UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

Implementación y Evaluación de Sistema de Monitoreo de Seguridad basado en flujos de paquetes IP

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

FRANCISCO DE BORJA ECHEVERRÍA SIERRALTA

PROFESOR GUÍA:
ALEJANDRO HEVIA ANGULO

MIEMBROS DE LA COMISIÓN:
JOSÉ MIGUEL PIQUER GARDNER
RODRIGO ARENAS ANDRADE

SANTIAGO DE CHILE
OCTUBRE 2008

Resumen

Internet es ampliamente utilizada como medio de comunicación y distribución de información en todo el mundo. La información del tráfico de datos, esto es, información de origen y destino de los paquetes IP, entre las distintas organizaciones, puede aportar información valiosa para la seguridad informática de las mismas.

Utilizando la información del tráfico de una red es posible identificar comportamientos de software maliciosos o "malware". Por ejemplo, una máquina infectada con un gusano (worm) intenta contagiar a un grupo de máquinas a través del tráfico IP. En este caso el tráfico muestra una gran cantidad de intentos de conexiones a direcciones IP distintas.

Aunque ya existen dispositivos de red y sistemas desarrollados para este fin, los sistemas son de carácter privado, funcionan para algunos dispositivos de red o no son fáciles de utilizar.

Durante la memoria se diseñó e implementó un sistema para el análisis offline del tráfico IP proveniente de redes distribuidas. El sistema permite recolectar información de flujos IP proveniente de varias redes geográficamente distribuidas y analizar dicha información para identificar comportamientos maliciosos de computadores en dichas redes. Se evaluó el sistema por medio de dos experimentos, el primero destinado a detectar comportamientos producidos por malware sobre el tráfico IP simulado, el segundo evalúa la efectividad del sistema para la detección de malware en un ambiente real al que se inyecta malware simulado.

En el transcurso de la memoria se encontró una dificultad con respecto a la traducción de direcciones de red hecha por los routers. Para estudiar este problema se configuró un ambiente distribuido y se hicieron consultas especiales.

El sistema desarrollado y los resultados obtenidos dan una base para la creación de nuevos métodos de detección, con el fin de mejorar la seguridad computacional dentro de las organizaciones.

A mi madre, abuela y familia por su
apoyo incondicional

Índice

1 Introducción

La seguridad informática tiene un rol muy importante dentro de las organizaciones con respecto a la confidencialidad, integridad y disponibilidad de la información. En cuanto a confidencialidad, la información importante no debe ser accesible por personas no autorizadas; la integridad garantiza que la información no haya sido modificada; la disponibilidad se refiere a que los servicios o recursos deben encontrarse disponibles cuando se necesiten.

Una amenaza usual a la seguridad informática de las organizaciones hoy en día es el malware. Se denomina malware a un programa computacional cuyo objetivo es infiltrarse y/o dañar un computador sin el conocimiento del usuario o dueño. Ejemplos de estos son los backdoors, botnet, troyanos, worms (gusanos), y el software spyware.

El monitoreo de las redes consiste en analizar el tráfico IP en una red para obtener métricas, medidas de efectividad e información estadística útil. Estas pueden ser utilizadas para mejorar la seguridad informática en las organizaciones. Por ejemplo, una gran cantidad de computadores externos enviando paquetes de red hacia una máquina interna, podría causar una denegación de servicio (DoS, denial of service) a la máquina receptora. Un análisis de tráfico IP puede identificar el origen de dichos paquetes. Con esta información se pueden realizar acciones como por ejemplo, no aceptar paquetes provenientes de la máquina atacante.

Actualmente existen sistemas para analizar el tráfico IP los cuales, sin embargo, son de origen comercial (y por ende su diseño y código es confidencial), están diseñados para dispositivos muy específicos, o son complicados de utilizar y/o limitados a redes locales o de pequeña escala.

Para analizar el tráfico IP es imprescindible almacenar la información útil de los paquetes que transitan sobre una red IP. Esta labor puede ser realizada por un dispositivo de red. Por ejemplo un router capacitado para enviar la información del tráfico IP hacia una máquina receptora.

Netflow es una tecnología utilizada por dispositivos de red Cisco, para coleccionar la información del tráfico IP. La información de tráfico es almacenada utilizando el formato con el mismo nombre en un flujo Netflow. Un flujo IP o flujo Netflow se refieren a un conjunto de paquetes que viajan en sentido unidireccional, compartiendo: IP de origen y destino, así como puertos de origen y destino, y protocolo.

El monitoreo puede ser realizado a diferentes escalas. Desde una red local, hasta un conjunto de estas. Por ejemplo, en una red local determinada puede ser interesante monitorear direcciones recurrentes de destino a una IP particular. Esta información puede

ayudar a detectar la existencia de malware en algunos de los equipos de la red. Si llevamos esto a una escala mayor, donde se relaciona esta información con un conjunto mayor de computadores, esta información puede ser útil, por ejemplo, para saber la magnitud del ataque o los posibles implicados.

El tema de estudio en esta memoria consistió en el diseño, implementación y evaluación de un sistema para el análisis del tráfico IP, utilizando los flujos Netflow provenientes de sensores distribuidos. Este sistema se compone de receptores de flujos IP emitidos por sensores (el sensor puede ser un programa o un dispositivo de red), un repositorio de flujos IP y varias herramientas para analizar los flujos recepcionados. Adicionalmente se implementó una interfaz web para facilitar el análisis de la información del tráfico distribuido.

Actualmente existen herramientas de código abierto capaces de recepcionar, almacenar y analizar los flujos IP. Estas herramientas facilitaron la implementación del sistema permitiendo centrarse en la recepción, integración y el análisis distribuido de los flujos IP y no así la implementación del protocolo y formato de los registros de paquetes.

Se evaluó el sistema mediante dos experimentos, el primero destinado a detectar comportamientos de malware (simulado) sobre el tráfico IP y el segundo evaluó la efectividad del sistema para la detección de malware en un ambiente real.

En el transcurso de la memoria se detectó una dificultad para analizar los flujos IP distribuidos. El problema consistió en no poder identificar las maquinas emisoras de paquetes IP entre dos redes donde al menos exista un router implementando NAT. Se dice que un router implementa NAT (Network Address Translation) si éste conecta a dos redes IP con direcciones incompatibles entre sí, las cuales deben ser traducidas. Este mecanismo de traducción se llama NAT y es constantemente utilizado para interconectar redes privadas con redes públicas. Aún cuando este problema no estaba dentro de los objetivos de esta memoria y por lo importante que es para el análisis de tráfico distribuido se realizó un experimento introductorio al problema. Este experimento, en un ambiente con dos máquinas en segmentos de red diferentes generando tráfico entre ellas, buscó encontrar consultas útiles para identificar una máquina emisora de paquetes IP en una red externa, utilizando la información de tráfico de las dos redes envueltas.

1.1 Motivación

En la actualidad el análisis de seguridad en base al tráfico IP en redes puede llegar a ser muy costoso en tiempo de transferencia y espacio en disco para sistemas de seguridad.

Analizar el tráfico utilizando flujos IP resulta ser de gran ayuda puesto que el espacio en disco es reducido gracias a la disminución de la información de tráfico almacenada, posibilitando una mayor cobertura en el tiempo del tráfico analizado. Reducir el tamaño de la información disminuye el tiempo de transferencia, mejorando la eficiencia en la recepción de la información.

El uso de flujos IP permite mantener la privacidad de la información para los usuarios de la red monitoreada debido a que el flujo IP solo utiliza la información disponible en la cabecera del paquete IP y no el contenido de la información transmitida.

Las herramientas comerciales consultadas para el análisis del tráfico IP en base a flujos IP no son extensibles. Por ejemplo, no permiten agregar consultas específicas, escogidas libremente por el usuario.

Las herramientas de código abierto son muy flexibles y difíciles de utilizar. En algunos casos realizar acciones comunes para experimentar es engorroso y además algunas solo funcionan para tráfico local.

El diseño propuesto pretende facilitar la experimentación con una sintaxis común para la creación de consultas sobre la información de flujos IP. Esta y otras características permitirán contar con una herramienta para analizar la información de tráfico fácilmente.

1.2 Situación Actual

En la actualidad existen sistemas para el monitoreo del tráfico IP, algunos utilizan la información completa del paquete IP y otros utilizan flujos IP como Netflow.

Las herramientas encontradas para el análisis del tráfico IP en base a flujos IP pagadas como “Scrutinizer”, “NetFlow Analyzer 9” o “NetFlow Traffic Analyzer” principalmente permiten realizar monitoreo de redes bastante complejo. Es posible monitorear el uso del ancho de banda, manejan una gran cantidad de sensores, permitir filtros a la información, mapear paquetes a servicios, crear alertas de seguridad y otros. Sin embargo, existen consultas a los flujos IP que ninguna de estas herramienta al parecer permite responder. De acuerdo a la información disponible, ningún software permite agregar consultas específicas, escogidas libremente por el usuario. Por ejemplo, no es posible encontrar máquinas relacionadas a una máquina que se relaciona a una o más direcciones IP maliciosas, sin tener que revisar las máquinas una por una.

Se encontraron herramientas de código abierto para el análisis de flujos IP como Ntop, Nfdump y Silk. Ntop sigue la misma línea de las herramientas comerciales mencionadas, sin la limitante del acceso al código fuente, por lo cual tampoco permite agregar consultas especificas, escogidas libremente por el usuario. Nfdump y Silk permiten hacer un análisis acabado del tráfico IP pudiendo ser tráfico local como distribuido.

El monitoreo de las redes contempla tres etapas:

- Recepción de datos distribuidos de flujos TCP/IP
- Almacenamiento de los datos de flujos TCP/IP
- Análisis de los datos de flujos TCP/IP

Tanto Nfdump como Silk permiten las tres etapas, lo cual posibilita el monitoreo sobre la información de flujos. Sin embargo, son difíciles de utilizar y la estructura propia de estas herramientas hace en algunos casos engorrosa la experimentación con ellas.

Estas herramientas funcionan por medio de la línea de comandos. Al utilizar la línea de comando es necesario estudiar bien la sintaxis de la herramienta para poder empezar a experimentar.

Nfsend es un programa que utiliza Nfdump para las 3 etapas mencionadas y funciona por una interfaz web. Uno de los problemas encontrados en Nfsend es que utiliza solo algunos comandos de Nfdump. Nfsend solo permite hacer filtros simples a la información de flujo. Esta no permitir guardar consultas ni cruzar información entre otros.

Respecto al análisis en base al Netflow, Yiming Gong [6] describe la existencia de cuatro técnicas para la detección de actividades anormales en base a Netflow. Estas técnicas se mencionan en el capítulo 2.

1.3 Objetivo General

El objetivo principal de esta memoria es diseñar, implementar y evaluar un sistema para analizar redes IP, en base al flujo de la comunicación, esto es, el origen/destino de los paquetes de comunicación utilizados, para la identificación y detección de tráficos no autorizados y otros estadísticos similares, como ataques de DoS, actividades de escaneo de puertos, etc., donde la identificación y detección es realizada principalmente en forma batch.

El diseño contempla facilitar las consultas al tráfico IP, almacenar consultas en el sistema, guardar resultados para un posterior análisis y poder almacenar datos externos para luego utilizarlos en consultas. Por ejemplo, una lista de direcciones IP es útil poder almacenarla para así poder utilizar esta información en nuevas consultas.

1.4 Objetivos Específicos

1.4.1 Diseño e implementación

- Selección de herramientas de código abierto más apropiadas para su complementación.
- Diseño sistema utilizando herramientas de código abierto.
- Definición de consultas para la detección de malware. Al referirse a consultas se refiere a preguntas cuya respuesta es información usualmente agregada, cuyo cálculo se realiza a partir de la información de flujo IP. Por ejemplo una consulta posible es cuantas máquinas están enviando más tráfico del que reciben.
- Diseño e implementación de sistema de almacenamiento de nuevas consultas mediante un módulo especializado.
- Visualizar información de forma externa, esto es, a través de un "portal".
- Estimación de tiempo de procesamiento, para análisis del tráfico IP.
- Estimación de espacio en disco para almacenamiento del tráfico IP.

1.4.2 Evaluación

- Explorar nuevas consultas interesantes desde una perspectiva de seguridad informática.
- Evaluar efectividad del sistema como mecanismo de detección de anomalías de seguridad computacional (malware).

1.5 Trabajo Realizado

En una primera etapa se investigaron herramientas de código abierto útiles para el análisis del tráfico IP como son Nfdump y Silk. Estas implementan la recepción, almacenamiento y análisis del flujo IP. Se definieron requisitos funcionales para los objetivos planteados y de acuerdo a éstos se evaluaron los sistemas. En base a las conclusiones y la definición de la herramienta más adecuada para analizar el tráfico IP, se diseñó el sistema utilizando funcionalidades propias de Nfdump y una base de datos relacional.

Implementado el sistema, se realizaron pruebas sobre el tiempo de procesamiento, tiempos de inserción y espacio en disco para distintos tamaños de registros.

Así mismo se especificaron consultas, basadas en técnicas existentes para la detección de anomalías, tanto en redes centralizadas (locales), como distribuidas.

Posteriormente se plantearon dos experimentos. El primero destinado a detectar comportamientos de malware sobre el tráfico IP generado artificialmente. Se realizaron cuatro simulaciones de tráfico IP acorde a comportamientos planteados. El segundo experimento evaluó la efectividad del sistema para la detección de malware en un ambiente real con datos provenientes de 2 semanas de almacenamiento de tráfico IP.

En el transcurso de la memoria se encontró un problema de gran importancia para el análisis del tráfico IP distribuido. Este problema afecta a las redes donde existe un dispositivo de red que esté utilizando el mecanismo de traducción de direcciones IP (NAT). Este mecanismo modifica los headers de los paquetes IP lo cual causa que se pierda la referencia de la máquina emisora del paquete IP en los datos de la red que recepciona los paquetes emitidos. Esto dificulta ostensiblemente el análisis de información de flujos IP. Para resolver este problema se realizó un experimento exploratorio en busca de consultas para poder relacionar los flujos emitidos y recibidos en dos redes distintas.

Finalmente, producto del trabajo realizado, se plantearon futuras mejoras en términos de eficiencia en la implementación, y una identificación de máquinas comprometidas más efectiva.

1.6 Estructura de la Memoria.

El capítulo 2 contiene información relevante para el desarrollo de esta memoria en cuanto a conceptos de seguridad informática, flujos Netflow, técnicas de análisis sobre Netflow, redes de bots (Botnets) y estructura del protocolo IP. El capítulo 3 detalla los pasos seguidos para el diseño e implementación del sistema. El capítulo 4 describe las técnicas para la detección de anomalías (malware) en base al flujo de paquetes, utilizándolas para la creación de nuevas consultas de seguridad, tanto en un ambiente local como distribuido. El capítulo 5 contiene la evaluación del sistema mediante dos experimentos: detección de malware sobre tráfico IP y efectividad de métodos de detección en ambiente real. El capítulo 6 describe el problema de análisis de flujos IP sobre redes con NAT, junto a una experimentación exploratoria del problema. El capítulo 7 describe trabajos futuros y propone extensiones a la presente memoria.

2 Antecedentes

En este cápitulo se hace una introducción a distintos conceptos importantes para el entendimiento de la memoria.

En primer lugar se describen las redes IP. Posteriormente conceptos de la seguridad informática son introducidos. Luego se describen tipos de malware los cuales el sistema busca detectar. En particular, describimos algunas características de los botnets en términos de tráfico IP.

Seguidamente se describe el protocolo Netflow utilizado por el sistema implementado. Luego se introducen los sistemas para el monitoreo en base al tráfico IP aportando con la estructura general utilizada por estos sistemas, la cual se tomará en cuenta para el diseño de la aplicación.

Adicionalmente se muestran cuatro técnicas actuales usadas en la detección de malware a partir de los flujos IP.

Finalmente se discute en más detalle, las características elementales de los paquetes que transitan en el tráfico IP, dando una base para la creación de nuevas consultas.

2.1 Redes IP

El estándar de comunicación de red más utilizado en la actualidad es el protocolo TCP/IP. Este se divide por capas de modo tal que se pueda sustituir una capa por otra equivalente, esto evita la sustitución del hardware y todo el software.

El estándar utilizado para el transporte de información son los paquetes UDP y TCP ambos utilizan IP (Internet Protocol). Tanto como TCP y UDP son estructuras formales con distintas características, TCP es utilizado para transacciones en las que se necesita un control permanente de la comunicación, UDP se utiliza generalmente para la transmisión de información rápidamente y poco confiable (ej, video, radio).

El proceso del envío de paquetes se asemeja al envío de cartas. Una carta para ser enviada necesita una dirección de origen y una de destino. El receptor de la carta por medio de la dirección de origen sabrá a quien responder. Las cartas enviadas no llegan directamente desde una dirección a otra, si no que pasan por diferentes agentes antes de llegar a su destino. Los paquetes IP se comportan de forma similar. Un paquete enviado no llega directamente al destino, si no que es transmitido a una dirección que sabe cómo hacer llegar el paquete a su destino. Cuando éste llega a destino el destinatario puede saber la dirección del emisor del paquete IP.

La dirección de origen y destino usada por el protocolo de internet se llama dirección IP. Además de esta información el paquete IP contiene más información propia del protocolo, la cual está detallada en el punto 2.9 de esta sección.

2.2 Seguridad en la Red

Las redes IP no solo han ayudado a aumentar y facilitar las comunicaciones globales, sino que también han posibilitado llegar a la información sensible de diferentes organizaciones a través de la red.

La seguridad informática vela por la confidencialidad, integridad y disponibilidad de las redes de las organizaciones. Para esto existen herramientas como algoritmos de encriptación, firmas digitales, políticas informáticas en las redes y hardware especializados.

Algunas de las herramientas especializadas para la seguridad son:

- Los firewall o cortafuegos permiten controlar las comunicaciones permitiéndolas o bloqueándolas dependiendo de las políticas propias de las organizaciones. Los firewalls son generalmente ubicados entre la red local y la red externa.

- Los IDS (sistemas de detección de intrusos, por su sigla en Inglés) son utilizados para la detección "online" de accesos no-autorizados, utilizando la información de los paquetes IP y contrastándola con comportamientos sospechosos conocidos.

- Sistemas de monitoreo en base al flujo Netflow aportan información de forma online o batch sobre el tráfico IP. Esta información puede ser analizada para detectar patrones de comportamiento malicioso.

2.3 Malware

Se denomina malware a un tipo de software intrusivo y malicioso. Este busca comprometer máquinas para poder abusar de éstas o dejarlas incapacitadas de cumplir su función.

- Gusanos (worms): son programas de carácter malicioso. Buscan afectar la máquina y luego de afectarla, afectar a las máquinas contiguas a esta. Para esto se autocopia e implanta en las máquinas. Generalmente los gusanos envían paquetes al azar intentando conectarse a las máquinas para infectarlas.

  En el contexto del tráfico IP este se verá afectado por paquetes enviados a rangos de IP y puertos aleatorios.

- Backdoor: software creado para permitir al creador tener acceso al sistema infectado, por medio de una puerta trasera. Tráfico poco frecuente en cierto puerto o sobre máquinas poco accesadas, son frecuentemente indicios de anomalías.

- Bot: software creado para realizar diversas acciones, imitando el comportamiento de un humano.

- Botnet: es una red de bots comandados desde un bot central. Este malware se describe más en detalle en la sección siguiente debido a su relevancia dentro de la memoria.

2.4 Botnet

Es una colección de software robots (o bots) que funcionan de forma autónoma. Tienen la capacidad de recibir comandos desde un bot central controlado por el atacante. También pueden infectar máquinas para implantarles otros bots, aumentando el número de bots en la botnet.

Estos bots generan comportamientos variados sobre el tráfico IP, dependiendo de su forma de comunicación y expansión. Algunas botnet funcionan de forma centralizada. Para este caso, el tráfico de la botnet es representado por un conjunto de máquinas que se comunican con una máquina especial, o viceversa, la máquina comandante se comunica con todas las máquinas en su botnet. De esta forma la cabeza de la botnet (herder) puede organizar un ataque.

Los bots traen su propio mecanismo de expansión. Este es similar al de los gusanos. Por ejemplo, utiliza técnicas de escaneo de redes en busca de máquinas y así transmitir el bot.

Las botnet son generalmente utilizadas para fines maliciosos. Pueden realizar ataques como los denominados ataques de denegación de servicios distribuidos (distributed denial of service, DDoS). Estos ataques buscan sobrecargar los recursos de la máquina atacada, en algunos casos mediante el envío masivo de paquetes IP.

2.4.1 Mecanismos de Control y Comando (C&C)

Las botnets poseen un mecanismo para su control llamado mecanismo de control y comando (C&C). El atacante en control de la botnet envía comandos al controlador de la bot y éste los envía a los bots.

Los medios de comunicación más utilizados por los bots y el C&C son:

1) IRC: Algunos *bot* están diseñados para conectarse a un servidor IRC y recibir comando a través de los canales de éste. El C&C puede enviar órdenes por el canal creado para la botnet. Este tipo de botnet son las más utilizadas en la actualidad. Los

botnet suelen ser combatidos en base a políticas como por ejemplo no permitir tráfico por el puerto predeterminado de IRC (6667).

2) Http: Los *bot* son capacitados para leer cada cierto tiempo alguna dirección en especial. El mecanismo de C&C publica en la dirección los comandos a ejecutar. Estos son más difíciles de encontrar debido a la utilización del puerto 80 (puerto utilizado por servidores web).

3) P2P: Estos *bot* se basan en la utilización de técnicas p2p para controlar y comandar, dejando de ser un mecanismo centralizado. Esto se debe a que los nodos pueden actuar como cliente o servidor.

En algunos casos los mismos bot pasan a ser C&C, aumentando la dificultad para detectar la botnet.

Los casos donde el controlador es centralizado, la botnet suele ser deshabilitada eliminando el bot controlador.

2.4.2 Formas de Propagación

Las bot cuentan generalmente con algún mecanismo de propagación. Para esto usualmente escanean rangos de direcciones IP intentando realizar alguna conexión con otras máquinas, y así poder infectarlas. La infección es lograda por medio de vulnerabilidades en los sistemas escaneados.

Las vulnerabilidades pueden ser de distinto tipo. A veces son fallas en la seguridad de servicios propios de los sistemas operativos, programas específicos o sólo servicios mal configurados o habilitados por error.

2.4.3 Formas de Detección Mediante el Tráfico IP

Como algunas botnets requieren comunicación por IRC, monitorear los puertos utilizados por IRC es una práctica recomendable (pero usualmente no suficiente). IRC por defecto utiliza principalmente el puerto 6667 y generalmente los comprendidos en el rango 6660-6669.

Utilizando una lista de servidores C&C reconocidos como maliciosos, es posible monitorear el tráfico entre las redes y los servidores. Esto ayudara a tener el conocimiento de las máquinas comprometidas.

Los botnet intentan muchas veces utilizar los puertos para el intercambio de archivos de Windows (puerto 135, 139 y 445) para expandirse. De la misma manera, el monitoreo de estos puertos junto a relaciones entre bot, mejoran la detección de la botnet.

La identificación de un comportamiento similar de un grupo de máquinas en un periodo de tiempo pequeño entrega indicios, en algunos casos, de la presencia de una botnet. Generalmente las Botnet pueden cambiar la dirección IP del C&C constantemente. Para esto utilizan servidores de DNS los cuales están actualizando constantemente la dirección IP asociada al nombre del dominio del C&C (técnica llamada DNS flux). Por ejemplo, un comportamiento usual es el envío de paquetes DNS desde un grupo de máquinas a un mismo servidor de DNS y luego las mismas máquinas envían paquetes IP hacia una dirección IP en común.

2.5 Netflow

Netflow es un protocolo desarrollado por Cisco. Para agrupar la información del tráfico. Esta información es exportada mediante el uso de datagramas UDP o SCTP hacia máquinas recolectoras del flujo Netflow.

Cada paquete IP que transita por un router o switch, que soporte Netflow, es examinado por atributos (campos IP) determinados. Estos atributos en su conjunto actúan como "huellas digitales" de los paquetes. Con estos se determina si estos son únicos o similares a otros paquetes. Los atributos son:

1) Dirección IP origen
2) Dirección IP destino
3) Puerto origen
4) Puerto destino
5) Tipo de protocolo de capa 3
6) Interface de ingreso
7) Tipo de servicio IP (type of service,TOS)

Todos los paquetes con los mismos valores de los atributos son agrupados en un flujo IP denominado flujo Netflow. Junto a los atributos almacenados se agregan el número total de paquetes observados y la suma total de bytes de los paquetes pertenecientes al flujo Netflow, timestamp para el inicio y término del flujo y la agrupación de los TCP flags en el flujo (sólo protocolo TCP) entre otros.

El uso de esta técnica para almacenar la información básica del tráfico IP disminuye el tamaño de la información recolectada. Esto logra una transmisión más rápida y permite un mayor almacenamiento de la información de tráfico. Si esta información es comparada con el almacenamiento y transmisión de información de tráfico por cada paquete, es claro que la información almacenada y transmitida para Netflow es mucho menor.

Las características mencionadas hacen posible el almacenamiento histórico de la información del tráfico IP, permitiendo el análisis posterior de la información.

Existen varias versiones del protocolo de red Netflow. La versión más utilizada es la versión 5 y permite el envío de los flujos Netflow utilizando el protocolo UDP. Actualmente se encuentra en la versión 9 y esta permite además de utilizar UDP como protocolo de transporte, utilizar el protocolo SCTP. El protocolo SCTP provee confiabilidad, control de flujo y secuenciación como TCP. SCTP también permite el envío de mensajes fuera de orden. Este es un protocolo orientado al mensaje (parecido al envío de datagramas UDP). Además la versión 9 permite incluir opcionalmente más información de los paquetes IP en el flujo Netflow.

Netflow ha surgido como un estándar y es llamado IPFIX (Internet Protocol Flow Information eXport) y se basa en la versión 9 de Netflow. Este estándar también es utilizado por otros dispositivos de red como los routers "Juniper".

2.6 Monitoreo de Paquetes IP

Una de las técnicas utilizadas para la detección de malware es el descubrimiento de anomalías en base al tráfico IP, usando la información del tráfico IP generado por una organización para deducir, detectar y prevenir comportamientos de malware.

Los gusanos por ejemplo buscan máquinas en la red para transmitirse. Estos envían paquetes de conexión a un rango de direcciones IP probables de existir y a distintos puertos, con la finalidad de realizar conexiones exitosas con las máquinas. Si se analiza la información del tráfico IP, se verá expresado los intentos de conexión de una máquina hacia un grupo de máquinas en un periodo de tiempo pequeño. Esto significará que probablemente estemos en presencia de un gusano en la red.

2.7 Sistemas para el Monitoreo de Paquetes IP

Existe una gran cantidad de aplicaciones para utilizar flujos IP. Existen aplicaciones propietarias de Cisco, de otros vendedores y herramientas de código abierto.

Las aplicaciones de código abierto varían en muchos factores, por ejemplo, no todas soportan todas las versiones de flujos IP (Netflow, IPFIX, sFlow). La forma de almacenar los datos en algunos casos es guardándolos en tablas en bases de datos. En otros casos, se utilizan archivos propios para guardar la información. También hay aplicaciones que no soportan múltiples exportadores de flujos.

La estructura básica de los sistemas para el monitoreo de flujos es la mostrada en la figura 1.

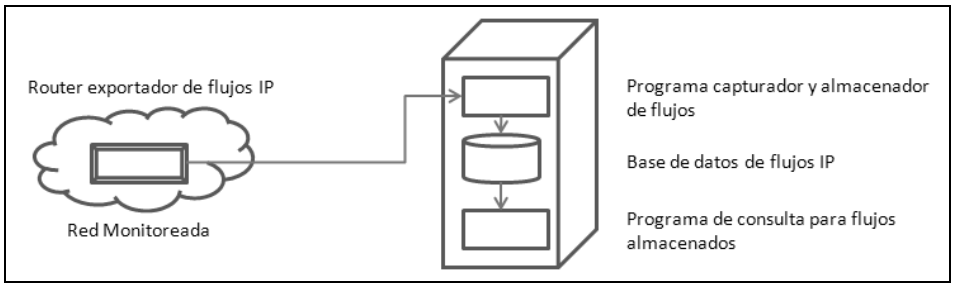

Figura 1: Estructura de sistema basado en flujos IP.

Existe un exportador de flujos IP inmerso en el tráfico IP de la red monitoreada. El exportador puede ser un programa o dispositivo de red (router) que transmite la información del tráfico IP hacia una máquina denominada "Colector de flujos". Los programas utilizados para exportar los flujos IP utilizan la interfaz de red en modo promiscuo (leer todos los paquetes que lleguen a la interface de red) para leer los paquetes IP y luego crear los flujos IP. Los exportadores (sensores) pueden ser configurados para transmitir a una IP determinada y a un puerto determinado la información de tráfico.

