PLANTILLA OFICIAL PARA LA PRESENTACIÓN DE TRABAJOS
II CONVENCIÓN CIENTÍFICA INTERNACIONAL
"II CCI UCLV 2019"

DEL 23 AL 30 DE JUNIO DEL 2019.
CAYOS DE VILLA CLARA. CUBA.

II CONFERENCIA INTERNACIONAL DE PROCESAMIENTO DE LA INFORMACIÓN (CIPI2019)

Análisis de las metodologías de pruebas de penetración para detectar vulnerabilidades en aplicaciones web

Analysis of penetration test methodologies to detect vulnerabilities in web applications

Henry Raúl González Brito[1], Raydel Montesino Perurena[2], Dainys Gainza Reyes[3]

1-Universidad de las Ciencias Informáticas, Cuba, henryraul@uci.cu
2- Universidad de las Ciencias Informáticas, Cuba, raydelmp@uci.cu
3-Universidad de las Ciencias Informáticas, Cuba, dgainza@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.

Información de contacto
convencionuclv@uclv.cu
www.uclv.edu.cu

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, 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.

Palabras Clave: Seguridad Informática, Análisis de Vulnerabilidades, Pruebas de Penetración, Aplicaciones Web, OWASP

Keywords: Computer security, Vulnerability Analysis, Penetration Test, Web Application, OWASP

1. 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 [1]. Desafortunadamente, la presencia de vulnerabilidades en sistemas informáticos, aumenta continuamente, no solo en número sino también en el impacto de su explotación individual [2].

Uno de los principales blancos de ciberataques son las aplicaciones web [3-6]. Las organizaciones y personas interactúan fundamentalmente en el ciberespacio a través de aplicaciones web, mediante navegadores y dispositivos móviles [7, 8].

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 [9, 10], pero su presencia no ha sido suficiente para disminuir o contener los ciberataques [11-13].

Un componente importante para mitigar estos problemas son las pruebas de penetración, conocidas también como pentesting o hacking ético, las cuales forman parte de las evaluaciones de seguridad [14, 15] y su empleo contribuye a garantizar que los sistemas informáticos y redes de datos cumplen con las normas y mecanismos de seguridad y puedan ofrecer la mayor protección contra las amenazas comunes.

En años recientes, diversos autores [14, 16-18] han publicado estudios que expresan deficiencias de las metodologías de pruebas de penetración ante determinados escenarios y dominios tecnológicos. 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.

2. Metodología

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.

3. Resultados y discusión

3.1.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 [19]. Un análisis de todos los reportes emitidos por OWASP[20] muestran las vulnerabilidades web más frecuentes:

V1. Inyección de código: Ocurren cuando la aplicación no está preparada para detectar códigos dañinos que puede ser insertado en secuencias de datos legítimos.

V2.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 y permitan tomar la identidad de los usuarios.

V3. Secuencia de Comandos en Sitios Cruzados (XSS): Ocurren cuando una aplicación envía datos no confiables al navegador web sin una validación y codificación apropiada.

V4. Control de Acceso Interrumpido: Las restricciones a las funciones que los usuarios tienen permiso para utilizar no se cumplen correctamente.

V5. Configuración de Seguridad Incorrecta: No se aplican adecuadamente las configuraciones de seguridad propuestas para los sistemas informáticos.

V6. Exposición de datos sensibles: Deficiencias en la protección adecuada de datos como credenciales de cuentas de usuarios, números de tarjetas de crédito y otros.

V7. Falsificación de peticiones en sitios cruzados (CSRF): Los ciberatacantes pueden obligar al navegador de un usuario autenticado a enviar una petición HTTP manipulada sin conocimiento de este.

V8. Utilización de componentes con vulnerabilidades conocidas: Está dado por la utilización de componentes vulnerables que debilitan las defensas.

V9. Entidades externas de XML (XXE): Pueden utilizarse para revelar archivos en servidores no actualizados, escaneo de puertos, ejecución de código, entre otros.

V10. Deserialización insegura: Ocurre cuando una aplicación recibe objetos serializados manipulados para realizar ataques de repetición, inyecciones o elevar sus privilegios de ejecución.

V11. 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.

3.2. Pruebas de Penetración

Las pruebas de penetración tuvieron un origen temprano en el año 1965 [21], investigaciones posteriores formalizaron la primera metodología de pruebas de penetración denominada FHM (Flaw Hypothesis Methodology) en el año 1973. El NIST (National Institute of Standards and Technology), las 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"[22].

Según el ISECOM[23], 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. Para OWASP es el "arte de probar una aplicación en ejecución remotamente para encontrar vulnerabilidades de seguridad, sin conocer el funcionamiento interno de la aplicación en sí"[24]

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.

