UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN, TELECOMUNICACIONES E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS

TEMA:

Análisis de Vulnerabilidades de Seguridad Informática, del Sistema de Gestión Médica SISMEDICALEC, de la empresa Incomsis.

Trabajo de Graduación. Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Ingeniero en Sistemas Computacionales e Informáticos

LÍNEA DE INVESTIGACIÓN: Sistemas Administradores de Recursos

AUTOR: Lisbeth Mariuxi Quirola Valarezo
TUTOR: Ing.David Omar Guevara Aulestia

Ambato - Ecuador
Agosto, 2019

CERTIFICACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema:

"Análisis de Vulnerabilidades de Seguridad Informática del Sistema de Gestión Médica SISMEDICALEC, de la empresa Incomsis", de la señorita, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería de Tecnologías de la Información, Telecomunicaciones e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para Obtener el Título Terminal de Tercer Nivel de la Universidad técnica de Ambato

Ambato, agosto del 2019

David Omar Guevara Aulestia

EL TUTOR

AUTORÍA

El presente trabajo de investigación titulado: "Análisis de Vulnerabilidades de Seguridad Informática del Sistema de Gestión Médica SISMEDICALEC, de la empresa Incomsis". Es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto del 2019

Lisbeth Mariuxi Quirola Valarezo

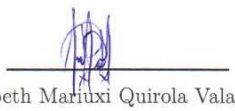
CC: 2101108427

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, agosto del 2019



Lisbeth Mariuxi Quirola Valarezo

CC: 2101108427

APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Dennis Chicaiza e Ing. Hernán Naranjo, revisó y aprobó el Informe Final del trabajo de graduación titulado "Análisis de Vulnerabilidades de Seguridad Informática del Sistema de Gestión Médica SISMEDICALEC, de la empresa Incomsis", presentado por la señorita Lisbeth Mariuxi Quirola Valarezo de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Mg. Elsa Pilar Urrutia Urrutia

PRESIDENTA DEL TRIBUNAL

Ing. Dennis Chicaiza
DOCENTE CALIFICADOR

Ing. Hernán Naranjo
DOCENTE CALIFICADOR

DEDICATORIA

El presente proyecto de Tesis está dedicado a Dios, que me ha permitido conseguir mi objetivo y concluir mi carrera estudiantil.

A mis padres que me han brindado su apoyo incondicionalmente y me han sabido guiar con amor y sabiduría en cada aspecto de mi vida.

Además a cada una de esas personas que me han brindado su atención, respeto y me han apoyado en este largo camino y a Manuel alguien muy importante en mi vida.

Lisbeth Mariuxi Quirola Valarezo

AGRADECIMIENTO

A Dios primeramente quien con su infinita misericordia y amor me ha brindado la fuerza y el valor para conseguir alcanzar mis metas.

A mi familia, por su amor su tiempo y palabras de aliento que supieron otorgarme para poder sobrellevar las adversidades que se me han presentado.

A mis maestros que, con su paciencia, experiencia, tiempo, dedicación y principalmente amor a su profesión me han enseñado los conocimientos necesarios para ser una buena profesional .

Y por último, a mis amigos que han sido como mi segunda familia, quienes han estado conmigo en los buenos y malos momentos y me han alentado a lo largo de toda mi formación humana, social y profesional.

Lisbeth Mariuxi Quirola Valarezo

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN EJECUTIVO

La finalidad del presente proyecto de investigación es consolidar cada conocimiento adquirido durante toda la trayectoria de mi carrera universitaria, y más en concreto, en el área de seguridad informática y administración de redes. Este documento que recoge todo el trabajo realizado durante el proyecto, en el que se puede reflejar tanto el estudio teórico del tema tratado, como la debida rea- lización práctica para el análisis de vulnerabilidades enfocado en el sistema web SISMEDICALEC perteneciente a la empresa Incomsis.

El presente proyecto, se plantea la problemática de una pequeña empresa que desea mejorar el desarrollo y seguridad de su software, para luego lograr diseñar páginas web más robustas y seguras para satisfacción de sus usuarios.

La inseguridad de las páginas web es un problema muy crítico en la actualidad. Es por ello por lo que las empresas que diseñan y desarrollan sistemas informáticos requieren una labor técnica que se condensa en un análisis de vulnerabilidades en la aplicación web. El proceso para llevar a cabo dicho análisis, está basado en la seguridad de aplicativos web, que se describen en este proyecto, que será útil para analizar detectar y otorgar un manual con las posibles soluciones para las vulnerabilidades web encontradas tanto en el código del software, sus configuraciones del sistema que aloja la aplicación web.

ABSTRACT

The purpose of this research project is to consolidate every knowledge acquired during the entire career of my university career, and more specifically, in the area of computer security and network administration. This document that includes all the work carried out during the project, which can reflect both the theoretical study of the subject, and the proper implementation for the vulnerability analysis focused on the web system SISMEDICALEC belonging to the company Incomsis. The present project, is the problem of a small company that wants to improve the development and security of its software, to then design more robust and secure web pages for the satisfaction of its users.

The insecurity of web pages is a very critical problem at present. It is for this reason that the companies that design and develop computer systems require a technical work that is condensed in a vulnerability analysis in the web application. The process to carry out this analysis is based on the security of web applications, which are described in this project, which will be useful to analyze detect and grant a manual with possible solutions for web vulnerabilities found in both the software code , your system configurations that host the web application.

INTRODUCCIÓN

El presente proyecto de investigación está enfocado en analizar las vulnerabilidades informáticas y elaborar un documento con los resultados obtenidos indicando las posibles soluciones en cada riesgo detectado.

CAPÍTULO I, "EL PROBLEMA", se describe el problema que estará sujeto a investigación, incluyendo también la justificación y el planteamiento de objetivos.

CAPÍTULO II, "MARCO TEÓRICO", contiene los antecedentes investigativos que fungirán como sustento al presente proyecto de investigación así también una propuesta de solución al problema planteados.

CAPÍTULO III, "METODOLOGÍA", se describe la forma en la que será llevado a cabo el presente proyecto justificando el uso de la metodología de investigación cualitativa y cómo será elaborado.

CAPÍTULO IV, "DESARROLLO DE LA PROPUESTA", se detalla cada uno de los pasos a seguir en el desarrollo de la investigación utilizando la metodología indicada en el capítulo anterior, además se elaborará un documento con los resultados obtenidos indicando las posibles soluciones en cada riesgo detectado.

CAPÍTULO V, "CONCLUSIONES Y RECOMENDACIONES", se presentan las conclusiones de los resultados obtenidos de la investigación y se detallan las recomendaciones a partir de un documento con los posibles correctivos de las vulnerabilidades detectadas, elaborado en el capítulo anterior.

CAPÍTULO 1

El problema

1.1. Tema de Investigación

Análisis de Vulnerabilidades de Seguridad Informática del Sistema de Gestión Médica EC, de la empresa Incomsis.

1.2. Planteamiento del problema

El avance tecnológico y el desarrollo de aplicaciones informáticas para soportar las necesidades del negocio de una organización hace imperioso cruzar fronteras, por ejemplo, acceder desde la Web hasta llegar a una base de datos que está gestionada por un software que corre sobre un equipo Mainframe. De este modo, la explotación de la aplicación se realiza atravesando diversas capas e integrando diferentes plataformas existentes en la organización. Dado que las capas tienen distintas naturalezas de seguridad, es necesario implementar un mecanismo eficiente que permita que las aplicaciones sean realmente seguras cumpliendo con los estándares respectivos y permaneciendo altamente alineadas con la tecnología La seguridad de las aplicaciones web al igual que con cualquier tecnología moderna, viene acompañada con una nueva variedad de vulnerabilidades de seguridad. El conjunto de los defectos que se detectan en los sitios web con mayor frecuencia ha evolucionado a lo largo del tiempo: los nuevos ataques están creados de forma tal que resulta un tanto complicado considerarlos, al momento del desarrollo de las aplicaciones y a su vez han surgido nuevas tecnologías que introdujeron nuevas posibilidades de explotación. Por otra parte, algunos problemas han perdido importancia en la medida que se ha incrementado la concientización respecto de ellos, y algunas categorías de defectos se han eliminado como consecuencia de los cambios introducidos en el software de los navegadores web. Es indiscutible que en la actualidad la seguridad en las aplicaciones web es el principal campo de batalla entre atacantes y aquellos que administran recursos y datos que se deben defender y es probable que siga así en el futuro mediato. Como parte de los mecanismos básicos de seguridad de las aplicaciones, los controles de acceso se encuentran construidos por encima de los mecanismos de autenticación

y de gestión de sesión; la principal razón por la que una aplicación necesita incluir estas funcionalidades, al menos en términos de la seguridad, es la necesidad de contar con algún mecanismo que le permita decidir si permite la ejecución de la acción indicada en una solicitud sobre los recursos indicados [1].

Muchas empresas actualmente no cuentan con planes de contingencia ante ataques de seguridad informática, y no es raro que aquellas dedicadas al desarrollo de software no vean la necesidad de implementar medidas para proteger la información, ya que se suele tener la creencia de que su tarea se limita a crear un producto que es ajeno a ataques de seguridad informática, pues tiende a verse como un asunto de la infraestructura y redes de telecomunicaciones, y que no concierne al mundo de la programación. En el entorno empresarial, a pesar de que la mayoría de las empresas no ven la importancia de implementar prácticas e incluir requerimientos de seguridad (ya sea porque los clientes no los solicitan o la propia empresa no los define), los riesgos informáticos están siempre presentes y pueden ocasionar grandes daños en las organizaciones. El problema está en que diversas empresas no tienen el conocimiento de cómo reflejar la baja calidad del software en los objetivos del negocio, y por tanto no se puede medir el impacto que estas situaciones tienen sobre los costos, la eficacia y la eficiencia de un proyecto [2].

El 24 de enero del 2015 Diario El Comercio en sus titulares dio a conocer que en Ecuador la mañana del 19 de enero del 2015, Cibermafias atacaron a 17 empresas en solo 5 días, donde encriptaron archivos valiosos: documentación levantada en Word, Excel y AutoCAD. Una de las empresas atacadas perdió carpetas donde almacenaba datos sensibles del departamento de contabilidad y reportes de la Fiscalía General del Estado del Ecuador misma que corroboran dicha afirmación. Según la publicación del Diario El Comercio del 28 de enero del 2015, el delito informático en el Ecuador evolucionó de la siguiente manera:

La **figura 1.1,** muestra la representación en Ecuador de los delitos informáticos, que indican que en el año 2008 hubo un porcentaje de solo el 20 %, mientras que en cinco años más creció al 46 %, lo que permite observar que los cibercrímenes cada año van en aumento.

Todas las empresas de servicios están ligadas a receptar y emitir información sensible, como por ejemplo bancos, aseguradoras, despachos jurídicos las mismas deben estar debidamente protegidas; en este proceso juega un papel importante el tipo de infraestructura/seguridad que utilicen y quien esté encargado de la misma; con el objeto de no sufrir ningún ataque o robo por personas no autorizadas y puedan contar con mayor confianza dentro del mercado al que se dedican [3].

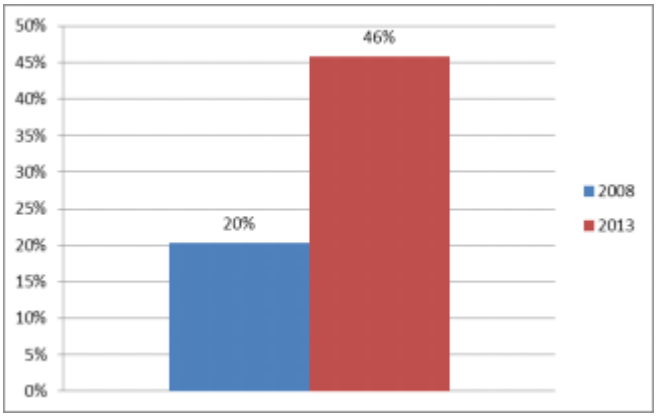


Figura 1.1: Representación de Evolución de delitos Informáticos en el Ecuador.

La empresa Incomsis, ha puesto a disposición de los profesionales de la salud, el Sistema de Gestión Médica, (SISMEDICALEC), el mismo que cumple con el objetivo de brindar una solución para consultorios médicos incluída el área de soporté técnico. El sistema actualmente no posee las políticas de seguridad de la información basada en las buenas prácticas existentes.

1.3. Delimitación

1.3.1. Delimitación de contenidos

El presente proyecto de investigación, su área de estudio está basada en Tecnología Informática.

Área académica: Hardware y Redes

Líneas de investigación: Sistemas administradores de recursos

Sublínea de investigación: Seguridad Informática.

1.3.2. Delimitación espacial

Este proyecto investigativo se llevará a cabo en la ciudad de Ambato, en las instalaciones de la empresa Incomsis.

1.3.3. Delimitación temporal

La investigación y desarrollo se efectuará durante el semestre septiembre-febrero 2019.

1.4. Justificación

Este proyecto de investigación es de gran interés para el personal de la empresa y los usuarios del sistema de gestión médica, pues almacena información de carácter confidencial y de importancia para las personas interesadas en el que consta datos

personales de cada usuario del sistema web.
En la actualidad se conoce que no se ha realizado un estudio minucioso en el área de seguridad informática, en las aplicaciones web de la institución, por lo que este proyecto investigativo representa un desarrollo tecnológico e innovador para los usuarios interesados.
Para realizar esta investigación se requiere de la asesoría del personal capacitado para el control y uso del sistema web SISMEDICALEC de la empresa Incomsis, en la cual se realizarán pruebas de seguridad informática, de los diferentes tipos de ataques que hoy en día son los de mayor vulnerabilidad para los sistemas web.
Los beneficiarios directos de este proyecto investigativo serán los usuario, propietarios y administradores del sistema de gestión médica EC de Incomsis, ofreciéndoles seguridad y confiabilidad de la información almacenada y distribuida en el mismo.

1.5. Objetivos

1.5.1. General

Analizar las vulnerabilidades informáticas para optimizar la seguridad de la información del sistema de gestión medica SISMEDICALEC,propiedad de la empresa Incomsis.

1.5.2. Específicos

- Realizar un estudio preliminar acerca de las tecnologías empleadas para la creación y utilización del sistema de gestión médica EC, de la empresa Incomsis.
- Identificar cuáles son las principales amenazas encontradas, al momento de realizar un análisis de las vulnerabilidades del sistema de gestión médica (SISMEDICALEC).
- Definir las tecnologías necesarias, para llevar a cabo los adecuados ataques informáticos para el sistema de gestión médica EC.
- Documentar los resultados obtenidos que indiquen las posibles soluciones a las vulnerabilidades encontradas en el sistema SISMEDICALEC.

CAPÍTULO 2

Marco Teórico

2.1. Antecedentes Investigativos

En la actualidad el mundo tecnológico avanza a pasos agigantados, por lo que es necesario que la información de carácter personal y empresarial sea almacenada en repositorios seguros, que brinden confiabilidad a los usuarios. Por lo cual el presente proyecto investigativo tendrá como objetivo principal detectar y analizar las diferentes vulnerabilidades informáticas encontradas dentro de una red que podría ser atacada, por ello se ha realizado una investigación de casos anteriores, acerca de debilidades en la web, incluido hacking ético.

Según la investigación realizada por los estudiantes de la Universidad Técnica de Babahoyo, Vega y Ramos en su artículo científico dan a conocer que no existe un software específico que gestione las actividades de seguridad informática y trate eficientemente las conexiones de usuarios de red y monitoree los accesos y usos de Internet.

Al tener un libre acceso a sitios no seguros dentro de la intranet y al no existir una adecuada administración en la infraestructura tecnológica, lo cual limita el trabajo al firewall de hardware de esta, lo que a su vez provoca una desconfianza total por las grandes vulnerabilidades internas de la Universidad [4].

En el proyecto de tesis de María Elena Hurtado Sandoval y Luis Alcides Mendaño Mendaño, dan a conocer que la información que fluye por las redes puede ser susceptible a diferentes tipos de ataques de esta manera, datos confidenciales de una organización en manos equivocadas podrían comprometer la integridad de la institución.

Y por ende con el pasar de los tiempos ha surgido la necesidad de implementar procesos de seguridad más robustos y con ello efectuar técnicas de intrusiones bajo un ambiente controlado, lo cual simule un ataque real. Esta simulación permite encontrar brechas en la seguridad, que los atacantes podrían aprovechar para infiltrarse en la red de una organización con fines maliciosos y de tal forma manipular la información, suplantar identidades, colapsar servicios, u otras actividades propias de un delincuente informático.

La función de un hacker ético es efectuar ataques controlados hacia una

infraestructura informática específica para detectar y explorar vulnerabilidades potenciales, pero sin poner en riesgo los sistemas y servicios auditados [5].

La especialista en Seguridad Informática de la Universidad Pontificia Bolivariana seccional Bucaramanga, Silvia Díaz, al finalizar con su investigación llegó a la conclusión que el proceso de aseguramiento de la calidad las empresas deben buscar definir los roles del equipo de calidad del software, las responsabilidades de los desarrolladores y las actividades respecto a los requisitos de seguridad de información, incorporando estos aspectos dentro de las pruebas de software, y no solamente cuando el cliente lo solicite.

Es fundamental que las empresas dedicadas al desarrollo de software establezcan sus políticas y procesos de aseguramiento de la calidad del software, definiendo los requerimientos de seguridad con base en un análisis de riesgos, teniendo en cuenta los tiempos, recursos (tecnológicos, humanos y económicos), limitaciones y ventajas competitivas del equipo (tanto de desarrollo como de calidad) invirtiendo a su vez en capacitación; con el fin de brindar un alto nivel de integración en las pruebas y por tanto de calidad de software, que se evidenciara en la satisfacción del cliente y de los usuarios finales, así como en la disminución de costos por trabajo extra o incumplimiento de tiempos [2]

2.2. Fundamentación Teórica

Auditoría de Seguridad de Sistemas de Información

Es el estudio que comprende el análisis y gestión de sistemas llevado a cabo por profesionales en tecnologías de la información (TI) con el objetivo de identificar, enumerar y describir las diversas vulnerabilidades que pudieran presentarse en una revisión exhaustiva de las estaciones de trabajo, redes de comunicaciones, servidores o aplicaciones [6].

Seguridad Informática

Es el conjunto de procedimientos, estrategias y herramientas que permitan garantizar la integridad, la disponibilidad y la confidencialidad de la información [7].

Un conjunto de métodos y herramientas destinadas a proteger la información y por ende los sistemas informáticos ante cualquier amenaza, un proceso en el cual participan además personas [8].

Análisis de Riesgos

Es un proceso que comprende la identificación de activos de información, sus

vulnerabilidades y las amenazas a los que se encuentran expuestos, así como la probabilidad de ocurrencia y el impacto de estas, a fin de determinar los controles adecuados para tratar el riesgo [6].

Amenazas

Cualquier circunstancia con el potencial suficiente para causar pérdida o daño al sistema. Ejemplos de amenazas son los ataques humanos, los desastres naturales, los errores humanos inadvertidos, fallas internas del hardware o el software.

Vulnerabilidades

Consistirá en cualquier debilidad que puede explotarse para causar pérdida o daño al sistema. De esta manera, en punto más débil de seguridad de un sistema consiste en el punto más débil de seguridad de un sistema, consiste en el punto de mayor vulnerabilidad de ese sistema. Ataque Es cualquier acción que explota una vulnerabilidad [9].

Hacking ético

Es un servicio de Auditoria de T.I, que ofrecen empresas especializadas, con el fin de evaluar la seguridad de un sistema informático de forma integral.

Hacker ético

Es un profesional que tiene las habilidades para evaluar la seguridad de un sistema informático de forma integral, llevando a la práctica una serie de pasos secuenciales y teniendo como un criterio trasversal una "Ética Profesional".

Cracker

Un cracker es alguien que viola la seguridad de un sistema informático de forma similar a como lo haría un hacker, sólo que, a diferencia de este último, el cracker realiza la intrusión con fines de beneficio personal o para hacer daño a su objetivo [10].

Pentest

Una prueba de penetración es un ataque a un sistema software o hardware con el objetivo de encontrar vulnerabilidades. El ataque implica un análisis activo de cualquier vulnerabilidad potencial, configuraciones deficientes o inadecuadas, tanto de hardware como de software, o deficiencias operativas en las medidas de seguridad. Este análisis se realiza desde la posición de un atacante potencial y

puede implicar la explotación activa de vulnerabilidades de seguridad. Tras la realización del ataque se presentará una evaluación de seguridad del sistema, indicando todos los problemas de seguridad detectados junto con una propuesta de mitigación o una solución técnica. La intención de una prueba de penetración es determinar la viabilidad de un ataque y el impacto en el negocio de un ataque exitoso [6].

Tipos de Prueba de Penetración

Caja Negra

Se proporciona poca o ninguna información sobre el objetivo de evaluación por adelantado. Es más frecuente en pruebas de penetración a redes.

Caja Cristal

Son realizadas generalmente por un equipo interno, pero ahora es más frecuen- te asignarlo a un equipo externo. El equipo de prueba usualmente es parte del equipo de QA, y como tal se convierte en una parte del ciclo de vida para el desarrollo del software. Se tiene acceso al código fuente para revisarlo y reportar las vulnerabilidades encontradas.

Caja Gris

Es el tipo de prueba más común. Requiere realizar más trabajo para obtener información necesaria. Es crítica la comunicación entre el equipo de pruebas y la empresa en evaluación [11].

Metodología OWASP Referencia de Escritorio en Seguridad de Aplicaciones de OWASP (OWASP Application Security Desk Reference).

La ASDR contiene las definiciones básicas y descripciones de todos los principios importantes de seguridad, agentes de amenaza, ataques, vulnerabilidades, contramedidas, impactos técnicos y de negocio en seguridad de aplicaciones. Esta es una referencia básica para todas las otras guías y es referenciada también por estos otros volúmenes [12].

Tipos de Ataques DDoS

Se entiende como denegación de servicio, en términos de seguridad informática, a un conjunto de técnicas que tienen por objetivo dejar un servidor inoperativo. Mediante este tipo de ataques se busca sobrecargar un servidor y de esta forma impedir que los usuarios legítimos puedan utilizar los servicios por prestados por

él. El ataque consiste en saturar con peticiones de servicio al servidor, hasta que éste no puede atenderlas, provocando su colapso. Un método más sofisticado es el ataque de Denegación de Servicio Distribuido (DDoS), mediante el cual las peticiones son enviadas, de forma coordinada entre varios equipos, que pueden estar siendo utilizados para este fin sin el conocimiento de sus legítimos dueños (por ejemplo, a través de una botnet).

Ataque de repetición

Es un tipo de ataque en el cual el atacante captura la información que viaja por la red, por ejemplo, un comando de autenticación que se envía a un sistema informático, para, posteriormente, enviarla de nuevo a su destinatario, sin que este note que ha sido capturada. Si el sistema informático o aplicación es vulnerable a este tipo de ataques, el sistema ejecutará el comando, como si fuera legítimo, enviando la respuesta al atacante que puede así obtener acceso al sistema. Para protegerse de este tipo de ataques el sistema informático puede tomar medidas como usar un control de identificación de comandos, de sellado de tiempos (timestamp), etc. junto con el cifrado y la firma de los comandos con el fin de evitar que sean reutilizados.

Spoofing

Es una técnica de suplantación de identidad en la Red, llevada a cabo por un ciberdelincuente generalmente gracias a un proceso de investigación o con el uso de malware. Los ataques de seguridad en las redes usando técnicas de spoofing ponen en riesgo la privacidad de los usuarios, así como la integridad de sus datos [6].

Virus

Es una secuencia de código que se inserta en un fichero ejecutable (denominado huésped), de forma que cuando el archivo se ejecuta, el virus también lo hace, insertándose a sí mismo en otros programas.

Gusanos

Un gusano es un programa capaz de ejecutarse y propagarse por sí mismo a través de redes, en ocasiones portando virus o aprovechando bugs de los sistemas a los que se conecta para dañarlos.

Caballos de Troya

Los troyanos o caballos de troya son instrucciones escondidas en un programa de

forma que este parezca realizar las tareas que un usuario espera de él, pero que realmente ejecute funciones ocultas sin el conocimiento del usuario [7].

Aplicación Web

Sitio web donde las entradas del usuario (navegación y entrada de datos) afectan al estado de negocio" Jim Conallen. Una aplicación web usa un sitio web como fachada de una aplicación tradicional [13].

CAPÍTULO 3

Metodología

3.1. Modalidad Básica de la investigación

Modalidad Bibliográfica

Este proyecto investigativo está basado en información encontrada en la Inter- net como artículos científicos, libros en pdf, enlaces bibliográficos también libros técnicos físicos, los cuales permiten obtener información clasificada para llevar a cabo la investigación.

Modalidad Aplicada

En esta modalidad se aplica los conocimientos adquiridos a lo largo del estudio académico basados en los módulos de Desarrollo de Software, Intranets-Extranets y Seguridad Informática.

Modalidad Campo

Para lograr conocer las actividades de esta empresa, en base a su entorno web y poder llevar a cabo el desarrollo del proyecto investigativo, ya sea para la detección de las vulnerabilidades informáticas, como también para diseñar las adecuadas correcciones, la investigación se desarrollará en las instalaciones de la empresa Incomsis.

3.2. Recolección de Información

Al realizar esta etapa del proyecto se necesitará información precisa y clara, pa- ra lograr su ejecución, se utilizará una herramienta de software libre orientada al hacking ético para el análisis de vulnerabilidades informáticas y al conocer cuál es su deficiencia se podrá diseñar los respectivos procesos de corrección aplicables. Se aplicará entrevistas a los directivos y personal de la institución para poder recopilar la mayor cantidad de información que será utilizada para la elaboración del proyecto investigativo.

Entrevista número uno

Nombre de los entrevistados: Diego Ocampo, Eduardo Masaquiza
Fecha: 25/10/2018
Cargo: programadores y analistas del sistema SISMEDICALEC
Lugar: Oficina para visitas en la empresa.

1. ¿Cuál es la principal funcionalidad del sistema?
Diego Ocampo, dice que: las funcionalidades del sistema son: - El agendamiento de citas medicas, Consulta externa, toma de signos vitales, CIE10 **Eduardo Masaquiza, respondió que:** las funcionalidades son: - Agendar citas y Control de citas médicas.

2. ¿El sistema se encuentra dividido en módulos, si es afirmativo cuántos y cuáles son?
Diego Ocampo, responde: - El sistema se compone de un solo modulo, es el de Agendamiento de Citas y la Consulta
Eduardo Masaquiza, dice: Si, está dividido módulos y son los siguientes: - Agendar citas, Consultas externas , Controlar signos vitales, CIE10 , Usuarios y roles, Pacientes, Doctores.

3. ¿Qué cantidad de usuarios se conectan en el sistema y cuantos son simultáneos?
Diego Ocampo, responde: - Está dividido en roles en administrador y enfermeros aproximadamente 5 personas.
Eduardo Masaquiza, dice:- Los que se conectan actualmente son tres personas .

4. ¿Existe un log que almacene los errores del sistema y en qué consiste el mismo?
Diego Ocampo, responde: - Si tiene un log que almacena todas las actividades, es un log completo, como ingresos y el total funcionamiento del sistema.
Eduardo Masaquiza, dice:- El Log lo genera el proveedor del hosting cuando hay errores en el sistema .

5. ¿El sistema ha sido desarrollado en capas, puede detallar cada una de ellas?

Diego Ocampo, responde: - Es un sistema solo está estructurado, no está en capas.
Eduardo Masaquiza dice:- El sistema web no está creado el sistema en capas.

6. ¿El sistema SISMEDICALEC, contiene certificados de seguridad?
Diego Ocampo, responde: - Si posee, el certificado AES256, emitido por COMODO.
Eduardo Masaquiza, dice:- El sistema web, posee el certificado AES256, emitido por COMODO.

7. ¿Qué tipo de cifrado utiliza el sistema para la seguridad de información vulnerable como las contraseñas?
Diego Ocampo, responde: - El cifrado que usa es ShA1 256 y md5.
Eduardo Masaquiza, responde:- El cifrado Sha1 256 y md5.

8. ¿El sistema se encuentra alojado en un housing con servidores propios o un hosting de pago en la nube?
Diego Ocampo, responde: - El sistema está almacenado en un hosting de pago y está en Argentina.
Eduardo Masaquiza, dice:- Está alojado en un hosting de pago en la nube.

9. ¿Cada qué tiempo se realiza respaldos de la información del sistema que tipos y en donde es almacenado?
Diego Ocampo, responde: - El tiempo para la realización de respaldos de información es variante dependiendo de factores relacionados al sistema web. Se emite un certificado en Google Chrome y luego de la correcta configuración del certificado, al navegar en la carpeta Home ahí está la almacenada la public_html, la que contiene los datos de la web.
Eduardo Masaquiza, dice:- El tiempo para la realización de respaldos de información es variante dependiendo de factores relacionados al sistema web. Se emite un certificado en Google Chrome y luego de la correcta configuración del certificado, al navegar en la carpeta Home donde está la almacenada public_html, la que contiene los datos de la web. Es almacenada la información en un hosting externo.

10. ¿El sistema web cuenta con un log de auditoria?
Diego Ocampo, responde: - No posee actualmente.
Eduardo Masaquiza, dice:- No posee actualmente.

11. Describa cuales son los roles de los usuarios del sistema y detalle cada uno de ellos.

Diego Ocampo,manifiesta que:- Rol administrador: acceso total , Rol usuario médico: que puede hacer todo, Rol de usuario del enfermero: que realiza la consulta de los signos vitales del paciente, Rol de usuario invitado: cliente: que hace el agendamiento de la cita.

Eduardo Masaquiza, respondió que:- Rol de administrador: administración para añadir, actualizar, eliminar y seleccionar los usuarios del sistema. Medico: realiza consultas y signos vitales, confirma la cita del paciente - Usuario: agendar citas , Enfermero: controla los signos vitales del paciente.

Entrevista número dos

Nombre del entrevistado: Ing. Adrián Figueroa
Fecha: 03/11/2018
Cargo: Gerente propietario de la empresa Incomsis.
Lugar: Oficina de gerencia en la empresa

1. ¿Se ha elaborado el debido proceso para el control de calidad del sistema web?

Responde:- El Control de la calidad del software se ha determinado de acuerdo con el factor de seguridad, es decir, si la aplicación es de fácil intercepción y vulnerable no podría salir a producción. Un tema bastante importante es resolver todos los requerimientos que tiene el cliente para así ofrecer un producto adapta- do al usuario final.

2. Ha tenido dificultades para ingresar al sistema SISMEDICALEC?

Responde:- Al principio no se ha tenido problema, ya que obedece a un flujo de trabajo utilizando la experiencia que tienen los médicos con sus pacientes. Los principales problemas se han dado debido a las pésimas conexiones de internet que tienen los clientes, los cuales hemos sugerido migrar a otro proveedor de internet con mejor estabilidad.

3. ¿Cómo maneja el sistema SISMEDICALEC, las políticas de contraseñas?

Responde:- Como política de contraseñas, se exige un mínimo de 8 caracteres, combinados entre letras mayúsculas, minúsculas, caracteres especiales y números.

4. ¿Piensa que es necesario simular un test de penetración para el sistema SISMEDICALEC, para poder determinar la seguridad de este?

Responde:- Claro que sí, es importante no solo ahora, sino continuamente, así vamos descubriendo vulnerabilidades a corregir en el sistema .

5. ¿Está de acuerdo en implementar un conjunto de buenas prácticas de seguridad para la información, de las funciones del sistema web, elaboradas a partir de recomendaciones, normativas y estándares?

Responde:- Muy importante, y totalmente de acuerdo, las buenas prácticas hacen que el producto se mantenga en una excelente calidad.

6. ¿Ha conocido casos de violación de información en el sistema web?
Responde:- En la actualidad nuestro sistema está en una fase de pruebas por lo cual aún no hemos detectado casos de violación de nuestra información.

7. ¿La empresa Incomsis ha desarrollado algún proceso de soporte y mantenimiento para el sistema SISMEDICALEC?
Responde:- Al momento el sistema se encuentra en evaluación y corrección de errores generales, para poder ponerlo en producción.

8. ¿La empresa Incomsis posee la adecuada documentación de creación y puesta en marcha del sistema SISMEDICALEC?
Responde:- Al momento se está generando toda la documentación necesaria para el sistema, la misma que no se encuentra disponible por el momento.

3.3. Procesamiento y análisis de datos

Recopilada la información se organiza y se analiza los procesos tomando en cuenta en primer lugar aquellos que representen un mayor riesgo dentro de la empresa. Aplicando procedimientos de estudio en las entrevistas los cuales serán expuestos posteriormente para facilitar su respectivo análisis e interpretación de los resultados obtenidos a través de ellas.

3.4. Analisis e interpretación de la información obtenida en base a las entrevistas

Interpretación y conclusiones de la entrevista realizada a los programadores del sistema web, "sismedicalec"

Pregunta número 1:
Interpretación de respuestas: Los dos programadores concuerdan que las funciones principales del sistema Sismedicalec, es el Agendamiento de Citas médicas y el Control de las citas médicas en base a una consulta externa, lo que manifiesta cual es el objetivo principal del sistema web para uso de los profesionales de la salud.
Conclusiones: - El sistema SISMEDICALEC, cuente con dos módulos que son, Agendamiento de Citas y Control de citas médicas, para beneficio de los profesionales de la salud.

Pregunta número 2:

Interpretación de respuestas: Los programadores entrevistados manifiestan respuestas diferentes Diego Ocampo responde que el sistema web solo está compuesto por un solo modulo, mientras Eduardo Masaquiza expresa que el sistema posee varios módulos y los nombra a cada uno de ellos, lo que da a entender que los programadores tienen una opinión totalmente diferente respecto a la cantidad de módulos que posee el sistema Sismedicalec.
Conclusiones: No existe una respuesta concordante entre los programadores entrevistados respecto al número de módulos que posee el sistema web.

Pregunta número 3:
Interpretación de respuestas: En respuesta a la cantidad de usuarios que se conectan simultáneamente en el sistema también expresan respuestas diferentes debido a que Diego Ocampo dice que se conectan 5 personas mientras que Eduardo Masaquiza dice que solo son 3 usuarios que se conectan simultáneamente, lo que genera una duda pues no está claro el número total de usuarios conectados.
Conclusiones: En este ítem no coinciden las respuestas de los programadores pues cada uno expresa un número diferente de los usuarios que se conectan simultáneamente al sistema informático.

Pregunta número 4:
Interpretación de respuestas: Los dos programadores concuerdan en decir que el sistema web si posee un log de errores que es generado y almacenado en el hosting contratado para el almacenamiento de Sismedicalec.
Conclusiones: - El sistema web posee un log de errores con su correcto almacenamiento en un hosting específico de uso de la empresa.