La máquina colectora recepciona y almacena los flujos. Para esto existe un proceso esperando los flujos y almacenándolos en el disco duro de la máquina.

Los flujos almacenados pueden ser consultados por medio de una herramienta especializada en el análisis de flujos IP.

2.8 Técnicas para la Detección de Malware

Un punto importante al analizar el flujo es conocer las técnicas más utilizadas para la detección de anomalías.

Las técnicas más utilizadas son Top N / Baseline, calce de patrones, TCP flags e ICMP.

2.8.1 Top N / *Baseline* sobre flujos IP

Un baseline de la actividad de red es una descripción de la actividad de red normal de acuerdo al historial de tráfico. Por ejemplo, un baseline típico es el promedio de conexiones realizadas por una máquina en la red.

Un top N rankea datos en base a una característica especial. Por ejemplo, un top N para un grupo de máquinas utilizando la característica del número total de paquetes enviados por máquina, genera una lista con las máquinas ordenadas de mayor a menor, con respecto al número total de paquetes enviados.

Análisis Top N / Baseline es la técnica más común y básica para analizar flujos. La idea es analizar los flujos que tengan alguna característica sobresaliente (top N), especialmente el valor de los campos que se desvían significativamente del comportamiento "normal" (baseline)

Normalmente hay dos maneras de hacer uso del método Top N / Baseline:

- Top N sesión

La estrategia "Top N sesión" identifica anomalías a través de ordenar los flujos por dirección de destino u origen, y así detectar patrones distintos al normal establecido por el baseline. Esta estrategia deja en evidencia la presencia de nuevos gusanos, ataques Dos/DDos, escaneo de red o cierta clase de abusos de red.

Si una máquina está infectada con un gusano, ésta actuará completamente diferente al baseline normal. La máquina tratará de conectarse a otros host para infectarlos, logrando que las peticiones de conexión aumenten. Por la misma razón cuando alguien esté escaneando un gran bloque de direcciones en busca de servicios vulnerables, se verá especialmente un alto volumen de sesiones enviadas por una sola dirección IP.

- Top N data

Este método puede ser definido como una gran transferencia de datos (por ejemplo, cantidad de paquetes IP) transferidos en un cierto periodo de tiempo, entre dos máquinas, o una máquina y un bloque de direcciones IP. Por ejemplo las máquinas que transfieren más cantidad de datos hacia el exterior deben ser rankeadas. Se podrían rankear por cantidad de bytes entrantes o bytes salientes. Si los valores difieren del baseline normal una alerta debiera ser invocada. Uno de los motivos posibles para esta diferencia es la existencia de un gusano copando el ancho de banda.

2.8.2 Calce de Patrones

El calce de patrones se refiere a la detección de ciertas cadenas específicas (patrones) dentro de los datos de flujo.

Los campos de IP y puertos, de origen y destinos son los más ocupados para el calce de patrones. Los flujos serán examinados y los host asociados a campos de flujos sospechosos serán alertados dependiendo de los criterios especificados.

- Calce de puerto

La manera más común de ocupar el calce de puerto para la detección de anomalías es cuando se conoce un puerto común de uso de un malware determinado. Por ejemplo, el puerto 1434 es el utilizado por "SQL Slammer" (gusano el cual causa un DoS a las máquinas

afectadas), entonces, si se quisiera encontrar este ataque se deberá analizar la información buscando el puerto 1434. Al encontrar rastros de utilización del puerto será recomendable analizar las máquinas. Un problema con esta solución es que es posible que algún programa no malicioso también ocupe el puerto, disminuyendo la efectividad para la detección de malware.

- Calce de dirección IP

Existen varias formas de utilizar el calce de direcciones:

- Calce de direcciones reservadas de la IANA (autoridad encargada de la asignación de números de internet).

La IANA tiene un gran número de bloques de direcciones reservados, los que no deben ser utilizados para un ruteo global, por lo que si encontramos cualquier flujo con estas direcciones, es indicador de actividad maliciosa.

El usuario debiera tener en cuenta que a veces no se puede rastrear la ruta de origen. Esto se debe a que usualmente los malware pueden cambiar su dirección de origen intencionalmente, por lo que se deberá recurrir a otro método para el rastreo del origen de los flujos maliciosos.

- Calce con direcciones especiales

Existen diferentes reglas para considerar los flujos anormales. Estas dependerán de las reglas internas de la organización y de las limitantes de las redes. A continuación se muestran solo algunas de esta:

- Tráfico Saliente a la red local

Cualquier flujo para el tráfico saliente donde la dirección de origen no sea parte de las máquinas del segmento, podrá ser considerado anormal.

- Tráfico Entrante a la red local

En casos especiales por políticas internas la red solo puede recibir flujos de ciertas direcciones. Esto es, cualquier flujo donde la dirección de origen que no sea parte de las direcciones descritas para el tráfico entrante, será considerado anormal.

- Direcciones Fijas

Algunas clases de anomalías tendrán una o más direcciones IP fijas a utilizar. Por ejemplo, W32/Netsky.c (gusano de envió masivo de correos) después de infectar envía consultas DNS a direcciones especificas. Por lo cual las máquinas que contengan tráfico con esas direcciones podrían estar comprometidas.

2.8.3 TCP flags para Netflow

Una de las tareas más difíciles cuando se hace análisis basado en los flujos, es que el administrador debe evaluar una gran cantidad de datos (flujos). Si sólo ocupara los métodos de Top N / Baseline, y calce de patrones, este estaría limitado a ver solo una porción de las anomalías de la red.

Varias veces se ha visto que en el flujo legítimo, se han encontrado rastros de gusanos y otras actividades anormales. En vista de esto, se requerirán otras maneras para la identificación de anormalidades. Una de estas es el análisis basado en los flags del paquete TCP. Esta información está contenida en los flujos IP. Con esta información se puede incrementar la eficiencia en la detección de malware mediante los TCP flags.

Por su naturaleza de replicación los gusanos están programados para buscar la mayor cantidad de víctimas; típicamente un gusano manda cientos de mensajes a grandes bloques de direcciones IP en un corto periodo de tiempo. En particular, si uno de estos estuviera diseñado para esparcirse vía TCP, durante su propagación éste mandará una gran cantidad de paquetes TCP con el flag SYN activado, intentando encontrar servicios vulnerables en otros host. El flag SYN en particular es utilizado para indicar inicio de conexión.

Los paquetes TCP tienen asociados distintos flags donde son utilizados para indicar el status o condición de la conexión. TCP ocupa el método llamado “three way handshake” para realizar una conexión.

2.8.3.1 Escenarios para “three way handshake”

En esta sección se detallan los tres posibles escenarios de conexión, especificando su uso para la técnica sobre TCP flags.

1) El host de destino existe y hay un servicio corriendo en el puerto objetivo.

Para esto la máquina con el gusano intentará establecer una conexión con la máquina afectada mediante el mecanismo de “three way handshake” especificado en el protocolo TCP.

- El cliente manda un paquete TCP con el flag SYN activado al host de destino
- El host de destino manda de vuelta un paquete con la activación de los flags SYN/ACK
- El cliente reconoce que el host lo reconoce con un ACK

Finalmente la conexión es establecida entre las máquinas. Si la conexión fue iniciada por el gusano y el servicio accedido puede ser explotado.

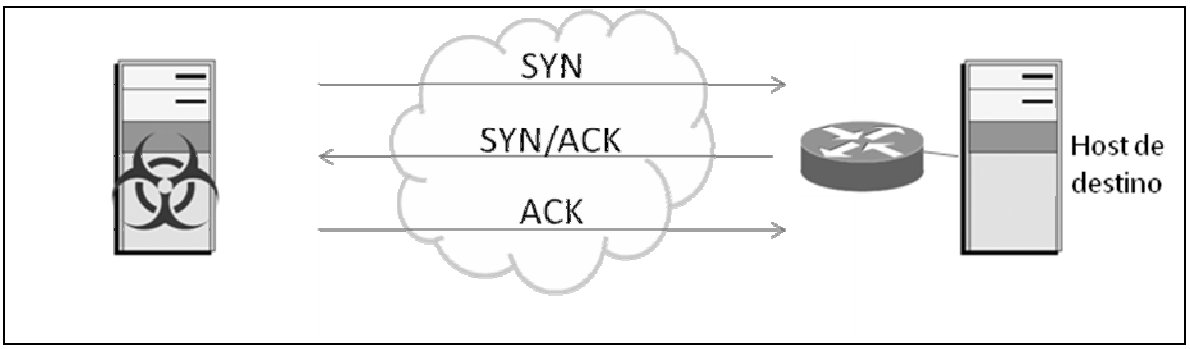

Figura 2: El host de destino existe y el puerto se encuentra abierto.

Usando flujos IP se esperará ver métodos para analizar combinaciones de TCP flags como ACK/SYN/FIN en los registros de ambas direcciones. Por ejemplo, un flujo perteneciente a una conexión de inicio a fin, contará con por lo menos los flag ACK/SYN/FIN. Supongamos que inicialmente una máquina emisora envía un paquete con el flag SYN activado pidiendo conectarse, luego en los siguientes paquetes se vera el flag ACK activado debido a la comunicación con la otra máquina y finalmente si la máquina termina la conexión se vera el flag FIN activado, luego el flujo tendra los tres flags acumulados en la información del flujo.

2) La segunda posibilidad resulta de los intentos de conexión que no recibieron respuesta, dado que no existía el host de destino.

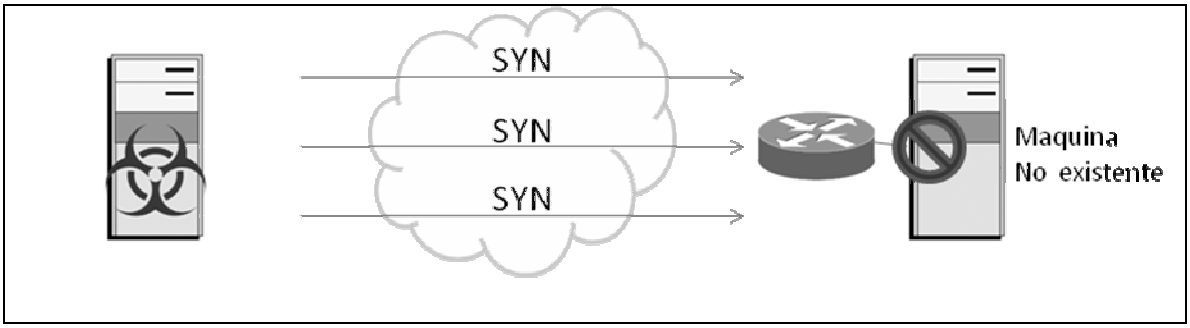

Figura 3: conexión a un host de destino no existente

Desde la perspectiva de flujos IP, se analizarán los flujos en los cuales solo el bit SYN fue configurado por el host del gusano y no recibieron respuesta.

3) La tercera posibilidad resulta que el host de destino existe pero el puerto buscado está cerrado.

Primero se manda un mensaje SYN y posteriormente el host de destino manda un mensaje RST/ACK. Acorde a las implementaciones TCP el host deberá parar cualquier próxima conexión TCP a ese puerto.

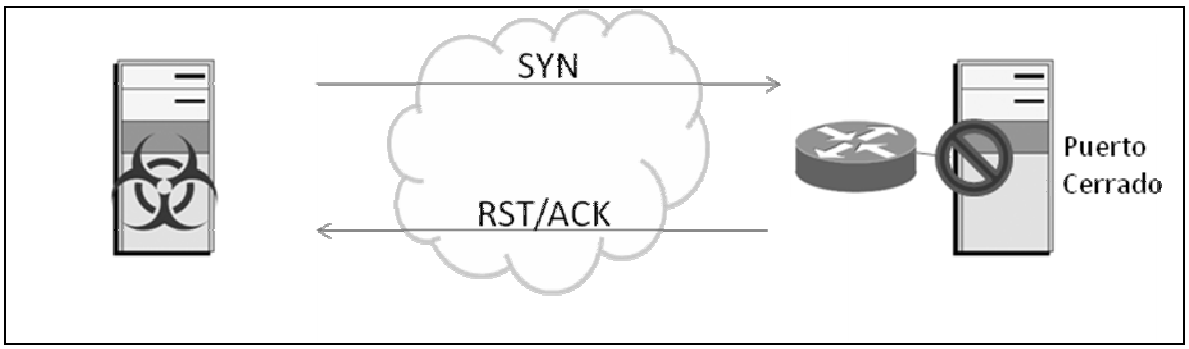

Figura 4: El host de destino existe y el puerto se encuentra cerrado.

Del punto de vista de flujos el host del gusano mostrará en los registros solo peticiones SYN hacia el host de destino y respuestas RST/ACK hacia el host del gusano.

En general se esperará ver una gran cantidad de paquetes de respuesta con los bits RST y ACK asociado a los records del host del gusano.

2.8.4 ICMP

El propósito de ICMP es proveer feedback automático de problemas en la red. Los mensajes ICMP en los flujos IP pueden también ser usados para encontrar los potenciales host maliciosos.

El tipo y código de ICMP son almacenados en los flujos IP, estos son registrados en el puerto de destino en los registros.

srcIP	dstIP	prot	srcPort	dstPort	octets	packets
135.169.9.116	137.54.111.144	ICMP	0	2048	28	1
135.169.9.116	137.62.249.241	ICMP	0	2048	28	1

Tabla 1: Ejemplo de información de flujos.

Por ejemplo, cuando el valor del protocolo es ICMP y el puerto de destino fuera 2048 (hexadecimal 800) quiere decir que el tipo de ICMP es 8 y el código es 00 (sin código), con lo que se concluye que el paquete es un "ICMP echo" de petición (ping).

De la misma manera un tipo 769 (301 hexadecimal) es un ICMP de tipo 3 y código 01, lo que significa "ICMP host inalcanzable".

Hay dos tipos interesantes de paquetes ICMP que pueden ser usados para la detección de flujos anormales cuando analizamos los flujos entrantes. Estos son ICMP destino inalcanzable y ICMP puerto inalcanzable.

2.8.4.1 ICMP destino inalcanzable y ICMP puerto inalcanzable

De acuerdo a la implementación de ICMP, si la red de destino o el host de destino son inalcanzables, el Gateway (equipo encargado de conectar dos redes) mandará mensajes de destino inalcanzable al host de origen.

Para peticiones UDP, máquinas con puertos cerrados podrían mandar mensajes de vuelta "ICMP puerto inalcanzable" hacia el host de origen. Si un gusano se esparciera con UDP, este gatillaría una gran cantidad de flujo de "ICMP puerto inalcanzable".

En resumen si un host tiene un volumen anormal de ICMP port/host/network inalcanzable, esto podría indicar que el host está actuando de forma anormal.

2.8.4.2 Método de calce de patrones para ICMP

Algunos gusanos y ataques de red utilizan ICMP. Por ejemplo, una máquina infectada con el gusano W32.Nachi.worm enviará paquetes ICMP echo (tipo 8) de largo fijo 92 bytes, simplemente, se necesitará aplicar un filtro a los paquetes que tengan ICMP tipo 8 con un largo de 92 bytes. Producto de esto, los host infectados con este gusano serán identificados.

2.9 Protocolo TCP/IP

En esta sección se describió más a fondo la composición de los paquetes que conforman el tráfico IP. El objetivo es ilustrar y entender los campos utilizados en el flujo IP en las siguientes secciones.

Lo primero es saber que el protocolo TCP/IP está compuesto de 4 capas, las cuales están relacionadas en cuanto a sus capacidades.

Capa	Nombre	Descripción
4	**Aplicación.**	**FTP** (**F**ile **T**ransfer **P**rotocol) **HTTP** (**H**yper **T**ext **T**ransfer **P**rotocol) **SSH** (**S**ecure **Sh**ell) **BitTorrent**
3	**Transporte.**	**TCP** (**T**ransmision **C**ontrol **P**rotocol) **UDP** (**U**ser **D**atagram **P**rotocol), **SCTP** (**S**tream **C**ontrol **T**ransmision **P**rotococol)
2	**Red.**	**IP** (incluyendo **IPv4** e **IPv6**) **ICMP** el sub-protocolo de IP
1	**Físico.**	Medio físico.

Tabla 2: Modelo de Capas de TCP/IP.

2.9.1 Capa de Aplicación

Es la capa que ocupan los programas más comunes para comunicarse por la red. Los mensajes de éstos van encapsulados en algún protocolo de la capa de transporte.

2.9.2 Capa de Transporte

Esta capa se encarga de conectar aplicaciones entre sí a través de puertos mediante un protocolo determinado por la aplicación.

2.9.2.1 TCP (Transmission Control Protocol)

Protocolo orientado a la conexión. Divide los datos en varios paquetes TCP. Resuelve numerosos problemas de fiabilidad para proveer una transmisión de bytes confiable. Se encarga que los paquetes lleguen en orden, para esto, descarta paquetes duplicados y maneja la pérdida y deterioro de éstos (cuando los paquetes no cumplen con su validación de error), siendo reenviados a su destino. A continuación se muestra la composición del paquete TCP centralizándose en los campos utilizados por los flujos IP.

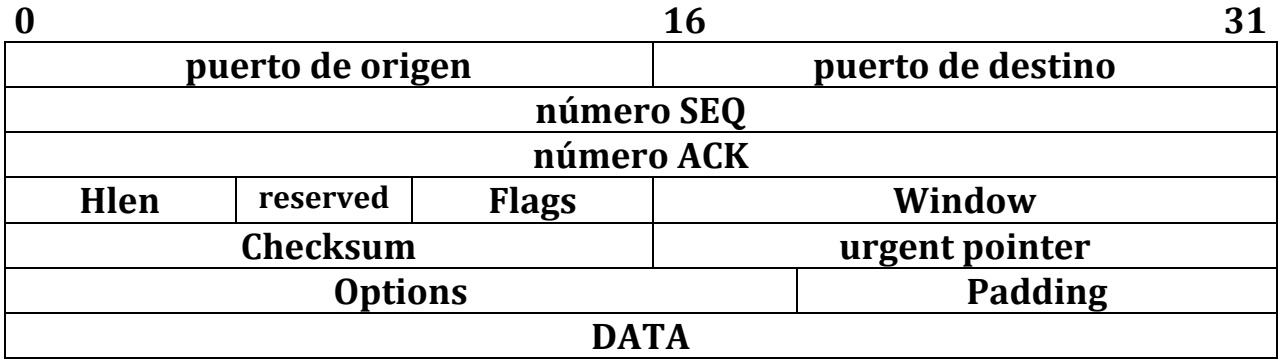

Figura 5: Paquete TCP.

Campos asociados al paquete TCP

- **Puerto de origen**: puerto de origen.
- **Puerto de destino**: puerto de destino.
- **Número SEQ** : número de secuencia (necesario para volver a armar los datos).
- **Número ACK**: número de ACK.
- **flags**: 6 bits:
    - URG: indica que el campo Urgent Ptr tiene información relevante.
    - ACK: indica que el campo "ACK Number" tiene un número de secuencia significativo. Este campo entrega el conocimiento que la máquina IP que origino el paquete recibió de forma correcta el paquete con el número de ACK -1.

- PSH: es la forma en que se indica que no se debiera seguir juntando bytes para pasárselos juntos al proceso.

- RST: se envía siempre cuando se recibe un paquete que no parece estar destinado a la conexión a la cual llegó. Sirve para volver a sincronizar ambos participantes.

- SYN: sincronización de números de secuencia. Usado al iniciar una conexión.
- FIN: usado para finalizar una conexión.

2.9.2.2 UDP (User Datagram Protocol):

Permite el envío de datagramas de forma casi directa, no existe orden de llegada, ni se garantiza que puedan llegar. Gracias a estas características es un protocolo más rápido y eficiente para tareas ligeras y sensibles en el tiempo.

0	16	31
puerto de origen	**puerto de destino**	
Largo	**Checksum**	
DATA		

Figura 6: Paquete UDP.

Campos asociados al paquete UDP

- **Puerto de origen**: Puerto de origen
- **Puerto de destino**: Puerto de destino

2.9.3 Capa de Red

Esta capa se preocupa principalmente de hacer lo necesario para lograr el envío desde el origen al destino abstrayéndose del tipo de conexión, para esto se vuelve a encapsular el paquete superior en un datagrama IP (o paquete IP).

Para lograrlo, realiza 3 funciones, la primera es marcar los paquetes con la identificación de la máquina originaria y de destino (se le agrega las direcciones IP de las máquinas correspondientes). La segunda función es asegurar la consistencia del paquete, debido al posible deterioro del paquete luego de su viaje por la red. Para esto utiliza un CRC (cyclic redundancy check) el cual es un algoritmo de verificación en base al contenido del mensaje. Finalmente, la capa de red proporciona mecanismos de control basados en mensajes (paquetes). Estos contienen instrucciones que manejan determinadas funcionalidades de la red.

2.9.3.1 Datagrama IP

Es la unidad base de transferencia utilizado por la capa de red para abstraerse del tipo de conexión.

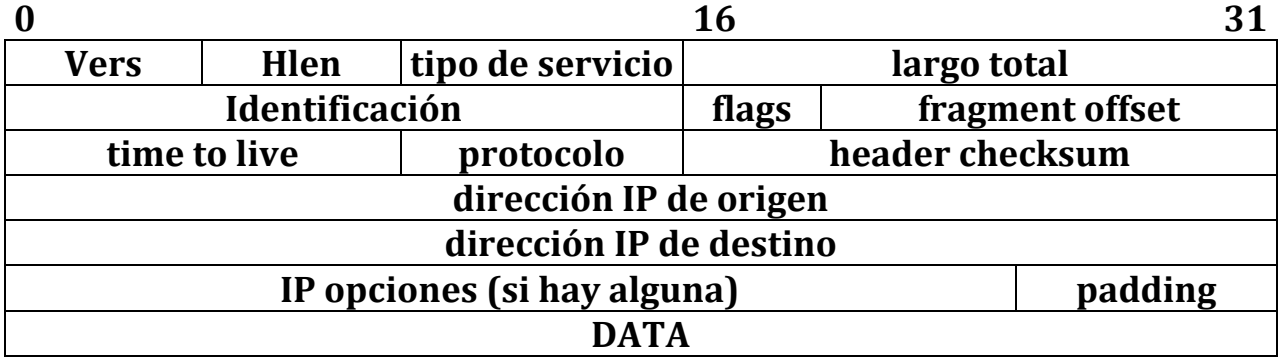

Figura 7: Datagrama IP.

Campos asociados al datagrama IP

- **Tipo de servicio (TOS)**: primeros 3 bits son la precedencia, luego vienen bits denominados D (low delay), T (high throughput), R (high reliability) y los últimos 2 no se usan.
- **Protocolo**: identifica el protocolo al cual se debe entregar los datos de este datagrama (ej: TCP, UDP, ICMP).
- **Dirección IP de origen**: direcciones en 32 bits consecutivos.
- **Dirección IP de destino**: direcciones en 32 bits consecutivos.

2.9.3.2 ICMP (internet control messaging protocol)

ICMP es un sub-protocolo del protocolo IP. Este es utilizado para detectar y avisar errores. También permite la depuración de la red en caso de problemas.

Los mensajes ICMP son construidos en la capa IP, usualmente por un datagrama IP que ha generado una respuesta ICMP. El protocolo de internet encapsula el mensaje ICMP con un nuevo header IP y lo transmite de la manera usual.

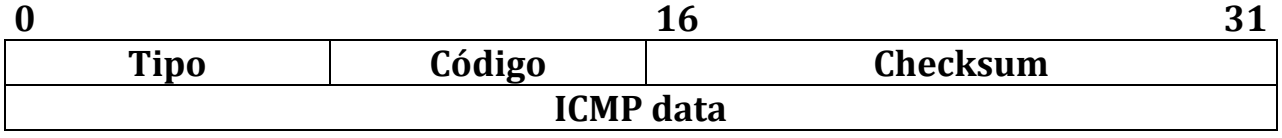

Figura 8: Header paquete ICMP.

Campos asociados al paquete ICMP

- **Tipo**: tipo de ICMP.
- **Código**: código del tipo de ICMP.
- **Checksum**: checksum calculado sobre la cabecera del paquete ICMP junto a los datos.
- **ICMP data:** este campo varía dependiendo del tipo de mensaje ICMP.

2.9.4 Capa Física

Esta capa no es realmente parte del conjunto de protocolos TCP/IP, contempla las características de la comunicación como la naturaleza del medio, potencias de señales y longitud de ondas, entre otras.

3 Desarrollo

Este capítulo está dedicado al diseño de la aplicación para el análisis del tráfico IP utilizando la información de flujos IP en redes distribuidas.

Inicialmente se plantearon requisitos funcionales que el sistema debe satisfacer para cumplir los objetivos propuestos. Para simplificar la implementación del sistema se evaluaron estos requisitos en dos herramientas de código abierto. Las herramientas evaluadas son utilizadas para la recepción, almacenamiento y análisis de la información de tráfico IP emitida por sensores. Estos sensores toman la información de tráfico IP y lo transforman a flujos IP utilizando el formato Netflow.

Seguidamente se diseñó el sistema utilizando una de las herramientas evaluadas (Nfdump) para el manejo de los flujos IP. El diseño creado requirió utilizar otros componentes para satisfacer los requisitos planteados. En especial se utilizó MySQL como motor de consultas.

Posteriormente se describió la implementación realizada del sistema por medio de las interfaces propias de cada módulo.

Finalmente, utilizando el sistema implementado, se estimó las capacidades del sistema con respecto al espacio utilizado en disco por la información de flujo almacenada, el tiempo de procesamiento para consultas hechas a los flujos y el tiempo de inserción de los flujos al sistema.

3.1 Requisitos Funcionales

De acuerdo a los objetivos planteados se investigaron herramientas comerciales existentes utilizadas para el análisis del tráfico IP en base a flujos IP. Como resultado de la investigación se llegó a la estructura utilizada por sistemas comerciales y de código abierto, mostrada en la figura 9.

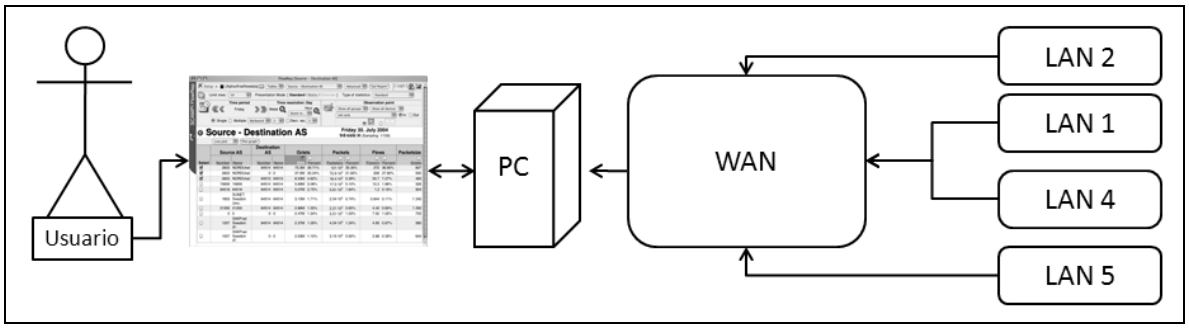

Figura 9: Estructura sistema de monitoreo.

En la figura 9 se muestran 4 redes locales (LAN) conectadas a internet (world area network, WAN). Las redes, por medio de sensores exportadores de flujos IP, envían la información del tráfico de red propio hacia una máquina colectora. Esta máquina almacena los flujos IP entrantes. La información del tráfico IP puede ser analizada utilizando herramientas especializadas.

De acuerdo a la estructura planteada y junto a los objetivos de esta memoria se infieren los siguientes requisitos funcionales.

3.1.1 Recibir y almacenar los flujos IP provenientes de sensores en redes distribuidas.

La información de flujo proviene de sensores especializados, estos suelen ser dispositivos de red (routers, switches etc.) o computadores actuando como sensor. El envío de la información de flujo es posible mediante configuraciones propias de los sensores. Para esto es necesario configurar la dirección IP y puerto de destino del flujo.

El sistema debe recibir y almacenar los flujos provenientes de sensores distribuidos.

3.1.2 Capacidad para filtrar la información almacenada para el posterior análisis.

La información de flujo proveniente de los sensores puede llegar a utilizar varios Gigabytes de espacio en disco. Mientras mayor sea la cantidad de información, mayor será el tiempo de cálculo utilizado en consultas sobre el tráfico IP. Dependiendo de la cantidad de información y consulta utilizada, estos cálculos pueden tomar hasta días de procesamiento.

Una forma de disminuir el tiempo de procesamiento es reduciendo la cantidad de información analizada. Por ejemplo, si se busca información sobre una máquina en el tráfico generado en 5 meses anteriores, es útil filtrar la información de tráfico seleccionando la información que hace referencia a la máquina. Esto reduce significativamente la cantidad de información procesada.

