Actas de la IX Reunión Española sobre Criptología y Seguridad de la Información

Casa de Convalescència, Hospital de la Santa Creu i Sant Pau

7, 8 y 9 de septiembre del 2006, Barcelona

Departament d'Enginyeria de la Informació i les Comunicacions,
Universitat Autònoma de Barcelona
Estudis d'Informàtica, Multimèdia i Telecomunicacions,
Universitat Oberta de Catalunya

Editores
Joan Borrell Viader
Jordi Herrera Joancomartí

Editores: Joan Borrell Viader y Jordi Herrera Joancomartí.

Primera edición: julio 2006.
ISBN: 84-9788-502-3

Prólogo

Esta publicación recoge las actas de la Reunión Española sobre Criptología y Seguridad de la Información (RECSI), celebrada los días 7, 8 y 9 de septiembre del 2006 en Barcelona.

La RECSI llega en el año 2006 a su novena edición, organizada de forma conjunta por el Departamento de Ingeniería de la Información y de las Comunicaciones de la Universidad Autònoma de Barcelona y el Departamento de Informática y Multimedia de la Universidad Oberta de Catalunya. Esta IX RECSI quiere seguir siendo el lugar de encuentro y el foro en el que los criptólogos y, en general, todos aquellos que trabajan en el campo de la Seguridad de la Información expongan sus hallazgos y debatan sus ideas. Se trata de un congreso bienal que se celebra en universidades y centros de investigación de España. Las ediciones anteriores se llevaron a cabo en Palma de Mallorca (U. Illes Balears), Madrid (CSIC), Barcelona (U. Politècnica de Catalunya), Valladolid (U. de Valladolid), Torremolinos (U. de Málaga), Santa Cruz de Tenerife (U. de La Laguna), Oviedo (U. de Oviedo) y Leganés (U. Carlos III).

La expansión de Internet, el incremento exponencial del volumen de datos automatizados que se maneja, la creciente inquietud por la protección de la intimidad y, en general, la entrada en la era de la información hace que la seguridad de ésta se configure como un campo de singular importancia, y por ello concentre un especial interés por parte de las empresas, las administraciones, los profesionales y más ampliamente, la sociedad entera. Por otro lado, la Criptología, en su doble vertiente de diseño de algoritmos criptográficos y de análisis de sus posibles debilidades, se ha convertido en la disciplina vertebral de la seguridad, habiendo abandonado los círculos impenetrables en los que se desplegaba históricamente, para ser tratada en universidades, centros de investigación, empresas y organismos de todo tipo interesados en proteger las informaciones que manejan.

Conscientes de lo anterior, en la IX RECSI se tratan y profundizan los aspectos de estas materias que más despiertan la atención en estos días, así como otros, aún en investigación, pero que están llamados a ser de capital importancia en los sistemas y mecanismos de seguridad en un inmediato futuro. A lo largo de las tres jornadas que conforman la Reunión se presentan 63 comunicaciones en 18 sesiones paralelas. Queremos agradecer desde estas líneas el trabajo realizado por el Comité Científico y los revisores en el proceso de revisión.

La IX RECSI, buscando mantener un elevado nivel académico y también un adecuado nivel de contacto de la comunidad investigadora con las empresas y la sociedad, incluye también:

- Tres conferencias magistrales a cargo de investigadores de reconocido prestigio en el ámbito de la Criptología y la Seguridad de la Información, el Dr. Moni Naor, del Weizmann Institute of Science (Israel), el Dr. Frédéric Cuppens de la Escuela Normal Superior de Telecomunicaciones de Bretaña

(Francia) y el Dr. Gene Tsudik de la Universidad de California en Irvine (USA).

- Dos presentaciones de empresas, Safelayer Secure Communications, compañía líder en el mercado de seguridad y confianza para las TIC, desarrollando tecnología de identificación electrónica, firma electrónica y protección de datos basada en Infraestructura de Clave Pública (PKI), y Scytl Secure Electronic Voting, compañía líder en el desarrollo de plataformas de votación electrónica seguras y confiables, aplicables desde procesos electorales clásicos a juntas generales de accionistas.
- La presentación de la Unidad Central de Informática Forense de la Policía de la Generalitat de Catalunya - Mossos d'Esquadra.

Manifestar también nuestro agradecimiento por la ayuda financiera y de difusión recibida de los distintos patrocinadores, cuya relación aparece en la página de agradecimientos de estas actas.

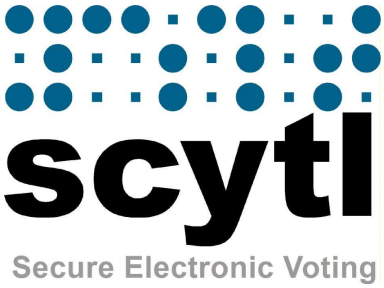
No quisieramos finalizar este prólogo sin recordar a nuestro amigo Andreu Riera Jorba, participante en varias Reuniones, tristemente fallecido en accidente de coche el 11 de marzo de 2006. Andreu, doctor por la UAB, era conocido tanto por su valiosa aportación en el campo de la criptografía aplicada al voto electrónico, como por su espíritu emprendedor que le llevó a fundar Scytl Secure Electronic Voting, empresa de la cual era Consejero Delegado.

Septiembre 2006

Joan Borrell Viader
Jordi Herrera Joancomartí

Agradecimientos

Los organizadores de la RECSI quieren agradecer a los patrocinadores de la Reunión su apoyo logístico y económico.



Organización

La IX RECSI ha sido organizada conjuntamente por el Departament d'Enginyeria de la Informació i les Comunicacions de la Universitat Autònoma de Barcelona y los Estudis d'Informàtica, Multimèdia i Telecomunicacions de la Universitat Oberta de Catalunya.

