IT·SS
Expertos en S.I & Ciberseguridad

CONTACTAR

ETHICAL HACKING

APP MOVIL

INFORME ETHICAL HACKING

Diciembre 28 del 2020

CONFIDENCIALIDAD

Este documento incluye información confidencial de uso exclusivo de las partes interesadas en la investigación del asunto.

Toda información contenida en el presente documento debe ser mantenida en forma estrictamente confidencial y utilizada exclusivamente para el desarrollo del objetivo del presente documento.

Las dependencias y/o funcionarios involucrados(as) en la lectura, revisión y/o aprobación del presente documento se obligan a darle el uso estrictamente necesario para el cumplimiento de los objetivos trazados, de igual forma se compromete a tomar las medidas necesarias para que la información no llegue a manos de terceros bajo ninguna circunstancia.

ESTE DOCUMENTO ES GENERADO EN FORMATO PDF CON PERMISOS PARA LECTURA EN PANTALLA UNICAMENTE.

CONTROL DE CAMBIOS

VERSION	FECHA	DESCRIPCION CAMBIO	ELABORO
1.0	Diciembre 28 de 2020	Documento Inicial	Carlos Enriquez

TABLA DE CONTENIDO

TABLA DE ILUSTRACIONES

1. Generalidades

Empresa Solicitante	CONTACTAR COLOMBIA
Equipo	Carlos Enriquez
Sistemas Auditados	Aplicación Móvil

2. Objetivo

- Realizar pruebas de Ethical Hacking a la aplicación móvil.
- Determinar si es posible y cómo un usuario malintencionado puede obtener acceso no autorizado a través de cualquiera de estos activos auditados.

3. Alcance

A fin de conocer vectores de riesgo relacionado con las aplicaciones, se realizan auditorias sobre las siguientes aplicaciones:

NOMBRE DE LA APLICACION	URL/IP	TIPO
Aplicación Móvil	FORMIIK	APK

4. Resumen ejecutivo

Para presentar los resultados de este análisis se establecen los siguientes niveles de riesgo en la evaluación de las vulnerabilidades encontradas:

Nivel de riesgo Alto: Fallas de seguridad que proporcionan información clara para acceder al sistema, o permiten el acceso directo al mismo.

Nivel de riesgo Medio: Fallas de seguridad que proporcionan información del sistema que podrían facilitar el acceso al mismo, utilizando técnicas de explotación y convirtiendo la falla en una de nivel de riesgo alto.

Nivel de riesgo Bajo: Fallas de seguridad que por sí solas no comprometen la seguridad del sistema analizado, pero combinado con otras fallas y utilizando técnicas de explotación podría aumentar el nivel de riesgo a medio o alto.

Las vulnerabilidades totales identificadas corresponden a la siguiente distribución:

Nivel de riesgo	Total
Alto	0
Medio	5
Bajo	0
Total encontrado	0

5. Metodología

Para la realización de las pruebas se usa como base la metodología de pruebas de penetración de IT SECURITY SERVICES, la guía de pruebas de penetración de PCI (Penetration Testing Guidance), teniendo en cuenta las siguientes fases:

- Recopilación de información / gestión de configuración.
- Test de Autenticación.
- Test de Autorización.
- Pruebas de gestión de sesión.
- Pruebas lógicas.
- Pruebas de manejo de errores.
- Validación de Data.
- Pruebas de almacenamiento con cifrado inseguro.
- Pruebas de comunicaciones inseguras.

Se realizaron pruebas manuales de inspección de código, y validación de data input, así la como verificación de usuarios por defecto, usuarios en blanco, contraseñas débiles, entre otras.

6. Analisis manual

En este caso se realiza la verificación del código en busca de funciones, procedimientos mal formados o variables flotantes que puedan exponer data sensible. Adicionalmente, se revisan los hallazgos arrojados por las herramientas de escaneo, corroborando la no existencia de falsos positivos en dichos resultados, esta comprobación se realiza revisando manualmente las peticiones que permitan explotar las vulnerabilidades asociadas a:

- Errores de inyección (SQLi, comandos de SO, LDAP, XPATH, entre otros). (6.5.1)
- Buffer overflow. (6.5.2)
- Almacenamiento y comunicaciones inseguras (no cifradas). (6.5.3 – 6.5.4)
- Manejo inadecuado de errores. (6.5.5)
- Cross site scripting (XSS). (6.5.7)
- Acceso no controlado a archivos y carpetas. (6.5.8)
- Cross Site Request Forgery (CSRF). (6.5.9)
- Session hijacking / Session fixation. (6.5.10)

Para dicha actividad se utilizan diferentes herramientas que están contenidas en la SUITE de Kali Linux, así como algunas herramientas de Windows. Una vez explotadas las vulnerabilidades, se extraen las evidencias correspondientes descartando los falsos positivos.