3.1.3 Administración de procesos para la recepción y el almacenamiento de los flujos IP

El sistema debe poder analizar la información proveniente de redes en distintas zonas geográficas. Para esto es necesario poder identificar el origen de la información en los datos almacenados, por lo cual se debe contar con algún mecanismo para administrar los procesos que recepcionan y almacenan los flujos de los sensores asignados a las redes.

3.1.4 Realizar consultas complejas a la información de flujo almacenada

Un requerimiento claro es poder realizar toda clase de preguntas a la información almacenada, distinguiendo entre redes y sensores distintos.

Las consultas deben obtener información sobre redes completas, grupos de sensores o máquinas existentes en la red.

El manejo de esta información debe ser lo más libre posible. Por ejemplo, hacer consultas anidadas, es decir, realizar consultas sobre resultados de otras consultas, poder agrupar la información y a la vez realizar cálculos a los datos de cada grupo. Por ejemplo obtener el promedio de cada grupo, etc.

3.1.5 Facilitar la creación, almacenamiento y utilización de consultas sobre el tráfico IP proveniente de sensores distribuidos.

El sistema debe proveer herramientas para facilitar el monitoreo y la experimentación en busca de nuevos métodos de detección de anomalías. Por ejemplo, almacenar las consultas realizadas permitiría aplicarlas a nuevos datos, facilitando la obtención de nueva información. Luego los resultados obtenidos por las consultas podrían ser almacenados para su posterior comparación o más aun, ser parte de una nueva consulta.

3.2 Evaluación de Herramientas para Análisis del Tráfico IP

En la actualidad existen herramientas utilizadas para el análisis del tráfico IP en base a los flujos IP. En particular se utilizó el formato Netflow para los flujos IP.

Se seleccionaron dos herramientas de código abierto para ser evaluados con respecto a los requisitos del sistema planteado (SILK, y Nfdump).

Estos sistemas fueron seleccionados por su carácter de código abierto, y satisfacción de los requisitos mencionados.

- Silk: es un conjunto de herramientas bastante complejo el cual permite almacenar flujos, administrar sensores y realizar todo tipo de consultas a los flujos. Esta herramienta es utilizada a través de la línea de comando.

- Nfdump: Nfdump es utilizado para la recepción, almacenamiento y consultas a los flujos IP. También funciona a través de la línea de comando. Esta herramienta puede ser utilizada en conjunto con Nfsend. Nfsend consulta la información de flujos utilizando Nfdump a través de una interfaz gráfica más amigable. Nfsend permite al usuario utilizar solo parte de las capacidades de Nfdump.

3.2.1 Recibir y almacenar los flujos IP provenientes de sensores en redes distribuidas.

Este requerimiento es cumplido por las dos herramientas. Los flujos son recepcionados mediante procesos que monitorean puertos determinados. La información de flujo es almacenada en archivos propios de cada implementación. La lectura de los datos es permitida principalmente con sus herramientas propias.

3.2.2 Capacidad para filtrar la información almacenada para el posterior análisis.

Los dos sistemas permiten luego de hacer consultas a los datos almacenar los resultados en otros archivos. Estos pueden ser utilizados como nueva fuente de datos, disminuyendo la cantidad de información analizada por las herramientas.

- SILK-Rwfilter: pertenece a Silk, selecciona parte de la información de los flujos IP. Los resultados pueden ser procesados por otras herramientas de Silk.

- Nfdump-Nfdump: lee conjuntos de flujos IP y permite realizar consultas a los datos seleccionados.

3.2.3 Administración de procesos para la recepción y el almacenamiento de los flujos IP

Las dos herramientas permiten la recepción de múltiples flujos provenientes de sensores externos. Silk necesita ser configurado mediante un archivo propio de configuración. Nfdump utiliza Nfcapd, el cual permite crear varias instancias para la recepción de flujos IP en puertos determinados de la máquina. Luego almacena los datos en directorios configurados. Las configuraciones de los procesos anteriores se hacen en base a argumentos en la inicialización del proceso. Esta característica hace a Nfdump por sí solo más adaptable a otros usos.

  - SILK-Rwflowpack: maneja la información necesaria para la recepción del flujo desde varios dispositivos. Esta herramienta corre como proceso demonio para los dispositivos. Por ejemplo, para capturar la información de tres dispositivos de red (Router) se debiera configurar el archivo apropiado y reiniciar el proceso Rwflowpack.

  - Nfdump-Nfcapd: es un proceso demonio el cual recibe parámetros de configuración por cada dispositivo de red (Router) para la recepción de flujos. Por ejemplo, si se quisiera capturar la información de flujo de 3 sensores, se debiera ejecutar 3 veces Nfcapd con argumentos personalizados por cada sensor, y con esto finalmente quedan 3 procesos corriendo en la máquina residente de Nfdump.

3.2.4 Realizar consultas complejas a la información de flujo almacenada

Este requisito es el más crítico al momento de evaluar las herramientas, debido a la necesidad de experimentar con esta.

Para la búsqueda de nuevos métodos de detección va a ser necesario poder tener libertad casi absoluta para el manejo de la información de tráfico, junto a la facilidad para hacer consultas a la información de flujos IP.

Silk provee distintas herramientas para consultar los datos, estas para ser utilizadas necesitan crear comandos en base a pipe's de la línea de comando. Por ejemplo, Rwfilter permite filtrar los flujos dependiendo de los argumentos, Rwsort es utilizado para ordenar, Rwcut es utilizado para mostrar la información, Rwcount es utilizado para sumarizar la información (sumar los bytes o numero de paquetes), Rwgroup es utilizado para agrupar, y otros.

Silk permite hacer toda clase de preguntas por medio de sus herramientas, sin embargo la manera de hacerlas requiere de un conocimiento muy específico de la herramienta. En consecuencia, utilizar la herramienta requiere un buen estudio, focalizando al usuario en aprender a utilizar Silk en vez de experimentar para la búsqueda de nuevas métodos de detección.

Nfdump tiene una complejidad mucho menor para realizar consultas nuevas. Esta permite agrupar, mostrar, sumarizar y filtrar en base a argumentos en la ejecución del proceso. En el caso de necesitar hacer consulta sobre los resultados, Nfdump permite exportar los resultados a su formato propio, permitiendo realizar consultas sobre resultados de consultas.

3.2.5 Facilitar la creación, almacenamiento y utilización de consultas sobre el tráfico IP proveniente de sensores distribuidos.

Las dos herramientas permiten la lectura de las consultas realizadas desde archivos de texto como medio de almacenamiento, dejando al usuario encargado del manejo de las consultas.

Estas herramientas permiten almacenar los resultados en sus propios formatos. Nfdump puede utilizar esta información para realizar un nuevo filtrado. En cambio Silk además permite utilizar los resultados para compararlos con otros existentes.

3.2.6 Selección de herramienta

Tanto Silk como Nfdump permiten de algún modo cumplir con los requisitos funcionales. En consecuencia, la importancia del diseño está en facilitar el análisis del flujo IP y no en la recepción y almacenamiento de este.

Los puntos 4 y 5 de los requisitos funcionales dan a entender la necesidad de facilitar la creación y almacenamiento de las consultas realizadas.

Una de las características de los sistemas descritos es que son soluciones probadas y utilizadas actualmente, como consecuencia, utilizar herramientas de estos sistemas para controlar el flujo IP resulta ser una buena solución. Esto permite concentrarse en la creación del sistema y no así en la creación de herramientas ya existentes para el manejo de flujos IP (Netflow).

Estas herramientas pueden realizar una gran cantidad de operaciones con la limitante de ser necesaria las herramientas propias de cada sistema para complementarse. Entonces habrá que elegir una de estas para facilitar la creación del sistema.

Silk al ser una herramienta más elaborada y compleja, dificulta la extensibilidad, tanto para el manejo de sensores como el análisis de la información.

Nfdump debido a sus características para el manejo independiente de sensores y la facilidad para analizar la información resulta ser más extensible que Silk. Sin embargo el análisis del flujo IP con Nfdump no es tan completo como el de Silk. Esto se debe a que Silk contiene más herramientas que Nfdump, permitiendo un mayor análisis.

Finalmente el conjunto de herramientas Nfdump debido a su facilidad para analizar el flujo Netflow, extensibilidad y junto a la importancia de facilitar el análisis, será utilizado para complementar el sistema.

En la siguiente sección se muestran las componentes del sistema haciendo uso de Nfdump.

3.3 Diseño

En esta sección se detalla el diseño del sistema en base a la evaluación y requisitos planteados.

3.3.1 Componentes generales

El sistema se basa en tres componentes principales mostradas en la figura 10.

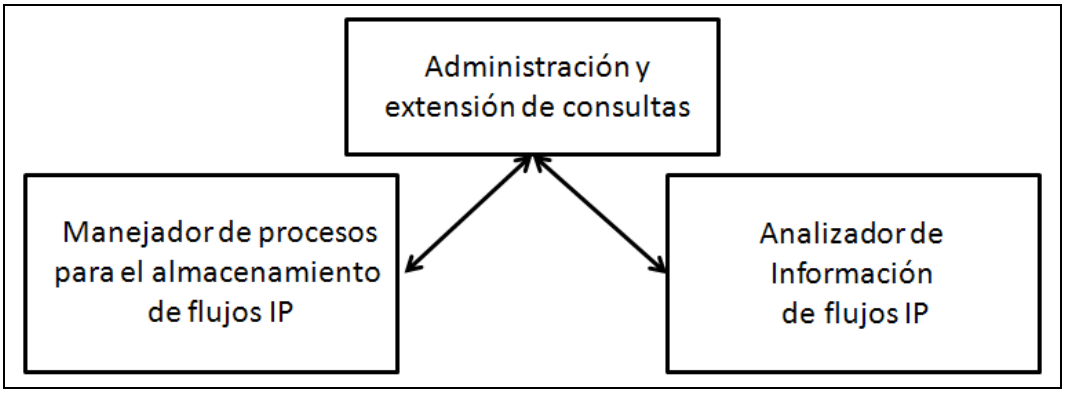

Figura 10: Componentes sistema de monitoreo.

El sistema para su funcionamiento utiliza principalmente un proceso encargado de almacenar los flujos IP, otro para leer los flujos IP y finalmente el proceso que administra los procesos anteriores y aumenta las capacidades del sistema.

3.3.1.1 Componente de manejo de procesos para el almacenamiento de flujos IP

Está encargada de manejar la recepción de los flujos de forma limpia. Esto es, recibir los flujos desde los sensores (router o PC) y almacenarlos en carpetas apropiadas.

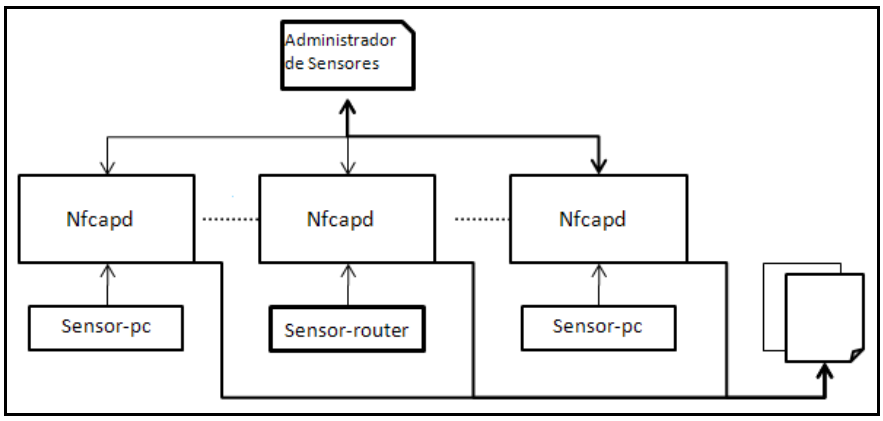

Figura 11: Detalle de manejador de procesos para el almacenamiento de flujos IP.

La figura 11 muestra un proceso llamado “Administrador de Sensores” encargado de manejar el funcionamiento de los procesos Nfcapd. Estos procesos almacenan toda la información de los flujos IP provisto por los sensores, en archivos propios de Nfdump.

3.3.1.2 Componente analizador de información de flujos IP

Este componente se encarga de tomar los flujos para realizar consultas a la información.

Este proceso se divide en dos partes: la primera parte realiza el primer filtrado de la información con comandos propios de Nfdump. La segunda parte recibe esta información y realiza un análisis más complejo sobre los datos.

Como se explicó anteriormente Nfdump no permite, de forma fácil, realizar consultas muy complejas a los datos de flujo. Por estos motivos, no se ocupará Nfdump de forma directa para analizar los flujos IP.

Al ejecutar una consulta con Nfdump sobre los datos, este permite la salida de los resultados como nuevos datos binarios utilizables por Nfdump o enviarlos a la salida estándar con un formato personalizado. Esto hace posible implementar o utilizar otra herramienta para realizar un análisis más complejo a los datos. Esto es posible debido a que al poder formatear la salida de los datos, se podrá construir un adaptador para la herramienta encargada del análisis.

Con respecto a qué herramienta utilizar para el análisis de la información de flujos IP, surgen dos soluciones de acuerdo a lo planteado:

- Solución 1: Consultar los datos mediante ejecuciones de Nfdump

Esta solución plantea basarse plenamente en la capacidad de análisis de Nfdump, y construir con él un motor para el análisis de la información. Por ejemplo para hacer consultas sobre los datos retornados por una primera ejecución de Nfdump, se podría retornar un archivo binario propio de Nfdump y luego en una nueva ejecución de Nfdump leer lo datos retornados por la primera ejecución de Nfdump, y realizar una nueva consulta.

La primera solución a la hora de implementarse, resulta ser igual de compleja para Silk o Nfdump, debido a que se deberá construir todo un motor para concatenar las distintas ejecuciones de Nfdump; más aun el usuario deberá realizar un estudio previo sobre la nueva sintaxis creada para este motor.

- Solución 2: Consultar los datos utilizando sentencias SQL en una base de datos relacional

Esta solución plantea que se podría insertar la información de flujo en una base de datos relacional. Y con esta información por medio de sentencias SQL se analizarán los datos.

Utilizando esta solución las capacidades de análisis quedarían asociadas directamente al poder de las consultas relacionales y al motor de base de datos.

En la segunda solución, debido a lo elaborado que se encuentran los motores de las bases de datos relacionales, se contará con toda la tecnología implementada para las bases de datos en cuanto a las consultas y a las capacidades de almacenamiento de los motores. Por ejemplo, se sabe que MySQL permite 65536 terabytes por tabla (un archivo). Pero la capacidad máxima posible está limitada al tipo de sistema de archivos utilizada por la máquina, se sabe que "ext 3" soporta hasta 4 terabytes como máximo por archivo y "solaris 10" hasta 16 terabytes.

Una de las primeras preguntas que surge con respecto a construir un motor para consultas de Nfdump o utilizar una base de datos, es sobre el performance al momento de consultar los datos debido al volumen de información y tiempo de procesamiento. Este es un punto importante en el diseño del sistema.

Debido a que el sistema principalmente utiliza información no en línea, se privilegió la facilidad para hacer consultas a la información, por lo cual se descartó la primera solución.

Como conclusión sobre la implementación del "Analizador de información de flujos IP", se utilizará un motor de base de datos relacional, principalmente por sus grandes capacidades de almacenamiento y estructura formal para la creación de consultas sobre los datos.

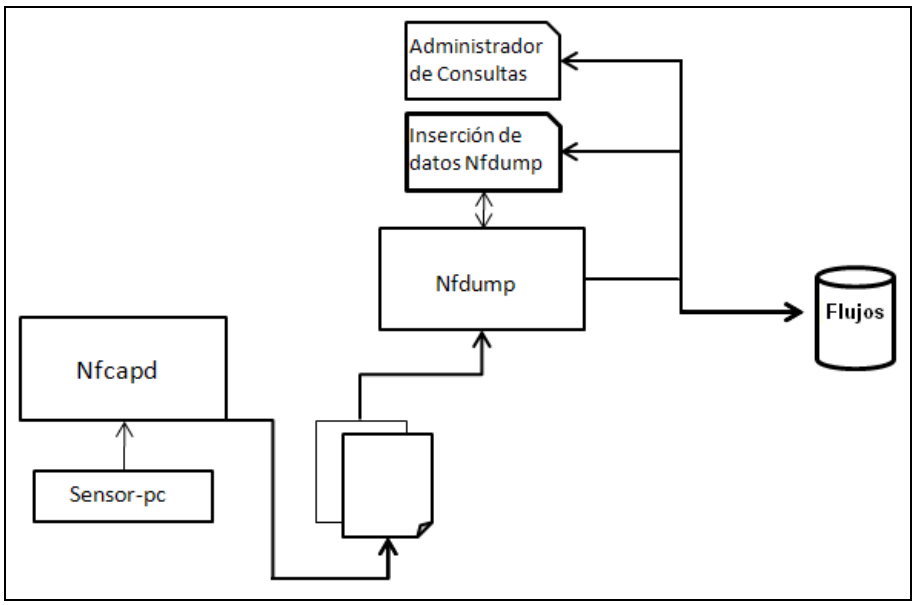

Figura 12: Detalle de analizador de información de flujos IP

La figura 12 muestra el detalle de esta componente. El módulo de "Inserción de datos Nfdump" maneja el proceso Nfdump para leer los datos almacenados por Nfcapd e insertarlos a la base de datos "Flujos". Con la información de flujos almacenada y utilizando el módulo "Administrador de Consultas" es posible analizar la información de flujos IP.

3.3.1.3 Componente de administración y extensión de consultas

Esta componente es la encargada de administrar las componentes anteriores y aportar con nuevas componentes para facilitar el análisis y experimentación para la creación de consultas sobre la información de flujo. Se creó una serie de administradores capaces de cumplir las siguientes funciones:

- Almacenar consultas
- Administración de filtros para Nfdump
- Administración de tablas para almacenamiento de flujos
- Administración de tablas para almacenar listas de IP o puertos para realizar cruces con la información almacenada.
- Administración de resultados provenientes de consultas realizadas.
- Administración de usuarios del sistema
- Administración de redes

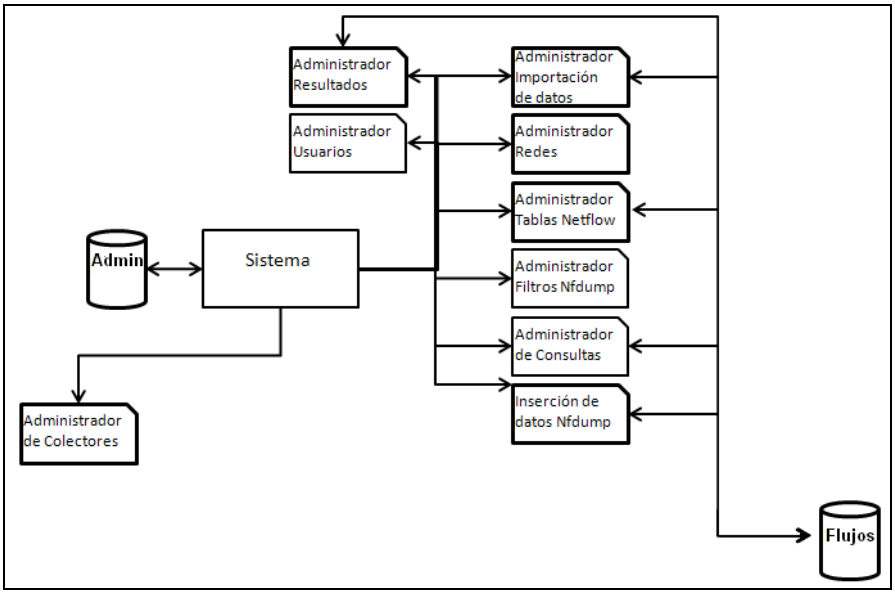

Figura 13: Detalle de componente de administración y extensión de consultas

La figura 13 muestra el detalle del módulo de "Administración y extensión de consultas". El sistema se compone de 8 módulos de administración y un módulo para almacenar en la base de datos la información de flujos de los archivos propios de Nfdump.

Estos administradores fueron implementados en un portal web utilizando PHP para el manejo de los administradores y MySQL como base de datos para la información del sistema.

3.3.2 Componentes específicos sistema Mon-dist

La relación entre las componentes descritas se detalla en la figura 14.

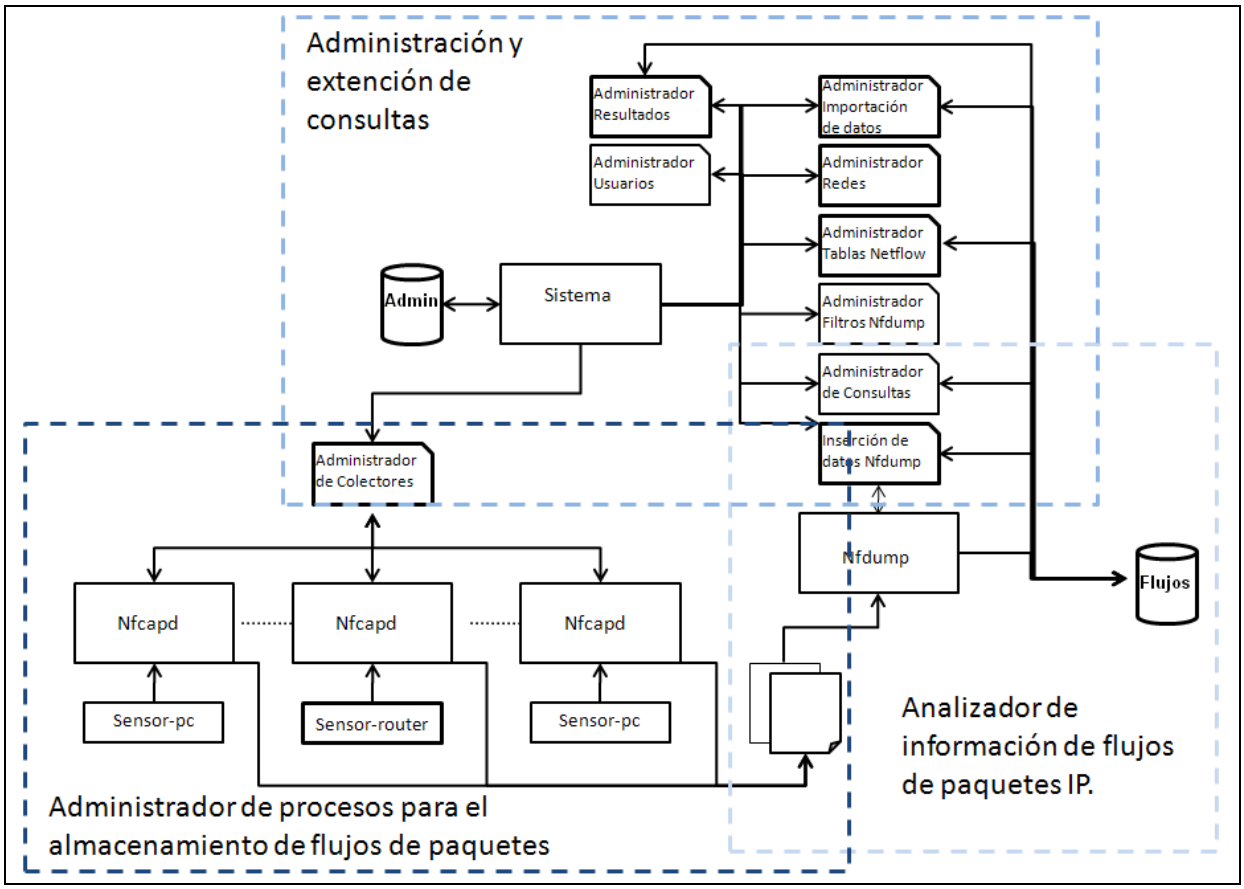

Figura 14: Estructura especifica sistema de monitoreo

El sistema está dividido en varios componentes. Los componentes llamados "administradores" permiten de forma general almacenar, modificar y eliminar la información propia de lo administrado.

3.3.2.1 Administrador de Redes

Las redes son utilizadas para agrupar un conjunto de sensores, pertenecientes a una red determinada. Existe por defecto la red llamada "General" la cual permite administrar todas las redes del sistema.

3.3.2.2 Administrador de Tablas de flujos IP

Se administran las tablas pertenecientes a la red, donde es almacenada la información importada desde los archivos binarios creados por Nfcapd.

3.3.2.3 Administrador de Importación de Datos

Se administran listas de IP o puertos, con los cuales es posible hacer cruces con la información del sistema. Las listas son creadas a partir de archivos de texto planos conteniendo la lista almacena.

3.3.2.4 Administrador de Consultas

Las consultas son creadas para redes determinadas, las consultas son sentencias SQL realizadas sobre las tablas de flujos IP, las listas importadas y los resultados de las consultas.

3.3.2.5 Administrador de Filtros

Los filtros son utilizados para filtrar la información de los archivos binarios al momento de insertarlos a las tablas de flujos IP.

3.3.2.6 Administrador de Colectores

Maneja los procesos demonios de Nfcapd permitiendo además de administrarlos, inicializar y detener los procesos.

3.3.2.7 Administrador de resultados

Este módulo administra las tablas contenedoras de los resultados de las consultas. Estas tablas pueden ser utilizadas para ver consultas anteriores, graficarlas y utilizarlas en nuevas consultas.

3.3.2.8 Administrador de Usuarios

Administra los usuarios del sistema. Los usuarios pertenecientes a la red llamada "General" tienen los permisos para administrar la información de todas las redes del sistema, y más aun, las consultas creadas para la red "General" son visibles en las otras redes.

Este módulo no es un requisito directo del sistema, pero fue implementado para facilitar una posible extensión del sistema. En especial, implementar el manejo de perfiles de usuarios. Por ejemplo, algunos usuarios podrían crear consultas y otros solo podrían visualizarlas.

3.3.2.9 Importador de datos binarios

Este módulo permite insertar los flujos IP desde los archivos binarios creados por Nfcapd a las tablas utilizadas para almacenar los flujos IP en la base de datos. Permitiendo distintos filtros a los datos antes de la inserción a la base de datos relacional.

3.3.2.10 Base de datos 1 (Admin)

Contiene las tablas para el almacenamiento de la información propia del sistema. Por ejemplo la información correspondiente a los usuarios del sistema, a las redes creadas, a los filtros creados, las tablas que almacenan los resultados guardados, las consultas

creadas, los receptores de los flujos, las tablas creadas en la base de datos “Admin” para el almacenamiento de los flujos IP y las tablas creadas en “Flujos” con las listas importadas.

3.3.2.11 Base de datos 2 (Flujos)

Contiene las tablas administradas por el administrador de tablas de flujos IP, donde es almacenada: la información de flujos IP proveniente de la importación de datos binarios. También contiene los resultados almacenados de las consultas realizadas y los datos importados, por ejemplo, listas de IP.

En la siguiente sección se muestra en detalle la implementación del sistema detallando la interfaz gráfica por cada componente.

3.4 Implementación Sistema Mon-dist

En esta sección se muestran las interfaces gráficas creada para cada componente del sistema implementado, se detalla las características principales.

3.4.1 Interfaz Inicial

La interfaz mostrada en la figura 15 es utilizada para el ingreso de usuarios registrados al sistema.

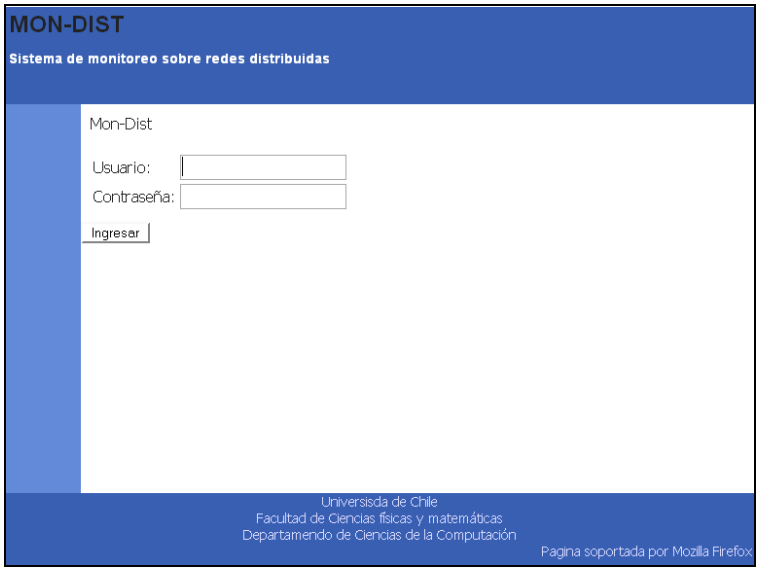

Figura 15: Interfaz inicial.

3.4.2 Administración de Redes

La interfaz mostrada en la figura 16 permite la creación, edición y eliminación de redes. A través de esta interfaz es posible utilizar otros módulos del sistema, permitiendo la administración de: sensores, consultas, tablas de flujos IP, tablas de importación de datos, filtros, usuarios, resultados y la inserción de flujos IP.

