UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

TÍTULO

IMPLEMENTACIÓN DE PRUEBAS DE PENTESTING PARA DETECCIÓN DE VULNERABILIDADES EN APLICACIONES WEB DE GRUPO ALMAR

AUTOR

Cerezo Zambrano, Jamil Javier

TRABAJO DE TITULACIÓN

Previo a la obtención del grado académico en
MAGÍSTER EN CIBERSEGURIDAD

TUTOR

Quintuña Padilla, Edisson Pompilio

Santa Elena, Ecuador

Año 2025

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

TRIBUNAL DE SUSTENTACIÓN

Firmado electrónicamente por:
ALICIA GERMANIA ANDRADE VERA

Ing. Alicia Andrade Vera, Mgtr.
COORDINADORA DEL PROGRAMA

Firmado electrónicamente por:
EDISSON POMPILIO QUINTUNA PADILLA

Ing. Edisson Quintuña Padilla, Mgtr.
TUTOR

Firmado electrónicamente por:
ALBERT GIOVANNY ESPINAL SANTANA

Ing. Albert Espinal Santana, Ph.D.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
DANIEL IVAN QUIRUMBAY YAGUAL

Lic. Daniel Quirumbay Yagual, Mgtr.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
MARIA MARGARITA RIVERA GONZALEZ

Abg. María Rivera González, Mgtr.
SECRETARIA GENERAL
UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por JAMIL JAVIER CEREZO ZAMBRANO, como requerimiento para la obtención del título de Magíster en Ciberseguridad.

TUTOR

Firmado electrónicamente por:
EDISSON POMPILIO QUINTUNA PADILLA

Ing. Edisson Pompilio Quintuña Padilla, Mgtr.

Santa Elena, 15 de diciembre de 2024

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

DECLARACIÓN DE RESPONSABILIDAD

Yo, JAMIL JAVIER CEREZO ZAMBRANO

DECLARO QUE:

El trabajo de Titulación, "Implementación de pruebas de pentesting para detección de vulnerabilidades en aplicaciones web de Grupo ALMAR" previo a la obtención del título en Magíster en Ciberseguridad, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 15 de diciembre de 2024

EL AUTOR

Firmado electrónicamente por:
JAMIL JAVIER CEREZO ZAMBRANO

Jamil Javier Cerezo Zambrano

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE CIENCIAS DE LA INGENIERÍA
INSTITUTO DE POSTGRADO

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado "Implementación de pruebas de pentesting para detección de vulnerabilidades en aplicaciones web de Grupo ALMAR", presentado por el estudiante, Jamil Javier Cerezo Zambrano fue enviado al Sistema Antiplagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 4%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

CERTIFICADO DE ANÁLISIS
magister

IMPLEMENTACIÓN DE PENTESTING PARA DETECCIÓN DE VULNERABILIDADES EN APLICACIONES WEB DE GRUPO ALMAR.

4% Textos sospechosos

4% Similitudes
0% similitudes entre comillas
< 1% entre las fuentes mencionadas
0% Idiomas no reconocidos
24% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: IMPLEMENTACIÓN DE PENTESTING PARA DETECCIÓN DE VULNERABILIDADES EN APLICACIONES WEB DE GRUPO ALMAR..pdf
ID del documento: d64406f3f4376876be3316e4f135201e273bc625
Tamaño del documento original: 5,34 MB
Autor: Jamil Javier Cerezo Zambrano

Depositante: EDISSON POMPILIO QUINTUÑA PADILLA
Fecha de depósito: 15/12/2024
Tipo de carga: interface
fecha de fin de análisis: 15/12/2024

Número de palabras: 11.253
Número de caracteres: 84.743

TUTOR

Firmado electrónicamente por:
EDISSON POMPILIO QUINTUNA PADILLA

Ing. Edisson Pompilio Quintuña Padilla, Mgtr.

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO

AUTORIZACIÓN

Yo, **JAMIL JAVIER CEREZO ZAMBRANO**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de mi propuesta metodológica y tecnológica avanzada con fines de difusión pública, además apruebo la reproducción de esta propuesta metodológica y tecnológica avanzada dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

Santa Elena, 15 de diciembre de 2024

EL AUTOR

Firmado electrónicamente por:
JAMIL JAVIER
CEREZO ZAMBRANO

Jamil Javier Cerezo Zambrano

AGRADECIMIENTO

A Dios, por su constante guía, por darme sabiduría y las oportunidades necesarias para avanzar en mi camino.

A mis padres, por brindarme una educación sólida que ha sido la base de todas mis decisiones y el pilar fundamental en mi formación personal.

A mis hermanos, por ser un modelo de profesionalismo y por el respaldo que me han ofrecido para alcanzar mis objetivos.

A mi esposa, por ser un apoyo esencial en mi vida, por estar a mi lado en los momentos más difíciles y por su constante impulso para que logre mis metas; no existen palabras suficientes para expresar mi gratitud por todo lo que haces por mí.

Jamil Javier, Cerezo Zambrano

DEDICATORIA

A mis padres, mis ángeles guardianes que me acompañan desde el cielo y son testigos de mi progreso.

A mis hermanos, quienes han sido un ejemplo que seguir tanto en su vida estudiantil como profesional.

A mi esposa, por permanecer a mi lado en los momentos más difíciles y brindarme su apoyo incondicional.

A mis tías Margarita y Lorena, por el cariño y apoyo maternal que me ofrecieron cuando más lo necesité, llenándome de fortaleza para seguir adelante.

Jamil Javier, Cerezo Zambrano

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo tiene como objetivo principal la implementación de pruebas de pentesting en las aplicaciones web de Grupo ALMAR, con el fin de identificar vulnerabilidades. Para las empresas partes del grupo este tipo de aplicaciones son componentes críticos, ya que gestionan la información que permite la continuidad de sus operaciones, y la amenaza de ataques cibernéticos hace necesario contar con mecanismos efectivos de evaluación y mitigación de riesgos.

Mediante el análisis de la infraestructura existente y poniendo a prueba su resistencia ante posibles amenazas, así como el uso de herramientas especializadas, se realizaron pruebas de intrusión controladas que simulan ataques reales, permitiendo identificar vulnerabilidades en las aplicaciones web.

Los resultados permitieron presentar las vulnerabilidades presentes en las aplicaciones web de Grupo ALMAR, esta información sirve de guía para proponer soluciones y mejoras para fortalecer la seguridad, reduciendo los riesgos de ataques y asegurando la integridad y confidencialidad de la información.

Palabras claves: Pentesting, Aplicaciones web, Ciberseguridad**.**

ABSTRACT

The main objective of this work is the implementation of pentesting tests in ALMAR Group web applications, to identify vulnerabilities. For the companies that are part of the group, these types of applications are critical components, since they manage the information that allows the continuity of their operations, and the threat of cyber-attacks makes it necessary to have effective risk assessment and mitigation mechanisms.

By analyzing the existing infrastructure and testing its resistance to possible threats, as well as the use of specialized tools, controlled intrusion tests were carried out that simulate real attacks, allowing vulnerabilities in web applications to be identified.

The results allowed us to present the vulnerabilities present in Grupo ALMAR's web applications. This information serves as a guide to propose solutions and improvements to strengthen security, reducing the risks of attacks and ensuring the integrity and confidentiality of the information.

Keywords: Pentesting, Web applications, Cybersecurity.

INTRODUCCIÓN

En la actual era digital, las aplicaciones web forman parte de la infraestructura de negocios, gobiernos y organizaciones de todo ámbito, este tipo de aplicaciones maneja una gran cantidad de datos, operativos, personales, transaccionales financieros, etc. Muchas de estas aplicaciones web presentan vulnerabilidades frente a posibles ataques cibernéticos, esto puede deberse a malas prácticas de desarrollo, configuraciones inseguras o a la complejidad inherente en los entornos web.

Para identificar las vulnerabilidades de las aplicaciones web, una de las herramientas más efectivas es el pentesting (pruebas de penetración) que permite a los equipos de seguridad simular ataques en un entorno controlado para descubrir vulnerabilidades antes de que los atacantes reales puedan explotarlas.

La aplicación de pentesting en las empresas enfrenta varios desafíos, como la complejidad técnica de las tecnologías que pueden resultar difíciles de analizar y asegurar, adicionalmente las vulnerabilidades pueden evadir las pruebas automatizadas y son solamente identificables mediante técnicas manuales y conocimiento profundo de las tecnologías, además los recursos se convierten en una limitante cuando las organizaciones no están equipadas con la tecnología, personal y presupuesto necesario para realizar pentesting de manera exhaustiva y regular.

Planteamiento del problema

Grupo ALMAR mediante el uso de sus aplicaciones web realiza transacciones y permite la interacción con clientes internos, externos y proveedores, lo que genera una gran cantidad de datos.

Las vulnerabilidades en las aplicaciones web son un problema para considerar, debido a que expone a la organización ante posibles pérdidas financieras, daño a su reputación y sanciones legales, mediante la divulgación de su información, sin un enfoque proactivo para detectar y corregir estas debilidades, la seguridad de las aplicaciones web de Grupo ALMAR podría ser comprometida.

Actualmente no se ha implementado en las aplicaciones web de Grupo ALMAR una metodología de detección de vulnerabilidades que permita identificarlas y mitigarlas antes de que puedan ser explotadas por atacantes internos o externos. Este trabajo busca

abordar esta necesidad mediante el diseño y ejecución de pruebas de pentesting que contribuyan a mejorar la postura de seguridad de la organización.

Justificación

Las vulnerabilidades en las aplicaciones web de Grupo ALMAR pueden llevar a brechas de seguridad significativas, aumentando los riesgos como la filtración de datos, daños reputacionales y pérdidas financieras.

Este trabajo permitirá determinar las herramientas y técnicas de pentesting que se ajustan a las aplicaciones web dominio de Grupo ALMAR, para posteriormente con la implementación del pentesting poder determinar si existen, cuáles son y la gravedad de las vulnerabilidades encontradas.

Con este trabajo se busca determinar las vulnerabilidades existentes, también se tiene como objetivo realizar un informe y un plan de acción que permita mitigar el riesgo asociado.

Formulación del problema

En el contexto de la creciente complejidad y sofisticación de las amenazas cibernéticas: ¿Cómo puede Grupo ALMAR implementar metodologías sistemáticas y eficaces de pentesting para identificar y mitigar vulnerabilidades en sus aplicaciones web?

Objetivo General

Implementar técnicas de pentesting para identificación de vulnerabilidades en las aplicaciones web de Grupo ALMAR.

Objetivos Específicos

1. Investigar estado del arte sobre técnicas de pentesting en las aplicaciones web para conocer los resultados sobre su implementación.
2. Identificar vulnerabilidades de las aplicaciones web de Grupo ALMAR mediante la implementación de pentesting.
3. Analizar resultados y elaborar un informe de las técnicas de pentesting utilizadas en las aplicaciones web de Grupo ALMAR.

4. Elaborar una propuesta de seguridad para la mitigación de vulnerabilidades encontradas al implementar pentesting en las aplicaciones web de Grupo ALMAR.

Planteamiento hipotético

"La aplicación de pentesting como herramienta de hacking ético permite la detección de vulnerabilidades en aplicaciones web, proporcionando una evaluación de la seguridad y contribuyendo a la mitigación de riesgos cibernéticos en el catálogo de aplicaciones web de Grupo ALMAR".

Alcance y limitaciones

El presente trabajo tiene como objetivo principal implementar pruebas de pentesting en las aplicaciones web de Grupo ALMAR, con el fin de identificar y mitigar vulnerabilidades de seguridad.

El trabajo se realizará utilizando el catálogo de aplicaciones web propiedad de Grupo ALMAR en ambientes controlados, abarcará un periodo de 3 meses desde la revisión de la literatura y selección de herramientas hasta la implementación de pruebas y análisis de resultados, el estudio se centrará en las aplicaciones web desarrolladas en tecnologías de la suite Microsoft Punto Net (.Net).

El acceso a las aplicaciones web reales para pruebas estará limitado a aquellas para las que se obtenga el permiso por parte de la empresa, así como de aquellas que se logren implementar en el laboratorio y escenarios de pruebas controlado.

Existirá un acuerdo de confidencialidad entre el autor de este trabajo y el representante de Grupo ALMAR para precautelar que la información o vulnerabilidades encontradas no se hagan de dominio público, o a su vez sean utilizadas por cualquier persona para beneficio propio.

CAPÍTULO 1. MARCO TEÓRICO

1.1. Definiciones

Aplicación Web

Una aplicación web es un programa en dónde el usuario ingresa a un servidor web mediante un navegador web, entre otras definiciones podemos decir que las aplicaciones web son herramientas para acceder a servidores utilizando una red, generalmente Internet o una Intranet (Valarezo Pardo et al., 2018).

Las aplicaciones web actualmente son vulnerables ante un sin número de ciberataques, los delincuentes cibernéticos aprovechan las vulnerabilidades de estas para acceder, comprometer y modificar la integridad de los datos y la seguridad del sitio web, estas vulnerabilidades son muy comunes en el diseño, mantenimiento e implementación de los aplicativos. El objetivo de estos ciberataques son realizar actividades ilícitas, y acceder a datos sensibles (Valarezo Pardo et al., 2018).

