Aproximación para un método de elicitación y especificación de requerimientos de seguridad para el desarrollo de software

Javier Antúnez, Marisa Panizzi

Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales, Universidad de Morón
Cabildo 134 – Morón (CP 1708)
Tel: 5627-2000

Javier.antunez@gmail.com, marisapanizzi@speedy.com.ar

1. Resumen

El propósito de esta línea de investigación consiste en el diseño de un método preliminar para la elicitación y especificación de requerimientos de seguridad de la información.
Las consideraciones de seguridad en los proyectos de desarrollo de software suelen ser incorporadas en etapas tardías del desarrollo, con la dificultad de incorporar medidas de protección en elementos ya diseñados (o incluso ya desarrollados), y los altos costos asociados a esta incorporación.
La idea fuerza de este trabajo es enfocarse en la etapa de elicitación de requerimientos.
Se propone un método de elicitación basado en el método ARM (Accelerated Requirements Method) y alineado con la metodología SQUARE-Lite (Security Quality Requirements Engineering - Lite), que considera elementos del contexto de la aplicación y aprovecha el conocimiento de las partes interesadas en la elicitación y en la priorización de los requerimientos.

2. *Palabras clave*

Seguridad de la información / Requerimientos de seguridad / Elicitación de requerimientos / Especificación de requerimientos / Requerimientos no funcionales (NFR)

3. Contexto

Dadas las características del proyecto, el mismo se encuentra enmarcado en el área sistemas de información-ingeniería de software, más específicamente en la problemática de ingeniería de requerimientos y elicitación y especificación de requerimientos de seguridad de la información. Esta línea de investigación, tuvo su origen en un proyecto de tesis de grado en desarrollo en la Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales de la Universidad de Morón.

4. Introducción

Mucho ha avanzado la tecnología y la ingeniería del software, las redes están cada vez mas interconectadas y es casi impensable desarrollar negocios sin el uso de tecnologías de información y conexión con Internet.

La evolución de los sistemas de información nos ha llevado desde los esquemas centralizados de hace un algunas décadas atrás, con lenguajes estructurados, a los nuevos sistemas distribuidos desarrollados en múltiples capas con lenguajes orientados a objetos (muchas veces, incluso con distintos lenguajes entre capas). El enfoque de proteger un único sistema o equipo de los esquemas centralizados, no es factible en la actualidad, ya que los sistemas desarrollados desde un tiempo a la fecha, funcionan completamente distribuidos, desplegados sobre múltiples plataformas, con diversos lenguajes de programación y muchas veces con servicios provistos por distintas organizaciones

Este nuevo escenario, genera un aumento en la cantidad de puntos que deben ser cubiertos para asegurar que un problema no afecte el normal desempeño de la organización.

Parte de estos problemas son cubiertos a través de distintas soluciones técnicas y herramientas de seguridad bien conocidas y cuentan con altos niveles de efectividad para su función

(Firewalls, IDS/IPS, VPNs, Antivirus, Antispywares, Sistemas de AAA, PKI, entre otros). La mayoría de estas herramientas actúan a nivel de red y se encuentran implementadas en porcentajes importantes de las organizaciones (en promedio del orden de más del 90% de las organizaciones medianas a grandes). **(RICHARDSON, 2012)**

Debido al nivel de evolución y madurez de las herramientas de protección, y su efectividad para la función que proveen, proliferaron nuevos focos de exposición, desplazando el tipo de ataques desde el nivel de red/infraestructura hacia el nivel de aplicación.

El nivel de madurez relativo a la seguridad de las aplicaciones es mucho menor comparada con la seguridad de otros componentes.

Es común encontrarse con fallas graves de seguridad en los desarrollos, esto se atribuye a que:

- Los requisitos de seguridad, en la mayoría de los casos, no son expresados en forma explícita por el cliente, lo que los convierte en requisitos no funcionales. **(Sindre, 2001)**
- La seguridad del software/servicio se suele considerar como algo "adicional", que se diseñará e implementará luego que este funcione.
- Los ingenieros de requerimientos no poseen formación adecuada, ni experiencia, como para identificar requisitos funcionales y no funcionales relacionados con la seguridad, especificarlos e incorporarlos en etapas tempranas del proceso de desarrollo.
- Existen malas prácticas de desarrollo llevadas adelante por algunos desarrolladores, muchas veces relacionadas con el bajo índice de madurez de la organización (que no incorpora un marco adecuado de control de los procesos).