El sistema contempla una red ingresada por defecto, esta es denominada “General” y permite a los usuarios pertenecientes a la red hacer análisis utilizando la información proveniente de todos los sensores del sistema. Las consultas creadas por los usuarios son visibles para las demás redes y finalmente es posible hacer modificaciones internas a las otras redes.

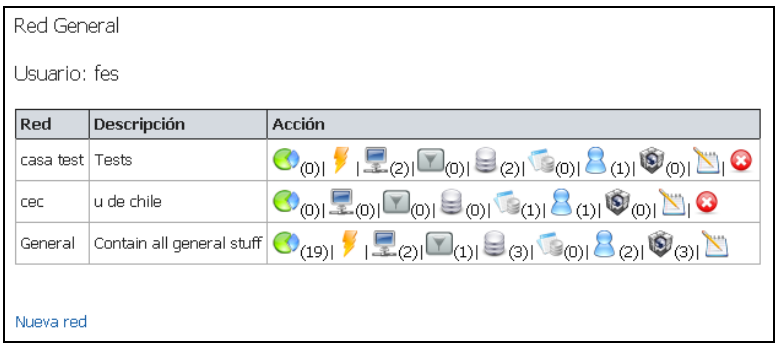

Figura 16: Interfaz administradora para red "General".

Solo los usuarios pertenecientes a la red "General" tienen la capacidad de ver y administrar todas las otras redes, en caso de no ser usuario de la red "General", la interfaz cambia a la mostrada en la figura 17.

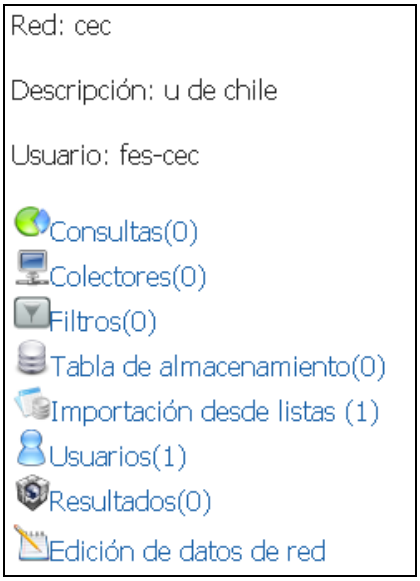

Figura 17: Interfaz administradora para red particular.

3.4.3 Administración de Usuarios

La interfaz mostrada en la figura 18 permite la creación, edición, eliminación y cambio de clave de los usuarios.

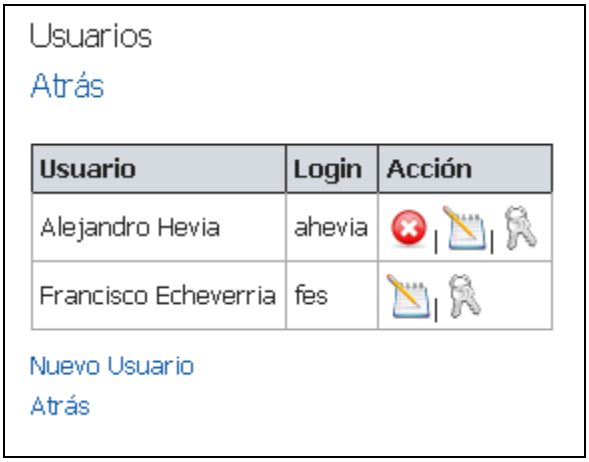

Figura 18: Interfaz administradora de usuarios.

Esta interfaz no es requisito del sistema pero se creó para facilitar una futura extensión, como soportar diferentes perfiles, por ejemplo, un usuario pueda solo ver resultados de estadísticas, y no así crear nuevas.

3.4.4 Colectores de flujos IP

Los sensores de tráfico IP envían su información de flujo a direcciones IP y puertos determinados. Un colector es un proceso Nfcapd recibiendo los flujos IP provenientes de un sensor y almacenándolos en archivos binarios.

Para mantener varias instancias de este proceso corriendo y recibiendo flujos se creó otro proceso que maneja las instancias de Nfcapd. El proceso cada 20 segundos verifica si el estado del receptor en el sistema concuerda con el estado de la ejecución del proceso. Al verificar se pueden dar tres casos:

- Estado en sistema es “Receptor inicializado” y el proceso no se está ejecutando: se lanza el proceso.
- Estado en sistema es “Receptor detenido” y el proceso se está ejecutando: se detiene el proceso.

Si el estado del sistema concuerda con el estado del proceso: no se toma acción.

3.4.4.1 Administrador de Colectores

La interfaz mostrada en la figura 19 permite la creación, modificación y eliminación de colectores. Además la inicialización y detención del colector.

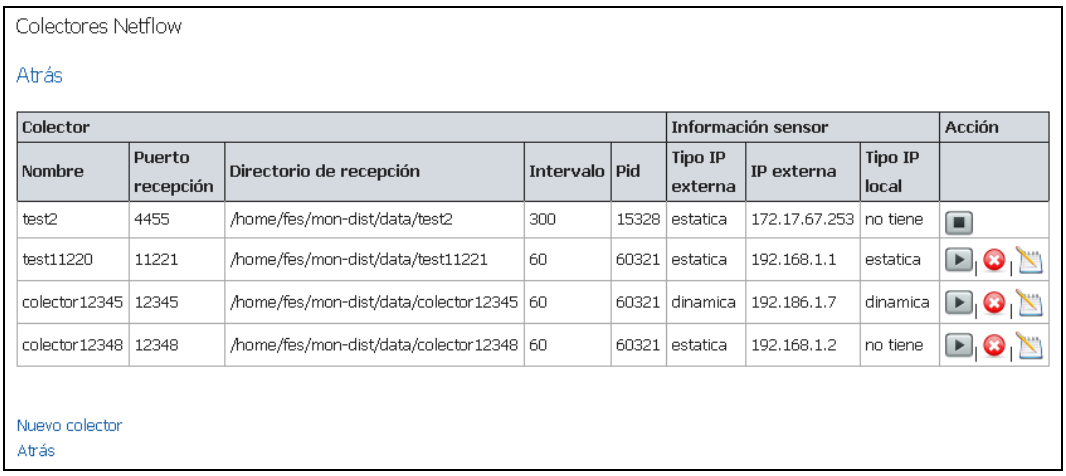

Figura 19: Interfaz administradora de routers.

3.4.4.2 Creación y Modificación de Colectores

La interfaz mostrada en la figura 20 es utilizada para la creación y modificar de colectores.

Información relevante:

- Puerto: puerto donde el colector recibe los flujos IP.
- Dir: directorio donde se alojara la información recibida por el colector, el directorio está restringido a los directorios después de /home/fes/mon-dist/data/, este es un path relativo configurado en un archivo de propiedades.
- Intervalo: los datos almacenados del flujo IP son separados en varios archivos por intervalo de tiempo. Por ejemplo, si se tiene un intervalo de 5 minutos y llevo recolectado tráfico por 1 hora, significa que tengo 12 archivos con el tráfico almacenado.
- Modo de obtención de IP: este campo es utilizado solo para conocer si la asignación de IP del sensor es de tipo estática o dinámica dependiendo del router que asigna la dirección. Esta información es útil para saber si la IP del sensor es mantenida en el tiempo y es posible identificar tráfico propio del sensor.
- IP externa: la IP externa es una referencia de la dirección IP de la red, es la IP asignada por el ISP a la red, no es la utilizada por el sensor. Esta información es útil para poder identificar flujos de una red en una red externa comunicada por Internet. La red externa solo verá la IP asignada por el ISP.
- Tipo de IP local: este campo es utilizado solo para mantener un registro del tipo de asignación de IP, bajo el sensor. Esta información es útil para saber si con los datos

almacenados en el segmento donde se encuentra el sensor, es posible identificar las máquinas en el tiempo.

- Asignación estática: el sensor es un router y las direcciones IP son asignadas particularmente.
- Asignación dinámica: el sensor es un router y las direcciones IP son asignadas dinámicamente por medio de dhcp.
- El sensor no asigna IP: el sensor es un PC normal y no asigna direcciones IP.

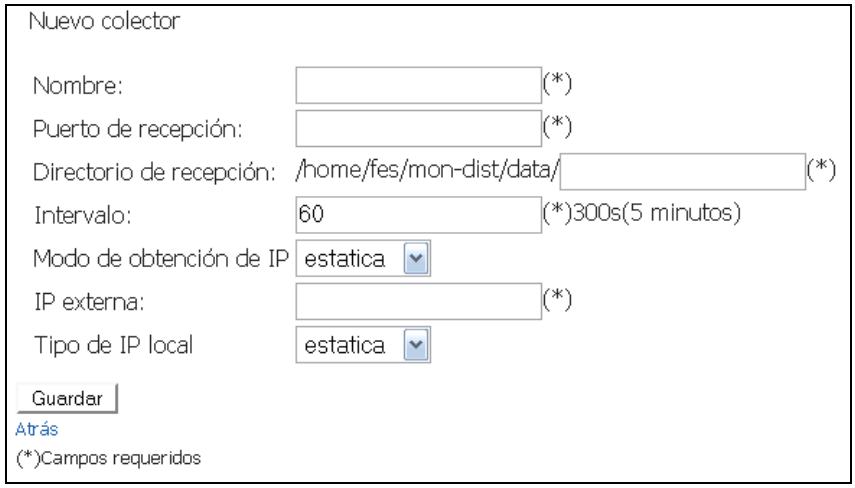

Figura 20: Interfaz de ingreso de nuevo router.

3.4.5 Administración de Tablas de flujos IP

La figura 21 muestra el administrador de tablas de flujos IP. Este permite agregar y eliminar tablas. Al momento de crear una nueva tabla esta es creada en la base de datos especializada para los datos de flujos.

El nombre de la tabla creada siempre ocupa el prefijo “packets_”.

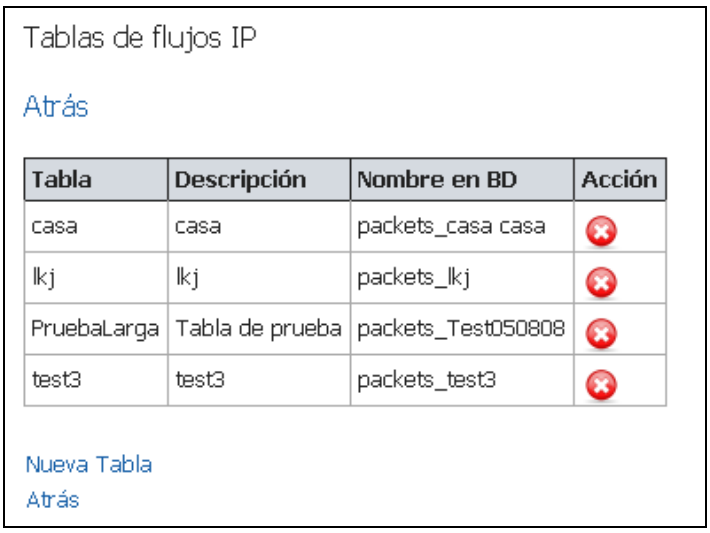

Figura 21: Interfaz administradora de tablas.

Al crear una tabla para almacenar los datos de los paquetes de flujos IP en la base de datos se crea una tabla con los campos predefinidos en la figura 22.

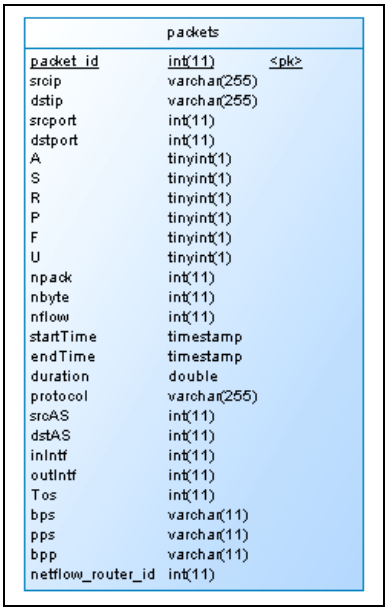

Figura 22: Estructura tabla con flujos IP.

Esto es así debido a la necesidad de manejar un formato en común para la inserción de la información de flujos IP utilizando Nfdump.

3.4.6 Consultas

Las consultas son sentencias SQL utilizadas para analizar la información proveniente de las Tablas de flujos IP. Adicionalmente es posible hacer cruces con la información de los sensores, tablas de importación de datos (Listas) y resultados generados por otras consultas.

3.4.6.1 Administrador de consultas

La interfaz mostrada en la figura 23 permite la creación, edición, eliminación de consultas. Además visualiza y grafica las consultas en el caso de ser posible.

Consulta	Red	Descripción	Query	Acción
testGraph	General	Prueba de Grafico	select srcip,sum(npack) from packets group by srcip	
total_test3	General	total columnas tabla de test3	select count(*) from packets_test3	

Figura 23: Interfaz administradora de consultas.

Las consultas generadas por el usuario general son visibles por todos los usuarios, estas son útiles para copiar su sintaxis y ocuparla en consultas propias.

En el caso de mostrar gráficamente las consultas, estas deben cumplir dos requisitos, el primero es que la segunda columna sea un valor numérico y además que la consulta tenga como mínimo 2 columnas de resultado.

3.4.6.2 Nueva Consulta

La creación de nuevas consultas permite realizar sentencias SQL, mostrar el resultado, graficarla y facilita el ingreso de una lista de IP o puertos mediante la función IN perteneciente al lenguaje SQL. También facilita las consultas sobre los datos. Este módulo aporta con información de los nombres de las tablas que almacenan la información de flujo IP, las IP correspondientes a los sensores de la red, el nombre de las tablas que tienen listas importadas al sistema y el nombre de las tablas con los resultados de las consultas SQL.

Este interfaz facilita la inserción de listas previamente almacenadas en un archivo de texto. El parser rápido recibe como entrada un archivo de texto. El archivo por línea debe contener los datos a insertar en la consulta. Luego es convertido a una cadena de texto de la forma “IN (A, B, C, D)” donde se separa por coma cada línea del archivo de texto. La cadena de texto puede ser copiada y agregado a la consulta.

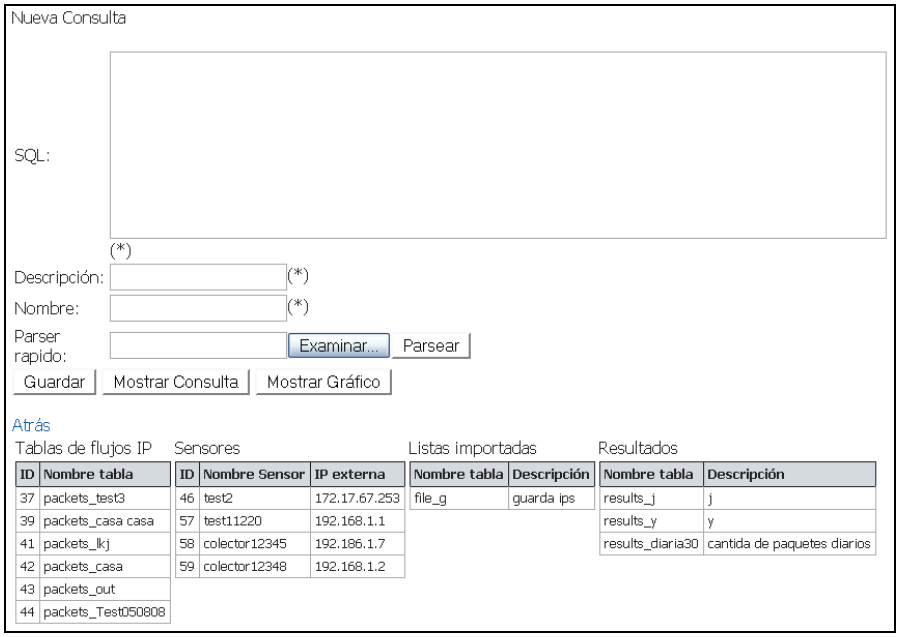

Figura 24: Interfaz de nueva consulta

3.4.6.3 Modificar Consulta

La interfaz mostrada en la figura 25 contiene las mismas características que la interfaz para agregar nuevas consultas, con la diferencia que en esta es posible la creación de una consulta a partir de una ya creada con anterioridad. Por ejemplo, si se quiere modificar cierta consulta y además la consulta anterior no se quiere perder. En este caso el sistema permite grabarla como una consulta nueva.

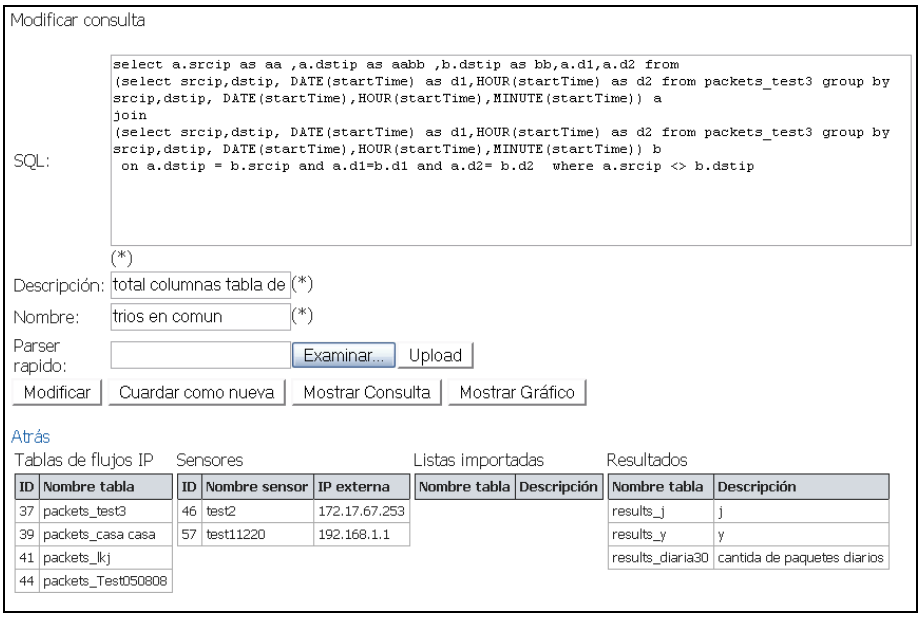

Figura 25: Interfaz de modificación de consultas.

3.4.6.4 Mostrar Consulta

Al ejecutar consultas estas se visualizan en la interfaz de la figura 26. Además de mostrar el resultado es posible almacenar este resultado sin la necesidad de volver a ejecutar la consulta. El resultado es almacenado en una tabla especial y esta puede ser usada para visualizar el resultado o ser parte de una nueva consulta.

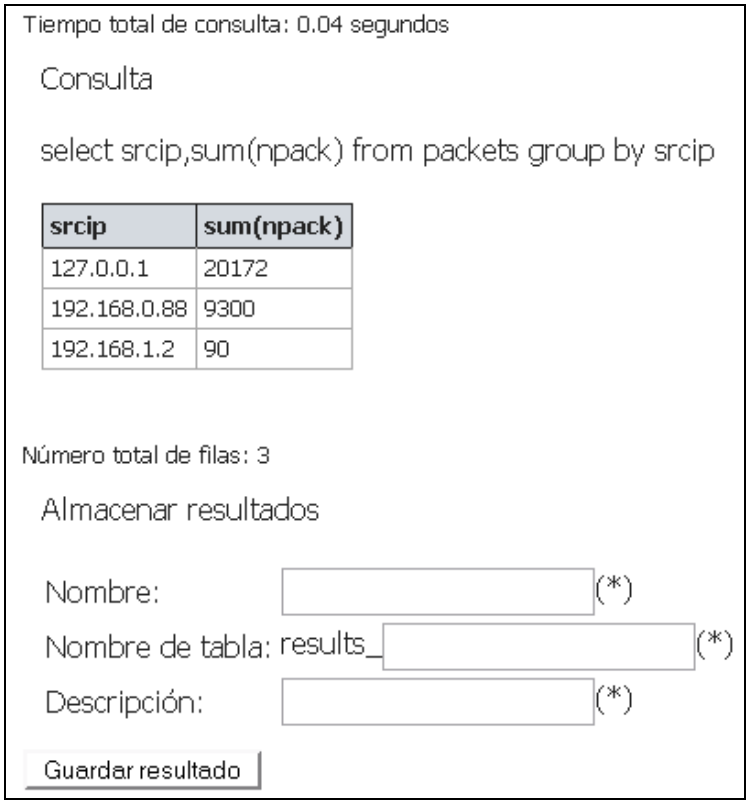

Figura 26: Interfaz de resultado de consulta.

3.4.6.5 Mostrar Gráfico

Una opción que permite los menús de creación, modificación y administración de consulta es mostrar las consultas gráficamente (Figura 27). Para esto existen 2 restricciones:

- La consulta debe tener a lo menos 2 columnas
- El valor de los elementos de la segunda columna debe ser de tipo Numérico.

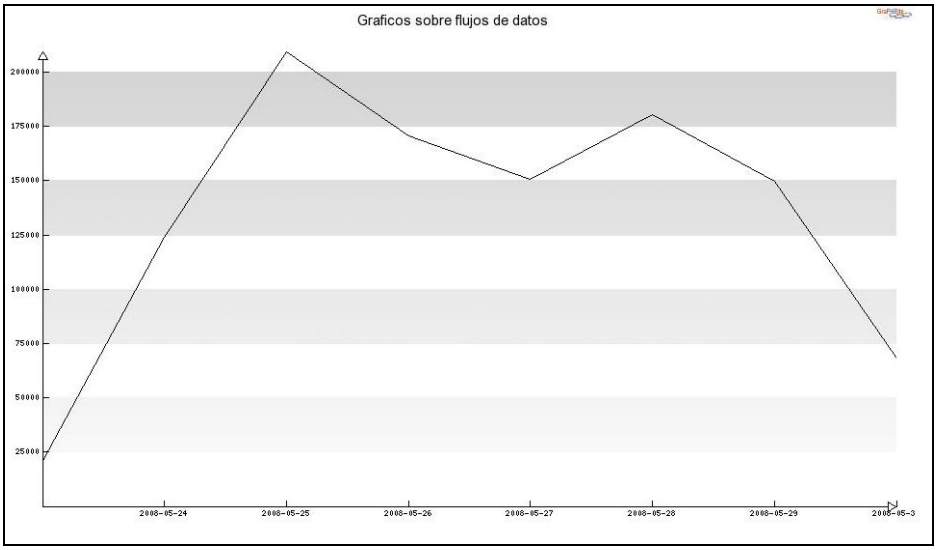

Figura 27: Interfaz de gráficos para consulta.

3.4.7 Administración de Filtros

La interfaz mostrada en la figura 28 permite la creación, modificación y eliminación de filtros.

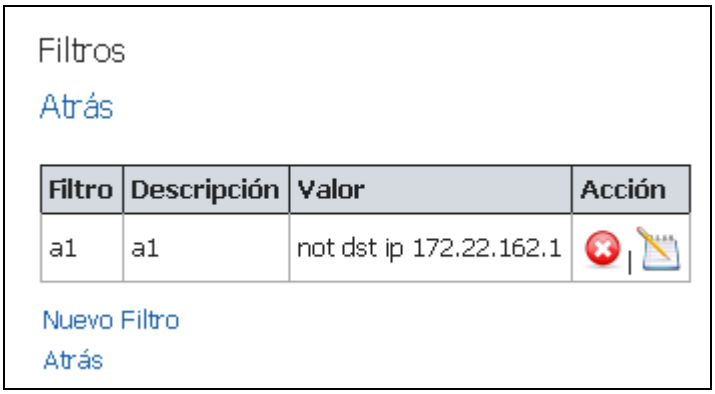

Figura 28: Interfaz administradora de filtros.

La sintaxis utilizada por los filtros es la sintaxis utilizada por Nfdump. En la interfaz para la creación y modificación de filtros, se encuentra las especificaciones de la sintaxis.

3.4.8 Administración de Tablas de Importación

La interfaz mostrada en la figura 29 permite la creación y eliminación de las tablas de importaciones, que incluyen información de listas: IP's, puertos o protocolos, subidos por el usuario desde archivos de texto.

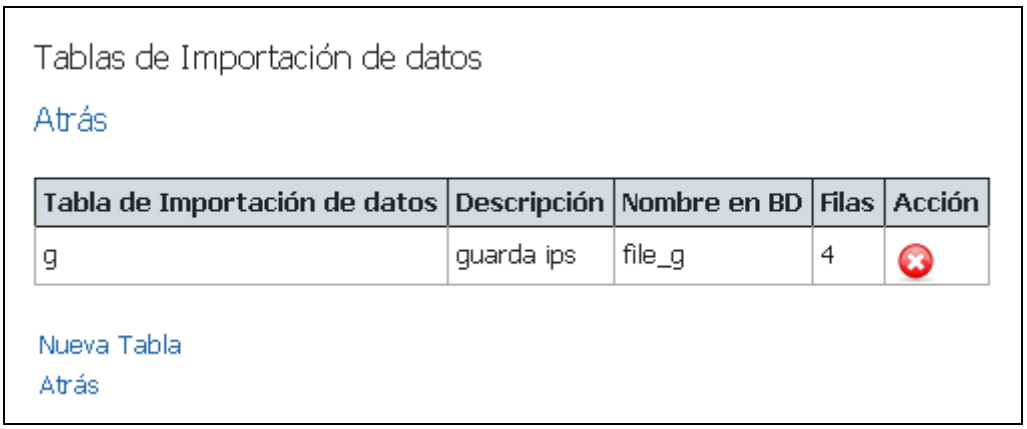

Figura 29: Interfaz administradora de tablas de importación.

La interfaz mostrada en la figura 30 permite insertar en la tabla de importación los datos descargados desde un archivo con información de una lista de IP o puertos separados por línea.

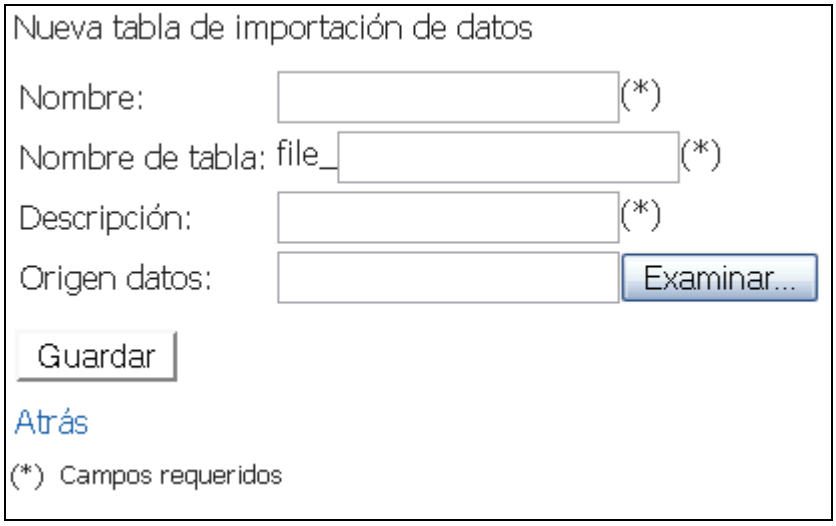

Figura 30: Interfaz de ingreso de tablas.

El nombre de la tabla en la base de datos siempre lleva el prefijo “file_”.

3.4.9 Importador de datos de flujos IP

La interfaz mostrada en la figura 31 permite la inserción de los datos binarios almacenados en las carpetas de cada colector seleccionado, en la tabla para llenar elegida.

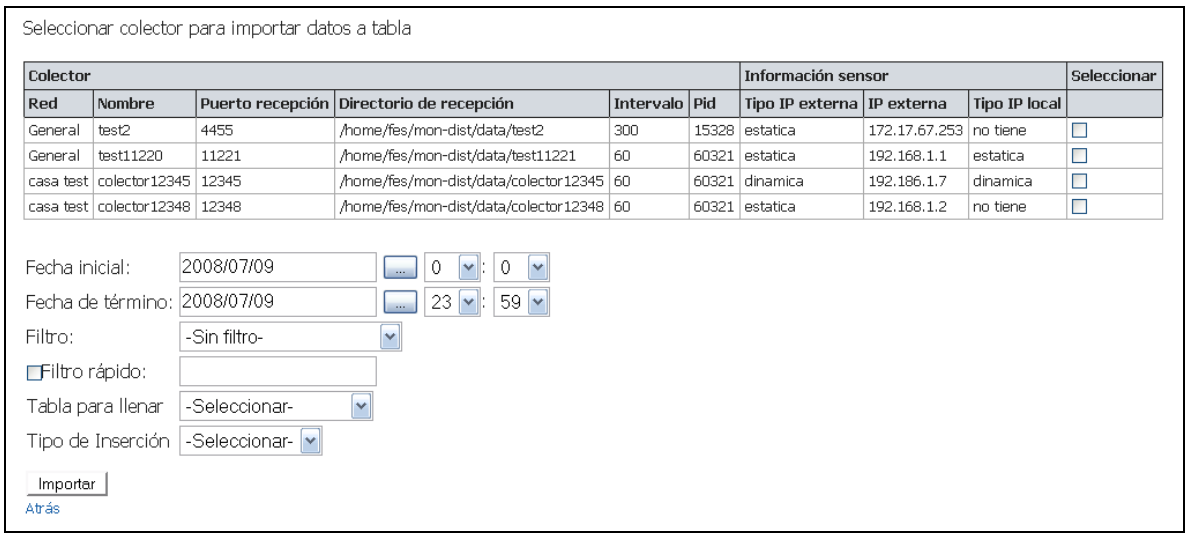

Figura 31: Interfaz para la inserción de flujos IP a tabla Netflow.