Seguridad informática

La seguridad informática son aquellas metodologías, técnicas y practicas establecidas para proteger sistemas contra amenazas cibernéticas que podrían causar el acceso, divulgación no autorizada, modificación y destrucción de la información, en otras palabras, debemos salvaguardar los recursos de la infraestructura tecnológica de atacantes dispuestos a hacer un mal uso de los datos (Vega Briceño, 2021).

Es importante tomar en cuenta que la seguridad informática no solo abarca la protección de los sistemas (software), aplicaciones, sino también a los recursos físicos (hardware) tales como, centros de datos, equipos de conectividad, cableado, etc. (Vega Briceño, 2021).

Ciberseguridad

La ciberseguridad es una rama muy amplia de la seguridad informática cuyo propósito es la protección de los sistemas informáticos y tiene como principio garantizar la disponibilidad, integridad y confidencialidad de la información, dentro de sus normas establece buenas prácticas y procesos específicos para la protección de aplicativos, dispositivos e información en la nube (Candau & Marco, 2021).

Ciberataque

Un ciberataque es cualquier acción malintencionada llevada a cabo por individuos o grupos con el objetivo de comprometer, dañar, o interrumpir sistemas informáticos, redes, o datos, estos ataques pueden variar en complejidad y objetivos, desde el acceso no autorizado a información confidencial hasta la interrupción completa de servicios críticos. Los ciberataques suelen estar motivados por factores como el lucro económico, el espionaje, el activismo político (hacktivismo) o la simple intención de causar daño(Cano M., 2020).

Hacker

Un hacker es un individuo con habilidades avanzadas en informática y tecnología, que utiliza sus conocimientos para explorar, modificar o mejorar sistemas computacionales, dependiendo de sus intenciones, estos actores pueden actuar de manera ética, maliciosa o neutral (Scariot Esquivel, 2022).

La actividad del hacking no necesariamente implica ilegalidad, es la intención y el método lo que define su carácter, los tipos de hackers según sus intenciones y roles se pueden clasificar de la siguiente manera (Andrés Maíllo Fernández & -México, 2021):

- Hackers de sombrero blanco (white hat): También conocidos como hackers éticos, trabajan para identificar y corregir vulnerabilidades en sistemas con el consentimiento del propietario, su objetivo principal es proteger sistemas y mejorar su seguridad.
- Hackers de sombrero negro (black hat): Actúan con intenciones maliciosas, buscando explotar vulnerabilidades para obtener beneficios financieros, realizar espionaje o causar daño.
- Hackers de sombrero gris (gray hat): Operan entre la ética de los White y Black Hats, pueden explorar vulnerabilidades sin permiso, pero sin intención de causar daño, a menudo con el fin de alertar a las organizaciones afectadas.
- Hacktivistas: Utilizan sus habilidades para promover causas políticas o sociales, mediante ataques a sistemas, sitios web o la filtración de información.
- Script Kiddies: Usuarios con conocimientos básicos que emplean herramientas creadas por otros para ejecutar ataques, generalmente motivados por notoriedad o simple curiosidad.

- Hackers de sombrero Azul (blue hat): Contratados por organizaciones para realizar pruebas de penetración y evaluar sistemas en busca de vulnerabilidades.
- Hackers de sombrero verde (green hat): Hackers en formación que buscan aprender sobre ciberseguridad, con el potencial de seguir caminos éticos o maliciosos.

Pentester

Los pentesters, también conocidos como hackers éticos, son profesionales especializados en evaluar la seguridad de los sistemas informáticos, su trabajo consiste en llevar a cabo una serie de pruebas para detectar posibles vulnerabilidades. Con su experiencia en seguridad de la información, emplean sus habilidades para identificar y gestionar de manera efectiva las debilidades en la infraestructura de una empresa (Astudillo, 2021).

Atacante

Un atacante o ciberdelincuente es una persona que accede sin permiso a sistemas informáticos con el propósito de dañar la integridad, disponibilidad y accesibilidad de la información en un sitio web o en un dispositivo electrónico (Instituto Nacional de Ciberseguridad de España, 2021).

Escaneo de vulnerabilidades

La actividad de buscar vulnerabilidades en redes y sistemas implica el uso de diversas técnicas y herramientas especializadas para identificarlas y corregirlas, con el objetivo de prevenir que los ciberdelincuentes las exploten en su favor, este proceso de escaneo se enfoca en aplicaciones, puertos y servicios presentes en una empresa (Instituto Nacional de Ciberseguridad de España, 2021).

Análisis estático de código fuente

El análisis estático de código fuente es un proceso técnico que implica examinar el código de una aplicación sin ejecutarlo, utilizando herramientas o inspección manual para identificar vulnerabilidades, errores, o problemas de calidad (Larrea, 2021).

Este tipo de análisis se enfoca en evaluar la estructura del código, las dependencias, y las posibles fallas de seguridad para garantizar que cumpla con estándares de calidad y seguridad, generalmente, este método se emplea en las primeras

etapas del ciclo de desarrollo de software para minimizar riesgos antes del despliegue (Larrea, 2021).

Exploit

Un exploit es un código que se utiliza para aprovechar una vulnerabilidad de seguridad en un sistema con el objetivo de obtener ventajas indebidas. Se trata de un software, un conjunto de datos, que explota un defecto en una aplicación o sistema para generar un beneficio propio (Pastor Ricós, 2020).

Pentesting (pruebas de penetración)

El pentesting es un método que se utiliza para encontrar fallos de seguridad o vulnerabilidades en los sistemas informáticos, básicamente, se realiza una serie de simulaciones o pruebas en entornos controlados con el fin de encontrar alguna brecha de seguridad en el sistema. Esta técnica es sumamente útil para las empresas dado que está diseñada para determinar las consecuencias o efectos que podrían causar los posibles ataques cibernéticos, los hackers éticos utilizan técnicas y herramientas que están reguladas y autorizadas por las organizaciones para garantizar la seguridad y confiabilidad de los datos (Vanegas Romero, 2021).

Importancia del pentesting

Entre los aspectos más importantes sobre el pentesting se indican los siguientes:

- Mitiga los riegos de perdida de información.
- Evaluación del nivel de seguridad.
- Cumplimiento de normas de seguridad.
- Incorporación de medidas de prevención y controles de seguridad.
- Reducción del riesgo de ataques.
- Fortalecimiento de la conciencia de seguridad (De la Torre & De la Torre, 2021).

Objetivos del pentesting

Los objetivos del pentesting son diversos y van a depender de acuerdo con los sistemas que se evalúan y a las necesidades específicas de las instituciones que requieran esta metodología, por mencionar los principales se tiene:

- Identificar vulnerabilidades y riesgos potenciales

- Evaluación de controles de seguridad.
- Mejorar la respuesta ante incidentes.
- Evaluar la efectividad de las medidas de seguridad.
- Mejorar el plan de acción de las organizaciones ante posibles ciberataques.
- Integrar mejores prácticas y concientizar acerca de los riesgos (De la Torre & De la Torre, 2021).

Principios éticos del pentesting

Los principios éticos del pentesting se basan en normas que permitan garantizar la legalidad, la responsabilidad y la ética al realizar este procedimiento, algunos de estos principios son los siguientes:

- Confidencialidad: La información que surja como resultado de las pruebas de pentesting debe ser manejada con suma cautela y solamente entregarse a personal autorizado.
- Integridad: Los hackers éticos al realizar pentesting deben actuar con absoluta integridad y evitar cualquier manipulación indebida de los datos o cambios en el sistema.
- Consentimiento informado: Antes de realizar las pruebas es necesario contar con una autorización de parte de la alta dirección informar cuales son los pasos para llevar a cabo el análisis de vulnerabilidades.
- Profesionalismo: Los pentesters deben respetar las políticas y estatutos de la organización, actuar con mesura, mantener profesionalismo y acatar normas de conducta.
- Disponibilidad: Durante el pentesting, se deben mantener controles para evitar alguna limitación con el acceso a la información y evitar un impacto negativo en el funcionamiento de los sistemas.
- Conocimiento y competencia: El personal que realice el pentesting debe estar plenamente capacitado y contar con las habilidades requeridas para realizar la labor de manera segura y confiable (Felipe Redondo & Núñez Cárdenas, 2024).

Los principios éticos mencionados, son necesarios para garantizar un procedimiento responsable, tomando en cuenta las políticas de privacidad y seguridad de la institución, esto ayuda a garantizar la integridad de la información y a promover la

confianza basado en el profesionalismo del personal (Felipe Redondo & Núñez Cárdenas, 2024).

Vulnerabilidades en aplicaciones web

Las vulnerabilidades en aplicaciones web se pueden presentar en diferentes fases de los sistemas como implementación, diseño o configuración, y pueden ser aprovechadas por los atacantes para poner en riesgo la seguridad de la aplicación, entre las principales vulnerabilidades detectadas en las aplicaciones web se pueden mencionar la siguientes:

- Inyección SQL (SQL Injection): Es un proceso que se basa en insertar comandos SQL maliciosos en campos de entrada de datos, luego son ejecutados en la base de datos del sistema, esto causa que los datos queden expuestos e incluso que se modifiquen o eliminen.
- Secuencia de comandos entre sitios (Cross-Site Scripting XSS): Facilita que un atacante introduzca scripts dañinos en los sitios web que otros usuarios visitan, la función de estos scripts es alterar la interfaz de usuario, robar cookies o secuestrar sesiones.
- Falsificación de petición en sitios cruzados (Cross-site Request Forgery CSRF): Su función es engañar al usuario para que ejecute operaciones no esperadas en una aplicación web, lo cual puede ocasionar cambios no autorizados en la cuenta del usuario.
- Redireccionamientos y reenvíos no validados (Unvalidated Redidrects and Forwards): Facilita que un atacante conduzca a los usuarios hacia sitios web dañinos o lleve a cabo redirecciones no permitidas a través de la alteración de parámetros de redirección.
- Exposición de datos sensibles (Sensitive Data Exposure): Se refiere a la divulgación de información confidencial, como contraseñas, datos de tarjetas de crédito o información personal, como resultado de la ausencia de cifrado adecuado o prácticas inadecuadas en la gestión de datos. (Safla, 2021).

Metodologías del pentesting

Existen diversas metodologías de pentesting que los expertos en seguridad emplean para llevar a cabo pruebas de penetración de forma organizada y efectiva, entre ellas se puede mencionar las siguientes:

- Metodología de Pruebas de Penetración de Open Web Application Security Project (OWASP): Esta metodología se especializa en la realización de pruebas de penetración en aplicaciones web, sus etapas clave abarcan la identificación de vulnerabilidades, la recolección de información, la explotación de dichas vulnerabilidades y la generación de Informes.
- Metodología de Pruebas de Penetración Estándar (PTES): Esta metodología ofrece un marco completo para realizar pruebas de penetración en sistemas informáticos. La PTES establece siete etapas clave: Preparación Inicial, Modelado de Amenazas, Inteligencia, Recolección de Información, Explotación, Post-explotación y gestión de Informes.
- Metodología de Pruebas de Penetración de Red Interna y Externa: Esta metodología está orientada a analizar la seguridad de las redes externas e internas de una institución, las etapas comunes abarcan el Escaneo de Red, la Explotación de Vulnerabilidades, la Enumeración de Servicios, la Post-explotación y la Producción de Informes. (Safla, 2021).

Tipos de pentesting

El pentesting se clasifica según el conocimiento del sistema y el nivel de acceso concedido. Los tipos incluyen pruebas de caja blanca, caja negra y caja gris, así como pruebas externas e internas, cada una abordando la seguridad desde diferentes ángulos. A continuación, se detallan los tipos de Pentesting:

Pentesting según el conocimiento:

- Pruebas de penetración de caja negra: Son también conocidas como pruebas de caja cerrada, en este proceso, el evaluador no dispone de información previa sobre el objetivo, este enfoque es más laborioso en comparación con otros métodos debido a la falta de conocimiento y acceso a datos internos. Solo evaluadores experimentados pueden llevar a cabo este tipo de prueba, que se realiza con un número limitado de casos de prueba, por lo que el evaluador simula el papel de un hacker potencial.
- Pruebas de penetración de caja gris: En las pruebas de penetración de caja gris, el evaluador cuenta con información parcial y acceso limitado a datos internos, se proporciona al evaluador detalles básicos sobre el cliente, como direcciones IP,

nombres de host, correos electrónicos y URL. A diferencia de las pruebas de caja blanca, el evaluador actúa desde la perspectiva de un usuario normal en lugar de un administrador, este método suele ser más ágil que las pruebas de caja blanca.

- Pruebas de penetración de caja blanca: En estas pruebas, el evaluador tiene acceso completo a toda la información del cliente, incluidas las credenciales de inicio de sesión, gracias a este conocimiento detallado, el proceso es más eficiente, permitiendo realizar un mayor número de casos de prueba en menos tiempo. Los evaluadores que llevan a cabo estas pruebas deben contar con sólidas habilidades en programación y lógica por lo cual suele ser más rápido y económico en comparación con otros métodos de prueba (Kongara & Krishnama, 2023).

