ANÁLISIS DE LAS HERRAMIENTAS Y TÉCNICAS UTILIZADAS EN PRUEBA DE PENETRACIÓN PARA LA DETECCIÓN DE VULNERABILIDADES EN APLICACIONES WEB

AUTORES: Kleber Germiniano Marcillo Parrales[1]
Jessica Carolina Marcillo Castro[2]
María Mercedes Ortiz Hernández[3]
Edwin Antonio Mero Lino [4]

DIRECCIÓN PARA CORRESPONDENCIA: kleber.marcillo@unesum.edu.ec

Fecha de recepción: 11/10/2020

Fecha de aceptación: 24/11/2020

RESUMEN

En vista de los riesgos de ataque cibernético que se presenta en los sistemas informático, en especial en las aplicaciones web, que maneja gran cantidad de información y datos de carácter privado como, por ejemplo, publicidad comercial, páginas web turísticas, marketing digital, entro otros, una manera de prevenirlo, es actuar anticipadamente, detectando a tiempo las vulnerabilidades potenciales, que pueden ser aprovechadas por los atacantes o hacker que carecen de consideraciones éticas. Se analizan las diferentes herramientas y técnicas utilizadas en prueba de penetración para la detección de vulnerabilidades en aplicaciones web, con la finalidad de detectar las posibles amenazas, que afecten, en el correcto funcionamiento de los diferentes sistemas informáticos. La metodología aplicada fue la bibliográfica, que permitió recopilar información relevante para realizar la investigación y el método analítico-sintético que se empleó para extraer las características principales y comparar las técnicas y herramientas para la detección de vulnerabilidades en aplicaciones web, además se realizó un análisis comparativo para determinar que herramienta fue la más utilizada en los ataques cibernéticos. Se determinó, que existen muchas herramientas y técnicas que proporcionan la detección de vulnerabilidad en aplicaciones web, entre la más utilizada por los usuarios de la informática, se tiene la Qualys Guard Web Application Scanning WAS. Este análisis permitió determinar, la eficacia de las herramientas y técnicas al ser aplicadas en prueba de vulnerabilidad considerando el análisis

[1] Ingeniero Eléctrico, Magister, Docente, Universidad Estatal del Sur de Manabí, Jipijapa, Manabí, Ecuador, email**:** kleber.marcillo@unesum.edu.ec,
[2] Estudiante, Universidad Estatal del Sur de Manabí, Jipijapa, Manabí, Ecuador. email**:** marcillo-jessica0021@unesum.edu.ec
[3] Ingeniero en Sistemas, Magister, Docente, Universidad Estatal del Sur de Manabí, Jipijapa, Manabí, Ecuador, email**:** maría.ortiz@unesum.edu.ec
[4] Universidad Estatal Del Sur De Manabí, Jipijapa, Ecuador. E-mail:

presentado por diferentes autores, entre ellos la matriz de trazabilidad, información detallada de los hallazgos de seguridad detectados, que sirvió para establecer los diferentes tipos de ataque cibernéticos, que se producen en las páginas web.

PALABRAS CLAVE: análisis de vulnerabilidades; prueba de penetración; seguridad informática

ANALYSIS OF THE TOOLS AND TECHNIQUES USED IN THE PENETRATION TEST FOR THE DETECTION OF VULNERABILITIES IN WEB APPLICATIONS

