ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"HACKING ÉTICO A UNA ORGANIZACIÓN PRIVADA QUE POSEE CONTENIDO DE SALUD ALOJADOS EN LA NUBE BASADO EN EL ASEGURAMIENTO DE LA INFORMACIÓN ELECTRÓNICA DE SALUD PROTEGIDA (ePHI)"

EXAMEN DE GRADO (COMPLEXIVO)

Previo a la obtención del título de:

MAGISTER EN SEGURIDAD INFORMÁTICA APLICADA

DENISSE AURORA CAYETANO CARVAJAL

Guayaquil – Ecuador

Año: 2016

AGRADECIMIENTO

La realización del siguiente proyecto no hubiera sido posible sin la aportación de muchas personas incluyendo mis profesores de maestría que compartieron sus conocimientos y experiencia y aportaron a mi crecimiento profesional. Agradezco a la empresa donde laboro por permitirme hacer uso de los recursos técnicos para la realización del siguiente trabajo.

DEDICATORIA

Dedico el siguiente trabajo a mi madre Marlene, a mi padre Raúl y a mis hermanos Stephanie, Raúl y Keyla, quienes durante el proceso de clases comprendieron el tener que perder fines de semana con el objetivo de concluir este proceso.

A mi prometido Walter que me ha apoyado y motivado a culminar con el proyecto. A mis compañeros de maestría Christian, Diego, Luis, Ivette y Margarita quienes, además son mis grandes amigos.

TRIBUNAL DE SUSTENTACIÓN

MGS. Robert Andrade

MIEMBRO DEL TRIBUNAL DE SUSTENTACÓN

MGS. Néstor Arrega

MIEMBRO DEL TRIBUNAL DE SUSTENTACÓN

RESUMEN

En el presente trabajo se realizara un hacking ético de caja blanco en una empresa privada que implementa y aloja sus servicios en la nube. Se ha seleccionado que sea de caja blanca puesto que se tiene como objetivo principal encontrar posibles vulnerabilidades que no sean conformes con las normas de HIPAA cuyo objetivo principal es la protección de información de salud y personal de pacientes.

El proceso para cubrir posibles vulnerabilidades fue iniciado 3 en los meses anteriores y se pretende medir si ha sido suficiente o es necesario implementar nuevos cambios para cumplir con la conformidad HIPAA.

Durante el hacking ético se detectó que las herramientas que se utilizan normalmente no encontraron vulnerabilidades de tipo de alto riesgo por lo que se hizo un análisis más profundo usando el código y las herramientas que se utilizan para el desarrollo, así también incluimos los procesos para obtener información de riesgos que podrían tener internamente. Gracias a este trabajo pudimos obtener vulnerabilidades que se han listado e indicado las posibles soluciones.

ÍNDICE GENERAL

ABREVIATURAS Y SIMBOLOGÍA

API	Interfaz de programación para aplicaciones
ePHI	Información de Salud Electrónica Protegida
HIPAA	Ley de Responsabilidad y Portabilidad del Seguro de Salud
HTTPS	Protocolo Seguro de Transferencia de Hipertexto
SQL	Lenguaje de Base de Datos Relacionales

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En Estados Unidos, desde el año 1996, se crea la Ley Health Insurance Portability and Accountability Act (HIPAA por sus siglas en ingles), donde se indican la infraestructura y procedimientos que debe cumplir una organización para el manejo seguro de la información de salud de un paciente, y así proteger a la persona.

Entre las recomendaciones de HIPAA están: obtener el consentimiento de los pacientes al recolectar información, especificar quien tendrá acceso, como se utilizara y cuando será revelada la información obtenida, además de incluir el hacking ético para evaluar la robustez de los controles implementados y de esta forma evaluar el cumplimiento de las conformidades [1].

Desde el año 2002, el incumplimiento de HIPAA puede tener consecuencias importantes para una organización que van desde sanciones económicas, civiles o legales; como por ejemplo: la posibilidad de demandas legales, multas y prisión si se obvia dicha falta de seguridad [1].

En la actualidad, la Ley HIPAA, incluye reformas que obliga a cualquier empresa privada, con servicios alojados en la nube, que tiene sede en Estados Unidos; cumplir la ley y asegurar la protección de dicha información. Por ello es importante que una empresa que maneja información de salud realice los procesos necesarios para el aseguramiento de las normas propuestas en HIPAA.

