



Temática: Seguridad Informática

Riesgos de Seguridad en Pruebas de Penetración Web

Security Risks in Web Penetration Testing

M. Sc. Henry Raúl González Brito ^{1*}, Dr. C. Raydel Montesino Perurena ², M. Sc. Dainys Gainza Reyes ³

¹ Subdirector del Centro de Telemática. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½, reparto Torrens, municipio Boyeros, La Habana, Cuba. henryraul@uci.cu

² Vicerrector Primero. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½, reparto Torrens, municipio Boyeros, La Habana, Cuba. raydelmp@uci.cu

³ Metodóloga. Dirección de Educación de Posgrado. Universidad de las Ciencias Informáticas. Carretera a San Antonio de los Baños, Km 2 ½, reparto Torrens, municipio Boyeros, La Habana, Cuba. dgainza@uci.cu

* Autor para correspondencia: henryraul@uci.cu

Resumen

En el presente trabajo se realiza una sistematización de los principales riesgos de seguridad que pueden estar asociados a las pruebas de penetración en aplicaciones web. Para la realización del estudio se consultaron fuentes bibliográficas y reportes de un alto nivel científico y técnico. Se identificaron y describieron 31 riesgos clasificados en dos grupos: los asociados a daños directos a la confidencialidad, integridad y disponibilidad de la información de la aplicación web y aquellos relacionados con la realización de una prueba de penetración deficiente y cuyos resultados parciales también afectan de manera indirecta la seguridad web, estos últimos fueron dividido en cuanto a riesgos de alcance y tiempo, infraestructura tecnológica y personal. Para el tratamiento de los riesgos descritos, se brinda un conjunto de 14 recomendaciones bases para la conformación de una estrategia de mitigación en función de los escenarios de pruebas. Se particulariza también en los modos de aplicación de las herramientas automatizadas de



evaluación de vulnerabilidades para limitar los daños en las aplicaciones web. Los resultados alcanzados tienen una alta pertinencia dada por la necesidad de los implicados en los procesos de pruebas de penetración de contar con una base de partida conceptual que favorezca el tratamiento de riesgos y contextualice mejor las decisiones tomados en función de solucionar las vulnerabilidades de seguridad halladas a través de este tipo de evaluación de seguridad.

Palabras clave: aplicaciones web; mitigación de riesgos; pruebas de penetración; riesgos de seguridad; seguridad web

Abstract

This paper systematizes the main security risks that may be associated with penetration testing in web applications. Bibliographic sources and reports of a high scientific and technical level were consulted for the study. Thirty-one risks were identified and described, classified into two groups: those associated with direct damage to the confidentiality, integrity and availability of web application information and those related to the performance of a deficient penetration test and whose partial results also indirectly affect the security of web portals, the latter were divided into risks of scope and time, technological infrastructure and personnel. For the treatment of the described risks, a set of 14 basic recommendations is provided for the conformation of a mitigation strategy according to the existing test scenarios. It also focuses on how to apply automated vulnerability assessment tools to limit damage to web applications. The results achieved are highly relevant given the need for those involved in penetration testing processes to have a conceptual starting point that favors the treatment of risks and better contextualizes the decisions taken in order to solve the security vulnerabilities found through this type of security assessment.

Keywords: penetration testing; risk mitigation; security risks; web applications; web security



Introducción

Las aplicaciones web representan una parte importante de los procesos de informatización y transformación digital. En la actualidad, su utilización abarca desde portales web desarrollados a la medida, basados en sistemas de gestión de contenidos de tipo general o especializados, redes sociales, hasta su funcionamiento en forma de paneles de administración de sensores industriales (Flaus, 2019), dispositivos de Internet de las Cosas (Anisetti et al., 2019), Computación en la Nube (Swathy Akshaya & Padmavathi, 2019) y API (Interfaz de Programación de Aplicaciones) para suministrar contenidos a dispositivos móviles (Papadopoulos et al., 2017).

La amplia difusión de las aplicaciones web, unido al valor de la información que gestionan, la han convertido en un objetivo permanente de los ciberataques desde Internet (Bartoli et al., 2018; Jamil, Asif, Ashraf, Mehmood, & Mustafa, 2018; Patel, 2019). Los reportes periódicos de organismos internacionales y de compañías líderes en el campo de la ciberseguridad muestran un crecimiento sostenido de incidentes de seguridad en los que se ven involucradas este tipo de aplicación (Kaspersky, 2020; Nguyen, Lin, & Hwang, 2019; Sucuri.net, 2020; Telefónica, 2020). Por ejemplo, los expertos de Positive Technologies encontraron en el 2019 que la mitad de las aplicaciones web tenían vulnerabilidades de alto riesgo y el 82% de las vulnerabilidades se localizaban en el código fuente (Positive_Technologies, 2020). Estos resultados fueron confirmados por Acunetix, la cual encontró que el 46% de las aplicaciones web contenían vulnerabilidades de alto riesgo y que el 87% del total contenían vulnerabilidades de riesgo medio (Acunetix, 2019). Este alto porcentaje de errores en el código fuente sugiere que las revisiones realizadas no son suficientes para detectar vulnerabilidades durante el desarrollo de software (Bishop & Rowland, 2019; Horton, 2020; Mohammed, Niazi, Alshayeb, & Mahmood, 2017; Muniz et al., 2018; Venson, Guo, Yan, & Boehm, 2019).

Para identificar las vulnerabilidades y fortalecer la seguridad de las aplicaciones web se aplican diversos controles de seguridad (Montesino Perurena, Baluja García, & Porvén Rubier, 2013) dentro de los cuales se encuentran las pruebas de penetración (Casola, De Benedictis, Rak, & Villano, 2018; Haber & Hibbert, 2018). Estas pruebas o evaluaciones de seguridad consisten en la recreación de las posibles acciones de un adversario en sistemas informáticos y redes de datos con el objetivo de comprobar si es posible evadir sus defensas y acceder a su estructura interna y datos almacenados (Cuzme-Rodríguez, León-Gudiño, Suárez-Zambrano, & Domínguez-Limaico, 2019; Rahalkar, 2016).

La aplicación de las pruebas de penetración brinda ventajas para la organización como: la concienciación sobre los problemas de ciberseguridad existentes, la comprobación de las capacidades para la detección de intrusiones y el



apoyo a la alta gerencia en los procesos de toma de decisiones en este campo (Kumar & Thagadikgora, 2019). De este modo, puede determinarse el nivel de tolerancia de los sistemas ante los patrones de ataque del mundo real, la barrera de complejidad que un adversario necesita superar para ello, y las medidas adicionales que se deben emplear para fortalecer la seguridad de la infraestructura tecnológica.

A pesar de ser una práctica reconocida en el campo de la ciberseguridad, la aplicación de las pruebas de penetración tiene asociadas un grupo de riesgos que pueden afectar tanto a los sistemas objetivos como el propio proceso de evaluación, lo que puede conllevar afectaciones debido a posibles daños provocados a la aplicación web o la obtención de resultados deficientes para la toma de decisiones en esta área.

Teniendo en cuenta lo anterior, en las secciones del presente trabajo se enuncian los principales elementos de las pruebas de penetración y se describen los entornos de despliegue en los que comúnmente se ejecutan. A partir de ello se formalizan los principales riesgos de seguridad que pueden afectar directamente a las aplicaciones web y también los que pueden dificultar el proceso de prueba de penetración. Se analiza la probabilidad de ocurrencia de los riesgos en los entornos planteados y finalmente se brindan estrategias para la mitigación de estos. De este modo, los implicados en procesos de pruebas de penetración podrán disponer de una base de partida suficiente para favorecer el tratamiento de riesgos y contextualizar mejor la toma de decisiones en función de solucionar las vulnerabilidades de seguridad halladas a través de este tipo de evaluación de seguridad.

