CIBSI 2009

16 al 18 de Noviembre
Montevideo, Uruguay

V Congreso Iberoamericano de Seguridad Informática

Actas del Congreso

Actas del V Congreso Iberoamericano de Seguridad Informática
CIBSI'09
Montevideo, Uruguay, 16 al 18 de Noviembre de 2009

Editores
Gustavo Betarte
Jorge Ramió Aguirre
Arturo Ribagorda Garnacho

ISBN: 978-9974-0-0593-8

Prefacio

Estimados colegas:

Este volumen contiene los trabajos presentados en el V Congreso Iberoamericano de Seguridad Informática (CIBSI'09) realizado en Noviembre en Montevideo, Uruguay.

Esta edición del Congreso Iberoamericano de Seguridad Informática, iniciativa de la Red Temática Iberoamericana de Criptografía y Seguridad de la información CriptoRed, ha convocado al igual que en sus anteriores ediciones a un gran número de investigadores y expertos de Latinoamérica, España y Portugal.

De 65 trabajos recibidos, el Comité de Programa Científico ha seleccionado 41 trabajos, 38 de los cuales se presentan en el evento. Los mismos proceden de investigadores de Argentina, Brasil, Chile, Colombia, España, EE.UU., Francia, México, Portugal, Uruguay y Venezuela.

El congreso cuenta asimismo con tres conferencistas que han sido invitados a presentar su trabajo de investigación en sesiones plenarias, el Dr. Gilles Barthe, de IMDEA Software de España, el Dr. Eduardo Giménez, de la Universidad de la República de Uruguay y el Dr. José Luis Piñar Mañas de la Universidad CEU San Pablo de España. Tendrá también lugar en el congreso un taller, titulado *Los retos de la protección de datos: la Ley 18331 de protección de datos personales*, a cargo del Dr. José Luis Piñar Mañas.

Desde estas páginas queremos hacer llegar nuestro profundo agradecimiento a los organizadores, autores, revisores, patrocinadores y asistentes, que son los que han hecho posible que una vez más tenga lugar este encuentro académico de expertos e investigadores en Seguridad Informática, esta vez en la ciudad de Montevideo.

Noviembre 2009

Gustavo Betarte
Jorge Ramió Aguirre
Arturo Ribagorda Garnacho
CIBSI'09

Organización de la Conferencia

CIBSI'09 es organizado por la Facultad de Ingeniería de la Universidad de la República en conjunto con la Universidad Politécnica de Madrid, a cuyos directivos asi como a todos y cada uno de los miembros del comité organizador queremos agradecer desde estas páginas.

Organización General

Gustavo Betarte, Universidad de la República, Uruguay
Jorge Ramió Aguirre, Universidad Politécnica de Madrid, España

Comité de Programa

Santiago Martín Acurio Del Pino, Pontificia Universidad Católica del Ecuador, Ecuador
Nicolás C.A. Antezana Abarca, Sociedad Peruana de Computación, Perú
Javier Areitio Bertolín, Universidad de Deusto, España
Walter Baluja García, Instituto Superior Politécnico José Antonio Echeverria, Cuba
Tomás Barros, NIC Labs, Chile
Gustavo Betarte (co-chair, Universidad de la República, Uruguay)
Joan Borrel Viader, Universidad Autónoma de Barcelona, España
Pino Caballero Gil, Universidad de La Laguna, España
Jeimy Cano Martínez, Universidad de los Andes, Colombia
Adriano Mauro Cansian, Universidade Estadual Paulista, Brasil
Hugo César Coyote Estrada, Instituto Politécnico Nacional, México
Enrique Daltabuit Godas, Universidad Nacional Autónoma de México, México
Jorge Dávila Muro, Universidad Politécnica de Madrid, España
Ángel Martín del Rey, Universidad de Salamanca, España
Josep Domingo-Ferrer, Universidad Rovira i Virgili, España
Josep Lluis Ferrer-Gomilla, Universidad de Las Islas Baleares, España
Amparo Fúster Sabater, Consejo Superior de Investigaciones Científicas CSIC, España
Luis Javier García Villalba, Universidad Complutense de Madrid, España
Roberto Gómez Cárdenas, ITESM Monterrey, México
Juan Pedro Hecht, Universidad de Buenos Aires, Argentina
Marco Aurelio Henriques, Universidade Estadual de Campinas, Brasil
Emilio Hernández, Universidad Simón Bolívar, Venezuela
Leobardo Hernández Audelo, Universidad Nacional Autónoma de México, México
Luis Hernández Encinas, Consejo Superior de Investigaciones Científicas CSIC, España
Alejandro Hevia, Universidad de Chile, Chile
Juan Guillermo Lalinde, Universidad EAFIT, Colombia
Javier López Muñoz, Universidad de Málaga, España
Julio César López, Universidade Estadual de Campinas, Brasil
Vincenzo Mendillo, Universidad Central de Venezuela, Venezuela
Carlos Mex Perera, ITESM Monterrey, México
Josep María Miret Biosca, Universidad de Lleida, España
Gaspar Modelo Howard, Universidad Tecnológica de Panamá, Panamá

Raúl Monge, Universidad Técnica Federico Santa María, Chile
Edmundo Monteiro, Universidad de Coimbra, Portugal
Guillermo Morales Luna, Centro de Investigación y Estudios Avanzados del IPN, México
Alberto Peinado Domínguez, Universidad de Málaga, España
Yoan Pinzón, Universidad Nacional de Colombia, Colombia
Tamara Rezk, INRIA Sophia Antipolis, Francia
Arturo Ribagorda Garnacho, (co-chair) Universidad Carlos III de Madrid, España
Josep Rifà Coma, Universidad Autónoma de Barcelona, España
Miguel Soriano Ibáñez, Universidad Politécnica de Cataluña, España
Horacio Tapia Recillas, Universidad Autónoma Metropolitana, México
Routo Terada, Universidade de São Paulo, Brasil
Alfredo Viola, Universidad de la República, Uruguay
Horst von Brand, Universidad Técnica Federico Santa María, Chile

Organización Local

Alejandro Blanco, Universidad de la República, Uruguay
Eduardo Carozo, ANTEL, Uruguay
Carlos Luna, Universidad de la República, Uruguay
Marcelo Rodríguez, Universidad de la República, Uruguay
Leonardo Vidal, Universidad de la República, Uruguay
Felipe Zipitría, Universidad de la República, Uruguay

Revisores Externos

Joao Afonso Abrunhosa, Universidade de Lisboa, Portugal
Almudena Alcaide, Universidad Carlos III de Madrid, España
Jorge Blasco Alis, Universidad Carlos III de Madrid, España
Philippe Camacho, Universidad de Chile, Chile
Eduardo Carozo, ANTEL, Uruguay
Eduardo Cota, Universidad de la República, Uruguay
Tiago Cruz, Universidade de Coimbra, Portugal
Eduardo Giménez, Universidad de la República, Uruguay
Jorge Granjal, Universidade de Coimbra, Portugal
Daniel Hedin, Chalmers University of Technology, Suecia
Julio César Hernandez-Castro, University of Southampton, Inglaterra
Jesús Manjón, Universitat Rovira i Virgili, España
Mireya Morales, Universidad Simón Bolivar, Venezuela
Gloria Pujol, Universitat Rovira i Virgili, España
Alejandro Russo, Chalmers University of Technology, Suecia
Miguel Torrealba, Universidad Simón Bolivar, Venezuela
Rolando Trujillo, Universitat Rovira i Virgili, España
Alexandre Viejo, Universitat Rovira i Virgili, España
Arnau Vives, Universitat Rovira i Virgili, España