CAPÍTULO 1

1. GENERALIDADES

1.1. Descripción del problema

Los administrados del servicio web que posee la organización privada, han iniciado un análisis de las posibles vulnerabilidades, basados en su experiencia y conocimiento del código e infraestructura, de tal manera se han realizado correcciones para apegarse a las directrices de las conformidades de HIPAA; sin embargo una de ellas es la implementación de controles para el aseguramiento de cumplimiento, controles que no han sido implementados en su totalidad y que son fundamentales con el fin de evitar multas o posibles demandas.

Entre los cambios previamente realizados se consideró encriptar datos como contraseñas, nombres, teléfonos, email, entre otras que representan información personal de un usuario tanto en las tablas, asi como también filtrar estos datos de logs del servidor y de la aplicación en general.

Es importante mencionar que los clientes de la entidad son otras compañías que tienen acceso a los servicios y para ello utilizan una clave de acceso; es decir el API de la empresa no esta intencionado para uso un usuario final, sino mas bien de otra entidad que pretende ofrecer un servicio en base a la funcionalidad e información que puede proporcionar el API.

1.2. SOLUCIÓN PROPUESTA

Con la finalidad de cumplir con los requisitos de HIPAA se desea seguir una de sus principales recomendaciones: La implementación de controles, uno de los recomendados, el hacking ético tanto externo como interno.

La organización posee varios servicios que contienen información de salud de usuario y si bien es requerida la autenticación para acceder a dichos servicios es necesario garantizar que son transmitidos en forma segura utilizando HTTPS y que existe un debido control en el acceso a

dicha información.

Se plantea hacking ético de caja blanca, puesto que, se pretende obtener resultados específicos, en función a que nuestro objetivo es puntual y requiere foco central en encontrar vulnerabilidades en el acceso de datos privados de usuarios [2].

Para asegurar que empleados no autorizados no cuenten con acceso a esta información protegida, se requiere una evaluación interna, donde se utilizaron técnicas de Ingeniería social para asegurar que los controles existentes y que van de la mano con las conformidades de HIPAA.

Como parte del presente trabajo se tiene como objetivo proporcionar una lista de recomendaciones necesarias para cubrir vulnerabilidades encontradas durante el proceso y reforzar los controles ya implementados con la finalidad de un mejoramiento de dichos controles.

CAPÍTULO 2

2. METODOLOGÍA DE DESARROLLO DEL PROCESO DE HACKING

2.1. PLANEACIÓN DEL PROCESO

Al utilizar un hacking ético de caja blanca contamos con información previa como: dominio objetivo a atacar, dirección IP, datos del servidor, lenguaje que utilizan para la codificación e información de la base de datos. Sin embargo incluiremos una etapa de reconocimiento.

En la etapa de reconocimiento no se invertirá mucho tiempo puesto que ya poseemos información previa, se utilizaran varias herramientas para

tener resultados de que tipo de información es pública o fácilmente accesible por un agente externo a la compañía.

Como paso siguiente se realizara un análisis de las vulnerabilidades a fin de encontrar un hueco de seguridad que nos permita realizar la explotación del servidor.

Posteriormente, realizaremos un análisis de los logs del servidor con el fin de recolectar información personal de los usuarios. Este paso se hace para saber que tanta información de este tipo podría recolectar un atacante en caso de tener acceso a ellos.

Para finalizar se nos facilitó una estación de trabajo de uno de los empleados de la compañía con las credenciales de acceso, de esta forma se podrá medir el impacto que podría tener un atacante interno con información y acceso a un equipo de la compañía.

2.2. HACKING ÉTICO

2.2.1. Reconocimiento

En la etapa de reconocimiento se tenía como objetivo encontrar la información más visible y accesible para un hacker que desea tener acceso a la información de la empresa. Para ello

hemos utilizado la herramienta Maltego, la cual fue corrida bajo la opción “Company Stalker” y “Huellas Nivel 3”, que realiza una búsqueda en la web para encontrar e-mails, servidores, páginas con información del dominio, etc.