Materiales y métodos o Metodología computacional

Para la realización del estudio se establecieron tres preguntas de investigación:

1. ¿Las pruebas de penetración pueden afectar la seguridad de las aplicaciones web?
2. ¿Cuáles pueden ser los riesgos de seguridad que pueden estar presentes durante una prueba de penetración web?
3. ¿Qué elementos pueden minimizar la ocurrencia de los riesgos de seguridad durante una prueba de penetración web?

Para la realización de la investigación se consultaron diferentes fuentes de información, principalmente memorias de conferencias, simposios y artículos de revistas indexadas en bases de datos referenciadas tales



como ACM Digital Library, IEEE Xplore, Scopus y Springer Link, así como reportes de seguridad de reconocidas compañías en el campo de la ciberseguridad a nivel mundial.

Para la selección de las fuentes bibliográficas se utilizaron diferentes combinaciones de las siguientes cadenas de búsqueda:

- Pruebas de penetración: penetration test, pentesting, ethical hacking, ethical hacker, offensive security
- Web: web, website
- Evaluación de vulnerabilidades: vulnerability assessment, vulnerabilities testing, security assessment, security testing
- Problemas: risks, issues, problems

A partir de estas búsquedas se aplicaron criterios de selectividad para utilizar solo documentos relevantes para la investigación que describan riesgos y problemas de seguridad asociados a las pruebas de penetración.

Resultados y discusión

Pruebas de Penetración

Las pruebas de penetración constituyen un proceso realizado por especialistas de seguridad para garantizar que los sistemas, activos, servicios y otros elementos en redes de datos sean inmunes a los distintos tipos de ciberataques a los que pueden estar expuestos (Alsmadi, 2019). Es por ello que deben simular las acciones típicas que puede llevar a cabo un adversario para comprometer los sistemas (Kettani & Wainwright, 2019; Saha, Das, Kumar, Biswas, & Saha, 2020). En el caso de productos de software que se encuentran en desarrollo, las pruebas de penetración se integran al proceso de validación de los requisitos de seguridad que deba tener el producto de software terminado (Mehta, Raj, & Singh, 2018; Schmittner, Griessnig, & Ma, 2018).

Por lo general, las pruebas de penetración están compuestas por una fase de planificación donde se determinan los objetivos a alcanzar y se crean las condiciones técnicas y organizativas necesarias. A continuación, se desarrolla la fase de descubrimiento de vulnerabilidades mediante el escaneo y la recopilación de información sobre los sistemas.



Posteriormente en la fase de ejecución se realiza la comprobación de las vulnerabilidades previamente descubiertas, incluyendo la explotación activa de estas. Si la evaluación de una vulnerabilidad resulta positiva, se enumeran los aspectos que la distinguen y las posibles medidas de mitigación. Por último, en la fase de documentación se emite un reporte con la caracterización de las vulnerabilidades encontradas, los riesgos que están representando para la organización y posibles vías para su solución (Murthy & Shilpa, 2018). Para comprobar cuestiones específicas se ejecutan diversas pruebas de seguridad (Felderer et al., 2016), cuyos resultados luego se integran y complementan en el proceso de pruebas de penetración.

Los especialistas se apoyan en diferentes tipos de herramientas de seguridad para llevar a cabo las pruebas de penetración (Al-Matari, Helal, Mazen, & Elhennawy, 2018; Brohi, Butt, & Zhang, 2019; Wang & Yang, 2017). Estas pueden tener diferentes alcances y con determinados niveles de automatización en su ejecución. Por ejemplo, en el caso de las aplicaciones web OWASP ZAP, Acunetix y Burp Suite, son escaneadores de seguridad de carácter general, con alto grado de automatización y pueden utilizarse en cualquier aplicación, independientemente de su tecnología (Thai & Hieu, 2019; Touseef et al., 2019), sin embargo, WPScan y JoomScan son escaneadores especializados ya que solo comprueban vulnerabilidades en los sistemas de gestión de contenidos WordPress y Joomla respectivamente (Alghofaili, 2018; Tetskyi, Kharchenko, & Uzun, 2018). También existen otras herramientas más específicas como SQLMap para estudiar las inyecciones SQL o BeEF para la explotación de vulnerabilidades en navegadores web (Ojagbule, Wimmer, & Haddad, 2018; Rodríguez, Torres, Flores, & Benavides, 2020).

Las fases, actividades, uso de herramientas y pruebas de seguridad se han formalizado en diferentes metodologías como NIST SP 800-115 (Technical Guide to Information Security Testing and Assessment) (Stouffer, Falco, & Scarfone, 2008), ISSAF (Information System Security Assessment Framework) (Rathore et al., 2006), OSSTMM (Open Source Security Testing Methodology Manual) (Barceló & Herzog, 2010), OWASP (Open Web Application Security Project Testing Guide) (Meucci & Muller, 2014) o PTES (Penetration Testing Execution Standard) (PTES, 2017). Aunque todas contienen fases para evaluar la seguridad en las aplicaciones web, solo la metodología de OWASP está especializada en este campo.

Cuando las metodologías de software se centran en productos de software y sobre todo en aplicaciones web, pueden encontrarse básicamente tres entornos de despliegue con características organizativas y tecnológicas diferenciadoras, a continuación, se describen los mismos.



Entorno de desarrollo

Se caracteriza por contar con las herramientas, servidores y servicios propios del proceso de desarrollo de software para la realización de la programación, integración y pruebas correspondientes. El producto de software no está terminado y por tanto no se aplican mecanismos de seguridad para proteger su ejecución y desempeño. El código fuente de la aplicación web sufre cambios continuos, ya sea por el proceso de codificación como por las probables modificaciones del alcance del producto acordado. Esta infraestructura, por norma general, no está diseñada para soportar cargas reales de uso intensivo. En ella no es común realizar una prueba de penetración completa, sin embargo, debe tratarse de aplicar el mayor número de pruebas de seguridad posible, enfocadas en detectar y corregir desde una fase temprana las posibles vulnerabilidades existentes, pero con un alcance unitario y muy específico por funcionalidades, componentes y subsistemas terminados.

Entorno de Prueba

Se caracteriza por contar con una infraestructura tecnológica que simula las condiciones de un despliegue en producción y presta especial atención a las medidas de configuración segura del servidor web, base de datos y mecanismos de seguridad como cortafuegos y detectores de intrusiones. Por tanto, ha concluido el desarrollo de la aplicación web y se está probando la versión que será liberada. Es el entorno ideal para hacer las pruebas de penetración en aplicaciones web debido a que un daño en las configuraciones, datos o código no debería afectar las operaciones de la organización. Sin embargo, es necesario señalar que no siempre es posible reproducir todas las situaciones reales que pueden presentarse debido a restricciones económicas para disponer del equipamiento y la integración con otros softwares necesarios.

Entorno de Producción

Es un entorno real de despliegue donde la aplicación web forma parte de un ecosistema de software (Jansen, Cusumano, & Popp, 2019) y está soportando procesos operacionales de la organización y cuyas fallas ocasionará afectaciones a su continuidad, provocando daños temporales o permanentes de diferentes tipos. Las pruebas de penetración deben planificarse cuidadosamente en todos los niveles y áreas involucradas, manteniéndose dentro de los límites de las regulaciones vigentes. Estas pruebas de penetración deben ser ejecutadas por personal experimentado y por lo general, el nivel de las afectaciones planificadas es monitoreado continuamente, priorizándose la protección de la información y vitalidad del servicio más que la explotación de vulnerabilidades activas.



