FACTOIDS: Modelos y Herramientas para el Análisis e Intercambio Seguro de Datos Colectados por Sensores

Carlos Martínez-Cagnazzo

Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, Montevideo, Uruguay

carlosm@fing.edu.uy

Resumen. Los *Computer Security Information Response Teams* (CSIRTs) son organizaciones de servicio que alojan equipos de trabajo altamente especializados que realizan respuesta a incidentes de seguridad ya sea a nivel de coordinación u operativa. Las actividades maliciosas en Internet no reconocen fronteras de ningún tipo por lo que para poder llevar adelante su tarea con la mayor efectividad posible, los CSIRTs deben establecer relaciones de confianza entre sí para poder compartir información. Para ser aceptables, estos intercambios de información deben además respetar las políticas de seguridad de la información de las organizaciones madre de los respectivos equipos de seguridad y para ser lo más efectivos posible deberían poder ser automatizables. Este trabajo presenta una introducción a este escenario de intercambio de información entre CSIRTs, para luego analizar algunas herramientas y arquitecturas relevantes encontradas en la literatura, realizar una análisis de requerimientos y proponer una arquitectura de alto nivel para intercambio de información automáticos entre CSIRTs a través de dominios administrativos de manera de respetar las políticas de SI de cada organización.

Palabras clave: CSIRTs, Logs, Information Exchange, FACTOIDS, IODEF, IDMEF, Sanitization, Anonymity, Intrusion Detection, Honeypots, Honeynets

1 Introducción

En cualquier instalación de red actual existen múltiples sensores que proporcionan datos sobre eventos de seguridad que pueden ser útiles como entrada para un grupo de respuesta a incidentes de seguridad (CSIRT). Estos sensores incluyen, pero no se limitan a: sistemas de detección de intrusos (IDS), firewalls, routers y diferentes clases de honeypots (1).

Típicamente un CSIRT opera y obtiene datos de una o más de estas fuentes. Sin embargo, la visibilidad que cada uno de los grupos tiene individualmente está limitada por los sensores que cada grupo dispone y administra, los que generalmente son

limitados en número y localizados bajo un único proveedor de servicios de Internet (ISP) En el caso de los sensores de tipo honeypot influye además el tamaño del rango de direcciones IP que el sensor utiliza y además el comportamiento observado puede tener componentes fuertemente locales.

Debido a esto el poder compartir estos datos entre grupos de trabajo permitiría a los investigadores tener una visión más global de los eventos que ocurren en la red y les permitiría extraer conclusiones de validez más general, así como también actuar mitigando amenazas en el caso de los CSIRTs.

Existen diversos obstáculos para que estos intercambios de datos entre grupos de trabajo sean posibles, entre ellos:

- Falta de normalización de los datos colectados por diferentes sensores
- Falta de buenas técnicas de representación interna de los datos y de almacenamiento de los mismos
- Protocolos de transporte de datos de seguridad entre grupos de trabajo
- Necesidad de sanitizar datos antes de compartirlos de acuerdo a las políticas de seguridad de la información del sitio origen de los datos
    - Dependiente del origen, tipo y destino de los datos
    - Teniendo en cuenta además posibles requisitos administrativos y/o legales del sitio origen de los datos
- Si bien distintos grupos tienen interés en consultar información colectada por otros, por un lado o todos los datos que colecta un grupo son de interés para otros, además es generalmente de interés el poder agregar o *minar* esta información de diferentes maneras.

  Esto sumado a los importantes volúmenes de información que se manejan, hace necesaria una forma de *consultar* los datos colectados por un grupo por parte de otros de forma automatizada.
- Necesidad de contar con garantías de autenticidad e integridad de los datos obtenidos de otros grupos
- Necesidad de establecer *relaciones de confianza* entre grupos que habiliten el intercambio de información. En general estas relaciones de confianza pasan por el contacto institucional y la firma de alguna forma de "MoU" (*Memorandum of Understanding*), documento que establece en líneas generales las reglas que hacen posible la colaboración.