- Fecha inicial: especificar una fecha inicial para el rango de los flujos.
- Fecha de término: especificar una fecha final para el rango de los flujos.
- Filtro: lista de filtros creada por el administrador de filtros Nfdump.
- Filtro rápido: si se quisiera agregar un filtro rápido hay que seleccionar el checkbox de "Filtro rápido" y escribir el filtro.
- Tabla para llenar: se selecciona la tabla donde se almacenara los flujos IP.
- Tipo de inserción: el tipo de inserción permite importar los flujos a continuación de los registros en la tabla seleccionada para llenar, o borrar los registros antes de insertarlos en la tabla para llenar.

3.4.10 Administración de Resultados

La interfaz mostrada en la figura 32 permite la visualización de los resultados como tabla y como gráfico en caso de ser posible. Además se permite la eliminación del resultado.

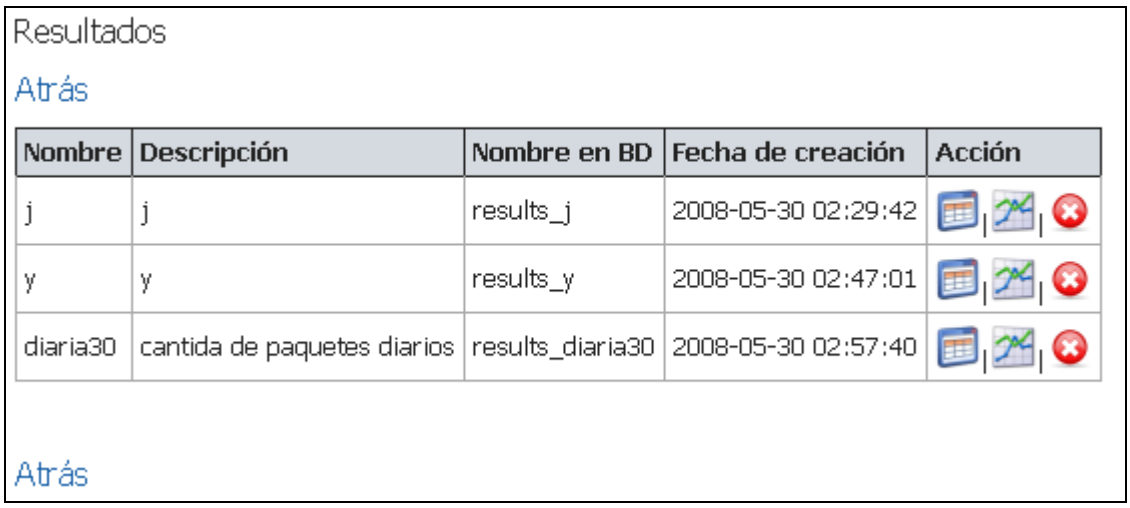

Figura 32: Interfaz de administración de resultados.

3.4.11 Base de Datos 1

La figura 33 muestra el diagrama de la base de datos utilizada por el portal web.

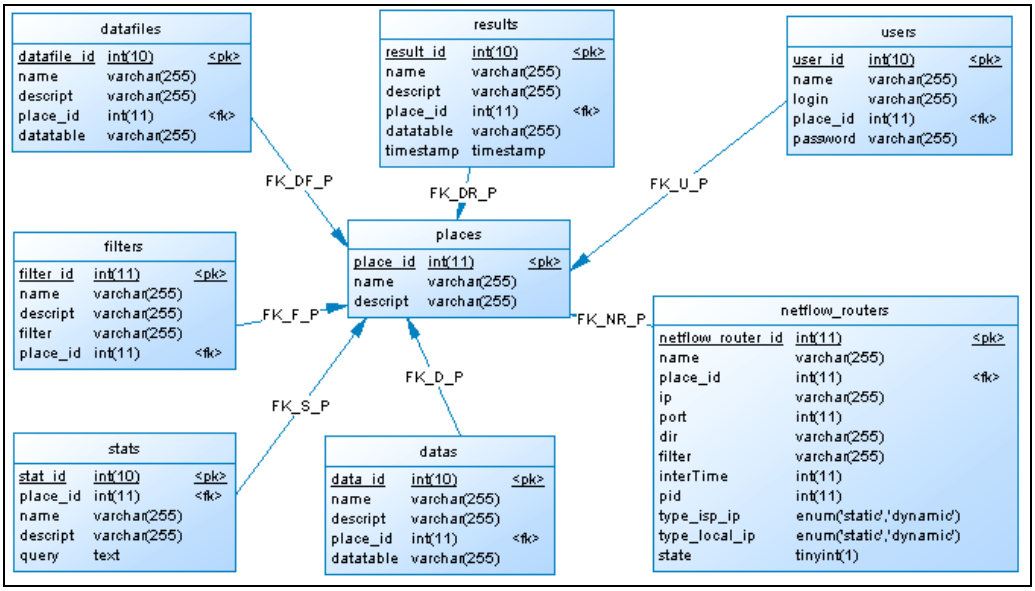

Figura 33: Base de datos Sistema.

3.4.11.1 Nomenclatura Básica

- **<pk>:** llave primaria
- **<fk>:** llave foránea
- **FK_DF_P:** llave foránea de tabla “datafiles” hacia “places”.
- **FK_F_P:** llave foránea de tabla “filters” hacia “places”.
- **FK_S_P:** llave foránea de tabla “stats” hacia “places”.
- **FK_D_P:** llave foránea de tabla “datas” hacia “places”.
- **FK_NR_P:** llave foránea de tabla “netflow_router" hacia “places”.

- **FK_DR_P:** llave foránea de tabla “result” hacia “places”.
- **FK_U_P:** llave foránea de tabla “users” hacia “places”.

3.4.11.2 Especificación de tablas

- Datas: Contiene la información de las tablas utilizadas para almacenar los flujos de tráfico IP, esto es nombre de tabla y red propietaria.

- Datafiles: Almacena la información de las tablas de importación de datos existentes para guardar la información de listas por ejemplo de IP, puertos o protocolos.

- Places: Almacena la información correspondiente a las redes. Por omisión, existe el grupo general el cual no es posible su eliminación.

- Netflow_routers: Guarda la información de los colectores del sistema, contiene la información necesaria para la identificación de los sensores, y para la inicialización y término del proceso de recepción del flujo IP.

- Filters: Contiene la información de los filtros Nfdump.

- Stats: Contiene la las consultas (sentencias SQL) utilizadas para el análisis del flujo IP.

- Results: Contiene las tablas que almacenan los resultados de las consultas calculadas.

- Users: Contiene el registro de los usuarios del sistema.

3.4.12 Base de datos 2

Esta Base de datos contiene las tablas utilizadas para hacer los cruces con la información de los flujos IP.

- Tablas de almacenamiento de flujos IP

- Tablas con los resultados de las consultas.

- Tablas con las importaciones de datos.

- Tablas con información de los sensores.

3.5 Lenguaje y Medio Ambiente

El sistema funciona en ambiente Linux por ser un sistema operativo de código abierto, utiliza apache como servidor web y MySQL como base de datos. El sistema Web está desarrollado ocupando HTML, PHP y Javascript.

La ejecución de procesos como Nfcapd, Nfdump y otros son lanzados desde programas escritos en Perl. En algunos casos estos son invocados desde PHP. El programa administrador de colectores, corre como proceso demonio.

3.6 Estimaciones de la Capacidad de procesamiento y almacenamiento del sistema

A fin de entender las limitaciones del sistema, se realizaron varios experimentos para estimar la capacidad de procesamiento y almacenamiento del sistema.

Las pruebas realizadas y los valores obtenidos son solo estimaciones debido a que las topologías de red y los usuarios que generan tráfico IP son diferentes para cada red. Adicionalmente las consultas realizadas a los datos pueden ser muy variadas en tiempo de procesamiento y en espacio temporal en disco.

Estas estimaciones se obtuvieron con información proveniente de 2 ambientes, el primero, un ambiente real con un solo sensor de flujos IP y el segundo, un ambiente experimental con tráfico generado artificialmente recibido por 3 sensores.

Las estimaciones se dividieron en dos tipos, las de tamaño de almacenamiento de datos y la de tiempo utilizado por una consulta o inserciones de datos.

3.6.1 Descripción de los Ambientes de Prueba

Las estimaciones utilizaron el sistema en versión beta, corriendo sobre una máquina con Ubuntu 8.04, procesador core duo de 1.8 GHZ y 1GB de RAM.

- Ambiente 1: El tráfico proviene de una red interna en la facultad de ingeniería de la Universidad de Chile, para 10 días de recepción utilizando un solo sensor.

Fecha	Número de paquetes IP	Número de flujos
24/05/2008	124029	16009
25/05/2008	209825	16791
26/05/2008	170642	23092
27/05/2008	150768	23527
28/05/2008	180422	23361
29/05/2008	149723	23241
30/05/2008	199918	22090
31/05/2008	112640	14834
01/06/2008	94566	14601
02/06/2008	141566	20481
Total	**1534099**	**198027**

Tabla 3: Registro de 10 días de almacenamiento de Netflow.

- Ambiente 2: La fuente de datos consistía en 32.998 registros obtenidos por 20 minutos de constantes inyecciones de paquetes TCP aleatorios en 3 interfaces de red virtuales.

Los datos anteriores se utilizaron para simular tráfico proveniente de 1 hora (90.000 registros) y 1 mes (62.351.698 registros) de tráfico.

3.6.2 Estimación de capacidades de almacenamiento de datos

- Para ambiente 1:

La tabla 4 muestra información para un día promedio sobre cantidad de paquetes, el tamaño de los archivos binarios en disco, tamaño de memoria utilizada por base de datos (al hacer inserción con el sistema), tamaño del índice de base de datos (sólo campo de Id) y el tiempo requerido para la inserción de los datos a la base de datos.

Tipo	Valor
Numero de Paquetes	19802,7
Memoria en Disco	1,06 MB
Memoria en BD	1,64 MB
Memoria índice en BD	0,2 MB
Memoria total en BD	1,83 MB
Tiempo de inserción	8,5 segundos

Tabla 4: Información promedio para un día de almacenamiento (ambiente 1).

La tabla 5 muestra la proyección de los valores de la tabla 4 para 365 días.

Tipo	Valor
Numero de Paquetes	7227985,5
Memoria en Disco	386,9 MB
Memoria en BD	598,6 MB
Memoria índice en BD	73 MB
Memoria total en BD	667,95 MB
Tiempo de inserción	51 minutos

Tabla 5: Información para 365 días de almacenamiento (ambiente 1).

En la figura 34 se muestra la relación espacio en disco y número de redes (un sensor) similares para 1 año de almacenamiento del flujo IP.

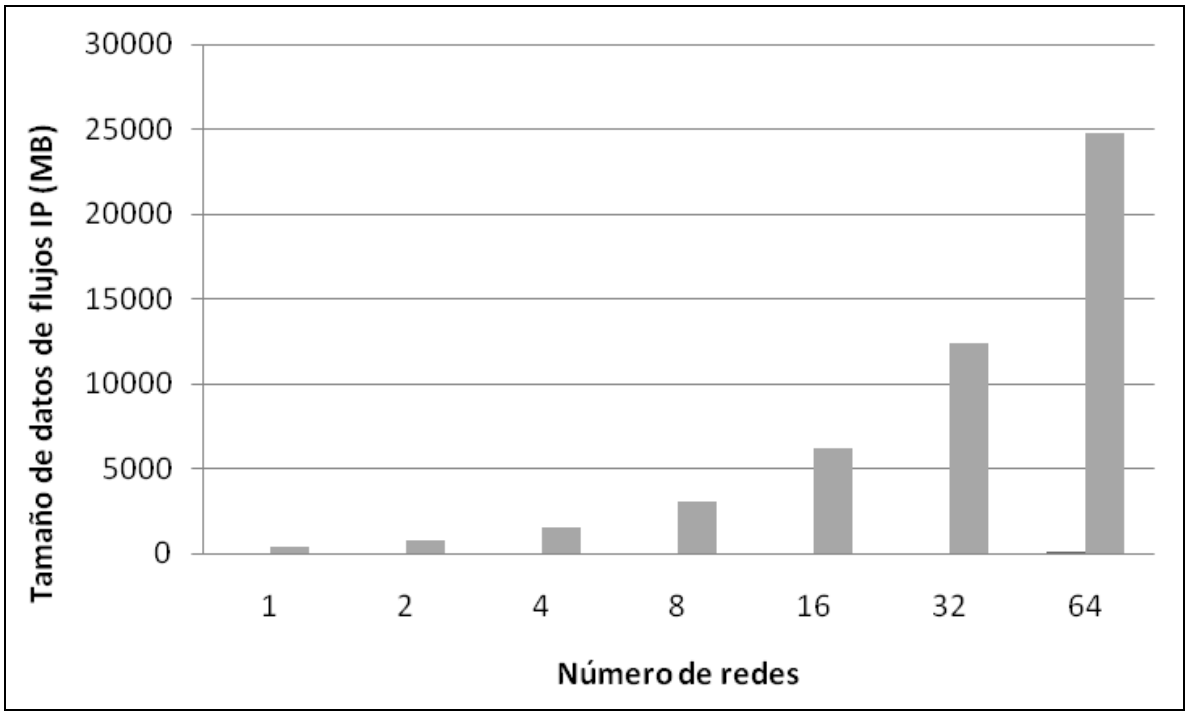

Figura 34: Número de redes v/s tamaño de datos.

- Para ambiente 2:

La tabla 6 muestra el tamaño para 62.351.698 flujos IP ingresados a una tabla en la base de datos del sistema. La base de datos contiene índices para (id, IP de origen y destino, puerto de origen y destino.

Tipo	Tamaño
Datos	4,979 GB
Índice	2,603 GB
Total	**7,583 GB**

Tabla 6: Tamaño en disco (ambiente 2).

3.6.3 Tiempo de procesamiento

Para el ambiente 1 los tiempos de inserción se detallan en las tablas 4 y 5.

Las siguientes mediciones son para el ambiente 2, calculando tiempos para la inserción de datos al sistema, consultas simples y complejas a datos en el sistema.

- Tiempo de Inserción de datos a la base de datos.

Acción	Tiempo
Inserción 90.000 registros	2 min 46.02 sec
Inserción 62.351.698 registros	1 hora 40 min 30 sec

Tabla 7: Medición Tiempo de inserción de registros.

- Tiempo para ejecutar consulta de ejemplo 1:

"select dstip,count(distinct dstport) from packets_TestAllData group by dstip having count(distinct dstport)>10"

Esta consulta agrupa los flujos por dirección IP de destino, a cada dirección de destino contabiliza el total de puertos distintos accedidos, con la restricción de que el total de puertos accedidos sea mayor a 10.

Acción	Tiempo
Consulta 90.000	3 min 51.17 sec
Consulta 62.351.698	Sin resultados (Time out)

Tabla 8: Medición de tiempo de consulta nivel medio.

La consulta para 62.351.698 registros necesitó más de 7 GB en disco duro y se paró el cálculo, por falta de espacio en disco duro, luego de 1 hora y 40 minutos.

Se estimó utilizando proporciones el tiempo necesario para obtener resultados sobre la consulta es de 44 horas y 29,23 minutos.

- Tiempo para ejecutar consulta de ejemplo 2:

"Select srcip,dstip,sum(npack) from packets_TestAllData group by srcip,dstip"

Esta consulta agrupa los flujos por dirección IP de origen y dirección IP de destino, además contabiliza el total de paquetes por cada grupo.

Acción	Tiempo
Consulta 90.000	0.705 segundos
Consulta a 62.351.698	44 min 25.11

Tabla 9: Medición de tiempo de inserción datos en base de datos.

3.6.4 Discusión

La tabla 6 muestra que 62.351.698 registros equivalentes a aproximadamente 5 GB de información de tráfico IP. Además, la información de tráfico del gráfico de la figura 34 para un año utiliza 5 GB de información en 16 redes. Al unir estos dos datos se concluye que los 62.351.698 registros equivalen a un total de 16 redes para un año. Cada una de las redes con un nivel de tráfico similar a la red del ambiente 1.

El tiempo de procesamiento para las consultas, indica que para un nivel de 16 redes similares el tiempo utilizado es razonable y es factible obtener estadísticas vía consultas a la base de datos.

4 Consultas para la detección de malware

En esta sección se describen consultas pertenecientes a las técnicas utilizadas para la detección de anomalías en flujos IP, presentadas en el capítulo de antecedentes.

Se formularon consultas puntuales para el análisis del tráfico IP sobre la detección de anomalías basadas en técnicas utilizadas por malware.

4.1 Flujos IP para creación de consultas

Los flujos IP almacenan parte de la información del paquete IP disminuyendo el espacio en disco para el almacenamiento de la información de tráfico. Esto posibilita el análisis de la información a una gran escala de tiempo y a un número mayor de redes distribuidas.

La reducción de información del tráfico IP restringe las consultas soportadas hacia los paquetes IP por la falta de campos en los header y además la información transportada por los paquetes.

4.2 Factores restrictivos sobre ambientes distribuidos

Esta sección está dedicada a mostrar los diferentes factores que hay que tener en cuenta cuando se analiza el tráfico IP en redes distribuidas.

4.2.1 Topología de red

La topología de red influye considerablemente cuando se crean consultas sobre los flujos IP. La topología de la red puede ser muy variada. Un ejemplo de esto son las dos redes mostradas en la figura 35.

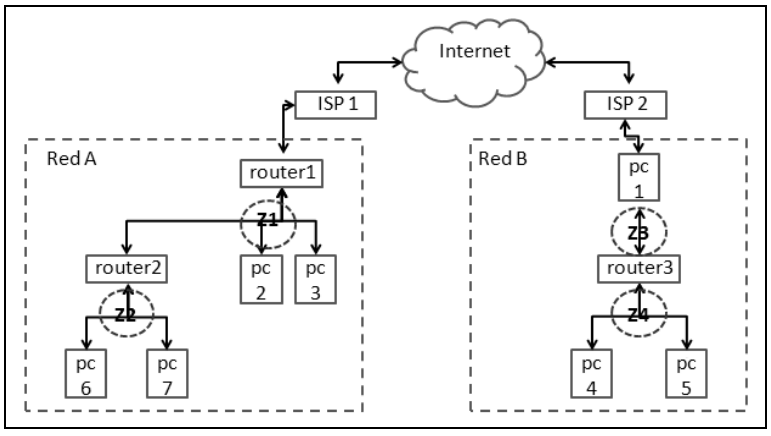

Figura 35: Topología de red.

La figura 35 muestra 2 redes con distintos proveedores de Internet. La red A obtiene su IP desde el ISP de forma estática y utiliza un router con asignación dinámica a máquina

internas por medio del NAT. Dentro de la red existen 2 máquinas y un router. El router utiliza NAT para asignar las direcciones IP internas de las máquinas 6 y 7.

En el caso de la red B el ISP asigna IP directamente al PC 1 de forma dinámica. El PC 1 reparte internet hacia el router asignándole una IP estática, luego el router asigna las direcciones IP de forma dinámica a los PC 4 y 5 utilizando NAT.

A continuación se explican 3 factores restrictivos con respecto a la topología de la red. Estos son la posición de los sensores, el número de sensores y cambios en la topología a través del tiempo.

4.2.1.1 Posición de sensores

Para la figura 35 la información obtenida por los sensores va a depender de la posición del sensor. Por ejemplo, un sensor capturando el tráfico para la zona Z1 ve el tráfico del router 2 como la IP asignada por el router 1, por lo cual, la presencia de los PC 6 y 7 es nula en la información del tráfico. En el caso de la red B al tener un sensor en la zona Z3 los PC 4 y 5 tampoco aparecen en la información del tráfico IP.

4.2.1.2 Número de sensores dentro de una red

Tener un número mayor de sensores dentro de una red local, ayuda a identificar las máquinas internas. Por ejemplo, utilizando la topología de la red de la figura 35, si se tuviera un sensor en la zona Z1 y además un sensor en la zona Z2, es posible co-relacionar la información entre las zonas para identificar los tráficos de la zona Z1.

4.2.1.3 Cambios de la topología a través del tiempo.

La topología de la red generalmente afecta al tráfico IP. Ejemplo de cambios son el ingreso de nuevas máquinas, cambios en tarjetas de red, cambios de ISP, etc. Consultas en el tiempo tienen que considerar esta información a la hora de realizar comparaciones y uso de datos en las consultas. Por ejemplo, una consulta entrega las máquinas que generan más tráfico para todo un año. Existe una máquina infectada la cual aparece en la lista de máquinas con más tráfico. Si la máquina cambia la tarjeta de red, la dirección MAC cambiará. Por otro lado, si el router a cargo del segmento asigna las direcciones IP dinámicamente, entonces la dirección IP de la máquina también cambiará en el tiempo. Como consecuencia se perderá la referencia de la máquina y la información de la consulta podría no mostrar la máquina comprometida.

4.2.2 Asignaciones dinámicas de IP

La asignación dinámica de IP puede ser un caso complejo en el análisis local como distribuido. Las máquinas que obtengan IP de forma dinámica, son difíciles de rastrear a menos que se cuente con alguna especie de mecanismo que haga esto posible. Por ejemplo, una máquina que realiza mucho tráfico se tiene identificada, luego de unos días la máquina se desconecta de la red. Posteriormente, al conectar la máquina recibe una nueva dirección

IP. Entonces, al buscar la máquina con la dirección IP antigua, no se podrá encontrar la máquina a la cual corresponde la dirección IP antigua.

4.2.3 NAT

Es un mecanismo utilizado por los router para hacer que máquinas internas en redes privadas puedan comunicarse con el exterior. Para esto traduce paquetes internos a externos o vice versa. En el caso donde se tengan sensores en un segmento y se utilice NAT, complica el análisis por el hecho de no saber la información de traducción utilizada por el NAT. Este problema se describe más a fondo en la sección 7.

4.2.4 Pérdida de paquetes

Los flujos IP informan la cantidad de paquetes y bytes en cada flujo, sin embargo, hay casos donde no todos los paquetes llegan al destino. Si se comparan los flujos entre dos segmentos, habrá flujos reales que no coincidirán. Esto provocará una disminución de la precisión en la identificación de flujos IP.

4.2.5 Configuración de sensor de flujos IP

Los sensores tienen distintas reglas para la creación de flujos. Al variar estas reglas entre los sensores, dificulta aun más encontrar similitudes entre los flujos. Las principales reglas utilizadas son:

- Tiempo de vida flujo: sensores por defecto tienen un tiempo de vida del flujo activo e inactivo generalmente de 15 segundos y 30 minutos respectivamente.
- Tamaño del cache: si el cache se llena los sensores cortan los flujos y los envían.
- Corte de flujo por flags FIN y RST: en conexiones realizadas por TCP, Si un paquete perteneciente a un flujo vienen los flags FIN o RST, el flujo es cortado.
- Número de flujos por datagrama: al caducar, los flujos se agrupan y se exportan en datagramas de hasta 30 records

Utilizar distintas reglas para la creación de flujos dificulta la identificación de flujos entre la información exportada por los sensores.

4.2.6 Conclusión

Los hechos mencionados son útiles para el conocimiento de las limitaciones al analizar la información de los flujos IP en redes distribuidas. Sin embargo, aunque no se

pueda identificar absolutamente un flujo, en algunos casos, el hecho de saber que un grupo de máquinas se encuentran comprometidas, ya ayuda a la detección de anomalías en la red.

En el capítulo 7 se plantea un experimento exploratorio que busca identificar, a través de la información de los flujos IP para dos redes distribuidas geográficamente, la máquina emisora de paquetes IP cuya dirección IP fue modificada por un NAT.

4.3 Top N / Baseline

Las técnicas de Top N y Baseline utilizadas en conjunto entregan información útil para la detección de anomalías sobre flujos IP.

Top N / Baseline permiten la creación de múltiples consultas de seguridad. En esta sección se formulan solo algunas de acuerdo a su contribución.

Malware por lo general hace cambiar el estado normal de la red. La búsqueda de estados fuera de lo normal da indicios de la existencia de malware.

4.3.1 Top N / Baseline para el monitoreo de una red local

Como se habló en los antecedentes, se utiliza esta técnica para datos de sesión y datos transmitidos.

- Ranking de IP de origen con un mayor número de IP's de destino diferentes.

  Si estas cantidades superan las normales se sabe que la máquina posiblemente esté infectada.

- Ranking de IP de origen con una mayor cantidad de bytes transferido por máquinas pertenecientes a la red.

  Es útil saber la tasa de transferencia normal de la red debido a que si esta es superada, es posible la existencia de algún gusano copando el ancho de banda

4.3.2 Top N para redes distribuidas

- Top N de paquetes transmitidos desde cada red hacia las otras.

  Esta información es útil para determinar las redes con más posibilidad de infectar o ser infectadas por otras del sistema.

4.4 Calce de direcciones y puertos

La manera de uso más general de esta técnica es partir con el conocimiento de alguna anomalía a buscar y por medio de sus características, buscarla en la información de las redes donde se quiera identificar.

4.4.1 Calce de direcciones y puertos para el monitoreo de la red

- Paquetes que no tengan ni IP de origen ni IP de destino, máquinas pertenecientes al mismo segmento de red del sensor.

  Se obtiene indicios de tráfico posiblemente malicioso.

- Calce de características propias de anomalías.

  Por ejemplo, botnet del tipo IRC ocupan generalmente el puerto 6667, monitorear máquinas que envía paquetes a ese puerto puede ser un indicio.

4.4.2 Calce de direcciones y puertos para redes distribuidas

- Búsqueda de IP externa de red Infectada sobre las otras redes.

  Esta información ayuda a las otras redes a encontrar un canal con el cual estuviera expuesta alguna red del sistema a la infección.

- Utilizar información proveniente de ataques ya realizados a las otras redes.

  Al tener información sobre ataques realizados a otras redes monitoreadas por el sistema ayudará a mejorar la seguridad de cada red. Por ejemplo sabiendo la forma de expansión de un ataque, es posible la creación de consultas de seguridad capaces de identificarla y posteriormente utilizar las consultas en las otras redes del sistema.

4.5 TCP flags para flujos IP

- Máquinas enviando paquetes con solo flag SYN y que no reciban respuesta.

  Esta información es útil para saber qué máquinas se intentan conectar a otras no existentes, en algunos casos esta información indica un escaneo de puertos.

- Máquinas enviando paquetes con flag ACK/RST.

  Indica que hay máquinas que intentan acceder por puertos cerrados a otras. Saber qué máquinas reciben una gran cantidad de paquetes con flags ACK/RST podría indicar las máquinas ya comprometidas.

4.5.1 TCP Flags para redes distribuidas

- Nivel de envío de paquetes entre las redes que solo hayan paquetes de ida con TCP flag SYN y no exista uno de vuelta.

  Un nivel muy alto, indica que la red que envía está escaneando, por lo cual se podría encontrar la red comprometida.

- Nivel de envíos de paquetes con solo el flag SYN y que hayan recibido paquetes de vuelta con el flag ACK/RST.

  Si el nivel es muy alto puede indicar que la primera red se encuentra comprometida y la segunda tiene posibilidades de ya estar infectada.

4.6 Consultas derivadas

Estas consultas de seguridad se refieren al uso de técnicas de forma conjunta. Por ejemplo, utilizar el calce de patrones en conjunto con top N / Baseline, produciendo estadísticas interesantes para el monitoreo y detección de malware en la red.

4.6.1 Top N y calce de dirección y puerto

- Ranking de máquinas con un número mayor de direcciones distintas de las máquinas propias de la red.

  Las máquinas que repentinamente realizan tráfico a un número bastante distinto del normal, pueden estar comprometidas.

4.6.2 Top N y TCP flag

- Ranking de máquinas con mayor número de intentos fallidos de conexión de acuerdo a TCP flag.

  Si alguna máquina tiene un número de intentos fuera de lo normal, puede estar comprometida.

- Ranking de máquinas con mayor número de intentos de conexión que no hayan recibido paquetes de respuesta.

  Las máquinas que tienen un número de intentos fuera de lo normal, indica las máquinas posiblemente comprometidas.

5 Evaluación

La evaluación se resume en dos experimentos creados para evaluar el sistema en cuanto a su eficiencia y eficacia para la seguridad informática.