Comité ejecutivo

Joan Borrell Viader
Jordi Herrera Joancomartí
Josep Rifà Coma

Comité científico

Abascal Fuentes, Policarpo (U. de Oviedo)
Arranz Chacón, Maria Luisa (Alcatel)
Areitio Bertolín, Javier (U. de Deusto)
Borrell Viader, Joan (U.Autònoma de Barcelona)
Caballero Gil, Pino (U. de La Laguna)
Dávila Muro, Jorge (U. Politécnica de Madrid)
Domingo-Ferrer, Josep (U. Rovira i Virgili)
Fernández-Medina Patón, Eduardo (U. de Castilla La Mancha)
Ferrer Gomila, Josep Lluis (U. de les Illes Balears)
Fúster Sabater, Amparo (CSIC)
Gómez Skarmeta, Antonio (U. de Múrcia)
González Jiménez, Santos (U. de Oviedo)
Guía Martínez, Dolores de la (CSIC)
Gutiérrez Gutiérrez, Jaime (U. de Cantabria)
Herrera Joancomartí, Jordi (U. Oberta de Catalunya)
Huguet Rotger, Llorenç (U. de les Illes Balears)
López Muñoz, Javier (U. de Málaga)
Martín del Rey, Angel (U. de Salamanca)
Mañas Argemí, José Antonio (U. Politécnica de Madrid)
Miret Biosca, Josep Maria (U. de Lleida)
Padró Laimon, Carles (U. Politècnica de Catalunya)
Peinado Domínguez, Alberto (U. de Málaga)
Ramió Aguirre, Jorge (U. Politécnica de Madrid)
Ramos Álvarez, Benjamín (U. Carlos III de Madrid)
Ribagorda Garnacho, Arturo (U. Carlos III de Madrid)
Rifà Coma, Josep (U. Autónoma de Barcelona)
Robles Martínez, Sergi (U. Autónoma de Barcelona)

Salazar Riaño, Jose Luís (U. de Zaragoza)
Sempere Luna, José Maria (U. Politécnica de Valencia)
Soriano Ibáñez, Miquel (U. Politècnica de Catalunya)
Rifà Coma, Josep (U. Autònoma de Barcelona)
Tena Ayuso, Juan (U. de Valladolid)
Villar Santos, Jorge (U. Politècnica de Catalunya)

Comité Organizador

Joan Arnedo (Universitat Oberta de Catalunya)
Carles Garrigues (Universitat Autònoma de Barcelona)
David Megías (Universitat Oberta de Catalunya)
Alvaro Moratalla (Universitat Autònoma de Barcelona)
Guillermo Navarro (Universitat Autònoma de Barcelona)
Josep Prieto (Universitat Oberta de Catalunya)
Segi Robles (Universitat Autònoma de Barcelona)
Jordi Serra (Universitat Oberta de Catalunya)
Pere Urbón (Universitat Autònoma de Barcelona)

Revisores

Guillermo Azuara Guillén
Óscar Cánovas Reverte
Jordi Castellà Roca
Sergio Castillo Pérez
Vanesa Daza Fernández
Oscar Esparza Martín
Juan M. Estévez Tapiador
Joaquín García Alfaro
Félix J. García Clemente
Maria Isabel González Vasco
Julio César Hernández Castro
Gabriel López Millán
Consuelo Martínez López
Antoni Martínez Ballesté
Gregorio Martinez Perez
José Luis Muñoz-Tapia
Josep Pegueroles
Joan Josep Piles Contreras
Helena Rifà Pous
Francesc Sebé Feixas
Agusti Solanas Gómez

Índice general

Sesión C1

Sesión S1

Sesión C2

Sesión S2

Sesión C3

Sesión S3

Sesión C4

Sesión S4

Sesión C5

Sesión S5

Sesión S6

Sesión S7

Sesión C6

Sesión S8

Sesión S12

Hacia un Proceso de Ingeniería de Requisitos de Seguridad para el Desarrollo de Sistemas de Información Seguros

Daniel Mellado[1], Eduardo Fernández-Medina[2], and Mario Piattini[2]

[1] Ministerio de Trabajo y Asuntos Sociales;
Instituto Nacional de la Seguridad Social;
Centro Informático; Madrid, España.
Daniel.Mellado@alu.uclm.es

[2] Grupo ALARCOS, Departamento de Tecnologías y Sistemas de Información,
Centro Mixto de Investigación y Desarrollo de Software UCLM-Soluziona,
Universidad de Castilla-La Mancha.
Paseo de la Universidad 4 - 13071, Ciudad Real, España.
Eduardo.FdezMedina, Mario.Piattini@uclm.es

Resumen En este artículo se presenta una propuesta para el establecimiento de requisitos de seguridad para el desarrollo de Sistemas de Información (SI) seguros. Proponemos un método basado en los activos y dirigido por el riesgo, el cual se apoya en la reutilización de requisitos de seguridad, mediante la utilización de un repositorio de recursos de seguridad. Y que además integra los Criterios Comunes en el modelo tradicional del ciclo de vida del software, de tal modo que es conforme a la ISO/IEC 15408. Basándonos en el concepto de la construcción iterativa de software, proponemos un micro-proceso para el análi-sis de de requisitos de seguridad, que es repetidamente efectuado en cada nivel de abstracción durante el desarrollo incremental. En resumen, presentamos una propuesta que trata con los requisitos de seguridad en las primeras fases del de-sarrollo software de una forma sistemática e intuitiva, y que además es confor-me al estándar ISO/IEC 17799:2005.