En los entornos de desarrollo, prueba y producción pueden encontrarse diferentes grupos de riesgos de seguridad y de procesos asociados a las pruebas de penetración los cuales son abordados en las siguientes secciones.

Riesgos en Pruebas de Penetración

Diversos autores han planteado los retos y deficiencias que se encuentran durante la utilización de las metodologías de pruebas de penetración en entornos reales y ante determinados escenarios y productos de software dentro de los que se encuentran las aplicaciones web (Antunes & Vieira, 2017; Dalalana Bertoglio & Zorzo, 2017; DeMarco, 2018; González Brito & Montesino Perurena, 2018; Knowles, Baron, & McGarr, 2016; Mansfield-Devine, 2017; Rahalkar, 2016). Las pruebas de penetración engloban un conjunto de acciones sobre los sistemas y redes de datos, muchas de las cuales pueden generar efectos colaterales no deseados (Blackwell, 2014). El empleo de herramientas automatizadas también representa una sobrecarga anómala de tramas de red (Kao, Wang, Tsai, & Chen, 2018; Shah et al., 2019). Los contenidos de las peticiones por lo general, causan efecto en las aplicaciones web, en forma de adición, modificación o eliminación de los datos contenidos (Negi, Kumar, Ghosh, Shukla, & Gahlot, 2019; Thakre & Bojewar, 2018).

Las peticiones masivas sobre los sistemas sobrecargan los servidores de aplicaciones y sistemas de gestión de bases de datos, afectando tanto la continuidad operacional de los procesos que soporta la aplicación web sujeta a evaluación (Laidlaw & Shoemaker, 2020), como otras con las cuales pueda estar integrada en el ecosistema de software de la organización (Jansen et al., 2019) o compartir recursos comunes en la infraestructura tecnológica (Manaseer, K.Al Hwaitat, & Jabri, 2018; Mansfield-Devine, 2018; Türpe & Eichler, 2009).

Resulta evidente, por tanto, que los especialistas de seguridad, en muchos casos deben evitar realizar acciones que un adversario no tendría reparos en ejecutar, lo que puede provocar la distorsión del nivel de semejanza con la realidad que puede alcanzarse y, por ende, afectar la calidad de los resultados esperados para fortalecer eficazmente la seguridad de las aplicaciones web (Hasan & Meva, 2018). Los adversarios además no disponen de límites de tiempo para estudiar la aplicación web hasta que puedan encontrar una vulnerabilidad (Miaoui & Boudriga, 2019), sin embargo, las pruebas de penetración tienen ventanas de tiempo reducidas y muchas veces insuficientes para analizar con suficiente nivel de detalles todos los problemas que pueden estar presente, por tanto, los reportes finales se hacen fundamentalmente con los datos obtenidos en los escaneos automatizados de vulnerabilidades (Knowles et al., 2016).



Otro aspecto importante surge de la posibilidad de que la organización, debido a su objeto social, no cuente con personal capacitado para comprender el contexto en que se desarrolló la prueba de penetración y por tanto, pueden encontrarse deficiencias en la aplicación de las recomendaciones de los reportes emitidos (Sina, 2019). Por este motivo existen otros riesgos asociados al propio proceso de pruebas de penetración (Such, Gouglidis, Knowles, Misra, & Rashid, 2016; Work, 2019). A continuación, se profundiza en ambos grupos de riesgos.

Riesgos de seguridad en las aplicaciones web

Desde la dimensión de la aplicación web como producto de software, los riesgos de seguridad pueden agruparse según el nivel de afectación que pueden causar a la confidencialidad, integridad y disponibilidad en los datos y procesos relacionados con la aplicación web.

Riesgos relacionados con la Confidencialidad

La confidencialidad es la preservación de las restricciones autorizadas para el acceso a la información y su divulgación, incluidos los medios para proteger la privacidad personal y la información de propiedad (Niele, Dempsey, & Pillitteri, 2017; Stallings & Brown, 2018). Una pérdida de confidencialidad es la divulgación o exposición no autorizada de información, ya sea por parte de personas u otros sistemas informáticos. En la tabla 1 se enuncian los principales riesgos de seguridad a la confidencialidad que pueden afectar a la aplicación web durante una prueba de penetración.

Tabla 1. Riesgos de seguridad a la confidencialidad que pueden afectar a la aplicación web durante una prueba de penetración.

ID	Riesgo	Consecuencias
RC1	Captura de credenciales de usuarios y acceso a contraseñas y tokens de sesión de otros servicios asociados.	La explotación de vulnerabilidades podría facilitar el acceso a los archivos de configuración de la aplicación web que contienen las credenciales de autenticación de bases de datos, APIs de tercero y otras aplicaciones web. Además, puede producirse la interceptación de paquetes de red conteniendo credenciales de autenticación o tokens de sesión de usuarios reales, lo que permitirá suplantar sus operaciones en el sistema.
RC2	Exposición de información sensible debido a la afectación en archivos de configuración y permisos del sistema de archivos.	La explotación de vulnerabilidades podría desconfigurar los mecanismos de autorización, permitiendo el acceso al contenido a terceros, incluyendo motores de indexación de Internet y botnets (Yin, Lv, Zhang, Tian, & Cui, 2018).
RC3	Inyección de puertas traseras o webshells que facilitan el acceso a	La explotación de las vulnerabilidades podría permitir la inyección de webshells (Wu, Sun, Huang, Jia, & Liu, 2019) genéricas y sin protección que podrían



	información sensible a terceros.	exponer contenido sensible desde otras subredes o Internet.
RC4	Acceso de los especialistas de seguridad a información sensible de los procesos de negocio que gestiona la aplicación web.	La explotación de vulnerabilidades de tipo Inyección SQL y otros podría facilitar el acceso de los especialistas de seguridad a información a los procesos gestionados por la aplicación web.
RC5	Acceso de los usuarios a funciones y datos no autorizados.	La explotación de vulnerabilidad podría afectar la capa de autorización, provocando la modificación de los permisos de acceso de los usuarios, lo que les permitiría ejecutar funcionalidades no relacionada con su cargo en la organización.

Riesgos relacionados con la Integridad

La integridad es la protección contra la modificación o destrucción incorrecta de la información, incluida la garantía de no repudio de la información y su autenticidad (Nieles et al., 2017; Stallings & Brown, 2018). Una pérdida de integridad, por tanto, es la modificación o destrucción no autorizada de información, cuestión que puede ocurrir con frecuencia durante la evaluación de las vulnerabilidades presentes mediante las herramientas de pruebas de penetración. En la tabla 2 se enuncian los principales riesgos de seguridad a la integridad que pueden afectar a la aplicación web durante una prueba de penetración.

Tabla 2. Riesgos de seguridad a la integridad que pueden afectar a la aplicación web durante una prueba de penetración.

