UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

IMPLEMENTACIÓN DE UN DETECTOR DE MALWARE CON SOFTWARE LIBRE EMPLEANDO EL PROTOCOLO NETFLOW

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACIÓN

PRESENTA:
ALDO IVAN GIRÓN CAPISTRÁN

Director de Tesis: Ing. José de Jesús Ramírez Pichardo

México, D.F Mayo 2011

Agradecimientos.

Juan,

Gracias por los valores inculcados papá, por la educación y apoyo incondicional proporcionado, por ser tan estricto y obligarme a superarme constantemente.

Por todo esto y demás he logrado ser la persona que soy ahora, gracias.

Margarita,

Gracias por ser como eres, por el amor incondicional que me has brindado, por el apoyo ofrecido en los buenos y malos momentos, por todos los retos que hemos enfrentado y superado juntos te quiero mamá.

Tíos y abuelos.

Por estar unidos en los buenos y malos momentos, por el apoyo ofrecido y por ponerme como ejemplo a seguir de mis primos, gracias.

Primos,

Por todas las experiencias, buenos y malos momentos que hemos compartidos juntos, por la solidaridad y apoyo mostrados, por todos los días de diversión que me han brindado, por ser como son, gracias.

Amelia.

Por haberme brindado tanto cariño y amor, por cuidarme desde el cielo, te llevo en mi corazón… ¡lo logré abuela! Gracias.

Ing. Jesús Pichardo,

Por todo el apoyo y la paciencia proporcionado en el desarrollo de este proyecto de tesis, por ser mi asesor, por tu amistad, gracias.

Ing. Hugo Bonilla

Por todo el apoyo y conocimientos que adquirí por parte tuya en mi estancia de becario en Banxico, por asesorarme en el desarrollo de esta tesis, por tu sincera amistad, gracias.

Banxico,

Por permitirme realizar mi servicio social e implementar mi proyecto en esta institución. Gracias al Ing. Humberto Brito por confiar en mí. A mis compañeros y amigos de la oficina de red interna por todo el apoyo proporcionado. A mis amigos becarios, excelentes momentos que pasamos juntos, gracias.

Facultad de Ingeniería, UNAM,

Por proporcionarme los conocimientos necesarios para ejercer dignamente esta hermosa carrera y por formar parte de la máxima casa de estudios, gracias a todos aquellos profesores que me exigieron al máximo y me demostraron de que estoy hecho. Orgullosamente UNAM,

Amigos

Por brindarme su amistad, por todos los buenos momentos que hemos pasado en diversas etapas de mi vida, por los consejos y buenos deseos, por no olvidarse de mí y por todo lo que nos falta recorrer juntos, gracias.

120'S

Gracias por permitirme formar parte de este excelente grupo, por mantenernos unidos desde el principio y hasta el final de la carrera. Gracias amigos por brindarme tantos momentos de alegría, de suspenso y de superación. Por brindarme apoyo cuando más lo necesité, ¡mi estancia en la Facultad no hubiera sido la misma sin ustedes!

Ricardo

Gracias por brindarme tu sincera amistad, por todo el tiempo que hemos convivido juntos, por todos los momentos buenos, malos y chuscos que hemos pasado, por tu apoyo incondicional, por preocuparte por mí. Te considero como mi hermano, ¡gracias por ser como eres!

Y recuerda esto: Οι φίλοι είναι σαν τα αστέρια, αλλά δεν είχε γνωρίσει ποτέ, ξέρεις ότι είναι εκεί.

Daniel y Victor

Compañeros de mi ciudad natal Netzahualcóyotl. Gracias por su apoyo, su sincera amistad, por la fortaleza y solidaridad que mostramos a lo largo de la carrera, por superar cada obstáculo que se nos puso enfrente, por todos los buenos momentos que compartimos juntos. Lo logramos amigos, gracias.

Índice

Introducción

Objetivo.
Implementar un detector de malware con software libre empleando el protocolo Netflow.

Descripción del problema.
Generalmente las herramientas de seguridad como los antivirus, firewalls, *IDS (Intrusion Detection System)* e *IPS (Intrusion Prevention System)*, operan en capas superiores del modelo OSI, y muy pocas de ellas operan en capas inferiores. Esto representa un problema cuando tratamos de detectar la actividad de un malware o código malicioso (Proviene del término en inglés "**mal**icious soft**ware":** Todo aquel software que perjudica a una computadora) que sea capaz de infectar a sistemas de cómputo mediante diversos ataques a capas inferiores del modelo OSI, como puede ser: *IP flooding*, *IP spoofing*, *Ping of Death*, *Arp spoofing*, entre otros, o explotando vulnerabilidades presentes en el protocolo TCP/IP.

A pesar de que las herramientas de seguridad han mejorado enormemente sus métodos de detección de malware, muy pocas son capaces de detectar algún malware de "día cero" (nuevo malware creado y que no se tiene ningún registro acerca de él), dado que para este tipo de malware no se tiene conocimiento sobre su comportamiento y no se tiene referencia alguna de él.

Una vez que el malware ha logrado infectar a un sistema de cómputo, buscará propagarse lo más rápido posible dentro de la red, comenzando con un escaneo de puertos o un escaneo de IP's en búsqueda de más "víctimas". Dependiendo del tipo de malware, éste podría enviar información confidencial hacia IP's desconocidas, o realizar un ataque de negación de servicio en conjunto con otras sistemas de cómputo infectados con el objetivo de saturar un servidor especifico, entre otras acciones.

Por todo lo mencionado acerca del malware y la capacidad de propagación y *polimorfismo* que presenta, es necesario investigar técnicas innovadoras que sean capaces de detectar malware. Una solución viable es la detección de malware mediante el monitoreo de red.

El monitoreo de red es una actividad que comúnmente desarrollan los administradores de redes de las empresas. Debido a que permite observar el comportamiento de la red en tiempo real. Específicamente:

- Se observa la cantidad de ancho de banda consumido.
- El porcentaje de utilización de protocolos *TCP, UDP*, *ICMP* entre otros.
- Utilización de servicios como HTTP, SSH, FTP, SNMP, SMTP, entre otros.
- La actividad presente en las redes de usuarios y redes de servidores.
- Comportamiento anormal de la red.

La siguiente tabla muestra los métodos más comunes utilizados en el monitoreo de red.

Tabla I.1 Diversas técnicas utilizadas para monitorear la red.

Método	Breve Descripción
Activar la tarjeta de red en modo promiscuo	La tarjeta de red captura todo el tráfico que circula a través de ella.
SNMP	Protocolo de la capa de aplicación del modelo OSI que facilita el intercambio de información de administración entre dispositivos de red
Netflow	Captura el tráfico directamente de un router o switch que soporte esta tecnología.

Sin embargo se tiene el inconveniente de que no existe un estándar enfocado en el monitoreo de red y que sea aplicable a todas las herramientas enfocadas en esta tecnología. El tener un estándar facilitaría la comunicación entre las diversas técnicas utilizadas para monitorear redes.

Para tratar de resolver este problema CISCO ha desarrollado un protocolo llamado Netflow enfocado en el monitoreo de red.

Como principales características de Netflow son:

- ✓ Estandarizado RFC 3954 (Cisco Systems Netflow Services Export Version 9).
- ✓ Actualmente tiene 10 versiones (Netflow V1-V9 y V10 IPFIX).
- ✓ Obtiene los datos directamente del router o switch, a diferencia de otras tecnologías de monitoreo.
- ✓ Permite contabilizar el ancho de banda consumido.
- ✓ Provee una visión detallada acerca del comportamiento de la red.
- ✓ Se logra tener un alcance del 100%, debido a que se puede activar en cada router o switch presente.

Sin embargo, la principal desventaja del protocolo Netflow es que el software propietario o comercial tiene un gran costo en ambientes de producción. Una alternativa a este inconveniente es utilizar software libre u open source (software distribuido y desarrollado libremente) que soporte completamente dicho protocolo.

Solución.
El problema expuesto en la sección “descripción del problema”, se puede resolver de las siguientes formas:

1. Mejorando el proceso de hardening en la institución. (Diversas herramientas de seguridad que trabajan en conjunto para proteger los bienes o activos).
2. Por medio de la compra de un software propietario o comercial Netflow enfocado en la detección de malware. Ejemplo: Arbok Netflow.
3. Por medio de la investigación sobre una alternativa software libre que soporte el protocolo Netflow y sobre ella implementar un detector de malware.

De las cuales las propuestas 1 y 3 son más viables, debido a que se investigará sobre alguna alternativa software libre basada en el protocolo Netflow y sobre ella se implementará un detector de malware, mientras que el aplicar la propuesta 2 implica pagar por la licencia y el uso del nuevo software.

La siguiente tabla muestra las ventajas y desventajas del proyecto propuesto.

Tabla I.2 Ventajas y desventajas del proyecto de tesis

Ventajas	Desventajas
✓ Propuesta de una solución innovadora en la detección de malware. ✓ Nueva herramienta enfocada en el monitoreo de red y en la detección de malware. ✓ Utilización del protocolo Netflow ✓ Utilización de software Libre ✓ Costos mínimos en su utilización. ✓ Posibles modificaciones hacia el nuevo software. ✓ Posible detección de malware de "día cero". ✓ La nueva herramienta de seguridad desarrollada será implementada en una institución importante en nuestro país.	× Escasas alternativas de software libre que soporten el protocolo Netflow. × Escasa información acerca del comportamiento del malware. × Existe software propietario o comercial muy robusto que soporta el protocolo Netflow. × Solo podrá detectar malware que deje evidencia en la red. × Capacitación a usuarios que no conozcan sistemas operativos Linux.

La nueva herramienta trabajará en conjunto con las demás herramientas de seguridad que se tienen implementadas en la institución, fortaleciendo el esquema de seguridad implementado en la institución.

Justificación

Por todo lo mencionado el presente trabajo tiene como objetivo investigar una alternativa de software libre que soporte el protocolo Netflow e implementar un detector capaz de encontrar malware en la red.

Con el éxito del presente proyecto de tesis, propondré una nueva herramienta de seguridad que cumpla con los siguientes objetivos:

- Implementado con software libre.
- Monitoreo de red basado en el protocolo Netflow.
- Detección de malware basado en patrones de comportamiento anómalo y posible detección de ataques de "día cero".
- Enfocada hacia capas inferiores del modelo OSI.
- Compatibilidad total con otras herramientas de seguridad.

Las restricciones encontradas en este proyecto es la escasa información acerca del comportamiento del malware, además de encontrar muy pocas alternativas de software libre que soporten en protocolo Netflow.

El proyecto de tesis se pretende implementar en una institución, por lo tanto no se podrán mostrar datos referentes a la institución donde será implementado, por cuestiones de integridad y confidencialidad de la información presente en la institución.

Método

Para la resolución del proyecto propuesto de tesis se aplicó la siguiente metodología:

1. Investigación acerca del funcionamiento del protocolo Netflow.
2. Investigación acerca de alternativas software libre que trabajen sobre el protocolo Netflow y en ellas se pueda implementar un detector de malware.
3. Elección de la mejor alternativa, con base en un análisis detallado realizado y puesta en marcha.
4. Investigación acerca del comportamiento del malware.
5. Creación del detector de malware.
6. Pruebas.
7. Puesta en marcha de la nueva herramienta sobre un ambiente de producción en una institución de alta importancia en el país.
8. Creación de toda la documentación requerida.

Estructura del proyecto de tesis.

El proyecto de tesis se ha dividido en cinco capítulos y cuatro anexos de la siguiente forma:

- El capítulo uno tiene el objetivo de proporcionar toda la información teórica requerida en la realización del proyecto tesis.
- El capítulo dos tiene el objetivo de explicar todo el proceso realizado en la elección del software libre que se implementó en la institución. Además de mostrar una comparación entre el software libre elegido con el software propietario utilizado en la institución.
- El capítulo tres tiene el objetivo de explicar la implementación realizada del protocolo Netflow en la institución y el funcionamiento del software elegido en el capítulo dos.
- El capítulo cuatro tiene el objetivo de explicar las diversas técnicas utilizadas para la detección del malware. Además del desarrollo y funcionamiento del plugin creado.
- El capítulo cinco tiene el objetivo de mostrar el correcto funcionamiento del software “Listry-AIGC”, software que incluye a la alternativa elegida en el capítulo dos, además de diversas funcionalidades añadidas a él; enfocado sobre el monitoreo de red y la detección de malware mediante el plugin explicado en el capítulo cuatro. Además de mostrar las conclusiones obtenidas en la realización del proyecto de tesis.
- El anexo A tiene un glosario creado.
- El anexo B explica la instalación del software “Listry-AIGC”.
- El anexo C se ha creado como una guía de usuario del software “Listry-AIGC”.
- El anexo D muestra el código del plugin creado y explicado en el capítulo cuatro.

Capítulo I

Marco Teórico

1.1 Software Libre

1.1.1 Historia, definición y características.

"El software libre es una cuestión de libertad, no de precio. Para entender el concepto, debería pensar en libre como en libre expresión, no como en barra libre" (Definición de Richard Stallman sobre el software libre)

El termino software libre nació en el año de 1988. Tiene como principal característica el permitir a cualquier usuario acceder a su código fuente sin pagar por su uso, a diferencia del software privativo en donde no se posible observar su código fuente y se tiene que pagar por su uso.

A principios de la década de los 70's y 80's comenzaron a desarrollarse notables proyectos como son SPICE y Tex, ambos enfocados a mantener la filosofía que poco a poco iba perdiendo fuerza, el primero desarrollado en 1973 en la Universidad de California por Donald Pederson estaba enfocado a la simulación de circuitos electrónicos y el segundo desarrollado por Donald Knuth en 1978 el cual es un sistema de escritura cuyo objetivo es producir documentos con un formato de calidad, además de uno de los más importantes en materia de software libre UNIX.

UNIX creado por Thompson y Ritchie desde 1972 e impulsado por los laboratorios Bellde AT&T, fue un pilar fundamental para el software libre, ya que éste fue desarrollado inicialmente bajo los términos de licencia que permitían su libre distribución,modificación y estudio, éste tuvo su mayor impulso en la Universidad de California en Berkeley quien posteriormente por problemas de licenciamiento y falta de acceso al mismo fue encareciendo el proyecto hasta llegar al grado en que dicha institución seguía un proceso legal con la división Unix System Laboratories de AT&T por publicar el código de este *sistema operativo*, lo que ocasionó que se perdieran los términos de distribución de versiones que actualmente están establecidos en la filosofía del software libre.

Otro principal fundador de la filosofía de software libre es Richard Stallman. Quien por problemas presentados con una impresora HP, y al no poder acceder a su código, lo motivó a desarrollar controladores de uso libre y distribuirlos sin ningún costo a usuarios que presentaran los mismos inconvenientes. Esta ideología fue rápidamente adoptada por personas que tenían los mismos problemas con software propietario, naciendo así la comunidad de software libre.

Los grupos de software libre comenzaron desarrollando controladores para hardware que presentaban problemas similares al presentado por Richard Stallman. Con el paso del tiempo se fueron creando alternativas de uso libre hacia el software propietario. En la actualidad se ha logrado avanzar a tal grado que existen sistemas operativos trabajando completamente con software libre: como es el caso de distribuciones Linux basadas en *Debian* o *Red Hat*, entre otras.

La idea del software libre radica en que al momento de liberar el código, se pretende que se realicen mejoras sobre este software a tal grado que logre igualar o inclusive superar en calidad al software propietario.

Todo software que sea considerado como software libre, deberá cumplir con las siguientes características:

- Libre redistribución.El poseedor del software tiene la libertad de venderlo o regalarlo. Sin embargo, existen licencias que obligan a que la distribución sea de forma gratuita.
- El código fuente podrá ser observado de manera libre sin tener que pagar por él o en su defecto se deberá obtener libremente.
- El poder realizar mejoras sobre dicho software; siempre y cuando se respete el copyright de algunas licencias.
- La integridad del código fuente del autor. Se debe respetar al autor de dicho software; los cambios realizados por terceras personas deberán ser publicados como parches o actualizaciones "las cuales pueden ser nuevas mejoras o nuevas funcionalidades" y no como un nuevo software libre.
- Toda persona que trabaje sobre alguna modificación deberá ser incluida.
- Todo software libre debe estar regido por alguna licencia; sin embargo esta licencia deberá ser neutral y no deberá obligar a que alguna versión o parche más reciente se deba distribuir sobre esta misma licencia, ni el volverlo software privativo.

1.1.2 GNU/Linux.

El proyecto GNU/Linux fue iniciado en 1983 por Richard Stallman. Esta filosofía tiene como principal objetivo el crear software libre con el fin de tener por lo menos alguna alternativa de uso libre hacia aplicaciones propietarias.

Fue tal el crecimiento de este proyecto que en cuestión de años se unieron cientos de personas. Logrando en 1991 la liberación del proyecto GNU/Linux cuyo objetivo radicó en la creación de un sistema operativo basado en su totalidad por software libre, bajo un núcleo Unix. Con la liberación de dicho proyecto se logró también el nacimiento del kernel Linux. En este kernen se han desarrollado diversos sistemas operativos, como lo son Red Hat, Centos, Fedora, Debian y Ubuntu entre otros.

Tal ha sido el éxito de la filosofía GNU/Linux que hoy en día es uno de los principales competidores de los sistemas operativos Windows y Unix. Teniendo un impacto a tal grado que cada día se unen más personas a esta filosofía.

Todo software que sea considerado como Software Libre, sin importar el kernel que utilice, debe cumplir con las siguientes libertades.

1.1.3 Libertades del Software Libre.

- **Libertad 0:** Libertad de ejecutar el programa con cualquier propósito.
- **Libertad 1:** Libertad de tener acceso al código fuente del software, poder estudiarlo y modificarlo para obtener el objetivo deseado.
- **Libertad 2:** Libertad de redistribución del software para ayudar al prójimo.
- **Libertad 3:** Libertad de redistribuir versiones modificadas a terceros, con el objetivo de realizar modificaciones y ayudar al mejor funcionamiento del software.

Todas las libertades mencionadas deberán cumplirse. Si por algún motivo se ignora alguna de ellas, el software no se considerará libre.

Aunque el software libre ha adquirido bastante popularidad en la actualidad, existen demasiadas trabas puestas por empresas, especialmente Microsoft, que evitan que siga creciendo esta ideología. En la siguiente sección se realiza una comparación entre el software propietario y el software libre.

1.1.4 Software propietario vs software libre

Como se ha mencionado, el principal objetivo del software libre es tener alguna alternativa de uso libre contra cada software propietario que exista. Sin embargo, en muchos casos las aplicaciones desarrolladas bajo software con licenciamiento de uso son mejores que las aplicaciones desarrolladas sobre la filosofía Open Source: el problema radica cuando los desarrolladores de software libre trabajan en desarrollar alguna alternativa de uso libre hacia el software propietario y no conocen, o pueden observar el código del software propietario. Este inconveniente causa dificultades en el desarrollo e implementación del nuevo software de uso libre.

Sin embargo, cada vez se logran desarrollar mayores aplicaciones de software libre que compiten a la par con el software propietario. Un ejemplo son los sistemas operativos (S.O.) Ubuntu y Fedora: en cada actualización disponible de sus versiones, los creadores logran volver a estos S.O. más amigables hacia el usuario final, ofreciendo software libre de fácil utilización y muy potente. Logrando que más personas prueben estos sistemas operativos y decidan instalarlos.

En contramedida, el software propietario ofrece una mejor calidad: las empresas desarrolladoras de software propietario se comprometen a dar mantenimiento y garantizar que el software funciona correctamente. Sin embargo, el software propietario es objeto de muchos ataques realizados por *perpetradores*. Como ejemplo, la cantidad de malware y ataques desarrollados a los S.O. Windows y software desarrollado en esta plataforma. Mientras que en los S.O. Linux prácticamente no se presenta este problema.

Otro inconveniente que presenta el software propietario es la piratería: la mayoría del software propietario es de alto costo, y no poder observar su código fuente motiva a los perpetradores a desarrollar aplicaciones con el objetivo de *crakear* o parchar el software propietario, logrando el uso de éste de forma ilegal.

La siguiente tabla muestra una comparación realizada sobre el software propietario y el software libre.

Tabla 1.1 Software Propietario vs software libre

	Ventajas	Desventajas
Software Propietario	✓ Soporte proporcionado por el fabricante. ✓ Actualizaciones. ✓ Funcionamiento deseado. ✓ Se desarrollan aplicaciones muy complejas. ✓ Robusto. ✓ Página web del proveedor. ✓ Guía de usuario. ✓ Fácil instalación.	× Licencia de uso (En algunos casos es muy cara). × Susceptible a virus. × Tener que pagar nuevas licencias para actualizaciones. × En la mayoría de los casos solo se desarrolla sobre los sistemas operativos más comerciales. × No se puede observar el código fuente. × No se pueden realizar cambios. × No se puede redistribuir, a excepción de comprar licencias que soporten múltiples equipos.
Software Libre	✓ Gratuito. ✓ Se mejora constantemente. ✓ Libertad de distribución. ✓ Observar el código fuente. ✓ Realizar cambios de acuerdo a las necesidades requeridas. ✓ Multiplataforma. ✓ Generalmente está libre de virus. ✓ La licencia impide que se vuelva privativos. ✓ Mejoras desarrolladas por terceras personas respetando los derechos de autor.	× En algunos casos es difícil su instalación. × No se tiene soporte en algunas aplicaciones desarrolladas × En algunos casos no funcionan correctamente. × No siempre se tienen alternativas software libre que logren sustituir completamente a algún software propietario. × No se tienen actualizaciones constantes del software. × En algunos casos se tiene una página web del software obsoleta.

Además de lo descrito del software libre, se tiene el inconveniente que los desarrolladores al utilizar software libre crean una nueva aplicación y la traten de volver propietaria. Por este motivo existen diversas licencias que protegen al software libre contra cualquier tipo de privatización y garantizan el respeto de los derechos de autor (copyleft).

1.1.5 Tipos de licencias de Software Libre.

Una licencia es una autorización formal que proporciona el autor del software para su uso. Cada una de las licencias descritas tiene como objetivo proteger los derechos de autor. Dependiendo del tipo de tipo de licencia puede ser posible convertir software libre a software propietario, siempre y cuando se respeten los términos establecidos en la licencia.

Las siguientes licencias son las más usadas por el Software Libre.

1.1.5.1 General Public License (GNU GPL)

Licencia creada por la Free Software Fundation en 1988. Su principal propósito es el declarar al software licenciado sobre ella como software libre y protegerlo de cualquier tipo de privatización. Tiene como restricciones la libre distribución, modificación y uso del software.

Otra característica radica en el respeto de los derechos de autor "copyleft" de las mejoras realizadas en el software. También es importante señalar que las nuevas versiones creadas deberán de estar regidas sobre la misma licencia (no se podrá cambiar el licenciamiento del software). Todo esto con el objetivo de no volver el software propietario al momento de realizar mejoras sobre él.

Actualmente esta licencia se encuentra en la versión tres (GNU GPL V3) publicada el 29 de junio del 2007.

1.1.5.2 Berkeley Software Distribution (BSD)

Licencia creada por la Universidad de California en 1990 y es otorgada principalmente para sistemas BSD (Berkeley Software Distribution).

Su característica principal radica en que el autor renuncia a todas las modificaciones realizadas a su software y no se hace responsable de los efectos que tengan dichas modificaciones. Esta licencia es más permisiva a comparación de la licencia GNU GPL, debido a que la persona que realice modificaciones sobre el software regido con dicha licencia tendrá la libertad de cambiar a cualquier otra licencia de software libre o incluso volver su versión propietaria.

Uno de los objetivo a cumplir en este proyecto de tesis es buscar alguna alternativa de uso libre que soporte el protocolo Netflow. El protocolo Neflow es un estándar creado por Cisco que se basa en la generación de estadísticas obtenidas por router o switch mediante el monitoreo de red.

El monitoreo de red es una técnica altamente utilizada en empresas por administradores de redes para observar la salud de la red. En la siguiente sección se describe esta técnica.

1.2 Monitoreo de red.

1.2.1 Definición.

El monitoreo de red es una técnica utilizada para observar el tráfico que circula en una red de datos, siendo de gran utilidad esta técnica en las empresas, debido a que un administrador se puede dar cuenta mediante su uso sobre los servicios que no estén levantados, enlaces que estén fallando o detectar anomalías presentes en la red.

Normalmente al encontrar alguna anomalía se avisa a los administradores de red mediante alertas, como puede ser el envió de correos electrónicos o generando alarmas encargadas de notificar la anomalía que ha ocurrido en la red.

Existen programas dedicados exclusivamente al monitoreo de la red; como puede ser Nagius, Netflow y HpOpenView, entre otros. Cada software genera estadísticas del tráfico que circula a través de la red y por medio de análisis realizado a los flujos obtenidos se calcula:

- Utilización de ancho de banda de interfaces de red.
- Utilización de protocolos TCP, UDP, ICMP
- Utilización y disponibilidad de servicios (HTTP, FTP, SSH, SNMP, etc.).
- HostS, redes que consumen el mayor ancho de banda.
- Detección de comportamientos anormales en la red, entre otras funcionalidades.

Existen diversos métodos para realizar monitoreo de red, como son:

- Activar la tarjeta de red en modo promiscuo.
- Utilizando el servicio SNMP
- Mediante el protocolo Netflow.

La tabla I.1 mostrada en la introducción describe brevemente estos métodos que utilizan el monitoreo de red.

En este proyecto de tesis se decidió utilizar el protocolo Netflow que ha adquirido bastante popularidad en la actualidad como una nueva técnica eficiente en el monitoreo de red, en la siguiente sección se describe profundamente esta técnica.

1.3 Netflow.

1.3.1 Definición

NetFlow es un protocolo abierto desarrollado por Darren Kerr y Barry Bruins, de CISCO Systems, que permite la recolección de tráfico de red.

La tecnología NetFlow describe la manera en la que un *router* y/o un *switch* inteligente exportan estadísticas sobre el tráfico que pasa por el mismo, mediante la generación de registros NetFlow (denominados flujos) que se exportan vía datagramas UDP a un dispositivo o máquina recolectora.

Actualmente Netflow cuenta con 10 versiones, Netflow V1-V9 e IPFIX (también conocida como V10). Tras las versión 5, la versión más utilizada en el mercado es la versión 9. Esta versión se basa en plantillas, permitiendo varios formatos para los registros NetFlow, siendo así mucho más flexible y extensible.

Aunque inicialmente el protocolo NetFlow fue implementado por CISCO, la necesidad de un protocolo estándar y universal que permita la exportación de información en flujos de red desde distintos dispositivos de red, ha hecho que NetFlow haya emergido como un estándar en la *IETF* (RFC 3954 Netflow v9).

Un punto muy importante a señalar sobre los registros NetFlow es que éstos no contienen información del usuario, sólo datos de conexión, lo que permite tener una visión detallada del comportamiento del segmento de red (tanto para el monitoreo como para el análisis efectivo de dicho tráfico), evitando problemas relacionados con la privacidad de los usuarios.

1.3.2 Flujo.

Un flujo es una cadena unidireccional de paquetes entre una fuente y un destino, los cuales están definidos por una dirección IP, puerto origen y destino respectivamente.

Cada paquete enviado se clasificara en un flujo a través de los siguientes datos

1. IP Origen
2. IP Destino
3. Puerto Origen
4. Puerto Destino
5. Tipo de Protocolo de capa 3
6. ToS (Type of service)
7. Interfaz utilizada en el dispositivo activo (router o switch)

Por medio de los siete campos contenidos en cualquier flujo se logra que cada paquete capturado sea único y la agrupación de paquetes con contenido igual en un mismo flujo [http://www.cisco.com/en/US/products/ps6601/prod_white_papers_list.html].

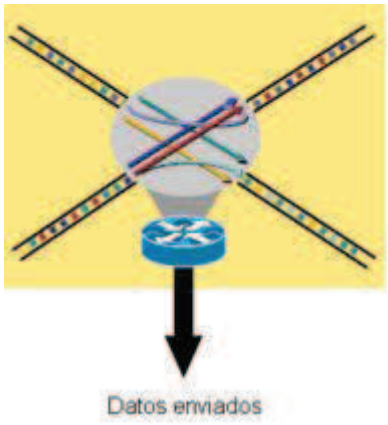


Figura 1.1 Representación de un flujo.

En la figura 1.1 se observa a un router que está clasificando la información que llega a él en distintos flujos de acuerdo con los siete campos que componen a un flujo. El dispositivo activo realizará las siguientes tareas:

- Enviar la información hacia su destino.
- Clasificar toda la información presente en flujos.
- Almacenar todos los flujos en una tabla (llamada cache).
- Enviar el contenido de la tabla hacia un dispositivo colector

1.3.2.1 Calculo del flujo

Una vez definido lo que es el flujo, los componentes presentes sobre él y los criterios utilizados para su clasificación. Para poder agrupar la información en flujos será necesario activar una cache (tabla) en el dispositivo activo. Toda cache activada sobre algún dispositivo activo, se encargará de:

- Contener toda la información que circule sobre el router en flujos
- Clasificar cada paquete presente en un flujo, tomando en cuenta los siguientes criterios:
  - Si el paquete analizado no se puede agrupar en algún flujo creado, se creara un nuevo flujo que contendrá información sobre este paquete y se añadirá a la cache.
  - Si el paquete analizado contiene información igual a algún flujo creado anteriormente. El paquete será agrupado sobre este flujo, sumando la cantidad de tráfico y paquetes del nuevo paquete al total contenido en el flujo.
- Crear un registro del total de flujos almacenados en la cache. Desglosando el tráfico presente en los protocolos TCP, UDP e ICMP y actualizar este registro cada que se añada un nuevo flujo a la cache.
- Agrupar el contenido de la cache en un paquete de exportación cada determinado tiempo.
- Enviar el paquete de exportación hacia un dispositivo colector cada determinado tiempo.

1.3.3 Funcionamiento del protocolo Netflow

El funcionamiento de esta técnica radica en la recolección e interpretación de los flujos creados en un determinado tiempo. Normalmente se realiza una comparación entre los flujos obtenidos actualmente y el promedio de los flujos almacenados en un lapso de tiempo específico. Si el resultado del análisis muestra un aumento inusual en el ancho de banda o se ha detectado un comportamiento anormal, se notifica inmediatamente sobre dicho evento a los administradores de red. Debido a la posibilidad de un ataque denegación de servicios o un malware presente en la red.

1.3.3.1 Conceptos Básicos

Los siguientes conceptos son necesarios para entender el funcionamiento del protocolo Netflow:

- **Observation Point:** Cualquier *dirección IP* que se desea observar dentro de la red.
- **Observation Domain:** Conjunto de Observation point que se encuentran dentro de un dispositivo activo en la misma red y tienen habilitado la exportación de datos vía Netflow.
- **IP Flow or flow:** Es el conjunto de paquetes que pasan a través del Observation Point en un intervalo de tiempo. También conocidos como flujos.
- **Flow record:** Provee información acerca de los flujos que se están observando en el Observation Domain.
- **Exportador:** Dispositivo activo (generalmente router) que tiene el servicio de Netflow habilitado; y que por medio de una *memoria volátil* dedicada almacenará, agrupará los flujos y creará el export packet que contendrá la cantidad de paquetes obtenidos de Observation Domain.
- **Export Packet:** Paquete originado en el dispositivo exportador con el objetivo de transportar los flow records generados por este dispositivo hacia el colector en un tiempo específico.
- **Colector:** Dispositivo que estará escuchando sobre un puerto UDP determinado con el objetivo de obtener los export packet enviados hacia él de uno o varios exportadores, este dispositivo almacenara la información en algún medio (Bases de datos, archivos, etc.) y tendrá la información lista para su interpretación mediante un software licenciado o de uso libre que soporte el protocolo Netflow.
- **Packet Header:** Es el primer campo que contiene un export packet. Contiene información acerca de ese paquete en específico. Ejemplo: la versión de Netflow utilizada.
- **Template record:** Define la estructura e interpretación de los campos en un flow data record.
- **Flow data record:** Son datos que contienen valores de los flujos (como puede ser dirección IP, puerto, protocolo, etc.) y que están asociados con el template record.
- **Options Template Record:** Define la estructura e interpretación de los campos de un Options Data Record.
- **Options Data Record:** Son datos que contienen valores e información sobre los parámetros de medición de los flujos, correspondientes a un Options Template Records.
- **Flowset:** Son flow record que tienen una estructura similar. En un export packet, uno o más flowset siguen el packet header. Se suelen dividir en:

- **Template Flowset:** Son uno o más templade records que han sido agrupados en un Export packet.
- **Options Template Flowset:** Son uno o más option templade records que han sido agrupados en un export packet.
- **Data Flowset:** Son uno o más flow records de un mismo tipo agrupados en un export packet. Cada registro es también un Flow Data Record o un Options Data Record definido por un Templade Record o un Options Template Record.

Para entender mejor acerca del formato de los datos presentados por Netflow la figura 2 muestra un ejemplo de un export packet generado para la versión 9 [http://www.cisco.com/en/US/products/ps6601/prod_white_papers_list.html].

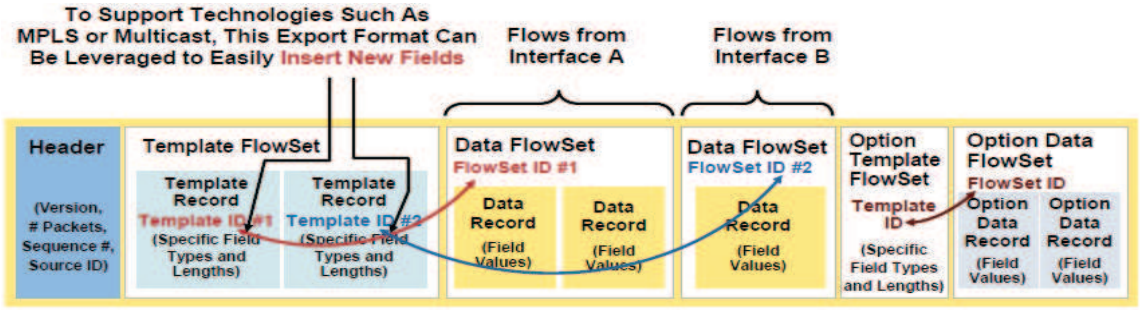


Figura 1.2 Formato Export Packet V9

Como se observa en esta figura, se tiene al inicio al packet header (Header) que indica: la versión utilizada, número de paquetes, número de templates ocupados, un número de identificación sobre el export packet (ID). Después del packet header se tendrán campos agrupados con una estructura similar (FlowSet). Los FlowSet se dividen en los mencionados anteriormente.

En el ejemplo mostrado se observa que el primer campo después del packet header corresponde al Templade Flowset. Este campo contiene información general acerca de los templates generados. A cada template se le asignará un número ID único. En el ejemplo se observa que cada template record creado corresponde a información obtenida de diferentes interfaces en el router.

En los siguientes dos campos se observan todos los datos obtenidos del template ID 1 (interfaz A) y el *templade* ID 2 (Interfaz B) respectivamente. Por último se contiene el campo Option template records que puede contener información acerca del tráfico consumido de los templates anteriores u otra información agrupada en un nuevo template.

Para entender el funcionamiento del protocolo Netflow, se describirán los tres componentes principales utilizados.

1.3.3.2 Componentes básicos en el protocolo netflow.

Se distinguen tres componentes básicos en el protocolo Netflow. Estos son:

Exportador. Router o switch capaz de generar registros NetFlow (Export packets), que se exportarán a un colector vía UDP.

Colector. Dispositivo que escucha en un puerto UDP determinado y que es capaz de almacenar o reenviar los flujos recibidos a otros colectores según la arquitectura definida.

Analizador. Software encargado de filtrar, mostrar, analizar y/o visualizar los flujos recibidos.

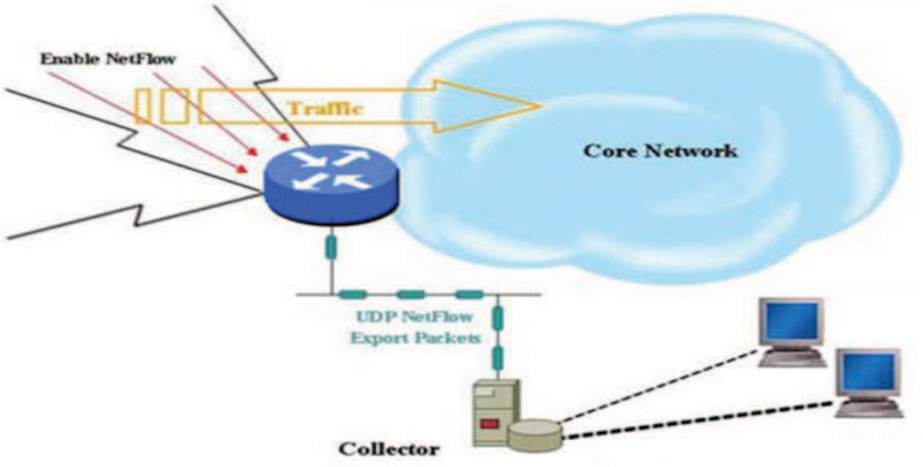


Figura 1.3 Funcionamiento protocolo Netflow

La figura 1.3 [http://www.cisco.com/en/US/products/ps6601/prod_white_papers_list.html] muestra el funcionamiento del protocolo Netflow de la siguiente forma:

- El tráfico presente sobre un router, que soporte el protocolo Netflow, será capturado, interpretado y clasificado en diferentes flujos. Dicho router se encargará de generar Export packets cada determinado tiempo (5 minutos por default).
- Los export packets se dirigirán hacia un equipo colector que se encargará de almacenar todos los datos provenientes de los exportadores.
- Con ayuda de un analizador podremos interpretar los archivos capturados y someterlos a un análisis detallado con ayuda de algún software licenciado o de uso libre.

Para entender mejor la forma en la que Netflow genera los Export Packets, es necesario explicar el concepto de Netflow cache.

1.3.3.3 Memoria volátil dedicada Netflow (cache).

Los exportadores operan construyendo una memoria volátil dedicada (Netflow cache) que contiene información sobre todos los flujos activos que pasan por el dispositivo. Cada flujo está representado por un flow record que contendrá los siete campos ocupados en la clasificación de un flujo, además de información extra relacionada con la conexión. La *caché* se actualiza cada vez

que se obtiene un nuevo flujo, llevando una cuenta de los paquetes y bytes por flujo. Después de un lapso de tiempo (definido por el administrador) se creará el *export packet* que contendrá todos los datos almacenados en la netflow cache. Al momento de ser enviado el export packet, la cache solo contendrá aquellos flow record que no hayan expirado.

Un flow record expira de la caché según una serie de criterios, algunos de ellos configurables en el dispositivo.

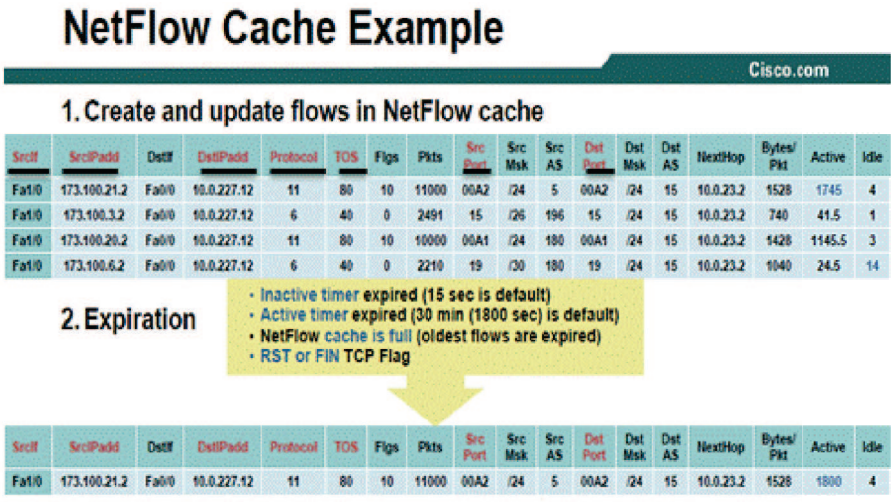
Estos criterios son:

- Cuando las conexiones TCP llegan a su fin (Flag FIN) o son reseteadas (se recibe un flag RST).
- Los flujos han estado inactivos por un tiempo determinado (Normalmente 15 segundos).
- La caché se llena o el router se queda sin recursos.
- Los flujos se mantienen activos en caché por un tiempo determinado (Normalmente 30 minutos). Una vez pasado este tiempo expiran de la caché, asegurando un reporte periódico. El envío se hace más frecuente si aumenta el tráfico de las interfaces configuradas con NetFlow.

Todos los flow record que han expirado en la caché se adjuntan en un datagrama de exportación UDP (export packet), que típicamente contendrá entre 20 y 50 registros. Este datagrama se envía a un puerto determinado al dispositivo colector configurado.

Para la colección y análisis de los flujos recibidos por los dispositivos de red se pueden utilizar diversos productos disponibles en el mercado tanto de libre distribución o comerciales.

En la figura 1.4 se muestra un ejemplo del contenido de una Netflow cache construyéndose en un router.



SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/ Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1745	4
Fa1/0	173.100.3.2	Fa0/0	10.0.227.12	6	40	0	2491	15	/26	196	15	/24	15	10.0.23.2	740	41.5	1
Fa1/0	173.100.20.2	Fa0/0	10.0.227.12	11	80	10	10000	00A1	/24	180	00A1	/24	15	10.0.23.2	1428	1145.5	3
Fa1/0	173.100.6.2	Fa0/0	10.0.227.12	6	40	0	2210	19	/30	180	19	/24	15	10.0.23.2	1040	24.5	14

SrcIf	SrcIPadd	DstIf	DstIPadd	Protocol	TOS	Flgs	Pkts	Src Port	Src Msk	Src AS	Dst Port	Dst Msk	Dst AS	NextHop	Bytes/ Pkt	Active	Idle
Fa1/0	173.100.21.2	Fa0/0	10.0.227.12	11	80	10	11000	00A2	/24	5	00A2	/24	15	10.0.23.2	1528	1800	4

Figura 1.4 Netflow cache

Como se observa en la figura 1.4 se tienen cuatro flujos activos, definidos por los campos subrrayados (criterios para la clasificación de cada flujo); cualquier flujo nuevo que contenga los

mismos datos de clasificación a alguno almacenado en caché, se añadirá a ese flujo sumando la cantidad de paquetes y tráfico consumido del nuevo flujo a la información ya almacenada. En caso de que el nuevo flujo contenga datos diferentes a los presentes, se creará una nueva entrada en la caché conteniendo la información de este nuevo flujo.

Según los criterios mencionados en la expiración del flow record, se observa que el primer flujo ha estado activo por un tiempo mayor a 1800 segundos. Por lo cual expira de la caché y se unirá al export packet a enviar al dispositivo colector.

Además se observa que en la caché se guardan solo datos de la conexión. No se observa ningún dato que contenga información referente al usuario. Lo que proporciona una gran ventaja en cuanto a la privacidad de la información.

1.3.4 Versiones de Netflow

Actualmente el protocolo Netflow cuenta con 10 versiones: La versión estándar es la 5; las versiones 1 a 4 son propietarias de cisco. La versión 9 es la que ofrece mejor compatibilidad con varios dispositivos mediante el uso de templates. Permitiendo la transmisión en diferentes formatos de datos Netflow en un mismo *export packet*, brindando con esto mayor flexibilidad.

La tabla 1.2 muestra un breve comentario sobre las versiones de Netflow más utilizadas.

Tabla 1.2 Principales características de las versiones de NetFlow

Versiones de Netflow	Comentario
1	Versión original, solamente soportada por router cisco.
5	Estandarizada y es la más usada en la actualidad, solo soporta Ipv4.
7	Solo es utilizada en las series de switch cisco C6500 y 7600
8	Múltiple compatibilidad entre swich, permite diferentes esquemas
9	Uso de templates. Al igual que la V8 tiene múltiple compatibilidad, además de esto es más flexible que las versiones anteriores, debido a que tiene soporte para campos adicionales y tecnologías como ejemplo: ➢ MLPS, BGP, Ipv6, entre otros.

Las versiones más utilizadas por el protocolo Netflow son la versión 5 y la versión 9. Por este motivo se realiza una comparación detallada entre cada una de ellas.

1.3.4.1 Netflow V5 vs V9

Si bien es cierto que la versión cinco de Netflow es la versión estándar, la versión nueve provee grandes mejoras en la representación de los datos en formato Netflow, la mejora más significativa entre estas versiones fue el agregar templates a la V9 de Netflow, permitiendo con esto mayor flexibilidad en los datos enviados.

La figura 1.5 [http://www.plixer.com/blog/netflow/netflow-v9-vs-netflow-v5/] muestra una comparación sobre el formato de un *export packet* entre la versión 5 y la versión 9.

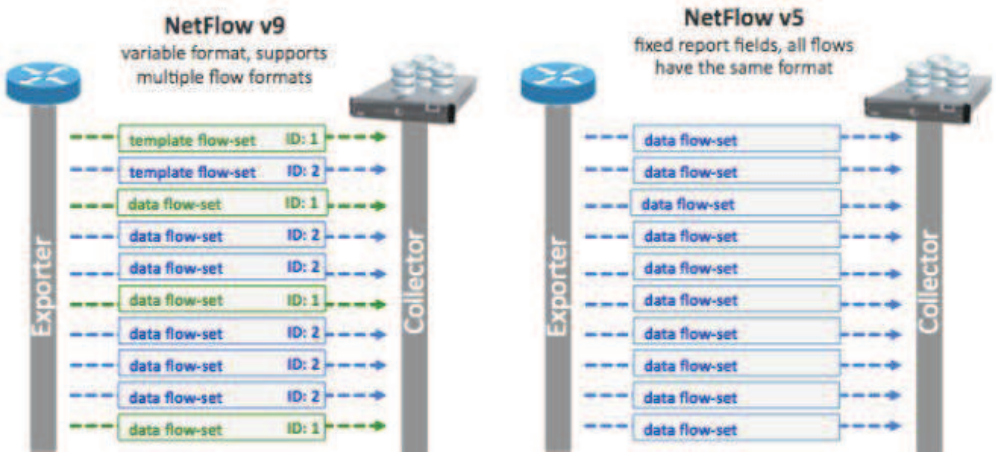


Figura 1.5 Netflow v5 Vs Netflow v9

El principal cambio entre estas versiones es el uso de templates en la versión 9. Logrando con esto el transmitir varios datos en distinto formato sin la necesidad de crear otro *export packet* para su transmisión.

Otra principal diferencia que se tiene entre la versión 5 (V5) y la versión 9 (V9) es que la V5 tiene un formato fijo de datos. Al utilizar la V5 solo podrán enviar datos hacia los routers que cumplan con el formato establecido por la V5. El export packet recibido no podrá ser interpretado por colectores que no soporten la V5. Mientras que la V9 con el uso de templates permite la combinación de varios campos en un mismo *export packet*. Logrando una configuración flexible dependiendo de la opción a realizar y que múltiples dispositivos puedan utilizar la V9 de Netflow adaptándola a sus necesidades.

Cada template creado contara con un único número que lo identificara (ID). Dependiendo de los datos recibidos se agruparan en el ID que les corresponda, logrando con esto tener múltiples datos con formatos distintos en un mismo export packet; pero cada dato o conjunto de datos solo pertenecerá al ID que hará referencia a su template.

1.3.5 Ventajas y consideraciones de utilizar Netflow

Netflow nos proporciona bastantes beneficios. Entre los cuales destacan:

- Responde a las preguntas qué, quién, cómo, dónde y cuándo acerca del tráfico cursado en la red.
- Provee estadísticas acerca de redes más utilizadas.
- Provee una visión detallada acerca del comportamiento de la red.
- Estandarizado**:** RFC 3954 Netflow V9.
- Monitoreo de la Red**:** Con técnicas de análisis de flujo.
- Monitoreo de Aplicaciones (basado en puertos TCP/UDP): Para planificar, entender nuevos servicios, y distribuir recursos y aplicaciones en la red.
- Monitoreo de Usuarios: para revisar de forma efectiva la utilización de los recursos por parte de los usuarios.

- Planificación de la Red: para anticiparse a los crecimientos de la red, ya sea en dispositivos, puertos y ancho de banda
- Análisis de seguridad: con el fin de detectar anomalías en el tráfico de la red.
- Contabilidad y la Facturación: Netflow es la principal tecnología desarrollada en estas áreas, debido a sus detalladas estadísticas. Permite poner precio al BW consumido.
- Almacenamiento de los Datos Netflow: Permite guardar estadísticas de los flujos capturados para realizar futuros análisis.
- No provee información sobre los usuarios: Solo nos muestra información sobre la conexión.
- Agrupar los datos: Mejor organización de los datos.
- Dispositivos Activos: Fue creado inicialmente para router y switch cisco, pero cada vez más empresas desarrolladoras de dispositivos activos incluyen este protocolo en sus equipos.
- Alcance: Debido a que NetFlow puede ser configurado en la mayoría de los routers y switch. Se tiene una excelente visión sobre el tráfico presente en la red.
- Entre otros.

1.3.6 Aplicación de la tecnología NetFlow en el área de seguridad informática.

El uso de NetFlow ha demostrado de gran utilidad para múltiples fines relacionados con el monitoreo, contabilidad y cobro de transmisión del tráfico o uso de red (del inglés “measurement, accounting and billing”).

Algunas funciones que provee Netflow orientado a seguridad informática son las siguientes:

- Detección en redes de la utilización de puertos TCP/UDP específicos.
- Propagación del malware en la red
- Detección de tráfico relacionado con un incidente ocurrido.
- Investigación de ataques DoS (Denial of Service).
- Host/subredes que consumen mayor tráfico en la red.

La principal ventaja de Netflow contra otras técnicas de monitoreo de red enfocadas hacia seguridad se presenta en omitir información acerca del usuario. Proporcionando la información de forma sencilla y fácil de comprender.

Netflow proporciona una serie de ventajas frente al uso de IDS y otros mecanismos de detección perimetrales, entre las cuales se pueden destacar:

- Dado que la mayoría de dispositivos de red tienen la capacidad de exportar registros NetFlow, el monitoreo se puede realizar desde cualquier router en la infraestructura. Incluidos routers de acceso a Internet donde normalmente se ubican IDSs y Firewalls, teniendo así además una vista única del tráfico total de la red a nivel de infraestructura.
- A diferencia de los IDSs, con la utilización de Netflow no se tiene acceso a información confidencial; debido a que los flujos no contienen información del usuario, sólo datos de conexión. Esta es una de las mayores ventajas de esta tecnología.
- La detección de ataques de día cero”zero-day” o mutaciones de ataques por medio de un análisis realizado hacia los registros obtenidos por Netflow. Especialmente útil cuando la detección basada en firmas no es válida.

1.3.6.1 NetFlow enfocado en la detección de ataques y anomalías

NetFlow permite detectar, en tiempo real, ataques o anomalías presentados en la red. Como puede ser:

- Equipos posiblemente comprometidos (presenten alguna infección).
- Conexión hacia servidores clientes desconocidos.
- Envío de información a keyloggers.
- Ataques DoS/DDoS.
- Escaneos de puertos y redes.
- Infecciones específicas de determinados gusanos.
- SPAM, entre otros.