- El primer experimento evalúa la capacidad del sistema para detectar anomalías. Se simularon 4 comportamientos vistos en el tráfico IP producto de malware. Por cada comportamiento se creó consultas de seguridad relacionadas a la detección del comportamiento.

- El segundo experimento evalúa la efectividad del sistema para detectar malware en un ambiente real. Se utilizó tráfico real proveniente de dos semanas de almacenamiento. Se utilizaron dos comportamientos del experimento 1 simulados artificialmente. Entre el tráfico real se agregó el tráfico simulado en una fecha determinada. Se crearon consultas enfocadas a la detección del comportamiento malicioso en el tráfico real.

5.1 Experimento 1

Este experimento principalmente evalúa la aplicación en una red artificial con tráfico IP simulado. Se simularon 4 comportamientos del tráfico IP producto de malware. La finalidad de la simulación fue probar la correctitud del sistema y la capacidad de este para la detección de diferentes malware.

Los comportamientos fueron simulados mediante inyecciones de paquetes IP en una interface de red de prueba. La interface de red fue configurada para solo aceptar los paquetes de las simulaciones. Con esto se redujo el margen de error en la experimentación. Mientras corrían las simulaciones, un sensor de flujo se encargó de crear flujos IP con la información del tráfico, para así enviarlos a un colector de flujos previamente configurado.

Mediante el sistema construido se hicieron consultas de seguridad al tráfico almacenado para detectar el comportamiento simulado.

En las siguientes secciones se detallan los comportamientos creados con sus resultados.

5.1.1 Comportamiento 1: Tráfico 1 a N

El tráfico de 1 a N corresponde al tráfico generado por una máquina hacia N máquinas en un periodo corto de tiempo. Específicamente se refiere a un grupo de paquetes IP donde la dirección de origen se ve repetida y la dirección de destino es distinta para el grupo.

Este comportamiento es visto generalmente en casos donde:

- Máquina maliciosa, o red maliciosa busca máquinas para infectar. Los gusanos generalmente hacen uso de esta técnica para esparcirse.
- Máquina o red externa respondiendo a escaneo distribuido realizado por un conjunto de máquinas.

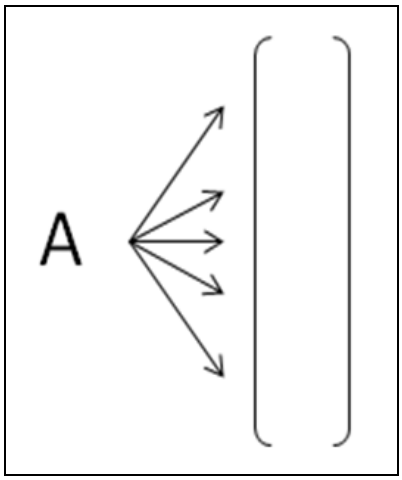

Figura 36: 1 a N.

5.1.1.1 Simulación

Se creó escenario, el cual consta de una red local con asignación de máquinas 192.168.2.2 hasta 192.168.2.30.

La máquina con IP 192.168.2.2 se encuentra infectada con un bot el cual realiza un escaneo de máquinas para reclutar nuevas máquinas a la botnet. La máquina bot específicamente realiza un escaneo a dos rangos 192.168.0.2-192.168.0.50 y 192.168.2.10-192.168.2.100. El escaneo se basa en el envió de paquetes TCP desde el puerto 2039 hacia el puerto 2040 en el rango de máquinas descrito.

Las máquinas existentes en la red responden a esta máquina de vuelta, las cuales dentro del rango 192.168.2.3-192.168.2.15 responden con el flag RST/ACK y la otra mitad con ACK/SYN.

5.1.1.2 Metodología

1) Una manera de detectar este comportamiento es monitorear los paquetes que provienen de un mismo origen. Se creó una consulta que agrupa los paquetes por IP de origen y cuenta las IP's de destino distintas.

IP de origen	Nº IP de destino distintas	Nº de paquetes
192.168.2.2	140	155
192.168.2.10	1	1
192.168.2.11	1	1
192.168.2.12	1	1
192.168.2.13	1	1
192.168.2.14	1	1
192.168.2.15	1	1
192.168.2.16	1	1
192.168.2.17	1	1
192.168.2.18	1	1
192.168.2.19	1	1
192.168.2.20	1	1

Tabla 10: Resultado 1.

Se ve claramente que 192.168.2.2 mandó a 140 equipos distintos y envió en total 155 paquetes. Lo cual corresponde a lo enviado en la simulación, 49 paquetes para rango 192.168.0.2-192.168.0.50, 91 paquetes para rango 192.168.2.10-192.168.2.100 esto entrega un total de 140 IP's distintas. Además 15 paquetes para rango 192.168.2.16-192.168.2.30 en total 155 paquetes enviados por 192.168.2.2.

El siguiente paso es identificar las máquinas posiblemente infectadas por la máquina con IP 192.168.2.2.

2) Una pregunta interesante es saber cuántas y cuales máquinas aceptaron la petición de conexión propuesta por la máquina infectada. Obtener esta información es posible para tráfico TCP. Analizando los flags TCP donde no se encuentre el flag RST, pero si los flag SYN y ACK en los paquetes IP, mostrará las máquinas que aceptaron la conexión.

La tabla 11 muestra el número de máquinas de origen distintas las cuales enviaron paquetes con los flags SYN, ACK y no RST.

IP de destino	Nº IP de origen distintas
192.168.2.2	15

Tabla 11: Resultado 2

La tabla 12 muestra el detalle de las máquinas posiblemente infectadas al responder a la conexión.

IP de origen	IP de destino
192.168.2.16	192.168.2.2
192.168.2.17	192.168.2.2
192.168.2.18	192.168.2.2
192.168.2.19	192.168.2.2
192.168.2.20	192.168.2.2
192.168.2.21	192.168.2.2
192.168.2.22	192.168.2.2
192.168.2.23	192.168.2.2
192.168.2.24	192.168.2.2
192.168.2.25	192.168.2.2
192.168.2.26	192.168.2.2
192.168.2.27	192.168.2.2
192.168.2.28	192.168.2.2
192.168.2.29	192.168.2.2
192.168.2.30	192.168.2.2

Tabla 12: Resultado 3

3) Una segunda pregunta interesante es para la misma máquina infectada saber cuántas y cuales máquinas rechazaron la conexión. La tabla 13 muestra el número de máquinas distintas que enviaron paquetes con los flags ACK, RST y sin el flag SYN. Los flujos se agruparon por IP de destino.

IP de destino	Nº IP de origen distintas
192.168.2.2	13

Tabla 13: Resultado 4

La tabla 14 muestra el detalle de las máquinas que rechazaron la conexión para la IP de destino igual a 192.168.2.2.

IP de origen	IP de destino
192.168.2.10	192.168.2.2
192.168.2.11	192.168.2.2
192.168.2.12	192.168.2.2
192.168.2.13	192.168.2.2
192.168.2.14	192.168.2.2
192.168.2.15	192.168.2.2
192.168.2.3	192.168.2.2
192.168.2.4	192.168.2.2
192.168.2.5	192.168.2.2
192.168.2.6	192.168.2.2
192.168.2.7	192.168.2.2
192.168.2.8	192.168.2.2
192.168.2.9	192.168.2.2

Tabla 14: Resultado 5

Utilizando las técnicas calce de IP y TCP flags se obtuvieron las máquinas que rechazaron las conexiones.

Mediante la primera consulta se logró determinar la máquina infectada luego con las siguientes consultas se determinaron las máquinas afectadas.

Un dato interesante es el número de máquinas rechazadas. Este valor puede indicar la naturaleza del servicio. Si el servicio fuera muy desconocido lo más probable es que tenga una tasa de rechazo mucho mayor. Por ejemplo, existe un proceso que realiza un escaneo sobre máquinas de la red para encontrar impresoras. Lo más probable es que nunca se rechace una conexión por ser un servicio conocido.

Las consultas propuestas en un medio real se comportarán bastante distintas. Esto se debe por no considerar el tiempo en sus cálculos. Por ejemplo, si una máquina un día no aceptó la conexión, y luego debido a cambios del usuario la máquina empieza a aceptar conexiones. Al analizar el tráfico la información podría no reflejar la existencia de una máquina infectada, obteniendo una consulta de seguridad poco efectiva para la seguridad informática.

5.1.2 Comportamiento 2: Tráfico N a 1

Este comportamiento se refiere al tráfico realizado por N máquinas hacia una máquina en común. Al analizar el tráfico IP este comportamiento es visto como un grupo de paquetes donde la IP de destino es la misma y la IP de origen es diferente para cada paquete del grupo.

Este comportamiento es visto en:

- Máquinas causando un DDoS, por inundación de paquetes: TCP con el flag SYN activado, paquetes ICMP echo o paquetes DNS, dirigidos a la dirección B.
- Máquinas realizando un escaneo distribuido a puertos de una misma máquina, o a una red en particular.

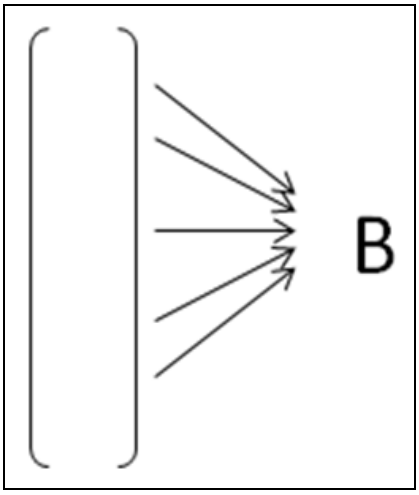

Figura 37: N a 1.

5.1.2.1 Simulación

Se creó escenario en una red local con asignación de máquinas 192.168.2.2 hasta 192.168.2.30.

Las máquinas con IP en el rango 192.168.2.3-192.168.2.15 se encuentran infectadas con un bot. Este realiza un escaneo distribuido sobre los puertos de una sola máquina para acelerar la infección. Las máquinas bot realizan un escaneo a la máquina 192.168.2.2 donde cada una escaneo rangos de 500 puertos. El escaneo se basa en el envío de paquetes TCP desde el puerto 2039 hacia los rangos mencionados.

5.1.2.2 Metodología

1) Este comportamiento puede ser detectado agrupando los paquetes por IP de destino y contabilizando el número de IP's de origen distintas para cada grupo. La tabla 15 muestra el resultado de la consulta propuesta.

IP de destino	Nº de IP de origen distintas
192.168.2.2	13

Tabla 15: Resultado 1.

El resultado 1 en la tabla 15 muestra el comportamiento fuera de lo normal sobre la máquina 192.168.2.2, pero aun así, esta máquina podría ser un servidor web y tener 13 máquinas distintas realizando tráfico.

2) Utilizando la consulta anterior y contabilizando los puerto de destino distintos

IP de destino	Nº de puertos de destino distintos
192.168.2.2	6500

Tabla 16: Resultado 2.

Se ve claramente en la tabla 16 que la IP 192.168.2.2 ha recibido paquetes hacia 6500 puertos distintos, lo que indica un posible escaneo de puertos. Se dice posible debido a que esta consulta no considera el tiempo. Posiblemente en un periodo largo de tiempo la máquina pueda llegar a tener un total de 6500 puertos distintos en los registros y no así estar infectada por un malware.

La consulta propuesta tampoco considera el tipo de flag, lo que podría dar conclusiones erróneas, por ejemplo los paquetes podrían ser paquetes con flags RST y ACK o ACK y SYN, lo cual indica que el rango en verdad está devolviendo la respuesta a un escaneo, en vez de escanear los puertos.

3) La tabla 17 muestra la consulta anterior considerando el tiempo, para esto se agrupo por fecha, hora y minuto.

IP de destino	Fecha y tiempo de flujo	Nº de puertos de destino distintos
192.168.2.2	12/05/2008 00:25	682
192.168.2.2	12/05/2008 00:26	1853
192.168.2.2	12/05/2008 00:27	1748
192.168.2.2	12/05/2008 00:28	1814
192.168.2.2	12/05/2008 00:29	403

Tabla 17: Resultado 3.

La máquina 192.168.2.2 fue atacada durante 5 minutos, reportando una gran cantidad de puertos distintos por cada minuto. Esto es un buen indicador y muestra la existencia de un malware.

5.1.3 Comportamiento 3: Tráfico 1 a N a 1

El comportamiento de tráfico 1 a N a 1 se basa en que una máquina envía paquetes a un rango de máquinas y luego el rango envía paquetes a una máquina en común distinta de la inicial.

Este comportamiento es visto en:

- Caso botnet: A es el mecanismo de control y comando, el rango son las máquinas controladas y B la máquina o red atacada.

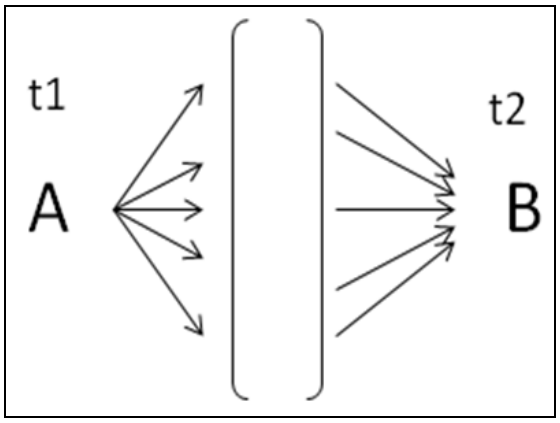

Figura 38: 1 a N a 1.

5.1.3.1 Simulación

La simulación muestra un ataque de DDoS hacia un servidor de DNS. En el ataque 15 máquinas son informadas para realizar un ataque dentro de 3 minutos, pero justo en ese tiempo algunas máquinas dejan de funcionar, solo una porción de estas realiza el ataque hacia el servidor de DNS.

Se tiene la dirección externa 244.2.2.3 enviando paquetes desde y hacia los puertos 6667 al rango de IP's 192.168.2.16-192.168.2.30, solo las IP en el rango 192.168.2.21-192.168.2.30 luego de 2 minutos envían paquetes DNS a la IP 211.2.2.4 y puerto 53.

5.1.3.2 Metodología

1) Una manera de detectar este comportamiento es contabilizar el número de máquinas que han recibido paquetes desde una máquina en común y han enviado paquetes a una máquina en común donde la máquina de origen y destino son distintas. La tabla 18 muestra el resultado del método de detección propuesto.

IP de origen máquina A	Nº de máquinas distintas	IP de destino máquina B
244.2.2.3	10	211.2.2.4

Tabla 18: Resultado 1.

2) La tabla 19 muestra el resultado de la consulta anterior considerando la fecha y la hora de los registros.

IP de origen máquina A	Nº de máquina distintas	IP de destino máquina B	Tiempo
244.2.2.3	10	211.2.2.4	12/05/2008 01

Tabla 19: Resultado 2.

Las consultas con parámetros de tiempo disminuye el margen de error significativamente, precisando la hora en que se manifestó el comportamiento.

5.1.4 Comportamiento 4: Tráfico N a 1 variando máquina de destino

El comportamiento N a 1 variando máquina de destino se refiere a un rango de máquinas generando tráfico hacia distintas máquinas en común.

Este comportamiento es visto en:

- Rango de máquinas comunicándose con mecanismo de control y comando cada cierto tiempo, debido a que la botnet cambia la dirección IP del mecanismo de control y comando

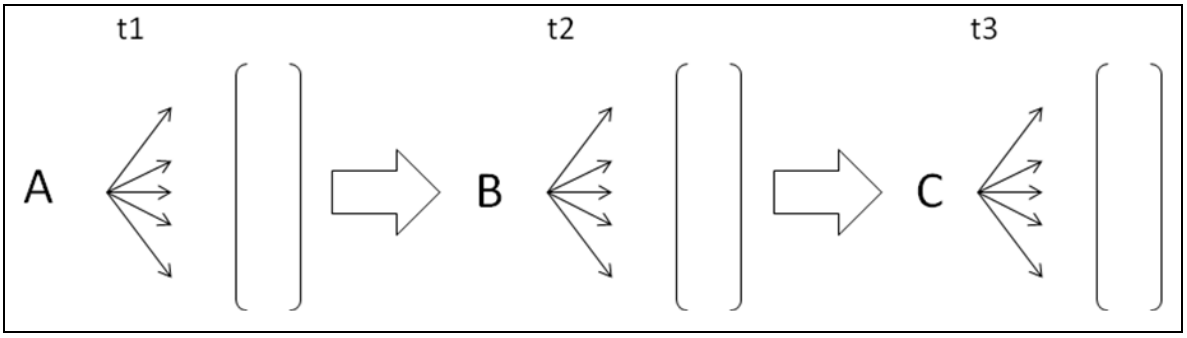

Figura 39: N a 1, variando máquina de destino

5.1.4.1 Simulación

Se tiene un rango de máquinas bot (192.168.2.20-192.168.2.30) enviando paquetes al mecanismo de control y comando de la botnet. Cada minuto el mecanismo de control y comando cambia su dirección IP a una distinta (200.10.0.2, 200.10.1.2, 200.10.2.2 y 211.2.4.4). En el proceso hay máquinas desconectadas de la red, por lo cual el rango disminuye. Esta técnica es conocida como fast-flux y es utilizada para dificultar la mitigación de la botnet.

5.1.4.2 Metodología

Este comportamiento es detectado luego de realizar las siguientes 3 consultas:

1) El resultado de la primera consulta es mostrada en la tabla 20 y agrupa las máquinas de destino que han recibido paquetes desde más de 5 máquinas distintas.

IP de destino	Nº IP de origen distintas
200.10.0.2	11
200.10.1.2	9
200.10.2.2	7
211.2.4.4	10

Tabla 20: Resultado 1

2) Utilizando los resultados anteriores se buscaron los tráficos donde la IP de destino pertenece al grupo resultante de la consulta anterior. El tráfico resultante se agrupó por IP de origen y se contabilizaron las IP distintas de destino (tabla 21).

IP de origen	Nº IP de destino distintas
192.168.2.20	2
192.168.2.21	3
192.168.2.22	4
192.168.2.23	4
192.168.2.24	4
192.168.2.25	4
192.168.2.26	4
192.168.2.27	4
192.168.2.28	4
192.168.2.29	3
192.168.2.30	1

Tabla 21: Resultado 2

3) En base al resultado anterior es útil saber la cantidad de repeticiones del número total de IP's de destino distintas (tabla 22)

Nº IP de destino distintas	Repeticiones
1	1
2	1
3	2
4	7

Tabla 22: Resultado 3

Luego de las 3 consultas creadas para este comportamiento se deduce que hay 7 máquinas que han tenido tráfico con las IP 200.10.0.2, 200.10.1.2, 200.10.2.2, 211.2.4.4, o sea las direcciones utilizadas por el C&C.

Una mejora a estas consultas es utilizar rangos de tiempo, para tener una mayor precisión. Para esta simulación no es de gran importancia ver su desarrollo, debido a que toda la información de tráfico correspondía a la simulación del comportamiento, por ende sólo se debió consultar toda la información, y no segmentada en el tiempo.

5.2 Experimento 2

Este experimento evalúa el sistema como medio de detección de malware en tráfico IP real. Se realizaron dos experimentos, el primero basado en el comportamiento 1 a N y el segundo en el comportamiento 1 a N a 1. Se almacenó información de tráfico real de 2 semanas.

Por cada experimentación se insertó en el tráfico real, en una fecha determinada, la simulación del comportamiento. Utilizando consultas derivadas del experimento 1 se buscó detectar con mejor precisión el campo simulado.

5.2.1 Simulación 1

La máquina con IP 172.17.67.2 se encuentra infectada con un bot el cual realiza un escaneo de máquinas para reclutar nuevas máquinas a la botnet. La máquina bot realiza un escaneo a dos rangos 172.17.0.2-172.17.0.50 y 172.17.67.10-172.17.67.100, El escaneo se basa en el envió de paquetes TCP con el flag SYN desde el puerto 2039 hacia el puerto 2040 en el rango de máquinas descrito.

Las máquinas existentes en la red responden a esta máquina de vuelta, las cuales dentro del rango 172.17.67.3-172.17.67.27 responden con el flag RST/ACK y la otra mitad con ACK/SYN.

La inserción de estos datos en el flujo real fue en el día 01/06/08.

5.2.1.1 Metodología

1) Se tomó 2 semanas de tráfico y por cada hora se contabilizaron las máquinas que hayan enviado, a más de 13 máquinas distintas, paquetes TCP. El número 13 fue estimado mediante pruebas con otros números buscando mostrar una cantidad razonable de máquina en la red.

Dirección IP de origen	Fecha y hora	Número de dirección IP de destino distintas
146.83.7.14	2008-05-29 10	15
146.83.7.14	2008-05-29 17	14
146.83.7.14	2008-05-30 15	14
172.17.67.2	2008-06-01 19	120
172.17.67.2	2008-06-01 20	20
146.83.7.14	2008-06-03 14	17
146.83.7.14	2008-06-04 15	14
146.83.7.14	2008-06-04 16	14
146.83.7.14	2008-06-04 17	15
146.83.7.14	2008-06-05 11	16

Tabla 23: Resultado 1

El resultado mostrado en la tabla 23 indica que para el primero de junio del 2008 a las 19 horas, 120 máquinas distintas recibieron paquetes IP desde un mismo origen. Posiblemente hay un gusano escaneando la red para esparcirse.

Mejoras

a. Se contabilizó por cada minuto y se redujo el número mínimo de direcciones IP distintas de destino a 6

Dirección IP de origen	Fecha y tiempo	Número de dirección IP de destino distintas
172.17.67.2	2008-06-01 19:59	120
172.17.67.2	2008-06-01 20:00	19
146.83.7.14	2008-06-03 11:38	6
146.83.7.14	2008-06-03 12:34	6
146.83.7.14	2008-06-03 13:04	6
146.83.7.14	2008-06-03 13:28	7
146.83.7.14	2008-06-03 14:06	7
172.17.67.46	2008-06-04 08:57	6

Tabla 24: Resultado 1.a

La reducción de la información por horas y minutos mostrada en la tabla 24 indica claramente que al primero de junio del 2008 a las 19:59 horas existió comunicación con 120 máquinas distintas, esto muestra que algo extraño está pasando en la red.

b. Se contabilizó por cada segundo, se disminuyó el número de direcciones IP distintas de destino a 3

Dirección IP de origen	Fecha y tiempo	Número de dirección IP de destino distintas
172.17.67.199	27/05/2008 15:32:08	3
172.17.67.25	28/05/2008 18:55:37	3
172.17.67.2	01/06/2008 19:59:56	9
172.17.67.2	01/06/2008 19:59:57	35
172.17.67.2	01/06/2008 19:59:58	38
172.17.67.2	01/06/2008 19:59:59	38
172.17.67.2	01/06/2008 20:00:00	19
172.17.67.46	04/06/2008 8:57:31	3
172.17.67.46	04/06/2008 8:57:37	3

Tabla 25: Resultado 1.b

La tabla 25 muestra claramente que por segundos existió tráfico sobre 30 máquinas. Esto indica que la máquina presenta posiblemente problemas.

El análisis hecho hasta ahora no toma en cuenta el tipo de comportamiento del tráfico IP, para tomarlo en cuenta es útil ocupar los TCP flags.

c. Se contabilizó por segundo, se agregó la restricción de flujos con el flag SYN activado y los flags RST y ACK desactivados, indicando que los flujos son de conexión y no de desconexión.

Dirección IP de origen	Fecha y tiempo	Número de dirección IP de destino distintas
172.17.67.2	2008-06-01 19:59:56	9
172.17.67.2	2008-06-01 19:59:57	20
172.17.67.2	2008-06-01 19:59:58	38
172.17.67.2	2008-06-01 19:59:59	38
172.17.67.2	2008-06-01 20:00:00	19
172.17.67.46	2008-06-04 08:57:31	3
172.17.67.46	2008-06-04 08:57:37	3

Tabla 26: Resultado 1.c

Se observa en la tabla 26 un comportamiento de 1 a N con un gran número de paquetes queriendo iniciar un gran número de conexiones, determinando un posible escaneo de IP.

d. Se contabilizó por hora y se agregó la restricción sobre los flag TCP RST y ACK activados y no activado el flag SYN.

Dirección IP de origen	Fecha y hora	Número de dirección IP de destino distintas
146.83.7.3	2008-06-04 9	3

Tabla 27: Resultado 1.d

La tabla 27 muestra el número de IP's que recibieron paquetes con flags RST y ACK. Ósea no se les permitió realizar la conexión. El número de direcciones IP's de destino distintas no muestra ningún valor fuera de lo normal, por lo tanto es difícil sacar conclusiones por medio de esta información.

2) Por cada hora se contabilizó las máquinas que recibieron paquetes TCP provenientes de 7 máquinas distintas. El número 7 fue estimado mediante pruebas con otros números buscando mostrar una cantidad razonable del número de direcciones IP de origen distintas.

Dirección IP de destino	Fecha y hora	Número de dirección IP de origen distintas
172.22.162.1	2008-05-23 22	7
172.22.162.1	2008-05-24 1	8
172.22.162.1	2008-05-24 3	7
172.22.162.1	2008-05-29 7	7
172.22.162.1	2008-06-01 14	8
172.17.67.2	2008-06-01 20	28
172.22.162.1	2008-06-05 3	7

Tabla 28: Resultado 2

Se ve claramente en la tabla 28 la máquina 172.17.67.2 ha recibido paquetes proveniente de 28 máquinas distintas el primero de junio del 2008 a las 20 horas, esto indica que la máquina 172.17.67.2 recibió tráfico, más o menos, a la misma hora que ocurrió el escaneo. Este resultado indicaría que posiblemente del mismo rango escaneado respondieron a la conexión.

Mejoras

a. Se contabilizó por minuto , limitando el número de direcciones IP de origen distintas a mayores de 2

Dirección IP de destino	Fecha y tiempo	Número de dirección IP de origen distintas
172.17.67.140	2008-05-27 16:24	3
172.17.67.187	2008-05-29 16:39	3
172.17.67.100	2008-05-30 16:32	4
172.17.67.2	2008-06-01 20:00	28
172.17.67.85	2008-06-03 13:56	3
172.17.67.73	2008-06-03 18:42	3
172.17.67.98	2008-06-04 13:13	3
146.83.7.3	2008-06-04 19:49	3
172.17.67.57	2008-06-05 14:16	3

Tabla 29: Resultado 2.a

La tabla 29 muestra que la dirección 172.17.67.2 es la dirección IP de la máquina comprometida.

b. Se contabilizó por segundo, limitando el número de direcciones IP de origen distintas a mayores de 2

Dirección IP de destino	Fecha y hora	Número de dirección IP de origen distintas
172.17.67.100	2008-05-30 16:32:09	3
172.17.67.2	2008-06-01 20:00:00	18
172.17.67.2	2008-06-01 20:00:01	10
172.17.67.73	2008-06-03 18:42:49	3
172.17.67.57	2008-06-05 14:16:30	3

Tabla 30: Resultado 2.b

La tabla 30 indica que la máquina 172.17.67.2 recibe tráfico desde 18 máquinas en tan solo un segundo, lo cual es aun más extraño indicando la presencia de un malware.

c. Además de contabilizar por fecha se agregó la restricción de flujos con el flag RST y ACK activado y desactivado los flags SYN, lo cual indicaría que son flujos de desconexión.

Dirección IP de destino	Fecha	Número de dirección IP de origen distintas
172.17.67.2	2008-06-01	13
172.17.67.146	2008-06-02	3

Tabla 31: Resultado 2.c

La tabla 31 muestra que la máquina de destino ha recibido 13 términos de conexiones TCP, indicando que esta máquina ha intentado conectarse seguidamente a distintas máquinas.

d. Además de contabilizar por minuto se agregó la restricción de flujos con el flag RST y ACK activado y desactivado los flags SYN, lo cual indicaría que son flujos de desconexión.

Dirección IP de destino	Fecha y tiempo	Número de dirección IP de origen distintas
172.17.67.224	2008-05-28 12:37	2
172.17.67.2	2008-06-01 20:00	13

Tabla 32: Resultado 2.d

La tabla 32 muestra claramente la existencia del malware simulado. Indicando que la máquina 172.17.67.2 recibió 13 rechazos de conexión en un minuto. Esto muestra que la máquina realizó un escaneo hacia las otras, comprobando la detección del malware en tráfico IP real.

5.2.2 Simulación 2

Para el mismo tráfico real se insertó el comportamiento de la máquina 244.2.2.3 generando tráfico hacia el rango de IP 172.17.67.1-172.17.61.61 y luego el rango de máquinas envía paquetes IP a la dirección 211.2.2.4.

5.2.2.1 Metodología

Se agrupó la información proveniente del sensor por fecha, hora, dirección IP de origen y dirección IP de destino. La información resultante se cruzó con la misma información con las siguientes restricciones.