  Estos MoU se establecen es escalas temporales que van de las semanas a meses, muchas veces excediendo el tiempo que los grupos disponen para responder con efectividad a un incidente.

Es de nuestro entender que la posibilidad de establecer estas relaciones de confianza de forma automatizada junto con las garantías de integridad de la información habilitarían nuevas áreas de cooperación así como permitiría una rápida respuesta a incidentes que abarcan grupos de trabajo que no han tenido contactos institucionales previos.

Existen en la literatura propuestas de arquitecturas para compartir información así como también múltiples trabajos que proponen protocolos, lenguajes o componentes que podrían ser utilizados en un sistema de este tipo.

Además de la comunicación entre grupos de trabajo en seguridad, la normalización de la representación de los datos colectados por sensores en sí misma permitiría la correlación de eventos y extracción de conclusiones de forma más ágil y sencilla.

En la segunda sección presentaremos los principales problemas que se deben resolver para habilitar el intercambio de información a través de fronteras administrativas, para luego continuar con referencias a sistemas que atacan problemas similares y terminar con una propuesta de arquitectura (FACTOIDS) que proponemos como modelo de referencia.

2 Aspectos generales y problemas a resolver para facilitar la comunicación entre organizaciones

2.1 Características generales de las fuentes de información de seguridad

En el entorno de red moderno los especialistas en seguridad cuentan con una multitud de fuentes de eventos de seguridad, como ser sistemas de detección de intrusos (IDSs), archivos de log de firewalls, archivos de log de distintos servidores (web, archivos, etc.) y fuentes *Netflow* (2) entre otros.

De particular interés para los equipos de respuesta a incidentes de seguridad (CSIRTs) resultan los *honeypots* y *honeynets.* De acuerdo a (1) y (3), un *honeypot* es un recurso cuyo valor reside en el uso no autorizado o ilícito del mismo. Los honeypots pueden ser de alta o baja interacción según el recurso expuesto sea virtualizado o consista en un sistema real. De acuerdo a (3) *honeynet* es un tipo de honeypot de *alta interacción*.

Como toda actividad detectada en un honeypot se asume maliciosa, estos dispositivos proveen fuentes de información extremadamente valiosa, permitiendo obtener inteligencia acerca de mecanismos de ataque, nuevas formas de propagación de *malware*, métodos de envío de correo *spam* (4), (5), etc.

Entre los distintos tipos de datos que podemos identificar en estas fuentes de eventos tenemos:

- Direcciones IP: Las direcciones IP aparecen como uno de los tipos de datos más frecuentes, ya que son un factor común a la mayoría de los eventos de seguridad.
- Encabezados de correo electrónico: Los mensajes de correo electrónico contienen *encabezados* (headers) que contienen información útil para estudiar el comportamiento de, por ejemplo, los envíos de correo no solicitado.
- Flujos de tráfico: Un flujo de tráfico es un conjunto de paquetes de red intercambiados entre dos puntos finales de comunicación. Cuando se envía un correo electrónico o se copia un archivo por red, la información se debe

paquetizar y estos paquetes fluyen entre dos procesos (los puntos finales de la comunicación), constituyendo así un *flujo.*

La forma más común en que un sensor almacena su información es lo que se conoce usualmente como archivo de *log*. Un archivo de log se puede abstraer como una serie temporal de líneas de texto.

Si bien existen algunos estándares de facto para archivos de log en algunos dominios específicos como el *Extended Log File Format* (6) del World Wide Web Consortium, en la mayoría de los casos la estructura de los mismos es específica de cada sensor y aplicación.

2.2 Necesidad de modelos de información para almacenamiento y transmisión

Si algo tienen en común todas estas fuentes de información es su heterogeneidad y falta de estructura uniforme. Por ello hace falta elaborar *modelos de información* que permitan almacenar y transmitir esta información con eficiencia y expresividad.