1. Introducción

Hoy en día los Sistemas de Información (SI) son vulnerables a multitud de amenazas. Además, cuanto más se incrementa la complejidad de las aplicaciones y servicios, más aumenta la potencialidad de sufrir brechas de seguridad [22]. Y en la actual Sociedad de la Información, dependiente de un gran número de sistemas software que tienen una misión crítica, es absolutamente vital que los SI sean asegurados apropiadamente desde el principio [1, 12], debido a las potenciales pérdidas a las que se enfrentan las organizaciones que confían en todos estos SI. Como sabemos, es ampliamente aceptado el principio que establece que la cons-trucción de la seguridad en las etapas tempranas del proceso de desarrollo es más eficaz respecto a los costes y tiene como resultado diseños más robustos [10]. Sin embargo, el gran problema es que en la mayoría de los proyectos software

la seguridad se trata una vez el sistema ha sido diseñado e implementado. Debido en muchos casos, a una gestión inapropiada de la especificación de los requisitos de seguridad del nuevo sistema, ya que la denominada fase de especificación de requisitos suele realizarse con unas cuantas descripciones o la especificación de objetivos plasmados en unos pocos folios [4]. Además, habitualmente los requisitos de seguridad no son bien entendidos en sí. De forma que, incluso cuando se intenta especificar los requisitos de seguridad, muchos desarrolladores tienden a describir soluciones de diseño en términos de mecanismos de protección en lugar de realizar proposiciones declarativas sobre el grado de protección requerido [5].

Una parte muy importante en el proceso de desarrollo software para conseguir sistemas software seguros es la denominada Ingeniería de Requisitos de Seguridad. La cual proporciona técnicas, métodos y normas para abordar esta tarea en el ciclo de desarrollo de los SI. Y debería implicar el uso de procedimientos repetibles y sistemáticos para asegurar que el conjunto de requisitos obtenidos es completo, consistente y fácilmente comprensible y analizable por parte de los diferentes actores implicados en el desarrollo del sistema [11]. Un buen documento de requisitos debe incluir tanto requisitos funcionales (relativos a los servicios que el software o sistema debe proporcionar) y los no-funcionales (relativos a las denominadas características de calidad, como rendimiento, portabilidad, seguridad etc.) [4]. Por su parte, la seguridad debe ser considerada durante todo el proceso de desarrollo y debería ser definida conjuntamente con la especificación de requisitos [16].

En este artículo, después de haber analizado en [14] y [15] diferentes propuestas técnicas relevantes relativas a los requisitos de seguridad para el desarrollo de SI seguros, presentamos SREP (Proceso de Ingeniería de Requisitos de Seguridad, cuyas siglas son en inglés y se descomponen en "Security Requirements Engineering Process"), en el cual se explica como integrar los requisitos de seguridad en el proceso de ingeniería del software de una forma sistemática e intuitiva. Nuestra propuesta está basada en la integración de los Criterios Comunes (CC) en el modelo tradicional de ciclo de vida del software, junto con la reutilización de requisitos de seguridad que son compatibles con el subconjunto del Marco de Trabajo de los CC (CCF). Además, para facilitar esta tarea y poder llevarla a cabo, proponemos utilizar varios conceptos y técnicas: un repositorio de recursos de seguridad (con activos, amenazas, requisitos, etc.), el uso de UMLSec [19], casos de mal uso [20], árboles de ataque, y casos de uso de seguridad [6]. En definitiva, planteamos un método basado en los activos y dirigido por el riesgo para el establecimiento de requisitos de seguridad. El resto del artículo esta organizado de la siguiente forma: en la sección 2, se ofrece una visión general de SREP. La sección 3 presenta las características principales del proceso. A continuación, en la sección 4, explicamos el repositorio de recursos de seguridad. Seguidamente, describimos el modelo del proceso en la sección 5. Y finalmente, presentamos nuestras conclusiones y futuros trabajos en la sección 6.

2. Visión General de SREP

SREP (Proceso de Ingeniería de Requisitos de Seguridad, cuyas siglas son en inglés y se descomponen en "Security Requirements Engineering Process") es un método

basado en los activos y dirigido por el riesgo para el establecimiento de requisitos de seguridad en el desarrollo de SI seguros. Este proceso básicamente describe cómo integrar los CC en el ciclo de vida tradicional del software, junto con el uso de un repositorio de recursos de seguridad que permita la reutilización de requisitos de seguridad (los cuales pueden estar modelados con UMLSec, o expresados como casos de uso de seguridad, o como texto plano con una especificación formal), activos, amenazas (que pueden ser representadas mediante casos de mal uso, árboles de ataque, diagramas UMLSec o en texto plano por ejemplo en forma de lista de verificación ('checklist')) y medidas de salvaguarda (contramedidas). El foco de este proceso busca construir la seguridad del SI en las primeras fases del ciclo de vida de desarrollo.

A continuación presentamos una descripción de SREP en la Fig. 1.