Existen varios métodos de análisis de flujos para detección de ataques y anomalías de seguridad. Como puede ser:

Realizando un análisis básico del tráfico. Se trata de un modelo basado en la descripción de las actividades consideradas como "normales" en la red de acuerdo a patrones históricos de tráfico. De manera que cualquier otro tipo de tráfico se marca como malicioso. La forma más básica de realizar esta tarea es mediante el uso de estadísticas, informes de datos y sesiones (estadísticas *TopN*).

Basado en expresiones regulares. Con el objetivo de detectar patrones de comportamiento anormal, cualquier campo de los incluidos en los registros NetFlow es susceptible de ser utilizado en una búsqueda por medio de expresiones regulares.

Por medio de alertas. La mayoría del software incorpora la opción de alertas. Esta opción nos permite realizar un análisis del comportamiento normal presente en la red. Dicho comportamiento sirve como línea base para realizar comparativos posteriores. Cuando se presenta un comportamiento anormal se ejecutará una alerta notificando un aumento en el tráfico o aumento en puertos específicos.

Con ayuda de filtros aplicados hacia los registros Netflow se logra acotar la información y poder realizar análisis más robustos.

Por medio de algoritmos. Se pueden realizar programas enfocados hacia la detección de actividades anormales presentadas en el tráfico capturado por Netflow. Como puede ser:

a. Escaneo de puertos o direcciones IP.
b. Ataques denegación de servicio.
c. Envío de información hacia servidores externos
d. Aumento del tráfico en la red, entre otros.

Una vez que se ha descrito lo que es el software libre, el monitoreo de red y el funcionamiento del protocolo Netflow, conceptos necesarios para el desarrollo de este proyecto de tesis. Se

explicarán brevemente conceptos requeridos de seguridad informática, necesarios para la correcta creación del detector de malware.

1.4 Conceptos de Seguridad Informática.

Lo que se pretende en esta sección es dar una visión general de lo que es la seguridad informática, los criterios usados para la clasificación de ataques y las buenas prácticas. Tomando como base estos criterios para la estrategia implementada en la detecciónel malware mediante el monitoreo de red.

1.4.1 Conceptos Básicos de Seguridad.

En esta sección se explicara a grandes rasgos diversos conceptos esenciales de seguridad informática

Seguridad: Diversas acciones y herramientas que me permiten proteger mis bienes o activos importantes para una persona o grupos de personas.

Tecnologías de la información: Son todos aquellos dispositivos o medios electrónicos que me permiten manipular la información.

Seguridad Informática: Diversas acciones y herramientas que me permiten proteger toda la tecnología de la información.

Servicios de seguridad.
Un servicio de seguridad es aquel que me permite mantener la seguridad en un sistema de cómputo.

Los servicios de seguridad están diseñados para contrarrestar los ataques a la seguridad y hacen uso de uno o más mecanismos de seguridad para proporcionar el servicio. Se clasifican en los siguientes.

- Control de acceso. Acceder a mi bien o activo siempre y cuando esté autorizado.
- Confidencialidad. Los usuarios deben de tener la privacidad de ver o realizar cambios en los bienes o activos.
- Integridad. Los usuarios deberán de obtener los bienes o activos tal y como ellos los modificaron (sin ninguna alteración por terceras personas).
- Disponibilidad. Todo bien o activo debe de estar presente al momento que un usuario autenticado acceda a ellos.
- No repudio. Al momento que un usuario realice cambios sobre los bienes o activos, deberá de aceptar que realizó dichas modificaciones.
- Autenticación. Serie de medios para garantizar que una persona es quien dice ser
  Factores de autenticación.
  - Algo que se sabe. Contraseña, clave.
  - Algo que se es. Administrador, jefe.
  - Algo que se posee. Certificado de seguridad, permisos.
  - Algo que se hace. Firma autógrafa, decir alguna frase
  - Desde algún lugar. Terminal remota, Lugar identificado por IP, *VPN(Virtual Private Network)*

Mecanismos de seguridad.
Son las herramientas o controles que permiten implementar un servicio de seguridad. Los mecanismos de seguridad pueden ser: aplicaciones (lógicos), físicos, buenas prácticas (reglas morales), reglas (políticas de seguridad), estándares, etc.

Las herramientas pueden ser: Preventivas, correctivas o detectoras.

1.4.2 Deficiniones de vulnerabilidades, amenazas y ataques

1.4.2.1 Vulnerabilidad.

Deficiencia o punto(s) débil(es) encontrado(s) que puede(n) ser explotado(s) por perpetradores con el objetivo de comprometer o dañar nuestros activos y bienes.

1.4.2.2 Amenaza.

Circunstancia o evento que aprovecha una vulnerabilidad y compromete la integridad, disponibilidad y confidencialidad. Generalmente pretende, puede o intenta destruir algo o a alguien. Siempre está latente, es importante señalar que puede o no materializarse.

Tanto las amenazas como las vulnerabilidades se clasifican en físicas, humanas, desastres naturales, hardware, software y de red.

1.4.2.3 Ataque.

Es una consumación de una amenaza, generalmente los ataques explotan una o varias vulnerabilidades y el grado de peligrosidad dependerá del perpetrador que los ha creado.

Los ataques responden al siguiente esquema: *Vulnerabilidad + Amenaza = Ataque*.

Todo ataque antes que el perpetrador lo ejecute sobre su objetivo específico, deberá de cumplir con las siguientes etapas:

1. *Planeación:* El perpetrador en esta etapa se encargará de recolectar información del objetivo(s) a atacar por cualquier medio posible. Además de visualizar el o los objetivos que pretende cuando se ejecute el ataque.
2. *Activación:* En esta etapa el ataque se encuentra en plena ejecución. Generalmente ya habrá explotado la vulnerabilidad(es) encontrada(s) resultado del análisis del punto anterior
3. *Ejecución:* Son los logros obtenidos por el perpetrador una vez que el ataque se encuentra en ejecución.

1.4.2.3.1 Clasificación de los ataques

Los ataques se suelen clasificar de acuerdo con los siguientes criterios:

- ¿Qué tan Intrusivo es? En este punto se clasifica al ataque de acuerdo con la capacidad de esconderse ante posibles detecciones de herramientas de seguridad o usuarios:
    A. *Ataque pasivo:* Son aquellos ataques que tienen el objetivo de pasar desapercibidos por las herramientas de seguridad y los usuarios.

B. *Ataque activo:* Son aquellos ataques que tienen el objetivo de causar alguna modificación evidente sobre los datos, bienes o activos.

- ¿A qué servicio de seguridad ataca? Se clasifica al ataque de acuerdo con el servicio de seguridad que pretende comprometer
  - A *Modificación:* Ataque realizado con el objetivo de crear algún cambio o alterar el flujo normal de comunicación. Atenta contra la integridad. Ejemplos: Backdoor, obtención de passwords, Arp Spoofing, IP Spoofing, Fuerza bruta, exploid, entre otros.
  - B *Suplantación:* Ataque realizado con el objetivo de obtener acceso hacia el flujo de comunicación robando la identidad de un usuario autorizado. Atenta contra la autenticación. Ejemplos: Ciber Graffiti, SQL Injection, Borrado de huellas, entre otros.
  - C *Interrupción:* Ataque que tiene como objetivo impedir el flujo normal en la comunicación. Atenta contra la disponibilidad. Ejemplos: DoS (Ataque de negación de servicios), DDoS (Ataque de negación de servicios distribuido), entre otros.
  - D *Intercepción:* Ataque que tiene como objetivo observar las acciones realizadas en el flujo de comunicación. Atenta contra la confidencialidad. Ejemplos: Sniffers, Keylogger, entre otros.

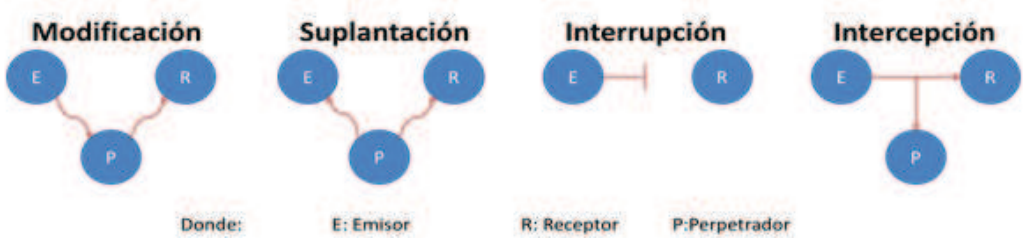


Figura 1.6 Esquema de ataques Fuente: Apuntes clase seguridad informática I, profesor: M.C. Cintia Quezada Reyes

¿Dónde se realiza? Se clasifica desde un punto de vista geográfico (lugar de ejecución).

- *Ataques Internos:* Ataque realizado por un usuario perteneciente a la empresa, institución, organización
- *Ataques externos:* Ataque realizado por un usuario externo a la empresa, institución, organización.

En la última sección de este capítulo se explicará a detalle el software malicioso, o también conocido como malware.

1.5 Malware.

1.5.1 Definición.

El malware es cualquier tipo de software dañino que puede afectar a un equipo electrónico de varias formas, como puede ser:

- Actuando como software espía.
- Obteniendo el password del equipo “víctima”.
- Realizando ataques pasivos o activos hacia los equipos “víctima”, con el fin de obtener información, causar daños o molestar.
- Engañando a la “víctima” mediante páginas web falsas, Ingeniería social, etc.
- Recolectar información del equipo “víctima”.

El Malware puede ser propagado por cualquier medio de comunicación disponible (Internet, Correo, P2P, Mensajería, etc.) y por supuesto por cualquier sitio web.

1.5.2 Clasificación del Malware

El malware se suele clasificar de acuerdo con la acción que realiza, su comportamiento o el grado de peligrosidad que suele tener en los equipos que logra infectar. El malware más común son:

1.5.2.1 Virus.

Programas creados para infectar sistemas u otros programas. Una vez que el virus logra infectar se encargara de realizar modificaciones y daños con el objetivo de provocar un mal funcionamiento general del equipo, registrar, dañar o eliminar datos, o bien propagarse por otros equipos a través de Internet.

1.5.2.2 Backdoor.

Son programas diseñados para abrir una "puerta trasera" en la víctima y permitirle a otro malware tener acceso.

1.5.2.3 Bootnets (Redes zombies).

Son computadoras infectadas por algún malware que actúan en conjunto enviando peticiones hacia servidores con el objetivo de saturarlo y por consecuente afectar la disponibilidad del sistema y/o aplicativo.

1.5.2.4 Exploit.

Programa o código que "explota" alguna vulnerabilidad del sistema o de parte de él, aprovechando esta deficiencia para beneficios propios.

Generalmente los exploit al encontrar alguna vulnerabilidad en el sistema accederán a ella y le permitirá el acceso a otro tipo de malware.

1.5.2.5 “Zero day” (Ataques de día cero).

Es cualquier tipo de malware del que no se tiene conocimiento. Generalmente cuando un exploit encuentra una nueva vulnerabilidad se encarga de ejecutar acciones para permitir a él mismo o a otro tipo de malware entrar y realizar ataques sobre el sistema.

Como es la primera vez que se ejecuta el ataque no se tendrá ningún parche, se categoriza como día cero al primer ataque realizado. Esta clasificación terminará cuando se tenga algún mecanismo de seguridad que contrarreste a este malware.

1.5.2.6 Gusanos (Worm)

Son programas desarrollados para reproducirse por algún medio de comunicación como el correo electrónico, redes P2P, memorias USB, internet, etc.

Su principal objetivo es llegar a la mayor cantidad de usuarios posible y lograr distribuir otros tipos de de malware (como Troyanos, Backdoors y Keyloggers, etc.).

Generalmente los gusanos están orientados a hacer ataques DoS contra sitios webs específicos y tiene como principal característica el saturar la red.

1.5.2.7 Hoax

Son mensajes enviados principalmente por correo electrónico que contienen contenido falso o tratan de engañan a sus víctimas por diferentes formas, como puede ser:

- Creación de nuevos virus y traen un adjunto con el parche que generalmente es un software espía o un virus.
- Mensajes de personas enfermas e incitan a ayudarlas por distintos métodos.
- Cadenas (SPAM).

Generalmente estos mensajes tienen el objetivo de saturar la red, los servidores de correo y obtener direcciones de correo.

1.5.2.8 Keylogger

Los keylogger son programas que capturan todo lo que teclea la máquina infectada e inclusive algunas aplicaciones capturan los clics efectuados con el mouse. Este tipo de malware envía las capturas a sus creadores. Por medio de un análisis obtienen información confidencial como nombres de usuario, contraseñas, números de cuentas, etc.

1.5.2.9 Phishing

Son páginas falsas creadas. Generalmente copias idénticas de las páginas de bancos o empresas importantes con la finalidad de engañar al usuario haciéndole creer que se encuentra en la página oficial. Al engañar al usuario el malware logra obtener información confidencial.

Generalmente los phishing llegan a las víctimas por medio de correos electrónicos haciéndose pasar por un correo de algún sitio oficial y al dar clic en sus ligas reenviará al usuario a la página web falsificada.

1.5.2.10 Spam

Son mensajes enviados a destinatarios sin su consentimiento, generalmente son mensajes publicitarios o contienen información de páginas falsas (phishing). Son enviados en forma masiva y tienen como objetivo el ser vistos por el usuario o saturar el servidor de correo utilizado.

1.5.2.11 Spyware

También conocidos como software espías. El objetivo de este malware es recolectar información de alguna persona o institución sin su consentimiento y distribuirlo a personas interesadas en esta información obtenida ilegalmente.

Generalmente recolectan información como nombres de usuario, contraseñas, direcciones IP, páginas web visitadas, etc.

1.5.2.12 Troyanos

Son virus informáticos o algún programa malicioso capaz de alojarse en computadoras y permitir el acceso a usuarios externos, a través de una red local o de Internet, con el fin de recabar información.

Suele ser un programa pequeño alojado dentro de una aplicación, una imagen, un archivo de música u otro elemento de apariencia inocente, que se instala en el sistema al ejecutar el archivo que lo contiene.

Generalmente cuando un troyano es ejecutado o instalado realiza acciones imperceptibles para el usuario y el sistema operativo, pueden actuar como espías o inclusive modificar registros del sistema logrando con esto tener un control total en sus acciones efectuadas y evitar ser detectados.

Como se observa, existe una enorme clasificación de malware. Por este motivo es de esperarse que cada malware tenga un comportamiento diferente, dependiendo del tipo de malware y el ataque que pretenda realizar.

1.5.3 Comportamiento del Malware

Debido a la extensa clasificación del malware no es posible definir un comportamiento general que se presente al momento de infectar a una víctima. A continuación se muestran algunos ejemplos del comportamiento del malware y las diferentes acciones realizadas.

1. Algún malware puede infectar a un equipo explotando alguna vulnerabilidad. Una vez infectado el equipo, el malware tratará de propagarse hacia otros equipos y realizar un ataque de negación de servicios.
2. Para la ejecución de algún malware será necesario que el usuario ejecute la aplicación que lo contiene. Una vez activado el malware recolectará información y la enviará hacia IP's desconocidas por el usuario.
3. Algún malware aprovechará alguna vulnerabilidad encontrada para su instalación en el sistema. Una vez instalado, el malware tratará de pasar desapercibido y abrirá una *backdoor* para la ejecución de otro malware.
4. El malware ha detectado alguna vulnerabilidad de día cero, se instalará y realizará acciones como el borrado de datos, cambio de registros, robo de contraseñas etc.
5. Se ha recibido un correo electrónico sospechoso. Algún usuario accede al contenido del correo y el malware se ejecuta. El malware tratará de consumir todo el ancho de banda

disponible en la red e infectar a la mayor cantidad de equipos en el menor tiempo posible. Entre otros.

Como se observa en los ejemplos mostrados, el comportamiento del malware varía dependiendo de la acción que se desee realizar con él. Sin embargo, se ha detectado que la mayoría del malware presenta algunos patrones similares. Normalmente un equipo infectado con algún tipo de malware tratará de realizar lo siguiente.

- Pasar desapercibido: Es de vital importancia para la mayoría del malware el pasar desapercibido. Debido a que si las herramientas de seguridad o el usuario no saben que su equipo se encuentra infectado no realizaran alguna acción para poder eliminar al malware.
- Tratar de infectar a otros equipos. Una vez que el malware ha logrado infectar a una víctima, tratará de explotar la vulnerabilidad encontrada en este equipo sobre otras víctimas pertenecientes a la misma red. Por medio de un escaneo de puertos o escaneo de IPs tratará de infectar a la mayor cantidad de equipos en el menor tiempo posible.
- Consumir recursos. Si el objetivo del malware es el de realizar algún ataque denegación de servicios, tratará de infectar a la mayor cantidad de computadoras posibles y utilizar todo el ancho de banda asignado en ellas para enviar múltiples peticiones hacia la víctima con el objetivo de saturar y tirar la conexión.
- Envió de información hacia IPs ajenas a la red infectada. El malware que presente este comportamiento recolectará información confidencial del equipo infectado y la enviará hacia IPs desconocidas para el usuario.

Estos son los principales comportamientos realizados por el malware, en esta sección me limite a describirlos brevemente. En la sección 4.3 se describe a detalle el funcionamiento de cada técnica descrita. Normalmente un equipo infectado por algún malware suele presentar las siguientes características.

- Se consume demasiado ancho de banda sin razón aparente.
- El equipo trabaja lento.
- Borrado de archivos sin autorización.
- Instalación de software sin autorización.
- Cambio en el registro del sistema; creación de cuentas de usuario sin autorización.
- No poder ejecutar herramientas de seguridad.
- Pérdida de archivos del sistema.
- Reinicio del S.O o imposibilidad de acceder a él.
- Software instalado deja de funcionar sin razón aparente.
- Aparición de páginas web no solicitadas.
- Pérdida de acceso a unidades rígidas.
- Entre otros.

Si el equipo presenta uno o varios de estos síntomas es recomendable el ejecutar alguna herramienta de seguridad en busca de malware que lo haya infectado.

Como último punto en este capítulo, se incluyen algunos consejos para evitar que un equipo sea infectado por malware

1.5.4 Mecanismos de prevención.

Para evitar daños ocasionados por algún malware. Es aconsejable seguir las siguientes reglas como medidas de prevención.

- Tener actualizado el S.O. y software.
- Instalar diversas herramientas de seguridad (como antivirus, antispyware, firewall, IDS, etc.).
- Realizar escaneos con herramientas de seguridad a su equipo de forma periódica (recomendable cada 2 meses).
- Evitar el uso de cuentas con privilegios de administrador.
- No abrir correos cuyo remitente sea sospechoso o desconocido.
- Evitar el uso de "cafés internet". De ser necesario su uso no teclear nombres de usuario o contraseñas en estas máquinas. Además al introducir dispositivos extraíbles en estos lugares realizar un escaneo con algún antivirus. (Más vale prevenir que lamentar).
- No proporcionar información confidencial.
- Usar contraseñas fuertes (al menos ocho dígitos incluyendo: letras, números, símbolos).
- Cambiar contraseña periódicamente (recomendable cada 6 meses).
- Tener un respaldo de la información.
- Evitar la instalación de software crackeado.
- Evitar el uso de barras en navegadores (como las barras de ask, yahoo, etc.) debido a que suelen instalar o descargar contenido malicioso sin el consentimiento del usuario.
- Buenas prácticas.

En caso de detectar algún malware que ha infectado el equipo recomiendo:

- No desesperarse.
- Investigar en internet acerca de los síntomas presentados por el equipo en busca de una solución.
- Aislar el equipo de cualquier red (evitar la propagación del malware).
- Ejecutar herramientas de seguridad en busca del código malicioso.
- Ejecutar el administrador de tareas y buscar procesos sospechosos.
- Ejecutar alguna herramienta enfocada hacia el monitoreo de red. En msdos es posible con el comando "netstat –na" obtener información acerca de las conexiones realizadas por su equipo a internet.
- Ejecutar el S.O en modo aprueba de fallos.
- De ser posible restaurar el sistema a un estado antes de la infección del malware.
- En caso de no poder acceder al S.O ejecutar un *live cd* de alguna distribución Linux y correr alguna herramienta de seguridad en busca del código malicioso.
- Respaldo de información en caso de ser necesaria la reinstalación.

Capítulo II

Investigación y elección del software libre a implementar

2.1 Introducción.

Uno de los objetivos del software libre, como se describió en el capítulo anterior, es desarrollar alternativas de uso libre sobre algún software propietario específico. En este caso el software propietario Netflow ofrece un desempeño eficiente y poderoso en la obtención de estadísticas, reportes, costo de ancho de banda, etc. Sin embargo se tiene el inconveniente de tener que pagar elevadas cantidades por su uso en ambientes de producción.

Esto ha motivado a que se realice una investigación con el objetivo de encontrar alguna alternativa de uso libre que soporte el protocolo Netflow. La alternativa encontrada deberá de cumplir con los siguientes requerimientos para su elección y puesta en marcha:

- ✓ Software libre.
- ✓ No se tenga restricciones en cuanto al número de exportadores instalados.
- ✓ Tenga un colector.
- ✓ Soportar el protocolo Netflow.
- ✓ Generar gráficas y poder observar los datos presentes en ellas.
- ✓ Que soporte por lo menos las versiones 5, 7 y 9 del protocolo Netflow.
- ✓ Que tenga interfaz cliente-servidor vía web.

En este capítulo se explicará todo el proceso realizado para la elección del software libre. El proceso de investigación y elección fue resultado de una búsqueda y análisis de la mejor alternativa que cumplió con las restricciones mencionadas. Además de mostrar una comparación entre el software libre elegido con el software propietario utilizado en la institución.

2.2 Investigación sobre las alternativas de software libre.

La investigación se realizó mediante una búsqueda en internet sobre alternativas de uso libre que cumplieran con los requerimientos descritos, descartando a alternativas que no cumplieran con algún requerimiento establecido.

Durante la investigación se presentaron los siguientes inconvenientes:

- ➢ La mayoría del software libre encontrado sólo soportaba un número limitado de exportadores (no permitían tener más de cinco dispositivos activos enfocados a recolectar información).
- ➢ La mayoría de las alternativas encontradas eran soportadas por S.O. Linux, y muy pocas soportaban S.O. Windows. Esto representa un problema para usuarios que no cuenten con conocimientos básicos del uso de S.O. Linux, debido a que no lograran utilizar el nuevo software al 100%.
- ➢ La versión comercial de Netflow utilizada (no se mencionará el nombre real de este software propietario utilizado, por cuestiones de integridad y confidencialidad de la información presente en la institución), trabaja en S.O. Windows ofreciendo estadísticas muy detalladas del tráfico presente en la red, además de diversas funcionalidades, como lo son:

- Cálculo de costo por tráfico consumido.
- Generación de reportes.
- Clasificación de redes en forma automática.
- Poderosa interfaz web.
- Entre otras.

Debido a las características mencionadas de la versión comercial de Netflow utilizada, fue muy complicado encontrar alguna alternativa software libre que logre sustituir completamente todas las funcionalidades ofrecidas por este software. Al final de este capítulo se muestra una comparación entre la alternativa software libre elegida y la versión comercial utilizada en la institución.

La tabla 2.1 muestra el resultado obtenido de la primera investigación realizada. Mostrando los posibles candidatos que podrían sustituir al software comercialNetflow utilizado en la institución.

Tabla 2.1 Alternativas Open Source a sustituir a la versión comercial de Netflow utilizada

Software	Licencia	S.O. Soportados	Comentario	Sitio Oficial.
Argus	GNU GPL	Linux, MAC, Windows NT	Soporta las versiones Netflow 1-8, actualmente se está trabajando en la lectura de datos de la versión 9. Sin embargo hay muchas diferencias entre los datos de argus y datos de Netflow, como son: ➢ Protocolo de soporte, ➢ Reportes, ➢ Precisión del tiempo, ➢ Tamaño de los registros, ➢ El estilo y el tipo de la métrica.	http://www.qosient.com/argus/argusnetflow.htm
Cflowd	Freeware	Linux	➢ Creado para recopilar y analizar la información obtenida disponible de los flujos en Netflow. Le permite al usuario almacenar la información. ➢ Solamente es utilizado como colector.	http://www.caida.org/tools/measurement/cflowd/
Nfdump y Nfsen	BSD	Linux	➢ Poderosa lectura, interpretación y análisis de datos por línea de comandos con Nfdump. ➢ Soporta las versiones 5, 7 y 9 de Netflow ➢ Ambiente web con la herramienta Nfsen, adaptada completamente para el soporte de Nfdump. ➢ El software Nfsen es capaz de crear alertas, acepta programación exterior además de la creación de filtros con el objetivo acotar la información.	http://nfdump.sourceforge.net/ http://nfsen.sourceforge.net/
EHNT	Freeware	Linux	➢ Solamente soporta la versión 5 de Netflow. ➢ Genera estadísticas de puertos tcp/udp. ➢ Solo disponible en modo texto.	http://www.networkuptime.com/tools/netflow/ehnt.html
F.L.A.V.I.O	GNU GPL	Linux	➢ Usado únicamente para graficar datos en formato Netflow.	http://www.networkuptime.com/tools/netflow/flavio.html
Flowd	BSD	Linux	➢ Recolecta rápidamente y de forma segura datos en formato Netflow. ➢ No puede almacenar flujos en múltiples formatos para realizar	http://www.mindrot.org/flowd.html

			análisis de datos. ➢ Solamente soporta versión 5.	
FlowScan	GNU GPL	Linux	➢ Analiza y reporta datos en formato Netflow mediante Cflow. ➢ Examina el flujo de datos y mantiene contadores reflejando lo que fue encontrado. ➢ Los valores del contador se almacenan con ayuda de RRDtool (BD utilizada sistemas cronológicos). ➢ Soporta la versión 5.	http://www.caida.org/tools/utilities/flowscan/
JNCA	Free	Linux, Windows	➢ JNCA (Java Netflow Collector and Analizer) es una solución a la administración de flujos de red basados en Netflow creada por java ➢ Diseñada para recolectar y analizar datos en formato Netflow.	http://www.networkuptime.com/tools/netflow/jnca.html
Ntop	GNU GPL	Windows, Linux	➢ Recolecta datos en formato Netflow. ➢ Genera tablas y gráficas. ➢ Utiliza el software Nscape como una interfaz web, sin embargo cuenta con una administración y configuración limitada.	http://www.ntop.org/
Silk	GNU GPL	Linux	➢ Creada por CERT. Encargada de coleccionar y analizar flujos de datos. ➢ Puede recolectar datos de Netflow v9 o v5.	http://tools.netsa.cert.org/silk/
Stager	GNU GPL	Linux	➢ Recolecta datos en formato Netflow basándose en el colector Nfdump. ➢ Utiliza Posgret SQL. ➢ Crea reportes, genera tablas y gráficas, ambiente web.	http://software.uninett.no/stager/

La información mostrada en la tabla 2.1 fue de vital importancia para elegir a los posibles candidatos capaces de sustituir al software comercial Netflow utilizado. La elección de las posibles alternativas, fue resultado de un análisis entre cada una de las alternativas presentadas, descartando aquel software que ofrecía un funcionamiento menor en comparación a los demás. Como resultado de la investigación y análisis realizado, los tres candidatos elegidos fueron:

1. Nfdump y Nfsen.
2. Flow Scan.
3. Stager.

El siguiente paso fue realizar una investigación acerca del funcionamiento, requerimientos, colector utilizado, última versión, etc., de las tres alternativas elegidas. Como resultado de la investigación realizada se obtuvo la tabla 1.2, esta tabla fue de vital importancia para efectuar un análisis entre cada software: observar cuál o cuáles mostraban un mejor comportamiento y descartar a aquel(los) que mostrara(n) un funcionamiento menor.

La investigación realizada hacia las tres posibles alternativas consistió en una búsqueda efectuada sobre las páginas WEB y foros de cada software, arrojando los siguientes resultados:

Tabla 2.2 Comparación general entre posibles alternativas

Software	NFSEN	FlowScan	Stager
Licencia	BSD	GNU GPL	GNU GPL
S.O.	Linux	Linux	Linux
Última Versión	1.3.4 04/07/2010	1.0.1 28/02/2001	4.0.1 12/01/2010
Versiones de Netflow soportadas.	V5, V7, V9	V5	V5,V7,V9
Colector	Nfdump	Cflow	NFdump
Ambiente WEB	Sí	Sí	Sí
B.D.	Archivos o RRDTool	RRDTool	Posgret SQL
Reportes	Sí	No	Sí
Graficas	Sí	Sí	Sí
Foros	Sí	No	Sí

Por medio del análisis realizado a la tabla seis, se observó que el software FlowScan tiene un desempeño menor en comparación con las otras dos alternativas. Esteproblema fue de vital importancia para la eliminación del software mencionado, además de encontrar los siguientes inconvenientes:

- Al momento de observar que la última versión del software Flow Scan fue creada en el año 2001, se adquirió demasiada desconfianza en la investigación efectuada hacia este software. Esto ocasionó dudas acerca del correcto funcionamiento del software. Por ejemplo: si contaba con un mantenimiento adecuado, funcionamiento óptimo, foros actualizados, etc. Las sospechas fueron comprobadas al encontrar escasa información del software y su funcionamiento (prácticamente solo se contaba con la información proporcionada por la página web del software).
- La interfaz web del software es muy limitada; su mejor funcionamiento es mediante línea de comandos.
- El encontrar escasa información del colector utilizado por el software Flow Scan, Cflow, representaría dificultades al tener un problema en el funcionamiento del software Flow Scan y tratar de resolverlo.

Como siguiente paso en la investigación se realizó una búsqueda, análisis y comparación detallada entre el software Nfsen y el software Stager.

2.3 Nfsen vs Stager.

El objetivo de realizar un análisis detallado entre las dos alternativas es observar cual mostraba un mejor funcionamiento, las ventajas y desventajas que presenta cada uno de ellos, su comportamiento una vez instalado, etc. Los puntos principales que debía contener la investigación fueron los siguientes:

- Requerimientos necesarios para su instalación
- Funcionamiento
- Colector Utilizado
- Documentación
- Versiones
- Foros
- Interfaz web

Todos estos factores se obtuvieron de la página web oficial de cada software, así como de investigaciones realizadas fuera de su página web. Cabe mencionar que las dos alternativas finales satisfacían los requerimientos planteados. Por este motivo se decidió observar el funcionamiento de cada software de manera detallada y así poder concluir cuál de ellos se adaptaba mejor a las necesidades requeridas.

Las dos alterativas fueron instaladas en un S.O. Centos 5.5 virtualizado en "VmWare Server 2.0". El servidor de pruebas utilizado tiene instalado un S.O Windows Server 2003; Dicho servidor está configurado con una IP clase C estática pertenece a una red de usuarios. La máquina virtual creada fue puenteada (modo bridge) hacia la misma red de usuarios asociándole una IP estática clase C.

La figura 2.1 muestra el esquema utilizado en el proceso de pruebas.

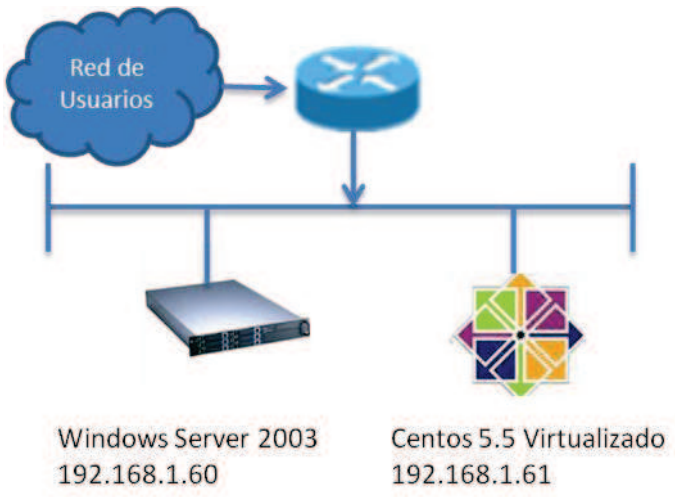


Figura 2.1 Esquema de Pruebas utilizado

La tabla 2.3 muestra una comparación realizada entre el software Nfsen y Stager. Resultado de la investigación realizada.

Tabla 2.3 Comparación entre Nfsen y Stager

Software	Nfsen	Stager
Requisitos para su instalación	➢ S.O. Linux ➢ Servidor Web ➢ Perl y PHP o Perl > 5.6.0 o PHP > 4.1 ➢ Módulos de perl. o Mail:: Header, Mail:: Internet ➢ Herramientas RRD o Todos los gráficos Netflow en NfSen requieren RRD. Por lo menos se requiere el módulo de Perl RRDs ➢ Herramientas Nfdump o Necesarias para recoger y procesar datos de Netflow o Instalar la versión 1.5.8	➢ S.O Linux o BSD ➢ Nfdump ➢ Perl o DBI (Modulo de interfaz de BD) ➢ PostgreSQL ➢ Servidor Http ➢ Php o Smarty: template engine o JpGraph: graph creating library
Colector Utilizado	Nfdump	Nfdump
Descripción General	El software Nfsen surgió como una aplicación creada con el objetivo de proporcionar una interfaz web al colector Nfdump. Por este motivo presenta una excelente compatibilidad con el software Nfdump. Además de lo mencionado también añadieron nuevas funcionalidades en el software Nfsen: como la creación de alertas, profiles y plugins (logrando realizar acciones específicas enfocadas hacia el monitoreo de red y análisis de datos)	Stager utiliza el colector Nfdump. Los autores del software decidieron enfocarlo más hacia las bases de datos, a tal modo que lograron implementar una base de datos en donde se guarda toda la información capturada por Nfdump. Sin embargo no implementaron todas las funcionalidades disponibles en el softwareNfdump.
Comparación entre graficas realizadas	Nfsen utiliza un software dedicado especialmente a graficar (RRDtool). Los desarrolladores enfocaron el software a la actualización de las gráficas en forma continua. Sin embargo, se tiene el inconveniente de solo tener un único formato de gráfica.	Stager utiliza jpGraph para la creación de sus gráficas. Esta herramienta permite crear gráficas en tercera dimensión y la elección de múltiples formatos para su creación. Sin embargo, las gráficas creadas solo muestran información general y no son actualizadas periódicamente.
Interfaz Web.	Nfsen presenta en su página de inicio gráficas basadas en el profile live (profile creado por default que contiene toda la información de los colectores instalados), divide la información en 12 gráficas; cada una de ellas grafica la cantidad de flujos, paquetes y bits que circulan en la red diario, semanalmente, al mes y al año. La interfaz web cuenta con un menú sencillo y potente que permite	Stager presenta en su página de inicio un conjunto de opciones relacionadas a seleccionar alguna interfaz y crear un reporte. Al realizar esta opción es posible observar las tablas creadas y graficarlas en caso que lo requeríamos. La interfaz web es de fácil utilización para el usuario.

	moverte fácilmente hacia cualquier opción del software, haciendo muy fácil el navegar sobre él.	
Procesamiento de datos	Nfsen tanto en sus gráficas como en tablas muestra la cantidad de flujos, paquetes y bits que circulan enla red. Se puede realizar un análisis más detallado mediante la aplicación de filtros, acotando la información a estadísticas muy específicas. Nfsen actualiza la información cada cinco minutos, logrando un monitoreo constante y la posibilidad de detectar en tiempo real alguna anomalía presentada.	Al igual que en Nfsen, Stager muestra la información de acuerdo a los flujos, bits y paquetes que circulan en la red. Sin embargo, como valor mínimo muestra lo que paso por la red cada hora y no permite realizar un análisis sobre los datos presentes.
Comentarios acerca de su funcionamiento	Nfsen al momento de ser implementado ha demostrado tener un buen funcionamiento, por los siguientes motivos: ➢ Las gráficas creadas se actualizan constantemente. ➢ Se observa información en sus tablas creadas referente a las gráficas ➢ Recolección de datos cada cinco minutos ➢ Creación de alertas y notificación inmediata sobre anomalías encontradas ➢ Capacidad de crear filtros específicos con el objetivo de acotar la información ➢ Capacidad de crear puntos de observación específicos sobre lo que está pasando a través de la red (profiles) ➢ Permitir ejecutar programación exterior al software mediante plugins.	Al momento de la instalación del software Stager se encontraron algunos errores de programación, esto causo dudas acerca de su funcionamiento. Una vez realizada la investigación acerca de los errores presentados y su resolución, se observó que solo está limitado a la representación de la información obtenida en tablas y gráficas. A comparación del software Nfsen, en Stager no es posible realizar un análisis de los datos obtenidos, ni el separar los datos o crear vistas específicas de ellos.
Conclusión	El software Nfsen mostró un mejor comportamiento en la fase de pruebas, debido a que además de su función principal la interpretación en ambiente web de los datos en formato Netflow proporcionados por el colector Nfdump, permite lo siguiente: ➢ Crear vistas específicas sobre los datos. ➢ La generación de alertas y notificación al detectar algo anormal ➢ El uso de filtros para observar información detallada. ➢ Gráficas de comparación por horas, día, semana, mes tanto de flujos, paquetes y bits. ➢ Posibilidad de implementar programación exterior sobre él.	El software Stager cumple con la función de recolectar, almacenar e interpretar los datos provenientes de exportadores de Netflow. Utiliza el colector Nfdump únicamente para la interpretación de los datos en formato Netflow. No aprovecha todo el potencial del colector mencionado y no proporciona mayores funcionalidades.

2.4 Elección de la alternativa Open Source

Como se observa en la tabla siete el software Nfsen mostró un mejor comportamiento que el software Stager. En la última prueba realizada antes de la elección, se decidió agregar un equipo exportador a Nfsen para observar su funcionamiento en tiempo real. Obteniendo lo siguiente:

- Al momento de agregar el dispositivo exportador al software Nfsen, se observó la actualización de las gráficas cada cinco minutos; en las tablas creadas se muestran las estadísticas obtenidas correspondientes al valor específico observado en la gráfica.
- Al probar la opción de filtros se observó el gran potencial que tiene para realizar análisis detallados y obtener información muy específica.
- Se comprobó que Nfsen fue desarrollado para mostrar en ambiente web todos los datos y opciones presentadas en línea de comandos por el software Nfdump. Cabe mencionar la gran facilidad proporcionada por Nfsen para generar comandos Nfdump y el gran potencial que tiene al aplicar filtros.
- La capacidad de crear puntos de vista específicos (profiles), y la creación de gráficas y tablas relacionadas con los datos aplicados sobre el/los filtros específicos, como puede ser:
  - Monitorear la utilización de puertos bien conocidos.
  - Monitorear la utilización de redes específicas.
  - Monitorear el tráfico sobre los protocolos TCP, UDP, ICMP, entre otras.
- La utilización de alertas para monitorear posibles anomalías que se presenten en nuestra red.
- La utilización de programación exterior (plugins) en el software Nfsen.

Por los motivos mencionados y como resultado de la investigación se decidió implementar el software Nfsen en la institución.

En el anexo B se muestra la guía de instalación del software mencionado.A continuación se muestran las características generales de la implementación realizada.

Sistema Operativo	Centos 5.5 (arquitectura de 32 bits)
Versión Nfdump Instalada	1.6.1
Versión Nfsen Instalada	1.3.2
Virtualización	Vmware Server 2.0
Dependencias Instaladas	➢ Servidor apache 2.0, habilitado con autenticación de usuarios ➢ Perl 5.6.0 ➢ Php 5.0 ➢ Modulos de perl: ▪ Mail:: Header ▪ Mail:: Internet ➢ RRD tool 1.4 ➢ Nfdump 1.5.8

La última sección en este capítulo pretende realizar una comparación entre el software libre implementado, Nfsen, y el software comercial Netflow utilizado por la institución.

2.5 Comparación entre versión comercial de Netlfow utilizada y Nfsen

En la tabla 2.4 se muestra una comparación realizada entre el software comercial Netlfow y el software libre Nfsen.

Tabla 2.4 Comparación entre la versión comercial de NetFlow utilizad y el software Nfsen

Software	Versión comercial de Netflow	Nfsen
S.O. Soportado	Windows XP, Vista, 7, Server 2003 o 2008	Linux
Tipo de Licenciamiento	Requiere licencia de uso de software.	BSD
Costo	Miles de dólares: depende de exportadores instalados e interfaces a monitorear.	Costo de Licencia **$0.00**: costos mínimos de capacitación a usuarios no familiarizados con el sistema operativo Centos.
Recursos Utilizados	La versión comercial de Netflow utilizada, por ser un software tan complejo consume demasiados recursos en el servidor instalado, en ocasiones llega a funcionar lento debido a las potentes funcionalidades desarrolladas sobre él.	Nfsen consume una mínima cantidad de recursos en el servidor instalado. Es óptimo para instalarse en computadoras con bajos recursos. Su funcionamiento es rápido en comparación del software comercial utilizado.
Descripción General	La versión comercial de Netlfow utilizada es un software enfocado hacia el monitoreo de red, presenta una excelente solución al analizar lo que está presente en la red, así como diversas funcionalidades.	El software Nfsen surgió como una aplicación creada con el objetivo de proporcionar una interfaz web al colector Nfdump. Por este motivo presenta una excelente compatibilidad con el software Nfdump.
Interfaz Web	La interfaz web del software comercial utilizado, es muy potente y de fácil utilización para el usuario. Además trae incorporado un manual de usuario y las gráficas creadas son en 2D o 3D.	La interfaz web del software Nfsen es más limitada en comparación con la versión comercial. Sin embargo es de fácil utilización. Además trae una pequeña explicación sobre las diferentes funcionalidades del software
Procesamiento de datos	El software comercial Netlfow utilizado, permite separar la información automáticamente en las redes presentes en el exportador. Además permite realizar análisis muy detallados (background) de lo que está presente en la red. Las gráficas creadas son actualizadas cada que el usuario lo desee (previa configuración), además se pueden observar datos en un lapso de tiempo específico. Los exportadores son agregados directamente sobre la interfaz web de una manera fácil y ágil. Además es posible mediante archivos el agregar redes de datos de forma automática.	Nfsen no separa la información automáticamente. Sin embargo con la ejecución de profiles es posible agrupar la información de igual forma que el software Netlfow comercial lo realiza. También es posible la realización de un background por medio de filtros e ir obteniendo cada vez información más específica. Las gráficas creadas se actualizan cada cinco minutos y es posible observar datos en un lapso de tiempo específico. Los exportadores y plugins creados en Nfsen tienen que ser agregados de forma manual en su archivo de configuración. Sin embargo una vez agregados presentan un comportamiento bastante eficaz sobre el software.
Soporte	Por ser software propietario, la versión comercial de Netflow	Nfsen no cuenta con soporte proporcionado directamente a sus

	utilizada cuenta con soporte técnico y de implementación proporcionado directamente por el fabricante. Sin embargo, solo está limitado al plazo de la licencia adquirida.	clientes. Sin embargo, es posible resolver problemas presentados preguntando en foros.
Funcionamiento	El software comercial NetFlow presenta un buen funcionamiento en términos generales. Se incorporaron a él diversas herramientas como son: generación de líneas bases, costo del ancho de banda consumido, generación de reportes muy precisos, notificación sobre anomalías encontradas entre otros. Sin embargo, en algunos casos no logra clasificar al 100% las redes de algún exportador específico y su uso se vuelve lento.	El software Nfsen no presenta funcionalidades tan complejas como la versión comercial de NetFlow. Sin embargo, es posible implementar en él, mediante la opción de plugins las funcionalidades realizadas por la versión comercial de Netlfow u otras que sean requeridas. Además tiene incorporada una opción de alertas basadas en filtros: mediante un análisis basado en filtros o plugins se logra encontrar tráfico anormal, haciendo muy potente el software.
Beneficios	La versión comercial de NetFlow utilizada, presenta los siguientes beneficios: ✓ Soporte prestado por profesionales las 24 horas del día. ✓ Costo de la licencia de acuerdo a las necesidades requeridas ✓ Compromiso de correcto funcionamiento. ✓ Fácil Instalación. ✓ Interfaz web muy potente y de fácil uso. ✓ Monitoreo de red potente, además de diversas funcionalidades complementarias. ✓ Creación de reportes muy potentes. ✓ Muy poca intervención del usuario.	Nfsen presenta los siguientes beneficios: ✓ Foro creado para la resolución de problemas presentes sobre el software. ✓ Software Libre. ✓ Observar el código fuente y hacer mejoras sobre él. ✓ Interfaz web limitada, pero de fácil utilización para el usuario. ✓ El monitoreo de red puede llegar a ser tan potente como el usuario lo requiera. ✓ Capacidad de agregar programación exterior (plugins) que satisfagan las necesidades del usuario. ✓ Una vez configurado de acuerdo a las necesidades requeridas, realiza los procesos de forma automática.
Conclusión	Por las características descritas, el software NetFlow comercial muestra ser muy robusto y potente. Sin embargo, se tiene el inconveniente que su licencia es de un alto costo, tomando en cuenta número de colectores instalados e interfaces monitoreadas.	A comparación con la versión comercial de NetFlow utilizada, Nfsen no es tan robusto. Sin embargo está diseñado para aprovechar al máximo sus funcionalidades. Puede llegar a ser tan robusto e inclusive superar al software comercial Netflow por medio de plugins, profiles y filtros enfocados a las necesidades requeridas.

Como se observa en la tabla ocho, la versión comercial de NetFlow utilizada proporciona un mejor desempeño que el software libre Nfsen. Sin embargo, una de las principales ventajas de Nfsen, además de ser libre y observar su código fuente, es el poder crear plugins enfocados hacia las necesidades requeridas. Esto ofrece un gran potencial al software Nfsen, a tal grado de igualar las funcionalidades desempeñadas por la versión comercial de NetFlow utilizada o incluso superarlas.

Cabe mencionar que el software Nfsen superó exitosamente la fase de pruebas y hasta el momento de publicación de este proyecto de tesis, no ha presentado errores en su funcionamiento. Logrando con esto el objetivo de encontrar una alternativa de uso libre capaz de poder sustituir al software Netflow comercial utilizado en la institución.

En conclusión, de todas las funcionalidades mostradas por el software Nfdump. Recomiendo el uso de este software frente a otros colectores por los siguientes motivos:

- Software Libre.
- Soporta las versiones 5, 7 y 9 de Netflow.
- Colector potente y configurable de acuerdo a las necesidades requeridas.
- Contabiliza el ancho de banda consumido.
- La herramienta nfdump es capaz de realizar análisis muy detallados.
- Pude ser enfocado hacia la seguridad informática.
- Muestra picos presentes sobre las redes o host que consumen un mayor BW (análisis mediante top n)
- Posibilidad de esconder la información.
- Filtros muy poderosos y de una sintaxis fácil.

CAPÍTULO III

Implementación del protocolo Neflow y del software “Listry-AIGC”

3.1 Introducción

Netflow, como se describió en el capítulo uno, es un protocolo de monitoreo de red creado por Cisco que parcialmente se ha convertido en un estándar en el uso de esta tecnología. Actualmente otras marcas soportan el protocolo Netflow en sus dispositivos activos con algunas variantes, logrando un crecimiento considerable en el uso de este protocolo para el monitoreo de redes.

Nfsen, software libre elegido del proceso de investigación realizado en el capítulo dos, es un software encargado de mostrar todos los datos capturados por Nfdump en una interfaz web amigable para el usuario; además los creadores de Nfsen añadieron diversas utilidades al software, como son:

- Graficas
- Alertas
- Proceso de *background* mediante el uso de filtros.
- Observación específica sobre los datos capturados (profiles)
- Posibilidad de añadir programación exterior al software(creación de plugins)

En este capítulo se explica la implementación realizada del protocolo Netflow en la institución, así como el funcionamiento del software Nfdump y Nfsen.

3.2 Implementación del protocolo Netflow

Antes de habilitar el protocolo Netflow, fue necesario conocer el esquema de red que se tiene implementado en la institución. La figura 3.1 muestra el esquema general de la red implementado en la institución.

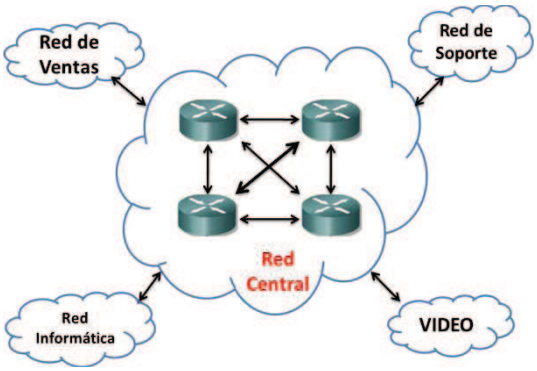


Figura 3.1 Esquema de red General

Como se observa en esta figura, se tienen cinco redes principales habilitadas: La red mostrada en el centro (Red Central) es la red principal de la institución. Esta red se encarga de las siguientes actividades:

- Interconexión con otras redes.
- Administración de redes de usuarios y servidores.
- Soporte a la operación, basado en las capas 2-4 del modelo OSI.

El esquema de red mostrado es aplicado en seis áreas de operación; se ha habilitado el protocolo Netflow en seis dispositivos activos asociados a un edificio específico. La figura 3.2

muestra el esquema de implementación del protocolo Netflow en la Red Central de la institución.

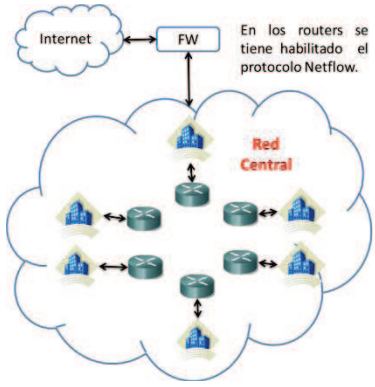


Figura 3.2 Habilitación de Netflow sobre RI

Como se observa en esta figura, cada router presente en la Red Central se encarga de proporcionar los servicios de red a un edificio en específico. En cada edificio se tienen creadas redes de usuarios, redes de servidores, redes de telefonía y otras redes de acuerdo al esquema de red mostrado en la figura ocho. Al habilitar el protocolo Netflow en estos routers se pretende tener un monitoreo constante (24*7) [Se pretende realizar un monitoreo 24 horas * 7 días a la semana, los 365 días del año], enfocado a observar las actividades realizadas en cada edificio, especialmente se observará el tráfico que circula en las redes de usuarios y servidores.

La figura 3.3 muestra las redes que tendrán una mayor observación mediante el monitoreo de red.

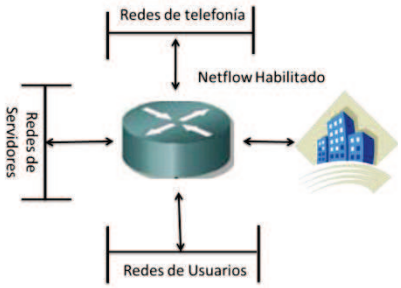


Figura 3.3 Redes a monitorear sobre los edificios.

Aunque se tiene implementado un esquema de seguridad robusto en la institución, las redes mencionadas son susceptibles a algún ataque realizado en ellas por usuarios internos o externos, esto ha motivado a que por medio del monitoreo de red, se realicé un algoritmo capaz de buscar anomalías presentes en estas redes, y que notifique inmediatamente al encontrar algún comportamiento inusual.