ABSTRACT

In view of the risks of cyber attack that occurs in computer systems, especially in web applications, which handles a large amount of information and private data such as, for example, commercial advertising, tourist web pages, digital marketing, I enter Others, a way to prevent it, is to act early, detecting potential vulnerabilities in time, which can be exploited by attackers or hackers who lack ethical considerations. The different tools and techniques used in penetration testing to detect vulnerabilities in web applications are analyzed, in order to detect possible threats that affect the correct functioning of the different computer systems. The applied methodology was the bibliographic one, which allowed to collect relevant information to carry out the research and the analytical-synthetic method that was used to extract the main characteristics and compare the techniques and tools for the detection of vulnerabilities in web applications, in addition an analysis was carried out comparative to determine which tool was the most used in cyber attacks. It was determined that there are many tools and techniques that provide vulnerability detection in web applications, among the most used by computer users, there is the Qualys Guard Web Application Scanning WAS. This analysis made it possible to determine the effectiveness of the tools and techniques when applied in vulnerability testing, considering the analysis presented by different authors, including the traceability matrix, detailed information on the security findings detected, which served to establish the different types cyber attacks, which occur on web pages.

KEY WORDS: vulnerability analysis; penetration test; Informatic security

INTRODUCCIÓN

Este trabajo está estructurado de la siguiente manera, en la primera sección se presenta la revisión bibliográfica de las evidencias teóricas. En la segunda sección se presentan los datos comparativos de las técnicas y herramientas para la detección de vulnerabilidades en aplicaciones web y finalmente se da a conocer la propuesta y conclusión de la presente investigación.

Un sitio web es un conjunto o directorio de sitios web enlazados entre sí y alojados en un servidor configurado como hosting e identificado con un nombre de 25 dominio, que tienen como objetivo publicar información o realizar trámites, transacciones y procedimientos desde cualquier sitio remoto. Hoy en día un sitio web no es una moda, teniendo en cuenta la evolución tecnológica, se ha convertido en una necesidad para cualquier institución, empresa y hasta personas tener un sitio web propio, que permita mostrar información, productos o servicios a

usuarios, clientes y proveedores en cualquier parte del mundo sin importar horarios ni fechas. Para el funcionamiento de un sitio web es necesario contar con un servidor de nombres de dominios DNS que permita resolver los nombres de dominios e indicando la ubicación de estos. De acuerdo con los servicios contratados con el Hosting donde se aloja el sitio web, así será el rendimiento y prestaciones del sitio ante las visitas de los usuarios. Cada sitio web publicado en internet posee un nombre de dominio único, que lo identificara cada vez que sea digitado en un navegador web y así poder desplegar la información alojada en el sitio. (Coutin García, 2019)

De acuerdo con los avances tecnológicos actuales y el uso masivo del internet, ha sido necesario que tanto socios, proveedores, empleados y clientes puedan interactuar con los recursos de la compañía por medios remotos como el internet, por tal motivo ha sido necesario aplicar una serie de controles para el acceso a los sistemas de información que permitan que los datos se mantengan confiables, íntegros y disponibles en tiempo real. (Coutin García, 2019)

El análisis de vulnerabilidades es el conjunto de pruebas de seguridad, en donde un especialista ejecuta técnicas y herramientas especializadas para la detección de fallas o malas configuraciones y vulnerabilidades asociadas a los servicios y activos de TI de una organización. Este servicio se realiza con el enfoque defensivo, de manera que no se realiza la explotación de las vulnerabilidades encontradas en los activos analizados, a diferencia del servicio de Pentest, se entrega un reporte a nivel técnico y ejecutivo con la información detallada de los hallazgos de seguridad detectados, el nivel de riesgo asociado, el escenario de riesgo posibles consecuencias y las respectivas recomendaciones para la mitigación del hallazgo de seguridad. (Ortiz Castillo, 2020), (González Brito & Montesino Perurena, 2018)

Las pruebas de penetración es una práctica para poner a prueba un sistema informático, red o aplicación web para encontrar vulnerabilidades que un atacante podría explotar, las pruebas de penetración pueden ser automatizadas con aplicaciones de software. De cualquier manera, el proceso incluye la recopilación de información sobre el objetivo antes de la prueba reconocimiento, la identificación de posibles puntos de entrada, intentos de entrar ya sea virtualmente o de manera real y el reporte de los resultados.