- Dirección IP de destino en la primera información debe ser igual a la IP de origen de la segunda información.
- Las fechas y horas deben coincidir en las dos informaciones resultantes.
- Dirección IP de origen en la primera información debe ser distinta a la IP de destino de la segunda información.
- Número total de máquinas enlazadas mayor a 10.

1) Tráfico sin simulación de malware.

Fecha y hora	Dirección IP de origen Tabla 1	Dirección IP de destino Tabla 2	Número total de máquinas enlazadas
2008-05-27 10	172.17.67.1	172.17.67.255	11
2008-05-27 11	146.83.7.14	172.17.67.255	11
2008-05-29 8	172.17.67.1	172.17.67.255	11
2008-05-29 10	146.83.7.14	172.17.67.255	12
2008-06-03 14	146.83.7.14	172.17.67.255	12
2008-06-04 15	146.83.7.14	172.17.67.255	11
2008-06-04 15	172.17.67.1	172.17.67.255	11
2008-06-04 16	146.83.7.14	172.17.67.255	11
2008-06-04 17	146.83.7.14	172.17.67.255	11
2008-06-04 18	172.17.67.1	172.17.67.255	12
2008-06-05 11	146.83.7.14	172.17.67.255	12
2008-06-05 14	172.17.67.1	172.17.67.255	11

Tabla 33: Resultado 1

La tabla 33 muestra los resultados de la consulta creada sin la inserción del tráfico simulado.

2) Tráfico con simulación.

Fecha y hora	Dirección IP de origen Tabla 1	Dirección IP de destino Tabla 2	Número total de máquinas enlazadas
2008-05-27 10	172.17.67.1	172.17.67.255	11
2008-05-27 11	146.83.7.14	172.17.67.255	11
2008-05-29 8	172.17.67.1	172.17.67.255	11
2008-05-29 10	146.83.7.14	172.17.67.255	12
2008-06-01 10	244.2.2.3	211.2.2.4	60
2008-06-03 14	146.83.7.14	172.17.67.255	12
2008-06-04 15	146.83.7.14	172.17.67.255	11
2008-06-04 15	172.17.67.1	172.17.67.255	11
2008-06-04 16	146.83.7.14	172.17.67.255	11
2008-06-04 17	146.83.7.14	172.17.67.255	11
2008-06-04 18	172.17.67.1	172.17.67.255	12
2008-06-05 11	146.83.7.14	172.17.67.255	12
2008-06-05 14	172.17.67.1	172.17.67.255	11

Tabla 34: Resultado 2

Se ve claramente en la tabla 34 que la máquina maliciosa (244.2.2.3) sobresale del resto, cumpliendo con los objetivos de la consulta creada.

Para esta simulación se logro mostrar la efectividad de las consultas creadas para la seguridad informática en cuanto a la detección de anomalías en tráfico IP.

6 Extensiones: Análisis de flujo en redes con NAT

Un problema que aparece cuando se analiza flujos de paquetes IP, es el análisis entre dos segmentos de red distintos. Las redes más comunes utilizadas en organizaciones reciben una IP externa asignada por el ISP de forma estática o dinámica. Las organizaciones para mantener una mayor cantidad de máquinas conectadas a internet utilizan un router y una red privada interna. El router asigna a las máquinas de la organización una IP especial. Además el router por medio del NAT mantiene la comunicación entre Internet y las máquinas. Para lograrlo los paquetes que pasan a través del router son modificados y redirigidos.

El problema surge debido al desconocimiento de la modificación realizada por el router. En casos más complejos puede existir más de un router modificando la información del paquete IP.

Si se quisiera identificar una máquina maliciosa que causó un problema en una red donde la máquina maliciosa no pertenezca a la red, la solución no es tan directa. Esto se debe principalmente por las modificaciones realizadas a los paquetes (NAT), pérdida de paquetes y fragmentación entre otros.

Este problema no está dentro de los objetivos iniciales de la memoria. Sin embargo se realizó un experimento para clarificar el problema y buscar consultas útiles para el problema de NAT en dos segmentos diferentes de red.

El experimento busca para dos redes con un sensor de flujos IP por red y un colector de flujos, encontrar consultas apropiadas para identificar de forma precisa la máquina emisora de flujos IP. Por medio de la información de tráfico propia de cada red.

Se obtuvieron resultados buenos para casos con condiciones favorables.

Los resultados toman el rol de ser una primera exploración al problema, y en ningún caso un estudio definitivo del problema.

6.1 Ambiente distribuido

La Figura 40 muestra la red A conformada por un PC con dirección IP 192.168.1.4 y un sensor capturando la información de tráfico del mismo segmento (PC 1 actúa como el sensor S1 del segmento). Se desconoce los otros sectores de la red A y se sabe que el PC 1 tiene acceso a Internet. Y además envía la información de tráfico hacia el colector en el puerto 12345 de la red B. La IP asignada por el ISP para la red A es 190.21.80.166.

La red B se conforma con un PC con dirección IP 192.168.1.2 y un sensor de tráfico IP en el mismo segmento de red (El PC 2 actúa como sensor de flujos IP). También se desconoce los otros sectores de la red. El PC 2 permite acceder a los puertos 22(SSH),

12345(sensor de flujos IP) y 12346(VNC). Para simplificar la experimentación, la red B está configurada de tal forma que los paquetes que lleguen por Internet hacia los puertos 22, 12345 y 12346 serán redirigidos al PC 2 a los puertos respectivos. Además el PC 2 actúa como colector de flujos IP y recibe los flujos del sensor S1 y S2 por medio del puerto 12345 y 12344 respectivamente. La IP asignada por el ISP para la red B es 190.160.228.185.

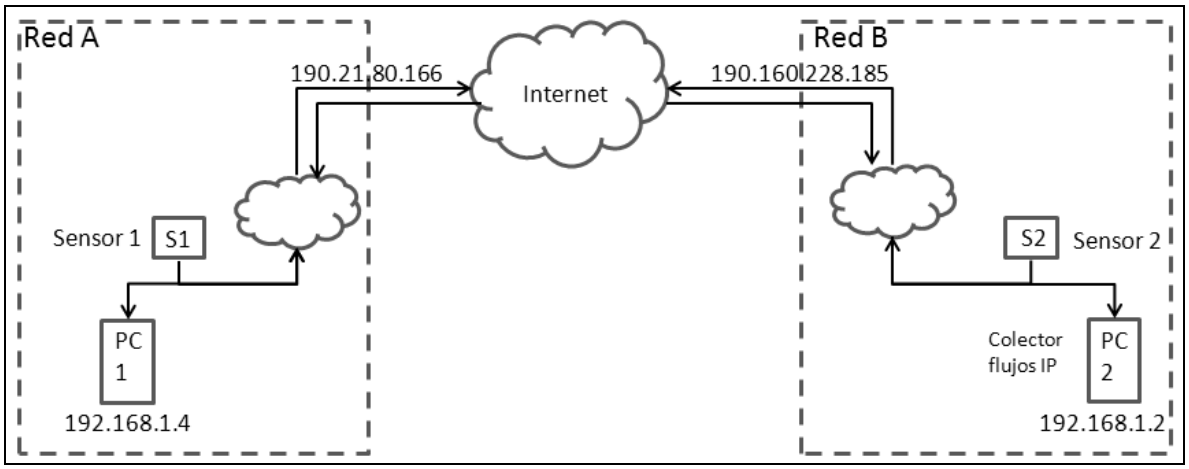

Figura 40: Ambiente distribuido

6.2 Ejemplificación de problema

Cuando la red A de la figura 40 envían paquetes IP hacia la red B por Internet. La información de tráfico de los paquetes es modificada. Por ejemplo utilizando la figura 40, el PC 1 envía paquetes IP al puerto 12345 de la red B. El sensor S1 almacena la información de la tabla 35.

IP origen	Puerto origen	IP destino	Puerto destino	N° Bytes	N° Paquetes	Protocolo	Tiempo de inicio
192.168.1.4	50	190.160.228.185	12345	4653	11	UDP	12:20:22

Tabla 35: Información de tráfico paquete IP sensor S1

La red B al recibir los paquete IP lo redirige al PC 2 y el sensor S2 almacena la información de la tabla 36.

IP origen	Puerto origen	IP destino	Puerto destino	N° Bytes	N° Paquetes	Protocolo	Tiempo de inicio
190.21.80.166	13	192.168.1.2	12345	4230	10	UDP	12:20:32

Tabla 36: Información de tráfico paquete IP sensor S2

Se ve que por medio de la información del sensor S1 la dirección IP de destino en el flujo es la dirección del ISP de la red B. Para la información del sensor S2 se ve que la

dirección IP de origen y el puerto de origen no corresponde a los del PC 1. El número de paquetes a veces no calza debido a la pérdida de paquetes en la red. Como tampoco el número de Bytes en casos donde exista fragmentación.

6.3 Datos de prueba

Luego de la configuración del ambiente distribuido, se almacenaron 2 horas de tráfico para cada sensor en el colector de flujos en la red B. En estas horas se realizaron constantemente cada 5 minutos envíos de paquetes UDP provenientes del sensor de flujos S1 hacia el colector de flujos. También se realizaron 3 conexiones por medio de VNC desde el PC 1 hacia el PC 2 y 3 conexiones por medio de SSH desde el PC 1 hacia el PC 2.

6.4 Estrategia de identificación

Se crearon para el ambiente distribuido y la información almacenada consultas posibles para relacionar los flujos enviados desde el PC 1 hacia el PC 2. La estrategia desarrollada pretende por medio de relaciones directas entre la información de los flujos almacenados por los sensores, identificar la máquina emisora de paquetes de la red A en la red B. Por ejemplo, se sabe que los flujos generados por el sensor S1 tiene en total “X” bytes e “Y” paquetes. La estrategia plantea seleccionar los registros del sensor S2 que tengan “X” bytes e “Y” paquetes. De forma intuitiva se sabe que posiblemente los flujos corresponden al tráfico real.

6.4.1 Heurística 1

La primera heurística supone que los flujos IP entre las dos redes siguen las siguientes relaciones:

- El puerto de destino del flujo IP asociado al sensor S1 y el puerto de destino del flujo IP asociado al sensor S2 son iguales.
- El tiempo de inicio del flujo en el sensor S1 y el tiempo de inicio del flujo en el sensor S2 no tiene más de 20 segundos de diferencia.
- El valor de la suma total de bytes para los flujos IP asociados a los sensores S1 y S2 son iguales.
- El número de paquetes totales para los flujos IP asociados a los sensores S1 y S2 son iguales.
- Los flags TCP ACK, SYN, FIN y RST para los flujos IP asociados a los sensores S1 y S2 son iguales. En caso de no ser Flujos asociados a paquetes TCP los flags son marcados con 0.
- El protocolo para los flujos IP asociados a los sensores S1 y S2 son iguales.

El resultado de la primara consulta se muestra en la tabla 23.

Sensor S1		Sensor S2	
Información de origen	**Información de destino**	**Información de origen**	**Información de destino**
192.168.1.4:48168	190.160.228.185:22	190.21.80.166:19101	192.168.1.2:22
192.168.1.4:52664	190.160.228.185:12345	190.21.80.166:19104	192.168.1.2:12345
192.168.1.4:57723	190.160.228.185:12346	190.21.80.166:20722	192.168.1.2:12346

Tabla 37: Resultado de utilizar heurística 1

La tabla 37 muestra claramente la correspondencia de los flujos entre la información almacenada por los sensores. En casos donde la red es relativamente buena, esto es, el envío de paquetes se demora menos de 20 segundos y no existe constantemente pérdida de paquetes IP en el camino. Junto a que los sensores de flujos IP utilicen las mismas reglas de creación y los segmentos donde se encuentren los sensores utilicen el mismo tamaño máximo para paquetes IP. En principio esta heurística funcionaría como primera aproximación.

El ambiente de prueba al seguir todos los supuestos mencionados anteriormente entrega buenos resultados para la heurística 1. Como resultado de esta consulta resulta interesante estudiar otros casos donde no se tenga tan buen ambiente.

Una manera para probar consultas en donde no se tenga un ambiente tan bueno, es relajando las condiciones en el ambiente construido, esto es quitando condiciones a la consulta.

6.4.2 Heurística 2:

Utilizando el mismo ambiente y datos de prueba se relaja la condición sobre el total de bytes por flujo en los sensores S1 y S2. La idea principal de esta heurística es ver la influencia del número de bytes por flujos en la consulta.

La tabla 38 muestra los resultados para la consulta 2 asociada a la heurística 2.

Sensor S1		Sensor S2	
Información de origen	**Información de destino**	**Información de origen**	**Información de destino**
192.168.1.4:34294	192.168.1.1:53	192.168.1.2:43696	192.168.1.1:53
192.168.1.4:34568	192.168.1.1:53	192.168.1.2:33148	192.168.1.1:53
192.168.1.4:34795	192.168.1.1:53	192.168.1.2:50411	192.168.1.1:53
192.168.1.4:35331	192.168.1.1:53	192.168.1.2:60417	192.168.1.1:53
192.168.1.4:36312	192.168.1.1:53	192.168.1.2:58490	192.168.1.1:53
192.168.1.4:37784	192.168.1.1:53	192.168.1.2:37604	192.168.1.1:53
192.168.1.4:37784	192.168.1.1:53	192.168.1.2:47851	192.168.1.1:53
192.168.1.4:37784	192.168.1.1:53	192.168.1.2:60098	192.168.1.1:53
192.168.1.4:38632	192.168.1.1:53	192.168.1.2:38417	192.168.1.1:53
192.168.1.4:39027	192.168.1.1:53	192.168.1.2:47016	192.168.1.1:53
192.168.1.4:39622	192.168.1.1:53	192.168.1.2:32888	192.168.1.1:53
192.168.1.4:41993	192.168.1.1:53	192.168.1.2:49266	192.168.1.1:53
.....	.....	.....	.....

Tabla 38: Resultado de utilizar heurística 2

Al relajar la restricción sobre el tamaño de los flujos se observó una gran cantidad de relaciones entre los flujos. Esta fue superior a 600 relaciones. Como se sabe que entre las redes solo hubo 3 puertos por los que se relacionaban. Las relaciones de la tabla 38 no aportan con información válida.

En la tabla 38 se muestran solo algunas relaciones como forma de ejemplificación.

Como conclusión a esta heurística se ve la gran influencia que tiene el número de bytes por flujo dentro de la consulta.

6.4.3 Heurística 3:

La segunda heurística mostró que al solo quitar la restricción sobre los bytes de los flujos IP, se pierden las relaciones verdaderas.

La condición sobre el total de bytes es bastante fuerte. Una manera para no usar la condición sobre el total de bytes por flujo de forma directa, es considerando los bytes por paquete IP promedio no así los bytes por flujos. Esta condición busca encontrar una mayor cantidad de relaciones correctas para un ambiente donde existan pérdidas de paquetes y los servicios del ambiente que generen tráfico utilicen un tamaño de bytes por paquete constante. Para esto se tomó el total de bytes por flujo y se dividió por el total de paquetes. Con esto se obtiene el número de bytes promedio posibilitando obtener las relaciones que hayan sufrido pérdida de paquetes.

Para cada flujo asociado a los sensores S1 y S2 se calculó el valor propuesto y se agregó como condición que estos valores fueran iguales.

El resultado de la consulta 3 se muestra en la tabla 39.

Sensor S1		Sensor S2	
Información de origen	**Información de destino**	**Información de origen**	**Información de destino**
192.168.1.4:48168	190.160.228.185:22	190.21.80.166:19101	192.168.1.2:22
192.168.1.4:52664	190.160.228.185:12345	190.21.80.166:19104	192.168.1.2:12345
192.168.1.4:57723	190.160.228.185:12346	190.21.80.166:20722	192.168.1.2:12346

Tabla 39: Resultado de utilizar heurística 3

Los resultados de la tabla 39 muestran que esta consulta, para un ambiente bueno, obtiene relaciones verdaderas. Esta consulta aunque considera los casos donde exista pérdida de paquetes necesita un supuesto bastante fuerte, como es el que los paquetes del servicio deban tener el mismo tamaño.

Una buena pregunta es saber cuan probable es que los paquetes enviados por un programa tengan el mismo tamaño. Si se piensa bien como primera aproximación se podría decir que va a depender del tipo de programa. Por ejemplo si el programa fuera de transferencia de datos, lo más probable es que la mayoría de los paquetes tengan valores constantes debido a que el programa tendrá que fragmentar el archivo para enviar la información.

6.4.4 Heurística 4:

Esta heurística, en vista de los resultados anteriores, en particular de la heurística 2, se basó en una nueva condición para relacionar los flujos. Los flujos por sensor en que los primeros 3 octetos de la dirección IP de destino es distinta de los primeros 3 octetos de la dirección IP de origen del mismo flujo. Esta nueva condición junto a la condición sobre la diferencia de los tiempos de inicio del flujo para los sensores, la igualdad de flags TCP e igualdad de puertos de destino, entregan los resultados mostrados en la tabla 40.

Sensor S1		Sensor S2	
Información de origen	**Información de destino**	**Información de origen**	**Información de destino**
192.168.1.4:33675	74.125.45.17:80	192.168.1.2:34733	77.247.176.134:80
192.168.1.4:48168	190.160.228.185:22	190.21.80.166:19101	192.168.1.2:22
192.168.1.4:52664	190.160.228.185:12345	190.21.80.166:19104	192.168.1.2:12345
192.168.1.4:57722	190.160.228.185:12346	190.21.80.166:20719	192.168.1.2:12346
192.168.1.4:57722	190.160.228.185:12346	190.21.80.166:20722	192.168.1.2:12346
192.168.1.4:57723	190.160.228.185:12346	190.21.80.166:20719	192.168.1.2:12346
192.168.1.4:57723	190.160.228.185:12346	190.21.80.166:20722	192.168.1.2:12346
192.168.1.4:57723	190.160.228.185:12346	190.21.80.166:20724	192.168.1.2:12346
192.168.1.4:57724	190.160.228.185:12346	190.21.80.166:20719	192.168.1.2:12346
192.168.1.4:57724	190.160.228.185:12346	190.21.80.166:20722	192.168.1.2:12346
192.168.1.4:57724	190.160.228.185:12346	190.21.80.166:20724	192.168.1.2:12346
192.168.1.4:59848	74.125.45.19:80	192.168.1.2:32884	77.247.176.134:80

Tabla 40: Resultado de utilizar heurística 4

Al analizar los resultados de la tabla 40 se concluye que se logra de alguna manera encontrar relaciones validas. Sin embargo estos flujos no todos existen, se ve que el flujo 4 y 5 de arriba hacia abajo, aparece por el cruce con la tabla de la red B.

Si se piensa un poco la idea de la condición creada sobre los primeros 3 octetos, busca filtrar los datos eliminando los paquetes que tiene que ver con las máquinas de la misma red.

Aunque se vea que esta condición funciona bastante bien, se cree que necesita muchas más pruebas para asegurar su efectividad. Una de las razones principales para pensar esto, es que en rango de tiempos mayores la cantidad de datos aumenta. Y al ser mayor la cantidad de datos, el número de cruces verdaderos o falsos aumentara considerablemente, sin poder distinguir los flujos reales de los falsos.

6.4.5 Heurística 5:

Luego de los casos vistos anteriormente aparece una nueva pregunta interesante, ¿Que pasaría si no se conocieran los servicios expuestos por la red B? . Para responder a esta pregunta se tomaron las restricciones sobre la igualdad de flags, la igualdad sobre el tamaño en bytes por paquete, el filtrado de los 3 primeros octetos y se condicionó la diferencia de tiempo entre los flujos de los sensores a valores entre 14 y 16 segundos. Por medio de pruebas se fue acotando las restricciones sobre el tiempo entre los flujos. Estos valores fueron producto de la búsqueda de una cantidad razonable de relaciones.

El resultado de la consulta es mostrada en la tabla 41.

Sensor S1		Sensor S2	
Información de origen	**Información de destino**	**Información de origen**	**Información de destino**
190.160.228.185:12346	192.168.1.4:57722	192.168.1.2:12346	190.21.80.166:20719
190.160.228.185:12346	192.168.1.4:57723	190.21.80.166:20722	192.168.1.2:12346
190.160.228.185:12346	192.168.1.4:57723	192.168.1.2:12346	190.21.80.166:20722
190.160.228.185:22	192.168.1.4:48168	192.168.1.2:22	190.21.80.166:19101
192.168.1.4:48168	190.160.228.185:22	190.21.80.166:19101	192.168.1.2:22
192.168.1.4:57722	190.160.228.185:12346	192.168.1.2:12346	190.21.80.166:20719
192.168.1.4:57723	190.160.228.185:12346	190.21.80.166:20722	192.168.1.2:12346
192.168.1.4:57723	190.160.228.185:12346	192.168.1.2:12346	190.21.80.166:20722

Tabla 41: Resultado de utilizar heurística 5

Las relaciones mostradas en la tabla 41 muestran claramente relaciones verdaderas. Además por no tener la limitación de los puertos se logra ver los flujos que viajan como respuesta de los enviados por el PC 1.

Los resultados de la consulta 5 corroboran, a primera vista, lo fuerte que es utilizar los tamaños de los flujos IP como condición.

6.4.6 Heurística 6:

Utilizando los resultados obtenidos en las heurísticas 1, 3, 4 y 5, se unieron las relaciones resultantes en una tabla y luego se agruparon los flujos resultantes por la información de origen y destino para los dos sensores. La información mostrada en la tabla 42 corresponde al número de relaciones repetidas dentro de las heuristicas 1, 3, 4 y 5.

Información de destino sensor S1	Información de origen sensor S2	Total repeticiones
190.160.228.185:12345	190.21.80.166:19104	2
190.160.228.185:12346	190.21.80.166:20719	3
190.160.228.185:12346	190.21.80.166:20722	3
190.160.228.185:12346	190.21.80.166:20724	2
190.160.228.185:12346	192.168.1.2:12346	2
190.160.228.185:22	190.21.80.166:19101	2

Tabla 42: Resultado de utilizar heurística 6

La idea principal de la heurística 6 es utilizar la información indirecta proporcionada por las consultas anteriores. En especial con la heurística 2 no se tiene certeza sobre las relaciones reales. Sin embargo, si las relaciones coinciden con alguna otra perteneciente a las otras consultas, será un buen indicio y aumenta la probabilidad de ser una relación correcta.

Los resultados en la tabla 42 muestran que la segunda y tercera fila muestran la mayor cantidad de relaciones repetidas, lo cual podría ser un indicador de relaciones correctas.

6.4.7 Heurística 7:

Para hacer las consultas anteriores sobre los datos de los dos sensores se utilizaron tablas por separado con la información de cada sensor. La información se consultaba de algún modo sabiendo que las tablas contenían la información de cada sensor. Esta heurística intenta ver qué pasa si re-agrupara la información de los sensores en una sola tabla y se ejecutara la consulta.

Se tomaron las restricciones sobre la igualdad de flags, la igualdad sobre el tamaño en bytes por paquete, el filtrado de los 3 primeros octetos y se condicionó la diferencia de tiempo entre los flujos de los sensores a valores entre 14 y 16 segundos. Estos valores fueron escogidos luego de probar varios rangos, valores menores a 14 segundos no entregaban resultados y los valores sobre 16 arrojaban muchos mas resultados, por lo cual se decidió restringir solo las relaciones comprendidas entre 14 y 16 segundos.

Sensor S1		Sensor S2	
Información de origen	**Información de destino**	**Información de origen**	**Información de destino**
190.160.228.185:12346	192.168.1.4:57722	192.168.1.2:12346	190.21.80.166:20719
190.160.228.185:12346	192.168.1.4:57723	190.21.80.166:20722	192.168.1.2:12346
190.160.228.185:12346	192.168.1.4:57723	192.168.1.2:12346	190.21.80.166:20722
190.160.228.185:22	192.168.1.4:48168	192.168.1.2:22	190.21.80.166:19101
….	….	….	….

Tabla 43: Resultado de utilizar heurística 7

El resultado de la consulta se muestra en la tabla 43, destaca una gran cantidad de incoherencias provenientes del cruce con toda la información de los sensores. Esto indica, a priori, que habrá que buscar otra manera de cruzar toda la información.

6.4.8 Heurística 8:

Esta heurística utiliza la consulta para la heurística 6 principalmente, pero en vez de utilizar dos tablas como fuente de datos considera una sola tabla con toda la información. La idea es probar que de alguna forma es posible hacer consultas a la información de los flujos proveniente de los sensores S1 y S2 sin tener que distinguir entre los flujos pertenecientes a cada sensor.

Lo útil de encontrar alguna consulta en este contexto es el poder a futuro unir la información de varios sensores y realizar la consulta para toda esa información. La manera normal sería ejecutando las consultas 1, 3, 4 y 5 por cada par de sensores, lo cual es muy engorroso.

Información de destino sensor S1	Información de origen sensor S2	Total repeticiones
190.160.228.185:12344	192.168.1.2:51012	2
190.160.228.185:12344	192.168.1.2:51013	2
190.160.228.185:12345	190.21.80.166:19104	2
190.160.228.185:12346	190.21.80.166:20719	3
190.160.228.185:12346	190.21.80.166:20722	3
190.160.228.185:12346	190.21.80.166:20724	2
190.160.228.185:12346	192.168.1.2:12346	2
190.160.228.185:22	190.21.80.166:19101	2
192.168.1.2:12345	192.168.1.4:52664	2
192.168.1.2:12346	192.168.1.4:57722	2
192.168.1.2:12346	192.168.1.4:57723	3
192.168.1.2:12346	192.168.1.4:57724	3
192.168.1.2:22	192.168.1.4:48168	2
218.186.17.110:18278	192.168.1.4:48075	2
77.247.176.134:80	192.168.1.4:59848	2

Tabla 44: Resultado de utilizar heurística 8

Los resultados mostrados en la tabla 44 mejoran considerablemente las fallas encontradas en la heurística 7, sin embargo el tiempo tomado por esta consulta, para el mismo número de flujos almacenados, es del orden de 12 veces superior.

En resumen se mostraron variadas heurísticas para distintos casos. Como resultado se cuenta, a priori, con consultas para la identificación de máquinas entre segmentos diferentes. Las consultas no lograron obtener una identificación absoluta para cualquier caso. Sin embargo, asumiendo algunas condiciones en los flujos es posible mejorar la identificación de los flujos IP.

7 Trabajos Futuros

Dado lo realizado en la memoria se pueden mencionar varias líneas de trabajo:

- Problema de NAT

Se piensa que este problema es de bastante importancia para el análisis de flujos IP en redes distribuidas. Un supuesto inicial de la memoria fue que aunque existiera NAT en el recorrido de los paquetes, el conocer el conjunto de máquinas que potencialmente hicieron posible el ataque, sería suficiente para detectar el problema. Sin embargo, esto no es tan cierto, a veces las redes tienen por segmento una cantidad considerable de máquinas. Por ejemplo, si se tuviera una red con 100 máquinas y se supiera que el tráfico malicioso proviene de este grupo. En este caso ya no es tan fácil detectar la máquina.

En la sección 6 se dieron los primeros pasos sobre este problema. Los experimentos desarrollados necesariamente necesitan una mirada más crítica, con más casos de prueba, diferentes ambientes, una muestra de datos mayor, etc.

- Problema de mucha información

Una duda que sale generalmente cuando se maneja cantidades grandes de información, es como procesarla. Como trabajo futuro es posible indagar en métodos para procesar cantidad de tamaños grandes de información.

- Mejoras a sistema

Luego del diseño del sistema y utilizarlo para las evaluación se llegaron a varios puntos que podrían ser útiles en una segunda iteración del desarrollo del sistema implementado.

    - Usuarios

El sistema actualmente permite tener usuarios distintos y con estos ingresar a las redes definidas por el administrador. Se plantea como mejora agregar roles a los usuarios limitándolos a distintas secciones del sistema. Por ejemplo, rol desarrollador podría encargarse de crear consultas de seguridad. Rol Cliente, este solo puede ver resultados de consultas hechas por el desarrollador.

    - On-line

El sistema implementado se basa principalmente en utilizar la información de forma batch para el análisis. Se plantea como mejora poder tener información en línea, esto es poder visualizar en tiempos cortos métricas, medidas de efectividad y otra información estadística. Por ejemplo, saber la cantidad de información transferida en los últimos 5

minutos. Además de estas características se le podrían agregar reglas on-line para la detección de anomalías.

Un punto a favor con esta característica es que se podría empezar comparar información on-line con información batch.

- Almacenamiento de problemas anteriores

Otra idea que se plantea es el crear un módulo que permita categorizar las consultas. Pudiendo asociarlas a malware o a problemas específicos, de forma fácil.

- Conocimiento de la red

Luego de la experimentación se encontró que saber la topología de la red donde se encuentran los sensores, es fundamental para no sacar conclusiones erróneas. Se plantea para este sistema agregar un módulo que permite ingresar información de la red. La información de la red ingresada debe poderse contrastar con la información proporcionada por el sistema.