La estrategia del plugin creado en el software Nfsen, se describe en el capítulo IV, en la siguiente sección se describe como fue implementado el software “Listry-AIGC”.

3.3 Implementación del software "Listry-AIGC"

3.3.1 ¿Qué es el software Listry-AIGC?

El software "Listry-AIGC" es un conjunto de módulos instalados, enfocados en dos principales actividades.

- En el monitoreo de red.
- En la detección de malware que deja evidencia en la red y opera en capas inferiores del modelo OSI.

Los módulos y herramientas instalados en el software "Listry-AIGC", permiten al administrador de red mayor seguridad y comodidad al utilizar este software. Debido a que proporciona todas sus funcionalidades en forma gráfica. Los módulos y herramientas que incluye este software son los siguientes:

- Habilitación del protocolo Hypertext Transfer Protocol Secure (HTTPS).
- Instalación y configuración de MySQL.
- Instalación y configuración del software OpenWebmail.
- Instalación y configuración del software Navicat.
- Instalación y configuración del colector Nfdump.
- Instalación y configuración del software Nfsen.
- Configuración del plugin "escaneo" en el software Nfsen.

El software "Listry-AIGC" fue instalado en un S.O. Centos 5.5 virtualizado en VmWare server 2.0.

Los seis dispositivos activos configurados en el software Nfsen, envían los export packet generados por cada uno de ellos hacia el colector Nfdump cada cinco minutos, mediante el software Nfsen se logran visualizar los datos presentes en una interfaz web.

Además, para que el router funcione como un dispositivo exportador, es necesario activar el protocolo Netflow de la siguiente forma:

1. Entrar al router y acceder al modo privilegiado
   - router$ enable
2. Entrar al modo de configuración
   - router# configure terminal
3. Entrar a la interfaz en donde se habilitara Netflow (repetir del paso 3 al 5 si se pretende activarlo en todas las interfaces)
   - Router(config)# interface GigabitEthernet x/x
   - Donde:
     - x/x = Numero de la interfaz del router
4. Se habilita la recolección de flujos
   - Router(config-if)# ip route-cache flow
5. Regresar a modo de configuración
   - Router(config-if)# exit
6. Redirigir los paquetes UDP hacia el colector
   - Router(config)# ip flow-export destination yyy.yyy.yyy.yyy zzzzz
   - Donde:

- yyy.yyy.yyy.yyy = Ip del colector
- zzzzz = Puerto UDP

7. Configurar la interfaz de donde se enviaran los datos (repetir el paso siete si se enviaran datos de todas las interfaces)
   - Router(config)# ip flow-export source GigabitEthernet x/x
8. Elegir la versión del protocolo Neflow utilizada
   - Router(config)# ip flow-export version 9
   - NOTA: Si la v9 no es compatible con el router, utilizar la versión v5.
9. Opcional: Configurar timeout
   - Router(config)# ip flow-cache timeout active 1
   - Router(config)# ip flow-cache timeout inactive 15
10. Salir del modo de configuración
    - Router(config)# ctrl + Z
11. Guardar los cambios
    - Router# write mem

La configuración puede variar dependiendo del dispositivo activo donde se habilitará Netflow, sin embargo, se muestra la configuración general del protocolo Netflow en un equipo Cisco.

La figura 3.4 muestra el esquema general de envío de los export packets hacia el dispositivo colector.

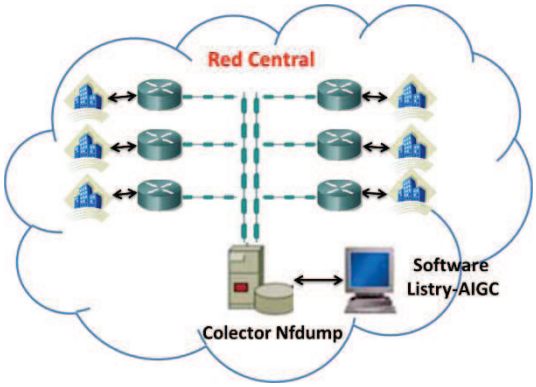


Figura 3.4. Envío de export packets hacia el colector.

Para poder utilizar al software "Listry-AIGC", como una alternativa enfocada a sustituir al software Netflow comercial, e implementarse en la institución. Se sometió dicho software a las siguientes etapas:

1. Investigación: En esta etapa se investigó acerca del funcionamiento del software, Nfsen, los requisitos necesarios para su instalación y su funcionamiento (1 mes).
2. Instalación: En esta etapa se instaló el software Nfsen mediante el proceso qua ya se ha descrito (1 semana).
3. Pruebas: En esta etapa al software Nfsen se le asignó una dirección IP perteneciente a un laboratorio de pruebas dentro de Red Central, además se asoció un dispositivo activo con el objetivo de monitorear las redes del edificio asociado a este dispositivo las 24 horas del día. Esto se realizó con el objetivo observar el comportamiento del Software Nfsen; la respuesta ofrecida y para verificar algún error en su funcionamiento antes de pasarlo a la etapa de producción (6 meses)

4. Implementación/Producción. Una vez superada exitosamente la fase de pruebas se configuraron los cinco routers faltantes para que enviaran los Export Packets generados hacia el colector Nfdump, como se observa en la figura 3.5. Además se asignó al software Nfsen una IP perteneciente a una red de servidores, completando con esto la fase de implementación/Producción.

En este capítulo se explica minuciosamente el funcionamiento del software Nfdump y Nfsen. La descripción de los demás módulos instalados en el software "Listry-AIGC", se encuentra en el glosario C "Guía de usuario del software Listry-AIGC".

El software Nfsen realiza monitoreo de las actividades presentes en las redes de usuarios y servidores las 24 horas del día. Durante la fase de pruebas, el software Nfsen mostro un óptimo funcionamiento debido a que recibía los Export Packet generados por un router cada cinco minutos (un paquete recibido contenía en promedio 300 flujos), además de explotar las diversas funcionalidades ofrecidas por Nfsen, en beneficio de la institución, las cuales son:

- Creación de Profiles: Se crearon profiles enfocados en el monitoreo de redes de usuarios y redes de servidores, monitoreo de servicios y monitoreo de conexiones realizadas hacia servidores;
- Alertas: Se crearon alertas enfocadas en la detección de picos presentes en la red de la institución y;
- Plugin: Se creó un plugin enfocado en la detección de malware mediante patrones típicos de comportamiento.

3.3.2 Nfdump Definición

Nfdump es un software regido bajo licenciamiento BSD, que se encarga de recolectar e interpretar datos en formato Netflow provenientes de dispositivos exportadores. Actualmente el software Nfdump se encuentra en la versión 1.5.8 y soporta las versiones 5, 7 y 9 del protocolo Netflow.

El software Nfdump tiene un conjunto de herramientas para capturar y procesar datos en formato Netflow en línea de comandos, las cuales son:

- nfcapd: Demonio que captura los datos en formato Netflow.
- nfdump: Aplicación creada para procesar el conjunto de datos en formato Netflow capturados (ficheros generados por nfcapd).
- nfprofile: Filtra los datos en formato Netflow guardados en función de los perfiles definidos (profiles).
- nfreplay: Lee los datos en formato Netflow guardados en ficheros por nfcapd y los reenvía a otro equipo.
- nfclean.pl: Script para borrar datos antiguos.

Aunque hay multitud de herramientas creadas para leer, procesar y representar datos en formato Netflow tanto en línea de comandos como en formato gráfico, pocas de ellas tienen la flexibilidad y potencia ofrecida por el software Nfdump a la hora de procesar los datos. Está herramienta puede ser muy útil en:

- La detección de ataques hacia la red.

- Generación de reportes.
- Realizar análisis forenses.
- Obtener diversas estadísticas del uso de: BW, puertos, redes, servicios, etc.

La figura 3.5 muestra el funcionamiendo del software Nfdump [http://nfdump.sourceforge.net/]:

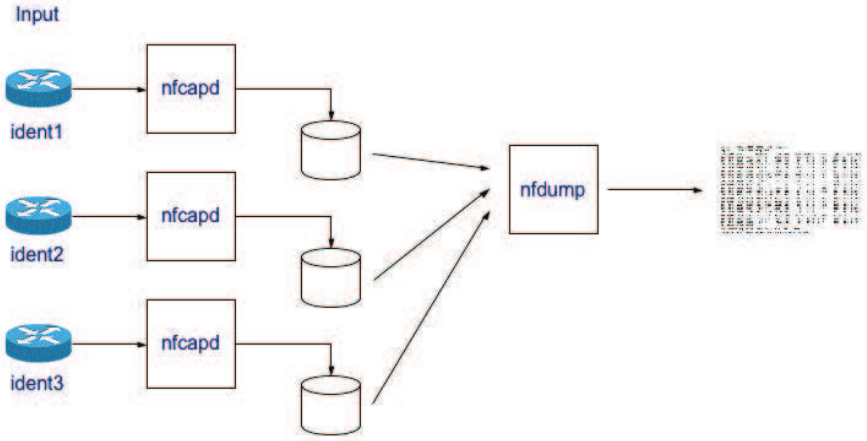


Figura 3.5 Funcionamiento nfdump

El comportamiento del software Nfdump se rige de la siguiente forma:

- El software Nfdump permite asociar uno o varios dispositivos exportadores que tienen habilitado el protocolo Netflow. Nfdump asocia a cada exportador un demonio nfcapd que se encarga de escuchar sobre un puerto UDP específico.
- El demonio nfcapd escucha y captura todos los Export Packets provenientes del dispositivo exportador asociados a él.
- Los Export Packets son enviados cada cinco minutos por el dispositivo exportador hacia el demonio Nfcapd que se ha asociado (se puede configurar el tiempo de envió de Export Packets). El colector Nfdump se encarga de guardar los Export Packets recibidos en archivos nfcapd (o en bases de datos, depende de la configuración realizada en la instalación del software Nfdump) que únicamente pueden ser interpretados por la herramienta nfdump.
- La herramienta nfdump se encarga de interpretar y mostrar toda la información guardada por nfcpad en texto claro. Además sobre esta herramienta se pueden aplicar filtros muy específicos con el objetivo de realizar un análisis detallado sobre los datos presentes.

3.3.2.1 Nfcapd funcionamiento

Nfcpad es el demonio encargado de recolectar los datos provenientes de los exportadores y almacenarlos cada cinco minutos en un archivo con el siguiente formato:

nfcapd.AAAAMMDDHHMINMIN (ej.: nfcapd.201008291420)

Estos archivos solo pueden ser interpretados por la herramienta nfdump y contienen información acerca de los flujos capturados por el exportador que los ha generado.

En la figura 3.6 se muestran algunos archivos capturados; estos archivos se generan automáticamente cada cinco minutos y contienen los flujos almacenados por el equipo exportador.

```
[root@localhost ll]# ls
nfcapd.201003110000  nfcapd.201003110155  nfcapd.201003110350  nfcapd.201003110545  nfcapd.201003110740
nfcapd.201003110005  nfcapd.201003110200  nfcapd.201003110355  nfcapd.201003110550  nfcapd.201003110745
nfcapd.201003110010  nfcapd.201003110205  nfcapd.201003110400  nfcapd.201003110555  nfcapd.201003110750
nfcapd.201003110015  nfcapd.201003110210  nfcapd.201003110405  nfcapd.201003110600  nfcapd.201003110755
nfcapd.201003110020  nfcapd.201003110215  nfcapd.201003110410  nfcapd.201003110605  nfcapd.201003110800
nfcapd.201003110025  nfcapd.201003110220  nfcapd.201003110415  nfcapd.201003110610  nfcapd.201003110805
```

Figura 3.6 Archivos nfcpad.

Al momento de añadir los equipos exportadores en el archivo de configuración del software Nfsen, Este software se encargará de configurar automáticamente el demonio nfcapd asociado al dispositivo exportador

Si es necesario configurar manualmente el demonio nfcapd, se realiza de la siguiente forma:

#nfcapd -w -D -l /flow_base_dir/exportador -p 23456

Dónde:

- W: Tiempo de rotación de los archivos (5 minutos por defecto)
- D: Background (Demonio)
- l: Directorio de salida
- Flow_base_dir: Directorio que contiene los archivos capturados
- Exportador: Nombre del equipo a observar

Para mayor información sobre las banderas utilizadas por nfcapd, teclear sobre shell:

#nfcapd –h

3.3.2.2 El intérprete nfdump

La herramienta nfdump sirve como un intérprete de los datos capturados por Nfcapd. Sus funciones principales son las siguientes.

- Ver el contenido de uno o varios archivos nfcpad.
- Realizar un análisis sobre los archivos seleccionados por medio de filtros basados en expresiones regulares.
- Visualizar el resultado de forma clara para el usuario o guardarlo en un archivo nfcapd.

El funcionamiento de la herramienta nfdump se muestra en la figura 3.7.

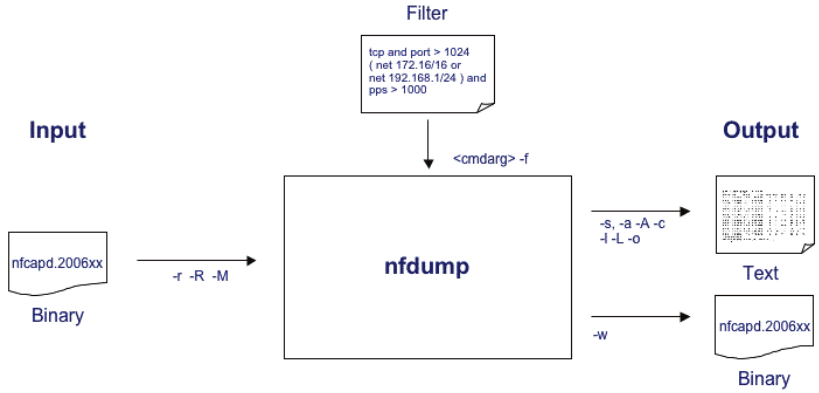


Figura 3.7 Procesamiento de datos Nfdump.

El procesamiento de datos mediante la herramienta nfdump se basa en:

- La lectura de archivos capturados por nfcapd mediante –r, -R o –M dependiendo del tipo de lectura:
  - -r: Solo lee un archivo;
  - -R: Lee un conjunto de archivos que se encuentran en el mismo directorio (/dir:file1:file2);
  - -M: Lee archivos desde múltiples directorios (/dir/dir1:dir2:dir3)
- El realizar un análisis detallado mediante la aplicación de filtros, acotando la información.
  - -z 'Filtro a aplicar';
  - Consultar el anexo C para mayor información de la sintaxis de los filtros.
- El modo de salida que puede ser:
  - Mostrar los datos en pantalla (-S –a –A –c –l –L –o);
  - Guardar los datos en un nuevo archivo (-w).

Las opciones descritas son las banderas básicas utilizadas por la herramienta nfdump. Cabe mencionar lo potente que es el software Nfdump en la realización de análisis background, en la detección de picos presentes sobre las redes, saber qué Ip's consumen mayor ancho de banda y en análisis enfocados sobre seguridad informática. Para mayor información sobre otras banderas teclear en Shell

```
# nfdump -h
```

3.3.2.3 Ejemplos de la herramienta nfdump.

En esta sección se proporcionan tres ejemplos sobre el uso de la herramienta nfdump.

1. Leer los datos de un archivo generado con la fecha 12-Marzo-2010 15:20

Solución: `# nfdump –r nfcapd.201003121520`

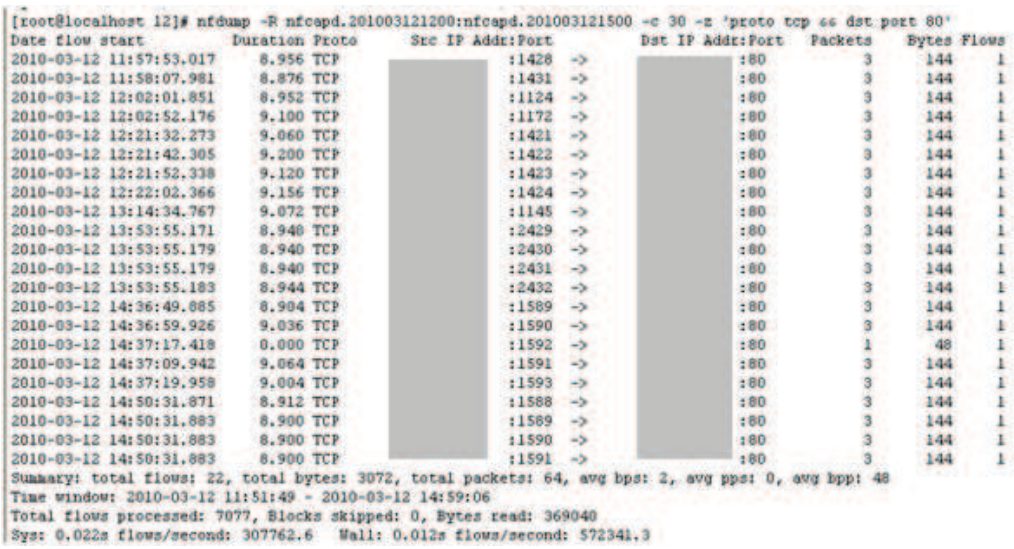
Nota: Se tendrá que estar situado en el directorio donde se encuentra el archivo o hacer referencia a él.

```
[root@localhost 12]# nfdump -r nfcapd.201003121520
Date flow start          Duration Proto     Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
2010-03-12 15:14:07.256     3.000 UDP      [redacted]           ->   [redacted]               2      656     1
2010-03-12 15:14:07.268     3.000 UDP                           ->                            2      656     1
2010-03-12 15:14:15.460     0.000 UDP                           ->                            1       87     1
2010-03-12 15:14:24.060     0.000 UDP                           ->                            1       68     1
2010-03-12 15:14:24.424     0.000 UDP                           ->                            1       68     1
2010-03-12 15:14:28.064     0.000 UDP                           ->                            1       68     1
2010-03-12 15:14:28.428     0.000 UDP                           ->                            1       68     1
2010-03-12 15:14:30.268     0.000 UDP                           ->                            1      229     1
2010-03-12 15:10:01.560   299.244 UDP                           ->                         3078   332424     1
2010-03-12 15:14:36.988     0.000 UDP                           ->                            1       84     1
2010-03-12 15:14:41.476     0.000 UDP                           ->                            1       87     1
2010-03-12 15:14:35.992     9.000 TCP                           ->                            3      152     1
2010-03-12 15:14:50.376     0.000 UDP                           ->                            1      229     1
2010-03-12 15:14:42.084     8.948 TCP                           ->                            3      144     1
2010-03-12 15:14:55.940     0.000 ICMP                          ->                            1       92     1
2010-03-12 15:14:45.316    14.504 UDP                           ->                            2      400     1
2010-03-12 15:15:02.105     0.000 UDP                           ->                            1       68     1
2010-03-12 15:15:03.201     0.000 UDP                           ->                            1       68     1
2010-03-12 15:15:06.109     0.000 UDP                           ->                            1       68     1
2010-03-12 15:15:07.209     0.000 UDP                           ->                            1       66     1
2010-03-12 15:15:07.413     0.000 UDP                           ->                            1      576     1
2010-03-12 15:15:07.493     0.000 UDP                           ->                            1       87     1
```

Figura 3.8 Resultado del ejemplo 1

Leer los datos de flujos que se obtuvieron el 12-Marzo-2010 entre las 12:00 y 15:00, mostrando solo los 30 primeros flujos que presentaron actividad en el protocolo TCP y enviaron datos a través de http.

Solución: #nfdump -R nfcapd.201003121200:nfcapd.201003121500 -c 30 -z 'proto tcp && dst port 80'



```
[root@localhost 12]# nfdump -R nfcapd.201003121200:nfcapd.201003121500 -c 30 -z 'proto tcp && dst port 80'
Date flow start          Duration Proto      Src IP Addr:Port           Dst IP Addr:Port   Packets    Bytes Flows
2010-03-12 11:57:53.017     8.956 TCP                    :1428  ->                   :80         3      144     1
2010-03-12 11:58:07.981     8.876 TCP                    :1431  ->                   :80         3      144     1
2010-03-12 12:02:01.851     8.952 TCP                    :1124  ->                   :80         3      144     1
2010-03-12 12:02:52.176     9.100 TCP                    :1172  ->                   :80         3      144     1
2010-03-12 12:21:32.273     9.060 TCP                    :1421  ->                   :80         3      144     1
2010-03-12 12:21:42.305     9.200 TCP                    :1422  ->                   :80         3      144     1
2010-03-12 12:21:52.338     9.120 TCP                    :1423  ->                   :80         3      144     1
2010-03-12 12:22:02.366     9.156 TCP                    :1424  ->                   :80         3      144     1
2010-03-12 13:14:34.767     9.072 TCP                    :1145  ->                   :80         3      144     1
2010-03-12 13:53:55.171     8.948 TCP                    :2429  ->                   :80         3      144     1
2010-03-12 13:53:55.179     8.940 TCP                    :2430  ->                   :80         3      144     1
2010-03-12 13:53:55.179     8.940 TCP                    :2431  ->                   :80         3      144     1
2010-03-12 13:53:55.183     8.944 TCP                    :2432  ->                   :80         3      144     1
2010-03-12 14:36:49.885     8.904 TCP                    :1589  ->                   :80         3      144     1
2010-03-12 14:36:59.926     9.036 TCP                    :1590  ->                   :80         3      144     1
2010-03-12 14:37:17.418     0.000 TCP                    :1592  ->                   :80         1       48     1
2010-03-12 14:37:09.942     9.064 TCP                    :1591  ->                   :80         3      144     1
2010-03-12 14:37:19.958     9.004 TCP                    :1593  ->                   :80         3      144     1
2010-03-12 14:50:31.871     8.912 TCP                    :1588  ->                   :80         3      144     1
2010-03-12 14:50:31.883     8.900 TCP                    :1589  ->                   :80         3      144     1
2010-03-12 14:50:31.883     8.900 TCP                    :1590  ->                   :80         3      144     1
2010-03-12 14:50:31.883     8.900 TCP                    :1591  ->                   :80         3      144     1
Summary: total flows: 22, total bytes: 3072, total packets: 64, avg bps: 2, avg pps: 0, avg bpp: 48
Time window: 2010-03-12 11:51:49 - 2010-03-12 14:59:06
Total flows processed: 7077, Blocks skipped: 0, Bytes read: 369040
Sys: 0.022s flows/second: 307762.6   Wall: 0.012s flows/second: 572341.3
```

Figura 3.9 Resultado del ejemplo 2

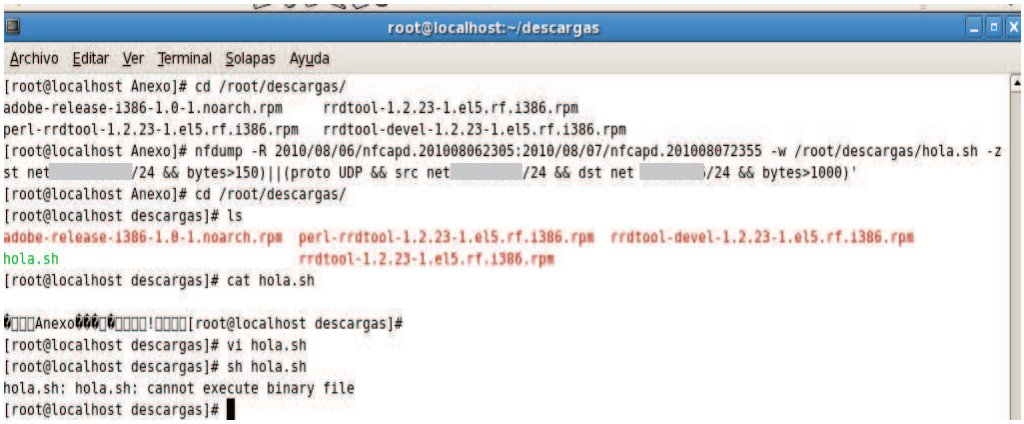
2. Leer los datos en el siguiente periodo de tiempo 06-Agosto-2010 23:05 a 07-Agosto-2010 23:55, limitándolo a 200 flujos con las siguientes condiciones:
    - Únicamente bandera habilitada de inicio de sesión sobre la subred x.x.x/24 con un tamaño mayor a 150 bytes.
    - O todo lo que pasa a través de UDP de la subred z.z.z/24 hacia y.y.y/24 con un tamaño mayor a 1000 bytes
    - El resultado guardarlo en /root/descargas con el nombre de 'hola.sh'

Solución:

#nfdump -R 2010/08/06/nfcapd.201008062305:2010/08/07/nfcapd.201008072355 -w /root/descargas/hola.sh -c 50 –z '(flags S && not flags APF && dst net x.x.x/24 && bytes >150) || (proto UDP && src net z.z.z/24 && dst net y.y.y/24 && bytes >1000)'

El resultado del comando aplicado será la generación del archivo "*hola.sh*"; este archivo creado solo podrá ser observado mediante nfdump: Si se quiere observar su contenido mediante algún editor de textos (o los comandos, vi, cat, more, etc.), la información contenida en este archivo será mostrada en forma ilegible hacia el usuario. De igual manera si se desea ejecutar el archivo "*hola.sh*" (por medio de sh ó ./), el shell notificara que no es un archivo ejecutable.

En la figura 3.10 se observa que no es posible acceder al contenido del archivo hola.sh por los medios tradicionales:



Al ejecutar Vi, la salida fue la siguiente:

Figura 3.10 Esconder datos con ayuda de nfdump

Como se observa la ejecución del comando nfdump crea un archivo disfrazado e ilegible con otros lectores de archivos. Destaqué esta funcionalidad de nfdump debido a que me parece muy útil como una técnica de *esteganografía*.

Al ejecutar el archivo “hola.sh” mediante nfdump, se podrá observar toda la información que tiene nuestro archivo disfrazado. El resultado se observa en la figura dieciséis.

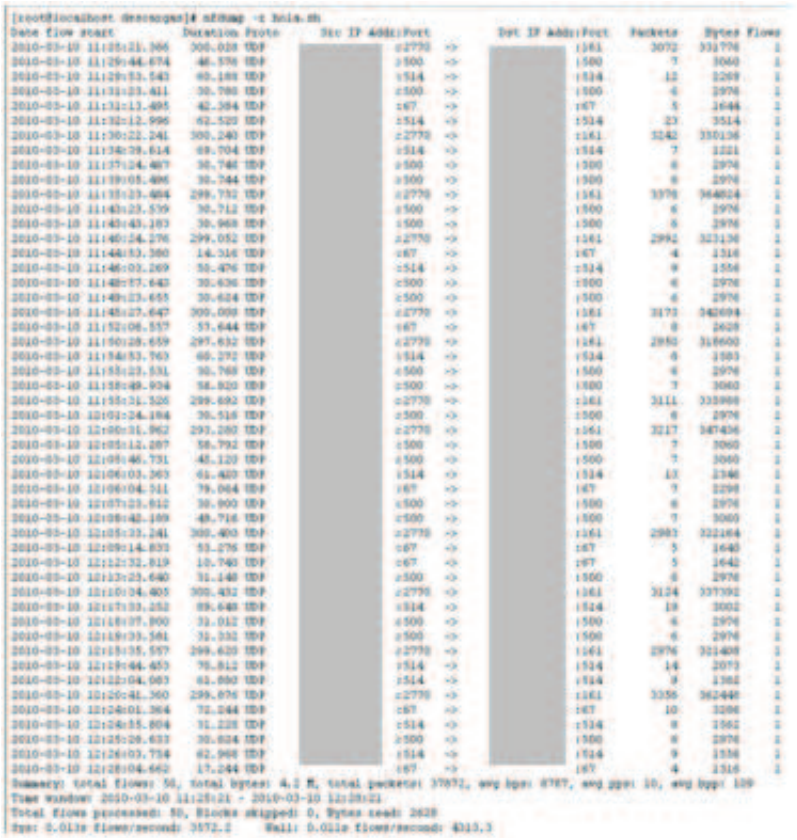


Figura 3.11 Salida del archivo hola.sh

3.3.3 Nfsen *Definición y funcionamiento*

El software Nfsen tiene el objetivo de proporcionar una interfaz gráfica al software Nfdump, además, Nfsen cuenta con las siguientes características:

- Muestra los datos almacenados por el demonio Nfcapd en graficas desglosadas de acuerdo a flujos, paquetes y el tráfico generado en los protocolos TCP, UDP, ICMP.
- Proporciona un ambiente web de fácil utilización.
- Permite observar y procesar los datos Netflow en lapsos de tiempo específicos.
- Actualización de las gráficas cada cinco minutos.
- Permite aplicar filtros para observar información acotada (utiliza la misma sintaxis soportada por Nfdump).
- Permite la creación de vistas específicas de los datos en formato Netflow (profiles) que pueden ser históricos o continuos.
- Permite la creación de alertas basadas en varias condiciones o en la ejecución de alertas mediante algún plugin.
- Permite añadir plugins para procesar datos de Netflow de acuerdo a las necesidades requeridas

En el anexo C “Guía de usuario Nfsen” se explica a detalle la configuración y utilización del software Nfsen.

3.3.3.1 Funcionamiento de Nfsen

El software Nfsen se encarga de graficar todos los datos obtenidos por el colector Nfdump, además de permitirnos opciones adicionales, como se ha descrito anteriormente. En la figura 3.12 se muestra el funcionamiento general del software Nfsen.

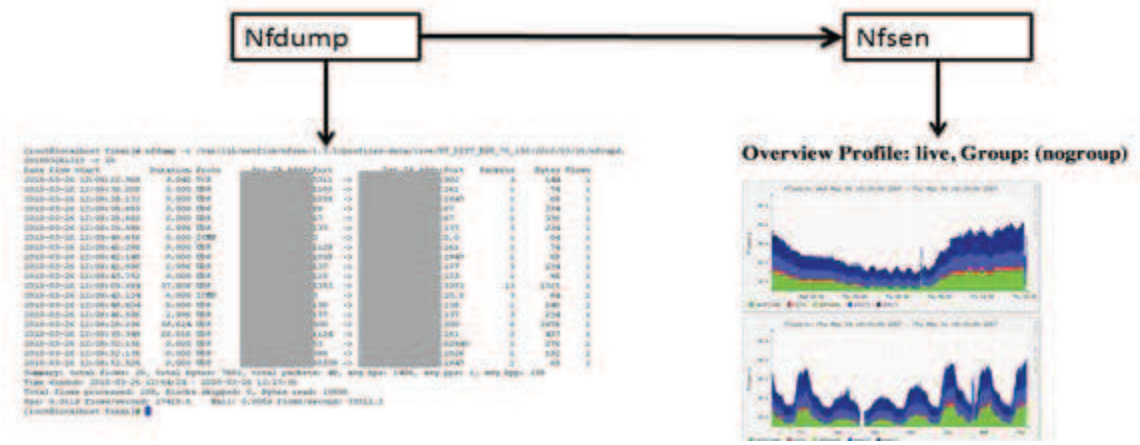


Figura 3.12 Funcionamiento nfsen

Cada archivo nfcapd obtenido por el colector Nfdump será mostrado y clasificado en las gráficas creadas por Nfsen automáticamente. Al desplazarse sobre una gráfica se muestra la fecha que hace referencia al punto específico observado; al dar clic en el Export Packet donde se encuentre posicionado el mouse se observará una tabla que contiene características generales de la cantidad de flujos, paquetes y tráfico presente en ese lapso de tiempo específico, además de permitir la aplicación de filtros para la realización de análisis y observar los datos del mismo modo que los presenta la herramienta nfdump.

Nfsen provee una gran flexibilidad en la visualización de los datos debido a que podemos observarlos de manera gráfica o en modo texto (mediante la herramienta nfdump). Además en sus tablas creadas se muestra el promedio de flow/s, paquetes/s y trafico/s de los protocolos TCP, UDP, ICMP y otros, asi como el tráfico consumido por estos mismos.

La figura 3.13 muestra un ejemplo de los Export Packets obtenidos en comparación con el tiempo y el tráfico consumido, tanto para una sola captura (archivo de 5 minutos) como para un lapso de tiempo en específico.

Figura 3.13 Presentación de datos

Desglosando esta imagen se observa un monitoreo constante del dispositivo exportador: las imágenes mostradas fueron obtenidas en el mismo lapso de tiempo; la diferencia entre ambas radica en observar en la imagen de la izquierda las estadísticas generadas por un solo Export Packet (archivo nfcapd.201003252345), mientras que en la imagen de la derecha se muestran las estadísticas generadas por un conjunto de Export Packets (archivos nfcapd 201003252220 al nfcapd 201003260255).

El poder observar un conjunto de Export Packets es de gran utilidad si se desea realizar un análisis sobre un lapso de tiempo específico, donde se haya presentado un aumento en el BW inusual al tráfico promedio obtenido. Además las tablas mostradas contienen la suma de todos los Export Packets observados en el lapso de tiempo específico y estas tablas son desglosadas de la manera descrita anteriormente.

El análisis sobre los datos en formato Netflow se realiza por medio de la aplicación de filtros (bajo la sintaxis descrita en el anexo C) y las diversas opciones incorporadas en el software Nfsen. Esto proporciona una gran potencia al software Nfsen en la investigación de eventos ocurridos, además de la posible detección de malware presente sobre la red y la pasible detección de malware de día cero.

3.3.3.2 Profiles

Como se explica en el anexo C Nfsen permite crear vistas personalizadas (profiles) con el objetivo de realizar observaciones específicas sobre los datos. Nfsen permite crear dos tipos de profiles:

- *Históricos:* Observación de datos en el pasado; se establece un punto inicial y un punto final para la observación de los datos.

- *Continuos:* Se empiezan a observar los datos en el pasado pero se continúan actualizando conforme se obtienen nuevos datos (el profile se actualizará cada cinco minutos).

La figura 3.14 se muestra un profile creado para monitorear redes de interés, tomando en cuenta la red origen y la red destino. El profile creado es de tipo continuo.

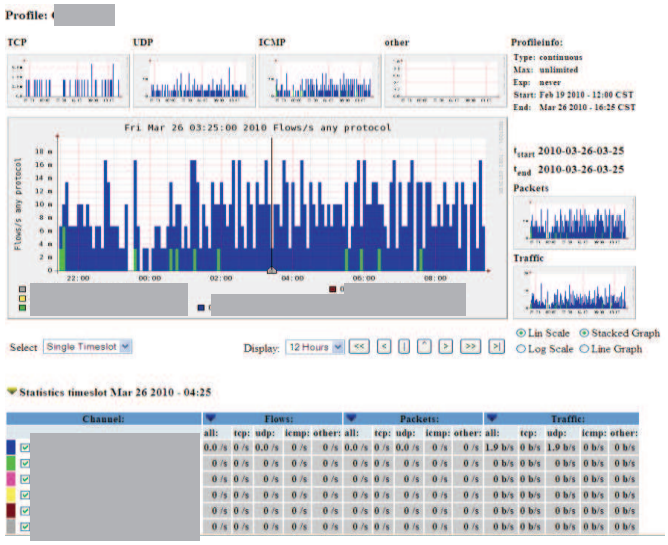


Figura 3.14 Ejemplo profile.

Desglosando esta figura, se observa que el profile creado separa las redes contenidas en el dispositivo exportador en los rangos establecidos en los filtros. A cada canal creado en el profile se le asignaron diversos filtros, logrando como resultado la gráfica mostrada que contiene las redes separadas. Cada color mostrado en la gráfica es resultado de una dirección IP perteneciente a una red de usuarios que adquiere un servicio proporcionado por una dirección IP perteneciente red de servidores.

Los profiles son de gran uso en el software Nfsen, debido a que permiten separar la información de acuerdo a las necesidades requeridas, por ejemplo:

- Observar el tráfico generado únicamente por redes de usuarios.
- Observar la utilización de puertos conocidos.
- Observar las conexiones realizadas hacia redes de servidores.
- Clasificar las conexiones realzadas en los puertos, conocidos, reservados y privados.
- Clasificar las conexiones realizadas de redes de usuarios hacia puertos bien conocidos.
- Entre otros.

3.3.3.3 Alertas

Otra característica a destacar del software Nfsen es la generación de alertas. Esta utilidad es de gran importancia debido a que permite mediante la aplicación de filtros, o algún plugin ejecutado en la alerta el detectar patrones anormales presentes en la red. Como puede ser:

- Detección de malware sobre la red.
- Aumento en el tráfico sin razón aparente.
- Host que consumen un mayor BW al asignado.
- Envío de información hacia redes no permitidas

- Utilización elevada de puertos conocidos, privados o reservados
- Entre otros.

Para mayor información acerca de la creación de una alerta y los estados de ejecución, consultar el anexo C.

En la figura 3.16 se muestra una alerta llamada conexión. Esta alerta monitorea las conexiones realizadas por cualquier cliente perteneciente a una red de usuarios y para su ejecución será necesario que se cumplan con las siguientes condiciones:

- Los flujos con los valores más altos (top 1) que excedan de 2 bits.
- Los paquetes con los valores más altos (top 1) que excedan de 10 Kb
- El tráfico con los valores más altos (top 1) que exceda de 10 Kb

Para que se ejecute y envíe un correo electrónico notificando, será necesario que todas sus condiciones sean verdaderas después de cinco ciclos (25 minutos).

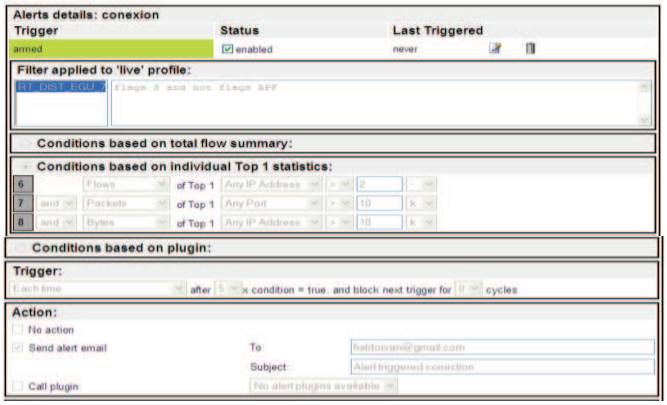


Figura 3.15 Creación de alerta.

La figura 3.15 tiene el objetivo de mostrar la configuración requerida para la creación de la alerta. En esta figura se observa un recuadro de color verde con la leyenda “armed”: Su significado es que la alerta se encuentra en ejecución, buscando que las condiciones programadas sobre ella se cumplan.

La figura 3.16 muestra la alerta creada en ejecución. En esta figura se observa una gráfica que contiene información acerca del promedio obtenido de los Export Packets (en el último ciclo; el promedio de 10 y 30 minutos anteriores; el promedio de 1, 4, 12 y 24 hora(s) anteriores, etc.). Además se observan dos tablas indicando lo siguiente:

- En la primera tabla se indica el valor (numérico) del promedio calculado por la alerta, tanto para los flujos, paquetes y bytes obtenidos en los lapsos de tiempo mencionados anteriormente.
- En la segunda tabla se observa el valor (booleano) devuelto por las condiciones creadas. El software Nfsen realiza una comparación AND sobre los resultados obtenidos de las condiciones; solamente la alerta se activará o pasará a otro estado de ejecución cuando todas las condiciones sean verdaderas.

Las alertas se actualizan cada cinco minutos, al igual que todas las funcionalidades del software Nfsen. En el ejemplo mostrado será necesario que la alerta se active cinco veces de forma consecutiva para cambiar al estado “fired” y que se ejecute.

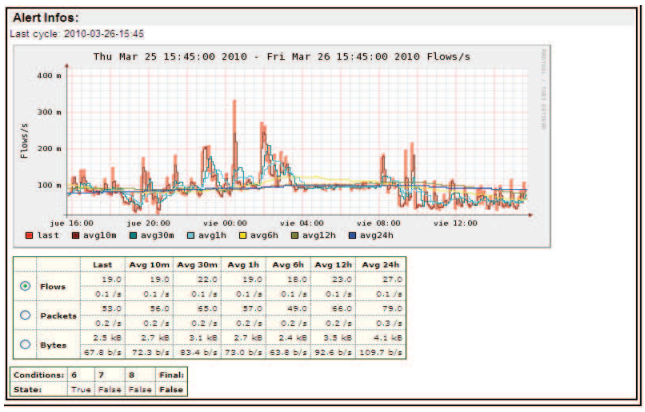


Figura 3.16 Ejecución de la alerta.

Las alertas pueden ser programadas para la ejecución de algún plugin con el objetivo de analizar por qué se presentó ese comportamiento anormal, brindando una mayor potencia al software Nfsen.

3.3.3.4 Plugins.

Los plugins son una potente herramienta utilizada por el software Nfsen, que permiten añadir programación exterior de acuerdo a las necesidades requeridas. Por medio de plugins creados en Nfsen se logra hacer a este software tan potente como se desee; además de enfocarlo hacia análisis de seguridad, poner precio al tráfico consumido, generar gráficas muy detalladas, entre otras cosas.

Nfsen soporta dos tipos de plugins:

- Backend: Módulos creados en perl con el objetivo de realizar acciones requeridas por el usuario y que no sean soportadas por Nfsen. Los plugin creados pueden ser enfocados de las siguientes maneras:
  - Condiciones de ejecución sobre alertas: Programación creada con el objetivo de cumplir condiciones creadas por el usuario y que no son soportadas directamente por Nfsen. Ejemplo: Monitorear si el ancho de banda se encuentra fuera del límite calculado por una línea base (baseline).
  - Acción específica al ejecutarse alguna alerta: Cuando las condiciones de ejecución en alguna alerta se cumplen, Nfsen lanzara el plugin creado bajo esta condición**.** Generalmente los plugin creados bajo esta acción buscaran, por medio del algoritmo creado, la causa de ejecución del plugin.
  - Actualización constante sobre nfsen: El plugin bajo esta condición se ejecutará en cada actualización de Nfsen (5 minutos por defecto). Dependiendo de la programación creada en el plugin, este podrá analizar cada archivo nfcapd obtenido en la última actualización de Nfsen, o analizar un conjunto de archivos nfcapd. Esta opción es muy utilizada cuando se programan algoritmos con el objetivo de buscar actividades anormales en la

red, o algoritmos que generan gráficas específicas y deben de ser actualizadas constantemente.

- Combinación de los puntos mencionados. Un pluglin creado bajo esta condición puede ser una combinación de las tres opciones mencionadas. Generalmente es usado sobre alertas: Se deberán de cumplir las condiciones programadas sobre el plugin para la ejecución de la alerta, así como al ejecutarse la alerta se lanzará el mismo plugin para verificar porque se ha presentado este comportamiento.

- Frontend: Programación creada sobre PHP con el objetivo de mostrar los resultados obtenidos en la ejecución del plugin backend asociado a él. Por ejemplo: al crear un plugin backend llamado "filtro.pm", el plugin generará como resultado un archivo de texto que contendrá el resultado de filtros programados sobre él; el plugin "filtro.php" tendrá como principal objetivo mostrar el contenido del archivo creado, sobre la interfaz web en Nfsen.

En el anexo C se muestra la programación necesaria para enfocar al plugin sobre alguna acción descrita en Backend, o el crear un plugin Frontend.

Como se ha descrito los plugins son una excelente herramienta utilizada en Nfsen. Por medio de la creación de plugins el software Nfsen se puede volver muy robusto e inclusive igualar o superar el funcionamiento de alguna alternativa propietaria.

En este trabajo de tesis se ha creado un plugin enfocado en la detección de malware presente en la red. El plugin fue creado para que se ejecute en cada actualización del software Nfsen analizando el último archivo nfcapd obtenido en búsqueda de algún comportamiento anormal sobre la red típico de un malware. Además se creó un módulo frontend con el objetivo de mostrar los resultados obtenidos del análisis realizado por el módulo backend. En el capítulo IV se explicara a detalle el funcionamiento del plugin creado.

La figura 3.17 muestra un plugin simple creado con el objetivo de mostrar el contenido del último archivo nfcapd obtenido.

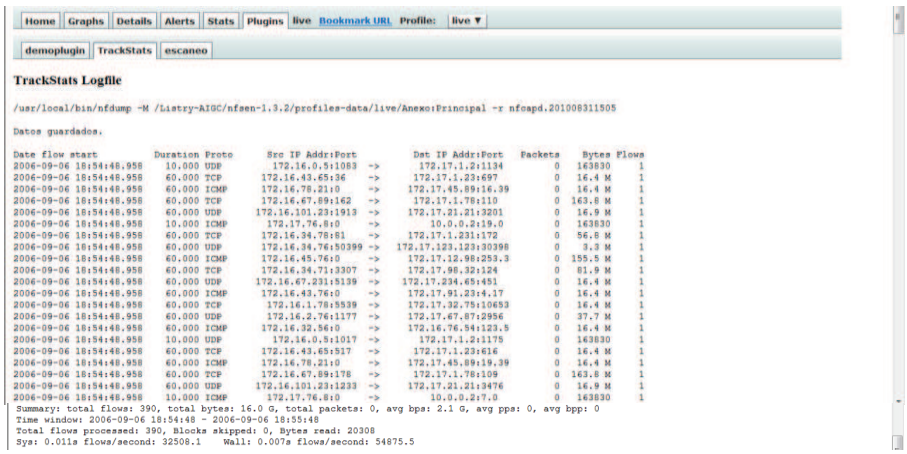


Figura 3.17 Ejecución de un plugin sobre nfsen.

Como se observa, los plugins pueden ser enfocados hacia cualquier funcionalidad requerida. Cada plugin creado en el software Nfsen permite añadir nuevas aplicaciones a este software,

enfocándolo hacia otras actividades diferentes al monitoreo de red, como puede ser la detección de malware presente la red, generación de gráficas de acuerdo con las necesidades, calcular costos por BW consumido, entre otros.

El plugin llamado "Trackstat", mostrado en la figura 3.17, se compone de los siguientes módulos:

- "Trackstat.pm": Módulo encargado de la obtención del último archivo nfcapd generado y guardarlo sobre un archivo de texto.
- "Trackstat.php": Módulo encargado de mostrar en la interfaz web el contenido del archivo creado por "Trackstat.pm".

El plugin "Trackstat" se ejecuta en cada actualización del software Nfsen, obteniendo el último archivo generado por nfcapd e interpretándolo.

En conclusión con todo lo mostrado acerca del funcionamiento del software Nfsen, se ha logrado cumplir con el objetivo de encontrar una alternativa Open Source capaz de sustituir en buena medida a software que requiere licenciamiento de uso. En este caso al software comercial utilizado en la institución Netflow. Además de ser la mejor alternativa software libre encontrada por los siguientes motivos.

- Software Libre.
- Soporta completamente el colector nfdump.
- Posibilidad de observar el código fuente y realizar mejoras sobre él.
- Foros web.
- Interfaz web amigable hacia el usuario.
- Las gráficas muestran los datos en lapsos de tiempo (12 horas, 1 día, 4 días, 1 semana, 1 mes; se pueden añadir más graficas con diferentes lapsos de tiempo).
- Las gráficas separan los datos de acuerdo a flujos, paquetes y tráfico contenido sobre los protocolos TCP, UDP, ICMP y otros.
- Actualización de datos constante.
- Posibilidad de realizar análisis sobre los datos presentes mediante filtros.
- Sobre cada Export Packet obtenido, se pueden observar las redes que consumen mayor ancho de banda (por medio de top n).
- Cataloga la información en base al ancho de banda consumido o el promedio con respecto al tiempo.
- Posibilidad de crear profiles.
- Creación de alertas y notificación.
- Creación de módulos compatibles con nfsen (plugins).
- Creación de alertas basadas en algoritmos (plugin enfocado hacia alertas).

Capítulo IV

Implementación del detector de malware

4.1 Introducción.

El malware, como se describió en el capítulo uno, es cualquier software dañino capaz de afectar a un equipo de diferentes formas. De acuerdo con la clasificación del malware descrita en el capítulo uno, los distintos tipos de malware pueden trabajar de forma individual o en conjunto para conseguir su objetivo.

A pesar de que existen herramientas de seguridad informática muy robustas basadas en la detección de malware por firmas digitales, algoritmos de comportamiento anormal, etc., los perpetradores explotan vulnerabilidades presentes en S.O. y crean ataques de día cero capaces de burlar la detección empleada por estas herramientas de seguridad. Este problema motivó a implementar una técnica de detección de malware diferente a las técnicas descritas. La técnica propuesta se basa en la detección de malware por patrones típicos de comportamiento referentes a capas del modelo OSI.

En este capítulo se explican las diversas técnicas utilizadas para la detección del malware. Además de explicar el desarrollo y funcionamiento del plugin, “escaneo”, creado para la detección de malware.

4.2 Estrategia de protección

Los conceptos de seguridad informática descritos en la sección 1.5 del capítulo 1 tienen el objetivo de proporcionar una visión general de lo que es la seguridad informática, los criterios usados para la clasificación de ataques y las buenas prácticas utilizadas, con el fin de poder desarrollar el detector de malware basado en el monitoreo de red eficazmente.

4.2.1 Medidas de protección.

Todo sistema de seguridad debe de contar con diferentes mecanismos de protección, como pueden ser antivirus, antispam, firewalls, IDS, IPS, entre otros. Entre mayores herramientas de seguridad se tengan implementadas se garantiza tener una seguridad más robusta. Sin embargo, ningún sistema es 100% infalible, por este motivo radican las buenas prácticas de los usuarios, el tener diferentes niveles de jerarquía en el acceso a la información, privilegios de acuerdo con el cargo realizado, herramientas de seguridad activadas las 24 horas del día, entre otras acciones.

Para proteger los activos o bienes es de gran utilidad plantearse las siguientes preguntas:

- ¿Qué se quiere proteger?
- ¿De qué se quiere proteger?
- ¿Cómo se quiere proteger?

El realizar un análisis con ayuda de estas preguntas garantiza una mejor aplicación de los mecanismos de seguridad en los bienes y activos a proteger. Estas tres preguntas fueron de gran utilidad en la creación del detector de malware:

- ¿Qué se quiere proteger? Se desea proteger la red central de la institución en donde esta implementado este proyecto de tesis, en especial se quiere proteger los activos y bienes de la institución que puedan generar pérdidas potenciales o una grave alteración en el funcionamiento de la red.

- ¿De qué se quiere proteger? De cualquier evento que presente un comportamiento anormal en la red de la institución. Especialmente de malware que deja evidencia en la red y sea observado mediante el monitoreo de red. Ya sea de malware ejecutado por usuarios internos o usuarios externos
- ¿Cómo se quiere proteger? Adicionalmente a las herramientas de seguridad (antivirus, antispam, firewalls, IDS, IPS, políticas de seguridad, entre otros) que ya se encuentran instaladas en la institución, se pretende integrar un nuevo mecanismo de seguridad basado en la detección de malware que genera comportamientos anormales en la red de la institución.

  Por medio del software Nfsen se creó un plugin que se ejecuta cada cinco minutos en búsqueda de comportamientos anormales descritos en la sección 4.3. Al detectar un comportamiento anormal se notifica inmediatamente vía email al administrador de red.

