Máster propio en ciberseguridad (Edición I)

•

Seguridad de aplicaciones

•

Práctica 3 SA – Seguridad en aplicaciones móviles

Pablo Sánchez Robles

En esta práctica se va ha hacer uso de la herramienta para análisis estático y dinámico de aplicaciones móviles MobSF, que sirve tanto para aplicaciones Android como IOS, como Windows mobile.

Como comienzo paso a analizar las versiones de Whatsapp de 2014 y 2015 para encontrar las diferencias entre ellas y posteriormente se pasará a probar la v049 de Pokemon Go.

Tras seguir los pasos de instalación que podemos encontrar en el github de la aplicación se nos arrancará un servidor web en local cuya interfaz se puede ver en la siguiente captura:

Como se puede apreciar el interfaz es altamente sencillo, sólo teniendo que pulsar el boton “updload & analyze” para seleccionar un archivo apk o la estructura de carpeta del apk deslincado, o arrastrarlo directamente al recuadro blanco del centro.

Una vez finalizado el análisis el programa despliega los resultados en forma de web. Dichos resultados se pueden exportar a un pdf para su almacenamiento o envío.

La estructura del resultado es la siguiente:

1. File Information: muestra los datos del fichero que se le pasa, como son el nombre, el tamaño del fichero y los hashes MD5 SHA1 SHA256.
2. App Information: muestra el nombre del paquete , la versión de android mínima así como el sdk mínimo, y el nombre del main activity o proceso que se ejecuta en el dispositivo móvil.
3. Code Nature: Da información sobre si se ejecuta en nativo o en remoto, si tiene mecanismos criptográficos y si utiliza ofuscación.

4. Certificate: Muestra información sobre el certificado digital con el que se firma la aplicación.
5. Permissions: Muestra toda la información acerca de los permisos que necesita la aplicación para su funcionamiento.
6. Manifest Analysis: Muestra información sobre todas las actividades internas que lleva a cabo la aplicación ya sea en primer plano o segundo plano.
7. Code Analysis: Te explica el funcionamiento de la aplicación evaluando la idoneidad de una serie de factores así como la severidad de dichas evaluaciones
8. Andoriod API: Muestra las diferentes APIs que utiliza la aplicación.
9. URLs: Urls a las que redirecciona la app, en blanco si no existen.
10. Malware check: Te muestra información si encuentra algún tipo de malware en la app, en caso contrario está en blanco.
11. Strings: No se lo que hace, en mi caso está en blanco.
12. Activities: Lista con todas las actividades de la app.
13. Providers: Proveedores de la app.
14. Receivers: Recibidores de la app
15. Services: Lista con los servicios que genera la app.
16. Libraries: Bibliotecas de las que hace uso la aplicación.
17. Files: Lista con todos y cada uno de los ficheros de la app.

Análisis estáticos

He realizado análisis estáticos de una versión de Whatsapp anterior al cifrado, una posterior y la última versión hasta el momento, y el juego Pokemon Go v0.49. Comenzaré por hacer una revisión de whatsapp. (Todos los documentos de análisis generados están adjuntos, así como los apks utilizados)

Tanto en la versión anterior al cifrado como en la posterior los permisos que necesita para funcionar son muchos y cuya severidad el programa califica como “High” ya que comparte datos con otras aplicaciones, toma control de dispositivos de entrada-salida tiene un receptor de mensajes “broadcast” que puede ser utilizado para recibir mensajes con código malicioso, etc.

Las diferencias mas grandes existentes entre ambas versiones, como era de esperar, es el cifrado de comunicaciones, que en la versión de 2014 no aparece, aunque el analizador nos dice que ambas versiones tiene riesgos de seguridad de severidad alta.

En la versión previa al cifrado el analizador no es capaz de analizar, valga la redundancia, el código de la aplicación así que paso a comentar el de la versión de 2015. Todos los fallos de seguridad de esta versión cabe esperar que serán como mucho igual de peligrosos que en versiones anterior (si suponemos que las actualizaciones mejoran dichos errores).

Los problemas encontrados en la versión de 2015 son:

- La aplicación utiliza un generador de números aleatorios inseguro: el generador aleatorio se utiliza para el establecimiento de las claves de cifrado entre otras cosas, por lo que ya nos está diciendo que el cifrado no es seguro 100% ya que existen problemas de seguridad en la generación de las claves que utiliza.
- La aplicación puede leer y escribir en el almacenamiento interno del dispositivo: Lo que nos puede llevar a la inserción de código arbitrario en el mismo si la aplicación fuera alterada para ser administrada por una entidad que no fuera el propio servidor de whatsapp.
- La aplicación registra información. La información sensible no debería ser registrada en ningún caso: Este fallo de seguridad podría registrar y compartir información sensible del usuario.
- La aplicación utiliza “Java Hash Code” un tipo de hash inseguro que no debería ser utilizado en criptografía: Este hash forma parte del cifrado entre otras cosas, lo que hace pensar que el cifrado de la comunicación es aún mas débil de lo anteriormente planteado.
- La aplicación utiliza SQLite y ejecuta consultas “raw”, o lo que es lo mismo, consultas sin paramentrizar y sin asegurar lo que podría producir inserciones de código SQL no deseadas.

Cómo se a podido observar en el análisis de código la aplicación Whatsapp en su versión de 2015 tiene grandes fallos de seguridad.

Para no alargar en demasía este documento voy a comentar las diferencias entre la versión de 2015 y la última, a grandes rasgos.

Los permisos de la aplicación final son los mismos que en sus versiones anteriores por lo que los problemas de seguridad deribados de ellos siguen presentes en esta versión. En el análisis de código se puede ver cómo esta versión mantiene todos y cada uno de los fallos de seguridad contenidos en la versión anterior por lo que mantiene dichas fallas igualmente. Las diferencias mas grandes entre las versiones de 2015 y 2016 vienen dadas en la sección de los archivos que utiliza la aplicación para su funcionamiento, que en el caso de la versión de 2016 vienen en una cantidad mayor. Esto da que pensar que no se han solucionado fallos anterior e incluso con la inclusión de todos esos nuevos ficheros puede que se haya añadido algún nuevo problema.

En el análisis estático de Pokemon Go vemos que la aplicación tiene permiso de acceso a GPS, vibrador, lectura-escritura en SD, manejo de la cámara, uso de credenciales, acceso completo a internet, bluetooth, etc. Muchos de ellos con riesgo de seguridad “dangerous” y algunos “normal”.

En la parte de análisis de código podemos ver:

- La aplicación utiliza un generador de numeros aleatorios inseguro.
- Una implementación insegura de un visor web(sólo da nivel “warning”).
- Aplicación puede leer y escribir en almacenamiento externo.
- Puede almacenar información sensible en los registros.
- Utiliza una versión no segura de hash (Java Hash Code).
- Los archivos puede contener información sensible “hardcoded” como nombres de usuario, contraseñas, claves, etc.
- Aplicación usa SQLite y consultas sin tratar.
- La aplicación puede tener capacidad de detección de “root”.

Como se puede observar en el análisis de código se puede ver que al igual que ocurria en Whatsapp la aplicación tiene problemas de seguridad de alto nivel que ponen en riesgos datos de carácter personal, así como credenciales y claves, pudiendo escribir en almacenamiento externo, lo cúal puede llevar a la inserción de código arbitrario.