Evaluación de la Seguridad en Sistemas Informáticos

Aristides Dasso, Ana Funes, Daniel Riesco, Germán Montejano

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
{arisdas, afunes, driesco, gmonte}@unsl.edu.ar

RESUMEN

En este trabajo presentamos los objetivos, lineamientos y resultados de una línea de investigación sobre la creación de modelos de evaluación de seguridad informática en organizaciones. En el ámbito del desarrollo de modelos de evaluación de sistemas complejos, esta investigación tiene como objetivo la creación, puesta a punto y aplicación de modelos que permitan obtener indicadores del nivel de madurez alcanzado en la seguridad de un sistema informático.

En general, el método empleado para el desarrollo de dichos modelos de evaluación es el método LSP. Asimismo, hemos tomado como referencia para la creación de partes de los modelos, estándares reconocidos como son la norma ISO 27000, o recomendaciones dadas por organismos internacionales, como la OWASP, que sirven de guía a las organizaciones para formular e implementar estrategias para la seguridad del software.

Palabras clave: Seguridad de Sistemas Informáticos. Evaluación de la Seguridad de Sistemas Informáticos. Métodos de Evaluación. Logic Scoring of Preference (LSP). SAMM. Software Assurance Maturity Model.

CONTEXTO

Este trabajo de investigación se viene llevando a cabo dentro del SEG (Software Engineering Group), en el ámbito de la Universidad Nacional de San Luis y se encuentra enmarcado dentro de una de las líneas de investigación del Proyecto de Incentivos código 22/F222 “Aspectos de alta sensibilidad en el ejercicio de la profesión de Ingeniero de Software” (Director: Daniel Riesco, Co-Director: Roberto Uzal. Acreditado con evaluación externa. Financiamiento: Universidad Nacional de San Luis).

INTRODUCCIÓN

Los modelos de evaluación de sistemas de seguridad permiten conocer, dentro de una organización, la madurez 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, tales 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 herramientas apropiadas para evaluar el grado de adecuación con dichos estándares.

En este sentido, existen en la literatura

múltiples propuestas. Por un lado, se encuentran diversas publicaciones de modelos de madurez de la seguridad, como por ejemplo en el trabajo e M. F. Saleh [18] o el trabajo de Karokola et al. [15], estos último, por ejemplo, proponen un modelo para servicios de e-gov; por otro lado, el modelo propuesto por Sjelin et al. [24], ayuda en la implementación de programas de seguridad en áreas comunitarias; la propuesta de S. Almuhammadi et al. [1], complementa el NIST Cybersecurity Framework for Critical Infrastructure; o la propuesta de S. Monteiro et al. [19], quienes presentan una metodología para evaluar la madurez de una organización con respecto a su seguridad. En [4] los autores también presentan una propuesta de modelo basado en los estándares dados en la ISO/IEC 27002.

Asimismo, existen múltiples organismos gubernamentales, educativos, así como instituciones internacionales, que certifican seguridad o que proponen estándares o pruebas para distintos aspectos de la seguridad; entre ellos el CERT, Division del Software Engineering Institute (SEI) [3] de la Carnegie Mellon University (CMU), creado en 1988 como respuesta al Morris worm y el National Institute of Standards and Technology (NIST) of USA que ha desarrollado SCAP (Security Content Automation Protocol) [20], el cual combina varios estándares, esquemas, etc. que pueden ser usados para automatizar diversas medidas de seguridad. La IEEE [17], [23] también tiene sus propios estándares e iniciativas en el campo de la seguridad informática, en particular el IEEE Centre for Secure Design (CSD). [16].

Dentro de los organismos internacionales, también se encuentra la Open Web Application Security Project (OWASP), una organización mundial sin fines de lucro, focalizada en mejorar la seguridad del software, la cual cuenta con varios proyectos relacionados con la seguridad informática, entre ellos el proyecto SAMM (Software Assurance Maturity Model) [22]. SAMM es un framework abierto creado para ayudar a las organizaciones en varios aspectos de la seguridad, tales como la evaluación, el desarrollo de programas de seguridad, programas de mejoras, y definición y medición de actividades relacionadas con la seguridad.

LÍNEAS DE INVESTIGACIÓN, DESARROLLO E INNOVACIÓN

La línea de investigación en la que se enmarca el trabajo presentado, es parte de una línea de investigación sobre el tema de la construcción de modelos de evaluación de sistemas complejos y que viene desarrollándose desde hace tiempo en el ámbito del SEG (Software Engineering Group), donde se han obtenido resultados plasmados en diversas publicaciones (ver por ejemplo [12][13][4] [14][2][5][6]).

