Módulo de Generación de Reportes Gráficos de una Honeynet a partir de los logs tcpdumps

Cayetano, Denisse; Rivadeneira, Christian; Abad, Cristina Ms.Sc.
Facultad de Ingeniería en Electricidad y Computación (FIEC)
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil-Ecuador

Resumen

En la actualidad, la mayoría de las redes mantienen conexión con el ancho mundo de la Internet. Esta conexión representa un constante peligro debido a la vulnerabilidad hacia los ataques que realizan los hackers. Esta es en efecto la razón principal por la cual en muchas redes, donde es primordial mantener la confidencialidad y consistencia de sus datos, se ejecutan algunos tipos de mecanismos de seguridad sobre sus conexiones. En el mercado existen varias herramientas que ayudan al análisis de, lo que podría ser, un posible ataque en la red; sin embargo, a pesar de que dichas herramientas se encuentran disponibles tanto de manera gratuita como propietaria, no abastecen la demanda para el análisis de una cantidad grande de información (en el orden de los GB y TB). El presente trabajo describe una herramienta escalable y distribuida para el procesamiento de logs de tráfico de red (en formato pcap) y la generación de reportes gráficos a partir de dichos logs, de tal manera que dichos reportes puedan ser utilizados como parte de procesos de análisis forenses de seguridad informática.

Palabras claves: *procesamiento masivo de datos, archivos pcap, hadoop, librería, logs, tcpdumps, reportes gráficos.*

Abstract

Nowadays, Computer Networks connected to the Internet continue to be compromised and exploited by hackers. This is in spite of the fact that many networks run some type of security mechanism at their connection to the Internet. Large Enterprise Networks, such as the network for a major university, are very inviting targets to hackers who are looking to exploit networks. In the market there are several tools that help the analysis, which could be a possible network attack, but despite of that these tools are available both for free and as owner, do not supply the demand for analysis a large amount of information (in the order of GB and TB). This paper describes a tool for scalable and distributed processing logs of network traffic (pcap format) and generate graphical reports from these logs, so that these reports can be used as part of review processes forensic computer security.

Keywords: *procesamiento masivo de datos, archivos pcap, hadoop, librería, logs, tcpdumps, reportes gráficos.*

1 Introducción

En la actualidad, la mayoría de las redes mantienen conexión con el ancho mundo de la Internet. Ésta conexión representa un constante peligro debido a la vulnerabilidad hacia los ataques que realizan los hackers.

Ésta es en efecto la razón principal por la cual en muchas redes, donde es primordial mantener la confidencialidad y consistencia de sus datos, se ejecutan algunos tipos de mecanismos de seguridad sobre sus conexiones.

Estudios han demostrado que las redes más atacadas son las empresariales [10], otro blanco muy común son las redes universitarias. Debido a esto, los administradores de redes tratan de manera cotidiana de controlar las vulnerabilidades a través de mecanismos que sirven de defensa para las redes en general. Una de estos mecanismos de defensa es el uso de una Honeynet con la cual podemos analizar los medios que se usan para realizar ataques.

En el mercado existen varias herramientas que ayudan al análisis de, lo que podría ser, un posible ataque en la red; sin embargo, a pesar de que dichas herramientas se encuentran disponibles tanto de manera gratuita como propietaria, no abastecen la demanda para el análisis de una cantidad grande de información (en el orden de los GB y TB).

El presente trabajo describe una herramienta escalable y distribuida para el procesamiento de logs de tráfico de red (en formato pcap) y la generación de reportes gráficos a partir de dichos logs, de tal manera que dichos reportes puedan ser utilizados como parte de procesos de análisis forense de seguridad informática.

2 Problema

2.1 Definición

El Capítulo Ecuador del Proyecto Honeynet ha montado una red que permite generar logs con datos obtenidos de las conexiones que se realizan a esta red. Considerando que sobre esta red se han montado "servicios falsos", todo tipo de requerimientos que se realice a uno de estos servicios es considerado como ataque. A partir de esta información se pueden generar reportes de gran utilidad.

