



UNIVERSIDAD  
**Finis Terrae**

VINCE IN BONO MALUM

## Proyecto Seguridad informática: Auditoria a Twitter

Jorge Gonzalez  
Profesor: Maximiliano Vega  
jgonzalezl@uft.edu  
Santiago, Chile.

29 de noviembre de 2017

# Índice

<b>1. Introducción</b>	<b>3</b>
1.1. ¿Que es Twitter? . . . . .	3
1.1.1. Historia de Twitter . . . . .	3
1.1.2. Tecnología usada . . . . .	3
1.2. tipos de ataques . . . . .	5
1.2.1. Medidas de protección . . . . .	5
1.2.2. Pruebas Previas . . . . .	5
1.3. Desencriptado de la aplicación . . . . .	6
1.4. Escaneo de puertos . . . . .	7
<b>2. que son las distribuciones basadas en Linux para auditorias</b>	<b>10</b>
2.1. Decompilación de la aplicación con apktool . . . . .	11
<b>3. Inyecciones SQL</b>	<b>13</b>
3.1. Que es SQLmap . . . . .	13
<b>4. modificación de la aplicación</b>	<b>16</b>
4.1. recompilacion de la aplicacion modificada . . . . .	16
<b>5. Conclusión</b>	<b>18</b>

# 1. Introducción

En la actualidad la gran mayoría de dispositivos portátiles utilizan sistemas operativos basado en Android, lo que hace que todas las empresas y desarrolladores independientes creen contenido para esta plataforma, por lo que la seguridad de estas es algo esencial en el día de hoy.

En este proyecto se desea poner a prueba algunas aplicaciones que utilizan funciones web y son conocidas en el rubro de las aplicaciones en Android; La aplicación a la cual se le hará la auditoria es Twitter, ya que esta aplicación como se verá es una de las plataformas web que ha tenido más casos de vulneraciones, por lo que se sabe que ha tenido que cambiar su seguridad en muchos casos, con esto en mente se pondrá a prueba muchas formas que sirven dentro de aplicaciones realizadas en Android, para dilucidar la seguridad de esta aplicación en lo que se refiere a seguridad.

## 1.1. ¿Que es Twitter?

### 1.1.1. Historia de Twitter

Twitter es un servicio gratuito de microblogging, que hace las veces de red social y que permite a sus usuarios enviar micro-entradas basadas en texto, denominadas "tweets", de una longitud máxima de 140 caracteres. El 21 de marzo del 2006, fecha del primer tweet lanzado por su fundador Jack Dorsey. En sus comienzos, la idea de Twitter surgió como proyecto de investigación dentro de Obvious, una pequeña compañía situada en San Francisco.



### Trending Topics

la red social amplio su ecosistema con los ya famosos "Trending Topics" o Temas del Momento. El 30 de abril del 2009, Twitter hacía oficial un cambio en su barra de búsquedas. Se trataba de fomentar aquello que originaba más "ruido", los temas que más se repetían entre el flujo de tweets veían como accedían a una categoría mayor, de manera que todos los usuarios podían reconocer o seguir los temas más candentes. Evidentemente y con el paso del tiempo, los Trending Topics han evolucionado, desde el año pasado se añaden también Temas del Momento promocionados.

### Hashtag

Otra de las claves de Twitter fue la inclusión de Hashtags, una etiqueta que bajo el símbolo de hash (#) seguida de una palabra o varias concatenadas, permitía realizar un seguimiento de temas a los usuarios.

### 1.1.2. Tecnología usada

Twitter está Escrita en Ruby on Rails, es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, la interfaz de Twitter es tremadamente

sencilla. Seguimos y nos siguen. Los usuarios que seguimos aparecen en el timeline y viceversa, lo que comúnmente se conoce en lenguaje “twitero” como followers y followings.

Todos los mensajes de Twitter van a parar al servidor de un software programado en Scala, además, la API está totalmente abierta a desarrolladores, motivo por el cual Twitter puede integrarse en todo tipo de webs, blogs o dispositivos móviles. Twitter confirmó en 2010 una de las actualizaciones más demandadas, incluyendo por fin la posibilidad de poder ver fotos, vídeos y contenido que llegaba de otros alojamientos. Además, desde la semana pasada la compañía añadió la opción que te permite utilizar siempre HTTPS cuando accedes a Twitter.com con el fin de mejorar la seguridad.

## 1.2. tipos de ataques

de los que se vieron online existen ataques que dicen funcionar como por ejemplo con exploits como los que se ven a continuacion:



Figura 1: ingreso a efinis con host modificado.

como se puede ver en su mayor parte se realizan acciones con exploits para saber que es un exploit se debe revisar la siguiente definicion: Las definiciones habituales hablan de un programa o código que se aprovecha de un agujero de seguridad (vulnerabilidad) en una aplicación o sistema, de forma que un atacante podría usarla en su beneficio.

### 1.2.1. Medidas de protección

- Mantener todas nuestras aplicaciones y sistemas actualizados: sabiendo que los exploits se aprovechan de los agujeros de seguridad, resulta vital cerrarlos cuanto antes. Por eso es necesario mantener una política de actualizaciones eficaz para evitar dejar una ventana de tiempo que pueda ser aprovechada por los atacantes.
- Mitigar los efectos de posibles exploits usados en nuestra contra. Puede suceder que el fabricante del sistema o aplicación vulnerable no haya lanzado todavía una actualización que solucione el problema. En este caso, se pueden utilizar herramientas como el Kit de herramientas de Experiencia de Mitigación mejorada (EMET) para Windows. Esto ayudará a evitar que tu sistema se infecte hasta que aparezca una solución definitiva.
- Contar con una solución de seguridad avanzada como ESET Smart Security, capaz de detectar y bloquear exploits pensados para aprovechar vulnerabilidades en navegadores web y lectores PDF, entre otros.