El principal objetivo de las pruebas de penetración consiste en determinar las debilidades de seguridad. Una prueba de penetración también puede ser utilizando para probar el cumplimiento de la política de seguridad de una organización, la conciencia de seguridad de sus empleados y la capacidad de la organización para identificar y responder a los incidentes de seguridad. Cuando existe incertidumbre acerca de la eficacia de los distintos mecanismos de seguridad como los controles de los firewalls, sistemas de detección de intrusiones, monitorización de integridad de archivos, lo mejor es realizar una prueba de penetración completa. Una de las técnicas más comunes para asegurarse del nivel de efectividad, es una prueba de penetración, que es un análisis de vulnerabilidad para localizar las flaquezas individuales del sistema. (Villacís & Morocho, 2017)

DESARROLLO

En este estudio, se analizaron información bibliográfica de diferentes investigadores que utilizaron el método analítico-sintético, para conocer las características principales y obtener información relevante sobre las metodologías de pruebas de penetración en aplicaciones web, se analizaron los instrumentos de evaluación aplicados para establecer resultados mediante un análisis comparativo.

Se determinó las vulnerabilidades más frecuentes en aplicaciones web, a través del método histórico-lógico, analizando su evolución a partir de reportes desde el año 2003.

Los ataques se pueden encontrar cuatro diferentes tipos de ataques: Ataques activos, ataques pasivos, ataques a contraseñas y ataques de código malicioso. (Rodríguez Cuadros, 2018), (Yánez et al., 2017)

Los Ataques activos son realizados para causar el mayor daño posible a un sistema, mediante la obtención de los servicios, con el fin de modificar la configuración de cada uno de ellos o detener su ejecución. Los ataques activos son visibles dado que sus consecuencias son detectables a simple vista. En esta categoría se encuentras:

• Denegación de servicios.
• Bufer Overflows
• Spoofing
• MITM – Man In The Middle
• TCP/IP Hijacking
• Ingenieria social (Rodríguez Cuadros, 2018)

Los ataques pasivos no afectan directamente a la red víctima, solo escuchan lo que viaja a través de esta, recopilando la información importante, desde conversaciones hasta claves de seguridad.

Entre estos ataques se encuentran:

• Análisis de vulnerabilidades.

• Escaneo de redes y espionaje.

Los ataques de contraseña son los ataques que más se aplican por su facilidad y por la cantidad de herramientas disponibles para esto. Hay dos tipos de ataques: • Ataque de fuerza bruta. • Ataque basado en diccionario. (Rodríguez Cuadros, 2018), (Dominguez, 2019)

A través de los años los malware conocidos como Ataque de código malicioso, han sido el tipo de infección informática que más tiene reconocimiento debido a su capacidad de cambio para ser detectado, esto sumado a la capacidad que tiene para propagarse en internet a través de correos electrónicos, memorias USB, descargas desde páginas poco conocidas. La ingeniería social es la forma más activa de propagación de este tipo de infección ya que por medio de enlaces o correos electrónicos con supuestos contenidos atractivos para los usuarios y se descargan en el equipo y allí permanecen hasta ejecutarse y cumplir su objetivo.

Entre los tipos de malware se encuentran:

• Virus.

- Troyanos.
- Bombas lógicas.
- Gusanos.
- Puertas traseras. (Rodríguez Cuadros, 2018)

Los Ataque de denegación de servicios DoS aprovecha la capacidad de uso de la red para saturar un sistema con envío de solicitudes simultáneas, con el objetivo de superar la capacidad de respuesta del sitio o sistema y de esta forma detener el buen funcionamiento de este. Normalmente los DoS se aprovechan de la capacidad limitada que tienen los recursos de red, como los servidores web, cuando el número de solicitudes supera este limite el servicio se ve afectado generando respuestas lentas y desechando las demás solicitudes de los usuarios. (Rodríguez Cuadros, 2018), (Zambrano & Valencia, 2017)