4.3 Comportamiento del malware.

Como se describió en el capítulo uno, el malware tiene una extensa clasificación y no es posible definir un comportamiento general sobre él. Sin embargo, por medio de investigaciones realizadas del comportamiento del malware se observa que generalmente presentan técnicas de infección y propagación comunes independientemente del tipo de malware. Todas estas técnicas fueron resultado de una investigación realizada acerca del comportamiento del malware en páginas orientadas a seguridad, como son:

- ✓ CERT
- ✓ SANS
- ✓ Insecure.org, entre otras.

Además de lectura de libros orientados a seguridad y búsqueda de información en internet.

A continuación se describirán las técnicas más comunes utilizadas por el malware para tratar de infectar a sus víctimas.

1) Tratar de pasar desapercibidos.

Uno de los principales objetivos de cualquier malware es pasar desapercibido. Debido a que si el usuario o la herramienta de seguridad no se dan cuenta de su presencia, el malware cumplirá con su objetivo sin ningún problema.

Un ejemplo típico de este comportamiento son los keylogger. Generalmente este malware es instalado por el usuario sin darse cuenta o por algún otro malware que ha abierto una *backdoor* en el sistema; el *keylogger* se encargará de recolectar la información tecleada por el usuario y enviarla a su creador. El usuario no se da cuenta de esta acción debido a que el ataque muestra un comportamiento pasivo y no causa daños al sistema. Sin embargo, este malware estará cumpliendo con su objetivo de recolectar y enviar información.

2) Tratar de infectar a otros equipos.

El malware que ha logrado infectar a un equipo, tratará de explotar la vulnerabilidad detectada en el equipo "víctima" sobre otros equipos pertenecientes a la misma red en el menor tiempo posible. Típicamente el malware que actúa bajo este comportamiento genera miles de conexiones en un tiempo pequeño, tratando de infectar a la mayor cantidad de equipos en el menor tiempo posible. Los métodos de infección más comunes son los siguientes.

Escaneo de puertos.

El malware que trate de infectar a diferentes equipos mediante un escaneo de puertos, presenta un comportamiento descrito de la siguiente forma:

- El malware que ha logrado infectar a una computadora con dirección IP *"xxx.xxx.xxx.xxx"*, buscara infectar a otra computadora perteneciente a la misma red con dirección IP *"yyy.yyy.yyy.yyy",* explotando la misma vulnerabilidad que encontró en el equipo víctima.
- Generalmente el malware que emplea esta técnica utilizará un puerto origen alto *"zzzz" y aleatorio*, tratando de infectar a una computadora con dirección IP *"yyy.yyy.yyy.yyy"* realizando un escaneo de puertos de forma secuencial hacia un puerto destino de la maquina víctima *"wwww"* de la siguiente forma:

xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww+1
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww+2
.
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyyy.yyy.yyy **:** wwww+n

Dónde: xxx.xxx.xxx.xxx = Ip origen zzzz: Pto Origen yyy.yyy.yyy.yyy = Ip destino wwww: Pto Destino

- Cuando el malware ha logrado infectar a una nueva víctima, repetirá el proceso descrito en otro equipo.

El escaneo de puertos que se realiza puede variar dependiendo del tipo de malware que utiliza esta técnica. Por medio de investigaciones en páginas enfocadas en seguridad como CERT, SANS, Insecure.org, entre otras, se observa que el malware frecuentemente realiza escaneo de puertos de forma secuencial y hacia puertos altos. Debido a que generalmente, los puertos altos no son muy utilizados y muchos usuarios tienen abiertos estos puertos.

Generalmente el malware que ejecuta la técnica de escaneo de puertos utiliza el protocolo orientado a conexión (TCP) realizando un "handshake". Sin embargo, algunos malware utilizan el protocolo no orientado a conexión (UDP) enviando paquetes y esperando la respuesta del equipo víctima, o utilizan el protocolo ICMP mediante un "echo request" (ping), buscando puertos libres.

Generalmente los gusanos es el malware que utiliza esta técnica descrita, sin embargo se puede presentar que otro tipo de malware haga uso de esta técnica por la capacidad de polimorfismo que tiene el malware. Algunos ejemplos de malware conocido que infectan a sus víctimas por medio de un escaneo de puertos son: Stumbler, Blaster, entre otros.

Escaneo de IP's.
El malware que trate de infectar a diferentes equipos mediante un escaneo de IP's presenta un comportamiento descrito de la siguiente forma:

- El malware que ha logrado infectar a una computadora con dirección IP *"xxx.xxx.xxx.xxx"*, buscará infectar a otra computadora perteneciente a la misma red o a otra red distinta con dirección IP *"yyy.yyy.yyy.yyy",* explotando la misma vulnerabilidad que encontró en el equipo víctima.
- Generalmente el malware que emplea esta técnica utiliza un puerto origen alto. Puede que en cada intento de infectar a una víctima varié el puerto origen utilizado. Por medio del socket establecido en la computadora infectada *"xxx.xxx.xxx.xxx:zzzz", el malware* tratará de infectar a una computadora con dirección IP *"yyy.yyy.yyy.yyy"* realizando un escaneo de IP's de forma secuencial en el tercer o cuarto octeto de la dirección IP de la víctima. El escaneo de IP's tiene como característica que el puerto destino *"wwww"* siempre será el mismo:

xxx.xxx.xxx.xxx **:**zzzzz**->** yyy.yyyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy.yyy+1 **:** wwww
.
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy.yyy+n **:** wwww

Ó

xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy+1.yyy **:** wwww
.
xxx.xxx.xxx.xxx **:** zzzzz **->** yyy.yyyy.yyy+n.yyy **:** wwww

Dónde: xxx.xxx.xxx.xxx = Ip origen zzzz: Pto Origen yyy.yyy.yyy.yyy = Ip destino wwww: Pto Destino

- Cuando el malware ha logrado infectar a una nueva víctima, repetirá el proceso descrito en otro equipo.

Al igual que en un escaneo de puertos, el protocolo utilizado por el malware al realizar un escaneo de IP's puede variar. Generalmente el malware que emplea esta técnica utiliza el protocolo TCP o el protocolo UDP.

Los gusanos es el malware que comúnmente utiliza la técnica de escaneo de IP's. Pero al igual que en un escaneo de puertos, puede que otro tipo de malware ocupe esta técnica. Algunos malware conocidos que conocidos que realizan un escaneo de IP's son: Conficker, Sasser, SQL Inyection, etc. Todos ellos tienen como característica el realizar escaneos de IP's de forma secuencial hacia un puerto destino igual en la víctima.

3) Consumir recursos.
El malware que presenta este comportamiento se caracteriza por tratar de consumir todos los recursos asignados en la computadora que ha infectado. Generalmente el malware genera cantidades de tráfico excesivas e impide que la víctima utilice eficazmente los recursos que se le han asignado. El comportamiento presentado por el malware que emplea esta técnica se describe de la siguiente forma:

- Generalmente el malware tratará de infectar a otras víctimas mediante un escaneo de puertos o escaneo de IP's.

- Cuando el malware ha logrado infectar a varios equipos, utilizará a todas las victimas para realizar un ataque de negación de servicios utilizando todo, o la mayor parte, del ancho de banda asignado a las computadoras zombies para enviar múltiples conexiones hacia la computadora víctima, generalmente servidor, con el objetivo de saturar y tirar la conexión de la computadora víctima.
- En el ataque DDoS realizado tendrá como objetivo enviar múltiples conexiones de máquinas infectadas hacia una dirección IP *"yyy.yyy.yyy.yyy"* y un puerto destino *"wwww"*, generando un tráfico excesivo. En las maquinas zombies puede haber variaciones en el uso del puerto origen de cada una de ellas y pueden pertenecer a una misma red o a redes distintas. El malware que emplea esta técnica actúa de la siguiente forma:

xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyy.yyy.yyy **:** wwww
.
xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyy.yyy.yyy **:** wwww
Dónde: xxx.xxx.xxx.xxx = Cualquier Ip origen zzzzz: Cualquier Puerto Origen
yyy.yyy.yyy.yyy = Ip destino wwww: Puerto Destino

Generalmente el malware que utiliza esta técnica son las bootnets. Este malware se encarga de infectar equipos y realizar ataques de negación de servicios (DoS) sobre la víctima. Las *botnets* se suelen instalar mediante *backdoors* o un gusano programado para que realice esta esa función específica. Sin embargo también hay virus y troyanos que emplean esta técnica.

Los ataques DoS o DDoS frecuentemente usan el protocolo TCP en su ejecución, aunque cierto malware también utiliza el protocolo UDP o aplican la técnica de "Ping of death" del puerto ICMP. La cantidad de tráfico generado por un ataque DoS es excesiva. El ataque DoS tiene como principal objetivo el saturar la máquina víctima y tirar, o interrumpir el servicio.

4) Actualización del malware y envió de información hacia direcciones Ip'S desconocidas por el usuario

La mayoría del malware una vez instalado en la victima, tratará de actualizarse constantemente, descargar nuevo malware o enviar información perteneciente de la máquina víctima a su creador. Hay que poner especial atención en este comportamiento si el malware ha infectado un servidor, debido a que puede haber fuga de información confidencial. El malware que utiliza esta técnica generalmente realiza las siguientes acciones:

- Actualizarse.
- Recolectar información de la maquina infectada con dirección IP *"xxx.xxx.xxx.xxx"* y enviarla hacia una dirección IP *"yyy.yyy.yyy.yyy"* utilizando puertos origen generalmente altos y de forma secuencial *"zzzz".* De la siguiente forma:

xxx.xxx.xxx.xxx **:** zzzz **->** yyy.yyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzz+1 **->** yyy.yyy.yyy.yyy **:** wwww
xxx.xxx.xxx.xxx **:** zzzz+2 **->** yyy.yyy.yyy.yyy **:** wwww
.
xxx.xxx.xxx.xxx **:** zzzz+3 **->** yyy.yyy.yyy.yyy **:** wwww
Dónde: xxx.xxx.xxx.xxx = Ip origen zzzz: Puerto Origen
yyy.yyy.yyy.yyy = Cualquier Ip destino wwww: Cualquier Puerto Destino

El puerto destino de las maquinas externas no es de gran importancia, debido a que el malware puede usar puertos destinos aleatorios. También hay que poner especial atención en el tráfico generado por cada conexión. A mayor cantidad de tráfico generado significa mayor fuga de información.

Normalmente el malware que utiliza esta técnica establece conexiones TCP hacia direcciones IP externas a la empresa y de rango sospechoso. Sin embargo, el malware también puede enviar información por medio del protocolo UDP.

Con base en las cuatro técnicas descritas, se pretende crear un plugin en el software Nfsen capaz de detectar malware que presente alguno de los comportamientos descritos en la red central de la institución. Hay que señalar que el objetivo del tema de tesis es detectar malware por medio del monitoreo de red, utilizando el protocolo Netflow, de tal modo que cualquier malware que no presente un comportamiento en la red o no cumpla con alguna de los cuatro métodos analizados no será detectado.

Sin embargo, el proyecto de tesis puede ser ampliado hacia la detección de cualquier tipo de malware o la creación de un IDS o IPS.

Para la creación de este plugin, fue necesario conocer el esquema de seguridad que se tiene implementado en la institución, este esquema sirvió como referencia para conocer las áreas más vulnerables en la institución.

4.4 Esquema de seguridad implementado en la institución

En la institución se tiene implementado un esquema de seguridad robusto; En cada red descrita se tienen implementadas diversas herramientas de seguridad, como lo son: antivirus, antispam, firewall, IDS, IPS, listas de control de acceso (ACL), iptables, entre otras. Además todo software nuevo a instalar en cualquier computadora deberá de ser verificado minuciosamente antes de su instalación. La figura 4.1 muestra a grandes rasgos el esquema de seguridad implementado.

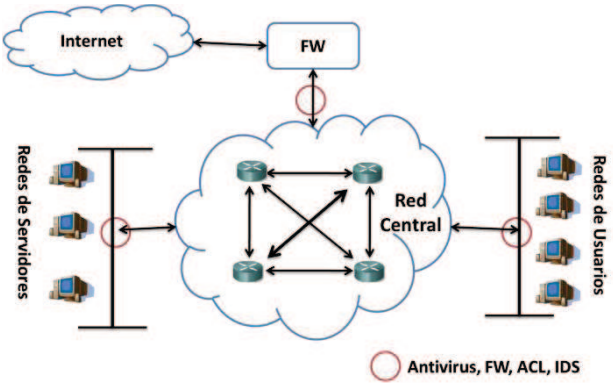


Figura 4.1 Esquema de seguridad implementado sobre la institución

Como se observa en esta imagen se hace énfasis en la protección de redes de usuarios y redes de servidores. Sin embargo, se puede presentar el caso de que un usuario accidentalmente o intencionalmente instale algún malware en la red central de la institución y este malware no sea detectado por las herramientas de seguridad instaladas, principalmente por ser un malware de día cero del cual no se tiene información alguna.

Por este motivo el objetivo del plugin a crear en el software Nfsen es trabajar en conjunto con las distintas herramientas de seguridad instaladas y la posible detección de malware de día cero, además con el desarrollo de este plugin se pretende una solución innovadora en la detección de malware.

4.5 Creación del plugin escaneo en Nfsen.

Como se describió en el capítulo tres, Nfsen permite crear módulos de programación adicionales al software y adaptados de acuerdo a las necesidades requeridas. En este proyecto de tesis se decidió programar un módulo enfocado en la detección de malware basado en los cuatro comportamientos descritos por las siguientes razones:

- La opción de alertas proporcionada por el software Nfsen permite acotar la información a rangos muy específicos, sin embargo no es posible al ejecutar la alerta, hacer referencia de las direcciones IP que se han detectado anormales, esto presenta un problema al tratar de analizar qué dirección IP es la causante de la anomalía detectada en la alerta.
- En los profiles soportados por Nfsen es posible crear un profile que observe un aumento en el tráfico anormal al promedio; por medio de análisis basados en filtros es posible obtener las direcciones IP que están causando este aumento inusual, sin embargo este procedimiento tiene que ser ejecutado manualmente por el usuario.

El objetivo de crear el plugin enfocado en la detección del malware es tener un módulo completamente automático y con una mínima intervención humana. El plugin creado se ejecutará cada que el software Nfsen reciba un nuevo archivo nfcapd. Este plugin se encargará de analizar el último archivo obtenido en búsqueda de los cuatro comportamientos del malware descritos. En caso de encontrar alguna anomalía, el plugin se encargará de notificar inmediatamente vía email de la anomalía detectada.

4.5.1 Estrategia de desarrollo del plugin escaneo.

El plugin enfocado en la detección de malware llamado *"escaneo"* será un módulo que formará parte del software Nfsen, encargándose de realizar las siguientes acciones en cada ejecución:

- Obtener e interpretar el último archivo nfcapd generado.
- Separar el contenido del archivo por medio de expresiones regulares.
- Verificar el resultado del punto anterior en búsqueda de patrones de comportamiento anormales, específicamente se buscará que en la red central se realice:
  - Escaneo de puertos,
  - Escaneo de IP's
  - Ataque de negación de servicios
  - Envió de información hacia el exterior.
- Generar el archivo de salida "anomalías" en caso de encontrar algún comportamiento anormal.
- Notificar inmediatamente vía email de la anomalía encontrada y guardar esta anomalía en una base de datos MySQL.
- Visualizar los resultados de la ejecución del plugin backend "escaneo.pm" Utilizando el módulo frontend "escaneo.php".

Los lenguajes de programación utilizados para la creación de este plugin fueron Perl y PHP, debido a que el software Nfsen solo permite la creación de módulos backend en perl y módulos frontend en PHP y HTML

4.5.2 Componentes del plugin Escaneo.

La tabla 4.1 tiene el objetivo de mostrar las funciones requeridas para la ejecución del plugin escaneo.

Tabla 4.1 Componentes del plugin escaneo

Plugin:	Escaneo.pm	Escaneo.php
Objetivo:	Ejecutar el plugin en cada actualización del software Nfsen en búsqueda de comportamientos anormales descritos en la sección 4.3	Mostrar los resultados de la ejecución del módulo "escaneo.pm" en la interfaz web del software Nfsen.
Lenguaje utilizado	Perl	Php
Funciones creadas	✓ Init ✓ Run ✓ Cleanup ✓ Separa ✓ Agrupa ✓ Agrupa_ip ✓ Guarda ✓ Envia_correo ✓ Epuertos ✓ Compara ✓ Eips ✓ Compara_Ip ✓ Exterior ✓ Compara_exterior ✓ DoS ✓ ComparaDos	✓ Escaneo_ParseInput ✓ Escaneo_run

Como se observa en esta tabla, el módulo escaneo.pm tiene como objetivo el analizar en cada actualización del software Nfsen el último archivo nfcapd generado por el colector Nfdump en búsqueda del comportamiento definido en la sección 4.3

Si en algún momento se llegase a detectar un comportamiento anormal que cumple con alguno de los puntos descritos se creará el archivo "anomalías" con el objetivo de guardar la información detectada como anormal y mandar llamar a la función envía_correo, notificando sobre el evento encontrado.

El módulo "escaneo.php" tiene como objetivo el mostrar los resultados de la ejecución del módulo "escaneo.pm" en la interfaz web del software Nfsen.

La descripción de cada función utilizada se mostrará mediante su diagrama de flujo específico y una breve explicación de las tareas ejecutadas por cada función.

El diagrama de flujo general de la ejecución del plugin escaneo se muestra en la figura 4.2.

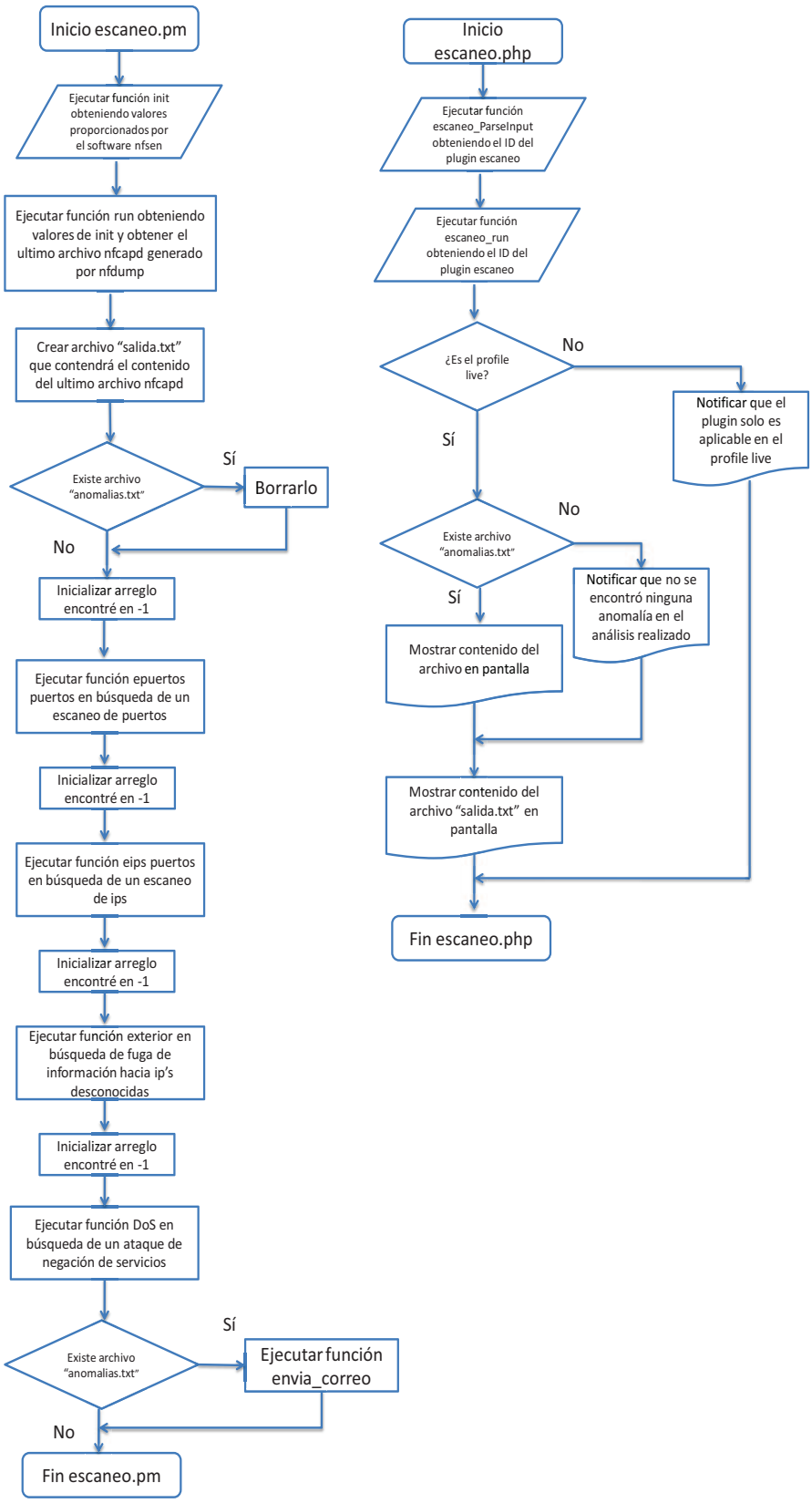


Figura 4.2 Diagrama de flujo general del plugin escaneo

4.5.3 Funcionamiento del plugin escaneo.

4.5.3.1 Funcionamiento del modulo *"escaneo.pm"*.

El objetivo del plugin backend 'escaneo.pm' es analizar el último archivo obtenido por el software Nfdump en búsqueda de patrones de comportamiento anormales presentes en la red central de la institución. Específicamente se buscará aquel malware que haya realizado un escaneo de puertos o un escaneo de IP's, o que esté enviando información mayor a 5 Mb de direcciones IP pertenecientes a redes de servidores hacia direcciones IP externas, o que realice ataques DoS o DDoS.

Las siguientes tres funciones son especiales en el uso de Nfsen. La figura 4.3 muestra el diagrama de flujo de cada una de ellas:

Init: Función especial creada en cualquier plugin a desarrollar en Nfsen. Esta función se encarga de contener toda la información necesaria para la ejecución de la función run.

Run: Función encargada de ejecutar el plugin en cada actualización del software Nfsen. Esta función recibe parámetros obtenidos de función init. Además se encarga de obtener el último archivo nfcapd generado, guardar este archivo en el arreglo "registros" y ejecutar a las funciones "epuertos, eips, DoS y exterior" en búsqueda de comportamientos anormales. Si la ejecución de las funciones mencionadas arroja un comportamiento anormal, se ejecuta a la función "envía_correo".

Cleanup: Función especial encargada de terminar la ejecución del plugin al momento de finalizar el software Nfsen.

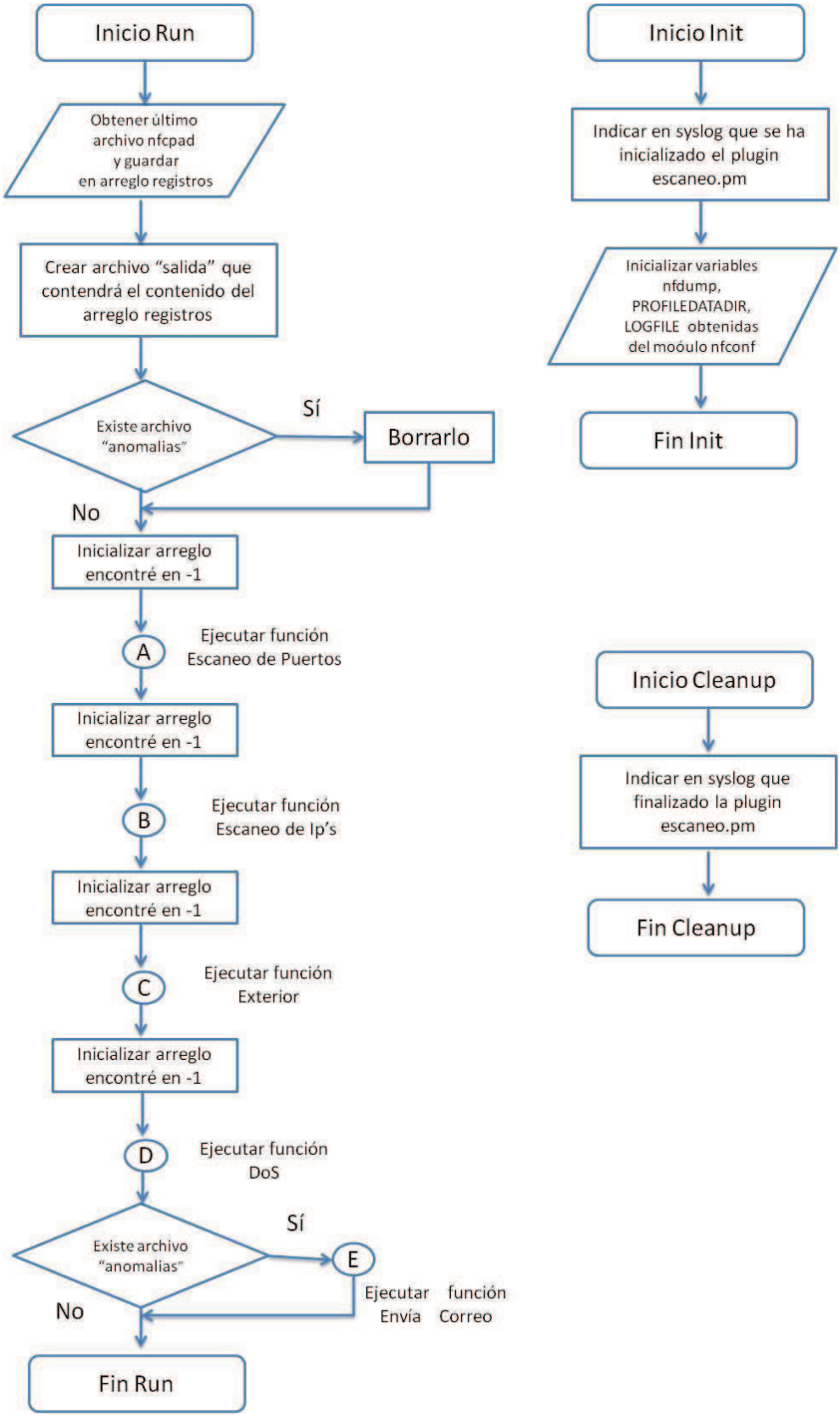


Figura 4.3 Diagrama de flujo de las funciones Init, Run y Cleanup

Las siguientes funciones fueron creadas con el objetivo de separar la información por medio de expresiones regulares:

Separa: Función creada para separar el contenido del arreglo "registros" por medio de expresiones regulares y guardar el resultado en el arreglo "ip" que contendrá la información con el formato "protocolo ip_origen**:**puerto_origen **->** ip_destino**:**puerto_destino paquetes tráfico".

El diagrama de flujo de esta función se muestra en la figura 4.4.

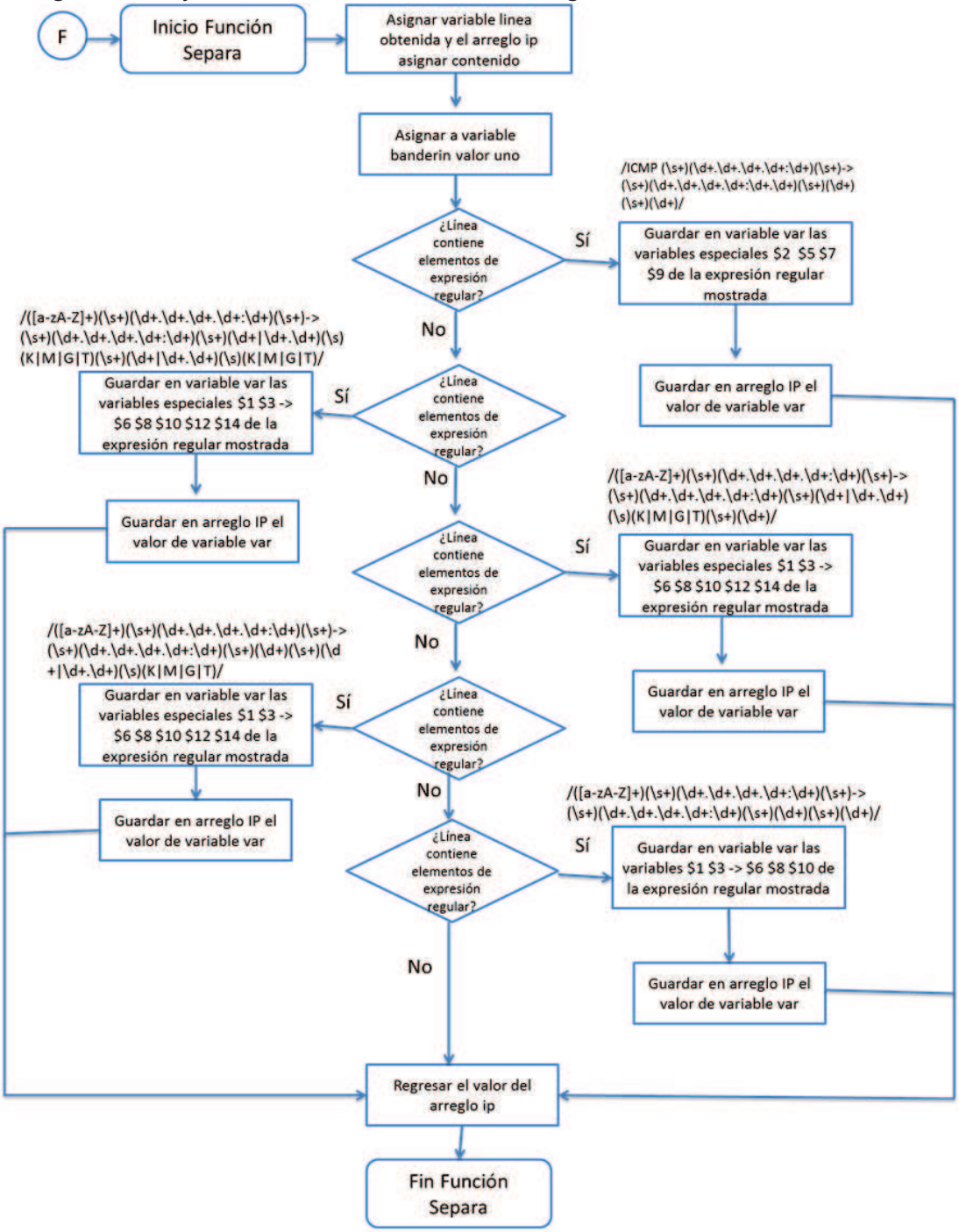


Figura 4.4 Diagrama de flujo de la función Separa

Agrupa: Función encargada de separar la información contenida en el arreglo "ip" por medio de expresiones regulares en el siguiente formato:

```
     aaa       xxx.xxx.xxx.xxx :  zzzz   ->  yyy.yyy.yyy.yyy :  wwww       bbb
(protocolo)   (ip ori)          (pto ori)       (ip dst)          (pto dst)  (trafico)
```

Cada valor separado se guarda en variables especiales con el objetivo de realizar análisis en estas variables en búsqueda de un escaneo de puertos y un ataque Dos o DDoS. Esta función es utilizada por las funciones "epuertos y DoS".

El diagrama de flujo de esta función se muestra en la figura 4.5.

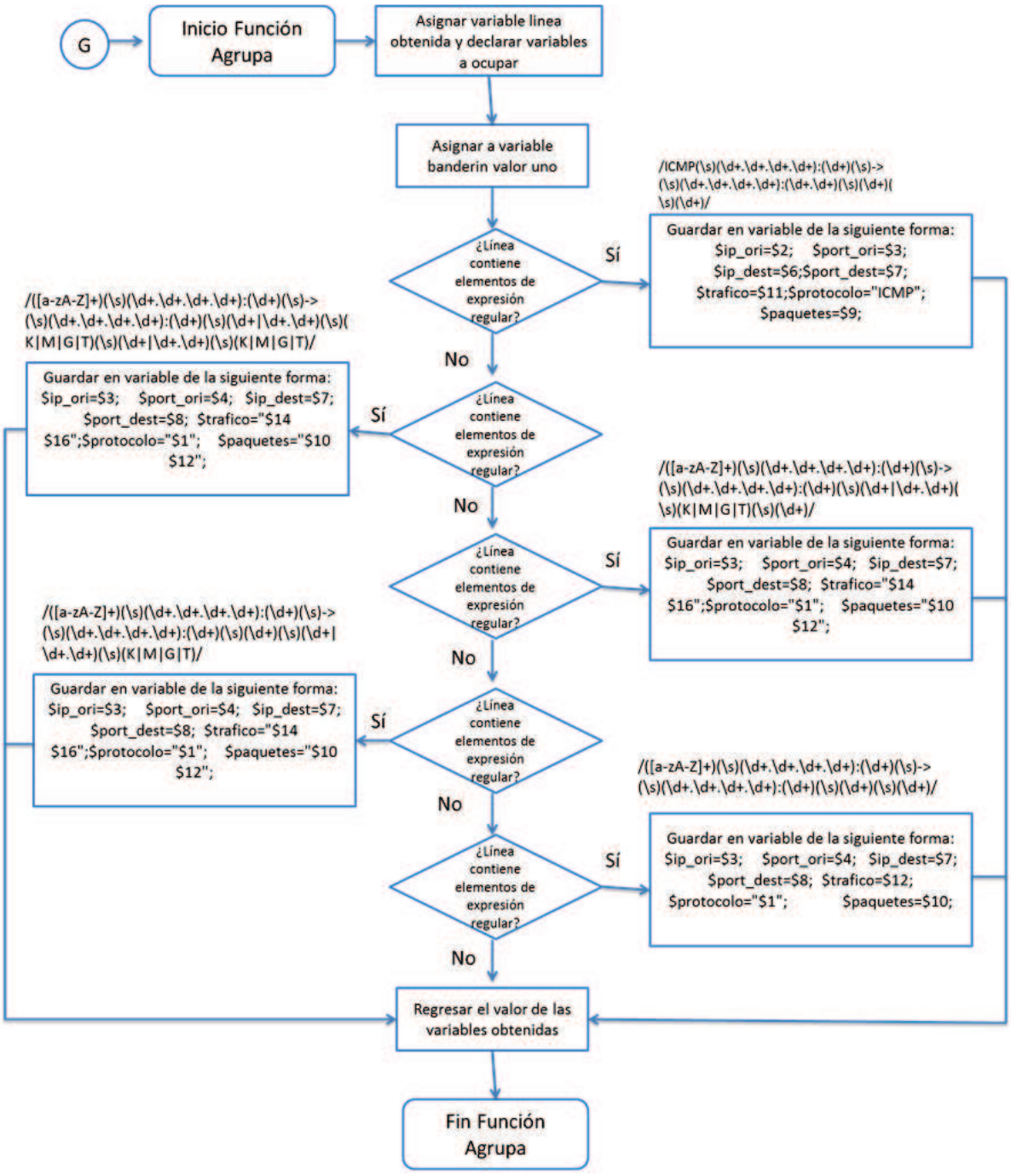


Figura 4.5 Diagrama de flujo de la función Agrupa

Agrupa_ip: Función encargada de separar la información contenida en el arreglo "ip" por medio de expresiones regulares en el siguiente formato:

```
        aaa    xxx.xxx.xxx.xxx  : zzzz  ->    yyy.yyy .    yyy  .   yyy    :  wwww    bbb
(Protocolo)   (ip ori)           (pto ori)    (ip dst1)  (ip dst2) (ip dst3)  (pto dst) (trafico)
```

Cada valor separado se guarda en variables especiales con el objetivo de realizar análisis sobre ellas en búsqueda de un escaneo de IP's y envío de información hacia el exterior. Esta función es utilizada por las funciones "eips y exterior".

El diagrama de flujo de esta función se muestra en la figura 4.6.

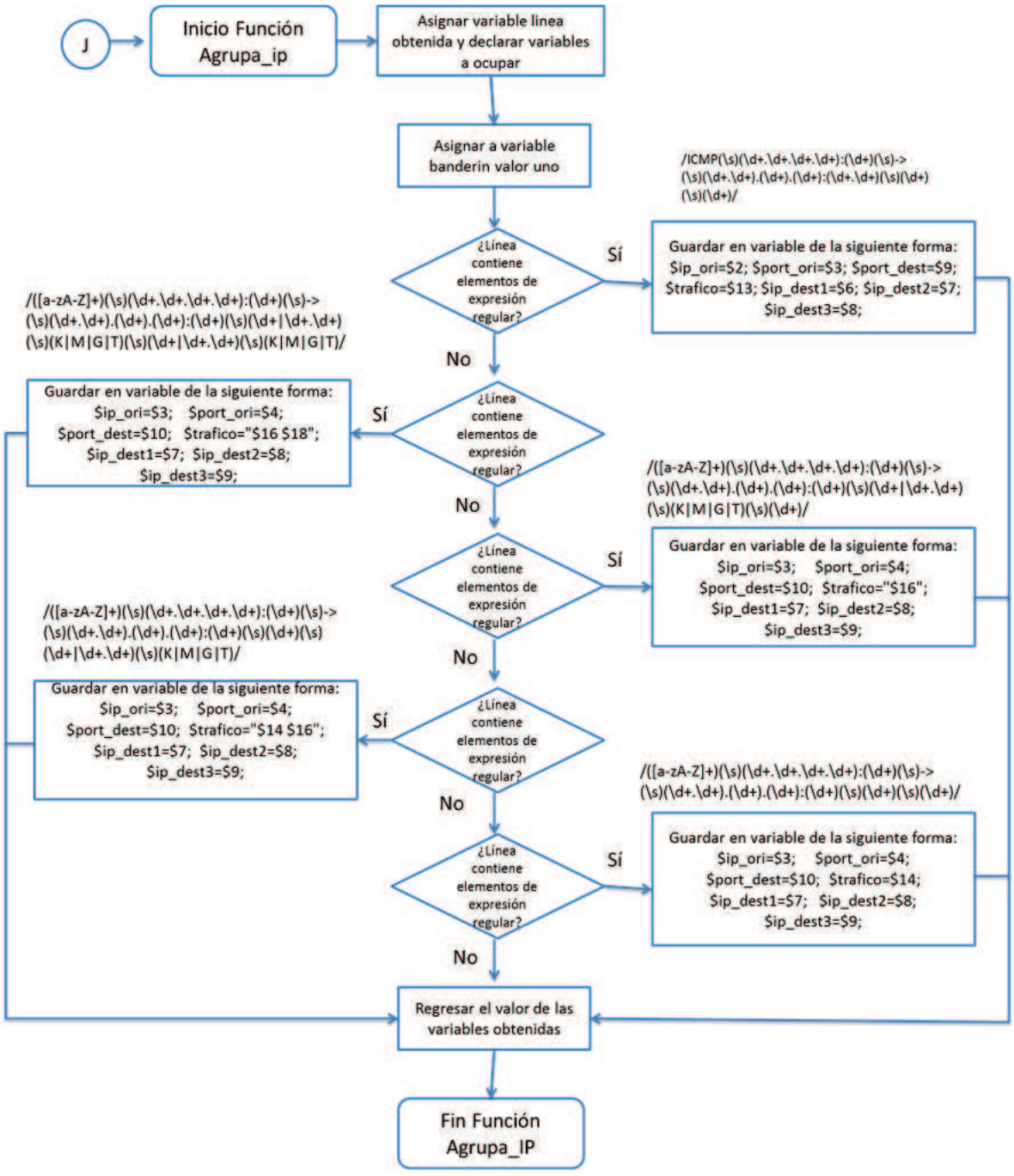


Figura 4.6 Diagrama de flujo de la función Agrupa_Ip

Las siguientes funciones fueron creadas con el objetivo de guardar y notificar en caso de encontrarse alguna anomalía en el análisis realizado.

Guarda: Función encargada de guardar las anomalías encontradas por las funciones epuertos, eips, exterior y DoS en el archivo de texto “anomalías”.

En la figura 4.7 muestra el diagrama de flujo creado para esta función.

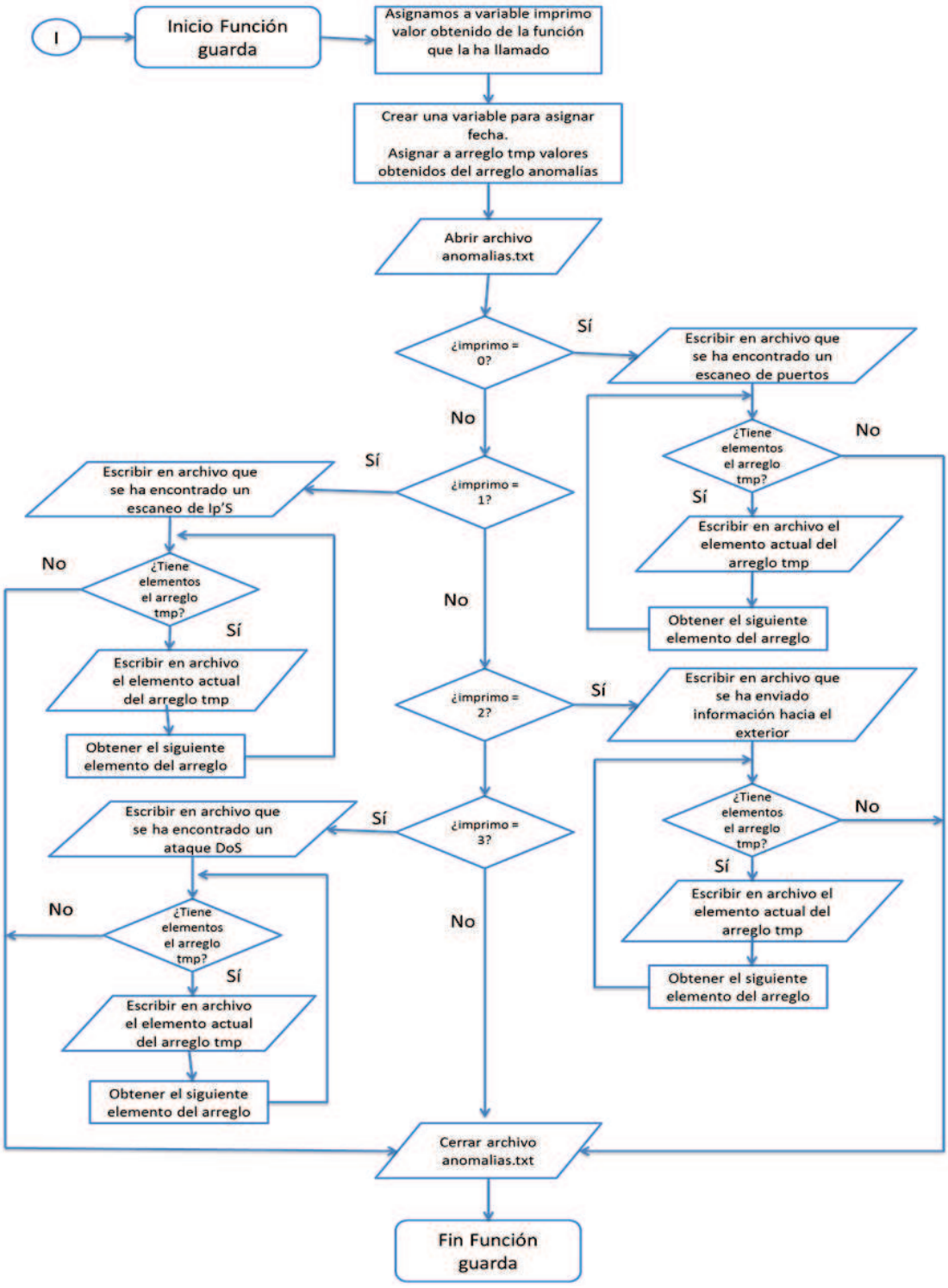


Figura 4.7 Diagrama de flujo de la función Guarda

Envía_correo: Función encargada de enviar un email con el contenido del archivo “anomalías” notificando sobre la(s) anomalía(s) encontrada(s).

Para él envío del email se utiliza al servicio sendmail. En la figura 4.8 se muestra el diagrama de flujo de esta función.

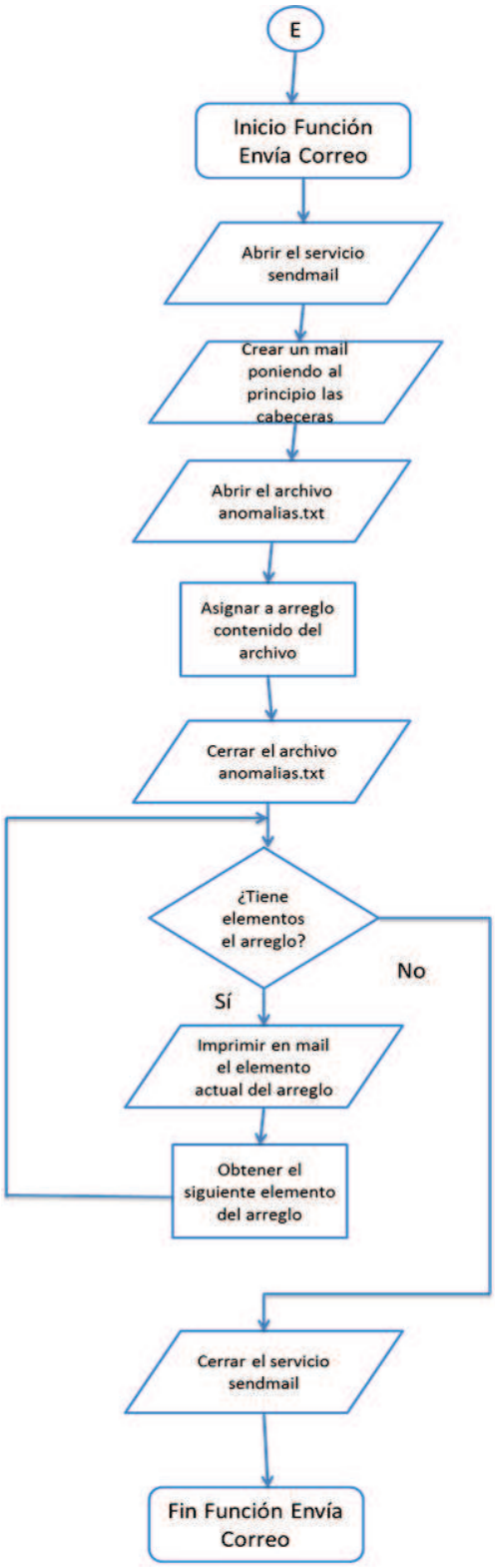


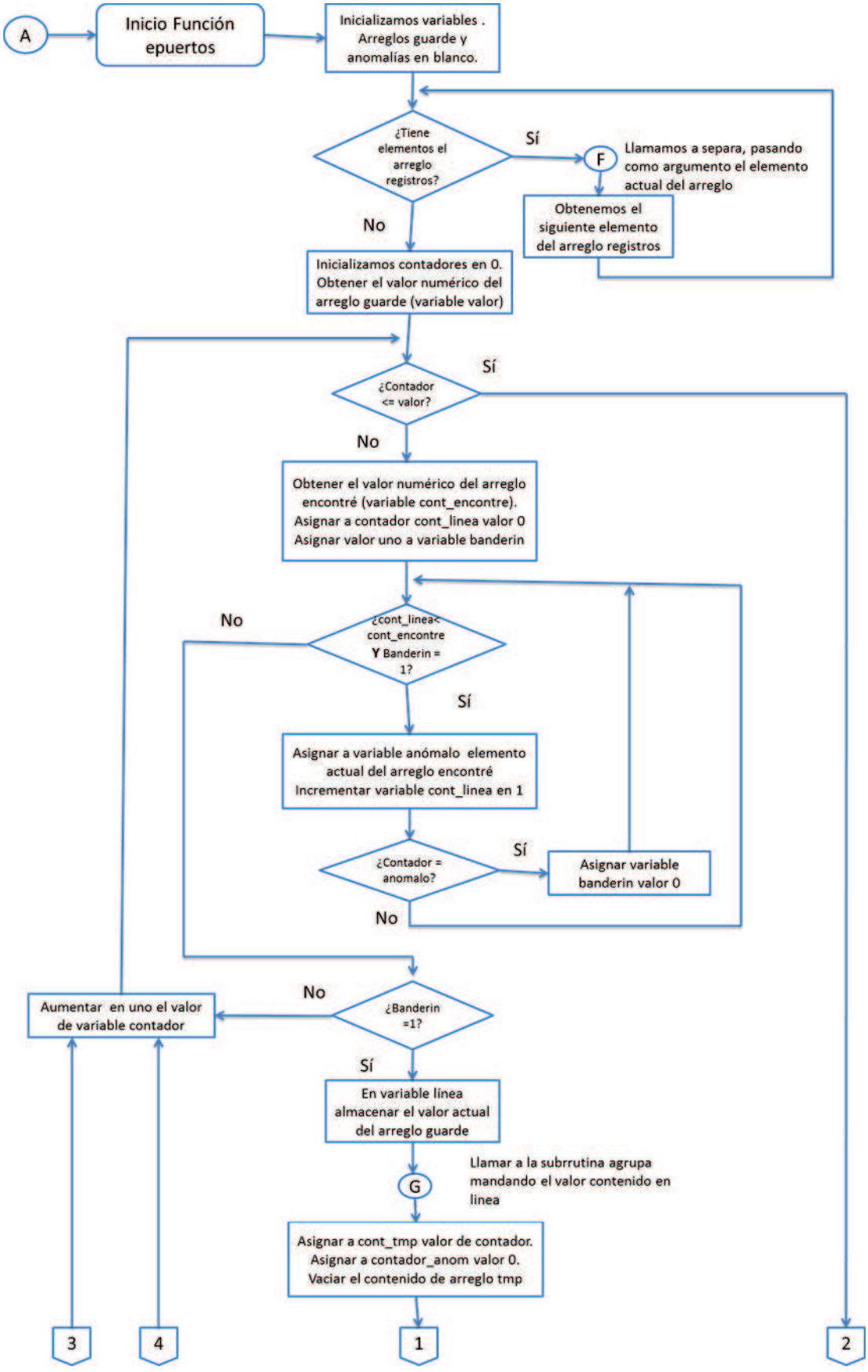
Figura 4.8 Diagrama de flujo de la función Envia_Correo

Las siguientes funciones fueron creadas con el objetivo analizar la información en búsqueda de comportamientos anormales presentes sobre la red interna de la institución:

Epuertos: Función creada con el objetivo de verificar si se ha realizado un escaneo de puertos. Su funcionamiento se describe a detalle a continuación:

1. Recibe el último archivo nfcapd obtenido en la función run y lo guarda en el arreglo *"registros"*.
2. Recorre todos los elementos contenidos en el arreglo *"registros",* separando cada elemento contenido en este arreglo mediante la función separa. El resultado obtenido se guarda en el arreglo *"guarde"*.
3. En la variable *"valor"* se hace referencia a cuantos elementos se han agrupado en el arreglo *"guarde"* (referencia numérica). Se inicializan contador y contador anormal en cero.
4. Se inicializa el primer ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"contador"* supere el número de la variable *"valor"*. Cuando se cumpla la acción indicada ir al paso diecisiete.
5. Se verifica si en previas iteraciones se ha detectado algún elemento como anormal. En caso afirmativo se asocia a variable *"banderin"* valor cero.
6. Si el valor de variable *"banderin"* es igual a uno se procede a obtener el elemento específico a analizar del arreglo *"guarde"* con ayuda del valor actual de variable *"contador"*, el resultado se guarda en variable línea. En caso contrario ir al paso quince.
7. Se manda llamar a función agrupa con el objetivo de separar la información contenida en la variable *"linea"* en las variables: *"ip_ori, pto_ori, ip_dest, pto_dest, trafico"*.
8. El valor contenido en variable *"contador"* se asocia a variable *"cont_tmp"*. El arreglo *"tmp"* que contiene elementos detectados como anormales se vacía.
9. Se ejecuta el segundo ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"cont_tmp"* supere el número de la variable valor. En caso de que la variable *"cont_tmp"* sea mayor a la variable *"valor"* ir al paso catorce.
   El objetivo del 2º ciclo while es recorrer los elementos que se tengan por debajo del valor actual de la variable *"contador"*.
10. Se incrementa en uno el valor de *"cont_tmp".* Se ejecuta la misma acción indicada en el paso cinco. La diferencia radica en que si la variable *"banderin"* es igual a cero, se regresa al paso nueve.
11. Se ejecuta la misma acción indicada en el paso seis. La diferencia radica en obtener el elemento específico de la arreglo *"guarde"* con ayuda del valor actual de la variable *"cont_tmp"*.
12. Se ejecuta la misma acción indicada en el paso siete. La diferencia radica en que se guardan los resultados en las variables temporales *"ip_ori_tmp, pto_ori_tmp, ip_dest_tmp, pto_dest_tmp, trafico_tmp".*
13. Se ejecuta la función compara en búsqueda de escaneo de puertos, pasando como argumentos las variables obtenidas del paso siete y once.
14. Regresar al paso nueve.
15. Si el valor de *"contador_anom"* obtenido como resultado de la función compara es mayor o igual a cinco, se guardan los elementos contenidos en arreglo *"tmp"* en el arreglo *"anomalias"*.
16. Incrementar el valor en uno de variable *"contador"* y regresar al paso cuatro.
17. Si existe el arreglo *"anomalías"*, asociar a variable imprimo valor cero y ejecutar función guarda, pasando como argumento el valor contenido en variable *"imprimo"* y el arreglo *"anomalias"*.
18. Fin función escaneo de puertos.