Pentesting según el acceso:

- Pentesting Interno: El pentesting interno se enfoca en revisar las medidas de seguridad dentro de una red, abarcando servidores y estaciones de trabajo, simula un ataque desde el interior de la red, como el de un dispositivo infectado, un empleado descontento o un cibercriminal con acceso interno, utilizando técnicas de ingeniería social o malware. Su propósito es evaluar la seguridad de la red interna y comprobar qué información y sistemas pueden ser accedidos sin permiso, destacando la necesidad de un análisis exhaustivo de vulnerabilidades.
- Pentesting Externo: El pentesting externo se dedica a analizar la seguridad de los sistemas que son accesibles desde Internet, tales como sitios web o infraestructuras en la nube, evalúa la seguridad de la infraestructura tecnológica desde la perspectiva de un atacante externo. La prueba simula un ataque dirigido a sistemas y aplicaciones expuestos a la red, como servidores web, cortafuegos y sistemas de correo electrónico, con el fin de descubrir y corregir vulnerabilidades en la frontera de seguridad antes de que puedan ser aprovechadas (Venza, 2023).

Herramientas del Pentesting

El pentesting emplea diversas herramientas y técnicas para descubrir vulnerabilidades y evaluar la seguridad redes, sistemas y aplicaciones, a continuación, se presentan algunas de las herramientas y métodos más habituales en el pentesting.

Herramientas para escaneo de Vulnerabilidades:

- Nmap: Es una herramienta de escaneo que facilita la detección de hosts y servicios en una red, además de identificar posibles vulnerabilidades.
- OWASP ZAP: Es una herramienta de código abierto creada para detectar vulnerabilidades en aplicaciones web, se usa principalmente para llevar a cabo pruebas de penetración en estas aplicaciones y evaluar su nivel de seguridad.
- Burp Suite: Es una suite de herramientas creada para la evaluación de la seguridad en aplicaciones web, que incorpora un proxy web, un escáner de vulnerabilidades y diversas utilidades complementarias. (Rodríguez Llerena, 2020).

Herramientas de explotación:

- Metasploit: Es una plataforma de pruebas de penetración que ofrece una extensa variedad de herramientas y exploits para comprometer aplicaciones y sistemas.
- ExploitDB: Es una base de datos de exploits públicos que permite localizar exploits específicos para vulnerabilidades ya identificadas.
- SQLMap: Es una herramienta dedicada a explotar vulnerabilidades de inyección SQL en aplicaciones web, que facilita la ejecución de ataques de manera automatizada.
- Hydra: Es una herramienta de fuerza bruta creada para examinar la robustez de los sistemas de autenticación en servicios y plataformas como FTP, SSH, HTTP y otros. (Rodríguez Llerena, 2020).

Herramientas de análisis de Tráfico:

- Wireshark: Es una herramienta de análisis de tráfico de red que captura y revisa paquetes de datos en la red, ayudando a detectar vulnerabilidades y comportamientos inusuales.
- Tcpdump: Es una herramienta de línea de comandos comparable a Wireshark, que facilita la captura y el análisis de paquetes de datos en tiempo real.
- Ettercap: Es una herramienta de análisis de tráfico desarrollada para interceptar y alterar comunicaciones en redes locales, facilitando la ejecución de ataques de tipo hombre en el medio (MITM). (Rodríguez Llerena, 2020).

Herramientas de análisis estático de código fuente:

Amazon Q Developer

Amazon Q Developer es un asistente conversacional impulsado por inteligencia artificial generativa, diseñado para ayudar a entender, desarrollar, expandir y gestionar aplicaciones de Amazon Q. Este asistente se actualiza de forma continua para ofrecer respuestas que sean siempre las más pertinentes y útiles en función del contexto (Amazon, 2024).

En un entorno de desarrollo integrado (IDE), Amazon Q ofrece soporte para el desarrollo de software al interactuar sobre el código, proporcionar sugerencias, generar código, escanear el código en busca de vulnerabilidades, realizar reajustes y mejoras (Amazon, 2024).

GitHub Copilot

GitHub Copilot es una herramienta creada en colaboración entre GitHub y OpenAI lanzada en junio de 2022, esta herramienta permite la generación de código de programación mediante palabras en lenguaje natural o comandos preestablecidos, en entornos de desarrollo integrado proporciona soluciones a errores de sintaxis, a advertencias de uso indebido del código, e incluso a vulnerabilidades identificadas (Huajie, 2023).

CAPÍTULO 2. METODOLOGÍA

2.1. Contexto del trabajo

El desarrollo de este trabajo se llevará a cabo en el campamento matriz de Grupo ALMAR, ubicado en la ciudad de Durán provincia de Guayas, en la vía Durán – Tambo Km. 8.

2.2. Diseño del trabajo

De acuerdo con el tipo de trabajo se empleará un enfoque descriptivo, experimental y cuantitativo.

2.2.1. Experimental

Se realizarán pruebas de penetración controladas en un entorno de desarrollo que simule aplicaciones web reales. Se aplicarán distintas técnicas y herramientas de pentesting, los resultados se medirán en términos de la cantidad y gravedad de vulnerabilidades detectadas.

2.2.2. Descriptivo

A través de una revisión de la literatura y análisis de casos de estudio, se documentarán las metodologías más utilizadas en pentesting y las vulnerabilidades frecuentes encontradas en aplicaciones web.

2.2.3. Cuantitativo

Medir y analizar los resultados de las pruebas de pentesting en términos de datos numéricos, como el número de vulnerabilidades encontradas, su gravedad, y el tiempo necesario para identificarlas.

Para este trabajo se recogerán y analizarán datos cuantitativos obtenidos durante el pentesting.

2.3. Fases del pentesting de caja blanca

Las fases de un pentesting de caja blanca son similares a las de otros enfoques, pero con un mayor grado de detalle debido al acceso completo a la información del sistema. Las fases principales son:

- Planificación y Reconocimiento: Se define el alcance y los objetivos del pentesting, además se recopila toda la información del sistema, como diagramas, configuraciones y código fuente.
- Análisis de Vulnerabilidades: Se revisa el código fuente y se identifican vulnerabilidades mediante herramientas automatizadas y manuales, también se evalúan las configuraciones de seguridad.
- Explotación: Se intentan explotar las vulnerabilidades descubiertas para evaluar el impacto y la posibilidad de acceso no autorizado o alteración del sistema.
- Post-Explotación: Se determina el nivel de acceso obtenido y el impacto potencial en la seguridad, también se evalúa la persistencia de un atacante en el sistema.
- Reporte y Documentación: Se genera un informe detallado con las vulnerabilidades, pruebas realizadas y recomendaciones de mitigación.
- Remediación y Retesting: Se implementan las correcciones y se vuelve a realizar pruebas para confirmar que las vulnerabilidades han sido resueltas.

Cada fase de un pentesting de caja blanca es más detallada y precisa debido al acceso privilegiado a la información interna, lo que permite detectar problemas que no serían visibles en un enfoque de caja negra o caja gris.

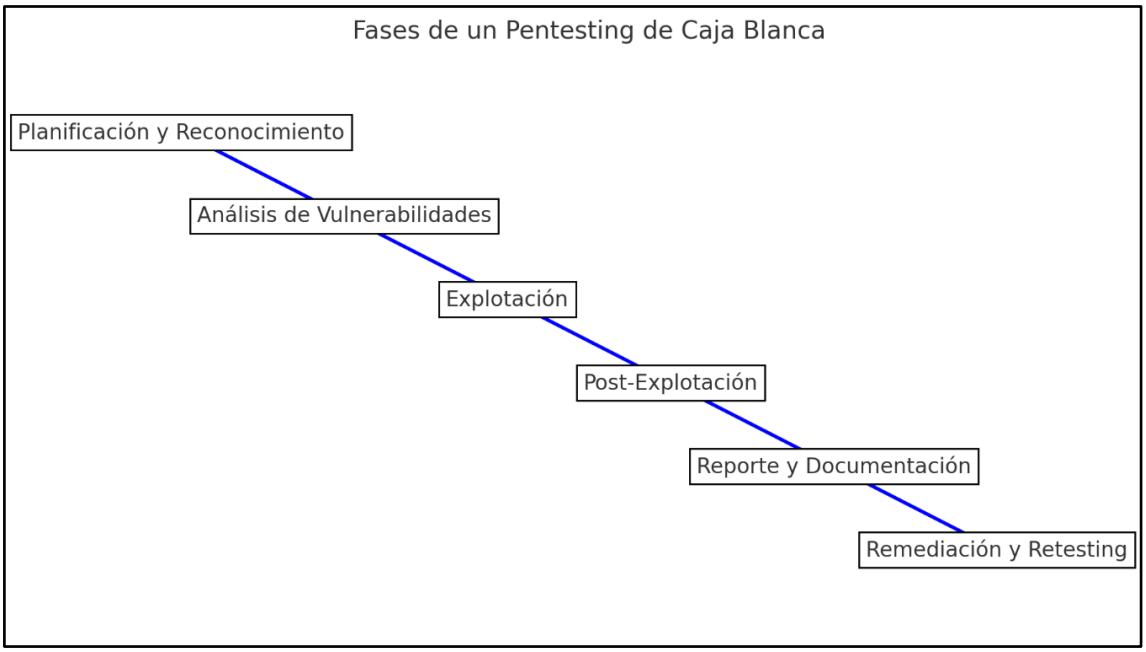


Ilustración 1 Fases de un Pentesting de Caja Blanca

Fuente: Elaboración propia

2.3.1. Planificación y Reconocimiento

En esta fase se establecerán los fundamentos del proceso de pentesting, lo cual incluye:

- Objetivos: Se delimitará el alcance y los objetivos del pentesting, especificando los sistemas que serán evaluados. Además, se definirán reglas claras para el desarrollo de las pruebas con el fin de respetar los límites y garantizar la seguridad de las operaciones.
- Recolección de información: Se procederá a reunir toda la información proporcionada por el cliente (Grupo ALMAR), esto incluye documentación técnica relevante, como diagramas de infraestructura de las aplicaciones web, configuraciones del sistema, código fuente, y cualquier otra información que pueda ser útil para comprender la infraestructura y sus vulnerabilidades potenciales.
- Identificación de áreas críticas: Con base en la información recopilada, se evaluarán las partes del sistema más propensas a sufrir ataques debido a vulnerabilidades conocidas, configuraciones incorrectas o puntos débiles en la seguridad.

Implementación en Grupo ALMAR: La planificación y el reconocimiento abarcarán las secciones 2.4, 2.5 y 2.6 de la infraestructura tecnológica.

2.3.2. Análisis de Vulnerabilidades

Durante esta fase, se llevarán a cabo las siguientes actividades clave:

- Revisión de código fuente: Dado que el pentester tiene acceso al código, se realizará una auditoría para identificar fallos comunes, como inyecciones SQL, desbordamientos de búfer, uso de malas prácticas de seguridad, y otras vulnerabilidades críticas.
- Escaneo de vulnerabilidades: Se emplearán tanto herramientas automatizadas como técnicas manuales para identificar posibles debilidades en el sistema, este análisis se centrará en detectar software desactualizado, configuraciones incorrectas y otras brechas que puedan ser explotadas.

Implementación en Grupo ALMAR: En esta fase, se utilizaron herramientas para realizar la revisión de código fuente, como Amazon Q y Github Copilot, y se hicieron análisis de vulnerabilidades con herramientas especializadas como Nessus y OWASP ZAP.

El código fuente de las aplicaciones web que son objetivo del pentesting fue escaneado bajo el análisis pasivo de vulnerabilidades con la herramienta AmazonQ, esta herramienta se instaló en su versión de prueba para el IDE (Integrated Development Environment) Visual Studio 2022.

Para utilizar AmazonQ primero se instaló la extensión de AWS Toolkit en Visual Studio, se configuró un usuario gratuito, una vez ubicado en alguna sección de código de programación se inició un escaneo de la solución dentro del ámbito y este arrojó el resultado de las vulnerabilidades encontradas en la ventana de salida propia de la herramienta.

