Alineación de los Requerimientos del Software de los Sistemas de Información a las Políticas de Seguridad de una Organización

Pablo Andrés Pessolani
Grupo de Investigación en Seguridad de la Tecnologías de Información y Comunicaciones
Facultad Regional Santa Fe - Universidad Tecnológica Nacional – Argentina
ppessolani@frsf.utn.edu.ar

Resumen. Una política de seguridad es un componente crítico de una arquitectura de seguridad y la plataforma necesaria en base a la cual debe desarrollarse un programa de seguridad [1]. Este programa debe abarcar a los procesos de desarrollo o adquisición de software para los Sistemas de Información, y sus requerimientos de seguridad deben alinearse con las políticas de seguridad establecidas. Aún siendo evidente esta necesidad, es escaso el progreso en el desarrollo de frameworks, metodologías y herramientas que permitan especificar las políticas de forma integral, desde el nivel más general hasta el nivel de detalle necesario para su incorporación como requerimientos en el Ciclo de Vida del Desarrollo de Software (SDLC en inglés), o como especificaciones para la adquisición de software de aplicación.

Keywords: SDLC, Políticas de Seguridad, Requerimientos.

1 Introducción

El software de aplicación, así como ocurre con otras maquinarias de ingeniería, requiere de especificaciones que describan sus objetivos, requerimientos, entradas, salidas, etc., sea para su diseño y desarrollo o para su adquisición. Una de esas especificaciones refiere a la forma en que el mismo debe desempeñarse frente a acciones que intenten reducir sus propiedades de confidencialidad, integridad y/o disponibilidad del software mismo, de la información que produce, o de los datos sobre los que opera, es decir sobre su seguridad.

Una organización establece sus objetivos a partir de su misión y de su visión. Las Tecnologías de Información y Comunicación (TICs) son herramientas necesarias en las que se sustentan las organizaciones para cumplir con esos objetivos. Las TICs, tal como otros activos, pueden ser sometidos a acciones maliciosas que pretendan reducir o impedir el normal desempeño de las mismas, y por lo tanto afectan la capacidad de la organización para que cumpla con sus objetivos establecidos, poniendo en riesgo el cumplimiento de su misión. A diferencia de otros activos, la seguridad de las TICs puede ser amenazada en forma remota, sin dejar ni rastros, ni huellas, en total anonimato y sin legislación o jurisprudencia específica sobre este tipo de delitos. Para la protección de éstos activos se requiere de la consideración y concientización de todos los niveles que componen la organización.

Las TICs están compuestas de una multiplicidad de componentes desarrollados en forma independiente, tales como el hardware, el software de base, los protocolos de

comunicaciones, el software de aplicación, las librerías de funciones o componentes, las bases de datos, etc. Esta lista no está completa sino que es solo enumerativa a modo de ejemplo. Solo si una organización construye o verifica la constitución de cada uno de los componentes de sus sistemas de información de extremo a extremo puede dar certeza (si acaso esto fuese posible) de que se cumplan todos los requerimientos de seguridad. Las organizaciones que más se acercan al cumplimiento de este requisito son las militares. Considérese qué sucedería con un software de cifrado de datos utilizado por las Fuerzas Armadas del País-A que utilice una librería desarrollada por el País-B la cual admite una clave oculta secreta de descifrado solo conocida y utilizable por el País-B.

En lo que a seguridad respecta, aplica adecuadamente el conocido refrán que dice: "*una cadena es tan fuerte como el más débil de sus eslabones*". Es decir, solo basta que un componente de un sistema no se encuentre adecuadamente protegido o un procedimiento establecido no se cumpla estrictamente para que éste se convierta en el principal objetivo de los ataques, poniendo en riesgo al sistema en general.

Para la mayoría de las organizaciones no es práctico, o es económicamente irrealizable, o directamente imposible verificar la seguridad de cada componente que utilizan sus aplicaciones o construirlos por sí mismas; por lo tanto deben confiar en la seguridad de esos recursos, componentes o tecnologías desarrollados o adquiridos por fuera de la organización que son utilizados para el desarrollo y construcción de sus sistemas de información. Esto permite realizar una afirmación práctica: "*La seguridad se basa en la confianza*"; por lo que para establecer la seguridad de un sistema se deben hacer hipótesis de confianza basadas en evidencias (como por ejemplo certificaciones), conocimiento experto, ingeniería inversa, etc.