Pregunta número 5:
Interpretación de respuestas: La respuesta expresada por los dos programadores coincide manifestando que el sistema web no ha sido desarrollado en capas lo cual sería una ventaja para la empresa si lo hubieran hecho ya que podrían hacer reutilización del código programado y seria de mejor y rápido entendimiento para otros programadores de la empresa.
Conclusiones: El sistema Sismedicalec no ha sido programado en capas, lo que permitiría la separación en partes de los componentes del sistema siendo más sencillo para mantención y creación de las interfaces.

Pregunta número 6:

Interpretación de respuestas: Los dos programadores entrevistados coinciden en que el sistema web posee un certificado de seguridad que es un algoritmo de cifrado simétrico denominado AES 256, emitido por COMODO, el mismo que fue diseñado para ser eficiente tanto en hardware como en software para asegurar información sensible, y admite una longitud de bloque de 128 bits y una longitud de clave de 128, 192 y 256 bits.
Conclusiones: - El sistema Sismedicalec, posee un certificado de seguridad (SSL) denominado AES 256 emitido por COMODO .

Pregunta número 7:
Interpretación de respuestas: Los programadores entrevistados concuerdan en su respuesta diciendo que el sistema Sismedicalec, posee un cifrado de seguridad para las contraseñas de usuarios el cual es el cifrado sha1 256 que es un algoritmo criptográfico y por medio de la encriptación, la información privada puede ser enviada públicamente por internet sin mayor riesgo de que otros puedan tener acceso a ella, ya que solo los que poseen la clave de encriptación podrían tener el acceso permitido.
Conclusiones: El sistema web contiene un cifrado de seguridad para la gestión de contraseñas de usuarios denominado sha1 256, algoritmo de criptografía.

Pregunta número 8:
Interpretación de respuestas: La respuesta expresada por los dos programadores coincide al decir que el sistema Sismedicalec, está alojado en un hosting de pago en Argentina contratado por la empresa Incomsis, para el correcto almacenamiento de los sistemas web, y así se cumple con un nivel de seguridad que expone que los respaldos de información deben estar alojados fuera de la infraestructura de la empresa..
Conclusiones: El sistema Sismedicalec, para su correcta administración ha sido alojado en un hosting de pago en el extranjero.

Pregunta número 9:
Interpretación de respuestas: Los dos programadores coinciden por completo en su respuesta manifestando que Google Chrome emite el certificado, y que se debe realizar la adecuada configuración del certificado según los requerimientos de la empresa, también mencionan que los contenidos de los datos web están alojados en la carpeta public_html, la cual contiene toda la información respectiva del sistema web Sismedicalec.

Conclusiones: El contenido del sistema web está alojado en la carpeta public_html, y que poseen un certificado emitido por Google Chrome y el tiempo de respaldos de información se realiza mediante una planificación dependiendo de diversos factores .

Pregunta número 10:

Interpretación de respuestas: La respuesta brindada por los dos programa- dores entrevistados expone que el sistema web en la actualidad aún no posee un log de auditoria, lo que es una desventaja para la misma pues el log almacena registros de auditoria de manera remota para proteger los datos del registro y evitar que sean modificados o suprimidos por un atacante informático.

Conclusiones: Actualmente el sistema Sismedicalec, no posee un log de auditoría.

Pregunta número 11:

Interpretación de respuestas: Los programadores entrevistados concuerdan en su respuesta manifestando que el sistema posee varios roles de usuario desde el rol de administrador hasta el rol de usuario invitado, lo cual es bueno para la empresa pues cada persona según su rol puede hacer uso de la información con las normas de seguridad adecuadas que son asignadas según el usuario que acceda.

Conclusiones: El sistema informático, posee diversos roles de usuario para su correcto gestionamiento y administración de la información.

Interpretación y conclusiones de la entrevista número dos realizada a el gerente propietario de la empresa "Incomsis"

Pregunta número 1:

Interpretación de respuestas: El Ing. Adrián Figueroa, manifiesta que efectivamente si se ha elaborado un control de calidad del software tomando en cuenta los requerimientos del cliente que es un factor sumamente importante para brindar un producto de calidad y eficiencia para lograr así satisfacer sus más altas exigencias.

Conclusiones: Se ha elaborado un control de calidad de software para el Sistema de Gestión Médica (Sismedicalec).

Pregunta número 2:

Interpretación de respuestas: El entrevistado, expresa que en cuanto a la funcionalidad del sistema no ha existido problemas al ingresar, ya que ellos se han presentado por el mal servicio de conectividad que los proveedores de internetles han brindado a sus clientes, por lo que se ha recomendado a los mismos migrar a otros proveedores que les ofrezcan más beneficios como una mejor conectividad y rapidez en la navegación web, y así mejorar la experiencia de los usuarios.

Conclusiones: Actualmente no se conoce que los usuarios hayan tenido algún problema al intentar ingresar al sistema web.

Pregunta número 3:

Interpretación de respuestas: El Ing. Figueroa, expresa en su contestación que evidentemente el sistema Sismedicalec, tiene políticas de seguridad para las contraseñas de los usuarios el mismo que consiste en crear una clase con un mínimo de caracteres establecido y que tenga una combinación de ellos como letras, números y caracteres especiales, lo cual le brinda al usuario un cierto grado de seguridad, pues de esa forma le sería al atacante mucho más complejo poder robar las contraseñas.

Conclusiones: El software web posee políticas de seguridad para la gestión de contraseñas de usuarios, con ciertas especificaciones en los caracteres para establecerla.

Pregunta número 4:

Interpretación de respuestas: El Ing. Figueroa, manifiesta que está totalmente de acuerdo que en su empresa se lleve a cabo pruebas de penetración al sistema Sismedicalec, lo cual ofrecería diversas ventajas y una de las más importantes es

mantener de una forma segura y confiable el almacenamiento de información, por medio del hallazgo y corrección de vulnerabilidades web que pueden o podrían existir.

Conclusiones: El gerente de la empresa Incomsis, está de acuerdo en que se elabore un análisis de vulnerabilidades en el sistema web.

Pregunta número 5:

Interpretación de respuestas: El Ing. entrevistado, expresa que efectivamente está de acuerdo en que se deben aplicar un conjunto de buenas prácticas para la seguridad de la información en base a normas y estándares de calidad del producto, lo que a su vez les brinde seguridad, confianza y satisfacción a los clientes.

Conclusiones: Ing. Figueroa gerente, está de acuerdo en la aplicación de un conjunto de buenas prácticas para la confianza y satisfacción de los clientes.

Pregunta número 6:

Interpretación de respuestas: El Ing. Figueroa, expresa que el sistema Sismedicalec, es un software que, por su corto tiempo de elaboración, aún no se han detectado casos de violación de información que pongan en riesgo la seguridad del usuario.

Conclusiones: Por el corto tiempo de creación del sistema web en la actualidad aún no se conocen de casos de manipulación incorrecta de la información.

Pregunta número 7:

Interpretación de respuestas: El sistema Sismedicalec, se encuentra en una fase de evaluación y corrección de errores generales para su posterior puesta en marcha, por lo cual momentáneamente no se ha elaborado un proceso de soporte y mantenimiento del sitio web.

Conclusiones: El sistema informático, por el momento se encuentra en una etapa de evaluación y corrección de errores generales.

Pregunta número 8:

Interpretación de respuestas: El Ing. Figueroa, expone que la empresa INCOMIS, que todavía están en proceso de generación de la documentación correspondiente al sistema web, pero que aún no se encuentra disponible para el uso e instrucción de sus clientes.

Conclusiones: El sistema Sismedicalec, temporalmente no tiene disponible su

documentación, pero que la empresa se encuentra en proceso de desarrollo.

3.5. Desarrollo del Proyecto

Se presentará, a continuación, una lista de actividades a llevar a cabo, en referencia a los objetivos específicos, que permitirá cumplir el objetivo general.

Realizar un estudio preliminar acerca de las tecnologías empleadas para la creación y utilización del sistema de gestión médica de la empresa Incomsis.

- Analizar el medio en donde se han creado el sistema web.
- Comprobar los diferentes servidores en donde ha sido almacenada la información de las aplicaciones web.
- Verificar los medios físicos que están disponibles para la seguridad del equipo de cómputo y servidores, en el caso de que cuenten con ellos.

Identificar cuáles son las principales amenazas encontradas, al momen- to de realizar un análisis de las vulnerabilidades de (Sismedicalec) sis- tema de gestión médica.

- Establecer los diferentes tipos de ataques informáticos hacia un entorno web, enfocado tanto en el cliente y el servidor.
- Analizar información archivada de los ataques más habituales ejecutados en las aplicaciones web en nuestro país.
- Recopilar datos de los daños más comunes realizados en el Sistema de Gestión Médica EC, de Incomsis.
- Precisar las tecnologías adecuadas y necesarias para realizar los ataques cibernéticos necesarios para las aplicaciones web.

Definir las tecnologías útiles, para llevar a cabo los adecuados ataques informáticos necesarios para el sistema de gestión médica EC.

- Realizar reuniones con el personal de interés para obtener previa autori- zación y poder ejecutar las pruebas en base a hacking ético al sistema de gestión médica de Incomsis.
- Investigar las diversas herramientas que faciliten el llevar a cabo las pruebas de penetración (PenTesting).

- Empezar con ataques informáticos más habituales, previamente analizados e ir avanzando hasta la realización de ataques con mayor riesgo y complejidad.
- Documentar los resultados de la ejecución de las pruebas de penetración.

Documentar los resultados obtenidos que indiquen las posibles soluciones a las vulnerabilidades encontradas en el sistema Sismedicalec.

- Estudiar los datos obtenidos de la realización de las correspondientes pruebas de penetración en el sistema web.
- Averiguar cuáles son las herramientas adecuadas de seguridad informática para reducir riesgos en el sistema web.
- Crear documentación necesaria sobre los mejores procesos a seguir para garantizar la seguridad informática, indicando las posibles soluciones en cada vulnerabilidad detectada.

CAPÍTULO 4

Desarrollo de la propuesta

4.1. Antecedentes de la propuesta

Establecer seguridad de la información no es solo cuestión técnica, se debe considerar a las personas, los procesos y funcionalidad de la naturaleza de los datos, así también como la protección de todos los recursos tanto lógicos como físicos de una institución.

Las aplicaciones web, deben cumplir con ciertos estándares de seguridad que permitan garantizar su adecuado funcionamiento, de manera que este disponible cuando sea necesario, que existan garantías de que los datos de caracter sensible sean procesados correctamente y que solo puedan acceder a ellos las personas autorizadas.

Los usuarios de la expuesta aplicación web Sistema de Gestión Médica Sismedicalec, quienes son las principales personas de interés. Además de ellos, los desarrolladores, como el personal que ofrece soporte al sistema web, son los mas preocupados en que el aplicativo web posea un adecuado nivel óptimo de seguridad.

En base a lo expuesto, se propone elaborar un informe donde se documente los resultados previamente obtenidos, indicando las posibles soluciones adecuadas para cada vulnerabilidad informática detectada en la aplicación Sismedicalec, de la empresa Incomsis, para lo cual se identificó las funcionalidades del aplicativo web, también se analizó la seguridad de la información del mismo, mediante ataques informáticos. Por último se presentará un documento final que exponga las posibles soluciones y recomendaciones para mejorar la seguridad de la información del sistema web.

4.2. Descripción del Sistema Operativo y herramientas que usa el servidor.

Sistema Operativo	**Versión empresa-rial**	**Versión ambiente de prueba**
Centos:distribución mantenida por su comunidad de usuarios que utiliza los ficheros fuente liberados por Red Hat para su distribución empresarial conocida como Red Hat Enterprise Linux (RHEL), lo que la hace ideal para servidores por ser bastante estables [14].	7	7

Tabla 4.1: Tabla de descripción del sistema operativo CentOs 7

Herramientas	Versión Empresa-rial	Versión ambiente de prueba
PHP: El lenguaje PHP (cuyo nombre es acrónimo de PHP: Hipertext Preprocessor) es un lenguaje interpretado con una sintaxis similar a la de C++ o JAVA. Aunque el lenguaje se puede usar para realizar cualquier tipo de programa, es en la generación dinámica de páginas web donde ha alcanzado su máxima popularidad. En concreto, suele incluirse incrustado en páginas HTML (o XHTML), siendo el servidor web el encargado de ejecutarlo [15].	5.6	5.6
Apache: El servidor HTTP Apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/4.3 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 4.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, a patchy server (un servidor "parcheado") [16].	2.4	2.4
MySQL: Es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha [17].	5.6	5.6

Tabla 4.2: Cuadro comparativo del uso de herramientas para el almacenamiento y funcionamiento del aplicativo web.

4.3. Documentación de los resultados obtenidos y las posibles soluciones a las vulnerabilidades encontradas en la aplicación Sismedicalec.

4.3.1. Herramienta Vega Scanner

Es un escáner de código abierto y libre y una plataforma de prueba para probar la seguridad de las aplicaciones web. Esta herramienta ayudar a encontrar y validar SQL Injection, Cross-Site Scripting (XSS), revelo inadvertidamente información confidencial y otras vulnerabilidades. Está escrito en Java, basado en GUI, y se ejecuta en Linux, OS X y Windows [18].

VEGA incluye un escáner automatizado para pruebas rápidas y un proxy de interceptación para la inspección táctica. El escáner de VEGA encuentra XSS (cross-site scripting), inyección de SQL y otras vulnerabilidades [19].

VEGA se puede ampliar usando una poderosa API en el lenguaje de la web: escáner automatizado de rastreadores y vulnerabilidades, interfaz de usuario coherente, crawler del sitio web, proxy de interceptación, SSL MITM, análisis del contenido, extensibilidad a través de un potente API de módulo Javascript, alertas personalizables, base de datos y modelo de datos compartidos. Para la identificación de las vulnerabilidades se utilizó el escáner VEGA de Linux, estas pruebas se realizaron a 10 de los 13 servicios web que se encontraron activos. En el escáner VEGA se introduce la Url de cada servicio web donde se identifican las vulnerabilidades, cada escaneo duro más de 48 horas [20].

Características de Vega Escáner

Vega es una herramienta que cuenta con dos modos de operación: escáner automático y proxy de interceptación. El escáner automatizado fue el módulo utilizado en el análisis del SGD, éste rastrea automáticamente sitios web, extrae enlaces, procesa formularios y ejecuta módulos en los posibles puntos de inyección que descubre. Estos módulos pueden hacer cosas como probar scripts de sitios cruzados (XSS) o inyección de SQL. Así como algunas herramientas o escáneres web, se tienen separadas las vulnerabilidades encontradas por categorías según el nivel de peligro, en rojo se encuentran las vulnerabilidades de alto riesgo, con naranja las de medio riesgo, con verde las vulnerabilidades que son de bajo riesgo y con azul la información que ha surgido del análisis [19].

Modo Escaneo

El mismo esta basado en un estudio de seguridad en el modo escaneo. En este modo, el programa descarga y rastrea la página web que deseemos, con el objetivo de encontrar posibles puntos de fallo. Para encontrar estos puntos de fallo utiliza módulos ampliamente conocidos como por ejemplo Cross Site Scripting (XSS), SQL Injection, Directorio transversal, Url Injection, Detección de errores, etc. Dentro de este modo, existen diferentes parámetros configurables explicados en: - Número total de paths descendentes. - Número total de paths hijos de un modo concreto: Limita el numero máximo de hijos escaneados en un nodo determinado. - Número máximo de subpaths escaneados: Limita el número de paths escaneados en la jerarquía de la página. - Número máximo de escaneos a paths repetidos. -Longitud máxima de los resultados generados: El usuario puede configurar el nivel de detalle en los resultados obtenidos. - Número máximo de escaneos por segundo [21].

- **Las imágenes que se presentan en el capítulo cuatro y los anexos A,B,C,D,E y F son creadas por el investigador para el presente proyecto.**

Pasos para iniciar y usar el escáner VEGA.

Para lanzar un escaneo por defecto simplemente se debe pinchar en el icono de "New Scan".

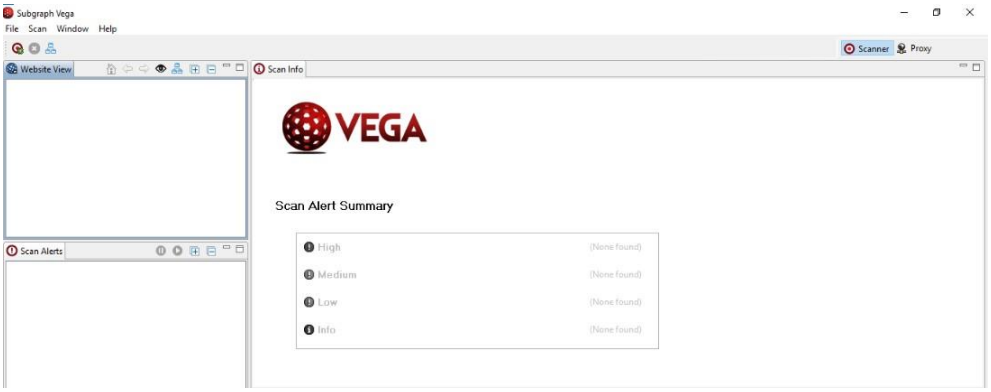


Figura 4.1: Interfaz de inicio de Vega Scan

Según la **figura 4.1:** Muestra la interfaz de inicio de la herramienta VEGA, misma que contiene 3 ventanas con opciones diferentes la una es Website View, que es en donde muestra la URL que se está analizando, también está la opción Scan Alerts que muestra las vulnerabilidades encontradas según el riesgo de cada

una, y Scan Alert Summary que muestra el resumen de el proceso final cuando ha terminado de ejecutar la herramienta el debido análisis lo muestra según su categoría y riesgo.

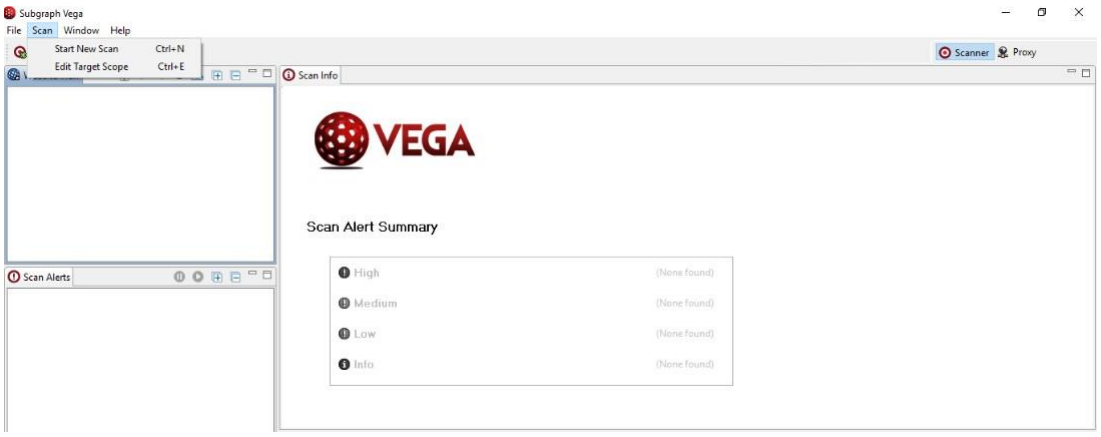


Figura 4.2: Interfaz de inicio de Vega Scan, donde se tiene que escoger "Start New Scan"

Según se observa **figura 4.2** muestra la pantalla de inicio de Vega al hacer clic sobre la opción Scan, que a su vez tiene dos funcionalidades Start New Scan, permite iniciar el análisis de vulnerabilidades al escribir el objetivo específico y Edit Target Scope que permite editar el objetivo a analizar en el caso que hubiese algún error en la URL.

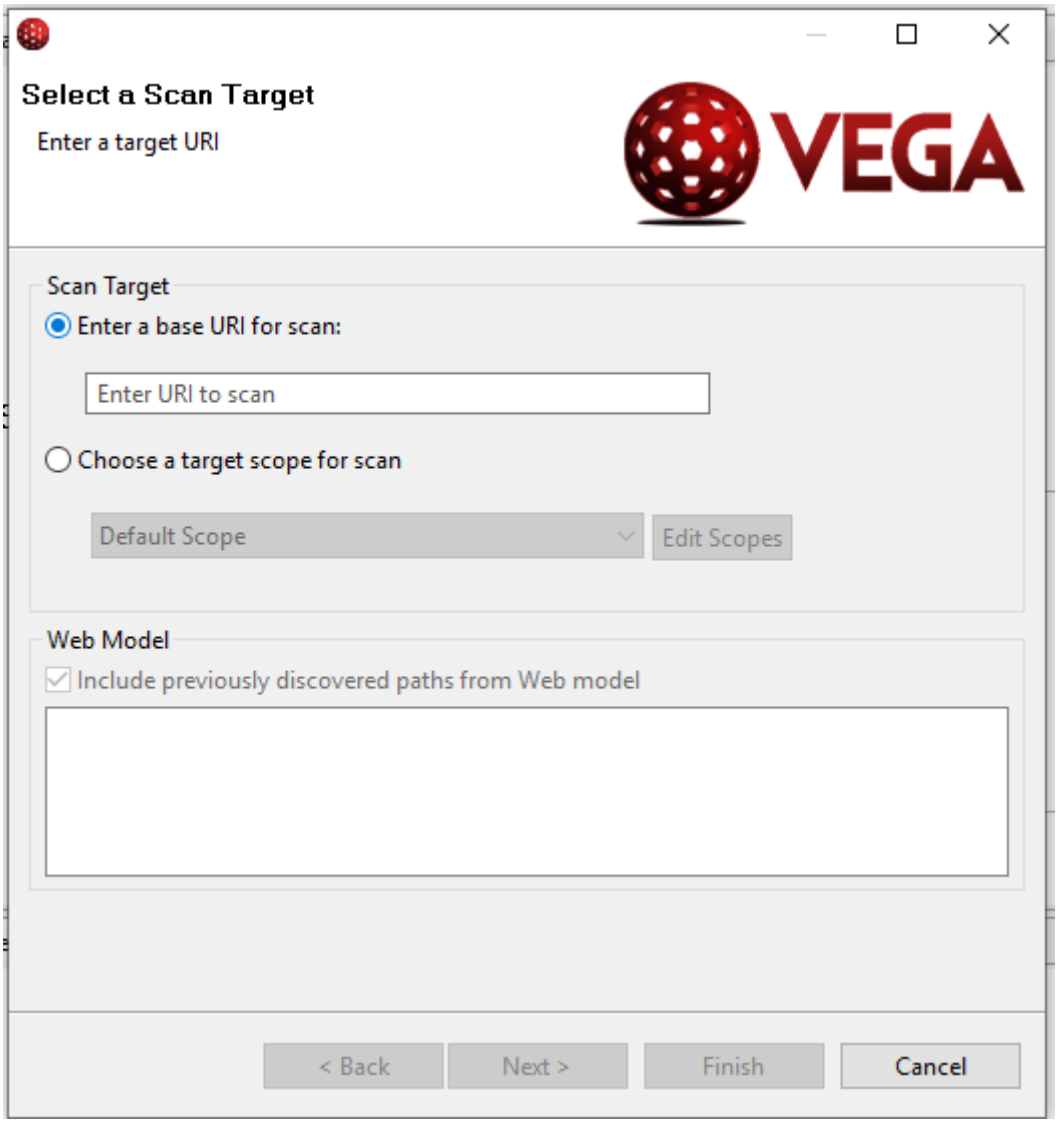


Figura 4.3: Interfaz que indica la opción para empezar un nuevo escáner de vulnerabilidades a analizar.

Como se observa la **figura 4.3:** muestra la ventana de la opción "Start New Scan", donde está seleccionado por defecto la elección de Enter a base URI for scan, donde se colaca la URL, del sitio web que se desea analizar.

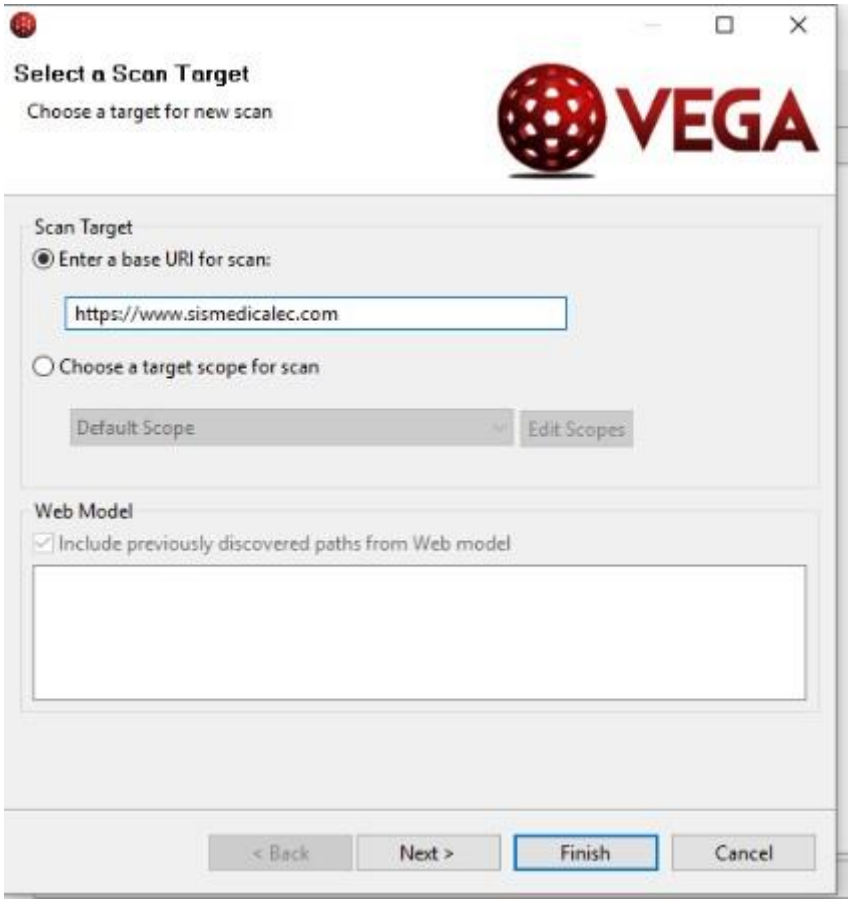


Figura 4.4: Interfaz donde se debe colorar la opción “Enter a base URI for scan” y se debe colocar la Url del sistema web que se va a analizar.

En la **figura 4.4:** Muestra la ventana donde se coloca el URL objetivo a analizar si es correcta se procede a finalizar la opción Scan Target para que inicie el análisis correspondiente.

Las demás opciones deben quedar por defecto y finalmente se hace un clic en "Finish". Una vez que inicia el proceso de análisis automáticamente comenzará el escaneo. Una vez finalice el proceso de escaneo, se obtendrá un reporte con los resultados de la prueba. En la pantalla principal de aplicaciones se puede ver un resumen de las vulnerabilidades obtenidas. En la ventana llamada "Scan Alerts" es posible encontrar un análisis detallado de las vulnerabilidades.

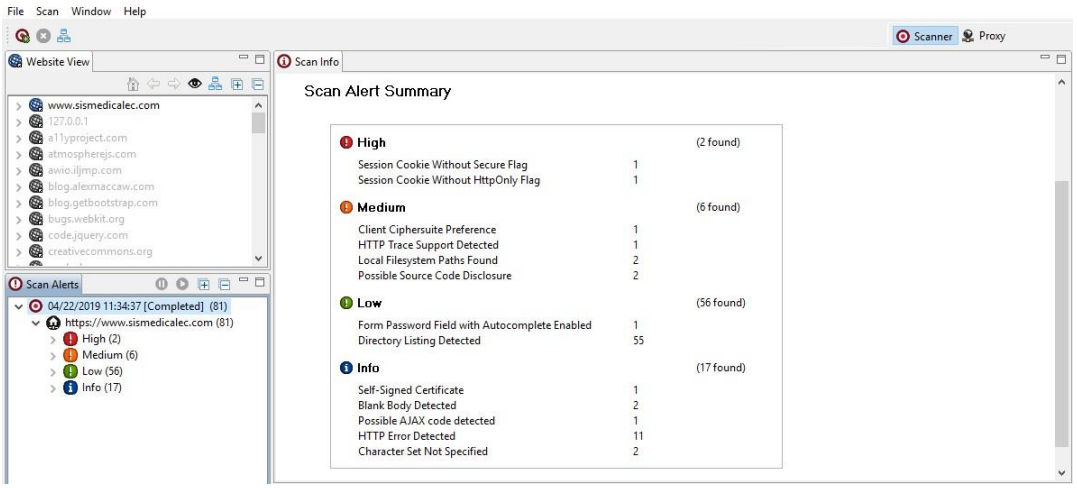


Figura 4.5: Esta interfaz muestra cuando el análisis del sistema web ha finalizado.

En la presente **figura 4.5:** Indica una pantalla la misma que presenta el análisis de alertas, la cantidad de las mismas según su nivel de riesgo que en este caso es Alto, Medio, Bajo y tipo Información, en la parte de "Scan Alert Summary" muestra un resumen de cada una de las vulnerabilidades detectadas.
Resultados del escaneo del sistema web con la herramienta VEGA.

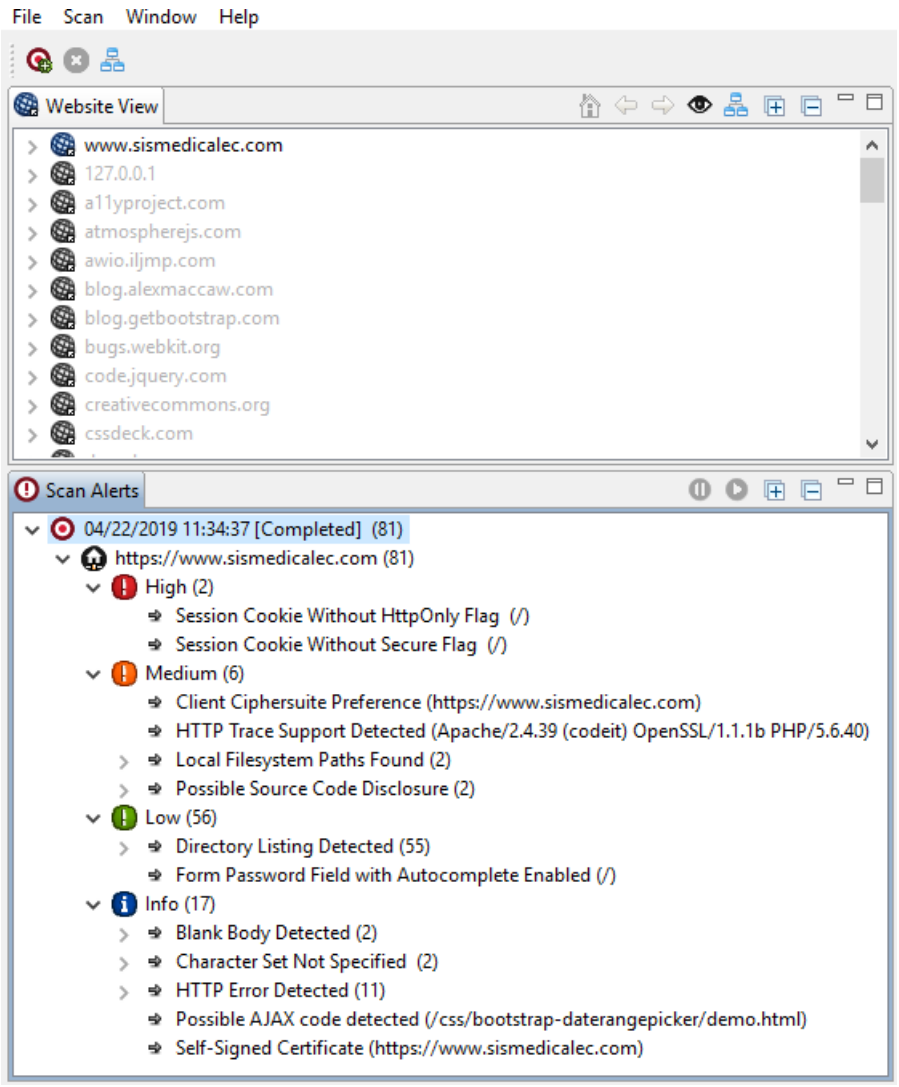


Figura 4.6: Interfaz de Vega Scan, muestra a detalle todas las alertas que emitió luego del escáner de vulnerabilidades.

Según la **figura 4.6:** Indica la ventana Scan Alerts, la misma que presenta la fecha y hora de el análisis de alertas que son presentadas según su nivel de riesgo, con un color diferente para poder diferenciarlas, el color rojo es asignado para las alertas de alto riesgo, el color naranja para las de riesgo medio, el color verde para las de riesgo bajo y el color azul es para las de tipo informativas.

Resultados del análisis web del sistema Sismedicalec, presentados según su grado de peligrosidad.

Vulnerabilidades de Alto riesgo

- Session Cookie Without HttpOnly Flag

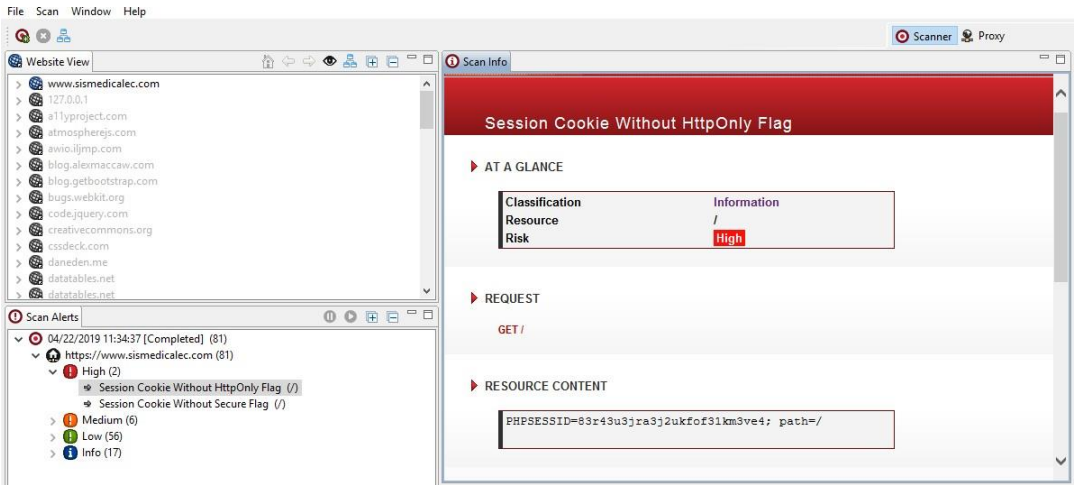


Figura 4.7: Muestra alerta, Session Cookie Without HttpOnly Flag

Según la **figura 4.7:** Presenta la primera alerta detectada en el nivel de riesgo alto, que es Session Cookie Without HttpOnly Flag y el resumen de dicha alerta que contiene también un contenido del recurso, la discusión y recomendación para corregir esta vulnerabilidad.

