Revista Cubana de Ciencias Informáticas
Vol. 12, No. 4, Octubre-Diciembre, 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 52-65
http://rcci.uci.cu

Tipo de artículo: Artículo original
Temática: Seguridad Informática
Recibido: 18/05/2018 | Aceptado: 08/10/2018

Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web

Capabilities of penetration test methodologies to detect frequent vulnerabilities of web applications

Henry Raúl González Brito[*[0000-0002-3226-9210]], **Raydel Montesino Perurena**

Universidad de las Ciencias Informáticas. Carretera San Antonio de los Baños Km 2 ½. La Lisa, La Habana, Cuba {henryraul, raydelmp}@uci.cu

***Autor para correspondencia**: henryraul@uci.cu

Resumen

En el estudio se analizan las capacidades para la detección de vulnerabilidades en aplicaciones web que proponen las principales metodologías de pruebas de penetración. El objetivo fue determinar hasta qué punto son válidos los procedimientos, herramientas y pruebas de seguridad propuestas en las metodologías ISSAF, OSSTMM, OWASP, PTES y NIST SP 800-115 para abordar los retos actuales de ciberseguridad en el campo del desarrollo y mantenimiento de las aplicaciones web. Se tomaron como base de comparación los informes de vulnerabilidades de OWASP, emitidos entre los años 2003 y 2017 y el análisis de la documentación de cada metodología de pruebas de penetración. Se elaboró una escala de evaluación cualitativa y su aplicación arrojó como resultado que la Guía de Pruebas de OWASP resultó la más completa, seguida de la metodología de ISSAF. No obstante, ninguna metodología demostró ser capaz de brindar métodos, herramientas o pruebas de seguridad para detectar todas las vulnerabilidades actuales. Los resultados alcanzados demuestran la necesidad de un proceso de adaptación y completamiento de las metodologías existentes.

Palabras clave: análisis de vulnerabilidades, aplicaciones web, OWASP, pruebas de penetración, seguridad informática.

Abstract

The study analyzes the capabilities for vulnerability detection in web applications that propose the main methodologies of intrusion tests. The objective was to determine the validity of the procedures, tools and tests proposed in the ISSAF,

Revista Cubana de Ciencias Informáticas
Vol. 12, No. 4, Octubre-Diciembre, 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 52-65
http://rcci.uci.cu

OSSTMM, OWASP, PTES and NIST SP 800-115 methodologies to address the current challenges of cybersecurity in the development and maintenance of Web applications. The OWASP vulnerability reports issued between 2003 and 2017 and the documentation of each intrusion methodology were taken as a base for comparison. A qualitative comparison scale was developed and its application showed that the most complete is OWASP Test Guide followed by the ISSAF methodology. However, no methodology proved to be able to provide security methods, tools or tests to detect all current vulnerabilities. The results show the need for a process of adaptation and complementation of existing methodologies.

Keywords: *computer security, OWASP, penetration test, vulnerability analysis, web application*

Introducción

En los últimos años, las continuas intrusiones en redes de datos y aplicaciones informáticas por parte de ciberdelincuentes a nivel internacional, han captado la atención del sector académico y empresarial para la búsqueda de soluciones que contribuyan a frenar o disminuir estos hechos (Dadkhah et al. 2018). Desafortunadamente, la presencia de vulnerabilidades en sistemas informáticos, aumenta continuamente, no solo en número sino también en el impacto de su explotación individual (Huang et al. 2017).

Uno de los principales blancos de ciberataques son las aplicaciones web (Agarwal and Hussain 2018; Bajovic 2017; Jhaveri et al. 2017). Estas son la base para la informatización de la sociedad moderna. Las organizaciones y personas interactúan fundamentalmente en el ciberespacio a través de aplicaciones web, mediante navegadores, aplicaciones nativas y dispositivos móviles (Bhandari et al. 2017; Wei and Wolf 2017).

Las aplicaciones web son una puerta de entrada a la infraestructura tecnológica de la organización. De los reportes emitidos en los últimos años puede inferirse que las soluciones tecnológicas basadas en antivirus, cortafuegos, sistemas de detección de intrusiones, han demostrado ser imprescindibles (Montesino Perurena et al. 2013; Topper 2018), pero su presencia no ha sido suficiente para disminuir o contener los ciberataques (Baş Seyyar et al. 2018; Nazir et al. 2017; Singh and Chatterjee 2017).