La norma ISO-27002[2] en su ítem 12.1.1 establece que "*Si se considera apropiado, por ejemplo por razones de costos, la gerencia puede desear hacer uso de productos independientemente evaluados y certificados*". Cuando una organización adquiere un software certificado por alguna entidad de prestigio, confía en que los tests realizados sobre el mismo fueron correctamente efectuados por expertos y que brinda protección adecuada contra la amenazas consideradas en los tests. La certificación del software no es garantía de inmunidad, solo la confianza en el prestigio de la entidad certificante hace que se pueda inferir que se reducen las posibilidades de incidentes respecto a las amenazas contempladas por la batería de tests realizados.

El presente trabajo se desarrolló en el ámbito del Grupo de Seguridad en las Tecnologías de Información y Comunicaciones (GISTIC) de la Facultad Regional Santa Fe de la Universidad Tecnológica Nacional, en el marco del proyecto "*Análisis de la Seguridad como atributo organizacional transversal en los proyectos de Sistemas y Tecnologías de Información*" con código 25/O108.

Este artículo se organiza de la siguiente forma: en la Sección 2 se analiza la relación entre las Políticas de Seguridad y el Software de las Aplicaciones. En la Sección 3 se presenta el Ciclo de Vida de las Políticas de Seguridad. En la Sección 4 se describen los Principios de Seguridad en el desarrollo de aplicaciones. Una serie de políticas de seguridad que tienen impacto en las aplicaciones se presentan en la Sección 5. En la Seccion 6 se exponen algunos modelos, metodologías, frameworks y herramientas que consideran la seguridad en el SDLC, para finalizar en la Sección 7 con las conclusiones.

2 Políticas de Seguridad y el Software de las Aplicaciones

El glosario del TCSEC define a una Política de Seguridad como "*El conjunto de leyes, reglas y prácticas que regulan la forma en que una organización gestiona, proteje y distribuye información sensible*". Generalmente una Política General de Seguridad se establece en el más alto nivel de dirección en congruencia con la misión y los objetivos de una organización. Las personas que integran la dirección de una corporación, de una gran empresa, de una PyME, seguramente desconocen los aspectos de detalle de la seguridad de las TICs. En las grandes corporaciones el CEO (Chief Executive Officer - Gerente General) delega en el CIO (Chief Information Officer – Gerente de Sistemas) estas responsabilidades, quien a su vez delega en el CISO (Chief Information Security Officer- Gerente de Seguridad de TICs) la gestión de la seguridad de las TICs.

Las TICs son tan amplias y variadas que es prácticamente imposible que una persona domine todas esas tecnologías a grado tal que le permitan conocer las amenazas, sus vulnerabilidades, como reducirlas, que tipos de ataques pueden realizarse sobre las mismas y que contramedidas existen. Este nivel de detalle, profundidad y conocimiento requiere de un equipo de profesionales interdisciplinario para establecer las Políticas de Seguridad específicas o de detalle. La norma ISO-27002 en su ítem 6 sugiere establecer una estructura de gestión para la seguridad de la información.

Como consecuencia de la diversidad de requerimientos algunas de las políticas de seguridad tienen un flujo descendente desde los niveles más altos de la organización (nivel de Dirección) hacia los más bajos (Top-Down) [3]; otras se generan a nivel medio tanto en el area de TICs como en otras áreas operativas, y muchas se generan en los niveles más bajos (Personal Técnico) las cuales deben ascender para que la dirección de la organización las respalde y formalice (Bottom-Up) [3].

Además de las políticas establecidas por la propia organización, la legislación vigente, la regulación en ciertas actividades, los acuerdos empresariales, etc., fuerzan a establecer políticas adicionales. Un ejemplo de estos son las certificaciones exigidas por la industria de las tarjetas de crédito a través del Payment Card Industry Data Security Standard (PCI-DSS), el cual requiere de restricciones de acceso a la información personal de los poseedores de tarjetas basado en el principio de "*lo que se necesita saber*".

