Análisis de pruebas de penetración en sistemas y servicios web

Erick Santiago González Mejía[1]
erick.gonzalez00@usc.edu.co

Dalessandro Chaves Cardenas[2]
dalessandro.chaves00@usc.edu.co

Erick Steven Mateus Rendon[3]
erick.mateus00@usc.edu.co

Javier Salvador Rojas Montes[4]
javier.rojas00@usc.edu.co

Estudiante, Universidad Santiago de Cali, Facultad de Ingeniería, Programa de ingeniería de Sistemas (1)
Estudiante, Universidad Santiago de Cali, Facultad de Ingeniería, Programa de ingeniería de Sistemas (2)
Estudiante, Universidad Santiago de Cali, Facultad de Ingeniería, Programa de ingeniería de Sistemas (3)
Profesor, Universidad Santiago de Cali, Facultad de Ingeniería, Programa de ingeniería de Sistemas (4)

Resumen

Los servicios informáticos enfrentan diversos riesgos, principalmente por ataques cibernéticos. Las pruebas de penetración web determinan si un sistema web es vulnerable a ataques utilizando herramientas y técnicas que suelen ser empleadas por especialistas de penetración quienes se dedican a identificar vulnerabilidades de estos. Se presentan casos de estudio con el fin de exponer una adecuada gestión de la ciberseguridad, utilizando herramientas especializadas como Nessus en el que se incluye un caso práctico de una universidad en donde se usa esta herramienta para evaluar la seguridad en sus sitios web, y también Metasploit la cual junto con el sistema operativo Kali Linux se aborda un caso práctico en donde se vulnera completamente dispositivos IoT. La revisión se elaboró con la metodología PRISMA, seleccionando 70 artículos y de los cuales se utilizaron 26 artículos. Esta metodología garantiza la calidad y confiabilidad de las fuentes. Las pruebas de penetración web evaluaron vulnerabilidades mediante enfoques de: caja negra, blanca y gris, empleándose en técnicas manuales y/o automatizadas. Las pruebas de penetración web consiste en una metodología de 5 fases: planificación, escaneo, explotación, mantenimiento del acceso y generación de informes. En conclusión, se evidenció que la combinación de enfoques de pruebas de penetración (caja negra, blanca y gris), junto con el uso de tecnologías emergentes como la inteligencia artificial, no solo optimiza la detección de vulnerabilidades, sino que también fortalece la seguridad de los sistemas. La incorporación de herramientas automatizadas y la simulación avanzada de ataques permiten a las organizaciones mitigar posibles impactos y proteger sus sistemas de manera más eficiente.
Palabras clave: Pentesting, Ciberseguridad, Vulnerabilidades, Hacking Ético, Metodología PRISMA

Abstract

IT services face various risks, mainly from cyber attacks. Web penetration testing determines whether a web system is vulnerable to attacks using tools and techniques that are usually used by penetration specialists who are dedicated to identifying vulnerabilities in these. Case studies are presented in order to expose an adequate management of cybersecurity, using specialized tools such as Nessus, which includes a practical case of a university where this tool is used to evaluate the security of its websites, and also Metasploit, which together with the Kali Linux operating system addresses a practical case where IoT devices are completely compromised. The review was prepared using the PRISMA methodology, selecting 70 articles and of which 26 articles were used. This methodology guarantees the quality and reliability of the sources. Web penetration testing evaluated vulnerabilities using black, white and gray box approaches, using manual and/or automated techniques. Web penetration testing consists of a 5-phase methodology: planning, scanning, exploitation, maintaining access and generating reports. In conclusion, it was found that the combination of penetration testing approaches (black, white and grey box), together with the use of emerging technologies such as artificial intelligence, not only optimizes vulnerability detection, but also strengthens system security. The incorporation of automated tools and advanced attack simulation allows organizations to mitigate potential impacts and protect their systems more efficiently.
Keywords: Pentesting, Cybersecurity, Vulnerabilities, Ethical Hacking, PRISMA Methodology

1. INTRODUCCIÓN

A pesar de los avances significativos en ciberseguridad, persisten importantes desafíos que necesitan ser abordados, uno de ellos son las múltiples vulnerabilidades que pueden tener los sistemas alojados en la web; para comprender la problemática que se va a abordar, se expondrá alguna de las vulnerabilidades más relevantes, para posteriormente conocer las técnicas que se usan para explotar dichas vulnerabilidades mediante las pruebas de penetración web. Los tópicos que se desarrollarán en este artículo se relacionan directamente con los riesgos que pueden impactar en un servicio de tecnología de información, en este caso el centro de la investigación se basa en los sistemas web y cómo pueden ser víctima de ataques de penetración mediante procesos relacionados al ámbito de ciberseguridad.

La sociedad moderna es dependiente al software el cual les ayuda a realizar diferentes actividades diarias de alta complejidad, como resultado estos sistemas están en constante exposición ante amenazas y riesgos que atenten contra su correcto funcionamiento. La complejidad misma de estos sistemas incrementa el riesgo de ser víctima de un ciberataque, e inclusive de ataques afectando su servicio, y en el peor caso deteriorar de manera crítica todo el sistema. Diferentes de estos incidentes se han presentado en todos los sectores de las industrias, todos los sistemas están expuestos, es por esto que es pertinente recurrir a medidas que mitiguen estos sucesos (Rodríguez et al., 2023).

Asegurar la información alojada en la web es importante, puesto que supone una garantía de que el área de la tecnología de información presta un servicio integral, dando así un sistema seguro y protegido de ataques malintencionados. Esta premisa se basa en los 3 principios del triángulo de la seguridad, el cual consiste en que la seguridad requiere de confidencialidad, integridad y disponibilidad. La confidencialidad establece que solo las fuentes autorizadas pueden acceder a cierta información. La integridad consiste en que sólo las personas y los recursos autorizados pueden modificar añadir o eliminar información y funciones sensibles. La disponibilidad se refiere a que el sistema, sus datos y su información deben estar disponibles para la demanda del usuario (Li & Liu, 2021).

Para el año 2021, se estima que el costo acumulado de una violación de datos es de 4 millones de dólares, y por cada registro perdido o robado que contenga información sensible y confidencial es de 158 dólares. Aproximadamente el 25% de las violaciones de datos se encuentra relacionado con las principales causas: el factor humano, que puede incurrir en errores por diversas razones, mientras que el 27% se atribuye al mal funcionamiento del sistema, a ataques maliciosos o a actividades criminales (Tiwari et al., 2021).