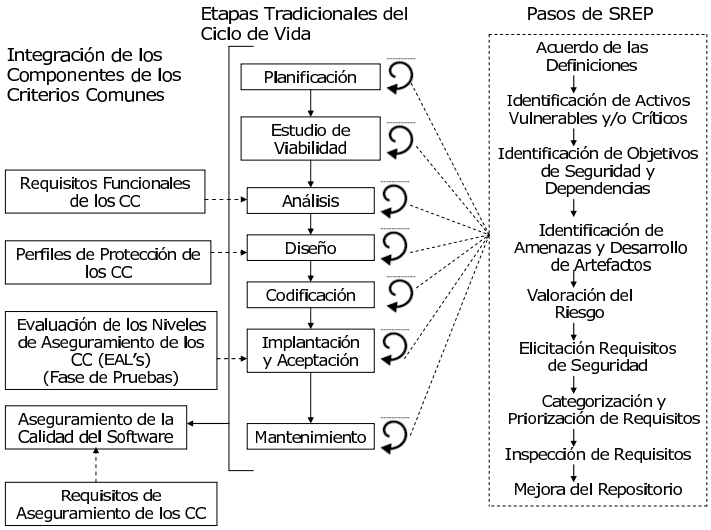


Fig. 1 Visión general de SREP

Tal y como se describe en la Fig. 1, el núcleo de SREP es un micro-proceso, que se compone de nueve pasos que son realizados repetidamente en cada fase del ciclo de vida. Al mismo tiempo, los Componentes de los CC son introducidos en el ciclo de vida del software de manera que SREP usa los diferentes Componentes CC según la fase, aunque las actividades de Aseguramiento de la Calidad (SQA) se realizan a lo largo de todas las fases del ciclo de vida de desarrollo del software. Y son en estas actividades de SQA donde los requisitos de aseguramiento de los CC deben ser incorporados. En las secciones siguientes se presentará una exposición más detallada de la integración de los CC y de SREP en ciclo de vida de desarrollo de SI.

3. Características de SREP

En términos generales las principales características de SREP son:

- Iterativo e incremental. El modelo elegido por SREP es iterativo e incremental, de este modo los requisitos de seguridad evolucionan a lo largo del ciclo de vida; por ejemplo durante el diseño se puede mejorar la especificación con requisitos relacionados con el entorno tecnológico y con las medidas de salvaguarda asociadas. El concepto clave es la existencia de un micro-proceso para el análisis de requisitos de seguridad [2], compuesto por nueve pasos, que es repetidamente efectuado en cada nivel de abstracción durante el desarrollo incremental. En cada iteración se realizan todos los pasos definidos en SREP, y cada salida de cada iteración completa, mejora y refina la Especificación de los Requisitos de Seguridad, mediante la adición de nuevos requisitos de seguridad o mediante la corrección o mayor especificación/detalle de los existentes.
- Facilita la reutilización. Proponemos un repositorio de recursos de seguridad (basado en la propuesta de Sindre, Firesmith and Opdahl [20]). El propósito del desarrollo con reutilización de requisitos es identificar descripciones de sistemas que pueden ser usadas (ya sea total o parcialmente) con un número mínimo de modificaciones, reduciéndose así el esfuerzo total de desarrollo [3]. Además, la reutilización de requisitos de seguridad ayuda a incrementar su calidad: inconsistencias, errores, ambigüedades y otros problemas pueden ser detectados y corregidos para poder ser usados en proyectos sucesivos [21]. De esta manera, se nos garantizará que los ciclos de desarrollo son realizados lo más rápido posible y estando a la vez basados en soluciones probadas.
- Facilita la trazabilidad de los requisitos de seguridad a lo largo de los distintos niveles de abstracción gracias a la estructura del repositorio.
- Soporta e incluye conceptos y técnicas dentro del campo de la Ingeniería de Requisitos de Seguridad y del Análisis y Gestión de Riesgos, como UMLSec, los casos de uso de seguridad, los casos de mal uso, o los árboles de ataque.
- Finalmente, SREP es conforme con varios estándares del campo de la Ingeniería de Requisitos y de la Gestión de la Seguridad, como la norma ISO/IEC 17799:2005 y la ISO/IEC 15408.

3.1 Alineación de SREP con los Estándares

SREP es conforme a las recomendaciones de la norma ISO/IEC 17799:2005 en lo relativo a requisitos de seguridad. Ya que en SREP se propone justo lo que dicha norma recomienda: “Los requisitos de seguridad deberían ser identificados y consensuados previamente al desarrollo y/o implementación de los SI. Todos los requisitos de seguridad deberían ser identificados y justificados en la fase de requisitos de un proyecto, consensuados y documentados como parte del proceso de negocio global para un SI”.

Además, tenemos en cuenta el estándar IEEE 830-1998, de tal modo que el paso de ‘Inspección de Requisitos’ del micro-proceso de análisis de requisitos de seguridad comprueba si los requisitos de seguridad son conformes a la norma, verificando que sean correctos, no ambiguos, completos, consistentes, ordenados por importancia y/o estabilidad, verificables, modificables y trazables. Por lo tanto, todos estos factores de

calidad son comprobados al final de cada iteración del micro-proceso, justo antes del paso de 'Mejora del Repositorio'.

Los CC (ISO/IEC 15408) son el catálogo de requisitos estándar para la evaluación de sistemas donde la seguridad es crítica. Usando los CC, un gran número de requisitos de seguridad pueden ser definidos en el propio sistema y en el desarrollo del sistema. Y el esquema de los CC puede incorporarse al ciclo de vida del software de aplicaciones nuevas y existentes para satisfacer requisitos de seguridad más estrictos. Por ello proponemos incorporarlos. Y esto puede realizarse según Kam [9], mediante: la incorporación de los requisitos funcionales de los CC en la Especificación de Requisitos del Software; integración de los requisitos de aseguramiento de los CC en las actividades de Aseguramiento de la Calidad (SQA); introduciendo los EALs (Evaluation Assurance Levels) en el plan de pruebas del software; e incorporando los Perfiles de Protección de los CC en el diseño de la arquitectura. Aunque una explicación detallada de las últimas formas de integración de los CC en el ciclo de vida esta fuera del alcance de este artículo.

