UNAH
UNIVERSIDAD NACIONAL
AUTÓNOMA DE HONDURAS

Facultad de Ingeniería

Departamento de Ingeniería en Sistemas

Auditoria Informática (IS-903)

Ing. Elmer Padilla

Herramientas para auditoria de dispositivos móviles.

Presentado por:

Génesis Raquel Izaguirre López 20151005232

Ciudad Universitaria, Tegucigalpa MDC, Francisco Morazán

Mayo, 2020.

Índice

Introducción

La proliferación de teléfonos inteligentes ha generado un escenario alarmante debido al rápido crecimiento de las aplicaciones móviles. Los delincuentes cibernéticos pueden obtener fácilmente credenciales para obtener acceso y robar datos críticos o información privada a través de las redes. El uso masivo de tecnologías inalámbricas coloca a los dispositivos móviles como uno de los principales objetivos de los atacantes cibernéticos. Las aplicaciones móviles se utilizan para uso personal, pero también pueden estar conectadas con su empresa, lo que compromete la seguridad de su organización si deja un espacio sin protección. Hoy en día, las noticias están llenas de historias sobre ciberataques y vulnerabilidades. Todos los desarrolladores deben asegurar sus aplicaciones de manera eficiente antes de lanzarlas. Comience ahora a auditar su aplicación móvil para evitar incidentes y brechas de seguridad pública que podrían dañar la reputación de su empresa.

Elementos vitales en una auditoria de dispositivos móviles

Seguridad en la transmisión de datos de la aplicación móvil

- → Mecanismos de autenticación existentes
- → Capa de transporte y mecanismos de cifrado (HTTP, HTTPS, SSL, TLS,..)
- → Comprobación de certificados digitales (certificate pinning)
- → Identificación de recursos con los que se establece la conexión.

Auditoría de aplicaciones móviles / revisión de la App Móvil

- → Desempaquetado de la aplicación dependiendo de su formato (APK, IPA, ALX, JAD, XAP)
- → Detección de protectores y ofuscadores de código.
- → Análisis y auditoría de código fuente.
- → Análisis de la información almacenada por la app móvil.
- → Análisis de los mecanismos de almacenamiento de la plataforma.
- → Mecanismos de protección de acceso a los datos de la app.
- → Carga o ejecución externa de contenido.

Revisión de seguridad del EndPoint

- → Análisis del sistema contra la que se conecta la app móvil (Webservice,..)
- → Análisis de controles de seguridad basados en metodología OWASP.
- → Pruebas sobre mecanismos de autenticación y autorización.
- → Pruebas de suplantación de identidad del servidor.
- → Interceptación de credenciales de acceso o de información intercambiada.
- → Consulte con nosotros cómo nuestro servicio de auditoría de aplicaciones móviles puede ayudar a su empresa a proteger la seguridad de sus Apps y de sus servicios online.

Herramientas

Santoku Linux

es una distribución basada en Linux especialmente desarrollada para auditar dispositivos móviles en busca de vulnerabilidades, fallos o simplemente cualquier aspecto que pueda comprometer nuestra privacidad al utilizar cualquiera de estos dispositivos móviles.

Este sistema operativo, que se distribuye de forma gratuita y con el código tanto del sistema como de todas las herramientas totalmente abierto y disponible para cualquiera interesado en revisarlo, incluye por defecto todos los SDK necesarios para que las herramientas funcionen sin problemas, así como controladores, interfaz gráfica para facilitar el uso de varias herramientas y una configuración que auto-detecta cualquier dispositivo que se conecte a nuestro PC para poder empezar a auditarlo.

En cuanto a las herramientas que incluye Santoku Linux podemos clasificarlas en 3 apartados:

Análisis forense

Este apartado ofrece a los usuarios una lista con imágenes de los sistemas operativos oficiales para un gran número de marcas y modelos de manera que se pueda instalar una versión del sistema operativo limpia en el dispositivo desde donde empezar a realizar los análisis. Igualmente disponemos de una serie de herramientas con las que poder analizar tanto la memoria interna como la ROM y la RAM en busca de información residente en dichas memorias.