El Desbordamiento de memoria es el tipo de ataque aprovecha los errores cometidos por los programadores. Estas deficiencias pueden ser explotadas por este ataque conocido como desbordamiento de memoria, este ataque envía demasiados datos al buffer con el fin de que este se sature, esta parte del sistema es un área de memoria temporal que almacena datos o instrucciones. Normalmente para ejecutar este tipo de ataques se reemplazan los datos de la memoria por otros datos que la mayoría de 25 las veces son caracteres sin ordenamiento, ocasionando que el programa deje de funcionar. (Rodríguez Cuadros, 2018)

Los Spoofing proporcionan información falsa acerca de su identidad, con el fin de ganar acceso no autorizado al sistema. El ejemplo más clásico es IP spoofing, en donde un atacante crea un paquete IP con la dirección de origen de otra máquina. (Rodríguez Cuadros, 2018), (Cornelio et al., 2012)

Los Ataque Man in The Middle Realiza sniffing a una red posicionándose en medio de la puerta de enlace y un servidor o red, esto se logra realizando un ataque al ARP (Protocolo de resolución de direcciones) que tome como puerta de enlace la máquina del atacante, para luego cambiar la Mac de la puerta de enlace por la Mac del atacante. (Rodríguez Cuadros, 2018)

Manual de la metodología abierta de testeo de seguridad (OSSTMM)

Es uno de los estándares profesionales más completos, provee de un manual con una metodología abierta del test de penetración de seguridad (Mayorga et al., 2018). La metodología está dividida en diferentes secciones:

- Sección A: Seguridad de la información.
- Sección B: Seguridad de los procesos.
- Sección C: Seguridad en las tecnologías de internet.
- Sección D: Seguridad en las comunicaciones.

• Sección E: Seguridad inalámbrica.

• Sección F: Seguridad Física. (Rodríguez Cuadros, 2018)

Dentro de las clasificaciones de ataques basados en vulnerabilidades se encuentran:

Inyección**:** Las fallas de inyección, tales como SQL, OS, LDAP, ocurren cuando datos no confidenciales son enviados a un intérprete como parte de un comando o consulta, tratando de engañar al intérprete en ejecutar comandos no intencionados o acceder datos no autorizados.

- Secuencia de Comandos en Sitios Cruzados: Las fallas XSS ocurren cada vez que una aplicación toma datos no confidenciales y los envía al navegador web sin una validación y codificación apropiada. XSS permite a los atacantes ejecutar secuencia de comandos en el navegador de la víctima los cuales pueden secuestrar las sesiones de usuario, destruir sitios web, o dirigir al usuario hacia un sitio malicioso.

- Configuración de Seguridad Incorrecta: Una buena seguridad requiere tener definidas e implementada una configuración segura para la aplicación, marcos de trabajo, servidores de aplicación, servidores web, base de datos, y plataformas. Todas estas configuraciones deben ser definidas, implementadas, y mantenidas ya que por lo general no son seguras por defecto.

- Exposición de datos sensibles: Muchas aplicaciones web no protegen adecuadamente datos sensibles tales como números de tarjetas de crédito, o credenciales de autenticación. Los datos sensibles requieren de métodos de protección adicionales tales como el cifrado de datos, así como también de precauciones especiales en un intercambio de datos con el navegador.

- Falsificación de Petición en Sitios Cruzados (CSRF): Un ataque CSRF obliga al navegador de una víctima autenticada a enviar una petición HTTP falsificado, incluyendo la sesión del usuario y cualquier otra información de autenticación incluida automáticamente, a una aplicación web vulnerable. (Saucedo, A. L. H., & Miranda, J. M. , 2016), (Mar Cornelio et al., 2020)

Se definen un conjunto de técnicas para detección de vulnerabilidades como son:

- Black-box: Es una técnica basada para descubrir vulnerabilidades en aplicaciones web, probando la aplicación desde el punto de vista del atacante (Sreenivasa & Kuman, 2012).

