



Informe de aplicación móvil

Este informe incluye información de seguridad importante acerca de la aplicación móvil.

Creado por:	IBM Application Security Analyzer - Mobile, Versión 1.1.4058 , Reglas: 1.1.4058
Nombre de escaneo:	formiik-android_6.4.3
Nombre de archivo de escaneo:	formiik-android_6.4.3.apk
Versión de aplicación:	228
Escaneo iniciado:	martes, 19 de junio de 2018 20:15:11 (UTC)
Sistema operativo:	Android
Modo de inicio de sesión:	Ninguno

Resumen de problemas de seguridad

Problemas de gravedad alta:	2
Problemas de gravedad media:	6
Problemas de gravedad baja:	3

Total de problemas de seguridad:	11
---	-----------

Tabla de contenido

Resumen

- Tipos de problemas
- Recomendaciones de arreglo
- Riesgos de seguridad
- OWASP Top 10

Problemas

- Cuelgue del código nativo (1)
- Se admiten las suites de cifrado SSL débiles (1)
- Generador de números aleatorios débiles (1)
- Filtración de información (3)
- Falta de fijación de certificados (2)
- Distintivo de copia de seguridad habilitado (1)
- Cuelgue del código Java (2)

Recomendaciones de arreglo

- No permita que se produzca un goteo de información confidencial.
- Valide la entrada de usuario
- No permita que la entrada de usuario se propague a las cadenas de formato.
- No utilice Generadores de números pseudoaleatorios (PRNG) débiles
- Evite las exportaciones de componentes de la aplicación que no sean necesarias.
- Establezca el atributo 'android:allowBackup' en false.
- Habilitar la fijación de certificados para esta conexión.
- Vuelva a configurar el servidor y la aplicación para evitar el uso de suites de cifrado débiles.

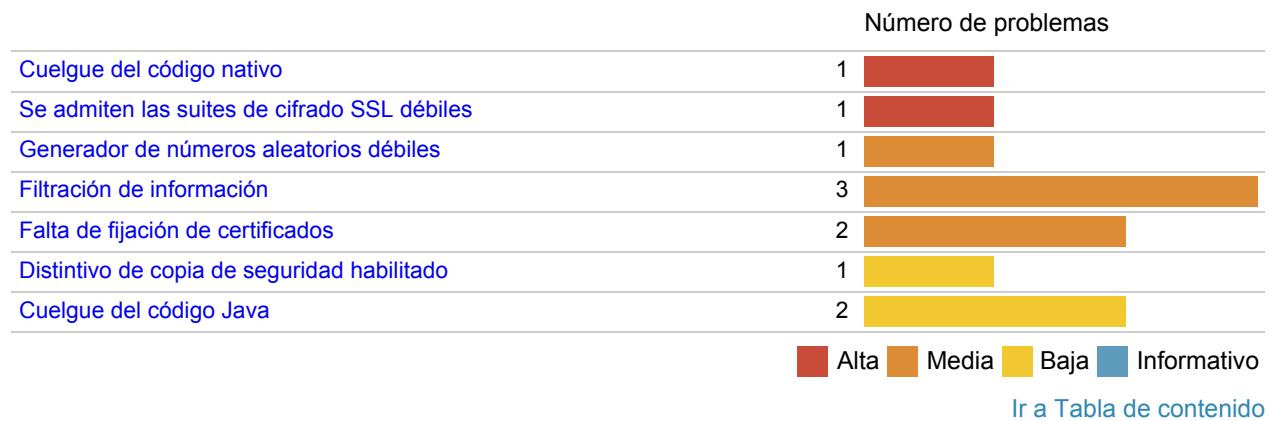
Cobertura

- Tipos de problemas
- Actividades

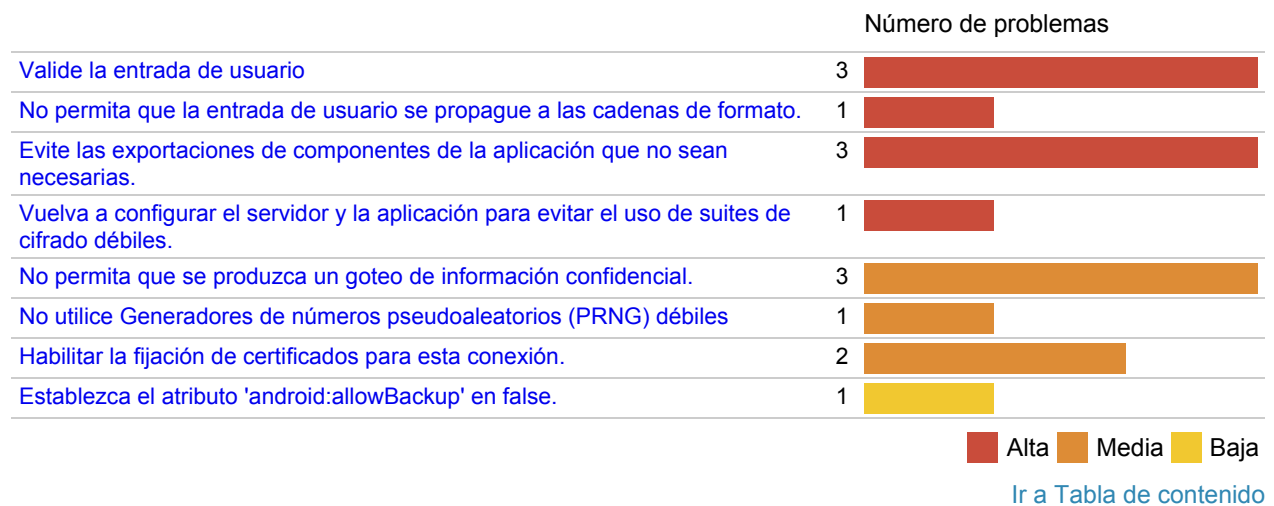
- Servicios
- Destinatarios

Resumen








Tipos de problemas: 7



Recomendaciones de arreglos: 8








Riesgos de seguridad: 7

	Número de problemas	
Una aplicación maliciosa puede hacer que la aplicación vulnerable deje de ser operativa, por ejemplo, DoS. Además, normalmente las excepciones nativas implican que la memoria está dañada, lo cual puede utilizarse para ejecutar código en el contexto de la aplicación vulnerable.	1	
Es posible robar o manipular la sesión y las cookies de un cliente, lo cual podría utilizarse para suplantar a un usuario legítimo, lo que permitiría al pirata informático visualizar o alterar registros de usuario y realizar transacciones como dicho usuario.	1	
El atacante puede predecir fácilmente los números aleatorios que genera esta aplicación. Esto puede permitir que mine la confidencialidad y/o la integridad de la aplicación vulnerable.	1	
Durante un ataque MitM o de acceso físico, los datos confidenciales del usuario están accesibles para un atacante.	3	
Si el atacante puede generar un certificado válido para el dominio de destino, por ejemplo, utilizando técnicas conocidas para instalar una Entidad emisora de certificados (CA) en el dispositivo, podrá suplantar al destino y descifrar tráfico, es decir, realizar un ataque Man-in-the-Middle. Esto puede provocar fugas de datos confidenciales, explotación de otras vulnerabilidades que, de otro modo, serían inaccesibles, o ataques de denegación de servicio.	2	
Un atacante malicioso puede minar la integridad y confidencialidad de la aplicación vulnerable realizando operaciones de copia de seguridad de ADB o de restauración de ADB.	1	
Una aplicación maliciosa puede hacer que la aplicación vulnerable deje de ser operativa, por ejemplo, DoS, o puede provocar que su estado sea imprevisto.	2	

 Alta  Media  Baja  Informativo

[Ir a Tabla de contenido](#)

OWASP Top 10

	Número de problemas	
M1: Uso inadecuado de la plataforma	0	
M2: Almacenamiento de datos no seguro	3	
M3: Comunicación no segura	3	
M4: Autenticación no segura	0	
M5: Criptografía insuficiente	1	
M6: Autorización no segura	-	
M7: Mala calidad del código	3	
M8: Manipulación indebida de código	0	
M9: Ingeniería inversa	0	
M10: Funcionalidad extraña	1	

[Ir a Tabla de contenido](#)

Problemas

A Cuelgue del código nativo 1

Problema 1 de 1

[Ir a Tabla de contenido](#)

Cuelgue del código nativo

Gravedad:	Alta
Sinopsis:	Se ha colgado la aplicación en el código nativo debido a que la validación de entrada es insuficiente o debido a una condición imprevista. Un cuelgue en el código nativo suele indicar que la memoria está dañada debido a un desbordamiento del almacenamiento intermedio o de enteros, una cadena de formato o un ataque de punteros en suspense.
Riesgo:	Una aplicación maliciosa puede hacer que la aplicación vulnerable deje de ser operativa, por ejemplo, DoS. Además, normalmente las excepciones nativas implican que la memoria está dañada, lo cual puede utilizarse para ejecutar código en el contexto de la aplicación vulnerable.
Causas:	Un atacante, mediante una aplicación maliciosa, puede hacer que la aplicación vulnerable se cuelgue en el código nativo.
X-Force:	93415
OWASP:	M7
Arreglo:	Valide la entrada de usuario No permita que la entrada de usuario se propague a las cadenas de formato. Evite las exportaciones de componentes de la aplicación que no sean necesarias.