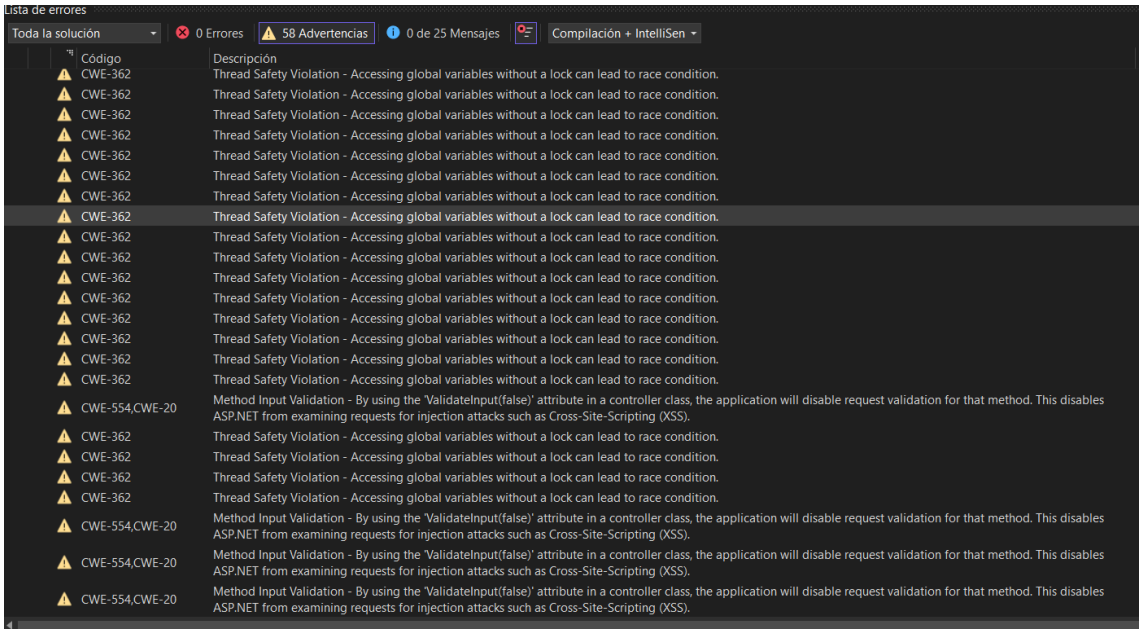


Ilustración 2 Ventana de salida de AmazonQ con las vulnerabilidades encontradas en análisis de código fuente

Este proceso fue repetido para las aplicaciones web que son objetivo de pentesting en este trabajo de investigación.

Las vulnerabilidades encontradas mediante el escaneo pasivo de código fuente con la herramienta AmazonQ, fueron analizadas con GitHub Copilot, esta herramienta forma parte del catálogo de soluciones adquiridas por parte de Grupo ALMAR y está a disposición de su personal de desarrollo, por lo que se pudo utilizar con las credenciales

propiciadas, y con esto la herramienta generó las sugerencias de código para solventar las vulnerabilidades encontradas.

2.3.3. Explotación

La fase de explotación se centrará en probar si las vulnerabilidades descubiertas pueden ser aprovechadas para comprometer el sistema, sin embargo, en el contexto del Grupo ALMAR esta fase estuvo limitada por las restricciones impuestas por la empresa y el tiempo disponible.

- Pruebas activas: El pentester intentará explotar las vulnerabilidades identificadas, lo que incluye realizar pruebas de acceso no autorizado, escalada de privilegios y acceso a datos sensibles, para medir el impacto real de dichas vulnerabilidades.
- Pruebas en tiempo real: Aprovechando el acceso completo al sistema, se simularán ataques en un entorno controlado para validar la gravedad de las vulnerabilidades y el nivel de acceso que un atacante podría obtener.

Implementación en el grupo ALMAR: Se realizaron pruebas mediante escaneos activos utilizando Nessus, OWASP ZAP, con el objetivo de detectar vulnerabilidades explotables.

2.3.3.1. OWASP ZAP

El pentesting realizado con la herramienta OWASP ZAP se hizo con las siguientes configuraciones:

- Se definió un contexto para cada aplicación web a probar (un contexto es un ámbito donde se establecen configuraciones de usuario, URLs incluidas o excluidas, estructura, tecnología, modo y credenciales de autenticación, etc.)
- En cada contexto se definió la URL de inicio de sesión (debido a que la página principal de las aplicaciones web es de este tipo) y las credenciales a utilizar para acceder a la página base donde se encuentra el menú con el resto de URLs.
- En el apartado de las tecnologías del contexto se marcaron las utilizadas por las aplicaciones web que forman parte del pentesting, Microsoft SQL Server para base de datos, ASP como lenguaje de programación, Windows para el sistema operativo y como servidor web Internet Information Services (IIS).

- Para las credenciales de usuario a utilizar en el pentesting se solicitó (a Grupo ALMAR) un usuario con todos los permisos en cada aplicación web, esto debido a que el tipo de pentesting definido para el trabajo de investigación es de caja blanca.
- Los análisis automáticos ejecutados para cada aplicación web son:
    - Spider, se utiliza para descubrir nuevas URLs en base a una semilla (una lista de enlaces predefinidos), posteriormente se visita cada hipervínculo buscando más hipervínculos, el proceso es recursivo siempre que se encuentren nuevos recursos.
    - Spider AJAX, integra un rastreador de sitios ricos en AJAX (Crawljax) para identificar páginas, usualmente se combina con el análisis Spider (normal) para mejores resultados.
    - Active Scan, o escaneo activo, permite identificar vulnerabilidades mediante la interacción con algoritmos como XSS en las URLs encontradas previamente con los análisis Spider.
- Las alertas encontradas para cada tipo de escaneo y aplicación web fueron recopiladas en un informe generado desde OWASP ZAP, mismo que se adjunta en anexos.

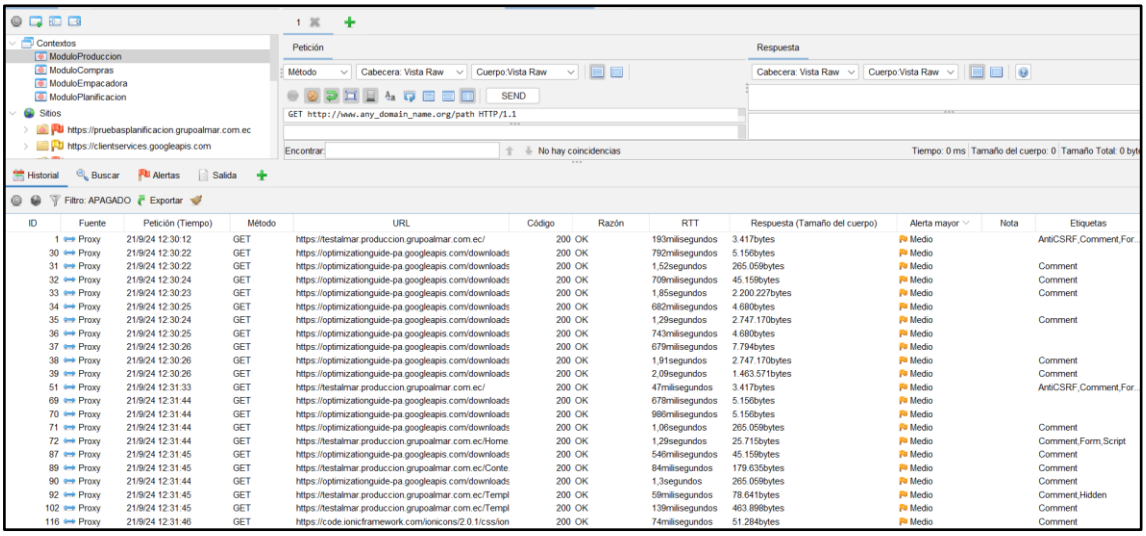


Ilustración 3 Contextos de las aplicaciones web y resultado de alertas en escaneo de vulnerabilidades – OWASP ZAP

2.3.3.2. Nessus

El pentesting realizado con la herramienta Nessus (en su versión profesional de prueba) se orientó a los servidores de aplicaciones y de base de datos que alojan las

aplicaciones web y las bases de datos (respectivamente), para ello se realizaron los siguientes escaneos de vulnerabilidades:

- Web Application Scan, permite escanear las vulnerabilidades en aplicaciones web publicadas y desconocidas, para ello se definieron los dos hosts (antes mencionados) como objetivos, no se utilizaron credenciales y las configuraciones fueron las predeterminadas por la plantilla de escaneo.

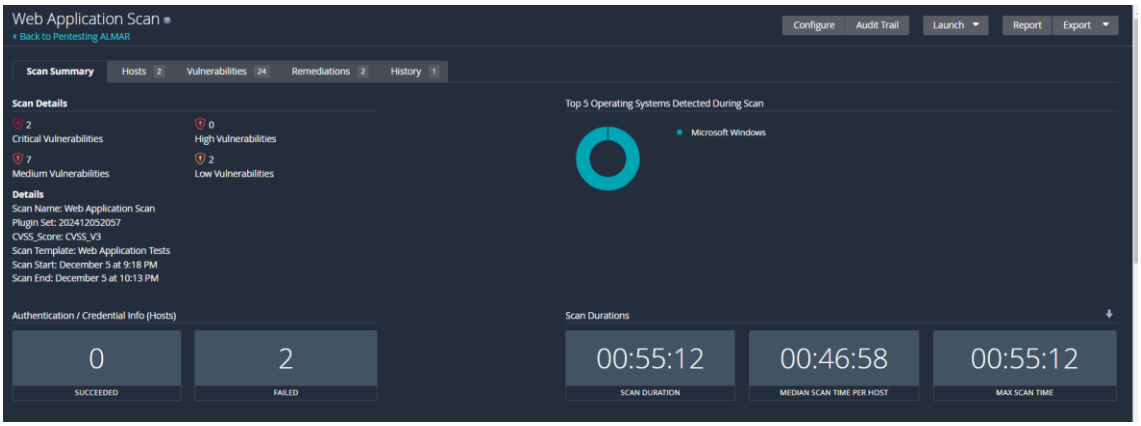


Ilustración 4 Dashboard resultado web application scan a los hosts de aplicaciones y base de datos

- Basic Network Scan, realiza un análisis completo del sistema adecuado para cada host, se definieron los dos hosts (antes mencionados) como objetivos, no se agregaron credenciales y las configuraciones utilizadas son las propias de la plantilla.

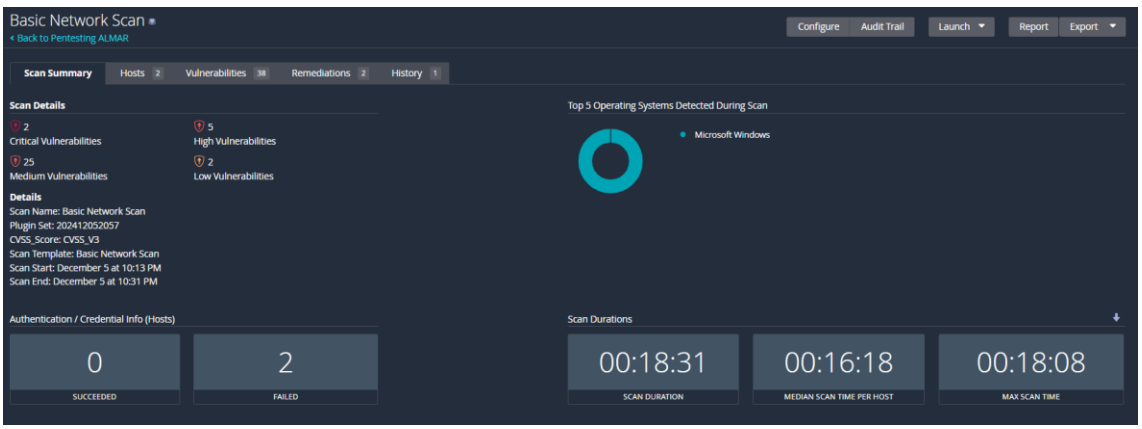


Ilustración 5 Dashboard resultado basic network scan a los hosts de aplicaciones y base de datos

- Advanced Dynamic Scan, permite definir escaneos o políticas con filtros de complementos dinámicos, para este trabajo se utilizó la plantilla definida por la herramienta, y se establecieron las URL's objetivos (los hosts antes mencionados), y para los plugins dinámicos se definieron las coincidencias con el

parámetro hostname y para los valores los mismos hosts que los definidos en las URL objetivo.

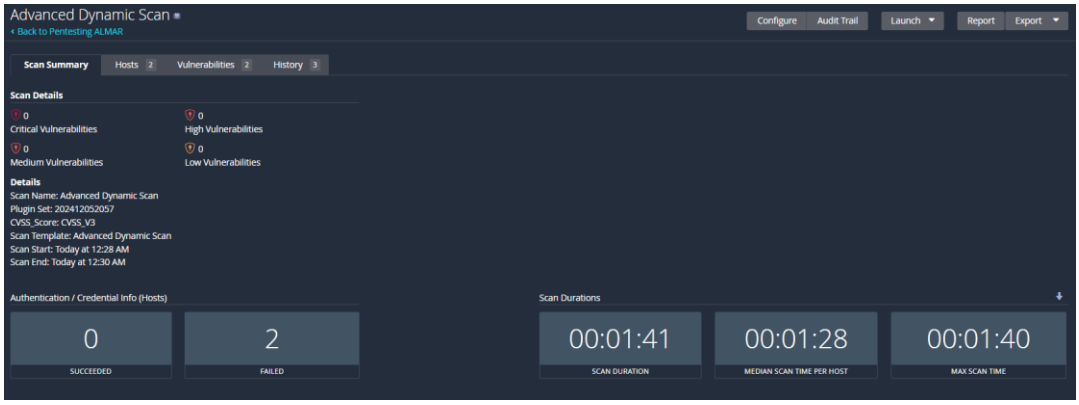


Ilustración 6 Dashboard resultado advanced dynamic scan a los hosts de aplicaciones y base de datos

2.3.4. Post-Explotación

Una vez explotadas las vulnerabilidades, el siguiente paso es evaluar el impacto que estos fallos de seguridad podrían tener sobre la organización:

Evaluación de impacto: El pentester determinará el nivel de acceso que un atacante podría obtener y evaluará las posibles consecuencias, tales como la fuga de datos, la interrupción de los servicios o la alteración de la integridad del sistema.