También son orígenes de nuevas políticas de seguridad las nuevas amenazas, los nuevos tipos de ataques, las vulnerabilidades detectadas, las nuevas tecnologías de acceso a las redes (Bluetooth, WI-FI, GPRS, WAP), etc.

Esta diversidad de orígenes de las políticas de seguridad, la diversidad de amenazas, la multiplicidad de tecnologías con sus particularidades, dificulta la sistematización integral de la generación de requerimientos de seguridad para el desarrollo o adquisición de software a partir de las políticas.

La falta de trazabilidad entre los requerimientos de seguridad de la organización y el entorno operativo de las TICs significa muchas veces que los niveles directivos no participan de la toma de decisiones respecto a las medidas de seguridad requeridas. Inicialmente, es necesario capturar, modelar y analizar las políticas de seguridad que sean consistentes con los objetivos de la organización para que sean traducidas en

requerimientos de operaciones de las TICS, en la gestión de la seguridad y en el desarrollo/selección del software de aplicación.

Un desafío clave es: "*Cómo aplicar consistentemente en toda la organización este creciente número de requerimientos?*". Es dificultoso traducir la política organizacional en políticas técnicas adecuadas para su uso en las TICs. La política de seguridad no es una excepción a esta regla. Muchas de las políticas y procedimientos son específicos de las distintas plataformas con capacidades y limitaciones propias de las tecnologías.

En los entornos actuales multiplataforma, cada plataforma debe ser configurada para adherir a la misma política de seguridad, cada una de ellas tiene su propio modelo de seguridad y su propia sintaxis que requiere un alto nivel de conocimiento técnico para realizar configuraciones correctas. Lograr la consistencia general entre los requerimientos de alto nivel y entre las plataformas entre sí es muy complejo y dificultoso [4].

En las organizaciones existe una gran diversidad de aplicaciones que provienen de la adquisición de software, desarrollos propios y uso de aplicaciones tercerizadas. Cada una de ellas fue desarrollada con su propia metodología que contempla (o no) los requerimientos de seguridad conformando diversos *Dominios de Seguridad*. Esta situación no es necesariamente negativa, pero reduce la posibilidad de gestionar las políticas de seguridad adecuadamente.

Si se realiza una gestión manual no sistematizada de todos estos requerimientos, configuraciones, etc. sobre plataformas heterogéneas seguramente se incrementarán las probabilidades de cometer errores y se dificultarán los posteriores cambios perdiendo la adherencia o alineación con respecto a las políticas. Las demandas crecientes para responder a los cambios continuos de los objetivos de las organizaciones impactan directamente sobre los costos del desarrollo/adquisición de software de aplicaciones y el mantenimiento de las mismas. Estos costos se incrementan cuando, como consecuencia de los cambios de objetivos, cambian los requerimientos de seguridad.

La complejidad en la gestión de la seguridad impone la necesidad de disponer de un marco de referencia flexible para alinearla con los objetivos de la organización e implementar seguridad consistentemente en todos los niveles, datos, aplicaciones y plataformas. Las soluciones puntuales no son escalables y no pueden capturar ni expresar toda la amplitud de las políticas asegurando su consistencia y cumplimiento. El objetivo a alcanzar es la integración de las aplicaciones para que se simplifique el desarrollo y la alineación de los requerimientos de seguridad, como así tambien simplificar la gestión manteniendo un modelo consistente que incluya a todas las aplicaciones. La solución de Gestión de Seguridad a estos problemas complejos debe ser de extremo a extremo.

3 Ciclo de Vida de las Políticas de Seguridad

Las Políticas de Seguridad no son estáticas, requieren de revisiones periódicas generando un ciclo de vida que incluye 5 pasos básicos [5]:

1. *Evaluación de Riesgo:* Brinda el contexto de la organización necesario para desarrollar las políticas de seguridad. De ésta evaluación surge un análisis que

permite detectar aquellas ***amenazas*** mas riesgosas, aquellos ***activos*** mas expuestos y ***vulnerables*** que serán el fundamento para el desarrollo de las Políticas de Seguridad.