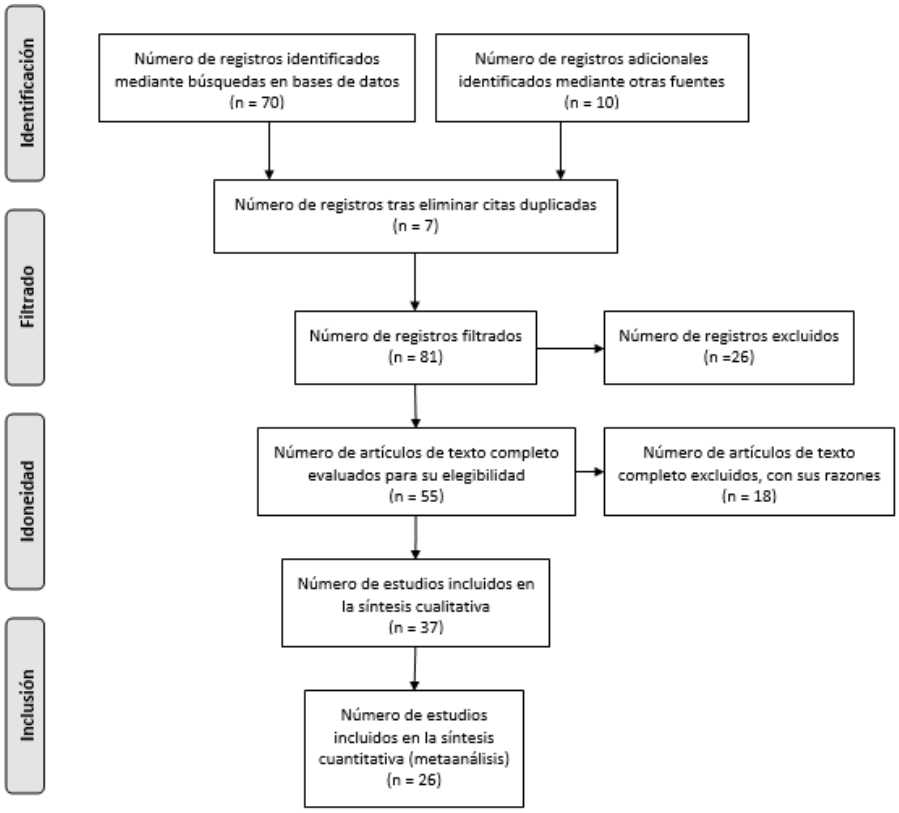
El hacking ético y pentesting tienen un rol importante en la identificación de vulnerabilidades y evaluación de la seguridad de un sistema usando las mismas técnicas de los atacantes. Las pruebas de penetración web consisten en evaluar la vulnerabilidad de una aplicación utilizando diversas herramientas y técnicas comúnmente empleadas por expertos autorizados. Estos profesionales actúan como atacantes simulados, buscando explotar posibles fallos de seguridad que podrían comprometer el sistema, desde el robo de datos hasta el control total del mismo, con el objetivo de identificar y demostrar debilidades en la seguridad del sistema para mejorar su seguridad (Auricchio et al., 2022). Existen diversos mecanismos con los que es posible acceder o penetrar un sitio web, por lo que en este artículo se expondrán algunos de estos mecanismos con el fin de contextualizar el entorno en el que se centran las pruebas de penetración web (Altulaihan et al., 2023).

La estructura para desarrollar este artículo de revisión se centró en la metodología PRISMA, la cual proporciona una guía detallada para documentar cada paso del proceso de revisión empleando el flujo de trabajo propuesto por la misma, que va desde la identificación y selección de estudios como también los respectivos criterios de inclusión y exclusión hasta la síntesis de resultados, aumentando así la confiabilidad de los hallazgos y su relevancia para futuras investigaciones.

2. METODOLOGÍA

La elaboración de este artículo de revisión se estructuró con base en la metodología PRISMA, proporcionando una guía confiable para desarrollar paso a paso un artículo de revisión de alta calidad y de total transparencia presentando hallazgos de manera clara y completa. Mediante esta metodología se realizaron búsquedas bibliográficas en donde se analizaron artículos publicados en bases de datos a partir de los temas pertinentes y relevantes al ámbito de la ciberseguridad, más específicamente en el escenario de penetración web, de esta forma garantizamos calidad y confiabilidad de la fuente de información estableciendo así el temario que dio forma al trabajo investigativo. En la búsqueda se seleccionaron aproximadamente 70 artículos, de los cuales se emplearon 26 para enriquecer la bibliografía de la presente revisión, extrayendo información relevante sobre diversos aspectos de la temática en estudio.

Ilustración 1: Diagrama de flujo de la metodología PRISMA para la elaboración del artículo



Fuente: Elaboración Propia

Para la fase de identificación se emplearon en su mayoría bases de datos las cuales tienen acceso los estudiantes de la Universidad Santiago de Cali, como por ejemplo Scopus, Science Direct, Springer entre otros, además de otros repositorios alojados en Google Scholar, cada artículo de revisión se tenía en cuenta su identificador DOI para validar su oficialidad. En la fase de filtrado, se consideraba la fecha de publicación, procurando que no superara los cinco años. Una vez revisados los artículos, se evaluaba el aporte que cada uno podía ofrecer a la bibliografía y a la composición del texto en desarrollo. Durante este proceso, tras realizar la búsqueda de artículos, se identificó que cierta información se repetía o resultaba similar. Por ello, se descartaron los artículos que ya no eran relevantes. Para el proceso de búsqueda de los artículos en las diferentes bases de datos se utilizaron palabras clave referentes al tema, las cuales se usaron de forma individual como también con conectores AND cuando se quería relacionar un tema específico de pentesting junto con una tecnología basada en la web. En la fase de idoneidad se emplearon los artículos los cuales una vez filtrados los artículos se consideraban (Alhamed & Rahman, 2023).

En el ámbito de ciberseguridad es importante delimitar las diferencias de algunos conceptos los cuales se van a abordar en el desarrollo del artículo, tenemos por un lado el concepto de vulnerabilidad el cual hace referencia a una debilidad

dentro de un procedimiento de seguridad la cual puede ser explotada o activada por amenazas internas o externas, y el concepto de ciberactivo se refiere a cualquier recurso digital que tiene valor para una organización y que puede ser objeto de gestión y protección en el ámbito de la ciberseguridad. Esto incluye no solo los datos e información, sino también los sistemas, aplicaciones y tecnologías que facilitan el manejo de dicha información (Pérez & Ortega, 2020). Además, el concepto de amenaza abarca cualquier evento con la capacidad de comprometer el ciberactivo mediante accesos no autorizados, destrucción, divulgación, alteración de la información o interrupción en la prestación de servicios. Finalmente, los ciberataques son actos cibernéticos no autorizados que buscan violar la política de seguridad de un ciberactivo, causando daños, perturbando o interrumpiendo los servicios o el acceso a la información de dicho recurso (Li & Liu, 2021).

