PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 10-09

Seguridad Informática en la Universidad de la República

Gustavo Betarte, Alejandro Blanco, Juan D. Campo, María E. Corti, Carlos Luna, Marcelo Rodríguez y Felipe Zipitría

2010

Seguridad informática en la Universidad de la República
Betarte, Gustavo; Blanco, Alejandro; Campo, Juan D.; Corti, María E.; Luna, Carlos; Rodríguez, Marcelo; Zipitría, Felipe.
ISSN 0797-6410
Reporte Técnico RT 10-09
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay, 2010

Seguridad Informática en la Universidad de la República

Gustavo Betarte, Alejandro Blanco, Juan D. Campo, María E. Corti, Carlos Luna, Marcelo Rodríguez y Felipe Zipitría

Grupo de Seguridad Informática
Instituto de Computación, Facultad de Ingeniería
Universidad de la República
Uruguay
{gustun,ablanco,jdcampo,mcorti,cluna,marcelor,fzipi}@fing.edu.uy
www.fing.edu.uy/inco/grupos/gsi.

Resumen Este artículo presenta al Grupo de Seguridad Informática de la Facultad de Ingeniería de la Universidad de la República (Uruguay), creado en 2006. Se describen los trabajos de investigación del grupo y los resultados generados, proveyendo referencias a sus artículos publicados. Se incluye asimismo una breve descripción de la actividad de formación curricular universitaria a cargo del equipo.

1. Introducción

El Grupo de Seguridad Informática (GSI) de la Facultad de Ingeniería (FING) de la Universidad de la República (UdelaR, Uruguay) fue creado a comienzos del año 2006. En estos años que han transcurrido desde su creación, el GSI ha realizado una intensa actividad orientada al desarrollo de trabajo de investigación y de formación de recursos humanos. El objetivo de este documento es presentar los resultados generados por el grupo, poniendo especial énfasis en los proyectos de I+D realizados así como en las metodologías y herramientas que han sido diseñadas e implementadas para dar soporte a las mismas. Es importante destacar, sin embargo, que la formación de investigadores y profesionales en el área de la Seguridad Informática ha sido una cuestión estratégica que, en muchos de los trabajos realizados, ha sido el factor motivador principal para el desarrollo de los mismos. Es por esta razón que en este documento también se ha destinado un espacio a describir la actividad curricular de enseñaza que realiza el GSI, así como al ambiente tecnológico que se ha construido para dar soporte a las actividades de entrenamiento que complementan la formación teórica impartida en los cursos dictados por el grupo.

La estructura del resto del documento es la siguiente. La Sección 2 presenta los proyectos de investigación recientemente realizados y describe brevemente los proyectos de investigación en curso. La Sección 3 describe la actividad desarrollada por el grupo en relación a la elaboración de metodologías y herramientas diseñadas e implementadas. La Sección 4 provee una breve descripción de la actividad de enseñanza universitaria desarrollada por el equipo. Finalmente, la Sección 5 exhibe las conclusiones.

2. Proyectos de Investigación y Desarrollo

En esta sección se presenta brevemente la actividad de investigación que ha desarrollado el GSI desde su creación. Las secciones 2.1, 2.2 y 2.4 describen proyectos de I+D que han sido finalizados y la sección 2.3 provee detalles sobre investigación en curso.

2.1. ReSeCo: Reliability of Security Components

El proyecto STIC-AMSUD (`www.sticamsud.org`) *ReSeCo* ha tenido como principal objetivo investigar la seguridad y fiabilidad en un modelo computacional, donde tanto las plataformas como las aplicaciones son dinámicas, de forma que componentes provistos por un agente externo puedan ser destinados a formar parte de la plataforma o ejecutar una aplicación de forma segura. El proyecto buscó además incentivar la colaboración entre la comunidad científica, e industrial, de Francia y de países Sudamericanos. En este proyecto colaboraron investigadores del FAMAF de la Universidad de Córdoba (Argentina), del Departamento de Ciencia de la Computación de la Universidad de Chile (Chile), del Instituto de Computación de la Universidad de la República (Uruguay) y de los grupos Everest y Oasis del INRIA Sophia Antipolis (Francia).

Los objetivos específicos que fueron definidos para este proyecto son resumidos a continuación:

1. Investigar la seguridad y fiabilidad en un modelo computacional, donde tanto las plataformas como las aplicaciones son dinámicas, de forma que componentes provistos por un agente externo puedan ser destinados a formar parte de la plataforma o ejecutar una aplicación de forma segura
2. Con el objetivo de poder aplicar técnicas de PCC (Proof Carrying Code) [4] y PCR (Proof Carrying Results) [6] a sistemas distribuidos, se consideraron escenarios que involucran diferentes actores consumidores de código y computaciones. En dicho escenario, los componentes de software son sintetizados, por un proceso automático, a partir de varios fragmentos de código originados por diferentes productores de código. También se consideraron las situaciones en que los componentes han transitado a través de dispositivos que potencialmente pueden alterar su comportamiento computacional y los correspondientes datos

La contribución por parte de miembros del GSI y del Grupo de Métodos Formales del InCo al proyecto ReSeCo fueron principalmente desarrolladas en el contexto del proyecto STEVE, que es descrito a continuación. A modo de resumen, en relación al primer objetivo de ReSeCo, se trabajó en la especificación formal y verificación de componentes críticos de la plataforma JME, una tecnología Java específica para dispositivos móviles. Respecto al segundo objetivo, se desarrolló una infraestructura PCR para asegurar computaciones distribuídas en el lenguaje Acute [31].

2.2. STEVE: Seguridad a Través de Evidencia Verificable

El proyecto STEVE [30] ha sido concebido en el marco del proyecto de cooperación ReSeCo arriba descrito. Sus objetivos generales coinciden con los

del mencionado proyecto. Concretamente, STEVE permitió elaborar y concebir mecanismos que ayudan a los desarrolladores de software a construir sistemas confiables a partir de componentes existentes, así como infraestructuras que garantizan al usuario final que el software que éste utiliza es seguro y confiable. A continuación describimos brevemente las diferentes líneas de investigación enmarcadas en el proyecto STEVE e incluimos los principales resultados obtenidos en cada una de ellas.