Carga útil

Paquete de Intent:	formiik.com.mobiiik.www
Clase de Intent:	activities.ActivityFormiikURL
Acción de Intent:	android.intent.action.VIEW

Vuelco:

```
received crash request for pid 27933
found intercept fd 512 for pid 27933 and type kDebuggerdNativeBacktrace
registered intercept for pid 592 and type kDebuggerdNativeBacktrace
received crash request for pid 592
```

```

found intercept fd 512 for pid 592 and type kDebuggerdNativeBacktrace
registered intercept for pid 844 and type kDebuggerdNativeBacktrace
received crash request for pid 844
found intercept fd 512 for pid 844 and type kDebuggerdNativeBacktrace
registered intercept for pid 845 and type kDebuggerdNativeBacktrace
received crash request for pid 845
found intercept fd 512 for pid 845 and type kDebuggerdNativeBacktrace
registered intercept for pid 846 and type kDebuggerdNativeBacktrace
received crash request for pid 846
found intercept fd 512 for pid 846 and type kDebuggerdNativeBacktrace
registered intercept for pid 851 and type kDebuggerdNativeBacktrace
received crash request for pid 851
found intercept fd 512 for pid 851 and type kDebuggerdNativeBacktrace
registered intercept for pid 852 and type kDebuggerdNativeBacktrace
received crash request for pid 852
found intercept fd 512 for pid 852 and type kDebuggerdNativeBacktrace
crash socket received short read of length 0 (expected 12)
registered intercept for pid 861 and type kDebuggerdNativeBacktrace
received crash request for pid 861
found intercept fd 512 for pid 861 and type kDebuggerdNativeBacktrace
registered intercept for pid 865 and type kDebuggerdNativeBacktrace
received crash request for pid 865
found intercept fd 512 for pid 865 and type kDebuggerdNativeBacktrace
crash socket received short read of length 0 (expected 12)
registered intercept for pid 1343 and type kDebuggerdNativeBacktrace
received crash request for pid 1343

```

A Se admiten las suites de cifrado SSL débiles 1

Problema 1 de 1

[Ir a Tabla de contenido](#)

Se admiten las suites de cifrado SSL débiles

Gravedad:	Alta
Sinopsis:	La prueba revela si la aplicación da soporte a las suites de cifrado SSL que no ofrecen cifrado o utilizan algoritmos de cifrado débil. En estos casos, un atacante podría descifrar la comunicación segura entre el cliente y el servidor, o ejecutar satisfactoriamente un ataque "man-in-the-middle" (intermediario) en el cliente, lo que le permitirá visualizar información sensible y llevar a cabo acciones en nombre del cliente.
Riesgo:	Es posible robar o manipular la sesión y las cookies de un cliente, lo cual podría utilizarse para suplantar a un usuario legítimo, lo que permitiría al pirata informático visualizar o alterar registros de usuario y realizar transacciones como dicho usuario.
Causas:	El servidor web o el servidor de aplicaciones no está configurado de forma segura.
X-Force:	89156
OWASP:	M3
Arreglo:	Vuelva a configurar el servidor y la aplicación para evitar el uso de suites de cifrado débiles.

Carga útil

Paquete de Intent:	formiik.com.mobiik.www
Clase de Intent:	activities.ActivityFormiikURL
Acción de Intent:	android.intent.action.VIEW
Datos de Intent:	http://G18B/

Firma de método:

```
javax.net.ssl.SSLParameters.setCipherSuites(java.lang.String[]):void
```

Parámetros de validación de problema:

Nombre Valor

cipher Suites	[TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_256_CBC_SHA,TLS_EMPTY_RENEGOTIATION_INFO_SCSV]
---------------	---

Pila de llamadas:

Función	Clase	Línea
javax.net.ssl.SSLParameters.setCipherSuites(java.lang.String[]):void	SSLParameters	(javax/net/ssl/SSLParameters.java:140)
[Framework Code, 21 lines removed]	...	(...)
com.android.okhttp.internal.huc.HttpURLConnectionImpl.getOutputStream():java.io.OutputStream	HttpURLConnectionImpl	(com/android/okhttp/internal/huc/HttpURLConnectionImpl.java)
io.fabric.sdk.android.services.network.HttpRequest.h():io.fabric.sdk.android.services.network.HttpRequest	HttpRequest	(io/fabric/sdk/android/services/network/SourceFile:2450)
io.fabric.sdk.android.services.network.HttpRequest.i():io.fabric.sdk.android.services.network.HttpRequest	HttpRequest	(io/fabric/sdk/android/services/network/SourceFile:2463)
io.fabric.sdk.android.services.network.HttpRequest.b(java.lang.String, java.lang.String):io.fabric.sdk.android.services.network.HttpRequest	HttpRequest	(io/fabric/sdk/android/services/network/SourceFile:2542)
com.crashlytics.android.core.DefaultCreateReportSpiCall.applyMultipartDataTo(io.fabric.sdk.android.services.network.HttpRequest, com.crashlytics.android.core.Report):io.fabric.sdk.android.services.network.HttpRequest	DefaultCreateReportSpiCall	(com/crashlytics/android/core/SourceFile:3526)
com.crashlytics.android.core.DefaultCreateReportSpiCall.invoke(com.crashlytics.android.core.CreateReportRequest):boolean	DefaultCreateReportSpiCall	(com/crashlytics/android/core/SourceFile:61)

	11	
<code>com.crashlytics.android.core.ReportUploader.forceUpload(com.crashlytics.android.core.Report):boolean</code>	Report Upload er	(com\crashlytics \android\core\SourceFile:99)
<code>com.crashlytics.android.core.ReportUploader.Worker.attemptUploadWithRetry():void</code>	Report Upload er\$Worker	(com\crashlytics \android\core\SourceFile:230)
<code>com.crashlytics.android.core.ReportUploader.Worker.onRun():void</code>	Report Upload er\$Worker	(com\crashlytics \android\core\SourceFile:173)
<code>cjz.run():void</code>	cjz	(Unknown:30)
<code>java.lang.Thread.run():void</code>	Thread	(java\lang\Thread.java:764)

Problema 1 de 1

[Ir a Tabla de contenido](#)**Generador de números aleatorios débiles****Gravedad:****Media****Sinopsis:**

Esta prueba identifica una llamada a un Generador de números aleatorios que utiliza una semilla aleatoria predecible. No se debe utilizar este RNG cuando se implementen mecanismos de seguridad, ya que el atacante puede adivinar correctamente el número generado con un número de intentos relativamente bajo (como promedio). Esta vulnerabilidad es relevante sólo cuando un mecanismo criptográfico/de seguridad se basa en el generador de números aleatorios débil (RNG). En el caso de utilizar un RNG débil con fines no criptográficos, la decisión de aplicar los cambios recomendados está a discreción del desarrollador.

Riesgo:

El atacante puede predecir fácilmente los números aleatorios que genera esta aplicación. Esto puede permitir que mine la confidencialidad y/o la integridad de la aplicación vulnerable.

Causas:

La aplicación utiliza un Generador de números aleatorios no seguro (predecible).