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) [25]. 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[26]. 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[27] por el ISECOM (Institute for Security and Open

Methodologies). 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 [28]. 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 Testing Guide

La Guía de Pruebas de OWASP (OWASP Testing Guide) versión 4, fue publicada en el año 2014 [24]. 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.

3.3.Análisis comparativo de las principales metodologías de pruebas de penetración

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. Presencia de pruebas de seguridad asociadas a 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 cómo hacer la prueba de seguridad para su detección.
2	Se describe cómo realizar la prueba de seguridad, pero el contenido presentado no es suficiente para realizar una prueba de seguridad real.
3	Se describe cómo 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:

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

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%).

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 un proceso de adaptación y completitud que dependerá de la experiencias de los equipos de seguridad.

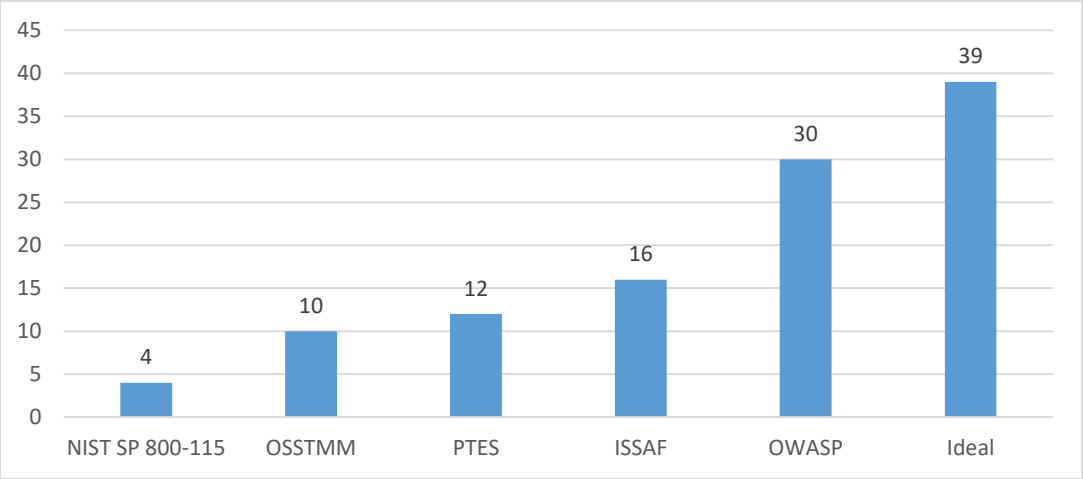

Figura 1. Orden de capacidades para la realización de pruebas de penetración en aplicaciones web.

¿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 (**¡Error! No se encuentra el origen de la referencia.**2). También 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.

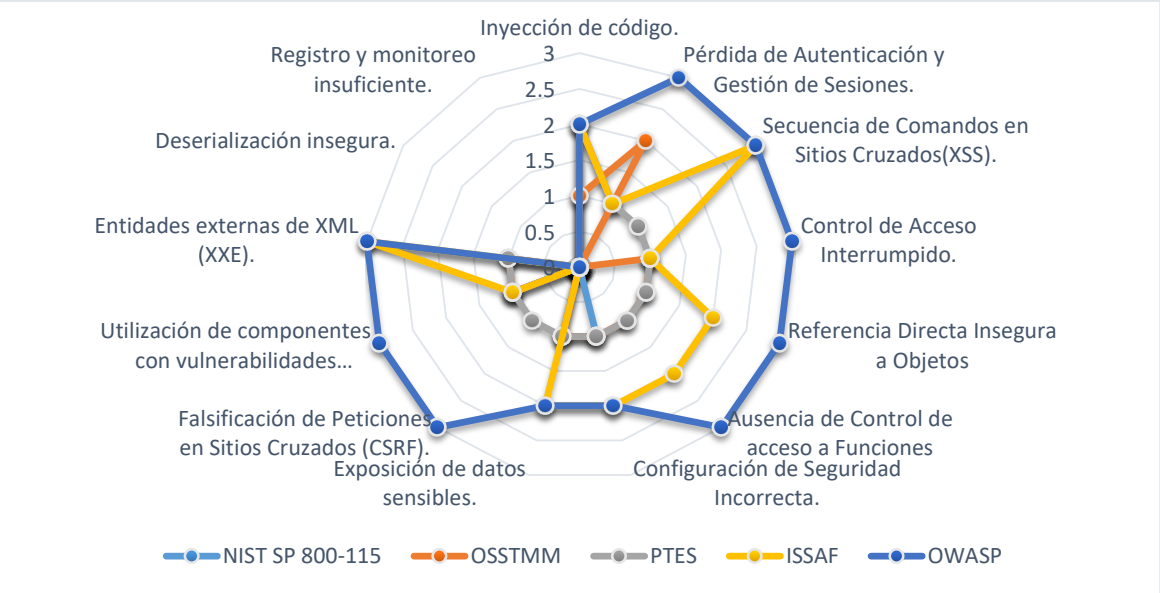

Figura 2. Metodología vs Vulnerabilidades.

