

Seguridad en sistemas y servicios móviles

# 5

## Auditorías de aplicaciones móviles



## ÍNDICE

---

|                                                                         |           |
|-------------------------------------------------------------------------|-----------|
| <b>MOTIVACIÓN .....</b>                                                 | <b>3</b>  |
| <b>PROPÓSITOS .....</b>                                                 | <b>4</b>  |
| <b>PREPARACIÓN PARA LA UNIDAD .....</b>                                 | <b>5</b>  |
| <b>1. AUDITORIAS DE APLICACIONES IPHONE, ANDROID Y BLACKBERRY .....</b> | <b>7</b>  |
| <b>2. ANDROID .....</b>                                                 | <b>14</b> |
| <b>3. IPHONE .....</b>                                                  | <b>18</b> |
| <b>4. BLACKBERRY .....</b>                                              | <b>23</b> |
| <b>CONCLUSIONES .....</b>                                               | <b>29</b> |
| <b>RECAPITULACIÓN .....</b>                                             | <b>30</b> |
| <b>AUTOCOMPROBACIÓN .....</b>                                           | <b>31</b> |
| <b>SOLUCIONARIO .....</b>                                               | <b>35</b> |
| <b>PROPUESTAS DE AMPLIACIÓN .....</b>                                   | <b>36</b> |
| <b>BIBLIOGRAFÍA .....</b>                                               | <b>38</b> |



## MOTIVACIÓN

---

Cada vez más accedemos a Internet desde dispositivos móviles, para lo que tener una aplicación, o APP, nos permitirá interconectarnos con el resto del mundo. Es por esto que una vez hayamos desarrollado una aplicación de iPhone, Blackberry o Android, deberemos proceder a realizar una auditoría sobre la misma para poder tener garantías de seguridad sobre la misma, no vaya a ser que el tener una aplicación para un Smartphone, con el que nos proponíamos abrir otro canal de comunicación y venta con nuestros clientes, se ha transformado en el caballo de Troya de una mafia.

En este tema se va a tratar de mostrar los rudimentos que nos van a permitir aprender cómo gestionar y controlar el acceso de dispositivos móviles, a la información confidencial de la empresa en la red corporativa, existente en los smartphones, para evitar fugas de información y poder crear APP seguras.

Y es que prevenir la fuga de información es una de las principales problemáticas que hoy enfrentan las organizaciones en materia de seguridad. Las fugas de información se generan principalmente en tres ámbitos, siendo uno de éstos por medio de una amenaza o código malicioso que se introduce en la organización y roba datos confidenciales. Por ejemplo, un troyano que modifica el gestor de correo de un terminal móvil, extrayendo información del usuario, o un virus que saca datos de la compañía almacenados en un terminal.

## PROPÓSITOS

---

Al finalizar el estudio de esta unidad deberías ser capaz de poder explicar las siguientes cuestiones:

Según ISACA (Asociación de Auditoría y Control de Sistemas de Información), “la Auditoría de Sistemas de Información es cualquier revisión y evaluación de todos los aspectos (o cualquier porción de ellos) de los sistemas automáticos de procesamiento de la información, incluidos los procedimientos no automáticos relacionados con ellos y los interfaces correspondientes”. Es por tanto vital conocer las técnicas para realizar una correcta auditoría informática de aplicaciones móviles, para poder cumplir con la seguridad de la organización.

Sin embargo, según muestra un estudio de White Hat Security las vulnerabilidades, independientemente del sector que cree la aplicación (Bancos, Educación, etcétera) sigue aumentando. Por tanto, ante la pregunta evidente de por qué no se corrigen estos problemas, la respuesta es sencilla pero alarmante: Faltan responsables para mantener el código y evitar vulnerabilidades; es más: los desarrolladores no saben ni entienden sobre “vulnerabilidades”, lo cual resulta más peligroso aún.

Si quieres conocer más sobre cómo se auditan las APP en los tres principales entornos móviles, te invitamos a proseguir con este tema. Si estás interesado en este tipo de temas, no dudes en seguir leyendo este tema para poder conocer cómo plantear el análisis de aplicaciones en dispositivos móviles.

## PREPARACIÓN PARA LA UNIDAD

---

En esta unidad vamos a tratar los siguientes temas:

1. En este tema veremos cómo plantear auditorías de aplicaciones iPhone, Android, Blackberry, así como veremos el proceso de auditorías de aplicaciones móviles y las pautas que se deben seguir.
2. Para sistemas ANDROID auditaremos el código fuente desensamblándolo de una aplicación en concreto, y viendo cómo se pueden visualizar datos muy interesantes
3. Para sistemas IPHONE, se auditan varias aplicaciones legítimas configurando en el iPhone usando el proxy interceptor Burp, para poder analizar las peticiones.
4. Para sistemas BLACKBERRY, se auditarán sus aplicaciones usando el emulador de Blackberry.





# 1. AUDITORIAS DE APLICACIONES IPHONE, ANDROID Y BLACKBERRY

Cuando hablamos de auditoría en móviles hemos de distinguir dos niveles, la auditoría del terminal y la de aplicación. En el primer caso, tal y como indica Informática64, los terminales Smartphone son sistemas operativos completos que pueden dar mucha información del uso que se le ha dado durante un periodo de tiempo. Con la auditoría de un terminal móvil los profesionales en análisis forense pueden extraer todos los datos almacenados en un Smartphone, y así trazar una línea temporal de la vida y uso que ha tenido dicho dispositivo.

Esta actividad permite conocer si alguien ha realizado un uso indebido de un terminal, y generar información relevante, que permita realizar el informe técnico y pericial, de cara a realizar acciones legales y a presentar un dictamen forense de calidad. Lo anterior puede interesar a:

- Hoy día, muchas de las grandes y pequeñas empresas, ya han integrado las plataformas móviles a nivel corporativo, y es fácil que se produzca un robo, una fuga de información, o un uso indebido.
- La lucha contra el crimen que libran las Fuerzas de Seguridad del Estado, cada día está más vinculada a la tecnología. Con esta actividad se pueden realizar análisis forense mediante herramientas de pago y otras gratuitas, de cara a realizar los informes necesarios a la hora de judicializar los casos.

Los analistas forenses pueden realizar este trabajo con terminales de las plataformas o sistemas operativos móviles más extendidos en el mercado, como son iOS (iPhone, iPad), Android, Windows Mobile, Symbian o BlackBerry.

En el segundo caso, en el proceso de auditorías de aplicaciones móviles se van a tratar de encontrar evidencias de un comportamiento anómalo por lo que para obtener esas evidencias se deben seguir una serie de pautas:

- a) Analizar el código de la aplicación si se tiene acceso, o en caso contrario intentar analizar en la medida de lo posible el código de la aplicación una vez desensamblado.
- b) Analizar los ficheros que lee y sobre todo crea la aplicación en el teléfono una vez es instalada y una vez es ejecutada. En estos ficheros se puede encontrar información sensible, como usuarios y contraseñas en texto claro.
- c) Analizar las comunicaciones de la aplicación. Este es el punto más importante, ya que mediante esta fase podremos intervenir las comunicaciones mediante ataques man in the middle y será posible manipular las comunicaciones y buscar vulnerabilidades de todo tipo.

Recordad que, como ya hemos visto para llevar a cabo auditorías sobre las aplicaciones para los dispositivos móviles de una forma homogénea y repetible, hace falta un estándar o metodología que establezca los pasos a seguir. A este respecto, OWASP está trabajando en una guía de Seguridad que ayude a estandarizar y homogeneizar las pruebas que se realizan para el análisis de seguridad en las aplicaciones móviles.

A continuación se detallarán a modo de ejemplo auditorías de aplicaciones móviles para una mejor comprensión del proceso. Se irán mostrando evidencias de auditorías reales en las que se detallan los fallos de seguridad encontrados.

## 1.1. ENTORNO DE TRABAJO.

Existen dos formas posibles de llevar a cabo una auditoría de una aplicación, bien utilizando un emulador o un dispositivo físico. Cada una de ellas, presenta sus ventajas y desventajas.

Por ejemplo, alguno de los inconvenientes que presenta el uso de emuladores es el hecho de que algunas aplicaciones comprueban el IMEI/MSISDN/IMSI del dispositivo, y si detecta que está siendo emulado automáticamente cesará la actividad de la aplicación. No obstante esto es posible solventarlo realizando algunas modificaciones a nivel de código, pero eso es algo que está fuera del ámbito de esta entrada.

Para la primera manera partimos utilizando un dispositivo físico y del SDK de Android instalado adecuadamente en nuestro equipo. Hay que tener en cuenta que en este caso es necesario tener el dispositivo rooteado (lograr esto depende de la versión que tenga instalada el terminal).



Un método rápido consiste en flashear el teléfono a la versión 2.3.6 descargando la imagen acorde al vuestro dispositivo (<http://shipped-roms.com/index.php?category=android>) y posteriormente utilizar el exploit zergRush ([dl-1.va.us.xda-developers.com/8/4/1/8/7/6/DooMLoRD\\_v4\\_ROOT-](http://dl-1.va.us.xda-developers.com/8/4/1/8/7/6/DooMLoRD_v4_ROOT-)

```
zergRush-busybox-  
su.zip?key=S5HLVyxRt_ucGabGSbnwA&ts=1450303811).
```

La segunda manera consiste en hacer las pruebas en un emulador, para lo cual se debe de instalar el SDK adecuadamente.

Una vez realizados los pasos anteriores, el siguiente objetivo consiste en poder interceptar las comunicaciones que son enviadas a través de HTTP por las aplicaciones o las páginas web que son visitadas. Utilizaremos como proxy la aplicación Burp.