X-Force:[93418](#)**OWASP:**[M5](#)**Arreglo:**[No utilice Generadores de números pseudoaleatorios \(PRNG\) débiles](#)**Carga útil****Paquete de Intent:**

formiik.com.mobiik.www

Clase de Intent:

background.Synchronization\$SyncBroadcastReceiver

Firma de método:

```
java.util.Random.nextDouble():double
```

Pila de llamadas:

Función

Clase

Línea

<code>java.util.Random.nextDouble():double</code>	Random	(java\util\Random.java:532)
<code>com.crashlytics.android.answers.RandomBackoff.randomJitter():double</code>	RandomBackoff	(com\crashlytics\android\answers\SourceFile:68)
<code>com.crashlytics.android.answers.RandomBackoff.getDelayMillis(int):long</code>	RandomBackoff	(com\crashlytics\android\answers\SourceFile:62)
<code>com.crashlytics.android.answers.RetryManager.canRetry(long):boolean</code>	RetryManager	(com\crashlytics\android\answers\SourceFile:1040)
<code>com.crashlytics.android.answers.AnswersRetryFilesSender.send(java.util.List):boolean</code>	AnswersRetryFilesSender	(com\crashlytics\android\answers\SourceFile:48)
<code>com.crashlytics.android.answers.EnabledSessionAnalyticsManagerStrategy.sendEvents():void</code>	EnabledSessionAnalyticsManagerStrategy	(com\crashlytics\android\answers\SourceFile:154)
<code>com.crashlytics.android.answers.AnswersEventsHandler\$3.run():void</code>	AnswersEventsHandler\$3	(com\crashlytics\android\answers\SourceFile:103)
<code>java.util.concurrent.Executors.RunnableAdapter.call():java.lang.Object</code>	Executors\$RunnableAdapter	(java\util\concurrent\Executors.java:457)
[Framework Code, 3 lines removed]	...	(...)
<code>java.util.concurrent.ThreadPoolExecutor.Worker.run():void</code>	ThreadPoolExecutor\$Worker	(java\util\concurrent\ThreadPoolExecutor.java:636)
<code>ckg.onRun():void</code>	ckg	(Unknown:75)
<code>cjz.run():void</code>	cjz	(Unknown:30)
<code>java.lang.Thread.run():void</code>	Thread	(java\lang\Thread.java:764)

Filtración de información

Gravedad:

Media

Sinopsis: Una vulnerabilidad de filtración de datos es un potencial para una violación de datos mediante: transmisiones anteriores a la filtración de datos, envío de datos confidenciales en texto sin formato, en movimiento (tráfico de red) y en descanso (almacenamiento de datos). Esta prueba detecta incidentes de filtración de datos: datos confidenciales revelados a un destino no autorizado mediante una intención maliciosa o un error. Los datos confidenciales podrían ser información del usuario o del dispositivo, geolocalización, credenciales de usuario, información de patentes o financiera, datos de la tarjeta de crédito, ID del dispositivo, contactos de usuario y otra información privada.

Riesgo: Durante un ataque MitM o de acceso físico, los datos confidenciales del usuario están accesibles para un atacante.

Causas: Los datos confidenciales del usuario o del dispositivo se enviarán a través de métodos no seguros.

X-Force: None

OWASP: M2

Arreglo: No permita que se produzca un goteo de información confidencial.

Carga útil

Paquete de Intent: formiik.com.mobiik.www

Clase de Intent: com.soundcloud.android.crop.CropImageActivity

Origen

Firma de método:

```
android.provider.Settings.Secure.getString(android.content.ContentResolver, java.lang.String):java.lang.String
```

Parámetros de validación de problema:

Nombre	Valor
<return>	079920b99f1285a7

Pila de llamadas:

Función	Clase	Línea
android.provider.Settings.Secure.getString(android.content.ContentResolver, java.lang.String):java.lang.String	Settings\$Secure	(android\provider\Settings.java:4654)
io.fabric.sdk.android.services.common.CommonUtils	CommonUtils	(io\fabric\sdk\androi

s.e(android.content.Context):boolean		d\services\common\SourceFile:538)
io.fabric.sdk.android.services.common.CommonUtils.f(android.content.Context):boolean	CommonUtils	(io\fabric\sdk\android\services\common\SourceFile:549)
com.crashlytics.android.core.CrashlyticsController.writeSessionOS(java.lang.String):void	CrashlyticsController	(com\crashlytics\android\core\SourceFile:1024)
com.crashlytics.android.core.CrashlyticsController.doOpenSession():void	CrashlyticsController	(com\crashlytics\android\core\SourceFile:512)
com.crashlytics.android.core.CrashlyticsController.access.500(com.crashlytics.android.core.CrashlyticsController):void	CrashlyticsController	(com\crashlytics\android\core\SourceFile:59)
com.crashlytics.android.core.CrashlyticsController.11.call():java.lang.Void	CrashlyticsController\$11	(com\crashlytics\android\core\SourceFile:418)
com.crashlytics.android.core.CrashlyticsController.11.call():java.lang.Object	CrashlyticsController\$11	(com\crashlytics\android\core\SourceFile:415)
com.crashlytics.android.core.CrashlyticsBackgroundWorker.2.call():java.lang.Object	CrashlyticsBackgroundWorker\$2	(com\crashlytics\android\core\SourceFile:99)
java.util.concurrent.FutureTask.run():void	FutureTask	(java\util\concurrent\FutureTask.java:266)
java.util.concurrent.ThreadPoolExecutor.runWorker(java.util.concurrent.ThreadPoolExecutor.Worker):void	ThreadPoolExecutor	(java\util\concurrent\ThreadPoolExecutor.java:1162)
java.util.concurrent.ThreadPoolExecutor.Worker.run():void	ThreadPoolExecutor\$Worker	(java\util\concurrent\ThreadPoolExecutor.java:636)
ckg.onRun():void	ckg	(Unknown:75)
cjz.run():void	cjz	(Unknown:30)
java.lang.Thread.run():void	Thread	(java\lang\Thread.java:764)

Receptor

Firma de método:

```
java.io.FileOutputStream.write(byte[], int, int):void
```

Parámetros de validación de problema:

Nombre Valor

```
b J 079920b99f1285a7 "AOSP on walleye(0E 8E|çN PZ(g$3baf6ee2-dc2
7-4d82-a757-567c7ea59ebdZd079920b99f1285a7`jGoogler@aosp_walleye
*****
*****
```

.....

Pila de llamadas:

Función	Clase	Línea
java.io.FileOutputStream.write(byte[], int, int) :void	FileOutputS tream	(java\io\FileOutputSt ream.java:322)
com.crashlytics.android.core.CodedOutputStream.r efreshBuffer():void	CodedOutput Stream	(com\crashlytics\andr oid\core\SourceFile:6 68)

com.crashlytics.android.core.CodedOutputStream.flush():void	CodedOutputStream	(com\crashlytics\android\core\SourceFile:678)
io.fabric.sdk.android.services.common.CommonUtils.a(java.io.Flushable, java.lang.String):void	CommonUtils	(io\fabric\sdk\android\services\common\SourceFile:722)
com.crashlytics.android.core.CrashlyticsController.writeSessionDevice(java.lang.String):void	CrashlyticsController	(com\crashlytics\android\core\SourceFile:1059)
com.crashlytics.android.core.CrashlyticsController.doOpenSession():void	CrashlyticsController	(com\crashlytics\android\core\SourceFile:513)
com.crashlytics.android.core.CrashlyticsController.access.500(com.crashlytics.android.core.CrashlyticsController):void	CrashlyticsController	(com\crashlytics\android\core\SourceFile:59)
com.crashlytics.android.core.CrashlyticsController.11.call():java.lang.Void	CrashlyticsController\$11	(com\crashlytics\android\core\SourceFile:418)
com.crashlytics.android.core.CrashlyticsController.11.call():java.lang.Object	CrashlyticsController\$11	(com\crashlytics\android\core\SourceFile:415)
com.crashlytics.android.core.CrashlyticsBackgroundWorker.2.call():java.lang.Object	CrashlyticsBackgroundWorker\$2	(com\crashlytics\android\core\SourceFile:99)
java.util.concurrent.FutureTask.run():void	FutureTask	(java\util\concurrent\FutureTask.java:266)
java.util.concurrent.ThreadPoolExecutor.runWorker(java.util.concurrent.ThreadPoolExecutor.Worker):void	ThreadPoolExecutor	(java\util\concurrent\ThreadPoolExecutor.java:1162)
java.util.concurrent.ThreadPoolExecutor.Worker.run():void	ThreadPoolExecutor\$Worker	(java\util\concurrent\ThreadPoolExecutor.java:636)
ckg.onRun():void	ckg	(Unknown:75)
cjz.run():void	cjz	(Unknown:30)
java.lang.Thread.run():void	Thread	(java\lang\Thread.java:764)

Problema 2 de 3

[Ir a Tabla de contenido](#)

Filtración de información

Gravedad:

Media

Sinopsis: Una vulnerabilidad de filtración de datos es un potencial para una violación de datos mediante: transmisiones anteriores a la filtración de datos, envío de datos confidenciales en texto sin formato, en movimiento (tráfico de red) y en descanso (almacenamiento de datos). Esta prueba detecta incidentes de filtración de datos: datos confidenciales revelados a un destino no autorizado mediante una intención maliciosa o un error. Los datos confidenciales podrían ser información del usuario o del dispositivo, geolocalización, credenciales de usuario, información de patentes o financiera, datos de la tarjeta de crédito, ID del dispositivo, contactos de usuario y otra información privada.