### 1.2.2. Pruebas Previas

Lo primero que se hizo fue probar con los supuestos exploits que se tenía en internet aquí un ejemplo del resultado:

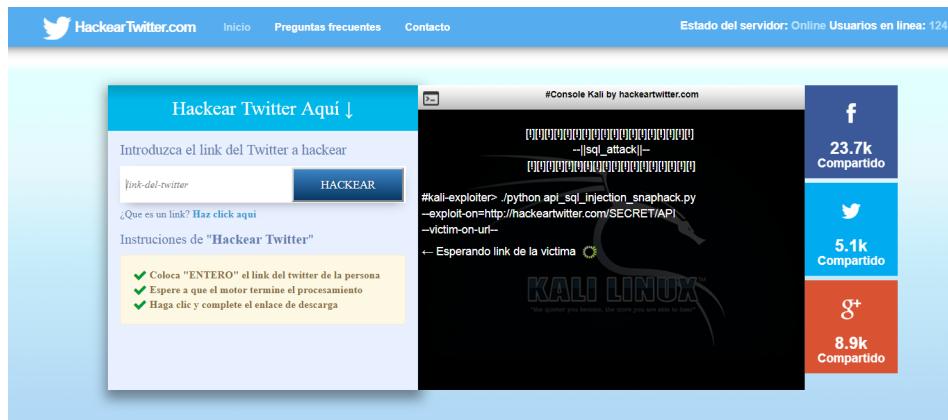


Figura 2:

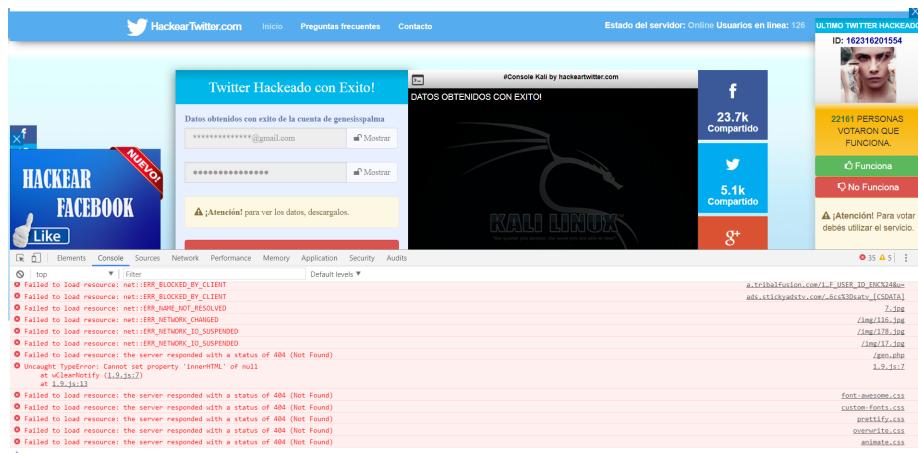


Figura 3:

Como es posible notar al inspeccionar la página se ve que la prueba solo era un “cazas bobos” ya que en el intento de conectarse con un socket solo lograba la falta de respuesta.

Luego de verificar esto se intentó ver a qué resultado se había llegado, pero para eso pedía un pago, lo que dejaba más en claro que solo se deseaba hacer perder el tiempo y dinero a quienes lo intentaban, luego de esto ya no se intentó más tratar de encontrar alguna falla con los similares a estos ya que se presumió que todos o la gran mayoría de ellos eran similares y serían perdidas de tiempo.

### 1.3. Desencriptado de la aplicación

Para poder desencriptar la aplicación se utilizó el sistema operativo lionsec y las herramientas APKtool en concreto lo que se muestra en la siguiente imagen:

Ya con la aplicación descargada se pasó a lionsec para poder descompilarla con las herramientas de este como se ve en la imagen:

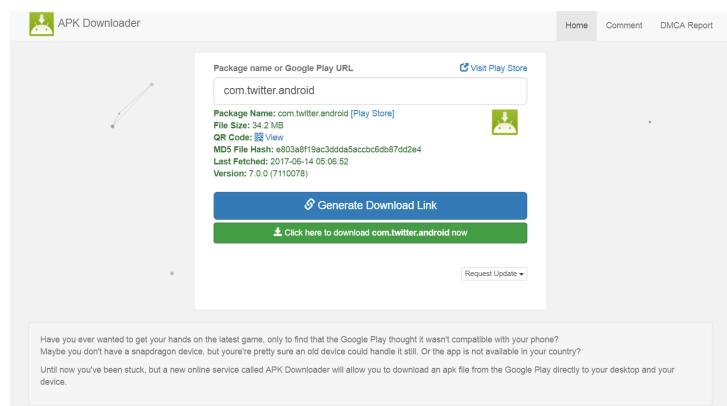


Figura 4:

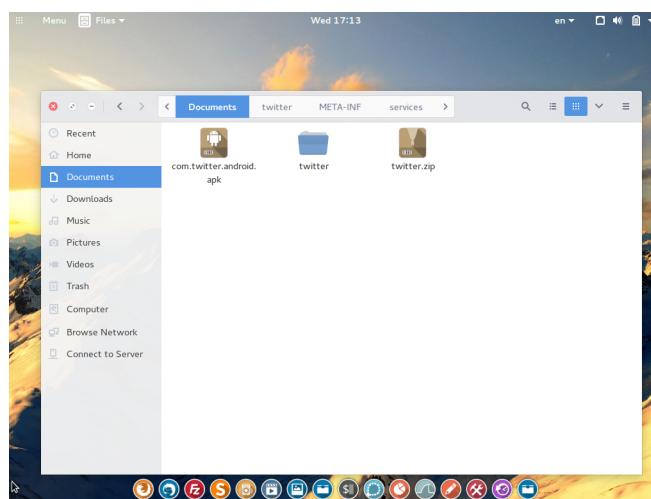


Figura 5:

Como se ve en la imagen después de descompilarla aplicación se creó una carpeta con algunos de los archivos que componen la aplicación para así lograr ver si se tiene alguna vulnerabilidad que se logre explotar. Como se muestra en la siguiente imagen lo único que se logró ver en la aplicación descompilada es el archivo META-INF el cual tiene el cifrado que se utiliza en la aplicación el cual es SHA-1 como se vio en el documento MANIFEST-MF que se ve a continuación:

#### 1.4. Escaneo de puertos

Después de esto se escanearon los puertos con nmap para ver si existía algún puerto descuidado o alguna otra falla lo cual solo se encontró los siguientes:

de esto solo se puede inferir que Los puertos abiertos son solo 2 lo que se ven son :

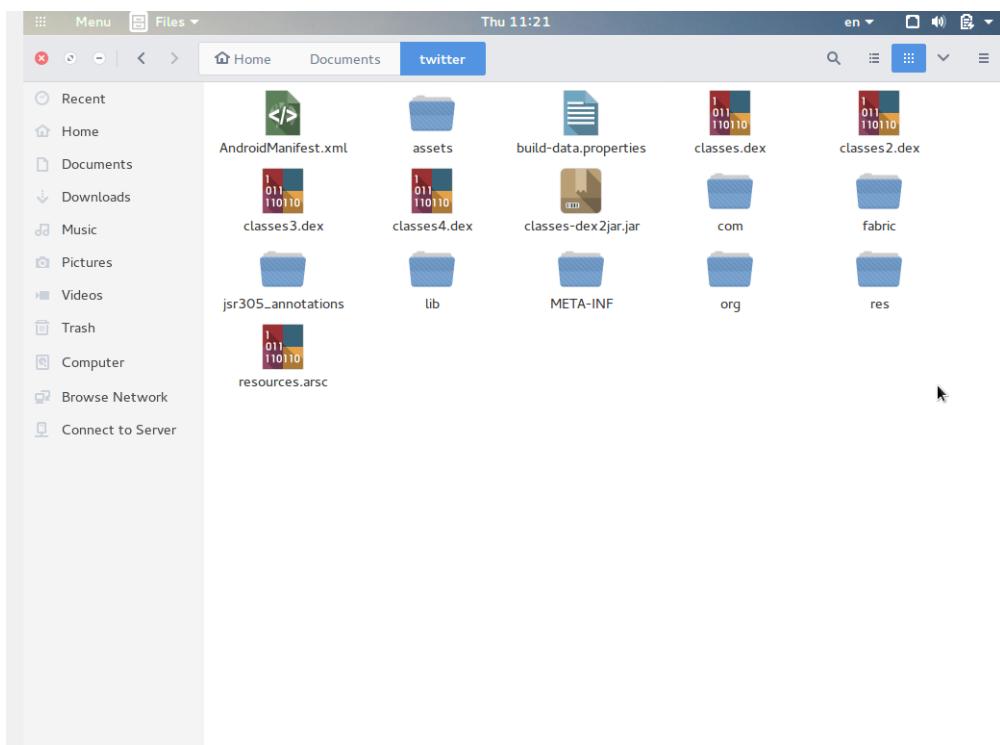


Figura 6:

Thu 22:14

MANIFEST.MF (~/Documents/twitter/META-INF) - gedit

```

File Edit View Search Tools Documents
  Open Save Undo Redo Cut Copy Paste Find Replace
  MANIFEST.MF x
Manifest-Version: 1.0
Built-By: Generated-by-ADT
Created-By: Android Gradle 2.3.0

Name: res/layout/totp_generator_ui.xml
SHA1-Digest: ETLCI65bgM0yG1NU8yqLnHNCnC

Name: res/drawable-anydpi-v21/vector_delete_icon.xml
SHA1-Digest: E+lKR1ADT2aN6eGU96gg1tVz0/I=

Name: res/layout/dm_rounded_conversation_suggestion_row_view.xml
SHA1-Digest: EL/MAVOXw2XhmlVh+hpSe5nide4=]

Name: res/layout/moments_guide_cell_item_thumbnail_icon.xml
SHA1-Digest: xT6se2B4A0zIMfk8oDOrtEG0iVC=

Name: res/layout/nativecards_consumerpoll_choices_row_moments.xml
SHA1-Digest: 4Imrr/b0D7xUbiz+0jJcfWRHfHs=

Name: res/drawable-hdpi-v4/ic_vector_medium_photo_stroke_tint.png
SHA1-Digest: 2wA85URwKaAvP2Ca3syHlMM/Hy8=

Name: res/drawable-xhdpi-v4/ic_tweet_monetize.png
SHA1-Digest: cmec9JUbeJ9rgp0Y7RJ3AOc4iIM=

Name: res/drawable-nodpi-v4/ic_compose_dm_102h.png
SHA1-Digest: kRFn8SCvaAkjhbaMxqUBJvvn9bM=

Name: res/drawable-nodpi-v4/ic_lightning_small_80h.png
SHA1-Digest: zGE+AcSLQFxrnA8sGRTRE15AoRg=

```

Plain Text Tab Width: 8 Ln 12, Col 42 INS

Figura 7:

```
lionsec@lionsec:~$ nmap -F twitter.com\
>

Starting Nmap 7.01 ( https://nmap.org ) at 2017-10-11 17:41 CLT
Stats: 0:00:03 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:04 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Stats: 0:00:05 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done

lionsec@lionsec:~$ nmap -F twitter.com

Starting Nmap 7.01 ( https://nmap.org ) at 2017-10-11 17:41 CLT
Nmap scan report for twitter.com (104.244.42.193)
Host is up (0.27s latency).
Other addresses for twitter.com (not scanned): 104.244.42.129
Not shown: 98 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 29.25 seconds
lionsec@lionsec:~$ m
```

Figura 8:

- Puerto 80
- Puerto 443

Por lo que se investigo el Puerto 80 es el puerto por default, por el medio del cual un servidor HTTP “escucha” la petición hecha por un cliente, es decir por una PC en específico Y el puerto 443 seria el que da el indicio de que las comunicaciones se realizan con certificados TLS lo que seria otro punto mas a investigar.

