Universidad de las Ciencias Informáticas

Facultad 2

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

"SOLUCIÓN INFORMÁTICA PARA EL DIAGNÓSTICO DE LA SEGURIDAD DESDE LA PLATAFORMA OSSIM, TENIENDO EN CUENTA LA INFORMACIÓN QUE GESTIONA SASGBD"

Autor: Elvis Javier Rodríguez Soto

Tutores: Msc. Yasser Azán Basallo

Ing. Fernando Ricardo Romero

La Habana, julio de 2016

"Año 58 de la Revolución"

Pensamiento

Declaración de autoría

Yo Elvis Javier Rodríguez Soto declaro ser el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a hacer el uso que estime pertinente con este trabajo.

Para que así conste firmo la presente a los ____ días del mes de julio del año 2016.

Elvis Javier Rodríguez Soto

Firma del Autor

Msc. Yasser Azán Basallo

Firma del Primer Tutor

Ing. Fernando Ricardo Romero

Firma del Segundo Tutor

Datos de contacto

Tutor: Msc. Yasser Azán Basallo (yazan@uci.cu): graduado de Ingeniero en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Pertenece al Centro de Telemática (TLM). Es profesor del Departamento de Práctica Profesional en la Facultad 2.

Tutor: Ing. Fernando Ricardo Romero (fricardo@uci.cu): graduado de Ingeniero en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Pertenece al Centro de Telemática (TLM).

Agradecimientos

Debo agradecer de forma muy especial al tribunal que evalúa la propuesta de solución. Con énfasis agradezco a la profe Idalis por ser casi una tutora para este trabajo de diploma, ya que por la paciencia que han demostrado poseer son responsables de la realización de mi acto de defensa. Son muy merecedores de mi reconocimiento mis tutores Yasser y Fernando que con su trabajo me han respaldado en momentos en los que pensé que no podía y quería que la tierra me tragara, momentos no tan lejanos como yo hubiera querido. A los trabajadores de ETECSA les doy las gracias por todo el apoyo brindado durante el desarrollo. Agradezco mucho a mi oponente y al técnico de seguridad informática Dennis Barrera Pérez que gracias a su paciencia y el sacrificio de su tiempo libre presento el trabajo.

También debo agradecer a mis compañeros de estudios, los actuales y los de la etapa de la FRA de Artemisa. A aquellos que no pudieron llegar les digo: “Yo llegué y en gran parte se los debo, a todos los llevo junto a mí siempre”. A los que ya se graduaron les digo: “Ya llegué, al final si pude” y a los que faltan por graduarse les digo: “Si yo pude ustedes también”. No haré mención de nombres en el documento por temor a olvidar a alguien.

Debo agradecer también y sería muy injusto de mi parte no hacerlo a mis profes e instructoras de Artemisa, de ahí en especial al profe Javier Gilberto y a mi vicedecana de formación Juana Elena, que dos veces me salvó de la baja.

De aquí de la UCI en cuanto a profesores debo agradecer a todos los que han contribuido a mi formación como profesional, con énfasis en el profe Alejandro, el profe Hugo y la profe Madelin que tanto se han preocupado por mí. También a mis instructoras de la residencia, no diré nombres porque son unas cuantas a las que debo agradecer.

Agradezco mucho el apoyo que me han dado muchas de mis amistades del barrio, del politécnico y sus familias, que casi son la mía.

Por último, debo agradecer a mi familia y a la de mi novia que es como la mía, pues me han dado buenos y malos ejemplos de cómo debo ser con respecto a la vida. A ellos les debo principalmente quién soy. No se me olvida agradecer a Arianna mi alegría, sustento y paño de lágrimas en los buenos y malos momentos, a ti más que a nadie le debo llegar aquí.

Dedicatoria

Dedico este trabajo a todo aquel que vio mi potencial para llegar aquí y me ayudó, aunque yo mismo no lo creyera posible. También a todo aquel que es capaz de luchar por sus objetivos y no rendirse aunque las situaciones adversas lo apremien. Por último pero no menos importante a todos mis amigos y familia, esos que están y esos que ya no están.

Resumen

Con el presente trabajo de investigación: Solución informática para el diagnóstico de la seguridad desde la plataforma OSSIM, teniendo en cuenta la información que gestiona SASGBD, se detallan las opciones de recepción de los datos de SASGBD por parte de OSSIM. El tema de la investigación está enfocado en la centralización de la información de los sensores en el sistema SIEM de AlienVault.

En la investigación, se propone el desarrollo de una solución que actúa como interfaz de transferencia de datos de la herramienta SASGBD para el diagnóstico de la seguridad desde la plataforma OSSIM. Esto se logra a través del nivel de integración Punto a Punto entre las soluciones.

De esta manera se creó una solución informática integrada por una aplicación y un plugin para OSSIM. Esto permite la correcta comunicación entre las plataformas SASGBD y OSSIM en ETECSA, aportando una vía de comunicación. La comunicación de las herramientas y permite centralizar los reportes de parámetros de sistemas gestores de bases de datos auditados que poseen configuraciones incorrectas. Este trabajo utilizó la metodología de desarrollo Programación Extrema o XP, Framework-OSSIM como Framework de desarrollo, PostgreSQL y MySQL como Sistemas Gestores de Bases de Datos.

Palabras claves: centralización, integrar, interfaz, plugin, sensores.

Índice

Introducción

El tratamiento y la transmisión de los datos recogidos desde fuentes tecnológicas ocupan un espacio importante en las empresas sin importar sus dimensiones u objetivos. Esto ha provocado que las tecnologías evolucionen de manera vertiginosa en conjunto a las técnicas de recogida, selección, envío y conservación de la información. En el campo de las redes informáticas se ha evolucionado desde los comienzos de la transmisión de datos donde se transmitía en forma de texto plano hasta la actualidad, con los protocolos estandarizados de transmisión y cifrado de datos.

Para asegurar los datos recolectados en el mercado existen aplicaciones de seguridad informática con fines específicos como son los cortafuegos (firewall), sistemas auditores, sistemas de detección de intrusos (IDS) y gestores de eventos de seguridad informática (SIEM) por sus siglas en inglés respectivamente. Los sistemas SIEM actúan como un repositorio central registrando eventos de seguridad generados en la red. Estos sistemas le permiten al administrador de red, escoger los eventos específicos de interés. (1)

Para la mayoría de las entidades a escala mundial es muy complejo el hecho de desarrollar una de estas herramientas, no solo por carencias de recursos financieros, sino que además la interacción con las mismas es compleja en algunos casos ya que generan numerosas notificaciones que deben ser supervisadas por especialistas. Las empresas y las comunidades de desarrollo de software de seguridad, ofrecen soluciones con el propósito de manipular la información digital. Ejemplos de estas herramientas son AlienVault OSSIM, Kaspersky Antivirus System, Trustwave SIEM, ManageEngine EventLog Analyzer.

En el ámbito nacional, la Empresa de Telecomunicaciones de Cuba S.A (ETECSA) está enmarcada en un complejo proceso de perfeccionamiento debido al sustancial aumento del cúmulo de datos generados por sus actividades, ejemplo de las cuales son: el aumento de clientes de líneas telefónicas fijas, el servicio de correo electrónico Nauta y la navegación por internet. Los datos a procesar y almacenar exigen la utilización de las tecnologías de la informática y las comunicaciones para su tratamiento, conservación y aseguramiento.

ETECSA conserva mayormente su información en bases de datos y en esos momentos no disponía de una herramienta de auditorías para bases de datos que detectara si estas presentaban vulnerabilidades. Esta situación atentaba contra la seguridad de los datos de sus clientes, por lo que la seguridad de sus Sistemas Gestores de Bases de Datos (SGBD) tomó un valor estratégico para el futuro de la entidad.

Bajo estas circunstancias la entidad decidió adquirir una solución informática capaz de auditar sus SGBD. Con este objetivo solicitó a la Universidad de las Ciencias Informáticas (UCI) el desarrollo de una solución informática que resolviera sus necesidades de auditoría de gestores de bases de datos en busca de posibles vulnerabilidades. Como institución de enseñanza de nuevo tipo, la UCI está estructurada por centros docentes y productivos. El centro Telemática (TLM) perteneciente a la Universidad desarrolló como resultado a la petición realizada por ETECSA la solución denominada Sistema para la realización de Auditorías a Sistemas Gestores de Bases de Datos (SASGBD).

Esta aplicación, siendo utilizada por ETECSA dio solución a las necesidades que originaron su desarrollo. El auditor de seguridad informática de dicha empresa, dispone para su trabajo de varias herramientas informáticas, siendo AlienVault OSSIM y SASGBD partes clave del manejo de la seguridad de la estructura digital de la entidad. Estas aplicaciones controlan eficientemente el entorno de trabajo digital de la entidad, generando de forma independiente informes sobre diferentes aspectos de seguridad.

Es de interés del auditor centralizar ambos informes, para acceder a los reportes que contengan información sobre el estado de los SGBD en conjunto al resto de los reportes de seguridad generados por OSSIM. A pesar de que ambas plataformas generan reportes en formato PDF no cuentan con una estructura de datos similar, lo que puede acarrear que la misma vulnerabilidad pueda estar representada en ambas plataformas pero estar descrita en formatos diferentes. Cada reporte generado debe ser revisado de forma individual por el auditor y hacer un reconocimiento de cada valor de las vulnerabilidades detectadas.

Los reportes generados por ambas soluciones informáticas, pueden contener elevados volúmenes de información, lo que trae consigo demoras por parte del auditor a la hora de realizar el análisis. Para la realización de un correcto análisis de seguridad usando ambas herramientas, se debe conocer cómo representan la información cada una de ellas.

Por lo antes expuesto se identifica como **problema** a resolver:

¿Cómo centralizar las vulnerabilidades detectadas en el OSSIM en el proceso de diagnóstico de seguridad informática, teniendo en cuenta la información que gestiona la aplicación SASGBD de la Empresa de Telecomunicaciones de Cuba S.A.?

El problema planteado se enmarca en el **objeto de estudio**: El proceso de diagnóstico de seguridad informática, tomando como **objetivo general** desarrollar una solución informática para monitorizar el diagnóstico de la seguridad desde la plataforma OSSIM, teniendo en cuenta la información que gestiona la aplicación SASGBD de la Empresa de Telecomunicaciones de Cuba S.A. Enmarcando el **campo de acción** en el proceso de diagnóstico de seguridad informática en la empresa ETECSA.

Para cumplir el objetivo general de la investigación se definen las siguientes tareas:

- Analizar soluciones existentes, en el ámbito nacional e internacional, relacionadas a la integración de los sistemas informáticos, estableciendo similitudes con la investigación en curso para la definición de requerimientos necesarios para la propuesta de solución.
- Analizar la aplicación SASGBD y la plataforma OSSIM, identificando las vulnerabilidades, para ser incluidas en la propuesta de solución.
- Asimilar herramientas, tecnologías y metodología, utilizadas en el desarrollo de la aplicación SASGBD y la plataforma OSSIM para el desarrollo de propuesta de solución.
- Desarrollar la propuesta de solución, que supervise el diagnóstico de la seguridad desde la plataforma OSSIM, teniendo en cuenta la información gestionada por la aplicación SASGBD haciendo uso de técnicas de integración.
- Desarrollar las pruebas necesarias para corroborar el correcto funcionamiento de la solución propuesta.

Para apoyar el desarrollo de la investigación se emplean los siguientes **métodos de investigación científica**:

Métodos Teóricos:

- Analítico – Sintético: Empleado en el análisis de materiales provenientes de las fuentes de datos y en la conformación del estado del arte, en el que se definen conceptos relacionados a la investigación. Permite sintetizar dentro del conjunto de funcionalidades de un agente de OSSIM aquellas que sean aplicables a la investigación, como por ejemplo la estructura que debe tener un plugin para la plataforma, la localización de sus ficheros dentro del servidor, el fichero tomado depósito para recolectar los logs. Además de permitir la selección de la información a enviar desde SASGBD.
- Modelación: Utilizado con el objetivo de plasmar de forma simple el proceso que realizará el auditor de seguridad informática de ETECSA a través de un diagrama de procesos de negocio que apoya la propuesta de solución.
- Histórico-Lógico: Se utilizó para realizar un análisis del estado del estado del arte de las principales aplicaciones orientadas a la integración y las que son parte de la plataforma OSSIM, relacionadas con el campo de acción.

Métodos Empíricos:

- Experimental: Empleado con el objetivo de obtener los efectos de la interacción de SASGBD y OSSIM con la solución. Esto se realiza mediante la monitorización del

rendimiento y estabilidad del servidor de bases de datos de SASGBD al ser consultado y el rendimiento del servidor de OSSIM al momento de incorporar los datos provenientes de la consulta realizada al servidor de bases de datos de SASGBD y mostrarlos en la paleta de mando.

- Observación: Utilizado para obtener y analizar el comportamiento normal de las herramientas SASGBD y OSSIM, mediante la percepción directa del funcionamiento de ambas sin existir interacción con la propuesta de solución. Después es empleado en analizar el nivel de influencia de la solución en el funcionamiento de las partes involucradas, midiendo la velocidad de respuesta de los servidores involucrados durante el uso normal y durante el período de mayor actividad.

La investigación se divide en cuatro capítulos, los cuales estarán estructurados de la siguiente forma:

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA: El capítulo contiene el marco conceptual que muestra las principales definiciones utilizadas en la investigación. Se incluye un estudio del estado del arte de posibles soluciones de integración a utilizar y los plugins existentes para el SIEM AlienVault OSSIM en la actualidad, a los que se le realizan estudios con el objetivo de seleccionar las características que se consideren resuelvan el problema y que serán parte de la propuesta de solución. En este capítulo también se hace referencia a la metodología de desarrollo, las herramientas y lenguajes de programación que apoyan el desarrollo de la solución.

CAPÍTULO 2. PLANIFICACIÓN Y CARACTERÍSTICAS DEL SISTEMA: En el presente capítulo se abordan los temas relacionados con las fases del trabajo de la metodología de desarrollo XP, se elabora una propuesta del sistema a desarrollar y se exponen las características del mismo para un mejor entendimiento en su desarrollo. Se confeccionan las Historias de Usuarios (HU) para cada iteración definida, con vista a documentar los procedimientos y técnicas empleados, proporcionando una mejor visión sobre lo que el cliente desea y además se realiza un análisis de la estimación del esfuerzo por cada historia de usuario.

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA: Se describe la arquitectura de AlienVault OSSIM y cómo se enmarca dentro de la misma la propuesta de solución. Muestra el patrón arquitectónico y el diseño que se utiliza en el proceso de desarrollo de la solución. Se diseña la estructura del plugin que formará parte de la propuesta de solución. También, se crean las tarjetas CRC y las tareas de la ingeniería para desglosar las actividades comprendidas en cada Historia de Usuario. Por último, en este capítulo se definen los estándares de codificación utilizados en la implementación del plugins.