Bajo estas condiciones, es difícil poder realizar un aseguramiento adecuado de las condiciones de calidad y seguridad en un entorno de sistemas dado.

Hay estudios que estiman, que el 40% de los errores se detectan en la fase de integración y el 25% en la etapa de beta y post-implementación. **(NIST, 2002).**

Según información del NIST **(NIST, 2002)** resolver un error detectado en fases tardías del proyecto, puede costar entre 10 veces (etapa de integración) y 30 veces (etapa de post implementación) más que si es detectado y resuelto en las fases tempranas.

En este sentido los desarrollos que dejan la seguridad "para después" y no incorporan los requisitos desde la especificación de requerimientos, se ven afectados fuertemente en lo que respecta a tiempos, costos y retrabajos.

Todo esto sin contar las consecuencias y costos de posibles incidentes de seguridad (según estudios recientes los incidentes de seguridad analizados tienen un costo promedio entre USD 100.000 y 300.000, pero se han dado casos de compromiso de la totalidad de la información con costos directos de millones de dólares e indirectos no mensurables -daño en reputación, pérdida de valor de las acciones de las compañías involucradas en el incidente, demandas legales, entre otros-). **(Ponemon Institute LLC, 2012)**

Para tratar la problemática de la seguridad en fases tempranas del proceso de desarrollo de software, hacia el inicio del proyecto, la etapa de elicitación de requerimientos resulta un momento oportuno, ya que sus resultados establecen en gran medida lo que se desarrollará en etapas posteriores. Es decir, los requerimientos constituyen la "materia prima" utilizada para muchas de las decisiones y actividades de fases posteriores del ciclo de vida.

La problemática de la seguridad en las aplicaciones, se viene tratando en distintos papers.

En el análisis de los antecedentes bibliográficos, hemos identificado que existen marcos metodológicos como SQUARE **(Mead, Houg, & Stehney, Security Quality Requirements Engineering (SQUARE) Methodology, 2005)**, SQUARE-Lite **(Gayash, Viswanathan, & Mead, 2008)**, SREP **(Mellado, Fernandez-Medina, & Piattini, 2006)** y STRIDE **(Hernan, Lambert, & Shostack, 2006)** para la elicitación, especificación y análisis de requerimientos de seguridad (y en algunos casos análisis de riesgos).

También trabajos que usan como base el modelado basado en árboles de ataque **(Schneier, 1999)**, el comportamiento de un potencial atacante **(Lamsweerde, Brohez, De Landsheer, & Janssens, 2003)**; el trabajo con metas y catálogos **(Saeki & Kaiya, 2009)** y restricciones de seguridad **(Mouratidis & Giorgini, 2007)**.

Existe también literatura donde se extienden los casos de uso de UML para analizar aspectos de seguridad **(Sindre, 2001)**, **(Firesmith, 2003)**; **(Kabasele Tenday, 2010)**.

Finalmente se revisaron los estándares que poseen apartados de relevancia, como ser aquellos aspectos relativos a la especificación de requerimientos de software (SRS) en los estándares IEEE std 830:1998, ISO/IEC/IEEE 15288:2008, ISO/IEC/IEEE 12207 y el estándar ISO/IEC/IEEE 29148:2011.