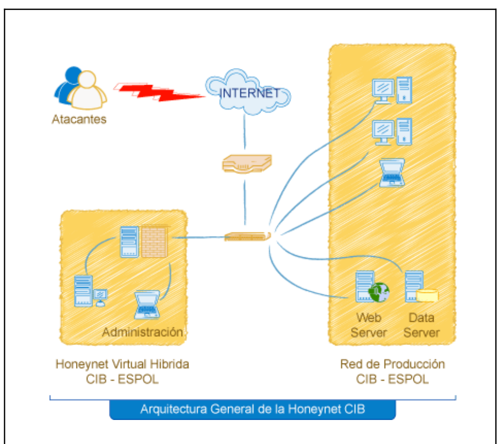

Figura 1.1.1 Arquitectura General de la Honeynet CIB

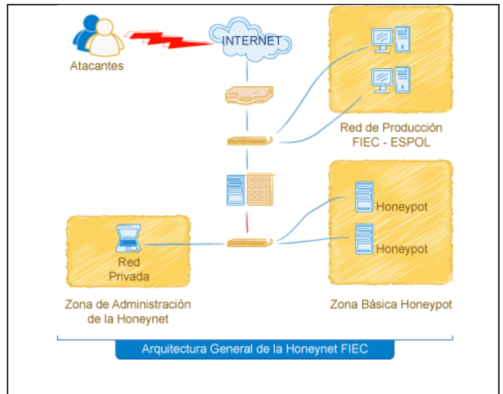

Figura 1.1.2 Arquitectura General de la Honeynet FIEC

Como parte de las actividades de este Capítulo, se realizó un estudio que generó un historial de ataques que se recibieron en dos honeynets instaladas dentro del campus universitario Gustavo Galindo de la ESPOL.

Ambas arquitecturas se las puede visualizar en la Figura 1.1-1 y en la Figura 1.1-2. Durante aproximadamente 4 meses (de Agosto a Noviembre de 2008), se monitoreó y registró todo el tráfico recibido por estas dos honeynets. A partir del tráfico capturado se realizaron varios análisis forenses y se identificaron ciertos patrones de ataque.

Lastimosamente, no se pudo realizar gráficas resumiendo el tráfico de todo el periodo analizado, ya que los registros de los logs al ser muy grandes (mayores a 4 GB) saturaban a las diversas herramientas disponibles en el mercado para este propósito. Un procesamiento tradicional de estos logs tampoco fue realizado, debido al largo tiempo que esto hubiera tomado. Es por esto que surge la necesidad de integrar tecnología de alta escalabilidad, capaz de procesar una gran cantidad de datos y proporcionar prontos resultados.

2.2 Justificación

La necesidad de una herramienta que permita el procesamiento de datos en gran escala, y que además permita el filtrado de datos para generar reportes, ofreciendo una visión global y realista de los posibles tipos de ataques a los que la red es susceptible; es la razón principal para enriquecer el conjunto de datos almacenados en archivos tcpdumps pcap, y poder mostrarlos a través de reportes gráficos que proporcionen información oportuna, precisa y concisa, que pueda ser analizada e interpretada por los administradores de red.

Con la integración de una herramienta de gran escalabilidad, los datos registrados dentro de los log podrán ser procesados y generaran un resultado que apoyará la visualización sistemática de los datos registrados en los archivos tcpdumps.

La implantación de éste módulo (PcapReports), pondrá a disposición de la comunidad, una herramienta que permita la visualización a través de reportes gráficos del comportamiento de las actividades capturadas a lo largo del Proyecto Honeynet.

2.3 Alcances y limitaciones

Este módulo básicamente procesará los datos registrados en los logs del Proyecto Honeynet generando un resultado que será visualizado en un reporte gráfico, y permitirà detallar la información desde lo más general hasta lo más particular posible.