CAPÍTULO 4. PRUEBAS: Se realizan las pruebas necesarias para comprobar que los plugins desarrollados cumplen con el objetivo por el cual fueron creados. Ellas son: pruebas unitarias y

pruebas de aceptación, para cada una se especifica la iteración en las que se fue realizando y los resultados obtenidos luego de aplicarlas.

Capítulo I: Fundamentación teórica

Introducción

El capítulo contiene el marco conceptual que muestra las principales definiciones utilizadas en la investigación, además incluye un estudio del estado del arte de los plugins existentes para el SIEM AlienVault OSSIM en la actualidad, a los que se le realizan estudios con el objetivo de seleccionar las características que se consideren resuelvan el problema y que serán parte de la propuesta de solución. En este capítulo también se hace referencia a la metodología de desarrollo, las herramientas y lenguajes de programación que apoyan el desarrollo de la solución.

1.1 Conceptos fundamentales

A continuación, se citan los conceptos necesarios para la comprensión de los términos técnicos empleados, además de la estructura y funcionamiento de los componentes de AlienVault OSSIM.

Log o registro: Reporte creado a partir de un evento generado por herramientas y dispositivos (2)

SIEM (Security Information Event Management): El acrónimo SIEM se atribuye a los analistas de Gartner Amrit Williams y Nicolett Marcos y se deriva de dos tecnologías independientes, pero complementarias: el Administrador de Eventos de Seguridad (SEM por sus siglas en inglés) y el Administrador de Información de Seguridad (SIM por sus siglas en inglés). Durante la última década, estas dos tecnologías han convergido en una única solución conjunta conocida hoy como SIEM. SEM fue una solución tecnológica que se centró en el seguimiento de eventos de seguridad en tiempo real, así como la correlación y el procesamiento. Estos eventos de seguridad eran típicamente alertas generadas por un dispositivo de seguridad de red, tales como un firewall o un Sistema de Detección de Intrusos (IDS por sus siglas en inglés). SIM, por otra parte, se centró en el análisis histórico de la información del archivo de registro para apoyar las investigaciones forenses y los informes. SIM a menudo analiza los mismos eventos que SEM, pero no lo hace en tiempo real. SIM centraliza el almacenamiento de registros y archivos, búsqueda y análisis de funciones y, sólidas capacidades de presentación de informes. Los sistemas SIEM combinan las capacidades de cada una de estas tecnologías en una única solución, de hecho, las soluciones SIEM actuales con frecuencia incorporan una función de gestión de registros mucho más amplia (1).

AlienVault OSSIM (Open Source Security Information Manager): OSSIM es una distribución de productos open source integrados para construir una infraestructura de monitorización de seguridad. Su objetivo es ofrecer un marco para centralizar, organizar y mejorar las capacidades de detección y visibilidad en la monitorización de eventos de seguridad de la organización.

Es un SIEM desarrollado por Dominique Karg y Julio Casal en el año 2000, que implementa la detección y prevención de intrusiones, y la seguridad de redes en general. Este sistema funciona a partir de múltiples herramientas populares de monitoreo y seguridad de código abierto (Open Source), como Nagios, Snort, y otros. Ofrece grandes capacidades y un alto rendimiento, creando así una inteligencia que traduce, analiza y organiza los datos de una forma única que la mayoría de sistemas SIEM no pueden conseguir, resultando en un diseño que gestiona, organiza y observa riesgos que los administradores pueden apreciar (1).

Correlación: Este se define como un algoritmo que ejecuta una operación de entrada de datos y regresa una salida de datos. El cual trabaja recolectando información y la monitorea de manera parcial, posteriormente este ilumina pequeñas áreas a través del espectro de toda la información que realmente nos interesa (1).

Agente o sensor: Son los encargados de recoger una amplia gama de información sobre su entorno local, procesar esta información y coordinar la detección y respuesta con el resto de la red OSSIM. Los sensores están instalados en los segmentos de red y lugares remotos, inspeccionan todo el tráfico, detectan ataques a través de diversos métodos y recolectan información sobre el tipo y forma de ataque sin afectar al rendimiento de la red (1).

Los sensores son los encargados analizar el tráfico que cursa por la red en tiempo real, recolecta los datos enviados por los dispositivos que conforman la red (realiza tareas de IDS, Escáner de Vulnerabilidades, detección de anormalidades, monitoreo de red, recolección de datos de routers, firewalls, etc.) y los normaliza para luego enviarlos al servidor (SIEM quien se encargará de realizar la clasificación y correlación de eventos) (1)

Plugin para OSSIM: Serie de expresiones regulares que permiten identificar y describir un evento (3).

Para OSSIM existen dos tipos de plugin:

Detectores: Encargados de leer los logs creados por las diferentes herramientas y estandarizarlos para que el agente pueda enviarlos al servidor. Ejemplos típicos de sensores con plugins de tipo detector son Snort, p0f, Arpwatch, Pads, etc (3).

Monitores: reciben pedidos del servidor OSSIM y los envían a la herramienta correspondiente, obtienen la respuesta y le avisan al servidor si la herramienta acepta lo que se le pide. Ejemplos de monitores son el Nmap y tcptrack (3).

Por lo general los plugins utilizados son de tipo detector, los monitores se utilizan para información muy específica y detallada que sea requerida.

1.2 Estudio de soluciones existentes para realizar integración

- Altova MissionKit: Ofrece una sencilla y robusta interfaz gráfica donde puede diseñar asignaciones de datos mediante operaciones de arrastrar y colocar. La interfaz gráfica es compatible con los formatos de datos más utilizados, incluidas las principales bases de datos relacionales. En las asignaciones de datos también puede usar varios orígenes y destinos, transformaciones en cadena y un gran número de funciones de procesamiento y filtrado de datos para que adapte sus proyectos de integración de datos a sus requisitos. Estas herramientas generan código en varios lenguajes libres de derechos de autor, incluido código Java (4).
- LANSA Composer: Herramienta altamente visual y sin código que fue diseñada para el uso de los analistas de negocio para diseñar e implementar soluciones a los problemas de integración. No es necesario escribir código para soluciones que automatizan procesos manuales y elimina el re-tecleo de datos para reducir la cantidad de papel, correo electrónico, fax e interacción humana requerida para completar un proceso de negocio determinado. Por ahora solamente compañías grandes podían pagar el costo y la complejidad del software del proceso de integración. LANSA brinda una solución de integración de procesos de negocio (BPI) para empresas de todo tamaño (5).
- LANSA Integrator: Elimina la necesidad de entrenar el personal en esas tecnologías complejas y provee un ambiente productivo para integrar datos entre diversas plataformas y aplicaciones. Oculta los aspectos técnicos de tareas de integración y brinda la máxima flexibilidad para incorporar esas tareas en sus aplicaciones LANSA, C, RPG y COBOL, en cualquier manera se requiera. También ofrece un asistente de desarrollo de servicios Web para consumir servicios Web externos o internos o exponer código existente LANSA o 3GL como un servicio Web (5).
- Qualisys Software and Technologies: Provee herramientas de integración le permiten interconectar y acceder a dispositivos y sistemas que no son normalmente compatibles. Los drivers que proporcionados permiten acceder a una gama de fuentes de datos para luego poner esos datos a disposición en el formato o protocolo que usted necesita. El aspecto más atendido en el diseño de estos sistemas es su robustez y tolerancia a fallos: han sido diseñados para funcionar constantemente en forma desatendida, recuperándose automáticamente de cualquier fallo (6).
- iWay Software: Ofrece la capacidad de integrar las diversas fuentes de datos e información dentro y fuera de su empresa, en un único marco coherente. Provee una infraestructura de información integrada puede ser compartida por las aplicaciones de misión crítica, portales de información ejecutiva, cuadros de mando, sistemas de informe y sistemas de cadena de suministro automatizados (7).

Conclusión sobre las soluciones de integración consultadas:

Todas las soluciones consultadas son de tipo propietario y de código cerrado, necesitándose la adquisición de las licencias correspondientes a su uso. Aunque ofrecen variantes viables para realizar procesos de integración, no son útiles a este trabajo. Para esta investigación no es posible verificar su funcionamiento interno mediante el examen de su código fuente. Por tanto las posibles alternativas consultadas no son una alternativa aceptable para realizar la integración de SASGBD y OSSIM.

1.3 Análisis de soluciones integradas a AlienVault OSSIM mediante plugins

Snort: Es el más importante IDS Open Source disponible en la actualidad. OSSIM contiene una versión personalizada de esta herramienta y es quien alerta sobre intentos de ataques a la red (1).

OpenVAS: Es la versión GPL (General Public License) de Nessus, una popular herramienta de escaneo de vulnerabilidades Open Source. Esta herramienta se utiliza para proporcionar búsqueda de vulnerabilidades de los recursos de red y añade esta valiosa información a la base de datos de OSSIM. Nessus también es incluido dentro de OSSIM (1).

Ntop: es una popular herramienta Open Source para la monitorización del tráfico de la red. Esta herramienta proporciona información muy valiosa sobre el tráfico en la red, que puede ser utilizada para detectar de una manera proactiva el tráfico anormal o malicioso (1).

Nagios: es una popular herramienta Open Source de monitoreo de dispositivos de red. Es una de las herramientas más complejas, pero le permite al administrador tener una única visión del estado de los hosts de la red. A través del monitoreo de hosts, Nagios puede enviar alertas en caso de fallas y posee una interface web desde donde se puede observar el estado de la red (1).

PADS: El Sistema de Detección Pasiva de Activos (PADS por sus siglas en inglés) es una herramienta única. La herramienta supervisa silenciosamente el tráfico de red, los registros de los host y las actividades de servicio, con el objetivo de detectar anomalías sin generar tráfico de red, realizando un inventario de activos y revisando los servicios que cada cual ejecuta (1).

P0f: La herramienta P0f toma pasivamente las huellas dactilares del sistema operativo (el descubrimiento del tipo de sistema operativo y su versión). Esta herramienta escucha silenciosamente el tráfico de red e identifica los sistemas operativos que se comunican en la red. Esta información resulta útil en el proceso de correlación (1).

OCS-NG (Open Computer and Software Inventory Next Generation): ofrece la capacidad multi-plataforma de gestión de recursos. Esta herramienta permite mantener un inventario actualizado en tiempo real de los dispositivos existentes en la red (1).

OSSEC: Sistema de Detección de Intrusiones de Host (HIDS por sus siglas en inglés) Open Source. Este se encarga de analizar los datos del host y detectar a través de ellos si un host está siendo víctima de algún ataque. OSSEC realiza esta tarea analizando logs, chequeando la integridad de archivos, monitoreando el registro de Windows, detectando rootkits, además de responder y alertar en tiempo real. Esta herramienta también ayuda a proteger al propio OSSIM (1).

OSVDB: La OSVDB (Open Source Vulnerability Database), es la base de datos que mantiene la información actualizada con respecto a las vulnerabilidades del sistema. Esta se ha utilizado por OSSIM durante el proceso de correlación y es quien proporciona un análisis cuando sea necesario (1).

NFSen/NFDump: Visor de flujos de red para la detección de anomalías en la red. Este además permite el procesamiento de Netflow v5, v7 y v9. NFSen proporciona una interfaz gráfica basada en web a NFDump. Ambos NFSen y NFDump se han integrado en OSSIM y han sido modificados para trabajar con las otras herramientas (1).

Inprotect: Interfaz basada en web para Nessus, OpenVAS y NMAP. Inprotect ofrece la posibilidad de definir perfiles de escaneo, programar sondeos, y exportar los resultados del análisis de distintos formatos (1).

OSSIM también tiene otras destacadas herramientas como Arpwatch, el cual es utilizado para detección de anomalías en el uso de direcciones MAC, MACSpade, el cual es un motor de detección de anomalías en paquetes utilizados para obtener conocimiento de ataques sin firma, Tcptrack, que es utilizado para conocer la información de las sesiones, con lo cual puede conceder información útil relativa a los ataques, Osiris, que es un HIDS, y Snare, quien colecciona los logs de sistemas Windows (1).

Resultados sobre los sensores estudiados y la plataforma OSSIM:

- Todos los sensores estudiados utilizan como medios de integración de sus reportes plugins para la plataforma OSSIM.
- Todos los ficheros contienen una estructura visible y accesible para su estudio desde el servidor de la plataforma OSSIM.
- Todos los plugins que colectan logs crean un fichero .conf y crean un fichero que funciona como almacén para concentrar la colecta de logs.
- Todos los plugins constan de cuatro ficheros.

1.4 Metodología, Lenguaje y Herramientas de desarrollo

Metodología de desarrollo de software

Una metodología de desarrollo de software es un conjunto de procedimientos, técnicas, herramientas y un soporte documental que guía a los desarrolladores en la creación de la solución de software.

Para el desarrollo de soluciones informáticas existen tres clasificaciones:

Las metodologías de desarrollo de software ágil buscan la satisfacción del cliente y la entrega rápida del software; equipos de proyecto pequeños y con alta motivación; están orientadas a la implementación y a la simplicidad general en el desarrollo. Están orientadas a proyectos pequeños donde existe gran incertidumbre con requisitos desconocidos o variables y el cliente es parte del proceso de desarrollo, lo que posibilita la retroalimentación constante y las respuestas rápidas a los cambios en el negocio debido a su gran capacidad de respuesta a los cambios (8).

Las metodologías de corte tradicional están diseñadas para equipos grandes de trabajo donde son generados elevados volúmenes de documentación, el período de tiempo para el desarrollo por lo general es relativamente largo y los requerimientos de manera general no cambian con mucha frecuencia.

Por otra parte las metodologías de tipo híbrido buscar agilizar el proceso de desarrollo de la solución mezclando artefactos y procedimientos de metodologías tradicionales con métodos de trabajo ágil.

Se decide adoptar una metodología ágil porque los requerimientos se tornan variables, se cuenta con un único desarrollador para elaborar la solución, el período de tiempo para la culminación del proyecto es inferior a un año y no será necesaria gran profundidad en la elaboración de los artefactos.

Metodología de desarrollo seleccionada. Programación Extrema (XP):

La actividad se realiza por un único desarrollador, por lo que no es posible utilizar la característica de desarrollo en dúo que normalmente caracteriza la metodología, pero se cuenta con alto dominio de las tecnologías a utilizar, el período de tiempo para el desarrollo es menor a un año, se trabajan directamente con el cliente haciendo pequeñas iteraciones y no se genera mucha documentación (8). . Para el marco de desarrollo de la propuesta de solución el desarrollador busca obtener resultados concretos en un corto período de tiempo y generando la documentación mínima necesaria para la compresión y continuación de la propuesta de desarrollo.

1.4.1. Lenguaje de modelado utilizado para la representación de los procesos

La notación de gestión de procesos de negocios de siglas en inglés BPMN, es una notación para el modelado de procesos de negocio capaz de representar de forma clara y concisa el proceso de obtención, envío y visualización de los datos de SASGBD añadidos a OSSIM en forma de logs.