Un componente importante para mitigar estos problemas son las pruebas de penetración, conocidas también como *pentesting o hacking ético*. Estas son una práctica reconocida a nivel internacional que persigue la recreación de las posibles acciones de un ciberatacante 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 (Rahalkar 2016; Sandhya et al. 2017).

Las pruebas de penetración son parte de las evaluaciones de seguridad (Rahalkar 2016) y su empleo contribuye a garantizar que los sistemas informáticos y redes de datos cumplen con las normas y mecanismos de seguridad aplicados en las organizaciones y puedan ofrecer la mayor protección contra las amenazas comunes (Bajovic 2017; Jhaveri, Cetin, Ga, Moore and Eeten 2017).

En los últimos tiempos, diversos autores (Antunes and Vieira 2015; Dalalana Bertoglio and Zorzo 2017; Knowles et al. 2016; Rahalkar 2016) han publicado estudios que ponen de manifiesto las capacidades y limitantes de las metodologías de pruebas de penetración ante determinados escenarios y dominios tecnológicos. Uno de los principales escenarios en la actualidad son las aplicaciones web (Dalalana Bertoglio and Zorzo 2017), las cuales por su importancia revisten un papel fundamental para garantizar la seguridad en el ciberespacio.

En correspondencia con lo anterior, el presente artículo muestra los resultados de una investigación que tuvo como propósito indagar sobre las potencialidades de las metodologías de pruebas de penetración ante la detección de las principales vulnerabilidades de seguridad en las aplicaciones web. Caracterizar este aspecto es esencial para determinar hasta qué punto son válidos los procedimientos, herramientas y pruebas de seguridad propuestas en las metodologías para abordar los retos actuales en el campo de las aplicaciones web. Los resultados obtenidos contribuirán a trazar estrategias más efectivas para la realización de evaluaciones de seguridad periódicas en las aplicaciones web.

Materiales y métodos o Metodología computacional

En el estudio se establecieron dos preguntas de investigación:

1. ¿Las metodologías de pruebas de penetración son capaces de evaluar las principales vulnerabilidades presentes en las aplicaciones web?
2. ¿Cuáles son las metodologías de pruebas de penetración más adecuadas?

El método analítico-sintético se empleó para extraer las características principales y comparar las principales metodologías de pruebas de penetración, enfocando a su utilización en las aplicaciones web. Para el estudio de las metodologías se establecieron dos instrumentos de evaluación los cuales se presentan en la sección Análisis Comparativo.

En la determinación de las vulnerabilidades más frecuentes en aplicaciones web empleó el método histórico-lógico para el estudio de su evolución mediante el análisis de todos los reportes Top 10 de OWASP, los cuales comenzaron a publicarse a partir del año 2003.

Resultados y discusión

Revista Cubana de Ciencias Informáticas
Vol. 12, No. 4, Octubre-Diciembre, 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 52-65
http://rcci.uci.cu

Vulnerabilidades en Aplicaciones Web

Las vulnerabilidades son errores, fallas, debilidades o exposiciones interna de una aplicación, dispositivo del sistema o servicio que podría conducir a un error de confidencialidad, integridad o disponibilidad (Franklin et al. 2014). La organización OWASP ha coordinado desde el 2003 la elaboración de reportes con las diez vulnerabilidades de seguridad más importantes en aplicaciones web. Un análisis de los reportes emitidos entre los años 2003 y 2017(Stock et al. 2017) muestran las vulnerabilidades más frecuentes:

- Inyección de código: Las vulnerabilidades de inyección de código SQL, HTML, OS, PHP y otros, ocurren cuando la aplicación no está preparada para validar y detectar códigos dañinos que puede ser insertado como parte de una secuencia de datos legítima(Dong et al. 2018).
- Pérdida de Autenticación y Gestión de Sesiones: Las funciones de la aplicación relacionadas con la autenticación y gestión de sesiones son implementadas de forma incorrectamente, lo que posibilita que los ciberatacantes puedan explotar fallas que les permitan tomar la identidad de los usuarios (Calzavara et al. 2017).
- Secuencia de Comandos en Sitios Cruzados (XSS): Las fallas XSS ocurren cuando una aplicación envía datos no confiables al navegador web sin una validación y codificación apropiada (Wang et al. 2017).
- Control de Acceso Interrumpido: Las restricciones a las funciones que los usuarios tienen permiso para utilizar no se cumplen correctamente. Dentro de este grupo de vulnerabilidades se destacan:
    - Referencia Directa Insegura a Objetos: Ocurre cuando un desarrollador expone una referencia a un objeto interno, como un archivo, directorio, o base de datos y no hay un chequeo de control de acceso efectivo.
    - Ausencia de Control de Acceso a Funciones: Las aplicaciones web no verifican los permisos de acceso a nivel de función antes de mostrarlas en la interfaz de usuario.
- Configuración de Seguridad Incorrecta: No se aplican adecuadamente las configuraciones de seguridad propuestas para las aplicaciones, marcos de trabajo, bases de datos, servidores web y sistemas operativos (Akiyama et al. 2017).
- Exposición de datos sensibles: Está provocado por las deficiencias en la protección adecuada de datos sensibles tales como credenciales de cuentas de usuarios, números de tarjetas de crédito y otros datos personales (Mansfield-Devine 2017).