## 2. que son las distribuciones basadas en Linux para auditorias

las distribuciones de Linux enfocados a las auditorias son, como lo dice su nombre variaciones del sistema operativo Linux que cuantas con una gran cantidad de herramientas y utilidades que son de gran ayuda a la hora de probar funcionalidades y seguridad de aplicaciones, sistemas, páginas web, sistemas web, etc...

estas distribuciones y recopilación de herramientas para la seguridad informática se pueden distribuir en tres categorías: distribuciones para penesting, para auditorias de seguridad y distribuciones seguras.

Las distribuciones más conocidas son:

- Kali Linux
- Parrot Security OS
- Xiaopan OS
- WifiSlax
- BackBox
- Samurai Web Testing Framework
- DEFT Linux
- Caine
- Nst
- Santoku Linux
- BlackArch
- Bugtraq

Entre otros, es importante tenerlos en cuenta ya que en muchos casos estas distribuciones son esécializadas en su ámbito lo que las hace muy útiles en el momento que se necesite una distribución para los distintos casos.

En esta ocasión se utilizaron algunas de las herramientas de la distribución de Linux kali, esta distribución está destinada principalmente para la auditoria de aplicaciones en distintos ámbitos. Para probar algunos de los puntos importantes de la aplicación se utilizaron dos herramientas de este sistema: APKtool y SQLmap.

## 2.1. Decomplilación de la aplicación con apktool

con la aplicación ya en el sistema se procedió a descompilarla como se ve en la imagen:

```
File Edit View Search Terminal Help
.-advanced --advanced prints advance information.
.-version --version prints the version then exits
sage: apktool if!install-framework [options] <framework.apk>
-p,--frame-path <dir> Stores framework files into <dir>.
-t,--tag <tag> Tag frameworks using <tag>.
sage: apktool d[ecode] [options] <file apk>
-d,--decode-decompile Decompile the apk and generate Java code
-f,--force sqlmap good for forcing sqlmap to use a specific target URL
-o,--output <dir> low? The name of folder that gets written. Default is apk.out
-p,--frame-path <dir> Uses framework files located in <dir>. SELECTED BY SOME KIND OF NAF/IPS/ID
-r,--no-res Do not decode resources.
-s,--no-src Do not decode sources.
-t,--frame-tag <tag> Uses framework files tagged by <tag>. might be dynamic
sage: apktool b[uild] [options] <app path> (Basic) test shows that URI parameter '#1' might
-f,--force-all NOT BE Skip changes detection and build all files.
-o,--output <dir> The name of apk that gets written. Default is dist/name.apk
-p,--frame-path <dir> Uses framework files located in <dir>.

for additional info, see: http://ibotpeaches.github.io/Apktool/
or small/baksmali.info, see: https://github.com/SerifReke/small
oot@kali:~# apktool d /root/Escriptorio/twitter.apk #1 does not seem to be injectable
input file (/root/Escriptorio/twitter.apk) was not found or was not readable. injectable. Try to r
oot@kali:~# apktool d /root/Desktop/twitter.apk to perform more tests. Also, you can try to r
: Using Apktool 2.2.4-dirty on twitter.apk
: Loading resource table...
: Decoding AndroidManifest.xml with resources...
: Loading resource table from file: /root/.local/share/apktool/framework/1.apk
: Regular manifest package...down at 19:09:01
: Decoding file-resources...
: Decoding values */* XMLs...
: Baksmaling classes.dex...
: Baksmaling classes2.dex...
: Baksmaling classes3.dex...
: Baksmaling classes4.dex...
: Copying assets and libs...
: Copying unknown files...
: Copying original files...
oot@kali:~#
```

Figura 9:

con la aplicación y descompilada se procedió a ver que archivos tenía y si había algunos datos de importancia dentro de esta misma y el archivo más relevante que se encontró fue el archivo Android manifest que es el archivo de configuración de la aplicación está situado en la raíz de esta misma y controla todos los permisos necesarios para que la aplicación funcione.