Existen múltiples aproximaciones a este problema, en su mayoría diseñadas para dominios de problema específicos. Entre ellos podemos mencionar a CVE *(Common Vulnerabilities and Exposures)* (7) (8), OVAL (*Open Vulnerability and Assessment Language)* (9) (10)

En particular, el *Incident Object Description Exchange Format* (IODEF) (11) es un modelo de información basado en XML definido por el IETF (*Internet Engineering Task Force).* De acuerdo a (11), IODEF es un formato diseñado para representar información de seguridad informática a intercambiar entre CSIRTs. Su principal objetivo es proveer a los grupos de respuesta a incidentes con información *normalizada* y *procesable automáticamente*.

IODEF define un esquema de documento XML y diferentes *clases* de información, como ser la clases "*Incident" (*Incidente), "*Contact"*(Contacto) y "*Report Time"* (Referencia temporal del reporte). IODEF es además un formato *extensible* y el mismo provee mecanismos para enriquecer el modelo.

IODEF no define como deberán lograrse las propiedades de confidencialidad, integridad y autenticidad, (11) explícitamente menciona que estas propiedades deberán ser garantizadas por los mecanismos de transporte o almacenamiento seleccionados.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This example demonstrates a report for a very old worm (Code Red) -->
<IODEF-Document version="1.00" lang="en"
  xmlns="urn:ietf:params:xml:ns:iodef-1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:schema:iodef-1.0">
  <Incident purpose="reporting">
    <IncidentID name="csirt.example.com">189493</IncidentID>
    <ReportTime>2001-09-13T23:19:24+00:00</ReportTime>
    <Description>Host sending out Code Red probes</Description>
    <!-- An administrative privilege was attempted, but failed -->
    <Assessment>
      <Impact completion="failed" type="admin"/>
    </Assessment>
    <Contact role="creator" type="organization">
      <ContactName>Example.com CSIRT</ContactName>
      <RegistryHandle registry="arin">example-com</RegistryHandle>
      <Email>contact@csirt.example.com</Email>
    </Contact>
```

Fig. 1. Ejemplo de un documento IODEF tomado de la RFC 5070 (11)

2.3 Normalización de eventos

La estructura de los archivos de log y en general de la información obtenida de las diferentes fuentes de información de seguridad posee estructuras dispares, con diferentes campos de información ocupando diferentes lugares, posiciones o con diferentes formatos.

Por ello es necesario *normalizar* esta información obteniendo los campos de información relevantes para el modelo de información. El proceso de *normalización* es que nos permitirá obtener por ejemplo direcciones IP, o direcciones de correo electrónico a partir de archivos de log de estructuras dispares para almacenarlos o transmitirlos de acuerdo a un modelo de información.

2.4 Sanitización de la información de seguridad

Uno de los principales obstáculos para la automatización del flujo de información de seguridad a través de fronteras administrativas de diferentes organizaciones es el temor de que en el proceso de divulgación de información se produzcan fugas de información considerada sensible.

Entre estos temores es posible distinguir dos casos, a saber:

- La divulgación de datos sobre infraestructura crítica (servidores de DNS, servidores de correo internos, etc.) o información sobre sistemas comprometidos o vulnerables de la organización. En este caso, el temor principal es que de caer esta información en manos de una entidad maliciosa, la misma pueda ser utilizada para realizar ataques dirigidos con mayor posibilidad de éxito.