Los datos con los que se cuenta son los proporcionados por el proyecto Honeynet Capítulo Ecuador que corresponden a la Facultad de Ingeniería en Electricidad y Computación (FIEC) y al Centro de Información Bibliotecaria (CIB).

Cabe recalcar, que el presente módulo no se limita a los archivos históricos recolectados hasta el momento. Por el contrario, se espera que conforme se siga monitoreando la red, se pueda continuar retroalimentando el historial de información a tal punto de poder seguir de cerca algún comportamiento malicioso que pueda afectar de alguna manera la seguridad de los datos que se encuentren viajando, y la integridad de los servicios que se encuentren disponibles en la red.

3 Análisis

3.1 ¿Qué estamos resolviendo?

El estudio realizado por el Capítulo Honeynet del Ecuador en una red específica dentro de la ESPOL, dejó como resultado logs tcpdump de aproximadamente 4 GB de tamaño, los mismos que contienen información de los distintos paquetes que fueron emitidos y recibidos por la honeypot a través de la red.

Con la integración de Hadoop, los datos registrados dentro de los logs tcpdump son procesados en gran escala, permitiendo conocer de manera sistemática cuáles son los patrones de ataques sufridos, los tipos de ataques, y las vulnerabilidades a las que la red fue susceptible durante ese periodo de tiempo.

Cabe recalcar, que el módulo desarrollado no se limita a los archivos históricos recolectados hasta el momento. Por el contrario, se espera que conforme se siga monitoreando la red, se pueda continuar retroalimentando el historial de información a tal punto de poder seguir de cerca algún comportamiento malicioso que pueda afectar de alguna manera la seguridad de los datos que se encuentren viajando, y la integridad de los servicios que se encuentren disponibles en la red.

3.2 ¿Por qué lo estamos resolviendo?

Este módulo busca poner a disposición de la comunidad una herramienta que permita visualizar a través de Reportes Gráficos el comportamiento de las actividades capturadas a lo largo del Proyecto Honeynet, permitiendo ir de lo general a lo particular.

La finalidad de estos Gráficos es poder analizar de manera detallada las vulnerabilidades que se presentan en los servidores actuales y que los distintos hackers están explotando. Este tipo de información será de utilidad para mejorar de manera contínua la seguridad en las redes actuales.

3.3 ¿Cómo lo estamos resolviendo?

Hemos utilizado Hadoop como herramienta para el procesamiento masivo de datos, a través de clústers levantados bajo demanda utilizando los servicios de Amazon Web Services. Con esta plataforma se procesa los archivos pcaps directamente en su formato binario, y se generan reportes en formato XML, los cuales pueden ser graficados utilizando una interfaz Web implementada con Adobe Flex. Entre los reportes que se generan, se encuentran:

- **Por Protocolo por cada mes**

En este reporte se puede visualizar el comportamiento mensual de los distintos protocolos, ya sean estos bajo la Capa de Red (IPv4, IPv6, ARP), bajo la Capa de Transporte (TCP, UDP, ICMP), ó bajo la Capa de Aplicación (HTTP, FTP, Telnet, SSH, SMTP, POP3).

- **Tráfico por País**

Debido a que los ataques provienen de cualquier punto geográfico del planeta, la identificación del origen de los mismos da una idea más clara sobre el nivel de fuentes maliciosas que existen alrededor de nuestro entorno. Esto es factible gracias al reconocimiento de origen de la IP fuente del ataque.

- **Tráfico por IP.**

El conocimiento de origen de los ataques a través de la IP fuente, ayuda a que los controles de acceso se encuentren con filtros cada vez más restrictivos, manteniendo bajo estudio aquellas IP que mantengan un comportamiento malicioso.