1.4.2. Herramienta CASE (Computer Aided Software Engineering) para el modelado

Se decide utilizar la herramienta profesional Visual Paradigm Suite 8 para BPMN en el modelado de los procesos de la solución, porque permite modelar de manera eficiente y rápida el proceso de negocio que enmarca la solución ahorrando tiempo de desarrollo y plasmando información de manera coherente y comprensible para futuros desarrolladores que interactuarán con la solución propuesta.

1.4.3. Lenguajes de programación

Expresiones regulares del lenguaje de programación Python 2.6 presentes en el plugin parte de la solución.

Java: Lenguaje de alto nivel que constituye una plataforma independiente al sistema operativo. Para esta plataforma existen elevados cúmulos de información referente a soporte, ejemplos de utilización de códigos fuente, bibliotecas de desarrollo y kit de desarrollo del lenguaje o JDK. Es tomado como lenguaje de desarrollo de la propuesta de solución aprovechando que es el lenguaje de desarrollo de SASGBD y las ventajas que brinda su portabilidad a múltiples sistemas operativos. Esto potencia la compatibilidad entre el sistema y el hardware que utilizará la propuesta de solución.

1.4.4. Framework de desarrollo

OSSIM-Framework: Framework nativo de la plataforma AlienVault OSSIM que simplifica y controla las interacciones con dicho sistema y está encargado de la interpretación de los ficheros que integran los plugins (9).Constituye una herramienta indispensable en la interacción con la plataforma ya que abstrae a los desarrolladores de la interacción con el sistema.

1.4.5. Entorno Integrado de desarrollo(IDE)

- Editor de textos Geany 0.16: Es un entorno de desarrollo que soporta varios lenguajes de programación y proporciona un ambiente de trabajo adecuado para la creación y modificación de los ficheros a desarrollar para crear el plugin de OSSIM.

- Netbeans IDE 8: Es un Entorno Integrado de Desarrollo (IDE por sus siglas en Inglés) que soporta al lenguaje Java y proporciona un ambiente de trabajo adecuado para el desarrollo debido al conjunto de bibliotecas que proporciona y la documentación existente sobre ejemplos de trabajo con sus componentes. Soporta de manera estable al lenguaje de programación Java ya que el mismo posee componentes internos desarrollados en dicho lenguaje.

1.4.6. Gestores de bases de datos

PostgreSQL 9.1: Sistema gestor de bases de datos relacional y de código abierto utilizado para el almacenamiento de los datos de la aplicación SASGBD mediante el uso de dos esquemas en su base de datos. En un esquema maneja datos relacionados a la información referente a las auditorías realizadas y en el otro esquema información referente a la seguridad del sistema (10).Es utilizado por la propuesta de solución para obtener los datos de SASGBD.

MySQL: Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario, software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Es el gestor utilizado por la herramienta AlienVault OSSIM en el almacenamiento de la información generada por los sensores. Contiene nueve campos definidos por las variables userdata para guardar información definida por el usuario en la en la base de datos de OSSIM (10). Es utilizado para almacenar los indicadores de los plugins y sus identificadores de regla de plugins de la plataforma OSSIM.

1.4.7. Bibliotecas utilizadas en el desarrollo

Java Data Base Conector (JDBC) para PostgreSQL: Utilizada para interactuar con el gestor de bases de datos PostgreSQL.

Syslog4j: Utilizada para crear y enviar logs al servidor OSSIM en formato syslog.

Java Native Access (JNA): Utilizada para la generación de logs en plataformas Unix.

1.4.8. Servidor de AlienVault OSSIM

AlienVault OSSIM 5.2.2, siendo la versión más actual de la plataforma con interactúa el desarrollador hasta el momento. Este no posee diferencias muy significativas con la versión más actual del producto en cuanto a compatibilidad con la solución a desarrollar.

1.4.9. Nivel de integración estructural seleccionado

Punto a Punto: Representa el nivel más simple de integración, ya que las tecnologías utilizadas se comunican a través de interfaces intermedias donde no se contempla la visión del negocio. Establece una infraestructura básica para el intercambio de datos entre aplicaciones. La interrelación entre los sistemas es baja, por lo que tienen un alto grado de independencia entre sí. La relación entre SASGBD y OSSIM estará determinada por la propuesta de solución, la cual no influirá en el proceso de funcionamiento de las soluciones a vincular (11).

1.4.10. Arquitectura de software

La arquitectura del software es un nivel de diseño que se ocupa de los problemas. Comprende más allá de los algoritmos y estructuras de datos de la computación, diseño y especificación de la

estructura general del sistema se perfila como un nuevo tipo de problema. Los problemas estructurales que incluyen la organización y estructura de control global; protocolos de comunicación, sincronización y acceso a datos, asignación de funciones a elementos de diseño, distribución física, la composición de los elementos de diseño, escalado y el rendimiento; y la selección entre alternativas de diseño (12).

Características:

- Representación de alto nivel de la estructura del sistema describiendo las partes que lo integran.
- Puede incluir los patrones que supervisan la composición de sus componentes y las restricciones al aplicar los patrones.
- Trata aspectos del diseño y desarrollo que no pueden tratarse adecuadamente dentro de los módulos que forman el sistema.

1.4.11. Patrón arquitectónico seleccionado

Cliente/Servidor: La solución propuesta hará peticiones al servidor de bases de datos de SASGBD. Dependiendo de los resultados obtenidos como respuesta a la consulta, será enviada información o no al servidor de OSSIM para su tratamiento y posterior visualización.

1.4.12. Técnica de integración seleccionada

Extracción, transformación y carga de datos: La técnica Extracción, Transformación y Carga de Datos (Extract, Transform and Load), como su nombre lo indica extrae datos de un sistema fuente, transforma esos datos en información para satisfacer los requerimientos del negocio y carga el resultado en el sistema destino. La solución extrae datos del servidor SASGBD, selecciona un grupo de interés y los transforma en logs para su envío al servidor de OSSIM. Ya enviados a OSSIM son colectados por el plugin parte de la solución y mostrados en la paleta de mando (13).

Conclusiones del capítulo

En este capítulo fueron abordados conceptos de relevancia para la investigación. Fueron expuestos ejemplos de soluciones informáticas para la integración de aplicaciones y de otras que usan plugins como medio de unión a la plataforma objetivo de integración. Se presentó la metodología de desarrollo a seguir y se presentaron las herramientas a utilizar. Como resultado del estudio del marco teórico, la investigación se llegó obtuvo como conclusión que la vía más adecuada para adicionar los reportes es la utilización de plugins. La utilización de esta vía y las herramientas propuestas disminuye el nivel de complejidad en la interacción de las herramientas cuyos reportes son objetivo de centralización.

Capítulo II: Planificación y propuesta de solución

Introducción

En el presente capítulo se abordan los temas relacionados a la propuesta de solución. Se presenta la metodología de desarrollo, se elabora una propuesta de sistema a desarrollar y se exponen las características que debe cumplir el mismo para un mejor entendimiento en su desarrollo.

2.1 Propuesta de sistema

- Se implementó un plugin para la plataforma AlienVault OSSIM de tipo detector porque la información almacenada por SASGBD será obtenida y normalizada por una aplicación desarrollada como parte de la propuesta de solución. Posteriormente a la obtención y tratamiento de los datos se produce su envío a OSSIM para que el servidor pueda procesarlos.
- Para la elaboración del plugin para AlienVault OSSIM desarrollado en la propuesta de solución se utiliza como vía de entrada de los datos logs de tipo syslog. Esto se logra modificando el fichero rsyslog.conf localizado en la dirección /etc para que dirija el flujo de logs al fichero SASGBD.conf. Este fichero redirecciona el flujo generado al fichero SASGBD.log situado en /var/log. Este último fichero mencionado será consultado por el plugin para obtener los datos para el SIEM.
- El plugin consta de dos ficheros SASGBD.cfg y SASGBD.sql que cumplen la estructura tratada en el marco teórico y que estarán ubicados en /etc/ossim/agent/plugins. Para ejecutar las consultas almacenadas en el fichero SASGBD.sql en el servidor MySQL de OSSIM se ejecuta el comando ossim-db < SASGBD.sql.

Para facilitar la comprensión de la propuesta de solución se presenta la siguiente figura.

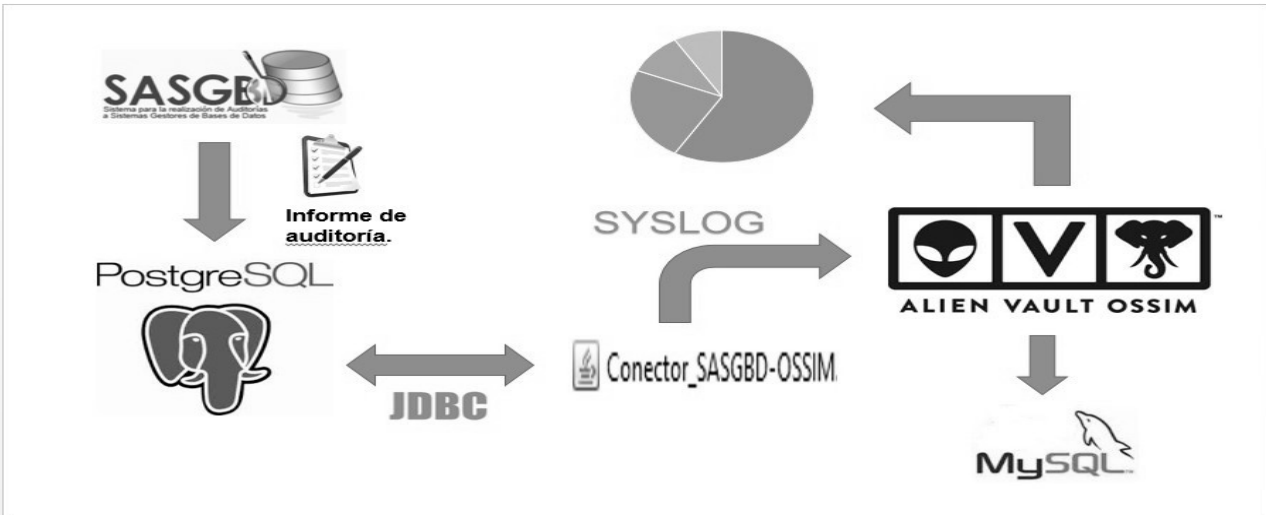

Figura-1 Flujo de datos en la propuesta de solución. Fuente: Elaboración propia.

2.2 Personal relacionado con el sistema

Administrador: Es el encargado de la gestión del sistema.

2.3 Funcionalidades de la propuesta de solución

- Consultar el servidor de bases de datos de SASGBD.
- Obtener el conjunto de datos de interés de la consulta SQL realizada.
- Transformar los datos al formato con que serán enviados.
- Enviar los datos obtenidos al servidor de OSSIM.
- Colectar los log recibidos.
- Mostrar los datos colectados en la interfaz web.

2.4 Características no funcionales del sistema

Las características no funcionales de la propuesta de solución están sujetas a las pautas usadas en AlienVault OSSIM y SASGBD.

- Apariencia o interfaz gráfica: Está sujeta a la plataforma AlienVault OSSIM ya que solo se le agregarán datos que se tomarán de SASGBD.
- Requerimientos de Hardware: Están sujetos a la plataforma AlienVault OSSIM por la parte del plugin desarrollado en la solución. La aplicación desarrollada como parte de la propuesta de solución necesitará que exista en el equipo anfitrión una Máquina Virtual de Java (MVJ) en la versión 7 o superior. Además, deben existir conexiones al servidor de AlienVault OSSIM y al sistema gestor de bases de datos que contenga la información de SASGBD. Por lo antes mencionado requerirá como recursos de hardware para su funcionamiento aquellos que requiera la MVJ para su correcto funcionamiento y conexiones de red estables.

- Requerimientos de fiabilidad: La precisión y exactitud de los datos de salidas del sistema se corresponderán a la calidad y exactitud de la información contenida en las bases de datos de SASGBD.
- Requerimientos de disponibilidad: El sistema debe estar operativo siempre que el auditor de seguridad lo emplee, exceptuando solo los días establecidos para mantenimiento a los servidores de OSSIM y SASGBD, cuando no se podrá establecer conexión.
- Requerimientos de seguridad: El sistema solo garantiza la estabilidad de los tres pilares de la seguridad informática si utiliza un usuario para hacer las consultas, solo con privilegios de lectura sobre las tablas de interés para la solución.

2.5 Fase de exploración

Primera fase de trabajo de la metodología XP, en la cual, los clientes plantean de manera general las funcionalidades que les son de interés para la elaboración de la solución. Transformándose dichas funcionalidades en historias de usuario. Partiendo de la información obtenida, el desarrollador evalúa de forma general el tiempo de codificación, de familiarización con las herramientas, tecnologías y prácticas que serán utilizadas en el proyecto. Además, se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo para ello (8).

El sistema a desarrollar debe ser capaz de obtener datos del servidor de bases de datos de SASGBD, seleccionarlos y enviarlos al servidor de AlienVault OSSIM en forma de log para su posterior tratamiento.

2.5.1. Historias de Usuario (HU)

Las HU son modelos que describen las funcionalidades de un sistema, especificando datos asociados a la misma. Son descritas por el cliente en su propio lenguaje, manifestando a muy alto nivel lo que el estiman que el sistema debe realizar. Su tratamiento es dinámico y flexible, lo que permite que en cualquier momento se puedan modificar, eliminar o reemplazar por otras más específicas o generales. El tiempo de desarrollo ideal para una HU varía entre 1 y 3 semanas (8).

Según Kent Beck, ingeniero de software estadounidense y uno de los creadores de las metodologías de desarrollo de software de programación extrema (XP) y el desarrollo guiado por pruebas (TDD). Cada HU recoge al menos los siguientes aspectos:

Número: Número asignado a la HU.

Nombre de HU: Atributo que contiene el nombre de la HU.

Usuario: El usuario del sistema que utiliza la HU.

Prioridad en el negocio: Evidencia el nivel de prioridad de la HU en el negocio. Se considera Alta en caso de que la HU sea imprescindible en el negocio, Media en caso de que su realización o no lo afecte considerablemente y Baja cuando no se considera una prioridad para el negocio.

Riesgo de desarrollo: Evidencia el nivel de riesgo en caso de no realizarse la HU. Se considera Alta, cuando el riesgo de no realizar la HU implica en el funcionamiento de la Plataforma. Media cuando el riesgo de no realizarla es medianamente importante y Baja en caso de que no se considere un riesgo el hecho de tardar en la realización de la HU y no implique en el funcionamiento de la Plataforma. (8)Puntos estimados: Este atributo no es más que una estimación hecha por el equipo de desarrollo del tiempo de duración de la HU. Cuando el valor es 1 equivale a una semana ideal de trabajo. En la metodología XP está definida una semana ideal en 5 días hábiles trabajando 40 horas, es decir, 8 horas diarias. Por lo que cuando el valor de

dicho atributo es 0.5 equivale a 2 días y medio de trabajo, lo que se traduce en 20 horas (8)Iteración asignada: Se especifica la iteración a la que pertenece la HU correspondiente.