- La divulgación de información personal que pueda incumplir alguna normativa de protección de la privacidad ya sea de la organización que divulga los datos o del país donde esta se encuentra, o que pueda comprometer información personal. El temor principal en este caso, además de posibles consecuencias legales es el de esta información personal pueda ser utilizada por una entidad maliciosa para realizar ataques dirigidos de *phishing* (12) o similares intentos de fraude.

En todo proceso de divulgación de información se hace necesario entonces *sanitizar* la información antes de transmitirla, ya sea para ocultar información de infraestructura o para anonimizar información personal. El trabajo de Bishop y Crawford (13) presenta los principales problemas que ocurren al sanitizar información de capa de red; problemas que se pueden generalizar a otros tipos de información.

La técnica clásica de sanitización es la conocida como "*black marker"*[1] (marcador negro): la información considerada sensible bajo algún criterio es suprimida de la salida cambiándola por una etiqueta fija, por ejemplo el mensaje "IP 192.168.1.10" se transforma en "IP X.X.X.X".

El problema principal con esta técnica es que elimina información sin preservar ninguna de las características del conjunto de datos original. En (14) se discute el impacto de la sanitización de flujos de eventos de seguridad en un entorno cooperativo de detección de intrusos.

En (13) se analiza el problema de sanitizar eventos de seguridad conteniendo direcciones IP como un problema de *mapeo* entre dos espacios a través de sustituciones guiadas por funciones o por tablas. En este trabajo también se establece que para que el analista pueda continuar obteniendo conclusiones útiles estos mapeos deben preservar la mayor cantidad de propiedades de los datos originales, como ser distancias, localidad, etc.

3 Referencias en la literatura

3.1 Sistemas de Gestión de Información de Seguridad (SIMs)

Los Sistemas de Gestión de Información o *Security Information Management Systems*, de los cuales el ejemplo mejor conocido es OSSIM (15), presentan en general arquitecturas distribuidas, basadas en el concepto de agregar información proveniente de múltiples sensores en una base de datos única para luego reprocesarla para obtener correlaciones y realizar detección de intrusos.

OSSIM presenta una arquitectura donde se identifican componentes de captura de eventos de seguridad, almacenamiento de información y un fuerte énfasis en la correlación de eventos para detección de intrusiones.

[1] El "*black marker*" es una analogía con las técnicas de eliminación de información de documentos, en los cuales ciertas líneas son borradas tapándolas con tinta negra.

El objetivo principal de estos sistemas es incrementar la capacidad de detección prematura de intrusiones y la capacidad de respuesta ante las mismas, presentando a un operador con información sumarizada y aplicando métodos de correlación de eventos.

Si bien muchos de estos sistemas incluyen funcionalidades de *exportar* información de seguridad en diferentes formatos (IODEF por ejemplo), la comunicación automatizada a través de fronteras administrativas no es su objetivo principal.

3.2 AirCERT

AirCERT (16) fue un proyecto llevado adelante por el CERT/CC (*CERT Coordination Center*) de la Universidad de Carnegie-Mellon, cuyo objetivo explícitamente consistió en implementar un sistema *escalable y distribuido* con el objetivo de *compartir información de seguridad a través de dominios administrativos.*