2.2.2. Análisis de Vulnerabilidades

Las Vulnerabilidades fueron analizadas usando Nessus que es una herramienta para este propósito y para ello utilizamos la opción “Scan Web Vulnerabilities (complex)” que realiza chequeo de vulnerabilidades como [3]:

- Inyección de código SQL
- Cross-site scripting (XSS)
- Inyección de cabeceras HTTP
- Inclusión de archivos remotos
- Ejecución de comandos

2.2.3. Análisis de Logs

El análisis de logs será necesario por lo que para este propósito nos han proporcionado el log de eventos, del servidor web, y logs creados por la aplicación. Para el análisis de logs realizaremos búsquedas a través de comandos y se analizarán las urls con mayor potencial a contener información personal de los usuarios que es el objeto a proteger en este trabajo.

2.2.4. Análisis del Código

Para esta parte del proceso se utilizó una herramienta automatizada que chequea el código en busca de posibles vulnerabilidades, este chequeo es un poco mas profundo puesto que nos interesa tener detalles en vista que se ha concedido el acceso al código que esta escrito en Ruby y utilizan el framework Rails. Para ello utilizaremos la herramienta Brakeman que es una librería especializada en código Ruby on Rails.

2.2.5. Análisis de los procesos

El objetivo en esta etapa es conocer que información es accesible por un empleado de la compañía y que tanta información podría

recolectarse. Para ello se ha incluido el equipo de un empleado de la compañía con el ambiente de desarrollo listo: base de datos, acceso a Github con el código fuente, herramientas utilizadas por el desarrollo y demás ambientes para el desarrollo de las búsquedas.

CAPÍTULO 3

3. ANÁLISIS DE RESULTADOS

3.1. RESULTADOS DE HACKING ÉTICO

En la etapa de reconocimiento pudimos descubrir que no existe mayor información pública: se logro recuperar 1 correo importante que es el jefe del departamento técnico, esto podría ser deducido a través del nombre en la página web donde se detalla los empleados de la compañía con sus cargos.

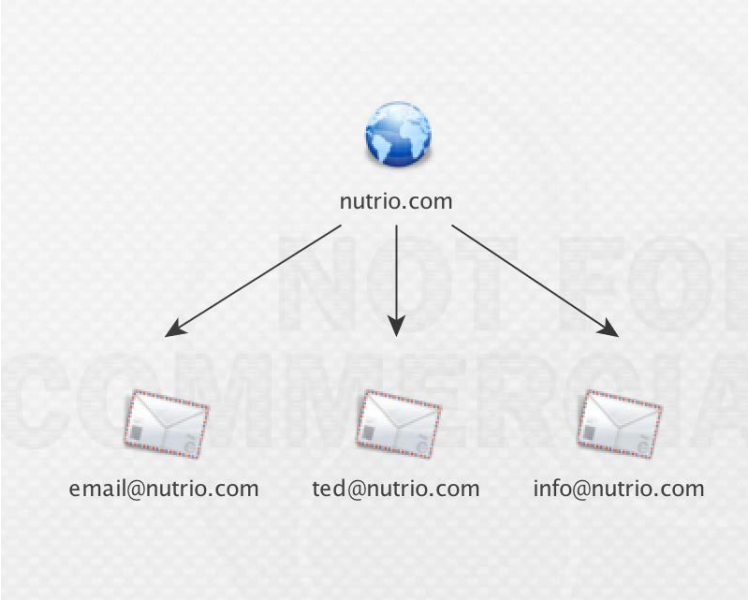

Figura 3.1 Lista de correos electrónicos. Recuperada por Maltego

A continuación, se realizó una búsqueda automatizada de huellas en la web con relación al dominio, encontrando los servidores de correo electrónico.