Tabla de Contenidos

iPhone 3G: Un nuevo reto para la informática forense 1
Andrea Ariza, Juan Ruíz, Jeimy Cano

Uso del DNIe para reforzar el anonimato en el voto telemático mediante tarjetas inteligentes 16
Emilia Pérez Belleboni, Justo Carracedo Gallardo

Command dimension reduction in masquerader detection 31
Carlos Benitez, Pablo Fierens

A Survey on Masquerader Detection Approaches 46
Maximiliano Bertacchini, Pablo Fierens

Propostas para apoiar a preservação documental de longo prazo na ICP-Brasil . . 61
Viviane Bertol, Rafael Timóteo de Sousa Júnior, Ricardo Custodio

Generación de ambientes para entrenamiento en seguridad informática 73
Alejandro Blanco, Juan Diego Campo, Lucía Escanellas, Carlos Pintado, Marcelo Rodríguez

Robust Declassification for Bytecode 88
Eduardo Bonelli, Francisco Bavera

Técnicas anti-forenses en informática: ingeniería reversa aplicada a TimeStomp . 103
Armando Botero, Ivan Camero, Jeimy Cano

Aplicar el modelo de amenazas para incluir la seguridad en el modelado de sistemas 118
Marta Castellaro, Susana Romaniz, Juan Carlos Ramos, Carlos Feck, Ivana Gaspoz

Sobre el número de funciones bent obtenidas a partir de funciones de máximo peso 133
Joan-Josep Climent, Francisco J. García, Verónica Requena

Um IDS Cooperativo para Redes de Acesso de Banda Larga 148
Tiago Cruz, Thiago Leite, Patricio Baptista, Rui Vilão, Paulo Simões, Fernando Bastos, Edmundo Monteiro

An Extended Reference Monitor for Security and Safety 163
Eduardo B. Fernandez, Michael VanHilst, David laRed Martinez, Sergio Mujica

Detección y limitaciones de ataques clásicos con Honeynets virtuales 173
Hugo Fernández, Jorge Sznek, Eduardo Grosclaude

Un modelo compacto de criptografía asimétrica empleando anillos no conmutativos 188
Juan Pedro Hecht

Watermarking in the encrypted Domain 202
Jordi Herrera-Joancomartí, David Megías

La protección de datos y el diseño de tratamientos de datos personales. Especificaciones funcionales necesarias 213
Angel Igualada Menor

Una marca de agua inteligente aplicada al dinero electrónico 225
Patricia Jaimes, Gabriel Hermosillo, Gomez Roberto

Aumento de la fiabilidad de la evidencia en un protocolo de intercambio justo mediante la división del entorno de firma 240
Jorge L. Hernandez-Ardieta, Ana I. Gonzalez-Tablas, Benjamin Ramos Álvarez, Arturo Ribagorda Garnacho

Análisis formal del estándar NIST para modelos RBAC 255
Carlos Luna, Cristian Rosa

FACTOIDS: Modelos y Herramientas para el Análisis e Intercambio Seguro de Datos Colectados por Sensores 270
Carlos Martinez-Cagnazzo

Autorización de Acceso en MIDP 3.0 283
Gustavo Mazeikis, Carlos Luna

Diseño básico de la seguridad para un servicio sacional de salud pública en Venezuela 298
Mireya Morales, Emilio Hernández

Estegoanálisis aplicado a la generación automática de estegotextos en lengua española. 310
Alfonso Muñoz Muñoz, Justo Carracedo Gallardo

Metodología de implantación de un SGSI en grupos empresariales de relación jerárquica 325
Gustavo Pallas, María Eugenia Corti

Verificación formal de la equidad de un protocolo de firma de contratos mediante Colored Petri Nets 340
M. Magdalena Payeras-Capellà, Macia Mut-Puigserver, Andreu Pere Isern-Deyà, Josep L. Ferrer-Gomila, Llorenç Huguet-Rotger

Criptoanálisis del generador Auto-Shrinking: Una propuesta práctica 355
María Eugenia Pazo Robles, Amparo Fuster Sabater

Diseño e implementación de una función hash basada en caos 368
César José Ramírez López, Leobardo Hernández Audelo

Gestión Automatizada de Requisitos de seguridad para proyectos de desarrollo de líneas de producto Software 383
Jesus Rodriguez, Daniel Mellado, Eduardo Fernandez-Medina, Mario Piattini

SLSB: improving the steganographic algorithm LSB 398
Juan José Roque, Jesús María Minguet

Hacia una arquitectura de servicios de seguridad para entornos Grid móviles 409
David G. Rosado, Eduardo Fernandez-Medina, Javier Lopez

Gestión de identidad en las administraciones públicas: Interoperabilidad pan-Europea 423
Sergio Sánchez, Ana Gómez

MGSM-PYME: Metodología para la gestión de la seguridad y su madurez en las PYMES 437
Luis Enrique Sánchez Crespo, Daniel Villafranca Alberca, Eduardo Fernández-Medina Patón, Mario Gerardo Piattini Velthuis

Metodología para la selección de métricas en la construcción de un cuadro de mando integral 452
Daniel Villafranca Alberca, Luis Enrique Sánchez Crespo, Eduardo Fernández-Medina Patón, Mario Piattini

Towards Secure Distributed Computations 467
Felipe Zipitría

Protocolo de creación de evidencias en entornos vehiculares 482
José María de Fuentes, Ana Isabel González-Tablas, Arturo Ribagorda, Benjamín Ramos

Control de calidad en imágenes de iris mediante razonamiento ontológico 497
Alberto de Santos Sierra, Javier Guerra Casanova, Carmen Sánchez Ávila, Vicente Jara Vera

Una propuesta de arquitectura biométrica de control de acceso basada en ontologías 509
Alberto de Santos Sierra, Javier Guerra Casanova, Carmen Sánchez Ávila, Vicente Jara Vera

Índice de Autores 520

Gestión Automatizada de Requisitos de Seguridad para Proyectos de Desarrollo de Líneas de Producto Software

Jesús Rodríguez[1], Daniel Mellado[2], Eduardo Fernández-Medina[1] y Mario Piattini[1]

[1] Universidad de Castilla – La Mancha, Departamento de Tecnologías y Sistemas de Información, Grupo de Investigación Alarcos, Universidad de Castilla – La Mancha, Paseo de la Universidad 4, 13071 Ciudad Real, España. (+34 925268800)
{Jesus.RValencia, Eduardo.FdezMedina, Mario.Piattini}@uclm.es

[2] Agencia Española de Administración Tributaria, Equipo de Auditoría Informática, Paseo de la Castellana, 108, 28046 Madrid, España.
Daniel.Mellado@uclm.es

Resumen. Tanto la seguridad como la ingeniería de requisitos son factores clave para el éxito en el desarrollo de una línea de productos software, debido a que la compleja estructura de dependencias entre las características de la línea de productos y de cada producto en particular puede verse amenazada si se encuentra una debilidad de seguridad en las características de la línea, pudiéndose propagar este problema a todos los productos de la misma. La aplicación de procesos de ingeniería de requisitos de seguridad o metodologías para definir requisitos de seguridad que mitiguen las posibles amenazas sobre una línea se convierte en una tarea muy difícil debido a que sería necesario realizarla manualmente. Por lo tanto, en este artículo presentamos una herramienta llamada SREPPLineTool que proporciona soporte automatizado, guiado e intuitivo para facilitar la aplicación del proceso de ingeniería de requisitos de seguridad para líneas de producto software que hemos desarrollado, denominado SREPPLine.