- White-box: Está del lado del servidor. En este tipo de enfoque se tiene acceso a información relevante de la organización (Sreenivasa & Kuman, 2012).

- Análisis estático de código (auditoria de código fuente): Es un método en el que no se requiere ejecutar el programa, este realiza un análisis de código fuente directo para determinar huecos en la seguridad (Sreenivasa & Kuman, 2012).

- Análisis dinámico de código: Se comunica con la aplicación web a través de front-end de la aplicación en orden de identificar vulnerabilidades de seguridad potenciales y debilidades en la arquitectura de la aplicación web (Sreenivasa & Kuman, 2012).

- Pruebas de penetración: Consiste en la simulación de un ataque de los maliciosos outssiders (que no tienen un medio autorizado de acceder a los sistemas de la organización) y de maliciosos insiders (que tienen algún nivel de acceso autorizado). El proceso implica un análisis activo del sistema en busca de posibles vulnerabilidades que podrían resultar de configuración deficiente o inadecuada del sistema, fallos de hardware o software, ya sea conocidos y desconocidos, o fallos operativos en proceso o contramedidas técnicas (Thompson, 2005).

- Pruebas pasivas: Las pruebas pasivas están diseñadas para el análisis del tráfico de telecomunicaciones. Permite detectar fallas y defectos de seguridad mediante el examen de los paquetes capturados (livetrafficor log files) (Mammar, Cavalli, & Jimenez, 2011).

- Pruebas activas: Utiliza un programador de subprocesos asignados al azar para verificar si las advertencias comunicadas por un análisis predictivo de programa son errores reales (Xiao-song Zhang, 2008).

- Fuzz testing (pruebas de caja negra): Consiste en estimular el sistema bajo prueba, utilizando datos aleatorios o mutados queridos, con el fin de detectar comportamientos no deseados como violación de confidencialidad (Xiao-song Zhang, 2008). (Saucedo, A. L. H., & Miranda, J. M. , 2016)

Las principales herramientas para la detección de vulnerabilidades se encuentran:

- QualysGuard Web Application Scanning WAS: Es una herramienta en la nube que permite realizar pruebas funcionales con selenium para aplicaciones web, además de pruebas de penetración. Permite encontrar vulnerabilidades del top 10 de OWASP (Qualys, 2014).

- WebSite Security Audit- WSSA: Permite examinar páginas web, aplicaciones y servidores web para encontrar vulnerabilidades de seguridad. Realiza pruebas de vulnerabilidades de código conocidas como: SQL Injection, XSS (Cross Site Scripting), entre otras (BeyondSecurity, 2014).

- Retina Web Security Scanner: Es una solución de escaneo de sitios web, aplicaciones web complejas para hacer frente a las vulnerabilidades de aplicaciones. Prioriza las vulnerabilidades por su nivel de riesgo (Beyontrust, 2014).

- WEBAPP 360: Enterprise Class web application scanning: Evalúa de manera completa la infraestructura de aplicaciones web, incluyendo aplicaciones web, sistemas operativos subyacentes y aplicaciones subyacentes en entorno de producción. Utiliza el Top 10 de OWASP para cerrar las brechas de seguridad en aplicaciones web (Tripwire, 2014).

- Frame-C: Es un software Open Source que permite analizar código fuente escrito en C. Reúne varias técnicas de análisis estático en una sola herramienta. (Frama-C, 2014).

• Parasoft C/C++ Test: Es una solución de pruebas para aplicaciones basadas en C y C++. Ayuda a desarrolladores a prevenir y eliminar defectos. Ayuda a eliminar problemas de seguridad, además vigila el cumplimiento de OWASP Top 10, CWE/SANS, FDA, entre otros (Parasoft, 2014).

• Fortify Static Code Analyzer: Proporciona análisis de código estático automatizado para ayudar a los desarrolladores a eliminar las vulnerabilidades y crear software de seguridad. Analiza el código fuente, identifica las causas originarias de las vulnerabilidades de la seguridad del software y correlaciona y prioriza los resultados (HP, 2014).