Especificación y verificación formal de modelos de seguridad para dispositivos móviles. Los dispositivos portátiles, tales como teléfonos celulares y asistentes de datos personales, permiten almacenar información confidencial y establecer comunicaciones con entidades externas. Generalmente, los usuarios pueden descargar e instalar nuevas aplicaciones de fuentes no confiables, que conviven junto con las instaladas por el fabricante del dispositivo o proveedor de servicios de comunicación. En este escenario, es importante garantizar la confidencialidad e integridad de los datos almacenados, así como la disponibilidad del servicio, aún cuando una aplicación maliciosa trate de hacer uso indebido de las funciones del dispositivo. La plataforma Java Micro Edition (JME)[32], una tecnología para desarrollo de software Java, provee el estándar Mobile Information Device Profile (MIDP) que facilita el desarrollo de aplicaciones y especifica un modelo de seguridad para el acceso controlado a recursos sensibles del dispositivo. El modelo está construido sobre la noción de dominio de protección, que puede ser concebido como un conjunto de permisos. Un modelo alternativo que extiende los permisos presentes en MIDP ha sido propuesto por Besson, Dufay y Jensen [8]. Este modelo introduce una noción de multiplicidad asociada a los permisos y flexibiliza la forma en la que el usuario puede conceder acceso a los recursos del dispositivo, a las aplicaciones que son utilizadas en el mismo. En el contexto del proyecto STEVE se trabajó en la especificación, en el cálculo de construcciones inductivas [7], y la verificación formal, usando el asistente de pruebas Coq [33], de componentes de modelos de seguridad para dispositivos móviles interactivos. En particular, se analizaron las tres generaciones de MIDP y se desarrolló un framework, en [16], que permite definir y comparar formalmente políticas de control de acceso, que pueden ser aplicadas por variantes de los modelos de seguridad considerados. En [35] desarrollamos una especificación general del modelo de seguridad de MIDP 2.0 [19], y en [29] implementamos un algoritmo de control de accesos, que certificamos respecto al comportamiento esperado del módulo correspondiente en MIDP 2.0. El artículo [24] reporta la extensión del análisis formal a la reciente introducción de MIDP 3.0 [20], haciendo especial énfasis en el modelo de seguridad a nivel de aplicación que se incorpora sobre el modelo de seguridad a nivel de plataforma, ya existente en MIDP 2.0. En [25] desarrollamos algoritmos críticos certificados para nuevas funcionalidades de MIDP 3.0 y reportamos algunas vulnerabilidades de seguridad de la más reciente generación de MIDP, junto con propuestas de soluciones. Los diferentes prototipos certificados construidos podrían ser usados como implementaciones de referencia para, por ejemplo, desarrollar casos de prueba para las implementaciones industriales que se utilicen.

Infraestructura de Proof Carrying Results para computaciones distribuidas en el lenguaje Acute. Acute [31] es una extensión del lenguaje

ML, que provee mecanismos robustos y seguros para desarrollar y ejecutar programas creados en forma separada. En particular, el lenguaje soporta computación de valores distribuidos por intermedio de procedimientos de (un)marshall, que permiten la cooperación de programas que envíen y reciban valores a través de canales de comunicaciones no tipados. Esto posibilita la aparición de errores de tipos cuando un programa hace el unmarshall de un valor recibido. Además, el chequeo de un tipo abstracto no es siempre posible en tiempo de hacer el unmarshal. En una red heterogénea, en presencia de adversarios activos, estos mensajes pueden ser modificados. Existe una técnica, denominada Proof Carrying Results (PCR) [6], que propone asegurar los resultados obtenidos del cómputo de una función, agregando una prueba de que el valor fue computado de forma correcta. El objetivo de esta línea de trabajo ha sido la de proponer un modelo y mecanismos para adicionar verificación de valores de tipos abstractos, que son el resultado de cálculos distribuidos, aplicados en el lenguaje de programación Acute. Para lograr esto, como punto inicial se definió el modelo para dicha integración dentro del lenguaje de programación distribuido Acute. Este modelo define las posibles interacciones entre productores, consumidores, y demás máquinas que funcionarán dentro de este esquema. Se definió un protocolo para la verificación de los valores intercambiados entre productores y consumidores. El protocolo introduce además la posibilidad de realizar cálculos en forma distribuida entre varias máquinas, dependiendo de la propiedad a verificar. Dentro de este esquema se realizan cálculos en uno o más equipos para obtener un testigo de la computación que permite verificar, potencialmente de forma automática, que el valor calculado corresponda con el valor obtenido por el consumidor desde el productor. Se implementó dentro del lenguaje el reconocimiento del agregado de información para las operaciones de intercambio de información entre productores y consumidores. Esto se hizo mediante la modificación de los componentes del lenguaje. Se definió e implementó un prototipo funcional para la verificación de computaciones distribuidas en el lenguaje Acute. El protocolo de verificación incluye la conexión con el asistente de pruebas Coq. Esta actividad ha sido enmarcada formalmente en el trabajo de Tesis de Maestría del Pedeciba Informática de Felipe Zipitría [37] y un resumen del trabajo desarrollado se puede encontrar en [38]. Para demostrar la funcionalidad integral del prototipo se desarrolló un ejemplo de complejidad media que permitió validar la aplicabilidad de las soluciones desarrolladas. Puede verse el código asociado en [36].