Palabras clave: requisitos de seguridad, línea de producto, ingeniería de requisitos de seguridad, Criterios Comunes, seguridad.

1 Introducción

En los últimos años se han observado una gran cantidad de organizaciones que se han visto comprometidas debido a la aparición de brechas de seguridad. De hecho, el número de vulnerabilidades que han aparecido en aplicaciones ha aumentado de 171 en 1995 a 7.236 en 2007 [1], esto está motivado en parte por la tendencia hacia sistemas de mayor envergadura que se distribuyen a través de Internet introduciendo de esta manera un gran conjunto de amenazas de seguridad que podrían manifestarse [2], lo que implica que hoy en día los sistemas de información son susceptibles de verse amenazados por cyber-ataques, usuarios maliciosos, ciber-terroristas, etc. [3].

Las Líneas de Productos Software (LPS) se han convertido en el enfoque más exitoso en el campo de la reutilización de componentes, reduciendo de un modo significativo tanto el coste de desarrollo como el tiempo de comercialización. En los sistemas de software complejos como las LPS, la seguridad es una preocupación transversal que debe ser objeto de un cuidadoso análisis y de aplicación de técnicas de ingeniería de requisitos adaptadas a la problemática de las líneas de producto [4].

Debido a esto, la seguridad en el software se está convirtiendo en un factor de gran interés para los ingenieros software [5]. Esto tiene como consecuencia que la disciplina de Ingeniería de Requisitos de Seguridad se considere una parte importante de la Ingeniería de Seguridad aplicada al proceso de desarrollo de sistemas de información, sin embargo hasta hace poco tiempo no se le había prestado mucha atención [6]. Esta disciplina aporta técnicas, métodos y estándares claves para conseguir productos y LPS seguras, además de asegurar que se cumplen los requisitos de seguridad definidos y las propiedades de seguridad establecidas en el modelo de variabilidad de las LPS a lo largo de todo el ciclo de vida.

A pesar de lo importante que resultan las disciplinas anteriores para la gestión de la seguridad en las LPS, las metodologías de ingeniería del software y las propuestas de estándares de ingeniería de LPS han ignorando tradicionalmente tanto los requisitos de seguridad como las cuestiones más específicas como la gestión de la variabilidad de seguridad, y aunque existen varios trabajos relacionados con procesos y herramientas orientadas a la gestión de requisitos de seguridad, ninguno de ellos da cobertura al paradigma de desarrollo de LPS, principalmente porque no tienen en cuenta la gestión de la variabilidad de los requisitos de seguridad.

En este artículo, describimos el prototipo de una herramienta de gestión de requisitos de seguridad para LPS que hemos desarrollado para proporcionar soporte automatizado al proceso SREPPLine (Proceso de Ingeniería de Requisitos de Seguridad para Líneas de Producto Software) [7], llamada SREPPLineTool. SREPPLineTool permite ejecutar el proceso SREPPLine de una forma sistemática, guiada e intuitiva. También facilita la integración con los Criterios Comunes (CC) [8] y con el estándar ISO/IEC 27001 [9] dentro del proceso de desarrollo software de los productos de una LPS, así como contribuye al cumplimiento del estándar IEEE 830:1998 [10]. Para conseguir esto, la herramienta se ayuda de las funcionalidades en cuanto a gestión de requisitos ofrecidas por 'IBM Rational RequisitePro' (herramienta CARE que extiende SREPPLineTool).

Además, la herramienta ayuda a desarrollar LPS y productos conforme a los estándares de seguridad más relevantes relacionados con la gestión de requisitos de seguridad de una forma asistida y haciendo de esta forma que no sea imprescindible tener un conocimiento absoluto de los estándares y reduciendo así la participación de expertos en seguridad, es decir, la herramienta mejora la eficiencia de SREPPLine. Asimismo, y gracias al modelo de referencia de seguridad implementado en SREPPLineTool, la gestión y visualización de los artefactos de variabilidad y su

trazabilidad se convierte en una tarea mucho más sencilla e intuitiva, mejorando así la calidad sucesivamente.

El resto del artículo está organizado de la siguiente forma: En la sección 2, se resumen algunas de las características básicas del proceso SREPPLine con el objetivo de que se entienda la posterior explicación de la herramienta. En la sección 3, introducimos las características de la herramienta, se explica detenidamente la funcionalidad de la herramienta aplicada sobre un caso de estudio que nos sirve para una validación inicial de la misma y expondremos las lecciones aprendidas. En la sección 4 presentamos el trabajo relacionado. Y finalmente, en la última sección comentaremos el trabajo futuro y las conclusiones.

2 Visión general de SREPPLine: Ingeniería de Requisitos de Seguridad para Líneas de Productos Software

Una línea de productos software es un sistema intensivo de software que comparte un conjunto común gestionado de características que satisfacen unas necesidades especificas de un segmento particular del mercado y que son desarrollados de una forma pre-establecida a partir de un conjunto común de componentes [11].

El paradigma de la ingeniería del software en líneas de producto diferencia dos procesos: la ingeniería del dominio, que se encarga de definir las partes comunes y variables de una línea; y la ingeniería de la aplicación, que es el proceso de la ingeniería de las LPS en el que se desarrollan los productos de la línea reutilizando artefactos del dominio y explotando la potencia de la variabilidad de la línea de productos [12].

El proceso SREPPLine [7] es un add-in de actividades (las cuales se muestran en la Fig. 1 usando la notación de SPEM 2.0 [13]), que se integran sobre el proceso de desarrollo de LPS existente en una organización, proporcionándole un enfoque en ingeniería de requisitos de seguridad específico para LPS. En este proceso hemos definido las actividades clave que deben ser parte de cada desarrollo de LPS. El orden en el que deben ser realizadas estas actividades depende del proceso particular que este establecido en cada organización. En concreto, proceso se integra dentro del marco de trabajo propuesto por Pohl et al. en [12], y está compuesto de dos subprocesos con sus respectivas actividades: el subproceso de Ingeniería de Requisitos de Seguridad del Dominio de las LPS (PLSecDomReq) y el subproceso de Ingeniería de Requisitos de Seguridad de la Aplicación de las LPS (PLSecAppReq). Aunque podría integrarse en otros procesos de desarrollo con las correspondientes adaptaciones puntuales en ciertas tareas concretas de las actividades que lo componen.