Análisis de malware móvil

Dentro de este apartado podemos mencionar una serie de emuladores que permiten utilizar máquinas virtuales con sistemas operativos móviles con el fin de poder analizar el funcionamiento de los diferentes sistemas operativos móviles. También dispone de herramientas para crear redes virtuales desde donde monitorizar las conexiones del malware y scripts que facilitan la tarea de de-compilación y des-ensamblado de este software malicioso.

Análisis de seguridad

Desde aquí podremos acceder a una lista de herramientas y scripts que nos facilitan la tarea de detectar vulnerabilidades y fallos comunes en el software convencional, así como varios scripts que facilitan la tarea de descifrado y de-compilación de herramientas y aplicaciones para un análisis más en profundidad.

Podemos descargar esta distribución forense de dispositivos móviles sin coste alguno desde su página web principal. Debemos recordar que Santoku Linux aún se encuentra en fase de desarrollo por lo que puede contener fallos y errores y es posible que en futuras versiones mejore su funcionamiento tanto interno como el de las herramientas que forman esta suite.

PidCat

es un *script* en Python que nos brinda la posibilidad de observar en tiempo real los registros de los procesos de una manera más colorida y ordenada.

Los elementos necesarios para la auditoría

Se utiliza un sistema operativo **Linux**, un emulador de **Android** y una aplicación con bajo nivel en lo que respecta a seguridad. Previamente debemos tener configurados en el archivo ".bashrc" los directorios (*paths*) de adb (Android Debug Bridge) y Android tools, que se encuentran integrados en el paquete de instalación de Android Studio. Es importante verificar estas configuraciones, ya que de lo contrario PidCat no funcionará. Para realizar ingeniería inversa

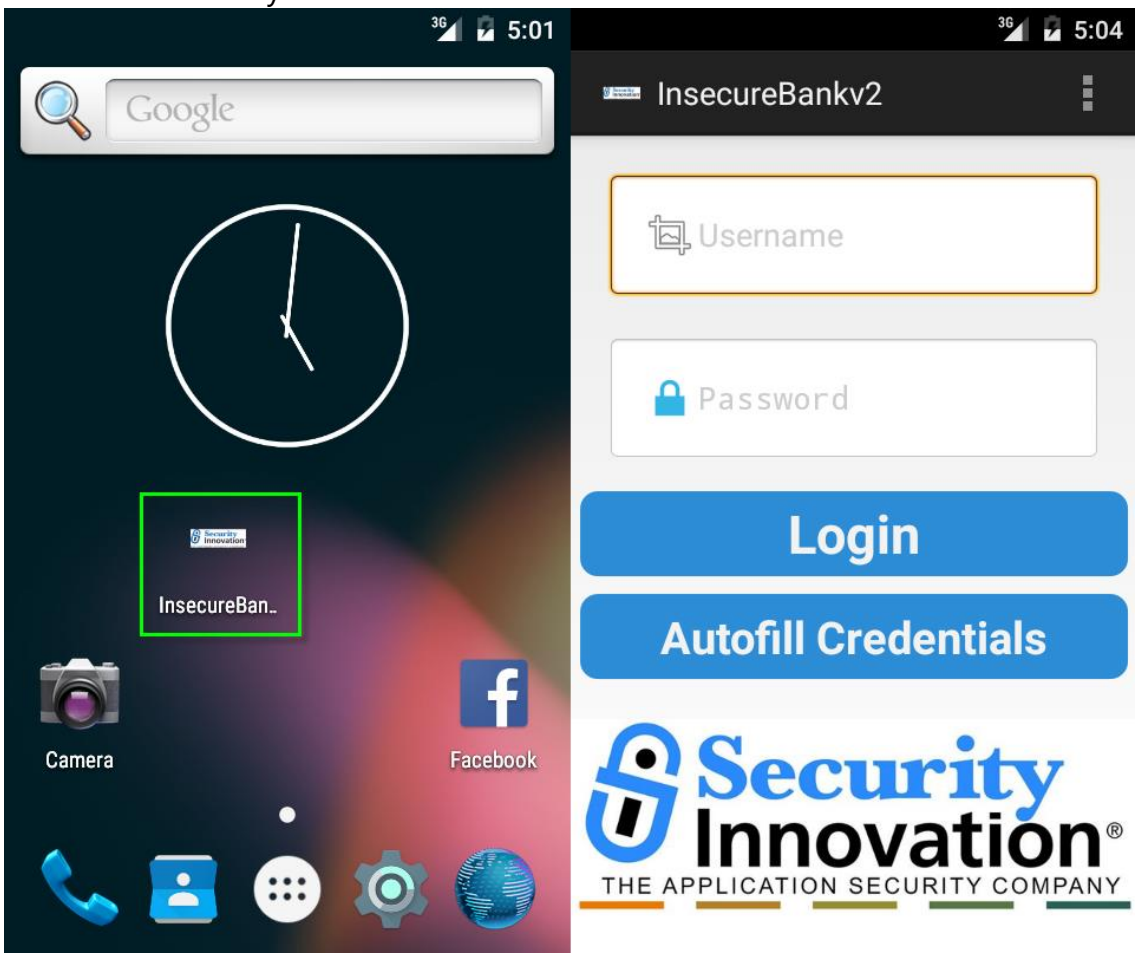
sobre la aplicación haremos uso de *Apktool*, que nos permitirá extraer información y también el archivo *AndroidManifest,* como puedes ver en la siguiente captura:

```
orion@orion-pc:~/sample$ apktool d InsecureBankv2.apk
I: Using Apktool 2.0.0-RC3 on InsecureBankv2.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/orion/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
orion@orion-pc:~/sample$ cd InsecureBankv2/
orion@orion-pc:~/sample/InsecureBankv2$ ls
AndroidManifest.xml  apktool.yml  original  res  smali
orion@orion-pc:~/sample/InsecureBankv2$
```

