Entrega № 3 Proyecto Seguridad Informática

Auditoria de la Aplicación y Decompilación

Camilo Zepeda Hoffmann

Profesor Maximiliano Vega

26 de Octubre 2017

1. Resumen

Una vez la decompilación de la aplicación se procede a utilizar una serie de herramientas capaz de analizar los distintos ambientes encontrados, de entre los cuales, se destaca un análisis de Manifest, puertos y página online. Entonces se procede a describir una serie de componentes y respuestas encontradas.

2. Análisis de la Aplicación

Previo a la obtención y análisis de la aplicación dado los datos a obtener se propone definir los requerimientos base que conforman la actividad realizada. El listado de Herramientas utilizadas es:

- **Página de Descarga**: se utiliza una página que disponga de la app.
- **Lionsec**: dado que es necesario decompilar la aplicación, se utiliza el nombrado sistema operativo, debido a sus multiples funciones y herramientas que faciliten el trabajo.
- **Apktool**: herramienta que realiza la lectura y depuración de la aplicación.
- **Dex2jar**: herramienta utilizada para convertir los diferentes .dex propios de android a .jar.
- **Java Decompiler**: utilizado para lograr depurar los archivos .jar.

2.1. Obtención de la Aplicación

En primera instancia es necesario conocer que la aplicación se encuentra disponible tanto para sistemas operativos iOs como Android, donde si bien es posible descargar ambos formatos no es posible decompilar los dos por igual. Cabe aclarar que se refiere a decompilar, como la obtención de toda la data proveniente de la aplicación. Entonces si bien, iOs dispone de los layouts o vistas, las cuales pueden ser editadas no se logra editar y ver los códigos base de la app.

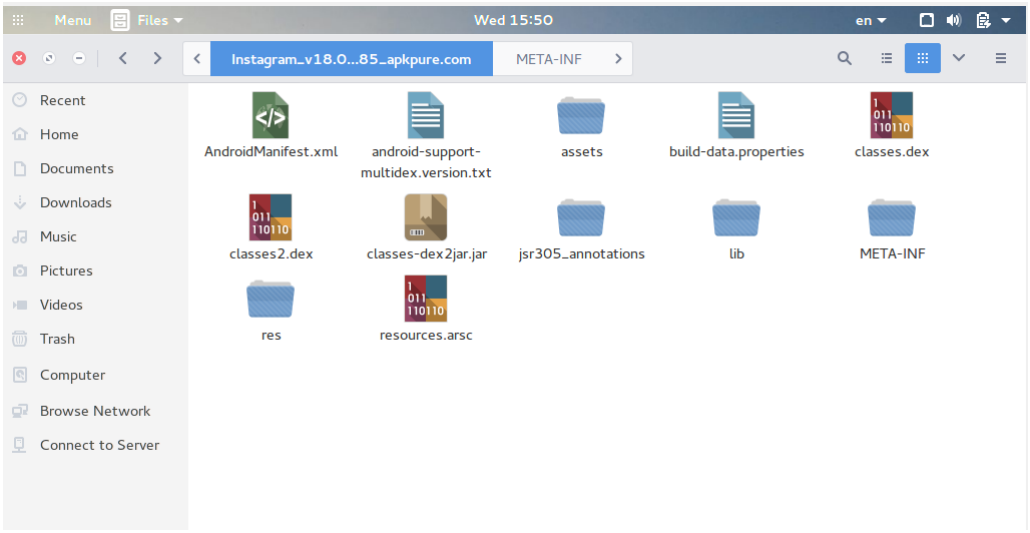
Entonces en concordancia con lo anterior, se realiza un análisis de la aplicación **.apk** (Sistema Operativo Android) al ser más amigable al momento de depurar, entregando una serie de archivos y datos correspondientes a la plataforma. Para lograr la obtención de la aplicación se hace uso de **ApkPure** página que dispone de la última versión de Instagram descargable para el computador. Ya en obtención de la aplicación y bajo el uso de las herramientas mencionadas anteriormente se procede a la depuración de la app.