Es un proceso basado en características y metas de seguridad, dirigido por el riesgo y los estándares de seguridad (más concretamente por la norma ISO/IEC 27001[9] y los Criterios Comunes [8]), que se ocupa de la elicitación de requisitos de seguridad y de la gestión de los artefactos relacionados con dichos requisitos, desde las primeras

etapas del desarrollo de las LPS de una forma intuitiva y sistemática especialmente adaptada para el desarrollo basado en el paradigma de las LPS. SREPPLine utiliza las últimas técnicas ampliamente extendidas para elicitar y modelar requisitos de seguridad, como casos de uso de seguridad [14] o casos de mal uso [2] ,además de la integración en el ciclo de vida de las LPS de los componentes pertenecientes a los Criterios Comunes (CC) (ISO/IEC 15408) y los controles descritos en el estándar ISO/IEC 27001, facilitando así la certificación de seguridad de los productos desarrollados en una LPS determinada.

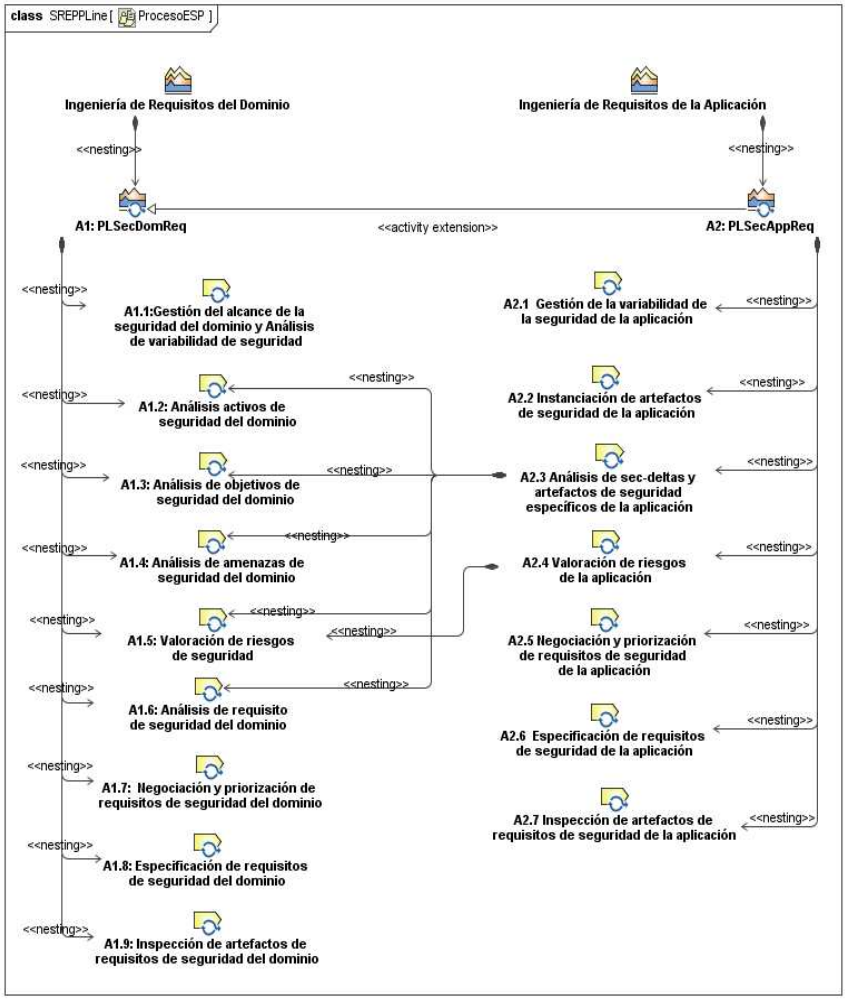

Fig. 1 Especificación de las actividades de SREPPLine usando SPEM 2.0

Además, el proceso SREPPLine sugiere el uso de un método para llevar a cabo la evaluación del riesgo que se ajuste al estándar ISO/IEC 13335 [15], concretamente para la evaluación del riesgo tanto de la línea como de los productos de una línea actualmente SREPPLine y SREPPLineTool utilizan la metodología Magerit [16] (que es conforme a dicho estándar y es una metodología oficialmente reconocida y utilizada en el ámbito de sistemas de información de la OTAN [17] y por la OCDE [18]).

También, como se menciono anteriormente, SREPPLine tiene el objetivo de minimizar el conocimiento necesario de estándares de seguridad así como la participación de expertos de seguridad durante el desarrollo. Para este fin, se propone el uso de un Repositorio de Activos de Seguridad facilitando así la reutilización de artefactos y la implementación del Meta Modelo de Referencia de Seguridad, el cual está compuesto por el Sub-Meta Modelo de Variabilidad de Seguridad y el Sub-Meta Modelo de Decisión de Requisitos de Seguridad, que dan soporte a la gestión de la variabilidad y trazabilidad de los artefactos de seguridad relacionados con las LPS y sus productos, como se detalla en SREPPLine [19].

Este meta modelo, el cual se explica en detalle en [19], es la base a través de la cual SREPPLine define, representa, y almacena el conocimiento sobre todos los artefactos de seguridad de una LPS, ayudando así a una posible certificación contra los estándares de seguridad más extendidos relacionados con la Ingeniería del Software. En esencia, estos modelos representan un repositorio de conocimiento estructurado para dar soporte a los requisitos de seguridad en la ingeniería de requisitos de la LPS.

3 SREPPLineTool

Para dar soporte automatizado a las actividades del proceso SREPPLine, hemos desarrollado una herramienta prototipo CARE (Computer Aided Requirements Engineering), llamada SREPPLineTool que nos permite aplicar el proceso en escenarios de aplicación reales donde el número de artefactos es elevado y la complejidad de su gestión manual se complica. La aplicación de SREPPLine nos permitió obtener experiencia para refinar el proceso y la herramienta, y así mejorar la calidad sucesivamente. La herramienta implementa el Meta Modelo de Referencia de Seguridad (explicado en [19]) por medio de los repositorios dinámicos de artefactos de seguridad, y guiando la ejecución del proceso de forma secuencial. Pudiendo así proponer artefactos de seguridad relacionados con cada actividad del proceso SREPPLine dependiendo de las categorías de artefactos del dominio de un proyecto de desarrollo de LPS.

Además, la forma de interacción con SREPPLineTool es guiada de una forma intuitiva, llevando la ejecución de una etapa a la siguiente, manteniendo en todo momento una representación visual de los artefactos que se manejan en dicha etapa y permitiendo de esta manera una gestión sencilla tanto de la variabilidad como de la trazabilidad entre los distintos artefactos de seguridad que se tratan en cada etapa. Así mismo, es posible generar documentos de seguridad sobre un proyecto de LPS y

exportar información sobre los modelos variabilidad en XML, más adelante se explicará en detalle los aspectos relativos a la funcionalidad aplicada a un caso de estudio representativo.