Descripción: Posee una descripción de lo que realizará la HU.

Luego de analizar los datos y especificaciones de cada campo comprendido en el modelo de HU, se obtuvo para el presente trabajo las historias que se muestran a continuación:

Declaración de las Historias de Usuario (HU).

Tabla 1.HU # 1

Historia de Usuario	
Número: 1	**Usuario:** Administrador
Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Consultar servidor de bases de datos de SASGBD.	
Prioridad en negocio: Muy Alta.	**Riesgo de Desarrollo:** Alto.
Puntos estimados:1	**Iteración asignada:** 1
Programador responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite obtener los datos almacenados por SASGBD en su servidor de bases de datos.	
Observaciones: El usuario conectado debe estar registrado con credenciales de lectura en el sistema gestor de bases de datos de SASGBD.	
Prototipo de interfaz: No aplica	

Tabla 2.HU # 2

Historia de Usuario	
Número: 2	**Usuario:** Administrador

Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Obtener el conjunto de datos de interés.	
Prioridad en negocio: Muy Alta.	**Riesgo de Desarrollo:** Alto.
Puntos estimados:1	**Iteración asignada:** 1
Programador responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite obtener los datos de interés de los resultados que arroja la consulta al servidor de bases de datos	
Observaciones: La propuesta de solución debe haber obtenido los datos resultantes de la consulta a SASGBD.	
Prototipo de interfaz: No aplica	

Tabla 3.HU # 3

Historia de Usuario	
Número: 3	**Usuario:** Administrador
Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Estructurar los datos a convertir en logs.	
Prioridad en negocio: Muy Alta.	**Riesgo de Desarrollo:** Alto.
Puntos estimados:1	**Iteración asignada:** 1
Programador responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite organizar los datos a convertir en logs de tipo syslog.	
Observaciones: El usuario debe estructurar los datos a guardar de forma	

que puedan ser adicionados a los campos userdata de la tabla de la base de datos de OOSIM
Prototipo de interfaz: No aplica

El resto de las HU están ubicadas en el anexo 1 del documento.

2.6 Planificación

2.6.1. Proceso de estimación del esfuerzo de duración de cada HU

Una vez asignadas las prioridades de las historias de usuario, se prosigue a estimar el esfuerzo necesario para la elaboración de cada una de ellas por parte de los desarrolladores. Esta estimación se basa principalmente en la velocidad del equipo de desarrollo y en la semejanza con historias de usuario desarrolladas con anterioridad. Las HU de la presente investigación tienen un valor 1 punto por cada hora ideal (HI) laborable trabajada (8 horas para un día ideal) y un punto por cada día laboral ideal (DI) culminado (5 días ideales son la medida utilizada normalmente para delimitar una semana ideal) (8).

Teniendo en cuenta las medidas adoptadas un día ideal equivale a 8 horas ideales cumplidas y una semana ideal equivale a 40 horas ideales cumplidas o sea 5 DI cumplidos totalmente. Para cumplir con la semana ideal de trabajo DI => 5 en la función matemática DI = 40/HI, donde 0< HI =< 8 (8)

Los puntos de esfuerzo estimados son expresados en DI, se debe tener en cuenta que muy pocas veces el cronograma se lleva a cabo exactamente como se planifica. El esfuerzo necesario para

construir las historias de usuario está basado en la técnica de estimación para el desarrollo ágil, quedando conformada de la siguiente manera:

Tabla 4. Puntos estimados de esfuerzo para las HU.

Historia de usuario	Puntos estimados en DI
Consultar al servidor de bases de datos de SASGBD.	**5**
Obtener el conjunto de datos de interés	**5**
Estructurar los datos a convertir en logs de SASGBD.	**5**
Convertir los datos en logs de tipo syslog.	**10**
Enviar datos a OSSIM.	**5**
Elaborar ficheros componentes del plugin para OSSIM/	**15**
Desplegar los ficheros del plugin en el servidor de OSSIM.	**10**
Mostrar los datos colectados por el servidor.	**5**

2.7 Plan de iteraciones

Una vez agrupadas las HU por su prioridad, se especificaron los datos que las comprenden y la estimación del esfuerzo dedicado a desarrollar cada una de ellas. Luego fue planificada la fase de implementación, para la que se establecieron tres iteraciones que se describen a continuación:

Iteración 1: Se llevó a cabo el desarrollo de las historias de usuario 1, 2 . Estas responden al proceso de obtención de los datos de SASGBD.

Iteración 2: Se llevó a cabo el desarrollo de las historias de usuario 3, 4, 5. Estas responden al proceso de estructuración de los datos a enviar y la elaboración del plugin para OSSIM.

Iteración 3: Se llevó a cabo el desarrollo de las historias de usuario 6. Esta responde también al proceso de estructuración de los datos a enviar y la elaboración del plugin para OSSIM, pero fue tratada de forma única por su duración estimada.

Iteración 4: Se llevó a cabo el desarrollo de las historias de usuario 7 y 8, que se corresponden a la etapa de realización de las pruebas de software.

2.7.1. Plan de duración de las iteraciones

Este plan consiste en mostrar la duración estimada en DI de cada iteración.

Tabla 5.Creación de la HU y su duración en DI.

No.Iteración	HU desarrolladas	Duración en DI
1	**HU #1, HU #2,HU # 3**	**15**
2	**HU #4, HU # 5**	**15**
3	**HU # 6**	**15**
4	**HU # 7, HU # 8**	**15**
Totales	**8**	**60**

2.8 Plan de liberaciones

En el momento de planificar la liberación de una aplicación se debe contar con un balance de esta, pues si se realiza de forma acelerada no se tendrán suficientes funcionalidades terminadas que avalen dicha liberación, por otro lado, esperar mucho tiempo, conlleva a que la solución desarrollada quede rezagada frente a una posible competencia (8).

Tabla 6.Plan de entregas.

Historia de usuario	Iteración 1	Iteración 2	Iteración 3	Iteración 4
Consultar al servidor de bases de datos de SASGBD.	**28/04/2016**	**Culminada**	**Culminada**	**Culminada**
Obtener el conjunto de datos de interés.	**4/05/2016**	**Culminada**	**Culminada**	**Culminada**
Estructurar los datos a convertir en logs de	**No iniciada**	**12/05/2016**	**Culminada**	**Culminada**

SASGBD.				
Convertir los datos en logs de tipo syslog.	**No iniciada**	**26/05/2016**	**Culminada**	**Culminada**
Enviar datos a OSSIM:	**No iniciada**	**3/06/2016**	**Culminada**	**Culminada**
Elaborar ficheros componentes del plugin para OSSIM.	**No iniciada**	**No iniciada**	**24/06/2016**	**Culminada**
Desplegar los ficheros en el servidor de OSSIM.	**No iniciada**	**No iniciada**	**No iniciada**	**28/06/2016**
Mostrar los datos colectados por el servidor.	**No iniciada**	**No iniciada**	**No iniciada**	**2/07/2016**

Conclusiones del capítulo

En este capítulo fue descrita la propuesta de solución y se enmarcó su papel dentro del proceso de envío y recepción de información de información de SASGBD por parte de OSSIM. Se especificaron los aspectos relacionados con la metodología de desarrollo XP, se identificó la fase inicial de Exploración, en la que se confeccionaron las HU, llegando a un acuerdo con el cliente. En la posterior fase de Planificación fueron estimados los esfuerzos necesarios para desarrollar cada HU (mediante puntos de estimación) y las iteraciones en las que estarán comprendidas con el objetivo de obtener productos funcionales de forma organizada y en el tiempo estimado, el cual fue especificado en correspondencia con la duración de cada iteración mediante el plan de liberaciones.

Utilizando lo expuesto anteriormente, se definió el diseño y la implementación de la propuesta de solución con todas las características que se deben cumplir para la integración mediante el uso de plugins. Esto permitió obtener un correcto flujo de trabajo para el desarrollo.

Capítulo III: Diseño e implementación del sistema

Introducción

Se describe la arquitectura de AlienVault OSSIM y cómo se enmarca dentro de la misma la propuesta de solución. Muestra el patrón arquitectónico y el diseño que se utiliza en el proceso de desarrollo de la solución. Se diseña la estructura del plugin de integración. También, se crean las tarjetas CRC y las tareas de la ingeniería para desglosar las actividades comprendidas en cada Historia de Usuario. Por último, en este capítulo se define la codificación utilizada en la implementación del plugin.

3.1 Estructura del plugin para AlienVault OSSIM

- Archivo .cfg: Este archivo contiene la información del log y de cómo interpretar el mismo (9).El fichero está ubicado en /etc/ossim/agent/plugins y se encuentra estructurado de la forma siguiente:

- Región de información sobre el plugin: Esta contiene informaciones de interés en forma de comentarios con respecto al autor del plugin, identificador, versión, última modificación, etc.
- Región default: Esta contiene el identificador del plugin (plugin_id), y las variables dst_ip y dst_port.
- Región de configuración: Está denotada por [config] y contiene los parámetros de configuración del plugin. Estos parámetros son:
  - ✓ type: contiene el tipo de plugin
  - ✓ enable: especifica si el plugin está o no habilitado
  - ✓ source: especifica el tipo de datos que se manejarán.
  - ✓ location: contiene la ruta al fichero que contiene los datos a usar.
  - ✓ create_file: especifica si se creará un fichero para la recolección de los datos.
  - ✓ process: denota el nombre del proceso del plugin.
  - ✓ start: denota si se iniciará el proceso del plugin.
  - ✓ stop: denota si se detendrá el proceso del plugin.
  - ✓ shutdown: contiene la dirección del fichero que detiene la ejecución del proceso del plugin.
  - ✓ startup: contiene la dirección del fichero que inicia la ejecución del proceso del plugin.
- Región de eventos del plugin: Contiene los eventos del plugin identificados por el nombre del evento dentro de [ ] y contiene los siguientes parámetros para cada evento:

- ✓ event_type: contiene que tipo de evento que recoge el plugin.
- ✓ plugin_sid: contiene el identificador del evento para el plugin.
- ✓ regexp: contiene la expresión regular que filtrará los datos del plugin.

Además puede contener variables como sensor, date, src_ip y userdata, las variables userdata desde la 1 hasta la 9 denotan los campos de la base de datos utilizables por los usuarios para entrar los datos que se deseen almacenar en la base de datos.

- Archivo .sql: Este archivo contiene el tipo de evento, sirve para crear la estructura en la base de datos y construir la tabla de plugin_sid (1). Este fichero contiene cuatro sentencias sql que serán añadidas a la base de datos MySQL ,dos sentencias de tipo DELETE y dos de tipo INSERT (9). Las sentencias de tipo DELETE se encargan de eliminar plugins que tengan igual identificador y de eliminar el sid (evento del plugin) de la tabla plugin_sid. Las sentencias INSERT luego de ser ejecutadas las de eliminar son ejecutadas para añadir el nuevo plugin y sus eventos. Este fichero se localiza comúnmente en /usr/share/doc/ossim-mysql/contrib/plugins/, aunque puede ser ubicado en cualquier otra ubicación si se añade a la base de datos mediante el comando ossim-db < archivo.sql.
- Archivo .conf: Este archivo será el encargado de filtrar mediante el uso de condicionales los logs que serán copiados al fichero que los almacenará (9).
- Archivo .log: Este archivo contendrá los log filtrados por el archivo de extención .conf (9).

3.2 Diagrama de la propuesta de solución

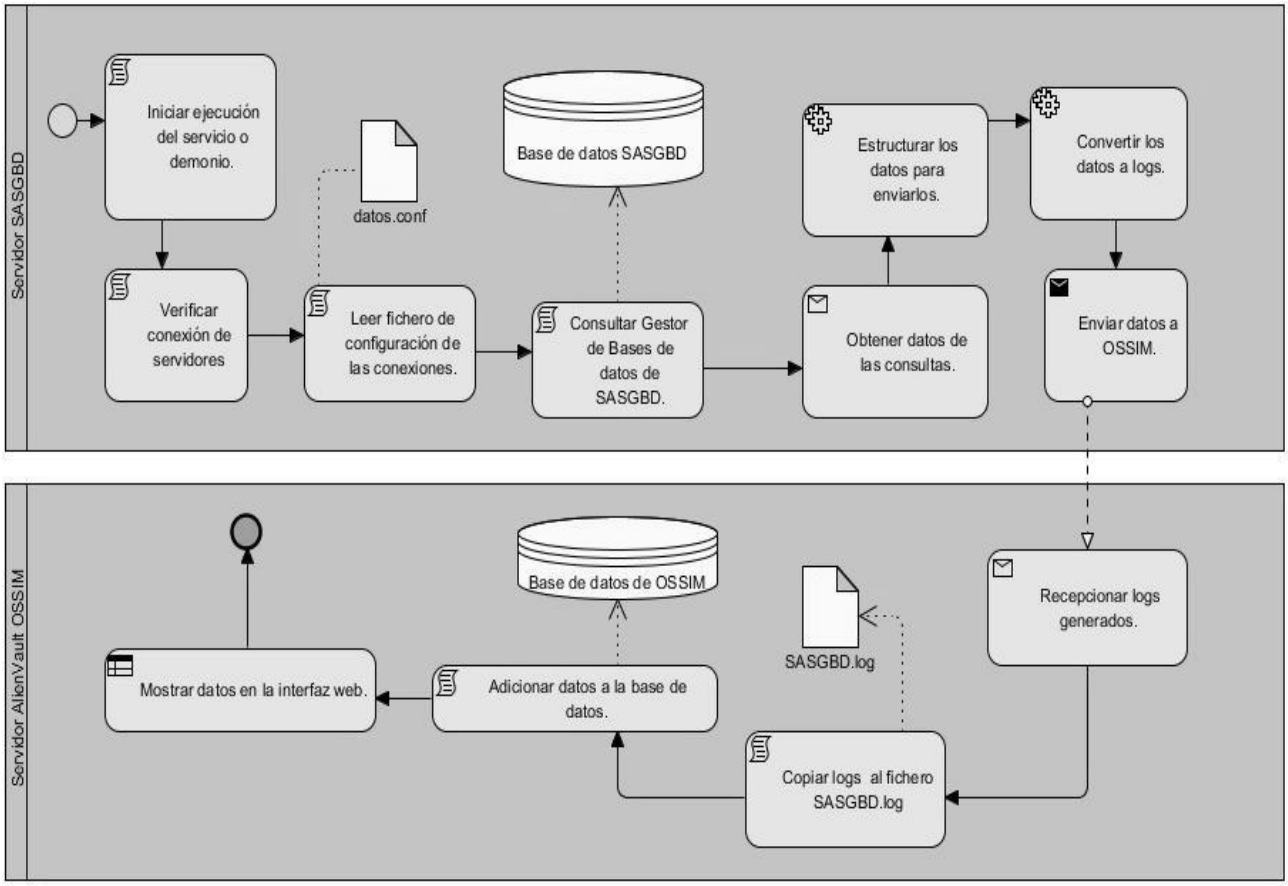

Figura-2. Diagrama de la solución propuesta.