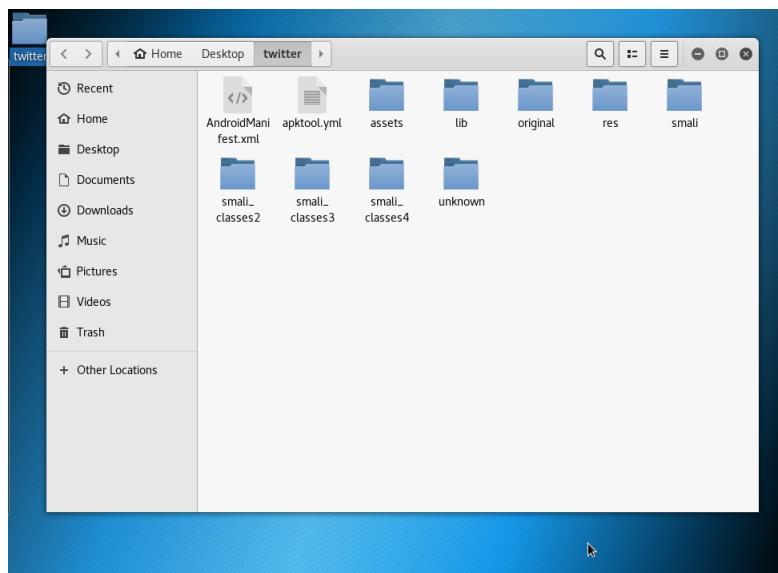
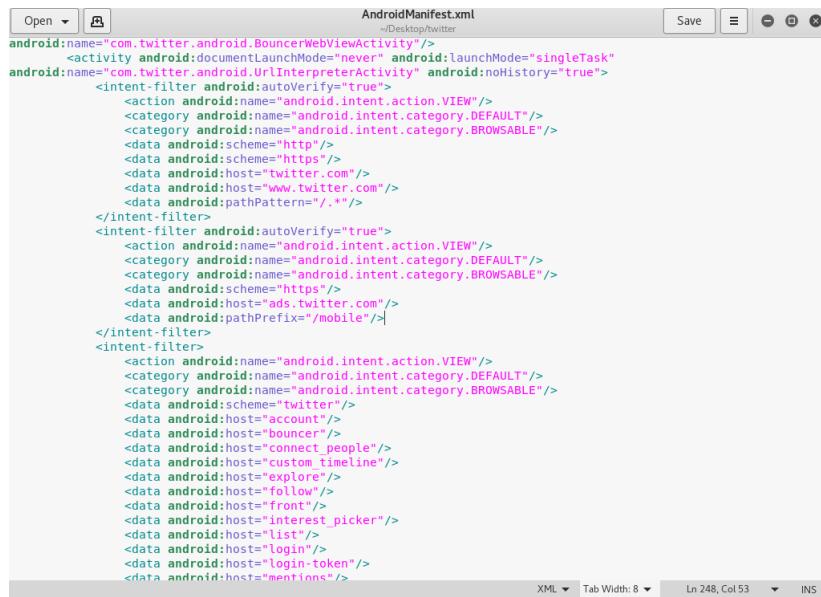


Figura 10:

En el Android manifest se encontraron todos los permisos de la aplicación, pero el dato más importante que se encontró fue el host de la aplicación y distintos tipos de tags con los cuales hacer consultas como se ve en la siguiente imagen:

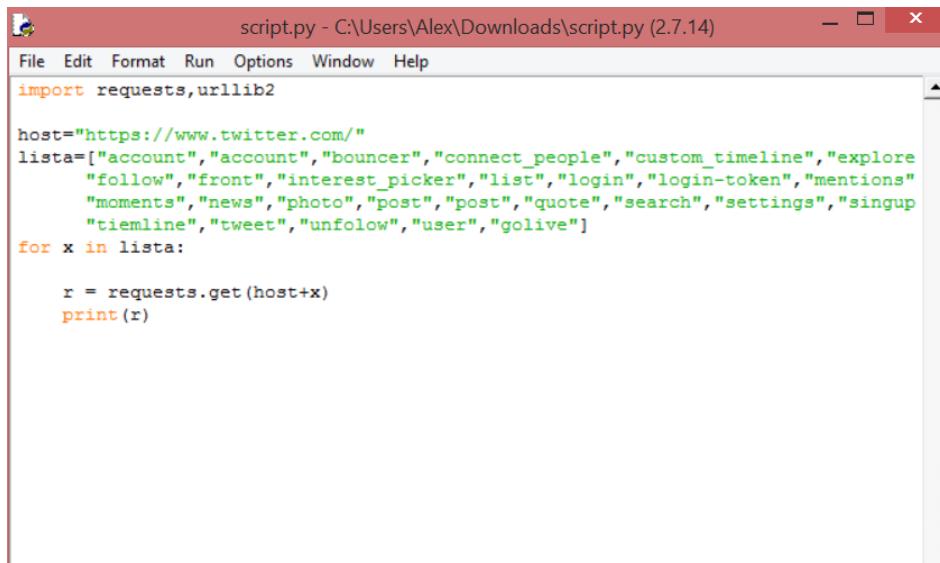


```

    Open [ ] AndroidManifest.xml ~Desktop\Twitter
    android:name=".com.twitter.android.BouncerWebViewActivity"/>
        <activity android:documentLaunchMode="never" android:launchMode="singleTask"
        android:name=".com.twitter.android.UrlInterpreterActivity" android:noHistory="true">
            <intent-filter android:autoVerify="true">
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:scheme="http"/>
                <data android:scheme="https"/>
                <data android:host="twitter.com"/>
                <data android:host="www.twitter.com"/>
                <data android:pathPattern="/.*/>
            </intent-filter>
            <intent-filter android:autoVerify="true">
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:scheme="https"/>
                <data android:host="ads.twitter.com"/>
                <data android:pathPrefix="/mobile"/>
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:scheme="twitter"/>
                <data android:host="account"/>
                <data android:host="bouncer"/>
                <data android:host="connect_people"/>
                <data android:host="custom_timeline"/>
                <data android:host="explore"/>
                <data android:host="follow"/>
                <data android:host="front"/>
                <data android:host="interest_picker"/>
                <data android:host="list"/>
                <data android:host="login"/>
                <data android:host="login-token"/>
                <data android:host="mentions"/>
            </intent-filter>
        
```

Figura 11:

Con estos datos se creo un script en Python junto con la librería request para generar una consulta sobre un URL tomando en consideración todos los urls que se obtuvieron dentro del manifest como se ve en la siguiente imagen:



```

script.py - C:\Users\Alex\Downloads\script.py (2.7.14)
File Edit Format Run Options Window Help
import requests,urllib2

host="https://www.twitter.com/"
lista=["account","account","bouncer","connect_people","custom_timeline","explore"
      "follow","front","interest_picker","list","login","login-token","mentions"
      "moments","news","photo","post","post","quote","search","settings","signup"
      "timeline","tweet","unfollow","user","golive"]
for x in lista:
    r = requests.get(host+x)
    print(r)