Las pruebas de penetración web determinan si un sistema web es vulnerable a ataques utilizando varias herramientas y técnicas que suelen ser empleadas por probadores de penetración o pentesters. Involucra encontrar y detectar fallos de seguridad en aplicaciones web para preservar la confianza y los datos de los usuarios (Bhesaniya & Kumar Agrawal, 2022). De tal manera, se establecen dos preguntas de investigación que fundamentan este artículo: ¿Qué metodologías y técnicas se usan recientemente para realizar pruebas de penetración en sistemas y servicios web y cómo se llevan a cabo estas pruebas de penetración?

Los expertos en pruebas de penetración son autorizados por una organización, para desempeñar el papel de atacantes con el fin de aprovechar las vulnerabilidades potenciales como puntos de acceso al sistema que será sometido a pruebas. El objetivo es demostrar la presencia de fallos de seguridad que pueden conducir a actividades ilegales, desde el robo de datos sensibles hasta el control total del sistema, con el propósito de conocer qué nivel de seguridad presenta su sistema.

Las pruebas de penetración web tienen en común la metodología general implementada para las pruebas de penetración estándar. Se trata de un proceso que aumenta la comprensión del estado de seguridad de un sistema objetivo, partiendo de la ausencia de información y hasta la construcción de una prueba de las vulnerabilidades encontradas (Auricchio et al., 2022).

El proceso de pruebas de penetración web se estructura en 5 etapas: Planificación/Reconocimiento, escaneo, explotación, mantenimiento del acceso/análisis de resultados e informes.

La etapa de planificación define el alcance y objetivos de las pruebas, busca comprender cómo funciona el sistema en cuestión e identificar sus posibles vulnerabilidades. La etapa de escaneo busca analizar cómo responde el sistema ante intentos de intrusión. En la etapa de explotación se emplean los diferentes mecanismos y técnicas que existen para explotar las vulnerabilidades encontradas y obtener acceso a ellas (Nagendran, Adithyan, Chethana, Camillus, & Bala Sri Varshini, 2019). En la etapa de mantenimiento del acceso, se analizan los resultados y se garantiza que los hallazgos encontrados no son falsos positivos sino vulnerabilidades reales. Finalmente, se registra un informe con detalles sobre los datos confidenciales a los que se pudo tener acceso, qué vulnerabilidades fueron explotadas y el tiempo que el atacante no fue detectado y demás información que se considere relevante (Pinto João, 2020).

Para comprender la relevancia de las amenazas que se puede enfrentar un sistema web existe el OWASP Top 10 siendo una guía clave en el pentesting web, que identifica las vulnerabilidades de seguridad más críticas en sistemas web. Este listado, basado en datos globales, permite a los profesionales de seguridad priorizar los problemas de mayor riesgo y enfocar sus esfuerzos en remediar fallos críticos, como la inyección de código y la configuración incorrecta de seguridad. Para desarrolladores y pentesters, seguir el OWASP Top 10 no solo mejora la protección contra ataques comunes, sino que también establece un estándar de seguridad que facilita la comunicación y colaboración en la creación de aplicaciones seguras (Aydos et al., 2021).

OWASP (Open Web Application Security Project) proporciona un documento sobre seguridad web en constante actualización. OWASP Top 10 expone una lista de los 10 riesgos críticos más importante en la seguridad de los sistemas web. Shahid et al., (2022), reporta la mayor clasificación de las vulnerabilidades para el 2021:

- Control de Acceso Quebrado
- Fallas Criptográficas
- Inyección
- Diseño Inseguro

- Configuración Errada de Seguridad
- Componentes Vulnerables y Desactualizados
- Fallas de Autenticación
- Fallas de Integridad de Software y Datos
- Fallas de Registro y Monitoreo de Seguridad
- Falsificación de Peticiones en Sitios Cruzados

2.1 Técnicas y prácticas

La elección de la estrategia depende del nivel de conocimiento que tenga el pentester sobre la red o sistema del cliente. Existen tres estrategias fundamentales para realizar pruebas de penetración, cada una con un enfoque diferente y características propias:

2.1.1 Caja Gris: En esta prueba la persona encargada de realizar la prueba tiene un conocimiento y acceso a las partes internas como, por ejemplo: direcciones IP, nombres de host, correos electrónicos, etc. Además, también puede tener conocimiento sobre el diseño de la documentación y la prueba se realiza desde una perspectiva de usuario. En comparación con la técnica de caja blanca, esta resulta ser más rápida.

2.1.2 Caja negra: Esta técnica también es llamada prueba de caja cerrada, su característica principal consiste en que el probador no tiene información sobre el objetivo. Requiere de mucho más tiempo en comparación con los demás enfoques debido a esta característica de carencia de información, y se ejecutan limitados casos de pruebas en las que el pentester examina desde la perspectiva de un hacker calificado.

2.1.3 Caja blanca: El pentester dispone de toda la información sobre el cliente. Se le proporciona la mayor parte de la información, de tal manera esto permite que se puedan realizar muchos más casos de prueba en comparación con la técnica de caja negra y puede llegar a ser menos costoso que cualquier otro enfoque. El evaluador en esta técnica debe tener buenos conocimientos de programación y lógica.

Para llevar a cabo un proceso de pruebas de penetración en sistemas web, se pueden emplear diversas técnicas y prácticas para ejecutarlas, a continuación, se dan a conocer algunos de estos métodos:

2.1.4 Pruebas de penetración manuales/convencionales: Involucran tareas tediosas como escribir código para ejecutar ataques de seguridad emulados. Esto incluye fases como planificación, reconocimiento, modelado de amenazas, explotación, post-explotación, generación de informes y nuevas pruebas.

(Kongara & Krishnama, 2023).

2.1.5 Pruebas de penetración no convencionales/automatizadas: Utilizan herramientas y técnicas más autónomas para realizar las pruebas de seguridad. Esto agrega una dimensión de automatización desde el escaneo hasta la fase de nuevas pruebas.