ID	Riesgo	Consecuencias
RI1	Manipulación de entidades en el sistema, siguiendo el patrón CRUD (Shon, 2019) a partir de los niveles de acceso de los usuarios de prueba que hayan sido suministrados a las herramientas automáticas de evaluación.	Para comprobar la presencia de vulnerabilidad, las herramientas automáticas pueden activar las funcionalidades de creación, lectura, modificación y eliminación de entidades en la aplicación web o durante la explotación de vulnerabilidades.
RI2	Desconfiguración de plantillas, temas y otros elementos de la capa de presentación.	Para comprobar la presencia de vulnerabilidad, las herramientas automatizadas pueden inyectar códigos que afecten la capa de presentación lo que afectaría la presentación visual de los contenidos de la aplicación web, también puede ocurrir durante la explotación de vulnerabilidades.
RI3	Perdida de la integridad en aplicaciones de terceros que consuman datos y recursos comunes gestionados por la aplicación web.	Las afectaciones a la integridad de los datos de la aplicación web puede afectar el funcionamiento de terceras con las que esté integrada en el ecosistema de aplicaciones de la organización.

RI4	Afectación en la generación de trazas de operaciones de la aplicación web.	La ejecución en lote de pruebas de seguridad realizadas por las herramientas automatizadas genera un número inusual de operaciones, lo que puede afectar los mecanismos establecidos de gestión de trazas, produciendo registros incompletos y afectación de formato.
RI5	Afectación en el funcionamiento de herramientas de monitoreo y Sistema de Gestión de Información y Eventos de Seguridad (SIEM)	El tratamiento de la información realizadas por las herramientas de monitoreo y sistemas SIEM puede verse afectada por la avalancha de falsos positivos generados por las pruebas de penetración, afectando su análisis para la toma de decisiones.

Riesgos relacionados con la Disponibilidad

La disponibilidad consiste en garantizar el acceso oportuno y confiable y el uso de la información (Nieles et al., 2017; Stallings & Brown, 2018). Una pérdida de disponibilidad es la interrupción del acceso o uso de la información o un sistema de información, cuestión que desafortunadamente puede ocurrir con frecuencia, entre otras cosas por el alto número de peticiones HTTP que generan las herramientas utilizadas durante una prueba de penetración. En la tabla 3 se enuncian los principales riesgos de seguridad a la disponibilidad que pueden afectar a la aplicación web durante una prueba de penetración.

Tabla 3. Riesgos de seguridad a la disponibilidad que pueden afectar a la aplicación web durante una prueba de penetración.

ID	Riesgo	Consecuencias
RD1	Afectación de operaciones del sistema debido a sobrecargas de buffers a distintos niveles.	Las pruebas de seguridad pueden generar sobrecargas de buffer a diferentes niveles de la aplicación web en dependencia de la tecnología en que haya sido desarrollada y las políticas de despliegue establecidas, lo que puede afectar tanto la disponibilidad como el estado e integridad de los datos. Está asociados en muchos casos con las pruebas de tipo fuzzy, las cuales son comunes en este tipo de evaluación de seguridad, así como pruebas de tipo rendimiento para comprobar la capacidad de defensa ante ataques de denegación de servicios de diferentes tipos.
RD2	Sobrecarga de servidores de bases de datos y de contenido debido al alto volumen de peticiones.	Las pruebas de seguridad sobrecargan las peticiones a los servidores de bases de datos y otros servicios, lo que puede hacerlos colapsar y afectar el funcionamiento de la aplicación web.
RD3	Saturación de los medios de almacenamiento por la generación de trazas de operaciones de la aplicación web.	La generación de trazas de la aplicación web está directamente relacionada con la cantidad de operaciones ejecutadas, por tanto, el elevado número de peticiones que se realizan durante una prueba de penetración puede provocar la ocupación de toda la capacidad disponible de los medios de almacenamiento debido a la persistencia de un alto número de trazas por lo que puede colapsar el funcionamiento de los servidores.

RD4	Saturación de la capacidad de almacenamiento de los servidores de bases de datos y de contenido.	La inyección de un elevado número de datos puede saturar las capacidades de almacenamiento de los servidores de bases de datos y contenidos, sobre todo en las infraestructuras más débiles.
RD5	Perdida de la disponibilidad en aplicaciones de terceros que utilicen componentes comunes de infraestructura.	Las pruebas de seguridad pueden sobrecargar las peticiones a los servidores de bases de datos y otros servicios compartidos, lo que puede hacerlos colapsar y dejar de prestar servicios a otras aplicaciones que dependan de ellos.

Riesgos que Afectan el Proceso de Pruebas de Penetración en las Aplicaciones Web

Además de los riesgos de seguridad que pueden afectar a las aplicaciones web, la gestión del proceso de pruebas de penetración también puede verse afectada por riesgos que pueden retrasar o impedir la ejecución normal de las actividades a realizar en cada fase, limitar el alcance y los resultados esperados o el tratamiento posterior de las vulnerabilidades detectadas.

Riesgos de Alcance y Tiempo

El alcance, desde una visión de proyecto, consiste en el trabajo que debe realizarse para crear un producto o servicio con las prestaciones solicitadas (PMI, 2017; PTES, 2017). Su contextualización en la prueba de penetración estará definida tanto por el conjunto de pruebas de seguridad y tareas de soporte a las mismas que deben realizarse, como por el número de aspectos de la aplicación web que se pretenden evaluar. El tiempo, por consiguiente, abarca el plazo para conseguir dicho propósito. Teniendo esto en cuenta, pueden identificarse varios riesgos que pueden afectar el alcance y tiempo planificado en una prueba de penetración, los cuales se enumeran en la tabla 4.

Tabla 4. Riesgos que pueden afectar el alcance y tiempo de una prueba de penetración.

ID	Riesgo	Consecuencias
RA1	Uso limitado de credenciales de acceso al sistema	La ausencia de disponibilidad de credenciales de usuarios puede retrasar o imposibilitar por completo el análisis de las funcionalidades según la configuración de la capa de autorización. Tener dos pares de credenciales por cada rol resulta esencial para comprobar determinadas vulnerabilidades como la referencia directa insegura a objetos, por ejemplo.
RA2	Insuficientes datos de prueba para analizar las funcionalidades del sistema.	La activación de diversas funcionalidades depende de la disponibilidad de datos en el sistema. Por ejemplo, resulta imposible analizar la seguridad del proceso de compras en una plataforma de comercio electrónico si no hay productos previamente introducidos en la base de datos que permitan instanciar y estudiar todas operaciones que se realizarán en un entorno real.

RA3	Limitaciones para ejecutar determinadas pruebas de seguridad	La Política de Seguridad de la organización, así como las regulaciones establecidas pueden limitar la ejecución de determinadas pruebas de seguridad, así como el uso de herramientas automatizadas de seguridad. Un ejemplo de ello puede ser la prohibición de ejecución de pruebas de seguridad desde una red externa cuando se traten de aplicaciones web de cara a Internet o el escaneo de vulnerabilidades de los servidores web desde la propia intranet.
RA4	Ejecución parcial de las pruebas de seguridad requeridas	Las características del entorno de prueba pueden imposibilitar la ejecución de determinadas pruebas de seguridad como por ejemplo la ejecución de actividades de OSINT(Open Source Intelligence) (Kothia, Swar, & Jaafar, 2019) desde Internet si se trata de una aplicación web que no se ha implantado en la práctica.
RA5	Análisis aislado de componentes del sistema	La evaluación aislada de la seguridad de los componentes del sistema no garantiza que durante su interconexión puedan surgir vulnerabilidades que pongan en riesgo a aquellos considerados seguros. Esto ocurre generalmente durante el proceso de desarrollo de software.
RA6	Selección y configuración deficiente de las herramientas automatizadas de seguridad	Si no se realiza una selección adecuada de herramientas de evaluación de seguridad y su posterior configuración, teniendo en cuenta el propósito de la prueba de penetración, su tecnología y plazos disponibles, puede verse afectado negativamente todo el proceso. Esto incluye sobrecargas y daños a la aplicación web, incumplimiento de los tiempos acordados y la detección parcial de las vulnerabilidades presentes.
RA7	Actividades de mantenimiento en la aplicación web	Si los desarrolladores cambian el código de la aplicación web durante el proceso de evaluación, será necesario reiniciar la prueba de penetración.
RA8	Insuficiente tiempo para la ejecución de la prueba de penetración	Normalmente es muy limitado el tiempo disponible para la prueba de penetración si se compara con la diversidad de pruebas de seguridad que deben ejecutarse y la integración y análisis posterior de los resultados. Esto se evidencia sobre todo en aplicaciones desarrolladas a la medida, donde se hace necesario probar funcionalidades de procesos de negocio que los especialistas de seguridad no dominan.
RA9	Ventana de tiempo limitada para el uso de herramientas automatizadas	Para limitar los riesgos de seguridad y garantizar la continuidad de los procesos que soporte la aplicación web, la organización puede establecer una ventana de tiempo para la realización de las pruebas de seguridad, pero esta puede ser insuficiente para el uso de las herramientas automatizadas de seguridad. Por ejemplo, un escaneo de seguridad con OWASP ZAP o ACUNETIX puede llevar más de un día.