- Falsificación de peticiones en sitios cruzados (CSRF): Esta vulnerabilidad permite que los ciberatacantes puedan obligar al navegador de un usuario autenticado a enviar una petición HTTP manipulada sin conocimiento de este (Martínez et al. 2017).
- Utilización de componentes con vulnerabilidades conocidas: Está dado por la utilización de componentes (librerías, marcos de trabajos, extensiones y otros módulos) con vulnerabilidades conocidas, que debilitan la defensas y amplían la superficie de ataque (Morrison et al. 2017).
- Entidades externas de XML (XXE): Las entidades externas en documentos XML pueden utilizarse para revelar archivos en servidores no actualizados, escaneo de puertos, ejecución de código, entre otros.
- Deserialización insegura: Ocurre cuando una aplicación recibe objetos serializados manipulados para realizar ataques de repetición, inyecciones o elevar sus privilegios de ejecución (Seacord 2017).
- Registro y monitoreo insuficiente: El deficiente monitoreo de los registros de operación de las aplicaciones web permiten a los ciberatacantes vulnerar los controles de seguridad (Shugrue 2017).

Atendiendo a que los reportes de OWASP son ampliamente referenciados en la literatura científica (Bhandari, Jaballah, Jain, Laxmi, Zemmari, Gaur, Mosbah and Conti 2017; Singh and Chatterjee 2017), se tomarán estos como indicadores para compararlos con las metodologías de pruebas de penetración actuales, cuestión que se aborda a continuación.

Pruebas de Penetración

Las pruebas de penetración tuvieron un origen temprano en el año 1965, cuando en una conferencia, especialistas de System Development Corporation (SDC) mostraron una docena de ejemplos de evasiones de seguridad en la computador IBM Q-32 (Hunt 2012). Investigaciones posteriores, financiadas por el Departamento de Defensa de los EE.UU., formalizaron la primera metodología de pruebas de penetración denominada FHM (*Flaw Hypothesis Methodology*) en el año 1973. El surgimiento de la World Wide Web (WWW), incrementó la atención por la seguridad de las aplicaciones y las redes de datos y difundieron las pruebas de penetración como un componente indispensable para mejorar la seguridad de la TIC.

Aunque existe cierta uniformidad entre investigadores y organizaciones respecto al concepto de pruebas de penetración, también hay elementos distintivos que son importante analizar (Dadkhah, Lagzian and Borchardt 2018). El NIST (National Institute of Standards and Technology), lo define como una "prueba de seguridad en la cual, los evaluadores simulan ataques del mundo real en un intento de identificar modos de evadir las características de seguridad de una

aplicación, sistema o red de datos. Las pruebas de penetración requieren la emisión de ataques reales en sistemas y datos reales, empleando las mismas técnicas y herramientas utilizadas por los atacantes actuales".

Según el ISECOM (Institute for Security and Open Methodologies), las pruebas de penetración son un tipo de prueba, de tipo doble ciego o caja negra, donde "el analista interactúa con el objetivo sin ningún conocimiento previo de sus defensas, activos o canales. En la Guía de Pruebas de OWASP, definen las pruebas de penetración como el "arte de probar una aplicación en ejecución remotamente para encontrar vulnerabilidades de seguridad, sin conocer el funcionamiento interno de la aplicación en sí"

Existen diversas metodologías y guías para la realización de pruebas de penetración. En esta sección se analizarán las principales metodologías referenciadas por los autores consultados. En cada caso, los análisis se realizarán desde la perspectiva de las aplicaciones web.