Por otro lado, se revisan las vulnerabilidades de alto riesgo detectadas en el proceso de identificación de vulnerabilidades (6.5.6), llevado a cabo por la organización para cada una de las IPs asociadas a las diferentes aplicaciones. En este proceso se ejecutan pruebas que permitan identificar si dichos fallos son falsos positivos o riesgos explotables que puedan comprometer la seguridad de los activos y/o aplicación.

Para las aplicaciones cliente servidor se realizan pruebas manuales de inserción de código, en donde existan campos de input data, adicionalmente se realizan intentos de acceso no autorizado sobre archivos y carpetas, pruebas de almacenamiento, comunicaciones inseguras y manejo inadecuado de errores.

7. Limitaciones

Durante las pruebas de aplicación se encontraron las siguientes limitaciones:

- Las pruebas se realizaron en caja negra, con desconocimiento total de los ambientes y activos auditados.

8. Herramientas tecnicas utilizadas

Para el desarrollo de las pruebas sobre aplicación interna, se emplearon un conjunto de herramientas especializadas.

A continuación, se mencionan algunos de los aplicativos más utilizados en el proceso:

HERRAMIENTA	DESCRIPCIÓN
Apktool	Herramienta de ingeniería inversa para APK
Decompiler	Descompilador específico para clases java
Python	Interprete para scripting del mismo lenguaje
Metasploit	Framework de explotación general
Burp suite	Proxy HTTP
Emulador	Emulador de Android

9. Prueba Narrativa

El objetivo de las pruebas realizadas, era diagnosticar el estado de seguridad de los activos de información establecidos en el alcance de este documento y si de alguna u otra manera se podía llegar a verse comprometidos.

Se realizó el análisis manualmente, validando los servicios disponibles dentro del emulador, como sus funciones o acciones.

Se anexan las evidencias de los análisis correspondientes con las vulnerabilidades encontradas dentro de la validación de las pruebas:

9.1. Infraestructura

9.1.1. Aplicación móvil FORMIIK

Ilustración 1 respuesta app modo root

APP SUMMARY	
App Name	Contactar
Android Package	com.kata.formiik
Date of Scan	9-DEC-2020, 9:15 AM
App Version	7.2.1
Android Min SDK Version	21
Android Target SDK Version	29
App Size	55.4 MB

Ilustración 2 Datos de la aplicación móvil

Ilustración 3 Prueba interceptación trafico

File Path:	l\a\a\a\c\g.java
Line	87. hashMap.put("redirect_uri", "http://localhost/");

Ilustración 4 Redirect

File Path:	l\a\a\d\e\a.java
Line	319. String format3 = String.format("http://%s.blob.core.windows.net/", new Object[]{G3[0]});

Occurrence : 23

File Path:	l\a\a\d\e\a.java
Line	333. String format4 = String.format("http://%s.blob.core.windows.net/", new Object[]{G4[0]});

Ilustración 5 uso de http

File Path:	e1\a\a\b\b.java
Line	38. int nextInt = new Random().nextInt();

Occurrence : 12

File Path:	e1\a\a\b\g.java
Line	21. Random random = new Random();

Ilustración 6 Función insegura random

File Path:	com\google\android\gms\measurement\internal\zzkw.java
Line	293. MessageDigest instance = MessageDigest.getInstance("MD5");

Ilustración 7 Hash Inseguros

File Path:	com\google\crypto\tink\subtle\AesCmac.java
Line	25. Cipher instance = EngineFactory.CIPHER.getInstance("AES/ECB/NoPadding");