Riesgos de Infraestructura Tecnológica

Las características de la infraestructura tecnológica pueden generar riesgos que dificulten la ejecución de la prueba de penetración en aplicaciones web. Estos riesgos se enumeran en la tabla 5:

Tabla 5. Riesgos de la infraestructura tecnológica que pueden afectar la ejecución de las pruebas de penetración.

ID	Riesgo	Consecuencias
----	--------	---------------

RT1	Limitaciones de las capacidades de la infraestructura de despliegue	La infraestructura tecnológica puede ser insuficiente para soportar las diferentes pruebas de seguridad que son necesarias desarrollar y por tanto puede sobrecargarse fácilmente, interrumpiendo el funcionamiento de la aplicación web. Esto es común en entornos de desarrollo y en ocasiones de pruebas donde se usan estaciones de trabajo normales para desplegar el producto de software.
RT2	Presencia de mecanismos de seguridad	Debido a las características de las pruebas de penetración, estas deben disparar necesariamente las alertas y acciones de los mecanismos de seguridad como IDS/IPS y cortafuegos, los cuales pueden limitar o impedir la interacción con la aplicación web. Esto ocurre también cuando hay presencia de mecanismos internos de seguridad de la propia aplicación web.
RT3	Interrupción de servicios	La interrupción de los servicios no solo puede retrasar las pruebas de seguridad sino también algunas pueden requerir que sean iniciadas nuevamente. Las interrupciones pueden ser debido a múltiples causas que pueden abarcar desde cortes de energía hasta la aplicación de políticas específicas de la entidad.

Riesgos asociados al personal

Tener en cuenta el factor humano es esencial en cualquier proceso y por tanto en una prueba de penetración en aplicaciones web. Es por ello que se enuncian en la tabla 6 los principales riesgos que puede surgir debido a ello:

Tabla 6. Riesgos asociados al personal que pueden afectar la ejecución de las pruebas de penetración.

ID	Riesgo	Consecuencias
RH1	Insuficiente uso de una metodología de pruebas de penetración.	La no adherencia a una metodología formalizada, ya sea de las comúnmente reconocidas a nivel internacional o propia de la entidad, puede ocasionar improvisaciones que afecten el proceso y dejar de probar determinadas cuestiones de la aplicación web.
RH2	Deficiencias en la planificación de la prueba de penetración.	Las deficiencias en la planificación de las pruebas de penetración pueden crear el escenario para la materialización de los diferentes riesgos presentados en el modelo. Sobre todo, se manifiesta en las organizaciones que no cuentan con personal capacitado para comprender el alcance de la prueba de penetración y las condiciones de aseguramiento necesarias para llevarla a cabo de manera efectiva.
RH3	Deficiencias en la interpretación de los reportes de las pruebas de penetración.	Las organizaciones, por la característica de su objetivo social, pueden carecer de especialistas que sean capaces de interpretar correctamente los resultados de las pruebas de penetración para poder tomar decisiones correctas sobre el método de mitigación de las vulnerabilidades reportadas.
RH4	Fallas en la comunicación de los involucrados durante la ejecución de las pruebas de seguridad.	A pesar de la planificación realizada, pueden darse diferentes condiciones que obliguen al personal a mitigar algún problema presentado durante la prueba de penetración, cuestión que puede tomar tiempo si no se establece un mecanismo de comunicación eficaz entre todas las partes involucradas.



Estrategia de Mitigación de Riesgos

Para el tratamiento de los riesgos mencionados, la organización puede evitar o eliminar estos riesgos, reducirlos o mitigarlos disminuyendo la probabilidad de ocurrencia o mediante la reducción de su impacto. También puede transferirlo o asignarlo a terceros o aceptarlos. En función de ello, pueden establecerse diferentes estrategias que pueden contribuir a minimizar los riesgos planteados. Estas estrategias están enfocadas precisamente en las aplicaciones web, por ello, según el alcance de la prueba de penetración, algunas pueden no ser válidas si se analizan otros tipos de sistemas y servicios o se enfocan en comprobar la resistencia de los mecanismos de seguridad de la organización.

Mitigación de Riesgos de Seguridad en Pruebas de Penetración en Aplicaciones Web

1. **Establecimiento de acuerdos de confidencialidad:** Como parte de los preparativos de la prueba de penetración los especialistas de seguridad firman acuerdos de confidencialidad alineados con las regulaciones y leyes vigentes, comprometiéndose a no divulgar ni conservar cualquier tipo de información obtenida. En determinadas situaciones esto puede significar la ejecución de las pruebas de penetración exclusivamente desde medios de cómputo suministrados por la organización para reforzar la confidencialidad, evitar filtraciones indeseadas y controlar la destrucción de la información generada posteriormente.
2. **Renovación de las credenciales de usuarios y servicios:** La organización sustituye las credenciales de acceso y de conexión con otros servidores (ej. sistemas de gestión de bases de datos) en la aplicación web, para garantizar que la información que pudo haberse obtenido en la prueba de penetración pierda validez.
3. **Restricción de acceso desde Internet:** Si es imprescindible comprobar la explotación de vulnerabilidades que puedan exponer información sensible de cara a Internet de una aplicación web desplegada en una infraestructura de producción, deben definirse mecanismos de restricción para evitar el acceso desde Internet de los motores de búsqueda e indexación y de usuarios en los tiempos que dure las pruebas de seguridad y así impedir el acceso de terceros a esta información.
4. **Reforzamiento de la política de respaldo y recuperación de la información:** Se ejecutan procedimientos reforzados de respaldo de la información durante la ejecución de las pruebas de seguridad, de manera que,



ante la ocurrencia de algún daño, esta pueda ser recuperada rápidamente, minimizando las pérdidas de información para la organización.

