Análisis de Vulnerabilidades de Sistemas Web en desarrollo y en producción

Juan C. Cuevas, Roberto M. Muñoz, M. Alejandra Di Gionantonio, Iris N. Gastañaga, Fabián A. Gibellini, Germán Parisi, Diego Barrionuevo, Milagros Zea Cárdenas.

Laboratorio de Sistemas / Dpto. de Ingeniería en Sistemas / Universidad Tecnológica Nacional / Facultad Regional Córdoba. Cruz Roja S/N, 5016

juancarloscue@hotmail.com, robertomunioz@gmail.com, ing.alejandradg@gmail.com, irisg@ciec.com.ar, fgibellini@bbs.frc.utn.edu.ar, germannparisi@gmail.com, santosdiegob@gmail.com, milyzc@gmail.com

Resumen

La realización de pruebas de penetración permiten detectar las vulnerabilidades de los sistemas de información, en el caso particular del proyecto de investigación y desarrollo que se lleva a cabo en UTN FRC, se han realizado acciones de esta metodología en sistemas web que desarrollan los estudiantes del último año de la carrera de Ingeniería en Sistemas de Información y con acciones similares que se realizaron en Sistemas web en producción de empresas privadas nacionales e internacionales.

Se presentará el abordaje teórico, la metodología utilizada y las técnicas con las que se llevan a cabo diversas pruebas manuales, se muestran los resultados obtenidos de aplicar la metodología para luego sentar las bases en búsqueda de la repetición de dichas pruebas ante un mismo sistema objetivo generando su automatización.

Palabras clave: Seguridad web; Auditorías; Ethical Hacking; Vulnerabilidad; Pentesting.

Contexto

Este proyecto de investigación se lleva a cabo en el ámbito del Laboratorio de Sistemas (LabSis) de la Universidad Tecnológica Nacional – Facultad Regional Córdoba (UTN - FRC).

Se enmarca dentro de las líneas Seguridad Informática y forma parte del proyecto de investigación Sistema Integrado de Soporte para Análisis de Vulnerabilidades en Sistemas Web (S.I.S.A.V.S.W.). Código: EIUTNCO0004084. Acreditado por la Secretaría de Ciencia, Tecnología y Postgrado, y financiado por la UTN – FRC.

Introducción

La seguridad de la información describe actividades relativas a la protección de la información y los activos de la infraestructura de la información contra riesgos de pérdida, uso inadecuado, revelación o daño.

Los riesgos de estos activos pueden ser calculados mediante el análisis de las siguientes cuestiones:

- Amenazas a sus activos: Eventos no deseados que pueden causar pérdida, daño o uso inadecuado de los activos en forma deliberada o accidental.
- Vulnerabilidades: Se refiere a cuán susceptibles son sus activos a ataques.
- Impacto: La magnitud de la pérdida potencial o la seriedad del evento.

Existen disponibles normas, modelos y estándares para asistir a las organizaciones en la implementación de programas y controles apropiados para mitigar estos riesgos, como por

ejemplo, las normas ISO como por ejemplo la ISO 27.001 [1] , los modelos ITIL [2], COBIT [3] y estándares del NIST [4].

En lo referido a las vulnerabilidades que es de interés para las pruebas de penetración diferentes autores abordan la temática.

Xie et al postulan que muchas de las vulnerabilidades de seguridad en las aplicaciones actuales son introducidas por los desarrolladores de software al escribir código inseguro los cuales se pueden deber a la falta de comprensión de programación segura [6].

Bates et al postulan que en los últimos años, los fabricantes de navegadores e investigadores han tratado de desarrollar filtros del lado del cliente para mitigar estos ataques. Pero algunos de estos filtros podrían introducir vulnerabilidades en sitios que antes estaban libres de errores. [7]

Tripp et al postulan que los autores de publicaciones recientes sugieren que las aplicaciones web son altamente vulnerables a ataques de seguridad. En referencia a lo cual citan un reporte reciente de la WASC que proveen estadísticas de seguridad sobre 12186 sitios web en producción, listando un total de 97554 vulnerabilidades detectadas en estos sitios web. Más severamente aún es que cerca del 49% de los sitios analizados se encontró que contenían vulnerabilidades de alto riesgo [8].

Según Shahriar las aplicaciones web contienen vulnerabilidades, las cuales pueden conducir a serias brechas de seguridad tal como el robo de información confidencial [9].

