EVALUACIÓN DE SEGURIDAD DE APLICACIONES WEB

Ana Funes, Aristides Dasso

SEG / Departamento de Informática / Facultad de Ciencias Físico-Matemáticas y Naturales / Universidad Nacional de San Luis

Ejército de los Andes 950, D5700HHW San Luis, Argentina

+54 (0) 266 4520300, ext. 2126

{afunes, arisdas}@unsl.edu.ar

RESUMEN

Aquí presentamos los objetivos, lineamientos generales y resultados esperados de una línea de investigación sobre la creación de modelos de evaluación de seguridad informática en organizaciones. Como parte integral del desarrollo de modelos de evaluación de sistemas complejos, y considerando que la evaluación de la estructura y metodología de implementación de un sistema de desarrollo de software, que incluya las tareas necesarias para la implementación de medidas de seguridad en el resultado final del sistema, implica una evaluación de un sistema complejo, es que esta investigación tiene como objetivo la creación, puesta a punto y aplicación de diversos modelos que permitan obtener indicadores del nivel alcanzado en la implementación de medidas de seguridad de aplicaciones web.

La metodología a seguir para el desarrollo de dichos modelos de evaluación está basada en la aplicación del método Logic Score of Preference (LSP) [8]. Asimismo, hemos tomado como referencia para la creación del modelo, estándares reconocidos como la Web Security Testing Guide de la OWASP [11], que sirven de guía a las organizaciones para formular e implementar estrategias para la seguridad del software.

Palabras clave: Redes. Seguridad Web. Métodos de Evaluación. Logic Score of Preference (LSP).

CONTEXTO

El trabajo de investigación aquí presentado se encuentra enmarcado dentro del ámbito del SEG (Software Engineering Group), de la Universidad Nacional de San Luis, ejecutándose dentro de una de las líneas de investigación del Proyecto de Incentivos código 22/F222 “Ingeniería de Software: Conceptos, Prácticas y Herramientas para el Desarrollo de Software de Calidad”, dirigido por el Dr. Daniel Riesco y co-dirigido por el Dr. Roberto Uzal. El mismo se encuentra acreditado con evaluación externa y financiamiento de la Universidad Nacional de San Luis.

INTRODUCCIÓN

En una organización, una vez elegido un proceso de desarrollo de software que implique la aplicación de metodologías de seguridad, resulta imperioso tener un modelo que permita, dentro de dicho proceso de desarrollo de software, la evaluación de las normas y reglas de seguridad a implementar. Este modelo, debe permitir conocer y controlar el grado de la puesta en práctica con la cual la organización lleva adelante sus políticas, actividades, usa sus herramientas y métodos, etc., en pos de su seguridad.

Si bien la construcción de modelos de evaluación de sistemas complejos, entre los que se encuentran los sistemas de seguridad informática, constituye una necesidad importante, no es una tarea sencilla. Múltiples aspectos deben ser considerados en esta tarea, teniendo en cuenta no solo los aspectos físicos, tales como las instalaciones y sus políticas de acceso, sino también medidas de seguridad del software, así como firewalls, permisos, codificación en línea, etc.

Por lo tanto, para una organización preocupada en su seguridad informática, resulta necesario contar con estándares así como con herramientas apropiadas para evaluar el grado de adecuación con dichos estándares.

En este sentido, existen en la literatura y en la web múltiples propuestas. En [4] hay una interesante revisión de la literatura del tema. Así, por ejemplo, empleando un sistema interactivo basado en grafos, desarrollado en el contexto de estándares de codificación segura para el manejo de vulnerabilidades, en [6] los autores esperan superar los posibles errores humanos que podrían aparecer cuando se aplican dichos estándares de codificación segura.

Khairul Anwar Sedek et al. [10] emplean el Open Source Security Testing Methodology (OSSTMM) usando las Top Ten Critical Vulnerabilities definidas por la OWASP, y crean un modelo aditivo para evaluar la performance basada en la propuesta de la Open Source Security Testing Methodology (OSSTMM).

Jun Zhu et al. [9] encaran el problema a través de un análisis estático interactivo, que integra análisis estático dentro de IDEs (Integrated Development Environment) para proveer, in-situ, programación segura de manera de ayudar a los desarrolladores en prevenir las vulnerabilidades en la etapa de construcción del código.

En [13] se presenta una revisión de las distintas vulnerabilidades de seguridad que se usan para asegurar la capa de aplicación web, los abordajes y técnicas empleadas en el proceso, así como las fases del desarrollo de software en las cuales se enfatizan dichas técnicas junto con las herramientas y mecanismos empleados en detectar vulnerabilidades.