2.2. Decompilación de la Aplicación

Con tal de lograr un mejor entendimiento de lo realizado se describe una serie de pasos de la mano con el material obtenido, en formato de imagen o similar.

- **Paso 1**: ya en posesión de la .apk correspondiente a la aplicación la primera actividad fue el transformar el archivo de:

```
Intagram.apk => Instagram.zip
```



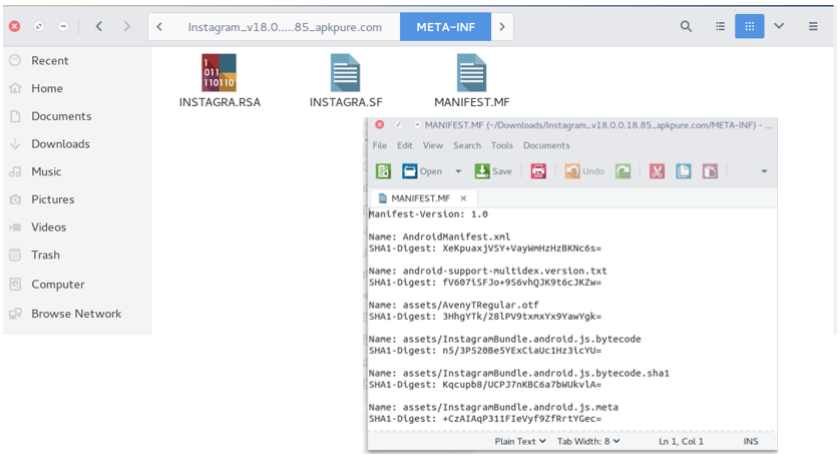
De la conversión anterior se obtiene acceso a los archivos en bruto de la aplicación, así no, no es posible observar los xml, java o react de la misma, ya que aún no se ha depurado a cabalidad. Igualmente, se obtienen los **.dex** equivalente a las clases utilizadas por la app.

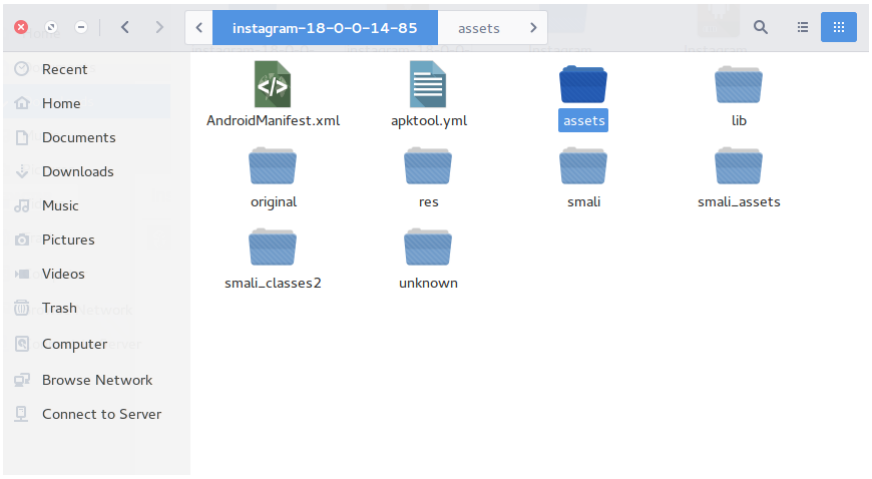
- **Paso 2**: ya extraídos los archivos anteriormente mencionados, se puede observar la carpeta **META-INF** equivalente al certificado de los desarrolladores de la aplicación, la cual al revisar los datos internos se determina que se usa Hashing Sha1 para el certificado de desarrollador y cifrado RSA, para el certificado entregado por Android. Lo dicho puede verse en la siguiente figura:

- **Paso 3**: Una vez realizado los pasos anteriores fue posible determinar que no se pueden ver tanto los xml, como los java provenientes de la aplicación. Es por esta razón que se hace uso de software **ApkTool** ya interno en Lionsec. Ya familiarizado con la herramienta se realiza el siguiente comando:

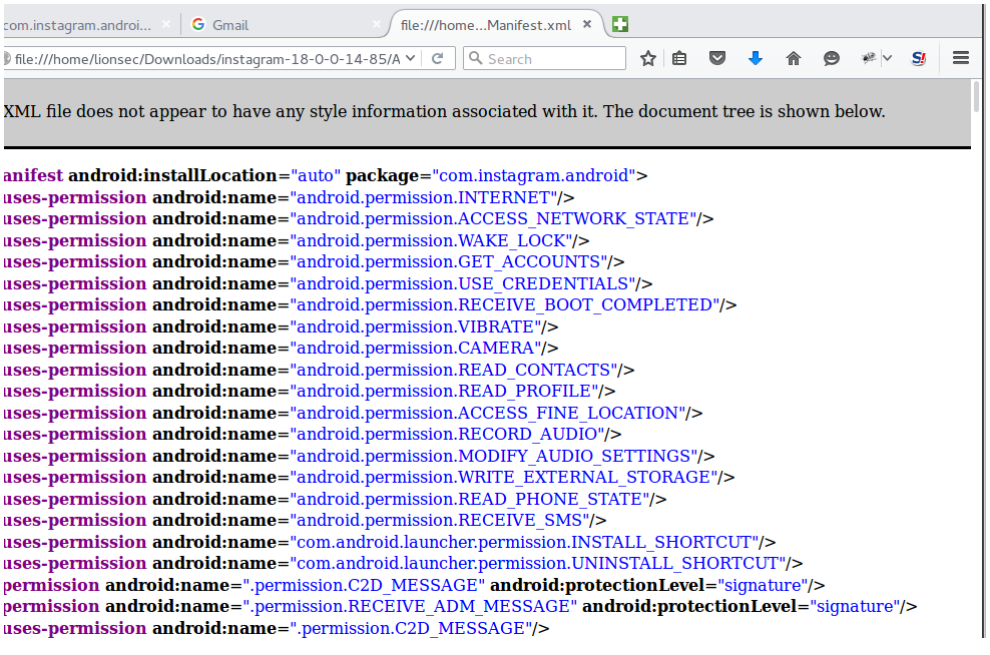
```
apktool d Intagram.apk
```

Donde se define **d** como la lectura del archivo, obteniendo entonces los siguientes archivos:





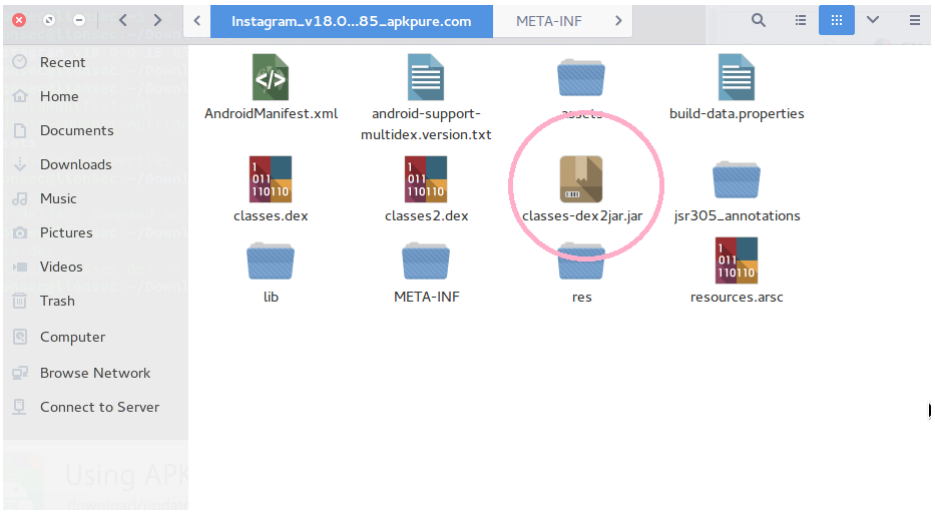
- **Paso 4**: Finalmente tras disponer de la información y datos pertinentes a la depuración de la aplicación. Es necesario revisar los datos obtenidos e información relevante. Los primeros datos revisados fueron los XmlAndroid, los cuales definen parte de los requerimientos y facultades que asocia la aplicación para su desempeño. La siguiente figura da una perspectiva de lo visto.