Riesgo: Durante un ataque MitM o de acceso físico, los datos confidenciales del usuario están accesibles para un atacante.

Causas: Los datos confidenciales del usuario o del dispositivo se enviarán a través de métodos no seguros.

X-Force: None

OWASP: M2

Arreglo: No permita que se produzca un goteo de información confidencial.

Carga útil

Paquete de Intent: formiik.com.mobiik.www

Clase de Intent: background.ServiceGoogleCloudMessages\$GcmIntentService

Origen

Firma de método:

```
android.provider.Settings.Secure.getString(android.content.ContentResolver, java.lang.String):java.lang.String
```

Parámetros de validación de problema:

Nombre	Valor
<return>	079920b99f1285a7

Pila de llamadas:

Función	Clase	Línea
android.provider.Settings.Secure.getString(android.content.ContentResolver, java.lang.String):java.lang.String	Settings\$Secure	(android\provider\Settings.java:4654)
io.fabric.sdk.android.services.common.IdManager.g	IdManager	(io\fabric\sdk\androi

() : java.lang.String		d\services\common\SourceFile:344)
io.fabric.sdk.android.services.common.IdManager.c () : java.util.Map	IdManager	(io\fabric\sdk\android\services\common\SourceFile:289)
com.crashlytics.android.answers.SessionMetadataCollector.getMetadata() : com.crashlytics.android.answers.SessionEventMetadata	SessionMetadataCollector	(com\crashlytics\android\answers\SourceFile:35)
com.crashlytics.android.answers.AnswersEventsHandler.4.run() : void	AnswersEventsHandler\$4	(com\crashlytics\android\answers\SourceFile:119)
java.util.concurrent.Executors.RunnableAdapter.call() : java.lang.Object	Executors\$RunnableAdapter	(java\util\concurrent\Executors.java:457)
[Framework Code, 3 lines removed]	...	(...)
java.util.concurrent.ThreadPoolExecutor.Worker.run() : void	ThreadPoolExecutor\$Worker	(java\util\concurrent\ThreadPoolExecutor.java:636)
ckg.onRun() : void	ckg	(Unknown:75)
cjz.run() : void	cjz	(Unknown:30)
java.lang.Thread.run() : void	Thread	(java\lang\Thread.java:764)

Receptor

Firma de método:

```
java.io.FileOutputStream.write(byte[], int, int):void
```

Parámetros de validación de problema:

Nombre Valor

```
b "Crashlytics Android SDK/2.3.17.dev#5B2976C000C3-0001-607F-F470353968AE;D
DT YX(X` :t formiik.com.mobiik.www2286.4.3** (e47cb00dd8d1f47f0d
1db72d2e1728f467b58a0e2 96fe747b69a046a8abf50a75f85b97a2PBB8.0.0REL
J 079920b99f1285a7"AOSP on walleye(0E 8E|çNPPZdd079920b99f1
285a7Z(g$3baf6ee2-dc27-4d82-a757-567c7ea59ebd`jGoogler@aosp_walleyeR©
n¥crash" "© % mainH*android.app.ActivityThread$H.handleMes
sageActivityThread.java 8) v"android.os.Handler.dispatchMessageHan
dler.java i(,android.os.Looper.loopLooper.java =(android.ap
p.ActivityThread.mainActivityThread.java 3(2java.lang.reflect.Meth
od.invokeMethod.java (L6com.android.internal.os.Zygote$MethodAndArgsC
aller.runZygote.java 1(A'com.android.internal.os.ZygoteInit.mainZ
ygoteInit.java „( Thread-16(java.lang.Object.waitObject.java(
//java.lang.Thread.parkFor$Thread.java (()*sun.misc.Unsafe.park
Unsafe.java 1(F+java.util.concurrent.locks.LockSupport.parkLockS
upport.java %((uKjava.util.concurrent.locks.AbstractQueuedSynchronizer
$ConditionObject.awaitAbstractQueuedSynchronizer.java <(T/ java.util
.concurrent.PriorityBlockingQueue.takePriorityBlockingQueue.java =(
hv.run SourceFile Z( Answers Events Handler1Q=com.crashlytics
.android.answers.AnswersRetryFilesSender.build SourceFile (s_com.cra
```

```

shlytics.android.answers.EnabledSessionAnalyticsManagerStrategy.setAnalytic
sSettingsData SourceFile B(N:com.crashlytics.android.answers.AnswersE
ventsHandler$1.run SourceFile F(L3java.util.concurrent.Executors$Runn
ableAdapter.callExecutors.java(=#java.util.concurrent.FutureTask.
runFutureTask.java ((sHjava.util.concurrent.ScheduledThreadPoolExec
utor$ScheduledFutureTask.run ScheduledThreadPoolExecutor.java (S1jav
a.util.concurrent.ThreadPoolExecutor.runWorkerThreadPoolExecutor.java (
T2java.util.concurrent.ThreadPoolExecutor$Worker.runThreadPoolExecut
or.java ((ckg.onRun SourceFile K(cjz.run SourceFile (*
java.lang.Thread.runThread.java ( 1 Queue(java.lang.Object
.waitObject.java (/java.lang.Thread.parkFor$Thread.java (*
sun.misc.Unsafe.parkUnsafe.java T(F+java.util.concurrent.locks.Lock
Support.parkLockSupport.java uKjava.util.concurrent.locks.Abstrac
tQueuedSynchronizer$ConditionObject.awaitAbstractQueuedSynchronizer.java
<(T/java.util.concurrent.PriorityBlockingQueue.takePriorityBlocking
Queue.java (p[io.fabric.sdk.android.services.concurrency.DependencyP
riorityBlockingQueue.performOperation SourceFile (cNio.fabric.sdk.a
ndroid.services.concurrency.DependencyPriorityBlockingQueue.get SourceFile
? (cOio.fabric.sdk.android.services.concurrency.DependencyPriorityBloc
kingQueue.take SourceFile A (cOio.fabric.sdk.android.services.concurre
ncy.DependencyPriorityBlockingQueue.take SourceFile . (Q/java.util.con
current.ThreadPoolExecutor.getTaskThreadPoolExecutor.java ; (S1java.
util.concurrent.ThreadPoolExecutor.runWorkerThreadPoolExecutor.java ((
T2java.util.concurrent.ThreadPoolExecutor$Worker.runThreadPoolExecutor
.java ((*java.lang.Thread.runThread.java ( Thread-12((
java.lang.Object.waitObject.java (/java.lang.Thread.parkFor$Thread
.java ((*sun.misc.Unsafe.parkUnsafe.java T(F+java.util.concu
rrent.locks.LockSupport.parkLockSupport.java uKjava.util.concurre
nt.locks.AbstractQueuedSynchronizer$ConditionObject.awaitAbstractQueuedSy
nchronizer.java <(T/java.util.concurrent.PriorityBlockingQueue.take
PriorityBlockingQueue.java (ic.run SourceFile \( - Queue((
java.lang.Object.waitObject.java (/java.lang.Thread.parkFor$Threa
d.java ((*sun.misc.Unsafe.parkUnsafe.java T(F+java.util.conc
urrent.locks.LockSupport.parkLockSupport.java uC)java.util.concurr
ent.FutureTask.awaitDoneFutureTask.java(=#java.util.concurrent.Fu
tureTask.getFutureTask.java( Z

```

Pila de llamadas:

Función	Clase	Línea
java.io.FileOutputStream.write(byte[], int, int):void	FileOutp utStream	(java\io\FileOutpu tStream.java:322)
com.crashlytics.android.core.CodedOutputStream.refreshBuffer():void	CodedOut putStrea m	(com\crashlytics\a ndroid\core\Source File:668)
com.crashlytics.android.core.CodedOutputStream.writeRawBytes(byte[], int, int):void	CodedOut putStrea m	(com\crashlytics\a ndroid\core\Source File:763)
com.crashlytics.android.core.CodedOutputStream.writeRawBytes(byte[]):void	CodedOut putStrea m	(com\crashlytics\a ndroid\core\Source File:745)
com.crashlytics.android.core.CrashlyticsController.copyToCodedOutputStream(java.io.InputStream, com.crashlytics.android.core.CodedOutputStream, int):void	Crashlyt icsContr oller	(com\crashlytics\a ndroid\core\Source File:1310)
com.crashlytics.android.core.CrashlyticsController.writeToCosFromFile(com.crashlytics.android.core.CodedOutputStream, java.io.File):void	Crashlyt icsContr oller	(com\crashlytics\a ndroid\core\Source File:1293)
com.crashlytics.android.core.CrashlyticsController.synthesizeSessionFile(java.io.File, java.lang.String, jav	Crashlyt icsContr	(com\crashlytics\a ndroid\core\Source