5. **Ejecución de las pruebas de seguridad invasivas en periodos menos riesgosos para la organización:** Las pruebas de seguridad que tengan mayor probabilidad de ocasionar daños a la aplicación web se planifican para ser ejecutadas fuera del horario laboral, de este modo se evita afectar los procesos operativos de la entidad y se establece un margen de tiempo para la recuperación a partir de los respaldos realizados si se produjera algún daño.
6. **Mapeo de sistemas relacionados para minimizar daños colaterales:** La organización realiza un análisis de las dependencias que tiene la aplicación web objeto de la prueba de penetración con otros sistemas y servicios que puedan sufrir efectos colaterales con las pruebas de seguridad, de forma tal que puedan diseñarse acciones para garantizar la continuidad de los procesos si esto ocurre. Se incluirán aquellos sistemas más interconectados como los servidores de sistemas de gestión de bases de datos, servidores de correo electrónico que gestionan los mensajes de alerta ante acciones de bloqueo de credenciales o cambios de contraseña y otros.
7. **Filtrado de trazas de seguridad:** Se aplicarán configuraciones específicas durante el tiempo que dure la prueba de penetración para evitar contaminar el análisis continuo de las trazas de seguridad y los comportamientos anómalos con aquellas acciones propias de la prueba de penetración. Para ello pueden utilizarse diversas estrategias como el filtrado de las direcciones IP o subredes utilizadas por los especialistas de seguridad.
8. **Establecimiento de condiciones tecnológicas en el entorno de pruebas:** Debe asegurarse que el entorno donde está desplegada la aplicación web cumpla con los requerimientos necesarios para soportar de forma razonable las sobrecargas de peticiones que deben generarse y que no dificulten la ejecución de las pruebas de seguridad. Es necesario prestar atención especialmente a esta cuestión en los entornos de desarrollo y de prueba.
9. **Completamiento de bases de datos de prueba y credenciales de usuarios:** En los entornos de desarrollo y de prueba deben garantizarse juegos de datos que permitan comprobar todas las funcionalidades de la



aplicación web y asociados a las credenciales de usuarios que puedan facilitarse y de este modo abarcar en un menor tiempo las pruebas de seguridad a la totalidad de las funcionalidades existentes.

10. **Establecimiento del contexto de utilización de los resultados:** En la fase de preparación debe definirse el alcance de los resultados que se pueden obtener según el entorno de despliegue facilitado por la organización y de este modo contextualizar como puede utilizarse los resultados obtenidos. Por ejemplo, si las pruebas se realizan en un entorno de desarrollo o de prueba, debe quedar registrado que aspectos no se tuvieron en cuenta y que son importantes evaluar posteriormente en el entorno de producción o, por ejemplo, la necesidad de repetir la prueba de penetración al final del proceso de desarrollo si se ejecuta a nivel de módulos que no están integrados.
11. **Estabilización del entorno de despliegue:** La prueba de penetración no debe realizarse hasta tanto no se hayan estabilizado todos los componentes en el entorno de despliegue y esto no debe variar hasta el fin del proceso de evaluación. Debe prohibirse la adición de nuevos códigos, actualizaciones y otras actividades de mantenimiento, estableciendo la necesidad de repetir las pruebas de seguridad si esto ocurre. Debe definirse un personal encargado de velar por el cumplimiento de esta cuestión.
12. **Configuración de los mecanismos de seguridad externos:** Los administradores de la infraestructura tecnológica establecen configuraciones permisivas que evitan ralentizar o impedir las pruebas de seguridad desde direcciones IP previamente convenidas. A su vez se evita la gestión de un número excesivamente inusual de falsos positivos que entorpezcan la toma de decisiones durante el tiempo de ejecución de la prueba de penetración. Una vez realizadas las suficientes pruebas de seguridad, se vuelven a habilitar las reglas normales para comprobar su eficacia.
13. **Adhesión a una metodología de prueba de penetración:** La prueba de penetración debe alinearse a una metodología descrita y que pueda ser de conocimiento de todos los implicados para lograr un entendimiento común de las acciones que serán llevadas a cabo, planificar mejor todas las condiciones organizativas y técnicas necesarias para ello, establecer un mecanismo de comunicación continuo, así como garantizar la auditabilidad de la prueba de penetración, aumentando la transparencia sobre las conclusiones reflejadas en el reporte final.



14. **Utilización de consultores externos:** La organización puede apoyarse en consultores externos para que le ayuden a evaluar el resultado de la prueba de penetración y puedan asesorar su traducción en políticas y controles técnicos efectivos para erradicar las vulnerabilidades encontradas.

Como fue descrito anteriormente, las herramientas automatizadas de seguridad desempeñan un papel importante durante una prueba de penetración, por este motivo, es necesario complementar estas medidas con el establecimiento de estrategias para focalizar su uso y ejecución (Bari & Ahamad; Hasan & Meva, 2018; Kothia et al., 2019; Manaseer et al., 2018; Mansfield-Devine, 2018; Miaoui & Boudriga, 2019; Nieves et al., 2017; PMI, 2017; Shon, 2019; Sina, 2019; Stallings & Brown, 2018; Such et al., 2016; Türpe & Eichler, 2009; Work, 2019; Wu et al., 2019; Yin et al., 2018), minimizando de este modo la afectación sobre el rendimiento y otros posibles impactos negativos que puedan tener sobre la aplicación web objetivo:

1. **Inferencia de vulnerabilidades mediante interacciones de bajo impacto:** Se extrae información de las peticiones HTTP resultantes de aplicar dinámicas similares a las realizadas por un usuario común. En caso de aplicaciones web de cara a Internet se recolecta también datos presentes en motores de búsqueda e indexadores especializados mediante técnicas OSINT. A partir del conocimiento adquirido se infieren las vulnerabilidades que pueden estar presente. Esta técnica es efectiva si la aplicación web contiene tecnologías comunes como sistemas de gestión de contenidos (CMS), componentes y API, entre otros elementos cuyas vulnerabilidades son publicadas periódicamente.
2. **Secuenciación de las pruebas de seguridad:** Las pruebas de seguridad son segmentadas para impedir que la ejecución paralela de las pruebas, ya sea realizadas por una herramienta o por varias al mismo tiempo, generen efectos no deseados en la aplicación web.
3. **Disminución de la frecuencia de pruebas de seguridad por unidad de tiempo:** Las herramientas de pruebas de seguridad son configuradas para emitir un menor número de peticiones que las habituales, de manera que no interfieran significativamente en el rendimiento de la aplicación web.
4. **Explotación selectiva de vulnerabilidades:** Los especialistas de seguridad se concentran en explotar vulnerabilidades específicas que constituyan una mayor importancia para el resultado final de la prueba de penetración. Los motivos pueden incluir a las posibles vulnerabilidades no documentadas en el proceso de análisis, las que están asociadas a los procesos de negocio soportados o aquellas que ya fueron explotadas con



anterioridad por los adversarios, dieron origen a un incidente que seguridad y se pretende comprobar si la solución dada fue efectiva.

5. **Establecimiento de mecanismos de interrupción y reanudación de las pruebas de seguridad:** Las ejecuciones de las pruebas de seguridad se diseñan para entrar en un estado de inactividad si es detectado algún comportamiento anómalo que pueda indicar afectaciones que se encuentran fuera del alcance de la prueba de penetración para luego reanudarse una vez que la situación haya sido resuelta.
6. **Automatización de casos de pruebas de seguridad:** Se diseñan pruebas de seguridad tomando en cuenta las funcionalidades de la aplicación web y se programan mediante los marcos de trabajo asociados a la tecnología utilizada. Estas acciones pueden resultar muy efectiva en el entorno de desarrollo y pueden extenderse a los entornos de prueba y producción como un artefacto resultante del proceso de desarrollo. Esto no evita el uso de herramientas estándares de seguridad durante una prueba de penetración, pero optimiza su utilización en aspectos más concretos.
7. **Evasión primaria de los mecanismos de bloqueo:** La secuencia de pruebas de seguridad se adapta para dejar la evaluación de los mecanismos de bloqueos en un plano posterior y así impedir la inhabilitación de las credenciales de usuarios en las pruebas de seguridad que requieran autenticación o la inclusión de las direcciones IP de los especialistas en las listas automáticas de restricciones de la aplicación web.