Implementación Grupo ALMAR: El impacto fue evaluado a partir de los reportes generados por las herramientas de análisis de vulnerabilidades, los cuales muestran los posibles efectos que podrían tener en la infraestructura tecnológica del grupo.

2.3.5. Reporte y Documentación

Una parte esencial del pentesting es documentar de forma clara y precisa todos los hallazgos obtenidos durante las fases anteriores:

- Informe detallado: El pentester elaborará un informe completo que incluirá todas las vulnerabilidades identificadas, las pruebas realizadas y las técnicas empleadas para explotarlas. Este informe también contendrá recomendaciones claras sobre cómo mitigar cada vulnerabilidad detectada.
- Evaluación de riesgos: Se proporcionará un análisis detallado del riesgo asociado con cada vulnerabilidad, considerando tanto el impacto como la probabilidad de que estas sean explotadas.

- Reunión de presentación: Finalmente, se realizará una presentación ante el equipo técnico y la gerencia de la organización para detallar los hallazgos y discutir los pasos a seguir para mejorar la seguridad del sistema.

Implementación en Grupo ALMAR: Se elaborará un informe detallado de las vulnerabilidades encontradas y se presentará al equipo de ciberseguridad de Grupo ALMAR.

2.3.6. Remediación y Retesting

Después de identificar las vulnerabilidades, la empresa tomará medidas para corregirlas. La fase final implica verificar que las soluciones aplicadas sean efectivas:

- Implementación de soluciones: El equipo técnico de la organización se encargará de aplicar las correcciones necesarias para mitigar las vulnerabilidades descubiertas.
- Reevaluación: Tras implementar las soluciones, se llevará a cabo una nueva ronda de pruebas para confirmar que las vulnerabilidades han sido eliminadas y que no han surgido nuevas debilidades.

Implementación en Grupo ALMAR: Esta fase se ejecutará tras la presentación del informe, y no está contemplada dentro del alcance del presente trabajo.

2.4. Población y muestra

La población para este trabajo está conformada por las aplicaciones web que formen parte del catálogo de Grupo ALMAR, indistintamente del sector al que pertenezcan, que además contengan características de acceso mediante clientes web.

Se seleccionarán las aplicaciones web a las que se tenga acceso autorizado para realizar pentesting sin violar normativas legales o éticas.

El tamaño de la muestra dependerá de las aplicaciones web que cumplan con las premisas antes mencionadas y además de la capacidad del autor para realizar el pentesting en el tiempo que se establece para el trabajo planteado.

2.5. Descripción de la empresa

Grupo ALMAR es un grupo de empresas acuícolas dedicadas a la larvicultura y producción de camarón.

Iniciando sus actividades en 1981 con la fundación de la primera de sus empresas (Produmar), desde entonces ha venido incorporando más empresas formando un grupo corporativo, que le ha permitido cerrar el ciclo de acuacultura uniéndose en un solo norte, "Acuacultura sostenible y produciendo el mejor camarón del mundo".

Grupo ALMAR se encuentra estructurado por departamentos que se encargan de la administración de todas sus operaciones, específicamente el departamento de tecnologías de la información (TI), se subdivide en tres subdepartamentos, infraestructura, soporte a usuarios y desarrollo. El equipo de desarrollo es el encargado de la definición, diseño, desarrollo, implementación y mantenimiento de los aplicativos que se utilizan para el almacenamiento de la información productiva, de abastecimiento, administración y análisis de todo Grupo ALMAR.

Es importante mencionar que todos estos aplicativos son accedidos solo mediante intranet, medida adoptada para brindar mayor seguridad.

Entre los aplicativos más relevantes se pueden mencionar los siguientes:

Aplicativo	Autoría	Descripción
Módulo Producción	Propio	Aplicación web que se utiliza para gestionar la información de la producción, costos y facturación de Grupo ALMAR.
App Producción	Propio	Aplicación móvil que se utiliza para gestionar la información de la producción de Grupo ALMAR.
Módulo Compras	Propio	Aplicación web que se utiliza para gestionar la información de abastecimiento y compras de Grupo ALMAR.
Módulo de RRHH	Propio	Aplicación web que se utiliza para gestionar la información del departamento de recursos humanos de Grupo ALMAR.
Sistema de Mantenimiento	Externo	Aplicación de escritorio que se utiliza para gestionar el mantenimiento de los equipos y maquinarias de Grupo ALMAR.
Módulo de Planificación	Propio	Aplicación web que se utiliza para la planificación y análisis de información de producción y compras de Grupo ALMAR.
Sistema de Empacadora	Propio	Aplicación web que se utiliza para la producción, planificación, ventas y exportación de la empacadora de Grupo ALMAR.
Módulo de activo fijo	Externo	Aplicación web que se utiliza para gestionar la información de los activos fijos de Grupo ALMAR.
SAP	Externo	Aplicación de escritorio que se utiliza para unificar los procesos empresariales y analizar los datos en tiempo real de Grupo ALMAR.

Tabla 1 Catálogo de aplicaciones de Grupo ALMAR

2.6. Selección de aplicaciones web a evaluar

Dentro del catálogo de aplicaciones de Grupo ALMAR se seleccionan para la aplicación de pentesting aquellas que coincidan con las premisas establecidas anteriormente:

Aplicativo	Autoría	Tipo	Tecnologías	Objetivo pentesting
Módulo Producción	Propio	Aplicación Web	• Microsoft .NET • Microsoft SQL Server	✅
App Producción	Propio	Aplicación Móvil	• Microsoft .NET • Microsoft SQL Server	❌
Módulo Compras	Propio	Sitio Web	• Microsoft .NET • Microsoft SQL Server	✅
Módulo de RRHH	Propio	Aplicación Web	• Microsoft .NET • Microsoft SQL Server	❌
Sistema de Mantenimiento	Externo	Aplicación Escritorio	• SMPROG	❌
Módulo de Planificación	Propio	Aplicación Web	• Microsoft .NET • Microsoft SQL Server	✅
Sistema de Empacadora	Propio	Aplicación Web	• Microsoft .NET • Microsoft SQL Server	✅
Módulo de activo fijo	Externo	Aplicación Web	• Cayman	❌
SAP	Externo	Aplicación Escritorio	• SAP S/4HANA	❌

Tabla 2 Catálogo de aplicaciones de Grupo ALMAR y sus características técnicas.

Las aplicaciones web que serán el objetivo del pentesting comparten una arquitectura y despliegue similar, en la ilustración 7 se puede apreciar lo siguiente:

- Los usuarios que acceden a los aplicativos son colaboradores de Grupo ALMAR.

- El acceso a las aplicaciones web se realiza a través de un equipo conectado mediante una VPN (Virtual Private Network), y es controlado por un firewall.
- Las peticiones a las aplicaciones web son resueltas por un servidor de aplicaciones que funcionan con el sistema operativo Windows Server.
- Las aplicaciones web están construidas con la suite de desarrollo de PuntoNet y se utiliza DevExtreme embebido en JavaScript como framework.
- Las aplicaciones web generan información que se almacena en bases de datos gestionadas mediante el sistema de gestión de bases de datos (DBMS) Microsoft SQL Server. Este DBMS está instalado en un servidor dedicado, el cual opera bajo el sistema operativo Windows Server.

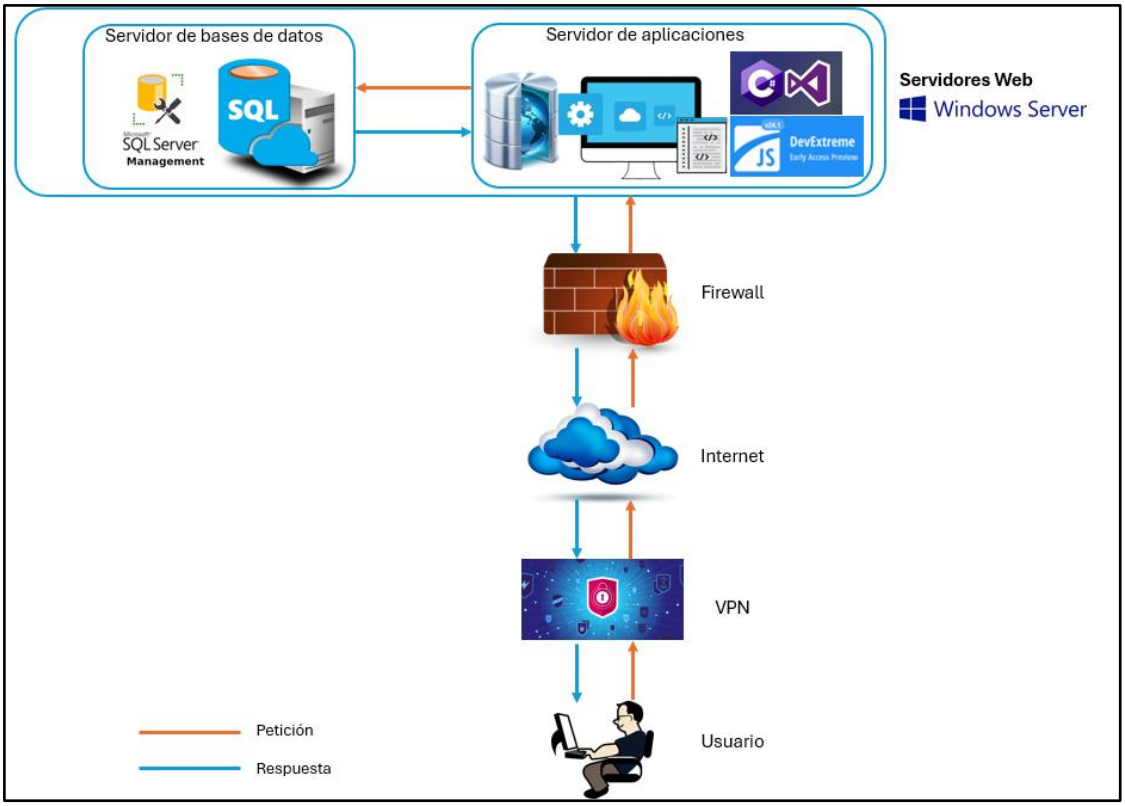


Ilustración 7 Infraestructura de las aplicaciones web de Grupo ALMAR

Fuente: Elaboración propia

CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

3.1. Análisis de código fuente - AmazonQ

En la ilustración 8, se observa la cantidad de vulnerabilidades encontradas con el análisis estático de código fuente para cada aplicación web, la aplicación que más vulnerabilidades tiene es la de Producción con un total de 150, luego le sigue la de Compras con una cantidad de 120, en tercer lugar, la de Empacadora con un total de 80, y finalmente la de Planificación con 40 vulnerabilidades.

En ese mismo orden (de cantidad de vulnerabilidades) las aplicaciones se sitúan de mayor a menor tamaño, entonces existe un patrón de acuerdo con el tamaño de cada aplicación, siendo así la aplicación de mayor tamaño (Producción) la que más vulnerabilidades tiene.

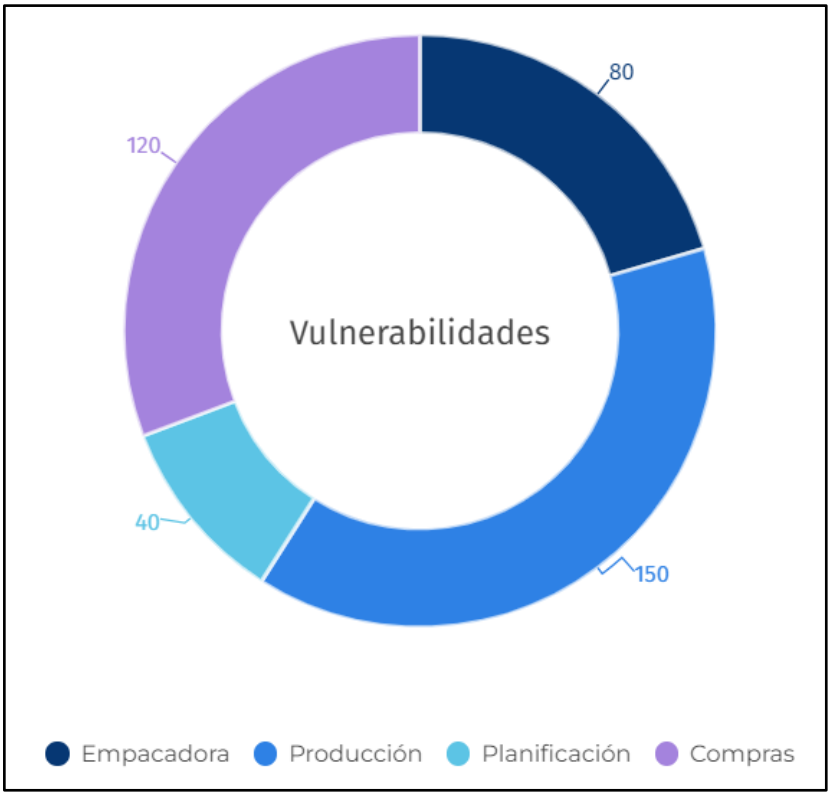


Ilustración 8 Cantidad de vulnerabilidades por aplicación web encontradas en el análisis de código fuente.