<code>a.io.File[], java.io.File):void</code>	oller	File:1218)
<code>com.crashlytics.android.core.CrashlyticsController.writeSessionPartsToSessionFile(java.io.File, java.lang.String, int):void</code>	CrashlyticsController	(com\crashlytics\android\core\SourceFile:1173)
<code>com.crashlytics.android.core.CrashlyticsController.closeOpenSessions(java.io.File[], int, int):void</code>	CrashlyticsController	(com\crashlytics\android\core\SourceFile:568)
<code>com.crashlytics.android.core.CrashlyticsController.closeSessions(cmv, boolean):void</code>	CrashlyticsController	(com\crashlytics\android\core\SourceFile:550)
<code>com.crashlytics.android.core.CrashlyticsController.closeSessions(cmv):void</code>	CrashlyticsController	(com\crashlytics\android\core\SourceFile:518)
<code>com.crashlytics.android.core.CrashlyticsController.call():java.lang.Void</code>	CrashlyticsController\$6	(com\crashlytics\android\core\SourceFile:298)
<code>com.crashlytics.android.core.CrashlyticsController.call():java.lang.Object</code>	CrashlyticsController\$6	(com\crashlytics\android\core\SourceFile:285)
<code>java.util.concurrent.FutureTask.run():void</code>	FutureTask	(java\util\concurrent\FutureTask.java:266)
<code>java.util.concurrent.ThreadPoolExecutor.runWorker(java.util.concurrent.ThreadPoolExecutor.Worker):void</code>	ThreadPoolExecutor	(java\util\concurrent\ThreadPoolExecutor.java:1162)
<code>java.util.concurrent.ThreadPoolExecutor.Worker.run():void</code>	ThreadPoolExecutor\$Worker	(java\util\concurrent\ThreadPoolExecutor.java:636)
<code>ckg.onRun():void</code>	ckg	(Unknown:75)
<code>cjz.run():void</code>	cjz	(Unknown:30)
<code>java.lang.Thread.run():void</code>	Thread	(java\lang\Thread.java:764)

Problema 3 de 3

[Ir a Tabla de contenido](#)

Filtración de información

Gravedad:	Media
Sinopsis:	Una vulnerabilidad de filtración de datos es un potencial para una violación de datos mediante: transmisiones anteriores a la filtración de datos, envío de datos confidenciales en texto sin formato, en movimiento (tráfico de red) y en descanso (almacenamiento de datos). Esta prueba detecta incidentes de filtración de datos: datos confidenciales revelados a un destino no autorizado mediante una intención maliciosa o un error. Los datos confidenciales podrían ser información del usuario o del dispositivo, geolocalización, credenciales de usuario, información de patentes o financiera, datos de la tarjeta de crédito, ID del dispositivo, contactos de usuario y otra información privada.
Riesgo:	Durante un ataque MitM o de acceso físico, los datos confidenciales del usuario están accesibles para un atacante.
Causas:	Los datos confidenciales del usuario o del dispositivo se enviarán a través de métodos no seguros.
X-Force:	None
OWASP:	M2
Arreglo:	No permita que se produzca un goteo de información confidencial.
Carga útil	
Paquete de Intent:	formiik.com.mobiik.www
Clase de Intent:	com.commonsware.cwac.wakeful.AlarmReceiver

Origen

Firma de método:

```
android.telephony.TelephonyManager.getIdentity() : java.lang.String
```

Parámetros de validación de problema:

Nombre	Valor
<return>	357537088137435

Pila de llamadas:

Función	Clase	Línea
android.telephony.TelephonyManager.getIdentity() : java.lang.String	TelephonyManager	(android\telephony\TelephonyManager.java:999)
cdv.c(android.content.Context) : java.lang.String	cdv	(Unknown:168)
cee.b(android.content.Context, java.lang.String)	cee	(Unknown:99)

ng):boolean		
formiik.com.mobiik.www.globals.Formiik.onCreate():void	Formiik	(formiik\com\mobiik\www\globals\SourceFile:5680)
android.app.Instrumentation.callApplicationOnCreate(android.app.Application):void	Instrumentation	(android\app\Instrumentation.java:1118)
[Framework Code, 8 lines removed]	...	(...)
com.android.internal.os.ZygoteInit.main(java.lang.String[]):void	ZygoteInit	(com\android\internal\os\ZygoteInit.java:772)

Receptor

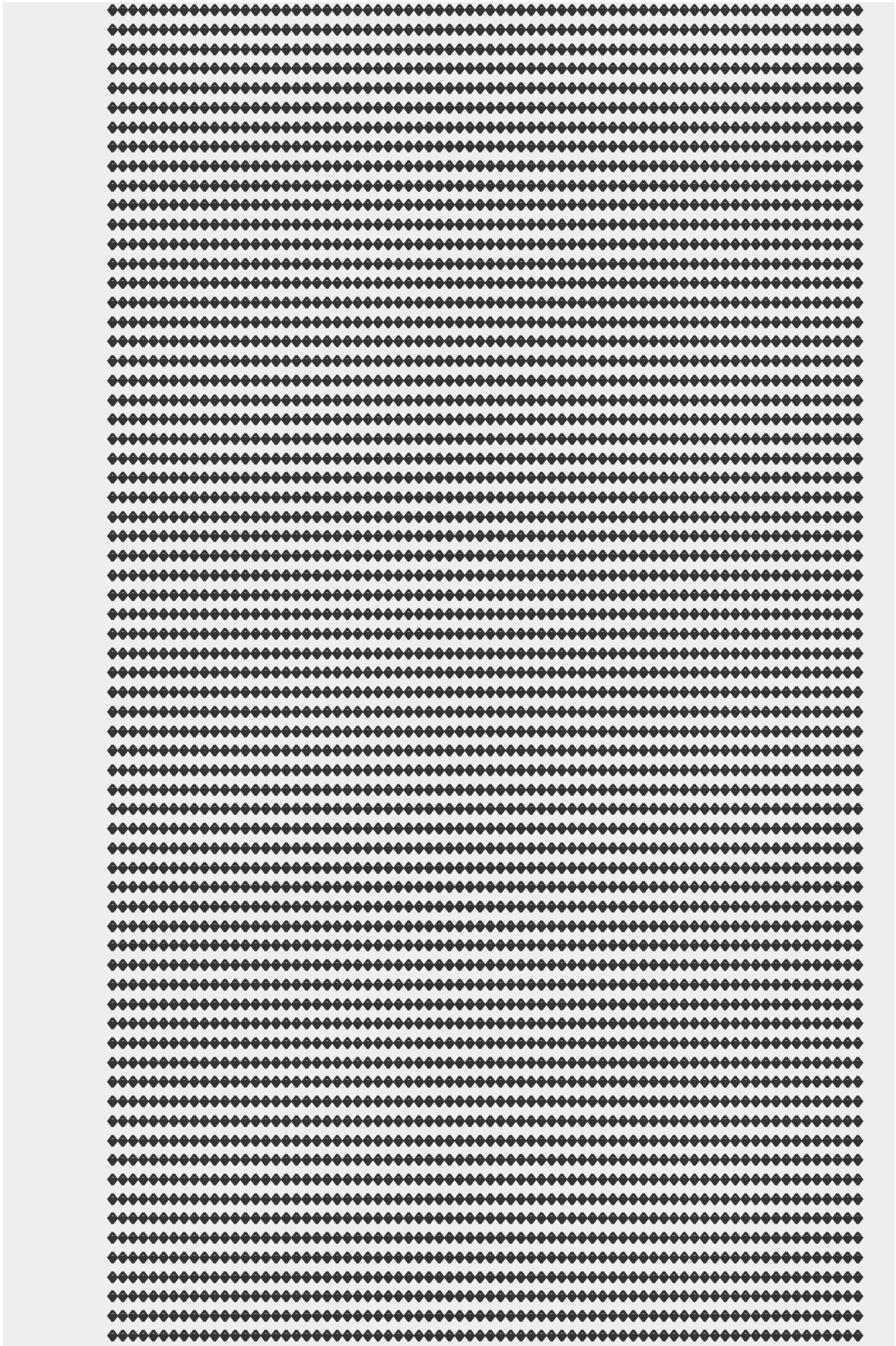
Firma de método:

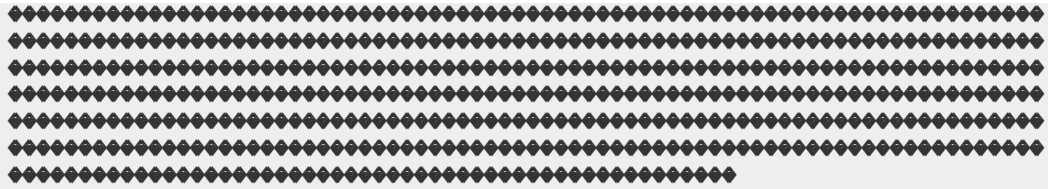
```
java.io.FileOutputStream.write(byte[], int, int):void
```

Parámetros de validación de problema:

Nombre	Valor
--------	-------

b	357537088137435
---	-----------------





Pila de llamadas:

Función	Clase	Línea
java.io.FileOutputStream.write(byte[], int, int):void	FileOutputStream	(java\io\FileOutputStream.java:322)
sun.nio.cs.StreamEncoder.writeBytes():void	StreamEncoder	(sun\nio\cs\StreamEncoder.java:221)
sun.nio.cs.StreamEncoder.implFlushBuffer():void	StreamEncoder	(sun\nio\cs\StreamEncoder.java:291)
sun.nio.cs.StreamEncoder.implFlush():void	StreamEncoder	(sun\nio\cs\StreamEncoder.java:295)
sun.nio.cs.StreamEncoder.flush():void	StreamEncoder	(sun\nio\cs\StreamEncoder.java:141)
java.io.OutputStreamWriter.flush():void	OutputStreamWriter	(java\io\OutputStreamWriter.java:229)
cee.b(android.content.Context, java.lang.String):boolean	cee	(Unknown:100)
formiik.com.mobiik.www.globals.Formiik.onCreate():void	Formiik	(formiik\com\mobiik\www\globals\SourceFile:5680)
android.app.Instrumentation.callApplicationOnCreate(android.app.Application):void	Instrumentation	(android\app\Instrumentation.java:1118)
[Framework Code, 8 lines removed]	...	(...)
com.android.internal.os.ZygoteInit.main(java.lang.String[]):void	ZygoteInit	(com\android\internal\os\ZygoteInit.java:772)

Falta de fijación de certificados

Gravedad:	Media
Sinopsis:	Esta prueba revisa si se ha habilitado y utilizado la Fijación de certificados para una conexión de transporte segura.
Riesgo:	Si el atacante puede generar un certificado válido para el dominio de destino, por ejemplo, utilizando técnicas conocidas para instalar una Entidad emisora de certificados (CA) en el dispositivo, podrá suplantar al destino y descifrar tráfico, es decir, realizar un ataque Man-in-the-Middle. Esto puede provocar fugas de datos confidenciales, explotación de otras vulnerabilidades que, de otro modo, serían inaccesibles, o ataques de denegación de servicio.
Causas:	La fijación de certificados está inhabilitada/no se ha implementado para esta conexión.
X-Force:	None
OWASP:	M3
Arreglo:	Habilitar la fijación de certificados para esta conexión.

Carga útil

Paquete de Intent:	formiik.com.mobiik.www
Clase de Intent:	background.Synchronization\$SyncBroadcastReceiver

Firma de método:

```
java.net.URL.openConnection():java.net.URLConnection
```

Parámetros de validación de problema:

Nombre	Valor
this	https://e.crashlytics.com/spi/v2/events

Pila de llamadas:

Función	Clase	Línea
java.net.URL.openConnection():java.net.URLConnection	URL	(java\net\URL.java:992)
clx.a(java.net.URL):java.net.HttpURLConnection	clx	(Unknown:315)
io.fabric.sdk.android.services.network.HttpRequest.d():java.net.HttpURLConnection	HttpRequest	(io\fabric\sdk\android\services\network\SourceFile:1298)
io.fabric.sdk.android.services.network.HttpRequest.a():java.net.HttpURLConnection	HttpRequest	(io\fabric\sdk\android\services\network\SourceFile:1318)

cjr.getRequest(java.util.Map):io.fabric.sdk.android.services.network.HttpRequest	cjr	(Unknown:3104)
cjr.getRequest():io.fabric.sdk.android.services.network.HttpRequest	cjr	(Unknown:117)
com.crashlytics.android.answers.SessionAnalyticsFilesSender.send(java.util.List):boolean	SessionAnalyticsFilesSender	(com\crashlytics\android\answers\SourceFile:34)
com.crashlytics.android.answers.AnswersRetryFilesSender.send(java.util.List):boolean	AnswersRetryFilesSender	(com\crashlytics\android\answers\SourceFile:49)
com.crashlytics.android.answers.EnabledSessionAnalyticsManagerStrategy.sendEvents():void	EnabledSessionAnalyticsManagerStrategy	(com\crashlytics\android\answers\SourceFile:154)
com.crashlytics.android.answers.AnswersEventsHandler.3.run():void	AnswersEventsHandler\$3	(com\crashlytics\android\answers\SourceFile:103)
java.util.concurrent.Executors\$RunnableAdapter.call():java.lang.Object	Executors\$RunnableAdapter	(java\util\concurrent\Executors.java:457)
[Framework Code, 3 lines removed]	...	(...)
java.util.concurrent.ThreadPoolExecutor\$Worker.run():void	ThreadPoolExecutor\$Worker	(java\util\concurrent\ThreadPoolExecutor.java:636)
ckg.onRun():void	ckg	(Unknown:75)
cjz.run():void	cjz	(Unknown:30)
java.lang.Thread.run():void	Thread	(java\lang\Thread.java:764)

Problema 2 de 2

[Ir a Tabla de contenido](#)

Falta de fijación de certificados	
Gravedad:	Media
Sinopsis:	Esta prueba revisa si se ha habilitado y utilizado la Fijación de certificados para una conexión de transporte segura.
Riesgo:	Si el atacante puede generar un certificado válido para el dominio de destino, por ejemplo, utilizando técnicas conocidas para instalar una Entidad emisora de certificados (CA) en el dispositivo, podrá suplantar al destino y descifrar tráfico, es decir, realizar un ataque Man-in-the-Middle. Esto puede provocar fugas de datos confidenciales, explotación de otras vulnerabilidades que, de otro modo, serían inaccesibles, o ataques de denegación de servicio.
Causas:	La fijación de certificados está inhabilitada/no se ha implementado para esta conexión.
X-Force:	None
OWASP:	M3
Arreglo:	Habilitar la fijación de certificados para esta conexión.
Carga útil	
Paquete de Intent:	formiik.com.mobiik.www
Clase de Intent:	com.hp.mss.hpprint.activity.PrintPluginManagerActivity

Firma de método:

```
java.net.URL.openConnection():java.net.URLConnection
```

Parámetros de validación de problema:

Nombre	Valor
this	https://print-metrics-w1.twosmiles.com/api/v2/events

Pila de llamadas:

Función	Clase	Línea
java.net.URL.openConnection():java.net.URLConnection	URL	(java\net\URL.java:992)
ir.createConnection(java.net.URL):java.net.HttpURLConnection	ir	(Unknown:169)
ir.openConnection(java.net.URL, com.android.volley.Request):java.net.HttpURLConnection	ir	(Unknown:179)
ir.performRequest(com.android.volley.Request, java.util.Map):org.apache.http.HttpResponse	ir	(Unknown:103)
ik.a(com.android.volley.Request):com.android.volley.NetworkResponse	ik	(Unknown:97)
ic.run():void	ic	(Unknown:114)

Problema 1 de 1

[Ir a Tabla de contenido](#)**Distintivo de copia de seguridad habilitado****Gravedad:** Baja

Sinopsis: El distintivo 'android:allowBackup' del archivo manifest de APK controla si la aplicación puede estar implicada en las operaciones de copia de seguridad y restauración de ADB. Habilitar este distintivo es peligroso, ya que un atacante malicioso podrá acceder a los datos de aplicación utilizando el mecanismo de copia de seguridad de ADB o podrá extraerlo del archivo de copia de seguridad creado previamente.

Riesgo: Un atacante malicioso puede minar la integridad y confidencialidad de la aplicación vulnerable realizando operaciones de copia de seguridad de ADB o de restauración de ADB.

Causas: La aplicación establece el indicador allowBackup en 'true', o no lo establece. (El valor predeterminado de este parámetro es true).