AirCERT identifica varios componentes, incluyendo *sensores, normalizadores, colectores, herramientas de análisis* y *herramientas para representar información.* En el sitio web de AirCERT se encuentran implementaciones de algunos de estos componentes, aunque el proyecto AirCERT parece estar abandonado, no habiéndose encontrado actualizaciones posteriores a 2003-2004, y consultas realizadas por el autor en diferentes listas de correo no recibieron respuesta.

3.3 HIDEF / HIDEAS

HIDEF (*Data Exchange Format for Information Collected in Honeypots and Honeynets*) (17) propone un formato basado en XML como una extension de IODEF que se propone resolver el problema de falta de interoperabilidad entre diferentes implementaciones de *honeypots* y *honeynets*.

HIDEAS (*Honeypots Information and Data Exchange and Analysis System*) (18) es una continuación de este mismo trabajo en el que se propone una arquitectura que intercabiando información en formatos como IODEF, IDMEF o HIDEAS habilita la correlación y posterior análisis y procesamiento de estos datos.

HIDEAS plantea que su objeto fundamental es el sensor *honeypot.* A partir de esta definición, se plantean requisitos funcionales que se plantean para HIDEAS se encuentran el almacenamiento y comunicación de información de seguridad de acuerdo, el soporte para sanitización de eventos colectados por honeypots. Se define que la seguridad en el transporte debe ser proporcionada por protocolos externos.

La hipótesis de que el sensor fundamental es el honeypot permite realizar hipótesis sobre los tipos de datos contenidos en los eventos capturados, como ser que en su mayoría serán eventos de red (direcciones IP, puertos TCP, etc.)

4 FACTOIDS

De acuerdo a lo expresado en la introducción, la visión de este trabajo es proponer una arquitectura general que brinde los servicios básicos identificados como requerimientos, es decir que permita recolectar datos de sensores heterogéneos, procesar estos datos para normalizarlos de acuerdo con un cierto modelo de información para luego compartirlos con una federación de sistemas de arquitectura similar, modelo que hemos llamado FACTOIDS.

La arquitectura propuesta incluye conceptos de OSSIM, AirCERT y HIDEAS. En particular, puede verse como una generalización de este último para incluir otras fuentes de información además del honeypot.

Los conceptos fundamentales que subyacen a la arquitectura son:

- La capacidad de *recolectar datos de sensores heterogéneos*, incluyendo pero no limitándose a honeypots, fuentes *Netflow*, archivos de log de firewalls y routers, etc.
- El *almacenamiento normalizado de estos datos* en alguna base de datos que soporte un *modelo de información bien especificado* y compatible con modelos estándar.
- El *soporte para realizar correlación de eventos y detección de intrusiones operando sobre datos propios y recibidos de otros sistemas similares.*
- El soporte para *sanitización de eventos* con la incorporación de *funciones de mapeo* adecuadas para los tipos de datos que el sistema maneja.
- La especificación de *interfaces bien definidas* con semánticas claras que permitan la interconexión sencilla de sistemas similares.
- El soporte para realizar *enriquecimiento contextual* de los eventos colectados. Existen casos donde los eventos colectados pueden enriquecerse con información adicional a la enviada por el sensor mismo, pero que debe obtenerse por otras vías más lentas o costosas que imposibiliten el hacerlo en línea.