- **Cantidad de Bytes por día de la semana y hora específica.**

El volumen de datos que se transfieren a través de la red, pueden llegar a ser incalculables en algunos casos. Sin embargo, el poder tener un control sobre la cantidad de ataques que se reciben en una Honeypot durante el tiempo funcional de la misma en la red, ayuda a regularizar el tráfico sobre la misma, y a prevenir el ingreso de ataques específicos que provengan, por ejemplo, de alguna IP determinada.

4 Solución

4.1 Diseño General

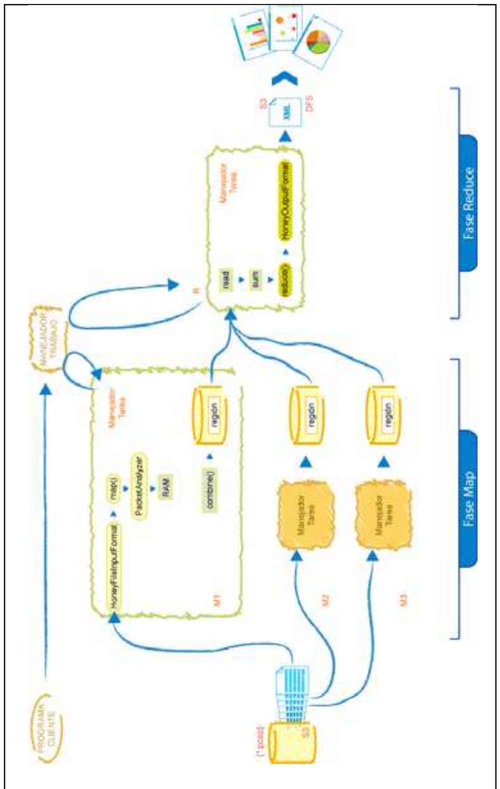

Figura 5.1.1 Diseño Arquitectónico del Módulo

4.1.1 Archivos de entrada

Cada uno de los archivos tcpdumps se almacenan en Amazon S3 para su posterior procesamiento. Hadoop nos provee de métodos para leer archivos de texto. Sin embargo, para poder procesar cada uno de los pcap almacenados en formato binario, fue necesario crear una clase que extienda de FileInputFormat y que pueda leer los datos almacenados en binario. A esta clase se la denominó HoneyFileInputFormat y permite leer paquetes almacenados en archivos pcap, sin permitir que estos sean divididos en el proceso.

4.1.2 El ambiente EC2

Lo componen básicamente dos tareas, la ejecución de los mappers y la ejecución de los reducers.

Cada mapper recibe un archivo de entrada y lo parsea de tal modo que pueda obtener los datos de interés. Es decir, en cada mapper se toma cada uno de los paquetes que se encuentran registrados en el archivo de

entrada y se le extraen los datos necesarios para el procesamiento. Como se manejan cuatro reportes diferentes, se ha definido cuatro mappers independientes para cada uno de ellos, descritos a continuación.
Para los Reportes Gráficos por Protocolo por cada mes, es necesario conocer el protocolo de comunicación que se utilizó durante la transmisión de datos. De igual forma, la fecha de captura del paquete, y el tamaño del mismo. Cada uno de estos datos permitirá en lo posterior realizar un correcto filtrado de información.

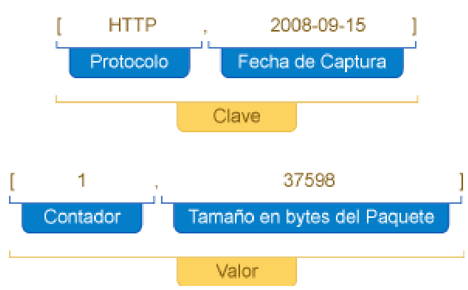

Figura 5.1.2.1 Formato de salida del Mapper de Protocolo por cada mes

Por cada paquete que se parsea en el mapper, se forma un registro con el formato que se presenta en la Figura 5.1.2-1, el cual detalla como está formado el modelo clave/valor. Dentro de la clave, se encuentran el protocolo y la fecha de captura, mientras que en el valor se encuentran un contador y el tamaño en bytes del paquete.