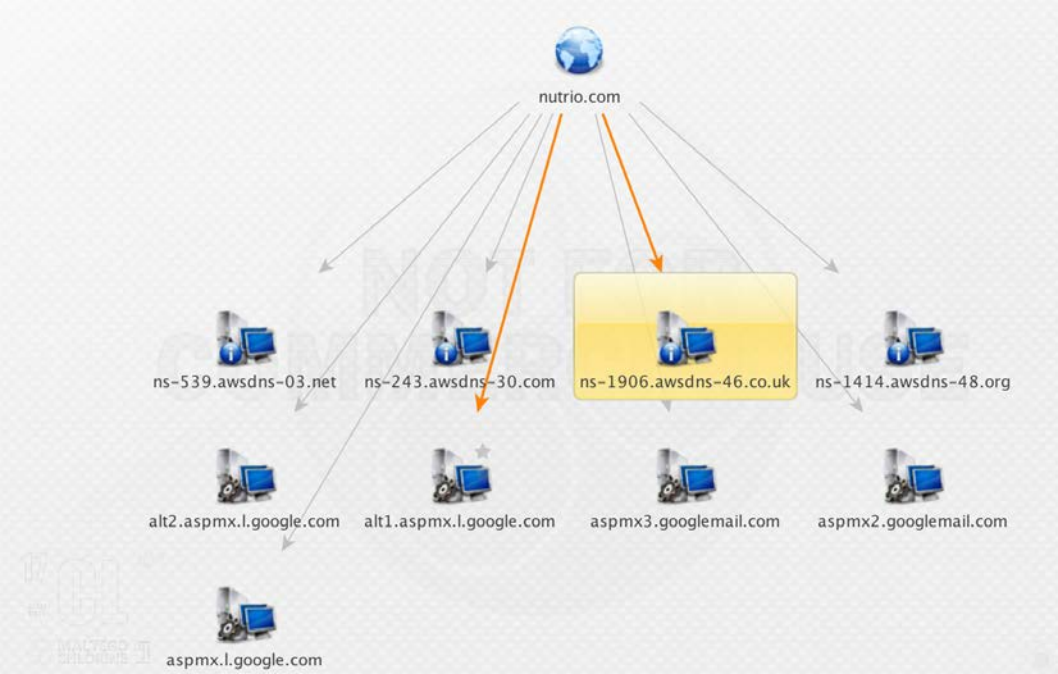

Figura 3.2 Servidores relacionados con el dominio

Los resultados del análisis de vulnerabilidades fueron satisfactorios para la compañía puesto que se realizo un análisis enfocado al API y la documentación que es publica en la web, sin embargo no se encontraron vulnerabilidades criticas o medias como permitir inyectar código por comandos o SQL, o la posibilidad de realizar Cross Site Scripting (XSS).

Summary

Critical	High	Medium	Low	Info	Total
0	0	0	0	14	14

Details

Severity	Plugin Id	Name
Info	10107	HTTP Server Type and Version
Info	10302	Web Server robots.txt Information Disclosure
Info	10386	Web Server No 404 Error Code Check
Info	10662	Web mirroring
Info	11032	Web Server Directory Enumeration
Info	11219	Nessus SYN scanner
Info	24260	HyperText Transfer Protocol (HTTP) Information
Info	33817	CGI Generic Tests Load Estimation (all tests)
Info	39470	CGI Generic Tests Timeout
Info	40406	CGI Generic Tests HTTP Errors
Info	43111	HTTP Methods Allowed (per directory)
Info	49704	External URLs
Info	84502	HSTS Missing From HTTPS Server
Info	85602	Web Application Cookies Not Marked Secure

Tabla 1. Resumen Ejecutivo Generado por Nessus

Con los resultados del análisis de código ya se pudo encontrar posibles vulnerabilidades como claves que se encontraban en el

código, representando un problema cuando estos archivos se encuentran usando versionamiento con git y un hueco de seguridad de este podría llevar a un atacante a conocer información de acceso a través del código. Así también fueron encontrados posibles bloques de código que podrían estar permitiendo ejecución de comandos e inyección SQL.