X-Force: [None](#)

OWASP: [M10](#)

Arreglo: [Establezca el atributo 'android:allowBackup' en false.](#)

Manifiesto:

```
XML:
<application android:icon="@formiik.com.mobiik.www:drawable/ic_launcher"
    android:label="@formiik.com.mobiik.www:string/app_formiik_name"
    android:largeHeap="true"
    android:name="formiik.com.mobiik.www.globals.Formiik"
    android:roundIcon="@formiik.com.mobiik.www:mipmap/ic_launcher"
    android:theme="@formiik.com.mobiik.www:style/ftd">
```

Problema 1 de 2

[Ir a Tabla de contenido](#)

Cuelgue del código Java

Gravedad:	Baja
Sinopsis:	Se ha colgado la aplicación en el código Java debido a que la validación de entrada es insuficiente o debido a una condición imprevista. Esta situación la puede reproducir una aplicación maliciosa para realizar un ataque de denegación de servicio local (DoS).
Riesgo:	Una aplicación maliciosa puede hacer que la aplicación vulnerable deje de ser operativa, por ejemplo, DoS, o puede provocar que su estado sea imprevisto.
Causas:	Un atacante, mediante una aplicación maliciosa, puede hacer que la aplicación vulnerable se cuelgue en el código Java.
X-Force:	93408
OWASP:	M7
Arreglo:	Valide la entrada de usuario Evite las exportaciones de componentes de la aplicación que no sean necesarias.
Carga útil	
Paquete de Intent:	formiik.com.mobiik.www
Clase de Intent:	activities.ActivityFormiikURL
Acción de Intent:	android.intent.action.VIEW

Firma de método:

```
formiik.com.mobiik.www.activities.ActivityFormiikURL.onCreate
```

Parámetros de validación de problema:

Nombre Valor

```
java.lang.RuntimeException: Unable to start activity ComponentInfo{formiik.com.mobiik.www/formiik.com.mobiik.www.activities.ActivityFormiikURL}: java.lang.NullPointerException: Attempt to invoke virtual method 'java.util.List android.net.Uri.getPathSegments()' on a null object reference filterResults=[]>
```

Pila de llamadas:

Función	Clase	Línea
formiik.com.mobiik.www.activities.ActivityFormiikURL.onCreate	ActivityFormiikURL	(Unknown:45)
android.app.Activity.performCreate	Activity	(Activity.java:6975)
[Framework Code, 10 lines removed]	...	(...)

Problema 2 de 2

[Ir a Tabla de contenido](#)

Cuelgue del código Java

Gravedad:

Baja

Sinopsis:

Se ha colgado la aplicación en el código Java debido a que la validación de entrada es insuficiente o debido a una condición imprevista. Esta situación la puede reproducir una aplicación maliciosa para realizar un ataque de denegación de servicio local (DoS).

Riesgo:

Una aplicación maliciosa puede hacer que la aplicación vulnerable deje de ser operativa, por ejemplo, DoS, o puede provocar que su estado sea imprevisto.

Causas:

Un atacante, mediante una aplicación maliciosa, puede hacer que la aplicación vulnerable se cuelgue en el código Java.

X-Force:

93408

OWASP:

M7

Arreglo:

Valide la entrada de usuario
Evite las exportaciones de componentes de la aplicación que no sean necesarias.

Carga útil

Paquete de Intent: formiik.com.mobiik.www

Clase de Intent:

activities.ActivityFormiikURL

Acción de Intent:

android.intent.action.VIEW

Datos de Intent:

http://G18B/

Firma de método:

```
android.app.ActivityThread$H.handleMessage
```

Parámetros de validación de problema:

Nombre	Valor
android.app.RemoteServiceException	<Param index=0 signature=java.lang.String instancetype=null name=java.lang.String value=Context.startForegroundService() did not then call Service.startForeground() filterResults=[]>

Pila de llamadas:

Función	Clase	Línea
<code>android.app.ActivityThread\$H.handleMessage</code>	<code>ActivityThread\$H</code>	<code>(ActivityThread.java:1778)</code>
<code>[Framework Code, 5 lines removed]</code>	<code>...</code>	<code>(...)</code>
<code>com.android.internal.os.ZygoteInit.main</code>	<code>ZygoteInit</code>	<code>(ZygoteInit.java:772)</code>

Recomendaciones de arreglo

H

Valide la entrada de usuario

[Ir a Tabla de contenido](#)

Tipos de problemas corregidos por esta tarea

- Cuelgue del código nativo
- Cuelgue del código Java

General

Valide siempre la entrada de usuario que puede controlar el adversario, por ejemplo, utilizando una aplicación maliciosa. Un modo recomendado para la validación de entradas de usuario es el uso de una lista blanca. En el ejemplo siguiente, la entrada de usuario se devuelve mediante la API `Intent.getDataString` y se valida para comprobar que se trata de un conjunto de cadenas (seguras).

```
final String[] SAFE_STRINGS = {"foo", "bar", "baz", "qux"};
Set<String> safeStrings = new HashSet<String>(Arrays.asList(SAFE_STRINGS));
String userInput = getIntent().getDataString();
if (!safeStrings.contains(userInput))
{
    FAIL...
}
sensitiveAPI(userInput);
```

H

No permita que la entrada de usuario se propague a las cadenas de formato.

[Ir a Tabla de contenido](#)

Tipos de problemas corregidos por esta tarea

- Cuelgue del código nativo

General

No permita que la entrada de usuario se propague a las API confidenciales, tales como el parámetro `format` de `printf`, (lo que también se conoce como ataque de cadenas de formatos)

H

Evite las exportaciones de componentes de la aplicación que no sean necesarias.

[Ir a Tabla de contenido](#)

Tipos de problemas corregidos por esta tarea

- Cuelgue del código nativo
- Cuelgue del código Java

General

Los componentes de aplicación, por ejemplo, las actividades, servicios y receptores, se pueden exportar accidentalmente al archivo Manifest de Android (AndroidManifest.xml). Dado que cuando se exporta un componente éste está accesible para una entrada maliciosa, no se debe llevar a cabo a menos que se requiera un acceso externo. Convertir un componente exportado en no exportado es una tarea fácil. Simplemente añada el atributo 'android:exported="false"' a la actividad vulnerable (según este informe) bajo el archivo Manifest de Android.

Por ejemplo:

```
<activity android:name="VulnerableActivityName" android:exported="false">
<intent-filter>
<action android:name="Foo" />
<category android:name="Bar" />
</intent-filter>
</activity>
```

Para obtener más información, consulte: <http://developer.android.com/guide/topics/manifest/activity-element.html#exported>.

H

Vuelva a configurar el servidor y la aplicación para evitar el uso de suites de cifrado débiles.

[Ir a Tabla de contenido](#)

Tipos de problemas corregidos por esta tarea

- Se admiten las suites de cifrado SSL débiles

General

Vuelva a configurar el servidor para evitar el uso de suites de cifrado débiles. Los cambios de configuración son específicos de servidor.

En el lado de la aplicación, no intente implementar suites de cifrado específicas que estén en la lista de suites de cifrado débiles.

Las suites de cifrado que dan soporte a sha1 también se consideran débiles.

Aquí encontrará la lista de suites de cifrado débiles:

M

No permita que se produzca un goteo de información confidencial.

[Ir a Tabla de contenido](#)

Tipos de problemas corregidos por esta tarea

- Filtración de información

General

Cuando está tratando con datos confidenciales, asegúrese de que la información se trata adecuadamente. No la almacene sin protección en el dispositivo ni la envíe a través de canales no seguros. No permita que datos procedentes del origen notificado alcancen el receptor notificado.

M

No utilice Generadores de números pseudoaleatorios (PRNG) débiles

[Ir a Tabla de contenido](#)

Tipos de problemas corregidos por esta tarea

- Generador de números aleatorios débiles

General

1. No confíe nunca en `Math.random` para que devuelva números aleatorios imprevisibles.
2. No confíe nunca en `java.util.Random` para que devuelva números aleatorios imprevisibles, protegiéndolos con semillas (esto es, invocando `java.util.Random.setSeed(long seed)` con una semilla que no sea previsible).
3. Utilice un Generador de números pseudoaleatorios (PRNG) que sea criptográficamente seguro, tal como `java.security.SecureRandom`, con su constructor predeterminado.
4. Evite aplicar semillas al PRNG por su cuenta. Confíe en `SecureRandom` para que genere las semillas. Por ejemplo, el código siguiente genera 1024 bytes pseudoaleatorios que son seguros criptográficamente:

```
SecureRandom random = new SecureRandom();  
byte[] data = new byte[1024];  
random.nextBytes(data);
```

M

Habilitar la fijación de certificados para esta conexión.