Conclusiones

Las pruebas de penetración constituyen un proceso importante para evaluar la seguridad de las aplicaciones web, pero existen diferentes riesgos que se deben tener en cuenta. En función de ello, en el presente trabajo se definieron 15 riesgos que pueden afectar la seguridad de las aplicaciones web y 16 riesgos que pueden retrasar o impedir la ejecución normal de las actividades a realizar en cada fase, limitar el alcance y los resultados esperados o el tratamiento posterior de las vulnerabilidades detectadas. Posteriormente se presentaron diferentes acciones y estrategias que pueden ser llevadas a cabo para mitigar estos riesgos.

A partir de los resultados aquí obtenidos, los implicados en procesos de pruebas de penetración podrán disponer de una base de partida que favorezca el tratamiento de riesgos y contextualizar mejor la toma de

decisiones en función de solucionar las vulnerabilidades de seguridad halladas a través de este tipo de evaluación de seguridad.

Referencias

- Acunetix. (2019). Acunetix. Web Application Vulnerability Report 2019. In.
- Al-Matari, O. M., Helal, I. M., Mazen, S. A., & Elhennawy, S. (2018). *Cybersecurity Tools for IS Auditing*. Paper presented at the 2018 Sixth International Conference on Enterprise Systems (ES).
- Alghofaili, R. (2018). *Security Analysis of Open Source Content Management Systems Wordpress, Joomla, and Drupal*. (Tesis de Maestría), California State Polytechnic University, EE.UU.
- Alsmadi, I. (2019). *The NICE Cyber Security Framework: Cyber Security Intelligence and Analytics*. Gewerbestrasse (Suiza): Springer.
- Anisetti, M., Asal, R., Ardagna, C. A., Comi, L., Damiani, E., & Gaudenzi, F. (2019). *A Knowledge-Based IoT Security Checker*, Cham.
- Antunes, N., & Vieira, M. (2017). Designing vulnerability testing tools for web services: approach, components, and tools. *International Journal of Information Security*, 16(4), 435-457. doi:10.1007/s10207-016-0334-0
- Barceló, M., & Herzog, P. (2010). *OSSTMM: Open Source Security Testing Methodology Manual*. Barcelona, España: Institute for Security and Open Methodologies (ISECOM).
- Bari, M. A., & Ahamad, S. Study of Ethical Hacking and Management of Associated Risks.
- Bartoli, A., De Lorenzo, A., Medvet, E., Faraguna, M., Tarlao, F. J. C., & Technologies, I. (2018). A Security-Oriented Analysis of Web Inclusions in the Italian Public Administration. 18(4), 94-110.
- Bishop, D., & Rowland, P. (2019). Agile and Secure Software Development: An Unfinished Story. *Issues in Information Systems*, 20(1).
- Blackwell, C. (2014). Towards a Penetration Testing Framework Using Attack Patterns. In *Cyberpatterns* (pp. 135-148). Switzerland: Springer.
- Brohi, A. B., Butt, P. K., & Zhang, S. (2019). *Software Quality Assurance: Tools and Techniques*, Cham.
- Casola, V., De Benedictis, A., Rak, M., & Villano, U. (2018). *Towards automated penetration testing for cloud applications*. Paper presented at the 2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE).
- Cuzme-Rodríguez, F., León-Gudiño, M., Suárez-Zambrano, L., & Domínguez-Limaico, M. (2019). *Offensive Security: Ethical Hacking Methodology on the Web*, Cham.
- Dalalana Bertoglio, D., & Zorzo, A. F. (2017). Overview and open issues on penetration test. *Journal of the Brazilian Computer Society*, 23(1), 1-16. doi:10.1186/s13173-017-0051-1
- DeMarco, J. V. (2018). An approach to minimizing legal and reputational risk in Red Team hacking exercises. *Computer Law and Security Review*, 34(4), 908-911. doi:10.1016/j.clsr.2018.05.033
- Felderer, M., Büchler, M., Johns, M., Brucker, A. D., Breu, R., & Pretschner, A. (2016). Security Testing: A Survey. In *Advances in Computers* (Vol. 101, pp. 1-51).
- Flaus, J.-M. (2019). *Cybersecurity of industrial systems*. EE.UU: John Wiley & Sons.
- 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.

- Haber, M. J., & Hibbert, B. (2018). Asset Attack Vectors: Building Effective Vulnerability Management Strategies to Protect Organizations.
- Hasan, A., & Meva, D. J. I. J. o. A. S. o. S. R. (2018). Web Application Safety by Penetration Testing. 3(9).
- Horton, S. (2020). *Are Software Security Issues a Result of Flaws in Software Development Methodologies?*, Utica College,
- Jamil, A., Asif, K., Ashraf, R., Mehmood, S., & Mustafa, G. (2018). *A Comprehensive study of Cyber Attacks & Counter Measures for web systems*. Paper presented at the Proceedings of the 2nd International Conference on Future Networks and Distributed Systems.
- Jansen, S., Cusumano, M., & Popp, K. M. (2019). Managing Software Platforms and Ecosystems. *IEEE Software*, 36(3), 17-21. doi:10.1109/MS.2019.2891795
- Kao, D., Wang, Y., Tsai, F., & Chen, C. (2018). Forensic analysis of network packets from penetration test toolkits. In *2018 20th International Conference on Advanced Communication Technology (ICACT)* (pp. 363-368). Nueva York, EE.UU: IEEE.
- Kaspersky. (2020). Kaspersky Security Bulletin 2020. Statistics.
- Kettani, H., & Wainwright, P. (2019). *On the top threats to cyber systems*. Paper presented at the 2019 IEEE 2nd International Conference on Information and Computer Technologies, ICICT 2019.
- Knowles, W., Baron, A., & McGarr, T. (2016). The simulated security assessment ecosystem: Does penetration testing need standardisation? *Computers & Security*, 62, 296-316. doi:10.1016/j.cose.2016.08.002
- Kothia, A., Swar, B., & Jaafar, F. (2019). *Knowledge Extraction and Integration for Information Gathering in Penetration Testing*. Paper presented at the 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C).
- Kumar, R., & Tlhagadikgora, K. (2019). *Internal Network Penetration Testing Using Free/Open Source Tools: Network and System Administration Approach*, Singapore.
- Laidlaw, G., & Shoemaker, D. (2020). Software assurance: the things a manager needs to know. *EDPACS*, 61(4), 1-8. doi:10.1080/07366981.2020.1753283
- Manaseer, S., K.Al Hwaitat, A., & Jabri, R. (2018). Distributed Detection and prevention of Web Threats in Heterogeneous Environment. 12, 13. doi:10.5539/mas.v12n10p13
- Mansfield-Devine, S. (2017). Open source software: determining the real risk posed by vulnerabilities. *Network Security*, 2017(1), 7-12. doi:10.1016/S1353-4858(17)30005-3
- Mansfield-Devine, S. (2018). Friendly fire: how penetration testing can reduce your risk. *Network Security*, 2018(6), 16-19. doi:10.1016/S1353-4858(18)30058-8
- Mehta, S., Raj, G., & Singh, D. (2018). *Penetration Testing as a Test Phase in Web Service Testing a Black Box Pen Testing Approach*, Singapore.
- Meucci, M., & Muller, A. (2014). *OWASP Testing Guide 4.0* (4 ed.). EE.UU: OWASP Foundation.
- Miaoui, Y., & Boudriga, N. (2019). Enterprise security investment through time when facing different types of vulnerabilities. *Information Systems Frontiers*, 21(2), 261-300. doi:10.1007/s10796-017-9745-3
- Mohammed, N. M., Niazi, M., Alshayeb, M., & Mahmood, S. (2017). Exploring software security approaches in software development lifecycle: A systematic mapping study. *Computer Standards & Interfaces*, 50, 107-115. doi:10.1016/j.csi.2016.10.001
- Montesino Perurena, R., Baluja García, W., & Porvén Rubier, J. (2013). Gestión automatizada e integrada de controles de seguridad informática. *Ingeniería Electrónica, Automática y Comunicaciones*, 34(1), 40-58.