Con el creciente interés en la inteligencia artificial, ha surgido una demanda considerable de su aplicación en el pentesting. Las técnicas de IA están siendo utilizadas para automatizar y mejorar los resultados de estas pruebas, que tradicionalmente son llevadas a cabo por un grupo selecto de pentesters expertos. Estos profesionales investigan las vulnerabilidades mediante un enfoque de varias etapas (Chowdhary et al., 2023). En este contexto, las ramas de la inteligencia artificial, como el aprendizaje automático (Machine Learning) y el aprendizaje profundo (Deep Learning), están siendo ampliamente utilizadas en sistemas defensivos como los sistemas de detección y prevención de intrusiones (IDPS), el análisis de malware y soluciones en antivirus, la mayoría de estas técnicas de antivirus pueden eludirse utilizando diferentes métodos, a raíz de estos avances diversos autores han hecho algunos aportes para emplear la inteligencia artificial para automatizar el proceso de pruebas de penetración. Por ejemplo, algunas de las propuestas son las siguientes: un framework que utiliza un árbol de decisión para seleccionar el exploit más apropiado, teniendo en cuenta el sistema operativo en ejecución, los servicios activos del objetivo, y los CVE en cuestión. El uso del Machine Learning para analizar el tráfico de red producido durante una prueba de penetración, con el fin de reconocer automáticamente qué herramienta utilizar. Se desarrollado un framework de pentesting basado en Deep Learning el cual se encarga de automatizar el paso de reconocimiento. En dicho framework se ejecutaron ataques de escaneo en un banco de pruebas que emulaba un aeropuerto

para validar el marco, que superó a otras técnicas como la máquina de vectores de soporte, k-nearest neighbours y el perceptrón multicapa. Además, un estudio analiza cómo pueden emplearse los métodos de Machine Learning para detectar malware en sistemas de información empresariales basados en el Internet de las Cosas, incluida la detección de malware estática, dinámica, promovida e híbrida automatizando el proceso de pentesting sin imponer restricciones a las especificaciones de la red objetivo, lo que las hace aplicables también a este entorno IoT (Lawrence et al., 2022).

2.2 Evaluaciones de vulnerabilidad de IoT

En el contexto de la seguridad de dispositivos IoT, herramientas como Shodan y Wappalyzer desempeñan un papel crucial al proporcionar información detallada sobre la infraestructura tecnológica y las posibles vulnerabilidades de los sistemas conectados. Estas herramientas se convierten en aliados indispensables para los expertos en ciberseguridad, al permitir una evaluación proactiva y detallada de la superficie de ataque de los entornos IoT que interactúan con distintas tecnologías web.

Shodan es un motor de búsqueda para IoT el cual es capaz de identificar información por ejemplo de dispositivos industriales, como dispositivos IoT domésticos, vehículos conectados y muchos de los dispositivos que son capaces de conectarse en todo el mundo. Shodan es una poderosa herramienta con la cual se puede realizar el reconocimiento en pentesting (Heiding et al., 2023).

De acuerdo con Shi et al., (2024), aplicó Wappalyzer herramienta que permite reconocer las tecnologías web utilizadas en un sitio, incluyendo CMS como WordPress y Drupal, frameworks de JavaScript, servicios de hosting, entre otros componentes de software. En la etapa de reconocimiento del pentesting, su función es recopilar información crucial sobre la estructura técnica del objetivo, lo que facilita el análisis de vulnerabilidades específicas de cada tecnología. Para ello, Wappalyzer envía una solicitud HTTP a la URL objetivo para obtener el encabezado y el cuerpo de la respuesta, en busca de patrones específicos que ayudan a identificar las tecnologías utilizadas. Estos patrones, conocidos como "huellas digitales", se basan en campos clave como encabezados HTTP, nombres de archivos de recursos (JavaScript, íconos, CSS) y comentarios en el código HTML. Al detectar coincidencias con sus reglas predefinidas, Wappalyzer puede identificar qué tecnologías están en uso en el sitio.

Además, Wappalyzer puede extraer características clave del sistema web a través de un proceso de selección, que incluye:

- Encabezados HTTP.
- Archivos de recursos: Analiza las rutas y nombres de archivos como JavaScript o íconos para identificar frameworks específicos.
- Código HTML: Examina los comentarios y meta tags dentro del HTML.

Si bien Wappalyzer es efectivo para reconocer una amplia variedad de tecnologías, puede tener dificultades con aplicaciones menos comunes o emergentes. Para mejorar esta situación, se recomienda combinar Wappalyzer con técnicas de reducción de dimensionalidad y algoritmos de clustering, lo que optimiza la precisión y el alcance de la identificación, adaptándose mejor a las nuevas tecnologías.

2.3 Caso de estudio: Ejemplo de fase de reconocimiento y escaneo

En las pruebas de penetración web, la fase de reconocimiento y escaneo es crucial para identificar posibles vulnerabilidades en la infraestructura. Existe una variedad de herramientas especializadas que facilita este proceso, permitiendo recolectar información sobre la arquitectura del sistema, explorar configuraciones inseguras y detectar puntos débiles.

Con relación a la eficacia de las pruebas de penetración no convencionales Arfaj et al., (2022), propone la siguiente tabla para que las organizaciones tengan la oportunidad de realizar una evaluación inicial mediante la recolección de información sobre diversas vulnerabilidades obtenidas desde las bases de datos Common Vulnerabilities and Exposures (CVE) y National Vulnerability Database (NVD), mediante el uso del sistema de calificación de vulnerabilidades Common Vulnerability (CVSS). Este sistema determina la puntuación base de cada vulnerabilidad y modifica las puntuaciones teniendo en cuenta la gravedad y la tendencia a cometer errores. El sistema de puntuación CVSS mide cuán fácil es explotar

una vulnerabilidad y evalúa la gravedad del impacto que un ataque de seguridad podría tener en un sistema determinado.

La puntuación de las debilidades se categoriza en tres grupos: Alto (7.0 a 10.0), Medio (4.0 a 6.9) y Bajo (0.0 a 3.9). Esta categorización entiende el riesgo al que se exponen las organizaciones, permitiéndole el análisis entre las vulnerabilidades y la priorización de la respuesta orientando sus esfuerzos de mitigación de forma más eficaz.

En un contexto de ciberseguridad donde los recursos son escasos Chawla et al., (2019), reconoce qué vulnerabilidades necesitan atención inmediata puede ser la distinción entre una vulnerabilidad y una defensa eficaz. Finalmente, la meta es generar un ambiente más seguro, y esta tabla se transforma en un instrumento imprescindible en el arsenal de cualquier experto en seguridad.