4. El Repositorio de Recursos de Seguridad

SREP esta basado en varias técnicas actuales que tratan con los requisitos de seguridad, con el fin de facilitar el tratamiento sistemático e intuitivo de los mismos en las primeras fases del desarrollo de software. Las principales técnicas que se emplean son las que a continuación se presentan:

- *UMLSec* te permite expresar información relativa a la seguridad dentro de los diagramas UML de especificación del sistema, de modo que su propósito es integrarse más en los artefactos producidos durante el proceso de desarrollo. Se proporciona una extensión en forma de perfil UML usando los mecanismos de extensión del estándar UML. Se usan estereotipos y etiquetas para formular los requisitos de seguridad y las suposiciones del entorno del sistema; las restricciones dan un criterio para determinar si el diseño del sistema cumple los requisitos [19]. En SREP, UMLSec puede ser usado para especificar las amenazas o los requisitos de seguridad, siendo en este último caso un método complementario a los casos de uso de seguridad.
- *Casos de Uso de Seguridad* son una técnica que usamos para especificar los requisitos de seguridad que la aplicación debe cumplir para protegerse satisfactoriamente de las amenazas de seguridad más relevantes [6]. Y son dirigidos por los casos de mal uso.
- *Casos de Mal Uso* son un tipo especializado de casos de uso que son usados para analizar y especificar amenazas de seguridad [6]. Son el caso opuesto a los casos de uso, representan una función que el sistema no debería permitir. Se puede definir de una forma más precisa como una secuencia completa de acciones cuyo resultado es la consecución de pérdidas en la organización o de alguna de las partes interesadas ('stakeholders') específicas [20]. En nuestra propuesta, los casos de mal uso pueden ser usados para especificar las amenazas y son los que dirigen los casos de uso de seguridad..

El *Repositorio de Recursos de Seguridad* (RRS) almacena todos los elementos reutilizables que pueden ser usados por los analistas o ingenieros de requisitos. Además, el repositorio entiende los conceptos de dominio y perfil [21]. El primero consiste en la pertenencia a un campo específico o a unas áreas funcionales de aplicación, como el comercio electrónico. En cambio, el concepto de perfil se refiere a un conjunto homogéneo de requisitos que pueden ser aplicados a diferentes dominios, como por ejemplo la legislación relativa a la protección de datos de carácter personal. En relación con esto, nosotros planteamos establecer un atributo a cada uno de los elementos del RRS que sirva para mostrar en que dominio/s podría ser usado dicho elemento. Por contra, los perfiles junto con los Perfiles de Protección de los CC son guardados como subconjuntos estandarizados de requisitos de seguridad específicos, y con ellos sus elementos asociados en el RRS (amenazas, etc.). En resumen, cada dominio o perfil son una vista del RRS global.

Adicionalmente, los elementos que el RRS contiene son establecidos de forma genérica, mediante el uso de mecanismos basados en parámetros, como plantillas parametrizadas reutilizables. Aunque hay también plantillas no parametrizadas y listas de verificación ('checklists'). Además, cada requisito de seguridad y su especificación son etiquetados como Requisito del Sistema (IEEE Std. 1233 y IEEE Std. 1207.1) o como Requisito Software (IEEE Std. 830-1998) [21].

A continuación se muestra en la Fig. 2 el meta-modelo del RRS, el cual es una extensión de la propuesta de repositorio formulada por Sindre, G., D.G. Firesmith, y A.L. Opdahl [20].

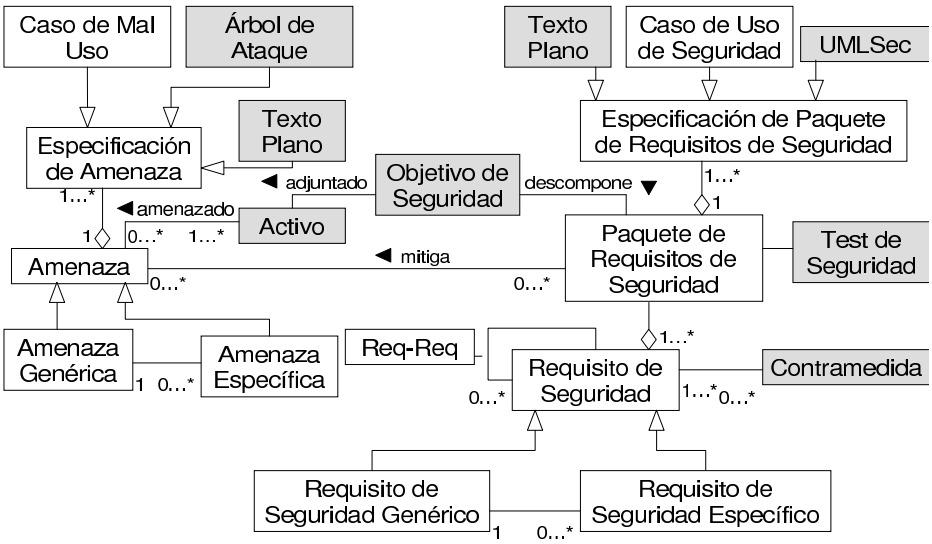


Fig. 2 **Meta-modelo del Repositorio de Recursos de Seguridad.**

Tal y como se puede observar en la Fig. 2, se trata de un meta-modelo dirigido por los activos y por las amenazas, porque los requisitos pueden obtenerse a través de los

activos o de las amenazas. Seguidamente, describimos brevemente los más importantes y/o complejos aspectos del meta-modelo.

- 'Amenaza Genérica' y 'Requisito de Seguridad Genérico' describen amenazas y requisitos independientemente de los dominios particulares. Y pueden ser representados con especificaciones distintas gracias a los elementos 'Especificación de Amenaza' y 'Especificación de Paquete de Requisitos de Seguridad'.
- 'Paquete de Requisitos de Seguridad' es un conjunto de requisitos que se juntan para satisfacer los mismos objetivos de seguridad y mitigar las mismas amenazas. Ya que estamos de acuerdo con Sindre, G., D.G. Firesmith, y A.L. Opdahl [20] en que en muchas ocasiones son una mayor y más efectiva unidad de reutilización.
- La relación 'Req-Req' permite establecer relaciones de inclusividad o exclusividad entre requisitos. Una relación de exclusividad entre requisitos implica que son alternativos mutuamente, como por ejemplo si están en conflicto o se solapan uno al otro. Mientras que una relación de inclusividad entre requisitos significa que para satisfacer un requisito dado, se necesita satisfacer también otro u otros.