Ilustración 8 modo ecb inseguro

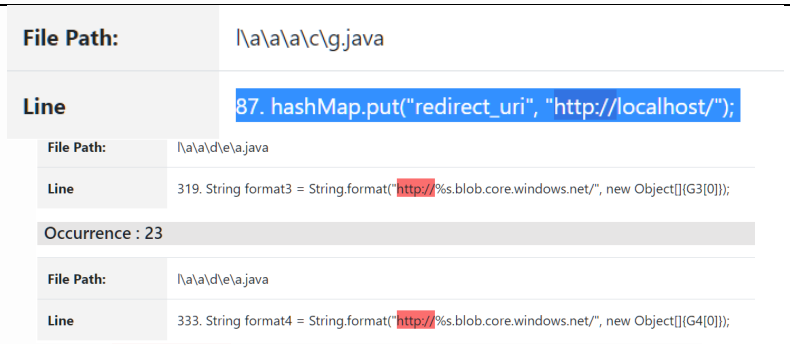
9.2. Resultados

Luego de las diferentes pruebas realizadas, se establecen los siguientes resultados

VULNERABILIDAD	TESTEADO	RESULTADO
Errores de inyección (SQLi, comandos de SO, LDAP, XPATH, entre otros). (6.5.1)	SI	No se encontró una vulnerabilidad asociada
Buffer overflow. (6.5.2)	SI	No se encontró una vulnerabilidad asociada
Almacenamiento y comunicaciones inseguras (no cifradas). (6.5.3 – 6.5.4)	SI	Se evidencias algoritmos débiles de cifrado.
Manejo inadecuado de errores. (6.5.5)	SI	No se encontró una vulnerabilidad asociada
Cross site scripting (XSS). (6.5.7)	SI	No se encontró una vulnerabilidad asociada
Acceso no controlado a archivos y carpetas. (6.5.8)	SI	No se encontró una vulnerabilidad asociada
Cross Site Request Forgery (CSRF). (6.5.9)	SI	No se encontró una vulnerabilidad asociada
Session hijacking / Session fixation. (6.5.10)	SI	No se encontró una vulnerabilidad asociada

9.3. Vulnerabilidades identificadas

ID	CONTACTAR-2020-APK-001	**Nivel**	Media
CVSS	5.3 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		
Descripción del hallazgo	**9.3.1. Código no ofuscado** Se identifico que el código del APK no se encuentra ofuscado, de tal manera que, al realizar una ingeniería inversa, se puede obtener en gran detalle, gran parte de la lógica de funcionamiento de la aplicación.		
Evidencia	**File Path:** res\layout\activity_comment.xml **Line** 8. <EditText android:textColor="@color/black" android:gravity="top" android:id="@+id/editText_Comment" android:background="@color/white" android:layout_width="match_parent" android:layout_height="203dp" android:layout_marginLeft="20dp" android:layout_marginTop="10dp" android:layout_marginRight="20dp" android:layout_marginBottom="10dp" android:ems="10" android:inputType="textMultiLine"/> *Ilustración 9 Código con la estructura de la API*		
Recomendación	• El código fuente debe ser ofuscado y hacer uso de Proguard para la definición de reglas.		
Referencias	https://cwe.mitre.org/data/definitions/649.html https://experto.dev/android-reducir-apk/ https://developer.android.com/studio/build/shrink-code?hl=es		

ID	CONTACTAR-2020-APK-002	Nivel	Media
CVSS	5.3 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		
Descripción del hallazgo	**9.3.2. Comunicaciones inseguras** Se encontró uso de conexiones que pueden ser inseguras, las cuales utilizan un protocolo no cifrado.		
Evidencia	 *Ilustración 10 conexiones HTTP*		
Recomendación	• Revisar las conexiones HTTP, y verificar si están enviando data sensible que pueda afectar la confidencialidad de la información. • Revisar la función redirect uri, ya que busca un recurso interno pero no se determina cual es.		
Referencias	N/A		

ID	CONTACTAR-2020-APK-003	Nivel	Media
CVSS	5.3 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		
Descripción del hallazgo	**9.3.3. Función Insegura Random** Se encuentra dentro del código el uso de la función insegura Random. File Path: c1\u\b.java Line 12. return new Random(); File Path: com\google\android\gms\internal\p000authapi\zzal.java		