```

Figura 12:

Los resultados que se obtuvieron son respuestas 200 y 404 las cuales son OK para recibir la consulta y no se encuentra la pagina como se aprecia en la imagen:

```
===== RESTART: C:/Users/Alex/Downloads/script.py =====
<Response [404]>
<Response [404]>
<Response [200]>
<Response [200]>
<Response [404]>
<Response [200]>
<Response [404]>
<Response [200]>
<Response [200]>
<Response [404]>
<Response [200]>
<Response [200]>
<Response [404]>
<Response [200]>
<Response [200]>
<Response [404]>
<Response [404]>
<Response [200]>
<Response [200]>
<Response [200]>
<Response [404]>
<Response [200]>
<Response [200]>
<Response [200]>
<Response [404]>
<Response [200]>
>>>
```

Figura 13:

Luego de saber cuales eran las URLs con las que se podía trabajar se probaron los estas en SQLmap para lograr ver si se podía obtener algo, lo cual como se suponía tuvo resultados negativos.

### 3. Inyecciones SQL

dentro de lo que se hizo en este incremental fue encontrar variadas herramientas para lograr realizar otras pruebas necesarias para verificar la seguridad de la aplicación en cuestión.

#### 3.1. Que es SQLmap

SQLmap es una herramienta desarrollada en el lenguaje de programación Python la cual es muy útil para realizar pruebas automatizadas a bases de datos sobre la seguridad de las mismas. El objetivo de esta herramienta es detectar y utilizar vulnerabilidades de inyección sql en aplicaciones web, una vez detectado el host se pueden realizar pruebas de inyecciones sql y otras opciones de las cuales se pueden utilizar serian:

- Enumerar bases de datos
- Enumerar las tablas y columnas de una base de datos.
- Descifrar los hashes de contraseñas .

- Leer archivos específicos del sistema de archivo de un host específico

```

root@kali: ~
File Edit View Search Terminal Help
local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 22:07:12

[22:07:13] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] y
[22:07:15] [INFO] testing connection to the target URL
[22:07:35] [WARNING] turning off pre-connect mechanism because of connection reset(s)
[22:07:35] [CRITICAL] connection reset to the target URL. sqlmap is going to retry the request(s)
[22:07:35] [WARNING] if the problem persists please check that the provided target URL is valid. In case that it is, you can try to rerun with the switch '--random-agent' turned on and/or proxy switches ('--ignore-proxy', '--proxy',...)
[22:09:00] [CRITICAL] connection reset to the target URL

[*] shutting down at 22:09:00

root@kali:~# sqlmap -u www.twitter.com/bouncer

```

Figura 14:

```

kali [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Applications Places Terminal Wed 18:29
File Edit View Search Terminal Help
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 18:27:00

[18:27:01] [WARNING] you've provided target URL without any GET parameters (e.g. 'http://www.site.com/article.php?id=1') and without providing any POST parameters through option '--data'
do you want to try URI injections in the target URL itself? [Y/n/q] y
[18:27:04] [INFO] testing connection to the target URL
[18:27:31] [WARNING] turning off pre-connect mechanism because of connection reset(s)
[18:27:31] [CRITICAL] connection reset to the target URL. sqlmap is going to retry the request(s)
[18:27:31] [WARNING] if the problem persists please check that the provided target URL is valid. In case that it is, you can try to rerun with the switch '--random-agent' turned on and/or proxy switches ('--ignore-proxy', '--proxy',...)
[18:28:37] [CRITICAL] connection reset to the target URL

[*] shutting down at 18:28:37

root@kali:~#

```

Figura 15:

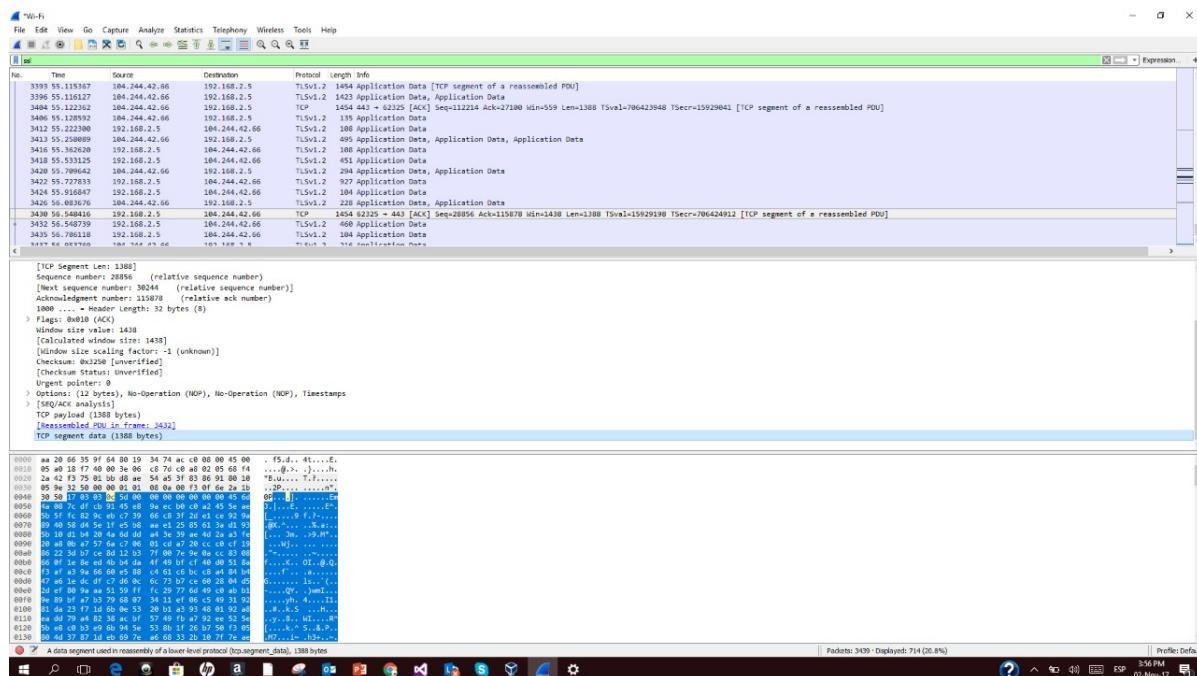


Figura 16:

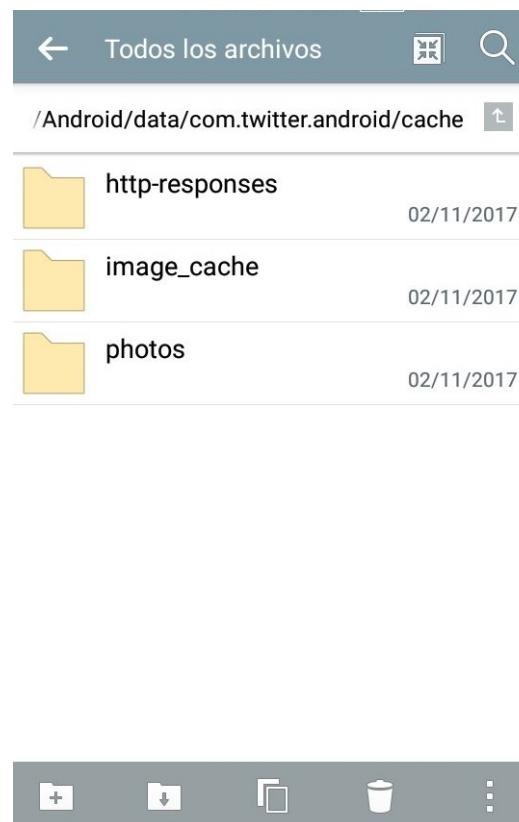


Figura 17:

## 4. modificación de la aplicación

en este incremental se realizaron dos acercamientos a encontrar alguna vulnerabilidad a la aplicacion twitter las cuales son las siguientes:

### 4.1. recompilacion de la aplicacion modificada

para lograr verificar si la aplicacion tiene la suficiente seguridad para evitar que se recompile si esta es modificada se realizo la siguiente prueba:

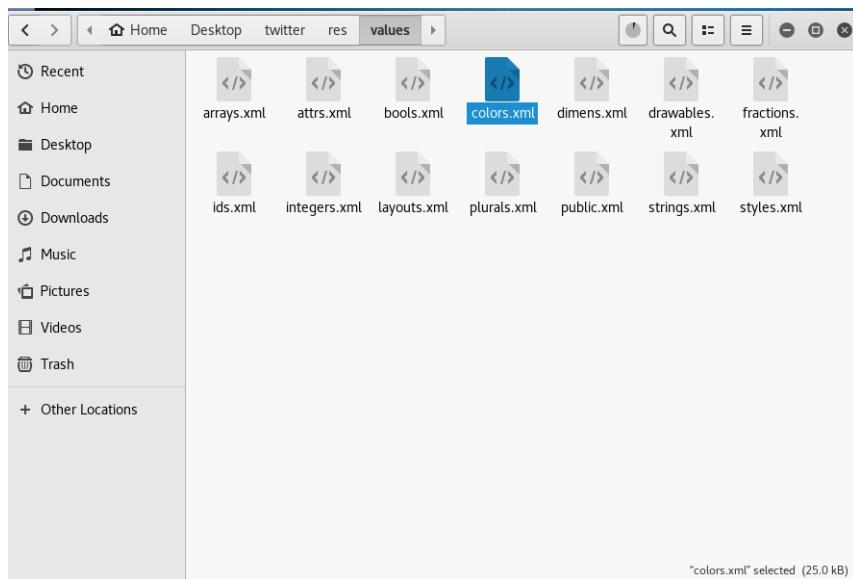


Figura 18:

se ingreso en la carpeta de resources donde se aloja el archivo de los colores de la aplicacion dentro de este se cambio el color de fondo de la aplicacion el que se cambio de blaco a negro como se ve en la siguiente imagen:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="app_background">@color/black</color>
    <color name="badge_protected">@color/text_black</color>
    <color name="badge_verified">@color/twitter_blue</color>
    <color name="border_color">@color/tertiary</color>
    <color name="deep_gray">#ff657786</color>
    <color name="deep_gray_30">#4d657786</color>
    <color name="deep_gray_50">#80657786</color>
    <color name="faded_gray">#fe6ecf0</color>
    <color name="faint_gray">#fff5f8fa</color>
    <color name="focused_bg">@color/faded_gray</color>
    <color name="footer_logo">@color/medium_gray</color>
    <color name="global_ripple_selector_color">#1d000000</color>
    <color name="light_gray">#ffcccd6dd</color>
    <color name="list_bg">@color/clear</color>
    <color name="medium_gray">#ffaab8c2</color>
    <color name="medium_gray_50">#80aab8c2</color>
    <color name="olive_holo_blue">@color/tertiary</color>
```

Figura 19:

luego de esto se realizo la recompilación de a aplicacion por lo que se utilizo la herramienta apktool con el siguiente resultado:

```

root@kali: ~/Desktop
File Edit View Search Terminal Help
Updated by Connor Tumbleson <connor.tumbleson@gmail.com>
twitter
usage: apktool
    -advanced,--advanced  prints advance information.
    -version,--version   prints the version then exits
usage: apktool if|install-framework [options] <framework.apk>
    -p,--frame-path <dir>  Stores framework files into <dir>.
    -t,--tag <tag>        Tag frameworks using <tag>.
usage: apktool d[ecode] [options] <file_apk>
    -f,--force            Force delete destination directory.
    -o,--output <dir>     The name of folder that gets written. Default is apk.out
    -p,--frame-path <dir>  Uses framework files located in <dir>.
    -r,--no-res           Do not decode resources.
    -s,--no-src            Do not decode sources.
    -t,--frame-tag <tag>  Uses framework files tagged by <tag>.
usage: apktool b[uild] [options] <app_path>
    -f,--force-all        Skip changes detection and build all files.
    -o,--output <dir>      The name of apk that gets written. Default is dist/name.apk
    -p,--frame-path <dir>  Uses framework files located in <dir>.