3.1 Desarrollo de la herramienta

Este prototipo se ha desarrollado utilizando la tecnología .NET de Microsoft y esta implementado en el lenguaje C#, además trabaja contra una base de datos Microsoft SQL Server 2005 y está vinculado con la herramienta de gestión de requisitos IBM Rational RequisitePro a través de una interfaz implementada en Visual Basic.NET como se describe en la Fig. 2 , pudiéndonos así servir de la popularidad de esta herramienta expandida en el mundo empresarial en las tareas de elicitación y gestión de requisitos en sistemas informáticos, permitiéndonos importar y exportar requisitos desde y hacia ella respectivamente.

Fig. 2. Arquitectura de SREPPLineTool.

3.2 Aplicando SREPPLineTool

En esta sección, describiremos como se puede utilizar SREPPLineTool en la práctica para automatizar el proceso SREPPLine, para ello aplicaremos SREPPLineTool sobre un caso de aplicación representativo en el que desarrollaremos los requisitos de seguridad para una LPS de una Administración Pública española encargada del servicio online de pago de tasas de examen. Esta LPS posee un conjunto de configuraciones diferentes para cada institución pública distinta, así como un conjunto común de funcionalidad que conforma el núcleo común de todos los sistemas, y además cuenta con un conjunto variable de parámetros de configuración y de requisitos no funcionales. Por lo tanto, la LPS de servicios online de pago de tasas de examen es una LPS cuyos productos varían en función de la configuración del sistema y de los servicios de negocio online, pero manteniendo todas las funcionalidades proporcionadas por el núcleo común. Se había realizado un caso de estudio del proceso SREPPLine en [7], pero sin utilizar ningún tipo de herramienta que diese soporte al proceso.

Este ejemplo se concentra en la aplicación de SREPPLineTool sobre la ingeniería del dominio para obtener todos los artefactos de seguridad de la LPS propuestos en el

subproceso PLSecDomReq de SREPPLine. Con el objetivo de facilitar la compresión del ejemplo se ha resumido y simplificado, destacando los puntos más importantes en cuanto al soporte automatizado que ofrece la herramienta.

Antes de comenzar la ejecución de SREPPLineTool, las características más importantes para nuestra LPS a la que llamaremos para abreviar "SOPE" deben ser identificadas y registradas en la herramienta IBM Rational RequisitePro. A continuación describiremos cada actividad (ordenadas en pestañas) de SREPPLineTool que coincide secuencialmente con cada etapa del proceso SREPPLine.

Actividad 1: Alcance de la Gestión de la Seguridad y Análisis de Variabilidad

Esta primera actividad se divide en varias sub-actividades que se describirán a continuación:

En la primera sub-actividad se realiza la actualización del repositorio, se introducen manualmente los artefactos de seguridad ya definidos en la LPS (que se desarrollaron sin usar SREPPLineTool) y además se recibió una petición de uno de los responsables de la LPS para importar un conjunto de características desde IBM Rational RequisitePro. Entonces, el ingeniero de Requisitos de Seguridad buscó en el Repositorio de Recursos de Seguridad de la herramienta para elicitar y proponer características de seguridad para la LPS, el sugirió relacionar las características de seguridad: Envíos Seguros y Autenticidad de Usuarios como características de seguridad variantes de la LPS SOPE.

La siguiente sub-actividad es la Selección del Personal Responsable, donde el director de desarrollo de la LPS selecciona los usuarios y roles con los que dichos usuarios participaran en el proyecto dentro de la herramienta.

La tercera sub-actividad es la Identificación de la Variabilidad, como se observa en la Fig. 3, SREPPLineTool nos permite representar en un árbol de variabilidad de características el modelo de variabilidad de la LPS previamente diseñado. Además, se observa la estructura de características de seguridad que se definieron en la primera sub-etapa, de esta manera en esta sub-actividad se diseña el modelo de variabilidad de seguridad, el cual se puede exportar a un fichero XML para que sea usado por otras herramientas.

La última sub-actividad es la de Acuerdos de Definiciones donde SREPPLineTool nos da soporte para conseguir acuerdos sobre un conjunto común de definiciones de seguridad como: Seguridad de la Información, Amenazas, Confidencialidad, etc., proporcionando las definiciones de estos conceptos de acuerdo con los estándares ISO/IEC 27002 [20] y ISO/IEC 27001. Además, SREPPLineTool nos permite definir nuevos estándares así como conceptos, pudiéndose así almacenar en el repositorio y también nos posibilita definir el estado de evaluación del nivel de aseguramiento (EAL) de los Criterios Comunes (CC), para nuestra línea SOPE se eligió el nivel EAL-2 de los CC.

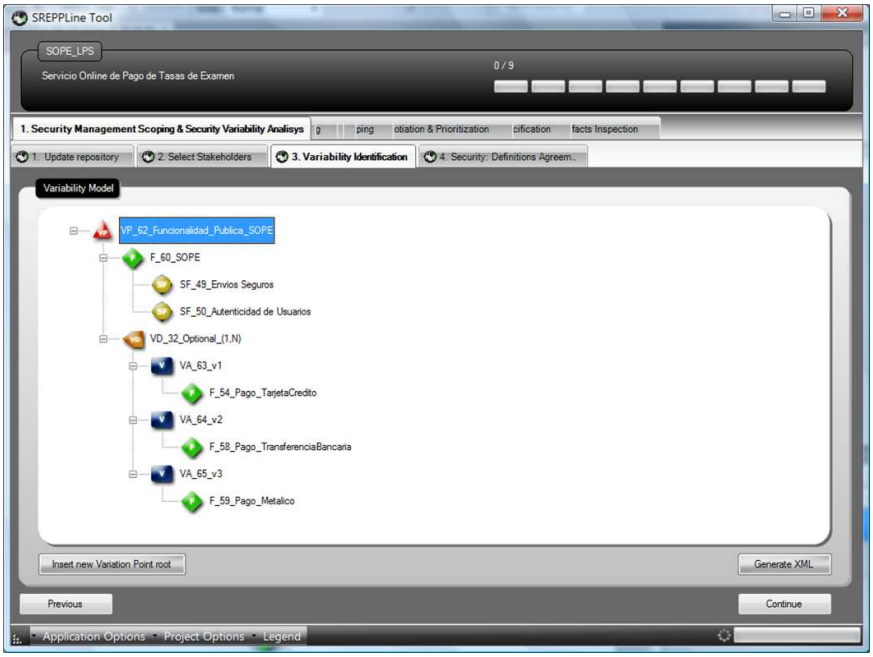

Fig. 3. Identificación de la Variabilidad.