Tabla 1: Estándar CVSS

PUNTUACIONES RESULTANTES CON TIPO DE VULNERABILIDAD					
Id de vulnerabilidad	**CVSS BASE**	**Memoria intermedia (BUFFER)**	**Condición de carrera**	**Entrada no valida**	**Autenticación**
CVE-2017-18015	4.3	5.3	4.8	4.4	4.1
CVE-2017-9964	5.8	6.5	6.1	5.6	5.3
CVE-2018-3814	6.5	7.2	6.7	6.2	6.0
CVE-2017-100432	6.0	6.8	6.3	5.8	5.6
CVE-2018-6476	1.0	10	9.5	9.1	8.8
CVE-2018-4837	5.0	5.8	5.3	4.9	4.6
CVE-2018-6205	6.1	7.0	6.5	6.1	5.8

Fuente: (Chawla et al., 2019)

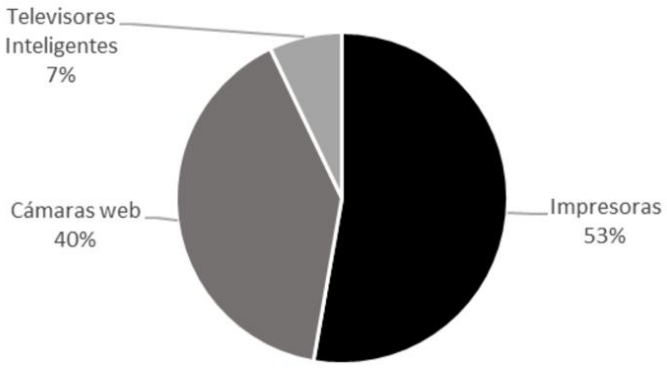
Este caso de estudio se lleva a cabo en la universidad STMIK Bina Patria, en el que se realizó una auditoría a 3 sitios web los cuales son utilizados por la comunidad académica. Se empleó la herramienta Nessus en su versión 8.13.1, la cual permite escanear vulnerabilidades de seguridad. Para ello se le indicaron al software Nessus las URLs de los sitios web a analizar, a lo que procedió a realizar diferentes tipos de escaneo. Nessus utiliza una combinación de sondas remotas (TCP/IP, SMB, HTTP, NTP, SNMP, etc) para recolectar información de los sistemas escaneados y con base en los resultados, Nessus determina posibles vulnerabilidades en los sitios web y las clasifica por nivel de severidad (Crítico, Alto, Medio, Bajo e Información), basado en la escala CVSSv2. CVSS (Common Vulnerability Scoring System por sus siglas en inglés) es un estándar para calificar la severidad de las vulnerabilidades informáticas (Chawla et al., 2019).

Para cada sitio web escaneado, Nessus generó un reporte detallando las vulnerabilidades encontradas y su correspondiente nivel CVSSv2. En el caso de web1 se encontraron 14 vulnerabilidades. Para web2, se hallaron 8 vulnerabilidades, siendo la mayoría (22%) de nivel medio. Finalmente, para web3 hubo 13 vulnerabilidades detectadas (Muhammad, Kapti, Yusnanto, Patria, & Managenent, 2020).

2.4 Caso de estudio con dispositivos IoT alojados en un servicio web: Ejemplo de fase de explotación y mantenimiento del acceso

Akhilesh et al. (2022) llevaron a cabo un estudio con 20.237 dispositivos IoT, de los cuales el 52,75 % correspondían a impresoras, el 40,16 % a cámaras web y el 7,1 % a televisores inteligentes. En este trabajo, se implementó un caso de prueba de penetración web simulando ataques reales para verificar la seguridad y evaluar el impacto potencial de las vulnerabilidades de los sistemas web asociados a la infraestructura en cuestión.

Gráfico 1: Representación de la muestra de elementos de hardware del estudio



Fuente: (Arreaga et al., 2023)

Muchos dispositivos IoT tales como sensores inteligentes, webcams, e incluso sistemas de cómputo integrados como Raspberry Pi presentan muchas vulnerabilidades que permiten a los atacantes tener acceso a estos dispositivos, o en su defecto causarles algún daño, afectando su operación o atacando la arquitectura en la que están localizados. Akhilesh et al. (2022) explica que algunas de estas vulnerabilidades pueden materializarse en forma de ataques de phishing, denegación de servicios o compromisos en los puertos de comunicación, permitiendo así el acceso a información confidencial.

Recurrentemente se realizan ciberataques dirigidos a aparatos electrónicos de consumo o artículos domésticos, como los ataques contra cámaras y refrigeradores inteligentes. Estos dispositivos suelen tener normas de seguridad deficientes lo que los hace más accesibles a los ataques informáticos (Heiding et al., 2022).

En este laboratorio se realizaron pruebas de vulnerabilidad en un dispositivo IoT Raspberry Pi utilizando herramientas como OracleVMVirtualBox, Kali Linux, Raspbian, RaspberryPi3modelBv1.2, AdvancedIPScanner, VNC Viewer, Wireshark, Metasploit, and Ettercap. Se llevaron a cabo tres tipos de ataques:

- Ataque Man-in-the-Middle (MITM):

Se utilizaron máquinas virtuales con Kali Linux (atacante), Ubuntu (servidor) y un dispositivo Raspberry Pi (víctima). Se emplearon las herramientas Ettercap y Wireshark para escanear hosts en la red, envenenar ARP y capturar el tráfico de red. Se pudo ver información como nombres de usuario y contraseñas sin encriptar enviadas desde el navegador de Raspberry Pi (Tabassum, Sharma, & Mohanan, 2021).

- Ataque de Puerta Trasera (Backdoor):

Se creó un payload con Metasploit en Kali Linux y se ejecutó en la víctima (Raspberry Pi y Windows 10 en máquinas virtuales separadas). Esto permitió al atacante controlar de forma remota la máquina víctima, obtener información del sistema y, en Windows 10, tomar capturas de pantalla y navegar por archivos.

- Ataque de Denegación de Servicio (DoS):

Se configuró un servidor web con Apache, MariaDB y PHP en Raspberry Pi y en una máquina virtual con Windows 10. Luego, se utilizó el módulo synflood de Metasploit desde Kali Linux para lanzar un ataque DoS contra el servidor, lo que provocó retrasos significativos en la carga de páginas web.

La herramienta Metasploit Framework permite ejecutar comandos para generar un payload o archivo malicioso que permite obtener acceso remoto a la máquina víctima.

Los comandos son los siguientes:

1. msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=[AttackerIP] LPORT=[ListeningPort] -f elf > shell.elf

Este comando genera un payload Meterpreter reverse_tcp para sistemas Linux de 32 bits. Los parámetros son:

- p: especifica el payload a usar (linux/x86/meterpreter/reverse_tcp)
- LHOST: la dirección IP del atacante a la que se conectará la víctima
- LPORT: el puerto en el que el atacante estará escuchando

- f elf: formato del payload generado (ejecutable ELF para Linux)
- shell.elf: nombre del archivo de payload generado

2. msfvenom -p windows/meterpreter/reverse_tcp LHOST=[AttackerIP] LPORT=[ListeningPort] -f exe > shell.exe

Este comando es similar, pero genera un payload Meterpreter reverse_tcp para Windows. Los parámetros son:

- p: especifica el payload (windows/meterpreter/reverse_tcp)
- LHOST y LPORT: igual que el caso anterior
- f exe: formato del payload generado (ejecutable .exe para Windows)
- shell.exe: nombre del archivo de payload

Luego, estos payloads generados (shell.elf o shell.exe) deben ejecutarse en la máquina víctima después de haber desactivado el Firewall de Windows Defender y desactivar todas las opciones de protección contra virus y amenazas, lo que permitirá al atacante obtener una sesión Meterpreter remota y controlar esa máquina a través de Metasploit, permitiendo realizar acciones como tomar capturas de pantalla, y navegar por medio de directorios, creando, eliminando, y modificando archivos por medio de comandos. (Arreaga et al., 2023)

2.5 Entorno controlado para pentesting empleando sistemas web

Por su parte, se han desarrollado múltiples datasets que son esenciales para el entrenamiento, prueba y comparación de algoritmos y sistemas de detección de intrusiones, permitiendo a los investigadores evaluar la eficacia de sus métodos en diferentes escenarios y tipos de ataques. Los pentesters participan en la creación de estos datasets para crear un dataset más realista y completo, que permite una evaluación más precisa de los sistemas de detección de intrusiones frente a amenazas humanas reales, además de las automatizadas.