Además, pueden producirse más relaciones más adelante en el momento de la especificación del diseño, de los casos de prueba de seguridad, medidas de salvaguarda, etc. Debido a que nuestro modelo de proceso propuesto esta basado en el concepto de la construcción iterativa de software, como se explica en la siguiente sección.

Por último, queremos destacar el hecho de que usando los CC, un gran número de requisitos de seguridad pueden ser definidos en el propio sistema y en el desarrollo del mismo. Sin embargo, los CC no proporcionan soporte metodológico, ni contienen criterios de evaluación de la seguridad relativos a las medidas de seguridad administrativas no directamente relacionadas con las medidas de seguridad del SI. Aunque se sabe que una gran parte de la seguridad que alcanza un SI se obtiene mediante la aplicación de estas medidas administrativas. Por tanto, y de acuerdo con la norma ISO/IEC 17799:2005, proponemos incluir el conjunto de requisitos legales, estatutarios, regulatorios, y contractuales que deberían satisfacer la Organización, sus socios comerciales, los contratistas, y los proveedores de servicios, y su entorno sociocultural. Con lo que después de adecuar estos requisitos al formato de los requisitos del sistema o bien requisitos software, éstos constituirán el subconjunto inicial de requisitos de seguridad del RRS para cualquier proyecto. Además, si la Organización realiza alguna actividad en España de manera directa o indirecta, nosotros planteamos que el RRS contenga todos los requisitos, así como sus listas de activos, amenazas y contramedidas, que se pueden obtener de MAGERIT (Metodología de Análisis y GEstión de Riesgos de los sisTemas de información, del Ministerio de Administraciones Públicas), que es también conforme a la ISO/IEC 15408. De forma que constituya un perfil mediante el cual se facilite la conformidad con la legislación española relativa a la seguridad y la protección de datos de carácter personal.

5. Modelo del Proceso

Según Kotonya y Sommerville [11], la Ingeniería de Requisitos se compone básicamente de las siguientes fases o pasos: i) elicitación de requisitos, ii) análisis y negociación de los requisitos, iii) documentación de los requisitos y iv) validación de los re-

quisitos. Partiendo del concepto de la construcción iterativa de software, proponemos un micro-proceso para el análisis de requisitos de seguridad, compuesto por nueve pasos, que son realizados repetidamente en cada nivel de abstracción a lo largo del desarrollo incremental. Ya que el documento de especificación de los requisitos de seguridad evolucionará durante todo el ciclo de vida; por ejemplo, durante el diseño, la especificación se puede enriquecer con requisitos relacionados con el entorno tecnológico. Además , cada requisito de seguridad puede ser rastreado a lo largo de los distintos niveles de abstracción. Y también, dado que el modelo entiende los conceptos de perfil y dominio (que pueden tener elementos de distinto nivel de abstracción), los requisitos pueden ser analizados por las partes interesadas ('stakeholders') que tengan mayor conocimiento y/o responsabilidad en ese dominio. Por ultimo, estamos de acuerdo con Nuseibeh [18] en que la Ingeniería de Requisitos y el diseño de la arquitectura son procesos concurrentes y que se influyen mutuamente.

Los nueve pasos en los que se basa SREP (basados en [20] y [13]) y que componen el micro-proceso de análisis de requisitos de seguridad son los siguientes:

- Paso 1: Acuerdo de las definiciones
- Paso 2: Identificación de activos vulnerables y/o críticos
- Paso 3: Identificación de objetivos de seguridad y dependencias
- Paso 4: Identificación de amenazas y desarrollo de artefactos.
- Paso 5: Valoración del riesgo
- Paso 6: Elicitación requisitos de seguridad
- Paso 7: Categorización y priorización de requisitos
- Paso 8: Inspección de requisitos
- Paso 9: Mejora del repositorio

i) Elicitación de Requisitos

ii) Análisis y Negociación de los Requisitos

iii) Documentación y iv)Validación

- *Paso 1: Acuerdo de las definiciones*. La primera tarea para la Organización es acordar un conjunto común de definiciones de seguridad, junto con la definición de los objetivos finales de seguridad de la Organización. Las definiciones candidatas serán las del IEEE y otros estándares. Y es importante que en este paso, que se realizará una sola vez, participen las partes interesadas ('stakeholders') y el equipo de requisitos.
- *Paso 2: Identificación de activos vulnerables y/o críticos*. Es en este punto donde el RRS puede ser usado por primera vez. Y consiste en la identificación de los diferentes tipos de activos valiosos o críticos y/o vulnerables, y es realizado por el ingeniero de requisitos, que puede ayudarse mediante:
  - Listas de activos extraídas del RRS, donde el analista / ingeniero de requisitos puede buscar por dominio, e incluso puede seleccionar un perfil concreto.
  - Los requisitos funcionales.
  - Las entrevistas con las partes interesadas ('stakeholders').
- *Paso 3: Identificación de objetivos de seguridad y dependencias*. En este paso el ingeniero de requisitos puede ayudarse también del RRS, ya que cada activo tiene adjuntado unos objetivos de seguridad. Para cada activo identificado en el paso

anterior se seleccionan los requisitos de seguridad apropiados para el activo, e identifica las dependencias entre ellos. Los objetivos de seguridad se expresan especificando el nivel de seguridad necesario en términos de probabilidad y en tipos probables de atacantes.