Para poder ayudar a reducir el riesgo presentado por el desarrollo de aplicaciones web en un entorno dado, los autores de [12] se basan en un análisis de las políticas de seguridad empleadas en otras instituciones similares, así como en el análisis de las regulaciones y el estado del arte de la seguridad en aplicaciones web.

Como se mencionó más arriba, pueden encontrarse en la web muchas otras propuestas sobre el tema, algunas, por ejemplo en [14], [7], [5].

Por nuestra parte, en este trabajo de investigación, nos proponemos como objetivo construir un modelo que, siguiendo el check list de la Web Security Testing Guide (WSTG) de OWASP [11], permita evaluar la seguridad de una aplicación web comprobando su adhesión a los puntos establecidos en la mencionada lista de control.

LÍNEAS DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN

Este trabajo se enmarca dentro de una línea de investigación dentro del SEG (Sofware Engineering Group) de la Universidad Nacional de San Luis; creemos que el mismo reviste un gran interés y potencial de desarrollo. En este sentido, cabe aclarar que se trata de una extensión de una línea de investigación más amplia y consolidada dentro del grupo, que se ocupa de la aplicación y desarrollo de técnicas de Evaluación de Seguridad en aplicaciones de software (ver p.e. [1], [3], [2]).

En este caso, en particular, nos enfocamos concretamente en producir un modelo que sirva para evaluar el cumplimiento de las normas establecidas en el Web Security Testing Guide (WSTG) de OWASP, siguiendo para ello su Check List [11]. A modo de ilustración, en la Tabla 1 se muestra

el primer ítem y sus correspondientes diez sub ítems. La tabla completa consiste de diez ítems en el primer nivel y un total de más de ciento diez ítems.

Para construir el modelo de evaluación, la tabla completa del Check List será parte integral del mismo, permitiendo, a quien decida adoptar el Web Security Testing Guide como parte de su desarrollo de software web, controlar el nivel de cumplimiento de la norma. Para ello, siguiendo los lineamientos del método LSP, cada sub ítem podrá ser evaluado y a su vez agregado bajo una función de agregación que brinda un indicador parcial de satisfacción; los sub ítems podrán ser agregados nuevamente de forma iterativa hasta obtener un indicador global de satisfacción.

Tabla 1. Primer item del check list (Information Gathering) con 10 sub ítems.

Information Gathering	Test Name	Description
OTG-INFO-001	Conduct Search Engine Discovery and Reconnaissance for Information Leakage	Use a search engine to search for Network diagrams and Configurations, Credentials, Error message content.
OTG-INFO-002	Fingerprint Web Server	Find the version and type of a running web server to determine known vulnerabilities and the appropriate exploits. Using "HTTP header field ordering" and "Malformed requests test".
OTG-INFO-003	Review Webserver Metafiles for Information Leakage	Analyze robots.txt and identify <META> Tags from website.
OTG-INFO-004	Enumerate Applications on Webserver	Find applications hosted in the webserver (Virtual hosts/Subdomain), non-standard ports, DNS zone transfers
OTG-INFO-005	Review Webpage Comments and Metadata for Information Leakage	Find sensitive information from webpage comments and Metadata on source code.
OTG-INFO-006	Identify application entry points	Identify from hidden fields, parameters, methods HTTP header analysis
OTG-INFO-007	Map execution paths through application	Map the target application and understand the principal workflows.
OTG-INFO-008	Fingerprint Web Application Framework	Find the type of web application framework/CMS from HTTP headers, Cookies, Source code, Specific files and folders.
OTG-INFO-009	Fingerprint Web Application	Identify the web application and version to determine known vulnerabilities and the appropriate exploits.
OTG-INFO-010	Map Application Architecture	Identify application architecture including Web language, WAF, Reverse proxy, Application Server, Backend Database

RESULTADOS Y OBJETIVOS

El objetivo principal, en líneas generales, que nos hemos propuesto, es poder construir un modelo que permita, a quien adopte la Web Security Testing Guide [11], conocer el grado de cumplimiento de los lineamientos de dicha guía así como decidir sobre qué aspectos de la misma se pondrá mayor o menor énfasis. Esto es posible debido al método de evaluación adoptado que va más

allá de un simple método de evaluación aditivo. En efecto, el método de evaluación elegido para la construcción del modelo de evaluación, el método Logic Score of Preference (LSP), permite que puedan darse no solo mayor o menor peso a distintos ítems, sino que algunos de estos ítems puedan ser considerados opcionales y otros mandatorios, pudiendo cada uno contar con un grado determinado de esa cualidad, permitiendo de esta manera adaptar el modelo a las necesidades particulares de la organización.