2.3. VirtualCert: Hacia una Plataforma Certificada de Virtualización

El proyecto VirtualCert aborda el estudio del comportamiento de plataformas de computación virtuales. En un ambiente virtualizado, el Virtual Machine Monitor (VMM) (por ejemplo VMWare Workstation, Microsoft Virtual Server, Xen) es el componente que provee a las máquinas virtuales (VMs) el acceso a los recursos del hardware subyacente, es decir, el ambiente computacional sobre el que opera cada máquina. Este ambiente incluye el microprocesador, la memoria, los dispositivos de E/S, entre otros. Por lo tanto, y en particular, el VMM es el responsable de que estos recursos sean compartidos por las diferentes VMs en forma segura. El acceso compartido a los recursos de memoria de la plataforma

es un tema crítico, en particular si las diferentes VMs pueden potencialmente hostear diferentes sistemas operativos y, consecuentemente, los aplicativos que son ejecutados sobre los mismos. En una plataforma virtualizada, sobre la que por ejemplo se implanten los diferentes sistemas de información de una organización, pueden llegar a convivir en tiempo de ejecución sistemas y aplicativos de orígenes diversos, con comportamientos no necesariamente controlados y de diferentes niveles de confianza. El entendimiento y modelado de mecanismos de control de acceso y gestión de memoria en plataformas virtualizadas constituye un desafío de interés científico que, al menos para nuestro conocimiento, no ha sido hasta ahora abordado sistemáticamente haciendo uso de herramientas de especificación y verificación formal. El objetivo principal del proyecto VirtualCert es, en primer lugar, definir y especificar formalmente un modelo idealizado de una plataforma de virtualización, donde el VMM es un hypervisor à la Xen [2, 13], las VMs básicamente operan de hosts de sistemas operativos, y donde interesa modelar y verificar formalmente el correcto comportamiento de los mecanismos de gestión de los recursos de memoria de la plataforma.

Una vez obtenido este modelo idealizado de virtualización, del cual tenemos ya una primera versión formalizada en Coq, será posible entonces desarrollar trabajo que permita, entre otros aspectos, razonar sobre las dependencias entre los componentes y módulos de la plataforma, establecer y probar propiedades de no interferencia entre los sistemas operativos hosteados por la plataforma, y derivar/construir, a partir de la especificación formal, un prototipo funcionalmente correcto de hypervisor cuya ejecución garantice determinadas propiedades de seguridad, en particular las probadas sobre el modelo. Este modelo a su vez podrá servir de guía para actividades de desarrollo e implementación de plataformas de virtualización.

Este proyecto se está desarrollando en colaboración con el Dr. Gilles Barthe de IMDEA Software de España.

2.4. Hacia un CSIRT Nacional

A mediados del año 2005, ANTEL y la Facultad de Ingeniería iniciaron una ronda de discusión en torno a la definición de una actividad a desarrollar en el contexto del convenio marco acordado entre las dos instituciones y con la seguridad informática como eje temático. El proyecto que se planteó tuvo como objetivo fundamental el desarrollar actividades que contribuyan con sus resultados al complejo proceso de planificar, organizar, instalar y gestionar un *Computer Security Incident Response Team* (CSIRT) nacional. Este proyecto ha estado efectivamente operativo desde marzo de 2006. La actividad ha estado orientada a dos ejes principales de trabajo: por un lado se ha desarrollado trabajo que contribuyó al proceso de formación e instalación de un CSIRT nacional y por otro lado se trabajó en la concepción, diseño e implantación de un laboratorio de Seguridad Informática (ver 4.2), con base en la Facultad de Ingeniería y sustentado por recursos (tanto humanos como estructurales) de esta institución educativa y de ANTEL.

El modelo de centro de respuestas a incidentes informáticos es reconocido a nivel mundial como una excelente herramienta para encarar la problemática general de la seguridad informática. Este proyecto ha tenido como principal objetivo contribuir al desarrollo del área de coordinación en seguridad informá-

tica a nivel nacional. En el mismo se ha trabajado en un modelo innovador de colaboración entre la industria y la academia, algo no tradicional en Uruguay.

Este proyecto además de cumplir con los objetivos planificados, ha logrado en sinergia con la actividad del CSIRT de ANTEL importantes reconocimientos nacionales y regionales. En particular se han establecido numerosos contactos a nivel internacional, como ser el CERT.BR de Brasil, ARCERT de Argentina y AUSCERT de Australia. Con el apoyo de algunos de estos centros el CSIRT de ANTEL fue admitido como miembro del FIRST (Forum of Incident Response and Security Teams). A fines de 2007, la Agencia para el Gobierno de Gestión Electrónica y la Sociedad de la Información y el Conocimiento (AGESIC) conformó ocho grupos de trabajo con el objetivo de definir líneas de acción para la implantación de políticas e instrumentos para la gestión del gobierno electrónico en el Uruguay. Uno de estos grupos, el de Seguridad de la Información, definió como uno de los proyectos estratégicos la creación del CSIRT nacional, encargado de coordinar las respuestas ante problemas asociados a la actividad maliciosa en informática en nuestro país. A fines del año 2007, se crea el Consejo Honorario Asesor de Seguridad de la Información de la AGESIC, que define como proyecto prioritario la creación del CSIRT nacional, finalmente llamado CERTuy. En setiembre de 2008 Presidencia de la República decreta la creación del centro, el que actualmente se encuentra operativo y trabajando bajo la égide de la AGESIC (`www.cert.uy`).

3. Metodologías y Herramientas

Esta sección describe brevemente trabajo desarrollado por el GSI orientado a la generación de metodologías, técnicas y herramientas para la implementación de mecanismos que contribuyan al aseguramiento de las plataformas computacionales y de comunicación de datos.