En el caso de los Reportes Gráficos de Tráfico por País, se necesita conocer primero la IP desde la cual se envió dicho paquete. Una vez que se obtiene la IP, se procede a buscar el País correspondiente con ayuda de la librería InetAddressLocator, capaz de transformar la dirección IP en un número entero, y de esta manera poder verificar la procedencia del paquete, adicional al dato del País de procedencia, también se extraen la fecha de captura y el tamaño del paquete.

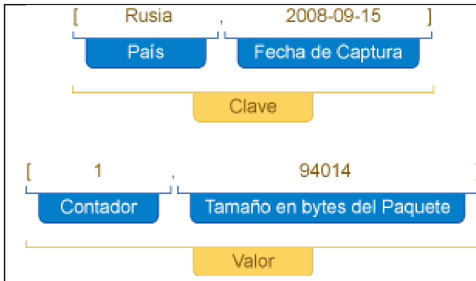

Figura 5.1.2.2 Formato de salida del Mapper de Tráfico por País

Por cada paquete que se parsea en el mapper, se forma un registro con el formato que se presenta en la Figura 5.1.2-2, el cual detalla como está formado el modelo clave/valor. Dentro de la clave, se encuentran el país de procedencia y la fecha de captura, mientras que en el valor se encuentran un contador y el tamaño en bytes del paquete.

Por otra parte, para los Reportes Gráficos de Tráfico por IP, es requerido conocer la IP fuente del paquete, así como la fecha de captura y el tamaño del mismo.

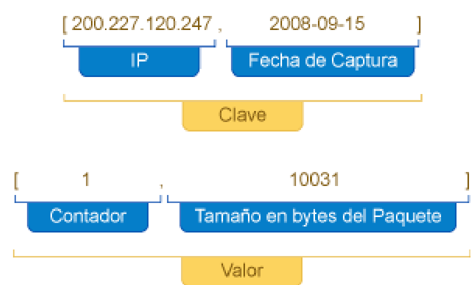

Figura 5.1.2.3 Formato de salida del Mapper de Tráfico por IP

Por cada paquete que se parsea en el mapper, se forma un registro con el formato que se presenta en la Figura 5.1.2-3, la cual detalla como está formado el modelo clave/valor. Dentro de la clave, se encuentran la IP de origen y la fecha de captura, mientras que en el valor se encuentran un contador y el tamaño en bytes del paquete.

Y por último, para los reportes de cantidad de bytes por día de la semana y hora específica, se requieren del día de envío del paquete, la hora de captura y el tamaño del mismo.

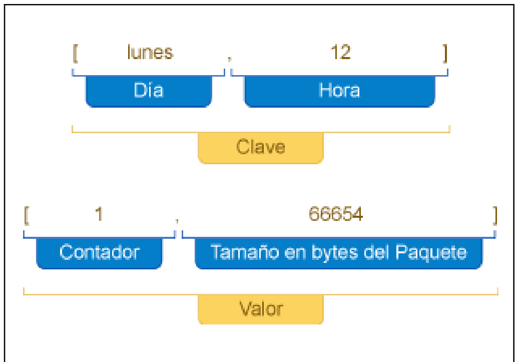

Figura 5.1.2.4 Formato de salida del Mapper de Cantidad de bytes por día de la semana y hora del día

Por cada paquete que se parsea en el mapper, se forma un registro con el formato que se presenta en la Figura 5.1.2-4, la cual detalla como está formado el modelo clave/valor. Dentro de la clave, se encuentran el día de la semana y la hora de captura, mientras que en el valor se encuentran un contador y el tamaño en bytes del paquete.

El proceso de parsear cada archivo de entrada se lo realiza gracias a la librería JNetStream, la misma que está orientada a la interpretación de paquetes de red de manera primitiva. Esta librería no posee métodos definidos para la extracción de datos, razón por la cual se implementó una clase PacketAnalyzer capaz de satisfacer las necesidades con respecto a la información requerida para la correcta generación de cada reporte.