Contenido del recurso

PHPSESSID=83r43u3jra3j2ukfof31km3ve4; path=/

Discusión

Vega ha detectado que una cookie de sesión puede haberse configurado sin la marca HttpOnly. Cuando este indicador no está presente, es posible acceder a la cookie a través del código de script del lado del cliente. La bandera HttpOnly es una medida de seguridad que puede ayudar a mitigar el riesgo de ataques de scripts entre sitios que se dirigen a las cookies de sesión de la víctima. Si se establece la marca HttpOnly y el navegador admite esta función, el código de secuencia de comandos provisto por el atacante no podrá acceder a la cookie.

Recomendación

Al crear la cookie en el código, establezca la marca HttpOnly en verdadero.

- Session Cookie Without Secure Flag

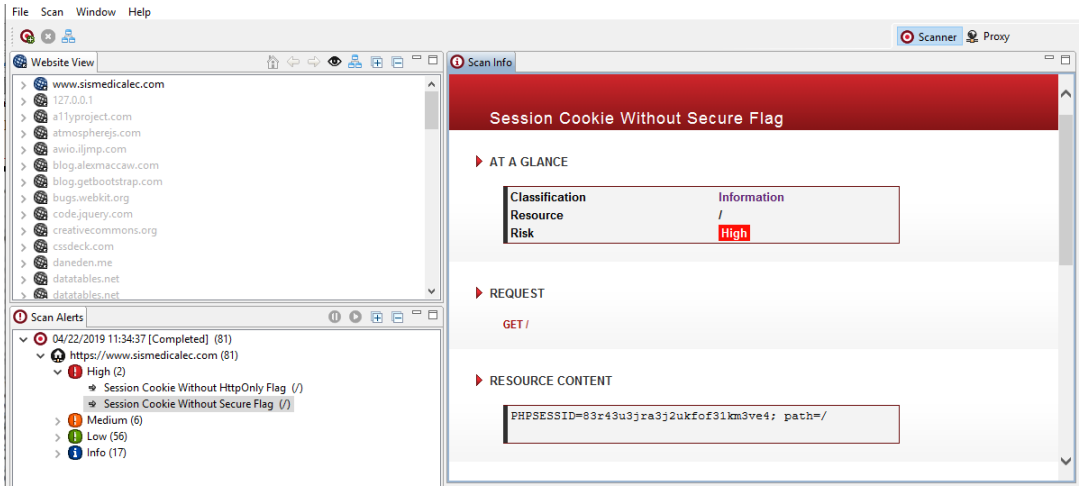


Figura 4.8: Muestra la alerta, Session Cookie Without Secure Flag

En la presente **figura 4.8:** Indica la ventana Scan Alerts en la parte izquierda con la alerta detectada "Session Cookie Without Secure Flag", el resumen de la misma incluido el contenido del recurso, la discusión de la alerta, impacto y su respectiva recomendación para corregir la vulnerabilidad.

Contenido de recurso

PHPSESSID=83r43u3jra3j2ukfof31km3ve4; path=/

Discusión

Vega ha detectado que una cookie de sesión conocida, puede haberse configurado sin el indicador de seguridad.

Impacto

Las cookies pueden estar expuestas a escuchas ilegales de la red. Las cookies de sesión son credenciales de autenticación; los atacantes que los obtienen pueden obtener acceso no autorizado a las aplicaciones web afectadas

Recomendación

Al crear la cookie en el código, establezca el indicador de seguridad en verdadero.

Vulnerabilidades de Riesgo Medio

- Client Ciphersuite Preference

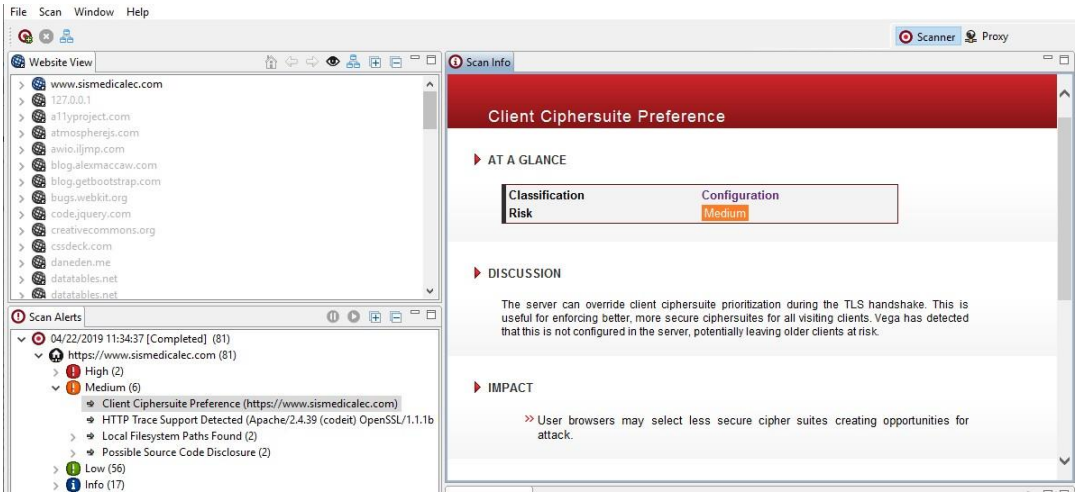


Figura 4.9: Muestra la alerta Client Ciphersuite Preference

Según la **figura 4.9:** Muestra la alerta de riesgo medio "Client Ciphersuite Preference", con su respectiva discusión, impacto y recomendación para corregir dicha vulnerabilidad.

Discusión

El servidor puede anular la priorización del conjunto de cifrado del cliente durante el protocolo de enlace TLS. Esto es útil para hacer cumplir conjuntos de cifrados mejores y más seguros para todos los clientes visitantes. Vega ha detectado que esto no está configurado en el servidor, lo que potencialmente deja a los clientes más antiguos en riesgo.

Impacto

Los navegadores de los usuarios pueden seleccionar suites de cifrado menos seguras y crear oportunidades de ataque.

Recomendación

El servidor HTTPS debe configurarse para imponer las preferencias del conjunto de cifrado del servidor. Cómo se configura esto variará según el servidor.

- HTTP Trace Support Detected

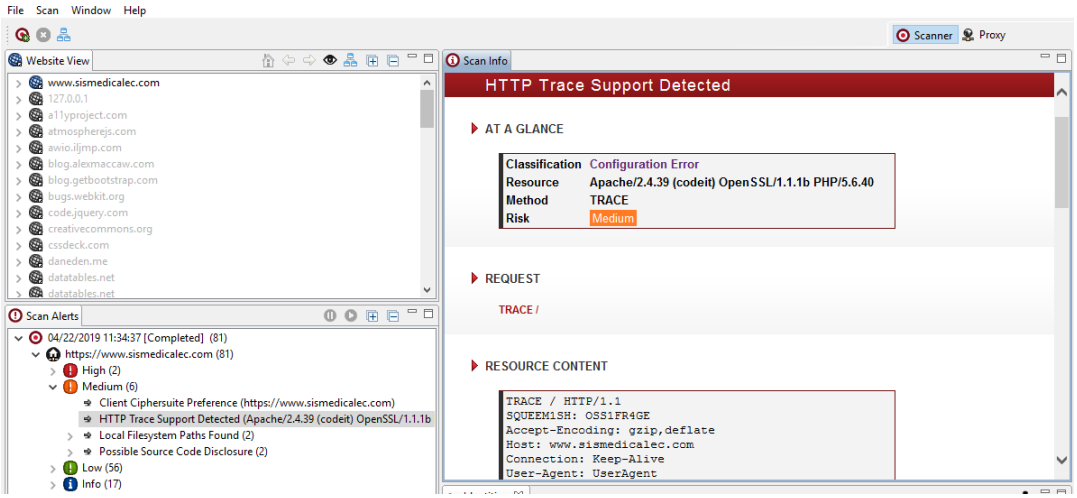


Figura 4.10: Muestra la alerta HTTP Trace Support Detected

Como esta en la **figura 4.10:** Indica la alerta " HTTP Trace Support Detected" con nivel de riesgo medio, incluye un resumen de la vulnerabilidad con su respectivo contenido del recurso detectado, discusión, impacto y recomendación para encontrar una posible solución.

Contenido de recurso

TRACE / HTTP/1.1
SQUEEM1SH: OSS1FR4GE
Accept-Encoding: gzip,deflate
Host: www.sismedicalec.com
Connection: Keep-Alive
User-Agent: UserAgent
Cookie: PHPSESSID=83r43u3jra3j2ukfof31km3ve4
Cookie2: $Version=1

Discusión

HTTP TRACE es un método HTTP que solicita que el servidor devuelva la solicitud TRACE al cliente. Esto incluye los encabezados que se enviaron junto con la solicitud. Se puede abusar del soporte para HTTP TRACE en escenarios donde se ha encontrado una vulnerabilidad de secuencias de comandos entre sitios, pero no se puede aprovechar para recuperar valores de cookies porque las cookies de destino se configuran con el indicador HttpOnly. La bandera HttpOnly indica a los navegadores que no permitan el acceso a la cookie con Javascript. Si se encuentra una vulnerabilidad de secuencias de comandos entre sitios, pero la cookie de sesión está configurada como HttpOnly, el soporte para HTTP TRACE abrirá una oportunidad para el robo de cookies. Un atacante puede usar la vulnerabilidad de los scripts entre sitios para que el navegador del usuario de

destino emita una solicitud TRACE al servidor a través de XMLHttpRequest (o una función similar) y luego recupere la cookie de la respuesta, que contendrá la solicitud que fue enviada por navegador, incluidas las cookies.

Impacto

Permitir HTTP TRACE puede permitir el rastreo entre sitios. Los atacantes pueden usar el rastreo entre sitios con scripts entre sitios para recuperar el valor de las cookies de HttpOnly

Recomendación

Para servidores basados en Apache, la directiva TraceEnable se puede usar para deshabilitar el soporte para HTTP TRACE. Para los servidores basados en IIS, la configuración del registro EnableTraceMethod controla la compatibilidad con HTTP TRACE.

- Local Filesystem Paths Found

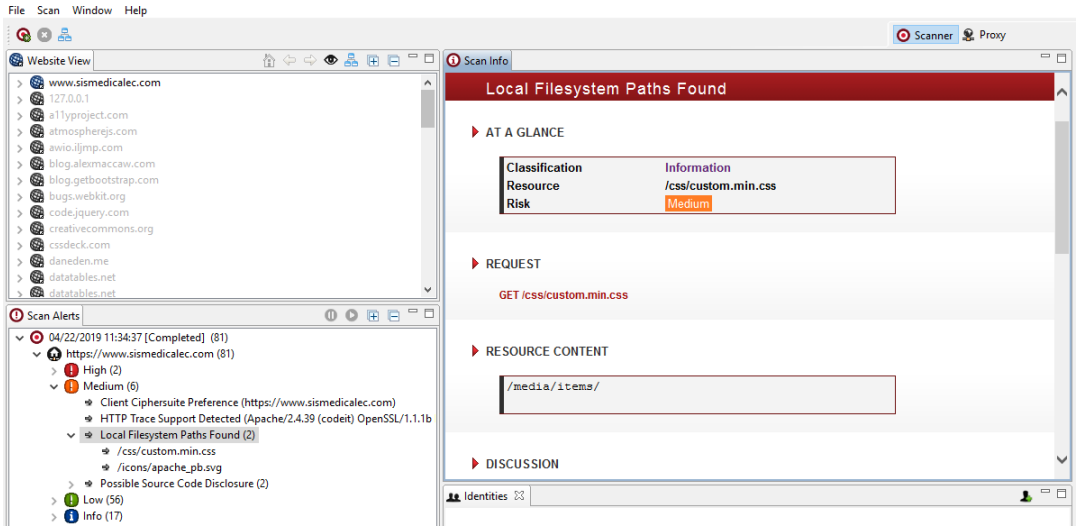


Figura 4.11: Muestra la alerta Local Filesystem Paths Found

En la **figura 4.11:** Presenta la alerta en el nivel de riesgo medio, “Local Filesystem Paths Found”, incluye el respectivo resumen con una solicitud, un contenido de recurso, la discusión, impacto y también las respectivas correcciones a tomar en cuenta para buscar corregir la vulnerabilidad.

Contenido del recurso

/media/items/

Discusión

Vega ha detectado una posible ruta absoluta del sistema de archivos (es decir, una que no es relativa a la raíz web). Esta información es confidencial, ya que puede revelar cosas sobre el entorno del servidor a un atacante. Conocer el diseño

del sistema de archivos puede aumentar las posibilidades de éxito de los ataques ciegos. Las rutas completas del sistema se encuentran muy a menudo en la salida de error Esta salida nunca debe enviarse a clientes en sistemas de producción. Debe ser redirigido a otro canal de salida (como un registro de errores) para que lo analicen los desarrolladores y administradores del sistema.

Impacto

Vega ha detectado lo que pueden ser rutas de acceso absolutas del sistema de archivos en el contenido escaneado. La divulgación de estas rutas revela información sobre el diseño del sistema de archivos. Esta información puede ser confidencial, su divulgación puede aumentar las posibilidades de éxito para otros ataques

Recomendación

Las rutas absolutas se encuentran a menudo en la salida de error. Tanto los administradores del sistema como los desarrolladores deben ser informados, ya que el problema puede deberse a un error de la aplicación o una mala configuración del servidor. La salida de error que contiene información confidencial, como las rutas de acceso absolutas del sistema, no debe enviarse a clientes remotos en servidores de producción. Esta salida debe enviarse a otra secuencia de salida, como un registro de errores.

- Possible Source Code Disclosure

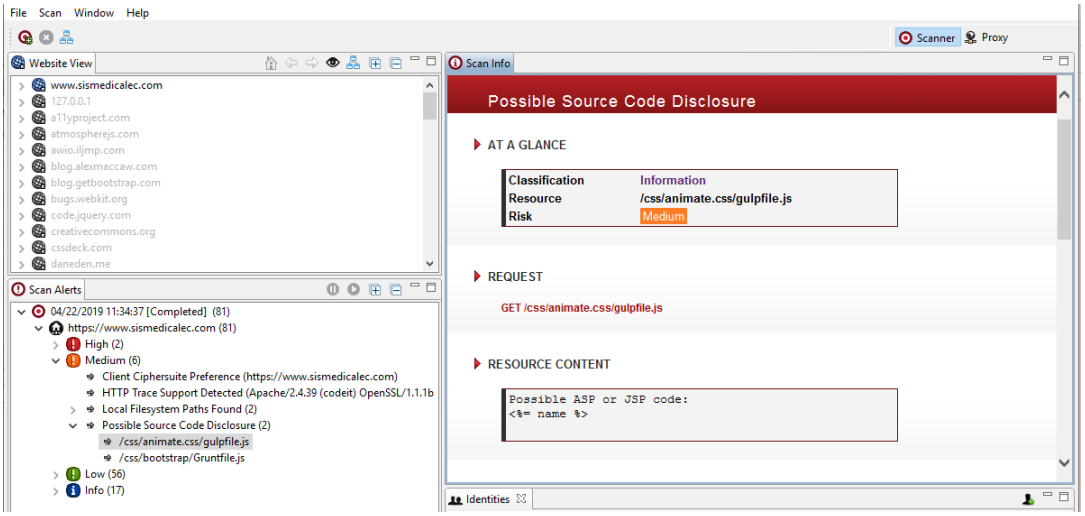


Figura 4.12: Muestra alerta Possible Source Code Disclosure

Según la **figura 4.12:** Muestra la última alerta de riesgo medio, " Possible Source Code Disclosure", incluyendo una solicitud, un contenido recurso, una discusión de la vulnerabilidad, el impacto que tiene la misma y las recomendaciones a seguir.

Contenido del recurso

Possible ASP or JSP code: < %= pkg.version %>

Discusión

Vega ha detectado fragmentos de texto que coinciden con las firmas del código fuente de la aplicación. El código fuente de la aplicación no visible para los clientes remotos puede ser una vulnerabilidad de seguridad. Esto puede ocurrir en aplicaciones que utilizan tecnologías como PHP y JSP, que permiten que el código se mezcle con el contenido de la presentación estática. Por ejemplo, el código en línea a veces se comenta utilizando comentarios HTML, lo que da como resultado que se transmita a clientes remotos. Para un atacante, el código fuente puede revelar información sobre la naturaleza de la aplicación, como su diseño o el uso de componentes de terceros. En ocasiones, la información confidencial, como una cadena de conexión a la base de datos, puede incluirse en el código fuente.

Impacto

Podría dar lugar a la divulgación de información sensible a los atacantes. Los fragmentos del código fuente pueden incluir información sobre el diseño / estructura de la aplicación, incluido el uso de componentes de terceros. Esta información no puede ser fácilmente conocida por un adversario. A veces, el código fuente también contiene información altamente confidencial, como contraseñas (cadenas de conexión de base de datos).

Recomendación

El desarrollador debe verificar que la salida detectada por Vega sea en realidad el código fuente de la aplicación. Se debe determinar la causa y eliminar o impedir que el material salga.

Vulnerabilidades de Riesgo Bajo

- Directory Listing Detected

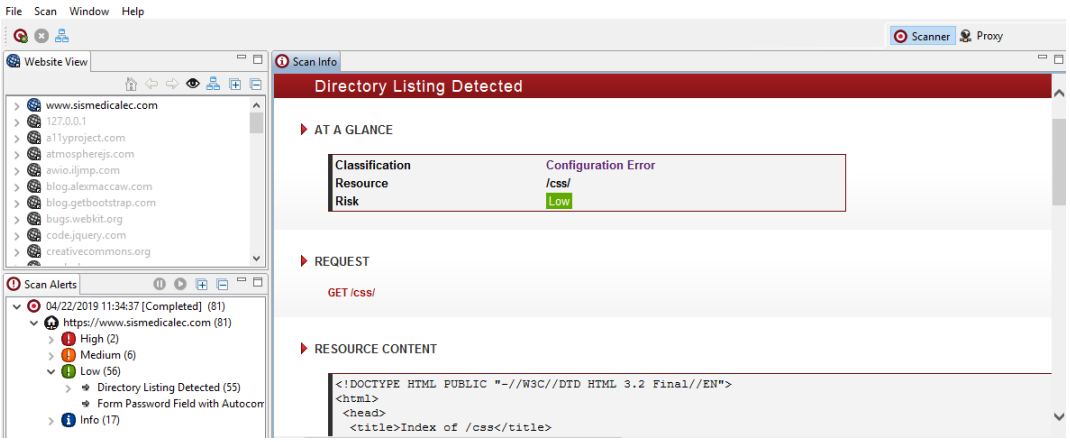


Figura 4.13: Muestra alerta Directory Listing Detected

En la **figura 4.13:** Presenta un nuevo nivel de riesgo que este caso es bajo, con la alerta "Directory Listing Detected", que contiene una solicitud, contenido del recurso, discusión de la alerta, el impacto que causa y sus respectivas recomendaciones a tomar en cuenta.