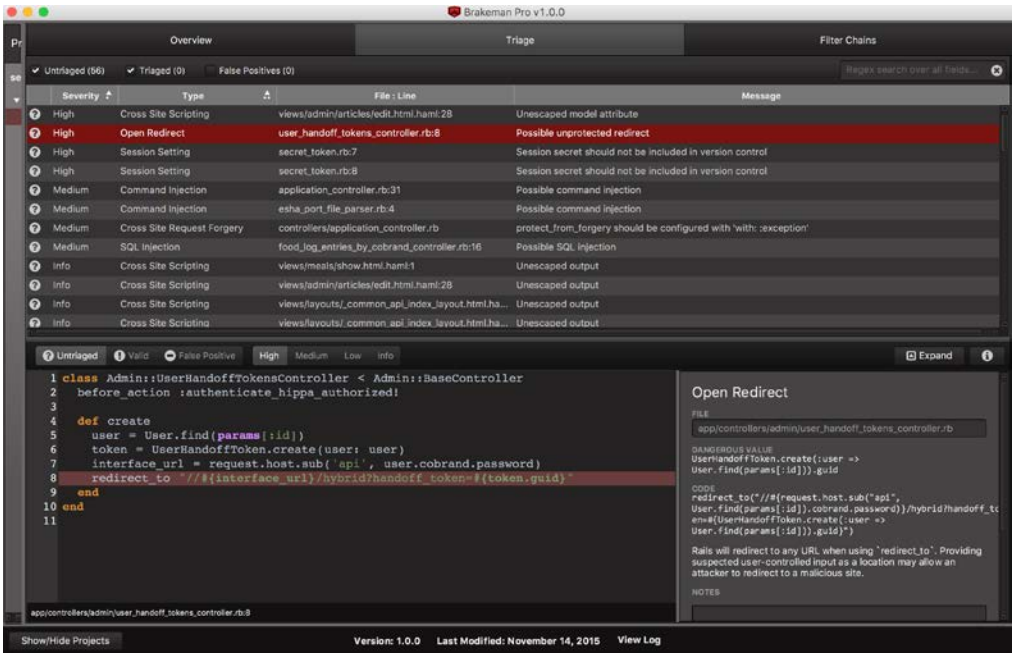

Figura 3.3 Captura de pantalla de resultados de Brakeman

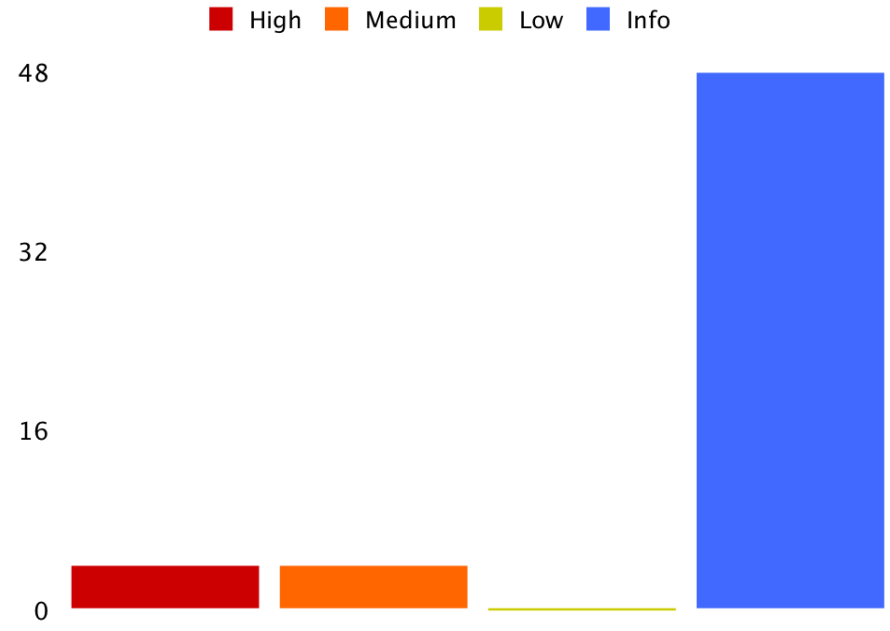

Figura 3.4 Grafica de vulnerabilidades en el código por tipo de riesgo

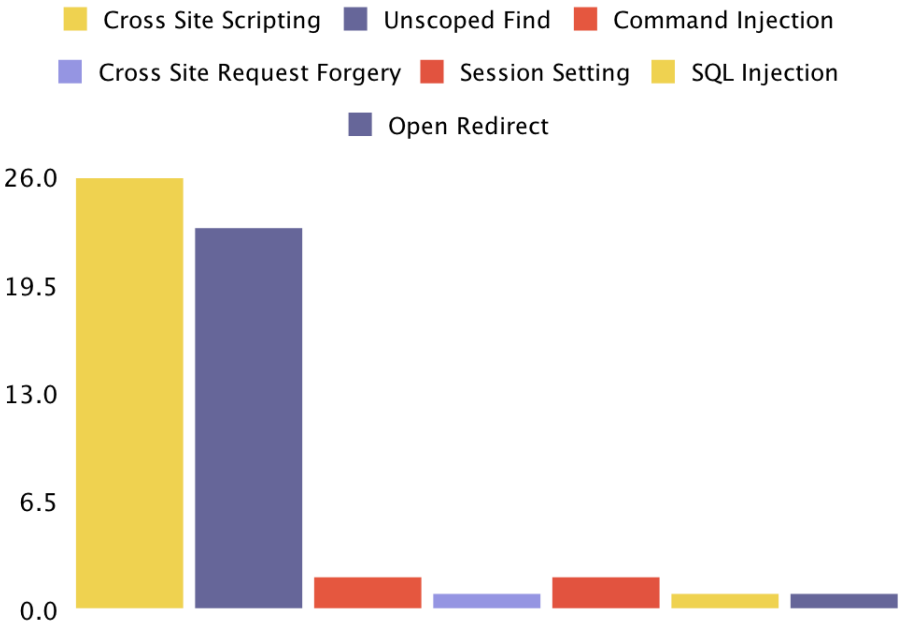

Figura 3.5 Grafica del porcentaje incidencias de vulnerabilidades por tipo

Gracias a los resultados por la herramienta de análisis de código pudimos explotar servicios específicos y obtener información como logs del sistema y ejecución de código no deseado.

Como resultados de la ultima etapa analizamos el equipo proporcionado con el ambiente de desarrollo y pudimos encontrar que la base de datos contiene información personal de usuarios encriptada, pero es posible realizar un query a través de rails para obtener la correcta, el hueco de seguridad existente en este caso es que los datos de prueba son datos reales de una versión anterior de la base de producción. Esto quiere decir que un desarrollador tiene la posibilidad de obtener información de usuarios reales.

3.2. PLAN DE RESOLUCIÓN DE VULNERABILIDADES ENCONTRADAS

Con las herramientas utilizadas no se pudieron encontrar vulnerabilidades de nivel alto, sin embargo al analizar los logs, código y equipo de un empleado desarrollador ya se pudieron encontrar vulnerabilidades que deben ser solucionadas.

A continuación una lista de las vulnerabilidades de alto riesgo:

- Logs de elasticsearch contienen información de usuarios.
  - **Solución:** La información personal de usurios deben ser filtrados de los logs de búsquedas con elasticsearch.

- Base de datos de deraollo contiene información de usuarios reales.
  - **Solución:** Ofuscar los datos personales usando técnicas como adicion de letras o números en forma aleatoria, eso permitirá preservar datos de usuarios pero les quitara validez.