Actividad 2: Alcance de los Activos de Seguridad

En esta actividad, identificamos los activos de seguridad para cada característica de seguridad y las dependencias entre dichos activos. Por ejemplo, para la característica de seguridad "Autenticidad de Usuarios" identificamos una serie de activos de seguridad, los cuales fueron propuestos por la herramienta al estar previamente clasificados en el repositorio como pertenecientes al dominio de componentes online de la administración, los activos fueron los siguientes: Contraseña de Usuario, Certificado Electrónico, Numero Nacional de identidad/Numero de Cedula. Además, la herramienta nos permite dar un valor numérico de 0 a 10, el cual describe como de importante o crítico es el activo, activos con valores altos representan una mayor importancia y degradación del producto si se llegase a manifestar una amenaza que afectase ha dicho activo.

En nuestro ejemplo de aplicación representativo, la característica de seguridad "Autenticidad de Usuarios" contiene información critica, y por tanto los activos de seguridad que representan la información fueron valorados de la siguiente forma (información de la Contraseña de Usuario, valor = 9; Información del Certificado, valor = 9; Información del Numero Nacional de Identidad, valor = 6). Además, SREPPLineTool nos permite definir jerarquías de dependencias entre activos, de esta forma si un activo depende de otro que tiene una mayor criticidad el valor de dicho

activo se propagará en el otro activo, de esta forma se crea una estructura en árbol en el que el valor de los activos se propaga del activo raíz a las hojas.

Actividad 3: Alcance de los Objetivos de Seguridad.

Seleccionamos los objetivos de seguridad para cada activo; la herramienta nos muestra los objetivos de seguridad y las relaciones actuales entre activos y objetivos de seguridad junto con el valor de cada par (activo, objetivo de seguridad), el cual representa cómo de importante es el cumplimiento de dicho objetivo para proteger ese activo, este valor definido en Magerit [16] varía de 0 (mínimo) a 10 (máximo).

En nuestro caso de estudio, se identificaron los siguientes objetivos de seguridad: integridad, confidencialidad, disponibilidad, autenticidad del origen de los datos, trazabilidad del uso del servicio y trazabilidad del acceso a los datos. Estos objetivos de seguridad se añadieron para los activos: Contraseña de Usuario, Certificado Electrónico y Numero Nacional de Identidad.

Actividad 4: Alcance de las Amenazas de Seguridad.

Esta actividad es la responsable de especificar y relacionar los pares de activo-objetivo de seguridad con las posibles amenazas que son susceptibles de crear brechas de seguridad en nuestra LPS. El repositorio de recursos de seguridad de SREPPLineTool nos permite seleccionar las amenazas desde las siguientes fuentes: amenazas previamente introducidas en el repositorio en este proyecto de LPS o en otros proyectos, Objetivos de Control de la ISO/IEC 27001 y Familias de los Criterios Comunes (ISO/IEC 15408). Cuando una nueva amenaza es creada, podemos especificar un conjunto de casos de mal uso y/o árboles de ataque [2] para definir el comportamiento de una amenaza cuando se manifiesta sobre un par activo-objetivo de seguridad.

Sobre el caso de estudio en el que nos centramos, SREPPLineTool nos permitió definir y relacionar las amenazas con los pares activo-objetivo de seguridad de la LPS, de modo que identificamos las siguientes amenazas que podrían manifestarse poniendo en peligro la LPS SOPE:

- Amenaza 1: Manipulación de la configuración.
- Amenaza 2: Enmascaramiento de la identidad del usuario.
- Amenaza 3: Modificación de los datos.
- Amenaza 4: Escuchas.
- Amenaza 5: Acceso no autorizado.

Actividad 5: Evaluación de riesgos de seguridad.

Una vez las amenazas han sido identificadas, se llevó a cabo la evaluación de requisitos (Fig. 4). Para realizar esta tarea, SREPPLineTool usa una técnica propuesta en Magerit [16] (técnicas reconocidas oficialmente por la OTAN [18]) que se basa en

el uso de un análisis cuantitativo. Primeramente, y con la ayuda de los responsables de la LPS, por cada par activo-objetivo de seguridad, se estimó la frecuencia con las que las amenazas podrían manifestarse (siendo los valores posibles: 0,1; 1; 10; 100. Así como se estimó la degradación (expresada como un porcentaje) del valor de un activo causada por una amenaza que se ha manifestado. Finalmente, con estos datos la herramienta calcula automáticamente el impacto y el riesgo para cada tupla activo-objetivo de seguridad-amenaza, en donde los valores altos indican un mayor impacto y riesgo.

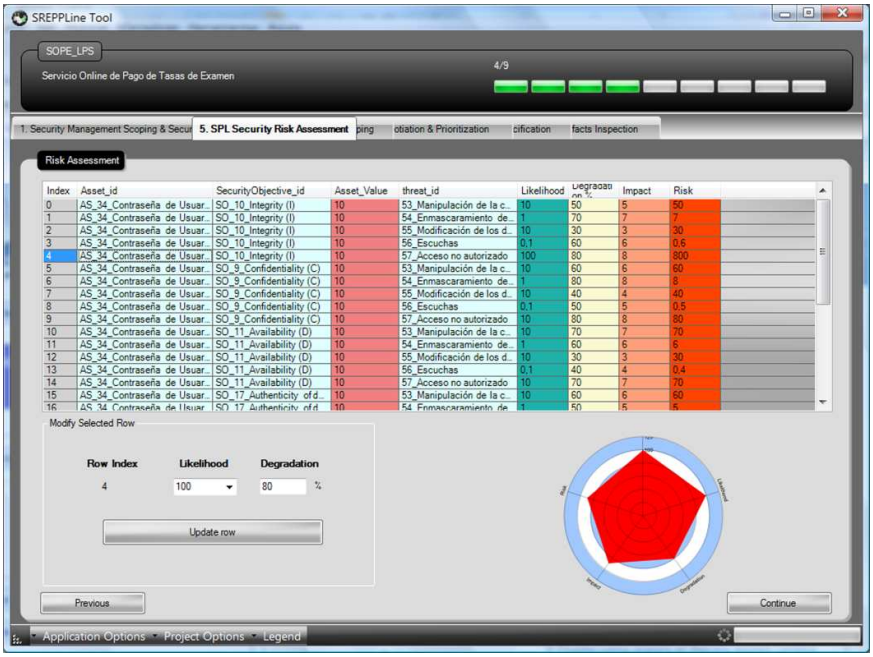

Fig. 4. Actividad 5 SREPPLineTool.

Actividad 6: Alcance de los Requisitos de Seguridad.

En esta actividad se definieron los requisitos o paquetes de requisitos de seguridad para cada tupla de [activo-objetivo]-amenaza en función del riesgo obtenido en la etapa anterior, para así mitigar las posibles amenazas en el caso de que se manifiesten. Por lo tanto, una vez fueron seleccionadas las amenazas, se elicitaron aquellos requisitos de seguridad que los responsables de la línea decidieron que eran necesarios. Para facilitar esta tarea SREPPLineTool nos permite:

- Seleccionar los requisitos de seguridad introducidos previamente en el repositorio de la herramienta ya sea en el proyecto de LPS actual o en otros proyectos o creando un requisito nuevo.

- Seleccionar los requisitos de seguridad desde los controles de la ISO/IEC 27001 o desde los componentes de los Criterios Comunes.
- Seleccionar o crear nuevos paquetes o test de seguridad.