Fuente: Elaboración propia

En la tabla 3, se listan las distintas vulnerabilidades encontradas con el análisis de código fuente, el código que las identifica, el nombre con que se conoce y la solución para corregirlas (la solución fue sugerida por la herramienta GitHub Copilot).

Código	Nombre	Solución
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	Mediante función reemplazar las comillas simples con dos comillas simples, así se neutraliza la entrada maliciosa
CWE-554	The ASP.NET application does not use an input validation framework, vulnerable to injection attacks such as Cross-Site Scripting (XSS)	1. Eliminar atributo ValidateInput(false) para volver a habilitar la validación de solicitudes. 2. Agregar atributos [HttpPost] y [ValidateAntiForgeryToken] para garantizar que los métodos estén protegidos contra ataques de falsificación de solicitudes entre sitios (CSRF).
CWE-362	Thread Safety Violation - Accessing global variables without a lock can lead to race condition.	Agregar una declaración de bloqueo alrededor de la sección crítica donde se accede a Constantes. Usuario. Esto garantiza que solo un subproceso pueda ejecutar esa sección de código a la vez, evitando condiciones de carrera.
CWE-20	Improper Input Validation	1. Eliminar atributo ValidateInput(false) para volver a habilitar la validación de solicitudes. 2. Agregar atributos [HttpPost] y [ValidateAntiForgeryToken] para garantizar que los métodos estén protegidos contra ataques de falsificación de solicitudes entre sitios (CSRF).

Tabla 3 Vulnerabilidades encontradas mediante análisis de código fuente y su respectiva solución.

A continuación, se listan las distintas vulnerabilidades encontradas por cada aplicación web, así como la cantidad de veces que ocurren:

- Producción, CWE-89 (20 ocurrencias), CWE-554 (80 ocurrencias) y CWE-20 (50 ocurrencias).
- Compras, CWE-89 (13 ocurrencias), CWE-554 (65 ocurrencias) y CWE-20 (42 ocurrencias).

- Empacadora, CWE-362 (40 ocurrencias), CWE-554 (25 ocurrencias) y CWE-20 (15 ocurrencias).
- Planificación, CWE-89 (18 ocurrencias), CWE-362 (13 ocurrencias) y CWE-20 (9 ocurrencias).

3.2. Análisis de vulnerabilidades dinámico - Nessus

En la ilustración 9, se visualiza la cantidad de vulnerabilidades encontradas en el servidor de aplicaciones mediante el análisis dinámico de la herramienta Nessus, así como su clasificación según el nivel de riesgo.

Se puede evidenciar que existen vulnerabilidades críticas (6), no existen de riesgo alto, para las de riesgo medio existe una cantidad de 22, las de riesgo bajo son 6, y finalmente las informativas ocupan la mayoría (66), este resultado refleja que si bien existen vulnerabilidades críticas (deben tratarse con importancia), la mayoría son informativas.

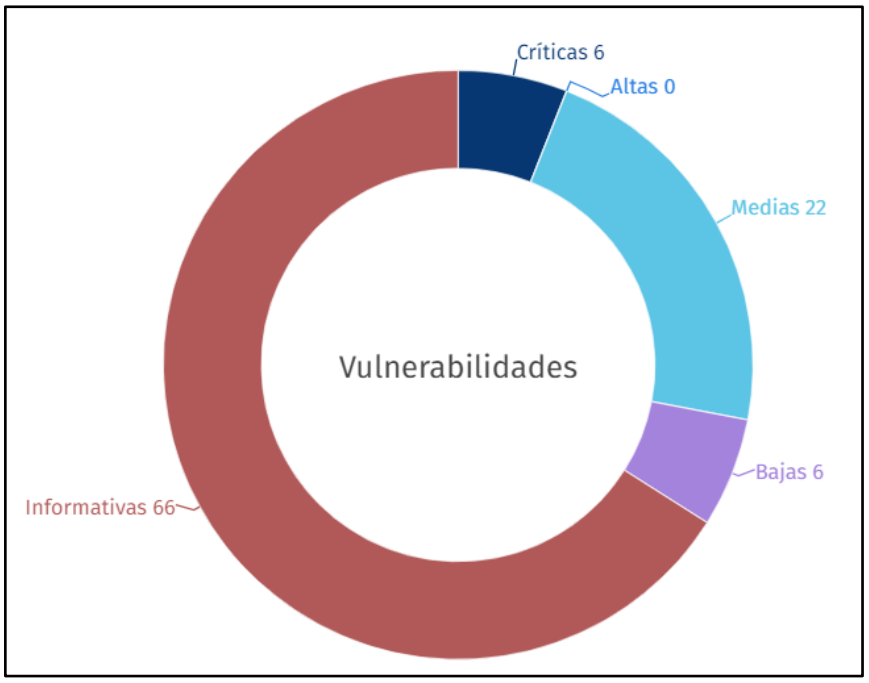


Ilustración 9 Cantidad de vulnerabilidades detectadas en el servidor de aplicaciones

Fuente: Elaboración propia

Luego de observar (ilustración 9) la cantidad de vulnerabilidades encontradas en el servidor de aplicaciones, en la tabla 4, se listan las vulnerabilidades críticas y medias

(no existen altas), incluyendo información como su nombre, la descripción y la solución para corregir la vulnerabilidad.

Severidad	Nombre	Descripción	Solución
Crítica	PHP Unsupported Version Detection	Según su versión, ya no se admite la instalación de PHP en el host remoto. La falta de soporte implica que el proveedor no lanzará nuevos parches de seguridad para el producto. Como resultado, es probable que contenga vulnerabilidades de seguridad.	Actualizar a una versión de PHP que sea compatible actualmente.
Crítica	PHP 8.0.x < 8.0.30 Multiple Vulnerabilities	La versión de PHP instalada en el host remoto es anterior a la 8.0.30. Por lo tanto, se ve afectado por múltiples vulnerabilidades como se menciona en el aviso de la versión 8.0.30. - En la versión PHP 8.0.* anterior a 8.0.30, 8.1.* anterior a 8.1.22 y 8.2.* anterior a 8.2.8, al cargar el archivo phar, mientras se leen las entradas del directorio	Actualizar a la versión PHP 8.0.30 o posterior.

		PHAR, una comprobación de longitud insuficiente puede provocar un desbordamiento de búfer de pila, lo que podría provocar daños en la memoria o RCE. (CVE-2023-3824)	
Media	Web Application Potentially Vulnerable to Clickjacking	El servidor web remoto no establece un encabezado de respuesta X-Frame-Options o un encabezado de respuesta 'frame-ancestors' de Content-Security-Policy en todas las respuestas de contenido. Esto podría exponer potencialmente el sitio a un ataque de secuestro de clics o de reparación de la interfaz de usuario, en el que un atacante puede engañar a un usuario para que haga clic en un área de la página vulnerable que es diferente de lo que el usuario percibe que es la página. Esto puede resultar en que un	Devolver el encabezado HTTP X-Frame-Options o Content-Security-Policy (con la directiva 'frame-ancestors') con la respuesta de la página. Esto evita que otro sitio muestre el contenido de la página cuando se utilizan las etiquetas HTML frame o iframe.

		usuario realice transacciones fraudulentas o maliciosas.	
Media	CGI Generic XSS (quick test)	El servidor web remoto aloja scripts CGI que no pueden desinfectar adecuadamente las cadenas de solicitud con JavaScript malicioso. Al aprovechar este problema, un atacante puede lograr que se ejecute código HTML y script arbitrario en el navegador de un usuario dentro del contexto de seguridad del sitio afectado. Es probable que estos XSS sean "no persistentes" o "reflejados".	Restringir el acceso a la aplicación vulnerable hasta desarrollar un parche o una actualización que solucione las vulnerabilidades de secuencias de comandos entre sitios.
Media	CGI Generic Cookie Injection Scripting	El servidor web remoto aloja al menos un script CGI que no puede desinfectar adecuadamente las cadenas de solicitud con JavaScript malicioso. Al aprovechar este	Restringir el acceso a la aplicación vulnerable hasta desarrollar un parche o una actualización que solucione las vulnerabilidades de secuencias de comandos entre sitios.

		problema, un atacante puede inyectar cookies arbitrarias. Según la estructura de la aplicación web, puede ser posible lanzar un ataque de "fijación de sesión" utilizando este mecanismo.	
Media	HSTS Missing from HTTPS Server (RFC 6797)	El servidor web remoto no aplica HSTS, según lo definido por RFC 6797. HSTS es un encabezado de respuesta opcional que se puede configurar en el servidor para indicarle al navegador que solo se comunique a través de HTTPS. La falta de HSTS permite ataques de degradación, ataques de intermediarios que eliminan SSL y debilita las protecciones contra el secuestro de cookies.}	Configurar el servidor web remoto para utilizar HSTS.

Tabla 4 Vulnerabilidades encontradas en el servidor de aplicaciones.

3.3. Análisis de vulnerabilidades dinámico – OWASP ZAP

Para la distribución y clasificación de las vulnerabilidades encontradas se utilizaron las tablas en formato matriz de riesgo de la herramienta OWASP ZAP.

En la tabla 5, se visualiza el total de las distintas vulnerabilidades para todas las aplicaciones, así como su clasificación según el nivel de riesgo y de confianza, se

evidencia que la mayoría de las vulnerabilidades se agrupan en un nivel de confianza medio y un riesgo bajo.

		Confianza			
		Alta	Media	Baja	**Total**
Riesgo	Alto	0 (0,0 %)	3 (9,1 %)	1 (3,0 %)	4 (12,1 %)
	Medio	1 (3,0 %)	4 (12,1 %)	1 (3,0 %)	6 (18,2 %)
	Bajo	2 (6,1 %)	10 (30,3 %)	1 (3,0 %)	13 (39,4 %)
	Informativo	0 (0,0 %)	4 (12,1 %)	6 (18,2 %)	10 (30,3 %)
	Total	3 (9,1 %)	21 (63,6 %)	9 (27,3 %)	33 (100,0 %)

Tabla 5 Matriz de riesgo - número de alertas por cada nivel de confianza y riesgo.

Nota: Los porcentajes entre paréntesis representan el número de alertas como porcentaje del total de alertas y redondeadas a un decimal.

A continuación, en la tabla 6 se observan las distintas vulnerabilidades por cada aplicación web, así como su clasificación según el nivel de riesgo, como se mencionó anteriormente, la aplicación de Producción es la que más vulnerabilidades posee teniendo estas agrupadas en su mayoría en un nivel de riesgo bajo.

Con esta herramienta de análisis dinámico se encontraron muchas menos vulnerabilidades que con la de análisis estático de código fuente.

		Riesgo			
		Alta (= Alto)	Media (>= Medio)	Baja (>= Bajo)	Informativo (>= Informativo)
Sitio	Planificación	2 (2)	0 (2)	1 (3)	1 (4)
	Empacadora	1 (1)	0 (1)	2 (3)	0 (3)
	Compras	0 (0)	0 (0)	1 (1)	0 (0)
	Producción	1 (1)	4 (5)	7 (12)	9 (21)
	Total	3 (9,1 %)	21 (63,6 %)	9 (27,3 %)	33 (100,0 %)

Tabla 6 Matriz de riesgo y la cantidad de alertas generadas para cada aplicación web en cada nivel de riesgo.

Nota: Los valores entre paréntesis son el número de alertas generadas para el sitio en ese nivel de riesgo o superior.

En la tabla 7 se listan las distintas vulnerabilidades encontradas en las aplicaciones web, el nivel de riesgo y la cantidad de ocurrencias para cada una de ellas, así como también el porcentaje que ocupa en función del total de vulnerabilidades.

Estos resultados reflejan que existe una cantidad de 22 vulnerabilidades altas en las aplicaciones web, existen 1.872 vulnerabilidades de riesgo medio, para las de rango bajo existen un total de 2.451 y finalmente las informativas suman una cantidad de 3.709.

En total el escaneo dinámico de la herramienta OWASP ZAP determina que existen 8.054 vulnerabilidades para las aplicaciones web.