Grossman en un reporte indica que las aplicaciones web de diferentes dominios de uso más frecuente (por ejemplo: banca, salud, TI, educación, redes sociales) son más propensas a ser vulneradas. [10]

Ha habido una gran puja por incluir la seguridad durante el ciclo de desarrollo de software y formalizar la especificación de requerimientos de una manera estándar. Además de un gran incremento en la cantidad de organizaciones dedicados a la seguridad de aplicaciones como la Open Web Application Security Project (OWASP) [11]. Sin embargo, todavía hay aplicaciones web descaradamente vulnerables, principalmente porque los programadores están más concentrados en la funcionalidad que en la seguridad [12].

En referencia a los ataques, según el último top 10 de las OWASP realizado cada 3 años, esta organización lista los ataques más usados para explotar vulnerabilidades. Estos son: Inyección (SQL, OS y LDAP), pérdida de autenticación y gestión de sesiones, cross site scripting (XSS), referencia directa e insegura a objetos, configuración de seguridad Incorrecta, exposición de datos sensibles, inexistente control de acceso a funcionalidades, falsificación de peticiones en sitios cruzados (CSRF), uso de componentes con vulnerabilidades conocidas, reenvíos y redirecciones no válidas. [5]

Respecto a las metodologías que son pasibles de ser utilizadas, existen mejores prácticas sobre las cuales basar la realización de este tipo de evaluación, aunque en general, cada profesional puede incorporar sus variantes. Algunos ejemplos pueden ser: el documento del NIST [4], el documento del Open Source Security Testing Methodology Manual (OSSTMM) [13], el marco de trabajo denominado Information System Security Assessment Framework (ISAAF) [14] y el Open Web Application Security Project (OWASP) [11].

Además existen herramientas que facilitan a los pentester realizar ataques de Injection, particularmente, existen varias relacionadas al ataque SQL Injection como por ejemplo: SQLmap, Havij y V1p3R. Estas herramientas se caracterizan principalmente por contener vectores de ataques más usados permitiendo automatizar un ataque [14] [15].

Cross-Site Scripting (XSS) aborda otro tipo de ataque común en aplicaciones web que consisten en aprovechar las características de los lenguajes que se ejecutan en el navegador, tales como el lenguaje JavaScript. Yusof y Pathan referencian tres tipos de ataques XSS: persistente, no persistente y basado en el DOM [16]. Hay varias maneras de prevenir estos

ataques XSS: mediante el "sanitizado" de todas las entradas de información al sistema [16] o un método que haga inútil la cookie que pueda robar el atacante [17].

En los siguientes puntos se define la metodología que se lleva a cabo para realizar las pruebas y obtener información para la generación de las bases que permitirán el desarrollo de un sistema automatizado de pruebas de penetración para los sistemas objetivos.

Metodología y técnicas

Entre las actividades desarrolladas en este proyecto orientadas a abordar esta temática se procedió a realizar una pormenorizada identificación y clasificación de técnicas y herramientas. Con respecto a estas últimas se realizó una búsqueda y selección de las mismas en el mercado, bibliografía, publicaciones científicas y en la web durante el 2016 - 2017. Inicialmente lo único que tenían que cumplir las herramientas era que fueran de software libre. Los criterios de selección fueron empíricos, ya que los integrantes del equipo de pentesters analizaron sus salidas en pruebas de seguridad realizadas a empresas privadas y a desarrollos de proyectos de la materia Proyecto Final en la carrera de Ingeniería en Sistemas de Información de la UTN FRC, verificaban si cada herramienta probada era lo suficientemente confiable y sus salidas eran comprensibles. Si cumplían con estos criterios, la herramienta era utilizada por los pentesters en próximas pruebas de seguridad.

Se realizaron trabajos de pentest a un proyecto de la materia de proyecto del último año de la carrera de Ingeniería en Sistemas de Información de la UTN FRC, con metodología de ethical hacking y dos pentest a una empresa privada nacional y otra internacional. Llevando a cabo las mismas acciones en todas:

- **Visibilidad** de tipo caja negra: El tester de seguridad no cuenta con ninguna información del objetivo. El tester de seguridad no tiene accesos, la única información sobre vulnerabilidades serán las que se puedan determinar mediante técnicas de prueba y "adivinación".