En estos dataset se modela una red y se implementan servicios similares al de un servidor web, servidor de correo, y una página de e-commerce, en donde los profesionales de pentesting pueden atacar esta infraestructura, y un determinado grupo intentaria defender la misma.

Entre algunos de estos datasets y sus características podemos resaltar los siguientes expuestos por Lawrence et al. (2022):

KDD99 y NSL-KDD:

- Establecieron una taxonomía de ataques: DoS, R2L, U2R, Probe y Normal.
- Se usaron ampliamente para entrenar y probar algoritmos de aprendizaje automático para NIDS

CTU-13:

- Contiene tráfico de botnets.
- Se usó para desarrollar y probar métodos de detección de botnets

UNSW-NB15:

- Creado para proporcionar datos más actualizados que KDD99.
- Incluye 9 tipos de ataques generados con herramientas modernas.
- Se usa para entrenar y evaluar sistemas NIDS más recientes.

WID (Aegean Wi-Fi Intrusion Dataset):

- Específico para tráfico Wi-Fi (802.11).
- Contiene tráfico normal y de ataques en redes inalámbricas.

CICIDS2017:

- Incluye una variedad de ataques modernos como inyección SQL, XSS, fuerza bruta, etc.
- Se usa para probar NIDS contra técnicas de ataque más recientes.

CUPID (Colorado University Pentesting Intrusion Dataset):

- Incluye tráfico generado por humanos (tanto normal como de ataque).
- Se usa para estudiar las diferencias entre ataques automatizados y humanos.

3. RESULTADOS Y DISCUSIÓN

Las metodologías y técnicas de penetración web varían según el enfoque, nivel de acceso al sistema objetivo y la experiencia del ejecutor, sin embargo, los métodos más usados en la actualidad son: Caja negra, caja blanca y caja gris, los cuales son esenciales en el pentesting web debido a la perspectiva única que cada uno aporta. El enfoque de caja negra es crucial para simular ataques reales desde el punto de vista de un atacante externo sin acceso previo, lo que permite identificar vulnerabilidades explotables por actores desconocidos (Kongara & Krishnama, 2023). Por su parte, el enfoque de caja blanca es indispensable para analizar en profundidad el código, la configuración y la infraestructura del sistema, detectando fallos internos que podrían ser pasados por alto. Finalmente, el enfoque de caja gris combina ambos mundos, proporcionando una visión equilibrada que refleja escenarios de amenazas realistas, maximizando la eficacia en la detección y mitigación de vulnerabilidades. Estos enfoques, utilizados estratégicamente, permiten a los pentesters realizar evaluaciones completas y adaptadas a los objetivos específicos del sistema.

En cuanto a las herramientas más utilizadas por pentesters en la actualidad, se tiene una variación dependiendo de las necesidades individuales de cada caso, no obstante, las siguientes herramientas son fuertemente mencionadas en la literatura y altamente recomendadas por la comunidad académica y la industria:

- Nessus: En el caso de estudio que se analizó, las vulnerabilidades de nivel medio estaban relacionadas con un servidor DNS débil en la configuración de los sitios web. Para concluir, estos resultados que permite este tipo de herramientas, mediante una adecuada interpretación nos resulta útil para desarrollar e implementar planes de mejora con el fin de corregir las vulnerabilidades encontradas (Chawla et al., 2019).
- Metasploit: Los ataques de Metasploit pueden prevenirse de manera más efectiva mediante medidas de seguridad tradicionales, como la instalación de parches, el uso de programas o sistemas con privilegios limitados, la restricción del acceso a redes únicamente a hosts confiables y la implementación de otros controles generales. (Tabassum, Sharma, & Mohanan, 2021).

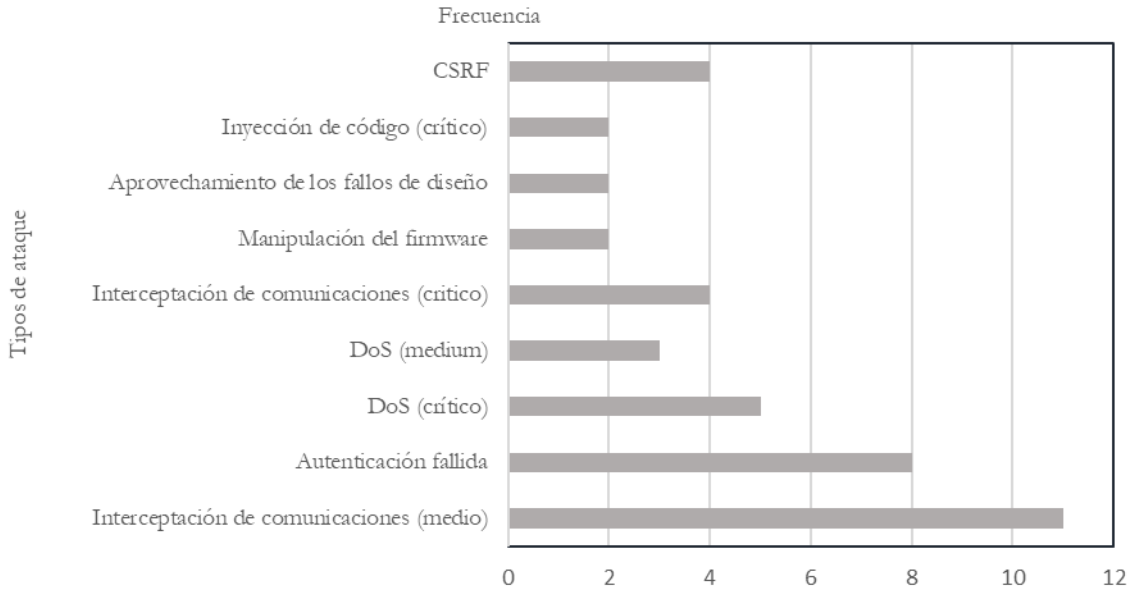
El proceso más común para llevar a cabo pruebas de penetración en servicios web incluye un proceso estructurado que consta principalmente de cinco etapas: planificación y reconocimiento, escaneo, explotación, mantenimiento del acceso y generación de informes.