- Si se utiliza un dispositivo físico, bastará con descargar una aplicación que ejerza de proxy, como por ejemplo ProxyDroid.
- Si por el contrario estamos bajo el emulador, podemos utilizar algunas de las opciones que por defecto la utilidad **emulator** (incluida en el SDK) dispone: `-http-proxy` | `-dns-server` | `-debug-proxy` | `-tcpdump`. Así pues, podríamos levantar el emulador utilizando la siguiente línea desde la terminal:

```
./emulator @device_name -http-proxy ip_proxy:proxy_port -debug-proxy
```

Independientemente del método elegido, será necesario tener ejecutando una instancia de Burp por detrás para que vaya interceptando todo el tráfico generado.

## 1.2. CONFIGURAR SDK Y BURP SUITE.



Burp Suite es una plataforma con herramientas de seguridad multipropósito, profesionalmente se utiliza para realizar auditorías de seguridad de aplicaciones web. Por tanto BURP es un proxy para interceptar comunicaciones, en este caso desde el emulador ANDROID hacia fuera (webs), que permite en la práctica ser un man-in-the-middle entre el browser y el emulador. Más información en <http://portswigger.net/burp/proxy.html>

Nuestro entorno para la auditoría de las aplicaciones estará compuesto por el emulador proporcionado por el SDK de Android, el proxy burp que nos pondrá en el medio de las comunicaciones entre el emulador y destino y la posibilidad de analizar el tráfico con wireshark.

El procedimiento es bastante sencillo: ejecutamos Burp Suite y configuramos en nuestro navegador el proxy para que la navegación sea redirigida a localhost (127.0.0.1) por el puerto 8080 (por defecto). Los pasos previos a realizar son:

1. Descargar de la web oficial Java <http://java.com/es/download/index.jsp>
2. Descargar Burp Suite Free Edition: <http://portswigger.net/burp/downloadfree.html>
3. Instalar Java y ejecutar burp\_free\_v.1.6.jar
4. Configurar el navegador con el proxy (Conexiones -> Configuración LAN -> Configurar servidor proxy: Dirección: 127.0.0.1 / Puerto: 8080)
5. En Burp Suite, apartado "Proxy" y la sección "Options", añadir las interfaces y puertos que estarán a la escucha (por defecto 127.0.0.1:8080)

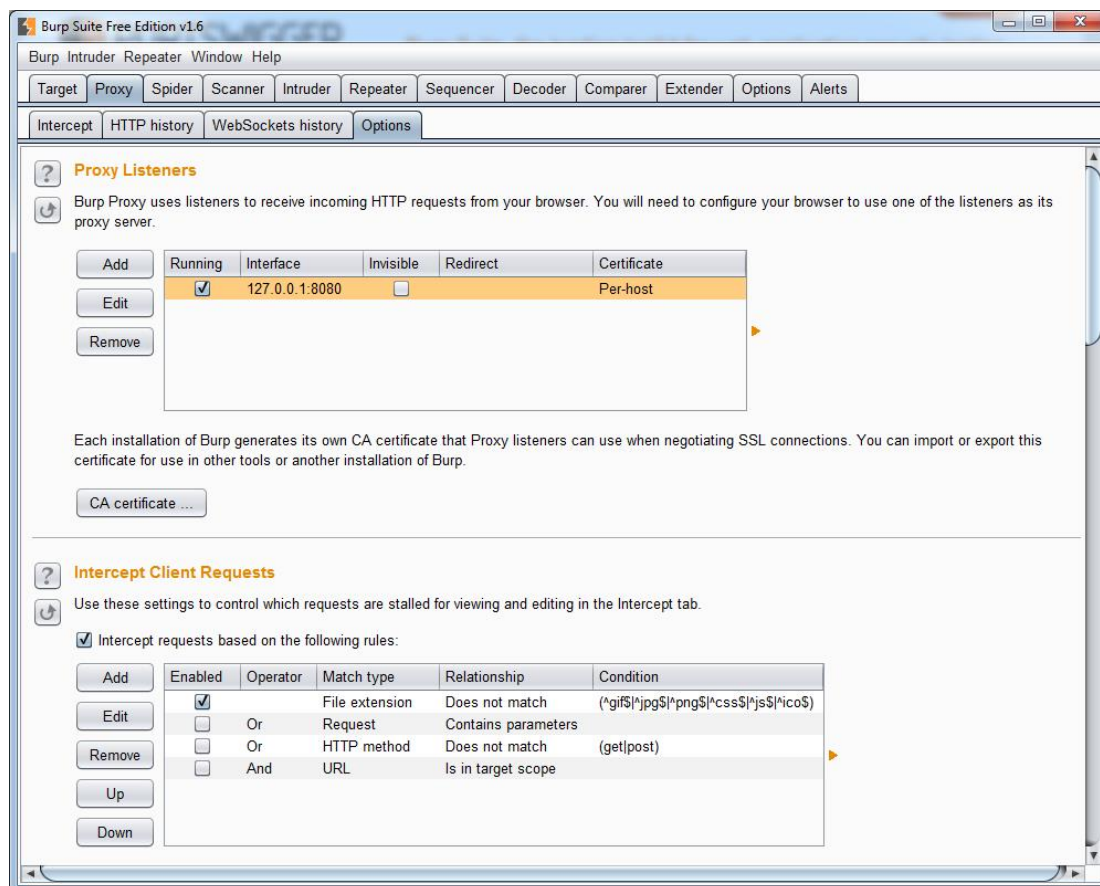


Figura 1: Fuente: Elaboración Propia

6. Dentro del apartado "Proxy" y en la sección "Intercept", Activa "Intercept is on" para comenzar a interceptar el tráfico del navegador en el que hayamos configurado el proxy.
7. Una vez realizamos este paso lanzamos nuestro emulador con la configuración siguiente para que las peticiones se hagan a través del proxy que arranquemos previamente.

### Using the Emulator with a Proxy

If your emulator must access the Internet through a proxy server, you can use the `-http-proxy <proxy>` option when starting the emulator, to set up the appropriate redirection. In this case, you specify proxy information in `<proxy>` in one of these formats:

```
http://<machineName>:<port>
```

or

```
http://<username>:<password>@<machineName>:<port>
```

The `-http-proxy` option forces the emulator to use the specified HTTP/HTTPS proxy for all outgoing TCP connections. Redirection for UDP is not currently supported.

Alternatively, you can define the environment variable `http_proxy` to the value you want to use for `<proxy>`. In this case, you do not need to specify a value for `<proxy>` in the `-http-proxy` command – the emulator checks the value of the `http_proxy` environment variable at startup and uses its value automatically, if defined.

You can use the `-verbose-proxy` option to diagnose proxy connection problems.

Figura 2: Fuente: Elaboración Propia

8. Lanzar el emulador con el proxy activado: `./emulator @device_name -http-proxy ip_proxy:proxy_port -debug-proxy`

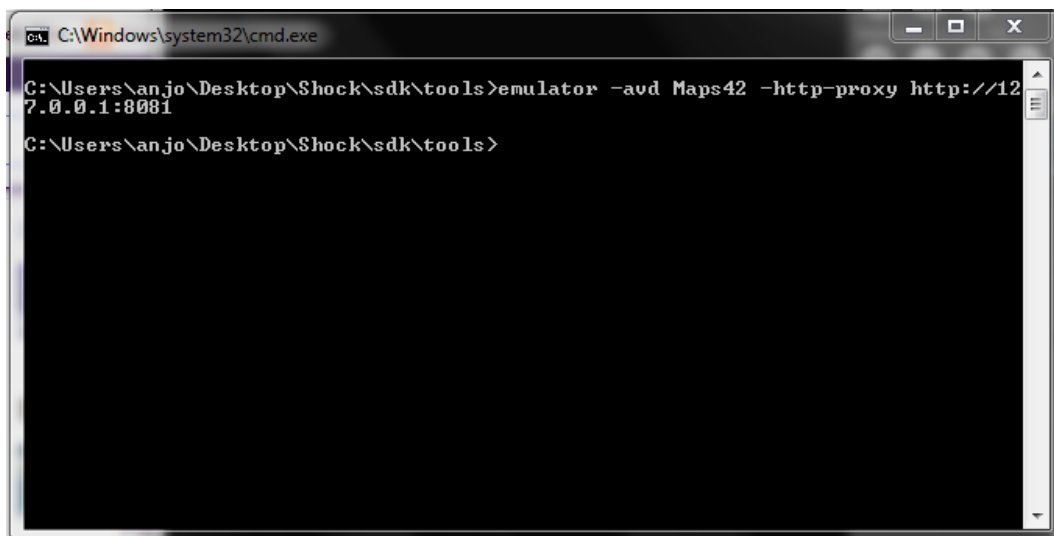


Figura 3: Fuente: Elaboración Propia

9. Una vez que ya tenemos nuestro emulador si realizamos cualquier petición con un navegador vemos que el proxy burp, si está activada la interceptación, se detendrá ante cada intento de comunicación.
10. Ahora el problema es el siguiente: si queremos interceptar peticiones https deberemos configurar algo más. Desde la herramienta burp podemos exportar un certificado de confianza para estas comunicaciones, lo haremos como en la siguiente captura:

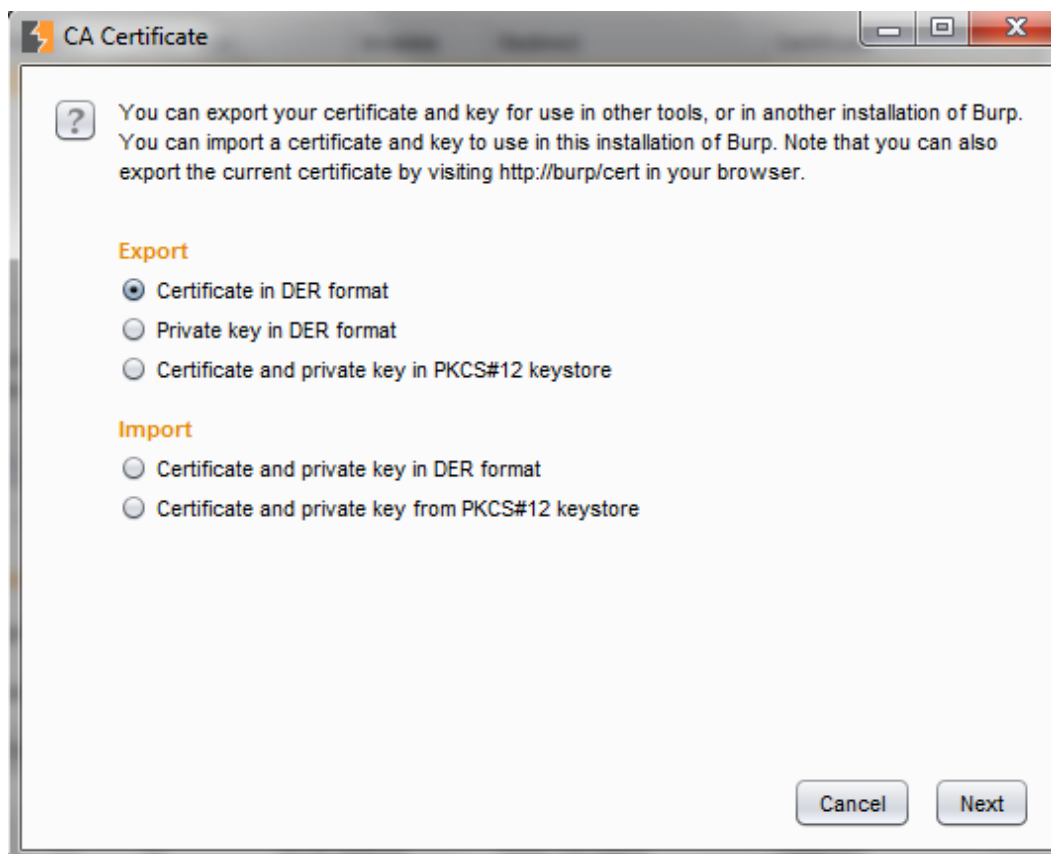


Figura 4: Fuente: Elaboración Propia

11. Una vez generado lo guardamos consultamos el hash y lo tendremos que pasar al emulador.

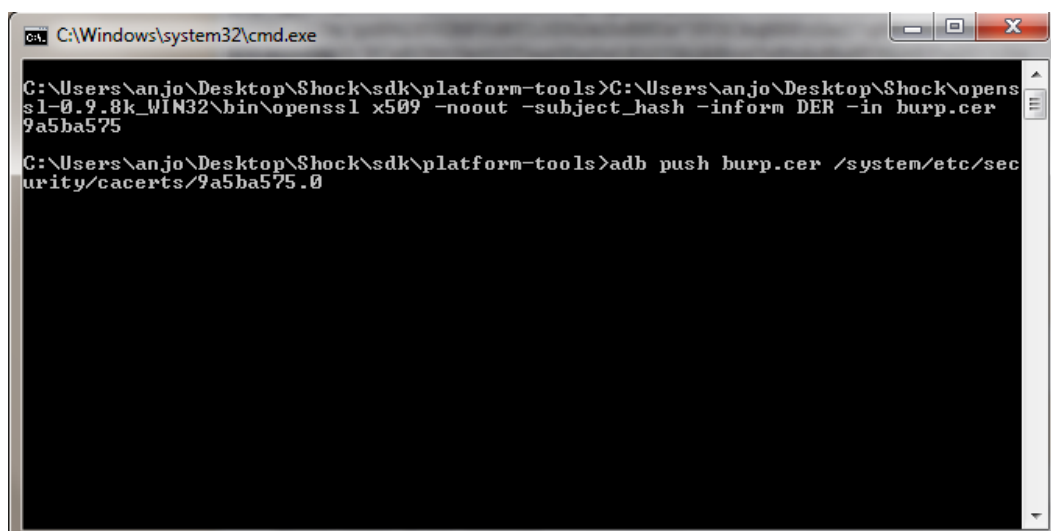


Figura 5: Fuente: Elaboración Propia

12. Una vez guardado en el emulador comprobamos dentro en Ajustes>Seguridad>Certificados

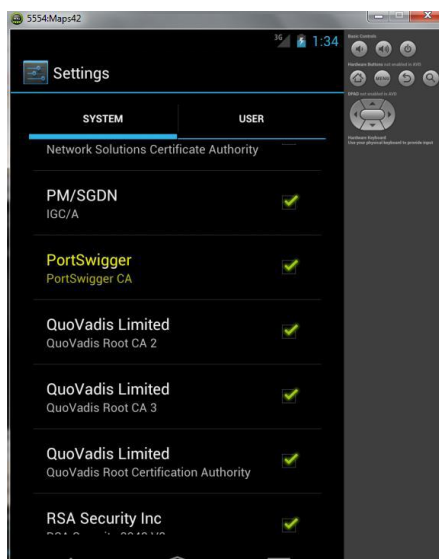


Figura 6: Fuente: Elaboración Propia

13. Una vez tenemos configurado este entorno pasamos a analizar una aplicación del market

## 2. ANDROID



Auditando el código fuente desensamblado de una aplicación en concreto se pueden visualizar datos muy interesantes, como rutas del servidor web que utiliza la aplicación, ficheros sensibles incluso en algunos casos usuarios y contraseñas válidos en la aplicación.

En la siguiente evidencia se puede comprobar cómo se pueden ver rutas http en el código de la aplicación:

```
hippo_sample_dex2jar.jar
- com.ku6.android.videobrowser
  - download
    + Download
    + HttpClients
    + Interfaces
    + InternetConnection
  - entity
  - network
    + AsyncImageLoader
    + ChannelHandler
    + ChargeHandler
    + Constants
    + FileHandler
    + HelperFactory
    + HotKeywordHandler
    + NetEventReceive
    + ServerStub
    + Utils
    + VersionUpdateHandler
    + VideoHandler
  - sms
    + BootReceiver
    + CallContentObserver
    + MessageService
    + About_Activity
    + Base_Activity
    + ChannelDetailAdapter
    + ChannelListViewAdapter

InternetConnection.class  Message.class  ServerStub.class
{
    HashMap localHashMap = new HashMap();
    localHashMap.put("method", "getrootcategory");
    String str = buildURL(localHashMap, "http://info.ku6.cn/clientRequest.htm");
    Log.i(getClass().toString(), str);
    ArrayList localArrayList;
    try
    {
        ChannelHandler localChannelHandler = (ChannelHandler)XMLParse(str, new ChannelHandler()
        {
            if (localChannelHandler != null)
            {
                localArrayList = localChannelHandler.channels;
            }
            else
            {
                localArrayList = new ChannelHandler().channels;
            }
        });
    }
    catch (Exception localException)
    {
        localException.printStackTrace();
        localArrayList = null;
    }
    return localArrayList;
}

public Charge retrCharge(Context paramContext, String paramString)
{
    String str = "http://info.ku6.cn/clientRequest.htm?method=startCharge&ct=android&chann";
    Log.d("HttpURL:", str);
    try
    {
    }
}
```

Figura 7: Fuente: Elaboración Propia



En la siguiente evidencia se comprueba cómo es posible obtener usuarios y contraseñas validos en el código fuente de la aplicación:

```
!#Enviando comando al dispositivo
Flecha
Dispositivo apagado
Barra de consumo
Dispositivo encendido
66Error al obtener los datos de la cabecera (HeaderData)
$$Error al conectar con el dispositivo
44Error al obtener los datos de la cabecera (HomeData)
    Siguiente
++Error al obtener el estado del PlugComputer
Anterior
..Error al obtener el estado de los dispositivos
&&Error al obtener el estado de la pinza
Icono de consejo
12Error al conectar con el servicio de meteorolog
CONSEJOS PARA AHORRAR
mil@.es Usuario
00000000 Password
Control Power Pack
Barra ControlPowerPack
Weather
Previous
Error in (HomeData)
Next
Error in (HeaderData)
Error connecting to the device
Consumption
```

Figura 8: Fuente: Elaboración Propia

A continuación se puede observar como en la siguiente evidencia es posible ver que ficheros genera la aplicación en el terminal.



En uno de los ficheros se puede observar cómo se guarda en texto claro el usuario y contraseña, lo que supone un grave fallo de seguridad.

```
# pwd
pwd
/data/data/[redacted]/shared_prefs
# ls -l
ls -l
-rw-rw---- app_39 app_39 112 2012-11-07 23:38 DDBB.xml
-rw-rw---- app_39 app_39 149 2012-11-07 23:38 SERVER_LIST.xml
# cat SERVER_LIST.xml
cat SERVER_LIST.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="user">mil@[redacted]</string>
<string name="user2">00000000</string>
</map>
#
```

Figura 9: Fuente: Elaboración Propia

En la siguiente evidencia se comprueba cómo se interceptan las comunicaciones de aplicaciones. En este caso es una aplicación funcionando en un terminal Android emulado y utilizando un proxy interceptor burp.