En la figura 4.9 se muestra el diagrama de flujo creado para esta función.



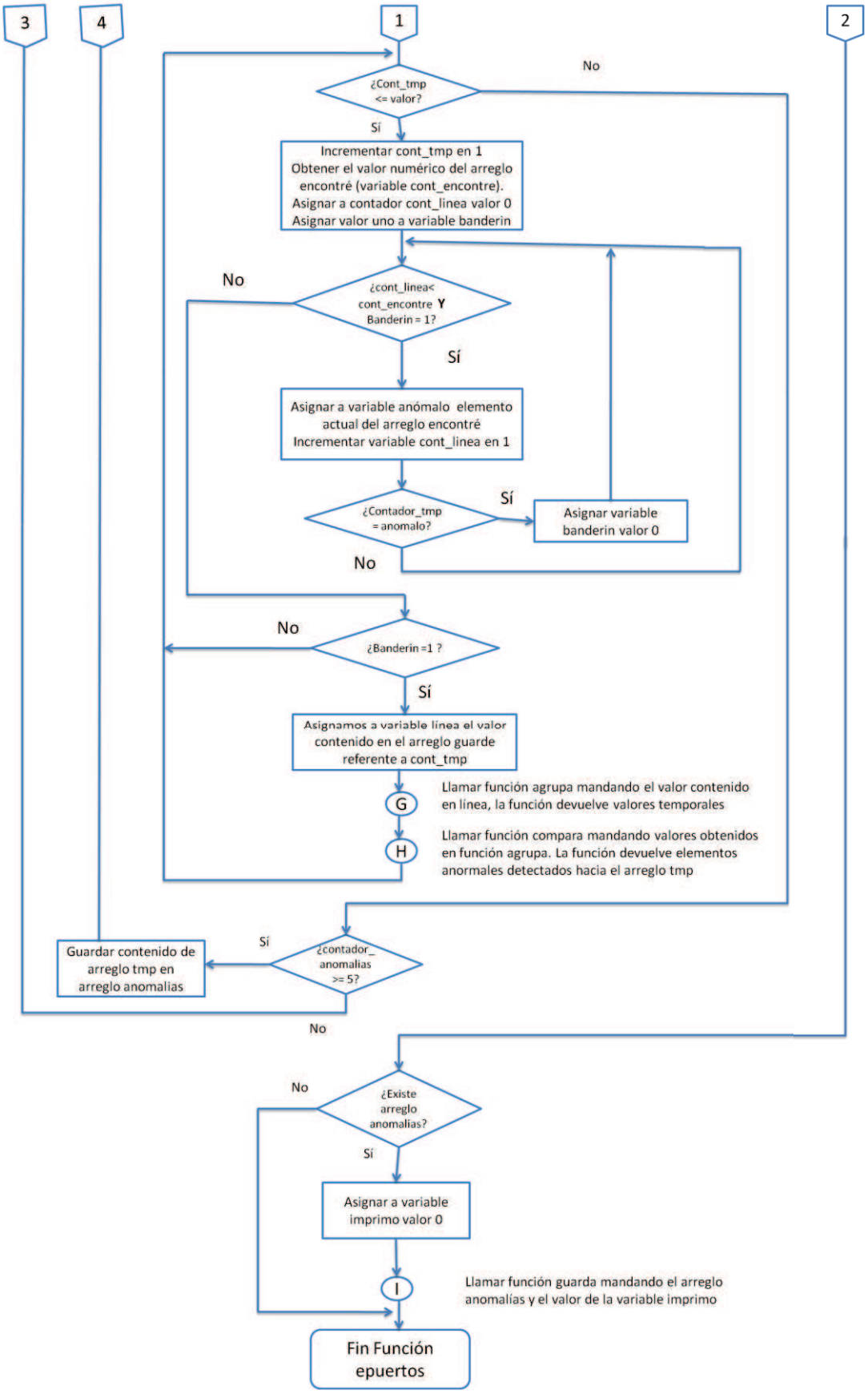


Figura 4.9 Diagrama de flujo de la función epuertos

Compara: Función utilizada por epuertos con el objetivo de verificar el elemento actual y el elemento temporal en búsqueda de un aumento en uno del puerto destino de la siguiente forma:

Ip_ori = Ip_ori_tmp **Y** Ip_dest=Ip_dest_tmp **Y** Port_dest = Port_dest_tmp.

En caso de detectar un elemento anormal, este elemento se guarda en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno.

El diagrama de flujo de esta función se muestra en la figura 4.10.

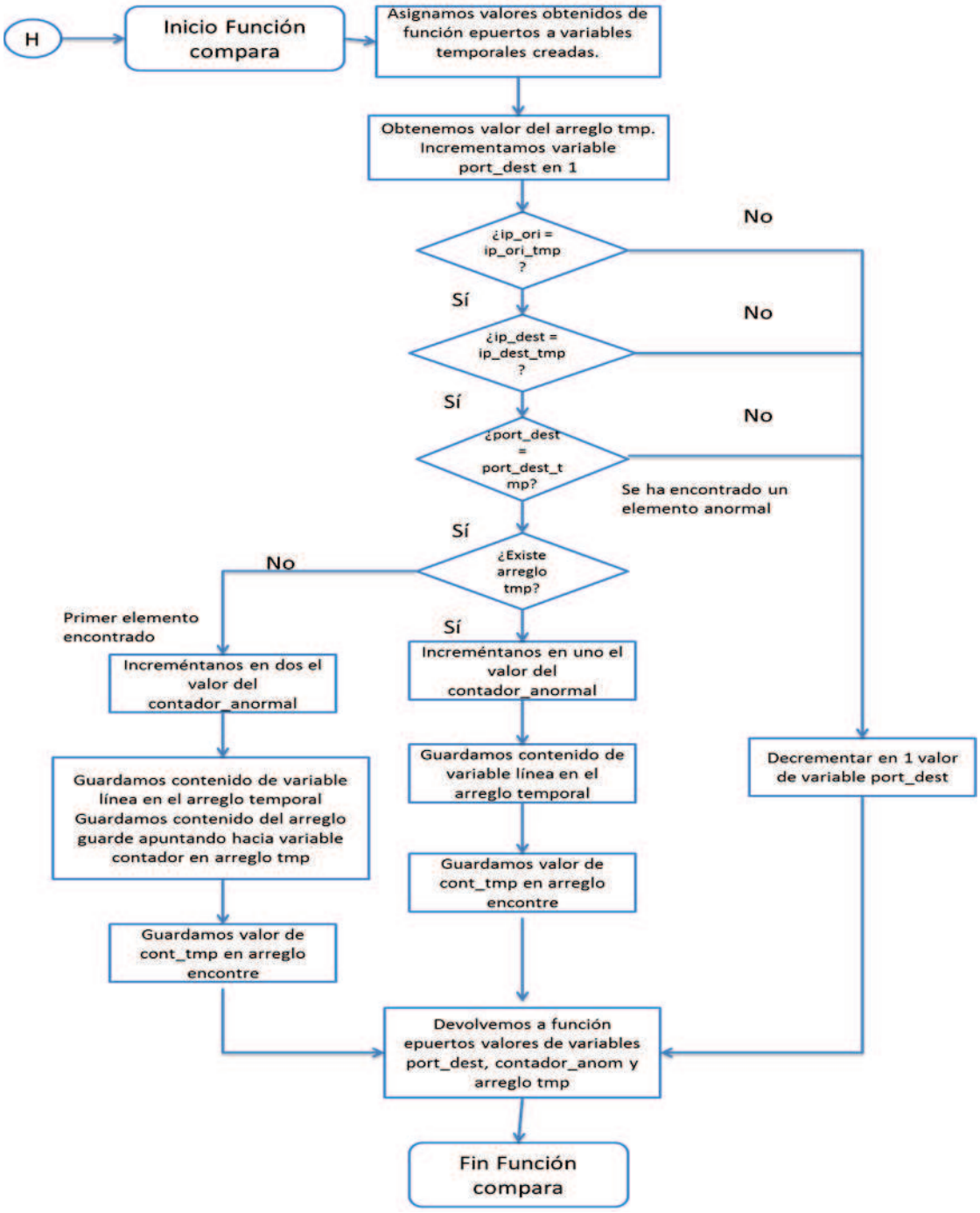
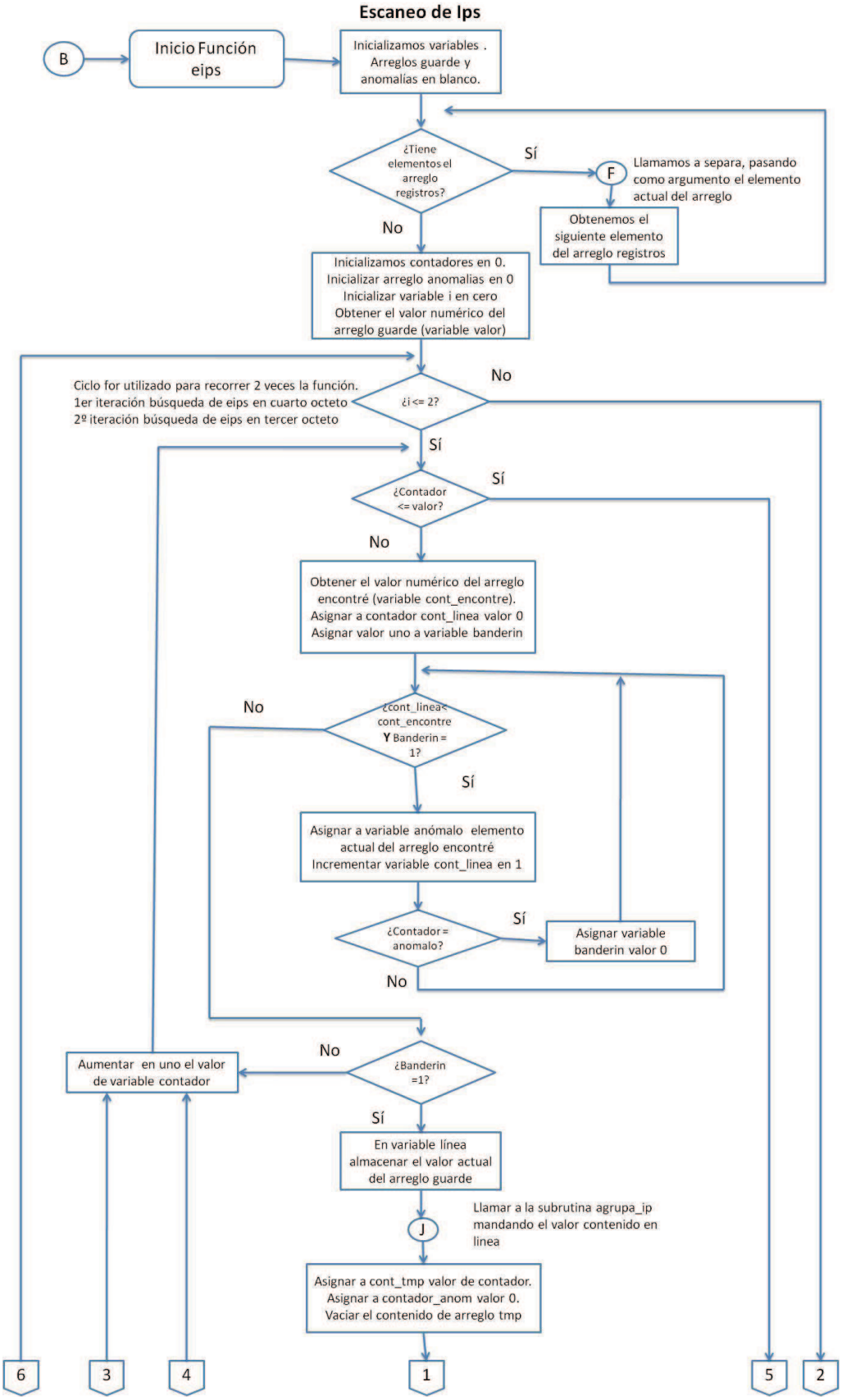


Figura 4.10 Diagrama de flujo de la función compara

Eips: Función creada con el objetivo de verificar si se ha realizado un escaneo de IP's en el tercer o cuarto octeto. Su funcionamiento se describe a detalle a continuación:

1. Recibe el ultimo archivo nfcapd obtenido en la función run y guardado en el arreglo *"registros"*.
2. Recorre todos los elementos contenidos en el arreglo *"registros",* separando cada elemento contenido en este arreglo ejecutando a la función separa. El resultado obtenido se guarda en el arreglo *"guarde"*.
3. Se inicializa un ciclo for que se ejecutará dos veces: En la primera iteración se buscará un escaneo de IP's en el cuarto octeto; en la segunda iteración se buscará un escaneo de IP's en el tercer octeto. En la tercera iteración ir al paso diecinueve.
4. En la variable *"valor"* se hace referencia a cuantos elementos se han agrupado en el arreglo *"guarde"* (referencia numérica). Se inicializan variables *"contador"* y *"contador anormal"* en cero.
5. Se inicializa el primer ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"contador"* supere el número de la variable *"valor"*. Cuando se cumpla esta acción ir al paso dieciocho.
6. Se verifica si en previas interacciones se ha encontrado algún elemento como anormal. En caso afirmativo se asocia a variable *"banderin"* valor cero.
7. Si el valor de variable *"banderin"* es igual a uno se procede a obtener el elemento específico a analizar del arreglo *"guarde"* con ayuda del valor actual de variable *"contador"*, el resultado se guarda en variable "línea". En caso contrario ir al paso diecisiete.
8. Se ejecuta la función *"agrupa_ip"* con el objetivo de separar la información contenida en la variable *"linea"* en variables: *"ip_ori, pto_ori, ip_dest1, ip_dest2, ip_dest3, pto_dest, trafico"*.
9. El valor contenido en variable "contador" se asocia a variable *"cont_tmp"*. El arreglo *"tmp"* que contiene elementos detectados como anormales se vacía.
10. Se ejecuta el 2º ciclo while que recorrerá todos los elementos del arreglo *"guarde"* hasta que la variable *"cont_tmp"* supere el número de la variable valor. En caso de que la variable *"cont_tmp"* sea mayor a la variable *"valor"* ir al paso quince.
    El objetivo del 2º ciclo while es recorrer los elementos que se tengan por debajo del valor actual de la variable *"contador"*.
11. Se incrementa en uno el valor de *"cont_tmp".* Se ejecuta la misma acción indicada en el paso seis.
12. Se ejecuta la misma acción indicada en el paso siete. La diferencia radica en obtener el elemento específico del arreglo *"guarde"* con ayuda del valor actual de la variable *"cont_tmp"*, y en que si la variable *"banderin"* es igual a cero se regresa al paso diez.
13. Se ejecuta la misma acción indicada en el paso ocho. La diferencia radica en que se guardan los resultados en las variables temporales *"ip_ori_tmp, pto_ori_tmp, ip_dest1_tmp, ip_dest2_tmp, ip_dest3_tmp, pto_dest_tmp, trafico_tmp"*.
14. Se ejecuta la función compara_ip en búsqueda de escaneo de Ip's en el tercer o cuarto octeto, depende en que iteración del ciclo for nos encontremos, pasando como argumentos las variables obtenidas del paso ocho y trece.
15. Regresar al paso diez.
16. Si el valor de *"contador_anom"* obtenido como resultado de la función compara es mayor o igual a cinco, se guardan los elementos contenidos en el arreglo *"tmp"* en el arreglo *"anomalias"*.
17. Incrementar el valor en uno de variable *"contador"* y regresar al paso cinco.
18. Incrementar en uno valor de variable i y regresar al paso tres.
19. Si existe el arreglo *"anomalías"*, asociar a variable imprimo valor uno y ejecutar función guarda, pasando como argumento el valor contenido en variable *"imprimo"* y el arreglo *"anomalias"*.
20. Fin Función escaneo de Ip's

En la figura 4.11 se muestra el diagrama de flujo creado para esta función.



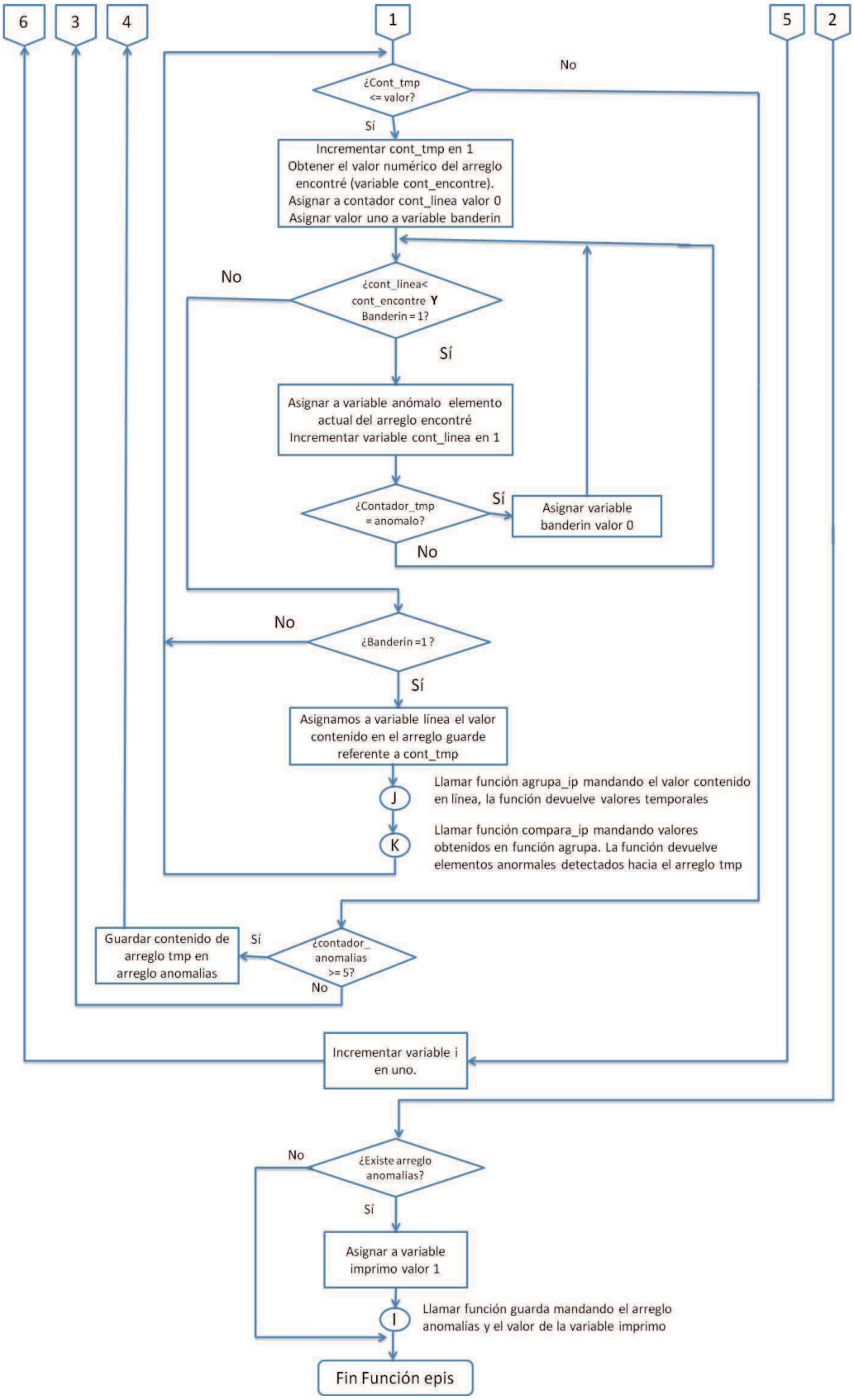


Figura 4.11 Diagrama de flujo de la función eips

compara_ip: Función utilizada por la función "eips" con el objetivo de verificar el elemento actual y el elemento temporal en búsqueda de un aumento en uno del tercer o cuarto octeto de la dirección IP destino, dependiendo de la iteración del ciclo for, de la siguiente forma:

var_ipori = var_ipori_tmp **Y** var_ipdst = var_ipdst_tmp

Donde *"var_ipori"* tiene el socket *"ip_ori:pto_ori"*, *"var_ipori_tmp"* tiene el socket *"ip_ori_tmpi:pto_ori_tmp", "var_ipdst"* tiene el socket *"ip_dst:pto_dst"* y *"var_ipdst_tmp"* tiene el socket *"ip_dst_tmpi:pto_dst_tmp"*

En caso de detectar un elemento anormal, se guarda este elemento en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno, el diagrama de flujo de esta función se muestra en la figura 4.12.

Exterior: Función creada con el objetivo de verificar si se ha enviado información de alguna red de servidores hacia direcciones IP no permitidas (generalmente externas al rango de la institución), con un tráfico mayor a 5 Mb y un aumento en el puerto origen de forma secuencial. Su funcionamiento es similar a la función "epuertos", solo cambia en los siguientes puntos:

- ✓ En el paso siete se manda a llamar a la función agrupa_ip, guardando el resultado en variables *"ip_ori, pto_ori, ip_dest1, ip_dest2, ip_dest3, pto_dest, trafico"*
- ✓ En el paso doce se manda a llamar nuevamente a función agrupa_ip, pero ahora se guarda el resultado en variables temporales: *"ip_ori_tmp, pto_ori_tmp, ip_dest_tmp1, ip_dest_tmp2, ip_dest_tmp3, pto_dest_tmp, trafico_tmp"*
- ✓ En el paso trece se ejecuta la función compara_exterior, pasando como argumento las variables obtenidas de los dos pasos anteriores.
- ✓ En el paso diecisiete se asigna a variable imprimo el valor de dos.

En la figura 4.13 se muestra el diagrama de flujo de esta función.

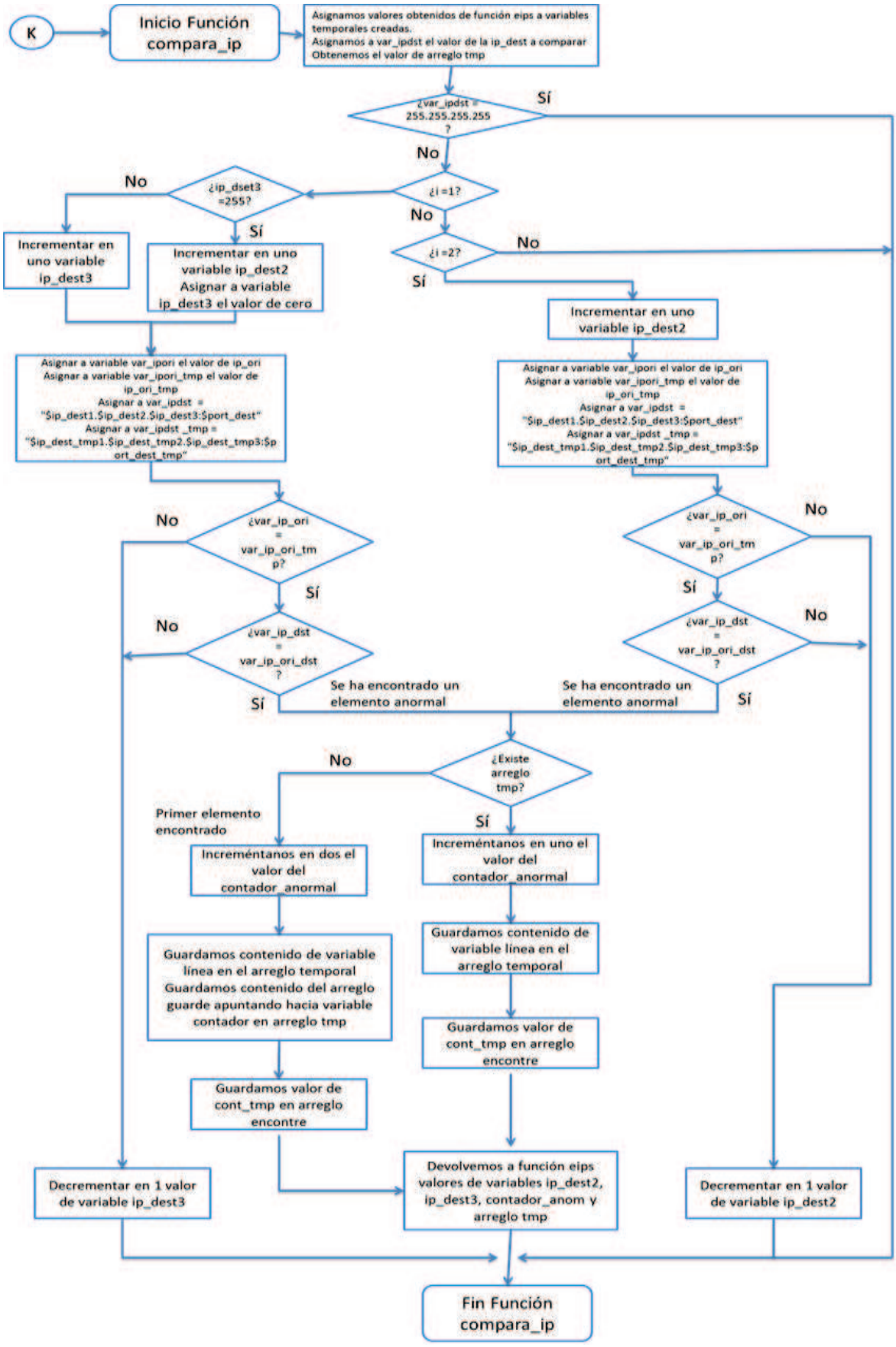
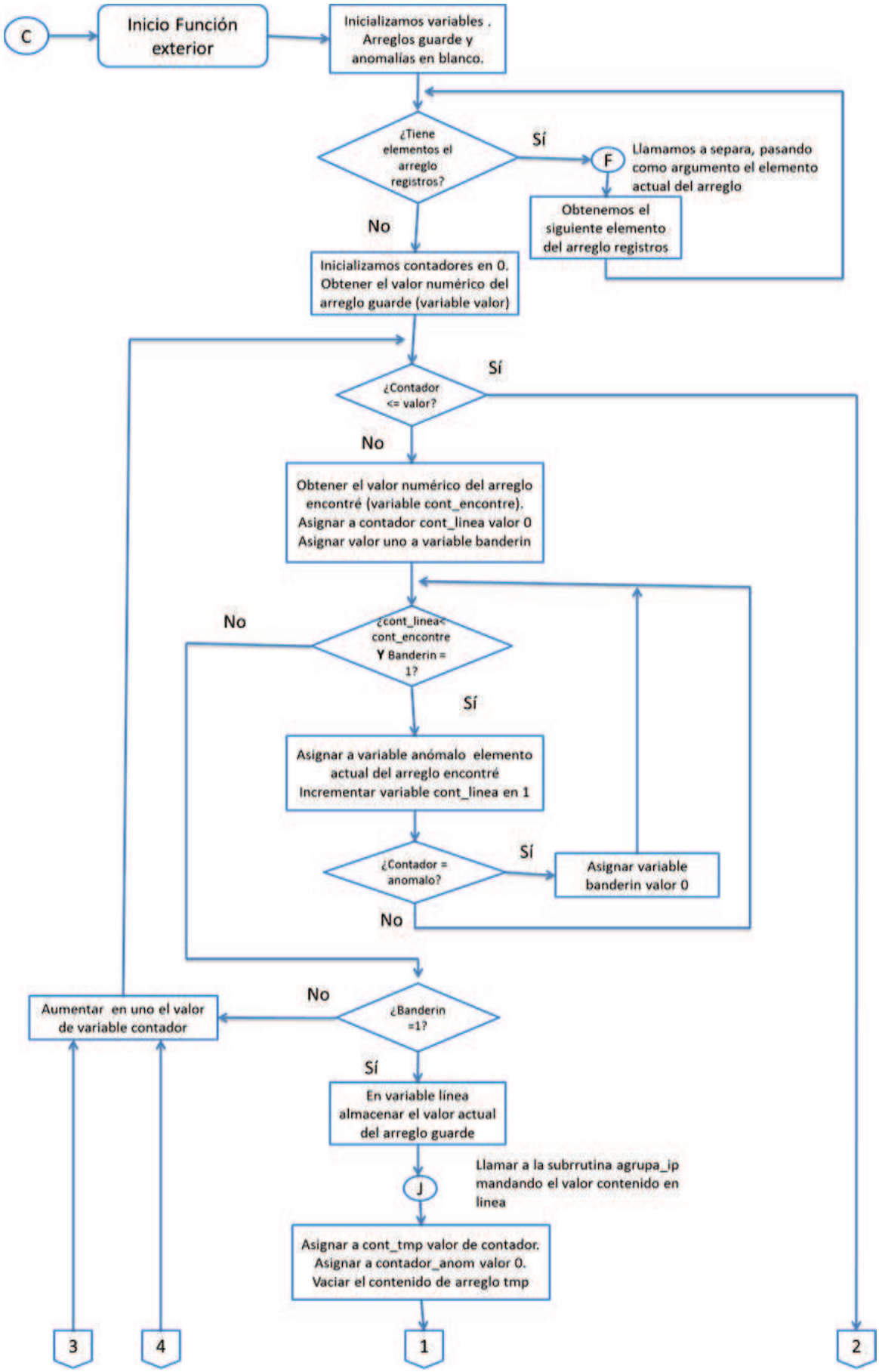


Figura 4.12 Diagrama de flujo de la función compara_ip


C
Inicio Función exterior
Inicializamos variables . Arreglos guarde y anomalías en blanco.
¿Tiene elementos el arreglo registros?
Sí
F
Llamamos a separa, pasando como argumento el elemento actual del arreglo
Obtenemos el siguiente elemento del arreglo registros
No
Inicializamos contadores en 0. Obtener el valor numérico del arreglo guarde (variable valor)
¿Contador <= valor?
Sí
No
Obtener el valor numérico del arreglo encontré (variable cont_encontre). Asignar a contador cont_linea valor 0 Asignar valor uno a variable banderin
¿cont_linea< cont_encontre Y Banderin = 1?
No
Sí
Asignar a variable anómalo elemento actual del arreglo encontré Incrementar variable cont_linea en 1
¿Contador = anomalo?
Sí
Asignar variable banderin valor 0
No
¿Banderin =1?
No
Aumentar en uno el valor de variable contador
Sí
En variable línea almacenar el valor actual del arreglo guarde
J
Llamar a la subrrutina agrupa_ip mandando el valor contenido en linea
Asignar a cont_tmp valor de contador. Asignar a contador_anom valor 0. Vaciar el contenido de arreglo tmp
3
4
1
2

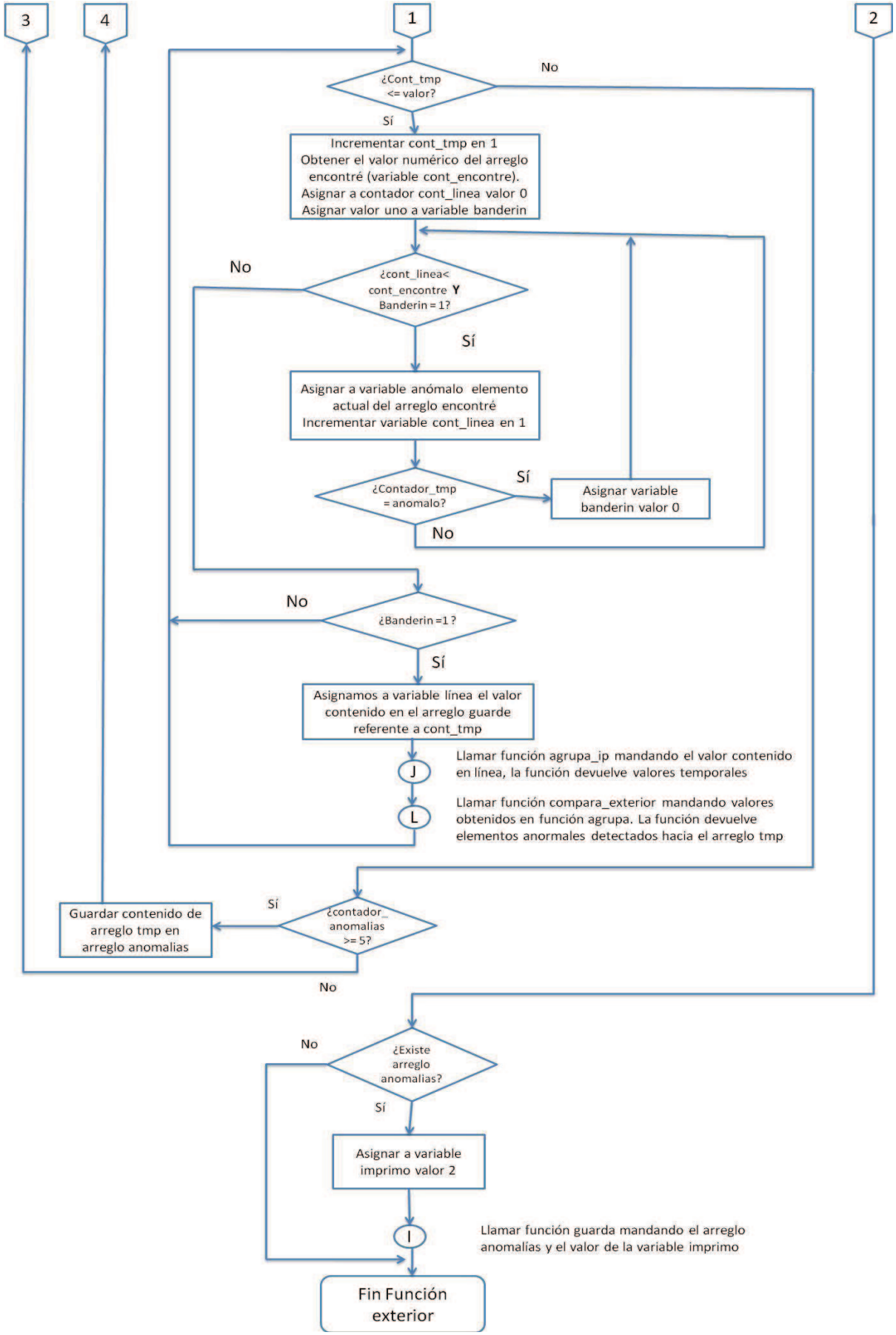


Figura 4.13 Diagrama de flujo de la función exterior

Compara_exterior: Función utilizada por la función "exterior" con el objetivo de verificar el elemento actual y el elemento temporal, pertenecientes a una red de servidores, en búsqueda de un aumento en uno del puerto origen y un envió de información mayor a 5 Mb hacia direcciones ip exteriores al rango permitido en la institución, de la siguiente forma:

"Ip_origen" pertenezca a una red de servidores **Y** *"ip_origen igual a ip_origen_tmp"* **Y** *"puerto_ori_tmp"* se haya incrementado en con respecto al valor de *"puerto_ori"* **Y** *"trafico_tmp sea mayor a 5242880"*.

En caso de detectar un elemento anormal, se guardara este elemento en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno.

El diagrama de flujo de esta función se muestra en la figura 4.14.

Dos: Función creada con el objetivo de verificar si se ha realizado un ataque DoS o un ataque DDoS en una dirección IP perteneciente a la institución con un tráfico mayor a 50 Mb. Su funcionamiento es similar a la función epuertos, solo cambia en los siguientes puntos:

- ✓ En el paso trece se ejecuta la función compara_DoS, pasando como argumento las variables obtenidas en el paso siete y doce.
- ✓ En el paso quince se compara con un valor de variable *"contador_anom"* mayor a veinticinco.
- ✓ En el paso diecisiete se asigna a variable imprimo el valor de tres.

En la figura 4.15 se muestra el diagrama de flujo de esta función.

Compara_dos: Función utilizada por la función "DoS" con el objetivo de verificar si en el elemento actual y el elemento temporal, se ha presentado una conexión mayor a 50 Mb, de la siguiente forma:

"ip_destino igual a ip_destino_tmp" **Y** *"puerto_destino igual a puerto_destino_tmp"* **Y** *"trafico_tmp sea mayor a 52428800"*.

En caso de detectar un elemento anormal, se guarda este elemento en el arreglo *"tmp"* y se incrementa el valor de *"contador_anomal"* en uno.

El diagrama de flujo se muestra en la figura 4.16.

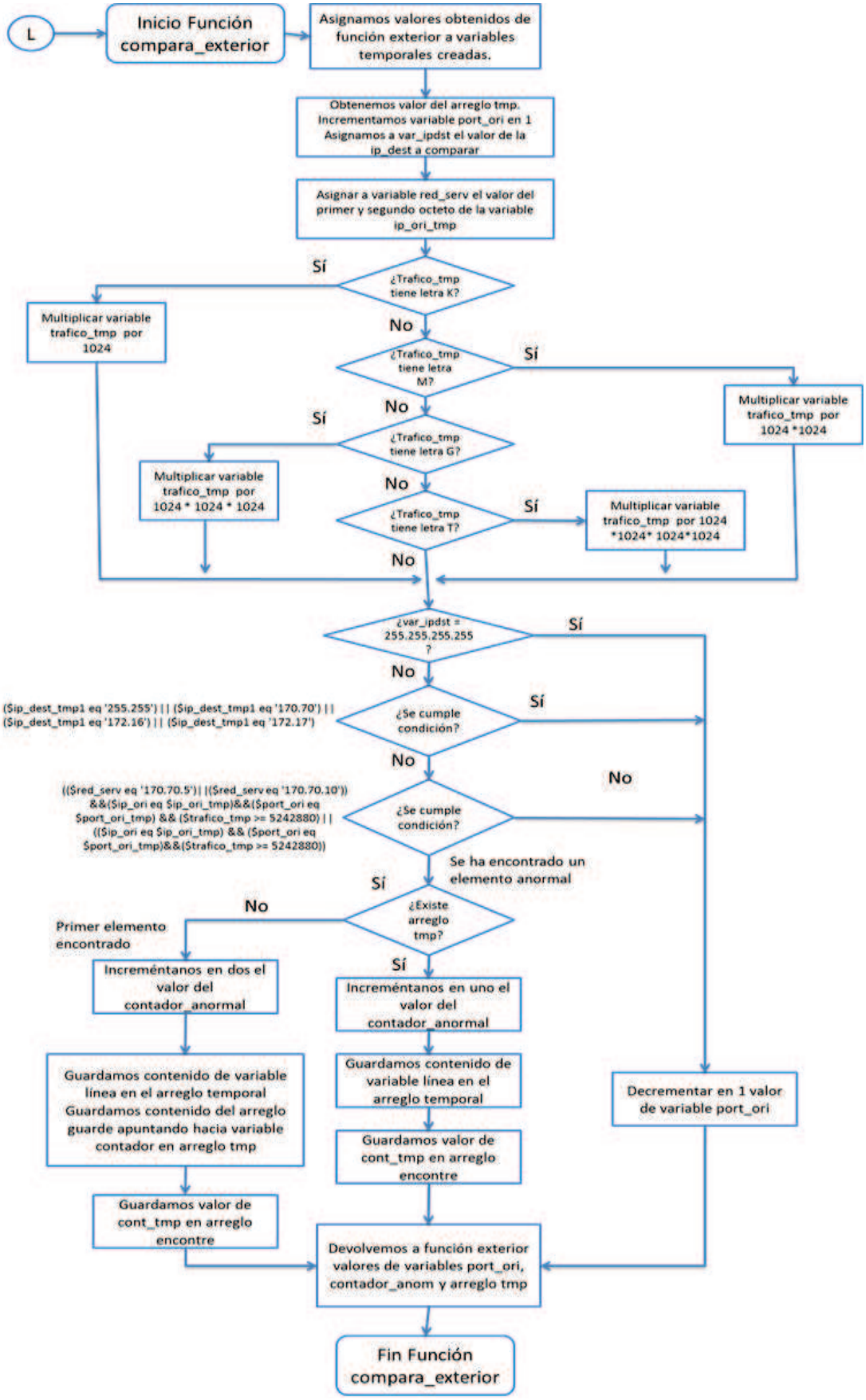
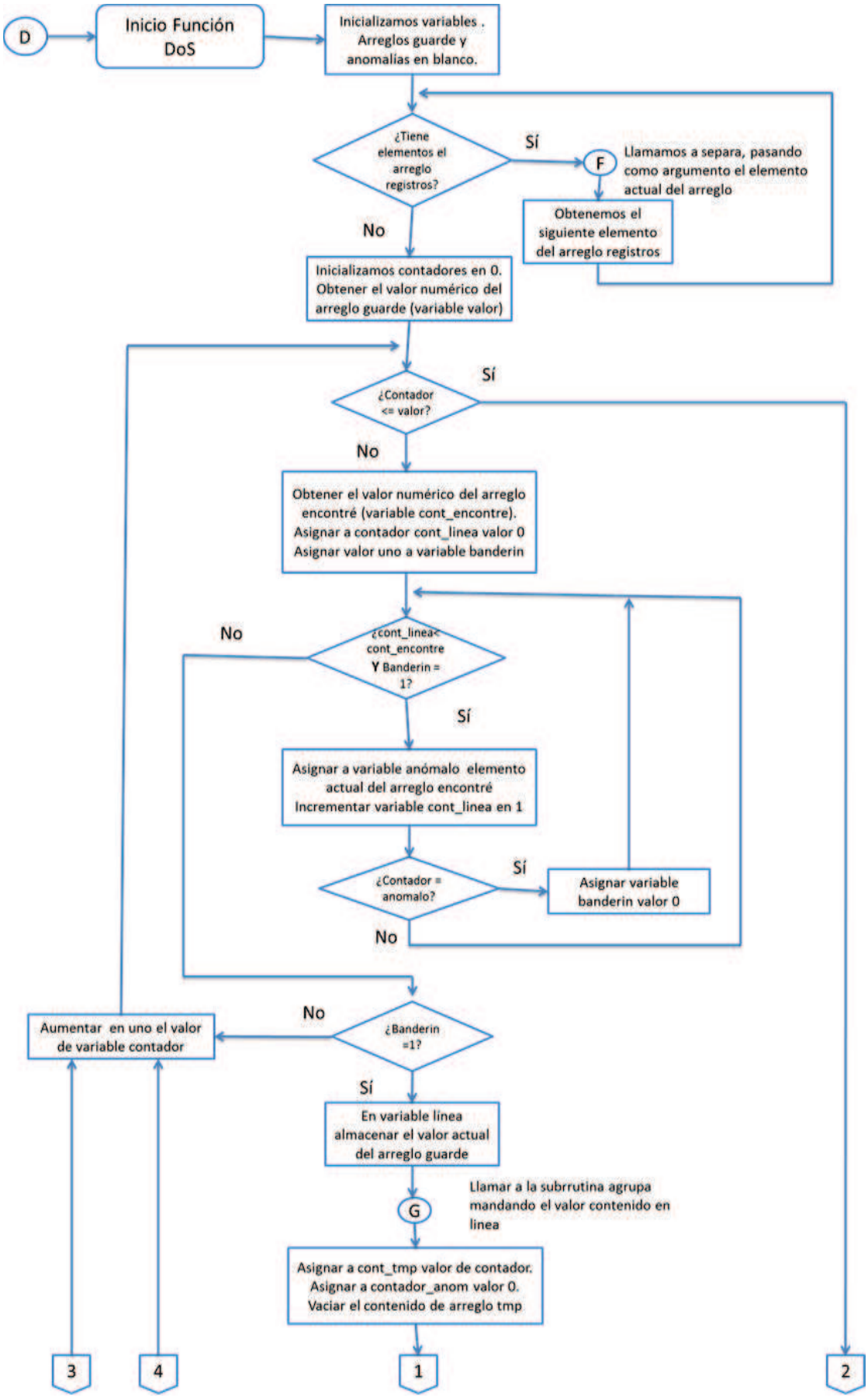


Figura 4.14 Diagrama de flujo de la función compara_exterior.


D
Inicio Función DoS
Inicializamos variables . Arreglos guarde y anomalías en blanco.
¿Tiene elementos el arreglo registros?
Sí
F
Llamamos a separa, pasando como argumento el elemento actual del arreglo
Obtenemos el siguiente elemento del arreglo registros
No
Inicializamos contadores en 0. Obtener el valor numérico del arreglo guarde (variable valor)
¿Contador <= valor?
Sí
No
Obtener el valor numérico del arreglo encontré (variable cont_encontre). Asignar a contador cont_linea valor 0 Asignar valor uno a variable banderin
¿cont_linea< cont_encontre Y Banderin = 1?
No
Sí
Asignar a variable anómalo elemento actual del arreglo encontré Incrementar variable cont_linea en 1
¿Contador = anomalo?
Sí
Asignar variable banderin valor 0
No
¿Banderin =1?
No
Aumentar en uno el valor de variable contador
Sí
En variable línea almacenar el valor actual del arreglo guarde
G
Llamar a la subrrutina agrupa mandando el valor contenido en linea
Asignar a cont_tmp valor de contador. Asignar a contador_anom valor 0. Vaciar el contenido de arreglo tmp
3
4
1
2

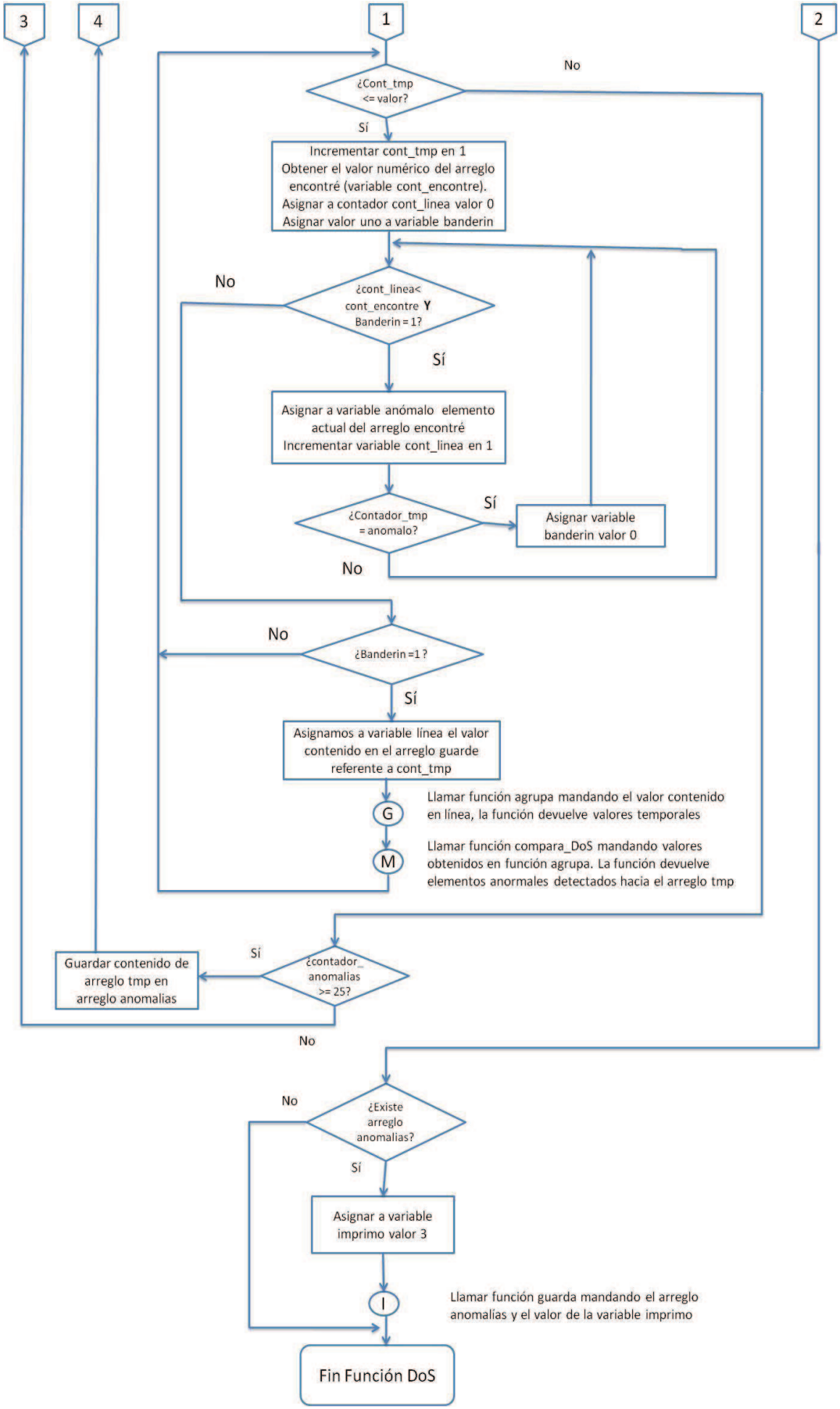


Figura 4.15 Diagrama de flujo de la función Dos.