• MCAfee Vulnerability Manager: Realiza monitorización activa y pasiva, además de realizar pruebas de penetración. Permite conocer los puntos en los que se debe centrar los esfuerzos de programación. Cubre las categorías de OWASP top 10 y CWE-25 (McAfee, 2014).

• Nessus Vulnerability Scanner: Permite realizar escaneo de vulnerabilidades en servidores web, servicios web, además de las vulnerabilidades de OWASP. Además de verificar la configuración errónea del sistema y parches faltantes. Muestra informes personalizados en formato XML, CVS, PDF nativo y HTML (Tenable, 2014).

• Nexpose Vulnerability Manager: Es una solución de gestión de vulnerabilidades que combina la evaluación de vulnerabilidades y controles, la validación de vulnerabilidades y la planificación de remediación. Maneja estándares de riesgo, vulnerabilidades y gestión de la configuración como PCI DSS, NERC CIP, FISMA, entre otros (Rapid7, 2014).

• Whatweb: Identifica el sitio web, reconoce tecnologías web, incluyendo los sistemas de gestión de contenidos (CMS por sus siglas en inglés), plataformas de blog, bibliotecas de JavaScript, servidores web. También identifica los números de versiones de correo electrónico, errores de SQL y más (MorningStartSecurity, 2014). (Saucedo, A. L. H., & Miranda, J. M. , 2016)

Fingerprint Web Server (OTG-INFO-002): Es una tarea critica para testear la penetración en un servidor web. Sabiendo la versión y el tipo de servidor web es posible conocer sus vulnerabilidades. Para conocer la versión y el tipo de servidor basta con utilizar la herramienta Netcat [6] (Delgado Benayas, 2019)

Enumerate Applications on Webserver (OTG-INFO-004): Con esta prueba se pretende averiguar el número de aplicaciones que se están ejecutando en un servidor web. En este caso sabemos que solo estamos ejecutando una aplicación en nuestro servidor apache, no obstante, podemos escanear fácilmente, mediante la herramienta Nmap [7] (Delgado Benayas, 2019)

Review webpage comments and metadata for information leakage (OTGINFO-005: Es importante revisar el código fuente en busca de comentarios. Es habitual que los desarrolladores pongan comentarios detallados en el código HTML. Hay que revisar estos comentarios ya que podrían revelar información a posibles atacantes. (Delgado Benayas, 2019)

Identify application entry points (OTG-INFO-006): El objetivo de esta prueba es entender cómo se realizan las peticiones y cómo son las respuestas que la petición realiza. (Delgado Benayas, 2019)

Map execution paths through appliacation (OTG-INFO-007): El objetivo de este apartado es mapear la aplicación y entender su flujo de trabajo. Es muy complicado realizar este estudio examinando todas las rutas de código, esto consumiría mucho tiempo. Para realizar este análisis es necesario alguna herramienta de spidering que descubra nuevas rutas. (Delgado Benayas, 2019)

Los principales ataques basados en vulnerabilidades y las técnicas y herramientas que son utilizadas actualmente para detectar vulnerabilidades en aplicaciones web. Los resultados que se obtuvieron son mostrados a continuación:

Ataques basados en vulnerabilidades.
Inyección.
Secuencia de Comandos en Sitios Cruzados.
Configuración de Seguridad Incorrecta.
Exposición de datos sensibles.
Falsificación de Petición en Sitios Cruzados (CSRF).

Técnicas para detección de vulnerabilidades.
Black-box.
White-box.
Análisis estático de código (auditoria de código fuente).
Análisis dinámico de código.
Pruebas de penetración.
Pruebas pasivas.
Pruebas activas.
Fuzz testing (pruebas de caja negra).