2. *Desarrollo de las Políticas de Seguridad*: Este es el desarrollo de las distintas capas o niveles de las políticas de seguridad (general, particular, procedimientos, guías, etc.). Las mismas deben ser informadas a la organización conforme se lo requiera.
3. *Implementar Políticas de Seguridad*: Se ponen en marcha las políticas usándolas en las operaciones normales de la organización.
4. *Gestión de Políticas de Seguridad*: Se evalúa la efectividad y actualidad de las distintas políticas a partir del uso de métricas y contrastándola con los objetivos propuestos.
5. *Auditoría de Políticas de Seguridad*: Se mide el grado de adhesión a las políticas y se identifican defectos luego del cual se retorna al paso de la reevaluación del riesgo.

   Este ciclo de vida se aplica a todos los niveles de políticas, desde la Política General hasta los niveles de procedimientos.

4 Principios de Seguridad en el Desarrollo de Aplicaciones

Además de las Políticas de Seguridad, quienes diseñan, desarrollan o deciden la adquisición de software de aplicación deben tener en consideración ciertos Principios de Seguridad [6] aceptados por la comunidad. Estos principios son un conjunto de propiedades, comportamientos y prácticas de diseño e implementación que intentan reducir la probabilidad de un incidente producto de una amenaza.

- *Defensa en Profundidad*: Si un ataque causa que un mecanismo de seguridad falle, otro mecanismo de diferente naturaleza mantendrá protegido al sistema.
- *Modelo de Seguridad Positivo*: Es aquel en el que se debe explicitar lo que está permitido, por defecto se niega el resto.
- *Completa Intermediación*: Cada pedido de acceso debe ser contrastado contra la autorización.
- *Fallos Seguros*: Si el sistema tiene un fallo, es mejor detenerlo en un estado seguro para impedir una probable ejecución riesgosa o un estado mas vulnerable. Ante un fallo o defecto es mejor impedir el acceso a alguien que no dispone de privilegios aún cuando se impida a alguien que si los tiene. Este principio se contrapone con el componente ***disponibilidad*** de la seguridad.
- *Ejecución con Mínimos Privilegios*: Los usuarios deben ejecutar las aplicaciones con los privilegios necesarios para llevar a cabo sus tareas. La implementación de este principio tiende a limitar el eventual daño que puede resultar de un error de sistema o de una acción maliciosa.
- *Evitar "Seguridad por Oscuridad"*: Refiere a la confianza en la seguridad de un sistema o componente producto de mantener secreta u oculta su implementación. Esto no significa que mantener secretos no sea bueno sino que el diseño debería utilizar principios de diseño conocidos y abiertos. Si más observadores analizan el diseño, se incrementan las probabilidades de detector errores.

- *Separación de Privilegios*: Los mecanismos de protección deberían depender de la satisfacción de múltiples condiciones, tal como requerir la cooperación de al menos dos entidades independientes (con distintos privilegios) para llevar a cabo una transacción.
- *Economía de Mecanismos*: El diseño debe ser pequeño y simple de tal forma que pueda ser evaluado y corroborar que es correcto.
- *Detección de Intrusos*: La detección de Intrusos requiere de la capacidad de registrar eventos relevantes, de disponer de procedimientos que monitoreen los eventos registrados y procedimientos que tomen acciones si se detectan intrusiones. No se debe confiar en otras tecnologías externas a la aplicación, ésta es la única que conoce los valores válidos de parámetros, datos de entrada, acciones permitidas, y que puede detectar intentos de evasión de controles.
- *No Confiar en la Infraestructura*: No se debe asumir que determinados componentes de la infraestructura (hardware, sistema operativo, middleware, librerías, componentes, etc.) evitarán todos los tipos de ataques contra la aplicación.
- *No Confiar en los Servicios Externos*: No se debe caer en el error de asumir que los sistemas externos, por ejemplo de socios de negocios, son de confiar. Se debe tomar todas las medidas de seguridad necesarias en el intercambio de información, datos, componentes, privilegios, etc. asumiendo que de allí podría partir un ataque.
- *Establecer Configuraciones por Defecto Seguras*: Es común que al iniciar la operación sobre un nuevo sistema se relajen las restricciones de seguridad a fin de facilitar su uso. Es en estos momentos donde el sistema presenta mayor vulnerabilidad, por lo que por el contrario, deben establecerse configuraciones seguras que, eventualmente después de analizadas, podrán relajarse.
- *Aceptabilidad*: Este principio suele ir en dirección opuesta respecto a los otros. En general, la defensa en profundidad, el principio de mínimos privilegios y otros principios restringen y quizás dificultan el uso de una aplicación haciendo que le usuario haga caso omiso a ciertas prácticas de seguridad (ejemplo: copiar la contraseña o utilizar el perfil y contraseña de otra persona) poniendo mayor riesgo al sistema. Para evitar esto, la complejidad del sistema de protección debe ser transparente al usuario.