Los resultados obtenidos destacan la importancia de combinar técnicas manuales y automatizada para realizar pruebas de penetración. Destacando también la importancia de la revisión regular y constante de la infraestructura tecnológica, subrayando especialmente el caso de los sitios web académicos, se puede evidenciar la importancia de configuraciones de seguridad fuertes, incluso en servicios que pueden no ser tan relevantes para la operación diaria.

A su vez, esta investigación ha logrado evidenciar las diferentes fallas de seguridad presentes en los sistemas y dispositivos de IoT, evidenciando una tendencia por parte de las compañías manufactureras de no aplicar políticas de seguridad importantes y el afán de interconectar todos los dispositivos en la red, exponiendo la seguridad e integridad de esta. En este contexto, la generación de payloads para ataques web ha experimentado una evolución constante, con un énfasis creciente en la evasión de antivirus. Si bien herramientas como Msfvenom ofrecen una funcionalidad básica, su capacidad para evadir las defensas de seguridad más modernas es limitada. Por otro lado, frameworks como Veil-Evasion, proporciona una capa adicional de ofuscación y cifrado, lo que dificulta significativamente la detección de los payloads por parte de los sistemas de seguridad (Marriwala et al., 2022).

El siguiente gráfico muestra los tipos de ataques que se ejecutaron y tuvieron exitos con más frecuencia, evidenciando qué vulnerabilidades pueden ser más comunes y representan un mayor riesgo. Esto permite identificar patrones de ataque y priorizar estrategias de mitigación según su impacto y frecuencia.

Gráfico 2: Representación de las vulnerabilidades más frecuentes en los sistemas IoT



Fuente: (Heiding et al., 2022)

La comparación entre Msfvenom y Veil-Evasion revela que este último ofrece una mayor flexibilidad y personalización en la generación de payloads, lo que permite a los atacantes adaptar sus ataques a las características específicas de cada objetivo. Además, Veil-Evasion incluye un conjunto de módulos y plugins que facilitan la integración con otras herramientas y la automatización de los procesos de ataque.

También está el uso de la inteligencia artificial, que al igual que en muchos otros campos, también plantea un rango de mejora en este, permitiendo anticipar fallas, encontrarlas en tiempo de ejecución y mitigarlas. De la misma manera que también plantea un riesgo para la ciberseguridad, permitiendo a los hackers encontrar y explotar las fallas de manera más rápida.

En cuanto a los servicios web, Auricchio et al. (2022) presenta la tabla 2 en donde se analiza tres métricas clave de un escáner o herramienta que se encarga de analizar el estado de seguridad:

- Precision (Precisión): Indica la proporción de resultados relevantes (detecciones correctas) entre el total de resultados positivos generados por el escáner. Un valor alto significa que pocas detecciones son falsos positivos.
- Recall (Cobertura o Sensibilidad): Mide la proporción de amenazas detectadas correctamente (verdaderos positivos) en relación con el total de amenazas existentes. Un valor alto indica que el escáner detecta la mayoría de las vulnerabilidades reales.
- F-measure: Es una métrica combinada que balancea la precisión y el recall. Se calcula como la media armónica de ambas métricas, proporcionando un único valor que resume el desempeño general.

Tabla 2. Comparación de scanner de vulnerabilidades web

Scanner	Precision	Recall	F-measure
WAPITI 3.0.4	99,54%	67,49%	80,44%
ARACHNI 1.5.1	99,92%	97,37%	98,63%
ZAP 2.10.0	99,72%	83,01%	90,60%
SKIPFISH 2.10B	99,02%	54,83%	70,58%
NESSUS Essential 8.13.1	98,92%	99,07%	99,00%
Syhunt Community 6.9.4	99,77%	99,07%	99,42%

Fuente: (Auricchio et al., 2022)

Estos resultados son fundamentales para seleccionar la herramienta más adecuada según las necesidades específicas de seguridad. Por ejemplo, un alto recall asegura que se detecten la mayoría de las vulnerabilidades, mientras que una alta precisión minimiza los falsos positivos. Herramientas como "ARACHNI 1.5.1" sobresalen al mostrar un equilibrio casi perfecto entre ambas métricas, lo que las hace ideales para escenarios donde la efectividad es crítica. En cambio, herramientas con menor recall, como WAPITI 3.0.4 o SKIPFISH 2.10B, podrían ser menos confiables en entornos de alto riesgo.

En definitiva, la ciberseguridad es un esfuerzo continuo y proactivo, donde el pentesting tiene un lugar importante, ya que su enfoque permite encontrar y corregir vulnerabilidades antes de que sean explotados por hackers, protegiendo datos sensibles y asegurando el correcto funcionamiento de los servicios. Esto resulta vital para sistemas de hospitales, sistemas de gubernamentales, sistemas de control de tráfico aéreo, entre otros, que no son exceptos de vulnerabilidades o penetraciones parte de grupos o personas malintencionadas.

4. CONCLUSIONES

En este trabajo se revisaron diversos enfoques de pruebas de penetración, destacándose los métodos de caja negra, caja blanca y caja gris como herramientas esenciales en la ciberseguridad moderna. Cada uno de estos métodos presenta ventajas específicas que, en combinación, permiten un análisis integral de la seguridad de los sistemas. Se concluye que la implementación conjunta de estos enfoques incrementa la eficacia en la detección de vulnerabilidades y fortalece la protección de los activos digitales de las organizaciones.

Asimismo, se resalta el impacto de las tendencias tecnológicas, como la inteligencia artificial, en el ámbito del pentesting. Estas tecnologías emergentes no solo optimizan el proceso mediante la automatización y la simulación avanzada de ataques, sino que también habilitan nuevos paradigmas en la identificación de amenazas, como se demuestra en los casos de estudio revisados.

Se considera que las etapas actuales del proceso de pruebas de penetración, desde el reconocimiento inicial hasta la remediación final, están adecuadamente estructuradas para abordar de forma eficiente y sistemática los desafíos de seguridad contemporáneos. Este enfoque permite a las organizaciones no solo detectar vulnerabilidades, sino también diseñar estrategias proactivas para mitigar riesgos a largo plazo.

Finalmente, se evidencia que las pruebas de penetración representan una herramienta clave para garantizar la resiliencia de los sistemas en un entorno tecnológico en constante evolución, beneficiando tanto a las empresas como a los usuarios finales.