Herramientas para la detección de vulnerabilidades.
Qualys Guard Web Application Scanning WAS.
Web Site Security Audit- WSSA.
Retina Web Security Scanner.
WEBAPP 360: Enterprise Class web application scanning.
Frame-C.
Parasoft C/C++ Test.
Fortify Static Code Analyzer.
MCAfee Vulnerability Manager.
Nessus Vulnerability Scanner.
Nexpose Vulnerability Manager.
Whatweb. (Saucedo, 2015)

CONCLUSIONES

Se concluye que existen diferentes tipos de ataques basados en vulnerabilidades, que afectan a los sistemas informáticos, pero también se da a conocer las técnicas y herramientas que proporcionan la detección de problemas de seguridad y mantener protegida la información de los sistemas.

Este análisis de sistemas de seguridad permitió a determinadas empresa, cliente, desarrollar estrategias para afrontar las vulnerabilidades.

REFERENCIAS BIBLIOGRÁFICAS

Coutin García, C. A. (2019). *Análisis de vulnerabilidades mediante pruebas de penetración avanzada Pentesting al sitio web oficial de la Alcaldía del municipio de Quibdó–Chocó.*

Delgado Benayas, V. (2019). Análisis de marcos de pruebas de seguridad para aplicación web y desarrollo de material didáctico (Bachelor's thesis). Madrid.

Ortiz Castillo, A. M. ((2020)). Introducción a las pruebas de penetración.

Rodríguez Cuadros, O. A. (2018). Diseño de manual básico de pruebas de hacking ético: Escaneo de red, de vulnerabilidades y ataque. Bucaramanga.

Saucedo, A. L. H., & Miranda, J. M. . (2016). Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones web. ReCIBE, 4(1). *ReCIBE, *, 4(1).

Cornelio, O. M., Moreno, N. C., Puig, P. M., & Hernández, R. C. J. (2012). Aplicación informática para el control energético de la tecnología utilizando herramienta de monitoreo de red Nmap. Revista Cubana de Ciencias Informáticas, 6(2), 1-10. https://www.redalyc.org/pdf/3783/378343676002.pdf

Dominguez, A. H. (2019). Sistema para la detección de ataques PHISHING utilizando correo electrónico. Telemática, 17(2), 60-70. https://revistatelematica.cujae.edu.cu/index.php/tele/article/download/304/280

González Brito, H. R., & Montesino Perurena, R. (2018). Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web. Revista Cubana de Ciencias Informáticas, 12(4), 52-65. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000400005

Mar Cornelio, O., Santana Ching, I., & Gulín González, J. (2020). Operador por selección para la agregación de información en Mapa Cognitivo Difuso. Revista Cubana de Ciencias Informáticas, 14(1), 20-39. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992020000100020

Mayorga, A. M., Solarte, S. P., & Donado, S. A. (2018). Pentesting sobre aplicaciones web basado en la metodología OWASP utilizando un cluster conformado por dispositivos SBC de bajo costo. Revista Ibérica de Sistemas e Tecnologias de Informação(E16), 1-14.

Villacís, G. V., & Morocho, R. A. R. (2017). Vulnerabilidades y amenazas a los servicios web de la intranet de la Universidad Técnica de Babahoyo. 3c Tecnología: glosas de innovación aplicadas a la pyme, 6(1), 53-66. https://dialnet.unirioja.es/descarga/articulo/6031033.pdf

Yánez, H., Barahona, A., Naranjo, P., Fassler, M., & García, C. (2017). Detección de vulnerabilidades en aplicaciones que funcionan sobre el sistema operativo Android, mediante el desarrollo de una aplicación tecnológica. ESPACIOS VOL, 39, 1-17. http://www.revistaespacios.com/a18v39n11/a18v39n11p07.pdf

Zambrano, S. M. Q., & Valencia, D. G. M. (2017). Seguridad en informática: consideraciones. Dominio de las Ciencias, 3(3), 676-688. https://dialnet.unirioja.es/descarga/articulo/6137824.pdf