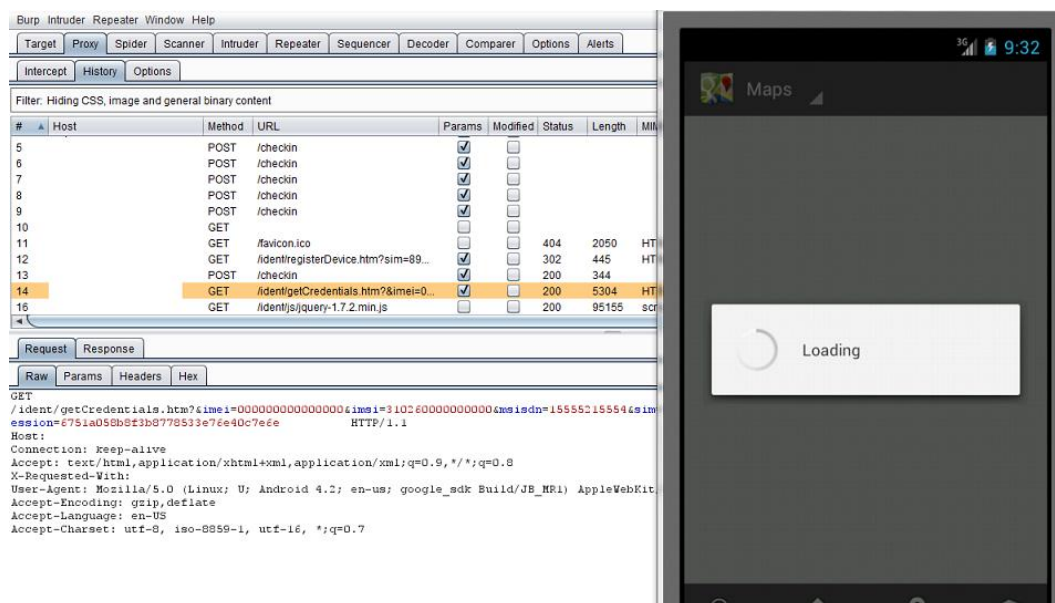


Figura 10: Fuente: Elaboración Propia

Tal y como se puede ver en la evidencia, se pueden interceptar tanto las comunicaciones http como https y manipularlas para poder buscar vulnerabilidades. Esta fase suele ser la más crítica ya que es donde se suele encontrar un mayor número de vulnerabilidades.

En la siguiente evidencia se puede observar cómo se manipulan las comunicaciones de una aplicación que utiliza el envío de SMS de un servidor de UK para validar el registro.



Interceptando estas comunicaciones es posible enviar el SMS de forma automatizada a cualquier número sin realizar ninguna comprobación, se puede automatizar este ataque con burp para enviar un número indeterminado de SMS para realizar ataques SMS bombing a cualquier teléfono y para generar costes de envío de SMS a la empresa desarrolladora de la aplicación.

Aquí se puede observar cómo se envía mediante protocolo SOAP la petición al servidor de SMS.

[illegible]

**Figura 11:** Fuente: Elaboración Propia

En la siguiente evidencia se observa cómo se automatiza el ataque para enviar 20 SMS a un determinado número.



Con un envío simultáneo de más de 200 SMS muchos móviles llegan a quedarse totalmente bloqueados.

| Request # | Payload | Status | Size                     | Timeout                  | Length | Comment |
|-----------|---------|--------|--------------------------|--------------------------|--------|---------|
| 9         |         | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 10        | 10      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 11        | 11      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 12        | 12      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 13        | 13      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 14        | 14      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 15        | 15      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 16        | 16      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 17        | 17      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 18        | 18      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 19        | 19      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |
| 20        | 20      | 200    | <input type="checkbox"/> | <input type="checkbox"/> | 640    |         |

RequestResponse

Raw

Headers

Hex

XML

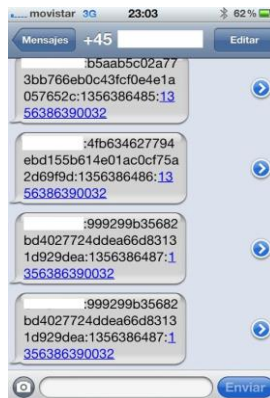
```

HTTP/1.1 200 OK
Server: nginx/0.8.55
Date: Mon, 24 Dec 2012 22:01:20 GMT
Content-Type: text/xml; charset=utf-8
Connection: close
X-Powered-By: Servlet: 2.5; JBoss-5.0/JBossWeb-2.1
Content-Length: 433

<?xml version="1.0" encoding="UTF-8"?><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><soapenv:Body><validatePhoneNumberSMLResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"><body
xmlns:xsd="xsd:string" true></body></validatePhoneNumberSMLResponse></soapenv:Envelope>

```

**Figura 12:** Fuente: Elaboración Propia



**Figura 13:** Fuente: Elaboración Propia

## 3. IPHONE

---

En este caso concreto, se auditan varias aplicaciones legítimas configurando en el iPhone nuestro proxy interceptor Burp, para poder analizar las peticiones.

[ch.smalltech.ledtorchfree](http://ch.smalltech.ledtorchfree)

---

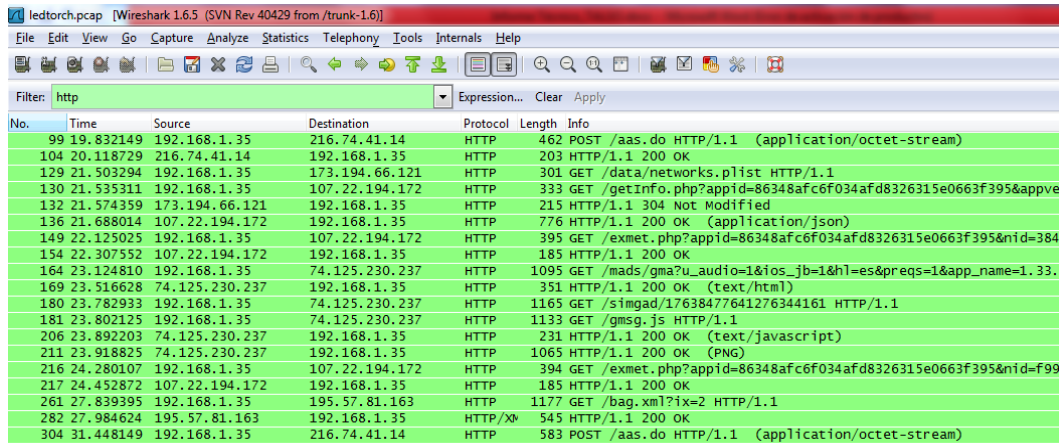
Esta aplicación se llama LinternaLed HD y es completamente legítima. Se trata de una aplicación que utiliza la luz del flash del móvil como cámara. A continuación se muestra una imagen de la ejecución de esta aplicación:



**Figura 14:** Fuente: Elaboración Propia

Se realiza un análisis dinámico de la aplicación buscando conexiones y comportamientos sospechosos.

A continuación se muestra una captura de red realizada al ejecutar la aplicación analizada:



| No. | Time      | Source         | Destination    | Protocol | Length | Info                                                            |
|-----|-----------|----------------|----------------|----------|--------|-----------------------------------------------------------------|
| 99  | 19.832149 | 192.168.1.35   | 216.74.41.14   | HTTP     | 462    | POST /aas.do HTTP/1.1 (application/octet-stream)                |
| 104 | 20.118729 | 216.74.41.14   | 192.168.1.35   | HTTP     | 203    | HTTP/1.1 200 OK                                                 |
| 129 | 21.503294 | 192.168.1.35   | 173.194.66.121 | HTTP     | 301    | GET /data/networks.plist HTTP/1.1                               |
| 130 | 21.535311 | 192.168.1.35   | 107.22.194.172 | HTTP     | 333    | GET /getinfo.php?appid=86348afc6f034afd8326315e0663f395&appver= |
| 132 | 21.574359 | 173.194.66.121 | 192.168.1.35   | HTTP     | 215    | HTTP/1.1 304 Not Modified                                       |
| 136 | 21.688014 | 107.22.194.172 | 192.168.1.35   | HTTP     | 776    | HTTP/1.1 200 OK (application/json)                              |
| 149 | 22.125025 | 192.168.1.35   | 107.22.194.172 | HTTP     | 395    | GET /exmet.php?appid=86348afc6f034afd8326315e0663f395&nid=384c  |
| 154 | 22.307552 | 107.22.194.172 | 192.168.1.35   | HTTP     | 185    | HTTP/1.1 200 OK                                                 |
| 164 | 23.124810 | 192.168.1.35   | 74.125.230.237 | HTTP     | 1095   | GET /mads/gma?u_audio=1&ios_jb=1&hl=es&preqs=1&app_name=1.33.1  |
| 169 | 23.516628 | 74.125.230.237 | 192.168.1.35   | HTTP     | 351    | HTTP/1.1 200 OK (text/html)                                     |
| 180 | 23.782933 | 192.168.1.35   | 74.125.230.237 | HTTP     | 1165   | GET /simgad/17638477641276344161 HTTP/1.1                       |
| 181 | 23.802125 | 192.168.1.35   | 74.125.230.237 | HTTP     | 1133   | GET /gmsg.js HTTP/1.1                                           |
| 206 | 23.892203 | 74.125.230.237 | 192.168.1.35   | HTTP     | 231    | HTTP/1.1 200 OK (text/javascript)                               |
| 211 | 23.918825 | 74.125.230.237 | 192.168.1.35   | HTTP     | 1065   | HTTP/1.1 200 OK (PNG)                                           |
| 216 | 24.280107 | 192.168.1.35   | 107.22.194.172 | HTTP     | 394    | GET /exmet.php?appid=86348afc6f034afd8326315e0663f395&nid=f99b  |
| 217 | 24.452872 | 107.22.194.172 | 192.168.1.35   | HTTP     | 185    | HTTP/1.1 200 OK                                                 |
| 261 | 27.839395 | 192.168.1.35   | 195.57.81.163  | HTTP     | 1177   | GET /bag.xml?ix=2 HTTP/1.1                                      |
| 282 | 27.984624 | 195.57.81.163  | 192.168.1.35   | HTTP/XV  | 545    | HTTP/1.1 200 OK                                                 |
| 304 | 31.448149 | 192.168.1.35   | 216.74.41.14   | HTTP     | 583    | POST /aas.do HTTP/1.1 (application/octet-stream)                |

Figura 15: Fuente: Elaboración Propia

Además se realizó un análisis estático buscando patrones de utilización indebida de la aplicación sin encontrarse ninguna información que lleve a pensar que se trate de un malware esta aplicación.