3.1. FRENESI: Un Framework para el Entrenamiento en Seguridad Informática

En estos últimos años se ha incrementado en forma sustancial el interés en la seguridad informática. En particular, el aseguramiento de las plataformas computacionales y de comunicación de las organizaciones y empresas es un aspecto crítico con el que los profesionales informáticos se han visto enfrentados desde hace ya un cierto tiempo, pero que en la actualidad se ha tornado ineludible. Los ataques informáticos representan hoy un serio problema, que se ha visto acentuado debido al incremento de las vulnerabilidades que presentan los sistemas computacionales y la complejidad de esos ataques. Un gran desafío que se plantea es la formación y entrenamiento de profesionales con capacidad para enfrentarse a las amenazas actuales y futuras, aplicando técnicas y tecnologías de seguridad en ambientes de producción. Dichos profesionales deben ser capaces de entender, habiendo identificado las vulnerabilidades de una plataforma de TI bajo su responsabilidad, cuales son los riesgos y las amenazas a las que se puede ver enfrentada esa plataforma. La experiencia práctica es un aspecto fundamental de esta preparación.

En este contexto, poder contar con ambientes de entrenamiento en los cuales se pueda experimentar sobre plataformas que reproduzcan ambientes

de producción, incluyendo sus vulnerabilidades, con técnicas, herramientas y reproduciendo problemas reales es fundamental. Un ambiente equipado para la experimentación, observación, práctica y prueba usualmente requiere, entre otros aspectos, una inversión significativa en hardware, dispositivos de red, cableados y equipamiento informático en general. Adicionalmente, para poder implantar un laboratorio de estas características es necesario disponer de una infraestructura aislada de los sistemas en producción, donde poder experimentar sin interferir con el funcionamiento normal de otros sistemas. La plataforma a utilizar debe permitir la instalación, configuración, funcionamiento y evaluación de herramientas de seguridad informática, así como el análisis en condiciones controladas de vulnerabilidades y ataques a los sistemas operativos y servicios.

El proyecto FRENESI tiene como objetivo general el diseñar, desarrollar e implantar una plataforma para el entrenamiento en seguridad informática. Esta plataforma debe satisfacer requerimientos esenciales para un ambiente de este tipo [34], esto es, el ambiente debe ser:

1. *Reconfigurable*: Diferentes temas a investigar o entrenamientos a realizar requieren diferentes configuraciones, ya sea de sistemas operativos y las aplicaciones que ejecutan sobre los mismos, o de topologías de red. Debe ser posible modificar la configuración del ambiente de forma fácil y eficiente
2. *Heterogéneo*: Debe involucrar diferentes plataformas de distintos proveedores
3. *Escalable*: Debe poder escalar fácilmente y debe soportar un número importante de usuarios sin que la performance se vea degradada
4. *Rentable*: El costo de configuración y mantenimiento debe ser considerablemente menor al costo de lo que se está intentando simular
5. *Robusto*: Debe poder soportar y reponerse rápidamente de daños producidos por los usuarios o por las propias instalaciones o pruebas que se estén realizando
6. *Mantenible*: Debe ser fácil de mantener. Tareas como respaldos, aplicación de parches o actualizaciones deben ser fáciles de realizar y lo más automatizadas posibles
7. *Realista*: Debe proveer escenarios que sean lo más próximos posibles a la realidad que se desea emular e investigar
8. *Aislado*: Las actividades desarrolladas sobre el mismo no deben afectar a otras instalaciones.

Una primera, y esencial, decisión de diseño para el desarrollo de esta plataforma ha sido hacer un uso intensivo de tecnologías de virtualización. El uso de esta tecnología permite particionar, por ejemplo, un único servidor físico en múltiples máquinas virtuales, que pueden definirse con variadas características de hardware y en las que se pueden instalar diversos sistemas operativos y aplicaciones. Esto se convierte en una base ideal para el objetivo de proveer una infraestructura que permita el testeo y evaluación de sistemas y aplicaciones diversas. Asimismo, la posibilidad de poder crear una máquina virtual y realizar varias copias de la misma, facilita el armado de entrenamientos y talleres de formación y la rápida repetición de los mismos y de cualquier experimentación que se desee realizar. Mediante la configuración de switches virtuales, por ejemplo, es posible soportar numerosas redes totalmente independientes entre sí. A su vez, cada uno de estos switches pueden conectar del orden de 40 maquinas virtuales.

Un primer paso hacia la implementación de un ambiente de este tipo consistió en concebir una arquitectura que permitiera satisfacer los requerimientos planteados y definir una metodología para la elaboración de entrenamientos en seguridad informática. Los resultados de esta actividad han sido reportados en [10]. En una siguiente etapa el trabajo se focalizó en concebir mecanismos que permitieran automatizar la generación de los ambientes, y en particular se definió e implementó un lenguaje, y su correspondiente compilador, para la especificación de escenarios de experimentación. Este trabajo ha sido reportado en [11]. El trabajo en curso tiene como objetivo principal refinar y extender el lenguaje de definición de escenarios, incorporar al framework un lenguaje para la administración de entrenamientos y definir e implantar mecanismos de control de acceso a los componentes de un ambiente de experimentación de este tipo.

3.2. Honey*

Los primeros trabajos relacionados con la tecnología de Honeypot datan del año 1999. Esta tecnología presenta un enfoque diferente a las técnicas tradicionales que se basan en tomar medidas de prevención o detección de los sistemas a través de, por ejemplo, Firewalls, IDSs y encriptación. Los Honeypots, en cambio, son sistemas o señuelos sin ningún tipo de valor de producción que en forma pasiva esperan ser atacados o vulnerados. De hecho, el valor de los mismos reside en ser recursos "atractivos" para los atacantes. Se los puede entender como sensores que están siendo monitoreados y registrando toda la actividad dirigida hacia los mismos.

Los tipos de Honeypot se clasifican de acuerdo al nivel del interacción que se le ofrezca al atacante. Los de *bajo nivel de interacción* son sensores que emulan o simulan el comportamiento de sistemas y servicios reales. En este caso el atacante interactúa únicamente con los servicios emulados (p.e. SMTP, POP, IMAP, HTTP, etc.) a través del Honeypot pero nunca con el sistema operativo base sobre el que se ejecuta el Honeypot o contra servicios reales.