A medida que el modelo clave/valor se va generando a partir de cada mapper, para cada tipo de reporte se va ejecutando el correspondiente reducer, cada uno de ellos tiene la finalidad de agrupar de manera correcta cada valor con su respectiva clave.

Una vez que todos los mappers han entregado sus correspondientes porciones de datos, el reducer procesa dichos datos y los agrupa en un solo resultado final.

4.1.3 Archivos de salida

El resultado del reducer (para nuestro caso) debe de almacenarse en un archivo XML. Para poder realizar el correspondiente almacenamiento, se debió de crear una clase que extienda de FileOutputFormat y que sobrescriba los métodos necesarios para que el resultado final se almacene como archivo XML en Amazon S3.

4.2 Diseño del Front-End

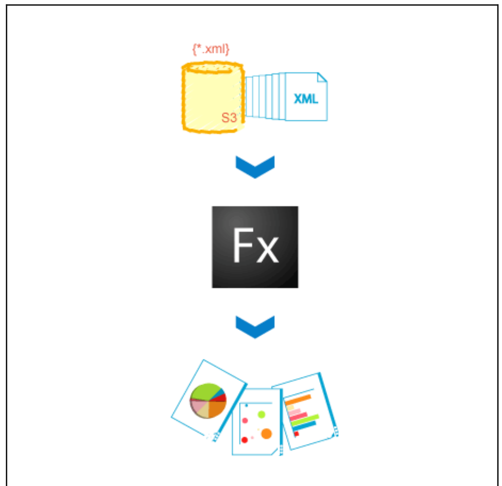

Figura 5.2.1 Diseño Arquitectónico del Front-End

4.2.1 Adobe FLEX

La interacción entre los reportes y el usuario debía ser muy práctica, por lo que el nivel de presentación se optó por desarrollarlo con Adobe FLEX, el cual trabaja usando lenguaje XML.

Mediante Gráficos Estadísticos como Barras, Pastel y Burbujas se puede visualizar, interpretar y analizar el comportamiento de la red, pudiendo detectar posibles ataques por parte de hackers. Cada uno de estos gráficos es el resultado del procesamiento de datos masivos almacenados en archivos tcpdumps.

5 Evaluación

5.1 Pruebas de eficacia

Para realizar estas pruebas fue necesario contrastar el trabajo realizado por el módulo de parseo de archivos *.pcap con otros programas de lectura existentes, sin embargo esto sólo fue posible con archivos en el orden de los MB, para archivos de mayor tamaño se encontraron errores en los programas probados por las limitantes que tienen al depender de las características de memoria y velocidad de procesamiento de una sola máquina.

Figura 7.1.1 Error tomado al abrir archivo de tamaño 1,25GB con Wireshark V

La Figura 7.1.1 muestra el mensaje correspondiente a "Fuera de Memoria" del programa Wireshark, al tratar de analizar la información contenida en el archivo de 1GB del mes de Septiembre de 2008.

Otros programas de distribución gratuita como JpcapDumper, simplemente se cerraron al tratar de analizar el mismo archivo.

5.2 Pruebas de eficiencia

Estas pruebas fueron realizadas en el Amazon EC2 y se han usado diferentes números de nodos para medir el tiempo que se requiere para completar la tarea.