Considerando cada uno de éstos principios, se pueden derivar requerimientos de seguridad, identificar posibles debilidades en los sistemas y tomar decisiones en cuanto a arquitectura y metodologías de implementación.

5 Políticas de Seguridad con Impacto en las Aplicaciones

La seguridad de las aplicaciones son el resultado de las decisiones que se han tomado en la organización [7]. Según la norma ISO 27002 [2] en su control A.12.1.1 requiere que "*Los requisitos de seguridad deberían ser identificados y consensuados antes de desarrollar los sistemas de información*".

Dado que cada organización es diferente a otras, no pueden establecerse condiciones rígidas impuestas por reglas o leyes que puedan abarcarlas a todas; pero si pueden proponerse estándares mínimos a cumplir tales como:

- Compromiso por parte de los niveles de decisión acerca de lograr altos niveles de seguridad.
- Una Política de Seguridad explícita derivada de un análisis de riesgo, de estándares nacionales e internacionales y de normativas vigentes.
- Un Sistema de Gestión de la Seguridad que imponga una metodología para desarrollar y verificar las actividades relacionadas con la seguridad.
- Una Metodología de Desarrollo y un conjunto de convenciones o estándares para la codificación, control, pruebas, versionamiento y documentación.

Probablemente todo el conjunto de Políticas de Seguridad tengan algún tipo de impacto en alguna de las instancias que atraviesa un software de aplicación:

- *Análisis*: En en el obtención, análisis y especificación de los requerimientos ,de casos de uso, casos de abuso, amenazas, etc.
- Diseño: En el proceso de selección de la metodología, definición de entidades, roles, planes de tests de seguridad.
- *Programación*: En la selección del framework de desarrollo de aplicaciones, del lenguaje de programación, de la infraestructura requerida, del acceso a datos de pruebas, de la selección de programadores, de la capacitación, proceso de revisión de código, etc.
- *Pruebas y Benchmarks*: Se requiere contemplar dentro del alcance de las políticas de seguridad a los entornos de pruebas tales como maquetas e implementaciones piloto [2].
- *Implementación en Producción*: En la infraestructura de red, de seguridad, de control de acceso, configuración cuidadosa y personalizada; aspectos necesarios para la operación segura de la misma.
- *Operación*: En los procesos de respaldos y restauraciones, planes de contingencia, manejo de incidentes, etc.
- *Mantenimiento y Actualización*: Para evaluar modificaciones producto de vulnerabilidades detectadas, cambios en los procesos organizacionales, cambios en las normas, en las políticas o como consecuencia de nuevas exigencias del mercado o de la sociedad.
- *Selección del Software*: En el caso de adquisición de software de aplicación, se debe realizar una prueba formal [2]. Los contratos con el proveedor deben indicar los requisitos de seguridad, la evaluación de las certificaciones, etc. La tendencia actual de brindar Software como Servicio (SAAS) hace que se deban realizar otro otro tipo de consideraciones referentes al proveedor, donde no solo se debe contemplarr la seguridad del software de aplicación sino también de todo el entorno operativo de la misma.

Particularmente se consideran Políticas de Seguridad de impacto directo aquellas relacionadas con los siguientes aspectos:

1. Control de acceso a los sistemas de cómputos, redes y aplicaciones consistente con los roles y responsabilidades.
2. Control de acceso a la información, de tal modo que sea consistente con su clasificación y con las políticas de privacidad.
3. Control del flujo de la información, de tal modo que sea consistente con su clasificación y con las políticas de privacidad.
4. Gestión de la disponibilidad, integridad y confiabilidad de los componentes.

5. Protección contra ataques maliciosos.
6. Sistema de identificación confiable para permitir el seguimiento o trazabilidad del acceso a los sistemas, datos e información.