Los de *alto nivel de interacción* no emulan ningún servicio sino que ofrecen servicios reales, ejecutados sobre un sistema operativo también real sobre el cual el atacante podría incluso interactuar. Estos últimos ofrecen un objetivo muy atractivo a los atacantes y permiten obtener un mayor conocimiento sobre los mismos y las herramientas y técnicas empleadas para vulnerar los sistemas. Son especialmente usados en ámbitos académicos y de investigación. Este tipo de Honeypots tiene asociado un riesgo considerablemente mayor y los costos de hardware y en especial de administración de los mismos son altos.

También se puede clasificar a los Honeypots de acuerdo al recurso que están simulando. Normalmente el termino Honeypot se utiliza cuando el recurso que es usado como carnada es un computador. Sin embargo este recurso podría tratarse de un Access Point WIFI (Wifipot), una red de computadoras (Honeynet), un servidor de correo para la captura de spam (Spampot), una tarjeta de crédito o un archivo (Tokenpot).

El GSI ha realizado un intenso trabajo orientado primero a adquirir un afinado manejo de los conceptos involucrados por esta área y luego al desarrollo de experimentos que permitan un acercamiento a las técnicas involucradas. En particular, la actividad se ha focalizado en trabajar sobre Honeypots de bajo

nivel de interacción, aunque ademas se estudiaron Honeypots de alto nivel de interacción y en especial implementaciones de Honeynet. El motivo de esto fue buscar puntos de contacto que nos permitiera escalar de soluciones de bajo a alto nivel de interacción con un mínimo costo. Para esto, se estudiaron las técnicas de implementación basadas en máquinas virtuales (virtual Honeynet) y en implementaciones de tercera generación de Honeynet.

Con el objetivo específico de experimentar con las técnicas más relevantes de esta disciplina, se trabajó sobre una solución que cumpliera con los siguientes requerimientos:

- El mecanismo de instalación o configuración debe facilitar la reubicación de los Honeypots de la solución.
- Se debe contar con un sistema de monitoreo y alarmas de la actividad del Honeypot.
- La solución debe ser independiente de las tecnologías empleadas para implementar el Honeypot y ser escalable.
- Se tiene que permitir el monitoreo de uno o más Honeypots, buscando así una solución al intercambio y manejo de datos recolectados de distintas fuentes o implementaciones de Honeypots.

El ambiente resultante fue diseñado e implantado exitosamente en la Facultad de Ingeniería. La solución consiste en un Honeypot de bajo nivel de interacción instalado sobre una maquina virtual, basado en la implementación de *Honeyd* [28], para la captura de datos y el procesamiento estadístico de los mismos. El objetivo de esta primer versión es brindar una instalación segura de honeyd que sea fácilmente puesta en producción en distintos puntos y sea adaptable a nuevas versiones del mismo. Adicionalmente, esta implementación ha permitido identificar las dificultades que presenta la recolección de datos generados por distintos Honeypots, debido a la ausencia de un modelo de datos estándar que permita uniformizar la gestión y su posterior tratamiento. Esto afecta, por ejemplo, la implementación de un sistema de alarmas y el proceso de la sanitización de los datos recolectados ante la necesidad de efectuar un intercambio de los mismos. Asimismo, concurrentemente se construyó un Spampot usando la implementación de Honeyd y scripts de emulación del servicio SMTP del *Cert.br* [1]. Esta tarea fue concebida como una prueba de concepto y de experimentación de la arquitectura desarrollada. Actualmente se están haciendo pruebas con la misma en ambientes de producción en Facultad de Ingeniería, en forma exitosa. Este trabajo ha sido reportado en detalle en [17].

3.3. Automatización de Recolección de Evidencia Digital

Debido a la creciente complejidad de los procesos en las investigaciones forenses digitales, es fundamental el desarrollo de metodologías y técnicas que permitan agilizar y automatizar las tareas realizadas.

En particular, la recolección de evidencia digital es una actividad delicada que exige esfuerzo y una considerable experiencia por parte del analista que realiza la tarea. En muchos casos, los procedimientos forenses empleados se construyen de manera informal pudiendo comprometer la eficacia y/o la integridad de la investigación.

A la hora de recolectar evidencia, el investigador forense se encontrará con una enorme cantidad de información que seguramente contenga evidencia potencial. Por consiguiente, se desprende la importancia de establecer el orden en el cual se recolectará dicha información. El RFC 3227 [21] establece este orden en base a la volatilidad de los datos.

En el artículo *A Formalization of Digital Forensics* [22] se propone un modelo teórico para analizar y construir procedimientos forenses desde una punto de vista formal. Según el ciclo de vida de una investigación forense digital propuesto en [3], dicho modelo se ubica en la tercera de las seis etapas propuestas, denominada *collection*, la cual se describe como extracción de ítems individuales o grupos de evidencia.

Por otro lado, *Open Vulnerability and Assessment Language (OVAL)* [14], concebido por MITRE Corporation, formaliza y normaliza los intercambios de información entre herramientas relacionadas con la gestión de vulnerabilidades. OVAL incluye un lenguaje utilizado para especificar sistemas de información y evaluar la presencia de vulnerabilidades en los mismos.

Este lenguaje estandariza los tres principales pasos en la evaluación de un sistema:

1. Representar la información de configuraciones de sistemas a ser evaluados.
2. Analizar el sistema en busca de estados de máquina específicos (vulnerabilidad, configuración, estado o versiones de parches de seguridad, etc.).
3. Reportar los resultados de la evaluación.

En este contexto, el objetivo de esta línea de trabajo ha sido extender el lenguaje propuesto por OVAL para especificar procedimientos forenses, tomando como marco de referencia el modelo propuesto en [22]. A continuación se describen los resultados obtenidos.

Se investigó el lenguaje OVAL en detalle para poder establecer una vinculación con el modelo formal, concluyendo que puede ser utilizado como una implementación del modelo propuesto, obteniendo así los beneficios de la formalidad y rigurosidad que el modelo ofrece.