La propuesta de solución ejecutarse debe verificar la conexión a los servidores involucrados y leer el fichero que guarda las configuraciones ya que de no existir la conexión o no encontrar el fichero de configuraciones en una estructura compresible no se iniciará la ejecución del mismo. Luego de iniciada la ejecución se leen, analizan y toman los resultados obtenidos y se transforman en logs de tipo syslog. Posterior mente se envían al servidor de OSSIM, este los colecta , los almacena y los muestra en la interfaz web.

3.3 Patrones de diseño utilizados

Patrón de diseño: Según Nicolás Tedeschi, analista de sistemas de la Facultad de Ingeniería del Uruguay; los patrones de diseño son considerados soluciones ya probadas a problemas de desarrollo de software sujeto a contextos similares. No se utilizan arbitrariamente, se debe tener en cuenta el nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).

3.3.1. Patrones GRASP (General Responsibility Assignment Software Pattern)

Para mayor entendimiento del uso de los patrones serán utilizados diagramas que no forman parte de la metodología XP.

Patrones generales de software para asignar responsabilidades. Se consideran buenas prácticas para el diseño de software porque el objetivo fundamental de su utilización es definir las clases de mayor jerarquía e implementar en ellas los métodos necesarios, además de asignar responsabilidades a las clases dependientes.

Experto: Determinó un modelo conceptual y la clase que posee la mayor jerarquía para asignarle una responsabilidad. Esto propició que el sistema se tornara más simple de desarrollar y de entender, disminuyendo las probabilidades de cometer errores de implementación y a su vez permita la reutilización de la propuesta para trabajos similares con SASGBD. La propuesta consta de dos clases que relacionadas entre sí poseen jerarquías bien definidas, la clase Demonio.java funge como clase controladora del proceso de recepción y envío. La clase datos se encarga de la interacción con el fichero datos.conf.

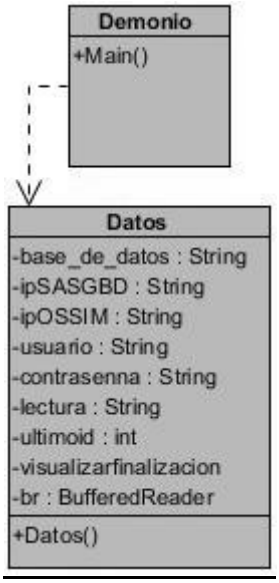

Figura-3. Diagrama de clases de la solución

La clase Demonio es la clase controladora de la solución, lo que la ubica en la sima jerárquica del diagrama de clases.

Bajo acoplamiento: Se utilizó con el objetivo de tener clases que dependan entre sí lo menos ligadas posible. De forma tal que cuando estas se vean de forma aislada se pueda entender la mayor parte de su estructura y su funcionamiento. En caso de que se produjese una modificación en alguna de ellas, tuviese la mínima repercusión posible en el resto de clases, potenciando la reutilización.

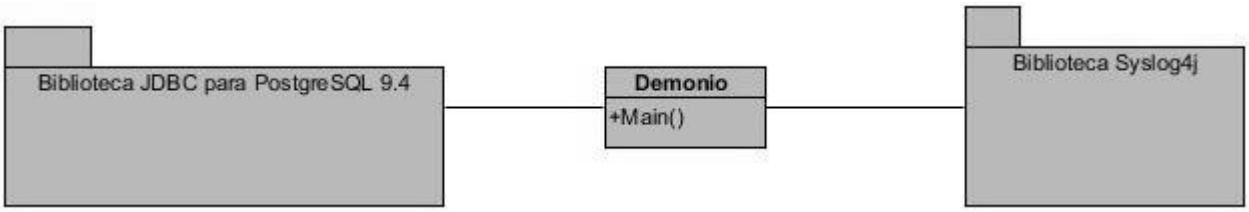

Figura-4. Relación de la clase Demonio con las bibliotecas usadas

La biblioteca jdbc puede ser reemplazada por otra versión siempre existan las clases a las que la solución hace referencia.

Creador: Este patrón se utilizó para crear instancias de clases a partir de una clase con alta jerarquía.

Se evidenció su utilización en la creación de instancias en la clase Demonio de las clases adicionadas mediante bibliotecas.

```
private final BufferedReader br;

    public Datos() throws FileNotFoundException, IOException {
    br = new BufferedReader( new FileReader("datos.conf"));
    lectura = br.readLine();
    br.close();
    base_de_datos = lectura.split(":")[0];
    ipSASGBD = lectura.split(":")[1];
    usuario = lectura.split(":")[2];
    contrasenna = lectura.split(":")[3];
    ultimoid = Integer.parseInt(lectura.split(":")[4]);
    ipOSSIM = lectura.split(":")[5];
    if(lectura.split(":")[6].equals("1"))
```

Figura 5.Ejemplo de utilización de objetos en la clase Demonio

3.3.2. Patrones GOF (Gang of Four, Banda de Cuatro)

Proporcionan a los programadores una estructura de código común a todos los proyectos que implemente una funcionalidad genérica. La utilización de estos patrones de diseño, permite ahorrar tiempo en la construcción del software y hacerlo más fácil de comprender, mantener y extender. Para el desarrollo de la solución fueron utilizados los siguientes:

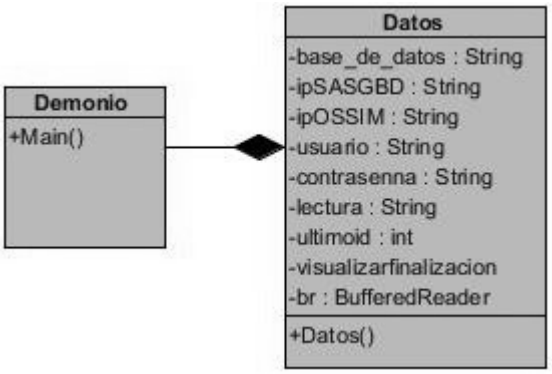

Figura-6.Relación entre las clases Demonio y Datos

• Singleton (instancia única): Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

Se refleja en la relación entre las clases Demonio y Datos, donde la primera ejecuta una única instancia de la segunda.

- Command (Orden): Encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma. Esto se ve reflejado en la solución a la hora de interactuar con los sistemas gestores de bases de datos.
  Se evidencia en el uso de objetos de la clase Datos en la clase Demonio para interactuar con el fichero datos.conf

```
        Class.forName("org.postgresql.Driver");
        //Conexion con postgresql
        c = DriverManager.getConnection("jdbc:postgresql://"+d.getIpSASGBD()+":5432/"+d.getBase_de_datos(),d
        stmt = c.createStatement();
        ResultSet rs = stmt.executeQuery(" select ve.valor,u.nombres,u.primer_apellido,u.segundo_apellido,mr
" from adbd.\"tb_dvalor_encontrado\" ve join adbd.\"tb_dmatriz_resultado\" mr on ve.fk_dmatriz_resultado_id
" join adbd.\"tb_dparametro\" p on ve.fk_dparametro_id = p.pk_id join seguridad.\"tb_dusuario\" u on mr.fk_
" join adbd.\"tb_raplicacion_servidor_versiong\" ra on mr.fk_rasvg_dservidor_id = ra.fk_dservidor_id join a
" join adbd.\"tb_nversion_gestor\" vg on ra.fk_nversiong_id = vg.pk_id join adbd.\"tb_ngestor\" gs on gs.pk_
" join adbd.\"tb_nriesgo_impacto\" ri on p.fk_nimpacto_id = ri.pk_id join adbd.\"tb_nevaluacion\" e on e.pk_
" join adbd.\"tb_nevaluacion\" e1 on e1.pk_id = ve.fk_neval_usuario_id where ve.pk_id >"+ultimoID+" and(e.n
" group by ve.pk_id,ve.valor,mr.fecha,s.ip,gs.nombre,vg.numero_version,p.nombre,ri.nivel,e.nivel,e1.nivel,u
" order by mr.fecha,ve.pk_id");
        //obtener datos de la consulta
        while ( rs.next()) {
```

Figura 7.Ejemplo de utilización de objetos en la clase Demonio

3.4 Tarjetas Clase – Responsabilidad – Colaborador (CRC)

Las tarjetas CRC, propias de la metodología XP, son una técnica simple e informal pero efectiva que ha sido propuesta para el diseño detallado de sistemas Orientado a Objetos. Se analizan basándose en sus responsabilidades con respecto al sistema y permiten que el equipo completo contribuya en la tarea del diseño (8).

Una tarjeta CRC representa un objeto, el nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda, y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente (8).

Las responsabilidades son los atributos y las operaciones relevantes para la clase. Dicho de una manera más simple una responsabilidad es ¨cualquier cosa que la clase sabe o hace ¨. Los colaboradores son aquellas clases que se requieren para que una clase reciba la información necesaria para completar una responsabilidad (8).

Tabla 7.Tarjeta CRC de la clase Demonio

Clase: Demonio	
Descripción: Es la encargada de interactuar con los sistemas gestores de bases de datos.	
Responsabilidad	**Colaboración**
Inicia la conexión a los gestores de bases de datos.	DriverManager, Connection
Escribir los datos en el fichero de configuración	FileWriter
Termina la conexión con los gestores de bases de datos.	Connection
Realiza consultas sql a los gestores de bases de datos.	ResultSet, Statement.
Clase propia de la biblioteca Syslog4j encargada de transformar los datos en logs y enviarlos	Syslog

Tabla 8.Tarjeta CRC de la clase Datos

Clase: Datos	
Descripción: Es la encargada de manejar el proceso de transferencia de datos desde el sistema gestor de base de datos fuente a sistema gestor destino.	
Responsabilidad	**Colaboración**
Leer los datos en el fichero de configuración	BufferedReader

3.5 Tareas de ingeniería

Tabla 9.Tarea de ingeniería 1

Tarea de Ingeniería	
Número Tarea: 1	**Historia de Usuario:** 1
Nombre Tarea: Consultar al servidor de bases de datos de SASGBD.	
Tipo de tarea: Desarrollo	**Puntos Estimados:**1
Fecha de inicio:22/04/2016	**Fecha de fin:**28/04/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite obtener los datos deseados del servidor SASGBD.	

Tabla 10.Tarea de ingeniería # 2

Tarea de Ingeniería	
Número Tarea: 2	**Historia de Usuario: 2**
Nombre Tarea: Estructurar los datos a convertir en logs de SASGBD.	
Tipo de tarea: Desarrollo	**Puntos Estimados:**1
Fecha de inicio:29/04/2016	**Fecha de fin:**4/05/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite seleccionar los datos almacenados en la base de datos de SASGBD a importar y estructurarlos de la forma en que serán enviados a OSSIM.	

Tabla 11.Tarea de ingeniería # 3

Tarea de Ingeniería	
Número Tarea: 3	**Historia de Usuario: 3**
Nombre Tarea: Crear fichero sql.	
Tipo de tarea: Desarrollo	**Puntos Estimados:**1
Fecha de fin:4/05/2016	**Fecha de fin:** 12/05/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	

Descripción: Permite introducir los datos de identificación del plugin la base de datos interna del OSSIM.

El resto de las tareas de ingeniería estarán ubicadas en el anexo 2

Conclusiones del capítulo

En este capítulo fueron tocados aspectos como los patrones de diseño a utilizar, el diagrama de negocios de la solución, las tarjetas CRC y las tareas de ingeniería. Los conceptos utilizados tributan al desarrollo del trabajo con metodología XP, lo que eleva la calidad de la solución y las entregas oportunas de resultados de iteraciones para realizarle pruebas.

La propuesta de solución fue implementada tomando en cuenta la prioridad de las funcionalidades a desarrollar y su tiempo de culminación fue cercano al estimado. Lo que propició que el comienzo de la siguiente fase del desarrollo no se realizara en la fecha planificada.

Capítulo IV: Pruebas de software

Introducción

En este capítulo es tratado el tema de la realización de las pruebas de software, el cual es un tema que suele denominarse verificación y validación. Verificación es un conjunto de actividades que aseguran que el software implemente correctamente una función específica. Validación es un conjunto diferente de actividades que aseguran que el software construido se corresponde con los requisitos del cliente (8).

Para comprobar el estado de la solución se implementan pruebas por parte del desarrollador de camino básico.

4.1 Prueba de caja blanca

Las pruebas de caja blanca son un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de caja blanca se comprueba la correcta ejecución del código auditado (14).

4.1.1. Técnica del camino básico

La prueba del camino básico es una técnica de prueba de caja blanca que permite al diseñador obtener una medida dela complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (14).

Pasos para la aplicación de la prueba:

Para la aplicación de la prueba se realizarán los siguientes eventos:

- Dibujar un grafo de flujo asociado.
- Calcular la complejidad ciclomática del grafo.
- Determinar un conjunto básico de caminos independientes.
- Preparar los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

La notación a usar es:

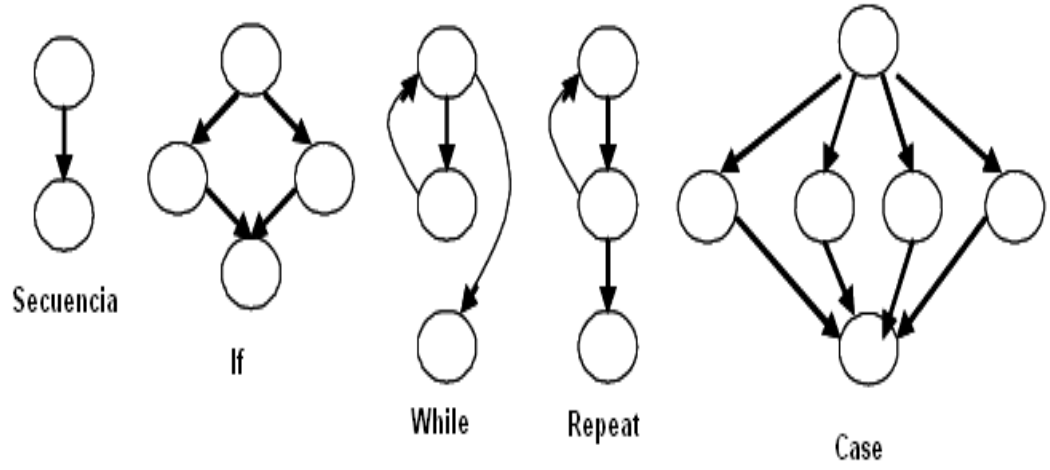

Figura-8.Tipos de estructura que puede contener el grafo de flujo

- Nodo: Cada círculo representado se denomina nodo del Grafo de Flujo.
- Arista: Las flechas del grafo se denominan aristas y representan el flujo de control del grafo.
- Regiones: Son las áreas delimitadas por las aristas y los nodos. También se incluye el área exterior del grafo, contándose como una región más.
- Complejidad ciclomática: La *complejidad ciclomática V (G)* define el número de *caminos independientes* del conjunto básico de un programa.
- Hay tres formas fundamentales de calcular la complejidad:
  - ✓ V (G) = número de regiones del grafo de flujo.
  - ✓ V (G) = *A - N + 2* donde: *A* es el número de aristas del grafo y *N* es el número de nodos.
  - ✓ V (G) = *P* + 1 donde: *P* es el número de nodos predicado (nodo del cual salen varias aristas) contenidos en el grafo *G*.
- Camino independiente: Un *camino es independiente* cuando introduce por lo menos un nuevo conjunto de sentencias de procesamiento o una nueva condición.
- Caso de prueba: Un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada.