ISSAF

El ISSAF (*Information System Security Assessment Framework*) o Marco de Evaluación de Seguridad de Sistemas de Información, fue desarrollada por la OISSG (Open Information Systems Security Group) (Rathore et al. 2006). El proceso de pruebas de penetración se desarrolla a través de tres fases:

- Fase I. Planificación y Preparación: Comprende los pasos para el intercambio de informaciones iniciales, planificación y preparación para las pruebas de seguridad.
- Fase II. Evaluación: Se aplican las pruebas de seguridad de la metodología de penetración de ISSAF.
- Fase III. Reportes, Limpieza y Destrucción de Artefactos: Toda la información creada y almacenada en los sistemas como parte de las pruebas de seguridad se eliminan.

Desde el punto de vista de las aplicaciones web, puede afirmarse presenta pruebas de seguridad y herramientas válidas, pero no permiten evaluar todos los aspectos requeridos actualmente.

NIST SP 800-115

La Guía Técnica para Evaluaciones y Pruebas de Seguridad de la Información NIST SP 800-115 (*Technical Guide to Information Security Testing and Assessment*), fue publicada en septiembre del 2008 por el NIST(Scarfone et al. 2008). Incluye la realización de pruebas de penetración y propone las siguientes fases:

- Fase de Planificación: Se identifican las reglas que deben seguirse durante la prueba de penetración y se crean las condiciones técnicas y organizativas requeridas.

- Fase de Descubrimiento: Se realiza el escaneo y recopilación de información de la infraestructura computacional de la entidad y el descubrimiento de vulnerabilidades.
- Fase de Ejecución: Se comprueban las vulnerabilidades previamente descubiertas.
- Fase de Documentación y Reporte: Se genera un reporte con los problemas de seguridad encontrados.

La NIST SP 800-115 considera que las evaluaciones de seguridad a nivel de aplicaciones es un tema complejo y por ese motivo no se trata en la metodología. Esto la hace inadecuada para ser empleada por sí sola, en la realización de pruebas de penetración en aplicaciones web.

OSSTMM

OSSTMM (*Open Source Security Testing Methodology Manual*) en su versión 3 fue publicada en el año 2010(Barceló and Herzog 2010) por el ISECOM (*Institute for Security and Open Methodologies*). Consta de cuatro fases:

- Fase de Inducción: Se establece el alcance, los requerimientos y restricciones de la auditoría.
- Fase de Interacción: Se trata de descubrir relaciones entre el alcance, los objetivos y los activos involucrados.
- Fase de requerimientos: Se realizan verificaciones de procesos, de configuraciones, capacitaciones, propiedad intelectual, información expuesta y otros.
- Fase de Intervención: Se enfoca en la penetración de los objetivos y su afectación.

En el caso de las aplicaciones web, no contiene fases, canales o módulos específicos para su evaluación.

PTES

El Estándar para la Ejecución de Pruebas de Penetración o PTES (*Penetration Testing Execution Standard*), es un proyecto constituido por diversas organizaciones y empresas (Amit 2012). Está compuesto por siete fases:

- Preacuerdo: Se define el alcance y los objetivos de la prueba de penetración.
- Recopilación de inteligencia: Se realiza la recolección de información de inteligencia desde fuentes abiertas.
- Modelado de amenazas: Se enuncian las posibles estrategias de penetración.
- Análisis de vulnerabilidades: Se descubren vulnerabilidades que puedan ser explotadas.
- Explotación: Se intentan explotar las vulnerabilidades descubiertas.

- Post explotación: Los especialistas de seguridad pueden continuar escalando el proceso de explotación.
- Reporte: Se comunica al cliente la información que le permita solucionar las vulnerabilidades encontradas.

Debe señalarse que algunas secciones de la metodología aún carecen de descripción y definitivamente no cubren todo el alcance requerido en el campo de las aplicaciones web.

OWASP

La Guía de Pruebas de OWASP (*OWASP Testing Guide*) versión 4, fue publicada en el año 2014 (Meucci and Muller 2014). Está divida en varios grupos de pruebas de seguridad que comprueban aspectos específicos de las aplicaciones web:

- Recopilación de Información.
- Pruebas de seguridad a la configuración y despliegue.
- Pruebas de seguridad a la gestión de la identidad.
- Pruebas de seguridad al proceso de autenticación.
- Pruebas de seguridad al proceso de autorización.
- Pruebas de seguridad al proceso de gestión de sesiones.
- Pruebas de seguridad a la validación de entradas.
- Pruebas de seguridad al manejo de errores.
- Pruebas de seguridad a los mecanismos criptográficos.
- Prueba de seguridad a la lógica de negocios.
- Pruebas de seguridad del lado del cliente.