Para la construcción de la solución se propone seguir el esquema conceptual de la **Figura Nro. 1.**

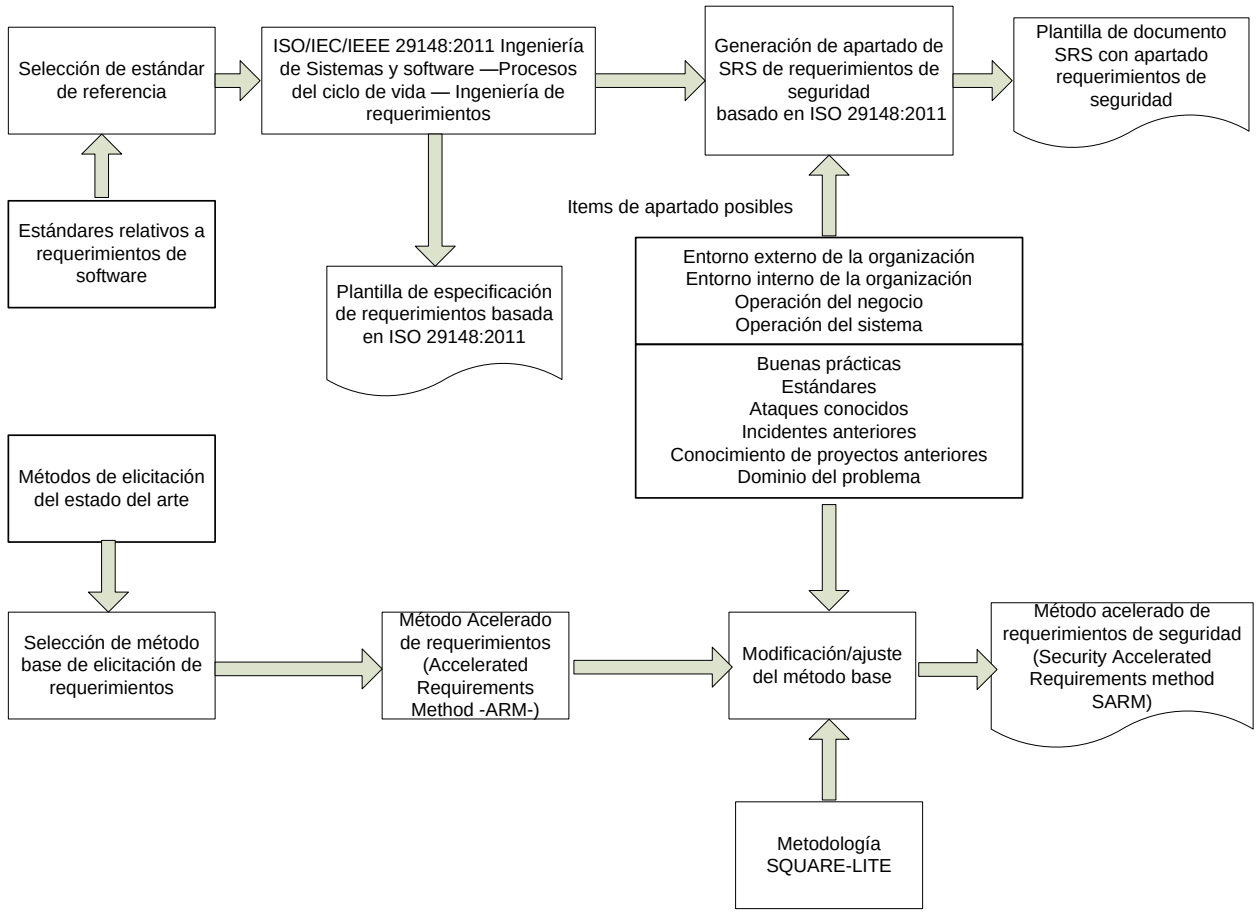

Figura Nro. 1 - Esquema conceptual de construcción de la solución

Trabajando en 2 grandes líneas. La primera busca sentar las bases para la especificación, tomando como base un estándar reconocido.

Entre los estándares disponibles relativos a ingeniería o especificación de requerimientos, se seleccionó el estándar ISO/IEC/IEEE 29148:2011, y se intenta crear un nuevo apartado para el documento de especificación de requerimientos de software (SRS), para la especificación de requerimientos de seguridad (ver **Figura Nro. 2**) y en conjunto con esto, una plantilla para la especificación individual de requerimientos que cumpla los requisitos especificados por el mismo estándar.