Debido que el intérprete de referencia de OVAL, Ovaldi [12], presenta múltiples falencias que dificultan la incorporación de nuevos procedimientos, se diseño e implementó un intérprete para el lenguaje OVAL extendido. Éste es totalmente compatible con la especificación original de OVAL y además contempla las funcionalidades necesarias para dar soporte a la extensión realizada para procedimientos forenses. La herramienta está diseñada para ser multiplataforma y es fácilmente extensible por su arquitectura basada en plugins. Los plugins eliminan la necesidad de trabajar en el intérprete como un todo. El desarrollador simplemente se enfoca en cómo resolver la recolección de la evidencia deseada.

Actualmente se está evaluando la funcionalidad de la herramienta y la adecuación del modelo propuesto. En los últimos tiempos ha cobrado importancia la recolección de evidencia en "sistemas vivos" (live system) [23] y se ha afianzado la necesidad de herramientas que automaticen los procesos de recolección de evidencia altamente volátil, como por ejemplo: lista de procesos activos, archivos abiertos y conexiones de red. Motivados por dicha necesidad, se crearon plugins para contemplar la recolección de evidencia en sistemas vivos y de esta manera poder evaluar la problemática involucrada en la adquisición de este tipo de evidencias. Este trabajo ha sido reportado en detalle en [5].

3.4. Metodologías para Implantación de SGSI

Cada vez más las organizaciones, sin importar su tamaño o actividad, se ven obligadas a considerar la seguridad de la información como un tema relevante y en la necesidad de implementar sistemas para gestionar la seguridad de la información, para proteger sus activos más sensibles. Por otro lado, el surgimiento de normas internacionales que promueven la implantación de Sistemas de Gestión de Seguridad de la Información (SGSI), y su rápida adopción, genera la necesidad de acelerar el proceso de implantación de los mismos. Sin embargo, existen estructuras organizacionales que requieren de un análisis detallado, ya sea por la criticidad de la información que manejan, su dimensión o su estructura empresarial, para lograr una implantación que provea los niveles de seguridad requeridos y brinde la confianza necesaria a la propia organización, a los socios de negocio y a los usuarios.

En este contexto, el GSI ha realizado trabajo en torno al desarrollo de metodologías y herramientas que faciliten y apoyen la implantación de SGSI en organizaciones con diferentes características. En [9] se presenta una metodología para la implementación y mejora continua de SGSI en empresas con características de pequeña y mediana. La misma, que surge como resultado de una tesis de maestría [15], complementa el modelo definido por la norma ISO/IEC 27001, contribuyendo a obtener una implantación efectiva de cada una de las actividades especificadas en el referido modelo, contemplando las limitaciones y estructura organizacional de este tipo de empresas. Como complemento a esta metodología se realizó la especificación de un software para facilitar la realización de las diferentes actividades propuestas y gestionar de forma efectiva la documentación, entendiendo que la documentación es parte fundamental en el proceso de certificación que se asocia a las normas mencionadas. Un prototipo con las principales funcionalidades de esta herramienta de software fue implementado y presentado en [18].

Recientemente se ha generado una metodología para la implantación de SGSI en grupos de empresas con una estructura empresarial jerárquica. Dicha metodología, descripta en [26], destaca la necesidad de una cooperación entre los SGSI de las empresas pertenecientes al grupo, pero a su vez indica que cada SGSI requiere de flexibilidad y agilidad operativa que permitan a cada empresa alcanzar los niveles de seguridad y objetivos específicos. Propone un enfoque de dirección centralizada pero con autonomía a nivel de cada dominio y cada empresa. La metodología propone, además, un organigrama que establece los roles necesarios y cómo se vinculan entre sí en cada una de las etapas de implantación del SGSI. También se plantea una estructura de documentación que contempla los requisitos de la norma ISO/IEC 27001 y las necesidades de la estructura empresarial analizada. Se utilizan grafos para representar los activos, su valoración y la relación de dependencia entre los mismos, proponiendo un algoritmo de ajuste y revisión de los valores definidos en función de la dependencia entre los activos y procesos de las diferentes empresas. El trabajo de tesis de maestría [27], del cual es resultado esta metodología, estudia el caso particular de una empresa telefónica y un proveedor de servicios de Internet.

4. Formación de Recursos Humanos

Esta sección describe las actividades de formación y enseñanza curricular que desarrolla el GSI.

4.1. Formación curricular

Fundamentos de la Seguridad Informática. Desde el año 2007, el GSI dicta en el primer semestre de cada año lectivo el curso *Fundamentos de la Seguridad Informática* (`www.fing.edu.uy/inco/cursos/fsi`). Este curso es ofrecido como electiva técnica de la carrera Ingeniero en Computación y como curso de posgrado del programa de Posgrado del Área Informática del Pedeciba (Programa de Desarollo de las Ciencias Básicas) del Uruguay.

La formación teórica del estudiante es complementada con trabajos prácticos que son desarrollados en el Laboratorio de Seguridad Informática, que ha sido diseñado e implantado por el GSI. La actividad que ha desarrollado el grupo en torno a este laboratorio ha sido reportada en [10, 11] y se describe en mayor detalle en la sección 4.2.

El GSI dicta asimismo una versión de este curso destinada a profesionales en el marco del Diploma de actualización profesional del Centro de Posgrado Profesional (CPAP) del InCo.

Taller de Seguridad Informática. Este taller tiene como objetivo principal introducir al estudiante a la implementación de servicios y funcionalidades orientadas al ámbito de la seguridad informática. Allí se experimenta desarrollando, por ejemplo, funciones de autenticación, plugins para herramientas de seguridad, y configurando funcionalidades complejas de los sistemas operativos. También se busca promover los conocimientos para el desarrollo de código en forma segura, mediante el estudio de las estructuras y funcionalidades provistas por diferentes frameworks de seguridad, disponibles para distintos sistemas operativos.