Para realizar este proceso se obtuvieron las librerías que carga la aplicación para entender si realmente necesita estas librerías o por el contrario está utilizando librerías que no se deberían utilizar.

```

iPhone-de-Administrador:/private/var/mobile/Applications/59697B94-D3F4-429B-8A34-5453A5038735/ledtorchfree.app root# otool -L ledtorchfree
ledtorchfree:
/usr/lib/libSystem.B.dylib (compatibility version 1.0.0, current version 161.1.0)
/System/Library/Frameworks/Twitter.framework/Twitter (compatibility version 1.0.0, current version 1.0.0)
/System/Library/Frameworks/Foundation.framework/Foundation (compatibility version 300.0.0, current version 890.1.0)
/System/Library/Frameworks/UIKit.framework/UIKit (compatibility version 1.0.0, current version 1700.0.0)
/System/Library/Frameworks/CoreGraphics.framework/CoreGraphics (compatibility version 64.0.0, current version 600.0.0)
/System/Library/Frameworks/QuartzCore.framework/QuartzCore (compatibility version 1.2.0, current version 1.7.0)
/System/Library/Frameworks/AVFoundation.framework/AVFoundation (compatibility version 1.0.0, current version 2.0.0)
/System/Library/Frameworks/AddressBook.framework/AddressBook (compatibility version 1.0.0, current version 30.0.0)
/System/Library/Frameworks/AddressBookUI.framework/AddressBookUI (compatibility version 1.0.0, current version 33.0.0)
/System/Library/Frameworks/iAd.framework/iAd (compatibility version 1.0.0, current version 1.0.0)
/System/Library/Frameworks/MediaPlayer.framework/MediaPlayer (compatibility version 1.0.0, current version 1.0.0)
/System/Library/Frameworks/SystemConfiguration.framework/SystemConfiguration (compatibility version 1.0.0, current version 432.0.0)
/usr/lib/libsqlite3.dylib (compatibility version 9.0.0, current version 9.6.0)
/usr/lib/libxml2.2.dylib (compatibility version 10.0.0, current version 10.3.0)
/usr/lib/libz.1.dylib (compatibility version 1.0.0, current version 1.2.5)
/System/Library/Frameworks/MessageUI.framework/MessageUI (compatibility version 1.0.0, current version 1.0.0)
/System/Library/Frameworks/AudioToolbox.framework/AudioToolbox (compatibility version 1.0.0, current version 359.0.0)
/System/Library/Frameworks/CFNetwork.framework/CFNetwork (compatibility version 1.0.0, current version 548.1.4)
/System/Library/Frameworks/CoreLocation.framework/CoreLocation (compatibility version 1.0.0, current version 1233.22.0)
/System/Library/Frameworks/CoreMotion.framework/CoreMotion (compatibility version 1.0.0, current version 1233.22.0)
/System/Library/Frameworks/EventKit.framework/EventKit (compatibility version 1.0.0, current version 100.0.0)
/System/Library/Frameworks/MapKit.framework/MapKit (compatibility version 1.0.0, current version 14.0.0)
/usr/lib/libobjc.2.dylib (compatibility version 1.0.0, current version 6.0.0)
/System/Library/Frameworks/CoreFoundation.framework/CoreFoundation (compatibility version 150.0.0, current version 690.1.0)
/usr/lib/libobjc.A.dylib (compatibility version 1.0.0, current version 228.0.0)

```

Figura 16: Fuente: Elaboración Propia

En la imagen anterior se observan las librerías cargadas por la aplicación y como se puede observar no hay librerías sospechosas que pudieran ser maliciosas.

com.yourcompany.TestWithCustomTabs

Esta aplicación se llama AccuWeather y es completamente legítima. Se trata de una aplicación para mostrar el tiempo en las distintas ciudades. A continuación se muestra una imagen de la ejecución de esta aplicación:



Figura 17: Fuente: Elaboración Propia

Se realiza un análisis dinámico de la aplicación buscando conexiones y comportamientos sospechosos. A continuación se muestra una captura de red realizada al ejecutar la aplicación analizada:

|     |           |               |               |          |      |                                                      |                                                                                              |
|-----|-----------|---------------|---------------|----------|------|------------------------------------------------------|----------------------------------------------------------------------------------------------|
| 112 | 25.762717 | 192.168.1.67  | 192.168.1.100 | HTTP     | 863  | GET                                                  | http://www.accuweather.com/adequest/adequest.aspx/getAdCode?strAppID=iphone51sponson&strP    |
| 113 | 25.778500 | 192.168.1.67  | 192.168.1.100 | HTTP     | 693  | GET                                                  | http://b.scorecardresearch.com/p?&name=start&c2=6005068&c1=19&c4=AccuWeather&c10=ios&c12=8CD |
| 117 | 26.075617 | 192.168.1.100 | 192.168.1.67  | HTTP     | 370  | HTTP/1.1                                             | 200 OK (GIF89a)                                                                              |
| 123 | 26.151869 | 192.168.1.100 | 192.168.1.67  | HTTP/XML | 779  | HTTP/1.1                                             | 200 OK                                                                                       |
| 158 | 30.018782 | 192.168.1.67  | 192.168.1.100 | HTTP     | 461  | GET                                                  | http://accuwx1snapshot.accu-weather.com/widget/accuwx1snapshot/weather-data.asp?location=cit |
| 159 | 30.042896 | 192.168.1.67  | 192.168.1.100 | HTTP     | 471  | GET                                                  | http://accuwx1snapshot.accu-weather.com/widget/accuwx1snapshot/weather-data.asp?location=cit |
| 160 | 30.055810 | 192.168.1.67  | 192.168.1.100 | HTTP     | 461  | GET                                                  | http://accuwx1snapshot.accu-weather.com/widget/accuwx1snapshot/weather-data.asp?location=cit |
| 161 | 30.071592 | 192.168.1.67  | 192.168.1.100 | HTTP     | 461  | GET                                                  | http://accuwx1snapshot.accu-weather.com/widget/accuwx1snapshot/weather-data.asp?location=cit |
| 163 | 30.086287 | 192.168.1.67  | 192.168.1.100 | HTTP     | 461  | GET                                                  | http://accuwx1snapshot.accu-weather.com/widget/accuwx1snapshot/weather-data.asp?location=cit |
| 179 | 32.372800 | 192.168.1.100 | 192.168.1.67  | HTTP/XML | 1077 | HTTP/1.1                                             | 200 OK                                                                                       |
| 184 | 32.377621 | 192.168.1.100 | 192.168.1.67  | HTTP/XML | 812  | HTTP/1.1                                             | 200 OK                                                                                       |
| 189 | 32.383585 | 192.168.1.100 | 192.168.1.67  | HTTP/XML | 771  | HTTP/1.1                                             | 200 OK                                                                                       |
| 194 | 32.397197 | 192.168.1.100 | 192.168.1.67  | HTTP/XML | 1038 | HTTP/1.1                                             | 200 OK                                                                                       |
| 255 | 32.512256 | 192.168.1.100 | 192.168.1.67  | HTTP     | 319  | POST                                                 | http://data.flurry.com/aap.do HTTP/1.1 (application/octet-stream)                            |
| 287 | 32.530843 | 192.168.1.100 | 192.168.1.67  | HTTP/XML | 392  | HTTP/1.1                                             | 200 OK                                                                                       |
| 302 | 33.157387 | 192.168.1.67  | 192.168.1.100 | HTTP     | 828  | GET                                                  | http://accuprod.amobee.com/upsteed/wap/adequest?time=1342004613&amobeeIncNw=c1Mmi,adMob,1ad  |
| 307 | 34.164206 | 192.168.1.100 | 192.168.1.67  | HTTP     | 215  | HTTP/1.1                                             | 200 OK                                                                                       |
| 315 | 34.575671 | 192.168.1.100 | 192.168.1.67  | HTTP     | 918  | HTTP/1.1                                             | 200 OK (text/html)                                                                           |
| 340 | 37.162569 | 192.168.1.67  | 192.168.1.100 | HTTP     | 249  | CONNECT                                              | 1adsdk.apple.com:443 HTTP/1.1                                                                |
| 341 | 37.178823 | 192.168.1.100 | 192.168.1.67  | HTTP     | 105  | HTTP/1.0                                             | 200 connection established                                                                   |
| 343 | 37.182612 | 192.168.1.67  | 192.168.1.100 | TLSv1    | 249  | client                                               | Hello                                                                                        |
| 351 | 37.326595 | 192.168.1.67  | 192.168.1.100 | HTTP     | 1256 | GET                                                  | http://ax.init.itunes.apple.com/bag.xml?ix=2 HTTP/1.1                                        |
| 460 | 39.180552 | 192.168.1.100 | 192.168.1.67  | HTTP/XML | 797  | HTTP/1.1                                             | 200 OK                                                                                       |
| 533 | 53.985881 | 192.168.1.100 | 192.168.1.67  | TLSv1    | 1495 | Server                                               | Hello, Certificate, Server Hello Done                                                        |
| 534 | 53.985948 | 192.168.1.100 | 192.168.1.67  | TLSv1    | 73   | Alert (Level: Fatal, Description: Handshake Failure) |                                                                                              |