Sobre nuestro caso de aplicación de SREPPLineTool se añadieron a nuestra LPS aquellos requisitos de seguridad que nos comunicó el experto en ingeniería de requisitos de seguridad que serian importantes para mitigar las amenazas definidas previamente.

A continuación describimos un requisito de seguridad que seleccionamos del repositorio de recursos de seguridad y que está relacionado con la amenaza 2 "Enmascaramiento de la identidad del usuario":

SR1: Aseguramiento de la Autenticidad del Usuario.

- Las funciones seguridad de [PV_LPS_dom] deberán identificar y autenticar a los [PV_tipo_usuario] usando [Variante] antes que un [PV_tipo_usuario] pueda conectarse al sistema [PV_LPS_dom]. (Variante = [Contraseña de Usuario, Certificado Electrónico]).

Además, SREPPLineTool nos permite relacionar los requisitos de seguridad con los requisitos funcionales y no funcionales de la LPS. También, facilita la especificación de requisitos de seguridad teniendo la posibilidad de especificar casos de uso de seguridad y árboles de ataque.

Actividad 7: Negociación y Priorización de los Requisitos de Seguridad.

En esta actividad se da prioridad a los requisitos de seguridad conforme al riesgo que poseen las amenazas que pretende mitigar y a las dependencias con otros requisitos funcionales y no funcionales. Para cada uno de los requisitos de seguridad establecidos en un nuestra LPS, seleccionamos el nivel de prioridad (de 0 a 10), y con esa información SREPPLineTool nos ordenará la lista de requisitos de seguridad desde la prioridad más alta a la más baja.

Actividad 8: Especificación de Requisitos de Seguridad.

En este punto se abordó la tarea de realizar la especificación de los requisitos de seguridad. Para llevarla a cabo, SREPPLineTool nos proporciona plantillas XML parametrizadas para definir la variabilidad en los requisitos de seguridad y sus relaciones con el modelo de variabilidad.

Actividad 9: Especificación de Requisitos de Seguridad.

En esta actividad, SREPPLineTool da soporte facilitando la tarea de verificación de que los requisitos de seguridad sean conformes al IEEE 830 [10] y al estándar ISO/IEC 15408 (Criterios Comunes), siendo más sencillo para los responsables de la

LPS el poder verificar y validar los requisitos de seguridad comprobando aquellas amenazas para las que no se han especificados requisitos que las mitiguen en el proyecto actual de LPS.

Por último, en esta actividad, la herramienta genera el documento de perfil de protección para la LPS conforme a los Criterios Comunes (ISO/IEC 15048), que integra toda la información relacionada con el resto de artefactos generados por SREPPLineTool en las actividades anteriores. Finalmente, SREPPLineTool nos permite seleccionar aquellos artefactos de seguridad modificados o generados en esta iteración del proyecto, para que sean añadidos al repositorio general de la herramienta, pudiendo ser así reutilizados en otras iteraciones de la LPS o en otros proyectos de LPS.

3.3 Lecciones Aprendidas

De entre las lecciones más importantes que hemos aprendido utilizando en la práctica SREPPLineTool sobre este caso de estudio, cabe destacar las siguientes:

- El soporte que ofrece la herramienta es fundamental para aplicar el proceso SREPPLine sobre sistemas software de gran envergadura y concretamente sobre el caso de estudio en el que nos hemos centrado, debido al gran número de artefactos que pueden llegar a manejarse y a la complejidad de las relaciones entre dichos artefactos, de tal forma que sin SREPPLineTool hubiera sido complicada la gestión de la trazabilidad y variabilidad de la LPS y hubiera el desarrollo hubiera necesitado de mayor esfuerzo y tiempo.
- Se echó en falta una mayor versatilidad en cuanto a la integración con otras herramientas relacionadas con el paradigma de desarrollo de LPS para obtener un trazabilidad óptima de los artefactos relacionados con los requisitos de seguridad, y para conseguir una implementación apropiada de la ingeniería de requisitos de seguridad dentro de una organización.
- Para la organización en la que se utilizó la herramienta, se obtuvo como mejora el hecho de tener un proceso (SREPPLine) sistemático y especifico para gestionar requisitos de seguridad en su LPS que cuenta con un soporte automatizado (SREPPLineTool) que facilita su aplicación, y además siendo conforme con los estándares ISO/IEC 15408 e ISO/IEC 27001, así como se implementó un repositorio de activos comunes que podían ser reutilizados en el desarrollo de las aplicaciones de la línea o en desarrollos futuros de nuevas LPS en la organización.

4 Trabajos Relacionados

En los últimos años se ha llevado a cabo una amplia labor de estudio sobre los requisitos de seguridad, tal y como se presentó en [21, 22], además de algunos trabajos que se ocupan de las herramientas de gestión de requisitos de seguridad, aunque ninguna de las herramientas existentes de ingeniería de requisitos hasta el momento da un soporte suficientemente especifico o adaptado al paradigma de

desarrollo basado en LPS, principalmente porque no se ocupan de la variabilidad de los requisitos de seguridad, el cual es un aspecto esencial de este paradigma, y que necesita de una automatización que facilite la gestión de las relaciones de trazabilidad y variabilidad de los numerosos artefactos que se generan.

Actualmente, en cuanto a los requisitos de seguridad se refiere, destacan una serie de herramientas que a continuación vamos a describir brevemente, y que además están relacionadas con SREPPLineTool en lo relativo a la gestión de requisitos de seguridad.

La herramienta SirenTool es un complemento de IBM RequisitePro que da soporte al método SIREN [23], que es un método para elicitar y especificar los requisitos software y de seguridad de un sistema, además incluye un repositorio de requisitos de seguridad basado en los artefactos de Magerit y que puede ser estructurado de acuerdo a dominios y perfiles de una forma parecida a las categorías que implementa SREPPLineTool. No obstante, SirenTool solo reutiliza requisitos basados en los activos de Magerit almacenados en el propio repositorio de la herramienta. Sin embargo, nuestra herramienta trabaja sobre líneas de producto software, que junto con el modelo de referencia de seguridad que implementa nos permite reutilizar requisitos de seguridad, amenazas, características de seguridad, objetivos de seguridad, activos, contramedidas y pruebas. Además, gracias a este modelo, la variabilidad de seguridad puede ser gestionada en el nivel de requisitos en lugar de en el nivel de diseño.

ST-Tool [24] es una herramienta CASE desarrollada para la modelización y análisis funcional, y para el diseño y modelización de requisitos de seguridad. STTool ha sido diseñada para dar soporte a la metodología Secure Tropos [24], y consiste en una herramienta de desarrollo de software orientado a agentes, que maneja el concepto de actor, servicio y relación. En contraste con SREPPLineTool, ST-Tool no trata la reutilización de artefactos de seguridad, tampoco incorpora una integración con los estándares de seguridad (como ISO/IEC 15408 o ISO/IEC 27001) y no facilita la generación de informes.