Para la realización de las pruebas a la propuesta de solución se tiene el siguiente grafo que muestra el proceso de ejecución de la aplicación encargada de obtener y enviar los datos de SASGBD. Se realiza la auditoría al algoritmo encargado de la recepción, transformación y el envío de los datos, porque de este depende el correcto funcionamiento de la solución.

4.1.2. Aplicación de la prueba

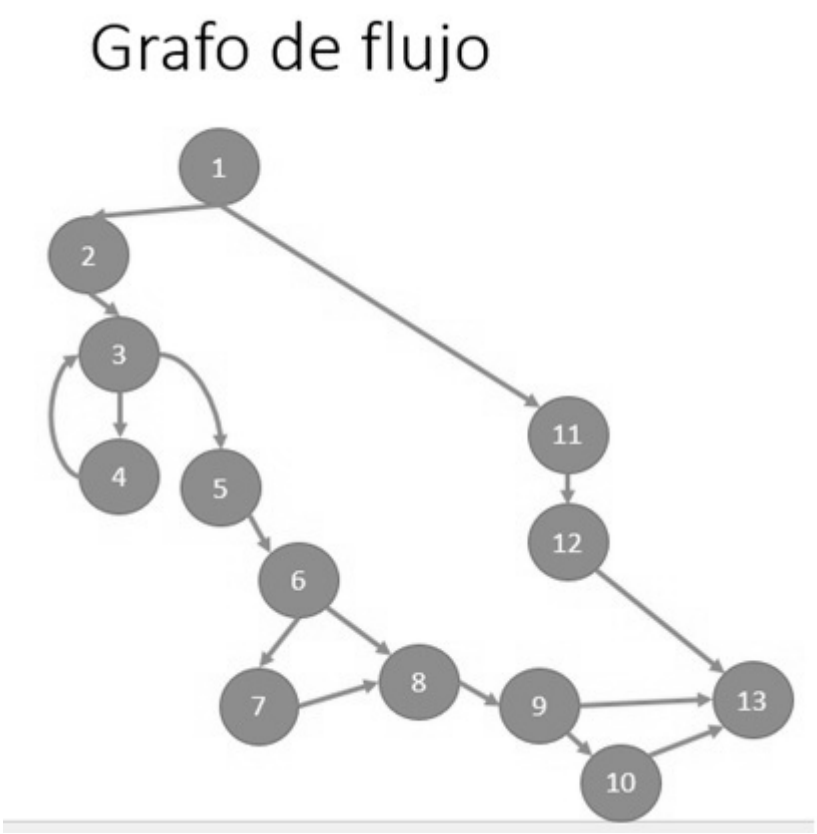

Figura 9. Grafo de flujo de los datos en el algoritmo auditado

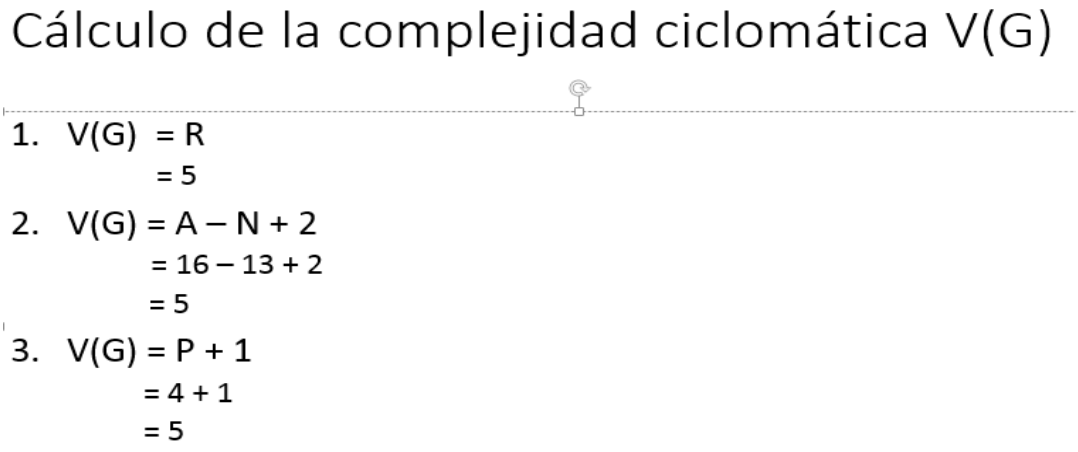

Figura 10.Cálculo de la complejidad ciclomática (14)

Caminos básicos

Nro. de camino	Secuencia de nodos
1	1,11,12,13
2	1,2,3,5,6,8,9,13
3	1,2,3,5,6,7,8,9,13
4	1,2,3,4,3,5,6,8,9,13
5	1,2,3,5,6,7,8,9,10,13

Figura 11.Caminos básicos encontrados en la aplicación de la prueba al código.

Tabla 12.Caso de pruebas seleccionado

Proceso: Obtención y envío de datos de SASGBD.
Caso de prueba: Obtención, reestructuración y envío de datos.
Entrada: Los datos a obtener, tratar y enviar están almacenados en la base de datos de SASGBD en una estructura diferente a la utilizada por OSSIM. El auditor de seguridad deberá ejecutar la aplicación con el fichero de configuración configurado para que audite la totalidad de casos existentes en la base de datos tomada como para realizar el envío al servidor OSSIM de los logs generados.
Resultados: La información es obtenida, transformada y enviada al servidor, generándose logs de tipo syslog para la recepción en OSSIM. Durante el proceso se pueden generar alertas por falta de conexión con el servidor de bases de datos de SASGBD y si se desea puede o no recibir la notificación de culminación del proceso de recepción y envío de los datos.
Condiciones: Los servidores de SASGBD y AlienVault OSSIM deben estar conectados mediante la red.

Conclusiones del capítulo

La ejecución de la prueba de caja blanca, camino básico, evidenció el correcto funcionamiento del algoritmo principal de la solución. Obteniéndose cinco secuencias posibles para la ejecución del código auditado y además propicia la elección de los casos de prueba en diferentes condiciones de ejecución. Los resultados obtenidos abalan el buen comportamiento de la aplicación implementada. Analizando detenidamente la secuencia de eventos que se sigue para la ejecución de la misma, el proceso de obtención, transformación y envío de datos desde el servidor de bases

de datos de SASGBD al servidor de OSSIM se realiza de forma estable de acuerdo a lo planificado.

Conclusiones generales.

Al término de la presente investigación se concluye lo siguiente:

1. Al realizar el análisis de soluciones informáticas que permiten realizar integración, no se identificó ninguna a la que se pudiera auditar su código fuente para comprender su modo de funcionamiento, por lo que fueron desechadas. Pero sirvieron como punto de partida para definir que la vía de solución que contendría las características que debía cumplir la propuesta de solución sería la de implementación de una solución que funcionara como interfaz de recolección. Esta en conjunto a un plugin que funcionara como colector, conforman la mejor vía de solución al proceso de desarrollo.
2. La aplicación de patrones de diseño y arquitectónicos, permitió obtener una solución Informática más flexible.
3. El análisis de las plataformas SASGBD y OSSIM, permitió identificar las vulnerabilidades que debían ser reportadas por la solución informática.
4. La arquitectura de integración seleccionada, permitió la correcta comunicación entre las plataformas SASGBD y OSSIM.
5. Con el desarrollo de la propuesta de solución, se centralizó la información referente a las vulnerabilidades de reportadas por SASGBD en sus reportes en OSSIM, permitiendo la correlación de la misma.

Recomendaciones

Se recomienda la implementación de mecanismos propios de cifrado y descifrado sin comprometer el rendimiento del servidor de AlienVault OSSIM para un mayor control de la privacidad de la información enviada. Utilizar estos mecanismos para el envío por parte de SASGBD y la recepción de los datos por parte del servidor de OSSIM teniendo en cuenta la variante de entrada de los logs al mismo.

También se recomienda crear un usuario con solo privilegios de lectura sobre las tablas a consultar por la solución dentro del servidor de bases de datos de SASGBD.

Referencias bibliográficas

1. *OSSIM, una alternativa para la integración de la gestión de seguridad en la red.* **Walter Baluja García, Cesar Camilo Caro Reina, Frank Abel Cancio Bello.** No. 1, La Habana : s.n., enero-abril de 2012, Revista Telem@tica, Vol. Vol. 11, págs. 11-19. ISSN 1729-3804.

2. **Olmos., Adrián Puchades.** *Análisis de la plataforma Ossim.* Valencia : s.n., 2008.

3. **Eduardo Hernán Amaya Guzmán, Laura Victoria Quiroga Martínez.** *INTEGRACIÓN Y EVALUACIÓN DEL PILOTO DE LA HERRAMIENTA DE MONITOREO ALIENVAULT EN LA PLATAFORMA TECNOLÓGICA DE TELEFONICA TELECOM.* Bogotá D.C. : s.n., 2012.

4. **Altova.** Altova. [En línea] 2016. [Citado el: 29 de Junio de 2016.] http://www.altova.com/es/.

5. **LANSA.** LANSA. [En línea] 2016. [Citado el: 28 de Junio de 2016.] http://www.lansa.com/.

6. •**Qualisys.** Qualisys. [En línea] 2016. [Citado el: 27 de Junio de 2016.] http://www.qualisyss.com/Sp/inicio.html.

7. **Information Builders. [En línea] Information Builders, 2016. [Citado el: 20 de Junio de 2016.] http://www.informationbuilders.es/products/integration/suite.**

8. Joskowicz, José. *Reglas y Prácticas en eXtreme Programming.* 2008.

9. Dominique Karg, Jesús D. Muñoz,David Gil,Santiago González,Julio Casal. *OSSIM Open Source Security Information Management Descripción General del Sistema.* 2003.

10. SlideShare. SlideShare. [En línea] 2015. [Citado el: 12 de Enero de 2016.] http://es.slideshare.net/arturoea1/postgresql-vs-mysql-postgresql-como-alternativa.

11. *Generalizando un modelo de desarrollo de ecosistemas de software.* González, MSc. Jessie Castell. La Habana : Guayaquil, 2014.

12. SG Buzz. SG Buzz. [En línea] 2010. [Citado el: 6 de Abril de 2016.] http://sg.com.mx/revista/27/arquitectura-software.

13. Universidad del País Vasco. *ARQUITECTURA DEL ENTORNO Y TÉCNICAS DE INTEGRACIÓN.* 2015.

14. Roger y Pressman, Roger S. *Ingeniería de software. Un enfoque práctico.* Madrid : s.n., 2007.

Bibliografía

1. **Roger y Pressman, Roger S.** *Ingeniería de software. Un enfoque práctico.* Madrid : s.n., 2007.

2. **Eduardo Hernán Amaya Guzmán, Laura Victoria Quiroga Martínez.** *INTEGRACIÓN Y EVALUACIÓN DEL PILOTO DE LA HERRAMIENTA DE MONITOREO ALIENVAULT EN LA PLATAFORMA TECNOLÓGICA DE TELEFONICA TELECOM.* Bogotá D.C. : s.n., 2012.

3. **Dominique Karg, Jesús D. Muñoz,David Gil,Santiago González,Julio Casal.** *OSSIM Open Source Security Information Management Descripción General del Sistema.* 2003.

4. **Olmos., Adrián Puchades.** *Análisis de la plataforma Ossim.* Valencia : s.n., 2008.

5. **Joskowicz, José.** *Reglas y Prácticas en eXtreme Programming.* 2008.

6. *OSSIM, una alternativa para la integración de la gestión de seguridad en la red.* **Walter Baluja García, Cesar Camilo Caro Reina, Frank Abel Cancio Bello.** No. 1, La Habana : s.n., enero-abril de 2012, Revista Telem@tica, Vol. Vol. 11, págs. 11-19. ISSN 1729-3804.

7. **Information Builders. [En línea] Information Builders, 2016. [Citado el: 20 de Junio de 2016.] http://www.informationbuilders.es/products/integration/suite.**

8. *Generalizando un modelo de desarrollo de ecosistemas de software.* González, MSc. Jessie Castell. La Habana : Guayaquil, 2014.

9. SlideShare. SlideShare. [En línea] 2015. [Citado el: 12 de Enero de 2016.] http://es.slideshare.net/arturoea1/postgresql-vs-mysql-postgresql-como-alternativa.

10. LANSA. LANSA. [En línea] 2016. [Citado el: 28 de Junio de 2016.] http://www.lansa.com/.

11. •Qualisys. Qualisys. [En línea] 2016. [Citado el: 27 de Junio de 2016.] http://www.qualisyss.com/Sp/inicio.html.

12. Altova. Altova. [En línea] 2016. [Citado el: 29 de Junio de 2016.] http://www.altova.com/es/.

13. SG Buzz. SG Buzz. [En línea] 2010. [Citado el: 6 de Abril de 2016.] http://sg.com.mx/revista/27/arquitectura-software.

14. Universidad del País Vasco. *ARQUITECTURA DEL ENTORNO Y TÉCNICAS DE INTEGRACIÓN.* 2015.

15. AlienVault. *Resumen de rutas y ficheros de configuración en OSSIM y AlienVault USM.* 2012.

16. —. *AlienVault Unified Security Management™ Solution How to create a data source plugin.* 2014.

17. —. *AlienVault Unified Security Management™ Solution Complete. Simple. Affordable Data Source Plugin Management.* 2014.

