IPC SA de CV

Reporte de Seguridad

Sistema Nodejs MongoDb

Esta copia fue impresa en: 20 de Abril de 2020

Ultima modificación en: 20 de Abril de 2020

Version: 1.0

Status: <Final>

Dueño del documento: IPC SA de CV

Autores del documento: Joaquin Miranda Castro

<Efrén>

<Manuel>

Contenido

Lista de Tablas 3

1 Introducción 4

1.1 Contexto 4

1.2 Objetivos del Reporte 4

1.3 Referencias 4

1.4 Acrónimos y Definiciones 4

2 Definición de las Limitaciones 5

2.1 Alcance 5

2.2 Fuera de alcance 5

3 Detalle de los Riesgos 6

3.1 SQL Injection 6

3.2 Intrusión 6

3.3 Cross-Site Scripting 6

3.4 Canal de Comunicación Inseguro 6

4 Servidores sin Mejoras 7

5 Resultados sobre el Sitio sin Mejoras 8

5.1 Pruebas de SQL Injection 8

5.2 Pruebas de Intrusión 9

5.3 Pruebas de XSS 10

5.4 Pruebas de Conexión Segura 10

6 Servidores Mejorados 11

7 Resultados sobre el Sitio Mejorado 12

7.1 Pruebas de SQL Injection 12

7.2 Pruebas de Intrusión 12

7.3 Pruebas de XSS 12

7.4 Pruebas de conexión segura 12

Lista de Tablas

[Tabla 1: Referencias 4](#_Toc38269201)

[Tabla 2: Acrónimos y Definiciones 4](#_Toc38269202)

[**Tabla 3: Pruebas que se realizan** 5](#_Toc38269203)

[Tabla 4: Pruebas que no se realizan 5](#_Toc38269204)

Introducción

Contexto

Este Reporte de Seguridad para la empresa IPC SA de CV fue encargado al equipo de seguridad 04, que se encargan de mantener el servidor 72.55178.76

La meta es utilizar los conocimientos adquiridos para hacer que el servidor sea robusto ante una variedad de posibles ataques a los que se podría ser vulnerable. Esta seguridad sirve no sólo para este proyecto, sino también a futuros productos de la empresa.

Objetivos del Reporte

Los objetivos de este reporte son los siguientes:

• Garantizar la robustez del sistema de backend Nodejs + MongoDb

• Hacer que los servidores sean elegibles para el expo sobre artículos promocionales.

• Crear un estándar de seguridad que se llegue a utilizar en todos los servidores de la empresa IPC SA de CV.

Referencias

Este documento obtuvo sus fuentes o se refirió de las siguientes fuentes.

Tabla 1: Referencias

|  |  |
| --- | --- |
| **Título** | **Link** |
| Protegido: Proyecto Final | <https://enlacenet.info/ciberseguridad/index.php/proyecto-final/> |
| Protegido: SQL Injection | <https://enlacenet.info/ciberseguridad/index.php/sql-injection/> |
| Protegido: Cross-Site Scripting (XSS) | <https://enlacenet.info/ciberseguridad/index.php/cross-site-scripting-xss/> |
|  |  |

Acrónimos y Definiciones

Esta sección define la terminología y los acrónimos que serán utilizados específicamente en este documento.

Tabla 2: Acrónimos y Definiciones

| **Acrónimo** | **Definición** |
| --- | --- |
| SNM | Sistema Nodejs + MongoDb |
| XSS | Prueba de Cross-Site Scripting |
| HTTP | Hyper Text Transfer Protocol |
| HTTPS | Hyper Text Transfer Protocol Secure |
| SQL | Structured Query Language |
| NVT | Network Vulnerability Tests |
| DB | Base de Datos |

Definición de las Limitaciones

Alcance

Este reporte de encargará de realizar las siguientes pruebas:

**Tabla 3: Pruebas que se realizan**

|  |  |  |
| --- | --- | --- |
| **ID de Prueba** | **Nombre** | **Definición** |
| P001 | Prueba de SQL Injection | Pruebas sobre la seguridad de la base de datos. |
| P002 | Prueba de Intrusión | Pruebas de NVT buscando vulnerabilidades. |
| P003 | Prueba de XSS | Pruebas sobre la inyección de código malicioso. |
| P004 | Prueba de canal de comunicación | Certificación SSL del sitio seguro por parte de los navegadores. |