Las vulnerabilidades son inherente al software, introducidas en el proceso de diseño arquitectónico, en su código o en los algoritmos que implementan [7], por lo tanto la solución para detectarlas, evitarlas, mitigarlas o eliminarlas debe ser abordada desde la Ingeniería de Software.

6 Modelos, Metodologías, Frameworks y Herramientas

La tarea de desarrollar software para aplicaciones que apliquen los principios de seguridad en forma consistente y efectiva tiene varios desafíos, entre ellos, la complejidad de integrar funciones específicas de seguridad dentro de los componentes, la dificultad que representa desarrollar un conjunto comprensible de requerimientos básicos de seguridad y la falta de metodologías estándares de diseño seguro [5].

Con la formalización de un criterio de evaluación de seguridad en el estándar Common Criteria (CC) se ha dado un paso importante en la forma de abordar el problema del desarrollo de arquitecturas de IT seguras.

Common Criteria [8] brinda una taxonomía para evaluar las funcionalidades de seguridad a través de un conjunto de ***requerimientos funcionales*** y de seguridad divididos en 11 clases:

1. Auditoría de Seguridad.
2. Comunicación.
3. Soporte Criptográfico.
4. Protección de datos de usuario.
5. Identificación y Autenticación
6. Gestión de funciones de seguridad
7. Privacidad
8. Protección de funciones de seguridad
9. Utilización de Recursos
10. Acceso a componentes
11. Canales confiables

Resulta sorprendente no encontrar ninguna referencia a las consideraciones de seguridad o protección en un libro clásico de UML como es [9]. Según [10] la seguridad y privacidad forman parte de los ***requerimientos no funcionales***. Según [11], las Políticas de Seguridad establecen restricciones sobre las acciones requeridas y permitidas sobre los activos de una organización. Desde el punto de vista del diseño de Software son ***restricciones funcionales*** y expresan los objetivos de seguridad del sistema en términos operacionales. En [12] se propone la consideración de ***anti-requerimientos***, que son los requerimientos de un usuario malicioso que pretende subvertir o trastornar un requerimiento existente. En [13] se plantea un framework en donde los requerimientos de seguridad generan tanto ***requerimientos no funcionales*** como ***restricciones funcionales***.

BSIMM [14] es modelo que permite medir como se ajusta un software a los requerimientos de seguridad (modelo de madurez) en donde se proponen un

framework de seguridad de software (SSF) consistente en 12 prácticas divididas en cuatro dominios

- *Gobernabilidad*: Son prácticas que ayudan a organizar, gestionar y medir las iniciativas de software seguro.
- *Inteligencia*: Son prácticas para llevar a cabo actividades tendientes al software seguro atravesando todas los sectores de una organización. Esto incluye orientación proactiva hacia la seguridad y modelo de amenazas.
- *Puntos de Contacto SDL*: Son prácticas asociadas con el análisis y aseguramiento de los procesos y artefactos del desarrollo de un software en particular.
- *Despliegue*: Son prácticas en donde interactúan los sectores deseguridad de redes tradicional y los sectores de mantenimiento de software. Aquí se consideran las configuraciones, mantenimiento y entorno necesarios con impacto directo sobre la seguridad del software.

No son muchas las herramientas disponibles actualmente que permiten gestionar de forma centralizada las políticas de seguridad de aplicaciones para mejorar el cumplimiento y así lograr una mejor gobernabilidad. Una de ellas es *Tivoli Security Policy Manager (TSPM)* [4], de IBM. TSPM permite a los administradores y responsables de las aplicaciones gestionar en forma sencilla las políticas complejas de autorización modelando y analizando los requerimientos de seguridad de alto nivel, asignar roles a los usuarios y crear plantillas de políticas de control de acceso a los recursos. En [15] son analizados otros modelos y metodologías vigentes en el SDLC que soportan el desarrollo de productos con garantías de seguridad como resultado de que las comunidades de la Ingeniería de Calidad y de la Ingeniería de la Seguridad comenzarán a afrontar este problema común.