Si bien se puede encontrar una serie de disposiciones y permisos relacionados a la aplicación, no se considera un punto relevante para la depuración, ya que son puntos relativamente comunes para la aplicación, dígase, no se encuentra anomalías que puedan ser explotables en futuros incrementales.

Por otra parte, mediante el uso de la herramienta dex2jar, se realiza una conversión de las clases intrinsecas en la aplicación, mediante el comando:

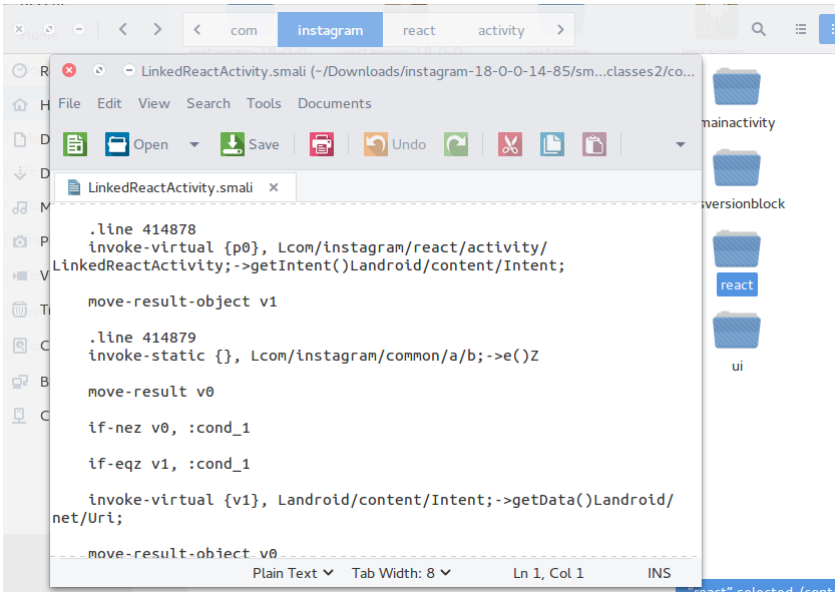
```
d2j-dex2jar classes.dex
```



Lo que convierte los archivos a un formato .jar, el que por medio de una Java Decompiler y jd-gui interno del mismo, puede lograrse un análisis de los archivos java desarrollados en la aplicación.

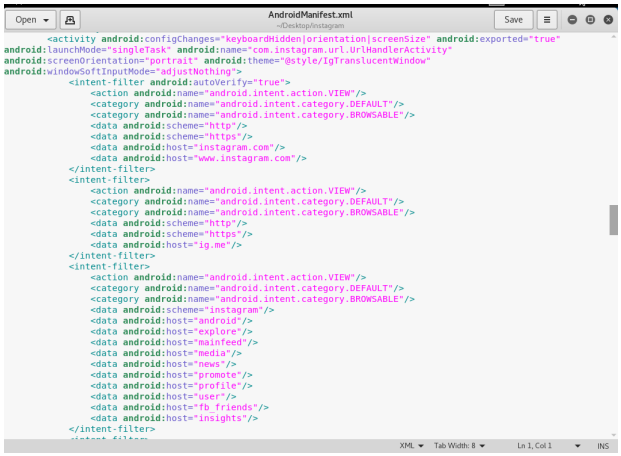
Por último se realiza una revisión de los archivos React, los cuales se encuentran en formato **.smali** propio de Java. Obteniendo entonces lo siguiente:

Cabe destacar que los archivos vistos en la figura anterior tienen una real validez en cuanto a vulnerabilidades explotables para la auditoría a realizar, ya que dichas opciones son las actividades, acciones y funciones propias de la aplicación, dando apertura a protocolos, urls, traspaso de información y otras actividades que compromenten la aplicación.