Fuera de alcance

Este reporte no se encarga de realizar ni certificar las siguientes:

Tabla 4: Pruebas que no se realizan

| **Item** | **Descripción** |
| --- | --- |
| 1 | Pruebas sobre la redundancia de los servidores. |
| 2 | Pruebas sobre la seguridad del sistema operativo. |
| 3 | Pruebas sobre la experiencia del usuario. |
|  |  |
|  |  |

Detalle de los Riesgos

Descripción de los diferentes ataques a los que el SNM podría llegar a ser vulnerable.

SQL Injection

Es una vulnerabilidad que permite al atacante enviar o “inyectar” instrucciones SQL de forma maliciosa y malintencionada dentro del código SQL programado para la manipulación de bases de datos, de esta forma todos los datos almacenados estarían en peligro.

Para el caso de MongoDB, ya no es como tal una SQL Injection, si no que la vulnerabilidad viene de una NoSQL Injection. Su objetivo es el mismo, solo cambian las estructuras de las querys.

Intrusión

Los riesgos de una intrusion son variados. Actualmente la lista actualizada de NVTs contiene más de 50,000 vunerabilidades, que van desde archivos de configuración del servidor hasta la detección de contraseñas fáciles de acceso al administrados.

Existen herramientas como OpenVAS, que son capaces de encontrar todas estas vulnerabilidades.

Cross-Site Scripting

XSS es un ataque de inyección de código malicioso para su posterior ejecución que puede realizarse a sitios web, aplicaciones locales e incluso al propio navegador.

Sucede cuando un usuario mal intencionado envía código malicioso a la aplicación web y se coloca en forma de un hipervínculo para conducir al usuario a otro sitio web, mensajería instantánea o un correo electrónico. Así mismo, puede provocar una negación de servicio.

Canal de Comunicación Inseguro

Un Certificado Digital consta de una pareja de claves criptográficas, una pública y una privada, creadas con un algoritmo matemático, de forma que aquello que se cifra con una de las claves sólo se puede descifrar con su clave pareja.

Si no se tiene este certificado, toda la información confidencial del sitio puede ser leída e interceptada por cualquiera que esté escuchando la comunicación entre el servidor y algún usuario.