5. REFERENCIAS

Akhilesh, R., Bills, O., Chilamkurti, N., & Chowdhury, M. J. M. (2022). Automated Penetration Testing Framework for Smart-Home-Based IoT Devices. Future Internet, 14(10). https://doi.org/10.3390/fi14100276

Alhamed, M., & Rahman, M. M. H. (2023). A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. Applied Sciences, 13(12), 6986. https://doi.org/10.3390/app13126986

Altulaihan, E. A., Alismail, A., & Frikha, M. (2023). A Survey on Web Application Penetration Testing. In Electronics (Switzerland) (Vol. 12, Issue 5). MDPI. https://doi.org/10.3390/electronics12051229

Arfaj, B. A. Bin, Mishra, S., & Alshehri, M. (2022). Efficacy of unconventional penetration testing practices. Intelligent Automation and Soft Computing, 31(1), 223–239. https://doi.org/10.32604/IASC.2022.019485

Arreaga, N. X., Enriquez, G. M., Blanc, S., & Estrada, R. (2023). Security Vulnerability Analysis for IoT Devices Raspberry Pi using PENTEST. Procedia Computer Science, 224, 223-230. https://doi.org/10.1016/j.procs.2023.09.031

Auricchio, N., Cappuccio, A., Caturano, F., Perrone, G., & Pietro Romano, S. (2022). An automated approach to Web Offensive Security. Computer Communications, 195, 248-261. https://doi.org/10.1016/j.comcom.2022.08.018

Aydos, M., Aldan, Ç., Coşkun, E., & Soydan, A. (2021). Security testing of web applications: A systematic mapping of the literature. Journal Of King Saud University - Computer And Information Sciences, 34(9), 6775-6792. https://doi.org/10.1016/j.jksuci.2021.09.018

Bhesaniya, P., & Agrawal, A. K. (2022). Web application pentesting approach & remediation. Journal of Cyber Security and Digital Forensics, 1, 11-14. https://jcsdf.nfsu.ac.in/Uploads/EJournal/1/1/(11-14)%20WEB%20APPLICATION%20PENTESTING%20APPROACH%20&%20REMEDIATION.pdf

Chawla, G., Sharma, N., & Rawal, N. K. (2019). IVSV: An improved CVSS base score mechanism with vulnerability type. International Journal of Engineering and Advanced Technology, 8(6), 4946–4950. https://doi.org/10.35940/ijeat.F9245.088619

Chowdhary, A., Jha, K., & Zhao, M. (2023). Generative Adversarial Network (GAN)-Based Autonomous Penetration Testing for Web Applications. Sensors, 23(18), 8014. https://doi.org/10.3390/s23188014

Heiding, F., Süren, E., Olegård, J., & Lagerström, R. (2022). Penetration testing of connected households. Computers & Security, 126, 103067. https://doi.org/10.1016/j.cose.2022.103067

Heiding, F., Katsikeas, S., & Lagerström, R. (2023). Research communities in cyber security vulnerability assessments: A comprehensive literature review. Computer Science Review, 48, 100551. https://doi.org/10.1016/j.cosrev.2023.100551

Kongara, D., & Krishnama, S. (2023). A process of penetration testing using various tools. Mesopotamian journal of Cybersecurity, 36(1), Article 014. https://doi.org/10.58496/MJCS/2023/014

Lawrence, H., Ezeobi, U., Tauil, O., Nosal, J., Redwood, O., Zhuang, Y., & Bloom, G. (2022). CUPID: A labeled dataset with Pentesting for evaluation of network intrusion detection. Journal Of Systems Architecture, 129, 102621. https://doi.org/10.1016/j.sysarc.2022.102621

Li, Y., & Liu, Q. (2021). A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. Energy Reports, 7, 8176-8186. https://doi.org/10.1016/j.egyr.2021.08.126

Marchand-Niño, W. R., & Santillan Ruiz, J. M. (2020). Laboratory for Vulnerability Analysis and CIS Controls on Layer 2 Switches. In Seguridad Informática. X Congreso Iberoamericano, CIBSI 2020. Universidad del Rosario. https://doi.org/10.12804/si9789587844337.18

Marriwala, N., Tripathi, C. C., Jain, S., & Mathapathi, S. (2022). Emergent Converging Technologies and Biomedical Systems. En Lecture notes in electrical engineering. https://doi.org/10.1007/978-981-16-8774-7

Muhammad, A. M., Kapti, N. D., Yusnanto, T., Patria, B. S., & Managenent, I. (2020). Campus website security vulnerability analysis using Nessus. International Journal of Computer and Information System (IJCIS), 3. https://ijcis.net/index.php/ijcis/indexx

Nagendran, K., Adithyan, A., Chethana, R., Camillus, P., & Bala Sri Varshini, K. B. (2019). Web application penetration testing. International Journal of Innovative Technology and Exploring Engineering, 8(10), 1029–1035. https://doi.org/10.35940/ijitee.J9173.0881019

Perez, J. C. C., & Ortega, A. E. (2020). Identificación de activos y ciber activos críticos en sistemas de transmisión de energía eléctrica. Tecnura, 24(65), 27-38. https://doi.org/10.14483/22487638.15388

Pinto, J. H. (2020). Web application penetration test: Proposal for a generic web application testing methodology [Tesis doctoral, ISCTE - Instituto Universitario de Lisboa]. ProQuest Dissertations & Theses Global.

https://www.proquest.com/openview/27aa8251bc5ba8d076a6fc2e9a535c7f/1?pq-origsite=gscholar&cbl=2026366&diss=y

Rodríguez, R. J., Marrone, S., Marcos, I., & Porzio, G. (2023). MOSTO: A toolkit to facilitate security auditing of ICS devices using Modbus/TCP. Computers & Security, 132, 103373. https://doi.org/10.1016/j.cose.2023.103373

Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., & Crespi, N. (2022). A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. Applied Sciences (Switzerland), 12(8). https://doi.org/10.3390/app12084077

Shi, Y., Yu, W., Zhao, Y., & Jia, Y. (2024). A Web Application Fingerprint Recognition Method Based on Machine Learning. Computer Modeling In Engineering & Sciences, 140(1), 887-906. https://doi.org/10.32604/cmes.2024.046140

Tabassum, M., Sharma, T., & Mohanan, S. (2021). Ethical hacking and penetrate testing using Kali and Metasploit Framework. International Journal of Innovation in Computational Science and Engineering, 2(1), 9–22. https://doi.org/10.353320995

Tiwari, A., Patel, J., & Sharma, P. (2021). Vulnerability Assessment and Penetration Testing Approach Towards Cloud-Based Application and Related Services. International Journal Of Scientific Research In Science Engineering And Technology, 395-403. https://doi.org/10.32628/ijsrset218346