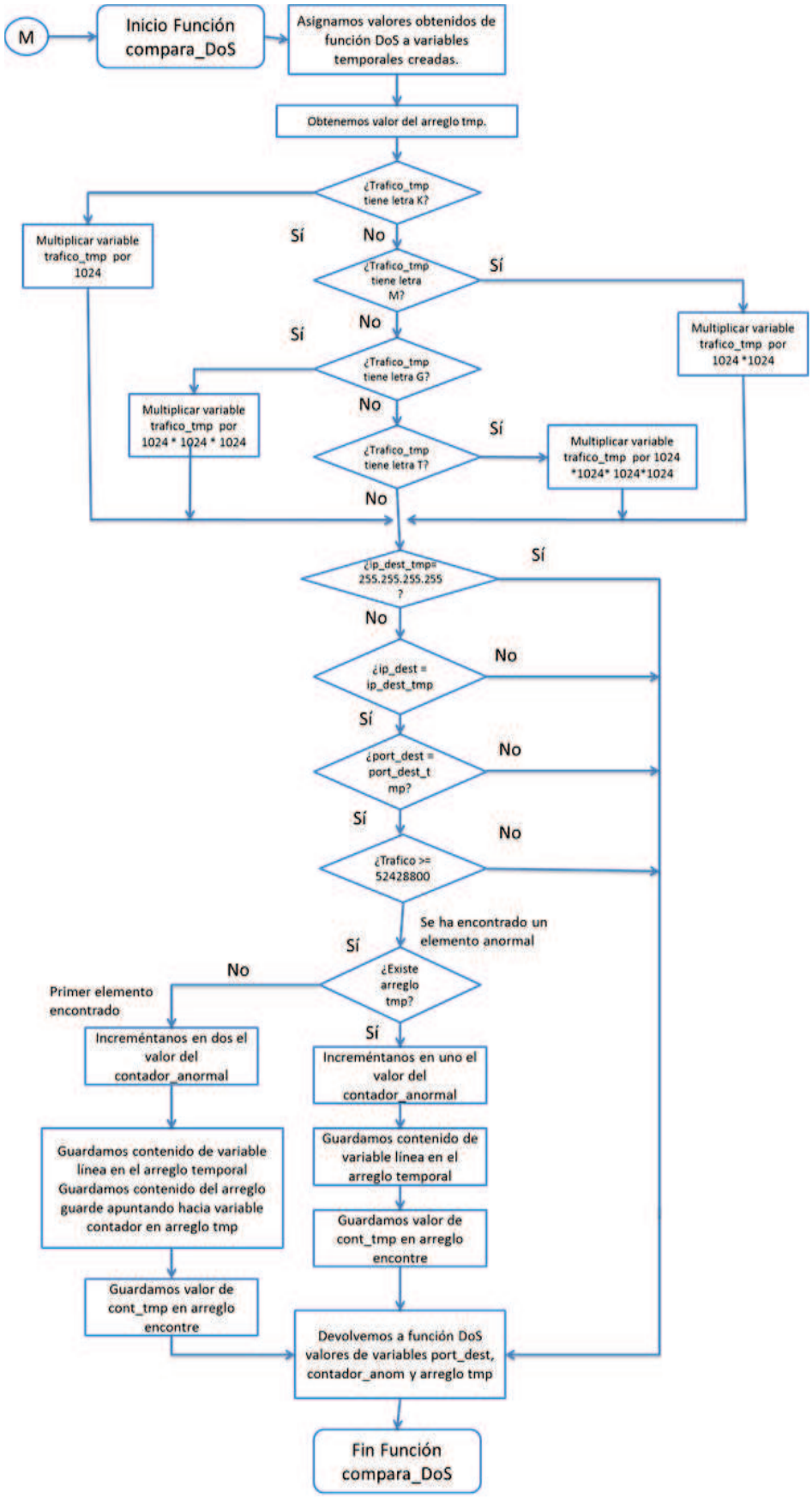


Figura 4.16 Diagrama de flujo de la función compara_dos

4.5.3.2 Funcionamiento modulo *"Escaneo.php"*.

El módulo frontend "escaneo.php" fue creado con el objetivo de visualizar los resultados obtenidos por el módulo "escaneo.pm" en la interfaz gráfica del software Nfsen. Este módulo se encarga de mostrar en pantalla el resultado del archivo "anomalías", en caso de haberse detectado una anomalía, y de mostrar el archivo nfcapd que se analizó.

El diagrama de flujo mostrado en la figura 4.2 es el utilizado por el modulo "escaneo.php". La función *"escaneo_ParseInput"* no es ocupada en este módulo frontend creado, sin embargo esta función debe de existir.

En la función "escaneo_Run" se realizan todas las acciones programadas y mostradas en el diagrama de flujo de la figura 4.2.

En el anexo D se muestra el código desarrollado en la creación del módulo frontend "escaneo.php" y el módulo backend "escaneo.pm".

En el capítulo V se mostrará el funcionamiento del software "Listry-AIGC", tanto para la función del monitoreo de red como en la detección de malware mediante el plugin creado. Este software está conformado por lo siguiente:

- Instalación y configuración del módulo HTTPS.
- Instalación y configuración del software Nfsen.
- La creación del plugin escaneo en el software Nfsen.
- Instalación y configuración del software MySQL y OpenWebmail.
- Instalación y configuración del software Navicat.

Capítulo V

Pruebas

5.1 Introducción

Este capítulo tiene como objetivo mostrar el correcto funcionamiento del software "Listry-AIGC" implementado en la institución en sus dos principales actividades:

- Enfocado hacia el monitoreo de la red interna de la institución.
- Enfocado hacia la detección de malware mediante el plugin "escaneo".

Por cuestiones de integridad y confidencialidad de la información presente en la institución se decidió realizar pruebas con ayuda de un software que genera paquetes en formato Netflow llamado "Paessler Netflow Generator".

5.2 Pruebas

Para la demostración del correcto funcionamiento del software "Listry-AIGC" se simularon dos edificios: En cada edificio trabajan aproximadamente 250 personas. La tabla 5.1 muestra algunas subredes creadas y asignadas a cada edificio para cuestiones de pruebas.

Tabla 5.1 Subredes creadas en el proceso de simulación

Edificio	A		B	
Red	Usuarios	Servidores	Usuarios	Servidores
Subred	172.16.5.0/24	172.16.10.0/28	172.16.6.0/24	172.16.11.0/28

La empresa requiere monitorear la actividad realizada en la red central de los trabajadores debido a que se ha detectado un consumo excesivo en las subredes presentes, sospechando de la presencia de un malware que ha infectado a la red central.

La Figura 5.1 muestra el esquema creado para la realización de las pruebas.

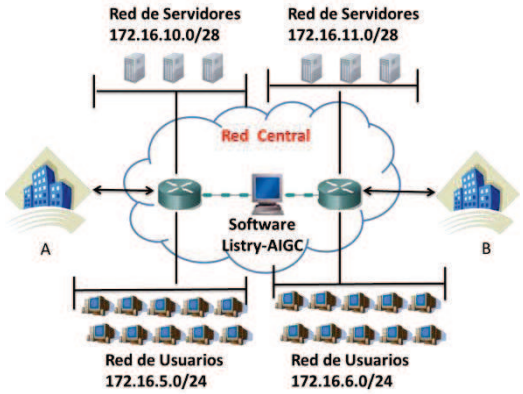


Figura 5.1 Esquema creado para ejemplos

5.2.1 Pruebas enfocadas a monitoreo.

En el capítulo III se mostraron algunas pruebas realizadas en el software Nfsen enfocadas en el monitoreo de red. Por este motivo en esta sección se crearon solamente profiles que observan conexiones realizadas hacia servidores de la siguiente forma:

- Monitorear todo el tráfico que llega de redes de usuarios al servidor de correos del edificio A con IP 172.16.10.3 y del edificio B con IP 172.16.11.6
- Monitorear todo el tráfico que llega al servidor web del edificio A con IP 172.16.10.4 y del edificio B con IP 172.16.11.7

- Monitorear todo el tráfico que llega al servidor de base de datos del edificio A con IP 172.16.10.5 y del edificio B con IP 172.16.11.8

La figura 5.2 muestra la simulación de flujos mediante el software "Paessler Netflow Generetor"

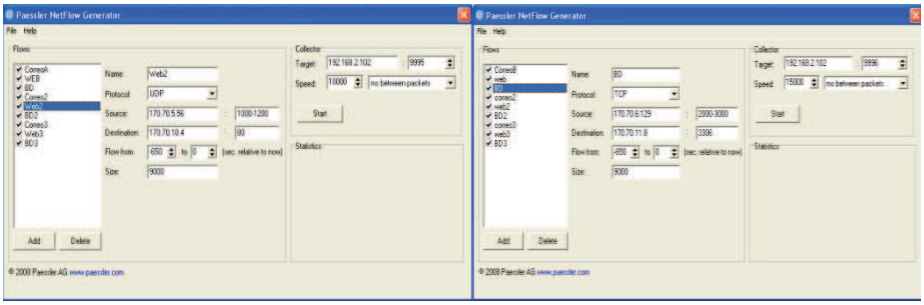


Figura 5.2 Simulación de flujos.

Mediante los flujos creados se está simulando la actividad presente en dos edificios. En la ventana izquierda de la figura se emula el tráfico generado en el edificio A y en la parte derecha el tráfico generado en el edificio B. En ambos edificios se crearon inicialmente nueve flujos; cada uno de ellos emula una conexión realizada hacia el servidor de correo, web o de bases de datos *(B.D.)* respectivamente. En la simulación se varia el puerto origen, el tamaño de los flujos y el tiempo que llevan activos, para tratar de hacer la simulación lo más real posible. En el edificio A inicialmente se enviaban flujos cada 10000 ms (10 S), mientras que en el edificio B se enviaban flujos cada 15000ms (15 S).

La figura 5.3 muestra el comportamiento de los edificios en el profile 'live' correspondiente al periodo 02-Octubre-2010 21:40 al 03 Octubre-2010 00:40.

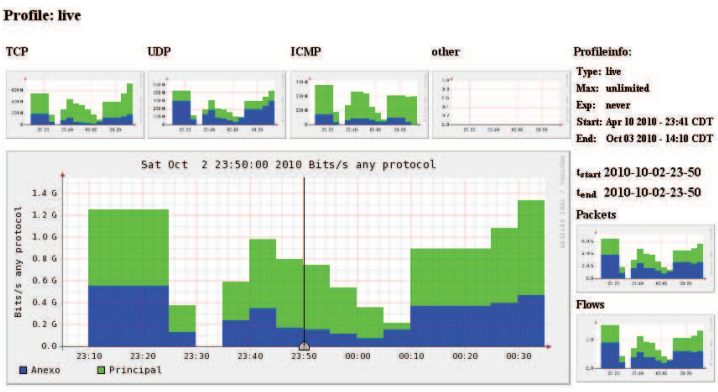


Figura 5.3 Grafica obtenida del profile "live"

Al realizar un análisis sobre esta grafica obtenida no se encuentran grandes variaciones con respecto al tráfico generado en cada edificio. Se observa que el edificio A tiene una mayor actividad que el edificio B. Por este motivo se decidió observar detalladamente un archivo estadísticas *TOPN* en este archivo, se obtuvo la siguiente información:

** nfdump -M /Listry-AIGC/nfsen-1.3.2/profiles-data/live/Principal:Anexo -T -r 2010/10/02/nfcapd.201010022350 -n 20 -s srcip/flows
nfdump filter: any
Top 20 Src IP Addr ordered by flows:

Date first seen	Duration	Proto	Src IP Addr	Flows(%)	Packets(%)	Bytes(%)	pps	bps	bpp
2030-01-27 11:30:48.634	60.000	any	172.16.5.10	19(7.2)	81.3 G(7.2)	311.3 M(1.1)	1.4 G	41.5 M	0
2030-01-27 11:21:48.634	600.000	any	172.16.5.15	19(7.2)	81.3 G(7.2)	311.3 M(1.1)	135.5 M	4.2 M	0

```
2030-01-27 11:20:58.634   650.000 any    172.16.5.56     19( 7.2)  81.3 G( 7.2)   2.8 G(10.1)  125.1 M  34.5 M    0
2030-01-27 11:28:48.634   180.000 any    172.16.5.123    19( 7.2)  81.3 G( 7.2)   1.4 G( 4.9)  451.6 M  60.9 M    0
2030-01-27 11:30:48.634    60.000 any    172.16.5.45     19( 7.2)  81.3 G( 7.2)  778.2 M( 2.8)   1.4 G  103.8 M    0
2030-01-27 11:21:48.634   600.000 any    172.16.5.156    19( 7.2)  81.3 G( 7.2)  336.2 M( 1.2)  135.5 M   4.5 M    0
2030-01-27 11:20:02.634   450.000 any    172.16.6.222    17( 6.4)  72.7 G( 6.4)   2.4 G( 8.7)  161.6 M  43.1 M    0
2030-01-27 11:12:32.634   900.000 any    172.16.6.127    17( 6.4)  72.7 G( 6.4)   1.4 G( 5.1)   80.8 M  12.5 M    0
2030-01-27 11:22:32.634   300.000 any    172.16.6.241    17( 6.4)  72.7 G( 6.4)   4.2 G(15.2)  242.4 M  112.9 M   0
2030-01-27 11:19:12.634   500.000 any    172.16.6.10     17( 6.4)  72.7 G( 6.4)   2.8 G( 9.9)  145.5 M  44.1 M    0
2030-01-27 11:16:42.634   650.000 any    172.16.6.14     17( 6.4)  72.7 G( 6.4)   3.1 G(11.1)  111.9 M  38.2 M    0
2030-01-27 11:16:42.634   650.000 any    172.16.6.129    17( 6.4)  72.7 G( 6.4)   2.5 G( 9.0)  111.9 M  30.9 M    0
2030-01-27 11:22:32.634   300.000 any    172.16.6.159    17( 6.4)  72.7 G( 6.4)   1.4 G( 5.2)  242.4 M  38.6 M    0
2030-01-27 11:20:02.634   450.000 any    172.16.6.234    16( 6.0)  68.5 G( 6.0)   1.5 G( 5.2)  152.1 M  25.9 M    0
2030-01-27 11:25:32.634   120.000 any    172.16.6.141    16( 6.0)  68.5 G( 6.0)   2.6 G( 9.4)  570.4 M  174.8 M   0
Summary: total flows: 265, total bytes: 27.9 G, total packets: 1.1 T, avg bps: 192.9 M, avg pps: 980.7 M, avg bpp: 0
Time window: 2030-01-27 11:12:32 - 2030-01-27 11:31:48
Total flows processed: 265, Blocks skipped: 0, Bytes read: 13836
Sys: 0.012s flows/second: 20389.3   Wall: 0.000s flows/second: 293466.2
```

Analizando estas estadísticas obtenidas no se observa algún valor que consuma el ancho de banda drásticamente, todos los flujos observados contienen valores aceptables (Consumen un tráfico no mayor al 20% del total de BW consumido). Sin embargo, se someterá este mismo lapso de tiempo establecido a un análisis más detallado mediante la creación deprofiles.

El primer profile creado, Monitoreo_redes, pretende observar la cantidad de tráfico que llega a cada servidor en el edificio A o B. La tabla 5.2 muestra los canales creados para este profile y los filtros aplicados a cada uno estos canales. El profile creado es de tipo continuo.

Tabla 5.2 Filtros aplicados para el profile "Monitoreo_redes"

Edificio	Canal	Acción Realizada	Filtro aplicado
A	1	Ver las conexiones que llegan al servidor de correos	src net 172.16.5/24 && dst ip 172.16.10.3 && dst port 25
	3	Ver las conexiones que llegan al servidor web	src net 172.16.5/24 && dst ip 172.16.10.4 && dst port 80
	5	Ver las conexiones que llegan al servidor de BD	src net 172.16.5/24 && dst ip 172.16.10.5 && dst port 3306
B	2	Ver las conexiones que llegan al servidor de correos	src net 172.16.6/24 && dst ip 172.16.11.6 && dst port 25
	4	Ver las conexiones que llegan al servidor web	src net 172.16.6/24 && dst ip 172.16.11.7 && dst port 80
	6	Ver las conexiones que llegan al servidor BD	src net 172.16.6/24 && dst ip 172.16.11.8 && dst port 3306

La figura 5.4 muestra la gráfica obtenida correspondiente al mismo periodo de tiempo indicado en el profile live. Esta gráfica muestra la actividad presente en los dos edificios, la información se ha clasificado mediante los filtros creados en el profile "Monitoreo_redes"

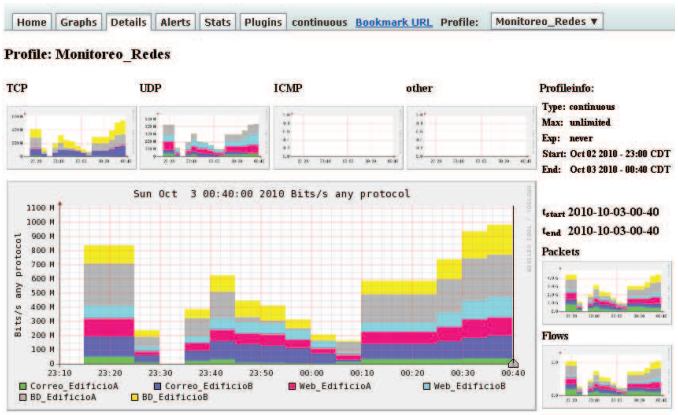


Figura 5.4 Grafica obtenida del profile “Monitoreo_redes”

En esta gráfica se observa que el máximo valor obtenido fue el 3 de octubre del 2010 a las 00:40 correspondiente a conexiones realizadas hacia el servidor de BD del edificio B. Seleccionando el máximo valor obtenido y sometiéndolo a un análisis, se observa que los siguientes flujos han generado este comportamiento:

```
** nfdump -M /Listry-AIGC/nfsen-1.3.2/profiles-data/Monitoreo_Redes/BD_EdificioB  -T  -r 2010/10/03/nfcapd.201010030040 -o long -c 20
nfdump filter: -c 6
Date flow  start        Duration      Proto     Src IP Addr:Port        Dst IP Addr:Port   Flags Tos   Packets    Bytes Flows
2030-01-27 12:59:06.634   650.000 TCP     172.16.6.129:2890  ->172.16.11.8:3306   0xff 255  4.3 G  147.4 M     1
2030-01-27 13:02:26.634   450.000 TCP     172.16.6.234:695   ->172.16.11.8:3306   0xff 255   4.3 G   91.0 M     1
2030-01-27 12:59:06.634   650.000 TCP     172.16.6.129:2841  ->172.16.11.8:3306   0xff 255  4.3 G  147.4 M     1
2030-01-27 13:02:26.634   450.000 TCP     172.16.6.234:630   ->172.16.11.8:3306   0xff 255   4.3 G   91.0 M     1
2030-01-27 12:59:06.634   650.000 TCP     172.16.6.129:2900  ->172.16.11.8:3306   0xff 255  4.3 G  147.4 M     1
2030-01-27 13:02:26.634   450.000 TCP     172.16.6.234:962   ->172.16.11.8:3306   0xff 255   4.3 G   91.0 M     1
Summary: total flows: 6, total bytes: 715.4 M, total packets: 25.7 G, avg bps: 8.8 M, avg pps: 39.5 M, avg bpp: 0
Time window: 2030-01-27 12:59:06 - 2030-01-27 13:09:56
Total flows processed: 68, Blocks skipped: 0, Bytes read: 3564
Sys: 0.020s flows/second: 3238.7   Wall: 0.002s flows/second: 27903.2
```

Mediante un análisis realizado a esta estadística obtenida, se observa que la dirección IP 172.16.6.129 esta generado una conexión con un tráfico mayor a 100Mb. Analizando esta dirección IP mediante estadísticas top N se observa lo siguiente:

```
** nfdump -M /Listry-AIGC/nfsen-1.3.2/profiles-data/Monitoreo_Redes/BD_EdificioB  -T  -r  2010/10/03/nfcapd.201010030040 -n 10 -s srcip/flows
nfdump filter: src ip 172.16.6.129 Top 10 Src IP Addr ordered by flows:
Date first seen        Duration Proto     Src IP Addr   Flows(%)     Packets(%)        Bytes(%)        pps        bps  bpp
2030-01-27 12:59:06.634   650.000 any 172.16.6.129   34(100.0)  145.5 G(100.0)   5.0 G(100.0)  223.8 M   61.7 M    0
Summary: total flows: 34, total bytes: 5.0 G, total packets: 145.5 G, avg bps: 61.7 M, avg pps: 223.8 M, avg bpp: 0
Time window: 2030-01-27 12:59:06 - 2030-01-27 13:09:56
Total flows processed: 68, Blocks skipped: 0, Bytes read: 3564
Sys: 0.006s flows/second: 9717.1   Wall: 0.000s flows/second: 184782.6
```

Por medio de las estadisticas Top N se observa que la dirección IP 172.16.6.129 esta generando un mayor tráfico que las demas direcciones pertenecientes al edificio B, sin embargo se observa un comportamiento normal: El plugin escaneo no arrojo algun comportamiento anormal generado en este lapso de tiempo, ademas el valor del tráfico no subió drásticamente.

El profile “puertos_conocidos” tiene el objetivo de clasificar la información de los dos edificios en base al puerto utilizado para su conexión, sin importar si es un puerto origen o puerto destino. La tabla 5.3 muestra la creación de los canales y los filtros aplicados para este profile.

Tabla 5.3 Filtros aplicados para el profile “puertos_conocidos”

Canal	Acción Realizada	Filtro aplicado
1	Ver las conexiones realizadas hacia FTP	port 20 || port 21
2	Ver las conexiones realizadas hacia SSH	port 22
3	Ver las conexiones realizadas hacia TELNET	port 23
4	Ver las conexiones realizadas hacia SMTP	port 25
5	Ver las conexiones realizadas hacia HTTP	port 80
6	Ver las conexiones realizadas hacia SNMP	port 161
7	Ver las conexiones realizadas hacia Netflow	port 9900

La figura 5.5 muestra la aplicación del profile “puertos_conocidos” en el mismo periodo de tiempo indicado en el profile live.

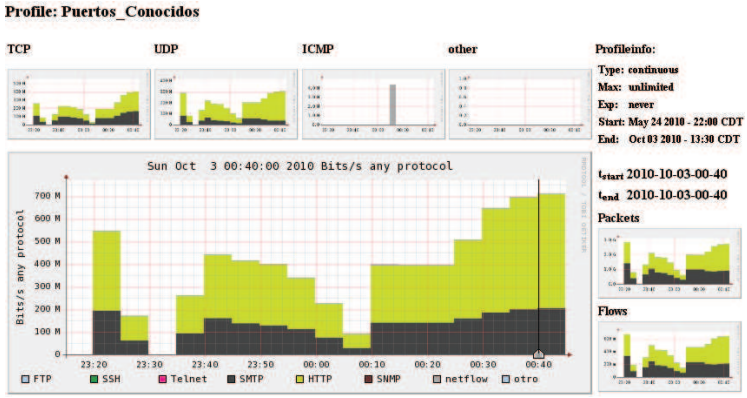


Figura 5.5 Grafica obtenida del profile “puertos_conocidos”

Analizando esta figura se observa que solo se tienen conexiones en el puerto 25 y 80 (en este profile no se creó un canal que este monitoreando la actividad generada por el servidor de B.D.); se observa que el servidor web está generando un mayor tráfico que el servidor de correos. Por medio de un análisis mediante estadísticas topN se observa lo siguiente:

```
** nfdump -M /Listry-AIGC/nfsen-1.3.2/profiles-data/Puertos_Conocidos/otro:netflow:SNMP:HTTP:SMTP:Telnet:SSH:FTP  -T  -r
2010/10/03/nfcapd.201010030035 -n 10 -s srcip/flows
nfdump filter: any
Top 10 Src IP Addr ordered by flows:
Date first seen             Duration Proto     Src IP Addr   Flows(%)     Packets(%)     Bytes(%)       pps    bps  bpp
2030-01-27 13:01:36.634   500.000 any     172.16.6.10    33(17.5)  141.2 G(17.5)   5.4 G(20.6)  282.4 M   85.6 M    0
2030-01-27 12:59:06.634   650.000 any     172.16.6.14    33(17.5)  141.2 G(17.5)   6.0 G(23.2)  217.2 M   74.2 M    0
2030-01-27 13:04:56.634   300.000 any     172.16.6.241   33(17.5)  141.2 G(17.5)   8.2 G(31.6)  470.6 M  219.1 M    0
2030-01-27 11:17:32.634   600.000 any     172.16.5.15    30(15.9)  128.3 G(15.9)  491.5 M( 1.9)  213.9 M    6.6 M    0
2030-01-27 11:22:32.634   300.000 any     172.16.5.12    30(15.9)  128.3 G(15.9)   1.5 G( 5.7)  427.8 M   39.3 M    0
2030-01-27 11:16:42.634   650.000 any     172.16.5.56    30(15.9)  128.3 G(15.9)   4.4 G(17.0)  197.5 M   54.4 M    0
Summary: total flows: 189, total bytes: 26.0 G, total packets: 808.6 G, avg bps: 30.6 M, avg pps: 119.0 M, avg bpp: 0
Time window: 2030-01-27 11:16:42 - 2030-01-27 13:09:56
Total flows processed: 189, Blocks skipped: 0, Bytes read: 10052
Sys: 0.005s flows/second: 31505.3   Wall: 0.000s flows/second: 353932.6
```

Analizando las estadísticas generadas se observa un tráfico normal en los servidores web y de correo, no se detecta un comportamiento anormal presente en estos servidores.

El ultimo profile creado, puertos, tiene el objetivo de monitorear las conexiones realizadas hacia puertos bien conocidos, registrados/dinámicos o privados sin importar si esta conexión fue realizada en un puerto origen o un puerto destino. La tabla 5.4 muestra los filtros creados en este profile.

Tabla 5.4 Filtros aplicados para el profile “puertos”

Canal	Acción Realizada	Filtro aplicado
1	Ver las conexiones realizadas por puertos altos (Dinámicos o privados)	port > 41151 && port < 65535
2	Ver las conexiones realizadas por puertos registrados	port > 1023 && port < 41152
3	Ver las conexiones realizadas por puertos bien conocidos	port > 0 && port < 1024

La figura 5.6 muestra la aplicación del profile “puertos” en el mismo periodo de tiempo indicado en el profile live.

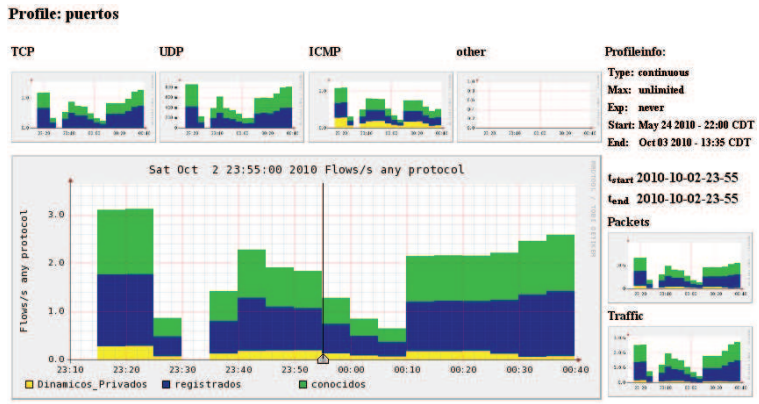


Figura 5.6 Grafica obtenida del profile “puertos_conocidos”

Analizando esta gráfica, se observa que se tienen mayores conexiones realizadas en puertos bien conocidos (en estos puertos opera el servidor de correos y el servidor web). Además se observan conexiones realizadas en puertos altos porque en el software “Paessler Netflow Generator” se simularon algunos puertos orígenes altos. Analizando un archivo nfcapd aleatorio por medio de estadísticas TopN se observa lo siguiente.

```
** nfdump -M /Listry-AIGC/nfsen-1.3.2/profiles-data/puertos/conocidos:registrados:Dinamicos_Privados -T -r
2010/10/02/nfcapd.201010022320 -n 10 -s srcip/flows
nfdump filter: any
Top 10 Src IP Addr ordered by flows:
Date first seen        Duration Proto      Src IP Addr   Flows(%)     Packets(%)     Bytes(%)       pps     bps  bpp
2030-01-27 11:21:48.634   600.000 any     172.16.5.156     90( 9.6)  385.0 G( 9.6)   1.6 G( 1.7)  641.7 M  21.2 M    0
2030-01-27 11:28:48.634   180.000 any     172.16.5.123     90( 9.6)  385.0 G( 9.6)   6.5 G( 6.8)   2.1 G  288.3 M    0
2030-01-27 11:26:48.634   300.000 any     172.16.5.12      60( 6.4)  256.7 G( 6.4)   2.9 G( 3.1)  855.6 M   78.6 M    0
2030-01-27 11:21:08.634   640.000 any     172.16.5.157     60( 6.4)  256.7 G( 6.4)   9.6 G(10.1)  401.1 M  120.4 M    0
2030-01-27 11:29:18.634   150.000 any     172.16.5.78      60( 6.4)  256.7 G( 6.4)  10.3 G(10.8)   1.7 G  550.5 M    0
2030-01-27 11:21:48.634   600.000 any     172.16.5.15      60( 6.4)  256.7 G( 6.4)  983.0 M( 1.0)  427.8 M  13.1 M    0
2030-01-27 11:30:48.634    60.000 any     172.16.5.10      59( 6.3)  252.4 G( 6.3)  966.6 M( 1.0)   4.2 G  128.9 M    0
2030-01-27 11:20:58.634   650.000 any     172.16.5.56      58( 6.2)  248.1 G( 6.2)   8.6 G( 8.9)  381.7 M  105.3 M    0
2030-01-27 11:25:32.634   120.000 any     172.16.6.141     52( 5.6)  222.5 G( 5.6)   8.5 G( 8.9)   1.9 G  567.9 M    0
2030-01-27 11:20:02.634   450.000 any     172.16.6.222     47( 5.0)  201.1 G( 5.0)   6.7 G( 7.0)  446.8 M  119.2 M    0
Summary: total flows: 935, total bytes: 95.6 G, total packets: 4.0 T, avg bps: 661.9 M, avg pps: 3.5 G, avg bpp: 0
Time window: 2030-01-27 11:12:32 - 2030-01-27 11:31:48
Total flows processed: 935, Blocks skipped: 0, Bytes read: 48704
Sys: 0.013s flows/second: 66800.0    Wall: 0.001s flows/second: 745019.9
```

En este archivo se observa que la subred 172.16.5.0/24 genera mayores conexiones hacia puertos conocidos que las demás subredes, este dato corresponde satisfactoriamente a la información mostrada en el profile live donde se observó que el edificio A tiene una mayor actividad que el edificio B.

Con los profiles creados se cumple con el objetivo de mostrar el software Nfsen enfocado en el monitoreo de red. Cabe destacar que se pueden crear profiles de acuerdo a las necesidades requeridas y por medio de los filtros creados en los profiles se pueden realizar análisis muy detallados.

5.2.2 Pruebas enfocadas en la detección de malware.

El objetivo de esta sección es mostrar el funcionamiento del plugin escaneo. En las pruebas anteriores se mostró el software Nfsen enfocado hacia el monitoreo de red y se simulo tráfico normal en el lapso de tiempo 02-Octubre-2010 21:40 al 03 Octubre-2010 00:40. Observando el plugin escaneo en un lapso de tiempo 2010-10-03 00:05 perteneciente al rango establecido se observa un tráfico normal sin presencia de anomalías delectadas, tal y como se analizó en los profiles creados. El resultado de la ejecución del plugin escaneo en este lapso de tiempo se observa en la figura 5.7.

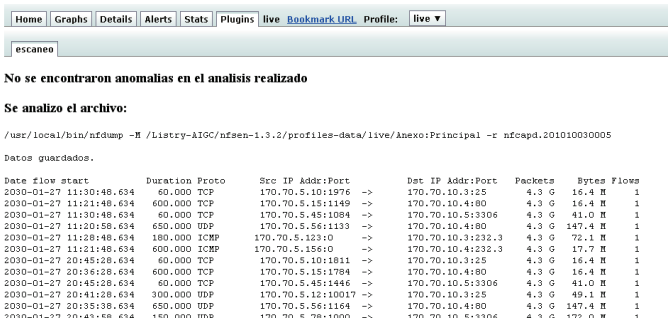


Figura 5.7 Comportamiento del plugin escaneo con tráfico normal

La figura 5.8 muestra el esquema utilizado en la realización de pruebas enfocadas en la detección de malware.

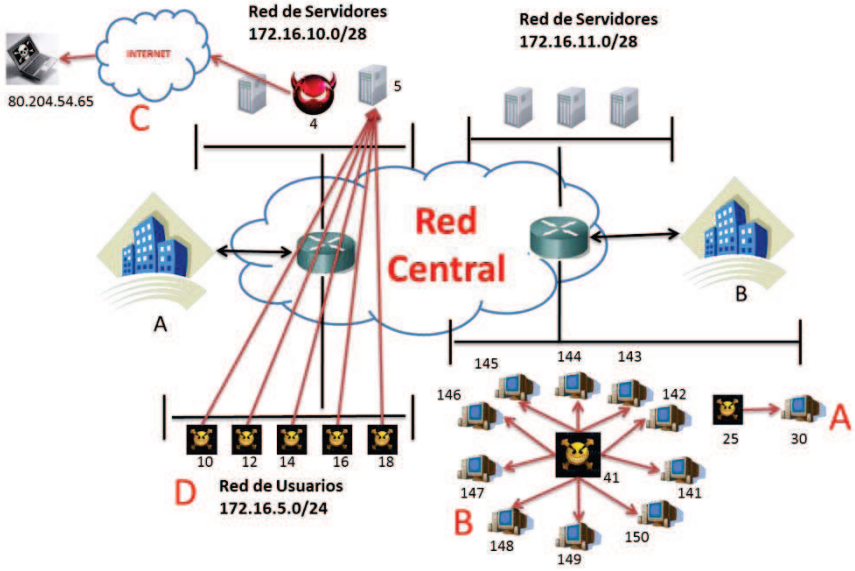


Figura 5.8 Esquema de pruebas realizadas simulando malware

En esta imagen se han simulado cuatro escenarios: cada escenario está representado por una letra y simula un comportamiento típico de algún malware. A continuación se describirá cada uno de ellos y se ejecutará el plugin 'escaneo' con el objetivo de observar los resultados que arroja el plugin.

5.2.2.1 Escenario A

En el escenario A se está simulando a una computadora infectada por el malware “blasser” con dirección IP 172.16.6.25. Este malware tratará de explotar la misma vulnerabilidad que encontró en el equipo “victima” para infectar mediante un escaneo de puertos a una computadora con dirección IP 172.16.6.30. Ambas computadoras se encuentran en la subred 172.16.6.0/24 perteneciente al edificio B. Simulado con ayuda del software “Paessles Netflow Generator” este comportamiento anormal y realizando un análisis con ayuda del plugin escaneo se detectó lo siguiente:

Figura 5.9 Escaneo de puertos encontrado

Como se observa en la figura 5.9 el plugin ‘escaneo’ ha detectado un escaneo de puertos realizado de forma secuencial en puertos altos el rango del 20000 al 20010. Además se observa que se ha notificado vía email sobre la anomalía y se ha creado un registro en la tabla ‘epuertos’ de esta anomalía detectada. La figura 5.10 muestra el uso del software OpenWebmail para visualizar los mensajes recibidos.

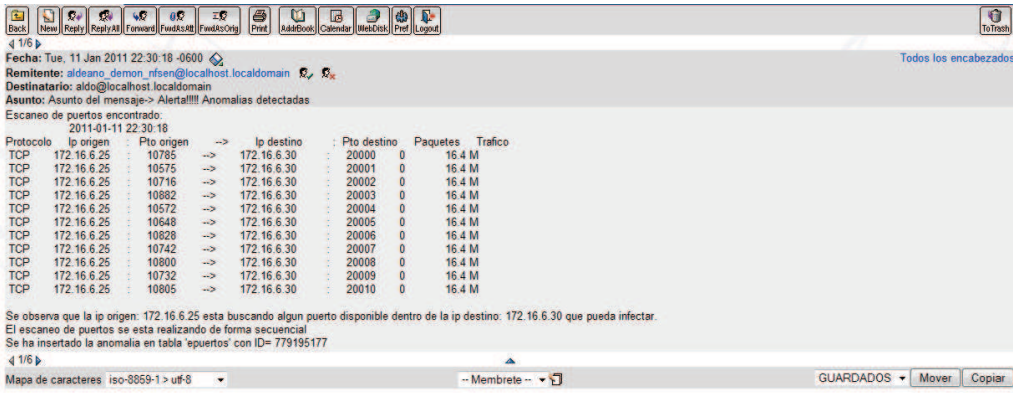


Figura 5.10 Notificación del escaneo de puertos encontrado

En la figura 5.11, mediante el uso del software Navicat, se observa que se ha creado un registro de la anomalía detectada en la tabla ‘epuertos’ con el ID ‘779195177’.

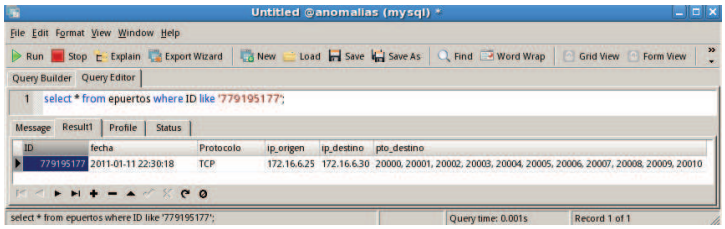


Figura 5.11 Registro de la anomalía guardado en la tabla 'epuertos'

5.2.2.2 Escenario B

En el escenario B se está simulando a una computadora infectada por el malware "conficker" con dirección IP 172.16.6.41. Este malware tratará de explotar la misma vulnerabilidad que encontró en el equipo "victima" para infectar mediante un escaneo de Ip'S a alguna computadora que se encuentre en el rango de direcciones Ip'S 172.16.6.141 al 172.16.6.150, perteneciente a la subred 172.16.6.0/24 creada en el edificio B. Simulado con ayuda del software "Paeesless Netflow Generator" este comportamiento anormal y realizando un análisis con ayuda del plugin escaneo se detectó lo siguiente:

Figura 5.12 Escaneo de IPS encontrado

Como se observa en la figura 5.12 el plugin 'escaneo' detecto un escaneo de IPS realizado de forma secuencial por la dirección IP origen 172.16.6.41 hacia un rango de direcciones IP destino 172.16.6.141 a 172.16.6.150 en el puerto destino 8060. Además se observa que se ha notificado vía email sobre la anomalía y se ha creado un registro en la tabla 'eips' de esta anomalía detectada. La figura 5.13 muestra el mensaje recibido.

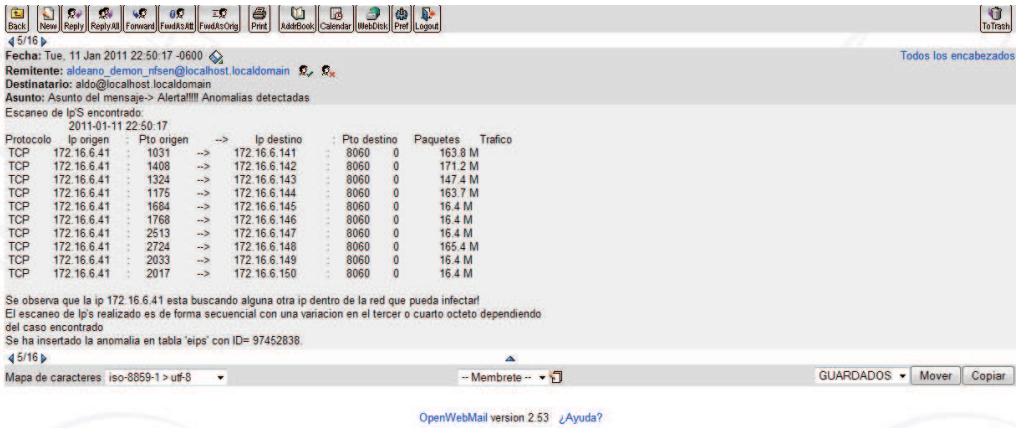


Figura 5.13 Notificación del escaneo de puertos encontrado

En la figura 5.14 se observa que se ha creado un registro de la anomalía detectada en la tabla 'eips' con el ID '97452838'

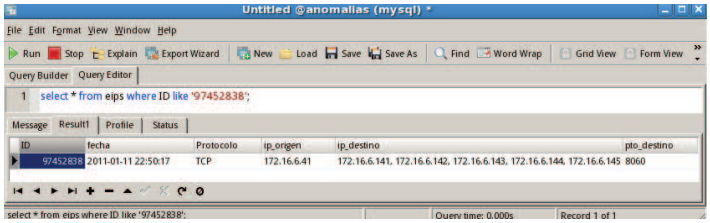


Figura 5.14 Registro de la anomalía guardado en la tabla 'epuertos'

5.2.2.3 Escenario C

En el escenario C se está simulando a un servidor web infectado por el malware "nspaint.exe". Este malware tratará de descargar a más malware y comprometer aún más al servidor web infectado con dirección IP 172.16.10.4. Adicionalmente el servidor infectado está enviando información hacia una computadora externa a la red de la institución con dirección IP 80.204.54.65 (IP asociada al creador del malware). Simulado con ayuda del software "Paessless Netflow Generator" este comportamiento anormal y realizando un análisis con ayuda del plugin escaneo se detectó lo siguiente:

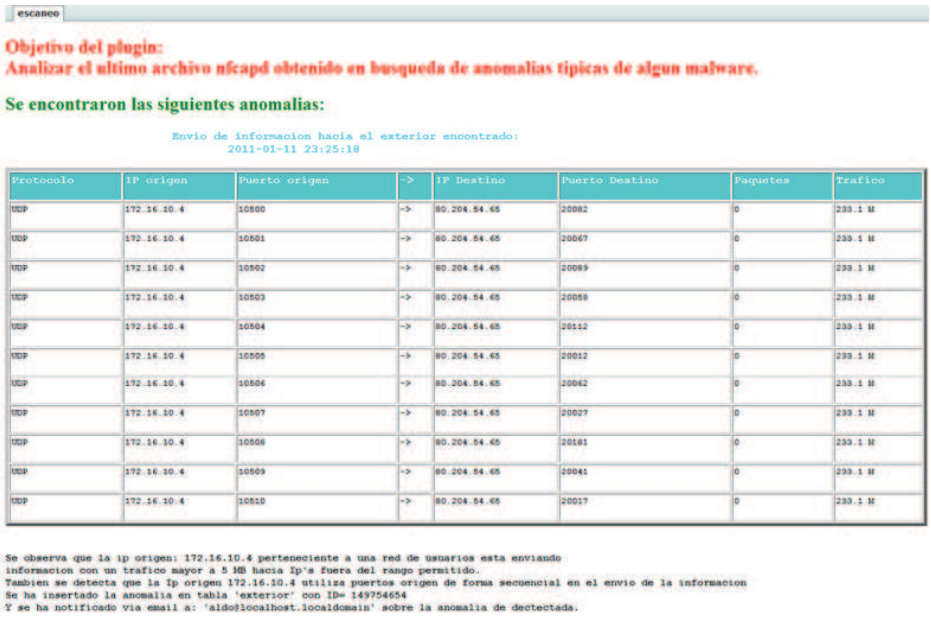


Figura 5.15 Fuga de información encontrada

Como se observa en la figura 5.15 el plugin ‘escaneo’ ha detectado una fuga de información del servidor de correo del edificio A con dirección IP 172.16.10.4, utilizando puertos orígenes secuenciales para él envió de información. Hay que resaltar que el plugin 'escaneo' detecte este tipo de comportamiento, la máquina infectada tendrá que enviar información con un tráfico mayor a 5MB y utilizar puertos origen de manera secuencial. Además se observa que se ha notificado vía email sobre la anomalía y se ha creado un registro en la tabla ‘exterior’ de esta anomalía detectada. La figura 5.16 muestra el mensaje recibido.

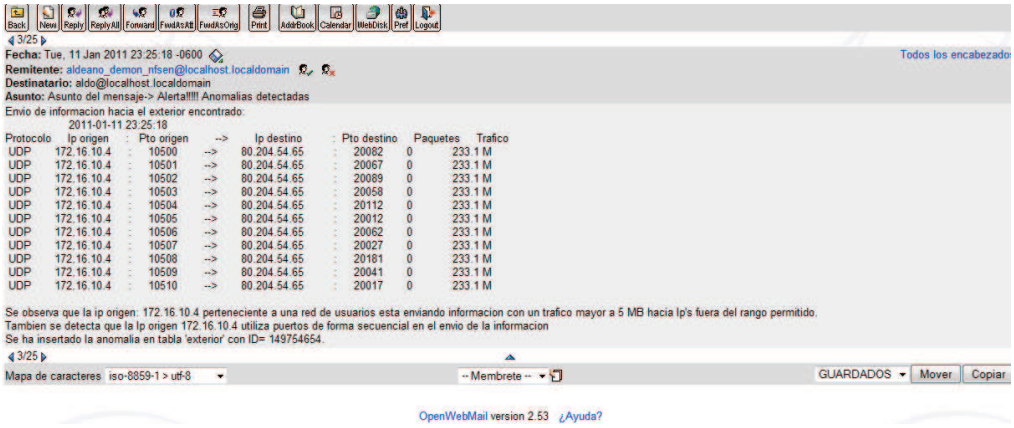


Figura 5.16 Notificación de la fuga de información encontrada

En la figura 5.17 se observa que se ha creado un registro de la anomalía detectada en la tabla ‘exterior’ con el ID ‘149754654’

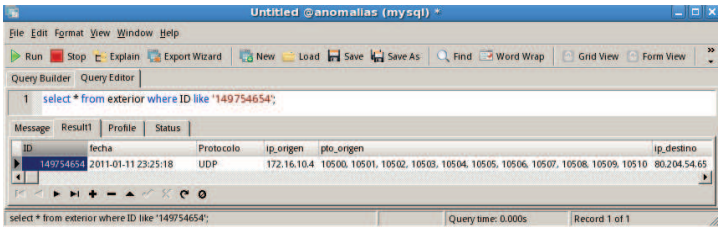


Figura 5.17 Registro de la anomalía guardado en la tabla ‘exterior’

5.2.2.4 Escenario D

En el escenario D se está simulando a un conjunto de computadoras, pertenecientes al rango de direcciones IP 172.16.5.10 al 172.16.5.18 del edificio A, que han sido infectadas por un malware “Postal.exe”. Este malware causa que las víctimas actúen como máquinas “zombies” y estén enviando múltiples peticiones hacia el servidor de bases de datos con dirección IP 172.16.10.5. El malware realiza esta acción con el objetivo de saturar o tirar al servidor. Simulado con ayuda del software “Paessless Netflow Generator” este comportamiento anormal y realizando un análisis con ayuda del plugin escaneo se detectó lo siguiente:


escaneo

Objetivo del plugin:
Analizar el ultimo archivo nfcapd obtenido en busqueda de anomalias tipicas de algun malware.

Se encontraron las siguientes anomalias:

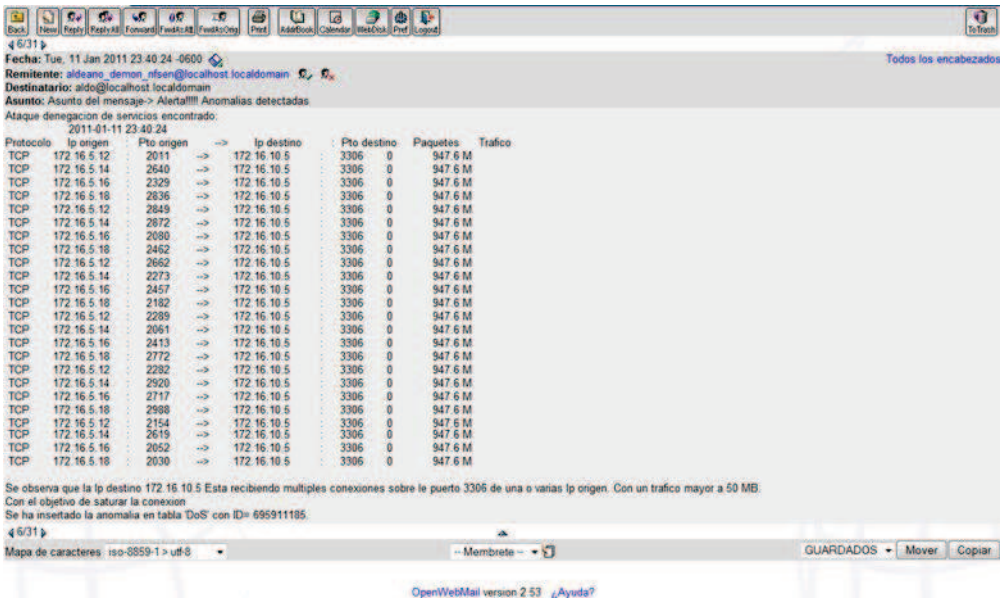
Ataque denegacion de servicios encontrado:
2011-01-11 23:40:24

Protocolo	IP origen	Puerto origen	->	IP Destino	Puerto Destino	Paquetes	Trafico
TCP	172.16.5.12	2011	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2640	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2329	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2836	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.12	2640	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2838	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2688	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2567	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.12	2927	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2619	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2052	->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2030	->	172.16.10.5	3306	0	947.6 M

Se observa que la Ip destino 172.16.10.5 Esta recibiendo multiples conexiones sobre le puerto 3306 de una o varias Ip origen. Con un trafico mayor a 50 MB. Con el objetivo de saturar la conexion
Se ha insertado la anomalia en tabla 'DoS' con ID= 695911185
Y se ha notificado via email a: 'aldo@localhost.localdomain' sobre la anomalia de dectectada.

Figura 5.18 Ataque de negación de servicios encontrado

Como se observa en la figura 5.18 el plugin ‘escaneo’ ha detectado a múltiples maquinas zombies que están realizando conexiones hacia el servidor de base de datos con dirección IP 172.16.10.5. Además todas las conexiones realizadas generan un tráfico anormal al que se tiene habitualmente en la institución. La figura 5.19 muestra la notificación recibida de esta anomalía que se detectó.


Back New Reply ReplyAll Forward FwdAtt FwdAttOrig Print AddrBook Calendar WebDisk Pref Logout ToTrash

6/31

Fecha: Tue, 11 Jan 2011 23:40:24 -0600 Todos los encabezados
Remitente: aldeano_demon_nfsen@localhost.localdomain
Destinatario: aldo@localhost.localdomain
Asunto: Asunto del mensaje-> Alerta!!!!! Anomalias detectadas

Ataque denegacion de servicios encontrado:
2011-01-11 23:40:24

Protocolo	Ip origen	Pto origen	-->	Ip destino	Pto destino	Paquetes	Trafico
TCP	172.16.5.12	2011	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2640	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2329	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2836	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.12	2849	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2872	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2080	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2462	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.12	2662	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2273	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2457	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2182	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.12	2289	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2061	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2413	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2772	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.12	2282	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2920	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2717	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2988	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.12	2154	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.14	2619	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.16	2052	-->	172.16.10.5	3306	0	947.6 M
TCP	172.16.5.18	2030	-->	172.16.10.5	3306	0	947.6 M

Se observa que la Ip destino 172.16.10.5 Esta recibiendo multiples conexiones sobre le puerto 3306 de una o varias Ip origen. Con un trafico mayor a 50 MB.
Con el objetivo de saturar la conexion
Se ha insertado la anomalia en tabla 'DoS' con ID= 695911185.