Figura 18: Fuente: Elaboración Propia

En esta captura se han detectado conexiones a la propia página de AccuWeather realizando la siguiente petición:

| request                                                                                                                                                                                                                                                                                                                                                                                                                              | response |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|
| raw                                                                                                                                                                                                                                                                                                                                                                                                                                  | params   |
| headers                                                                                                                                                                                                                                                                                                                                                                                                                              | hex      |
| GET /adequest/adequest.aspx/getAdCode?strAppID=iphone51sponson&strPartnerCode=iphone51sponson&strIpAddress=192.168.1.67&strUserAgent=Mozilla/5.0%20(iPhone%20U; %20CPU%20iPhone%20OS%202_ %20like%20Mac%20OS%20X; %20en-us) %20AppleWebKit/525.18.1%20(KHTML, %20like%20Gecko) %20Version/3.1.1%20Mobile/5F136%20Safari/525.20&strCurrentC ipCode=cityId=30852&strWeatherIcon=01&strUID=0CA99C7-B115-4D07-855B-1E83BDC53E9C HTTP/1.1 |          |
| Host: www.accuweather.com                                                                                                                                                                                                                                                                                                                                                                                                            |          |
| User-Agent: Mozilla/5.0%20(iPhone%20U; %20CPU%20iPhone%20OS%202_ %20like%20Mac%20OS%20X; %20en-us) %20AppleWebKit/525.18.1%20(KHTML, %20like%20Gecko) %20Version/3.1.1%20Mobile/5F136%20Safari/525.20                                                                                                                                                                                                                                |          |
| Accept: */*                                                                                                                                                                                                                                                                                                                                                                                                                          |          |
| Accept-Language: es-es                                                                                                                                                                                                                                                                                                                                                                                                               |          |
| Accept-Encoding: gzip, deflate                                                                                                                                                                                                                                                                                                                                                                                                       |          |
| Connection: keep-alive                                                                                                                                                                                                                                                                                                                                                                                                               |          |
| Proxy-Connection: keep-alive                                                                                                                                                                                                                                                                                                                                                                                                         |          |

Figura 19: Fuente: Elaboración Propia



En la captura anterior se observa cómo se manda la dirección ip y nuestro user agent además de nuestro identificador de UUID único del dispositivo que aunque no se trata de un comportamiento totalmente adecuado tampoco es algo muy grave en cuanto a la privacidad de datos se refiere.



Además de lo comentado anteriormente en el análisis dinámico no se detectó nada relevante.

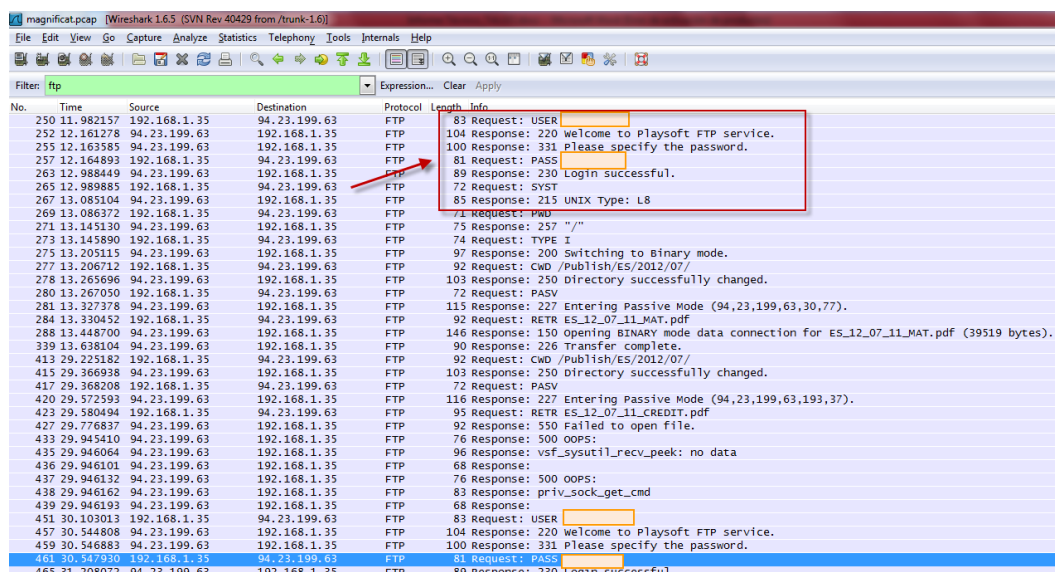
[es.magnificat.magnificat](https://es.magnificat.magnificat)

Esta aplicación se llama Magnificat y es completamente legítima. Se trata de una aplicación de carácter religioso. A continuación se muestra una imagen de la ejecución de esta aplicación:



**Figura 20:** Fuente: Elaboración Propia

Se realiza un análisis dinámico de la aplicación buscando conexiones y comportamientos sospechosos. A continuación se muestra una captura de red realizada al ejecutar la aplicación analizada:



| No. | Time      | Source       | Destination  | Protocol | Length | Info                                                                                         |
|-----|-----------|--------------|--------------|----------|--------|----------------------------------------------------------------------------------------------|
| 250 | 11.982157 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 83 Request: USER [redacted]                                                                  |
| 252 | 12.161278 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 104 Response: 220 Welcome to Playsoft FTP service.                                           |
| 255 | 12.163585 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 100 Response: 331 Please specify the password.                                               |
| 257 | 12.164893 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 81 Request: PASS [redacted]                                                                  |
| 263 | 12.988449 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 89 Response: 230 Login successful.                                                           |
| 265 | 12.989885 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 72 Request: SYST                                                                             |
| 267 | 13.085104 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 85 Response: 215 UNIX Type: L8                                                               |
| 269 | 13.086372 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 71 Request: PWU                                                                              |
| 271 | 13.145130 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 75 Response: 257 "/"                                                                         |
| 273 | 13.145890 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 74 Request: TYPE I                                                                           |
| 275 | 13.205115 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 97 Response: 200 Switching to Binary mode.                                                   |
| 277 | 13.206712 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 92 Request: CWD /Publish/Es/2012/07/                                                         |
| 278 | 13.265696 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 103 Response: 250 Directory successfully changed.                                            |
| 280 | 13.267050 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 72 Request: PASV                                                                             |
| 281 | 13.327378 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 115 Response: 227 Entering Passive Mode (94,23,199,63,30,77).                                |
| 284 | 13.330452 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 92 Request: RETR ES_12_07_11_MAT.pdf                                                         |
| 288 | 13.448700 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 146 Response: 150 opening BINARY mode data connection for ES_12_07_11_MAT.pdf (39519 bytes). |
| 339 | 13.638104 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 90 Response: 226 Transfer complete.                                                          |
| 413 | 29.225182 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 92 Request: CWD /Publish/Es/2012/07/                                                         |
| 415 | 29.366938 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 103 Response: 250 Directory successfully changed.                                            |
| 417 | 29.368208 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 72 Request: PASV                                                                             |
| 420 | 29.572593 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 116 Response: 227 Entering Passive Mode (94,23,199,63,193,37).                               |
| 423 | 29.580494 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 95 Request: RETR ES_12_07_11_CREDIT.pdf                                                      |
| 427 | 29.776837 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 92 Response: 550 Failed to open file.                                                        |
| 433 | 29.945410 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 76 Response: 500 OOPS:                                                                       |
| 435 | 29.946064 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 96 Response: vsf_sysutil_recv_peek: no data                                                  |
| 436 | 29.946101 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 68 Response:                                                                                 |
| 437 | 29.946132 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 76 Response: 500 OOPS:                                                                       |
| 438 | 29.946162 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 83 Response: priv_sock_get_cmd                                                               |
| 439 | 29.946193 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 68 Response:                                                                                 |
| 451 | 30.103013 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 83 Request: USER [redacted]                                                                  |
| 457 | 30.144808 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 104 Response: 220 Welcome to Playsoft FTP service.                                           |
| 459 | 30.546883 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 100 Response: 331 Please specify the password.                                               |
| 461 | 30.547930 | 192.168.1.35 | 94.23.199.63 | FTP      |        | 81 Request: PASS [redacted]                                                                  |
| 465 | 31.308075 | 94.23.199.63 | 192.168.1.35 | FTP      |        | 89 Response: 230 Login successful.                                                           |

Figura 21: Fuente: Elaboración Propia



En esta captura de red la aplicación se conecta a un servidor FTP con las credenciales en texto plano para descargarse un fichero PDF donde se encuentran las oraciones que luego el usuario final visualiza en la aplicación por lo cual esta aplicación resulta potencialmente peligrosa ya que un usuario ilegítimo podría subir contenido malicioso a dicho servidor web pudiendo comprometer finalmente el teléfono del usuario.



## 4. BLACKBERRY



Para auditar aplicaciones blackberry se puede usar el emulador de blackberry que facilita mucho en la tarea de analizar el comportamiento de las aplicaciones.

<http://us.blackberry.com/sites/developers/resources/simulators.html>

En este caso concreto, se audita una aplicación legítima configurando en el terminal emulado nuestro proxy interceptor Burp, para poder analizar las peticiones.