En esta instancia, nos encontramos abocados a estudiar la propuesta de la OWASP, particularmente en su proyecto SAMM [22], con el objetivo de proponer un nuevo modelo de evaluación de la seguridad, que si bien, se basa en SAMM, apunta a mejorar la forma de evaluación. Más concretamente, como parte de las herramientas de OWASP, se encuentra el SAMM Assessment Toolbox v1.5 [21]. Se trata de un template que sirve de guía al equipo evaluador durante las entrevistas llevadas a cabo dentro de la organización a ser evaluada. Usando este Toolbox, dicho equipo cuenta con una guía sobre qué aspectos debe abordar durante las entrevistas y, además, puede asignar puntajes a cada pregunta. La herramienta, de forma automática, al finalizar la evaluación devuelve un puntaje global que es interpretado como un indicador de la madurez alcanzada dentro de la organización en su seguridad informática. Para esto, la herramienta adopta un simple método de puntaje aditivo. Sin embargo, creemos que la evaluación de la seguridad informática involucra una serie de decisiones del tipo y/o que deben considerarse en su momento.

En este sentido, el método LSP (Logic Score of Preference) [7][8][9][10][11] viene a cubrir esta necesidad ya que se trata de un método que permite la creación de modelos

de evaluación, especialmente útil para sistemas complejos, en los cuales un número importante de decisiones deben ser consideradas.

RESULTADOS Y OBJETIVOS

Actualmente, nos encontramos desarrollando el modelo de evaluación, para lo cual hemos adoptado el método LSP.

En primer lugar, ya hemos establecido un conjunto de requerimientos, bajo la forma de una estructura jerárquica, llamado árbol de requerimientos en LSP. Estos requerimientos provienen directamente de cada nivel del SAMM.

Como se puede ver en la Tabla 1, SAMM se encuentra organizado en cuatro funciones principales del negocio (Governance, Construction, Verification and Operations). Cada función del negocio comprende a su vez tres prácticas de seguridad, y cada una de esas tres prácticas cuenta con tres niveles de seguridad u objetivos. Cada uno de esos objetivos se encuentra definido por métricas y actividades específicas.

Tabla 1. Los tres primeros niveles de SAMM

Business Functions	Security Practices	Objectives
1. Governance	1.1. Strategy & Metrics	1.1.1. SM1: Establish a unified strategic roadmap for software security within the organization.
		1.1.2. SM2: Measure relative value of data and software assets and choose risk tolerance.
		1.1.3. SM3: Align security expenditure with relevant business indicators and asset value.
	1.2. Policy & Compliance	1.2.1. PC1: Understand relevant governance and compliance drivers to the organization
		1.2.2. PC2: Establish security and compliance baseline and understand per-project risks.
		1.2.3. PC3: Require compliance and measure projects against organization-wide policies and standards.
	1.3. Education & Guidance	1.3.1. EG1: Offer development staff access to resources around the topics of secure programming and deployment.
		1.3.2. EG2: Educate all personnel in the software lifecycle with role-specific guidance on secure development.
		1.3.3. EG3: Mandate comprehensive security training and certify personnel for baseline knowledge.
2. Construction	2.1. Threat Assessment	2.1.1. TA1: Identify and understand high-level threats to the organization and individual projects.
		2.1.2. TA2: Increase accuracy of threat assessment and improve granularity of per-project understanding.
		2.1.3. TA3: Concretely align compensating controls to each threat against internal and third-party software.
	2.2. Security Requirements	2.2.1. SR1: Consider security explicitly during the software requirements process.
		2.2.2. SR2: Increase granularity of security requirements derived from business logic and known risks.
		2.2.3. SR3: Mandate security requirements process for all software projects and third-party dependencies.
	2.3. Secure Architecture	2.3.1. SA1: Insert consideration of proactive security guidance into the software design process.
		2.3.2. SA2: Direct the software design process toward known secure services and secure-by-default designs.
		2.3.3. SA3: Formally control the software design process and validate utilization of secure components.
3. Verification	3.1. Design Review	3.1.1. DR1: Support ad-hoc reviews of software design to ensure baseline mitigations for known risks.
		3.1.2. DR2: Offer assessment services to review software design against comprehensive best practices for security.
		3.1.3. DR3: Require assessments and validate artifacts to develop detailed understanding of protection mechanisms.
	3.2. Implementation Review	3.2.1. IR1: Opportunistically find basic code-level vulnerabilities and other high-risk security issues.
		3.2.2. IR2: Make implementation review during development more accurate and efficient through automation.
		3.2.3. IR3: Mandate comprehensive implementation review process to discover language-level and application-specific risks.
	3.3. Security Testing	3.3.1. ST1: Establish process to perform basic security tests based on implementation and software requirements.
		3.3.2. ST2: Make security testing during development more complete and efficient through automation.
		3.3.3. ST3: Require application of specific security testing to ensure baseline security before deployment.
4. Operations	4.1. Issue Management	4.1.1. IM1: Understand high-level plan for responding to issue reports or incidents.
		4.1.2. IM2: Elaborate expectations for response process to improve consistency and communications.
		4.1.3. IM3: Improve analysis and data gathering within response process for feedback into proactive planning.
	4.2. Environment Hardening	4.2.1. EH1: Understand baseline operational environment for applications and software components.
		4.2.2. EH2: Improve confidence in application operations by hardening the operating environment.
		4.2.3. EH3: Validate application health and status of operational environment against known best practices.
	4.3. Operational Enablement	4.3.1. OE1: Enable communications between development teams and operators for critical security-relevant data.
		4.3.2. OE2: Improve expectations for continuous secure operations through provision of detailed procedures.
		4.3.3. OE3: Mandate communication of security information and validate artifacts for completeness.