- **Paso 5**: dar a conocer los host encontrados dentro del archivo **Manifest**, donde se encuentra el siguiente host:

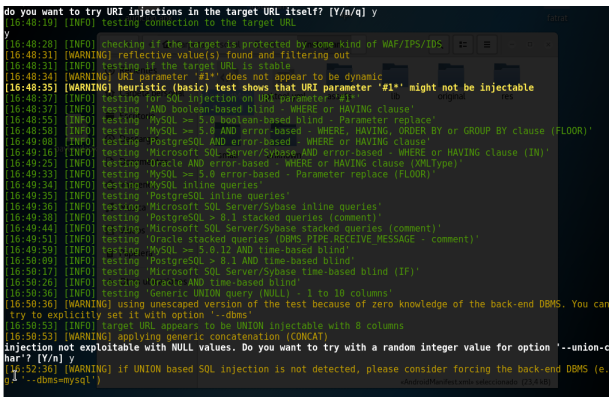
www.instagram.com



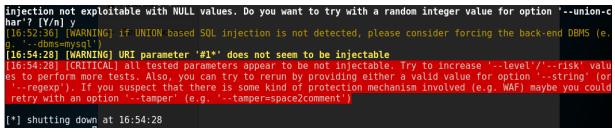
De las cuales se obtiene un listado perteneciente a los métodos navegables del mismo Host, lo que implica un número acotado de url's a analizar. El listado mencionado consta de: Android, explore, mainfeed, news, promote, profile, user, fb-friends y insights.

Por tanto al utilizar la herramienta **SQLMAP**, se generará una serie de inyecciones sql en las direcciones mencionadas, lo que implica una búsqueda de elementos o posibles aperturas dentro del sistema navegable. Al utilizar el siguiente comando:

sqlmap -u www.instagram.com/**Hosts**



En la figura anterior se da una vista de lo anteriormente mencionado, de lo que se obtiene finalmente:

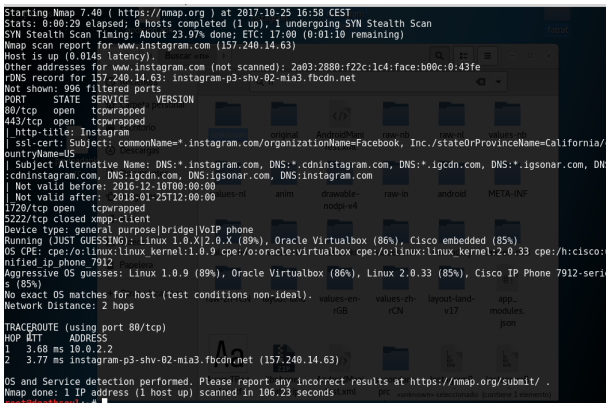


Es entonces que se infiere que no existe ningún resultado concluyente frente a las direcciones probadas, ya que si bien existen ciertas respuestas, sólo implican la respuesta de vista de usuario o imágenes obtenidas, pero no se encuentra ningún tipo de apertura a explotar dentro de la app.