- **Posicionamiento** Externo: El posicionamiento externo brinda al tester de seguridad la posición de atacantes externos a la organización de donde suelen provenir la mayor cantidad de ataques, en la mayoría de las organizaciones el nivel de madurez de la protección perimetral es alto. El tester de seguridad realiza las pruebas desde fuera de la organización. Busca probar el perímetro expuesto a internet.

- **Etapas del análisis de seguridad:**
  - Reconocimiento Pasivo
  - Reconocimiento activo superficial.
  - Reconocimiento activo en profundidad.
  - Análisis de Vulnerabilidades.

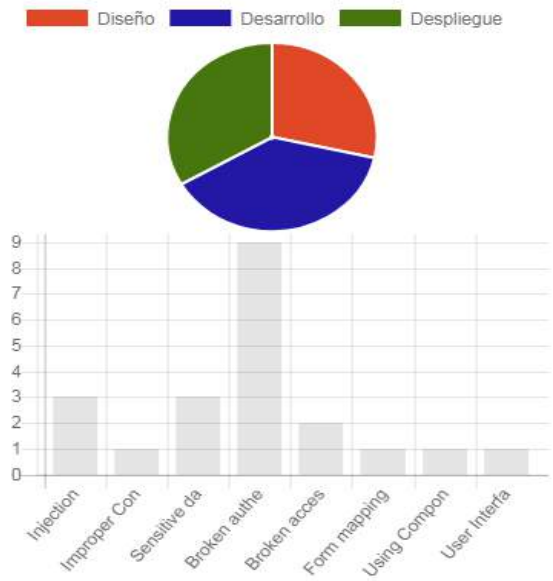
De lo expuesto anteriormente se obtuvo como resultados los siguientes valores expresados en una tabla.
La tabla 1 identifica las categorías, basadas en OWASP y de acuerdo a las etapas del ciclo de desarrollo donde se han identificado las vulnerabilidades.
En la tabla no se encuentran todas categorías definidas por OWASP debido a que muchas de estas no han sido detectadas como vulnerabilidades en los sistemas objetivos.
En el caso de la categoría Form Mapping, no se encuentra identificada en las OWASP, pero se pudo detectar vulnerabilidades en los frameworks web que se analizaron.

Tabla 1

Categoria	Diseño	Desarrollo	Despliegue	Total
Injection	0	3	0	3
Improper Control of Interaction Frequency	0	1	0	1
Sensitive data exposure	1	1	1	3
Broken authentication	5	0	4	9
Broken access control	0	2	0	2
Form mapping	0	1	0	1
Using Components with Known Vulnerabilities	0	0	1	1
User Interface Misrepresentation of Critical Information	0	0	1	1
Total	6	8	7	21

Se puede observar (Gráfico 1) que los mecanimos de desarrollo de autenticación son los que mas fallan y se nota que hay un pico alto de bugs de inyecciones y esto concuerda con el top tep de 2017 de OWASP

Gráfico 1



El resto de las vulnerabilidades detectadas a pesar de que en números es menor, no deja de ser un llamado de atención para los líderes de estos sistemas.

Líneas de Investigación, Desarrollo e Innovación

Este proyecto se inscribe dentro de esta línea de investigación de seguridad de la información, más específicamente pruebas de penetración en sistemas web en producción, de forma que permitan identificar las vulnerabilidades existentes dentro de un ambiente controlado.

Resultados y Objetivos

Como resultado del análisis surgen algunas reflexiones primigenias: Los Sistemas web objetivos de estudio poseen vulnerabilidades bien conocidas, ya sea en los desarrollos de los estudiantes como en el ámbito profesional privado en empresas nacionales e internacionales. Así mismo, es necesario tener en cuenta que las acciones que se deben llevar a cabo para realizar estas pruebas y poder obtener datos fehacientes dependen de un acuerdo de confiabilidad y confidencialidad de los propietarios del sistema con el equipo que lleva a cabo la metodología de ethical hacking por lo que se dificulta poder contar con gran cantidad de muestras para poder llegar a conclusiones definitivas, pero las que aquí se pudieron obtener constituyen, el comienzo de un camino que permita definir las bases de una sistema automatizado de pruebas de penetración. Otro aspecto que se suma también como dificultad son los tiempos que demanda llevar estas acciones sobre los sistemas web.