18. Management, Data Source Plugin. *Data Source Plugin Management.* 2011.

19. AlienVault. *Plugins Management Guide.* 2015.

20. Marcofi Andretti Torres Manrique, Diego Alejandro Villegas Oliveros. *INTEGRACIÓN DE OSSIM Y UNTANGLE.* 2010.

21. Martínez, Mariana Mercedes Tenorio. *MANUAL DE OSSIM.* 2009.

22. Párrigas, Angel Alonso. *OSSIM, una plataforma clave para la seguridad en profundidad.*

Anexos

Anexo 1:

Tabla 13.HU # 4

Historia de Usuario	
Número: 4	**Usuario:** Administrador
Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Convertir datos en logs de tipo syslog.	
Prioridad en negocio: Alta.	**Riesgo de Desarrollo:** Alto.
Puntos estimados:2	**Iteración asignada:** 2
Programador responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite convertir los datos seleccionados, estando estructuralmente tratados, en logs de tipo syslog.	
Observaciones: El usuario debe poseer los datos obtenidos de SASGBD en una estructura adecuada.	
Prototipo de interfaz: No aplica	

Tabla 14.HU # 5

Historia de Usuario	
Número: 5	**Usuario:** Administrador
Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Enviar datos a OSSIM.	
Prioridad en negocio: Alta.	**Riesgo de Desarrollo:** Alto.
Puntos estimados:1	**Iteración asignada:** 2

Programador responsable: Elvis Javier Rodríguez Soto.	
Descripción: Constituye una de las principales funciones que realiza la aplicación parte de la propuesta de solución.	
Observaciones: La propuesta de solución envía los logs utilizando funciones de la biblioteca syslog4j.	
Prototipo de interfaz: No aplica	

Tabla 15.HU # 6

Historia de Usuario	
Número: 6	**Usuario:** Administrador
Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Elaborar ficheros componentes del plugin de OSSIM.	
Prioridad en negocio: Media	**Riesgo de Desarrollo:** Alto.
Puntos estimados:3	**Iteración asignada:** 2
Programador responsable: Elvis Javier Rodríguez Soto.	
Descripción: De la correcta elaboración de estos ficheros que componen el	

plugin para OSSIM dependerá la calidad de la recepción de los logs en el servidor.
Observaciones: Los ficheros deben poseer la estructura mostrada anteriormente en el capítulo de diseño e implementación.
Prototipo de interfaz: No aplica

Tabla 16.HU # 7

Historia de Usuario	
Número: 7	**Usuario:** Administrador
Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Desplegar los ficheros del plugin dentro del servidor de OSSIM.	
Prioridad en negocio: Baja.	**Riesgo de Desarrollo:** Bajo.
Puntos estimados:2	**Iteración asignada:** 4
Programador responsable: Elvis Javier Rodríguez Soto.	

Descripción: Permite la ubicación de los ficheros del plugin, de forma acorde a la dirección que deben ocupar dentro del servidor para luego activarlo.
Observaciones: El servidor debe ser reiniciado para que el sensor sea reconocido.
Prototipo de interfaz: No aplica

Tabla 17.HU # 8

Historia de Usuario	
Número: 8	**Usuario:** Administrador
Modificación de Historia de Usuario #: Ninguna	
Nombre de Historia de Usuario: Mostrar los datos colectados por el servidor	
Prioridad en negocio: Baja.	**Riesgo de Desarrollo:** Medio.
Puntos estimados:1	**Iteración asignada:** 4
Programador responsable: Elvis Javier Rodríguez Soto.	

Descripción: Permite verificar el cumplimiento del objetivo del proceso de desarrollo al mostrar la información manejada.
Observaciones: El usuario debe enviar logs desde la aplicación parte de la solución y debe tener configurado el servidor de OSSIM para mostrar la información enviada.
Prototipo de interfaz: No aplica

Anexo 2:

Tabla 18.Tarea de ingeniería # 4

Tarea de Ingeniería	
Número Tarea: 4	**Historia de Usuario:** 4
Nombre Tarea: Convertir los datos en logs y enviarlos a OSSIM.	
Tipo de tarea: Desarrollo	**Puntos Estimados:**2
Fecha de inicio: 13/05/2016	**Fecha de fin:**26/05/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	

Descripción: Permite ordenar estructurar los datos en forma de syslog y enviarlos para la recolección en OSSIM.

Tabla 19.Tarea de ingeniería # 5

Tarea de Ingeniería	
Número Tarea: 5	**Historia de Usuario:** 5
Nombre Tarea: Enviar datos al servidor de OSSIM	
Tipo de tarea: Desarrollo	**Puntos Estimados:**1
Fecha de inicio:27/05/2016	**Fecha de fin:**3/06/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite crear llenar el fichero colector de log parte del plugin.	

Tabla 20.Tarea de ingeniería # 6

Tarea de Ingeniería	
Número Tarea: 6	**Historia de Usuario:** 6
Nombre Tarea: Elaborar los ficheros del plugin de OSSIM.	
Tipo de tarea: Desarrollo	**Puntos Estimados:**3
Fecha de inicio:6/06/2016	Fecha de fin:24/06/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite crear los ficheros que constituirán el plugin para OSSIM.	

Tabla 21.Tarea de ingeniería # 7

Tarea de Ingeniería	
Número Tarea:7	**Historia de Usuario:** 7
Nombre Tarea: Desplegar los ficheros del plugin en el servidor.	
Tipo de tarea: Desarrollo	**Puntos Estimados:** 2
Fecha de inicio:25/04/2016	**Fecha de fin:**28/06/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite seleccionar las pruebas a aplicar para el control de la calidad de la solución.	

Tabla 22.Tarea de ingeniería # 8

Tarea de Ingeniería	
Número Tarea: 8	**Historia de Usuario: 8**
Nombre Tarea: Verificar los datos mostrados por OSSIM.	
Tipo de tarea: Corrección	**Puntos Estimados:**1.0
Fecha de inicio:29/06/2016	**Fecha de fin:**2/07/2016
Programador Responsable: Elvis Javier Rodríguez Soto.	
Descripción: Permite validar los resultados del envío de los datos al servidor de OSSIM.	

Anexo 3:

Figura 12.Interfaz inicial de la vista web del servidor OSSIM

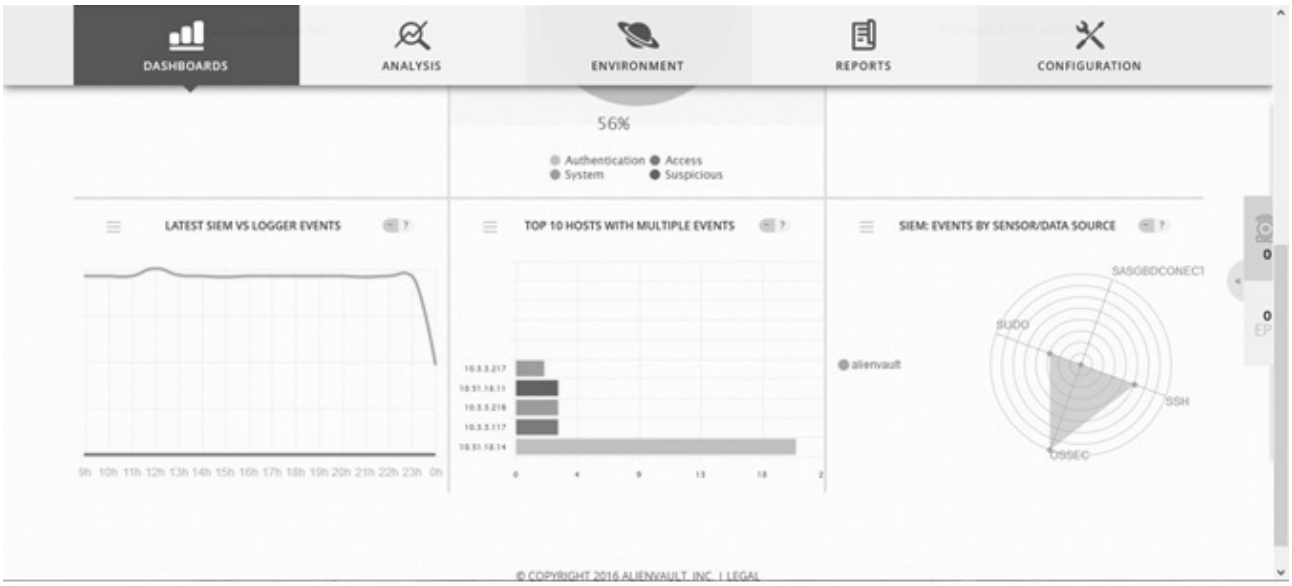

Figura 13.Parte inferior de la interfaz inicial que muestra a la derecha los plugins activos en el servidor

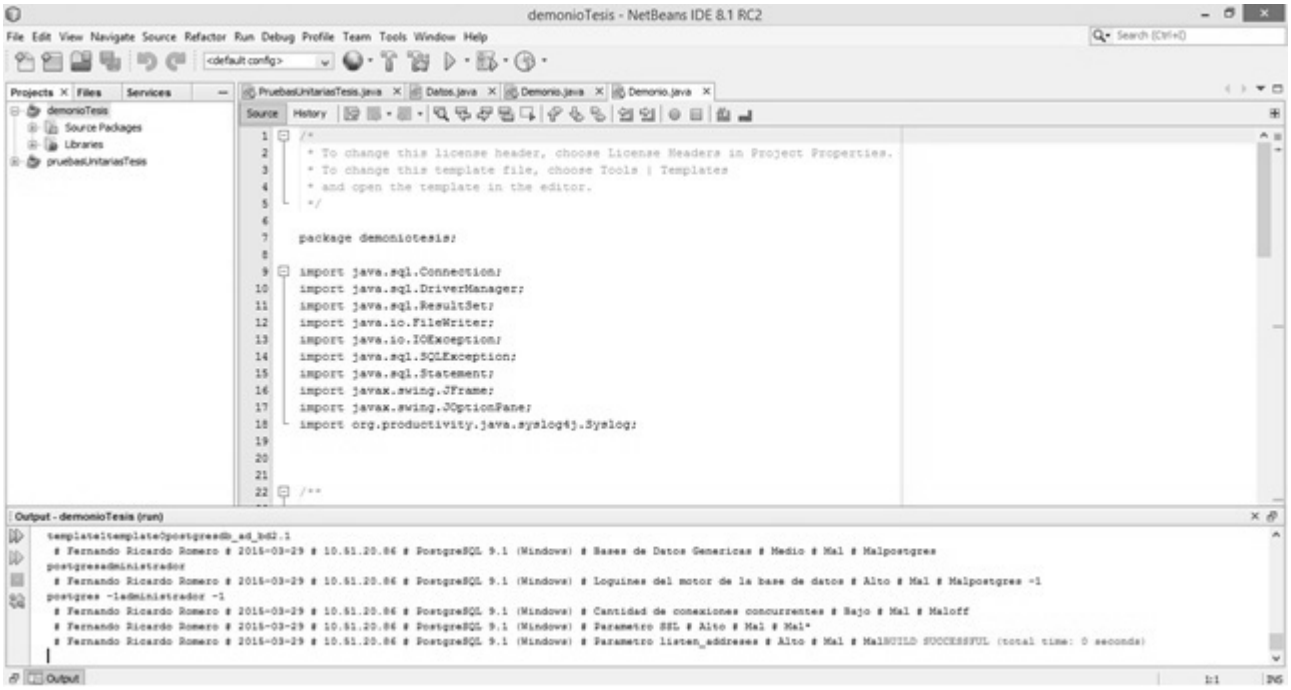

Figura 14.Imagen que muestra el envío de logs utilizando el compilador del IDE Netbeans.

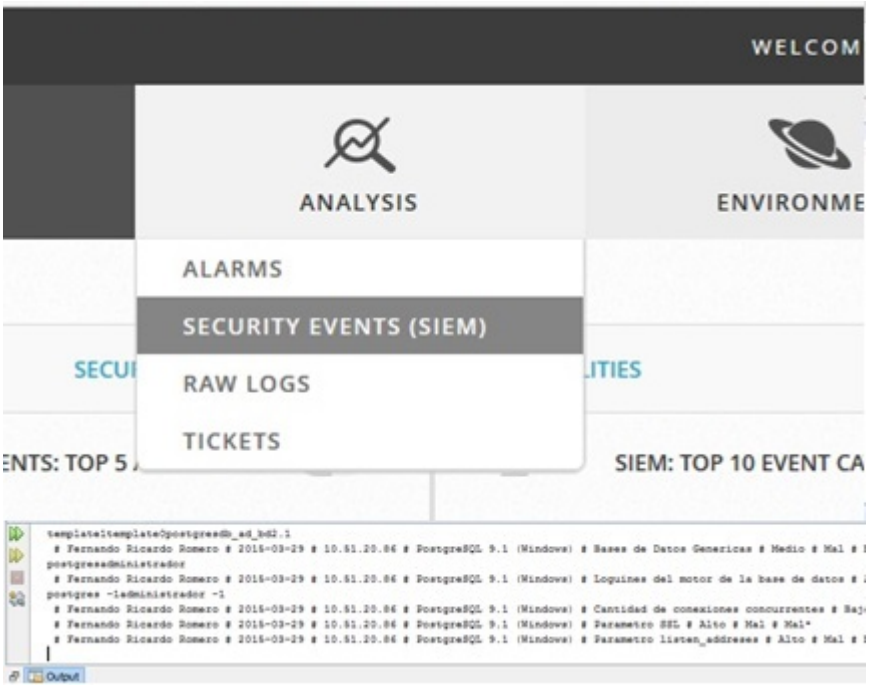

Figura 15.Menú de acceso a la vista donde se visualizan los datos de los sensores del OSSIM.