Contenido del recurso

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<html>
 <head>
  <title>Index of /css</title>
 </head>
 <body>
<h1>Index of /css</h1>
  <table>
   <tr><th valign="top"><img src="/icons/blank.gif" alt="[ICO]"></th><th><a href="?C=N;O=D">Name</a></th><th><a href="?C=M
   <tr><th co...
```

Figura 4.14: Contenido gráfico de Resource Content

Como indica **figura 4.19:** Presenta a detalle el código html en el contenido del recurso de la alerta "Directory Listing Detected".

Discusión

Listado de contenidos del directorio cuando no hay ningún archivo de índice presente en una configuración errónea común. El contenido del directorio puede proporcionar información útil a un atacante, especialmente si hay archivos que no están destinados a ser accesibles, como el código fuente o las copias de seguridad. La lista de directorios también puede proporcionar información útil sobre los hábitos de la administración del servidor y / o los desarrolladores web, como la convención de nomenclatura de archivos, que podría utilizarse para aumentar el éxito probable de la fuerza bruta u otros ataques.

Impacto

El servidor está generando el contenido de los directorios. Esto podría exponer archivos no destinados a la recuperación del usuario (archivos htaccess antiguos,

copias de seguridad, código fuente). La lista de directorios también puede proporcionar información útil sobre el diseño y las características del sistema, como las convenciones de denominación utilizadas por los desarrolladores y administradores. Esta información puede aumentar la probabilidad de éxito para ataques ciegos y adivinanzas de fuerza bruta.

Recomendación

Para Apache, realice una de las siguientes acciones: agregue "IndexIgnore *" al archivo .htaccess del directorio, o bien elimine "Índices" de la "Options All Indexes FollowSymLinks MultiViews" en su archivo de configuración de Apache. Para lighttpd, cambie "dir-listing.activate =" enable "" a "dir-listing.activate =" disable "" en su archivo de configuración lighttpd.

- Form Password Field with Autocomplete Enabled

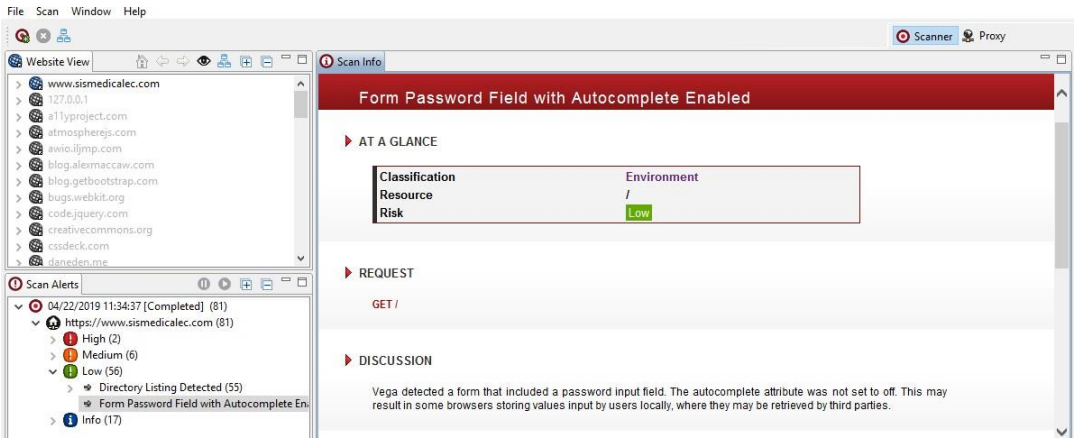


Figura 4.15: Muestra alerta Form Password Field with Autocomplete Enabled

En la **figura 4.15:** Muestra en el riesgo bajo la alerta "Form Password Field with Autocomplete Enabled", con su respectiva solicitud, discusión de la misma y también las recomendaciones a seguir.

Discusión

Vega detectó un formulario que incluía un campo de entrada de contraseña. El atributo autocompletar no se desactivó. Esto puede hacer que algunos navegadores almacenen valores ingresados por los usuarios localmente, donde pueden ser recuperados por terceros.

Impacto

Un valor de contraseña se puede almacenar en el sistema de archivos local del cliente. Las contraseñas almacenadas localmente pueden ser recuperadas por otros usuarios o código malicioso.

Recomendación

La declaración de formulario debe tener un atributo de autocompletado con su valor establecido en "off".

Información que arrojó como resultado del análisis al aplicativo web

- Blank Body Detected

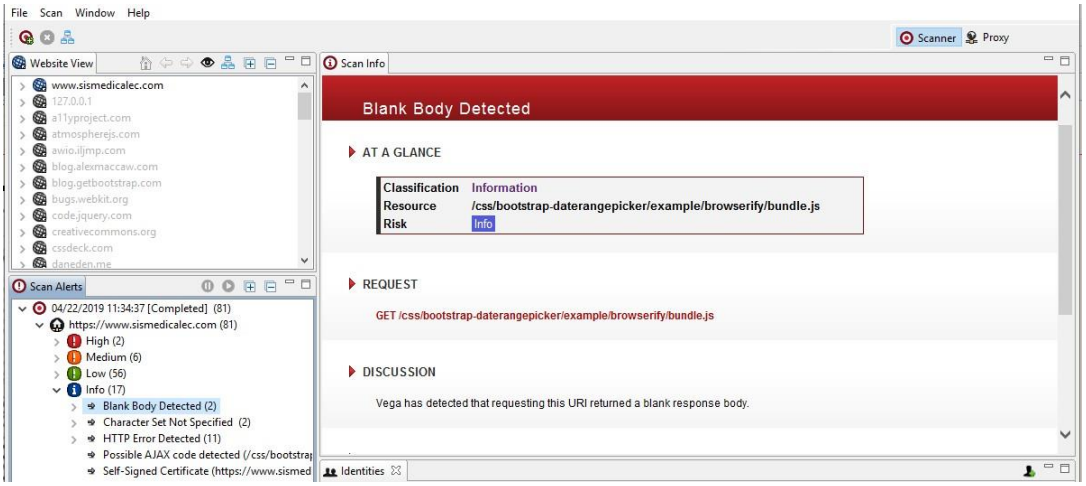


Figura 4.16: Muestra de información Blank Body Detected

Según la **figura 4.16:** Presenta el nivel de tipo información, “Blank Body Detected”, y un resumen donde detalla la solicitud, la discusión respectiva, el impacto y las recomendaciones a tomar en cuenta.

Solicitud

GET /css/bootstrap-daterangepicker/example/browserify/bundle.js

Discusión

Vega ha detectado que la solicitud de este URI devolvió un cuerpo de respuesta en blanco.

Impacto

Esto puede ser indicativo de una condición de error y debe investigarse manualmente.

Recomendación

El desarrollador debe investigar por qué ocurrió esto y si existen implicaciones de seguridad.

- Character Set Not Specified

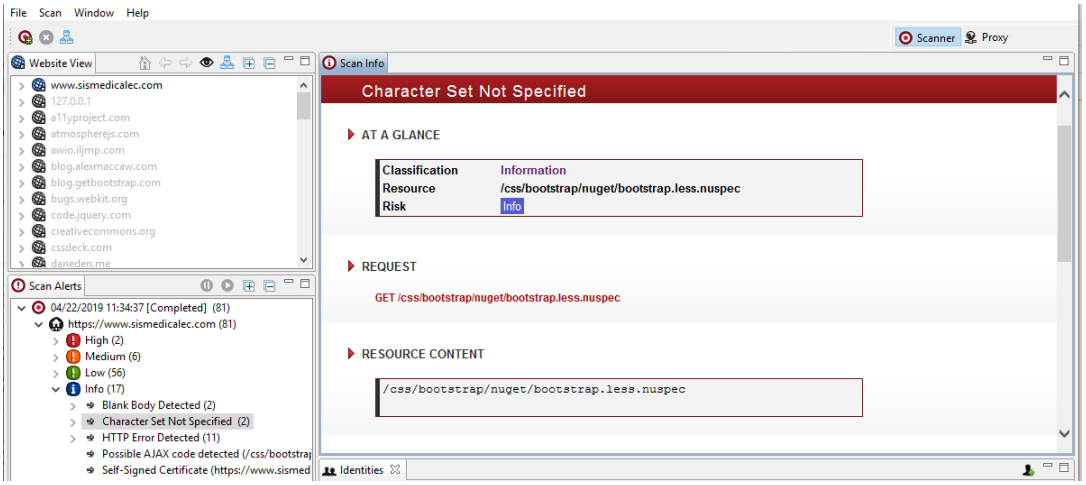


Figura 4.17: Muestra de informácion Character Set Not Specified

En la **figura 4.17:** Indica la opción "Character Set Not Specified", de tipo información con su respectivo contenido del recurso, discusión y recomendaciones a seguir, en esta sección no presenta el impacto como en las alertas antes expuestas.

Contenido del recurso

/css/bootstrap/nuget/bootstrap.less.nuspec

Discusión

Vega ha detectado que el recurso no ha especificado un conjunto de caracteres en la respuesta. Si no se especifica el conjunto de caracteres, el navegador puede hacer suposiciones sobre el conjunto de caracteres en función del contenido del recurso. Esto puede presentar un problema de seguridad si el recurso afectado contiene contenido generado dinámicamente que se origina de los usuarios. En tal caso, los usuarios malintencionados pueden aprovechar la forma en que los navegadores específicos interpretan los caracteres para provocar la representación del contenido malicioso. Por ejemplo, un atacante puede omitir un filtro de scripts entre sitios al codificar su carga maliciosa en un conjunto de caracteres alternativo, que puede ejecutarse según la forma en que el navegador interpreta el contenido codificado.

Recomendación

Especifique un conjunto de caracteres bien definido (como UTF-8) dentro del tipo de contenido del encabezado de respuesta o el cuerpo de la respuesta.

- HTTP Error Detected

Como muestra **figura 4.18:** Presenta la opción de tipo información, "HTTP Error Detected", un resumen con el contenido del recurso, su discusión, impacto y las recomendaciones a tomar en cuenta.

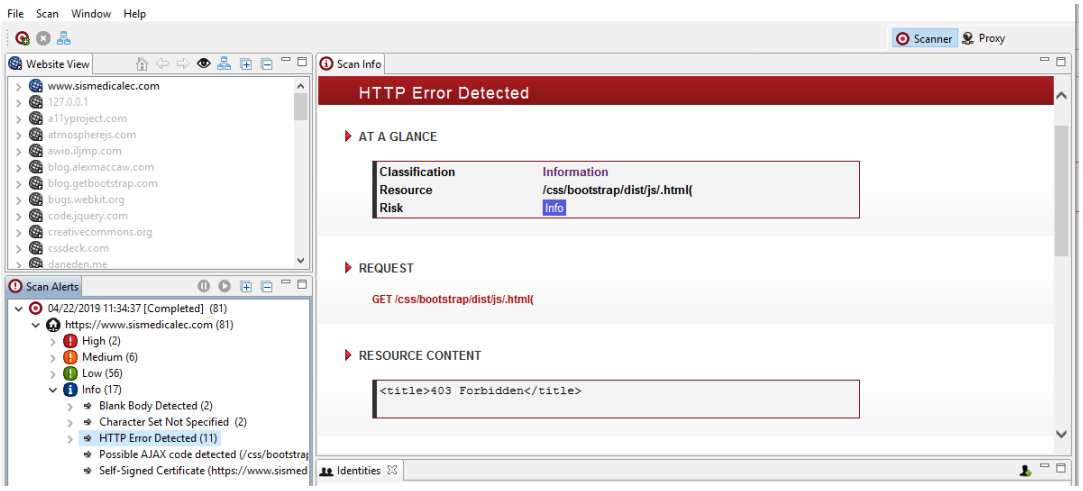


Figura 4.18: Figura que muestra la opción HTTP Error Detected

Contenido del Recurso

<title>403 Forbidden</title>

Discusión:

Una solicitud que Vega ha enviado ha dado como resultado una respuesta HTTP con un código de estado de error. Esto debe investigarse inspeccionando tanto la solicitud como la respuesta.

Impacto:

El código de estado de error indica un evento no identificado en el servidor que puede estar asociado con una vulnerabilidad o un problema de configuración.

Recomendación:

El desarrollador debe investigar cómo o por qué ocurrió este error y asegurarse de que no haya ninguna vulnerabilidad presente.

- Possible AJAX code detected

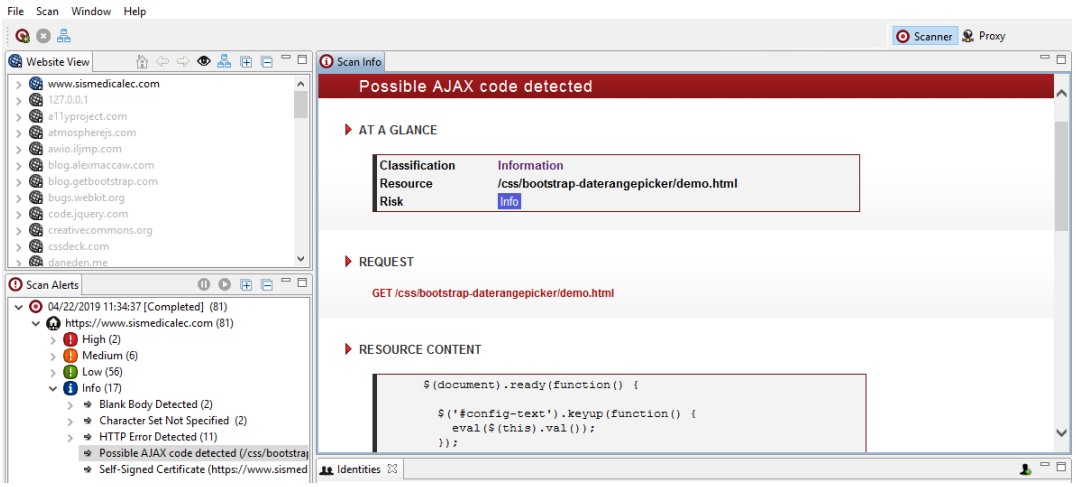


Figura 4.19: Muestra de información Possible AJAX code detected

Según la **figura 4.19:** Muestra la opción de "Possible AJAX code detected", de tipo informativo, con su correspondiente resumen que incluye una solicitud, contenido del recurso, Discusión, impacto que causa y sus respectivas recomendaciones necesarias para dar una posible solución.

Contenido del recurso

$(document).ready(function() {
$('#config-text').keyup(function() {
eval($(this).val());
});
$('.configurator input, .configurator select').change(f...)

Discusión

AJAX (Javascript asíncrono y XML) se refiere a una colección de tecnologías utilizadas para hacer que la experiencia del usuario de las aplicaciones web sea más interactiva. La funcionalidad AJAX a menudo implica el envío asíncrono de solicitudes y el procesamiento de sus respuestas utilizando Javascript, sin necesidad de recargar la página. Los puntos finales en el lado del servidor a menudo aceptan parámetros, lo que los convierte en puntos de inyección donde podrían existir vulnerabilidades.

Impacto

Vega ha detectado contenido en el uso de AJAX, lo que indica la existencia de posibles puntos de inyección donde pueden existir vulnerabilidades. La API de back-end de AJAX debe ser inspeccionada manualmente para detectar vulnerabilidades.

Recomendación

Esto no es una vulnerabilidad. Esta alerta solo indica que el código asociado

con el uso de AJAX se ha detectado en el contenido escaneado. Las interfaces de back-end AJAX pueden exponer posibles vulnerabilidades y la inspección manual debe incluirse en cualquier evaluación de seguridad integral.

- Self-Signed Certificate

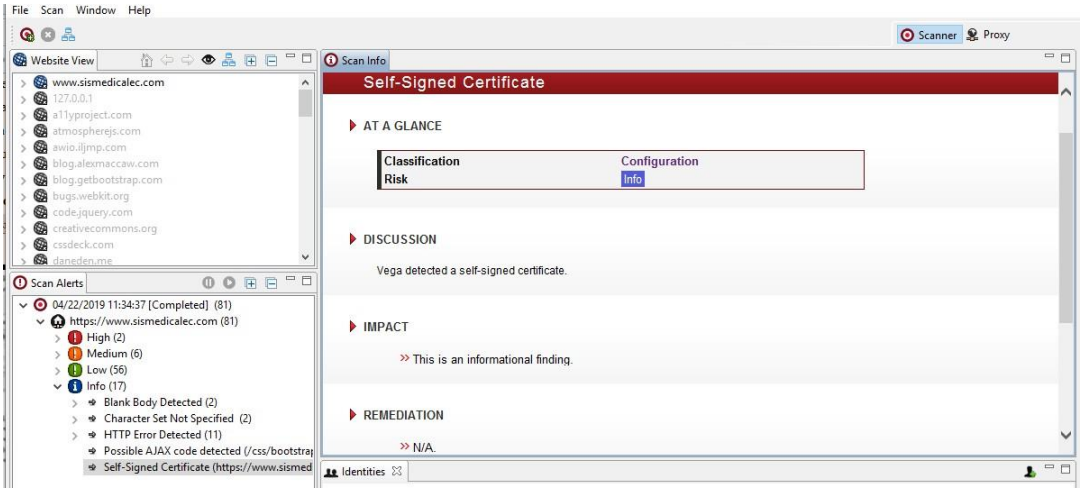


Figura 4.20: Muestra de información Self-Signed Certificate

Como se presenta en la **figura 4.20:** Indica la opción de tipo informativa, “Self-Signed Certificate” la última en el análisis que incluye únicamente una discusión, impacto.

Discusión

Vega detectó un certificado autofirmado.

Impacto

Este es un hallazgo informativo.Estas son las Requests que muestra la herramienta vega luego de realizar el escaneo de vulnerabilidades.

Opcion Requests, resultados finales

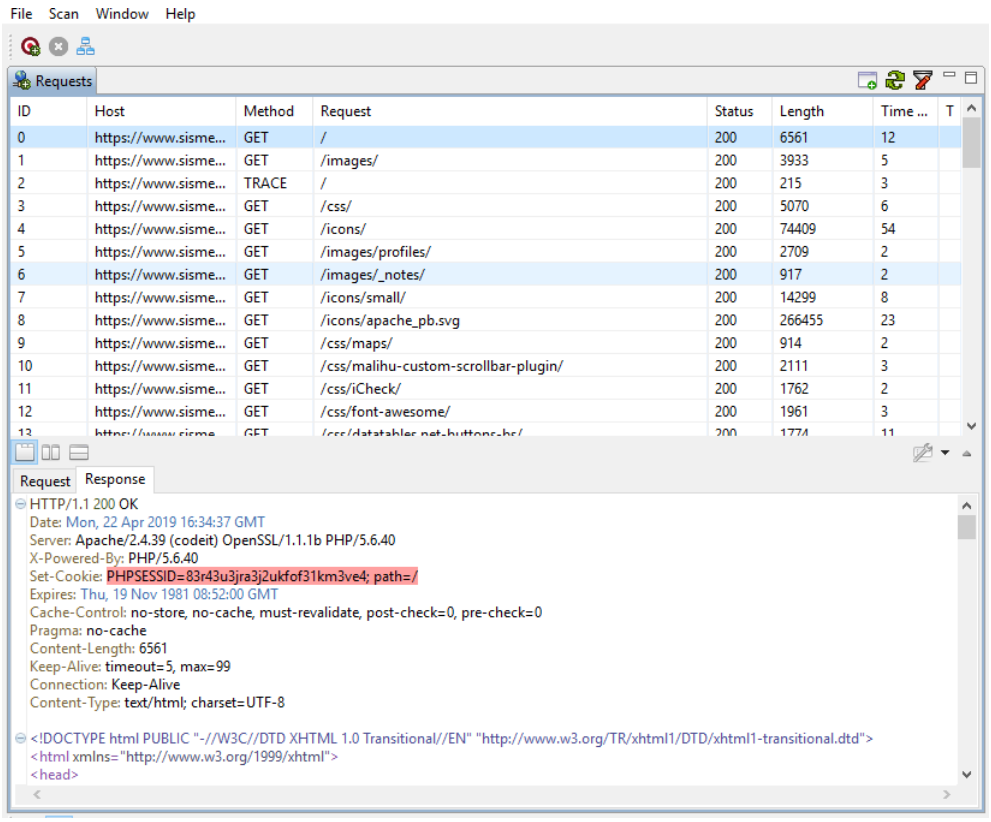


Figura 4.21: Interfaz que muestra la ventana Requests

En la **figura 4.21:** Presenta la ventana Request que contiene los resultados finales del análisis de su respectivo objetivo analizado.

4.3.2. Herramienta Owasp Zap

Herramienta de análisis Owasp Zed Attack Proxy (ZAP) Zed Attack Proxy (ZAP): Herramienta de fácil uso para encontrar vulnerabilidades en aplicaciones web. Está diseñada para ser utilizada tanto por desarrolladores y probadores funcionales (que son nuevos en test de intrusión) como por personas con una amplia gama de experiencia en seguridad. Permite automatizar las pruebas y también facilita un número de herramientas para hacerlas manualmente [22].
Con la finalidad de prevenir que la base de datos muestre alguna, en relación con nuestra herramienta de la OWASP "Zed Attack Proxy" permite a los programadores realizar testeos durante el desarrollo de la aplicación con el fin de encontrar vulnerabilidades y poder corregir alguna vulnerabilidad encontrada en la aplicación al igual permitir realizar la corrección o el análisis en el código directamente en un ambiente gráfico, caso contrario al SQL rand donde se debe implementar este modelo sobre la aplicación que debe estar actualizando este tipo de código para no ser vulnerable (OWASP Zed Attack Proxy Project) [23].

Las principales características con las que cuenta un auditor al utilizar esta herramienta son:

- Herramienta totalmente gratuita y de código abierto.
- Herramienta multi-plataforma, compatible incluso con Raspberry Pi.
- Fácil de instalar, dependiendo únicamente de Java 7 o superior.
- Posibilidad de asignar un sistema de prioridades.
- Traducida a más de 12 idiomas, entre ellos, el español.
- Excelente manual de ayuda y gran comunidad en la red [24].

Algunas de las funcionalidades de ZAP se citan a continuación:

- Funciona como un proxy que intercepta todas las peticiones que se realizan y las respuestas recibidas (útil sobre todo para ver peticiones que no sean evidentes a simple vista). También permite establecer puntos de ruptura para cambiar peticiones y respuesta sobre la marcha.
- Análisis activo de vulnerabilidades. ZAP realiza ataques conocidos contra las aplicaciones web seleccionadas con el objetivo de encontrar vulnerabilidades potenciales. Este análisis sólo puede encontrar ciertos tipos de

vulnerabilidades, por lo que debe completarse con pruebas de penetración manual.

- Análisis pasivo de vulnerabilidades. Este tipo de análisis no modifica las respuestas y no ralentiza la exploración ya que se realiza en segundo plano.
- Detección automática de nuevos recursos (spider). A partir de una lista de enlaces, esta herramienta identifica todos los hipervínculos de la páginas y accede a ellos repitiendo el proceso de forma recursiva. Autenticación y gestión de sesiones.
- Proporciona un API Rest (disponible en JSON, HTML y XML) para la integración con otras herramientas, que permite el uso de las funcionalidades de escaneo activo y spider, aunque en futuras versiones de ZAP se aumentará el número de funcionalidades disponibles a partir del API.
- Actualizaciones automáticas. Plugins integrados y un marketplace de plugins en constante actualización.
- Plugins integrados y un marketplace de plugins en constante actualización [25].

Vulnerabilidades con Owasp Zap

• Procedimiento para el uso de la herramienta Owasp Zap.
Se busca **Owasp Zap** en el listado de herramientas que proporciona Kali Linux, comienza a cargar y se muestra de la siguiente manera.

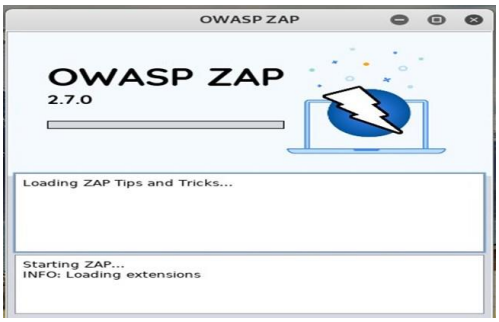


Figura 4.22: Iniciando la herramienta Owasp Zap

Según la **figura 4.22:** Presenta el inicio del escáner OWASP ZAP cuando es seleccionada la herramienta que tiene incluida el sistema operativo Kali Linux para la el análisis y detección de vulnerabilidades web.

Cuando se inicia la aplicación se puede observar una pantalla donde permite restaurar una sesión anterior o iniciar una nueva.
En este ejemplo ya que se procede a iniciar un análisis nuevo no es necesario restaurar ninguna sesión y se despliega una pantalla de bienvenida e inicio del análisis, como se observa en la siguiente figura.

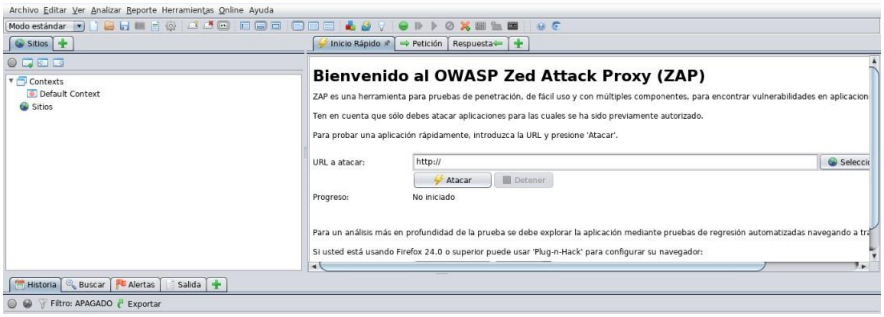


Figura 4.23: Página de bienvenida de la herramienta Owasp Zap.

Como se presenta la **figura 4.23:** Muestra la página de inicio y bienvenida que presenta el escáner Owasp Zap para sus usuarios, también indica la opción para colocar la URL que se va a atacar y el botón Attack que sirve para iniciar el respectivo análisis del sitio web específico.
En la parte derecha de la pantalla contiene un modo de ataque automático o activo. Si se coloca en input la Url de la Web, **https://www.sismedicalec.com** a auditar y se clickea sobre la opción Attack, comenzará a navegar a través de los diferentes enlaces y a descubrir ficheros ocultos de la misma usando diferentes técnicas como la lectura del fichero robots.txt, spider, etc.

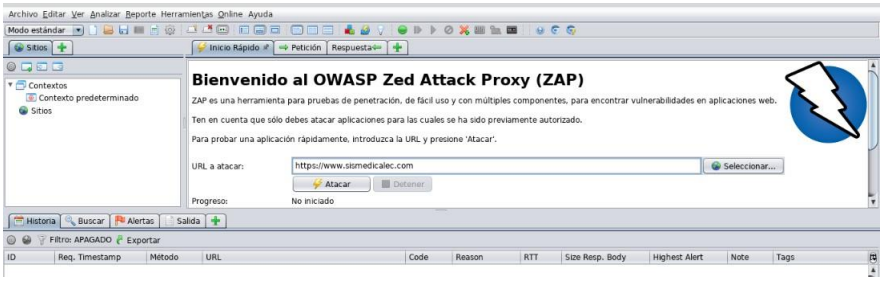


Figura 4.24: Colocación de la Url del sistema web a analizar, al presionar en Atacar empieza el análisis.

Según la **figura 4.24:** Muestra la página de inicio de Owasp Zap con una bienvenida y descripción de la herramienta también está la opción URL atacar donde se escribe el objetivo específico para iniciar el análisis correspondiente.

Alertas encontradas según su nivel de riesgo

Es recomendable empezar a describir cada alerta, por su mayor grado de riesgo encontrado, en este caso seria por **"Riesgo Medio".**

- Encabezado X-Frame-Options no establecido

Url: **https://www.sismedicalec.com/**

Riesgo: medio

Confianza: media

Origen: Pasivo (10020 - Opciones del encabezado del escáner X-Frame)

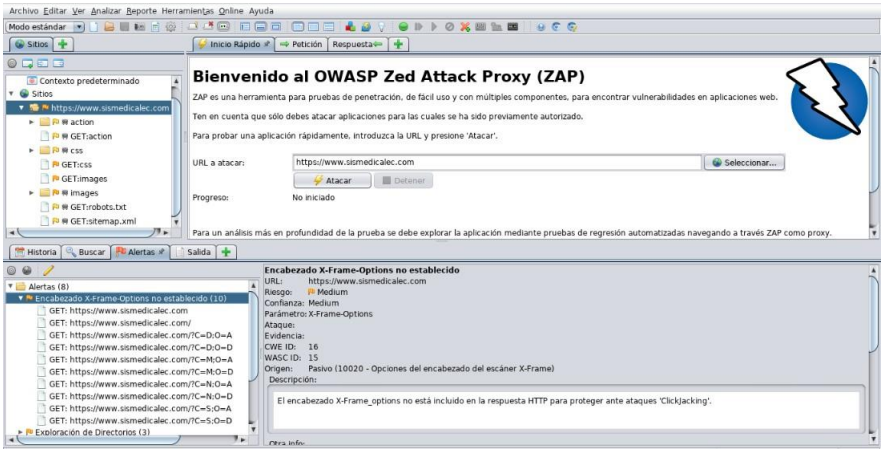


Figura 4.25: Muestra la alerta nombrada como: Encabezado X-Frame-Options no establecido.

Según la **figura 4.25:** Indica la primera alerta detectada en este caso es de riesgo medio, "Encabezado X-Frame-Options no establecido", la URL, parámetros, el origen, pero sobre todo la correspondiente descripción de la alerta, la posible solución sugerido por la herramienta.

Descripción de alerta

El encabezado X-Frame_options no está incluido en la respuesta HTTP para proteger ante ataques 'ClickJacking'.

Solución propuesta por Owasp Zap

Los navegadores de web más modernos apoyan la cabecera HTTP X-Frame-Options. Por ello se debe asegurar que esté establecido en todas las páginas web, de la aplicación, (si se espera que la página este enmarcada solo por páginas en su servidor (por ejemplo, es parte de un FRAMESET, que define la organización de marcos dentro de la ventana del usuario) por otro lado si especifica SAMEORIGIN, puede usar la página en un marco mientras el sitio que la incluya sea el mismo que la sirve, de otra forma o si nunca se espera que la página esté enmarcada, debería usar DENY. ALLOW-FROM para que permite

a los sitios web específicos enmarcar la página web en navegadores web que sean compatibles).

- Exploración de Directorios

Url: **https://www.sismedicalec.com/css/**

Riego: Medio

Confianza: Media

Ataque: Parent Directory

Origen: Activo (0 - Exploración de Directorios)

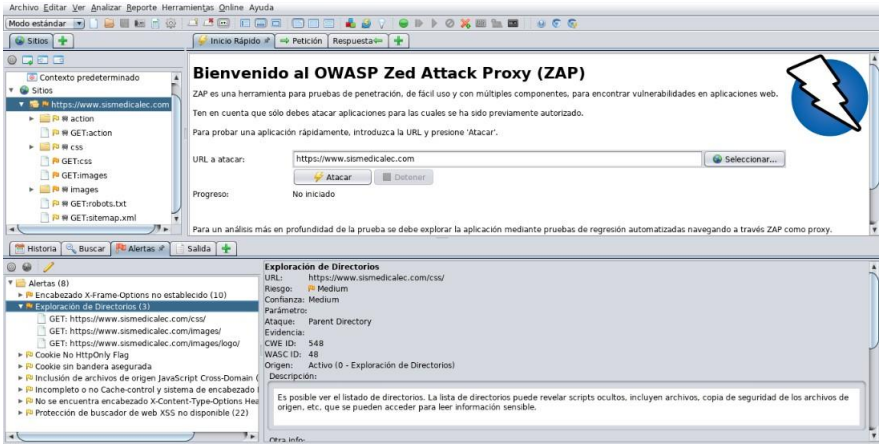


Figura 4.26: Muestra alerta, Exploración de Directorios

Según como se observa en la **figura 4.26:** Presenta la siguiente alerta también de riesgo medio, "Exploración de Directorios", con su respectiva url, al tipo de ataque, origen y la correspondiente descripción de la alerta, y la posible solución recomendada por la herramienta Owasp Zap.

Descripción de la alerta

Es posible ver el listado de directorios. La lista de directorios puede revelar scripts ocultos, incluyen archivos, copia de seguridad de los archivos de origen, etc., que se pueden acceder para leer información sensible.

Solución

Una solución rápida es la creación de un archivo index.html sin contenido en la carpeta publicada, de manera que al ingresar a la ruta solicitada, se expondrá el index.html sin contenido.

Desactivar la exploración de directorios. Si esto es necesario, se debe asegurar de que el archivo de la lista no induce riesgos.

Vulnerabilidades de Riesgo Bajo

- Cookie No HttpOnly Flag

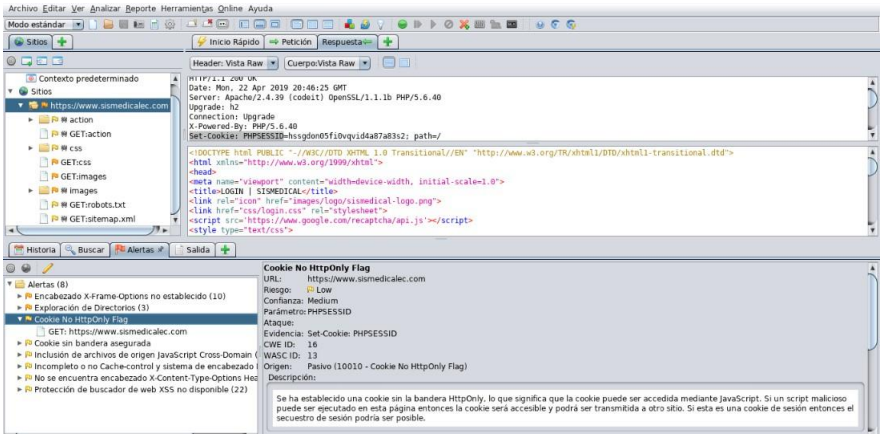


Figura 4.27: Muestra la alerta Cookie No HttpOnly Flag

Como se muestra en la **figura 4.27:** Indica la primera alerta de riesgo bajo, "Cookie No HttpOnly Flag", con su respectiva url analizada, el nivel de confianza, el parámetro, evidencia, origen y lo más destacado la descripción de la alerta y la posible solución a tomar en cuenta.

Descripción de la alerta:

Se ha establecido una cookie sin la bandera HttpOnly, lo que significa que la cookie puede ser accedida mediante JavaScript. Si un script malicioso puede ser ejecutado en esta página entonces la cookie será accesible y podrá ser transmitida a otro sitio. Si esta es una cookie de sesión entonces el secuestro de sesión podría ser posible.

Solución

Debe estar seguro que la bandera HttpOnly esta establecida para todas las cookies.

- Cookie sin bandera asegurada

Url: **https://www.sismedicalec.com**

Riesgo: Bajo

Confianza: Media

Parámetro: Evidencia: Set-Cookie: PHPSESSID

Origen: Pasivo (10011 - Cookie sin bandera asegurada)

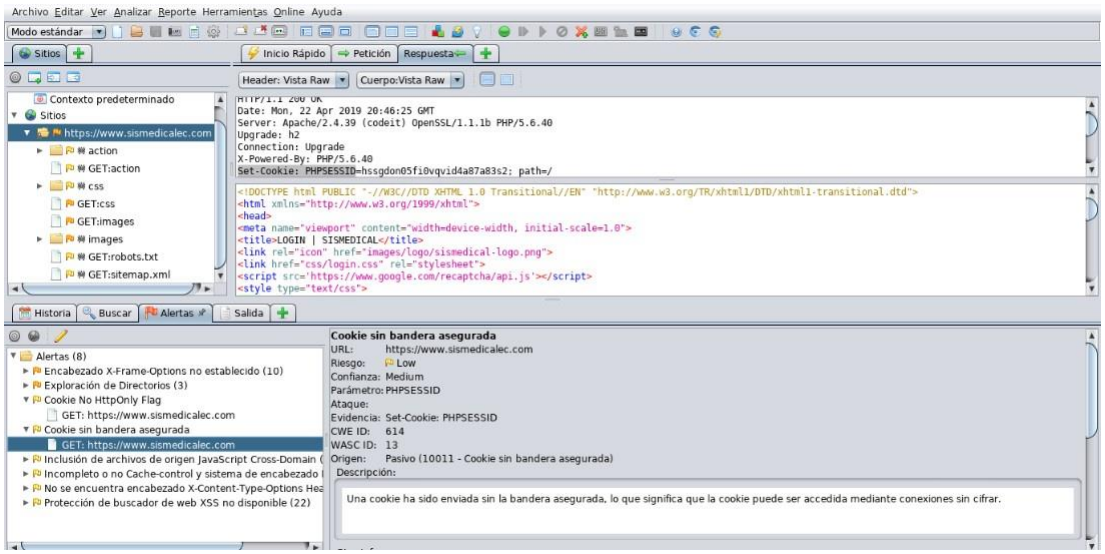


Figura 4.28: Muestra de alerta Cookie sin bandera asegurada.

La **figura 4.28:** Presenta una alerta “Cookie sin bandera asegurada”, de riesgo bajo, con su respectiva url, el nivel de confianza, el parámetro, evidencia, origen y sobre todo la respectiva Descripción de la alerta y solución propuesta por el escáner Owasp Zap.

Descripción de alerta:

Una cookie ha sido enviada sin la bandera asegurada, lo que significa que la cookie puede ser accedida mediante conexiones sin cifrar.

Solución:

Cuando una cookie contiene información sensible o es un token de sesión, debería ser siempre pasada usando un canal cifrado. Debe estar seguro de que la bandera asegurada está establecida para cookies conteniendo información sensible.

- Inclusión de archivos de origen JavaScript Cross-Domain

Url: **https://www.sismedicalec.com/**

Riesgo: Bajo

Confianza: Media

Parámetro: https://www.google.com/recaptcha/api.js

Evidencia: <script src=’https://www.google.com/recaptcha/api.js’></script>

Origen:Pasivo (10017 - Inclusión de archivos de origen JavaScript Cross-Domain)

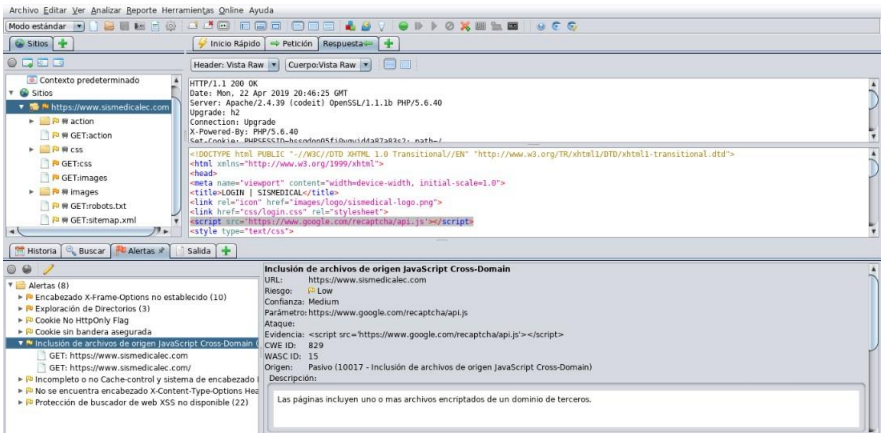


Figura 4.29: Muestra la alerta Inclusión de archivos de origen JavaScript Cross-Domain

Como se muestra en la **figura 4.29:** Indica detalladamente la alerta de riesgo bajo "Inclusión de archivos de origen JavaScript Cross-Domain", con su respectiva url, confianza, el parámetro, evidencia, origen sobre todo la descripción de la alerta y la posible solución a tomar en cuenta.

Descripción de la alerta:

Las páginas incluyen uno o más archivos encriptados de un dominio de terceros.

Solución:

Es importante asegurar que los archivos de la fuente JavaScript están descargados solo de sus fuentes confiables, y las fuentes no pueden ser controladas por los usuarios finales de la aplicación.

- Incompleto o no Cache-control y sistema de encabezado HTTP Pragma

Url: **https://www.sismedicalec.com**

Riesgo: Bajo

Confianza: Media

Parámetro: Cache-Control

Origen: Pasivo (10015 - Incompleto o no Cache-control y sistema de encabezado HTTP Pragma)

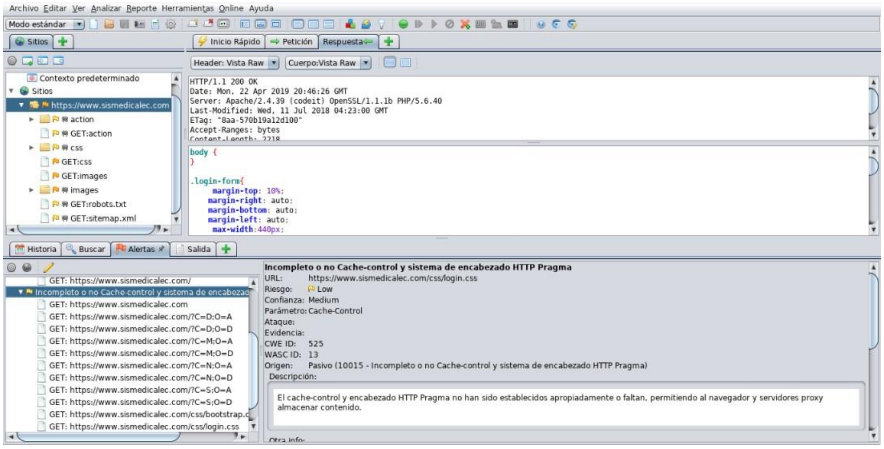


Figura 4.30: Interfaz acerca de la alerta Incompleto o no Cache-control y sistema de encabezado HTTP Pragma

Según la **figura 4.30:** Presenta la alerta con nivel de riesgo bajo denominada " Incompleto o no Cache-control y sistema de encabezado HTTP Pragma", incluye también la url, el nivel de confianza, el parámetro, origen y una descripción correspondiente a la alerta presentada con una posible solución a seguir.

Descripción de la alerta:

El cache-control y encabezado HTTP Pragma no han sido establecidos apropiadamente o faltan, permitiendo al navegador y servidores proxy almacenar contenido.

Solución:

Siempre que sea posible asegurarse que el encabezado HTTP cache-control está establecido con no-cache, no-store, must-revalidate, y que el encabezado HTTP pragma esté establecido con no-cache.

- No se encuentra encabezado X-Content-Type-Options Header.

Url: **https://www.sismedicalec.com/**

Riesgo: Bajo

Confianza: Media

Parámetro: X-Content-Type-Options

Origen: Pasivo (10021 - No se encuentra encabezado X-Content-Type-Options Header)

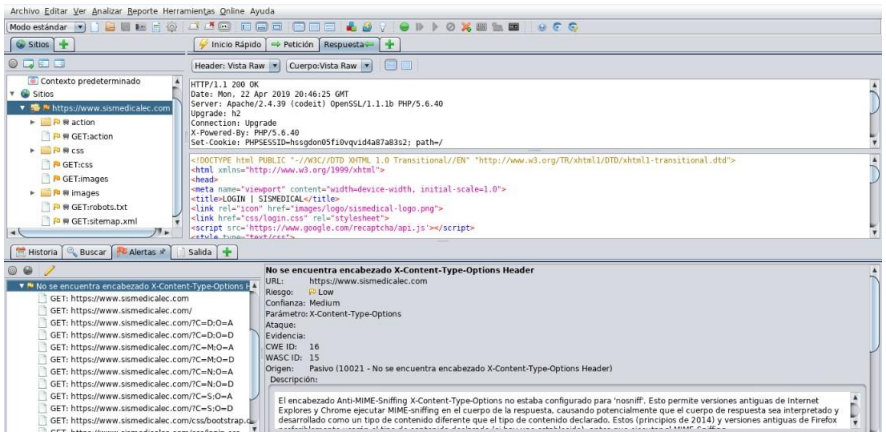


Figura 4.31: Interfaz sobre la alerta No se encuentra encabezado X-Content-Type-Options Header.

Según se observa en la **figura 4.31:** Indica la alerta de riesgo bajo denominada como "No se encuentra encabezado X-Content-Type-Options Header", incluye la url, el nivel de confianza, el parámetro, origen y su respectiva descripción de la alerta presentada con su correspondiente posible solución a seguir.

Descripción de la alerta:

El encabezado Anti-MIME-Sniffing X-Content-Type-Options no estaba configurado para 'nosniff'. Esto permite versiones antiguas de Internet Explores y Chrome ejecutar MIME-sniffing en el cuerpo de la respuesta, causando potencialmente que el cuerpo de respuesta sea interpretado y desarrollado como un tipo de contenido diferente que el tipo de contenido declarado. Estos (principios de 2014) y versiones antiguas de Firefox preferiblemente usarán el tipo de contenido declarado (si hay uno establecido), antes que ejecutar el MIME-Sniffing.

Información:

Este inconveniente aún aplica para páginas de error (401, 403, 500, etc.) ya que esas páginas a menudo todavía están afectadas por problemas de inyección, en cuyos casos aún hay preocupación de buscadores rastreando páginas fuera de su tipo de contenido verídico. En límite 'alto' este escáner no alertará sobre las respuestas de error al cliente o servidor.

Solución:

Se debe asegurar que el servidor de la aplicación/web establezca el encabezado Content-Type apropiadamente, y que esté establecido el encabezado X-Content-Type-Options en 'nosniff' para todas las páginas web. Si es posible, se tiene que asegurar que el último usuario usa un navegador web compatible con los estándares y moderno que no ejecute MIME-sniffing en absoluto, o que pueda ser dirigida por el servidor de la aplicación/web para no ejecutar MIME-sniffing.

- Protección de buscador de web XSS no disponible

Url: **https://www.sismedicalec.com/**

Riesgo: Bajo

Confianza: Media Parámetro: X-XSS-Protección

Origen: Pasivo (10016 - Protección de buscador de web XSS no disponible)

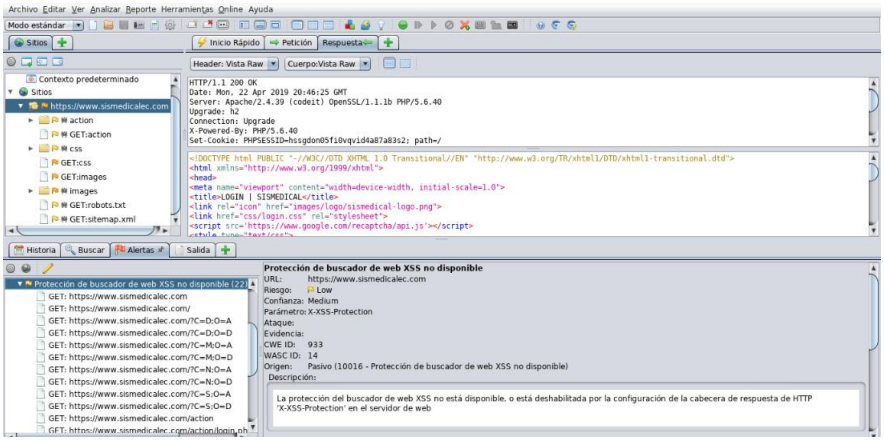


Figura 4.32: Interfaz que muestra la alerta Protección de buscador de web XSS no disponible

Como se muestra en la **figura 4.32:** Muestra la última alerta del nivel de riesgo bajo, denominada "Protección de buscador de web XSS no disponible", la url, el nivel de confianza, el parámetro, origen , también la descripción de la alerta detectada con la solución propuesta por la herramienta.

Descripción de la alerta

La protección del buscador de web XSS no está disponible, o está deshabilitada por la configuración de la cabecera de respuesta de HTTP 'X-XSS-Protección' en el servidor de web

Información

El encabezado de respuesta HTTP X-XSS-Protection le permite al servidor web habilitar o deshabilitar el mecanismo de protección del navegador web XSS. Los siguientes valores intentan habilitarlo: X-XSS-Protection: 1; mode=bloqueo X-XSS-Protection:1; reporte=http://www.example.com/xss Los siguientes valores lo deshabilitarían: X-XSS-Protection: 0 El encabezado de respuesta HTTP X- XSS-Protection es actualmente compatible en Internet Explorer, Chrome y Safari (WebKit). Tenga en cuenta que esta alerta solo se produce si el cuerpo de respuesta podría potencialmente contener una carga útil XSS (con un tipo de contenido basado en texto, con una longitud distinta de cero).

Solución

Se debe asegurar que el filtro XSS del navegador web está habilitado, estableciendo el encabezado de respuesta HTTP X-XSS-Protection en ’1’.
Progreso del escaneo, al terminar en analisis del sistema web **“SISMEDICA-LEC”.**

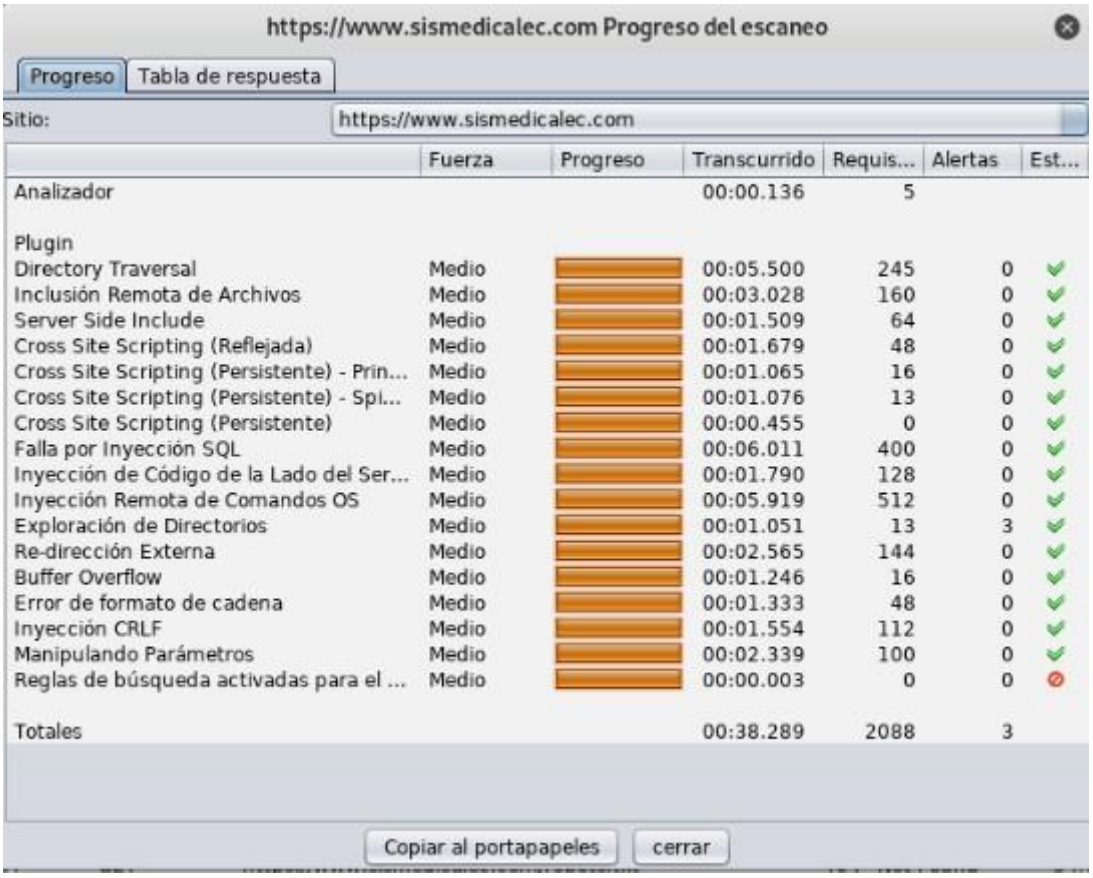


Figura 4.33: Interfaz que muestra el progreso final del escaneo de vulnerabilidades al sistema web.

Según la **figura 4.33:** Muestra una ventana luego de presentar y describir cada vulnerabilidad web de acuerdo a su grado de riesgo, la herramienta OWASP ZAP, tambien proporciona una Ventana adicional donde muestra el progreso final del escaneo, con el objetivo cumplido, de acuerdo a su analisis y descripción.

4.3.3. Herramienta web Uniscan

Es un escáner de vulnerabilidades Web, dirigido a la seguridad informática, cuyo objetivo es la búsqueda de vulnerabilidades en los sistemas web. Está licenciado bajo GNU GENERAL PUBLIC LICENSE 3.0 (GPL 3).
Uniscan se ha desarrollado utilizando el lenguaje de programación Perl, destaca por su facilidad para trabajar con el texto, usar expresiones regulares, además de ser multihilo.

Sus características principales son:

- Identificación de las páginas del sistema a través de un rastreador web.
- El uso de threads en el rastreador.
- Controla el número máximo de peticiones.
- Control de la variación de las páginas identificadas por el sistema rastreador web.
- Control de las extensiones de archivo que se ignoran.
- Prueba de páginas encontradas a través del método GET.
- Prueba de los formularios que se encuentran a través del método POST.
- Soporte de peticiones SSL (HTTPS).
- Soporte de proxy.
- Interfaz gráfica simple.
- Añadido nuevo plugin "PHP inyección argumento CGI" para las pruebas dinámicas.
- Búsqueda para los plugins de Drupal, Joomla y WordPress [26].

Arquitectura de Uniscan

Un escáner de vulnerabilidades tiene módulos esenciales, como rastreador y comprobador de vulnerabilidades. Estos componentes son responsables de actividades vitales. El rastreador intenta encontrar todos los archivos y vínculos dentro del sitio objetivo. Por otro lado, el verificador de vulnerabilidades es responsable de realizar diferentes tipos de pruebas sobre cada archivo o vínculo

encontrado. Además, el probador puede realizar un gran conjunto de pruebas, siendo que no todos son conocidos o pueden realizarse en el momento de diseñar el escáner de vulnerabilidades. Bajo esta óptica, una forma de diseñar un escáner de vulnerabilidades flexibles y extensibles es a través de la modularización y el uso de plug-ins [27].

Para la versión de terminal: Primero, abra el terminal y escriba Uniscan y presione enter. Se mostrarán todos los parámetros disponibles que puede utilizar. Puede iniciar el archivo de ayuda usando "uniscan -h" y se detallan todas las opciones de esta herramienta.

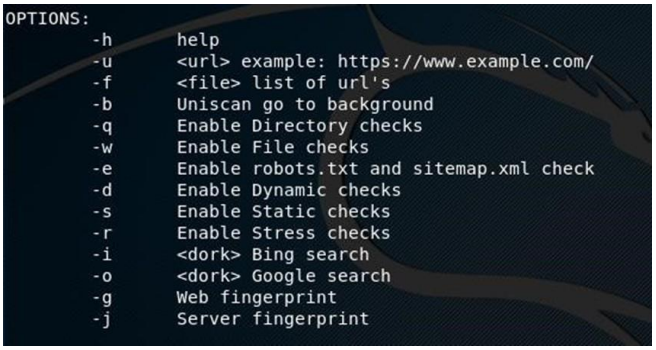


Figura 4.34: Pantalla del listado de opciones que ofrece UNISCAN

Como se observa en la **figura 4.34:** Presenta el listado de tareas que puede ejecutar la herramienta Uniscan ya sea una por una o diferentes de ellas o a su vez todas en un análisis.

Listado de vulnerabilidades que detecta la herramienta Uniscan

- consultar directorio
- verificar archivos
- revisar robots.txt
- pruebas dinámicas
- pruebas estáticas
- pruebas de estrés
- huella digital
- huella digital del servidor

Cómo funciona Uniscan-GUI.

Para realizar un análisis con Uniscan es tan sencillo como seleccionar las opciones y elegir el objetivo. Para posteriormente estar disponible un informe con los resultados obtenidos al pulsar en la opción Open log file, que muestra un informe del análisis al sistema web. Uniscan tiene línea de comandos, así como interfaz GUI.
La versión GUI se puede cargar usando el siguiente comando.

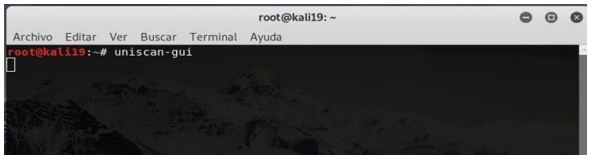


Figura 4.35: Comando para ver la Interfaz de Uniscan

Como se muestra en la **figura 4.35**: Indica mediante el uso del terminal desde el sistema operativo Kali Linux, que al escribir uniscan-gui inmediatamente empieza a cargar la interfaz de la herramienta Uniscan.

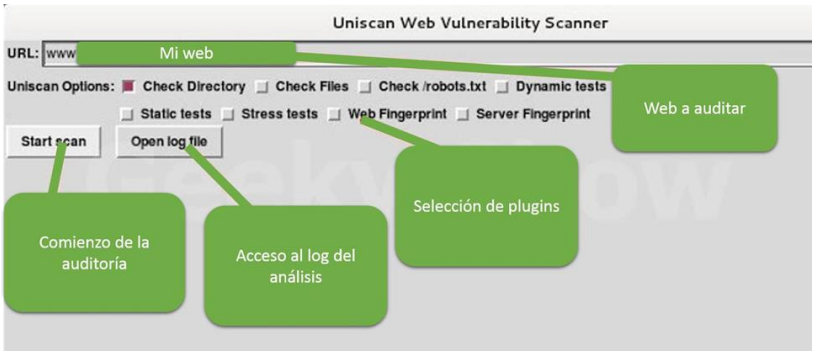


Figura 4.36: Listado de opciones de pruebas mediante el escáner Uniscan con interfaz.

En la siguiente **figura 4.36:** Muestra la versión GUI de la herramienta Uniscan, detallando cada tarea que puede ejecutar la misma, al momento de realizar el análisis del objetivo, una aplicación web.

Escaneo del sistema web sismedicalec

Herramienta usada (UNISCAN)

en modo consola comando:
uniscan -u http://www.sismedicalec.com / -qwedsgj

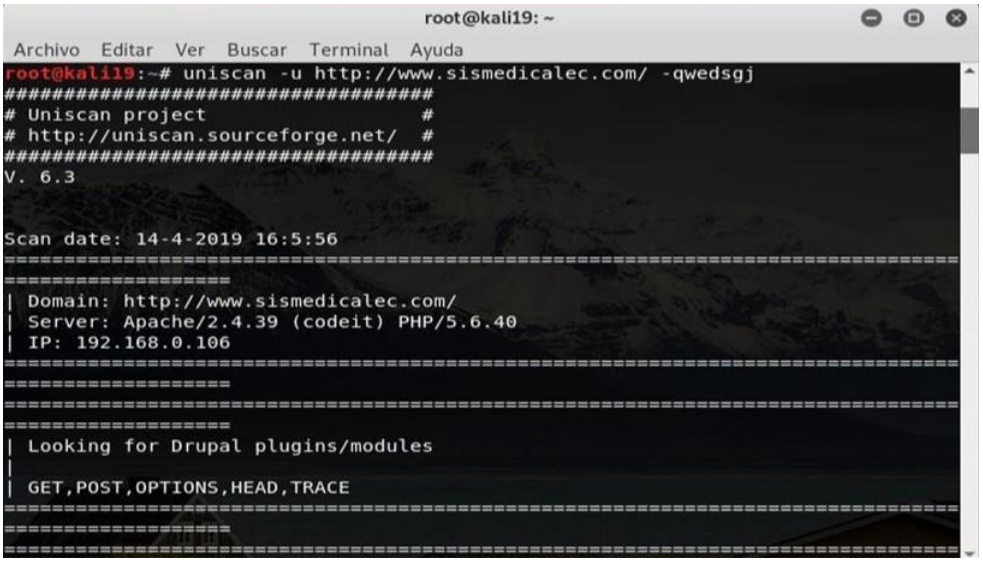


Figura 4.37: Interfaz con el comando para ejecutar el análisis de vulnerabilidades web.

Según la **figura** 4.37: Indica una pantalla donde se escribe la línea de comando, uniscan -u http://www.sismedicalec.com/ acompañado de **qwedsgj,** el mismo que ordena a la herramienta que ejecute en su análisis 7 diferentes opciones de tareas.

Al digitar este comando ejecutará 7 tareas diferentes como:

- q Activa la comprobación de directorios.
- w Activa la comprobación de archivos.
- e Analiza los ficheros robots.txt y sitemap.xml.
- d Habilita las comprobaciones dinámicas.
- s Activa las comprobaciones estáticas.
- g Realiza un seguimiento de huella digital a la web.
- j Realiza un seguimiento de huella digital al servidor.

Vulnerabilidades detectadas al momento de analizar

Vulnerabilidad Check Directory: - Consulta de directorios .
Esta herramienta permite rastrear los directorios ubicados en el destino clickeando la opción Check Directory desde la interfaz de Uniscan.

Directory check:
CODE: 200 URL: http://www.sismedicalec.com/action/
CODE: 200 URL: http://www.sismedicalec.com/ajax/
CODE: 200 URL: http://www.sismedicalec.com/css/
CODE: 200 URL: http://www.sismedicalec.com/doc/
CODE: 200 URL: http://www.sismedicalec.com/font/
CODE: 200 URL: http://www.sismedicalec.com/fotos/
CODE: 200 URL: http://www.sismedicalec.com/icons/
CODE: 200 URL: http://www.sismedicalec.com/images/
CODE: 200 URL: http://www.sismedicalec.com/js/
CODE: 200 URL: http://www.sismedicalec.com/lib/
CODE: 200 URL: http://www.sismedicalec.com/makefont/
CODE: 200 URL: http://www.sismedicalec.com/phpmyadmin/

Figura 4.38: Muestra la, opción de análisis de la prueba Directory Check

En la presente **figura 4.38:** Indica la opción Directory Check o Verificación de directorios, que el escáner Uniscan descubrió con algunos directorios que podrían ser de interés para los atacantes, incluyendo lo que parece ser documentación e información de configuración del sistema web escrito en PHP .

Vulnerabilidad, File Check:

El test indicado también permite habilitar las verificaciones de archivos con la opción File Check.

File check:
CODE: 200 URL: http://www.sismedicalec.com/config.php
CODE: 200 URL: http://www.sismedicalec.com/css
CODE: 200 URL: http://www.sismedicalec.com/error_log
CODE: 200 URL: http://www.sismedicalec.com/index.php
CODE: 200 URL: http://www.sismedicalec.com/js
CODE: 200 URL: http://www.sismedicalec.com/login.php

Figura 4.39: Resultado de la prueba con la presente herramienta, opción Files Check

Según la **figura 4.39:** Muestra el resultado de File Check o chequeo de archivos, donde se puede observar que en el análisis se encontró algunos archivos que podrían proporcionar información valiosa al atacante acerca del aplicativo web, lo cual se convierte en un fallo de seguridad, que podría poner en evidencia información confidencial del sistema web.

Vulnerabilidad, Web Backdoors, E-mails y Php info () Disclosure:

Uniscan cargará algunos complementos para realizar verificaciones dinámicas en el destino, incluida la identificación del correo electrónico, la detección de puerta

trasera y el descubrimiento de SQL y otros tipos de puntos de inyección. Para ello se debe seleccionar la opción Dynamic Test. Esto puede demorar bastante tiempo en ejecutarse y mostrar los resultados.

Web Backdoors:

Ignored Files:
http://www.sismedicalec.com/css/datatables.net-buttons-bs/Readme.md
http://www.sismedicalec.com/js/Chart.js/docs/04-Bar-Chart.md
http://www.sismedicalec.com/js/datatables.net-keytable/Readme.md
http://www.sismedicalec.com/js/Chart.js/docs/06-Polar-Area-Chart.md
http://www.sismedicalec.com/js/Chart.js/LICENSE.md
http://www.sismedicalec.com/js/Chart.js/docs/10-Notes.md
http://www.sismedicalec.com/js/datatables.net/Readme.md
http://www.sismedicalec.com/css/datatables.net-fixedheader-bs/Readme.md
http://www.sismedicalec.com/css/bootstrap/CHANGELOG.md
http://www.sismedicalec.com/js/datatables.net-scroller/Readme.md
http://www.sismedicalec.com/js/Chart.js/docs/08-Bubble-Chart.md
http://www.sismedicalec.com/js/Chart.js/docs/00-Getting-Started.md
http://www.sismedicalec.com/css/bootstrap-progressbar/CHANGELOG.md
http://www.sismedicalec.com/js/datatables.net-buttons/Readme.md
http://www.sismedicalec.com/js/Chart.js/docs/09-Advanced.md
http://www.sismedicalec.com/js/Chart.js/docs/03-Line-Chart.md
http://www.sismedicalec.com/js/Chart.js/docs/07-Pie-Doughnut-Chart.md
http://www.sismedicalec.com/js/Chart.js/docs/05-Radar-Chart.md
http://www.sismedicalec.com/js/datatables.net-fixedheader/Readme.md
http://www.sismedicalec.com/js/moment/CHANGELOG.md
http://www.sismedicalec.com/js/Chart.js/CONTRIBUTING.md
http://www.sismedicalec.com/css/datatables.net-bs/Readme.md
http://www.sismedicalec.com/js/Chart.js/docs/01-Chart-Configuration.md
http://www.sismedicalec.com/css/datatables.net-responsive-bs/Readme.md
http://www.sismedicalec.com/css/datatables.net-scroller-bs/Readme.md
http://www.sismedicalec.com/js/datatables.net-responsive/Readme.md
http://www.sismedicalec.com/js/Chart.js/docs/02-Scales.md
http://www.sismedicalec.com/css/bootstrap-progressbar/CONTRIBUTORS.md

Figura 4.40: Interfaz sobre Web Backdoors, opción seleccionada Dynamic Tests.

Como se muestra **figura 4.40:** Muestra las Web Backdoors encontradas por Uniscan, una puerta trasera o backdoor es una vulnerabilidad que permite entrar en un servidor, página web, red local o empresarial sin ser detectado y con ciertos privilegios (o no, depende) para poder hacer casi lo que quieras. La inmensa mayoría de las puertas traseras son errores genuinos, es decir, o bien se generan por un error del usuario o del programador, o de los administradores de red. Esto es peligroso porque el usuario desconoce que existe esa puerta abierta sin ninguna protección adecuada.

El escáner detecto más de 300 correos electrónicos guardados en el sitio web.

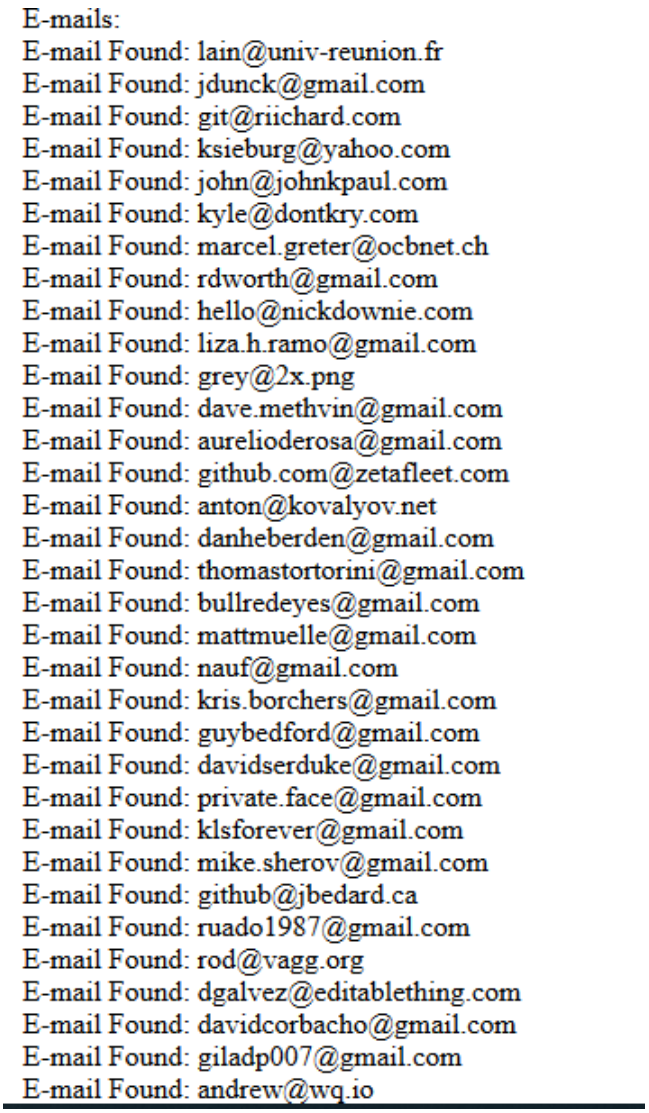


Figura 4.41: Interfaz que indica como son publicados los E-mails de la página Web, un fallo de seguridad.

En esta **figura 4.41:** se puede comprobar que existe una lista de direcciones de correos electrónicos, esta información puede ser utilizado por un hacker para hacer conjeturas de posibles formatos de E-mails internos que, utilizada la empresa, también se puede saber los posibles nombres de usuarios. Hay que asegurar que los correos electrónicos que están publicados no ofrecen ninguna información que pueda ser utilizado por un hacker, ya que es un evidente fallo de seguridad.

External hosts:

Un host de entrada externa es una ubicación fuera del servidor de contraseñas que almacena información de contraseñas. Una vez que se configura un Host de Entrada Externa en el Servidor de Contraseña, el Servidor de Contraseña puede conectarse a él para importar y actualizar información. Para algunas aplicaciones,

la lógica empresarial personalizada puede necesitar conectarse a un host externo. En este caso, el nombre de host debe estar registrado en la configuración de backend y está sujeto a un proceso de aprobación.


External hosts:
http://www.fpdf.org
http://server
https://icf.improvely.com
https://en.wikipedia.org
http://github.com
http://www.ozerov.de
http://bugs.mysql.com
http://www.acko.net
https://www.google.com
http://www.dangrossman.info
https://oss.maxcdn.com
https://gist.github.com
http://travis-ci.org
http://sf.net
http://host
https://maxcdn.bootstrapcdn.com
http://www.tcpdf.org
http://www.incomsis.com
https://saucelabs.com
http://www.google.com
http://www.awio.com
https://tools.ietf.org
https://demo.phpmyadmin.net
https://www.w3counter.com
http://wiki.phpmyadmin.net
https://github.com
http://bugs.php.net
https://disqus.com
http://www.gnu.org
http://mariadb.org
http://en.wikipedia.org
http://pecl.php.net
http://httpd.apache.org
http://www.mysql.com

Figura 4.42: Primera parte de la lista de Host de entrada externa

Según la **figura 4.42:** Indica las la primera parte de las Url encontradas en la opción “External Host” o Host Externos, presentada por el escáner Uniscan.

https://www.phpmyadmin.net
http://www.drizzle.org
http://sphinx-doc.org
http://dev.mysql.com
http://bugzilla.mozilla.org
https://launchpad.net
http://ghbtns.com
http://software.opensuse.org
https://sourceforge.net
https://www.improvely.com
http://www.wikipedia.org
https://wiki.phpmyadmin.net
https://hosted.weblate.org
http://www.apachefriends.org
http://pear.php.net
https://code.jquery.com
http://netdna.bootstrapcdn.com
http://cdnjs.cloudflare.com
http://php.net
http://www.hardened-php.net
http://www.php.net
http://fedoraproject.org

Figura 4.43: Segunda parte de la lista de Host de entrada externa

Como se muestra en la **figura 4.43:** Indica las la segunda parte de las Url encontradas en la opción "External Host" presentada por el escáner Uniscan.

Descripción:

El archivo host externo enumera las máquinas remotas conectadas directamente al MSS (Tamaño Máximo de Segmento) para las cuales no se usa DNS. Para ver la lista de hosts externos actualmente definidos en el sistema, vaya a la página Administrar hosts externos , que enumera la dirección IP, el nombre de host y el alias (si está definido) de cada máquina remota directamente conectada al MSS.

Revelación de información, Php info ()

PHPinfo() Disclosure:

Source Code Disclosure:
Source Code Found: http://www.sismedicalec.com/phpmyadmin/doc/html/_sources/faq.txt
Source Code Found: http://www.sismedicalec.com/js/pdfmake/build/pdfmake.min.js.map
Source Code Found: http://www.sismedicalec.com/js/morris.js/spec/viz/run.sh
Source Code Found: http://www.sismedicalec.com/phpmyadmin/doc/html/_sources/setup.txt

Figura 4.44: Revelación de información mediante Php info (), opción Dynamic Tests.

Según la **figura 4.44:** Muestra la alerta Phpinfo(), y la parte de Revelación de

código fuente con sus respectivas URL, del sistema web analizado.

Descripción

Php info () es una funcionalidad de depuración que imprime información detallada tanto del sistema como de la configuración de PHP.

Al encontrar un fallo en la seguridad, como es Php info (), un atacante puede obtener información como, por ejemplo:

- Versión exacta de PHP.
- Nombre del sistema operativo en uso y su versión.
- Detalles de la configuración de PHP.
- Direcciones IP internas.
- Variables de entorno del servidor.
- Extensiones PHP cargadas y sus configuraciones.

Esta información puede ayudar a un atacante a obtener más información sobre el sistema web. Después de obtener información detallada, el atacante puede investigar las vulnerabilidades conocidas para ese sistema bajo revisión. El atacante también puede usar esta información durante la explotación de otras vulnerabilidades. En esta imagen sobre Php info(), indica un archivo morris.js que es una librería Javascript que trabaja con jQuery y que permite de forma muy sencilla crear gráficas de barra. Información FCKeditor correos electrónicos.

4.4. Identificación de las principales amenazas encontradas al realizar el análisis de las vulnerabilidades del sistema de gestión médica (SISMEDICALEC).

Listado de vulnerabilidades de alto riesgo alto

Vulnerabilidades	Riesgo	Herramienta de análisis	Herramienta de ataque o método	Forma de uso
Session Cookie Without HttpOnly Flag	Alto	Vega	Envenenamiento ARP dentro de una LAN	Robo de cookies, se debe aplicar un ataque MITM. Consiste en introducirse en la comunicación entre dos equipos para que todo el tráftco pase por nosotros y poder así desencriptar sus datos, contraseñas, etc. Para este tipo de ataques se necesitan dos máquinas víctima, que bien podría ser el servidor y un equipo de una red empresarial, o bien el router y el equipo de nuestra víctima real, además de nuestro propio equipo.
Session Cookie Without Secure Flag	Alto	Vega	Envenenamiento ARP dentro de una LAN	Robo de cookies aplicando el ataque MITM que consiste en introducirse en la comunicación entre dos equipos para que todo el tráftco pase por nosotros y poder así desencriptar sus datos, contraseñas, etc. Para este tipo de ataques se necesitan dos máquinas víctima, que bien podría ser el servidor y un equipo de una red empresarial, o bien el router y el equipo de nuestra víctima real, además de nuestro propio equipo.

Tabla 4.3: Vulnerabilidades de riesgo alto analizado con Vega Scan

Listado de vulnerabilidades de alto riesgo medio.

Vulnerabilidades	Riesgo	Herramienta de análisis	Herramienta de ataque o método	Forma de uso
Client Ciphersuite Preference	Medio	Vega	Inyección de código malicioso	Ataque Poodle, para este ataque, se supone que el atacante puede hacer peticiones desde el navegador al servidor, pero debe estar conectado en la misma red de la víctima, por ejemplo con un código en Javascript. De la siguiente forma GET ruta Cookie: ???? \r\n\r\n cuerpo de la petición MAC padding.
HTTP Trace Support Detected	Medio	Vega	Inyección de código malicioso	Ataque XSS, Un atacante puede usar la vulnerabilidad de los scripts entre sitios para que el navegador del usuario de destino emita una solicitud TRACE al servidor a través de XMLHttpRequest (o una función similar) y luego recupere la cookie de la respuesta, que contendrá la solicitud que fue enviada por navegador, incluidas las cookies.
Local Filesystem Paths Found	Medio	Vega	Inyección de código malicioso	Sí permite modiftcar los parámetros de lo que se incluye: GET pasa los datos por URL, por lo que en la barra de direcciones URL se vería algo así: http://localhost/index.php?pagina= 78se3.php, donde se incluye el archivo 78se3.ph. La manipulación de este archivo puede modiftcar lo que se incluye mostrar, por ejemplo, el ftchero de contraseñas de Linux /etc/hosts.

Tabla 4.4: Vulnerabilidades de riesgo medio analizado con Vega Scan

Segunda parte de el listado de vulnerabilidades con riesgo medio.

Vulnerabilidades	Riesgo	Herramienta de análisis	Herramienta de ataque o método	Forma de uso
Possible Source Code Disclosure	Medio	Vega	Inyección SQL, Metasploit	Crear un shell web que debe colocarse en una parte no restringida del sistema de archivos del servidor, que es fácil de acceder. Puede usar el siguiente shell web PHP simple que le permite ejecutar un comando del sistema pasándolo a través de un parámetro GET (por ejemplo ?cmd=whoami): <?phpe-cho(system($_GET["cmd"])); ?> Ejecutar la siguiente sentencia: SELECT * FROM stats WHERE ftlename = '' UNION SELECT '<?php system($_GET["cmd"]); ?>', '', '' INTO OUTFILE '/tmp/cmd.php';#
Encabezado X-Frame-Options no establecido	Medio	Owasp Zap	Inyección de código malicioso	Ataque XSS, Un atacante puede usar la vulnerabilidad de los scripts entre sitios para que el navegador del usuario de destino emita una solicitud TRACE al servidor a través de XMLHttpRequest (o una función similar) y luego recupere la cookie de la respuesta, que contendrá la solicitud que fue enviada por navegador, incluidas las cookies.
Exploración de Directorios	Medio	Owasp Zap	DirBuster	Se puede realizar un ataque de fuerza bruta, que deja a los directorios y archivos ocultos, sin nada que ocultar. Si seleccionamos el diccionario de fuerza bruta que viene por defecto, veremos cómo tal vez encontramos ciertos directorios interesantes. Se coloca la URL, con el puerto 80 y clic en start.
Shell webs	Medio	Uniscan	Remot3d	Herramienta creada para generar una puerta trasera para controlar y explotar un servidor donde el servidor ejecuta el programa en PHP.

Tabla 4.5: Vulnerabilidades de riesgo medio analizado con Vega Scan, Owasp Zap y Uniscan

Listado de vulnerabilidades de alto riesgo bajo

Vulnerabilidades	Riesgo	Herramienta de análisis	Herramienta de ataque o método	Forma de uso
Password Field with Autocomplete Enabled	Bajo	Vega	Inyección de código malicioso	Las credenciales almacenadas pueden ser capturadas por un atacante que obtiene el control sobre la computadora del usuario. Además, un atacante que encuentra una vulnerabilidad de aplicación separada, como los scripts entre sitios, puede explotar esto para recuperar las credenciales almacenadas en el navegador de un usuario.
Inclusión de archivos de origen JavaScript Cross-Domain	Bajo	Owasp Zap	Inyección de código malicioso	Se aplica un ataque MITM, Hombre en el medio. Dicho ataque permite a una tercera persona inyectar en páginas web visitadas por el usuario, código JavaScript en una URL del sitio web de forma camuflada, desde una misma LAN.
No Cache-control y sistema de encabezado HTTP Pragma	Bajo	Owasp Zap	Falla en la conftguración en el entorno web.	No se encontró forma de llevar a cabo un ataque informático.
No se encuentra encabezado X-Content-Type-Options Header		Owasp Zap	Inyección de código malicioso	Respuestas json puede ser explotado mediante la invalidación de la Matriz de constructores o si hostiles de los valores de cadena de javascript-se escapó. preftjo de todos json con algo como:throw 1; < don't be evil' > <script src="http://yourbank.com/accountStatus.json">
Protección de buscador de web XSS no disponible	Bajo	Owasp Zap	Inyección de código malicioso	Para la inyección, solo debe remplazar el valor del texto que se mostrará con una secuencia de comandos de modo que aparezca en la página web. Siempre y cuando el navegador del usuario esté conftgurado para ejecutar dichas secuencias de comandos, el código malicioso tendrá acceso a todos los datos compartidos por la página web y el servidor del usuario (cookies, campos de entrada, etc).

Tabla 4.6: Vulnerabilidades de riesgo bajo analizado con Vega Scan

Como se observa los **tablas 4.3, 4.4 y 4.5 y 4.6** se enlistan las vulnerabilidades

web detectadas en el aplicativo mencionado, mediante tres escáneres diferentes como son **VEGA, OWASP y UNISCAN**, cada una de estas herramientas arrojó un número determinado posibles debilidades detectadas en el sistema, cada una de ellas con su respectiva descripción, posible solución y el nivel de riesgo e impacto que tendría en el sistema web en el caso que personas ajenas a la empresa INCOMSIS, realicen ataques de explotación de las mismas.

Al usar las 3 herramientas usadas se repetían las vulnerabilidades informáticas que muchas veces solo cambiaba algo en el nombre pero en si eran las mismas. Se realizó una búsqueda extensa en diferentes sitios especializados en seguridad y la en la mayoría de casos se reporta como forma de ataque el uso de inyección de código malicioso. Debido al nivel de acceso a la información estas son las 14 vulnerabilidades en total.

4.5. Explotación de las principales vulnerabilidades de Sistema de Gestión Médica

4.5.1. Vulnerabilidad detectada denominada "Session Cookie Without HttpOnly Flag", ataque a realizar para su explotación Man in the Middle (Hombre en el medio)

Un ataque "Hombre en el Medio", representa en la actualidad una seria amenaza para la seguridad de la información en redes de comunicación, estos constituyen uno de los ataques sofisticados a sistemas de telecomunicaciones. Ocurren cuando un usuario agresor se inserta en la comunicación entre dos entidades y esta es controlada sin que ninguna de ellas conozca que el enlace ha sido violado. Al ser implementado y alcanzar su objetivo permite la realización de ataques en tiempo real como escuchas en el medio, las que tienen como propósito capturar paquetes transmitidos en una red, y leer el contenido de datos en busca de cualquier tipo de información, y secuestro de sesión. Durante dicho proceso el atacante puede modificar los paquetes, gestionar e incluso silenciosamente disminuir el tamaño de los mismos [28].

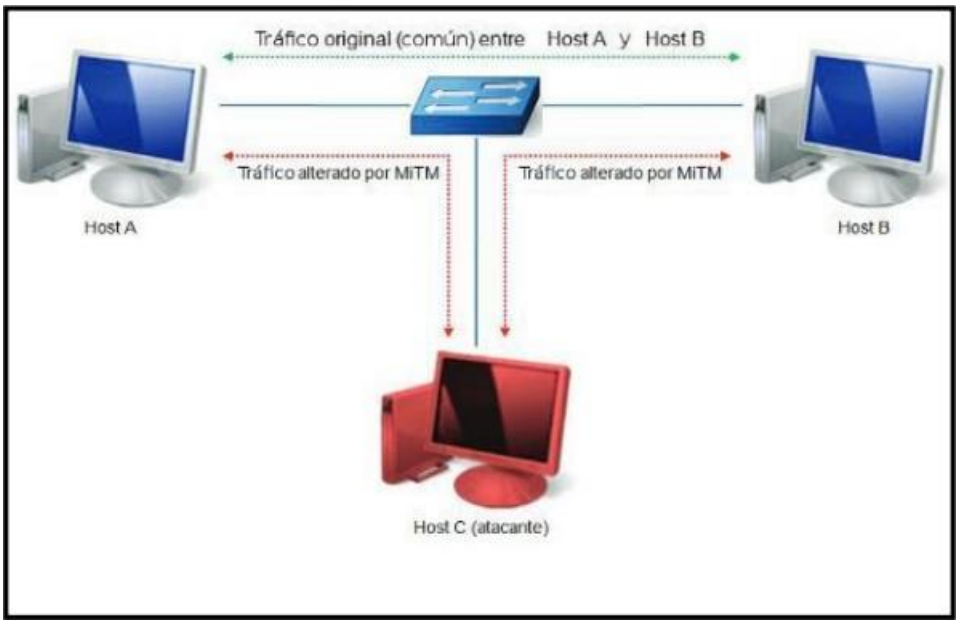


Figura 4.45: Ataque “Hombre en el medio”, MITM

Ataques MITM en redes LAN

Una red de área local (LAN) permite a usuarios comunicarse con otros host de la misma red. Está definida por dos o más dispositivos tales como estaciones de trabajo, impresoras o servidores. Estos dispositivos están vinculados con el propósito de compartir información, recursos o ambos [29].

Un ataque MITM en dicha infraestructura puede ejecutarse implementando ataques como los que se exponen a continuación.

Ataque Envenenamiento ARP

El protocolo ARP (Address Resolution Protocol) es el encargado de traducir direcciones IP a direcciones MAC (Media Access Control), cuando un host desea comunicarse con una IP emite una trama ARP–Request a la dirección de Broadcast solicitando la MAC del host destino, primando la IP. El ordenador con la IP solicitada responde con un ARP–Reply indicando su MAC. Los switchs y los host guardan una tabla local con la relación IP-MAC llamada “tabla ARP”. Esta es sensible a cambios y se actualiza con frecuencia; dicha tabla puede ser modificada por un ordenador que emita tramas ARP–Reply indicando su MAC como destino válido para una IP específica. Además no existen mecanismos para verificar si el par de direcciones IP y MAC obtenidas son las correctas [29].

El ataque ARP Spoofing (figura 1.4) hace referencia a la construcción de tramas de solicitud y respuesta ARP modificadas, asociando la dirección MAC del atacante con la dirección IP de otro nodo (el nodo atacado), como por ejemplo la puerta de enlace predeterminada (Gateway). Cualquier tráfico dirigido a la

dirección IP de ese nodo será enviado al atacante en lugar de su destino real, el agresor en este instante puede elegir entre reenviar el tráfico a la puerta de enlace predeterminada real (ataque pasivo o escucha), o modificar los datos antes de reenviarlos (ataque activo),este puede ser ejecutado desde una máquina controlada (el atacante ha conseguido previamente hacerse con el control de la misma), o bien la máquina del atacante está conectada directamente a la LAN Ethernet [30].

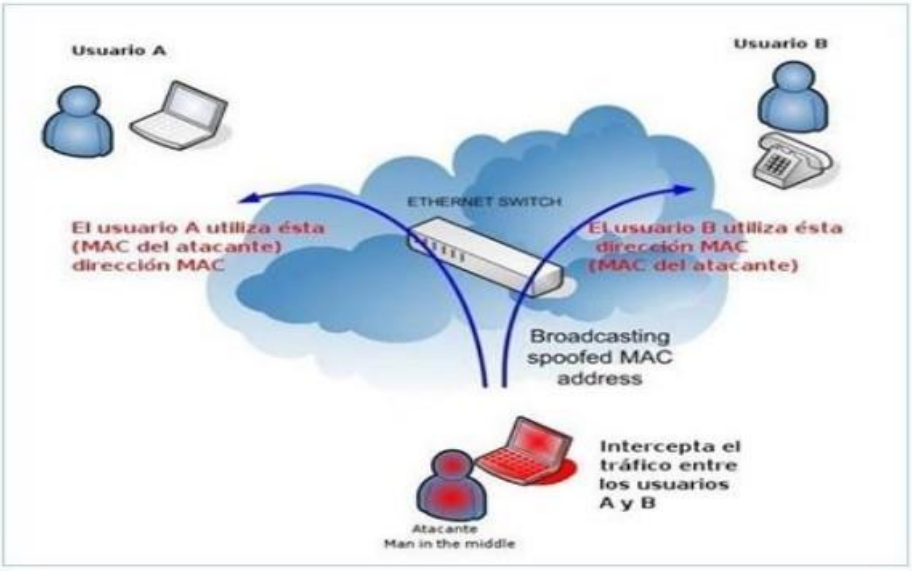


Figura 4.46: Ataque “Hombre en el Medio” variante envenenamiento ARP

Para realizar un ataque hombre en el medio en este caso se utilizó una herramienta sniffer muy popular como es sslstrip.
En el anexo C, se puede observar paso a paso como se hizo para realizar el ataque informático y conseguir los resultados esperados.