En la práctica, a partir de experiencias de los integrantes del equipo se identificaron, por un lado, la necesidad de trabajar fuertemente en incluir los requerimientos de seguridad en la planificación de nuevos proyectos de desarrollo de productos software, como así también la necesidad de gestionar la ejecución de múltiples pruebas de penetración en el contexto de la seguridad de la información de sistemas web en producción, basándose en metodologías abiertas, para identificar y analizar sus vulnerabilidades. Para lo que es necesario:

- Permitir a los pentesters identificar vulnerabilidades y automatizar el proceso de identificación.

- Lograr que el sistema emita un diagnóstico respecto a las vulnerabilidades del sistema web analizado.
- Crear una base de datos que facilite al pentester vincular metodologías, técnicas y herramientas para abordar la evaluación de vulnerabilidades de un sistema web determinado.
- Generar un sistema que contribuya y facilite el desarrollo de las actividades del pentester.
- Obtener un producto (sistema) que sea simple de utilizar por los profesionales de la seguridad.

Formación de Recursos Humanos

En el equipo trabajarán estudiantes avanzados de la carrera de Ingeniería en Sistemas de Información, los cuales actualmente se desempeñan en el Laboratorio de Sistemas (LabSis), con la finalidad de que inicien su formación en investigación científica y tecnológica profundizando sus conocimientos en temas significativos en la seguridad de la información. Los estudiantes podrán realizar la Práctica Supervisada.

Se llevan a cabo competencias de casos de test de penetración ético, para fomentar en los estudiantes el pensamiento en el desarrollo seguro, e inculcar conceptos de ética ante estas acciones.

Se realizaron capacitaciones a estudiantes sobre cómo resolver vulnerabilidades bien conocidas, a partir de realizar pruebas de penetración ético.

Referencias

[1] ISO/IEC 27001. "Tecnología de la información". Técnicas de la seguridad. Sistemas de gestión de la seguridad de la información (SGSI). Requisitos. ISO Ginebra, Suiza 2013.

[2] ITIL. Information Technology Infrastructure Library. Vs. 3 - 2001.

[3] COBIT Control Objectives for Information and Technology. Vs. 5 - 2012.

[4] Technical Guide to Information Security Testing and Assessment. SP 800-115. NIST National Institute of Standards and Technology. 2008.

[5] "OWASP top 10 2013 Project". Open Web Application Security Project. https://www.owasp.org/index.php/Top_10_2013-Top_10. 2013.

[6] XIE, J.; Chu B.; Lipfort, H. R.; Melton, J. T.: "ASIDE: IDE Support for Web Application Security". ACSAC '11, Orlando, Florida USA. Dec. 5-9/2011

[7] Bates,D; Barth, A.; Jackson, C.: "Regular Expressions Considered Harmful in

Client-Side XSS Filters", Raleigh, NC, USA. April 26-30/2010.

[8] Tripp, O.; Weisman, O. y Guy, L.: "Finding Your Way in the Testing Jungle". ISSTA '13, July 15-20/2013, Lugano, Switzerland.

[9] Shahriar, H.: "Security Vulnerabilities and Mitigation Techniques of Web

Applications". SIN'13, November 26-28/2013, Aksaray, Turkey.

[10] Grossman, J.: "How does your website security stack up against peers?" White Hat Report, Summer. 2012.

[11] Guía OWASP (Open Web Application Security Project). Vs. 3.

[12] Josh Pauli: "The Basics of Web Hacking: Tools and Techniques to Attack the Web". Ed. ELSEVIER. 2013. 25 Wymnan Street, Waltham, MA 02451, USA.

[13] "Open Source Security Testing Methodology Manual (OSSTMM)". Institute for Security and Open Methodologies (ISECOM). Diciembre 2010. Cataluña. España.

[14] Ciampa, A.; Visaggio, C. A.; y Di Penta, M. "A heuristic-based approach for detecting SQL-injection vulnerabilities in Web applications". SESS. Cape Town, South Africa. Mayo 2/2010.

[15] Nagpal, B; Chauhan, N.; Singh, N.; Panesar, A.: "Tool Based Implementation of SQL Injection for Penetration Testing". International Conference on Computing, Communication and Automation (ICCCA2015). 2015

[16] Yusof, I; Pathan, A. S.; "Preventing Persistent Cross-Site Scripting (XSS) Attack By Applying Pattern Filtering Approach". IEEE. 2014.

[17] Takahashi H.; Yasunaga K.; Mambo M.; Kim K.; Youl Youm H.: "Preventing Abuse of Cookies Stolen by XSS". Eighth Asia Joint Conference on Information Security. IEEE 2013.