La herramienta UMLsec-Tool soporta UMLsec [25] y proporciona una extensión al proceso convencional de desarrollo dirigido por casos de uso para sistemas de seguridad críticos. Considera los aspectos de seguridad tanto en el modelo de dominio estático como en la especificación funcional. Para la elaboración de los aspectos funcionales introduce un catálogo de cuestiones y en el modelo del dominio introduce una extensión de UML, UMLsec. Sin embargo, la herramienta no facilita la definición de la variabilidad de la seguridad, que es una diferencia fundamental entre el desarrollo de sistemas tradicionales y el de LPS.

5 Conclusiones y Trabajo Futuro

Actualmente, la seguridad en el software está generando un creciente interés y aún más en el desarrollo orientado a LPS, debido a que la gestión de requisitos de

seguridad es extremadamente importante en las LPS porque una debilidad en la seguridad de una línea puede producir problemas a lo largo de todo el ciclo de vida de los productos de dicha línea.

Aunque se han realizado algunos intentos de cubrir el agujero que separa la ingeniería de requisitos de la ingeniería de requisitos para LPS, actualmente no existe una aproximación sistemática o herramienta que soporte la definición de requisitos de seguridad, que gestione la variabilidad de los mismos, de las relaciones con otros artefactos de seguridad de los modelos de LPS. Además, las herramientas tradicionales de gestión de requisitos no son capaces de dar un soporte directo a los requisitos de seguridad que se manejan en la ingeniería de las LPS.

En este artículo, se ha mostrado como una integración completa de los conceptos de ingeniería de requisitos de seguridad y de ingeniería de LPS, junto con las últimas técnicas de especificación de requisitos de seguridad (como los casos de uso de seguridad [14], casos de mal uso y árboles de ataque [2], UMLsec [25]) y junto con los estándares de seguridad más relevantes que tratan la gestión de requisitos de seguridad (tales como ISO/IEC 15408, ISO/IEC 27001,o ISO/IEC 27002), en estas herramientas es posible.

Teniendo lo anterior en cuenta, podemos decir que las herramientas como SREPPLineTool facilitan que la industria acepte e integre técnicas de ingeniería de requisitos de seguridad dentro del desarrollo de sus LPS, este hecho fue ratificado en el caso de estudio real que realizamos en la Seguridad Social de España [7].

Por último, hay una serie de aspectos previstos para el futuro de la herramienta que nos permitirán aumentar el nivel de automatización del proceso SREPPLine y por lo tanto, una mayor eficiencia del proceso de ingeniería de requisitos dentro de la ingeniería de las LPS de una organización. Entre ellos, podemos destacar los siguientes: ampliar los tipos de especificaciones de requisitos soportadas para así dar soporte a UMLSec [25]; desarrollar nuevos módulos de extensibilidad que permitan que SREPPLineTool sea soportada por otras herramientas CARE; y automatizar la creación de casos de uso de seguridad a través de la creación de casos de mal uso.

6 Referencias

[1] CERT/CC, "CERT/CC Statistics 1995-2007," Pittsburgh, 2008.

[2] A. L. Opdahl and G. Sindre, "Experimental comparison of attack trees and misuse cases for security threat identification," *Information and Software Technology. In Press, Corrected Proof*, 2008.

[3] K.-K. R. Choo, R. G. Smith, and R. McCusker, "Future directions in technology-enabled crime: 2007–09," in *Research and Public Policy Series*. vol. 78, Australian_Government, Ed.: Australian Institute of Criminology, 2007.

[4] A. Birk and G. Heller, "Challenges for requirements engineering and management in software product line development," *International Conference on Requirements Engineering (REFSQ 2007)*, pp. 300-305, 2007.

[5] J. Viega and G. McGraw, *Building Secure Software: How to Avoid Security Problems the Right Way*. Boston: Addison-Wesley, 2002.
[6] A. v. Lamsweerde, "Elaborating Security Requirements by Construction of Intentional Anti-Models," *26th International Conference on Software Engineering (ICSE 2004)*, pp. 148-157, 2004.
[7] D. Mellado, E. Fernández-Medina, and M. Piattini, "Towards security requirements management for software product lines: a security domain requirements engineering process," in *Computer Standards & Interfaces*. vol. 30, 2008, pp. 361-371.
[8] ISO/IEC, "ISO/IEC 15408:2005 Information technology - Security techniques - Evaluation criteria for IT security, (Common Criteria v3.0)," 2005.
[9] ISO/IEC, "ISO/IEC 27001 Information technology -- Security techniques -- Information security management systems -- Requirements.," 2005.
[10] IEEE, "IEEE 830: 1998 Recommended Practice for Software Requirements Specifications," 1998.
[11] P. Clements and L. Northrop, *Software Product Lines: Practices and Patterns*: Addison-Wesley, 2002.
[12] K. Pohl, G. Böckle, and F. v. d. Linden, *Software Product Line Engineering. Foundations, Principles and Techniques*. Berlin Heidelberg: Springer, 2005.
[13] OMG, "Software & Systems Process Engineering Meta-Model Specification v.2.0," 2008.
[14] D. G. Firesmith, "Engineering Security Requirements," *Journal of Object Technology*, vol. 2, pp. 53-68, 2003.
[15] ISO/IEC, "ISO/IEC 13335 Information technology - Security techniques - Management of information and communications technology security," 2004.
[16] Ministry_for_Public_Administration_of_Spain, *Methodology for Information Systems Risk Analysis and Management*: Ministry for Public Administration, 2005.
[17] CCN-CERT, "Últimos avances en ciberseguridad (9th NATO cyberdefense workshop)," in *Revista auditoria y Seguridad (www.revista-ays.com)*. vol. nº23-junio, 2008, pp. 70-71.
[18] OECD, "The promotion of a culture of security for information systems and networks in OECD countries," Organisation for Economic Co-operation and Development 2005.
[19] D. Mellado, E. Fernández-Medina, and M. Piattini, "Security Requirements Variability for Software Product Lines," *Symposium on Requirements Engineering for Information Security (SREIS 2008) co-located with ARES 2008*, pp. 1413-1420, 2008.
[20] ISO/IEC, "ISO/IEC 27002:2005 Information technology - Security techniques - Code of practice for information security management," 2007.
[21] D. Mellado, E. Fernández-Medina, and M. Piattini, "A Comparison of the Common Criteria with Proposals of Information Systems Security Requirements," *"First International Conference on Availability, Reliability and Security" (ARES'06)*, pp. 654-661, 2006.
[22] D. Mellado, E. Fernández-Medina, and M. Piattini, "A Systematic Review of Security Requirements Engineering," in *Computers and Security (Being processed - Under review)*, 2008.
[23] A. Toval, J. Nicolás, B. Moros, and F. García, "Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach," in *Requirements Engineering*. vol. 6, 2002, pp. 205-219.
[24] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "ST-Tool: A CASE Tool for Security Requirements Engineering," in *IEEE International Conference on Requirements Engineering (RE'05)*, 2005.
[25] J. Jürjens, "UMLsec: extending UML for secure systems development.," *UML 2002 - The Unified Modeling Language. Model Engineering, Languages,Concepts, and Tools. 5th International Conference.*, vol. LNCS 2460, pp. 412-425, 2002.