```
nterfaz: 192.168.0.102 --- 0x3
  Dirección de Internet          Dirección física      Tipo
  192.168.0.1           60-36-dd-49-3a-59     dinámico
  192.168.0.101         08-00-27-1f-57-9f     dinámico
  192.168.0.109         60-36-dd-49-3a-59     dinámico
  192.168.0.255         ff-ff-ff-ff-ff-ff     estático
  224.0.0.2             01-00-5e-00-00-02     estático
  224.0.0.22            01-00-5e-00-00-16     estático
  224.0.0.252           01-00-5e-00-00-fc     estático
  239.255.255.250       01-00-5e-7f-ff-fa     estático
  255.255.255.255       ff-ff-ff-ff-ff-ff     estático
```

Figura 4.47: Tabla ARP, desde una máquina con Windows

La figura 4.47 muestra la información de las ip y la duplicación de las direcciones MAC, en la ip 192.168.0.1 y la ip 192.168.0.109, con lo que se puede comprobar que existe un “hombre en el medio”, el que está escuchando el tráfico de su víctima.

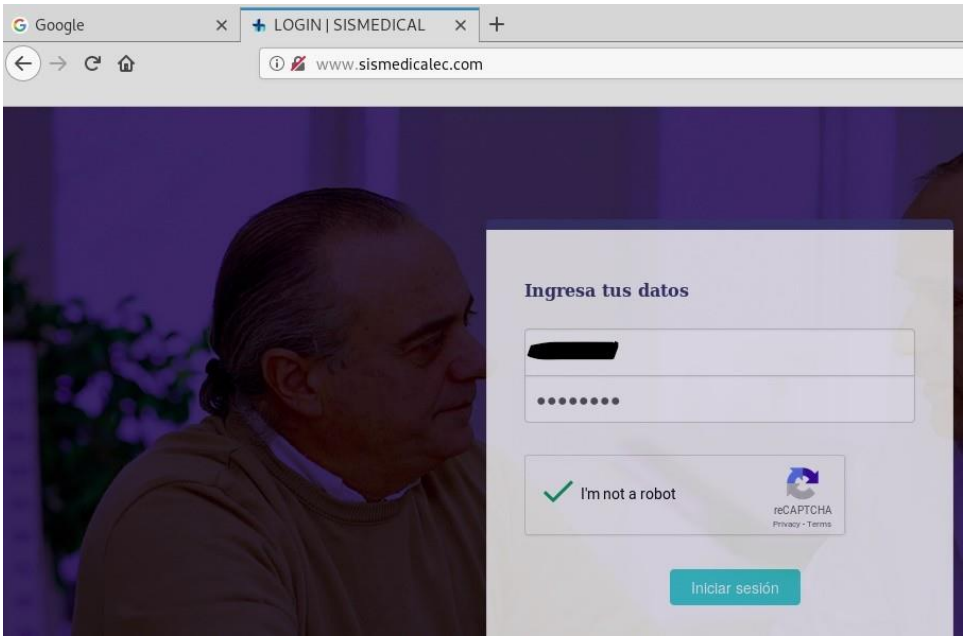


Figura 4.48: Ingreso de credenciales del sistema web desde el navegador del atacante

Como se observa en la figura 4.48, desde un navegador google Chrome el atacante ingresa las credenciales de logueo para ingresar a la aplicación web deseada.
La ejecución de la "Escucha de tráfico SSL por medio de ataque mitm", se realizó con éxito y dicho ataque se puede encontrar en una forma más detallada en el **Anexo C.**

4.5.2. Vulnerabilidad detectada denominada "Possible Source Code Disclosure" y la explotación se realizará mediante Ataques de Inyección de Código SQL

"Structured Query Language" (Lenguaje de Consulta Estructurado), es un lenguaje textual utilizado para interactuar con bases de datos relacionales. La unidad típica de ejecución de SQL es la consulta ("query"), conjunto de instrucciones que permiten modificar la estructura de la base de datos (mediante instrucciones del tipo "Data Definition Language", DDL) o manipular el contenido de la base de datos (mediante instrucciones del tipo "Data Manipulation Language", MDL). En los servidores Web se utiliza este lenguaje para acceder a bases de datos y ofrecer páginas dinámicas o nuevas funcionalidades a sus usuarios.
El ataque por inyección de código SQL se produce cuando no se filtra de forma adecuada la información enviada por el usuario. Un usuario malicioso podría incluir y ejecutar textos que representen nuevas sentencias SQL que el servidor no debería aceptar. Este tipo de ataque es independiente del sistema de bases de

datos subyacente, ya que depende únicamente de una inadecuada validación de los datos de entrada. Como consecuencia de estos ataques y, dependiendo de los privilegios del usuario de base de datos bajo el cual se ejecutan las consultas, se podría acceder no sólo a las tablas relacionadas con la operación de la aplicación del servidor Web, sino también a las tablas de otras bases de datos alojadas en el mismo servidor Web. También pueden propiciar la ejecución de comandos arbitrarios del sistema operativo del equipo del servidor Web [31].

Como protegerse:

- No utilice sentencias SQL construidas dinámicamente.
- No utilice cuentas con privilegios administrativos.
- No proporcione mayor información de la necesaria.
- Verifique el tamaño como el tipo de datos de las entradas del usuario [32].

Según la herramienta Vega, detectó la vulnerabilidad denominada "Possible Source Code Disclosure" con nivel de riesgo medio, y al recopilar información respecto a esta, se encontró que puede ser explotada esta debilidad mediante un ataque de inyección de código sql, por ende se llevó a cabo las pruebas pertinentes para verificar si es o no explotable esta vulnerabilidad, en el ANEXO D, se puede observar paso a paso como se intentó vulnerar el sistema web mediante inyección de código malicioso y se muestra los resultados obtenidos del mismo.

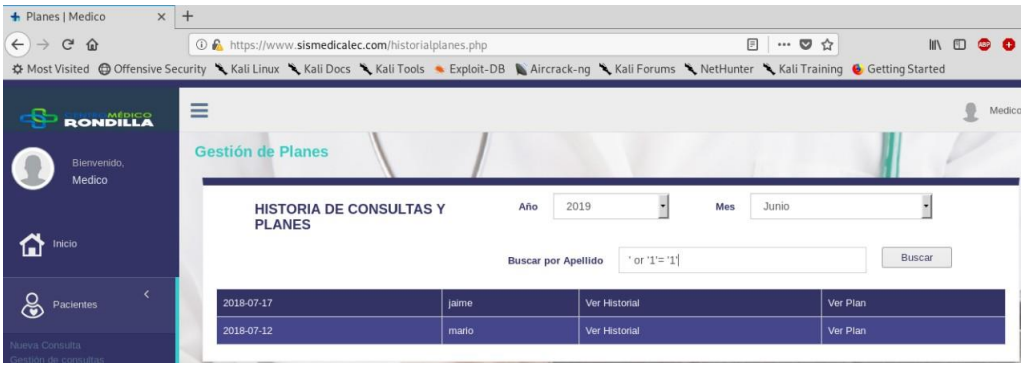


Figura 4.49: Ingreso de código en una caja de texto para identificar si acepta código sql

Según la figura 4.47 Ingreso de caracteres especiales en la caja de texto de búsquedas para comprobar la posible explotación de la vulnerabilidad mediante un ataque de inyección de código sql.

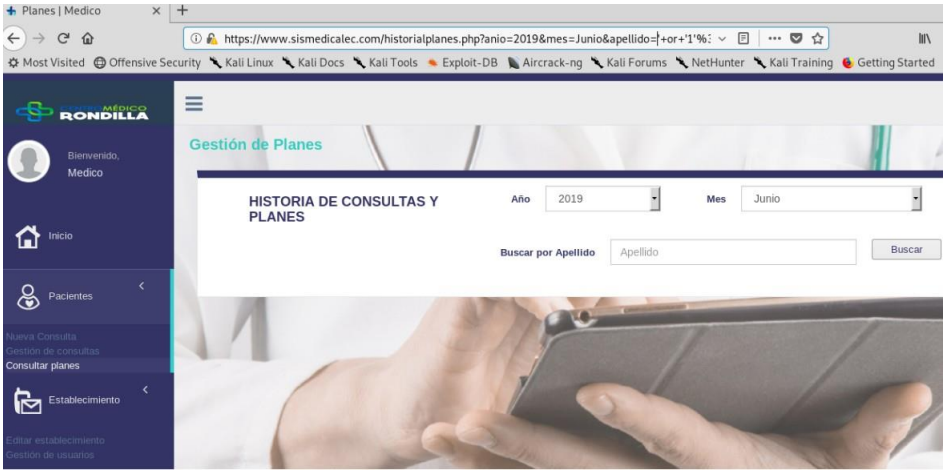


Figura 4.50: Resultado al probar inyectar código sql en la caja de texto, para búsquedas por apellido.

Según la figura 4.48 El aplicativo web sismedicalec.com pasó las pruebas satisfactoriamente al ignorar el ingreso de código sql, con caracteres especiales, por lo cuál se comprueba que la vulnerabilidad "Possible Source Code Disclosure" no puede ser explotada mediante inyección de código malicioso.

4.5.3. Vulnerabilidad detectada denominada "Exploración de Directorios" y la explotación se realizará con la herramienta DirBuster

Muchas instituciones publican archivos confidenciales con información sensible en la Internet, y por medio de alguna herramienta se puede vulnerar dicha información, de interés personal.
La herramienta Owasp Zap detectó la siguiente vulnerabilidad, **"Exploración de directorios"**, para lo cual se investigó sobre diferentes herramientas para su explotación y para este caso se escogió DirBuster.

Herramienta DirBuster:

Es una herramienta que fuerza a los directorios de servidores web. para encontrar aplicaciones ocultas o configuraciones inseguras, que proporcionar otro vector de ataque. DirBuster es muy simple de detectar, porque usa el mismo patrón URI cada vez y genera muchas respuestas "404 Página no encontrada". Debido a la relativamente baja complejidad de la firma de DirBuster, la mayoría de los métodos de clasificación de aprendizaje automático generarán resultados similares. Nuestros resultados sugieren, las mejoras de el mecanismo de votación de clasificación es muy limitado, ya que los diferentes algoritmos de clasificación

pueden generar similares. modelos sin embargo, nuestros experimentos mostraron que la votación de clasificación es al menos tan buena como la mejor de los algoritmos utilizados, al aprovechar las ventajas de varias clasificaciones. resultados combinados en un modelo [33].

Explicación del uso de DirBuster:

Para la explotación de la vulnerabilidad "Exploración de directorios" detectada con la herramienta Owasp Zap, se usó la herramienta DirBuster que se recomienda usar según **la tabla 4.5**, la misma que realizó con éxito la explotación de dicha vulnerabilidad como se puede observar en la figura 4.51.

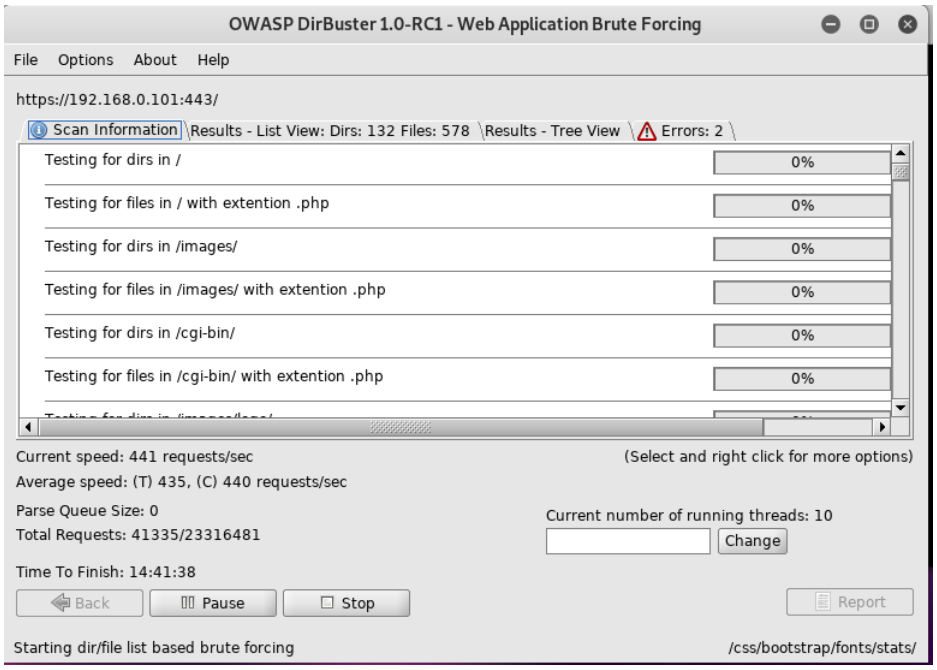


Figura 4.51: Herramienta Dirbuster en su pantalla con la opción Scan Information

La figura 4.51 muestra la pantalla con el inicio del escáner de cada uno de los directorios que están adjuntos al aplicativo web sismedicalec.

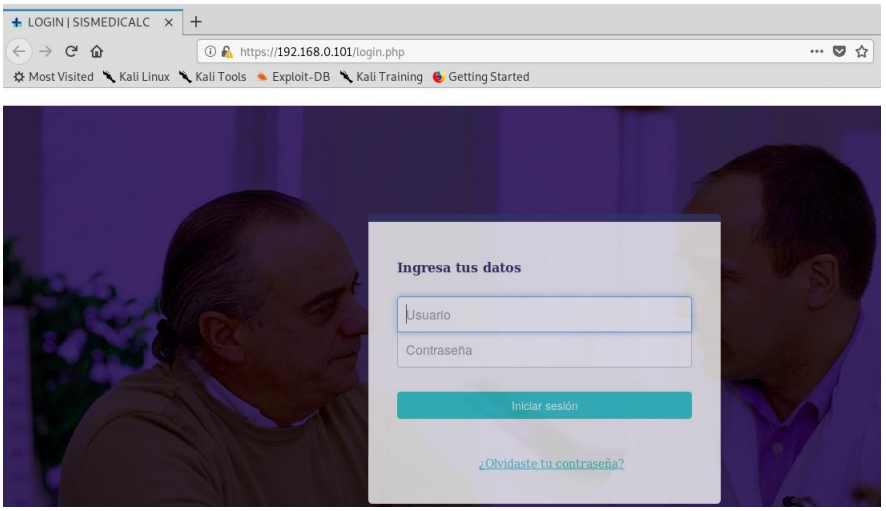


Figura 4.52: Pantalla de un directorio listado, que muestra la interfaz de la página login.php

Una demostración de como se realizó el ataque a la debilidad de "Exploración de directorios", en el ANEXO E, se puede observar paso a paso como se realizó con éxito dicho ataque informático. Lo cual pone en riesgo la aplicación web en su contenido pues sus directorios como se muestra han sido listados y expuestos dentro de un navegador web.

Pruebas de stress

Las pruebas de carga, también conocidas en algunos casos como pruebas de stress o test load tiene por objeto emular la conexión a un aplicativo web de determinado número de usuarios con el fin de medir la reacción de este y del sistema donde está instalado cuando la concurrencia alcanza niveles específicos [34].
Uno de los análisis que suelen integrar cualquier plan de QA(aseguramiento de calidad) es la prueba de stress. Esta evaluación pone a prueba la robustez y la confiabilidad del software sometiéndolo a condiciones de uso extremas. Entre estas condiciones se incluyen el envío excesivo de peticiones y la ejecución en condiciones de hardware limitadas. El objetivo es saturar el programa hasta un punto de quiebre donde aparezcan bugs (defectos) potencialmente peligrosos para el aplicativo web que se esté analizando.

Herramienta Apache Bench

Herramienta de línea de comandos disponible en los Unix y Gnu/Linux. Software que está diseñado para medir el rendimiento de los sitios web. Permite hacer peticiones de forma concurrente hacer pruebas de carga a cualquier sitio web[35].

Para la realización de las pruebas de stress según se puede observar detalladamente en el Anexo F, fue necesario crear un nuevo usuario con privilegios limitados, ya que no era recomendado hacer dichas pruebas desde un super usuario.
Para ello se ejecutó el siguiente comando.
useradd -m -d /home/test -s /bin/bash -g sudo pruebas

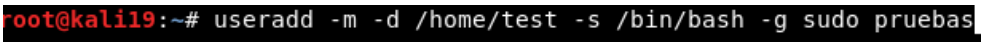


Figura 4.53: Comando para crear un nuevo usuario con privilegios inferiores a root

Lo que este comando logra:

- useradd - crea un nuevo usuario
- -m - crea el directorio de inicio
- -d / home / test: establece el directorio de inicio del usuario en / home / pruebas
- -s / bin / bash - hace que el shell de bash predeterminado del usuario (Ubuntu usa el guión de forma predeterminada)
- -g sudo - agregar usuario al grupo sudo (para ejecutar comandos con sudo)
- prueba - el nombre del nuevo usuario

Ejecución de comandos para la realización de la prueba de stress, a modo de ejemplo, como se observa en la figura 4.52