  Un ejemplo de esto es el agregar a cada dirección IP mencionada en un evento colectado la información de su mapeo reverso de DNS y su sistema autónomo. Ambos datos pueden obtenerse haciendo consultas a bases de datos en Internet, pero ambas pueden resultar uno o más órdenes de magnitud más lentas que la normalización y almacenamiento de eventos.
- El soporte para la *realización de consultas* por parte de sistemas externos apropiadamente autorizados.
  En la mayoría de las arquitecturas similares encontradas en la literatura, el énfasis en la comunicación con sistemas externos está puesto en la *exportación* de eventos. Creemos que en muchos casos puede resultar más eficiente y ágil el permitir que los sistemas externos adecuadamente autorizados realicen *consultas* sobre los repositorios locales de información.

En el resto de esta sección presentaremos las interfaces y componentes de la arquitectura FACTOIDS.

4.1 Interfaces

4.1.1 Interfaz "A" – Recolección de datos de sensores

La interfaz "A" representa el punto de contacto entre sensores y la base de datos de eventos normalizada. Esta interfaz es de un lado dependiente del tipo de sensor, mientras que en su lado interno conecta a los sensores con la base de datos normalizada aplicando el modelo de datos identificado.

Por ello, la interfaz A se especializa según los diferentes sensores de los que se busque recibir datos. En el ejemplo del a figura tenemos:

1. **A1** : sistema de detección de intrusos
2. **A2** : firewall / router
3. **A3** : Honeypot
4. **A4** : Fuente Netflow

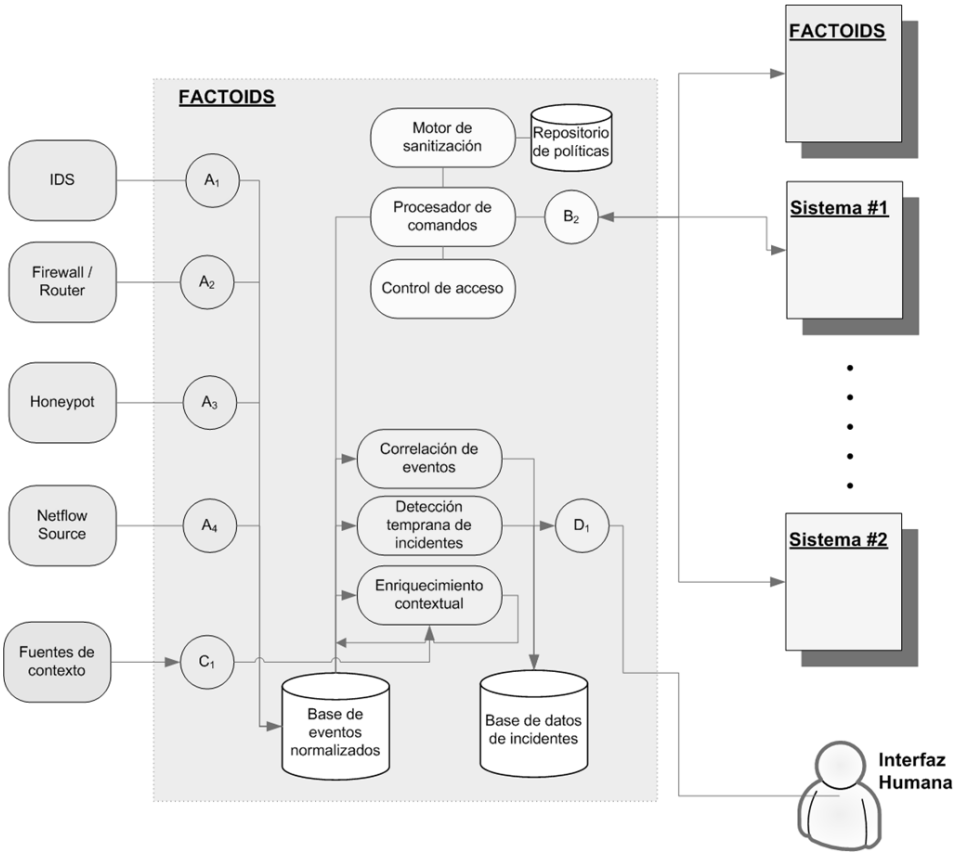

Fig. 2. FACTOIDS

4.1.2 Interfaz "B" – Intercambio de información con sistemas externos al grupo

La interfaz "B" es la interfaz que provee el punto de contacto entre fronteras administrativas, posiblemente con otros sistemas de arquitectura similar

Esta interfaz se alimenta por un lado de los datos ya normalizados y deberá implementar un conjunto de operaciones que permitan la realización consultas por parte de sistemas similares apropiadamente autorizados para acceder al repositorio local de información.

Es a este flujo saliente de información al que se le deben aplicar las políticas de seguridad de la información de la organización madre, las que se almacenarán en el *repositorio de políticas.*

4.1.3 Interfaz "C" – Interfaz para enriquecimiento contextual de eventos

Esta interfaz provee el punto de contacto con bases de datos de información externa como ser los mapeos IP-sistema autónomo (ASN) (19) o búsquedas reversas en DNS. Como se mencionó anteriormente, no es posible realizar estas búsquedas a nivel de las interfaces "A" por motivos de performance, y además es posible que no tenga sentido realizarlas para *todos* los eventos colectados si no para un subconjunto de los mismos.

4.1.4 Interfaz "D" – Interfaz humana

Interfaz con operadores humanos, donde se prestan funcionalidades como:

- Interfaz de administración
- Detección temprana de incidentes, mediante el motor de correlación
- Notificación de alarmas por consola u otras vías (e-mail, etc.)

4.2 Componentes

FACTOIDS define tres grupos de componentes básicos, a saber: Componentes de *Almacenamiento,* componentes de *Reprocesamiento* y componentes de *Control.*

4.2.1 Componentes de almacenamiento

El sistema necesita almacenar la información recolectada de los sensores, así como también permitir la operación de otros componentes que procesen y enriquezcan esta información.

La naturaleza heterogénea y el volumen de la información recolectada a partir de los sensores, así como la necesidad de procesarla con eficiencia, hace que los componentes de almacenamiento sean uno de los puntos más críticos del sistema.