	Line 9. public static final Random zzcv = new Random(); File Path: com\google\android\libraries\maps\gj\zzm.java Line 143. Random random = new Random(); File Path: com\google\android\libraries\maps\ik\zzau.java Line 9. public static final Random zzb = new Random(); File Path: com\google\android\libraries\maps\jw\zzd.java Line 71. Random random = new Random(); File Path: com\google\android\libraries\maps\mw\zzbr.java Line 57. public final Random zzs = new Random(); File Path: com\google\android\libraries\maps\mw\zzbw.java Line 9. public Random zza = new Random(); File Path: com\google\android\libraries\maps\mw\zzfj.java Line 30. public static Random zzw = new Random(); File Path: com\google\firebase\perf\internal\zzv.java Line 26. float nextFloat = new Random().nextFloat(); File Path: com\google\firebase\remoteconfig\RemoteConfigComponent.java Line 26. public static final Random DEFAULT_RANDOM = new Random(); File Path: e1\a\a\b\b.java Line 38. int nextInt = new Random().nextInt(); File Path: e1\a\a\b\g.java Line 21. Random random = new Random(); File Path: e1\a\a\h\i.java Line 62. Random random = new Random();

	File Path: l\i\a\a\l.java Line 9. public final Random e = new Random();
Evidencia	 *Ilustración 11 Función no segura*
Recomendación	• Para procesos que requieran proteger datos o accesos se debe hacer uso de funciones seguras como secure random.
Referencias	Random vs Secure Random numbers in Java - GeeksforGeeks

ID	CONTACTAR-2020-APK-004	**Nivel**	Media
CVSS	5.3 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		
Descripción del hallazgo	**9.3.4. Funcion Hash no segura** Se observa el uso inseguro de la función MD5 para procesos criptográficos. File Path: com\google\android\gms\measurement\internal\zzkw.java Line 293. MessageDigest instance = MessageDigest.getInstance("MD5"); File Path: com\google\android\libraries\maps\gu\zze.java Line 51. MessageDigest instance = MessageDigest.getInstance("MD5");		

	File Path: l\a\a\d\p\b.java Line 14. MessageDigest instance = MessageDigest.getInstance("MD5"); File Path: l\a\a\d\p\b.java Line 32. byte[] digest = MessageDigest.getInstance("MD5").digest(str.getBytes()); File Path: l\i\a\a\b0\b.java Line 170. this.h = MessageDigest.getInstance("MD5"); File Path: l\i\a\a\b0\u.java Line 38. MessageDigest instance = MessageDigest.getInstance("MD5"); File Path: l\i\a\a\c0\l.java Line 58. messageDigest = MessageDigest.getInstance("MD5"); File Path: l\i\a\a\c0\l.java Line 294. iVar2.d = MessageDigest.getInstance("MD5"); File Path: l\o\a\a0\k.java Line 164. return g1.b0.a.j(h.p(MessageDigest.getInstance("MD5").digest(str.getBytes("UTF-8"))));</td>
Evidencia	**File Path:** com\google\android\gms\measurement\internal\zz **Line** 293. MessageDigest instance = MessageDigest.getIn *Ilustración 12 USO de MD5*
Recomendación	• Hacer uso de funciones hash seguras como SHA 256
Referencias	N/A

ID	CONTACTAR-2020-APK-005	**Nivel**	Media
CVSS	5.3 AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N		

Descripción del hallazgo	**9.3.5. Cifrado inseguro ECB** Se identifico el uso de cifrado no seguro ECB
Evidencia	**File Path:** com\google\crypto\tink\subtle\AesCmac.java **Line** 25. Cipher instance = EngineFactory.CIPHER.getInstance("AES/ECB/NoPadding"); *Ilustración 13 Código con la estructura de la API*
Recomendación	• Cambiar el cipher inseguro por AES/GCM.
Referencias	N/A

10. Conclusiones

Basado en el estudio de las pruebas realizadas,

- Se evidencia buenas prácticas de seguridad en el desarrollo de la aplicación.
- Es posible obtener el código de la aplicación por ingeniería inversa.
- No se permite el uso de la aplicación en un dispositivo root.
- Se tiene control para prevenir el uso de dispositivos de análisis de tráfico.

11. Recomendaciones

- Revisar el uso de protocolo no seguros, y cifrado inseguro como http, MD5 para servicios de la aplicación que entreguen información confidencial.
- Se debe revisar el uso de funciones no seguras que puedan representar un riesgo para la aplicación.