- Código vulnerable a Cross Site Scripting.
  - **Solución:** Usar métodos de sanitizacion de las columnas antes de presentación en el código HTML.

- Servicios vulnerables a ejecución de código no deseado
  - **Solución:** Cambiar el código que usa métodos no seguros que permite la ejecución de código no deseado.

CONCLUSIONES Y RECOMENDACIONES

Luego de realizar las pruebas de vulnerabilidades podemos estimar que la empresa ha iniciado de manera exitosa con el proceso de conformidades con HIPAA.

Posterior a la implementación de las correcciones necesarias a las vulnerabilidades encontradas se recomienda:

1. Planear el proceso de hacking ético semestralmente, para que la mejora sea continua.

2. Iniciar la implementación de controles por escrito, puesto que, si bien los empleados han sido notificados en forma verbal; es indispensable poner por escrito las normas de la conformidad y las recomendaciones para los desarrolladores con el fin de que el nuevo código implementado preserve los actuales estándares.

BIBLIOGRAFÍA

[1] Rebecca Herold, K. B. (2014). *The Practical Guide to HIPAA Privacy and Security Compliance.* Auerbach Publications.

[2] Karina, A. B. (2013). *Hacking Ético 101, Cómo hackear profesionalmente en 21 días o menos!*

[3] Brian Marind, C. F. (2013). Web Application Scanning with Nessus. *Detecting Web Application Vulnerabilities and Environmental Weaknesses* .

[4] Muniz, J. (2013). *Web Penetration Testing with Kali Linux.* Packt Publishing Ltd.

[5] Steven DeFino, L. G. *Official Certified Ethical Hacker.*

ANEXOS

1. Reporte ejecutivo generado por Nessus con datos específicos de las vulnerabilidades encontradas.

2. Reporte generado por Brakeman con las vulnerabilidades encontradas a nivel de código.