Para este componente se debe definir un *modelo de información,* que estandarice la representación de los diferentes tipos de eventos enviados por los sensores.

4.2.2 Componentes de control

Los componentes de control agrupan las funcionalidades de aplicación de políticas de seguridad (incluyendo un repositorio adecuado para el almacenamiento de estas políticas), el procesamiento de comandos y respuestas recibidos de sistemas externos y de los operadores a través de la interfaz "B".

El motor de políticas en particular deberá aplicar las políticas de sanitización definidas por la organización, para ello deberá contar con un *repositorio de políticas* y acceso a un conjunto de *funciones de mapeo* que le permitan implementar estas políticas para los diferentes tipos de datos y eventos colectados.

Los componentes de control deberán incluir módulos de *control de acceso* que permitan definir privilegios y establecer las relaciones de confianza con otros dominios administrativos.

4.2.3 Componentes de reprocesamiento

Los componentes de reprocesamiento implementan funcionalidades de reprocesamiento de los eventos recolectados a través de la interfaz "A", incluyendo por ejemplo *motores de correlación* que permitan realizar detección temprana de incidentes, procesos de archivo y eliminación de información antigua.

5 Conclusiones y trabajo a futuro

El escenario de intercambio de información de seguridad entre CSIRTs, a través de fronteras administrativas encierra la promesa de permitir a investigadores, personal de respuesta a incidentes y otros grupos como ser equipos policiales el acceder a vastos repositorios de información.

Para obtener el máximo valor de esta información con agilidad y eficiencia, es necesario resolver ciertos problemas que incluyen el almacenamiento normalizado, transmisión en formatos estándar y acceso a través de interfaces bien definidas.

Si bien se han propuesto soluciones y arquitecturas con fines relativamente similares, creemos que es necesario definir una arquitectura lo suficientemente general, basada en todo el trabajo previo existente, que permita acomodar las necesidades de un público amplio.

A diferencia de AirCERT, el objetivo de este trabajo no es el de proveer una implementación exhaustiva de todos los componentes si no el de definir un marco que permita incorporar diferentes propuestas de modelos de información, funciones de sanitización, motores de correlación y otros eventos.

Como generalización de HIDEAS, FACTOIDS propone el incorporar fuentes de información adicionales a los honeypots, así como poner un énfasis en el control de acceso, privilegios de acceso y sanitización de eventos.

Como concepto adicional, FACTOIDS incorpora la noción de la bidireccionalidad de la interfaz "B" (interfaz con otros dominios administrativos), a través de la cual no solo sería posible exportar información sino también realizar consultas.

FACTOIDS es un trabajo en progreso y por ello hay varios aspectos todavía no definidos completamente. Una vez establecidas todas las interfaces del sistema, se pasará a definir detalladamente la semántica de las diferentes interfaces y continuar con la implementación de un prototipo basado en IODEF como modelo de información y proporcionando funciones de sanitización para los eventos de red.

6 Referencias

1. **Spitzner, Lance.** Honeypots - Definitions and Value of Honeypots. *Sitio web "Tracking Hackers".* [Online] Marzo 2003. [Cited: Marzo 15, 2009.] http://www.tracking-hackers.com/papers/honeypots.html.