- *Paso 4: Identificación de amenazas y desarrollo de artefactos*. Cada activo es amenazado por una/s amenaza/s que pueden impedir la consecución del objetivo/s de seguridad de dicho activo. Con lo que primeramente hay que encontrar todas las amenazas que apuntan sobre los activos, con la ayuda del RRS. Además, puede que sea necesario desarrollar artefactos (como casos de mal uso, árboles de ataque, o casos de uso y diagramas de clases o de secuencia o estado usando UMLSec) para especificar nuevas amenazas y/o requisitos genéricos y específicos. Ya que es necesario buscar por nuevas amenazas que no estén relacionadas con los activos a través del RRS, porque es posible que algunos objetivos de seguridad y activos se hayan podido olvidar en pasos anteriores o bien dichas amenazas no hayan sido introducidas aún en el RRS.
- *Paso 5: Valoración del riesgo*. El riesgo generalmente tiene que ser determinado específicamente para cada aplicación, y teniendo siempre en cuenta que la meta final es conseguir el 100% de aceptación del riesgo. En primer lugar hay que valorar si las amenazas son relevantes según el nivel de seguridad especificado por los objetivos de seguridad. Después estimar los riesgos de seguridad basándose en las amenazas relevantes, su probabilidad de ocurrencia y su potencial impacto negativo. Para ello se puede usar cualquier metodología de análisis de riesgos de seguridad, como por ejemplo MAGERIT. De este modo, esta valoración nos permite saber cómo la tolerancia a riesgos de la Organización se ve afectada respecto de cada amenaza. En este paso, es recomendable que el ingeniero de requisitos sea ayudado por un experto en riesgos y por las partes interesadas (‘stakeholders’).
- *Paso 6: Elicitación requisitos de seguridad*. El RRS puede ser usado de nuevo en este paso. Para cada amenaza extraída del repositorio, se puede encontrar uno o más paquetes de requisitos de seguridad asociados. El ingeniero de requisitos debe seleccionar los requisitos de seguridad o los paquetes de requisitos de seguridad más adecuados que mitiguen las amenazas al nivel necesario según la valoración del riesgo asociado a la misma. Sin embargo, puede que también se encuentren requisitos de seguridad o paquetes de requisitos de seguridad adicionales por otros medios. Además, debe de especificarse el test de seguridad para cada paquete de requisitos de seguridad así como las contramedidas para cada requisito de seguridad, aunque éstos sean refinados a nivel de diseño, haciéndose un esbozo de los mismos en la fase de análisis. Ya que coincidimos con Firesmith [5] en que se debería tener cuidado en evitar la especificación innecesaria y prematura de mecanismos arquitectónicos.
- *Paso 7: Categorización y priorización de requisitos*. Cada requisito es categorizado y priorizado de forma cualitativa. De tal manera que los requisitos más importantes (en términos de impacto y probabilidad de ocurrencia) sean gestionados primero. Esta tarea es realizada por el ingeniero de requisitos junto con otro tipo de especialistas en dominios concretos (si es necesario). Al final de este proceso, se procede a la documentación del conjunto de requisitos obtenidos.

- *Paso 8: Inspección de requisitos*. La inspección de requisitos se realiza para validar los requisitos, así como los elementos modificados del modelo y los nuevos elementos generados. Tiene como fin la revisión de la calidad del trabajo realizado por el equipo y de los entregables generados. Y es realizada por el equipo de inspección, de manera que sirva como comprobación final de la calidad de los resultados obtenidos en la iteración.
- *Paso 9: Mejora del repositorio*. En primer lugar se realiza la validación de los requisitos con la participación de las partes interesadas ('stakeholders') y del ingeniero de requisitos. Después, los nuevos elementos del modelo (amenazas, requisitos, etc…) encontrados a lo largo del desarrollo de los pasos anteriores y susceptibles de ser usados en futuras aplicaciones, son introducidos en el RRS. Además, los elementos del modelo ya existentes en el repositorio pueden ser modificados para mejorar su calidad.

Por último, y al mismo tiempo que integramos los requisitos funcionales de los CC en estos nueve pasos, proponemos esbozar/perfilar los EAL's en el plan de pruebas software y verificarlos durante las pruebas de ejecución. Y paralelamente, planteamos incorporar los requisitos de aseguramiento de los CC en las actividades de aseguramiento de la calidad (SQA), como las actividades de control de calidad, prevención y eliminación de errores [9]. Por tanto, de este modo el plan de gestión de la configuración es la primera actividad que explícitamente tiene que seguir los requisitos de aseguramiento de los CC, para conseguir nuestro objetivo de que los CC sean integrados en el ciclo de vida de desarrollo tradicional.

6. Conclusiones y Futuros Trabajos

Hoy en día, en la llamada Sociedad de la Información, dada la criticidad creciente de los SI unido a los nuevos requisitos legales y gubernamentales, se hace necesario el desarrollo de enfoques cada vez más sofisticados para asegurar la seguridad de la información. Normalmente, la seguridad de la información se abordaba desde el punto de vista técnico en la fase de implementación, y aunque se trata de un aspecto importante, consideramos fundamental el tratamiento de la seguridad en todas las fases del desarrollo de SI, especialmente en el establecimiento de los requisitos de seguridad, ya que constituyen la base para la consecución de un SI robusto.