Teniendo en cuenta que la Guía de Pruebas de OWASP es una metodología para un dominio específico, podía haber desarrollado mejor la fase de reportes. Tiene pruebas de seguridad repetidas en varias fases y tampoco es inusual encontrar fuertes dependencias de pruebas de seguridad entre fases sin abordar cuestiones de cómo gestionar esta interrelación para evitar la repetición de acciones que conllevarán a obtener el mismo resultado.

Análisis comparativo de las principales metodologías de pruebas de penetración

Revista Cubana de Ciencias Informáticas
Vol. 12, No. 4, Octubre-Diciembre, 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 52-65
http://rcci.uci.cu

Durante la caracterización de las principales metodologías de pruebas de penetración, se desarrolló un análisis cualitativo respecto a los principales requerimientos de seguridad en las aplicaciones web. Para profundizar dicho análisis, se creó una escala de evaluación cualitativa (Tabla 1) para analizar cómo se abordan las vulnerabilidades más frecuentes en aplicaciones web. Los resultados de la aplicación de la escala de evaluación se enuncian en la Tabla 2.

Tabla 1. Escala de evaluación de las metodologías respecto a las principales vulnerabilidades de seguridad en las aplicaciones web.

Valor	Descripción
0	No se hace ninguna alusión a la vulnerabilidad ni a pruebas de seguridad o comprobación relacionada con esta.
1	Se hace mención a la vulnerabilidad, pero no se describe como hacer la prueba de seguridad para su detección.
2	Se describe como realizar la prueba de seguridad, pero el contenido presentado no es suficiente para realizar una prueba de seguridad real.
3	Se describe como realizar la prueba de seguridad con suficientes detalles para ser aplicada directamente en una prueba de seguridad real.

Tabla 2. Presencia de pruebas de seguridad asociadas a aplicaciones web.

Principales Vulnerabilidades	NIST SP 800-115	OSSTMM	PTES	ISSAF	OWASP
Inyección de código.	1	1	2	2	2
Pérdida de autenticación y gestión de sesiones.	0	2	1	1	3
Secuencia de comandos en sitios cruzados (XSS).	0	0	1	3	3
Control de acceso interrumpido.	0	1	1	1	3
Referencia directa insegura a objetos	0	1	1	2	3
Ausencia de control de acceso a funciones	0	1	1	2	3
Configuración de seguridad incorrecta.	1	1	1	2	2
Exposición de datos sensibles.	1	1	1	2	2
Falsificación de peticiones en sitios cruzados (CSRF).	0	0	1	0	3
Utilización de componentes con vulnerabilidades conocidas.	1	1	1	1	3
Entidades externas de XML (XXE).	0	1	1	0	3
Deserialización insegura.	0	0	0	0	0
Registro y monitoreo insuficiente.	0	0	0	0	0
Totales	**4**	**10**	**12**	**16**	**30**

A partir de los datos obtenidos puede darse respuesta a las interrogantes de investigación:

1. **¿Las metodologías de pruebas de penetración son capaces de evaluar las principales vulnerabilidades presentes en las aplicaciones web?**

Revista Cubana de Ciencias Informáticas
Vol. 12, No. 4, Octubre-Diciembre, 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 52-65
http://rcci.uci.cu

Como se muestra en la tabla 2, ninguna de las metodologías estudiadas abarca la evaluación completa de las principales vulnerabilidades en las aplicaciones web. La Guía de pruebas de OWASP es la que presenta un nivel de completitud mayor (76%), le siguen ISSAF (41%), PTES (31%), OSSTMM (26%) y finalmente NIST SP 800-115 (10%). Esta comparación puede apreciarse en la Figura 1, donde se compara con el patrón ideal con un valor de 39 puntos según el instrumento diseñado.

Resulta interesante señalar que la metodología de ISSAF, a pesar de ser del año 2006, tiene un mayor nivel de completitud relacionado con vulnerabilidades en aplicaciones web que otras más recientes como PTES u OSSTMM (Figura 1). Desafortunadamente no se han publicado nuevas versiones de esta metodología y esto ha sido un factor determinante que la pone en desventaja respecto a otras más actuales.

Por tanto, puede afirmarse que ninguna de las metodologías de pruebas de penetración analizadas enuncia todas las evaluaciones de seguridad que se requieren para detectar al menos las principales vulnerabilidades en aplicaciones web. Necesitan por tanto un proceso de adaptación y completitud que dependerá de las competencias y experiencias de los equipos de seguridad que tengan la misión de realizarlas.