El cuarto nivel del árbol, que no se muestra en la tabla por razones de espacio, contiene un número de preguntas que son formuladas durante las entrevistas, mientras que en el quinto, se acomodan las hojas del árbol, llamadas variables de perfomance en LSP, que corresponden a las guías de entrevista dados en el SAMM Assessment Toolbox v.1.5.

De acuerdo al método LSP, durante el proceso de evaluación, cada variable de performance debe ser transformada en una preferencia elemental, un valor entre 0 y 100, que representa el grado de satisfacción alcanzado en ese ítem. Para esto, en ocasiones resulta necesario definir algunas funciones, llamadas criterios elementales en LSP. En nuestro caso, dado que cada preferencia elemental es asignada en forma directa por el evaluador no hay necesidad de definir ningún criterio elemental.

La última etapa de la construcción del modelo es la creación y puesta a punto de la estructura de agregación o función de criterio LSP. La misma se encuentra aún en desarrollo ya que el proceso de puesta a punto implica la evaluación de varios sistemas para ver cómo se comporta el modelo y su adaptación para ir ajustándolo.

Dicha estructura de agregación se construye a partir de las hojas del árbol de requerimientos, por medio de un proceso iterativo que comienza por agregar grupos de preferencias elementales, por medio de operadores provisto por el método, y generando un número de preferencias agregadas. Este es un proceso bottom up que se repite sobre las preferencias agregadas hasta que se obtiene una única preferencia global. Este resultado global, que es un valor en el intervalo [0..100], representa el porcentaje de cumplimiento con respecto a los requisitos de seguridad establecidos en el árbol (y en SAMM) y es un indicador, en nuestro, caso del nivel de madurez alcanzado por la organización bajo evaluación.

Como parte del trabajo futuro, esperamos, en una etapa siguiente, finalizar con la calibración de la estructura de agregación y poder aplicar el modelo a la evaluación de casos reales.

FORMACIÓN DE RECURSOS HUMANOS

Dentro del SEG (Software Engineering Group), en el ámbito de la Universidad Nacional de San Luis, en el que se realiza el Proyecto de Incentivos código 22/F222 “Aspectos de alta sensibilidad en el ejercicio de la profesión de Ingeniero de Software”, se han llevado a cabo numerosas tesis de grado y de posgrado.

Entre otros, nos hemos concentrado en la evaluación de sitios de gobierno electrónico lo que ha dado como resultado una tesis de maestría en 2010 [2], mientras que hay otra en etapa de finalización sobre modelos de evaluación de la accesibilidad web [14]. La construcción del modelo aquí expuesto, también, tiene como objetivo ser motivo de tesis, como lo han sido la construcción de otras herramientas en el ámbito del proyecto.

REFERENCIAS

[1] Almuhammadi, Sultan, Alsaleh, Majeed. “INFORMATION SECURITY MATURITY MODEL FOR NIST CYBER SECURITY FRAMEWORK”. Computer Science & Information Technology, Sixth International Conference on Information Technology Convergence and Services (ITCS 2017). Sydney, Australia, February 25~26, 2017.

[2] Castro,M. Dasso, A., Funes, A. “Modelo de Evaluación para Sitios de Gobierno Electrónico”. 38 JAIIO/SIE 2009, Simposio de Informática en el Estado 2009, Mar del Plata, Argentina, August 26-28, 2009.

[3] CERT Division of the Software Engineering Institute (SEI). Carnegie Mellon University (CMU). http://www.cert.org/ (Retrieved March 2015)

[4] Dasso, Aristides, Funes, Ana, Montejano, Germán, Riesco, Daniel, Uzal, Roberto, Debnath, Narayan; “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.

[5] Debnath, N. Dasso, A., Funes, A., Montejano, G., Riesco, D., Uzal, R. “The LSP Method Applied to Human Resources Evaluation and Selection”, Journal of Computer Science and Information Management, Publication of the Association of Management/International Association of Management, Volume 3, Number 2, 2000, ISBN 1525-4372, pp.1-12.