```
pruebas@kali19:/root$ ab -n 1000 -c 100 https://www.sismedicalec.com/
This is ApacheBench, Version 2.3 <$Revision: 1843412 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking www.sismedicalec.com (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        Apache/2.4.39
Server Hostname:        www.sismedicalec.com
Server Port:            443
SSL/TLS Protocol:       TLSv1.2,ECDHE-RSA-AES256-GCM-SHA384,2048,256
Server Temp Key:        X25519 253 bits
TLS Server Name:        www.sismedicalec.com

Document Path:          /
Document Length:        6561 bytes

Concurrency Level:      100
Time taken for tests:   4.648 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      6999000 bytes
HTML transferred:       6561000 bytes
Requests per second:    215.14 [#/sec] (mean)
Time per request:       464.819 [ms] (mean)
```

Figura 4.54: Ingreso de comando para iniciar las respectivas solicitudes a la aplicación web.

Como se muestra en la figura **4.54** se ejecutó el comando con 1000 peticiones que tuvieron un inicio y un final, con un nivel de concurrencia 100 y un total de 4.648 segundos para que la actividad sea ejecutada con éxito.

Se puede observar en el **Anexo F**, de forma detallada la realización de la prueba de stress aplicada al sistema “Sismedicalec”.

CAPÍTULO 5

Conclusiones y Recomendaciones

5.1. Conclusiones

- La empresa INCOMSIS no ha realizado en ninguna ocasión una auditoría de seguridad informática, por lo que el presente proyecto ha sido de vital importancia para lograr mitigar riesgos en su software.
- El análisis de vulnerabilidades informáticas es una acción necesaria para toda empresa que desarrolle y comercialice sistemas informáticos. Para ello la empresa INCOMSIS, debe tener como prioridad el poder garantizar la aplicación de normas de seguridad en todos sus dominios y aplicativos web.
- La protección a las Cookies de sesión de un usuario es ineficaz lo que provoca un alto riesgo de secuestro de sesiones.
- Actualmente no poseen un mecanismo de cierre automático al ingresar un usuario credenciales erróneas, lo que puede provocar riesgos y permitir ataques de fuerza bruta.
- Los objetivos planteados en el presente proyecto investigativo fueron cumplidos con éxito mediante el análisis, explotación y descripción de las posibles soluciones para cada una de vulnerabilidades informáticas encontradas.
- Posterior a la detección de las vulnerabilidades existentes en la aplicación web, se realizó un informe dirigido al director y propietario de la organización INCOMSIS.

5.2. Recomendaciones

- Se recomienda a la empresa realizar frecuentemente un análisis de seguridad informática para evitar riesgos en sus sistemas web y así garantizar el

correcto funcionamiento del sitio web a los usuarios, de la misma forma poder garantizar la seguridad de envío y recepción de los datos personales de los usuarios de la aplicación web.

- Se sugiere el análisis de técnicas de aseguramiento de la información y la calidad de la misma, para brindar seguridad y satisfacción a los usuarios.
- Se recomienda de forma puntual el tomar en cuenta las posibles soluciones brindadas para cada una de las vulnerabilidades detectadas y así lograr aminorar los riesgos en lo máximo posible.

Bibliografia

[1] S. C. Romaniz, "Seguridad de aplicaciones web: vulnerabilidades en los controles de acceso," in *XIV Congreso Argentino de Ciencias de la Computación,* 2008.

[2] S. D. Diaz, "Pruebas de seguridad en aplicaciones web como imperativo en la calidad de desarrollo del software,"

[3] M. Pilay, L. Alexander, M. González, and C. Leonardo, "Modelo de negocio para la creación de una empresa de seguridad en la información digital para pymes.," B.S. thesis, Universidad Estatal de Guayaquil, 2017.

[4] G. V. Villacís and R. A. R. Morocho, "Vulnerabilidades y amenazas a los servicios web de la intranet de la universidad técnica de babahoyo," *3c Tecnología,* vol. 6, no. 1, pp. 53–66, 2017.

[5] M. E. Hurtado Sandoval, M. Mendaño, and L. Alcides, "Implementación de técnicas de hacking ético para el descubrimiento y evaluación de vulnerabilidades de la red de una cartera de estado," B.S. thesis, Quito, 2016., 2016.

[6] C. A. Avila Maldonado, A. F. Fula Amaya, N. I. Pérez, C. J. Rodríguez Rodríguez, *et al.*, "Módulo de educación presencial del sistema de información sobre ciberseguridad y plataforma educativa," B.S. thesis, Universidad Piloto de Colombia, 2013.

[7] R. Gómez, D. H. Pérez, Y. Donoso, and A. Herrera, "Metodología y gobierno de la gestión de riesgos de tecnologías de la información," *Revista de ingeniería,* no. 31, pp. 109–118, 2010.

[8] P. Soxo and P. Xavier, "Propuesta de técnicas de aseguramiento de aplicaciones web desarrolladas en java," Master's thesis, Escuela Superior Politécnica de Chimborazo, 2013.

[9] E. Bernardis, H. Bernardis, M. Berón, and G. A. Montejano, "Seguridad en servicios web," in *XIX Workshop de Investigadores en Ciencias de la Computación (WICC 2017, ITBA, Buenos Aires),* 2017.

[10] S. C. Solá, *Fundamentos de sistemas operativos: teoría y ejercicios resueltos*. Editorial Paraninfo, 2007.

[11] E. A. Quezada Caballero, "Pruebas de Penetración contra Aplicaciones Web," *Reydes*, 2014.

[12] O. Foundation, "Guía de pruebas OWASP v3.," vol. 0, p. 372, 2008.

[13] D. Roldan, P. Valderas, and O. Pastor, "Aplicaciones web: Un enfoque práctico," *México: Alfaomega Ra-Ma*, 2010.

[14] R. M. G. Labrador, "14127 administración de servidores linux (ubuntu/fedora/centos)," 2014.

[15] M. A. Arias, *Introducción a PHP*. IT Campus Academy, 2013.

[16] C. E. G. Montoya, C. A. C. Uribe, and L. E. S. Rodríguez, "Seguridad en la configuración del servidor web apache," *Inge Cuc*, vol. 9, no. 2, pp. 31–38, 2013.

[17] L. A. C. Santillán, M. G. Ginestà, and Ó. P. Mora, "Bases de datos en mysql," *Universitat Oberta de Catalunya*, 2014.

[18] J. N. Castillo Fiallos, "Propuesta de implementación de un modelo para la reducción de riesgos de seguridad informática en servicios web de la espoch," Master's thesis, Escuela Superior Politécnica de Chimborazo, 2016.

[19] R. A. MATUZ and M. G. D. J. R. JORDÁN, "Análisis de la seguridad dentro del desarrollo web e implementación de testing tesis,"

[20] J. N. Castillo, A. S. C. Barahona, P. M. M. Naranjo, and D. F. J. Segovia, "Modelo para la reducción de riesgos de seguridad informática en servicios web," *Cumbres*, vol. 4, no. 2, pp. 19–30, 2018.

[21] S. Ortega Sancho, "Diseño de un sistema de seguridad para un servidor web apache," 2018.

[22] E. N. Yánez Romero, "Guía de buenas prácticas de desarrollo de aplicaciones web seguras aplicado al sistema control de nuevos aspirantes empresa grupo laar," B.S. thesis, 2014.

[23] J. W. Amado Ballén, A. Calderón, C. Andrés, A. A. Sierra Morales, *et al.*, "Análisis de vulnerabilidades en aplicaciones web desarrolladas en php

versión 5.6. 24 con base de datos mysql versión 5.0. 11 a partir de ataques sql inyección," 2017.

[24] D. L. L. Chávez and P. A. Ramirez, "Pruebas de seguridad utilizando herramientas para la detección de vulnerabilidades. security tests using tools for the detection of vulnerabilities.,"

[25] E. González de Canales González, "Generación de reportes de vulnerabilidades y amenazas para aplicaciones web,"

[26] T. Llorente Cabello *et al.*, "Laboratorio virtual para el estudio de vulnerabilidades en la nube," 2016.

[27] D. P. Rocha, "Uniscan: um scanner de vulnerabilidades para sistemas web," 2012.

[28] V. Ramachandran and C. Buchanan, *Kali Linux wireless penetration testing: beginner's guide*. Packt Publishing Ltd, 2015.

[29] V. d. R. Ochoa Villalba, "Análisis de tráfico de datos en la capa de enlace de una red lan, para la detección de posibles ataques o intrusiones sobre tecnologías ethernet y wifi 802.11," B.S. thesis, SANGOLQUÍ/ESPE/2011, 2011.

[30] L. d. l. C. Esquerra Blanco, *Pruebas de penetración con la herramienta Kali Linux en la Universidad Central Marta Abreu de las Villas*. PhD thesis, Universidad Central"Marta Abreu"de Las Villas, 2014.

[31] Á. G. Vieites, "Tipos de ataques e intrusos en las re- des informáticas," *línea]. Disponible en: http://www. edisa. com/wp-content/uploads/2014/08/Ponencia_-_Tipos_de_ataques_y_de_intrusos_en_las_redes_informaticas. pdf.[Accedido: 09-ago-2017]*, 2013.

[32] L. J. Cañon Parada *et al.*, "Ataques informáticos, ethical hacking y conciencia de seguridad informática en niños," B.S. thesis, Universidad Piloto de Colombia, 2015.

[33] P. Fruehwirt, S. Schrittwieser, and E. Weippl, "Using machine learning techniques for traffic classification and preliminary surveying of an attackers profile," in *Proc. of Int. Conf. on Privacy, Security, Risk and Trust*, 2014.

[34] C. D. J. C. VELÁSQUEZ, “Propuesta metodológica para la realización de pruebas de de software en un ambientes productivos,” *Universidad Nacional De Colombia, Medellín*, 2009.

[35] H. Sansano Miralles, “Pruebas sobre sitios web,” 2017.

Anexos y Apéndices

Anexo A

Anexo

INCOMSIS

Quito D.M., 6 de junio de 2018

Ingeniera

Elsa Pilar Urrutia Mg.

DECANA

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Universidad Técnica de Ambato

Presente.

Señora Decana:

En mi calidad de Director y propietario de la organización IMCOMSIS, por medio de la presente manifiesto a usted, que la Srta. Lisbeth Marixxi Quirola Valarezo, portador (a) de la cédula de ciudanía N° 2101108427, estudiante de Décimo nivel en el período académico marzo-agosto 2018, de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, tiene mi autorización y respaldo para desarrollar en la empresa que represento, el Trabajo de Titulación denominado **"Análisis de vulnerabilidades de seguridad informática, del Sistema de Gestión Médica SISMEDICALEC, de la empresa INCOMSIS"**, para lo cual designo como Tutor Empresarial al Ing. Luis Alberto Pazmiño Msc., quien es consultor de Seguridad Informática de nuestra organización.

Con estos antecedentes informo que la realización de este Trabajo de Titulación es de gran importancia para la empresa, el estudiante tiene total apoyo para su desarrollo y ejecución, por lo expuesto anteriormente solicito se apruebe y se proceda con el trámite correspondiente.

Atentamente,

INCOMSIS
Adrian Eduardo Figueroa Jaro
R.U.C. 0603230013001

Ing. Adrian J. Figueroa

Director

16:00

Matriz Quito: Mariano Aguilera E7-219 y Diego de Almagro. Edificio Giovannini.
Teléfono: 02 2549200 - 09 95982411

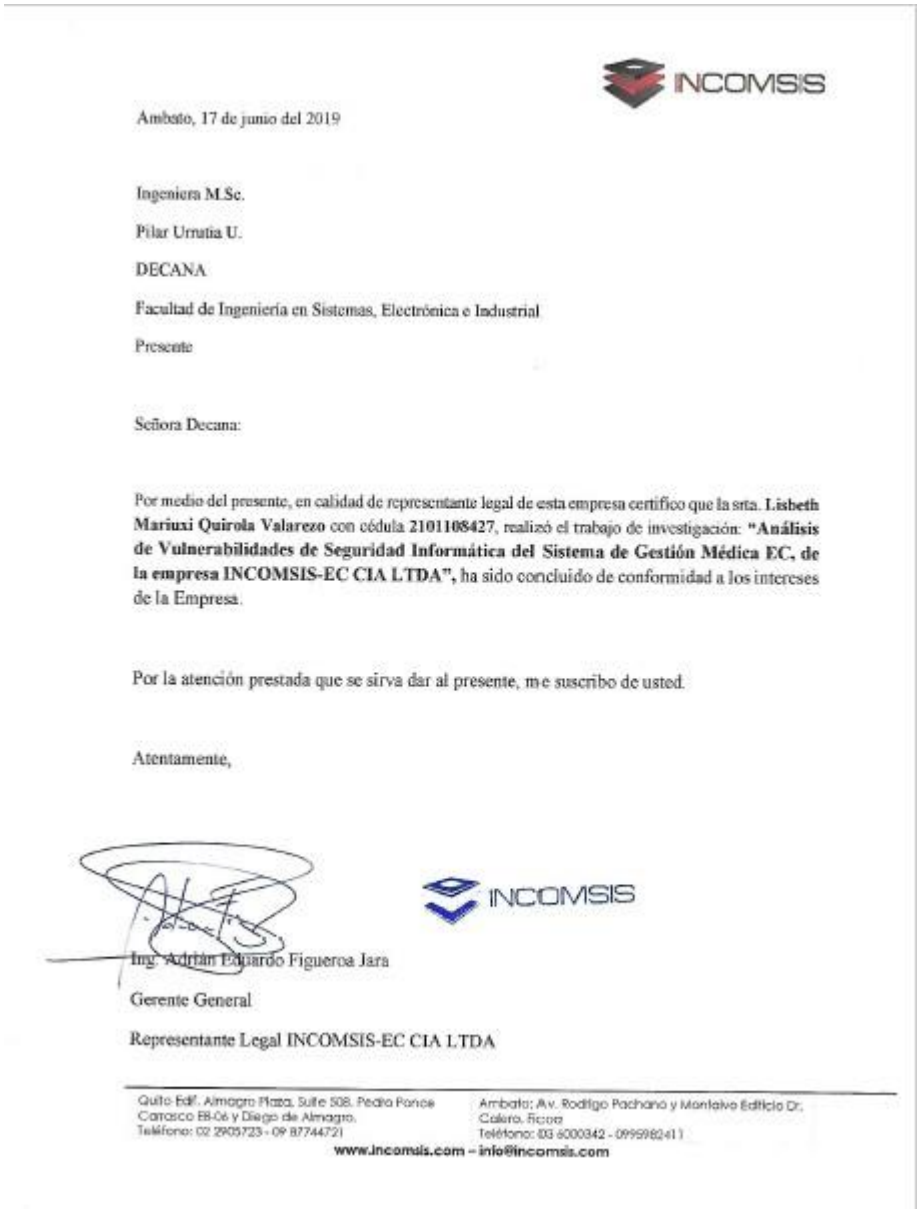
Sucursal Ambato: Los Tilos y Los Nuecos.
Ficoa
Teléfono: 03 6000347 - 098774721

www.incomsis.com – info@incomsis.com

Figura A.1: Aprobación por la empresa INCOMSIS para realizar el proyecto de investigación

Anexo B

Anexo



INCOMSIS

Ambato, 17 de junio del 2019

Ingeniera M.Sc.

Pilar Urrutia U.

DECANA

Facultad de Ingeniería en Sistemas, Electrónica e Industrial

Presente

Señora Decana:

Por medio del presente, en calidad de representante legal de esta empresa certifico que la srta. **Lisbeth Mariuxi Quirola Valarezo** con cédula **2101108427**, realizó el trabajo de investigación: **"Análisis de Vulnerabilidades de Seguridad Informática del Sistema de Gestión Médica EC, de la empresa INCOMSIS-EC CIA LTDA"**, ha sido concluido de conformidad a los intereses de la Empresa.

Por la atención prestada que se sirva dar al presente, me suscribo de usted.

Atentamente,

INCOMSIS

Ing. Adrián Eduardo Figueroa Jara

Gerente General

Representante Legal INCOMSIS-EC CIA LTDA

Quito Edif. Almagro Plaza, Suite 508. Pedro Ponce Carrasco E8-06 y Diego de Almagro. Teléfono: 02 2905723 - 09 8774472

Ambato: Av. Rodrigo Pachano y Montalvo Edificio Dr. Calero. Ricaurte Teléfono: 03 6000342 - 0999982411

www.incomsis.com – info@incomsis.com

Figura B.1: Certificado de culminación del proyecto investigativo para la empresa INCOMSIS

Anexo C

Anexo

Prueba de explotación de vulnerabilidades mediante el ataqu**e Man in the middle**, MITM

Para realizar esta práctica se usa “sslstrip” como herramienta principal y estar en la misma red local la víctima y el atacante.

Como primer paso se debe conocer las ip de las tres máquinas necesarias para la realización exitosa del ataque mitm

Que consta de la ip del servidor, la ip del atacante y la ip de la víctima, cabe destacar que estas máquinas deben estar dentro de una LAN.

En la figura C.1 es la ya mencionada ip del servidor que aloja el sistema web analizado.

Figura C.1: Ip del servidor que contiene la aplicación web

Figura C.4: Comando para la habilitación de reenvío de paquetes

Figura C.2: Ip de la máquina del atacante cibernético

Como se figura C.2 es la ya mencionada ip del atacante cibernético, que para este caso es el “hombre en el medio”.

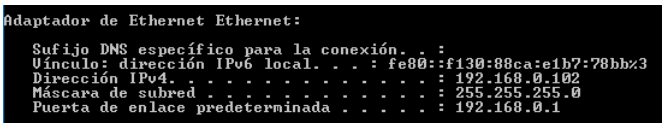


Figura C.3: Ip de la víctima, el objetivo de ataque

Esta máquina y su ip corresponden a un sistema operativo Windows, como se puede observar en la figura **C.3**

Según muestra la figura C.4, se digita el comando presente, desde la máquina atacante con ip 192.168.0.109 para hacer un reenvío de paquetes mediante ip versión 4.

Figura C.5: Línea de comando para la creación de una regla ip-tables

Como se puede ver en la figura C.5 se escribe el presente dígito para la creación de una regla dentro del ip-tables, que es desde el puerto 80 redirigido al puerto 8080.

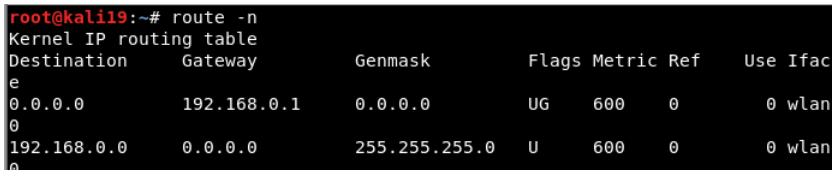


Figura C.6: Puerta de enlace de la dirección ip del atacante

En la figura C.6 se digitó el presente comando para conocer el Gateway de la red del atacante.

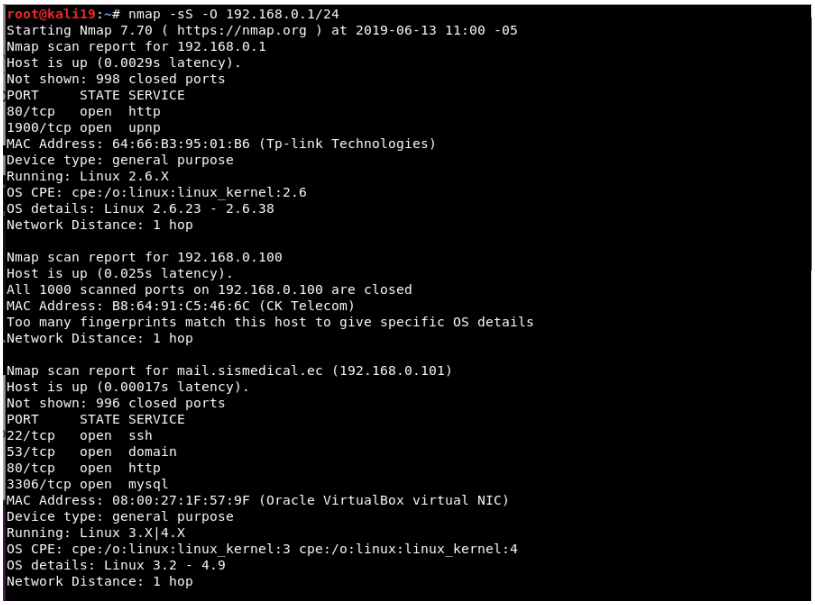


Figura C.7: Escaneo de direcciones ip dentro de la LAN.

En la figura C.7 muestra el comando para verificar las ip que están dentro de la LAN, con ayuda de la herramienta **NMAP.**

Figura C.8: Muestra la ip de la víctima, mediante el escaneo de ips

Segunda parte de el listado de ip y puertos dentro de la LAN, por medio de un NMAP.

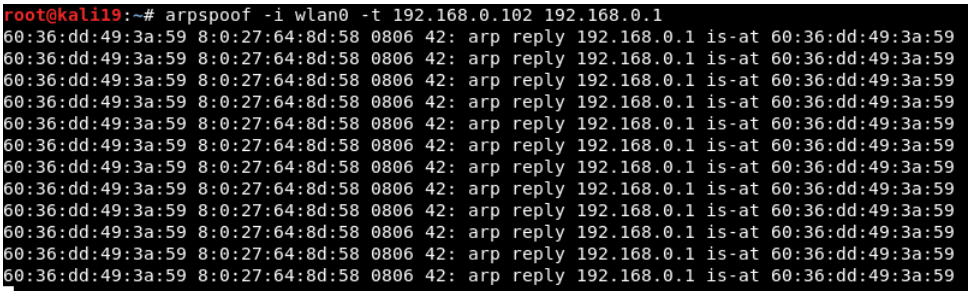


Figura C.9: Ingreso de comando para activación del ataque hombre en el medio

Según se observa en la figura C.9 es un comando para envenenamiento de ip, que consta de interfaz de red del atacante, wlan0, -t por target o el objetivo, 192.168.0.102 es la máquina víctima, mientras que 192.168.0.1 representa el Gateway del atacante, el que va a escuchar el tráfico de la red.

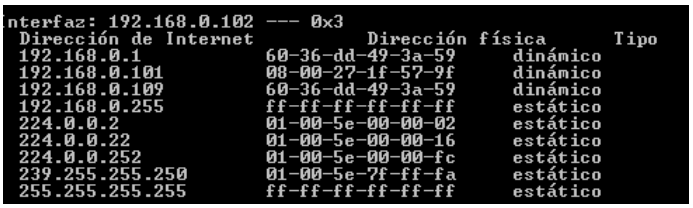


Figura C.10: Tabla ARP desde consola en Windows

Luego de haber empezado el ataque mitm, como se mostró en la figura C.9, la figura C.10 muestra una consulta de las ip desde la máquina de la víctima lo cual permite verificar y asegurar que el ataque está siendo exitoso, pues como se observa la ip 192.168.0.1 y la ip 192.168.0.109 tienen la misma dirección MAC.

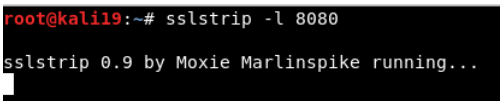


Figura C.11: Inicio de la herramienta para re-direccionamiento de tráfico de la red

La herramienta sslstrip viene por defecto instalada en Kali Linux, y se procede a iniciarla se observa en la figura C.11, se usa el puerto 8080 que es el que está abierto y tiene como objetivo escuchar el tráfico SSL de la víctima.

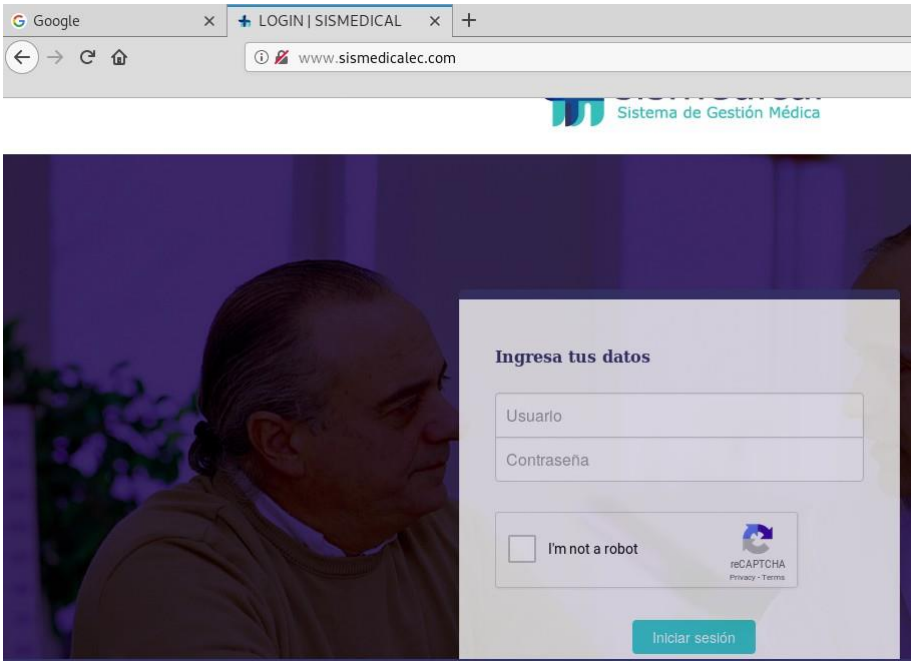


Figura C.12: Páginas de navegación de ingreso de la víctima

Como ya inició la escucha del tráfico SSL de la víctima y el objetivo de este ataque es obtener las credenciales de acceso al sistema web, se espera hasta que el objetivo abra su navegador e ingrese al aplicativo deseado en este caso es **sismedicalec.com,** como se puede observar en la figura C.12

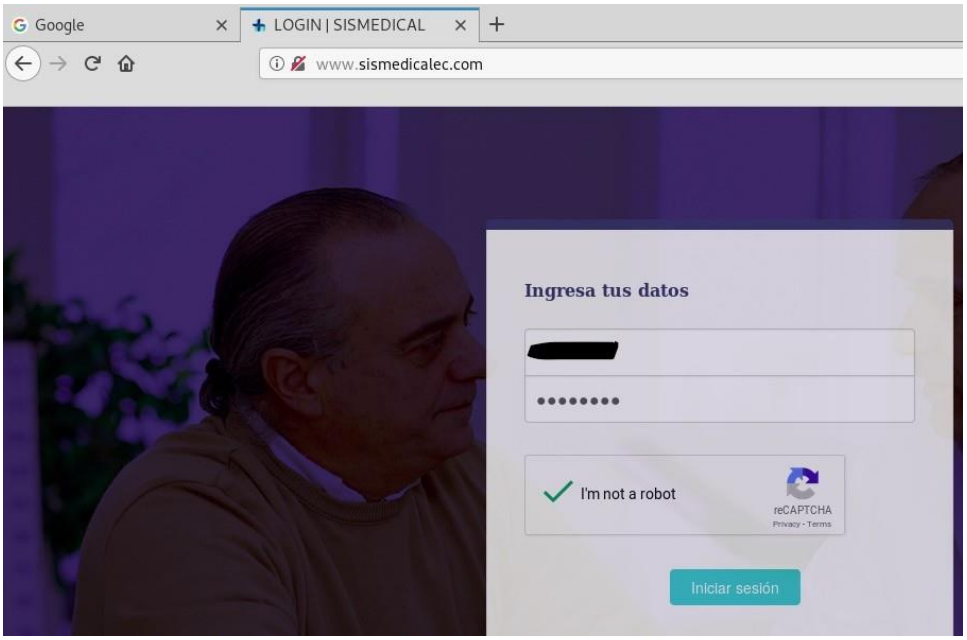


Figura C.13: Ingreso de credenciales a la aplicación web por parte de la víctima

Como la víctima desconoce que en su red hay un “hombre en el medio” procede a loguearse dentro del aplicativo ingresando su nombre de usuario y contraseña.

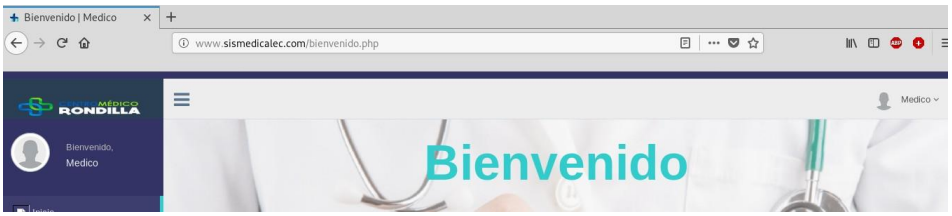


Figura C.14: Ingreso exitoso al aplicativo que realizó la víctima

Como el usuario de la aplicación ingresó sus credenciales entonces se muestra una pantalla de Bienvenida que presenta el aplicativo en su inicio.

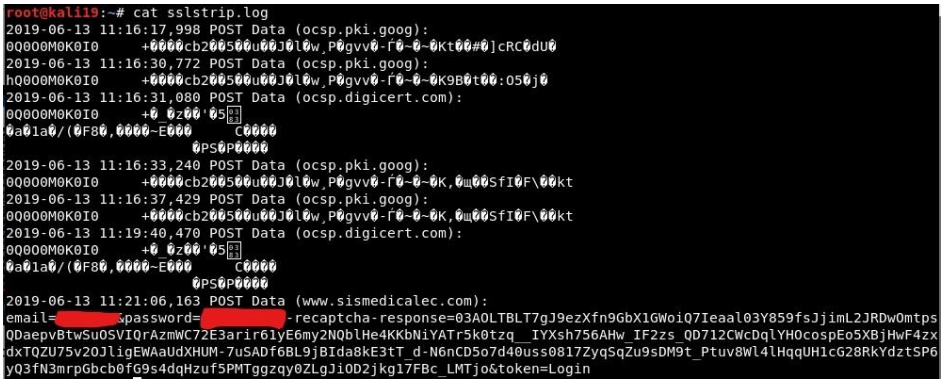


Figura C.15: Captura de información con las credenciales de la víctima

Luego de que el usuario del aplicativo ingresó al sistema con sus respectivas credenciales, la captura de información que realizó la **herramienta sslstrip,** como se puede observar en la figura C.15 en la parte inferior de la misma donde dice email= ?, está el nombre del usuario y en la parte que dice password=? corresponde a la contraseña de dicho usuario, lo que significa que se obtuvo satisfactoriamente las credenciales de ingreso, de la víctima.

Figura C.16: Navegador con la URL, para ingresar al sistema web

En la figura C.16 se observa la URL del respectivo aplicativo web, ingresada desde el navegador del atacante.

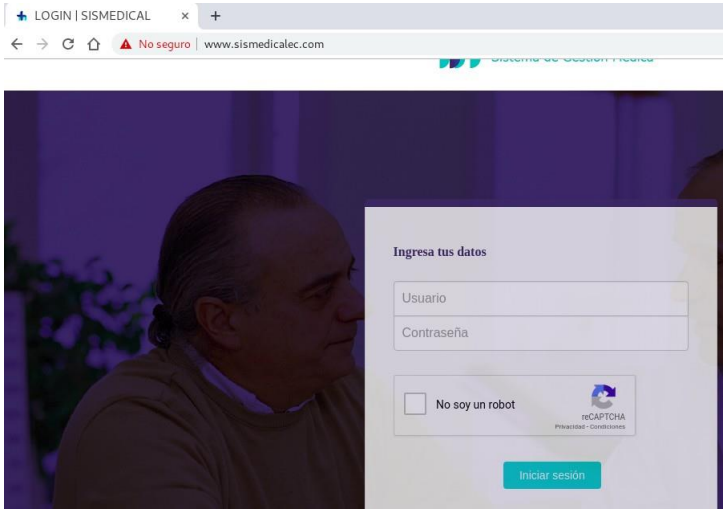


Figura C.17: Página de login del sistema

Página de ingreso de credenciales que presenta al sistema web sismedicalec.com, como se observa en la figura C.17

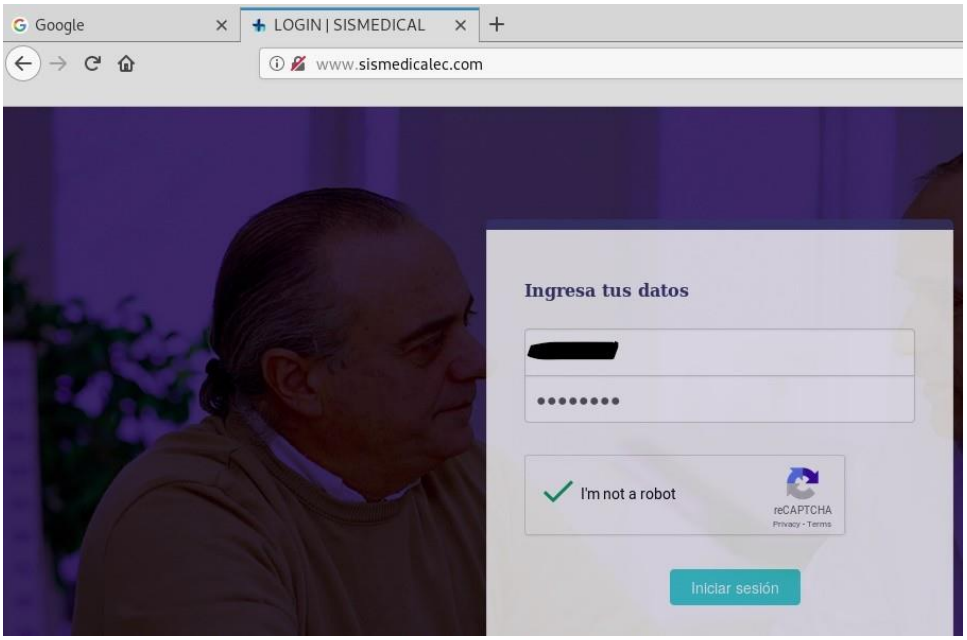


Figura C.18: Página para ingreso del login en la aplicación web

Luego de que el atacante obtuvo las credenciales procede a su respectivo ingreso dentro del sistema web como se muestra en la figura C.18

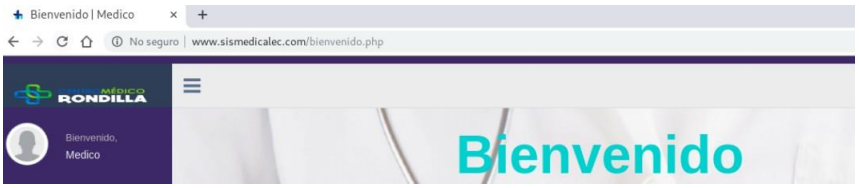


Figura C.19: Ingreso exitoso al sistema, pantalla de bienvenida

En la figura C.19 se puede observar que el ingreso de credenciales fue correcto y desde la máquina del atacante logró satisfactoriamente, el robo de información y acceso total al sistema.

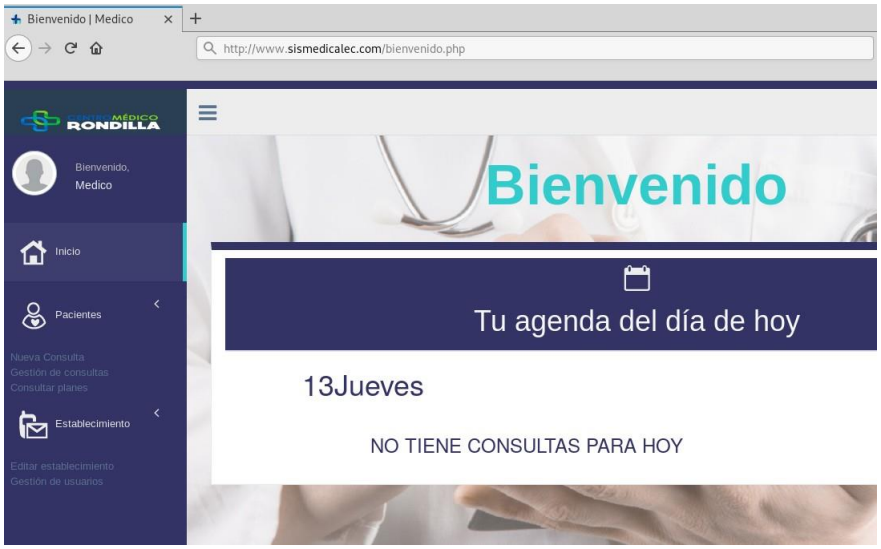


Figura C.20: Pantalla de bienvenida del sistema web

Según muestra la figura C.20 luego de haber ingresado al sistema, el atacante empieza a navegar dentro del aplicativo “sismedicalec.com”.

Anexo D

Anexo

Test de inyección SQL, para tratar de vulnerar la posible debilidad en la aplicación web denominada “Possible Source Code Disclosure”

1. Acceder al navegador de su preferencia, para este caso se escogió Mozilla Firefox

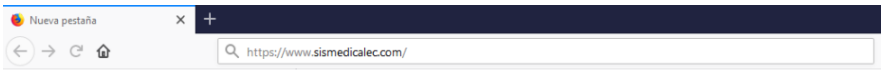


Figura D.1: Ingreso de la URL, en el buscador Mozilla Firefox

2. Posteriormente al digitar la URL, del sistema web se muestra de la siguiente forma.

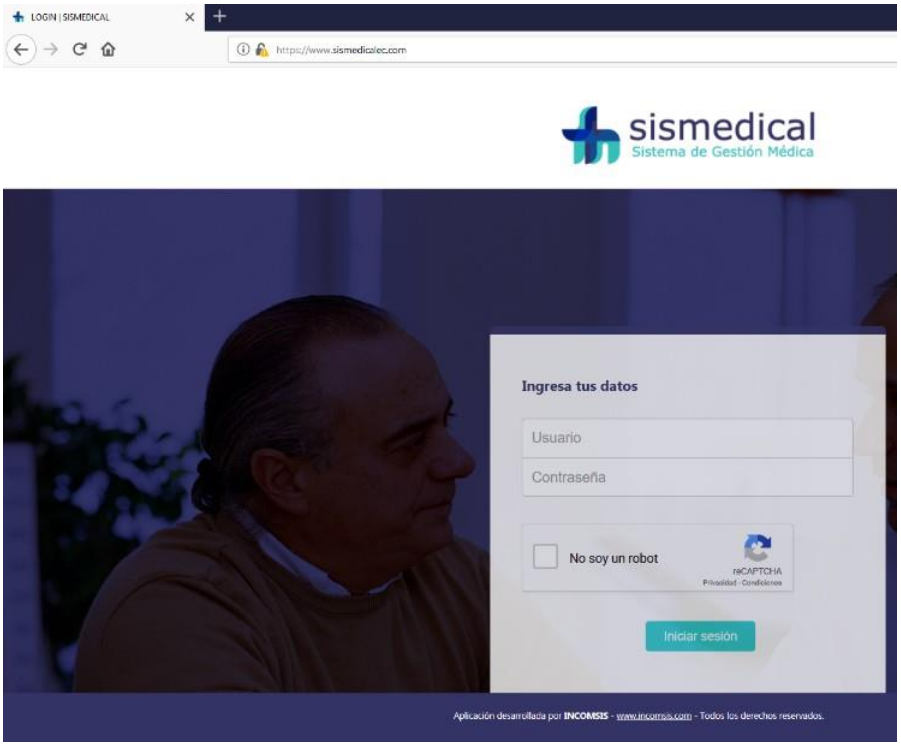


Figura D.2: Inicio de sesión en el aplicativo web Sismedicalec

3. En la página de inicio de sesión se escribe el código sql para intentar ingresar al aplicativo web sin las credenciales correspondientes

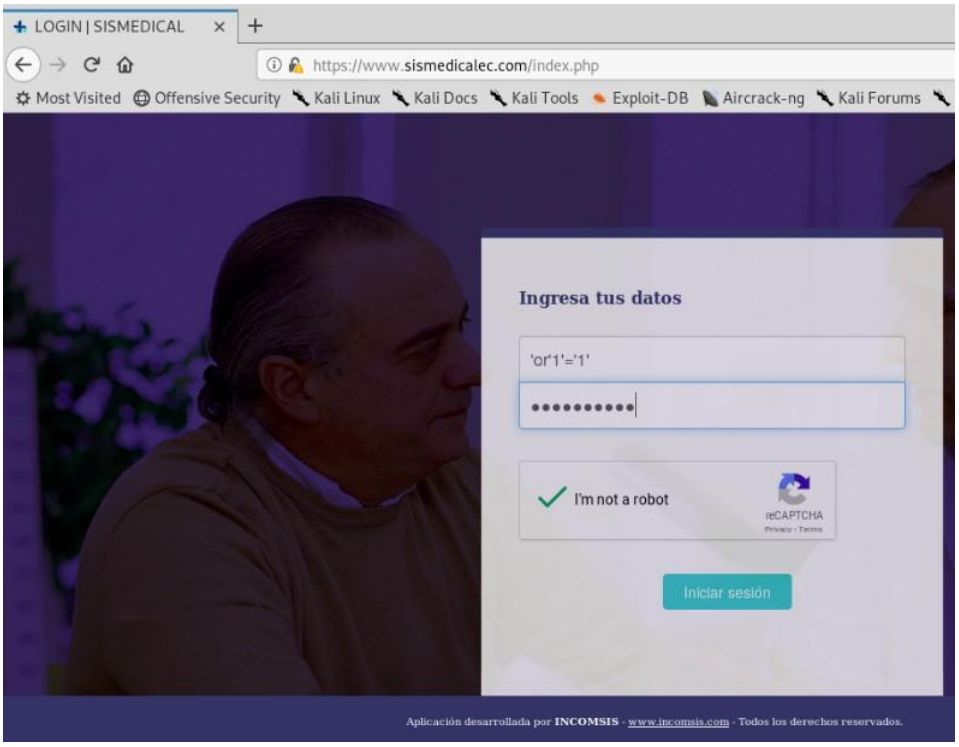


Figura D.3: Ingreso de código para inyección sql, ingresado en el aplicativo web

Como se observa en la figura D.2 se procede a ingresar el código sql para intentar la inyección de código malicioso.

4. Ahora se observa el resultado después de ingresar el código sql, **'or'1'='1'** en la parte del login del aplicativo web

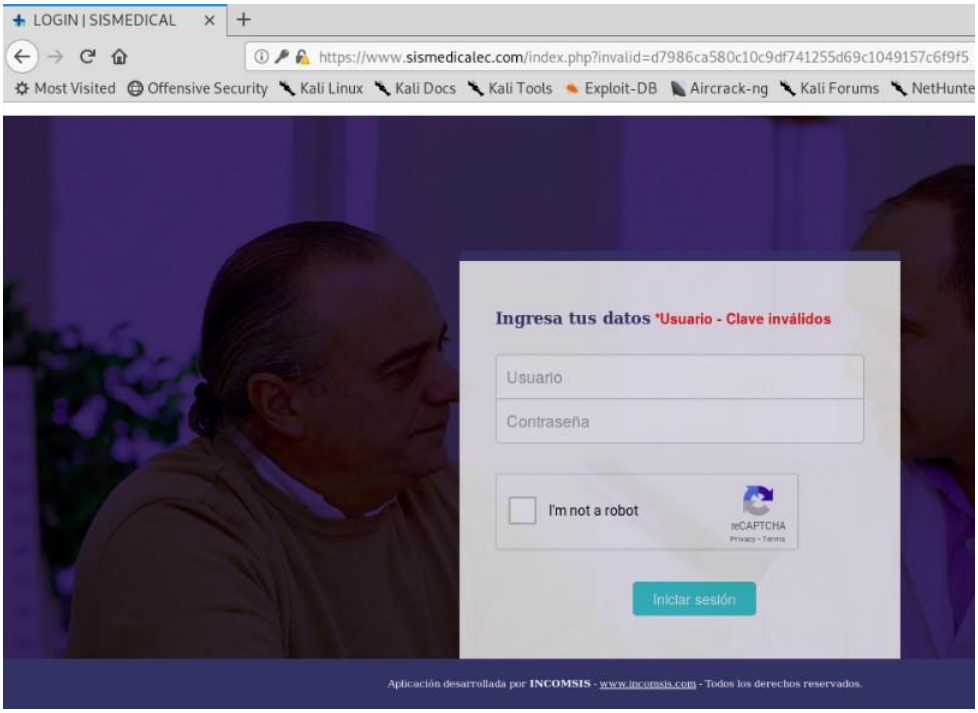


Figura D.4: Prueba de inyección sql, desde el login del aplicativo web

Como se muestra en la pantalla de la figura D.3 se intentó vulnerar el inicio de sesión mediante código sql, y el resultado da que no es vulnerable a inyección de código malicioso.

5. Como no se logró ingresar mediante código sql, se procede a ingresar con las credenciales de ingresa otorgadas por la empresa INCOMSIS, para verificar que tampoco al ingresar al sistema contenga alguna vulnerabilidad de inyección de código, por el paso de parámetros.

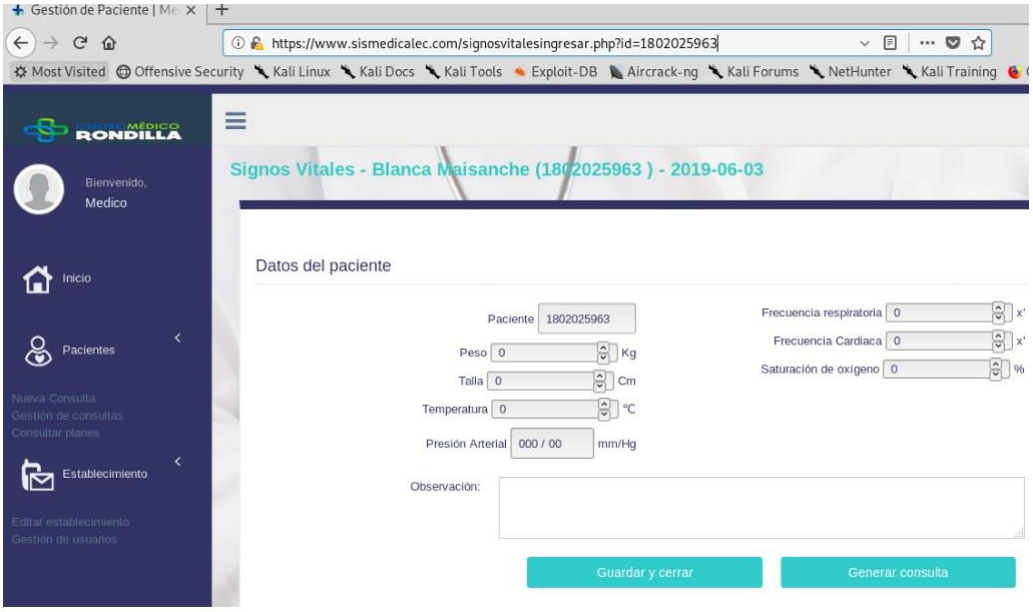


Figura D.5: Página de ingreso de signos vitales del paciente

En la **figura D.4** se muestra una página web para el ingreso de signos vitales del paciente, donde se permite paso de parámetros y es posible ingresar caracteres especiales para verificar si es vulnerable a un ataque por inyección sql a continuación se ingresa una comilla simple, dentro de la URL, que permite paso de parámetros.

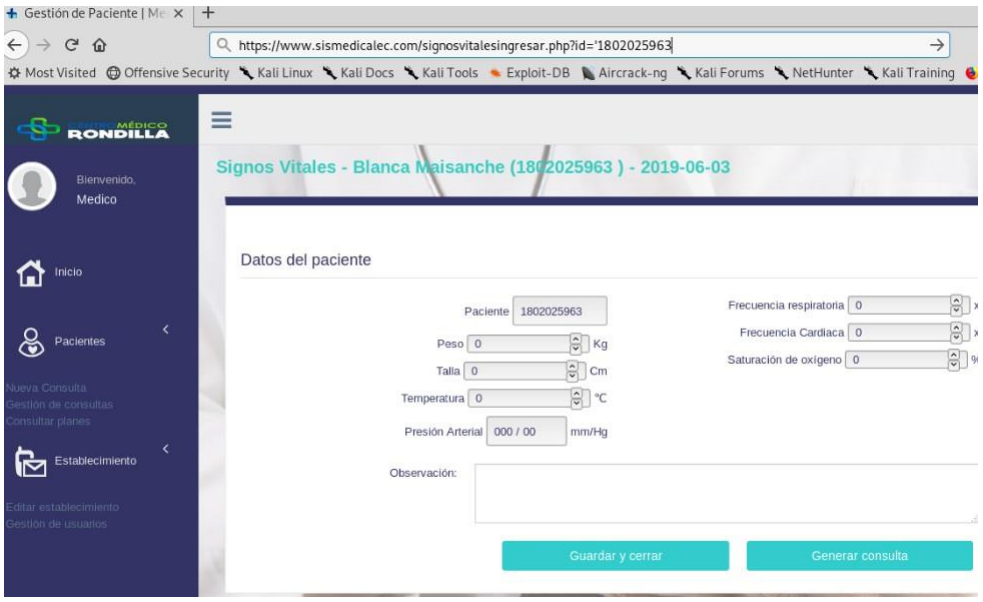


Figura D.6: Ingreso de caracter especial en el paso de parámetros de la página signos vitales

Después del signo **(=)** igual mediante paso de parámetros se ingresa una comilla

simple, para probar si lee el caracter adicional ingresado.

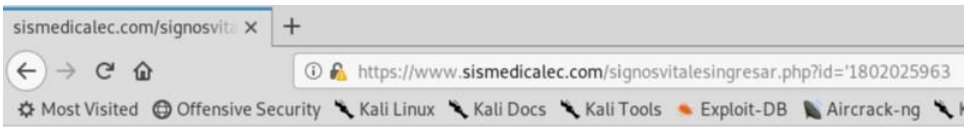


Figura D.7: Muestra del resultado al ingresar un caracter especial ' en el paso de parámetros.

Al hacer una simple prueba mediante ingreso de un caracter especial dentro de la URL, y al no encontrar información relacionada mediante los parámetros, deja la pantalla del sistema en blanco, lo que quiere decir que no se logró recuperar información de la base de datos, y con esto se confirma que el aplicativo web ha pasado el test de inyección sql, exitosamente.

Anexo E

Anexo

Procedimientos explicados paso a paso sobre la explotación de la vulnerabilidad **"Exploración de directorios"** con la herramienta DirBuster

Para iniciar el ataque informático se debe conocer la víctima como se conoce la URL del sitio web a atacar que es **"www.sismedicalec.com"**, se procede a detallar los pasos a seguir.

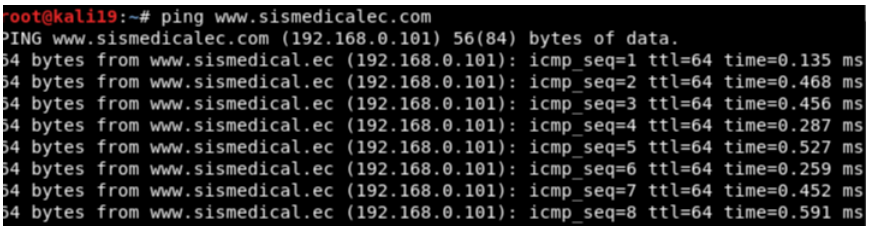


Figura E.1: Terminal que incluye un comando ping, para conocer la ip del servidor a vulnerar.

Mediante la realización de un ping a la página web se puede conocer la ip del servidor, este proceso funciona siempre y cuando la víctima y el atacante se encuentren dentro de la misma red.

Como ya se conoce la ip de la víctima en este caso es 192.168.0.101, ip del servidor, se procede a iniciar la respectiva herramienta de explotación, mediante una terminal de Kali Linux.

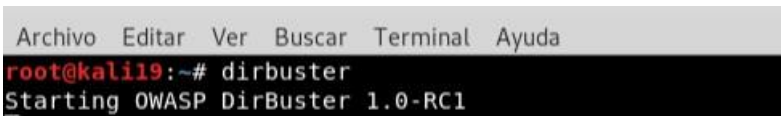


Figura E.2: Terminal en Kali Linux donde se inicia la herramienta Dirbuster, con su respectivo comando

La siguiente figura muestra la interfaz de inicio de DirBuster.

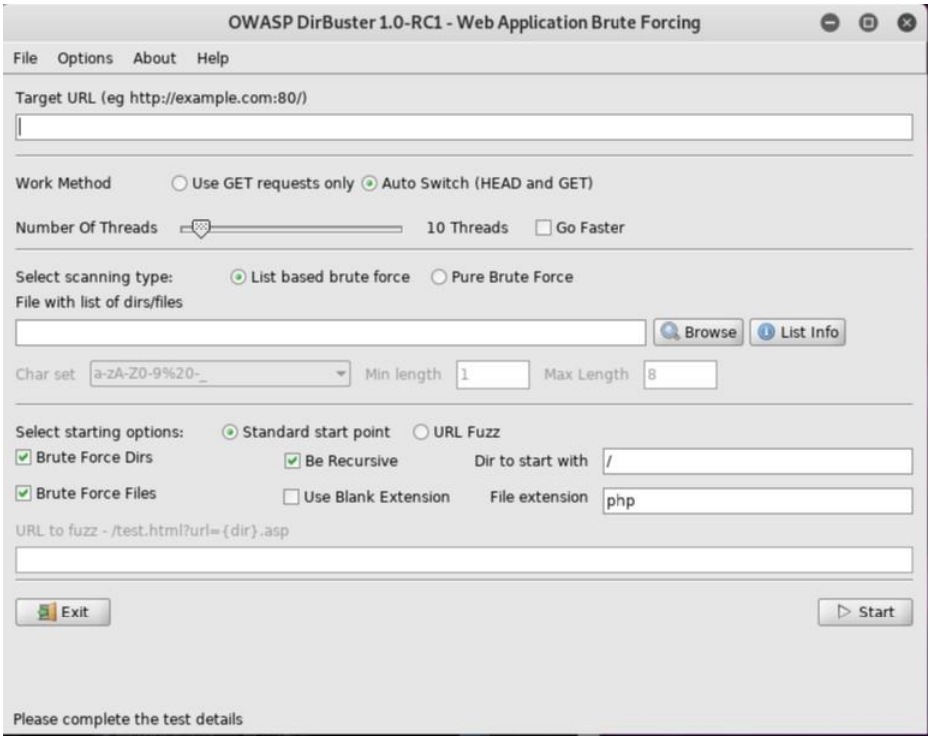


Figura E.3: Pantalla de inicio de la herramienta DirBuster

El siguiente paso es colocar la ip de la víctima seguido del puerto en este caso es el puerto para páginas con https, 192.168.0.101:443, se coloca en el Target URL.

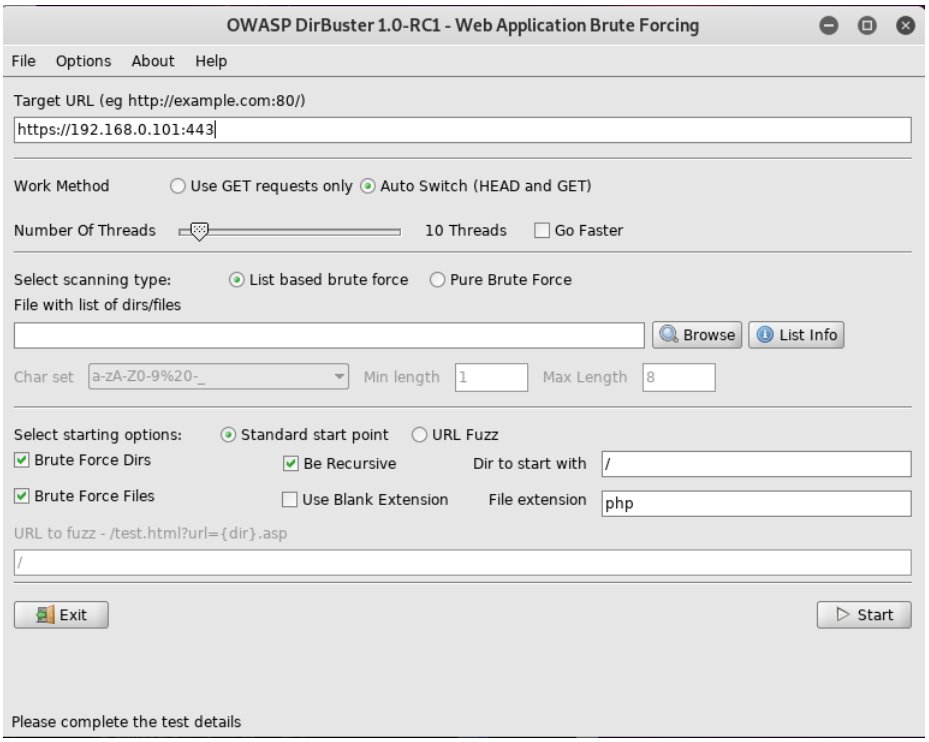


Figura E.4: Pantalla de ingreso del target a escanear

Como siguiente paso se busca un archivo .txt wordlists, en el directorio que se encuentra en DirBuster para ello se escribe en la terminal **locate dirbuster.**

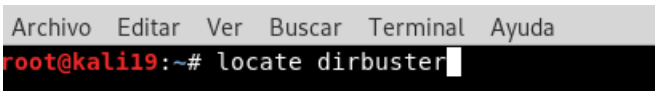


Figura E.5: Comando para buscar el directorio de los wordlists de DirBuster

El wordlists que se necesita para la práctica es **small.txt,** en la figura E.6 indica el directorio para encontrar dicho archivo luego se busca mediante interfaz dicho directorio.

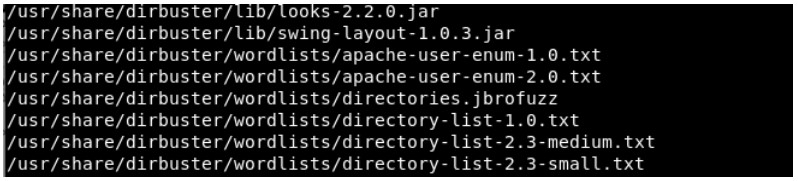


Figura E.6: Directorio de archivos de DirBuster donde se encuentra small.txt

La figura E.6 muestra el parte del directorio que contiene por defecto la herramienta DirBuster, e indica el archivo que se necesita para esta ocasión, small.txt

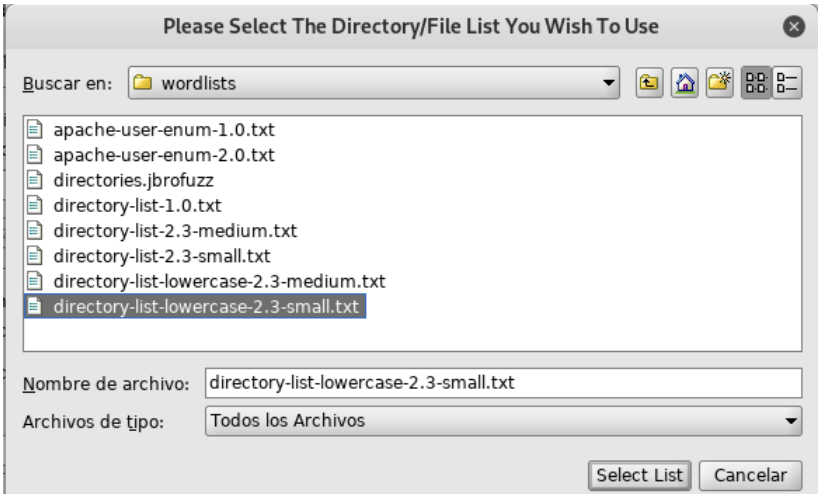


Figura E.7: Pantalla para seleccionar el directorio del archivo small.txt

En dicho directorio se encuentra el archivo que se debe elegir para que inicie el escaneo de los directorios, luego de escoger dicho archivo la interfaz queda de la siguiente forma:

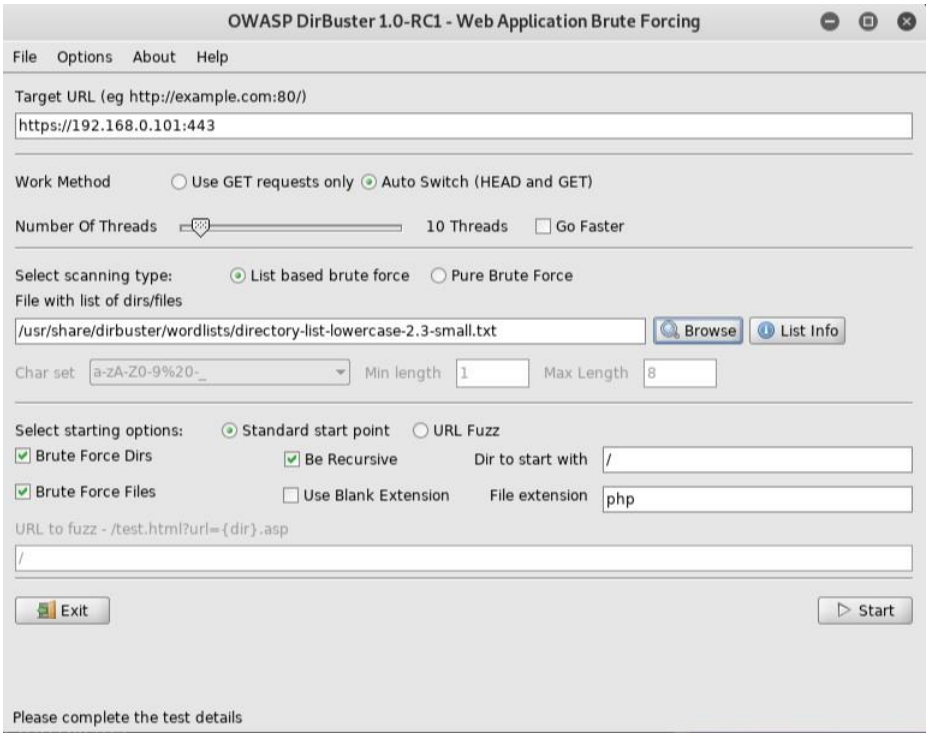


Figura E.8: Pantalla con los datos necesarios par iniciar el escáner de directorios.

Cuando los datos correctos han sido ingresados en la pantalla, se procede a hacer clic en Start para que empiece el escaneo de directorios.

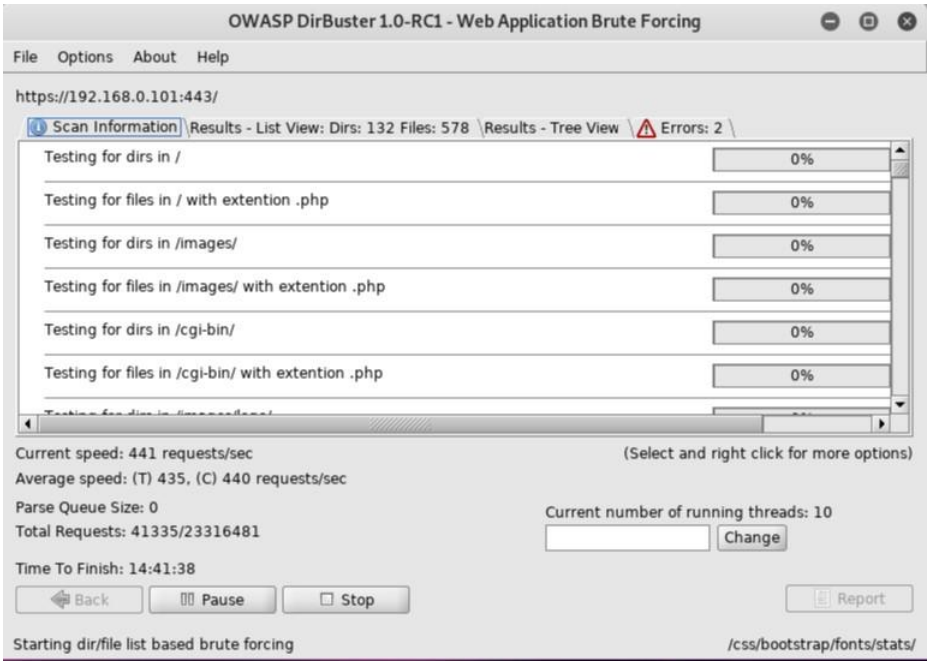


Figura E.9: Inicio del escáner de directorios del sistema web, mediante la colocación de una ip.

Como se observa en la figura E.8 ya inició el escáner de directorios, y se puede observar el listado mediante consola e interfaz como se observa a continuación

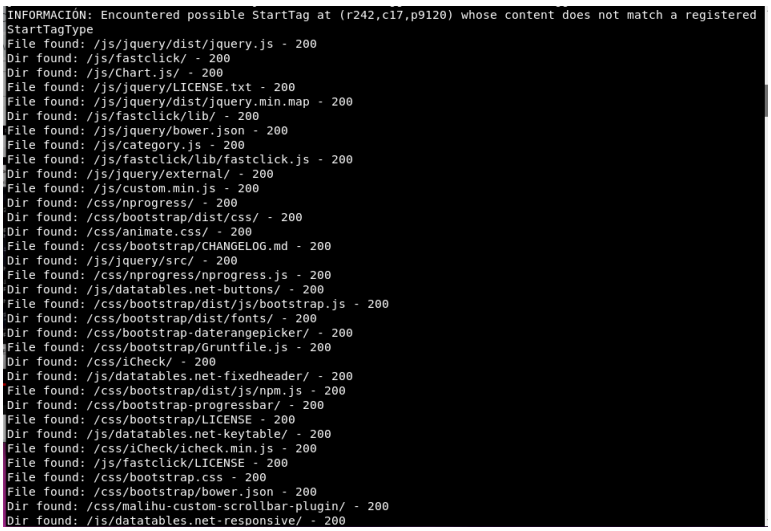


Figura E.10: Terminal donde se muestra el directorio de imágenes y archivos incluidos en el aplicativo web

Y mediante la interfaz de DirBuster en la opción Result-List View que muestra 752 vistas de directorios

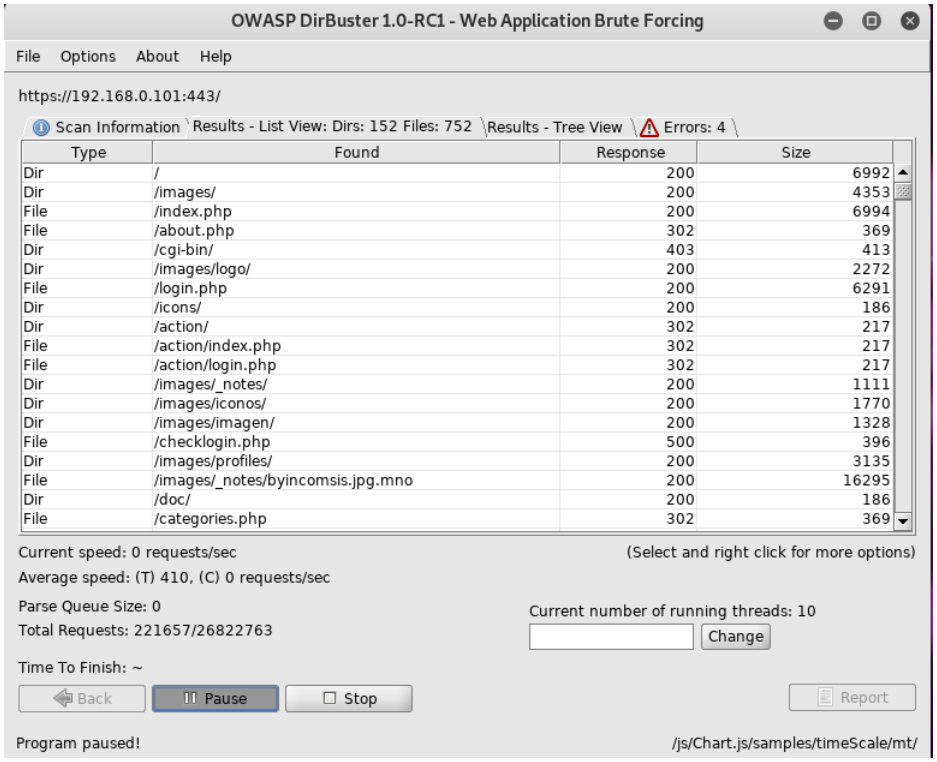


Figura E.11: Pantalla de directorios incluidos en la opción Resul-List View

Como se puede observar en la figura E.10 ya empezó a listar todos los directorios encontrados en la aplicación web "sismedicalec", ahora para observar la información que hay en dicho directorio se hace un clic derecho sobre algún de estos directorios que contenga una respuesta número 200. A continuación se indica lo que se debe realizar para buscar el archivo específico.

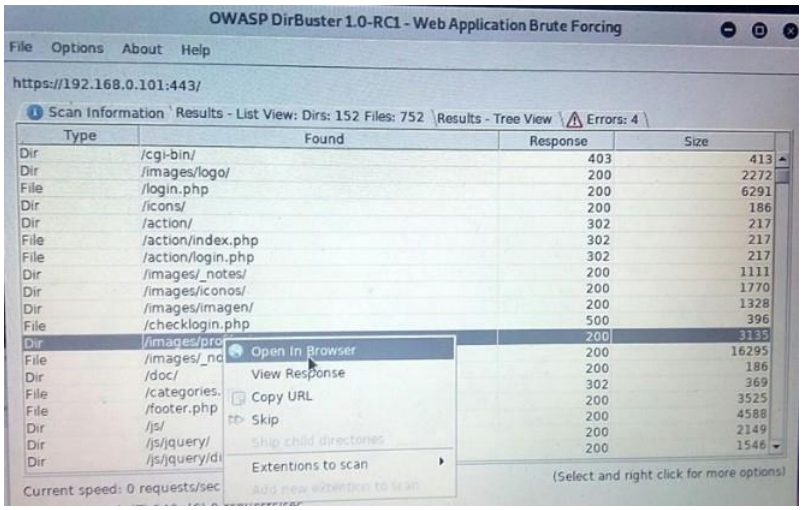


Figura E.12: Pantalla de Result-List View con un clic derecho para abrir los archivos.

Al hacer un clic sobre el directorio que enlista las imágenes se muestran en el navegador como se indica a continuación.

Figura E.13: Pantalla de directorios mostrados mediante un navegador web.

En la figura E.12 se puede observar lo que contiene el directorio de imágenes de perfil, y al hacer un clic sobre alguna de estas imágenes se muestra como tal.
Se selecciona dentro del directorio por ejemplo donde dice hojamenbretada1.png

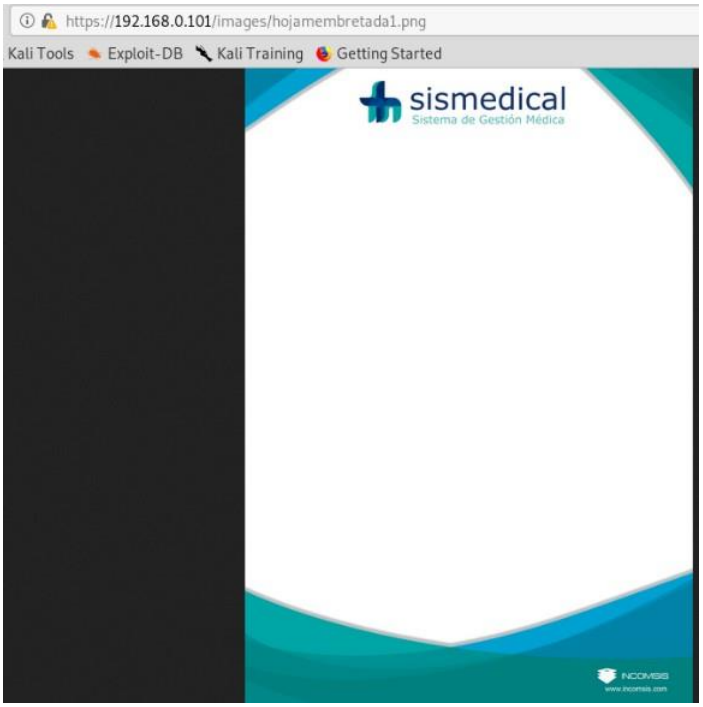


Figura E.14: Muestra de la imagen desde el navegador, con el nombre de hojamembretada1

La exploración de directorios lista, imágenes, archivos y también la ubicación de las páginas del sistema web, como se puede observar en la imagen E.14

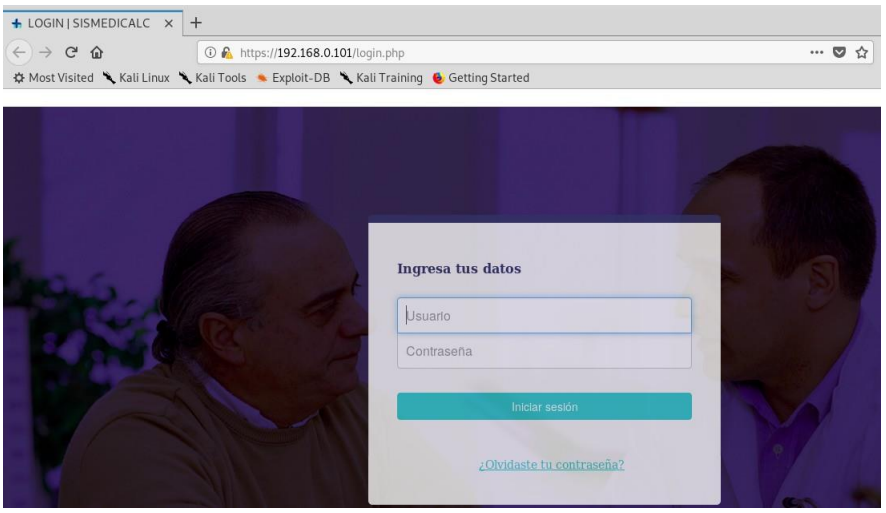


Figura E.15: Pantalla desde un navegador donde muestra la interfaz de la página login.php

Ataque a la vulnerabilidad “Exploración de directorios”, realizada con éxito, los resultados obtenidos son los esperados para el presente proyecto de investigación.

Anexo F

Anexo

Para realizar una prueba de stress al aplicativo web, se necesitará la ayuda de una herramienta llamada Apache Bench, la cuál va a ayudar a que el test se realice con éxito.
Como primer paso se procede a actualizar los repositorios para instalar dicha herramienta.