En las trazas enviadas por la aplicación se puede observar la siguiente petición HTTP, en la que manipulando la información enviada (letras en color rojo) se consigue un error en la aplicación generando una denegación de servicio. Además se observó más adelante que al modificar estas peticiones se corrompía la base de datos del servidor destino que provocaba una serie de errores graves en el sistema.

POST /SAKTU\_RE\_ALUF/ws/WLANS\_DOE\_Listener HTTP/1.1

User-Agent: kSOAP/2.0

SOAPAction:

https://example.com/SAKTU\_RE\_ALUF/ws/WLANS\_DOE\_Listener/ValidarTID

Content-Type: text/xml

Connection: close

Content-Length: 525

Host: example.com

Accept-Encoding: gzip

```
<soapenv:Envelope xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns:d="http://www.w3.org/2001/XMLSchema"
xmlns:c="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"><soapenv:Header
/><soapenv:Body><ValidarTID tan-
de="LOWKSI"><tande2><id>12345345555555555555555555555555444444444444444AQAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</id>
</tande2></ValidarTID></soapenv:Body></soapenv:Envelope>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<soap-env:Envelope xmlns:soap-
env="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<soap-env:Header/>

<soap-env:Body>

<soap-env:Fault>

<faultcode>soap-env:Server</faultcode>

<faultstring>Executing Facade: LOWKSI, Method: ValidarTID</faultstring>

<detail>

<prefixRigel0:Exception
xmlns:prefixRigel0="http://example.com/ws/genericError">

<faultCode>soap-env:Server</faultCode>

<Value>Receiver</Value> <Subcode> <Value/>
```

```
</Subcode> </code> <reasons> <reason> <lang>en</lang> <text>The request
message could not be processed, because it happened an unexpected error in
the Server</text> </reason> </reasons> <technicalException> <technicalPlat-
form>TER</technicalPlatform> <platformCode></platformCode> <message>An
error occurred when locating element 'id' in the collection. Element 'id' is not in
the collection (key does not exist) or the data type is null.</message> <stack-
Trace>services.messaging.ExtendedMessagingException: [BKSE04023001363]
: Error processing class: 'messag-
ing.facadecaller.FacadeCallerEngineBridgeImpl'. Message '[BKSE08003000108]
: An error occurred when executing the facade'
at
com.example.test.testa..FacadeCallerEngineBridgeImpl.onMessage(FacadeCalle
rEngineBridgeImpl.java:152)
at
com.example.test.testa..srcimpl.messaging.MessagingServiceImpl.onRequest(
MessagingServiceImpl.java:1257)
```

```
at
com.example.test.testa..srcimpl.messaging.http.MessagingSubsystemHTTPImpl
l.onMessage(MessagingSubsystemHTTPImpl.java:1257)
at
com.example.test.testa..srcimpl.messaging.MessageBridge.notify(MessageBrid
ge.java:107)
at
com.example.test.testa..srcimpl.messaging.http.HTTPServletListener.doPost(H
TTPServletListener.java:104)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:738)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:831)
at
com.example.test.testa.ws.webcontainer.servlet.ServletWrapper.service(Servlet
Wrapper.java:1657)
at
com.example.test.testa.ws.webcontainer.servlet.ServletWrapper.handleRequest(
ServletWrapper.java:939)
at
com.example.test.testa.ws.webcontainer.servlet.ServletWrapper.handleRequest(
ServletWrapper.java:502)
at
com.example.test.testa.ws.webcontainer.servlet.ServletWrapperImpl.handleRequ
est(ServletWrapperImpl.java:179)
at
com.example.test.testa.ws.webcontainer.servlet.CacheServletWrapper.handleRe
quest(CacheServletWrapper.java:91)
at
com.example.test.testa.ws.webcontainer.WebContainer.handleRequest(WebCon
tainer.java:864)
at
com.example.test.testa.ws.webcontainer.WSWebContainer.handleRequest(WS
WebContainer.java:1583)
at
com.example.test.testa.ws.webcontainer.channel.WCChannelLink.ready(WCCha
nnelLink.java:186)
at
com.example.test.testa.ws.http.channel.inbound.impl.HttpInboundLink.handleDis
crimination(HttpInboundLink.java:445)
at
com.example.test.testa.ws.http.channel.inbound.impl.HttpInboundLink.handleNe
wRequest(HttpInboundLink.java:504)
```

```
at
com.example.test.testa.ws.http.channel.inbound.impl.HttpInboundLink.processRe
quest(HttpInboundLink.java:301)
at
com.example.test.testa.ws.http.channel.inbound.impl.HttpInboundLink.ready(Http
InboundLink.java:275)
at
com.example.test.testa.ws.tcp.channel.impl.NewConnectionInitialReadCallback.s
endToDiscriminators(NewConnectionInitialReadCallback.java:214)
at
com.example.test.testa.ws.tcp.channel.impl.NewConnectionInitialReadCallback.c
omplete(NewConnectionInitialReadCallback.java:113)
at
com.example.test.testa.ws.tcp.channel.impl.AioReadCompletionListener.futureC
ompleted(AioReadCompletionListener.java:165)
at
com.example.test.testa.io.async.AbstractAsyncFuture.invokeCallback(AbstractAs
yncFuture.java:217)
at
com.example.test.testa.io.async.AsyncChannelFuture.fireCompletionActions(Asy
ncChannelFuture.java:161)
at
com.example.test.testa.io.async.AsyncFuture.completed(AsyncFuture.java:138)
at
com.example.test.testa.io.async.ResultHandler.complete(ResultHandler.java:204
)
at
com.example.test.testa.io.async.ResultHandler.runEventProcessingLoop(Result
Handler.java:775)
at com.example.test.testa.io.async.ResultHandler$2.run(ResultHandler.java:905)
at com.example.test.testa.ws.util.ThreadPool$Worker.run(ThreadPool.java:1563)
Caused by: com.example.test.testa..bl.facadecaller.FacadeCallerException:
[BKSE08003000008] : An error occurred when executing the facade
at
com.example.test.testa..bl.facadecaller.FacadeCallerImpl.callFacade(FacadeCall
erImpl.java:178)
at
com.example.test.testa..FacadeCallerEngineBridgeImpl.onMessage(FacadeCalle
rEngineBridgeImpl.java:132)
... 28 more
Caused by: com.example.test.testa..bl.facadecaller.FacadeCallerException:
[BKSE04023000367] : An error occurred when mapping the input
```

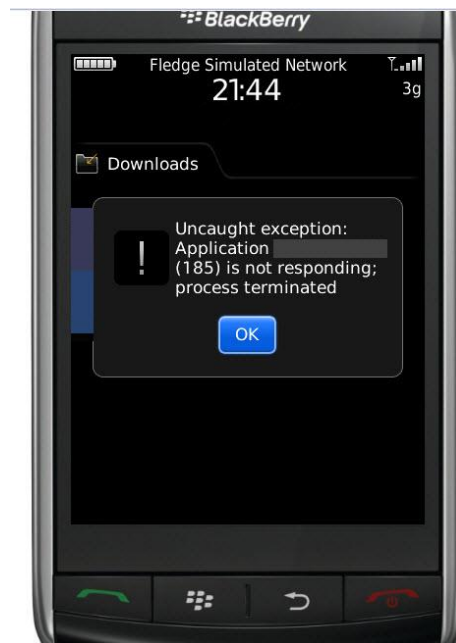
```
at
com.example.test.testa..mappers.RigelMSFacadeCallMapperImpl.map(RigelMSF
acadeCallMapperImpl.java:81)
at
com.example.test.testa..bl.facadecaller.FacadeCallerImpl.callFacade(FacadeCall
erImpl.java:109)
... 29 more
Caused by: com.example.test.testa..common.data.MapperException: An error
occurred when mapping element 'tande2'
at
com.example.test.testa..common.mapper.CompDataMapper.mapData(CompDat
aMapper.java:92)
at
com.example.test.testa..mappers.RigelMSFacadeCallMapperImpl.map(RigelMSF
acadeCallMapperImpl.java:72)
... 30 more
Caused by: com.example.test.testa..common.data.MapperException: An error
occurred when mapping element 'id'
at
com.example.test.testa..common.mapper.CompDataMapper.mapData(CompDat
aMapper.java:92)
at
com.example.test.testa..common.mapper.CompDataMapper.mapData(CompDat
aMapper.java:86)
... 31 more
Caused by: com.example.test.testa..common.data.MapperException: An error
occurred when locating element 'id' in the collection. Element 'id' is not in the
collection (key does not exist) or the data type is null.
at
com.example.test.testa..common.mapper.CompDataMapper.mapData(CompDat
aMapper.java:61)
... 32 more
----- Previous exception stack trace -----
Exception: com.example.test.testa..bl.facadecaller.FacadeCallerException, mes-
sage: [BKSE08003000008] : An error occurred when executing the facade
----- Previous exception stack trace -----
Exception: com.example.test.testa..bl.facadecaller.FacadeCallerException, mes-
sage: [BKSE04023000367] : An error occurred when mapping the input
----- Previous exception stack trace -----
Exception: com.example.test.testa..common.data.MapperException, message:
An error occurred when mapping element 'tande2'
```

—— Previous exception stack trace ——

Exception: com.example.test.testa..common.data.MapperException, message:  
An error occurred when mapping element 'id'

—— Previous exception stack trace ——

Exception: com.example.test.testa..common.data.MapperException, message:  
An error occurred when locating element 'id' in the collection. Element 'id' is not  
in the collection (key does not exist) or the data type is null.</stackTrace>  
</technicalException> <additionalInfo/> </prefixRigel0:Exception> </detail>  
</soap-env:Fault> </soap-env:Body> </soap-env:Envelope>



**Figura 22:** Fuente: Elaboración Propia

## CONCLUSIONES

---

A lo largo del tema hemos podido comprobar que cada día nuevas noticias confirman que actualmente es muy importante tener cuidado con lo que se comparte en la red y se hace público para todo el mundo, ya que con teclear el nombre de una persona en un buscador como Google o en páginas especializadas de búsquedas de personas es posible obtener información casi de cualquier persona que esté activa en internet.