6/31

Mapa de caracteres iso-8859-1 > utf-8 -- Membrete -- GUARDADOS Mover Copiar

OpenWebMail version 2.53 ¿Ayuda?

Figura 5.19 Notificación de la fuga de información encontrada

En la figura 5.20 se observa que se ha creado un registro de la anomalía detectada en la tabla ‘exterior’ con el ID ‘695911185’

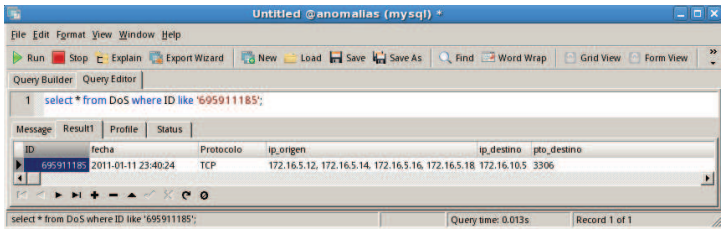


Figura 5.20 Registro de la anomalía guardado en la tabla 'DoS'

En estos ejemplos creados se decidió mostrar al plugin escaneo enfocado en la detección, de forma individual, de cada técnica implementada para detectar malware presente en la red central de la institución, Esto se realizó con el objetivo de poder hacer un análisis detallado en cada ejemplo creado y mostrar los resultados de la ejecución del plugin escaneo. Hay que señalar que este plugin es capaz de detectar las 4 técnicas implementadas de detección de malware de forma simultánea.

Con estos ejemplos creados se cumple con el objetivo de mostrar al software "Listry-AIGC" enfocado, mediante el plugin escaneo, en la detección de malware basado en patrones típicos de comportamiento.

Conclusiones

El objetivo de este trabajo de tesis fue implementar un detector de malware con software libre empleando el protocolo Netflow.

Para ello:

- Se realizó una investigación profunda acerca del protocolo Netflow para poder entender el funcionamiento de esta técnica enfocada en el monitoreo de red mediante la recolección de flujos directamente del switch o router que soporten dicha técnica, esta investigación fue descrita en la sección 1.3.
- Se realizó una investigación detallada acerca de alguna alternativa de uso libre que fuera capaz de sustituir, en buena medida, al software comercial Netflow que esta implementado en la institución, esta investigación fue descrita en el capítulo 2.
- Se implementó el software "Listry-AIGC" en la institución. El funcionamiento de este software fue descrito en el capítulo 3 y, en los anexos B y C.
- Se realizó una investigación acerca de las diversas técnicas utilizadas por el malware para infectar a sus víctimas en capas inferiores del modelo OSI. Obteniendo como resultado los cuatro comportamientos habitúales utilizados por el malware y que fueron descritos en la sección 4.3.
- Aprovechando la posibilidad de implementar pluglins en el software Nfsen, se creó el plugin 'escaneo', enfocado en la detección de malware de acuerdo con los comportamientos descritos en la sección 4.3. El funcionamiento de este plugin se describió en la sección 4.4.
- Se realizaron pruebas para demostrar el correcto funcionamiento del software "Listry-AIGC" enfocado en el monitoreo de red y en la detección de malware mediante el plugin escaneo. Las pruebas realizadas se describieron en la sección 5.2.1 y 5.2.2 respectivamente.

El software Nfsen implementado demostró ser la mejor alternativa de uso libre capaz de sustituir a software propietario que utiliza el protocolo Netflow. Debido a que brinda estadísticas muy detalladas de lo que está pasando en la red, y mediante filtros se puede acotar la información obtenida en estadísticas muy detalladas, además de permitir la creación de vistas específicas y alertas enfocadas en el monitoreo de red.

Además el software Nfsen ofrece grandes beneficios al poder ser enfocado, mediante plugins, a realizar alguna acción complementaria al monitoreo de red. En este caso se aprovechó esta ventaja para crear un plugin enfocado en la detección de malware. El plugin creado es capaz de detectar de forma individual o en conjunto las cuatro técnicas de detección de malware descritas en la sección 4.3. Con la principal ventaja de no requerir intervención alguna del usuario final, debido a que el plugin notifica automáticamente mediante él envió de un email la(s) anomalía(s) detectada(s) y guarda un registro de esta(s) anomalía(s) en una base de datos creada en MySQL para futuros análisis. Naciendo con estas herramientas añadidas el software "Listry-AIGC"

Para verificar el correcto funcionamiento del software "Listry-AIGC" enfocado en el monitoreo de red se realizaron las siguientes pruebas:

- El software "Listry-AIGC" se mantuvo en pruebas por 6 meses. En este periodo de tiempo, el software Nfsen recolectaba información de un router asociado a un edificio. Las estadísticas obtenidas eran en promedio 250 flujos capturados cada 5 minutos.
- Se crearon diversos profiles y alertas enfocados en el monitoreo de red y descritos en la sección 5.2.1.
- Se añadieron graficas que permiten visualizar las estadísticas obtenidas cada 3 horas. Además de acceder a la interfaz web mediante HTTPS (Hypertext Transferer Protocol Secure).

El plugin escaneo fue desarrollado como un módulo backend que se actualiza cada que el software Nfsen recibe un nuevo archivo Nfcapd y un módulo frontend que muestra los resultados obtenidos de la ejecución del módulo backend

Para verificar el correcto funcionamiento del software Nfsen enfocado en la detección de malware presente en la red central de la institución, se realizaron las siguientes pruebas:

- Se crearon cuatro escenarios distintos. Cada uno de estos escenarios simulaba el comportamiento de un malware mediante alguna de las técnicas descritas en la sección 4.3.
- Se observó la respuesta devuelta por el plugin escaneo ante cada uno de estos escenarios sometidos, descrita en la sección 5.2.2.
- Se verifico el correcto envió de los correos generados por el plugin escaneo al administrador de red.
- Se verifico el correcto almacenamiento de anomalías detectadas en las tablas creadas en MySQL

Con el desarrollo exitoso de este proyecto de tesis se logró tener una nueva herramienta de uso libre, que es capaz de sustituir en buena medida a software propietario que utiliza el protocolo Netflow. Además de desarrollar una técnica innovadora en la detección de malware que opera en capas inferiores del modelo OSI mediante el monitoreo de red.

Por todo lo descrito se cumplió con el objetivo establecido, ya que se encontró una alternativa de uso libre capaz de sustituir al software comercial Netflow implementado en la institución. Además de implementar un detector basado en la detección de malware por medio del monitoreo de red mediante patrones típicos de comportamiento en capas inferiores del modelo OSI.

Anexos

A

Glosario

ARP Spoofing. Técnica utilizada para infiltrase en una red. El principio del ARP Spoofing es enviar mensajes ARP falsos (falsificados, o spoofed) a la Ethernet. Normalmente la finalidad es asociar la dirección MAC del atacante con la dirección IP de otro nodo (el nodo atacado), como por ejemplo la puerta de enlace predeterminada (gateway). Cualquier tráfico dirigido a la dirección IP de ese nodo, será erróneamente enviado al atacante, en lugar de a su destino real

Background. Técnica utilizada en el monitoreo de red, se refiere a realizar un análisis detallado, generalmente al realizar un análisis mediante backgrund se llegan a tener estadísticas muy detalladas de los archivos analizados.

Bases de Datos. Es un "almacén" que nos permite guardar grandes cantidades de información de forma organizada para que luego sea posible acceder y utilizar esta información fácilmente

Crakear. Término utilizado cuando se aplican parches orientados a software propietario, que tienen el objetivo de alterar el funcionamiento del software original. Generalmente un software "crakeado" ocasiona que el software propietario al que ha sido aplicado el parche sea utilizado sin tener que pagar por su uso.

Debian. Es una comunidad conformada por desarrolladores y usuarios, que mantiene un S.O. GNU basado en software libre. Este sistema se encuentra precompilado, empaquetado y en un formato "deb". Debian nació como una apuesta por separar en sus versiones el software libre del software no libre. Por este motivo no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuirlo comercialmente mientras se respete su licencia.

Dirección IP. Se llama Dirección IP al número único asignado a un "host" en la red. Dichonúmero consta de 32 bits dividido en cuatro campos de 8 bits.Cada campo de 8 bits, es representado por un número decimal entre 0 y255, separado por periodos.

Cada dirección IPv4 identifica una red y un host único en cada red. Elvalor del primer campo determina cual porción de la dirección IP es elnúmero de la red y cual porción es el número del host. Los números dered están divididos en cuatro clases:

- Clase A (0.0.0.0 a 127.255.255.255)
- Clase B (128.0.0.0 a 191.255.255.255)
- Clase C (192.0.0.0 a 223.255.255.255)
- Clase D Multicast (224.0.0.0 a 239.255.255.255)

EITF (Internet Engineering Task Force). Es una organización internacional abierta de normalización, creada en Estados Unidos en 1986, que se encarga de regular las propuestas y los estándares de Internet, conocidos como RFC.

IP flooding. Ataque que se basa en la inundación masiva de la red mediante datagramas IP. Estos ataques se pueden utilizar para degradar el rendimiento de la red a la cual está conectado el perpetrador, generando paquetes con origen y destino aleatorio.

Además del degradado de la red, también pueden colapsar un equipo, con un ataque dirigido contra una víctima.

IP Spoofing. Ataque que tiene el objetivo de sustituir la dirección IP origen de un paquete TCP/IP por otra dirección IP a la cual se desea suplantar. Esto se consigue generalmente gracias a programas destinados a ello y puede ser usado para cualquier protocolo dentro de TCP/IP como ICMP, UDP o TCP. Hay que tener en cuenta que las respuestas del host que reciba los paquetes alterados irán dirigidas a la IP falsificada.

Estadísticas TOPN: Técnica utilizada en el monitoreo de red, que permite observar a los host o subredes que consumen el mayor BW.

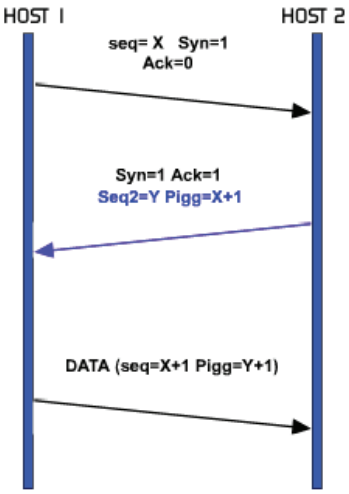
Estenografía. Rama perteneciente a la criptografía que permite ocultar información de distintas formas, como puede ser en imágenes, archivos, música, etc.

Falso Positivo. Son supuestos ataques generados por actividades legítimas. Generalmente son actividades normales clasificadas como anomalías. Ejemplo: El antivirus detecta un programa "casero" que genera múltiples conexiones como un gusano.

Falso Negativo. Son ataques reales considerados como actividades legítimas. Generalmente son anomalías clasificadas como actividades normales. Ejemplo: Múltiples conexiones de una misma IP hacia un servidor Web.

Handshake. Técnica utiliza en el protocolo TCP para conectar dos equipos electrónicos mediante los siguientes pasos:

- El servidor se mantiene a la espera de una conexión ejecutando las primitivas LISTEN y ACCEPT.
- El host que desea iniciar la conexión ejecuta una primitiva CONNECT especificando la dirección IP y el puerto con el que se desea conectar, el tamaño máximo del segmento que está dispuesto a aceptar. Entonces la primitiva CONNECT hace una apertura activa, enviando al otro host un paquete que tiene el bit SYN activado, indicándole también el número de secuencia inicial "x" que usará para enviar sus mensajes.



- El host receptor recibe el segmento revisa si hay algún proceso activo que haya ejecutado un LISTEN en el puerto solicitado. Si lo hay, el proceso a la escucha recibe el segmento TCP entrante, registra el número de secuencia "x" y, si desea abrir la conexión, responde con un acuse de recibo "x + 1" con el bit SYN activado e incluye su propio número de secuencia inicial "y", dejando entonces abierta la conexión por su extremo. El número de acuse de recibo "x + 1" significa que el host ha recibido todos los octetos hasta e incluyendo "x", y espera "x + 1" a continuación. Si no desea establecer la conexión, envía una contestación con el bit RST activado, para que el host en el otro extremo lo sepa.
- El primer host recibe el segmento y envía su confirmación, momento a partir del cual puede enviar datos al otro extremo, abriendo entonces la conexión por su extremo.

- La máquina receptora recibe la confirmación y entiende que el otro extremo ha abierto ya su conexión, por lo que a partir de ese momento también puede ella enviar datos. Con esto, la conexión ha quedado abierta en ambos sentidos.

ICMP (Internet Control Message Protocol). Protocolo estandarizado (RFC 792), que se utiliza como un medio de control y notificación de errores del Protocolo de Internet (IP). Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.

ICMP difiere del propósito de TCP y UDP ya que generalmente no se utiliza directamente por las aplicaciones de usuario en la red. La única excepción es la herramienta ping y traceroute, que envían mensajes de petición Echo ICMP (y recibe mensajes de respuesta Echo) para determinar si un host está disponible, el tiempo que le toma a los paquetes en ir y regresar a ese host y cantidad de hosts por los que pasa.

IDS (Intrusion Detection System). Mecanismo de seguridad encargado de detectas anomalías o actividad fuera de lo normal, que posiblemente se trate de un ataque o un falso.

Es importante mencionar que los IDS son mecanismos de detección no de prevención y son complementarios a mecanismos de seguridad existentes (Firewall, routers, etc.).

IPS (Intrusion Prevention System). Mecanismo de seguridad que monitorea la actividad en busca de anomalías, ataques o intrusiones y puede reaccionar de forma preventiva (bloquear) en tiempo real.

LAN (Local Area Network). Son redes de propiedad privada, de hasta unos cuantos kilómetros de extensión. Por ejemplo una oficina o un centro educativo.

Se usan para conectar computadoras personales o estaciones de trabajo, con objeto de compartir recursos e intercambiar información.

Están restringidas en tamaño, lo cual significa que el tiempo de transmisión, en el peor de los casos, se conoce, lo que permite cierto tipo de diseños (deterministas) que de otro modo podrían resultar ineficientes. Además, simplifica la administración de la red.

Suelen emplear tecnología de difusión mediante un cable sencillo al que están conectadas todas las máquinas.

- Operan a velocidades entre 10 y 100 Mbps.
- Tienen bajo retardo y experimentan pocos errores.

Live CD. Es un sistema operativo (normalmente acompañado de un conjunto de aplicaciones) almacenado en un medio extraíble, tradicionalmente un CD o un DVD, que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de archivos.

Algunos Live CD incluyen una herramienta que permite instalarlos en el disco duro. Otra característica es que por lo general no se efectúan cambios en la computadora utilizada, aunque algunos pueden almacenar preferencias si así se desea.

MAC (Media Access Control). Una dirección MAC es una dirección física de 48 bits que identifica a una computadora de forma única en una trama Ethernet o alguna otra tecnología utilizada en la capa 2 del modelo OSI. Generalmente una MAC se divide en los primeros 24 bits que indican la dirección del fabricante y los últimos 24 bits indican el número de serie de nuestra computadora.

La dirección MAC es utilizada por los switch para identificar a una computadora en un segmento de red.

Memoria caché: Memoria en la que se almacenas una serie de datos para su rápido acceso.

Memoria Volátil. Es aquella memoria cuya información se pierde al interrumpirse el flujo de corriente eléctrica.

Modelo OSI El modelo OSI fue creado en 1984, surgió como un método para estandarizar todas las redes. Antes de su aparición las redes presentes tenían problemas para comunicarse entre sí; debido a que no se contaban con una serie de normas que debieran de cumplir dichas redes por lo cual las empresas podían crear sus redes de cualquier forma y esto causaba problemas en la comunicación de unas redes con otras redes pertenecientes a otras empresas.

Objetivo del modelo OSI: Una o más computadoras y el software asociado, periféricos, operadores, procesos físicos y significado de las transferencias que forman algo autónomo, el cual es capaz de procesar y/o transferir información.

En el modelo de referencia OSI, hay siete capas numeradas, cada una de las cuales ilustra una función de red específica. Estas capas son:

1. **Capa física.** Provee la transmisión binaria:
    a. Cables, conectores, voltajes, velocidades de transmisión de datos.
2. **Capa de enlace de datos.** Provee un control directo de enlaces, acceso a medios:
    a. Provee la transferencia confiable de los datos a través de los medios
    b. Conectividad y selección de ruta entre sistemas.
    c. Direccionamiento lógico.
    d. Entrega de mejor esfuerzo.
3. **Capa de red.** Provee dirección de red y determinación de la mejor ruta.
    a. Provee transferencia confiable de los datos a través de los medios.
    b. Conectividad y selección de ruta entre los sistemas.
4. **Capa de transporte.** Provee la conexión extremo a extremo.
    a. Se ocupa de aspectos de transporte entre host.
    b. Contabilidad de transporte de datos.
    c. Establecer, mantener y terminar circuitos virtuales.
    d. Detección de fallas y control de flujo de la información.
5. **Capa de sesión.** Provee la comunicación entre host.
    a. Establece, mantiene y termina sesiones entre aplicaciones
6. **Capa de presentación:** Provee la presentación de los datos.
    a. Garantizar que los datos sean legibles para el sistema receptor.
    b. Formato de datos.

c. Estructuras de datos.
d. Negocia la sintaxis de transferencia de datos para la capa de aplicación.

7. **Capa de aplicación:** Provee procesos de red a aplicaciones
    a. Suministra procesos de red a los procesos de aplicaciones (como por ejemplo: correo electrónico, transferencia de archivos y emulación de terminales).

Password: También conocido como contraseña, es una clave utilizada para impedir que cualquier usuario pueda tener acceso a la información propietaria. Es recomendable utilizar contraseñas con un mínimo de 6 dígitos, incluyendo mayúsculas, minúsculas, números, símbolos y caracteres especiales para evitar el fácil robo de la contraseña por un perpetrador.

Perl. Lenguaje de programación diseñado por Larry Wall en 1987. Este lenguaje toma características del lenguaje C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesador de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

Ping of Death. Ataque que consiste en mandar numerosos paquetes ICMP muy pesados (mayores a 65.535 bytes) con el fin de colapsar el sistema atacado.

Perpetradores. También conocidos como atacantes, piratas informáticos o entidades maliciosas, son personas que poseen conocimientos de seguridad informática y orientan sus habilidades en la obtención ilegal de información, bienes o activos mediante el uso de diverso malware creado por el (ellos) o por terceras personas.

La clasificación de los distintos tipos de perpetradores es muy amplia, sin embargo, los que más destacan son: crackers, hackers, phreakers, entre otros.

PHP. Lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

Polimorfismo. Técnica utilizada por diversos malware (como virus informáticos y gusanos) para modificar partes de su código dificultando su detección.

Programa: Conjunto de instrucciones ordenadas correctamente que permiten realizar una tarea o trabajo especifico.

Protocolo. Es un conjunto de reglas específicas, relacionadas al formato y tiempo de los datos transmitidos entre dos dispositivos.

Router. Dispositivos activos que operan en la tercera capa del modelo OSI. Su función es la de conectar dos o más LAN y proveer una transmisión fiables de los paquetes enviados.

Los routers son capaces de proveer conectividad para mezclar ambientes MAC y poder trabajar con un protocolo en la capa superior. Esto permite la conexión de diferentes segmentos de red.

Sin embargo, los routers no son capaces de traducir protocolos utilizados en capas superiores, por lo que un router debe ser equipado con software apropiado para poder soportar dichos protocolos.

El rol del router es dirigir paquetes a lo largo de manera eficiente, utilizando algoritmos de enrutamiento para obtener la ruta más corta o económica.

Sistema Operativo: Conjunto de programas relacionados entre sí que permiten administrar y controlar los recursos de la computadora de manera segura y eficaz, de tal manera que permite conectarse al usuario con la máquina.

Switch. Dispositivo activo que opera en la capa dos del modelo OSI, su función es evitar tener colisiones en la red y asignar a cada puerto un nodo específico, logrando con esto una separación entre la red y una mejor transmisión de la información. También nos sirve para interconectar dos o más segmentos en la red, pasando de un segmento a otro de acuerdo a las MAC que tienen los dispositivos conectados a los switch.

TCP (Transmision Control Protocol). Protocolo estandarizado (RFC 793) que opera en la capa de transporte del modelo OSI. El protocolo TCP permite establecer una conexión entre dos equipos, garantizando la entrega de datos sin errores y en el mismo orden en que se transmitieron. Algunos protocolos de aplicación que operan sobre TCP son; HTTP, SMTP, FTP, SSH, entre otros.

UDP (User Datagram Protocol). Protocolo estandarizado (RFC 768) que opera en la capa de transporte del modelo OSI. El protocolo UDP está orientado a enviar mensajes sin establecer una conexión mediante datagramas, además este protocolo no garantiza la entrega secuencial de los paquetes enviados ni que todos los paquetes se reciban.

Red Hat. Es la compañía responsable de la creación y mantenimiento S.O. Red Hat Enterprise Linux, Fedora y Centos entre otros. Red Hat es famoso en todo el mundo por los diferentes esfuerzos orientados a apoyar el movimiento del software libre. No sólo trabajan en el desarrollo de una de las distribuciones más populares de Linux, sino también en la comercialización de diferentes productos y servicios basados en software de código abierto. Algunas de las contribuciones más notables han sido la creación de un sistema de empaquetación de software (RPM), y varias utilidades para la administración y configuración de equipos, como sndconfig o mouseconfig.

TopN. Técnica implementada en cualquier software de monitoreo de red que muestra los host o subredes que han consumido el mayor tráfico en un lapso de tiempo específico.

VPN (Virtual Private Network). Tecnología que permite la conexión virtual segura entre puntos remotos cuya localización geográfica impide que la red local sea física. Generalmente la información que viaja a través de este túnel, viaja encapsulada y a través de un túnel cifrado.

B

Guía de instalación del software Listry-AIGC

Requisitos para la instalación de Nfsen.

Nfsen ocupa las siguientes dependencias para poder ser instalado.

- Apache
- Perl y PHP
  - Perl > 5.6.0
  - PHP > 4.1
- Módulos de perl.
  - Mail:: Header, Mail:: Internet
- Herramientas RRD
  - Todos los gráficos netflow en NfSen requieren RRD. Por lo menos se requiere el módulo de Perl RRDs
- Herramientas Nfdump
  - Necesarias para recoger y procesar datos de Netflow
  - Instalar la versión 1.5.8

Todas estas dependencias se instalaron en un S.O. Centos 5.5 de 32 bits. En caso de requerir ser instalado en alguna otra versión de Linux se deberán de realizar los ajustes para dicha versión.

Instalación apache.

Normalmente en Centos viene instalado apache, esto se puede verificar de la siguiente forma:

```
# service httpd status
```

Si el script devuelve la leyenda “httpd: service desconocido”, significa que no se tiene instalado apache. La forma más fácil para su instalación es ejecutar el siguiente comando en una terminal:

```
# yum –y install httpd
```

Al momento de terminar la instalación, verificar que el servicio se encuentre levantado, en caso de estar detenido, se iniciara de la siguiente forma:

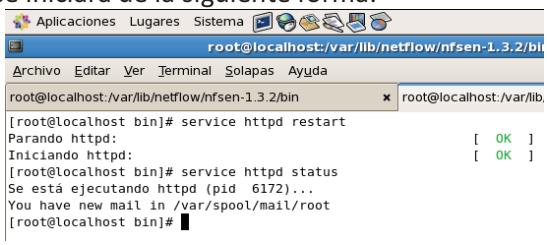


Figura B.1 Estado del servicio httpd

Para comprobar que apache ha sido iniciado exitosamente, abrir un navegador web y escribir la siguiente URL:

```
http://localhost:80
http://IP:80
```

Aparece una ventana como la siguiente:

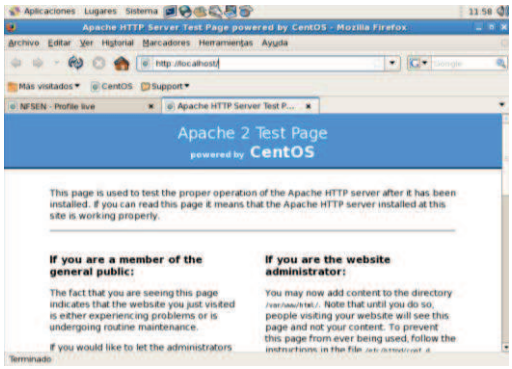


Figura B.2 Comprobación del servicio http en el servidor web.

Todos los archivos que se visualizan en apache están guardados en "/var/www/", más adelante se explica el procedimiento para crear directorios virtuales y poder visualizar carpetas con diferentes rutas.

Instalación PHP

Normalmente php viene instalado por defecto en CentOS, podemos verificar si se tiene instalado PHP, con el siguiente comando.

```
# php --version
```

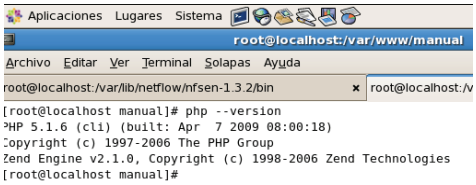


Figura B.3 Verificación de la correcta instalación de PHP.

En caso de no tener instalado php, se instalará de la siguiente forma:

```
# yum -y install php
```

Generalmente al instalarse PHP se configura automáticamente para que sea añadido a apache. Como comprobación, crear un archivo de prueba llamado "hola.php" y guardarlo en /var/www/manual

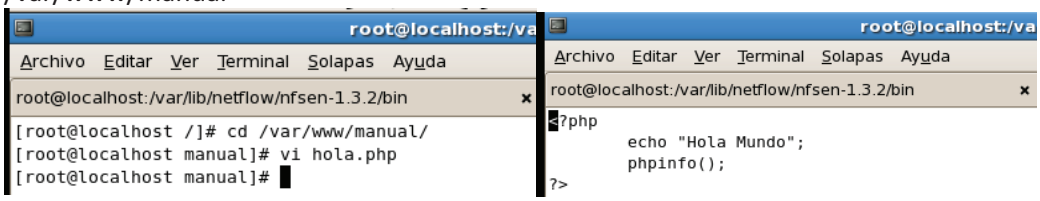


Figura B.4 Creación del script "hola.php".

Abrir en el navegador web y escribir lo siguiente

URL: http://localhost/manual/hola.php

El resultado del script ejecutado es:

Figura B.5 Resultado de la ejecución del script “hola.php”.

Instalación de perl

Normalmente Perl viene instalado por defecto en CentOS, se puede comprobar si se tiene instalado perl, tecleando en la terminal:

```
# perl –version
```

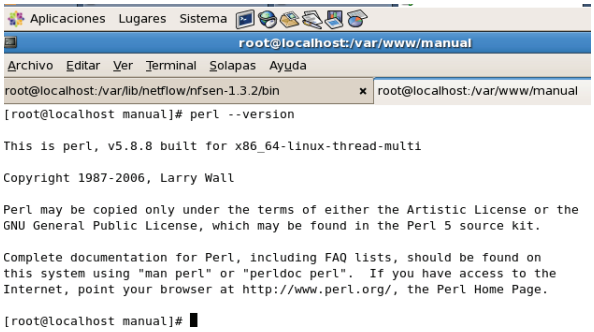


Figura B.6 Verificación de la correcta instalación de perl.

En caso de no tener instalado perl, se instala de la siguiente forma en la terminal:

```
# yum –y install perl
```

Dependencias de perl.

Nfsen requiere de las siguientes dependencias de perl:

- Mail::Header
- Mail::Internet

Se instalaran estas dependencias mediante la herramienta de perl “cpan”.

```
# perl –MCPAN –eshell
```

Esto hace que se cambie el prom a “cpan”. Para instalar las dependencias teclear.

```
cpan> install Mail::Header
cpan> install Mail::Internet
```

Al terminar la instalación de estas dependencias, salir de cpan.

```
cpan> exit
```

Instalación de RRDTool.

RRD (Round Robin Database) es un sistema encargado de almacenar y mostrar datos, este software se puede utilizar por medio de una terminal o en cualquier software disponible en ambiente gráfico. RRDTool es de gran utilidad debido a que genera graficas de los datos capturados.

Nfsen se apoya de RRDtool para generar graficas referentes al consumo de internet (BW ocupado, porcentaje de puertos ocupados, disponibilidad de servicios, etc.)

Para la instalación y configuración de RRDtool, realizar lo siguiente:

Dependencias requeridas por RRDtool:

- zlib
- libpng
- Cairo
- Glib
- Pango

Instalar estas dependencias mediante los siguientes comandos en la terminal:

```
# yum install cairo-devel libxml2-devel pango-devel pango libpng-devel freetype freetype-devel libart_lgpl-devel
```

Ahora es necesario descargar y descomprimir el software RRDTool

```
# cd /opt
# wget http://oss.oetiker.ch/rrdtool/pub/rrdtool-1.4.2.tar.gz
# tar –zxvf rrdtool-versión.tar.gz
```

Compilar e instalar rrdtool, ejecutando los siguientes comandos:

- Acceder al directorio descomprimido:

```
# cd rrdtool-version
```

- Exportar la variable "PKG_CONFIG_PATH" que tiene como referencia el archivo"pkgconfig":

```
# export PKG_CONFIG_PATH=/usr/lib/pkgconfig
```

- Compilar el programa

```
# ./configure
```

- Si no arroja ningún error la compilación, instalar

```
# make
# make install
```

- En caso de mostrar algún error el proceso de compilación, verificar si falta alguna dependencia que utilice el software RRDtool.

Nota: para la instalación de RRDtool en un S.O. CentOS de 32 bits, fue necesario instalar RRDtool de repositorios contenidos en la página oficial de la siguiente forma:

Descargar los archivos:

```
# wget http://dag.wieers.com/rpm/packages/rrdtool/perl-rrdtool-1.2.23-1.el5.rf.i386.rpm
# wget http://dag.wieers.com/rpm/packages/rrdtool/rrdtool-1.2.23-1.el5.rf.i386.rpm
# wget http://dag.wieers.com/rpm/packages/rrdtool/rrdtool-devel-1.2.23-1.el5.rf.i386.rpm
```

Instalar de la siguiente forma:

```
# rpm -ivh perl-rrdtool-1.2.23-1.el5.rf.i386.rpm rrdtool-1.2.23-1.el5.rf.i386.rpm rrdtool-devel-1.2.23-1.el5.rf.i386.rpm
```

Además es necesario instalar una librería extra de la siguiente forma:

```
# yum install libcap-devel
```

Opcional: Crear un acceso directo hacia el directorio donde se ha instalado rrdtool:

```
# cd /usr/local
# ln –s rrdtool-versión/ rrdtool/
```

Verificación de la instalación.

- Acceder hacia la siguiente ruta:
  # cd /opt/rrdttool-version/share/rrdtool/examples
- Ejecutar el siguiente script para verificar el correcto funcionamiento de rrdtool
  # ./stripes.pl
- Al momento de ejecutar este script, se genera una gráfica llamada "stripes.png", copiar esta grafica en el directorio de apache.
  # ls –l
  # cp stripes.png /var/www/manual/
- Verificar la gráfica obtenida en el navegador web.

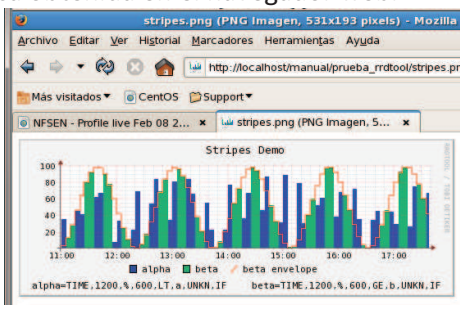


Figura B.7 Resultado de la ejecución del script "stripes.pl".

Instalación de Nfdump.

Nfdump es una herramienta la cual nos permite recolectar e interpretar datos provenientes de routers que soportan el protocolo Netflow, El colector Nfdump soporta las versiones 5, 7 y 9 de Netflow.

Para instalar Nfdump es necesario tener instalado los siguientes módulos de perl

- Mail::Header
- Mail::Internet

Instalación

- Descargar el software de la página oficial.
  http://sourceforge.net/projects/nfdump/
- Descomprimir el archivo descargado
  # tar xzvf nfdump-versión.tar.gz
- Configurar Nfdump, habilitando la opción "nfprofile":
  # ./configure --enable-nfprofile
- Al observar que la compilación fue exitosa, instalar el software
  # make
  # make install
- En caso contrario instalar las dependencias faltantes.
- Para mayor información del software Nfdump consultar:
  http://nfdump.sourceforge.net/
- Verificar la instalación en la terminal:
  # nfdump –V

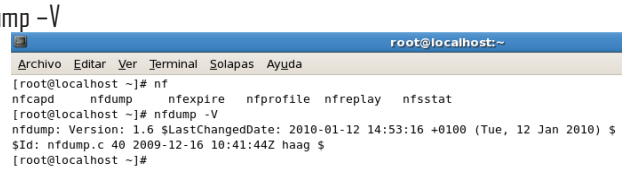


Figura B.8 Verificación de la correcta instalación del software nfdump.

NfSen.

NfSen provee una interfaz gráfica del colector Nfdump, para poder instalar el software Nfsen, es necesario tener instaladas correctamente todas las dependencias y software anteriores.

Nfsen crea una instalación de acuerdo al siguiente esquema:

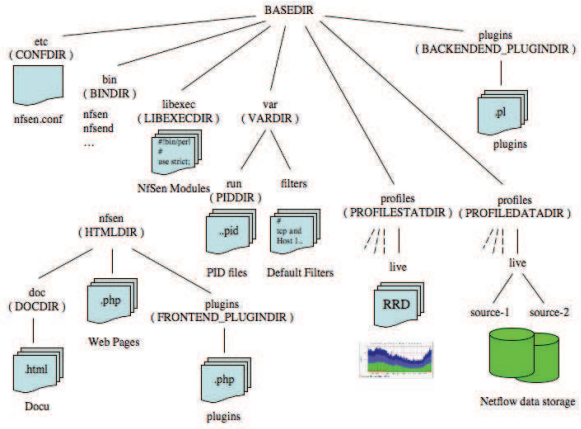


Figura B.9 Esquema de instalación del software Nfsen

Donde BASEDIR es la carpeta en donde se desea instalar el software NFsen.

Antes de instalar el software Nfsen, es necesario crear usuarios que solo serán utilizados por este software y configurar el archivo nfsen.conf de la siguiente forma:

Usuarios.

NfSen corre bajo un usuario local, por defecto “netflow”, por lo cual procedemos a crear el usuario "netflow" dentro del grupo APACHE.

```
# useradd –G apache –d /Lystri-AIGC/nfsen-version netflow
```

Es importante proporcionar permisos de lectura y escritura, ademas de cambiar al propietario de este grupo para el correcto funcionamiento del software Nfsen:

```
# chown netflow:apache ~netflow
# chmod 750 ~netflow
```

Instalación.

- Descargar el software Nfsen de la página oficial y descomprimirlo:
  - http://sourceforge.net/projects/nfsen/

    ```
    # tar xzvf nfsen-version.tar.gz
    ```

- Nfsen requiere de un archivo de configuración especial, el cual es necesario descargarlo:
  - http://nfsen.sourceforge.net/nfsen-dist.conf

- Sobrescribir el contenido de nfsen.conf con el archivo nfsen-dist.conf en *"etc"* (se encuentra dentro del directorio de instalación no en la carpeta /etc general)

  ```
  # cp nfsen-dist.conf etc/nfsen.conf
  ```

- Editar el archivo nfsen.conf cambiando las siguientes líneas:
  - $BASEDIR (Poner la ruta de instalación de Nfsen en este caso “/Lystri-AIGC/nfsen-version”
  - $WWWUSER/$WWWGROUP (Cambiar a apache)
  - %sources (Eliminar los dos ejemplos y añadir los flujos correspondientes, ver anexo C)

- Instalar Nfsen de la siguiente forma:

```
# ./install.pl etc/nfsen.conf
```

- El script de instalación solicita la ubicación de librerías de perl, verificar que tenga marcadas /usr/bin/perl y dar enter.

- Para ejecutar el software Nfsen, cambiar a su directorio de instalación y ejecutar el script "nfsen" para iniciar los servicios de la siguiente forma:

```
# cd /Lystri-AIGC/nfsen-version /bin
# ./nfsen start
```

Configurando directorios virtuales en apache.

Debido a cuestiones de seguridad, no es posible poder mostrar en el servidor apache todos los directorios que se encuentran en "/var/www".

Al momento de instalar el software Nfsen se crea el directorio "/var/www/Nfsen". Es necesario realizar configuraciones en el servidor apache, para poder visualizar este directorio creado de la siguiente forma:

- Crear un archivo de configuración dentro de httpd, este archivo contendrá las opciones necesarias para poder acceder al directorio y una opción extra de seguridad que pedirá autenticarse antes de entrar a este directorio:

```
# vi /etc/httpd/conf.d/nfsen.conf
```

- El nuevo archivo creado tendrá lo siguiente:

```
Alias /nfsen /var/www/nfsen
<Directory /var/www/nfsen/>
    DirectoryIndex nfsen.php
      Options -Indexes
      AllowOverride all
    order allow,deny
    allow from all
      AuthType Basic
      AuthUserFile /etc/httpd/conf/htpasswd.nfsen
      AuthName "Access"
    require valid-user
    </Directory>
```

- Estas opciones indican que para poder acceder al directorio es necesario autenticarse, además de indicar el directorio virtual al que se tendrá acceso en el servidor apache.
- Es necesario crear el archivo en donde se almacenará la contraseña correspondiente para que pueda acceder el usuario "netflow" al directorio creado.

```
# htpasswd –c /etc/httpd/conf/htpasswd.nfsen admin
```

- Escribir la contraseña.
- Reiniciar el servidor apache para poder visualizar los cambios realizados:

```
#service httpd restart
```

Acceder al software Nfsen por medio de la siguiente URL:
http://localhost/nfsen/index.php

Solucionando errores presentados al momento de acceder a nfsen.

Al momento de iniciar Nfsen por primera vez, se presentaron los siguientes errores:

- ERROR: nfsen connect() error: Permission denied!
- ERROR: nfsen – conecction failed!
- ERROR: Cannot initialize globals!

Estos errores se solucionaron de la siguiente forma:

- Verificar en el archivo "php.ini" que se tenga encendida la etiqueta "short_open_tag=on"

  # vi /etc/php.ini

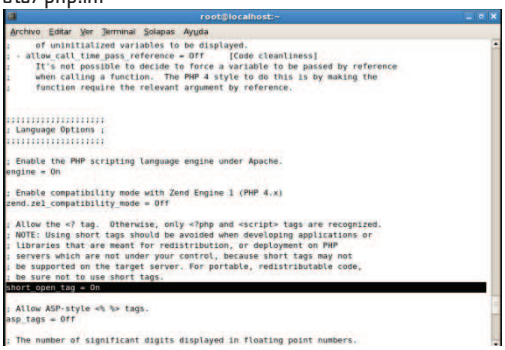


Figura B.10 Archivo "php.ini"

- Aplicar un parche en el archivo "Nfcomm.pm":

  "@@-770,6 +770,7@@
  return undef;
  }
  chmod 0660, $socket_path;
  + chown $NfConf::UID, $NfConf::GID, $socket_path;
  } else {
  # TCP Internet socket

Nota: la línea + es lo que se agregara a este archivo, quedando de la siguiente forma:

cd /var/lib/netflow/nfsen-versión/libexec
vi Nfcomm.pm

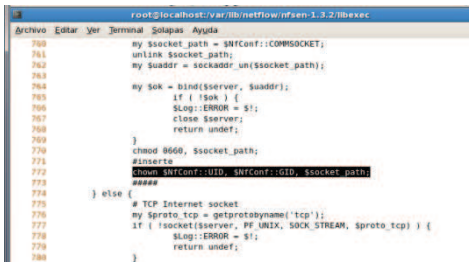


Figura B.11 Archivo "Nfcomm.pm"

Nfsen tiene un socket llamado "nfsen.comm", el cual se encarga de la comunicación del software, este socket necesita permisos de lectura y escritura para su correcto funcionamiento. Asignar estos permisos de la siguiente forma:

cd /var/lib/netflow/nfsen-versión/var/run
chmod 660 nfsen.comm
ls –l

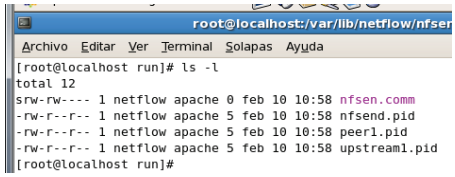


Figura B.12 Verificación de la asignación de correctos permisos al socket "nfsen.comm"

Reiniciar el software Nfsen

```
# cd /Lystri-AIGC/nfsen-version /bin
# ./nfsen restart
```

Ahora será necesario detener el firewall Selinux para probar la correcta ejecución de nuestro software de la siguiente forma.

- Sistema→Administración→Nivel de seguridad y Cortafuegos
  - Deshabilitamos SElinux y las opciones de cortafuegos.

Configuración de apache con el módulo HTTPS.

HTTPS es la versión segura del protocolo HTTP, inventada en 1996 por Netscape Communications Corporation. No es un protocolo separado de HTTP. Se trata de una combinación de este último con un mecanismo de transporte SSL o TLS, garantizando una protección razonable durante la comunicación cliente-servidor. Es ampliamente utilizado en la red mundial (WWW o World Wide Web) para comunicaciones como transacciones bancarias y pago de bienes y servicios.El servicio utiliza el puerto 443 por TCP para realizar las comunicaciones

Para la habilitación del módulo SSL en apache, es necesario realizar lo siguiente:

Instalar el módulo SSL

```
# yum -y install mod_ssl
```

A fin de mantener cierta organización, y un directorio dedicado para cada sitio virtual SSL, es conveniente crear un directorio específico para almacenar los certificados de cada sitio virtual SSL. Igualmente, por motivos de seguridad, debe ser solamente accesible para el usuario root.

```
# mkdir -mp 0700 /etc/ssl/midominio.org
```

Generando clave y certificado.

Se debe crear una clave con algoritmo RSA de 1024 octetos y estructura x509, la cual se cifra utilizado Triple DES (Data Encryption Standard), almacenado en formato PEM de modo que sea interpretable como texto ASCII. En el proceso descrito a continuación, se utilizan 5 ficheros comprimidos con gzip, que se utilizan como semillas aleatorias que mejoran la seguridad de la clave creada (server.key).

```
# openssl genrsa -des3 -rand \
  fichero1.gz:fichero2.gz:fichero3.gz:fichero4.gz:fichero5.gz \
  -out server.key 1024
```

Si se utiliza este fichero (server.key) para la configuración del sitio virtual, se requerirá de interacción del administrador cada vez que se tenga que iniciar, o reiniciar, el servicio httpd, ingresando la clave de acceso de la clave RSA. Este es el procedimiento más seguro, sin embargo, debido a que resultaría poco práctico tener que ingresar una clave de acceso cada vez que se inicie el servicio httpd, resulta conveniente generar una clave sin Triple DES, la cual permita iniciar normalmente, sin interacción alguna, al servicio httpd. A fin de que no se sacrifique demasiada seguridad, es un requisito indispensable que esta clave (fichero server.pem) solo sea accesible para root. Ésta es la razón por la cual se crea el directorio /etc/ssl/midominio.org con permiso de acceso solo para root.

```
# openssl rsa -in server.key -out server.pem
```

Opcionalmente se genera un fichero de petición CSR (Certificate Signing Request) que se hace llegar a una RA (Registration Authority o Autoridad de Registro), como Verisign, quienes, tras el correspondiente pago, envían de vuelta un certificado (server.crt) firmado por dicha autoridad.

```
# openssl req -new -key server.key -out server.csr
```

Lo anterior solicitará se ingresen varios datos:

- Código de dos letras para el país.
- Estado o provincia.
- Ciudad.
- Nombre de la empresa o razón social.
- Unidad o sección.
- Nombre del anfitrión.
- Dirección de correo.
- Opcionalmente se puede añadir otra clave de acceso y nuevamente el nombre de la empresa.

Si no se desea un certificado firmado por un RA, puede generarse uno certificado propio utilizando el fichero de petición CSR (server.csr). En el ejemplo a continuación, se crea un certificado con estructura X.509 en el que se establece una validez por 730 días (dos años).

```
# openssl x509 -req -days 730 -in server.csr \
  -signkey server.key -out server.crt
```

Con la finalidad de que solo el usuario root pueda acceder a los ficheros creados, se deben cambiar los permisos de éstos archivos a solo lectura para root.

```
# chmod 400 /etc/ssl/midominio.org/server.*
```

Configuración de Apache.

Crear la estructura de directorios para el sitio de red virtual.

```
# mkdir -p /var/www/midominio.org/{cgi-bin,html,logs,etc,var}
```

De todos directorios creados, solo /var/www/midominio.org/html, /var/www/midominio.org/etc, /var/www/midominio.org/cgi-bin y /var/www/midominio.org/var pueden pertenecer al usuario, sin privilegios, que administrará éste sitio de red virtual. Por motivos de seguridad, y a fin de evitar que el servicio HTTPD no sea trastornado en caso de un borrado accidental de algún directorio, tanto /var/www/midominio.org/ como /var/www/midominio.org/logs, deben pertenecer al usuario root.

Añadir al archivo /etc/httpd/conf.d/nfsen.conf el siguiente contenido:

```
NameVirtualHost *:443
<VirtualHost *:443>
  ServerAdmin root@localhost.localdomain
  DocumentRoot /var/www/localhost/html
  SSLEngine on
  SSLCertificateFile      /etc/ssl/localhost/server.crt
  SSLCertificateKeyFile   /etc/ssl/localhost/server.pem
  #SSLCertificateChainFile /etc/apache/domain.com/CA_issuing.crt
  ServerName localhost
<Directory /var/www/localhost>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
allow from all
</Directory>
</VirtualHost>
```

A fin de que surtan efecto los cambios, es necesario reiniciar el servicio httpd.

```
# service httpd restart
```

Instalación del software OpenWebmail

OpenWebmail es un proyecto libre, de código abierto (Open Source) que permite visualizar el correo electrónico de forma gráfica. Para la instalación de Openwebmail, realizar lo siguiente:

Descargar e instalar dependencias de perl ocupadas por OpenWebmail

```
#   cd /usr
#   wget http://packages.sw.be/perl-Text-Iconv/perl-Text-Iconv-1.4-1.2.el5.rf.i386.rpm
#   rpm –ivh perl-Text-Iconv-1.4-1.2.el5.rf.i386.rpm
```

Añadir e instalar el repositorio oficial de OpenWebmail a los repositorios de CentOS

```
#   cd /etc/yum.repos.d
#   lftpget http://openwebmail.org/openwebmail/download/redhat/rpm/release/openwebmail.repo
#   yum install openwebmail
```

Instalación y configuración de MySQL y Navicat.

Verificar si se tiene instalado MySQl:

```
#   rpm -q mysql mysql-server
```

En caso de no estar instalado, ejecutar:

```
#   yum -y install mysql mysql-server
```

Iniciar y configurar el servicio de MySQL para que se cargue al inicio del S.O.

```
#   service mysqld start
#   chkconfig --level 345 mysqld on
```

Entrar a MySQL y asignar contraseña al usuario root:

```
#   mysql
use mysql
update user set Password=PASSWORD('nuevo_password') where user='root';
```

Verificar que se ha asignado correctamente la contraseña y creación de la base de datos utilizada.

```
mysql –u root –p
create database anomalias;
```

Se utiliza el software Navicat para la administración de la BD creada. Descargar y descomprimir el software.

```
#   cd /Listry-AIGC
#   wget http://download2.navicat.com/download/navicat9_lite_en.tar.gz
#   tar –xzvf navicat9_lite_en.tar.gz
```

Iniciar el servicio de Navicat en modo "background":

```
#    /Listry-AIGC/navicat9_lite_en/start_navicat &
```

Al ejecutar el comando anterior aparece la ventana principal del software Navicat, configurar la conexión a MySQL, dar clic en “conection->MySQL” y asignar el nombre de usuario y contraseña.

Una vez realizado esto dar clic en “mysql->anomalías” para acceder a la base de datos creada.

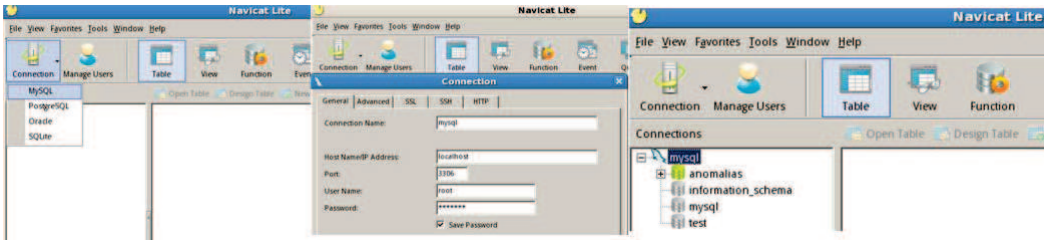


Figura B.13 Acceder al bases de datos mediante el software Navicat

Crear las tablas ‘epuertos’, ‘eips’, ‘exterior’ y ‘anomalias’.

Tabla ‘epuertos’, ‘eips’ y ‘DoS’

Tabla B.1 Campos creados en las tablas ‘epuertos’, ‘eips’ y ‘DoS’

Campo	Tipo	Longitud (bytes)	Llave primaria
ID	int	12	Sí
Fecha	datetime	0	Sí
Protocolo	varchar	1000	No
Ip_origen	varchar	5000	No
Ip_destino	varchar	5000	No
Pto_destino	varchar	5000	No

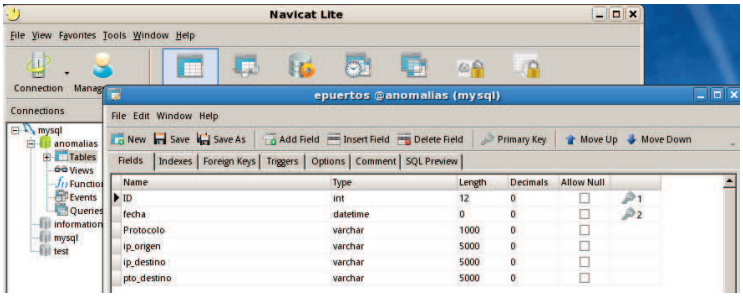


Figura B.14 Verificación de la correcta creación de las tablas ‘epuertos’, ‘eips’ y ‘DoS’

Tabla ‘exterior’

Tabla B.2 Campos creados en las tabla2 ‘exterior’

Campo	Tipo	Longitud (bytes)	Llave primaria
ID	int	12	Sí
Fecha	datetime	0	Sí
Protocolo	varchar	1000	No
Ip_origen	varchar	5000	No
Ip_destino	varchar	5000	No
Pto_origen	varchar	5000	No

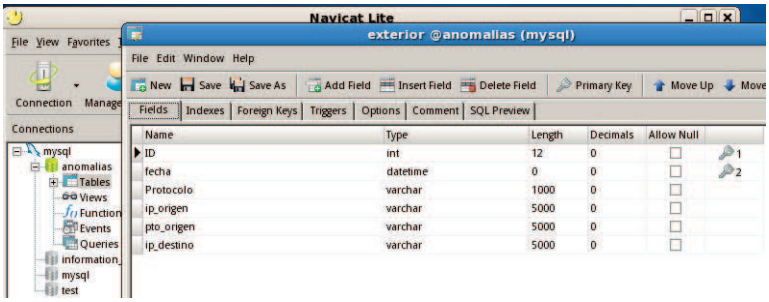


Figura B.15 Verificación de la correcta creación de la tabla ‘exterior’

Verificación de la correcta instalación del Software “Listry-AIGC”

Https y Nfsen.