Las contradicciones expuestas en las propuestas de los distintos autores, no hace más que evidenciar la inmadurez vigente respecto a modelos y metodologías referentes a las consideraciones de seguridad en el SDLC, pero, por otro lado se puede considerar que se está ante el surgimiento de una nueva rama de la ingeniería. Algunos investigadores proponen denominarla Ingeniería del Software Seguro [16], cuyo alcance comprende, entre otras, a la ingeniería de requerimiento de seguridad, el modelo de seguridad y el desarrollo de software seguro. Su principal objetivo como campo de investigación es la producción de técnicas, métodos, procesos y herramientas que integren los principios de la ingeniería de seguridad y de calidad, y que le permitan a los desarrolladores de software analizar, diseñar, implementar, testear y desplegar sistemas de software seguro.

7 Conclusiones

La variedad de modelos, frameworks, metodologías, etc., las contradicciones conceptuales entre ellas, la falta de estándares, y la escacez de herramientas que permitan dar un tratamiento integral a la problemática de la seguridad en el proceso de desarrollo de software para las aplicaciones, revela un importante grado de inmadurez en esta área de investigación y desarrollo. Esto tiene al menos dos aspectos que pueden considerarse positivos: en primer lugar la conciencia por parte de la comunidad de desarrollo de software de la necesidad de incorporar a la Seguridad como atributo organizacional transversal en los proyectos de desarrollo o adquisición

de software para aplicaciones, y en segundo lugar, se abre un amplio campo de investigación en lo que se ha denominado "*Ingeniería del Software Seguro*" en pos de una metodología estandarizada y un framework integral.

Quedan pendientes de análisis el impacto en la seguridad de los nuevos paradigmas de procesamiento de la información surgidos de tecnologías emergentes tales como "*Computación en la Nube*" (Cloud Computing), la cual abarca una amplia variedad de modelos para ejecutar software de aplicación, incluso por fuera de las fronteras de una organización.

Referencias

1.Hamdi, Bouhoula, Mosbah; "*Declarative Approach for Easy Specification and Automated Enforcement of Security Policy*"; February 2008.
2.INTERNATIONAL STANDARD ISO/IEC FDIS 27002:2005; "*Tecnología de la Información – Técnicas de Seguridad – Código de Práctica para la Gestión de la Seguridad de la Información*".
3.Canavan, Diver: "*Information Security Policy - A Development Guide for Large and Small Companies*"; SANS Institute 2007.
4.IBM Corp; "*IBM Tivoli Security Policy Manager*"; 2009.
5. IBM Corp, "*Enterprise Security Architecture Using IBM Tivoli Security Solutions*", 2007.
6.OWASP, "*Principle*", http://www.owasp.org/index.php/Category:Principle
7.Feck, "*Modelado de Amenazas, una herramienta para el tratamiento de la seguridad en el diseño de sistemas*", CNEISI 2010.
8.CC/CEM Documentation , http://www.niap-ccevs.org/cc_docs/
9.Rumbaugh, Jacobson, Booch; "*The Unified Modeling Language Reference Manual*"; Addison-Wesley, 1999.
10.Booch, Maksimchuk, Engle, Young, Conallen, Houston;"*Object-Oriented Analysis and Design with Applications*";Third Edition, Addison-Wesley, 2007.
11.Lalanam, Finin; "*Modeling Conversation Policies using Permissions and Obligations*"; University of Maryland Baltimore County; 2004
12.Crook, Ince, Lin,Nuseibeh;"*Security Requirements Engineering:When Anti-requirements Hit the Fan*"; The Open University Walton Hall
13.Moffett, Nuseibeh; "*A Framework for Security Requirements Engineering*", University of York; Report YCS 368.
14.McGraw, Chess, Migues; "*Building Security In Maturity Model*".
15.Castellaro, Romaniz, Ramos, Pessolani, "*Hacia la Ingeniería de Software Seguro*", 2009.
16.Mouratidis, Giorgini;"*Integrating Security and Software Engineering: Advances and Future Visions*" ; Idea Group Publishing; 2007; ISBN Idea Group Publishing.

Agradecimientos

El autor agradece las sugerencias propuestas y la colaboración brindada por la Lic. Marta Castellaro, y al resto de los integrantes del proyecto "*Análisis de la Seguridad como atributo organizacional transversal en los proyectos de Sistemas y Tecnologías de Información*" de Universidad Tecnológica Nacional - Facultad Regional Santa Fe, por el apoyo brindado, sus consejos y comentarios durante el proceso de redacción y revisión del presente artículo.