Las aplicaciones móviles, o APP, son cada vez más usadas en entornos como las redes sociales, donde internautas jóvenes aceptan ejecutar cualquiera de ellas sin leer las políticas de privacidad de la App, por lo que aún menos se puede esperar que entiendan de permisos o comportamientos extraños. Sin embargo, la realidad nos indica que cada día más y más APP aparecen ligadas al malware móvil.

Muchas APP son tan famosas que los usuarios piensan que son seguras. Pero esto no siempre es así. Pero no siempre los fallos de privacidad y de seguridad en las aplicaciones son intencionales: Muchas veces a los desarrolladores se les pasa bloquear algunas cuestiones de privacidad. Es aquí donde la auditoría de las APP adquiere su verdadera razón de ser.

## RECAPITULACIÓN

---

Tras lo visto en el tema, como recapitulación, comentar que como es ya bien sabido, resulta más práctico y útil aplicar la seguridad en el desarrollo de software, ya sea APP o cualquier otro software utilizando un ciclo de desarrollo seguro. Esto ayuda a disminuir costos de desarrollo y mantenimiento.

En el caso del desarrollo de aplicaciones para terminales móviles, las lecciones aprendidas para el desarrollo seguro de software, y la auditoría de las mismas, son perfectamente aplicables para este entorno. Lo único que sucede es que hay que adaptar las herramientas y las técnicas a utilizar.

Hemos visto que para cada entorno hay que buscar una estrategia que nos permita entender el funcionamiento de la aplicación a auditar, viendo cómo es su comportamiento en el sistema, ya sea emulándolo o interceptando sus comunicaciones.



## AUTOCOMPROBACIÓN

---

### 1. En el proceso de auditorías de aplicaciones móviles se debe:

- a) Analizar el código de la aplicación si se tiene acceso o en caso contrario intentar analizar en la medida de lo posible el código de la aplicación una vez desensamblado.
- b) Analizar el código de la aplicación si se tiene acceso o en caso contrario intentar analizar en la medida de lo posible el código de la aplicación sin desensamblar.
- c) No hace falta analizar el código de la aplicación pero siempre intentar analizar en la medida de lo posible el código de la aplicación una vez desensamblado.
- d) Sólo analizar el código de la aplicación si se tiene acceso.

### 2. En el proceso de auditorías de aplicaciones móviles se debe:

- a) Analizar sólo los ficheros que lee la aplicación en el teléfono una vez es instalada y una vez es ejecutada.
- b) Analizar los ficheros que lee y sobre todo crea la aplicación en el teléfono una vez es ejecutada.
- c) Analizar los ficheros que lee y sobre todo crea la aplicación en el teléfono una vez es instalada y una vez es ejecutada.
- d) Analizar los ficheros que crea la aplicación en el teléfono una vez es instalada y una vez es ejecutada.

**3. Al analizar las comunicaciones de la aplicación es importante, ya que:**

- a) Mediante esta fase podremos intervenir las comunicaciones mediante ataques man in the middle y aunque no será posible manipular las comunicaciones si es el buscar vulnerabilidades de todo tipo.
- b) Mediante esta fase podremos intervenir las comunicaciones mediante ataques man in the middle y será posible manipular las comunicaciones. Para buscar vulnerabilidades de todo tipo es necesario otras técnicas.
- c) Mediante esta fase podremos intervenir las comunicaciones mediante ataques man in the middle y será posible manipular las comunicaciones y buscar vulnerabilidades de todo tipo.
- d) Mediante esta fase no podremos intervenir las comunicaciones mediante ataques man in the middle pero si será posible manipular las comunicaciones y buscar vulnerabilidades de todo tipo.

**4. Al analizar los ficheros que lee y sobre todo crea la aplicación en el teléfono una vez es instalada y una vez es ejecutada, se puede encontrar:**

- a) Información sensible, como usuarios y contraseñas cifradas.
- b) Información sensible, como usuarios y contraseñas en texto claro.
- c) Información inservible.
- d) Información no sensible en texto claro.

**5. Para poder comprobar cómo se interceptan las comunicaciones de aplicaciones**

- a) Se puede hacer con una aplicación funcionando en un terminal Android emulado y utilizando un proxy interceptor burp.
- b) Se puede hacer con un terminal real y utilizando un proxy interceptor burp.
- c) Sólo se puede hacer con una aplicación funcionando en un terminal Android emulado.
- d) Ninguna de las restantes respuestas.

**6. Se pueden:**

- a) Interceptar tanto las comunicaciones http pero no las https y manipularlas para poder buscar vulnerabilidades.
- b) Interceptar tanto las comunicaciones http como https y manipularlas para poder buscar vulnerabilidades.
- c) Interceptar tanto las comunicaciones http como https pero no manipularlas para poder buscar vulnerabilidades.
- d) Ninguna de las restantes respuestas.

**7. La interceptación de las comunicaciones http como https y manipularlas es:**

- a) Esta fase no es posible, ya que nos puede interceptar https.
- b) Esta fase suele ser la más crítica pero no se suele encontrar un mayor número de vulnerabilidades.
- c) Esta fase suele ser la más crítica ya que es donde se demuestra que el OS es vulnerable
- d) Esta fase suele ser la más crítica ya que es donde se suele encontrar un mayor número de vulnerabilidades.

**8. Interceptando las comunicaciones es posible:**

- a) Enviar SMS de forma automatizada a cualquier número sin realizar ninguna comprobación pero no se puede automatizar este ataque para enviar un número indeterminado de SMS.
- b) Enviar SMS de forma automatizada a cualquier número sin realizar ninguna comprobación y se puede automatizar este ataque para enviar un número finito de SMS.
- c) Enviar SMS de forma automatizada a cualquier número sin realizar ninguna comprobación y se puede automatizar este ataque para enviar un número indeterminado de SMS.
- d) Sólo encontrar vulnerabilidades básicas.

**9. Una aplicación que se conecta a un servidor FTP con las credenciales en texto plano para descargarse un fichero**

- a) Resulta potencialmente peligrosa ya que un usuario ilegítimo podría subir contenido malicioso a dicho servidor web pudiendo comprometer finalmente el teléfono del usuario.
- b) No resulta potencialmente peligrosa ya que un usuario ilegítimo no podría acceder a las credenciales.
- c) El peligroso pero el OS detectaría cualquier intrusión.
- d) Ninguna de las restantes respuestas.

**10. Para auditar aplicaciones blackberry se puede usar:**

- a) El emulador de blackberry que facilita mucho en la tarea de analizar el comportamiento de las aplicaciones.
- b) El emulador de blackberry pero que es muy complicado de usar para analizar el comportamiento de las aplicaciones.
- c) Un terminal real solamente.
- d) Ninguna de las restantes respuestas.

## SOLUCIONARIO

|    |   |    |   |    |   |    |   |     |   |
|----|---|----|---|----|---|----|---|-----|---|
| 1. | a | 2. | c | 3. | c | 4. | b | 5.  | a |
| 6. | b | 7. | d | 8. | c | 9. | a | 10. | a |

## PROPUESTAS DE AMPLIACIÓN

---

Una vez has visto este tema, te proponemos tres posibles líneas para ampliar lo aquí tratado. La primera es en sistemas ANDROID, identificar y des-ensamblar la aplicación, viendo cómo se pueden visualizar los datos de la misma, para lo cual sólo necesitas hacerlo como con cualquier aplicación JAVA.

La segunda es que configures el proxy interceptor Burp, para poder analizar las peticiones que se realizan contra un sistema iOS. Por otro lado, prueba a conseguir iEMU, como emulador de sistemas iOS, y haz lo mismo que en el anterior punto.

Por último, y para sistemas BLACKBERRY, instálale el emulador de Blackberry, y analiza el comportamiento de aplicaciones en este entorno, tal y como se han visto en el tema estudiado.



## BIBLIOGRAFÍA

---

- Troyano, R., (2006). La auditoría informática. Su necesidad y metodología. Ed. Pratida Doble. Recuperado de <http://pdfs.wke.es/5/4/8/1/pd0000015481.pdf>
- Varios autores. (2013) BlackBerry Simulators . Ed RIM. Recuperado de [es.blackberry.com/developers/resources/simulators.jsp](http://es.blackberry.com/developers/resources/simulators.jsp)
- Varios autores (2013) Burp Suite Tutorial, The Intruder Tool. Ed. Security Ninja Recuperado de <http://www.securityninja.co.uk/hacking/burp-suite-tutorial-the-intruder-tool/>,
- Hoog, A. Strzempka, K. (2011) "IPHONE AND iOS FORENSICS" Ed Syngress
- Hoog, A (2011) "Android Forensics Investigation, Analysis, and Mobile Security for Google Android" Ed. Syngress
- Daniel, L. (2011) "Digital Forensics for Legal Professionals", Ed. Syngress
- Varios autores (2011) Malware en Smartphones. Ed. Consejo Nacional Consultivo de Cyberseguridad. Recuperado de [http://www.bdigital.org/Documents/Malware\\_Smartphones.pdf](http://www.bdigital.org/Documents/Malware_Smartphones.pdf)