4. 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.

5. Referencias bibliográficas

1. Dadkhah, M., M. Lagzian, and G. Borchardt, *Academic Information Security Researchers: Hackers or Specialists?* Science and Engineering Ethics, 2018. **24**(2): p. 785-790.
2. Huang, H.C., et al., *Web Application Security: Threats, Countermeasures, and Pitfalls.* Computer, 2017. **50**(6): p. 81-85.
3. Bajovic, V., *Criminal Proceedings in Cyberspace: The Challenge of Digital Era*, in *Cybercrime, Organized Crime, and Societal Responses: International Approaches*, E.C. Viano, Editor. 2017, Springer International Publishing Switzerland: Washington. EE.UU. p. 87-101.
4. Jhaveri, M.H., et al., *Abuse Reporting and the Fight Against Cybercrime.* ACM Computer Surveys, 2017. **49**(4): p. 1-27.
5. Agarwal, N. and S.Z. Hussain, *A closer look on Intrusion Detection System for web applications.* arXiv preprint arXiv:1803.06153, 2018.
6. González Brito, H.R., *Estudio de patrones de intentos de ciberataques asociados a las vulnerabilidades del complemento RevSlider.* Revista Cubana de Ciencias Informáticas, 2018. **12**(1): p. 43-57.
7. Wei, X. and M. Wolf, *A Survey on HTTPS Implementation by Android Apps: Issues and Countermeasures.* Applied Computing and Informatics, 2017. **13**(2): p. 101-117.
8. Bhandari, S., et al., *Android inter-app communication threats and detection techniques.* Computers & Security, 2017. **70**: p. 392-421.
9. 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): p. 40-58.
10. Topper, J., *Compliance is not security.* Computer Fraud & Security, 2018. **2018**(3): p. 5-8.
11. Singh, A. and K. Chatterjee, *Cloud security issues and challenges: A survey.* Journal of Network and Computer Applications, 2017. **79**: p. 88-115.

12. Nazir, S., S. Patel, and D. Patel, *Assessing and augmenting SCADA cyber security: A survey of techniques.* Computers & Security, 2017. **70**: p. 436-454.
13. 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. **14**(1): p. 28-36.
14. Rahalkar, S.A., *Certified Ethical Hacker (CEH) Foundation Guide*. 2016, Pune, Maharashtra: Springer. 207.
15. Sandhya, S., et al. *Assessment of website security by penetration testing using Wireshark*. in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. 2017.
16. Knowles, W., A. Baron, and T. McGarr, *The simulated security assessment ecosystem: Does penetration testing need standardisation?* Computers & Security, 2016. **62**: p. 296-316.
17. Dalalana Bertoglio, D. and A.F. Zorzo, *Overview and open issues on penetration test.* Journal of the Brazilian Computer Society, 2017. **23**(1): p. 2.
18. 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): p. 269-283.
19. Franklin, J., C. Wergin, and H. Booth, *CVSS implementation guidance.* National Institute of Standards and Technology, NISTIR-7946, 2014.
20. Stock, A.v.d., et al., *OWASP Top 10 2017. The Ten Most Critical Web Application Security Risks*. 2017, EE.UU: The OWASP Foundation. 50.
21. Hunt, E., *US Government Computer Penetration Programs and the Implications for Cyberwar.* IEEE Annals of the History of Computing, 2012. **34**(3): p. 4-21.
22. Stouffer, K., J. Falco, and K. Scarfone, *NIST SP 800-115: Technical Guide to Information Security Testing and Assessment*. 2008, Maryland: National Institute of Standards and Technology.
23. Shrestha, N., *Security Assessment via Penetration Testing: Network and System Administrator's Approach: Security, Network and System Administrator, Penetration Testing*. 2012.
24. Meucci, M. and A. Muller, *OWASP Testing Guide 4.0*. 2014, EE.UU: OWASP Foundation. 224.
25. Rathore, B., et al., *Information Systems Security Assessment Framework (ISSAF)*. 2006, Colorado Springs: Open Information Systems Security Group. 845.
26. Scarfone, K., et al., *NIST SP 800-115: Technical Guide to Information Security Testing and Assessment.* 2008, Maryland: National Institute of Standards and Technology. 80.
27. Barceló, M. and P. Herzog, *OSSTMM: Open Source Security Testing Methodology Manual.* 2010, Barcelona: Institute for Security and Open Methodologies (ISECOM). 213.
28. Amit, I.I., *PTES: Penetration Testing Execution Standard.* . 2012, The Penetration Testing Execution Standard.