Por ello presentamos una propuesta que trata con los requisitos de seguridad en las primeras fases de desarrollo del software, y que está basado en la reutilización de requisitos de seguridad, para lo cual proporciona un Repositorio de Recursos de Seguridad (RRS). Y que propone también la integración de los Criterios Comunes en el modelo tradicional de ciclo de vida de desarrollo del software. Además, dicha propuesta es conforme a los estándares ISO/IEC 15408 e ISO/IEC 17799:2005. Y tiene como fin la consecución del 100% de aceptación del riesgo, ya que a pesar de los mejores esfuerzos de los investigadores en seguridad es imposible garantizar el 100% de la seguridad [17]. Partiendo del concepto de la construcción iterativa de software, proponemos un micro-proceso para el análisis de requisitos de seguridad, compuesto por nueve pasos, que son realizados repetidamente en cada nivel de abstracción a lo largo del desarrollo incremental. Finalmente, uno de los aspectos más relevantes de

nuestra propuesta es que integra otros enfoques o técnicas, como SIREN [21], UML-Sec [19], los casos de uso de seguridad [6] o los casos de mal uso [20]. Y es compatible además con la fase WSSecReq (Web Services Security Requirements) del proceso PWSSec (Web Services Security Development Process) [7], así como puede incorporar también en su repositorio el catálogo de plantillas de requisitos de seguridad para servicios web basadas en SIREN, que Gutiérrez et al. proponen en [8].

Por ultimo, en futuros trabajos será necesario proporcionar una herramienta CARE (Computer-Aided Requirements Engineering) que soporte SREP. Al igual que un refinamiento teórico del proceso mediante su aplicación en un caso de estudio real.

Agradecimientos

Este artículo ha sido desarrollado en el contexto de los proyectos DIMENSIONS (PBC-05-012-2) Proyecto de la Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla- La Mancha y el FEDER, y los proyectos CALIPO (TIC2003-07804-CO5-03) y RETISTIC (TIC2002-12487-E) de Dirección General de Investigación del Ministerio de Ciencia y Tecnología.

Referencias Bibliográficas

1. Baskeville, R., *The development duality of information systems security.* Journal of Management Systems, 1992. **4**(1): p. 1-12.
2. Breu, R. and Innerhofer–Oberperfler, F., *Model based business driven IT security analysis.* 2005.
3. Cybulsky, J. and Reed, K., *Requirements Classification and Reuse: Crossing Domains Boundaries.* ICSR'2000, 2000: p. 190-210.
4. Fernández-Medina, E., Moya, R., and Piattini Velthus, M., *Gestión de Requisitos de Seguridad*, in *Seguridad de las Tecnologías de la Información "La construcción de la confianza para una sociedad conectada"*, AENOR, Editor. 2003. p. pp 593-618.
5. Firesmith, D.G., *Engineering Security Requirements.* Journal of Object Technology, 2003. **2**(1): p. 53-68.
6. Firesmith, D.G., *Security Use Cases.* Journal of Object Technology, 2003: p. 53-64.
7. Gutierrez, C., Fernández-Medina, E., and Piattini, M., *PWSSec: Process for Web Services Security.* IEEE ICWS'05, 2005.
8. Gutiérrez, C., Moros, B., Toval, A., Fernández-Medina, E., and Piattini, M., *Security Requirements for Web Services based on SIREN.* Symposium on Requirements Engineering for Information Security (SREIS-2005), together with the 13th IEEE International Requirements Engineering Conference – RE'05, 2005.
9. Kam, S.H., *Integrating the Common Criteria Into the Software Engineering Lifecycle.* IDEAS'05, 2005: p. 267-273.

10. Kim., H.-K., *Automatic Translation Form Requirements Model into Use Cases Modeling on UML.* ICCSA 2005, LNCS, 2005: p. 769-777.
11. Kotonya, G. and Sommerville, I., *Requirements Engineering Process and Techniques*. Hardcover ed. 1998. 294.
12. McDermott, J. and Fox, C. *Using Abuse Case Models for Security Requirements Analysis*. in *Annual Computer Security Applications Conference*. 1999. Phoenix, Arizona.
13. Mead, N.R. and Stehney, T. *Security Quality Requirements Engineering (SQUARE) Methodology*. in *Software Engineering for Secure Systems (SESS05), ICSE 2005 International Workshop on Requirements for High Assurance Systems*. 2005. St. Louis.
14. Mellado, D., Fernández-Medina, E., and Piattini, M., *A Comparative Study of Proposals for Establishing Security Requirements for the Development of Secure Information Systems.* The 2006 International Conference on Computational Science and its Applications (ICCSA 2006), Springer LNCS 3982, 2006: p. 1044-1053.
15. Mellado, D., Fernández-Medina, E., and Piattini, M., *A Comparison of the Common Criteria with Proposals of Information Systems Security Requirements.* "First International Conference on Availability, Reliability and Security" (ARES'06), 2006: p. 654-661.
16. Mouratidis, H., Giorgini, P., Manson, G., and Philp, I. *A Natural Extension of Tropos Methodology for Modelling Security*. in *Workshop on Agent-oriented methodologies, at OOPSLA 2002*. 2003. Seattle, WA, USA.
17. Myagmar, S., J. Lee, A., and Yurcik, W., *Threat Modeling as a Basis for Security Requirements*. 2005: SREIS 2005.
18. Nuseibeh, *Weaving Together Requirements and Architectures.* IEEE Computer, 2001: p. 115-117.
19. Popp, G., Jürjens, J., Wimmel, G., and Breu, R., *Security-Critical System Development with Extended Use Cases*. 2003: 10th Asia-Pacific Software Engineering Conference. p. 478-487.
20. Sindre, G., Firesmith, D.G., and Opdahl, A.L. *A Reuse-Based Approach to Determining Security Requirements*. in *Proc. 9th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'03)*. 2003. Austria.
21. Toval, A., Nicolás, J., Moros, B., and García, F., *Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach*. 2001: Requirements Engineering Journal. p. 205-219.
22. Walton, J.P., *Developing a Enterprise Information Security Policy*. 2002, ACM Press: Proceedings of the 30th annual ACM SIGUCCS conference on User services.