- Muniz, R., Braz, L., Gheyi, R., Andrade, W., Fonseca, B., & Ribeiro, M. (2018). *A Qualitative Analysis of Variability Weaknesses in Configurable Systems with# ifdefs*. Paper presented at the Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems.
- Murthy, P., & Shilpa, R. (2018). *Vulnerability Coverage Criteria for Security Testing of Web Applications*. Paper presented at the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI).
- Negi, R., Kumar, P., Ghosh, S., Shukla, S. K., & Gahlot, A. (2019). Vulnerability Assessment and Mitigation for Industrial Critical Infrastructures with Cyber Physical Test Bed. In *2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS)* (pp. 145-152). Nueva York, EE.UU: IEEE.
- Nguyen, V. L., Lin, P. C., & Hwang, R. H. (2019). Web attacks: defeating monetisation attempts. *Network Security*, 2019(5), 11-19. doi:10.1016/S1353-4858(19)30061-3
- Nieves, M., Dempsey, K., & Pillitteri, V. (2017). *An introduction to information security*. Retrieved from Ojagbule, O., Wimmer, H., & Haddad, R. J. (2018). Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP. In *SoutheastCon 2018* (pp. 1-7). Nueva York, EE.UU: IEEE.
- Papadopoulos, E. P., Diamantaris, M., Papadopoulos, P., Petsas, T., Ioannidis, S., & Markatos, E. P. (2017). *The long-standing privacy debate: Mobile websites vs mobile apps*. Paper presented at the Proceedings of the 26th International Conference on World Wide Web.
- Patel, K. (2019, 23-25 April 2019). *A Survey on Vulnerability Assessment & Penetration Testing for Secure Communication*. Paper presented at the 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI).
- PMI. (2017). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* (6 ed.). Pensilvania (EE.UU): Project Management Institute.
- Positive_Technologies. (2020). PositiveTechnologies. Web Application Vulnerabilities Statistics for 2019. In PTES. (2017). The Penetration Testing Execution Standard Documentation. Retrieved from <http://bit.ly/3qmRjXy>
- Rahalkar, S. A. (2016). *Certified Ethical Hacker (CEH) Foundation Guide*. Pune (India): Springer.
- Rathore, B., Brunner, M., Dilaj, M., Herrera, O., Brunati, P., Subramaniam, R. K., . . . Chavan, U. (2006). *Information Systems Security Assessment Framework (ISSAF)*. Colorado Springs, EE.UU: Open Information Systems Security Group.
- Rodríguez, G. E., Torres, J. G., Flores, P., & Benavides, D. E. (2020). Cross-site scripting (XSS) attacks and mitigation: A survey. *Computer Networks*, 166, 1-43. doi:10.1016/j.comnet.2019.106960
- Saha, S., Das, A., Kumar, A., Biswas, D., & Saha, S. (2020). *Ethical Hacking: Redefining Security in Information System*, Singapore.
- Schmittner, C., Griessnig, G., & Ma, Z. (2018). *Status of the Development of ISO/SAE 21434*, Cham.
- Shah, M., Ahmed, S., Saeed, K., Junaid, M., Khan, H., & Ata Ur, R. (2019). Penetration testing active reconnaissance phase - Optimized port scanning with nmap tool. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies, iCoMET 2019* (pp. 1-6). Nueva York, EE.UU: IEEE.
- Shon, M. D. (2019, 2-5 July 2019). *Information Security Analysis as Data Fusion*. Paper presented at the 2019 22th International Conference on Information Fusion (FUSION).
- Sina, B. J. (2019). *Identifying the Efficacy of Various Penetration Testing Practices*. Utica College,
- Stallings, W., & Brown, L. (2018). *Computer Security: Principles and Practice* (4 ed.). New York (EE.UU): Pearson.
- Stouffer, K., Falco, J., & Scarfone, K. (2008). *NIST SP 800-115: Technical Guide to Information Security Testing and Assessment*. Maryland, EE.UU: National Institute of Standards and Technology.

- Such, J. M., Gouglidis, A., Knowles, W., Misra, G., & Rashid, A. (2016). Information assurance techniques: Perceived cost effectiveness. *Computers & Security*, *60*, 117-133. doi:10.1016/j.cose.2016.03.009
- Sucuri.net. (2020). Web Professionals Security Survey. How agencies approach website security and protect their clients' websites.
- Swathy Akshaya, M., & Padmavathi, G. (2019). *Taxonomy of Security Attacks and Risk Assessment of Cloud Computing*, Singapore.
- Telefónica. (2020). Informe de Tendencias: Ciberamenazas Hacktivistas.
- Tetskyi, A., Kharchenko, V., & Uzun, D. (2018). Neural networks based choice of tools for penetration testing of web applications. In *2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT)* (pp. 402-405). Nueva York, EE.UU: IEEE.
- Thai, N. D., & Hieu, N. H. (2019). A framework for website security assessment. In *ACM International Conference Proceeding Series* (pp. 153-157). Nueva York, EE.UU: ACM.
- Thakre, S., & Bojewar, S. (2018). Studying the Effectiveness of Various Tools in Detecting the Protecting Mechanisms Implemented in Web-Applications. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 1316-1321). Nueva York, EE.UU: IEEE.
- Touseef, P., Alam, K. A., Jamil, A., Tauseef, H., Ajmal, S., Asif, R., . . . Mustafa, S. (2019). Analysis of Automated Web Application Security Vulnerabilities Testing. In *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems* (pp. 1-8). Nueva York, EE.UU: ACM.
- Türpe, S., & Eichler, J. (2009, 4-6 Sept. 2009). *Testing Production Systems Safely: Common Precautions in Penetration Testing*. Paper presented at the 2009 Testing: Academic and Industrial Conference - Practice and Research Techniques.
- Venson, E., Guo, X., Yan, Z., & Boehm, B. (2019). *Costing Secure Software Development: A Systematic Mapping Study*. Paper presented at the Proceedings of the 14th International Conference on Availability, Reliability and Security, Canterbury, CA, United Kingdom.
- Wang, Y., & Yang, J. (2017, 27-29 March 2017). *Ethical Hacking and Network Defense: Choose Your Best Network Vulnerability Scanning Tool*. Paper presented at the 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA).
- Work, J. (2019, 3-4 June 2019). *In wolf's clothing: Complications of threat emulation in contemporary cyber intelligence practice*. Paper presented at the 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security).
- Wu, Y., Sun, Y., Huang, C., Jia, P., & Liu, L. (2019). Session-Based Webshell Detection Using Machine Learning in Web Logs. *Security and Communication Networks*, *2019*, 11. doi:10.1155/2019/3093809
- Yin, J., Lv, H., Zhang, F., Tian, Z., & Cui, X. (2018). *Study on Advanced Botnet Based on Publicly Available Resources*, Cham.