se podrán utilizar posteriormente para el análisis de las aplicaciones Android. La segunda máquina virtual tendrá instalado el sistema Android x86 el cual simula un dispositivo Android y el que se utilizará para instalar las aplicaciones Android y analizar su comportamiento como si se tratara de un dispositivo móvil.

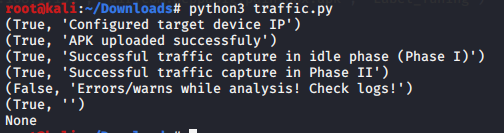
Una vez instaladas las dos máquinas virtuales, es necesario establecer los adaptadores de red adecuados para conseguir la conexión entre









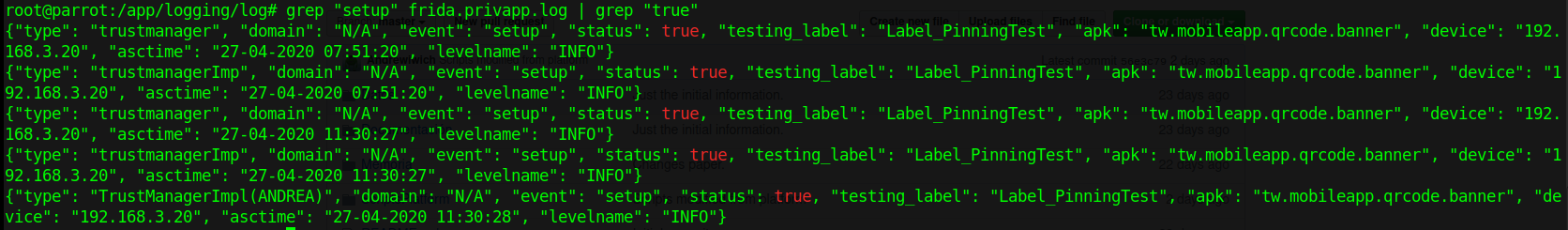


# ESTUDIO DE CASOS DE APKS QUE NO LOGRAN BYPASS DE CP

**Caso 1:**

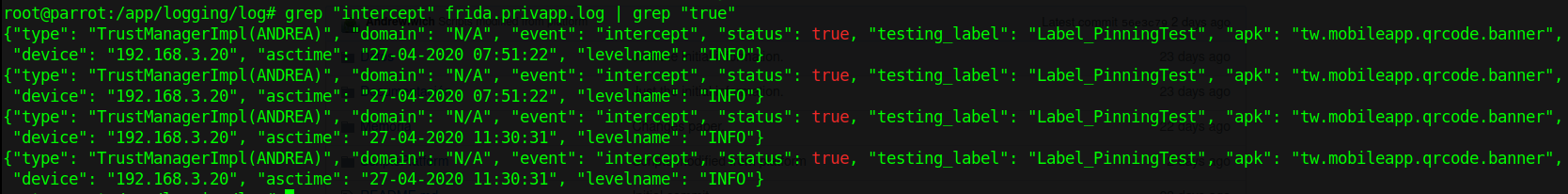
APK 🡪 tw.mobileapp.qrcode.banner

Fallo bypass 🡪 Se detecta que el APK hace uso de las librerías trustmanager y TrustaManagerImpl



Solución:

* Librería TrustManagerImpl. Se analizan las diferentes funciones de la librería y se utiliza el código que se muestra a continuación, encontrado en <https://gist.github.com/1mm0rt41PC/cd492f24dc061019fb25222ba0b96d20>. Tras realizar modificaciones e implementar el bypass para esa librería se consigue saltar tal y como se muestra en la siguiente imagen.



# Bibliografía

|  |  |
| --- | --- |
| [1] | «Redalia,» [En línea]. Available: https://www.redalia.es/ssl/protocolo-ssl/. [Último acceso: 08 03 2020]. |
| [2] | «IBM Mobile Foundation,» 27 02 2020. [En línea]. Available: https://mobilefirstplatform.ibmcloud.com/tutorials/es/foundation/8.0/authentication-and-security/certificate-pinning/. [Último acceso: 08 03 2020]. |