La segunda línea de actividades, constará de la selección de un método de elicitación existente como base, para construir un método modificado adaptado a la elicitación de requerimientos de seguridad. Preliminarmente se utilizará como base el método Accelerated Requirements Method (ARM), que demostró gran flexibilidad y (pese a ciertas limitaciones identificadas, principalmente con el tipo de requerimientos obtenidos) buenos resultados en las pruebas realizadas por el equipo del Software Engineering Institute de la Universidad de Carnegie Mellon **(Mead, Requirements Elicitation Case Studies Using IBIS, JAD, and ARM, 2006-2008)**.

Se intentará a través de las modificaciones, resolver las limitaciones del método original identificadas por investigaciones previas, principalmente que el resultado de la elicitación eran requerimientos funcionales.

Para conseguir este resultado, se introducirán nuevos pasos y modificaran algunos pasos existentes. Por ejemplo parece oportuno introducir un o varios pasos para realizar la identificación de metas y objetivos de seguridad, asegurarse que las partes interesadas que pueden realizar aportes significativos desde el punto de vista de los requerimientos de seguridad (como por ejemplo los sectores de Seguridad de la Información, Legales, Cumplimiento, Auditoría, etc.).

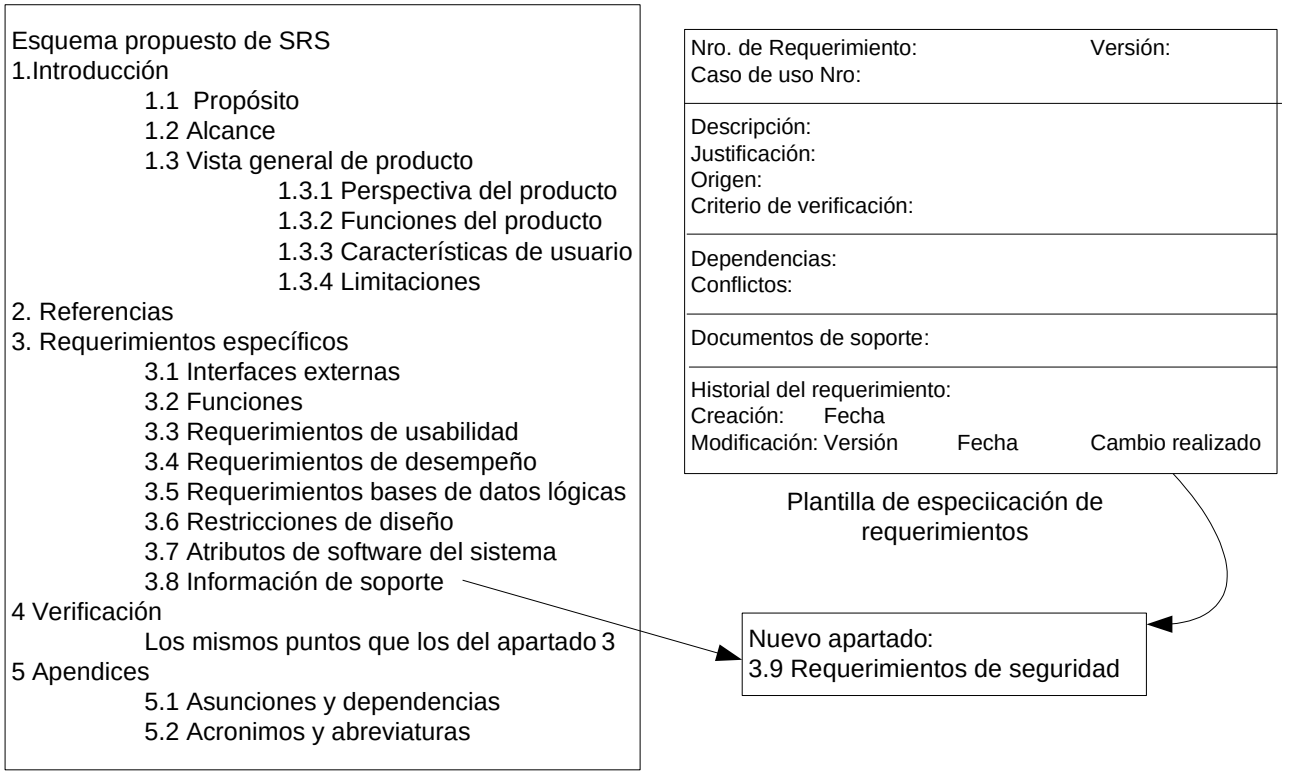

Figura Nro. 2 - Esquema general propuesto para el documento SRS y plantilla de requerimientos

Finalmente podría modificarse la etapa de cierre para que se requiera que los entregables (documento de Especificación de Requerimientos de Software – SRS), cumplan con la plantilla de requerimientos y del apartado de requerimientos de seguridad.