Tipo de alerta	Riesgo	Cantidad
Inyección SQL	Alto	9 (27,3 %)
Inyección SQL - MsSQL	Alto	2 (6,1 %)
Inyección SQL - Oracle - Time Based	Alto	1 (3,0 %)
Ruta Transversal	Alto	10 (30,3 %)
Ausencia de Tokens Anti-CSRF	Medio	137 (415,2 %)
Cabecera Content Security Policy (CSP) no configurada	Medio	1085 (3.287,9 %)
Configuración Incorrecta Cross-Domain	Medio	378 (1.145,5 %)
Falta de cabecera Anti-Clickjacking	Medio	189 (572,7 %)
Inyección XSLT	Medio	60 (181,8 %)
Librería JS Vulnerable	Medio	23 (69,7 %)
Cookie Sin Flag de Seguridad	Bajo	23 (69,7 %)
Cookie con el atributo SameSite a None	Bajo	2 (6,1 %)
Cookie sin el atributo SameSite	Bajo	5 (15,2 %)
Divulgación de Información - Mensajes de Error de Depuración	Bajo	49 (148,5 %)
Divulgación de Marcas de Tiempo - Unix	Bajo	171 (518,2 %)

Divulgación de error de aplicación	Bajo	13 (39,4 %)
El servidor filtra información de versión a través del campo "Server" del encabezado de respuesta HTTP	Bajo	275 (833,3 %)
Falta encabezado X-Content-Type-Options	Bajo	368 (1.115,2 %)
Inclusión de archivos fuente JavaScript entre dominios	Bajo	35 (106,1 %)
Las Páginas Seguras Incluyen Contenido Mixto	Bajo	4 (12,1 %)
Revelación de IP privada	Bajo	3 (9,1 %)
Strict-Transport-Security Header No Establecido	Bajo	1502 (4.551,5 %)
Versión obsoleta de Asp.Net en uso	Bajo	1 (3,0 %)
Aplicación Web Moderna	Informativo	168 (509,1 %)
Atributo de elemento HTML controlable por el usuario (XSS potencial)	Informativo	161 (487,9 %)
Divulgación de Información - Información sensible en URL	Informativo	4 (12,1 %)
Divulgación de información - Comentarios sospechosos	Informativo	497 (1.506,1 %)
Evento JavaScript Controlable por el Usuario (XSS)	Informativo	3 (9,1 %)
Petición de Autenticación Identificada	Informativo	3 (9,1 %)
Petición de Verificación Identificada	Informativo	10 (30,3 %)
Reexaminar las Directivas de Control de Caché	Informativo	188 (569,7 %)
Respuesta de Gestión de Sesión Identificada	Informativo	376 (1.139,4 %)
User Agent Fuzzer	Informativo	2299 (6.966,7 %)
Total		**33**

Tabla 7 Cantidad de ocurrencias por cada tipo de alerta, junto con el nivel de riesgo.

Nota: Los porcentajes entre paréntesis representan cada recuento como porcentaje, redondeado a un decimal en función del número de alertas.

3.4. Discusión

Los resultados a partir del análisis de vulnerabilidades en las aplicaciones web de Grupo ALMAR evidencian la relevancia del uso de técnicas de pentesting para identificar

y mitigar amenazas cibernéticas, este enfoque se alinea con lo propuesto por (Tomanek & Klima, 2015), quienes destacan que el pentesting es una metodología efectiva para evaluar el nivel de seguridad de los sistemas y aplicaciones al simular ataques reales. La implementación de estas técnicas permitió identificar múltiples vulnerabilidades relacionadas con inyecciones SQL y configuraciones inadecuadas, problemas que también se documentan como prevalentes en estudios como el de OWASP Foundation (Larson, 2021), donde estas categorías son clasificadas entre las más comunes en aplicaciones web.

En términos metodológicos, el uso de herramientas dinámicas y estáticas para el pentesting ofreció una perspectiva integral, tal como lo sugieren (Correa et al., 2021), quienes resaltan que la combinación de enfoques proporciona resultados más precisos y fiables, sin embargo, los desafíos encontrados, como la limitación de tiempo para implementar medidas de mitigación, subrayan la necesidad de incorporar prácticas de seguridad en etapas tempranas del desarrollo, un aspecto que está respaldado por (Vovk, 2020), quienes enfatizan la importancia del "shift left" (enfoque de calidad, pruebas, seguridad y validación en el desarrollo de software) en seguridad.

Por otro lado, las vulnerabilidades detectadas en Grupo ALMAR no son únicas, sino que reflejan un patrón común en aplicaciones desarrolladas bajo modelos tradicionales sin un enfoque claro en seguridad (Rajapakse et al., 2022) esto pone de manifiesto la necesidad de promover prácticas de desarrollo seguro y adoptar marcos como DevSecOps, que integran la seguridad como parte del ciclo de vida del software.

Finalmente, a pesar de los hallazgos, la falta de tiempo para implementar una propuesta completa de mitigación de vulnerabilidades representa una limitación significativa de este estudio, esto refuerza la recomendación de priorizar proyectos futuros que permitan no solo la identificación de vulnerabilidades, sino también la implementación y evaluación de estrategias de mitigación, siguiendo lineamientos como los propuestos por (Larson, 2021).

CONCLUSIONES

A partir del trabajo realizado, se determinó que las técnicas de pentesting son herramientas para identificar y gestionar riesgos en aplicaciones web, entre las técnicas más utilizadas se encuentran las pruebas de inyección de SQL, pruebas de fuerza bruta y análisis de configuraciones de seguridad, cada una con resultados efectivos en diferentes entornos. Además, se evidenció que el uso de frameworks como OWASP ayuda a garantizar la cobertura adecuada en las pruebas, esto demuestra que el conocimiento actualizado y sistematizado de estas técnicas sirve para mejorar la seguridad en las aplicaciones web.

La implementación de pruebas de pentesting en las aplicaciones web de Grupo ALMAR permitió identificar vulnerabilidades críticas, tales como versiones obsoletas de frameworks para el desarrollo y configuraciones incorrectas del servidor. Es importante aplicar pruebas periódicas para garantizar la integridad y seguridad de los sistemas en un entorno empresarial, las vulnerabilidades detectadas indican que los sistemas web son un punto de exposición ante posibles amenazas.

El análisis de los resultados permitió validar las técnicas de pentesting implementadas en algunas de las aplicaciones web de Grupo ALMAR, el informe elaborado detalla el impacto de las vulnerabilidades detectadas, clasificándolas según su nivel de riesgo y proponiendo estrategias inmediatas de mitigación.

Aunque se planteó la elaboración de una propuesta de mitigación como parte de este trabajo, no fue posible llevarla a cabo debido a las limitaciones de tiempo inherentes al desarrollo de la tesis, sin embargo, los resultados y las vulnerabilidades identificadas ofrecen una base sólida para que futuros esfuerzos en la empresa puedan enfocarse en diseñar estrategias de seguridad específicas y efectivas.

RECOMENDACIONES

Dado que no fue posible elaborar una propuesta de mitigación dentro del tiempo asignado a la tesis, se recomienda que Grupo ALMAR priorice la implementación de un plan de acción basado en las vulnerabilidades identificadas, este plan debe considerar tanto soluciones técnicas inmediatas como estrategias de largo plazo para la mejora continua de la seguridad.

Se recomienda a Grupo ALMAR adoptar un calendario regular de pruebas de seguridad mediante pentesting, esto permitirá identificar y abordar nuevas vulnerabilidades que puedan surgir debido a actualizaciones del software o cambios en la infraestructura tecnológica.

Se recomienda al equipo de desarrollo de software adoptar prácticas de codificación seguras desde las etapas iniciales del ciclo de vida de desarrollo (SDLC), esto incluye la integración de herramientas de análisis estático y dinámico de código, así como la realización de revisiones de seguridad antes de poner en producción las aplicaciones.

Dado que el pentesting puede requerir conocimientos especializados y herramientas avanzadas, se sugiere que Grupo ALMAR colabore con consultores externos o empresas especializadas en ciberseguridad para llevar a cabo auditorías periódicas y garantizar que se implementen las mejores prácticas del sector.

REFERENCIAS

Amazon. (2024). *amazonq-developer-ug*.

Andrés Maíllo Fernández, J., & -México, B. (2021). *Hackers Técnicas y herramientas para atacar y defendernos Informática*.

Astudillo, K. (2021). *Beneficios de realizar Pentesting*.

Candau, J., & Marco, D. (2021). *Ciberseguridad. Evolución y tendencias Palabras clave*.

Cano M., J. J. (2020). Ciberataques. *Revista SISTEMAS*, *157*, 67–74. https://doi.org/10.29236/sistemas.n157a6

Correa, R. A., Higuera, J. R. B., Higuera, J. B., Montalvo, J. A. S., Rubio, M. S., & Alberto Magreñán. (2021). Hybrid security assessment methodology for web applications. *CMES - Computer Modeling in Engineering and Sciences*, *126*(1), 89–124. https://doi.org/10.32604/CMES.2021.010700

De la Torre, C., & De la Torre, M. (2021). *CyberNoticia48-20171016*.

Felipe Redondo, A. M., & Núñez Cárdenas, F. de J. (2024). Criterios de selección de herramientas para pentesting. *Ciencia Huasteca Boletín Científico de La Escuela Superior de Huejutla*, *12*(24), 31–35. https://doi.org/10.29057/esh.v12i24.12763

Huajie, X. (2023). Github Copilot : A Groundbreaking Code Autocomplete Tool. *ResearchGate*, *10.13140/RG.2.2.29962.24002*.

Instituto Nacional de Ciberseguridad de España. (2021). *Glosario de términos de ciberseguridad*.

Kongara, D., & Krishnama, S. (2023). A Process of Penetration Testing Using Various Tools. *Mesopotamian Journal of CyberSecurity*, *2023*, 93–103. https://doi.org/10.58496/MJCS/2023/014

Larrea, M. L. (2021). *El Análisis Estático como Herramienta de Evaluación en Cátedras con Proyectos de Programación*.

Larson, N. (2021). *Who We Be*.

Pastor Ricós, F. (2020). *Pentesting y generación de exploits con Metasploit*.

Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. In *Information and Software Technology* (Vol. 141). Elsevier B.V. https://doi.org/10.1016/j.infsof.2021.106700

Rodríguez Llerena, A. E. (2020). Herramientas fundamentales para el hacking ético Fundamental Tools for Ethical Hacking. *Revista Cubana de Informática Médica*, *2020*(1), 116–131. http://scielo.sld.cu

Safla, D. G. (2021). *"PROYECTO ABIERTO DE SEGURIDAD DE APLICACIONES WEB" Proyecto de investigación previo a la obtención del título de Magíster en Ciberseguridad Autor*.

Scariot Esquivel, N. (2022). ¿Qué es ser hacker? *Question/Cuestión*, *3*(71), E661. https://doi.org/10.24215/16696581e661

Tomanek, M., & Klima, T. (2015). Penetration Testing in Agile Software Development Projects. *International Journal on Cryptography and Information Security*, *5*(1), 01–07. https://doi.org/10.5121/ijcis.2015.5101

Valarezo Pardo, M. R., Honores Tapia, J. A., Gómez Moreno, A. S., & Vinces Sánchez, L. F. (2018). Art_2. *3C Tecnología*. https://3ciencias.com/wp-content/uploads/2018/09/Art_2.pdf

Vanegas Romero, A. Y. (2021). Pentesting, ¿Porque es importante para las empresas? In *Universidad Piloto de Colombia*.

Vega Briceño, E. (2021). *SEGURIDAD DE LA INFORMACIÓN*.

Venza. (2023). *Pruebas de Penetración y Análisis de Vulnerabilidades*.

Vovk, V. (2020). *Testing for concept shift online*. http://arxiv.org/abs/2012.14246

ANEXOS

Acuerdo de confidencialidad y no divulgación.

AUTORIZACIÓN DE ACCESOS, ACUERDO DE CONFIDENCIALIDAD Y NO DIVULGACIÓN

Conste por el presente instrumento una autorización de accesos, acuerdo de confidencialidad y no divulgación que se determina y contiene al tenor de las siguientes cláusulas:

PRIMERA.- INTERVINIENTES: Celebran el presente acuerdo:

1.1.- La compañía SOCIEDADES ACUÍCOLAS AL-MAR, SOCALMAR S.A. ubicada en Durán, Km. 8 Vía Durán-Tambo y debidamente representada por su Gerente General, señor Wolfgang Harten Alava, parte a la que se podrá identificar en adelante como la "Empresa" y/o "SOCALMAR"; y

1.2.- El señor Jamil Javier Cerezo Zambrano, por sus propios y personales derechos, estudiante de la maestría en Ciberseguridad de la Universidad Península de Santa Elena (UPSE), quien tiene domicilio en Guayaquil, Sauces 5, Mz. 253, Villa 27, parte a la que en adelante se podrá identificar como el " Estudiante".

SEGUNDA.- ANTECEDENTES:

2.1. El Estudiante se encuentra cursando una maestría en Ciberseguridad en la UPSE y, como parte de los requisitos académicos que el programa requiere, debe realizar un trabajo de grado (tesis) en el que ha estado trabajando y al que denomina: "Implementación de pruebas de Pentesting para detección de vulnerabilidades en aplicaciones web de Grupo ALMAR", teniendo en consideración que el referido Estudiante es, además, colaborador de dicho Grupo, por lo que el presente Acuerdo y sus obligaciones se extienden no solo a SOCALMAR sino a todas las compañías que forman parte de Grupo Almar, entre éstas: Produmar S.A., Limbomar S.A., Produpesada S.A., Biotecnología & Genética Marina S.A. (BIOGEMAR) y cualquier otra que se incorpore al Grupo.