Este archivo nos proporciona la configuración básica de la aplicación. Dentro podremos buscar la etiqueta "android.debuggable", la cual nos indica si, dependiendo de su configuración, la aplicación es depurable o no. En la siguiente imagen observamos que dicha configuración se encuentra activada; este es un error grave, ya que se podría estar **exponiendo información** sensible de cualquier índole. Así, gracias a los registros o *logs* del sistema podremos extraer mucha información.

```
    <android:uses-permission android:name="android.permission.READ_PHONE_STATE"/
>
    <android:uses-permission android:maxSdkVersion="18" android:name="android.pe
rmission.READ_EXTERNAL_STORAGE"/>
    <android:uses-permission android:name="android.permission.READ_CALL_LOG"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-feature android:glEsVersion="0x20000" android:required="true"/>
    <application android:allowBackup="true" android:debuggable="true" android:ic
on="@mipmap/ic_launcher" android:label="@string/app_name" android:theme="@androi
d:style/Theme.Holo.Light.DarkActionBar">
        <activity android:label="@string/app_name" android:name="com.android.ins
```

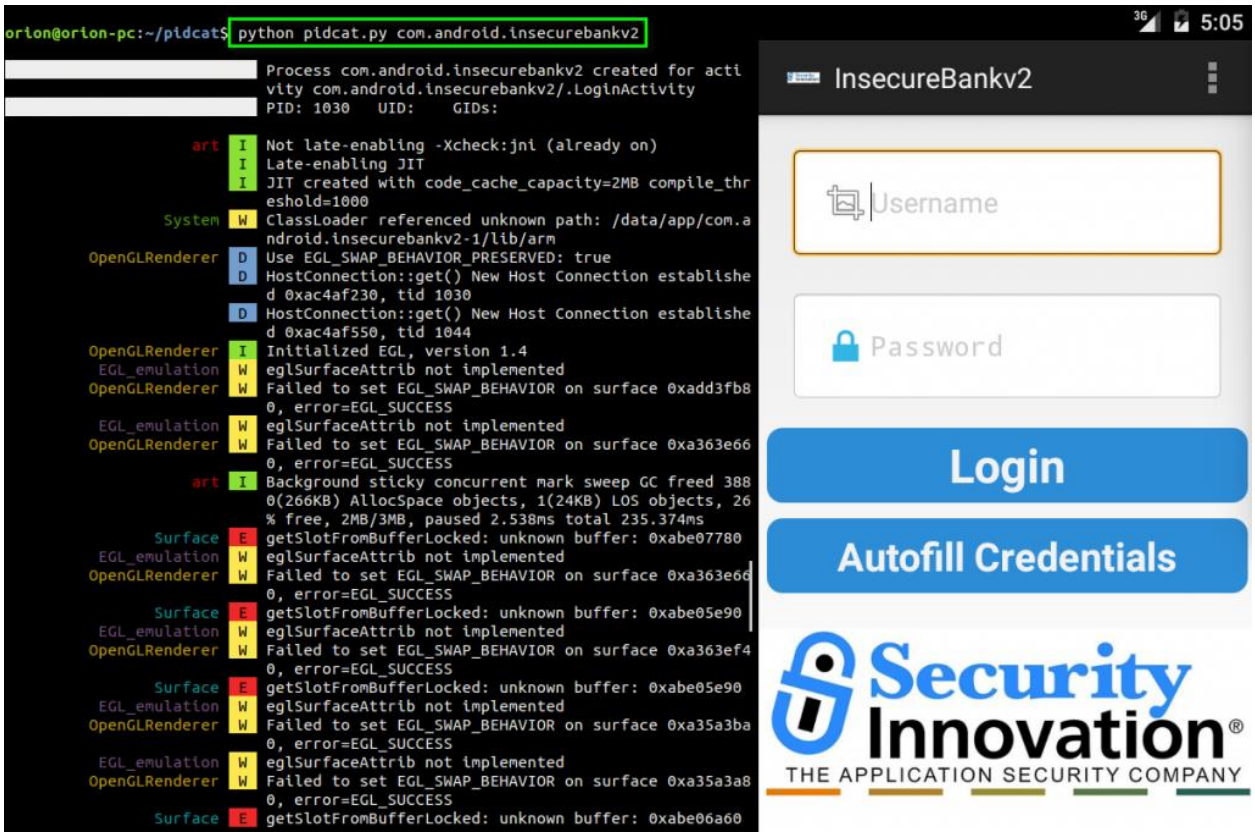
Una vez configurado el emulador de forma adecuada, lo iniciamos e instalamos la aplicación, abriendo una consola y usando el comando adb install Insecurebankv2.apk. Después de unos minutos observaremos que la aplicación vulnerable logró instalarse correctamente. En las siguientes capturas, observaremos que el aplicativo proporciona un formulario en donde solicita un usuario y una contraseña:



Para capturar los registros del sistema normalmente utilizaríamos el comando adb logcat, pero a veces puede ser más que desafiante tener que lidiar con una enorme cantidad de sucesos al momento de **analizar o encontrar errores**. Para ello, PidCat nos proporciona una vista más agradable y estructurada de los registros capturados. Luego de haber descargado PidCat procedemos a ingresar al directorio en el que se encuentre el *script*, en donde le especificaremos qué aplicación queremos auditar. En caso de no conocer concretamente el nombre del aplicativo, utilizaremos el comando adb shell pm list packages –f <packageName>.

