MINTIC

servinformación
LOCALIZACIÓN INTELIGENTE

INFORME DE PRUEBAS DE ETHICAL HACKING PREPRODUCCIÓN 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	Actualización del documento

TABLA DE CONTENIDO

INTRODUCCIÓN

Este documento presenta el informe de los resultados obtenidos en la ejecución de las pruebas de Ethical Hacking realizadas a la aplicación Yo Cuido Lo Público, en un ambiente de preproducción.

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-InformeDePruebasDeVulnerabilidadPreproduccion-EBM.docx” 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.

1. INFORME DE PRUEBAS DE ETHICAL HACKING

Dentro de las pruebas de Ethical Hacking, adicional a la identificación de vulnerabilidades realizada en las pruebas de vulnerabilidad ya realizadas a la aplicación, se realizan pruebas de explotación de las vulnerabilidades identificadas, las cuales permiten medir, basado en el impacto generado, la problemática real de seguridad a la que se encuentran enfrentadas 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.

1.1 ESCENARIO DE LAS PRUEBAS

Las aplicaciones para Android fueron montadas en ambientes virtualizados, para Android, una máquina virtual con Android, con herramientas de medición que permiten determinar el comportamiento de las mismas en ejecución, frente a las pruebas realizadas.

1.2 PRUEBAS

1.2.1 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.

1.2.1.1 VARIABLE NO USADA

- **Descripción**: Existe una variable inicializada en el código que no es utilizada dentro del mismo.
- **Detalles**: Es posible manipular la variable dentro del código para colocar valores que puedan afectar la aplicación de manera negativa.
- **Impacto**: Bajo.
- **Tipo de vulnerabilidad**: Manipulación de Variables.
- **Recomendaciones**: Eliminar la variable si no es necesario en el programa.

1.2.1.2 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.

1.2.1.3 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 Explícito.

1.2.1.4 LLAMADOS EN 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.

2. CONCLUSIONES

A continuación se presentan las conclusiones y recomendaciones.

- La explotación de las vulnerabilidades no genera 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.