[Ir a Tabla de contenido](#)

Tipos de problemas corregidos por esta tarea

- Falta de fijación de certificados

General

La fijación de certificados se puede manejar mediante un `SSLSocketFactory`. Para obtener más información, consulte https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning#Android

L

Establezca el atributo 'android:allowBackup' en false.

Tipos de problemas corregidos por esta tarea

- Distintivo de copia de seguridad habilitado

General

Establezca en "false" el atributo 'android:allowBackup' de la etiqueta Application en el archivo Manifest de Android (AndroidManifest.xml).

Por ejemplo:

```
<application android:allowBackup="false">
```

```
...
```

```
</application>
```

Para obtener más información, consulte: <http://developer.android.com/guide/topics/manifest/application-element.html#allowbackup>.

Cobertura

Tipos de problemas: 71

[Ir a Tabla de contenido](#)

Tipos de problemas contra los que ASOC ha probado su aplicación.

Apropiación de actividad
Suplantación de carga de clases de Android
Inyección de fragmentos de Android
Inyección de argumentos
Ejecución del código de serialización no seguro de Adobe Creative (Aviary) SDK
Distintivo de copia de seguridad habilitado
Vulnerabilidad de agujero de gusano "wormhole" Baidu Moplus SDK
Robo de difusión
Desbordamiento de almacenamiento intermedio
BuildConfig.DEBUG es true en la versión de release
Serialización Jumio SDK no segura (CVE-2015-2000)
Serialización MetaIO SDK no segura (CVE-2015-2001)
Serialización Esri ArcGis SDK no segura (CVE-2015-2002)
Serialización no segura Pjsip Pjsua2 SDK (CVE-2015-2003)
Serialización GraceNote GNSDK no segura (CVE-2015-2004)
Serialización MyScript SDK no segura (CVE-2015-2020)
Okhttp Weak Certificate Chain Validation (CVE 2016-2402)
Apache Cordova Readable Logged Data in Previous Android Versions (CVE-2016-6799)
Inyección de mandatos
ConnectManipulation
IV constante
Contraseña constante
Prefijo constante de la contraseña (Salt)
Clave secreta constante
Semilla constante
Señal constante
Scripts entre aplicaciones Apache Cordova (CVE-2014-3500/1/2)
Explotación remota Apache Cordova de Variables de configuración secundarias (CVE-2015-1835)
Aleatorización débil Apache Cordova de BridgeSecret (CVE-2015-8320)
Credentials Leakage

El distintivo de depuración está habilitado en la versión del release

Se ha detectado una versión de depuración

Depuración habilitada de WebView en versión de release

Depuración habilitada de WebView en versión de depuración

Vulnerabilidad de Dropbox SDK para Android (CVE-2014-8889)

Cargar una biblioteca externa de forma no segura

Manipulación de archivos

Manipulación de cadena de formato en código Java

Filtración de información

Filtración de información

Función hash criptográfica quebrada

Criptografía quebrada

Permiso de archivos no seguro

Intent pendiente no seguro

La versión de SSL en desuso está soportada

Serialización no segura

Cuelgue del código Java

APK probablemente malicioso

MiTM (Man-in-the-Middle)

Falta de código de autenticación de mensajes criptográficos

Uso de MixedContentMode no seguro

Cuelgue del código nativo

Usurpación mediante MiTM (Man-in-the-Middle)

Proveedor de contenidos exportados

Vulnerabilidades críticas de Adobe Air (CVE-2016-[0986-1002])

Inyección de SQL del lado del cliente

Falta de fijación de certificados

Falta validación de nombre de dominio

Verificador de nombre de host TLS/SSL no seguro

Fábrica de sockets TLS/SSL no segura

Gestor de confianza TLS/SSL no seguro

Manejador de errores WebView TLS/SSL inseguro

Usurpación de servicio

Suplantación de interfaz de usuario

attUSILeakage

Reflejo no seguro

Existe un archivo binario con información de depuración en APK

Se admiten las suites de cifrado SSL débiles

Generador de números aleatorios débiles

Scripts entre aplicaciones

Scripts entre sitios (XSS) a través de MiTM (Man-in-the-Middle)

Actividades: 43

[Ir a Tabla de contenido](#)

Actividades que se han probado para encontrar vulnerabilidades de seguridad, definidas en el manifiesto de la aplicación.

com.hp.mss.hpprint.activity.PrintHelp
com.hp.mss.hpprint.activity.PrintPluginManagerActivity
com.hp.mss.hpprint.activity.PrintPreview
com.hp.mss.hpprint.activity.PrintServicePluginInformation
com.soundcloud.android.crop.CropImageActivity
formiik.com.mobiik.www.activities.ActivityAbout
formiik.com.mobiik.www.activities.ActivityConversation
formiik.com.mobiik.www.activities.ActivityConversations
formiik.com.mobiik.www.activities.ActivityExternalApsFiles
formiik.com.mobiik.www.activities.ActivityForgotPassword
formiik.com.mobiik.www.activities.ActivityFormiikGame
formiik.com.mobiik.www.activities.ActivityFormiikGameDay
formiik.com.mobiik.www.activities.ActivityFormiikGameMonthGraph
formiik.com.mobiik.www.activities.ActivityFormiikURL
formiik.com.mobiik.www.activities.ActivityHtmlViewer
formiik.com.mobiik.www.activities.ActivityImageEditor
formiik.com.mobiik.www.activities.ActivityImageViewer
formiik.com.mobiik.www.activities.ActivityImageViewerFullscreen
formiik.com.mobiik.www.activities.ActivityLogin
formiik.com.mobiik.www.activities.ActivityMap
formiik.com.mobiik.www.activities.ActivityOrder
formiik.com.mobiik.www.activities.ActivityOrderResults
formiik.com.mobiik.www.activities.ActivityOrderUpdatable
formiik.com.mobiik.www.activities.ActivityProfile
formiik.com.mobiik.www.activities.ActivityScanner
formiik.com.mobiik.www.activities.ActivitySettings
formiik.com.mobiik.www.activities.ActivitySplash
formiik.com.mobiik.www.activities.ActivitySupportNewTicket
formiik.com.mobiik.www.activities.ActivitySupportTicket
formiik.com.mobiik.www.activities.ActivitySupportTickets
formiik.com.mobiik.www.activities.ActivityTutorial
formiik.com.mobiik.www.activities.ActivityTutorials
formiik.com.mobiik.www.activities.ActivityVideoPlayer
formiik.com.mobiik.www.activities.ActivityVideos
formiik.com.mobiik.www.activities.ActivityVideosOrderHelp
formiik.com.mobiik.www.activities.ActivityWidgetFormEdit
formiik.com.mobiik.www.activities.ActivityWidgetImage
formiik.com.mobiik.www.activities.ActivityWidgetMeeting
formiik.com.mobiik.www.activities.ActivityWidgetSignature
formiik.com.mobiik.www.activities.delivery.ActivityOrders
formiik.com.mobiik.www.activities.financial.ActivityOrders
formiik.com.mobiik.www.activities.responseview.ActivityOrderResponseView
formiik.com.mobiik.www.activities.responseview.ActivityWidgetFormEditResponseView

Servicios: 6

[Ir a Tabla de contenido](#)

Servicios que se han probado para encontrar vulnerabilidades de seguridad, definidas en el manifiesto de la aplicación.

formiik.com.mobiik.www.background.ServiceDownloadBlobFiles
formiik.com.mobiik.www.background.ServiceFetchAddress
formiik.com.mobiik.www.background.ServiceGoogleCloudMessages\$GcmIntentService
formiik.com.mobiik.www.background.ServiceGooglePlayServicesLocation
formiik.com.mobiik.www.background.ServiceTaskExecutor
formiik.com.mobiik.www.background.ServiceUploadBlobFiles

Destinatarios: 7

[Ir a Tabla de contenido](#)

Destinatarios que se han probado para encontrar vulnerabilidades de seguridad, definidas en el manifiesto de la aplicación.

com.commonsware.cwac.wakeful.AlarmReceiver
formiik.com.mobiik.www.background.AlarmWorkTime
formiik.com.mobiik.www.background.BroadcastReceiverTaskWorker
formiik.com.mobiik.www.background.BroadcastReceiverTimeZone
formiik.com.mobiik.www.background.BroadcastReceiverTracking
formiik.com.mobiik.www.background.ServiceGoogleCloudMessages\$GcmBroadcastReceiver
formiik.com.mobiik.www.background.Synchronization\$SyncBroadcastReceiver