Servidores sin Mejoras

<Efren, explicacion de la versión insegura del sitio, y como fue implementada>

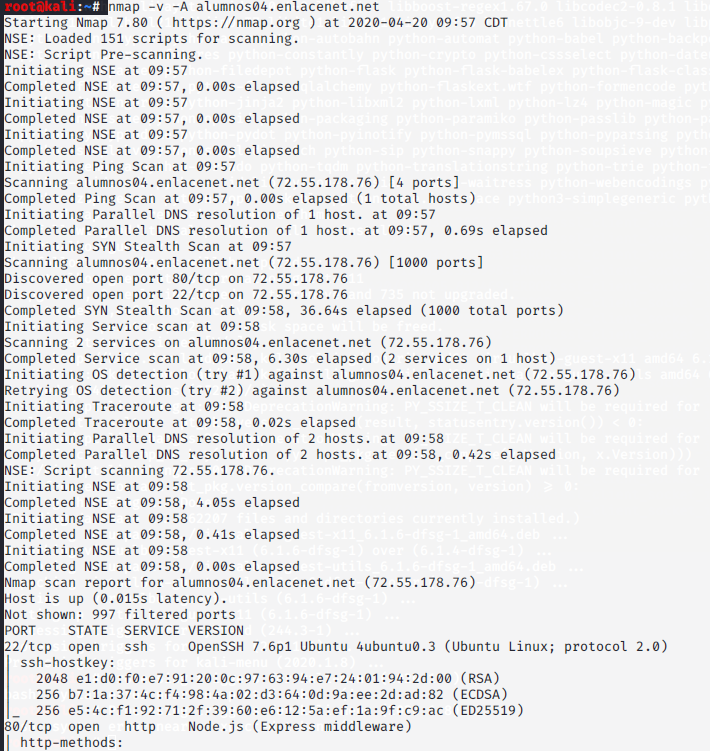
Resultados sobre el Sitio sin Mejoras

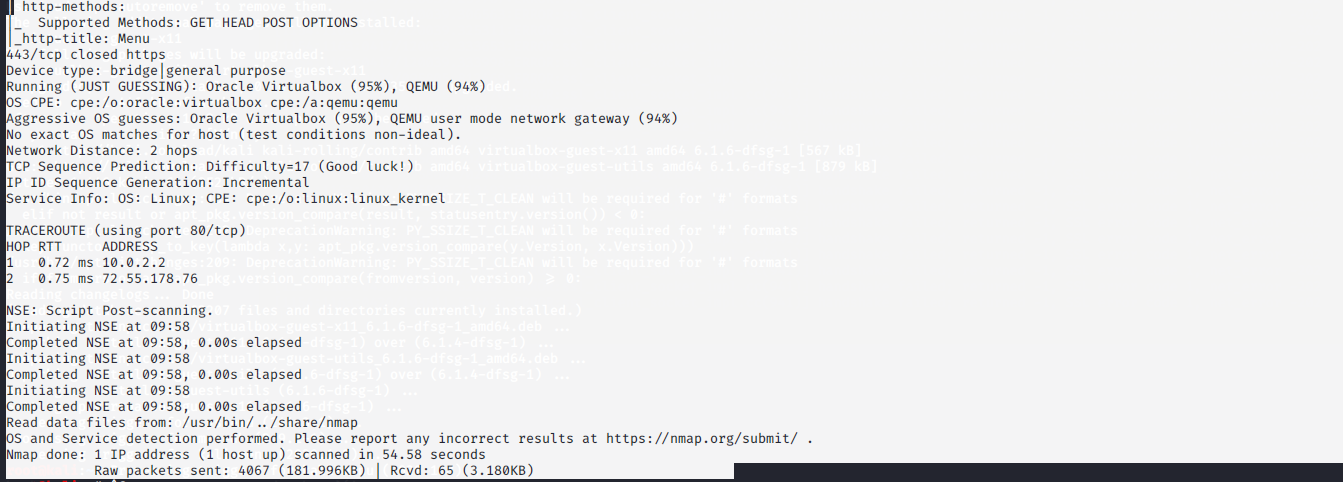
Estos son los resultados de las pruebas realizadas sobre el sitio sin seguridad.

Pruebas de SQL Injection

Se ejecutó la herramienta nmap, que de acuerto a su documentación (<https://nmap.org/nsedoc/scripts/mongodb-databases.html>) es capaz de realizar análisis sobre la vulnerabilidad de un servidor mongo.

Esto se ejecutó desde un equipo KaliLinux, y estos fueron los resultados:



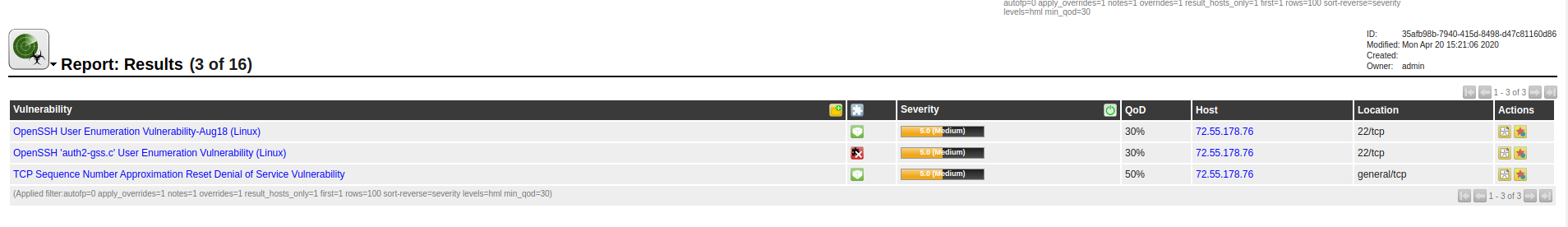


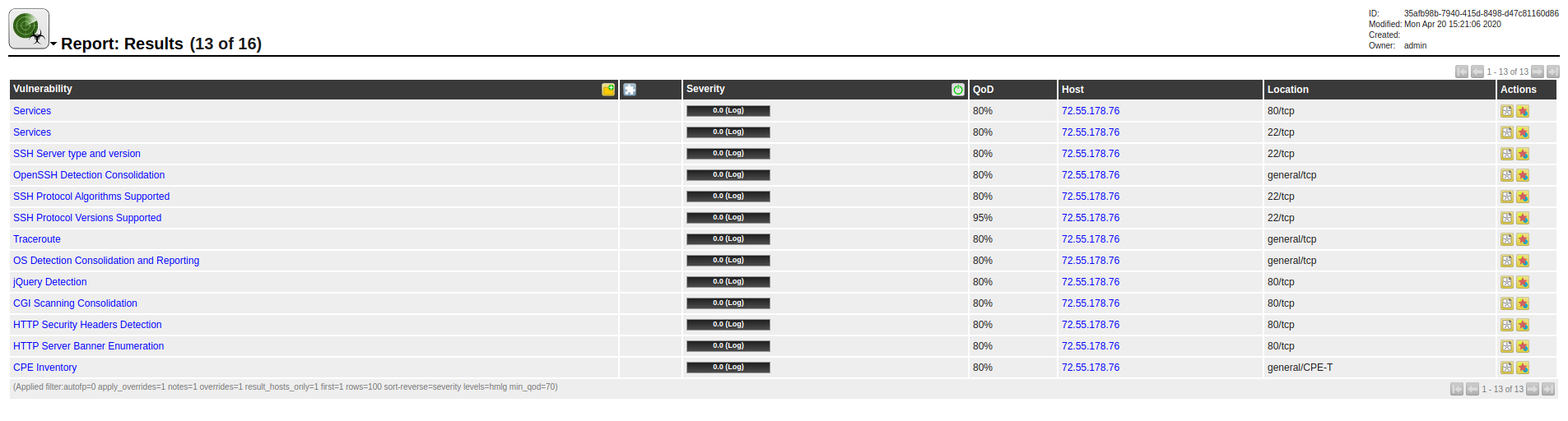
Esto indica que nmap no encontró este tipo de vulnerabilidad, incluso en este sitio inseguro.

¿Por qué? MongoDb utiliza por default BSON data, que es un tipo de JSON Binario, tanto en sus querys como en sus documentos. Esto le permite un cierto nivel de encriptación y protección que al parecer es suficiente.

Pruebas de Intrusión

Para esta prueba se utilizó la herramienta open source OpenVas, que analiza la IP proporcionada con la librería de NVTs actualizada.





Las pruebas no revelaron demasiados resultados, salvo la conexión insegura y algunas vulnerabilidades del servidor. NodeJs es de las comunidades más activas en desarrollos. Probablemente tiene muchas vulnerabilidades, pero son rápidamente reportadas y arregladas. (<https://groups.google.com/forum/#!forum/nodejs-sec>).

Esto, más la seguridad de MongoDB, hace que el servidor haya tenido pocos resultados desde OpenVas.

Pruebas de XSS

<Manuel>

Pruebas de Conexión Segura



Para esto solo fue necesario ingresar a la página desde el navegador. Automáticamente el navegador detecta que el sitio no es seguro, y te advierte de no ingresat ningún tipo de información sensitiva.

Servidores Mejorados

<Efren, explicacion de la versión segura del sitio, como fue implementada y las diferencias frente a la anterior versión¨.

Resultados sobre el Sitio Mejorado

Estos son los resultados de las pruebas realizadas sobre el sitio una vez que se le implementaron diferentes cambios para mejorar su seguridad:

Pruebas de SQL Injection

<Joaquin, cuando ya este arriba el sitio Seguro y con certificado>

Pruebas de Intrusión

<Joaquin, cuando ya este arriba el sitio Seguro y con certificado>

Pruebas de XSS

<Manuel>

Pruebas de conexión segura

<Manuel>