- Búsqueda de nuevas consultas para la detección de malware

Gran parte de la experimentación fue crear consultas útiles para la detección de malware. Se plantea que en base a las consultas creadas, se puede seguir mejorándolas o creando nuevas consultas para las búsquedas de anomalías mediante la información del tráfico IP.

8 Conclusiones

En cuanto al trabajo realizado se logra concluir principalmente que el sistema es una ayuda para la seguridad informática, aumentando y complementando la seguridad computacional en las organizaciones.

Primero se debió diseñar el sistema para monitorear redes IP. Este consistió en un sistema que utilizaba la información flujos de paquetes IP (flujos Netflow). Para enfocarse en la creación del sistema de monitoreo se ocupó una herramienta de código abierto para recibir, almacenar y analizar flujos IP (Nfdump). Esta herramienta fue seleccionada por cumplir con los requisitos más importantes y al mismo tiempo ser adaptable.

El diseño necesitaba ser flexible en cuanto al análisis de la información de flujo. Para esto se seleccionó MySQL como motor de consulta. Las razones principales son el poder contar con una sintaxis de consulta conocida, aguantar una cantidad grande de información y ser una herramienta consolidada. Con las componentes principales se decidió hacer un sistema web con PHP, MySQL y Perl.

Luego se diseñaron las componentes necesarias para hacer la experimentación con los flujos IP, lo más flexible posible. Se crearon módulos para filtrar la información de flujo proveniente de los sensores, hacer consultas vía SQL, cruzar la información con información externa y almacenar los resultados provenientes de las consultas.

Con el sistema desarrollado se hicieron test de capacidad con respecto a tiempos de ejecución de consultas y tamaños de almacenamiento de flujos IP. En resumen se concluyó que los tiempos de consulta son poco medibles debido a que dependen absolutamente de la consulta. La información propia de la red es poco parametrizable debido a la variabilidad de las topologías. Por ejemplo, una red podría tener 100 máquinas conectadas a la red, pero dependiendo del uso que tenga, se podría encontrar una red con solo 10 máquinas que realizan la misma cantidad de tráfico. Esto dificulta la estimación de información a almacenar por el sistema.

Se midió para una red normal el total de flujo IP almacenados por 10 días. Con esta información se proyectó la cantidad total de información almacenada en un año. Luego se hicieron pruebas con información artificial. Con estas pruebas se concluyó que para 16 redes parecidas y en un periodo de tiempo cercano a 1 año, se ocuparían 5 GB de datos aproximadamente.

Por medio de 2 experimentos se evaluó la herramienta, como conclusión de la experimentación se concluyó, en parte, con buenas prácticas para la construcción de consultas para la seguridad informática.

La segmentación por tipo de protocolo resulta ser muy importante a la hora de consultar la información de flujo, la omisión de esta información puede entregar información falsa del tráfico de la red. Por ejemplo, en algunos casos puede haber mucho tráfico de paquetes para una máquina. Sin embargo, estos pueden corresponder a mensajes ICMP normales y no a tráfico malicioso entre máquinas.

En vista a los experimentos, se vio la importancia de hacer cónsulas segmentadas por fecha, hora, minutos y segundos. Esto se debe a que mientras mayor sea el tiempo en estos rangos, el tráfico malicioso puede ser confundido como tráfico normal.

Muchas veces se puede confundir el tráfico normal con tráfico malicioso si no se conoce las características de las máquinas de la red. Por ejemplo, se puede tener una máquina servidor, actuando al mismo tiempo como Gateway, servidor web y dando otra clase de servicios. Posiblemente el servidor web parezca malware, por la cantidad de tráfico 1 a N producido por el servicio. Lo que corrobora que obviar esta información puede entregar información no válida.

Hay que tener cuidado al construir consultas con respecto a la dirección del tráfico y al protocolo. Por ejemplo, para UDP no se sabe si corresponde a información de conexión o de respuesta a alguna conexión. En el caso de TCP, los flag TCP ayudan a este conocimiento.

Finalmente como conclusión de la experimentación se logró utilizar el sistema para detectar distintas clases de malware entre tráfico IP real proveniente de 2 semanas de almacenamiento.

En el tráfico producido entre máquinas de segmentos de red distintos es difícil identificar las máquinas emisoras. Esto se debe principalmente a la existencia del NAT. Se experimentó con respecto a este problema y se logró concluir principalmente que contando con un ambiente favorable, donde la red es bastante buena (sin pérdida de paquetes, poca fragmentación y tiempos de transferencia con poca variación), se cree que será posible identificar los flujos directamente. Sin embargo, las pruebas realizadas fueron sólo de tipo preliminar y muy específicas para esta situación, por lo que el problema está todavía abierto.

9 Referencias y Bibliografía

[1] Acquisition and Analysis of Large Scale Network Data, John McHugh,

http://users.cs.dal.ca/~mchugh/netanalysis/slides/

[2] Network Forensics, Ryan Connolly,Team Cymru.

http://www.apricot2007.net/presentation/tutorial/ryan-network-forensics-tut.pdf

[3] Sets, Bags, and Rock and Roll?, Analyzing| Large Data Sets of Network Data , John, McHugh.

http://www.cert.org/netsa/publications/Esorics2004-mchugh-sets.pdf

[4] Watch your Flows whit NfSen and Nfdump, Peter Haag,

http://www.ripe.net/ripe/meetings/ripe-50/presentations/ripe50-plenary-tue-nfsen-nfdump.pdf

[5] Introduction to Cisco IOS Netflow - A Technical Overview,

http://www.cisco.com/en/US/products/ps6601/products_white_paper0900aecd80406232.shtml

[6] Detecting Worms and Abnormal Activities with Netflow, Part 1,

http://www.securityfocus.com/print/infocus/1796

[7] Detecting Worms and Abnormal Activities with Netflow, Part 2,

http://www.securityfocus.com/print/infocus/1802

[8] Introducción a TCP/IP,

http://www.alcancelibre.org/staticpages/index.php/introduccion-tcp-ip/print

[9] Prolexic Distributed Denial of Service Attack Alert

http://www.prolexic.com/news/20070514-alert.php

[11] Botnets: Funcionamiento, usos y deteccin, ArCERT

http://www.arcert.gov.ar/webs/tips/botnets_200710_v1.pdf

[12] Resurge la tecnologa P2P para crear redes de BOTS

http://www.vsantivirus.com/nugache-p2p-botnet.htm

10 Glosario

Protocolo Internet (IP): Este protocolo permite la transmisión de datos a través y entre redes de área local, los datos viajan sobre una red basada en paquetes IP.

Flujo de comunicación o de paquetes IP (Netflow): Entiéndase por secuencia unidireccional de paquetes IP que comparten cinco elementos en común: Dirección IP de origen, Dirección IP destino, Número de puerto de origen, Número de puerto de destino y Protocolo.

Escaneo de puertos: Se emplea para designar la acción de analizar por medio de un programa, el estado, de los puertos de una máquina conectada a una red de comunicación.

Se utiliza para detectar que servicios comunes está ofreciendo la máquina. También puede ser usado por usuarios malintencionados que intentan comprometer la seguridad de una máquina.

Ataque de DoS (Denial of Service): Ataque a un sistema de computadores o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos.

Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema de la víctima.

Malware: Variedad de software o programas de códigos hostiles e intrusivos. Tiene como objetivo infiltrarse o dañar un computador sin el conocimiento de su usuario. Ejemplos de estos, *backdoors*, *bots*, *bugs*, troyanos, gusanos, *spyware* y otros.

IDS (Instrusion Detection System): Un Sistema de detección de intrusos es un sistema de monitoreo. Utilizado para detectar accesos desautorizados a un computador o a una red.

Estos se componen de: sensores los cuales generan eventos de seguridad, una consola para monitorear eventos, alertas y controlar los sensores, y un motor central que almacena la información proveniente de los sensores en una base de datos y utiliza un sistema de reglas para generar alertas de seguridad.

IANA (Internet Assigned Numbers Authority): Es la Agencia de Asignación de Números de Internet. Era el antiguo registro central de los protocolos Internet, como puertos, números de protocolo y empresa, opciones y códigos. Fue sustituido en 1998 por ICANN.

Gateway: Un Gateway es un equipo que permite interconectar redes con protocolos y arquitecturas completamente diferentes a todos los niveles de comunicación.

NAT (Network Address Translation): Mecanismo utilizado por routers para intercambiar paquetes entre dos redes con direcciones incompatibles. Los router's convierten en tiempo real las direcciones utilizadas en los paquetes IP.

Anexos

Anexo A: Consultas SQL para evaluaciones del capítulo 5

Experimento 1:

Comportamiento N a 1

1)

```
select
srcip,count(distinct dstip),sum(npack)
from packets_acto1
group by srcip
```

2)

```
select
dstip,count(distinct srcip)
from packets_acto1
where dstip like '192.168.2.2'
and R=0
and S=1
and A=1
group by dstip
```

3)

```
select
srcip ,dstip
from packets_acto1
where dstip like '192.168.2.2'
and R=0
and S=1
and A=1
group by srcip ,dstip,R,A,S
order by srcip
```

4)

```
select
srcip ,dstip
from packets_acto1
where dstip like '192.168.2.2'
and R=1
and S=0
and A=1
group by srcip ,dstip,R,A,S
order by srcip
```

5)

```
select
dstip,count(distinct srcip)
from packets_acto1
where dstip like '192.168.2.2'
and R=1
and S=0
and A=1
group by dstip
```

Comportamiento 1 a N

1)

```
select
dstip,count(distinct srcip)
from packets_acto2
group by dstip having count(distinct srcip)>10
```

2)

```
select
dstip,count(distinct dstport)
from packets_acto2
group by dstip having count(distinct dstport)>10
```

3)

```
select
dstip,concat(DATE(startTime),'  ',HOUR(startTime),':',MINUTE(startTime)) as fecha
,count(distinct dstport)
from packets_acto2
group by dstip ,
DATE(startTime),
HOUR(startTime),
MINUTE(startTime) having count(distinct dstport)>10
```

Comportamiento 1 a N a 1

1)

```
select
aa , count(distinct aabb), bb
from
(
  select
  a.srcip as aa ,a.dstip as aabb ,b.dstip as bb
  from
  (
    select
    *
    from packets_acto3
  )
  a join (select * from packets_acto3) b on a.dstip = b.srcip
  where a.srcip <> b.dstip
)
temp
group by aa, bb
```

2)

```
select
aa , count(distinct aabb), bb
from
(
  select
  a.srcip as aa ,a.dstip as aabb ,b.dstip as bb,a.d1,a.d2
  from
  (
    select
    srcip,dstip, DATE(startTime) as d1,HOUR(startTime) as d2
```

```
    from packets_acto3
    group by srcip,dstip, DATE(startTime),HOUR(startTime),MINUTE(startTime)
  )
  a join
  (
    select
    srcip,dstip, DATE(startTime) as d1,HOUR(startTime) as d2
    from packets_acto3
    group by srcip,dstip, DATE(startTime),HOUR(startTime),MINUTE(startTime)
  )
  b on a.dstip = b.srcip
  and a.d1=b.d1
  and a.d2= b.d2
  where a.srcip <> b.dstip
)
temp
group by aa, bb
```

Comportamiento N a 1 variando en el tiempo

1)

```
SELECT
dstip,count(distinct srcip)
FROM `packets_acto4`
group by dstip having count(distinct srcip) > 5
```

2)

```
select
srcip, count( distinct dstip) as total
from packets_acto4
where dstip in
(
  SELECT
  dstip
  FROM `packets_acto4`
  group by dstip having count(distinct srcip) > 5
)
group by srcip
```

3)

```
select
total,count(*)
from
(
  select
  srcip, count( distinct dstip) as total
  from packets_acto4
  where dstip in
  (
    SELECT
    dstip
    FROM `packets_acto4`
    group by dstip having count(distinct srcip) > 5
  )
  group by srcip
)
d
group by total
```

Experimento 2:

Simulación 1

1)

```
select
concat(date(startTime),' ', hour(startTime)) fecha ,
srcip,
count(distinct dstip)
from packets_TestAllData
where protocol = 'TCP'
group by date(startTime),
hour(startTime),
srcip having count(distinct dstip) > 13
```

a.

```
select
concat(date(startTime),' ', hour(startTime),':',minute(startTime)) fecha ,
srcip,
count(distinct dstip)
from packets_TestAllData
where protocol = 'TCP'
group by date(startTime),
hour(startTime),
minute(startTime) ,
srcip having count(distinct dstip) > 5
```

b.

```
Select
concat(date(startTime),'  ',  hour(startTime)  ,':',  minute(startTime)  ,':',second(startTime))
fecha ,
srcip,
count(distinct dstip)
from packets_TestAllData
where protocol = 'TCP'
group by date(startTime),
hour(startTime),
minute(startTime),
second(startTime) ,
srcip having count(distinct dstip) > 2
```

c.

```
select
concat(date(startTime),'  ',  hour(startTime)  ,':',minute(startTime)  ,':',second(startTime))
fecha ,
srcip,
count(distinct dstip)
from packets_TestAllData
where S=1
and R=0
and A=0
and protocol = 'TCP'
group by date(startTime),
hour(startTime),
minute(startTime),
second(startTime) ,
srcip having count(distinct dstip) > 2 fecha srcip count(distinct dstip)
```

d.

```
select
concat(date(startTime),' ', hour(startTime)) fecha ,
srcip,
count(distinct dstip)
from packets_TestAllData
where S=0
and R=1
and A=1
and protocol = 'TCP'
group by date(startTime), hour(startTime),srcip having count(distinct dstip) > 2
```

2)

```
select
concat(date(startTime),' ', hour(startTime)) fecha ,
dstip,
count(distinct srcip)
from packets_TestAllData
where protocol = 'TCP'
group by date(startTime),
hour(startTime),
dstip having count(distinct srcip) > 6
```

a.

```
select
concat(date(startTime),' ', hour(startTime),':',minute(startTime)) fecha ,
dstip,
count(distinct srcip)
from packets_TestAllData
where protocol = 'TCP'
group by date(startTime),
hour(startTime),
minute(startTime) ,
dstip having count(distinct srcip) > 2 fecha dstip count(distinct srcip)
```

b.

```
select
concat(date(startTime),'     ',     hour(startTime),':',minute(startTime),':',second(startTime))
fecha ,
dstip,
count(distinct srcip)
from packets_TestAllData
where protocol = 'TCP'
group by date(startTime),
hour(startTime),
minute(startTime),
second(startTime) ,
dstip having count(distinct srcip) > 2
```

c.

```
select
date(startTime),dstip,count(distinct srcip)
from packets_TestAllData
where protocol = 'TCP'
and R=1
and A=1
group by date(startTime),dstip having count(distinct srcip) >2
```

d.

```
select
concat(date(startTime),' ', hour(startTime),':',minute(startTime)) fecha ,
dstip,
count(distinct srcip)
from packets_TestAllData
where protocol = 'TCP'
and R=1
and A=1
and S=0
group by date(startTime),
hour(startTime),
minute(startTime),
dstip having count(distinct srcip) >1
```

Simulación 2

1)

```
select
date,hour,aa,bb,count(distinct aabb)
from
(
  select
  a.date ,a.hour, a.srcip as aa ,a.dstip as aabb ,b.dstip as bb
  from
  (
    select
    srcip,dstip, date(startTime) as date , hour(startTime) as hour
    from packets_TestAllData
    group by date(startTime) , hour(startTime),srcip,dstip
  )
  a join
  (
    select
    srcip,dstip ,date(startTime) as date, hour(startTime) as hour
    from packets_TestAllData
    group by date(startTime) , hour(startTime),srcip,dstip
  )
  b on a.dstip = b.srcip
  where a.date=b.date
  and a.hour = b.hour
  and a.srcip <> b.dstip
)
d
group by date , hour, aa, bb having count(distinct aabb) >10
```

Extensiones: Análisis de flujo en redes con NAT:

Heuristica 1:

```
select
srcA,dstA,srcB,dstB,sum(npack),sum(nbyte)
from
(
  select
  concat(a.srcip,':',a.srcport) as srcA,
  concat(a.dstip,':',a.dstport) as dstA,
  concat(b.srcip,':',b.srcport) srcB,
  concat(b.dstip,':',b.dstport) as dstB,
  a.startTime stA,
  b.startTime stB
  from
  (
    select
    *
    from packets_colector12345mb
  )
  a join
  (
    select
    *
    from packets_colector12348mb
  )
  b using(npack,nbyte,dstport,A,S,R,F)
  where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
  order by 1,2,3,4
)
a
group by srcA,dstA,srcB,dstB
```

Heuristica 2:

```
select
srcA,dstA,srcB,dstB,sum(npack),sum(nbyte)
from
(
  select
  concat(a.srcip,':',a.srcport) as srcA,
  concat(a.dstip,':',a.dstport) as dstA,
  concat(b.srcip,':',b.srcport) srcB,
```

```
  concat(b.dstip,':',b.dstport) as dstB,
  a.startTime stA,
  b.startTime stB
  from
  (
    select
    *
    from packets_colector12345mb
  )
  a join
  (
    select
    *
    from packets_colector12348mb
  )
  b using(npack,dstport,A,S,R,F)
  where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
  order by 1,2,3,4
)
a
group by srcA,dstA,srcB,dstB
```

Heuristica 3:

```
select
srcA,dstA,srcB,dstB,sum(npack),sum(nbyte)
from
(
  select
  concat(a.srcip,':',a.srcport) as srcA,
  concat(a.dstip,':',a.dstport) as dstA,
  concat(b.srcip,':',b.srcport) srcB,
  concat(b.dstip,':',b.dstport) as dstB,
  a.startTime stA,
  b.startTime stB
  from
  (
    select
    *
    from packets_colector12345mb
  )
  a join
  (
    select
    *
```

```
    from packets_colector12348mb
  )
  b using(dstport,A,S,R,F)
  where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
  and (a.nbyte/a.npack) = (b.nbyte/b.npack)
  order by 1,2,3,4
)
a
group by srcA,dstA,srcB,dstB
```

Heuristica 4:

```
select
srcA,
dstA,
srcB,
dstB
from
(
  select
  concat(a.srcip,':',a.srcport) as srcA,
  concat(a.dstip,':',a.dstport) as dstA,
  concat(b.srcip,':',b.srcport) srcB,
  concat(b.dstip,':',b.dstport) as dstB,
  a.startTime stA,
  b.startTime stB ,
  a.npack npackA,
  b.nbyte nbyteB,
  b.npack npackB ,
  a.nbyte nbyteA
  from
  (
    select
    *
    from packets_colector12345mb
    where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
  )
  a join
  (
    select
    *
    from packets_colector12348mb
    where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
```

```
  )
  b using(dstport,A,S,R,F)
  where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
  order by 1,2,3,4
)
a
group by srcA,dstA,srcB,dstB
```

Heuristica 5:

```
select
srcA,
dstA,
srcB,
dstB
from
(
  select
  concat(a.srcip,':',a.srcport) as srcA,
  concat(a.dstip,':',a.dstport) as dstA,
  concat(b.srcip,':',b.srcport) srcB,
  concat(b.dstip,':',b.dstport) as dstB,
  a.startTime stA,
  b.startTime stB ,
  a.npack npackA,
  b.nbyte nbyteB,
  b.npack npackB ,
  a.nbyte nbyteA
  from
  (
    select
    *
    from packets_colector12345mb
    where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
  )
  a join
  (
    select
    *
    from packets_colector12348mb
    where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
  )
  b using(A,S,R,F)
  where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 17
```

```
  and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) > 13
  and (a.nbyte/a.npack) = (b.nbyte/b.npack)
  order by 1,2,3,4
)
a
group by srcA,dstA,srcB,dstB
```

Heuristica 6:

```
select
dstA,srcB,count(*)
from
(
  (
    select
    srcA,
    dstA,
    srcB,
    dstB,
    ((sum(npackA) + sum(npackB))/2) avgnpack,
    ((sum(nbyteA) + sum(nbyteB))/2) avgnbyte
    from
    (
      select
      concat(a.srcip,':',a.srcport) as srcA,
      concat(a.dstip,':',a.dstport) as dstA,
      concat(b.srcip,':',b.srcport) srcB,
      concat(b.dstip,':',b.dstport) as dstB,
      a.startTime stA,
      b.startTime stB ,
      a.npack npackA,
      b.nbyte nbyteB,
      b.npack npackB ,
      a.nbyte nbyteA
      from
      (
        select
        *
        from packets_colector12345mb
        where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
      )
      a join
      (
        select
        *
```

```
      from packets_colector12348mb
      where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
    )
    b using(A,S,R,F)
    where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 17
    and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) > 13
    and (a.nbyte/a.npack) = (b.nbyte/b.npack)
    order by 1,2,3,4
  )
  a
  group by srcA,dstA,srcB,dstB
)
union
(
  select
  srcA,dstA,srcB,dstB,sum(npack) as avgnpack,sum(nbyte) as avgnbyte
  from
  (
    select
    concat(a.srcip,':',a.srcport) as srcA,
    concat(a.dstip,':',a.dstport) as dstA,
    concat(b.srcip,':',b.srcport) srcB,
    concat(b.dstip,':',b.dstport) as dstB,
    a.startTime stA,
    b.startTime stB ,
    a.npack ,
    b.nbyte
    from
    (
      select
      *
      from packets_colector12345mb
    )
    a join
    (
      select
      *
      from packets_colector12348mb
    )
    b using(npack,dstport,A,S,R,F)
    where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
    and (a.nbyte/a.npack) = (b.nbyte/b.npack)
    order by 1,2,3,4
  )
  a
  group by srcA,dstA,srcB,dstB
```

```
)
union
(
  select
  srcA,
  dstA,
  srcB,
  dstB,
  ((sum(npackA) + sum(npackB))/2) avgnpack,
  ((sum(nbyteA) + sum(nbyteB))/2) avgnbyte
  from
  (
    select
    concat(a.srcip,':',a.srcport) as srcA,
    concat(a.dstip,':',a.dstport) as dstA,
    concat(b.srcip,':',b.srcport) srcB,
    concat(b.dstip,':',b.dstport) as dstB,
    a.startTime stA,
    b.startTime stB ,
    a.npack npackA,
    b.nbyte nbyteB,
    b.npack npackB ,
    a.nbyte nbyteA
    from
    (
      select
      *
      from packets_colector12345mb
      where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
    )
    a join
    (
      select
      *
      from packets_colector12348mb
      where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
    )
    b using(dstport,A,S,R,F)
    where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
    order by 1,2,3,4
  )
  a
  group by srcA,dstA,srcB,dstB
)
union
(
```

```
    select
    srcA,
    dstA,
    srcB,
    dstB,
    ((sum(npackA) + sum(npackB))/2) avgnpack,
    ((sum(nbyteA) + sum(nbyteB))/2) avgnbyte
    from
    (
      select
      concat(a.srcip,':',a.srcport) as srcA,
      concat(a.dstip,':',a.dstport) as dstA,
      concat(b.srcip,':',b.srcport) srcB,
      concat(b.dstip,':',b.dstport) as dstB,
      a.startTime stA,
      b.startTime stB ,
      a.npack npackA,
      b.nbyte nbyteB,
      b.npack npackB ,
      a.nbyte nbyteA
      from
      (
        select
        *
        from packets_colector12345mb
        where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
      )
      a join
      (
        select
        *
        from packets_colector12348mb
        where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
      )
      b using(A,S,R,F)
      where abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 17
      and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) > 13
      and (a.nbyte/a.npack) = (b.nbyte/b.npack)
      order by 1,2,3,4
    )
    a
    group by srcA,dstA,srcB,dstB
  )
)
aux
group by dstA,srcB having count(*)>1
```

Heuristica 7:

```
select
srcA,
dstA,
srcB,
dstB,
((sum(npackA) + sum(npackB))/2) avgnpack,
((sum(nbyteA) + sum(nbyteB))/2) avgnbyte
from
(
  select
  concat(a.srcip,':',a.srcport) as srcA,
  concat(a.dstip,':',a.dstport) as dstA,
  concat(b.srcip,':',b.srcport) srcB,
  concat(b.dstip,':',b.dstport) as dstB,
  a.startTime stA,
  b.startTime stB ,
  a.npack npackA,
  b.nbyte nbyteB,
  b.npack npackB ,
  a.nbyte nbyteA
  from
  (
    select
    *
    from packets_total
    where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
  )
  a join
  (
    select
    *
    from packets_total
    where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
  )
  b using(A,S,R,F)
  where a.netflow_router_id <> b.netflow_router_id
  and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 17
  and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) > 13
  and (a.nbyte/a.npack) = (b.nbyte/b.npack)
  order by 1,2,3,4
)
a
group by srcA,dstA,srcB,dstB
```

Heuristica 8:

```
select
dstA,srcB,count(*)
from
(
  (
    select
    srcA,
    dstA,
    srcB,
    dstB,
    ((sum(npackA) + sum(npackB))/2) avgnpack,
    ((sum(nbyteA) + sum(nbyteB))/2) avgnbyte
    from
    (
      select
      concat(a.srcip,':',a.srcport) as srcA,
      concat(a.dstip,':',a.dstport) as dstA,
      concat(b.srcip,':',b.srcport) srcB,
      concat(b.dstip,':',b.dstport) as dstB,
      a.startTime stA,
      b.startTime stB ,
      a.npack npackA,
      b.nbyte nbyteB,
      b.npack npackB ,
      a.nbyte nbyteA
      from
      (
        select
        *
        from packets_total
        where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
      )
      a join
      (
        select
        *
        from packets_total
        where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
      )
      b using(A,S,R,F)
      where a.netflow_router_id <> b.netflow_router_id
      and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 17
      and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) > 13
      and (a.nbyte/a.npack) = (b.nbyte/b.npack)
```

```
    order by 1,2,3,4
  )
  a
  group by srcA,dstA,srcB,dstB
)
union
(
  select
  srcA,dstA,srcB,dstB,sum(npack) as avgnpack,sum(nbyte) as avgnbyte
  from
  (
    select
    concat(a.srcip,':',a.srcport) as srcA,
    concat(a.dstip,':',a.dstport) as dstA,
    concat(b.srcip,':',b.srcport) srcB,
    concat(b.dstip,':',b.dstport) as dstB,
    a.startTime stA,
    b.startTime stB ,
    a.npack ,
    b.nbyte
    from
    (
      select
      *
      from packets_total
    )
    a join (select * from packets_total ) b using(npack,dstport,A,S,R,F)
    where a.netflow_router_id <> b.netflow_router_id
    and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
    and (a.nbyte/a.npack) = (b.nbyte/b.npack)
    order by 1,2,3,4
  )
  a
  group by srcA,dstA,srcB,dstB
)
union
(
  select
  srcA,
  dstA,
  srcB,
  dstB,
  ((sum(npackA) + sum(npackB))/2) avgnpack,
  ((sum(nbyteA) + sum(nbyteB))/2) avgnbyte
  from
  (
```

```
    select
    concat(a.srcip,':',a.srcport) as srcA,
    concat(a.dstip,':',a.dstport) as dstA,
    concat(b.srcip,':',b.srcport) srcB,
    concat(b.dstip,':',b.dstport) as dstB,
    a.startTime stA,
    b.startTime stB ,
    a.npack npackA,
    b.nbyte nbyteB,
    b.npack npackB ,
    a.nbyte nbyteA
    from
    (
      select
      *
      from packets_total
      where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
    )
    a join
    (
      select
      *
      from packets_total
      where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
    )
    b using(dstport,A,S,R,F)
    where a.netflow_router_id <> b.netflow_router_id
    and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 20
    order by 1,2,3,4
  )
  a
  group by srcA,dstA,srcB,dstB
)
union
(
  select
  srcA,
  dstA,
  srcB,
  dstB,
  ((sum(npackA) + sum(npackB))/2) avgnpack,
  ((sum(nbyteA) + sum(nbyteB))/2) avgnbyte
  from
  (
    select
    concat(a.srcip,':',a.srcport) as srcA,
```

```
      concat(a.dstip,':',a.dstport) as dstA,
      concat(b.srcip,':',b.srcport) srcB,
      concat(b.dstip,':',b.dstport) as dstB,
      a.startTime stA,
      b.startTime stB ,
      a.npack npackA,
      b.nbyte nbyteB,
      b.npack npackB ,
      a.nbyte nbyteA
      from
      (
        select
        *
        from packets_total
        where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
      )
      a join
      (
        select
        *
        from packets_total
        where SUBSTRING_INDEX(srcip,'.',3) <> SUBSTRING_INDEX(dstip,'.',3)
      )
      b using(A,S,R,F)
      where a.netflow_router_id <> b.netflow_router_id
      and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) < 17
      and abs(time_to_sec(a.startTime) - time_to_sec(b.startTime)) > 13
      and (a.nbyte/a.npack) = (b.nbyte/b.npack)
      order by 1,2,3,4
    )
    a
    group by srcA,dstA,srcB,dstB
  )
)
aux
group by dstA,srcB having count(*)>1
```