5. Resultados esperados

Se pretende probar el método en un caso real, sobre un proyecto ya finalizado, para analizar los resultados obtenidos originalmente y compararlos con los resultados de la aplicación del método propuesto.
Se espera como resultado, identificar requerimientos que no fueron obtenidos mediante el proceso de análisis de requerimientos tradicional (aplicado en originalmente en el proyecto testigo) e idealmente que todos los requerimientos obtenidos originalmente surjan como resultado de aplicar el método propuesto.

6. Bibliografía

Firesmith, D. G. (2003). Security use cases. Journal of Object Technology , 53-64.

Hernan, S., Lambert, S. O., & Shostack, A. (2006, November 1). Uncover Security Design Flaws Using The STRIDE Approach. Retrieved Octubre 06, 2012, from MSDN Magazine: http://msdn.microsoft.com/hi-in/magazine/cc163519%28en-us%29.aspx

Kabasele Tenday, J.-M. (2010). Using Special use case for security in the software development life cycle. Information Security Applications: 11th International Workshop WISA 2010 (pp. 122-134.). Springer.

Lamsweerde, A. v., Brohez, S., De Landsheer, R., & Janssens, D. (2003). From System Goals to Intruder Anti-Goals:Attack Generation and Resolution for Security Requirements Engineering. 11th IEEE International Requirements Engineering Conference RE03 - 2nd Workshop on Requirements for High Assurance Systems REHAS 03 (pp. 49-56). Monterrey Bay, CA, USA: Carnegie Mellon - Software Engineering Institute.

Mead, N. R. (2006-2008, 09 22). Requirements Elicitation Case Studies Using IBIS, JAD, and ARM. Retrieved 10 14, 2012, from Build Security In:https://buildsecurityin.uscert.gov/bsi/articles/best-practices/requirements/532-BSI.html

Mead, N. R., Houg, E. D., & Stehney, T. R. (2005). Security Quality Requirements Engineering (SQUARE) Methodology. Pittsburgh, PA: Carnegie Mellon University - Software Engineering Institute (CMU-SEI).

Mellado, D., Fernandez-Medina, E., & Piattini, M. (2006). A Common Criteria Based Security Requirements Engineering Process for the Development of Secure Information Systems. In Varios, Computer Standard and Interfaces (pp. 244-253). Madrid y Ciudad Real - España: Elsevier.

Mouratidis, H., & Giorgini, P. (2007). SECURE TROPOS: A SECURITY-ORIENTED EXTENSION OF THE TROPOS METHODOLOGY. World Scientific Publishing , 17 (2) 285-309 .

NIST. (2002). NIST Report "The Economic impacts of inadequate infrastructure for software testing". NIST.

Ponemon Institute LLC. (2012). The impact of cybercrime on Business. Ponemo Institute LLC sponsored by Checkpoint Software Technologies.

RICHARDSON, R. (2012). Computer Security Institute/Federal Bureau of Investigations. COMPUTER CRIME AND SECURITY SURVEY 2010-2011. CSI-FBI.

Saeki, M., & Kaiya, H. (2009). Using Common Criteria as Reusable Knowledge in Security Requirements Elicitation. Tokio, Japón: Dept. of Computer Science, Tokyo Institute of Technology.

Schneier, B. (1999). Modeling Security Threaths. Dr. Dobb´s Journal , December ISSUE.

Sindre, G. O. (2001). Capturing security requirements trough misuse cases. Retrieved 05 01, 2006, from folk.uio.no: http://folk.uio.no/nik/2001/21-sindre.pdf