2.2. En este sentido, el Estudiante ha solicitado a SOCALMAR que le permita acceder a cierta información sensible de la Empresa, entre ésta, configuraciones de sistemas, datos internos y vulnerabilidades, siempre bajo la supervisión de quien la Empresa designe para el efecto.

2.3.- La Empresa ha accedido al pedido del Estudiante, con las limitaciones y obligaciones que aquí se establecen y por un período limitado al levantamiento de la información que el Estudiante requiere para la conclusión del trabajo de maestría, tiempo que en todo caso no podrá ser superior a 5 meses. Sin perjuicio de lo dicho, la Empresa podrá en cualquier momento revocar la autorización de accesos concedida al Estudiante si estima que ello es lo pertinente.

TERCERA.- ASPECTOS PARTICULARES:

3.1. Definición de Información Confidencial

Para efectos de este acuerdo, "Información Confidencial" es cualquier información técnica, datos, archivos, configuraciones, contraseñas, metodologías, procesos, o cualquier otra información que la Empresa considere de carácter sensible y que el Estudiante pueda tener acceso durante la realización de las pruebas de penetración. Esto incluye, pero no se limita a:

- Configuraciones de redes y servidores.
- Credenciales de acceso a sistemas.
- Resultados de las pruebas de vulnerabilidad.

3.2. Obligación de No Divulgación

El Estudiante se compromete a no divulgar, difundir, compartir o de cualquier manera poner a disposición de terceros la Información Confidencial sin el consentimiento expreso y por escrito de la Empresa, salvo que sea requerido por ley o autoridad judicial.

3.3. Uso de la Información Confidencial

El Estudiante acuerda utilizar la Información Confidencial únicamente para la realización de las pruebas de penetración requeridas por su programa académico y para ningún otro propósito, ya sea personal, comercial o de cualquier otra índole. La Información Confidencial solo podrá ser

AUTORIZACIÓN DE ACCESOS, ACUERDO DE CONFIDENCIALIDAD Y NO DIVULGACIÓN

incluida en su proyecto de maestría de forma anonimizada o con el consentimiento de la Empresa.

3.4. Protección de la Información

El Estudiante se compromete a:

Tomar todas las medidas razonables para proteger la Información Confidencial contra acceso no autorizado, pérdida, uso indebido o divulgación.

Limitar el acceso a la Información Confidencial exclusivamente a sí mismo o a cualquier persona expresamente autorizada por la Empresa y por la institución académica, si fuese necesario.

Destruir o devolver a la Empresa toda la Información Confidencial a la conclusión de las pruebas de penetración y la presentación de su informe académico.

3.5. Exclusiones de la Información Confidencial

No será considerada Información Confidencial aquella que:

Sea de conocimiento público o se haga pública sin que ello implique incumplimiento de este acuerdo por parte dell Estudiante.

Haya sido desarrollada por el Estudiante de forma independiente y sin referencia a la Información Confidencial de la Empresa.

Haya sido obtenida legalmente de un tercero sin restricciones de confidencialidad.

3.6. Duración de las obligaciones de Confidencialidad y no Divulgación

Las obligaciones de confidencialidad y no divulgación entrará en vigor a partir de la firma del presente instrumento y estarán vigentes de forma indefinida.

3.7. Propiedad Intelectual

El Estudiante reconoce que toda la Información Confidencial y cualquier trabajo derivado de ella sigue siendo propiedad exclusiva de la Empresa. El informe académico derivado del pentesting puede ser utilizado por el Estudiante exclusivamente para fines educativos, a menos que la Empresa otorgue su permiso para otro uso.

3.8. Responsabilidad por Incumplimiento

El Estudiante será responsable por cualquier daño que se derive del incumplimiento de este acuerdo, y La Empresa podrá tomar las acciones legales que considere necesarias para proteger su información y obtener una compensación por los perjuicios causados.

3.9. Datos Personales

Queda prohibido el acceso y divulgación de datos personales cuya custodia y tratamiento tenga la Empresa, de manera que dicha información no es materia del presente Acuerdo.

Para constancia de todo lo cual, se suscribe el presente instrumento en 2 ejemplares de igual valor y tenor a 11 de septiembre de 2024.

Por SOCALMAR

Wolfgang Harten Alava

Gerente General

El Estudiante

Jamil Javier Cerezo Zambrano

Cédula No.: 1207195882

Informe de vulnerabilidades generado por la herramienta OWASP ZAP

Análisis de vulnerabilidades - Aplicaciones web ALMAR

Generated with ZAP on vie 4 oct 2024, at 21:59:56

ZAP Versión: 2.15.0

ZAP is supported by the Crash Override Open Source Fellowship

Contents

About this report

Report description

Análisis spider, spider AJAX y escaneo activo de las aplicaciones web producción, empacadora, compras y planificación

Report parameters

Contexts

No contexts were selected, so all contexts were included by default.

Sites

The following sites were included:

- https://pruebasplanificacion.grupoalmar.com.ec
- https://testlimbopack.grupoalmar.com.ec
- https://test.compras.grupoalmar.com.ec
- https://testalmar.produccion.grupoalmar.com.ec

(If no sites were selected, all sites were included by default.)

An included site must also be within one of the included contexts for its data to be included in the report.

Risk levels

Included: Alto, Medio, Bajo, Informativo

Excluded: None

Confidence levels

Included: Confirmado por Usuario, Alta, Media, Baja

Excluded: Confirmado por Usuario, Alta, Media, Baja, Falso positivo

Summaries

Alert counts by risk and confidence

This table shows the number of alerts for each level of risk and confidence included in the report.

(The percentages in brackets represent the count as a percentage of the total number of alerts included in the report, rounded to one decimal place.)

		Confidence				
		Confirmado por Usuario	Alta	Media	Baja	Total
Risk	Alto	0 (0,0 %)	0 (0,0 %)	3 (9,1 %)	1 (3,0 %)	4 (12,1 %)
	Medio	0 (0,0 %)	1 (3,0 %)	4 (12,1 %)	1 (3,0 %)	6 (18,2 %)
	Bajo	0 (0,0 %)	2 (6,1 %)	10 (30,3 %)	1 (3,0 %)	13 (39,4 %)
	Informativo	0 (0,0 %)	0 (0,0 %)	4 (12,1 %)	6 (18,2 %)	10 (30,3 %)
	Total	0 (0,0 %)	3 (9,1 %)	21 (63,6 %)	9 (27,3 %)	33 (100%)

Alert counts by site and risk

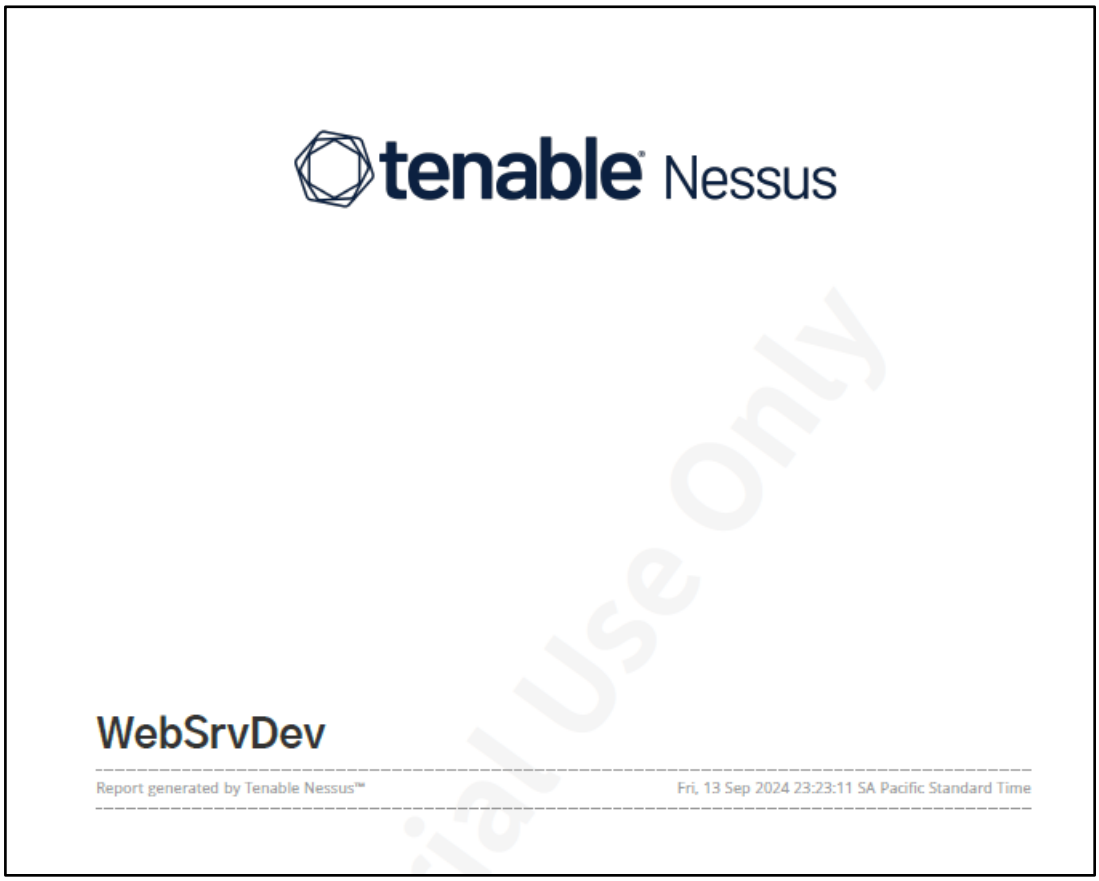
This table shows, for each site for which one or more alerts were raised, the number of alerts raised at each risk level.

Alerts with a confidence level of "False Positive" have been excluded from these counts.

(The numbers in brackets are the number of alerts raised for the site at or above that risk level.)

		Risk			
		Alto (= Alto)	Medio (>= Medio)	Bajo (>= Bajo)	Informativo (>= Informativo)
Site	https://pruebasplanificacion.grupoalmar.com.ec	2 (2)	0 (2)	1 (3)	1 (4)
	https://testlimbopack.grupoalmar.com.ec	1 (1)	0 (1)	2 (3)	0 (3)
	https://test.compras.grupoalmar.com.ec	0 (0)	0 (0)	1 (1)	0 (1)
	https://testalmar.produccion.grupoalmar.com.ec	1 (1)	4 (5)	7 (12)	9 (21)

Informe de vulnerabilidades generado por la herramienta Nessus Professional



testalmar.produccion.grupoalmar.com.ec

CRITICAL	HIGH	MEDIUM	LOW	INFO
2	0	8	2	24

Vulnerabilities Total: 36

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
CRITICAL	9.8	6.7	0.0013	179364	PHP 8.0.x < 8.0.30 Multiple Vulnerabilities
CRITICAL	10.0	-	-	58987	PHP Unsupported Version Detection
MEDIUM	6.5	-	-	142960	HSTS Missing From HTTPS Server (RFC 6797)
MEDIUM	6.1	5.7	0.0627	136929	JQuery 1.2 < 3.5.0 Multiple XSS
MEDIUM	5.3	-	-	11229	Web Server info.php / phpinfo.php Detection
MEDIUM	4.3	1.4	0.0005	177509	PHP 8.0.x < 8.0.29
MEDIUM	4.3*	-	-	44136	CGI Generic Cookie Injection Scripting
MEDIUM	4.3*	-	-	49067	CGI Generic HTML Injections (quick test)
MEDIUM	4.3*	-	-	39466	CGI Generic XSS (quick test)
MEDIUM	4.3*	-	-	85582	Web Application Potentially Vulnerable to Clickjacking
LOW	N/A	-	-	42057	Web Server Allows Password Auto-Completion
LOW	2.6*	-	-	26194	Web Server Transmits Cleartext Credentials
INFO	N/A	-	-	47830	CGI Generic Injectable Parameter
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
INFO	N/A	-	-	84502	HSTS Missing From HTTPS Server
INFO	N/A	-	-	69826	HTTP Cookie 'secure' Property Transport Mismatch
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	-	-	106658	JQuery Detection
INFO	N/A	-	-	50344	Missing or Permissive Content-Security-Policy frame-ancestor HTTP Response Header
INFO	N/A	-	-	50345	Missing or Permissive X-Frame-Options HTTP Response Head
INFO	N/A	-	-	11219	Nessus SYN scanner
INFO	N/A	-	-	19506	Nessus Scan Information
INFO	N/A	-	-	48243	PHP Version Detection
INFO	N/A	-	-	66334	Patch Report
INFO	N/A	-	-	33139	WS-Management Server Detection
INFO	N/A	-	-	85601	Web Application Cookies Not Marked HttpOnly
INFO	N/A	-	-	85602	Web Application Cookies Not Marked Secure
INFO	N/A	-	-	40773	Web Application Potentially Sensitive CGI Parameter Detectio
INFO	N/A	-	-	91815	Web Application Sitemap
INFO	N/A	-	-	11032	Web Server Directory Enumeration
INFO	N/A	-	-	10662	Web mirroring
INFO	N/A	-	-	156439	jQuery UI Detection

* indicates the v3.0 score was not available; the v2.0 score is shown