- **Paso 6:** analizar los posibles puertos expuestos dentro de la aplicación. En la realización de dicha tarea se utilizará el mismo host encontrado en el paso anterior (www.instagram.com), la cuál por medio de la herramienta **NMAP**, se utilizará el siguiente comando:

  nmap -O www.instagram.com

  De lo que obtiene el siguiente resultado:

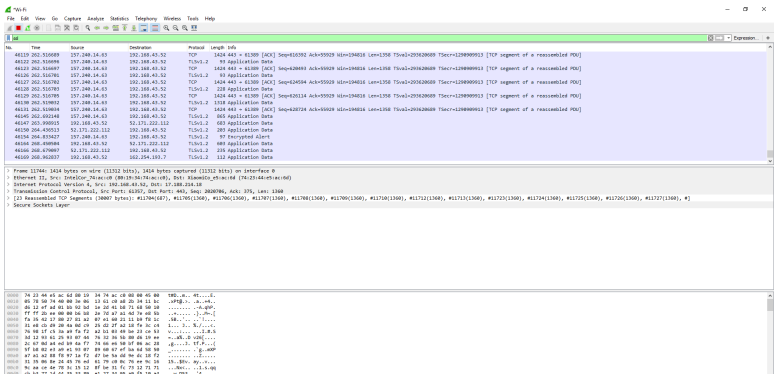


  Donde se encuentra sólo la apertura del puerto 80, perteneciente a flujos HTTP y 443 correspondiente a HTTPS, lo que significa que no existen puntos a explotar o bien puertos expuestos dentro de la aplicación.

- **Paso 7:** se realiza una prueba de man-in-the-middle, donde, con el fin de ver los paquetes captados se utiliza la herramienta **Wireshark** y el celular en versión móvil. Para lograr entender lo anterior mencionado se describe lo siguiente:

- Utilizar el computador personal, como fuente de internet para analizar los paquetes.
- Conectar el celular a la señal del computador.
- Cerrar toda aplicación, página u similar dentro del computador, con tal de captar sólo los paquetes del celular.
- Utilizar wireshark como captador de las querys realizadas dentro del instagram.

Ya realizados los pasos anteriores se obtiene el siguiente resultado:



Donde se encuentran todos los paquetes cifrados bajo el certificado ssl, afectando así, en la visualización de la información. Por tanto, se da a entender que no es posible encontrar data **AL MENOS CON WIRESHARK** proveniente de la aplicación.

2.3. Propuestas para Cuarto Incremento

Con el fin de lograr una mayor comprensión de lo propuesto se define el siguiente listado.

- Analizar en profundidad las funciones Internas React, depurando la información.
- Generar certificados SSL para el desarrollo de Man-in-the-Mi
- Buscar posibles Scripts utilizados para robar información.
- Analizar Servicios utilizados por la aplicación.
- Instalar la aplicación, con tal de analizar la BD interna generada por la misma.

Referencias

[1] A. Plaza, Ün fallo de seguridad en Instagram muestra la vulnerabilidad de las cuentas | Hipertextual", Hipertextual, 2017. [Online]. Available: https://hipertextual.com/2013/05/fallo-de-seguridad-en-instagram. [Accessed: 11- Oct- 2017].

[2] "La posibilidad de usar varias cuentas en Instagram presenta un fallo de seguridad - TreceBits", TreceBits, 2017. [Online]. Available: http://www.trecebits.com/2016/02/16/la-posibilidad-de-usar-varias-cuentas-en-instagram-presenta-un-fallo-de-seguridad/. [Accessed: 11- Oct- 2017].

[3] .El hackeo a Instagram, más grave de lo esperado: 6 millones de cuentas", ADSLZone, 2017. [Online]. Available: https://www.adslzone.net/2017/09/02/el-hackeo-instagram-mas-grave-de-lo-esperado-6-millones-de-cuentas/. [Accessed: 11- Oct- 2017].

[4] https://www.buzztouch.com/forum/thread.php?tid=B60C6EE41CD7D289048B0F4

[5] https://github.com/arashpayan/appirater

[6] https://iphoneros.com/44273/asi-es-el-modo-reachability-para-poder-utilizar-el-iphone-6-plus-con-una-sola-mano-video

[7] https://curl.haxx.se/docs/history.html