

**INFORME DE PRUEBAS  
DE ETHICAL HACKING PRODUCCIÓN  
YO CUIDO LO PÚBLICO MÓVIL**

---

**Dirección de Gobierno Digital**

Bogotá, D.C, octubre de 2017



**CONTROL DE CAMBIOS**

VERSIÓN	FECHA	No. SOLICITUD	RESPONSABLE	DESCRIPCIÓN
1.0	2017-10-09	No aplica	Servinformación	Creación del documento

**TABLA DE CONTENIDO**

1.	INTRODUCCIÓN .....	4
2.	INFORME DE PRUEBAS DE ETHICAL HACKING .....	5
2.1	ESCENARIO DE LAS PRUEBAS .....	5
2.2	PRUEBAS .....	5
2.2.1	EXPLOTACION DE VULNERABILIDADES .....	5
3.	CONCLUSIONES.....	8



## 1. INTRODUCCIÓN

**E**ste documento presenta el informe de los resultados obtenidos en la ejecución de las pruebas de Ethical Hacking realizadas a la aplicación solución móvil Yo Cuido Lo Público, en un ambiente de producción

Las pruebas de vulnerabilidad permitieron identificar las vulnerabilidades de la solución. Las pruebas de Ethical Hacking están enfocadas en explotar las vulnerabilidades encontradas, cabe destacar que todas las vulnerabilidades presentadas en el informe de pruebas de vulnerabilidad “GLFS2-SM4-INF-InformeDePruebasDeVulnerabilidadProduccion-EBM” el cual está localizado en la siguiente ruta: 03. Fase de Ejecución / 06. Producción / 01. Entrega/02. Pruebas de Seguridad, en las aplicaciones desarrolladas para Android son nivel de bajo y no generan un impacto alto a la confidencialidad, integridad o disponibilidad de la información manejada por la aplicación.

El desarrollo de las pruebas de Ethical Hacking, para este caso específico, necesita la generación de código que aproveche las vulnerabilidades específicas identificadas en el código dentro de las pruebas dinámicas y estáticas de la aplicación.

## **2. INFORME DE PRUEBAS DE ETHICAL HACKING**

---

**D**entro de las pruebas de Ethical Hacking se identifican las vulnerabilidades de la aplicación, se ejecutan pruebas de explotación de las vulnerabilidades identificadas, las cuales permiten medir la problemática real de seguridad a la que se encuentran enfrentados las aplicaciones y los servicios.

La ejecución de las pruebas de Ethical Hacking, permiten generar una medición real de las consecuencias de las vulnerabilidades identificadas en el paso anterior, estableciendo recomendaciones específicas para la generación de controles con el fin de mitigar los aspectos identificados en estas pruebas. Igualmente la explotación de vulnerabilidades, puede permitir la identificación de errores o problemas adicionales de seguridad en las aplicaciones y servicios analizados.

### **2.1 ESCENARIO DE LAS PRUEBAS**

La aplicación para Android fue montada en ambiente virtualizado, en una máquina virtual, con herramientas de medición las cuales se encuentran descritas en el informe GLFS2-SM4-INF-InformeDePruebasDeVulnerabilidadProduccion-EBM.docx localizado en la siguiente ruta: 03. Fase de Ejecución / 06. Producción / 01. Entrega/02. Pruebas de Seguridad, que permiten determinar el comportamiento de las mismas en ejecución, frente a las pruebas realizadas.

### **2.2 PRUEBAS**

#### **2.2.1 EXPLOTACION DE VULNERABILIDADES**

En esta parte específica, se explotan las vulnerabilidades identificadas en las pruebas de vulnerabilidad de la solución, para comprometer la solución a nivel de confidencialidad, integridad y/o disponibilidad.

La explotación se realiza basada en las vulnerabilidades detectadas en el análisis de código, algunas vulnerabilidades identificadas en el proyecto son de nivel bajo y para su prueba es necesario generar código específico para su explotación,



debido al nivel de las vulnerabilidades encontradas, no fue generado ningún tipo de código para explotar las mismas.

#### 2.2.1.1 INCORRECTA ESTRUCTURA DE METODOS

- **Descripción:** Fueron identificados métodos con una estructura que puede permitir la ejecución de código en los mismos. Vulnerabilidad referente a la aplicación en sistema Android.
- **Detalles:**
  - No se hace una correcta finalización de los métodos, lo que puede permitir que la manipulación de los mismos.
- **Impacto:** Medio
- **Tipo de vulnerabilidad:** Manipulación de Métodos
- **Recomendaciones:** Correcta definición de los métodos usados.

#### 2.2.1.2 USO DE ALCANCE EXPLICITO

- **Descripción:** Algún paquete o método tiene visibilidad de paquetes. Todos los paquetes que se encuentren dentro de las clases deben estar escondidos.
- **Detalles:** El alcance de los paquetes o métodos debe estar escondido dentro de las clases.
- **Impacto:** Bajo
- **Tipo de vulnerabilidad:** Alcance Explicito

#### 2.2.1.3 USO DE LITERALES EN CONDICIONALES

- **Descripción:** Fue posible identificar literales en algunos condicionales utilizados en algunos condicionales del código.
- **Detalles:** La definición de algunas declaraciones condicionales fueron realizadas con código estático incluido en la aplicación.
- **Impacto:** Bajo

- **Tipo de vulnerabilidad:** Manejo de Variables
- **Recomendaciones:** Hacer uso de variables dentro de las declaraciones generadas para condicionales.

### **2.2.1.4 CADA CLASE DEBE TENER COMO MINIMO UN CONSTRUCTOR**

- **Descripción:** Es necesario definir un constructor como mínimo para la clase. Es posible inicializar un objeto de manera arbitraria si el constructor no se encuentra inicializado en la clase.
- **Detalles:** Si un constructor no está descrito por clase es posible inicializar el objeto de manera arbitraria, dependiendo de las necesidades del atacante.
- **Impacto:** Bajo
- **Tipo de vulnerabilidad:** Definición de constructores



### 3. CONCLUSIONES

---

**A** continuación se presentan las conclusiones y recomendaciones.

- La explotación de las vulnerabilidades no generan un impacto alto para la aplicación por su misma naturaleza y clasificación de nivel bajo.
- Es necesario mitigar las vulnerabilidades encontradas, con el fin de no dejar posibles vectores de ataque en la plataforma que puedan ser aprovechados por usuarios mal intencionados.
- Es necesario revisar variables y constructores en las aplicaciones Android, con el fin de asegurar que no puedan ser manipuladas de manera arbitraria en la ejecución de la aplicación.
- Se recomienda mitigar la vulnerabilidad encontrada en la estructura de los métodos, teniendo en cuenta la necesidad de terminar los puntos de salida como última sentencia dentro del código.
- Se debe realizar la mitigación de las vulnerabilidades encontradas para Android.