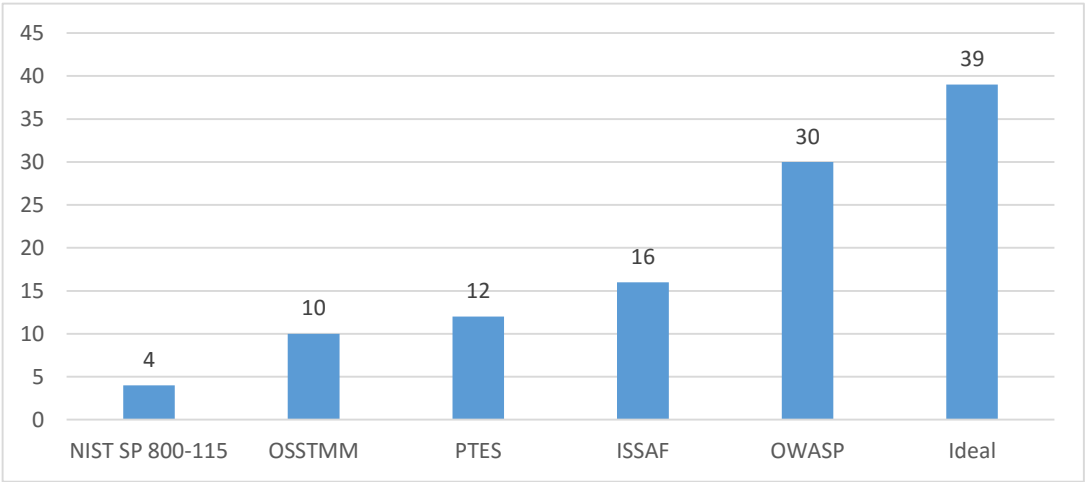

Figura 1. Orden de capacidades para la realización de pruebas de penetración en aplicaciones web.

2. ¿Cuáles son las metodologías de pruebas de penetración más adecuadas?

La Guía de Pruebas de OWASP es la más adecuada para ser tomada como base en una prueba de penetración en aplicaciones web. La comparación con las vulnerabilidades más frecuentes en aplicaciones web, muestran que sus mayores carencias en el caso de la Guía de Pruebas de OWASP está en la necesidad de contar con pruebas de seguridad que permitan evaluar la deserialización insegura y el registro y monitoreo de la aplicación web (Figura 2). También

Revista Cubana de Ciencias Informáticas
Vol. 12, No. 4, Octubre-Diciembre, 2018
ISSN: 2227-1899 | RNPS: 2301
Pág. 52-65
http://rcci.uci.cu

se aprecia la necesidad de integrar pruebas de seguridad en función de comprobar con mayor efectividad problemas de configuración en los servidores web.

Sin embargo, si se compara con otras metodologías de pruebas de penetración, pueden encontrarse importantes deficiencias asociadas al poco o nulo tratamiento de la gestión del proceso. Por ejemplo, se encuentran pruebas de seguridad repetidas entre grupos de pruebas. No se mencionan aspectos organizativos como por ejemplo las actividades de establecimientos de alcances y contratos de confidencialidad entre las partes o procesos de planificación y seguimiento y control (Dalalana Bertoglio and Zorzo 2017; Singh and Chatterjee 2017). Estas cuestiones deberán ser aportadas por los equipos de seguridad.