FORMACIÓN DE RECURSOS HUMANOS

Dentro del SEG (Software Engineering Group), en el ámbito de la Universidad Nacional de San Luis, en el que se ejecuta el Proyecto de Incentivos código 22/F222 "Ingeniería de Software: Conceptos, Prácticas y Herramientas para el Desarrollo de Software de Calidad", se vienen llevando a cabo numerosas tesis de grado y de posgrado.

En este sentido, creemos que la línea de investigación aquí descripta, la cual es una extensión de una línea más amplia sobre aplicación y desarrollo de Análisis de Seguridad en aplicaciones de Software, seguirá dando sus frutos, tanto en publicaciones nacionales e internacionales como en la formación de recursos humanos (2 tesis de maestría presentadas más una tesis de maestría en ejecución). Asimismo, de momento, se ha encarado la posibilidad de la ejecución de una nueva tesis de maestría basada en los objetivos que aquí nos hemos propuesto.

REFERENCIAS Y BIBLIOGRAFIA

[1] Ana Funes, Aristides Dasso, Germán Montejano, Daniel Riesco. "A SAMM-based model for assessing Cybersecurity Implementations", actas de CoNaIISI 2018, 29 y 30 de Noviembre de 2018, Mar del Plata, Buenos Aires, Argentina.

[2] Aristides Dasso y Ana Funes, "Threat and Risk Assessment Using Continuous Logic", Encyclopedia of Organizational Knowledge, Administration, and Technologies, 1st. edition. IGI Global. Aceptado para su publicación en 2020.

[3] Aristides Dasso, Ana Funes, Germán Montejano, D. Riesco, R. Uzal, Roberto, N. Debnath; "Model Based Evaluation of Cybersecurity Implementations". ITNG 2016. Las Vegas, Nevada, USA, 11-13 abril 2016. In S. Latifi (ed.), Information Technology New Generations, Advances in Intelligent Systems and Computing 448. DOI: 10.1007/978-3-319-32467-8_28. Springer International Publishing, Switzerland 2016.

[4] Bala Musa Shuaibu, Norita Md Norwawi; Mohd Hasan Selamat; Abdulkareem Al-Alwani. "Systematic review of web application security development model". Artificial Intelligence Review February 2015 https://doi.org/10.1007/s10462-012-9375-6

[5] Cody Arsenault- 11 Web Application Security Best Practices. Updated on March 4, 2019. https://www.keycdn.com/blog/web-application-security-best-practices

[6] Divya Rishi Sahu & Deepak Singh Tomar. "Analysis of Web Application Code Vulnerabilities using Secure Coding Standards". Computer Engineering and Computer Science. Arabian Journal for Science and Engineering volume 42, pages 885–895 (2017)

[7] guru99. Web Application Testing: 8 Step Guide to Website Testing. https://www.guru99.com/web-application-testing.html

[8] Jozo Dujmović. "Soft Computing Evaluation Logic. The LSP Decision Method and Its Applications". Wiley, IEEE Press. Hoboken, NJ : John Wiley & Sons, 2018

[9] Jun Zhu, Jing Xie, Heather Richter Lipford, Bill Chu; “Supporting secure programming in web applications through interactive static analysis”. Cairo University. Journal of Advanced Research. 2013 Production and hosting by Elsevier B.V. on behalf of Cairo University. http://dx.doi.org/10.1016/j.jare.2013.11.006

[10] Khairul Anwar Sedek, Norlis Osman, Mohd Nizam Osman, Hj. Kamaruzaman Jusoff, “Developing a Secure Web Application Using OWASP Guidelines”. Vol. 2, No. 4 (2009), Computer and Information Science.

[11] OWASP Web Security Testing Guide (WSTG). https://owasp.org/www-project-web-security-testing-guide/

[12] S Vargas, M Vera and J Rodriguez. “Security strategy for vulnerabilities prevention in the development of web applications”. Published under licence by IOP Publishing Ltd. Journal of Physics: Conference Series, Volume 1414, V International Conference Days of Applied Mathematics 15–17 May 2019, Barranquilla, Colombia.

[13] Sajjad Rafique, Mamoona Humayun, Zartasha Gul, Ansar Abbas, Hasan Javed. “Systematic Review of Web Application Security Vulnerabilities Detection Methods”. Journal of Computer and Communications, 2015, 3, 28-40. Published Online September 2015 in SciRes. http://www.scirp.org/journal/jcc. http://dx.doi.org/10.4236/jcc.2015.39004.

[14] Top 30+ Web Application Testing Tools In 2020 (Comprehensive List). https://www.softwaretestinghelp.com/most-popular-web-application-testing-tools/ Last Updated: March 19, 2020