For additional info, see: http://ibotpeaches.github.io/Apktool/
For smali/baksmali.info, see: https://github.com/JesusFreke/smali
root@kali:~/Desktop# b twitter/
bash: b: command not found
root@kali:~/Desktop# apktool b twitter/
I: Using Apktool 2.2.4-dirty
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes4 folder into classes4.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes3 folder into classes3.dex...
I: Checking whether sources has changed...
I: Smaling smali_classes2 folder into classes2.dex...
I: Checking whether resources has changed...
I: Building resources...
W: warning: string 'ps_unban_progress dialog' has no default translation.
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:4: error: No resource identifier found for attribute 'shortcu

```

Figura 20:

```

root@kali: ~/Desktop
File Edit View Search Terminal Help
I: Checking whether resources has changed...
I: Building resources...
W: warning: string 'ps_unban_progress dialog' has no default translation.
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:4: error: No resource identifier found for attribute 'shortcu
tId' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:4: error: No resource identifier found for attribute 'shortcu
tShortLabel' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:4: error: No resource identifier found for attribute 'shortcu
tLongLabel' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:9: error: No resource identifier found for attribute 'shortcu
tId' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:9: error: No resource identifier found for attribute 'shortcu
tShortLabel' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:9: error: No resource identifier found for attribute 'shortcu
tLongLabel' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:14: error: No resource identifier found for attribute 'shortcu
tId' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:14: error: No resource identifier found for attribute 'shortcu
tShortLabel' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:14: error: No resource identifier found for attribute 'shortcu
tLongLabel' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:19: error: No resource identifier found for attribute 'shortcu
tId' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:19: error: No resource identifier found for attribute 'shortcu
tShortLabel' in package 'android'
W:
W: /root/Desktop/twitter/res/xml-v22/shortcuts.xml:19: error: No resource identifier found for attribute 'shortcu
tLongLabel' in package 'android'

```

Figura 21:

como se ve en las fotos el resultado de cambiar los datos en la aplicación solo fueron negativos ya que la aplicación cuenta con una defensa para cuando uno modifique esta para que no lleve a ser compilada ya que representaría un gran riesgo a la hora de distribuir versiones alteradas.

## 5. Conclusión

Por lo visto dentro de este informe y los incrementales se puede notar que Twitter en la parte de aplicación es lo suficientemente segura para lograr mover la cantidad de clientes que utilizan esta red social, ya que como se pudo ver en este proyecto se probaron distintos tipos de vulnerabilidades de las que se estudiaron en este proyecto se logró llegar a la culminación de que esta aplicación no tiene vulnerabilidades fáciles de atacar.

Ya en el pasado Twitter sufrió como muchas otras redes sociales ataques por parte de hackers organizados y en solitario lo que causo que esta misma multiplicara su seguridad ya que cada vez que esta era vulnerada se cambiaban las debilidades de estas mismas.

Por esto se concluye que la aplicación de Twitter es lo suficientemente segura para la cantidad de clientes que maneja ya que la única vulnerabilidad además de la fuerza bruta fueron los ataques DDos realizados por más de 500 computadores zombies haciendo que esta cantidad de computadores al mismo tiempo sea casi la única forma de vulnerar esta aplicación y no se descarta que se haya tomado este último punto y se hayan hechos los cambios haciendo que ni esta misma vulnerabilidad sea en la actualidad posible.

## Referencias

- [1] Hipertextual. (2017). Historia de Twitter. [online] Available at: <https://hipertextual.com/archivo/2011/03/historia-twitter/> [Accessed 28 Sep. 2017].
- [2] Cad.com.mx. (2017). Historia de Twitter. [online] Available at: [http://www.cad.com.mx/historia\\_de\\_twitter.htm](http://www.cad.com.mx/historia_de_twitter.htm) [Accessed 28 Sep. 2017].
- [3] Rubyonrails.org.es. (2017). Ruby on Rails - El desarrollo web que no molesta. [online] Available at: <http://rubyonrails.org.es/> [Accessed 28 Sep. 2017].
- [4] Como hackear una cuenta de Twitter gratis — Hackear Cuentas. [online] Hackearcuentas.com. Available at: <http://hackearcuentas.com/twitter/> [Accessed 28 Sep. 2017].
- [5] Albors, J. and Albors, J. (2017). ¿Sabes qué es un exploit y cómo funciona?. [online] WeLiveSecurity. Available at: <https://www.welivesecurity.com/la-es/2014/10/09/exploits-que-son-como-funcionan/> [Accessed 28 Sep. 2017].
- [6] <https://stackoverflow.com/questions/41216047/apktool-decompile-with-and-apktool/41216097>
- [7] <http://www.openexpo.es/mejores-distribuciones-linux-seguridad/>
- [8] <http://computerhoy.com/paso-a-paso/software/que-es-kali-linux-que-puedes-hacer-41671>
- [9] <http://www.tuprogramacion.com/glosario/que-es-el-android-manifest/>