Los resultados los describimos a continuación:

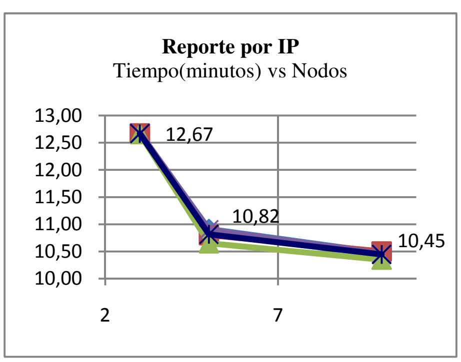

Figura 7.2.1 Reporte Gráfico por IP

Como se muestra en la Figura 7.2.1 se realizaron tres pruebas de procesamiento masivo para el Reporte de IP, teniendo una diferencia de tiempo dentro de un intervalo de +/- 10 seg. Con tres nodos se obtuvo un tiempo de procesamiento de 12,67 seg., en cambio con cinco nodos se obtuvo un tiempo de procesamiento de 10,82 seg., mientras que con diez nodos se obtuvo un tiempo de procesamiento de 10,45 seg.

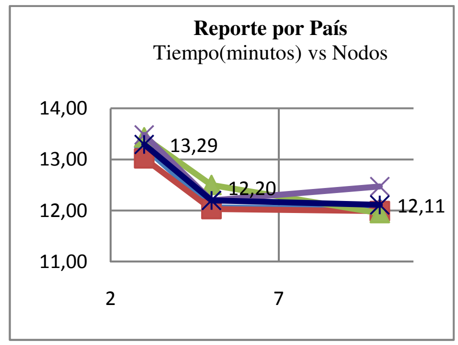

Figura 7.2.2 Reporte Gráfico por País

Como se puede observar en la Figura 7.2.2, en el caso del Reporte por País, se realizaron tres pruebas de procesamiento masivo, obteniendo una diferencia dentro de un intervalo de +/- 10 seg. Con tres nodos se obtuvo un tiempo de procesamiento de 13,29 seg., en cambio con cinco nodos se obtuvo un tiempo de procesamiento de 12,20 seg., mientras que con diez nodos se obtuvo un tiempo de procesamiento de 12,11 seg.

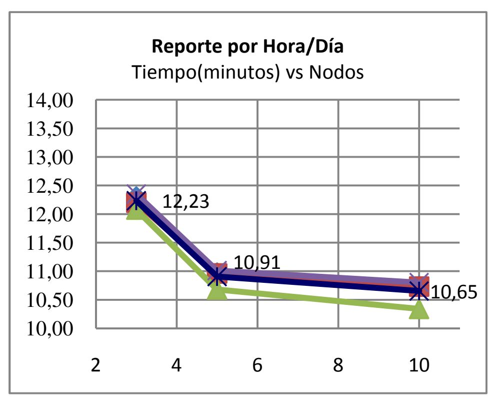

Figura 7.2.3 Reporte Gráfico por Hora y Día

Como se muestra en la Figura 7.2.3, en el caso del Reporte por Hora/Día, se realizaron tres pruebas de procesamiento masivo, obteniendo un intervalo de diferencias de +/- 10 seg. Con tres nodos se obtuvo un tiempo de procesamiento de 12,23 seg., en cambio con cinco nodos se obtuvo un tiempo de procesamiento de 10,91 seg., mientras que con diez nodos se obtuvo un tiempo de procesamiento de 10,65 seg.

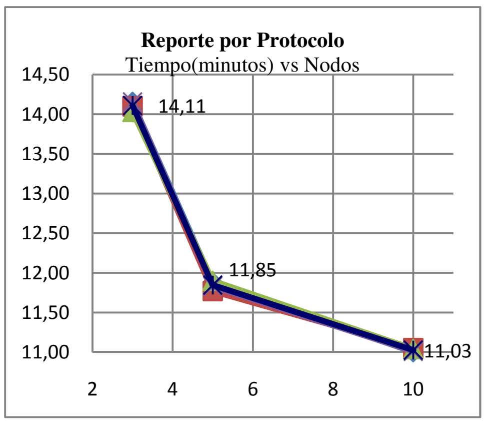

Figura 7.2.4 Reporte por Protocolo

Como se muestra en la Figura 7.2.4, se realizaron tres pruebas de procesamiento masivo para el Reporte por Protocolo, teniendo una diferencia de tiempo dentro de un intervalo de +/- 10 seg. Con tres nodos se obtuvo un tiempo de procesamiento de 12,67 seg., en cambio con cinco nodos se obtuvo un tiempo de procesamiento de 10,82 seg., mientras que con diez nodos se obtuvo un tiempo de procesamiento de 10,45 seg.