[6] Debnath, N. Dasso, A., Funes, A., Uzal, R., Paganini, J. “E-

government Services Offerings Evaluation Using Continuous Logic". 2007 ACS/IEEE International Conference on Computer Systems and Applications, AICCSA '2007, Amman, Jordan. Sponsored by IEEE Computer Society, Arab Computer Society, and Philadelphia University, Jordan. May 13-16, 2007.

[7] Dujmovic, J. J. and Elnicki, R., "A DMS Cost/Benefit Decision Model: Mathematical Models for Data management System Evaluation, Comparison, and Selection", National Bureau of Standards, Washington, D.C., No. NBS-GCR-82-374, NTIS No. PB82-170150 (155 pages), 1982.

[8] Dujmovic, J. J.: "A Method for Evaluation and Selection of Complex Hardware and Software Systems", The 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG96 Proceedings, vol. 1, pp.368-378, 1996.

[9] Dujmovic, J. J.: "Quantitative Evaluation of Software", Proceedings of the IASTED International Conference on Software Engineering, edited by M.H. Hamza, pp. 3-7, IASTED/Acta Press, 1997.

[10] Dujmovic, J. J.; Bayucan, A.: "Evaluation and Comparison of Windowed environments", Proceedings of the IASTED Interna Conference Software Engineering (SE'97), pp 102-105, 1997.

[11] Dujmovic, Jozo J.: "Continuous Preference Logic for System Evaluation", IEEE Transactions on Fuzzy Systems, Vol. 15, Nº 6, December 2007

[12] Funes, A., Dasso, A., "Web Application Frameworks Evaluation", CONAIISI 2014, 13 y 14 de noviembre de 2014, San Luis, Argentina. pp. 1063-1070. ISSN: 2346-9927.

[13] Funes, A., Dasso,A. Salgado, C., Peralta, M., "UML Tool Evaluation Requirements". Argentine Symposium on Information Systems ASIS 2005. Rosario, Argentina. September 29-30, 2005.

[14] Gallardo, Cecilia, Funes, Ana. "Un Modelo para la Evaluación de la Calidad de la Accesibilidad al Contenido Web", CONAIISI 2015, 19 y 20 de Noviembre de 2015, Buenos Aires, Argentina, ISBN: 978-987-1896-47-9.

[15] Geoffrey Karokola, Stewart Kowalski and Louise Yngström. "Towards An Information Security Maturity Model for Secure e-Government Services: A Stakeholders View". Proceedings of the Fifth International Symposium on Human Aspects of Information, Security & Assurance (HAISA 2011). Editors: Steven Furnell, Nathan Clarke. London, United Kingdom 7-8 July 2011. Publisher: Lulu.com - 2011

[16] IEEE Computer Society Center for Secure Design. http://cybersecurity.ieee.org/center-for-secure-design.html (Retrieved March 2015)

[17] IEEE Cyber Security Initiative. http://cybersecurity.ieee.org/about.html (Retrieved March 2015)

[18] Malik F. Saleh. "Information Security Maturity Model". INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND SECURITY (IJCSS). Edited by DR. NABEEL TAHIR. Volume 5, Issue 3, August 2011. Publishing Date: July / August 2011. ISSN (Online): 1985 -1553

[19] Monteiro, S. "Information Security Maturity Level: A Fast Assessment Methodology". In proceedings of Ambient Intelligence–Software and Applications –8th International Symposium on Ambient Intelligence (ISAmI 2017). 21-23 June, Porto, Portugal. J.F. De Paz, V. Julián, G. Villarrubia, G. Marreiros, P. Novais (Eds.). ISBN 978-3-319-61118-1

[20] National Institute of Standards and Technology (NIST), Security Content Automation Protocol (SCAP) http://scap.nist.gov/index.html (Retrieved March 2015)

[21] OWASP. SAMM_Assessment_Toolbox_v1.5_FINAL.xlsx. Retrieved 28/11/2017 from: https://www.owasp.org/index.php/OWASP_SAMM_Project

[22] OWASP. Software Assurance Maturity Model. A guide to building security into software development. Version 1.5. OWASP The Open Web Application Security Project. Retrieved 28/11/2017 from: https://www.owasp.org/index.php/OWASP_SAMM_Project

[23] Rozenfeld, M.: "IEEE Standards on Cyber Security". THE INSTITUTE. IEEE. Volume 39, Issue1, March 2015.

[24] Sjelin N., White G. "The Community Cyber Security Maturity Model". In: Clark R., Hakim S. (eds) Cyber-Physical Security. Protecting Critical Infrastructure, vol 3. Springer, Cham.