La primera instancia de este taller, que también hará uso de la infraestructura del laboratorio de seguridad antes mencionado, tendrá lugar en el segundo semestre de 2010.

4.2. Laboratorio de Seguridad Informática

El Laboratorio de Seguridad Informática (LaSI) ha sido concebido como un centro de actividades de difusión, formación, experimentación e investigación en torno a los múltiples aspectos de la seguridad informática. El conocimiento y manejo de las distintas herramientas que permiten implementar medidas de protección, detección y prevención de incidentes es una necesidad en el ámbito de las Tecnologías de la Información y la Comunicación, y se considera que un enfoque práctico es particularmente enriquecedor. El laboratorio permite disponer de una infraestructura, aislada de los sistemas en producción, donde poder experimentar. Esta plataforma permite la instalación, configuración, funcionamiento y evaluación de herramientas de seguridad informática así como el análisis en condiciones controladas de vulnerabilidades y ataques a los sistemas operativos y servicios.

Uno de los objetivos principales del LaSI es poder consolidarse como un ámbito donde se concentren actividades vinculadas a la seguridad informática. Éste debe propiciar la creación de grupos de trabajo, integrados por docentes y profesionales, proveyendo un ambiente que propicie la formación e investigación en el área. Otro objetivo importante es el de contribuir a la formación de recursos humanos capacitados en seguridad informática mediante la realización de talleres, cursos y seminarios con importante contenido práctico, así como generar un espacio de experimentación e investigación para el desarrollo de maestrías y doctorados en el área.

El LaSI ha sido implantado en el Instituto de Computación (INCO) de la Facultad de Ingeniería de la UdelaR. Los estudiantes acceden al ambiente de entrenamiento y sus funcionalidades desde una aula taller que cuenta con varios PCs y que se encuentra en una red independiente de la red de producción, separada de la misma a través de un firewall. Esta aula es compartida por variadas actividades docentes y estudiantiles, lo que limita la adecuación de esta infraestructura a las necesidades particulares del laboratorio. Cualquier modificación que se realice a las máquinas de este salón podría afectar las otras actividades. Este fue el principal motivo por el cual se decidió implementar el LaSI mediante una arquitectura mixta. En la misma se utiliza la infraestructura física (PCs, switches, cables) provista por la mencionada aula del INCO, para acceder a una red virtual donde realmente se realizan las prácticas. Esta arquitectura esta soportada por tres servidores que disponen de capacidad suficiente para soportar el entorno virtual donde se ejecutan las máquinas virtuales, sobre las que se efectúan los experimentos y entrenamientos.

Por una presentación más detallada del laboratorio y las actividades desarrolladas en torno al mismo ver [10, 11].

5. Conclusión

Desde su concepción, en el año 2006, el GSI definió como objetivo general desarrollar actividad orientada a la investigación, innovación y formación en seguridad informática. En este documento se presentó, en forma sucinta, el trabajo realizado por el GSI y se describieron brevemente los resultados obtenidos en estos primeros años, proveyendo asimismo referencias a los artículos publicados por miembros del equipo.

El objetivo principal en el corto plazo es profundizar el trabajo sobre las líneas de investigación en curso, la mayoría de ellas presentadas en este artículo. Por otra parte, el GSI se propone afianzar y consolidar sus vínculos con instituciones y grupos, tanto internacionales como regionales, que desarrollan investigación de primer nivel en el área de la Seguridad Informática. La participación en redes de cooperación y la formación de recursos humanos a nivel de postgrado son sin duda objetivos esenciales sobre los que se trabajará intensamente en los próximos años.

Referencias

[1] *Centro de Estudos, Resposta e Tratamiento de Incidentes de Segurança do Brasil.* `http://www.cert.br`.

[2] *The Xen Hypervisor*. `http://www.xen.org/products/xenhyp.html`.

[3] *A Road Map for Digital Forensics Research*. Informe técnico, Digital Forensics Research Workshop, 2001.

[4] Appel, Andrew W.: *Foundational Proof-Carrying Code*. En *Logic in Computer Science*, 2001. http://citeseer.ist.psu.edu/appel01foundational.html.

[5] Barrere, Martín: *Automatización de Procesamiento de Evidencia Digital*. Tesis de Grado, Facultad de Ingeniería, Universidad de la República, 2010.

[6] Barthe, G. y F. Pastawsky: *Notes on Proof Carrying Results*. INRIA Sophia-Antipolis Technical Report, 2006.

[7] Bertot, Y. y P. Castéran: *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. Springer-Verlag, 2004. http://www.labri.fr/publications/l3a/2004/BC04.

[8] Besson, F., G. Duffay y T. Jensen: *A Formal Model of Access Control for Mobile Interactive Devices*. En *11th European Symposium on Research in Computer Security (ESORICS'06), LNCS 4189*, páginas 110–126, 2006.

[9] Betarte, G., M. Corti y R. de la Fuente: *Hacia una Implementación exitosa de un SGSI*. En *III Congreso Iberoamericano de Seguridad Informática, CIBSI'05*, Noviembre 2005.

[10] Betarte, G., M. E. Corti y M. Rodríguez: *Concepción, Diseño e Implantación de un Laboratorio de Seguridad Informática*. En *IV Congreso Iberoamericano de Seguridad Informática, CIBSI'07*, Noviembre 2007.

[11] Blanco, A., J.D. Campo, L. Escanellas, C. Pintado y M. Rodríguez: *Generación de Ambientes para Entrenamiento en Seguridad Informática*. En *V Congreso Iberoamericano de Seguridad Informática, CIBSI'09*, Noviembre 2009.

[12] Buttner, A.: *The OVAL Interpreter*. `http://sourceforge.net/projects/ovaldi/`.

[13] Chisnall, D.: *The Definitive Guide to the Xen Hypervisor*. Prentice Hall, 2007.