En cada gráfico se puede observar el comportamiento del rendimiento del módulo bajo tres condiciones de recursos. De manera general, se observa que a mayor número de nodos, se tiene un menor tiempo de procesamiento. Sin embargo, el correcto aprovechamiento del número de nodos depende del tamaño total de información a procesar, del número de mappers a utilizar y del tamaño configurado para cada chunk.

6 Conclusiones

1. Los reportes gráficos fueron generados bajo un ambiente altamente usable, pues se ha explotado varias características visuales que provee el lenguaje utilizado para la implementacion del front end (MXML, Action Script).

2. A través de cada uno de los reportes, se puede demostrar el resultado del parseo, procesamiento y filtrado de los correspondientes logs. Pudiendo visualizar bajo varias perspectivas, diferentes enfoques de análisis, apoyando el monitoreo de las redes y controlando las debilidades y fortalezas de las mismas, dando soporte a la toma de decisiones y el mejoramiento continuo a la administración de la red.

3. Proveemos una alternativa diferente a las herramientas existentes, debido a que las actuales no brindan soporte para archivos de mayor tamaño (1GB) y mucho menos generar reportes a partir de varios archivos, si estos superan el tamaño soportado por la herramienta.

4. Se desarrolló un módulo que puede ser adaptable a las necesidades de los administradores de red para el procesamiento de los logs en formato pcap.

7 Recomendaciones

1. Los reportes presentados son una muestra de lo que se puede realizar a partir de los archivos de salida generados por map/reduce. Sin embargo, se pueden generar reportes mas complejos, exportarlos a otros tipos de formato e inclusive visualizarlos de una forma diferente utilizando las herramientas adecuadas (muchas herramientas soportan la generación de graficos a partir de un archivo XML).

2. El analisis de archivos pcap en el ambiente Hadoop ya ha sido discutido en foros de Internet , donde no han encontrado una solución viable. Nosotros hemos compartido nuestra propuesta y esperamos una pronta retroalimentacion con el objetivo de mantener el mejoramiento continuo del módulo.

3. Los administradores de red podrían adaptar la propuesta que hemos detallado para generar reportes personalizados a sus necesidades, sin embargo, tendrían que familiarizarse con el ambiente distribuido que utiliza Hadoop para el procesamiento de datos.

8 Referencias

[1] Lance Spitzner, “Honeypots: Tracking Hackers”, Addison Wesley professional, 2002.

[2] Apache. Hadoop. http://lucene.apache.org/hadoop/, 2008

[3] Apache. Pig. http://incubator.apache.org/pig/, 2008

[4] Chu, L., Tang, H., Yang, T., and Shen, K. 2003. Optimizing data aggregation for cluster-based internet services. In Proceedings of the Ninth ACM SIGPLAN Symposium on Prin-ciples and Practice of Parallel Programming, 2003.

[5] Jeffrey Dean, Sanjay Ghemawat: MapReduce: simplified data processing on large clusters. OSDI 2004: 137-149.

[6] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, Den-nis Fetterly: Dryad: distributed data-parallel programs from sequential building blocks. EuroSys 2007: 59-72.

[7] Fay Chang et al, Bigtable: a distributed storage system for structured data, OSDI 2006, 205-218

[8] Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung: The Google file system. SOSP 2003: 29-43.

[9] OpenSolaris. Hadoop. http://opensolaris.org/os/project/livehadoop/, 2008

[10] Herringshaw C., "Detecting attacks on networks", Diciembre 1997.

[11] Jeffrey Dean, Sanjay Ghemawat, "Simplified Data Processing on Large Clusters", Diciembre 2004.