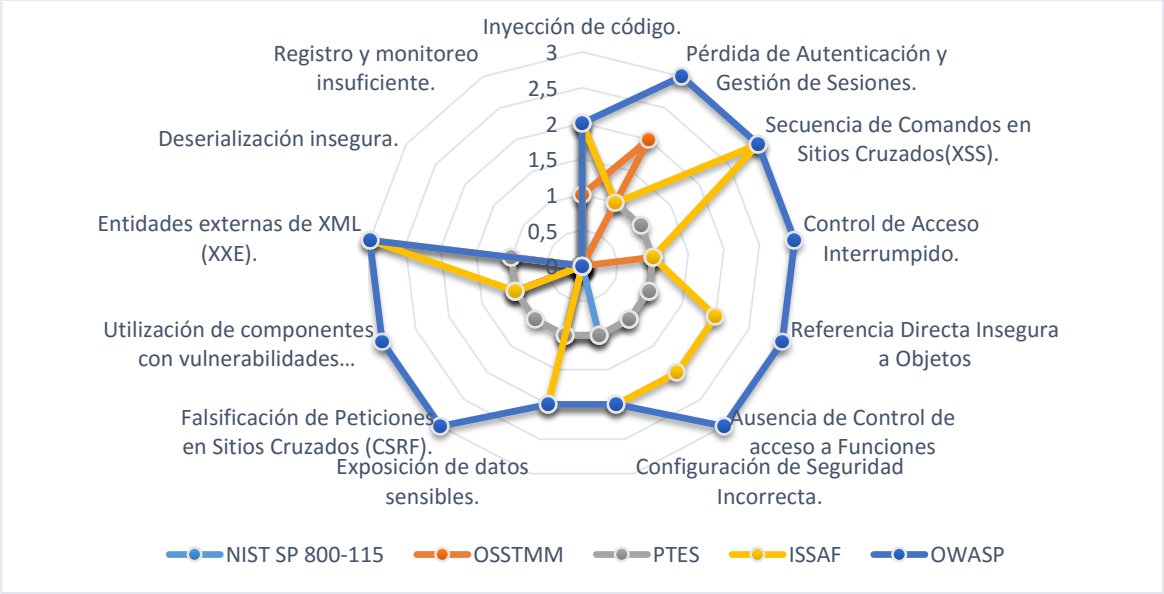

Figura 2. Metodología vs Vulnerabilidades.

Conclusiones

En el artículo se presentaron los resultados de la comparación sobre las capacidades de las metodologías de pruebas de penetración para detectar las principales vulnerabilidades en aplicaciones web. La Guía de Pruebas de OWASP resultó la más completa, seguida de la metodología de ISSAF. No obstante, ninguna metodología demostró ser capaz de brindar métodos, herramientas o pruebas de seguridad para detectar todas las vulnerabilidades comparadas. Los resultados alcanzados demuestran la necesidad de un proceso de adaptación y completamiento de las metodologías existentes ya

que ninguna, por sí sola, contiene todos los elementos requeridos para realizar una evaluación de seguridad actual en aplicaciones web.

El incremento del uso de las aplicaciones web como base para la informatización de servicios en la sociedad, así como las continuas noticias de incidentes de seguridad que involucran este tipo de aplicaciones, hace necesario seguir profundizando en formas de evaluación basadas en pruebas de penetración y otras que permitan minimizar la ocurrencia e impacto de estos incidentes.

Agradecimientos

La presente investigación se ha desarrollado en el marco del Proyecto Metodología Ágil de Prueba de Penetración en Aplicaciones Web, perteneciente al Programa de prioridad nacional de ciencia, tecnología e innovación "Informatización de la Sociedad".

Referencias

AGARWAL, N. AND S. Z. HUSSAIN A closer look on Intrusion Detection System for web applications. arXiv preprint arXiv:1803.06153, 2018.

AKIYAMA, M., T. YAGI, T. YADA, T. MORI, et al. Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots. Computers & Security, 2017, 69, 155-173.

AMIT, I. I. PTES: Penetration Testing Execution Standard. . In.: The Penetration Testing Execution Standard, 2012.

ANTUNES, N. AND M. VIEIRA Assessing and Comparing Vulnerability Detection Tools for Web Services: Benchmarking Approach and Examples. IEEE Transactions on Services Computing, 2015, 8(2), 269-283.

BAJOVIC, V. Criminal Proceedings in Cyberspace: The Challenge of Digital Era. In E.C. VIANO ed. *Cybercrime, Organized Crime, and Societal Responses: International Approaches.* Washington. EE.UU: Springer International Publishing Switzerland, 2017, p. 87-101.

BARCELÓ, M. AND P. HERZOG *OSSTMM: Open Source Security Testing Methodology Manual.* Edtion ed. Barcelona: Institute for Security and Open Methodologies (ISECOM), 2010. 213 p.

BAŞ SEYYAR, M., F. Ö. ÇATAK AND E. GÜL Detection of attack-targeted scans from the Apache HTTP Server access logs. Applied Computing and Informatics, 2018/01/01/ 2018, 14(1), 28-36.

BHANDARI, S., W. B. JABALLAH, V. JAIN, V. LAXMI, et al. Android inter-app communication threats and detection techniques. Computers & Security, 2017, 70, 392-421.

CALZAVARA, S., R. FOCARDI, M. SQUARCINA AND M. TEMPESTA Surviving the Web: A Journey into Web Session Security. ACM Computing Surveys, 2017, 50(1), 13.

DADKHAH, M., M. LAGZIAN AND G. BORCHARDT Academic Information Security Researchers: Hackers or Specialists? Science and Engineering Ethics, 2018, 24(2), 785-790.