[14] Corporation, MITRE: *Open Vulnerability Assessment Language*. `http://oval.mitre.org/`.

[15] Corti, María Eugenia: *Análisis y Automatización de la Implantación de SGSI en Empresas Uruguayas*. Tesis de Maestría, Universidad de la República, Junio 2006.

[16] Crespo, J., G. Betarte y C. Luna: *A Framework for the Analysis of Access Control Models for Interactive Mobile Devices TYPES*. En al, S. Berardi et (editor): *Types for Proofs and Programs 2008*, volumen 5497 de *Lectures Notes in Computer Science*, páginas 49–63. Springer-Verlag, January 2009.

[17] Cócaro, F. y M. García: *Diseño e Imlementación de un Honeypot*. Tesis de Grado, Facultad de Ingeniería, Universidad de la República, 2007.

[18] Gelós, M. y N. De Maio: *Automatización de Actividades de Implantación y Mejora Continua de un SGSI*. Tesis de Grado, Universidad de la República, 2007.

[19] Group, JSR 118 Expert: *Mobile Information Device Profile for Java 2 Micro Edition. Version 2.0*. Informe técnico, Sun Microsystems, Inc. and Motorola, Inc., 2002.

[20] Group, JSR 118 Expert: *Mobile Information Device Profile for Java 2 Micro Edition. Version 3.0*. Informe técnico, Sun Microsystems, Inc. and Motorola, Inc., 2008.

[21] Killalea, Bcp T.: *Network Working Group D. Brezinski Request for Comments: 3227 In-Q-Tel BCP: 55 T. Killalea Category: Best Current Practice neart.org February 2002 Guidelines for Evidence Collection and Archiving*, Marzo 27 2002. `http://citeseer.ist.psu.edu/652246.html;http://www.tzi.de/~cabo/pdfrfc/rfc3227.txt.pdf`.

[22] Leigland, Ryan y Axel W. Krings: *A Formalization of Digital Forensics*. International Journal of Digital Evidence, 3, 2004.

[23] Lessing, Marthie: *Live Forensic Acquisition as Alternative to Traditional Forensic Process*. En Göbel, Oliver, Sandra Frings, Detlef Günther, Jens Nedon y Dirk Schadt (editores): *IMF*, volumen 140 de *LNI*, páginas 107–124. GI, 2008, ISBN 978-3-88579-234-5.

[24] Mazeikis, G., G. Betarte y C. Luna: *Formal Specification and Analysis of the MIDP 3.0 Security Model*. En *XXVIII International Conference of the Chilean Computer Science Society*, IEEE CS Press, 2010. To be published.

[25] Mazeikis, G. y C. Luna: *Autorización de Acceso en MIDP 3.0*. En *V Congreso Iberoamericano de Seguridad Informática, CIBSI'09*, Noviembre 2009.

[26] Pallas, G. y M. Corti: *Metodología de Implantación de un SGSI en Grupos Empresariales de Relación Jerárquica*. En *V Congreso Iberoamericano de Seguridad Informática, CIBSI'09*, Noviembre 2009.

[27] Pallas, Gustavo: *Metodología de Implantación de un SGSI en un Grupo Empresarial Jerárquico*. Tesis de Maestría, Universidad de la República, 2010.

[28] Provos, N.: *Honeyd web site*. `http://www.honeyd.org`.

[29] Roushani, R., G. Betarte y C. Luna: *A Certified Access Controller for JME-MIDP 2.0 enabled Mobile Devices*. En *XXVIII International Conference of the Chilean Computer Science Society*, IEEE CS Press, 2010. To be published.

[30] Seguridad Informática, Grupo de: *STEVE: Seguridad a Través de Evidencia Verificable*. `http://www.fing.edu.uy/inco/grupos/gsi`, Proyecto Clemente Estable, Edición 2006.

[31] Sewell, Peter, James J. Leifer, Keith Wansbrough, Mair Allen-Williams, Francesco ZappaÑardelli, Pierre Habouzit y Viktor Vafeiadis: *Acute: High-level programming language design for distributed computation. Design rationale and language definition*. Informe técnico UCAM-CL-TR-605, University of Cambridge Computer Laboratory, Octubre 2004. http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-605.html, Also published as INRIA RR-5329. 193pp.

[32] Sun Microsystems, Inc.: *Java Platform Micro Edition*, Last accessed: Jul 2008. http://java.sun.com/javame/index.jsp.

[33] The Coq Development Team: *The Coq Proof Assistant Reference Manual – Version V8.2*, 2009. `http://coq.inria.fr`.

[34] Yang, T. Andrew, Kwok Bun Yue, Morris Liaw, George Collins, Jayaraman T. Venkatraman, Swati Achar, Karthik Sadasivam y Ping Chen.: *Design of a distributed computer security labs*. En *Rocky Mountain Conference*, página 332–346. Consortium for Computing Sciences in Colleges, 2004.

[35] Zanella, S., G. Betarte y C. Luna: *A Formal Specification of the MIDP 2.0 Security Model*. En *Formal Aspects in Security and Trust, Fourth International Workshop, FAST 2006, Hamilton, Ontario, Canada, August 26-27, 2006, Revised Selected Papers*, volumen 4691 de *LNCS*, páginas 220–234. Springer-Verlag, 2006.

[36] Zipitría, Felipe: *PCR extension to the Acute source code*, 2008. http://www.fing.edu.uy/inco/grupos/gsi/sources/acute-pcr/index.html.

[37] Zipitría, Felipe: *Towards Secure Distributed Computations*. Tesis de Maestría, Universidad de la República, Noviembre 2008. http://www.fing.edu.uy/ fzipi/tesis/tesis.pdf, PEDECIBA Informática - Technical Report - TR 08-22.

[38] Zipitría, Felipe: *Towards Secure Distributed Computations*. En *V Congreso Iberoamericano de Seguridad Informática, CIBSI'09*, Noviembre 2009.