2. **Cisco Systems, Inc.** RFC 3954 - Cisco Systems NetFlow Services Export Version 9. *Internet Engineering Task Force.* [Online] 2009. [Cited: marzo 15, 2009.] http://www.ietf.org/rfc/rfc3954.txt.

3. **The Honeynet Project.** Know Your Enemy: Honeynets. *The Honeynet Project.* [Online] Mayo 31, 2006. [Cited: Marzo 15, 2009.] http://old.honeynet.org/papers/honeynet/.

4. **CERT.br: Computer Emergency Response Team - Brazil.** Spam Metrics: The SpamPots Project. *Sitio web CERT.br.* [Online] 2007. [Cited: marzo 15, 2009.] http://www.cert.br/docs/palestras/certbr-icann2007.pdf.

5. **Hoepers, Cristine and Steding-Jessen, Klaus.** New Developments in the SpamPots Project. *Sitio web CERT.br.* [Online] 2008. [Cited: marzo 15, 2009.] http://www.cert.br/docs/palestras/certbr-national-csirts-meeting2008.pdf.

6. **World Wide Web Consortium.** Extended Log File Format; W3C Working Draft WD-logfile-960323. *W3C Web Site.* [Online] 1996. [Cited: marzo 15, 2009.] http://www.w3.org/TR/WD-logfile.html.

7. **The Mitre Corporation.** *Making Security Measurable.* [Online] enero 14, 2009. [Cited: marzo 15, 2009.] http://measurablesecurity.mitre.org/.

8. *Towards a Common Enumeration of Vulnerabilities.* **Mann, David E. and Christey, Steven M.** West Lafayette, Indiana, USA : s.n., 1999. 2nd Workshop on Research with Security Vulnerability Databases, Purdue University.

9. **The Mitre Corporation.** About OVAL. *Mitre's OVAL Website.* [Online] marzo 6, 2008. [Cited: marzo 15, 2009.] http://oval.mitre.org/oval/about/index.html.

10. —. An Introduction to the OVAL Language. *Mitre's OVAL Website.* [Online] 2006. [Cited: marzo 15, 2009.] http://oval.mitre.org/oval/documents/docs-06/an_introduction_to_the_oval_language.pdf.

11. **Danyliw, R., Meijer, J. and Demchenko, Y.** RFC 5070 - The Incident Object Description Exchange Format. *Internet Engineering Task Force (IETF).* [Online] Diciembre 2007. [Cited: marzo 15, 2009.]

12. *Why phishing works.* **Dhamija, Rachna, Tygar, J. D. and Hearst, M.** Montréal, Québec, Canada : ACM New York, NY, USA, 2006. ISBN:1-59593-372-7 .

13. *Some Problems in Sanitizing Network Data.* **Bishop, M., et al., et al.** s.l. : IEEE Computer Society Washington, DC, USA, 2006. Proceedings of the 15th IEEE

International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. ISBN ~ ISSN:1524-4547 , 0-7695-2623-3 .

14. *Impact of Sanitized Message Flows in a Cooperative Intrusion Warning System.* **Tolle, J. Jahnke, M. Felde, N.G. Martini, P.** Washington DC : IEEE, 2006. Military Communications Conference, 2006. MILCOM 2006. IEEE. ISBN: 1-4244-0617-X.

15. **AlienVault.** OSSIM: Open Source Security Information Management. *Sitio web de OSSIM.* [Online] [Cited: Abril 14, 2009.] http://www.ossim.net/.

16. **CERT Coordination Center.** AirCERT. *Sitio web de AirCERT.* [Online] [Cited: marzo 15, 2009.] http://aircert.sourceforge.net/.

17. *HIDEF: a Data Exchange Format for Information Collected in Honeypots and Honeynets.* **Hoepers, C, Vijaykumar, NL and Montes, A.** 2007.

18. **Hoepers, Cristine.** *Projeto E Implementacao De Uma Infra-Estrutura Para Troca E Analise De Informacoes De Honeypots E Honeynets.* INPE - Instituto Nacional de Pesquisas Espaciais. Sao Paulo : INPE, 2008. Tesis doctoral. INPE-0000-TDI/000.

19. **Team Cymru.** IP to ASN Mapping. *Team Cymru Website.* [Online] 2009. [Cited: marzo 15, 2009.] http://www.team-cymru.org/Services/ip-to-asn.html.

20. **Debar, H., Curry, D. and B.Feinstein.** RFC 4765. *The Intrussion Detection Message Exchange Format (IDMEF).* [Online] marzo 2007. [Cited: marzo 15, 2009.] http://www.ietf.org/rfc/rfc4765.txt.