DALALANA BERTOGLIO, D. AND A. F. ZORZO Overview and open issues on penetration test. Journal of the Brazilian Computer Society, 2017, 23(1), 2.

DONG, Y., Y. ZHANG, H. MA, Q. WU, et al. An adaptive system for detecting malicious queries in web attacks. Science China Information Sciences, 2018, 61(3), 032-114.

FRANKLIN, J., C. WERGIN AND H. BOOTH CVSS implementation guidance. National Institute of Standards and Technology, NISTIR-7946, 2014.

HUANG, H. C., Z. K. ZHANG, H. W. CHENG AND S. W. SHIEH Web Application Security: Threats, Countermeasures, and Pitfalls. Computer, 2017, 50(6), 81-85.

HUNT, E. US Government Computer Penetration Programs and the Implications for Cyberwar. IEEE Annals of the History of Computing, 2012, 34(3), 4-21.

JHAVERI, M. H., O. CETIN, C. GA, T. MOORE, et al. Abuse Reporting and the Fight Against Cybercrime. ACM Computer Surveys, 2017, 49(4), 1-27.

KNOWLES, W., A. BARON AND T. MCGARR The simulated security assessment ecosystem: Does penetration testing need standardisation? Computers & Security, 2016, 62, 296-316.

MANSFIELD-DEVINE, S. Open source software: determining the real risk posed by vulnerabilities. Network Security, 2017, 2017(1), 7-12.

MARTÍNEZ, S., V. COSENTINO AND J. CABOT Model-based analysis of Java EE web security misconfigurations. Computer Languages, Systems & Structures, 2017, 49, 36-61.

MEUCCI, M. AND A. MULLER *OWASP Testing Guide 4.0*. Edtion ed. EE.UU: OWASP Foundation, 2014. 224 p.

MONTESINO PERURENA, R., W. BALUJA GARCÍA AND J. PORVÉN RUBIER Gestión automatizada e integrada de controles de seguridad informática. Ingeniería Electrónica, Automática y Comunicaciones, 2013, 34(1), 40-58.

MORRISON, P., B. H. SMITH AND L. WILLIAMS 2017. Surveying Security Practice Adherence in Software Development. In *Proceedings of the Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, Hanover, MD, USA2017 ACM, 3055312, 85-94.

NAZIR, S., S. PATEL AND D. PATEL Assessing and augmenting SCADA cyber security: A survey of techniques. Computers & Security, 2017, 70, 436-454.

RAHALKAR, S. A. *Certified Ethical Hacker (CEH) Foundation Guide*. Edtion ed. Pune, Maharashtra: Springer, 2016. 207 p.

RATHORE, B., M. BRUNNER, M. DILAJ, O. HERRERA, et al. *Information Systems Security Assessment Framework (ISSAF)*. Edtion ed. Colorado Springs: Open Information Systems Security Group, 2006. 845 p.

SANDHYA, S., S. PURKAYASTHA, E. JOSHUA AND A. DEEP. Assessment of website security by penetration testing using Wireshark. In *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 2017, p. 1-4.

SCARFONE, K., M. SOUPPAYA, A. CODY AND A. OREBAUGH *NIST SP 800-115: Technical Guide to Information Security Testing and Assessment.* Edtion ed. Maryland: National Institute of Standards and Technology, 2008. 80 p.

SEACORD, R. C. Java Deserialization Vulnerabilities and Mitigations. In *2017 IEEE Cybersecurity Development (SecDev)*. 2017, p. 6-7.

SHUGRUE, D. Fighting application threats with cloud-based WAFs. Network Security, 2017, 2017(6), 5-8.

SINGH, A. AND K. CHATTERJEE Cloud security issues and challenges: A survey. Journal of Network and Computer Applications, 2017, 79, 88-115.

STOCK, A. V. D., B. GLAS, N. SMITHLINE AND T. GIGLER *OWASP Top 10 2017. The Ten Most Critical Web Application Security Risks*. Edtion ed. EE.UU: The OWASP Foundation, 2017. 50 p.

TOPPER, J. Compliance is not security. Computer Fraud & Security, 2018, 2018(3), 5-8.

WANG, R., G. XU, X. ZENG, X. LI, et al. TT-XSS: A novel taint tracking based dynamic detection framework for DOM Cross-Site Scripting. Journal of Parallel and Distributed Computing, 2017.

WEI, X. AND M. WOLF A Survey on HTTPS Implementation by Android Apps: Issues and Countermeasures. Applied Computing and Informatics, 2017, 13(2), 101-117.