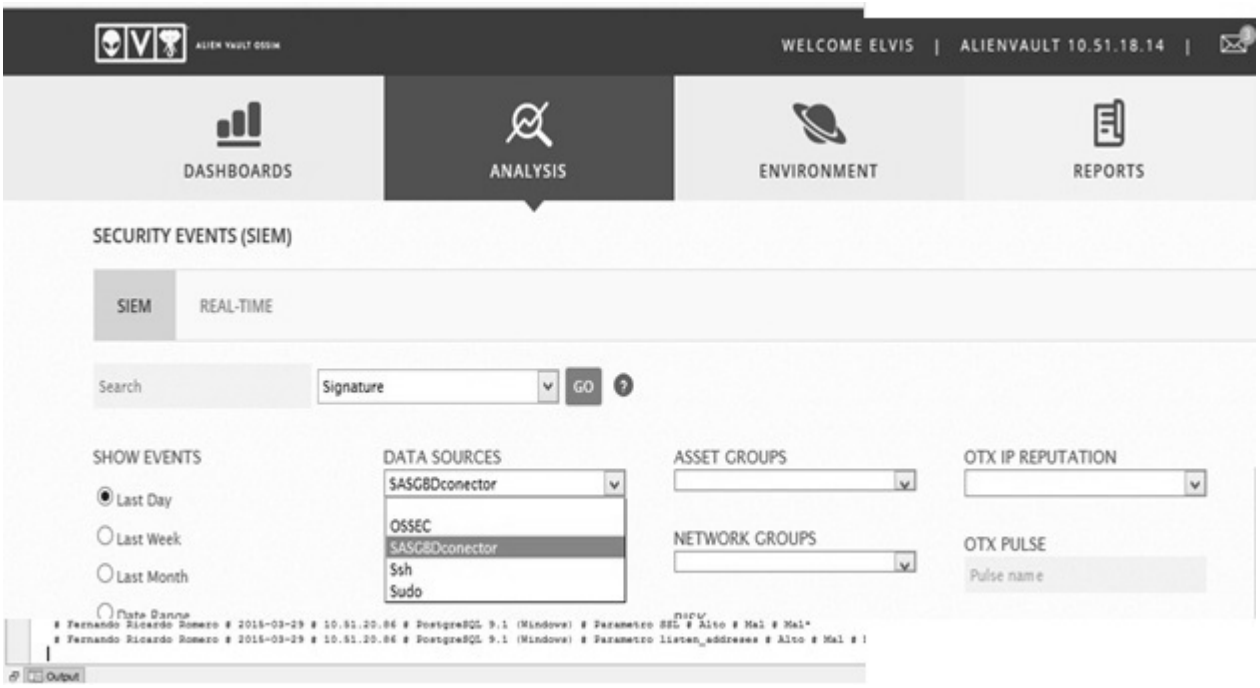

Figura 16.Filtros para ver los eventos del SIEM OSSIM

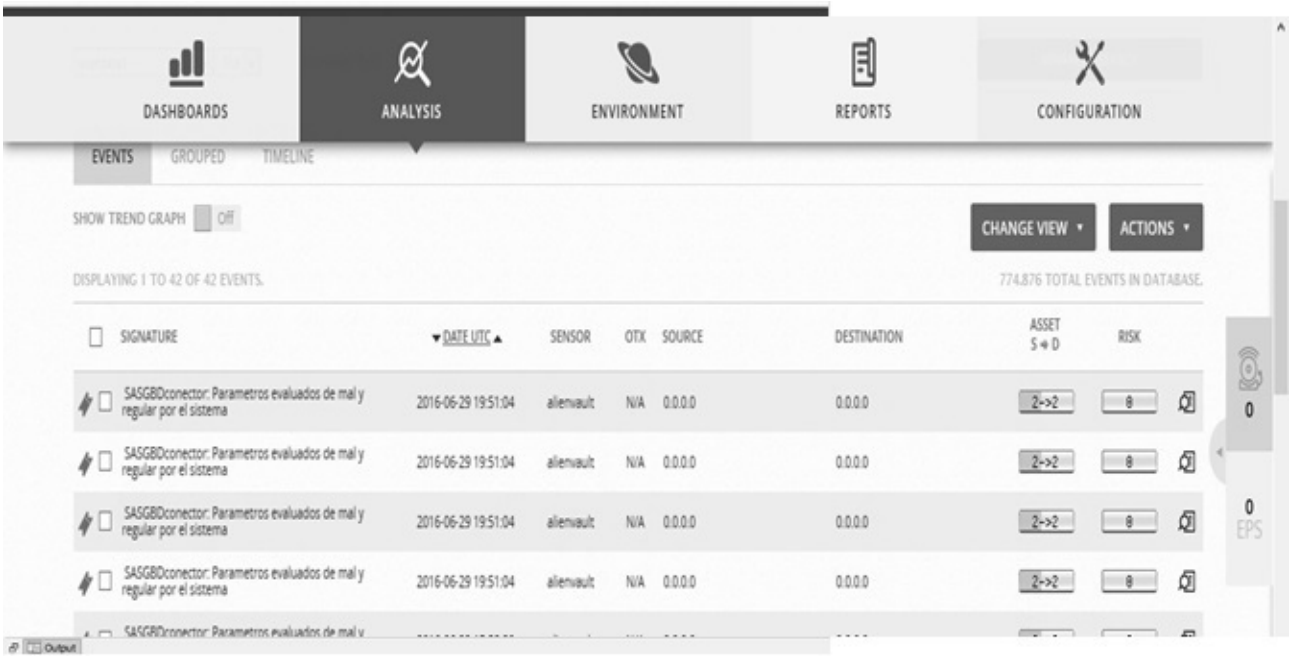

Figura 17.Eventos recibidos en OSSIM provenientes de la aplicación parte de la propuesta de solución.

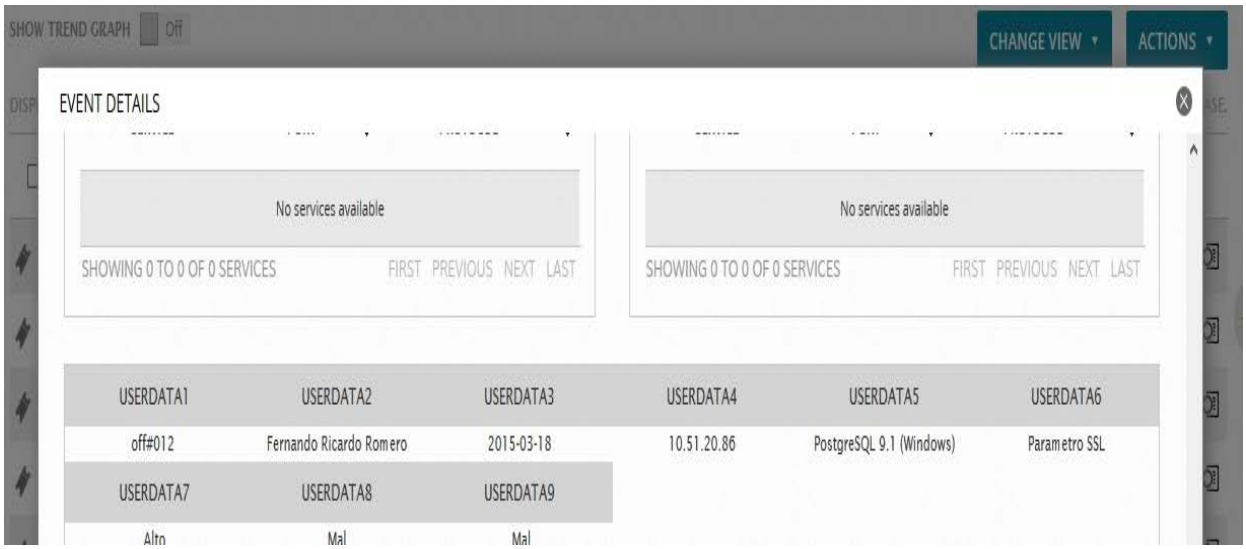

Figura 18.Visualización de los datos al dar clic sobre los eventos registrados por OSSIM

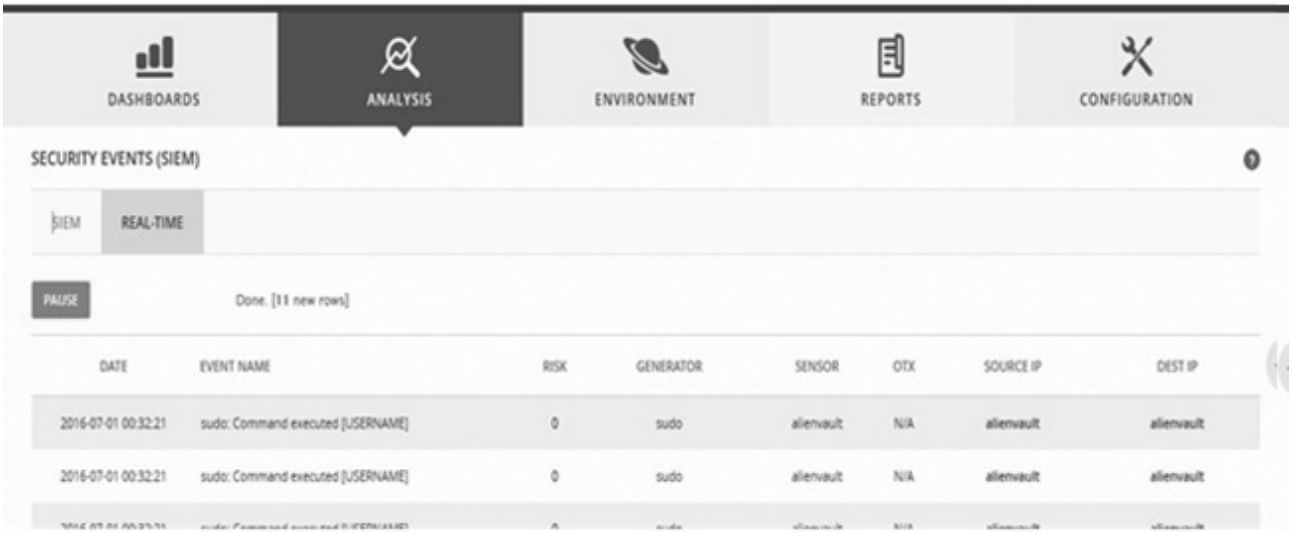

Figura 19.Interfaz de visualización de la entrada de los datos en tiempo real.

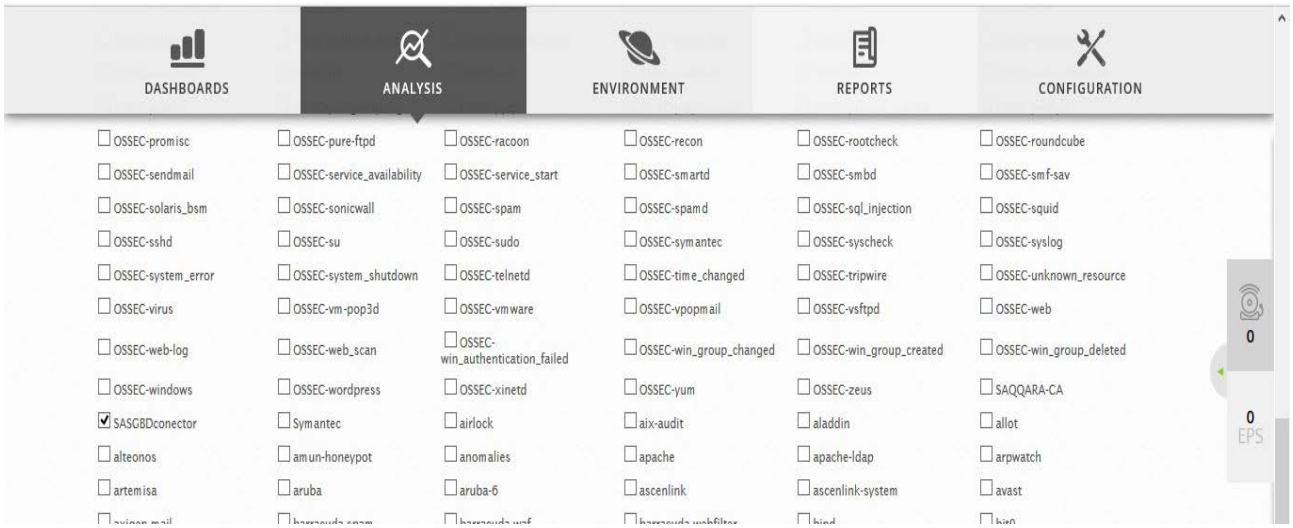

Figura 20.Aplicar un filtro a la entrada de logs en tiempo real.

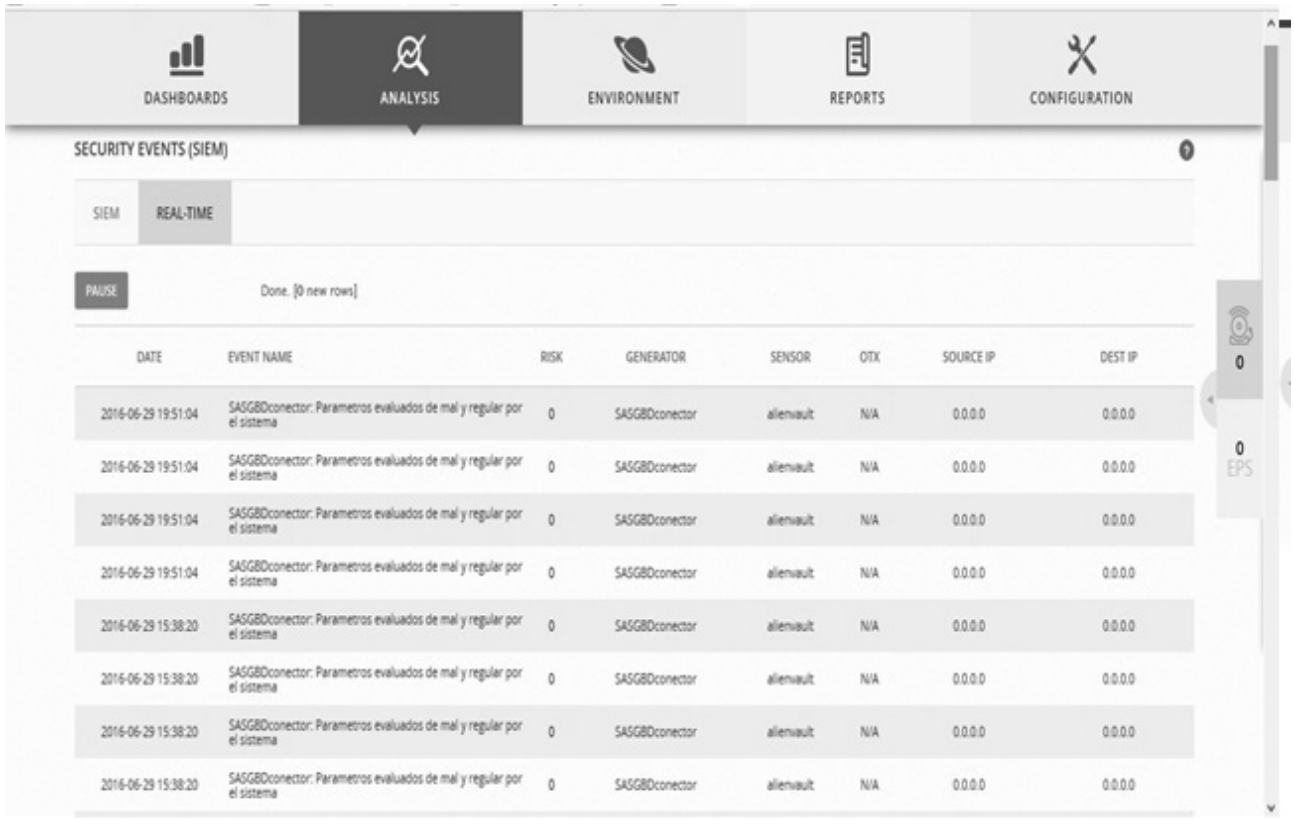

Figura 21.Entrada de logs enviados en tiempo real por la aplicación parte de la solución

Glosario

Propuestas de solución: Es el resultado del trabajo del desarrollador para solucionar la situación problemática existente.

Liberaciones: Entrega de prototipos no funcionales durante el proceso de desarrollo para la obtención de no conformidades, incluyendo la entrega del producto final.

Entorno Integrado de Desarrollo: Plataforma sobre la cual se desarrolla la propuesta de solución. En este proceso de desarrollo se utilizó Netbeans 8.

Patrones de diseño: Modelos generales aplicables condiciones del proceso de desarrollo que cumplan con determinados aspectos para el tipo de situación presentada.