```
root@kali19:~# apt-get update
```

Figura F.1: Comando para actualización de repositorios en Kali Linux

Como segundo paso se debe instalar el paquete apache2-utils para obtener acceso a Apache Bench.

```
root@kali19:~# apt-get install apache2-utils
```

Figura F.2: Comando de instalación de la herramienta Apache Bench

A continuación, crea el usuario que manejará la ejecución de actividades. No es una buena idea ejecutar algunos de los comandos en la siguiente sección como root

```
root@kali19:~# useradd -m -d /home/test -s /bin/bash -g sudo pruebas
```

Figura F.3: Comando para agregar un nuevo usuario y usarlo para las pruebas de stress

A continuación se debe ingresar como el usuario pruebas para empezar a ejecutar los comandos con el uso de Apache Bench.

```
root@kali19:~# su pruebas
pruebas@kali19:/root$
```

Figura F.4: Comando para cambiar de usuario root a usuario pruebas

Una vez que se ingresa con este usuario se procede a ejecutar los comandos para el test de stress

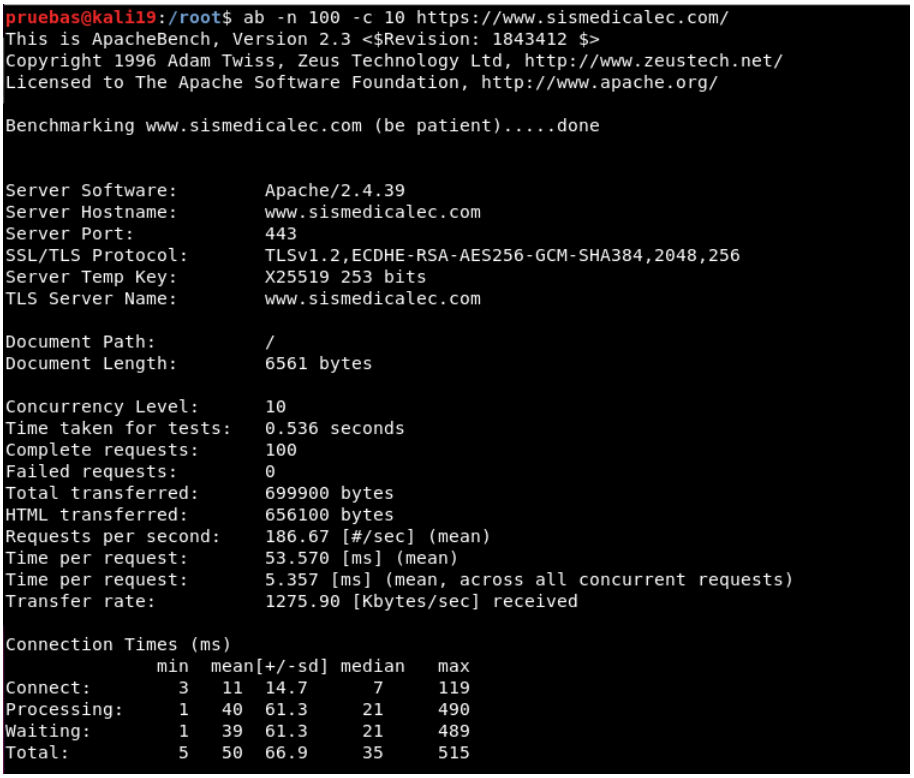


Figura F.5: Comando para iniciar solicitudes a la aplicación web

Para empezar la primera petición se hace en una escala baja, como se puede observar en la figura F.5, la misma que representa el ejecutar una prueba con Apache Bench, donde se utiliza 100 solicitudes a la aplicación web con una concurrencia de 10. Y com se puede observar esta petición se llevó con éxito y tomó tiempo de 536 segundos.

Ahora se inicia una petición con rangos más altos, tanto en solicitudes como en concurrencia, como se puede observar en la figura F.6

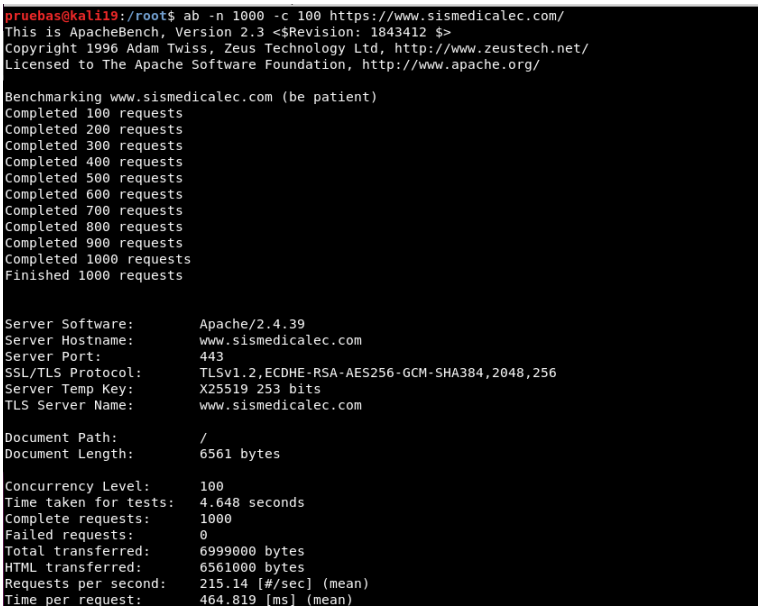


Figura F.6: Ingreso de comando para la ejecución de pruebas de stress

Como esta solicitud es más alta que la anterior se puede observar que el tiempo de la petición según la figura F.6 es de 4.648 segundos en total en la respuesta. Ahora se lleva a cabo una nueva petición estableciendo parámetros diferentes como es: -c parámetro especifica el número de conexiones, -k son soportes para HTTP Keep-Alive, y -t es el parámetro que establece el tiempo en segundos durante el cual cada conexión está activa.

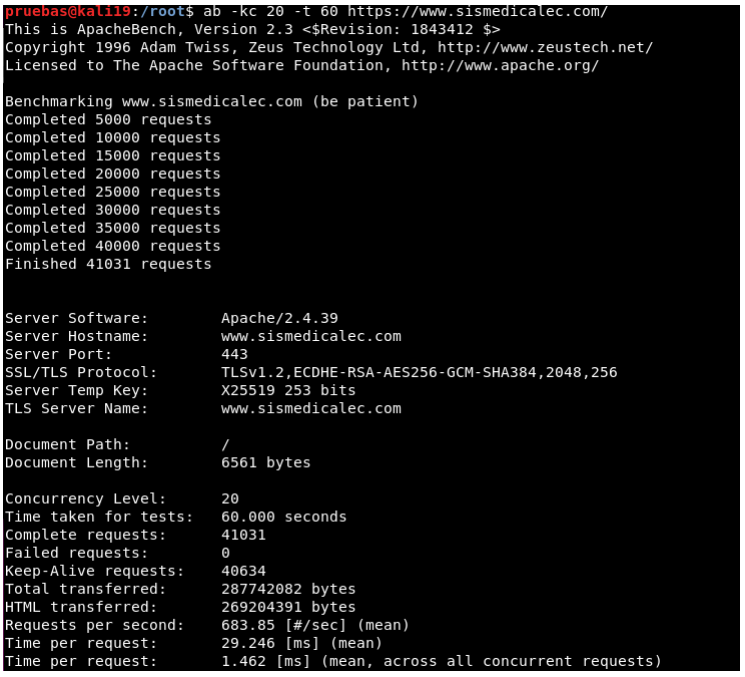


Figura F.7: Ingreso de comando para una nueva solicitud al aplicativo web

Esta petición muestra información requerida según los parámetros ingresados en el comando como se observa en la figura F.7 y con un tiempo de 60 segundos como fue establecido.

Para realizar una prueba de "inundación", se establecemos el número de solicitudes (-n) a digamos 5000, y se asigna el número de conexiones concurrentes (-c) a algo así como 200.

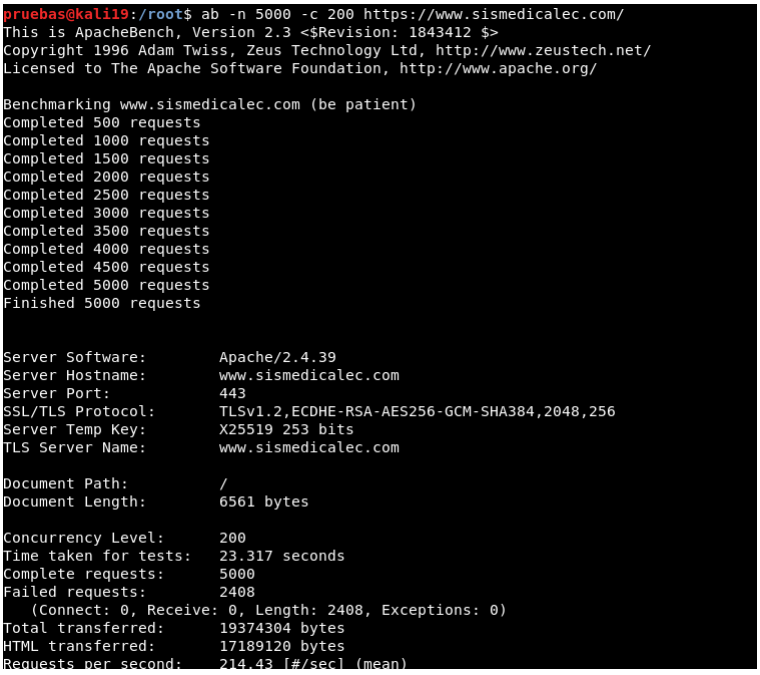


Figura F.8: Comando para realizar solicitudes al sistema “sismedicalec”

Según se observa en la figura F.8 la petición se llevó con éxito y revela cierta información relevante como es la versión de Apache que usa la aplicación web que está en investigación.

Y también revela el tiempo que demoró cuando terminó las 5000 solicitudes que fue de 23.317 segundos.