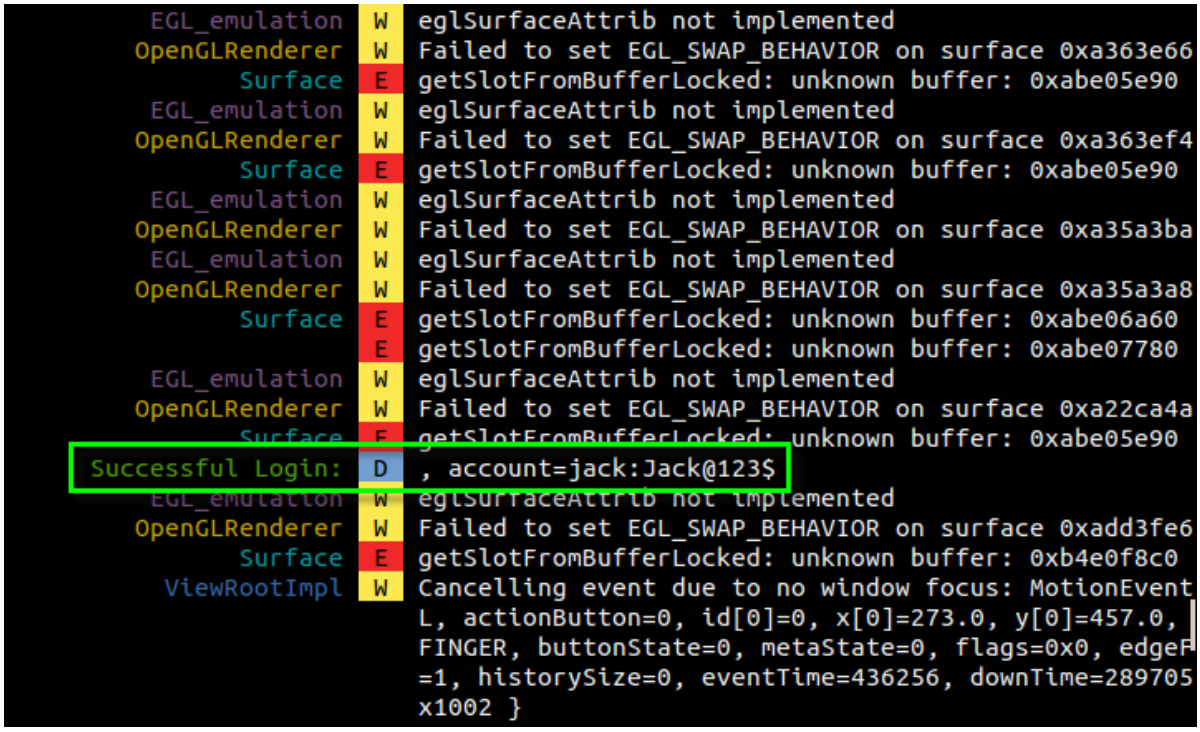
En la siguiente imagen podemos ver cómo esta poderosa herramienta comienza a registrar cada acción que se ejecuta. Por medio de etiquetas y colores, categoriza cada uno de los

sucesos, con lo cual nos da la posibilidad de que nuestra labor sea más confortable y podamos ahorrar tiempo al analizar y encontrar *bugs*.



Contraseñas expuestas

A continuación, observaremos cómo PidCat logró capturar una grave falla, gracias a la cual se puede ver claramente un usuario y su correspondiente contraseña en texto plano. Estaríamos ante un error o vulnerabilidad de la que algún cibercriminal o usuario mal intencionado podría sacar provecho.



Existen muchos otros tipos de herramientas o aplicaciones que nos pueden ayudar a encontrar o incluso solucionar algún error de seguridad en el desarrollo de software. PidCat es una buena y eficiente opción al momento de auditar qué es lo que sucede mientras algún aplicativo se encuentra en estado de ejecución.

Es importante tomar conciencia de estas cuestiones porque, de omitirse, no solo se estaría poniendo en riesgo a los usuarios que utilizan estas aplicaciones sino también al negocio de la o las compañías que formen parte del proyecto.

Existen diferentes cuestiones al momento de preguntarnos por dónde comenzar a preocuparnos, o dónde podríamos tener un mal diseño o implementación en el código fuente del software. Por ejemplo, de qué forma viajan los datos a través de la red, los permisos que cada usuario tiene y la autenticación, de qué forma se validan los datos, la cantidad de intentos que tiene una persona al momento de ingresar al sistema, entre otros.

Bibliografía

- https://www.redeszone.net/2015/03/14/santoku-linux-un-sistema-operativo-para-auditar-dispositivos-moviles/
- https://blogs.deusto.es/master-informatica/que-impacto-produce-el-mundo-movil-en-la-auditoria-interna-que-hacemos-byod/
  https://www.hackbysecurity.com/servicios-empresas/auditoria-informatica/auditoria-de-dispositivos-moviles