En un navegador web, teclear la siguiente URL:
https://ip_server/nfsen/nfsen.php

El navegador web indica que no es una conexión confiable, obtener el certificado de seguridad.

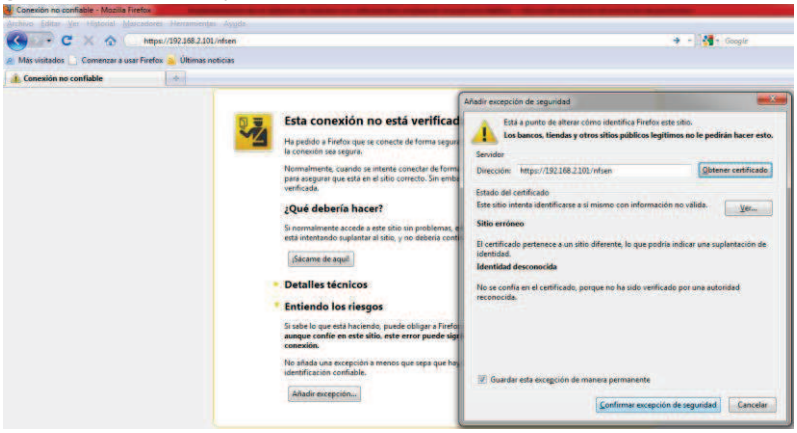


Figura B.16 Verificación de la correcta ejecución de HTTPS

Al momento de acceder aparece una ventana de autenticación, teclear:

- ✓ Usuario: admin
- ✓ Contraseña: *****,

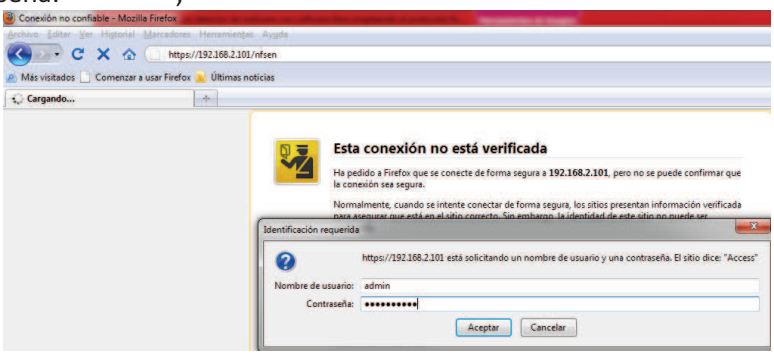


Figura B.17 Autenticación del software Nfsen

Finalmente se observa la página de inicio del software Nfsen.

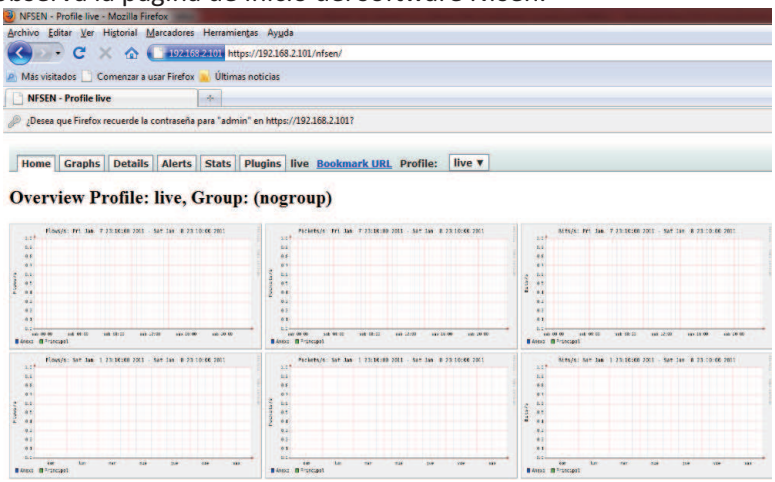


Figura B.18 Página de inicio del software Nfsen

OpenWebmail.

Para acceder al software OpenWebmail. Teclear en un navegador web la siguiente URL https://ip_server/webmail

Autenticarse con una cuenta que exista en el sistema y no sea la cuenta de root (por cuestiones de seguridad el software OpenWebmail deshabiliota esta cuenta.

Figura B.19 Autenticación del software OpenWebmail

Para la visualización del correcto funcionamiento del software OpenWebmail, se creó un script en perl que envía un correo electrónico, este script se llama "envía_correo.pl"

```
#!/usr/bin/perl
use strict;
my $fecha;
my @arreglo;
my $tmp;
open (MAIL,"|/usr/sbin/sendmail -t");
print MAIL "Content-Type: text/plain; charset=iso-8859-1\n";
print MAIL "To: aldo\@localhost.localdomain\n";
print MAIL "From: aldo\@localhost.localdomain\n";
print MAIL "Subject: Hola mensaje de prueba voy a enviar un email\n";
print MAIL "Content-Type: text/plain; charset=iso-8859-1\n";
print MAIL "Hola, este es un mensaje de prueba, saludos!!\n\n\n";
close (MAIL)
```

Ejecutar el script envia_correo.pl

```
#       perl envia_correo.pl
```

Observar en la página principal del software el nuevo correo.

Figura B.20 Verificación del correcto funcionamiento del software Openwebmail

Anexo C

Manual de usuario del software Listry-AIGC

Nfsen.

Verificación si los servicios están encendidos.

Antes de ejecutar el software Nfsen hay que verificar si el servidor apache esta encendido:

```
# service httpd status
```

En caso de que el servicio httpd se encuentre, encenderlo y añadirlo al nivel de arranque.

```
#       service httpd start
#       chkconfig –level 35 httpd on
```

Verificar el estado del servicio del software Nfsen:

```
#       cd /var/lib/netflow/nfsen-1.3.2/bin
#       ./nfsen status
```

De igual manera que con el servicio httpd, si nfsen se encuentra apagado encenderlo.

```
#       ./nfsen start
```

Añadir equipos a Nfsen.

Dirigirse a la carpeta de instalación del software Nfsen y abrir el archivo “nfsen.conf”:

```
#       cd/Listry-AIGC/nfsen-1.3.2/etc/
#       vi nfsen.conf
```

Buscar “Sources” y añadir los dispositivos activos, con el siguiente formato:

- 'ident' => { 'port' => '<portnum>', 'col' =>'<colour>', 'type' => '<type>' }
  - <Iden>: Nombre del equipo
  - <Portnum>: Número de puerto de escucha.
  - <Colour>: Formato del color (Escrito en hexadecimal).
  - <type>: Tipo de datos que recolectara (Netflow).

Ejemplo:

- 'NOMBRE' => { 'port' => '2001', 'col' => '#00aa00','type' => 'netflow' },

Añadir todos los equipos que generarán estadísticas en formato Netflow, separados por “,” :

- *%sources = (*
  'upstream1' => { 'port' => '9995', 'col' => '#0000ff', 'type' => 'netflow' },
  'peer1' =>{ 'port' => '9996', 'col' => '#ff0000' },
  'routin2' => { 'port' => '9997', 'col' => '#00aa00','type' => 'netflow' },
  );

Guardar el archivo de configuración y reiniciar el servicio de nfsen:

```
#       cd /var/lib/netflow/nfsen-1.3.2/bin
#       ./nfsen reconfig
```

El script preguntará si se desean borrar los datos antiguos de los equipos. Dependiendo de la opción que se seleccione, el script procede a configurar y añadir a los nuevos equipos. Al momento que el script termina de realizar los cambios, acceder a la interfaz web y observar los nuevos equipos añadidos en la página de inicio del software Nfsen.

Acceso a Nfsen

Página de inicio de Nfsen.

Para acceder a la interfaz web del software Nfsen, en un navegador web ejecutar la siguiente URL y autenticarse:

- https://ip_host/nfsen/nfsen.php
  - User: admin
  - Password: *****,

Aparece una ventana de inicio similar a la siguiente.

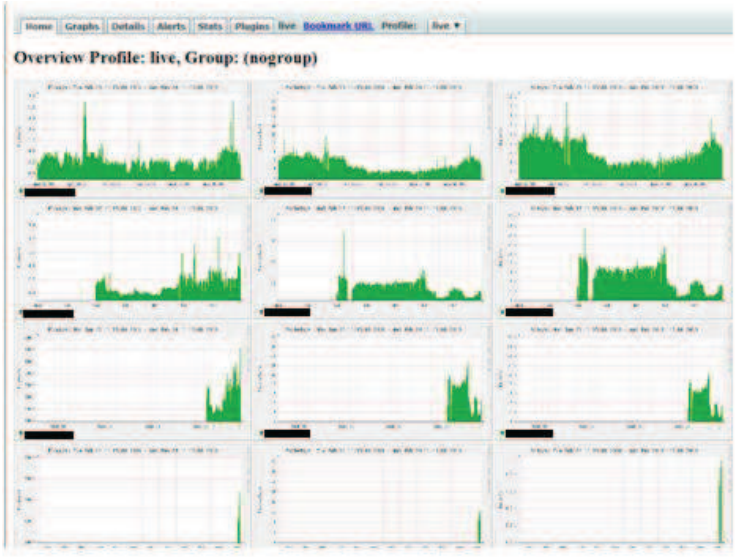


Figura C.1 Ventana de inicio del software Nfsen.

Nfsen por default captura información proveniente de los routers configurados cada 5 minutos y muestra esta información en el profile “live”, este profile captura todos los datos provenientes de los routers y gráfica estos datos.

Menú principal

La siguiente imagen muestra el menú de opciones del software Nfsen:

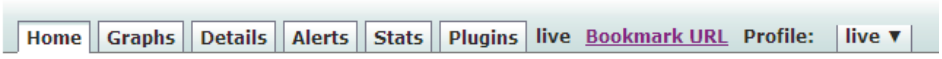


Figura C.2 Menu del software Nfsen.

Home

La opción **Home** se carga por default y muestra todas la gráficas generadas (hora, semana, día y mes).

Las gráficas están divididas en 4 grupos de 3 graficas cada grupo. De la siguiente forma.

- Gráfica los flujos, paquetes y el tráfico generado cada hora.
- Gráfica los flujos, paquetes y el tráfico generado cada día.
- Gráfica los flujos, paquetes y el tráfico que se generado cada semana.
- Gráfica los flujos, paquetes y el tráfico que se generado cada mes.

Graphs

La opción **Graphs** permite observar mejor las gráficas generadas.

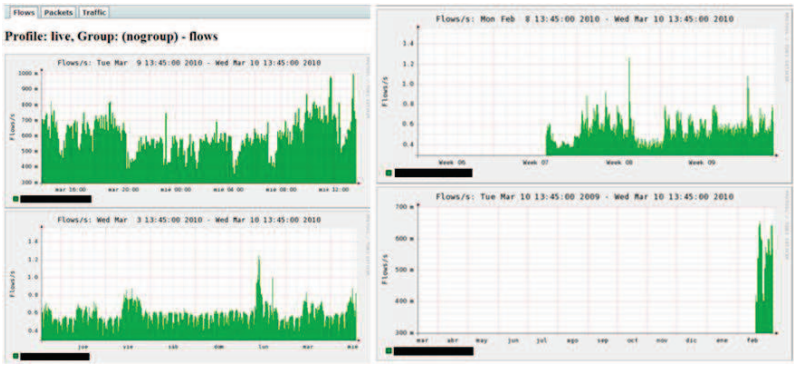


Figura C.3 Ventana correspondiente al menu “graphs” en el software Nfsen.

Tanto en la opción **Home**, como en la opción **Graphs**, es posible dar clic en cualquier gráfica y acceder a la opción **Details**, esta sección permite observar detalladamente el comportamiento de la gráfica seleccionada.

Details

Esta opción permite visualizar en la gráfica los datos capturados en tiempo real y, en base a estos datos, permite realizar análisis aplicando filtros con condiciones específicas.

Figura C.4 Ventana correspondiente al menu “details” en el software Nfsen.

En la imagen mostrada, se observa una gráfica principal, previamente seleccionada, dicha gráfica esta generada de manera general (sin tomar en cuenta ningún filtro) y muestra la cantidad de flujos capturados.

En la parte superior de la imagen, se observan cuatro gráficas, estas graficas desglosan la información de acuerdo a lo que se capturo en el protocolo TCP, UDP, ICMP y otros.

En la parte derecha de la gráfica principal se observan las gráficas obtenidas para los paquetes y el tráfico.

Las letras:

Muestran información de la gráfica en un punto de observación específico, de la siguiente forma:

- ***Type:*** Indica el tipo de profile que se está observando, en este caso es el profile default "Live".
- ***Max:*** Indica el máximo tamaño que tendrá el colector para guardar sus datos (en este caso ilimitado).
- ***Exp.*** Indica el tiempo en el cual el colector dejará de recolectar datos (en este caso nunca).
- ***Start.*** Indica la fecha en la cual se empezaron a capturar los datos.
- ***End.*** Indica la fecha (Momentánea) en la cual se termina de capturar los datos, normalmente es el último archivo capturado en el momento que se observa la gráfica.
- t_{start}**:** Tiempo específico que se está observando (En este caso es el 24/02/2010 00:15hrs). Esto es utilizado para obtener información sobre la cantidad de flujos, paquetes y el tráfico que se generó en ese momento en las tablas.
- t_{end}**:** Tiempo en el cual terminara de observarse (Utilizado en la opción "Time Window").

Las siguientes opciones permiten navegar en la gráfica:

Figura C.5 Panel de navegación en las gráficas del software Nfsen.

En el campo **select** se puede seleccionar:

- ***Single Timeslot:*** Muestra información de un flujo capturado.
- ***Time Windows:*** Permite seleccionar el tiempo de inicio y el tiempo final de la captura, en base a esto se observa un promedio de los flujos capturados en ese rango de tiempo.

En el campo ***display*** permite observar la graficas en diferentes periodos de tiempo (cada 12 horas, diario, cada 2 días, semanalmente, cada mes, etc.

Es posible observar un dato específico moviendo el cursor ubicado en la gráfica o dar clic en el área de interés.

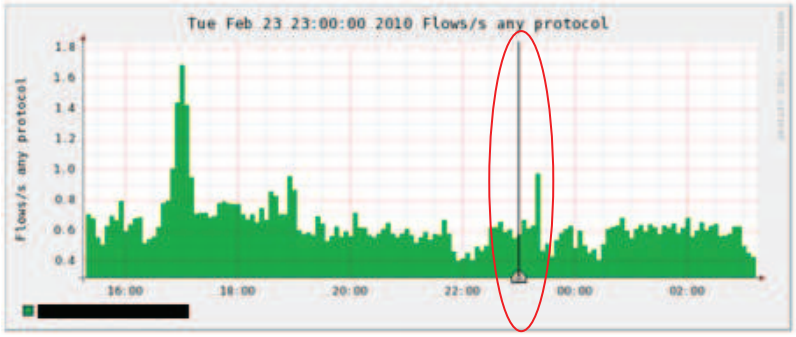
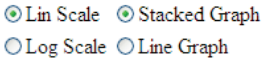


Figura C.6 Grafica creada en el software Nfsen.

- El botón "**<<**" permite regresar en la gráfica el tiempo seleccionado en "Display".
- El botón "**<**"permite regresar a la captura anterior.
- El botón "**|**"centra el cursor.
- El botón "**^**"permite centrar el cursor en el máximo pico encontrado.
- El botón **">"**permite ir a la captura siguiente.

- El botón **">>"**permite adelantar la gráfica el tiempo seleccionado en "Display"
- El botón **">|"**redirige el cursor al último valor capturado.

Las opciones ***"Lin Scale"***, ***"Stacked Graph"***, ***"Log Scale"***, ***"Line Graph"*** son utilizadas para dar formato a la gráfica. De la siguiente forma:



- **Lin Scale:** Escala de la gráfica en formato lineal.
- **Log Scale:** Escala de la gráfica en modo logarítmico.
- **Stacked Graph:** Dibuja todo el contorno de la gráfica.
- **Line Graph:** Solo dibuja la línea superior de la gráfica.

La parte ***"Stadistic"*** se observa una tabla que contiene información del total de los flujos, paquetes y el tráfico capturados en ese instante de tiempo, esta información se divide en cuatro filas (All, tcp, udp, icmp).

La información se puede visualizar en dos formas.

- "***Rate":*** Se observa la información en base al tiempo [S].

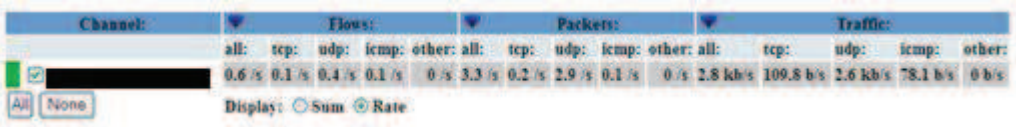


Figura C.7 Estadisticas correspondientes a la opción "Rate" en el software Nfsen.

- ***"Sum":*** Se observa la información en base al ancho de banda consumido.

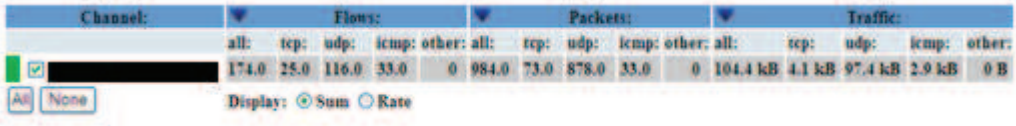


Figura C.8 Estadisticas correspondientes a la opción "Sum" en el software Nfsen.

La última parte de esta sección "**netflow processing**", permite aplicar filtros específicos para observar información de manera detallada.

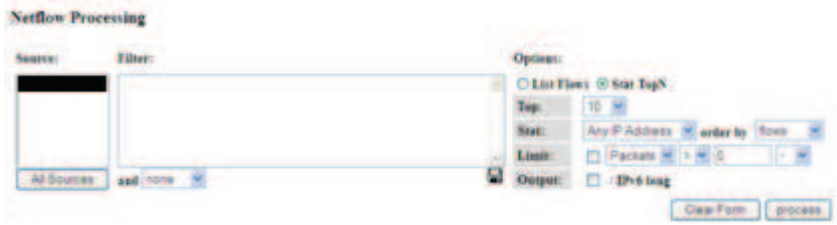


Figura C.9 Filtros a aplicar sobre flujos.

En la opción ***"source"*** seleccionamos el colector sobre el cual deseamos realizar un análisis más detallado.

Filtros.

Los filtros son operaciones específicas que permiten acotar la información. Para la creación de filtros se utiliza la siguiente sintaxis:

expr and expr, expr or expr, not expr, (expr)

Se puede aplicar múltiples filtros mediante el uso de los operadores lógicos:

- ✓ and
- ✓ or
- ✓ not

Los campos 'expr' pueden ser sustituidos por las siguientes opciones.

- Protocolo: **proto TCP, UDP, ICMP, GRE, ESP, AH, RSVP**
  o**proto num**(donde *'num'* es el número del protocolo)
- Puerto: ***port num*** (Donde 'num' representa el número del puerto).
- IP: ***ip a.b.c.d*** (donde 'a.b.c.d' representa el número de la dirección IP).
- IP Origen: ***src ip a.b.c.d***
- IP Destino: ***dst ip a.b.d.c***
- Red: ***net a.b.c.d m.n.r.s*** (donde 'm.n.r.s' representa la máscara correspondiente en decimal.
  Ó ***net a.b.c.d/num*** (donde 'num' representa los números de unos encendidos en la máscara.
- Red origen: ***src net a.b.c.d/num***
- Red destino: ***dos net a.b.c.d/num.***

Nfdump soporta las opciones de comparación: "**<, >, =, ==**"

- Packetes ***packets comparador num*** (donde 'num' será el número a comparar).
- Bytes ***bytes comparador num***
- Flujos ***flow comparador num***
- Bandera activa: ***flags Y*** (donde 'Y' es la letra inicial de la bandera utilizada).

Banderas. Nfdump provee de una serie de banderas que pueden estar activas e indicar información del estado de la red. Las banderas usadas por el colector Nfdump son las siguientes:

A ACK
S SYN
F FIN
R Reset
P Push
U Urgent
X All flags on

Ejemplos de la creción de filtros:

1) Realizar un filtro para observar todo el flujo que pasa a través de la dirección IP 192.168.1.32
   ip 192.168.1.32

2) Realizar un filtro para observar todo el flujo que pasa a través de la IP origen 192.168.1.32 hacia la IP 192.168.233.33
   src ip 192.68.1.32 and dst ip 192.168.1.33

3) Realizar un filtro que observe todo el flujo que pasa a través de la red origen 192.168.0.0 con la bandera SYN activada, u observar lo que pasa a través de la red destino 192.168.2.0 donde todos los bytes sean mayores a 100.
   (src net 192.168.0/24 and flags S) or (dst net 192.168.1/24 and bytes >100)

Alertas.

Una alerta es una acción específica que se ejecuta en el profile "live". Si se cumplen la(s) condicione(s) programadas en la alerta, se ejecutan ciertas acciones, como el enviar un correo electrónico al administrador o el ejecutar algún trigger o plugin que realizará alguna acción en específico.

Toda alerta debe de ser creada en el profile live y debe de cumplir con el siguiente esquema:

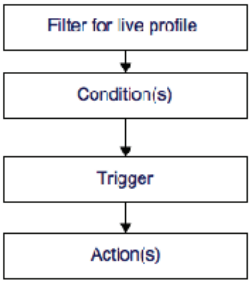


Figura C.10 Esquema de ejecución de una alerta.

En la opción ***"Alerts"*** del menú principal del software Nfsen es un donde se crean y administraran las alertas de la siguiente forma.

En el menú alertas, seleccionar el botón "+" (add new alert), aparece la siguiente ventana:

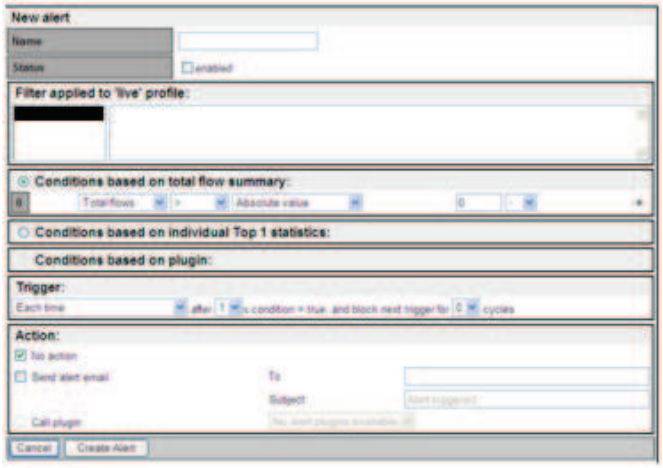


Figura C.11 Ventana de creación de una alerta

- ✓ La sección "Filter applied to 'live' profile:" es utilizada para aplicar filtros a esta alerta
- ✓ Seleccionar la opción "Enabled".
- ✓ Seleccionar alguna de las 3 opciones de acuerdo a la acción que se desee realizar en la alerta:
  - **Conditions based on total flow summary.** Esta opción permite habilitar varias condiciones que se deberán de cumplir para que la alerta sea ejecutada.

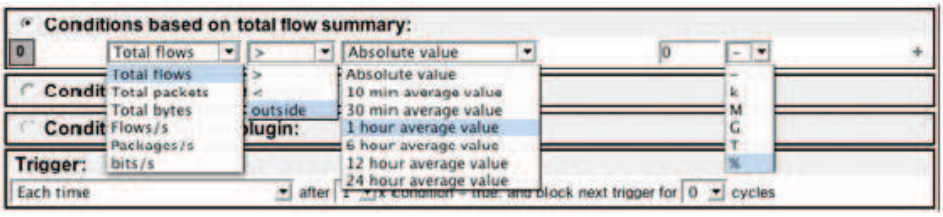


Figura C.12 Posibles condiciones a aplicar en una alerta

- **Conditions based on individual Top 1 statistics.** En esta opción, se deberá de cumplir la condición o condiciones deseadas, con la característica que se basara en estadísticas topN (primer lugar de la lista).

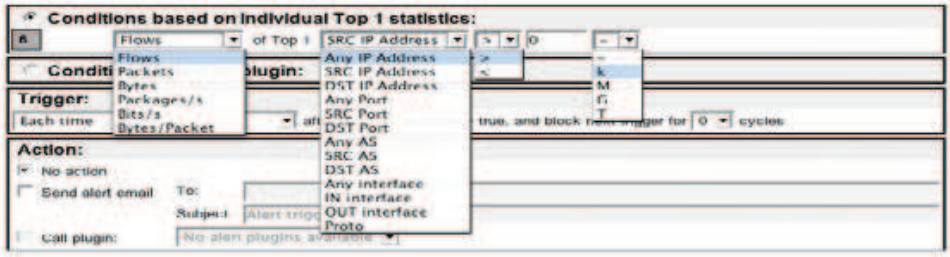


Figura C.13 Posibles condiciones a aplicar en una alerta basándose en estadísticas TOPN

- **Conditions based on plugin.**Un plugin es unaacción programada por el usuario, la cual se deberá de cumplir para que se ejecute la alerta.

Trigger.

Al momento en que las condiciones se cumplen, se ejecuta un trigger. Este trigger puede ser programado para que se ejecute una sola vez o se ejecute cada que la condición sea verdadera.

Al activarse un trigger realiza una acción específica, como puede ser: mandar un email al administrador o ejecutar algún plugin específico.

Estados de alerta.

inactive	Indica que la alerta no se está ejecutando.
armed	La alerta esta activa y es evaluada cada ciclo hasta que se cumpla la condición.
armed 1/3	La alerta esta activa y es evaluada cada ciclo. Al momento que una condición sea verdadera, se necesitará que las siguientes dos condiciones (los siguientes dos ciclos) sean verdaderos para que se ejecute esta alerta.
fired	La alerta esta activa y es evaluada cada ciclo. El trigger se ejecuta en el último ciclo y realiza la acción indicada.
fired --	Esta alerta solo se dispara una sola vez y volverá a estar activa, a menos que se reactive manualmente.
blocked 1:2	La alerta esta activa pero se bloquea cada dos ciclos. En el siguiente ciclo se continúa ejecutando normalmente.

Ejemplo de creación de alertas.

Crear una alerta que observe todo tráfico a través del protocolo tcp y que cumpla con las siguientes condiciones para que se ejecute:

- La cantidad de flujos/segundo sea mayor en un diez por ciento al valor promedio que fluye cada 30 minutos.
- La cantidad de paquetes/segundo sea mayor en un veinte por ciento al valor promedio que fluye cada 30 minutos.
- La cantidad de bit/segundo sea mayor en un diez por ciento al valor promedio que fluye cada 30 minutos.

- La alerta se active cada que se detecte un valor anormal en tres ciclos seguidos y se envié un email para notificar.

Filtro aplicado: proto tcp
Condiciones:

- Flows/s > 30 min average value + 20%
- && Packages/s > 30 min average value + 20%
- && Bit/s > 30 min average value + 10%

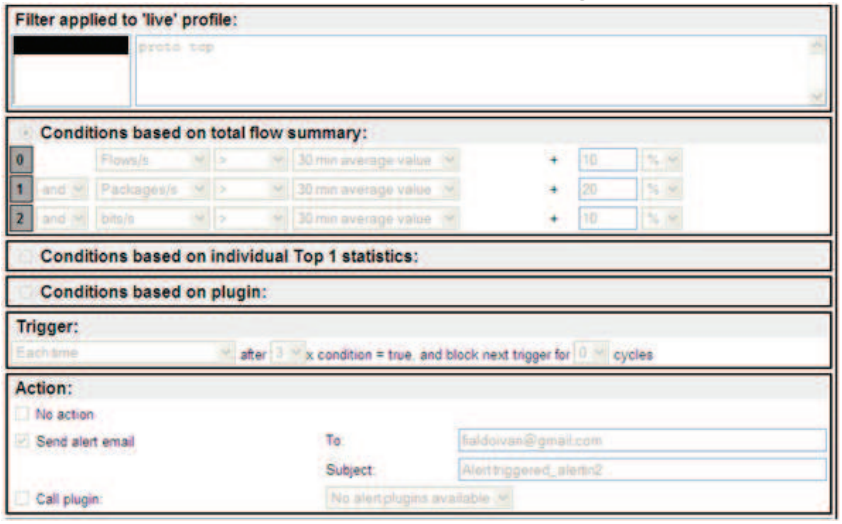


Figura C.14 Ejemplo de creación de una alerta

Después de ejecutar la alerta 8 horas, se observa la siguiente grafica generada:

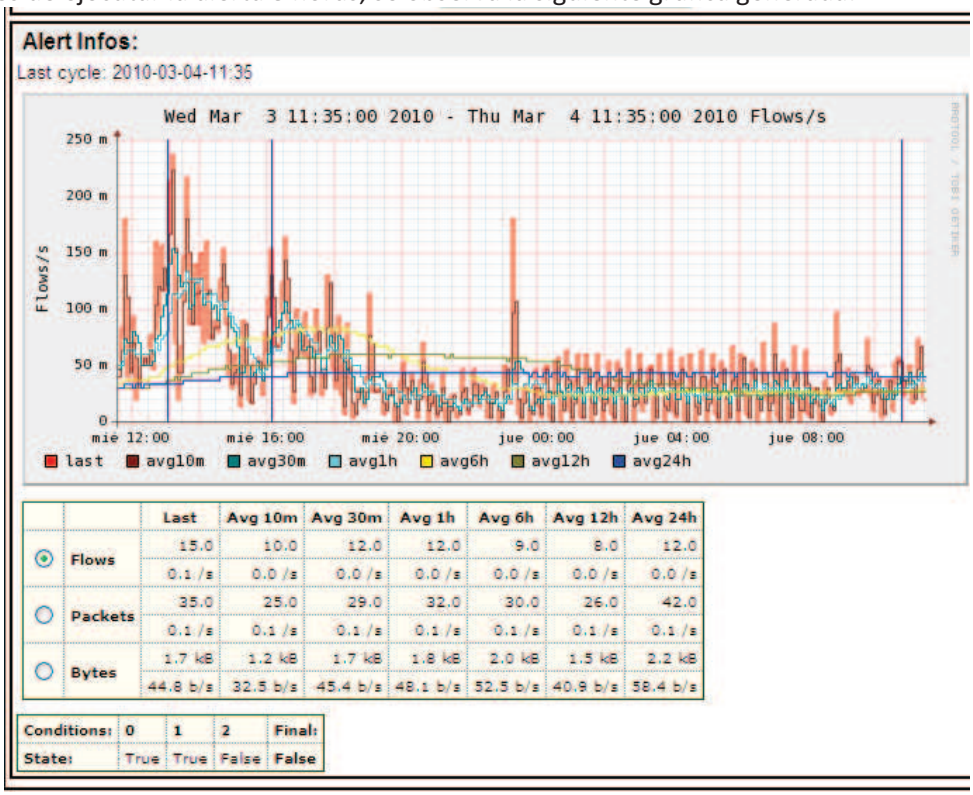


Figura C.15 Ejemplo de una alerta en ejecución

La gráfica mostrada es una comparación del valor obtenido en el ciclo pasado (cada cinco minutos) en comparación con los valores calculados cada diez minutos, treinta minutos, una hora, seis horas doce horas, veinticuatro horas.

En la tabla se muestran los valores obtenidos y las condiciones evaluadas. Toda condición se evalúa con lo calculado el ciclo pasado en comparación con las condiciones deseadas. Se aplica una operación and, en la cual todos los valores deben de cumplirse para activar el primer ciclo de la alerta.

Profile.

Un profile es un punto de vista específico de los routers. Se pueden crear diferentes profiles que realicen diferentes acciones, como observar el tráfico que pasa a través de puertos conocidos, u observar el comportamiento de ciertas direcciones IP o redes, etc.

Los profiles pueden pertenecer a estos grupos:

- **Históricos.** Se define el tiempo inicial y el tiempo final de observación de datos que se han obtenido en el pasado (datos ya capturados).
- **Continuos.** Este tipo de profile empieza a capturar datos en una fecha específica y se continuara ejecutando en cada actualización del software Nfsen.
- **Shadow.** Estos profiles no recolectan datos de con formato Netflow.

Canales.

Al momento de definir un profile, es posible anexar uno o varios canales. Un canal es un punto de observación específico, el cual genera datos que se observarán tanto en la gráfica como en la tabla correspondiente al instante de tiempo observado.

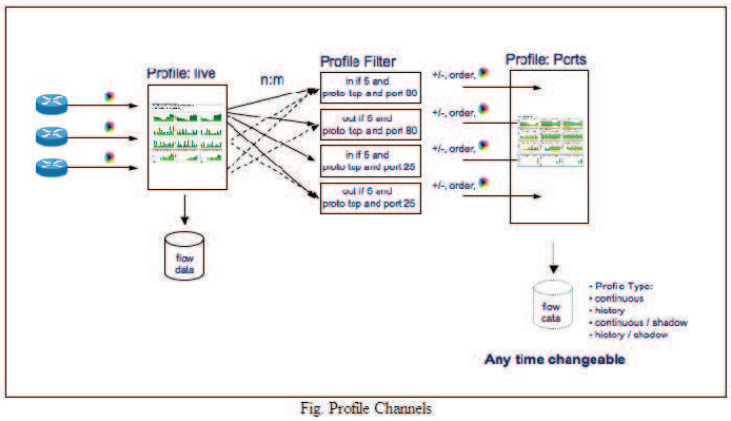


Fig. Profile Channels

Figura C.16 Esquema de creación de un profile

Al momento de definir un canal, es necesario aplicar un filtro específico.

Los canales pueden estar definidos en una o más fuentes de información (colectores) y son independientes del número de fuentes de información.

Creación de un profile.

En la pestaña live. Seleccionar new profile.

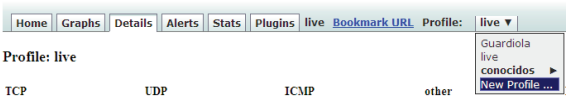


Figura C.17 Creación de un nuevo profile

Aparecerá la siguiente ventana:

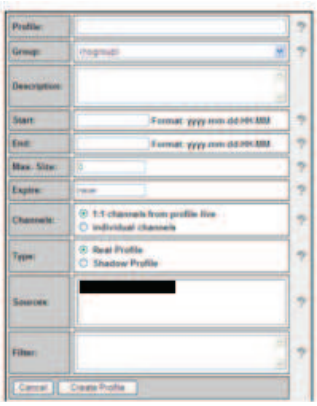


Figura C.18 Datos requeridos en la creación de un profile

Si se desea, agrupar el nuevo profile a un grupo. En la opción ***'start'*** seleccionar el día en el cual empezará a operar el profile (formato aaaa-mm-dd-hh-minmin), debe de existir alguna captura realizada por nfcapd que corresponda al día seleccionado.

Las opciones '**Max Size'** y '**Expire'** dependen del número de canales que se desean seleccionar y del tipo de profile.

En la opción '**channels'** se puede seleccionar alguna de estas dos opciones:

- **1:1 channels from profile live:** Al seleccionar esta opción solo se tiene 1 canal, en la opción '***filter'***, escribir el filtro.
- **Individual channels:** Al seleccionar esta opción el profile se compondrá de múltiples channels, cada canal creado deberá de tener su propio filtro.

En la opción '**type'** seleccionar el tipo de profile que se desea crear:

- **Real profile:** Profile que se basa en datos en formato Netflow y pude ser histórico o continuo.
- **Shadow profile:** Profile que no se basa en datos en formato Netflow.

En la opción '**sources'** seleccionar el colector (router) que se desea analizar.

Una vez seleccionados todos los datos, crear el profile seleccionando el botón "***create profile***". Si el profile se compone de un solo canal inmediatamente se creará y se podrá observar su funcionamiento, en caso de que se componga de múltiples canales el profile se crea, pero se tendrán que agregar los canales correspondientes.

Cuando el profile es creado, aparece la siguiente frase.

Esta frase Indica que se ha creado correctamente el profile. En el menú principal de Nfsen, seleccionar la opción "Stats" para observar la información creada del profile.

Para este ejemplo se ha creado un profile llamado 'protocolos', este profile pertenece al grupo 'conocidos' y contiene múltiples canales.

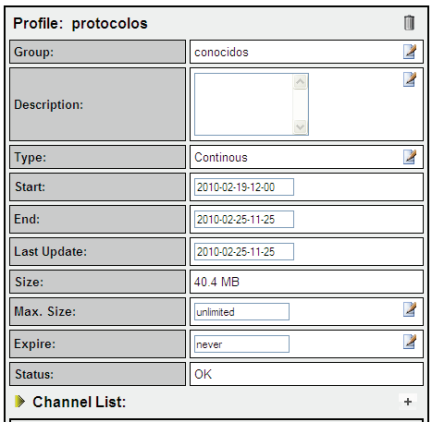


Figura C.19 Ejemplo de creación de un profile

Agregar un canal.

Ahora se procederá a crear los canales correspondientes para este profile.

Para cada canal que se desea agregar, dar clic en el botón "Add new channel" (+):

Figura C.20 Ventana de creación de un canal

Escribir el nombre del canal. En la opción "**Colour"** seleccionar el color que distingue a este canal (este color será para observar su comportamiento tanto en la gráfica como en la tabla), es posible seleccionar un amplio abanico de colores en la opción color ***'picker'***

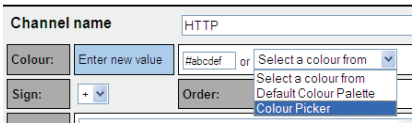


Figura C.21 Diversos métodos de seleccionar colores en el canal a crear

Una vez seleccionado el color. En la opción "**filter**", crear el filtro de acuerdo a las necesidades requeridas, en este ejemplo se aplicó el siguiente filtro:

- Port 80

En la opción **"sources",** seleccionar el colector a observar, se pueden seleccionar múltiples colectores. Para añadir estos colectores al nuevo canal, dar clic en el botón ***">> ".*** La fuente seleccionada se mueve de ***Avaliable Sources*** a **Selected Sources**.

Finalmente, dar clic en el botón ***'Add Channel'*** para agregar el nuevo canal en el profile protocolos.

Figura C.22 Ejemplo de creación de un canal

Quedando el profile de la siguiente forma:

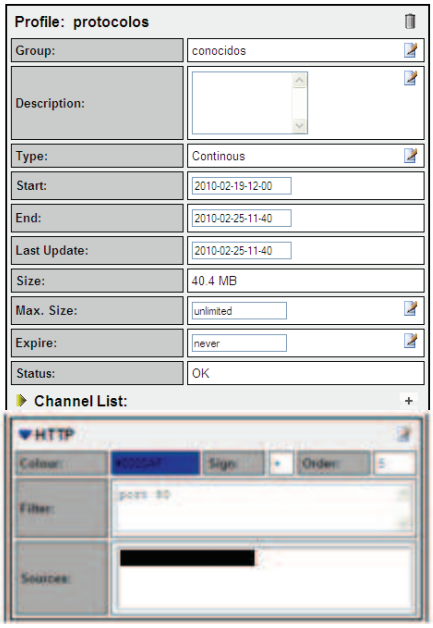


Figura C.23 Ejemplo de creación de un profile con sus canales

Para este profile además del puerto http se agregaron los puertos ftp, ssh, telnet, smtp, snmp con el mismo procedimiento. Una vez creados todos los canales, en la opción "***Status***" se observa la leyenda "new", dar click en ***"Commit new profile"*** (√).

En la siguiente figura, se observa el proceso de construcción del profile (en la parte status aparece build % - locked)

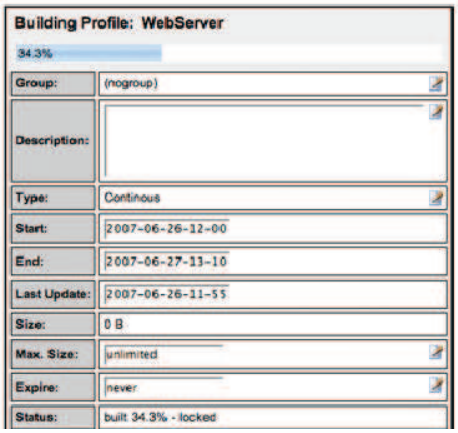


Figura C.24 Proceso de construcción de un profile

Al terminar de construirse el profile correctamente, se observa en la opción ***'status'*** la leyenda OK, quedando el profile de la siguiente forma:

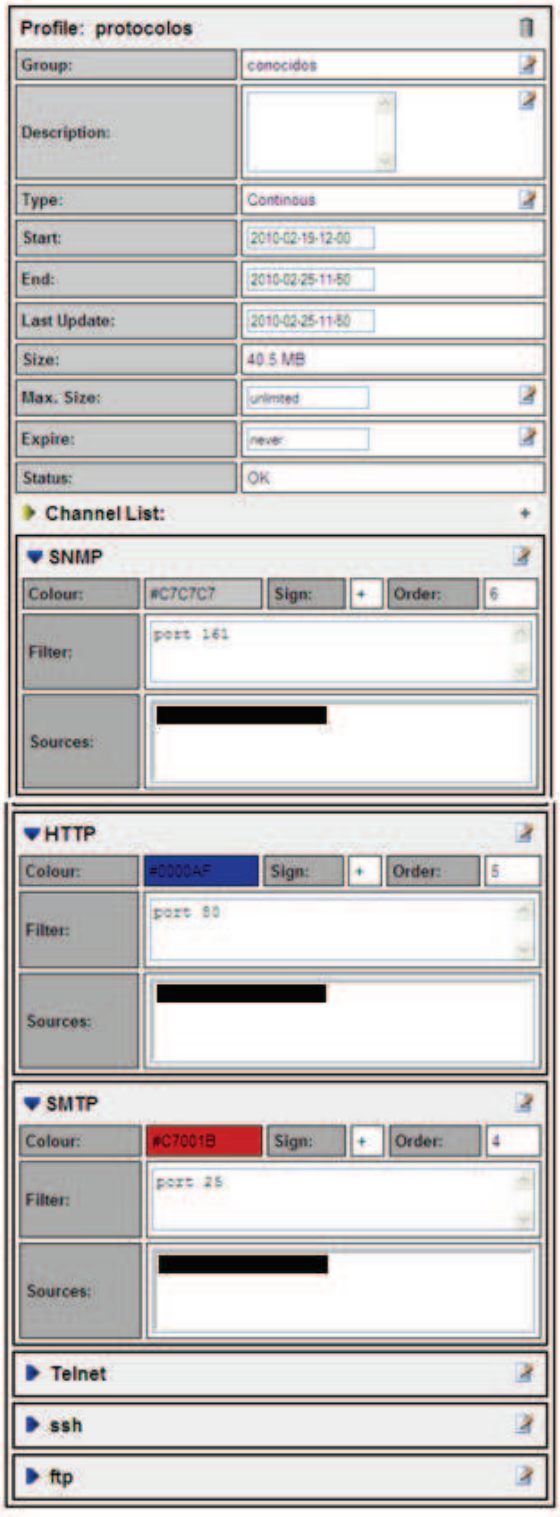


Figura C.25 Ejemplo completo de creación de un profile con sus canales

Es posible observar en la opción del menú home, las gráficas creadas correspondientes a este perfil.

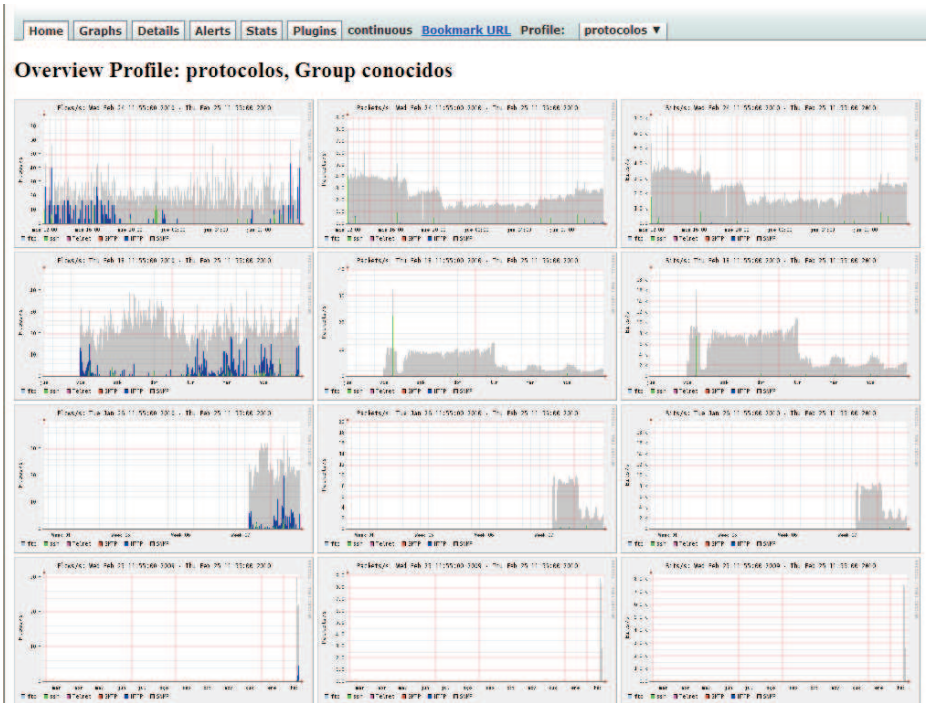


Figura C.26 Gráficas creadas por un profile

Al seleccionar cualquier gráfica al igual que en el profile live, se pueden obtener datos de una manera más detallada.

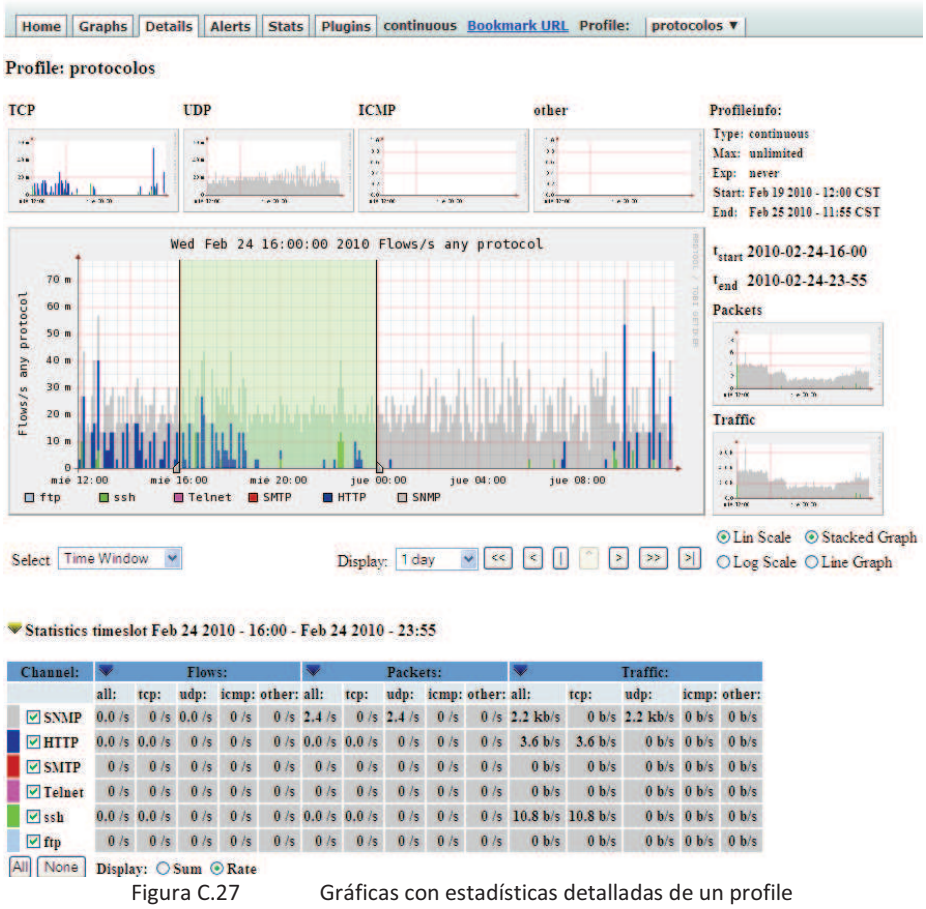


Figura C.27 Gráficas con estadísticas detalladas de un profile

Y aplicar filtros específicos para realizar análisis específicos sobre los profile creados.

Plugin.

Los plugins son una potente herramienta utilizada por el software Nfsen, que permiten añadir programación exterior de acuerdo a las necesidades requeridas. Por medio de plugins creados en Nfsen se logra hacer a este software tan potente como se desee; además de enfocarlo hacia análisis de seguridad, poner precio al tráfico consumido, generar gráficas muy detalladas, entre otras cosas.

En el capítulo 3, sección 3.3.2.4 se describieron los tipos de plugins que soporta el software Nfsen, en esta parte se describirán las subrutinas y configuración necesarias para la creación de un plugin.

Creación de un plugin Backend.

Un plugin backend es escrito como un módulo en perl. Cualquier plugin backend escrito debe de contener el siguiente código:

```
# Name of the plugin
package PluginName;
use strict;

# This string identifies the plugin as a version 1.3.0 plugin.
our $VERSION = 130;

sub Init {
return 1;
}
1;
```

La subrutina 'init' es inicializada cuando el plugin es cargado, con el propósito de dar la posibilidad de que el plugin se ejecute por sí mismo. La función init regresa '1' si se cargó correctamente o '0' se existió algún error.

Dependiendo de la función del plugin, se pueden añadir otras subrutinas, estas subrutinas son:

Cleanup.

Subrutina creada para limpiar el plugin, cuando el software Nfsen termina se ejecución. El propósito de esta subrutina es dar la posibilidad al plugin de terminar por sí mismo, sin necesidad de tener que terminar el proceso forzosamente. El código de esta subrutina es el siguiente:

```
sub Cleanup {
syslog("info", "demoplugin Cleanup");
# not used here
}
```

run

Subrutina necesaria cuando el plugin es recargado en cada actualización del software Nfsen. El código necesario en esta subrutina es el siguiente:

```
sub run {
my $profile = shift;
my $timeslot = shift;
syslog('debug',"Plugin escaneo run: Profile: $profile, Time: $timeslot");
## Añadir codigo aqui
}
```

alert_condition

Subrutina necesaria cuando un plugin es utilizado como un módulo requerido para la ejecución de una alerta, como se observa en la figura.

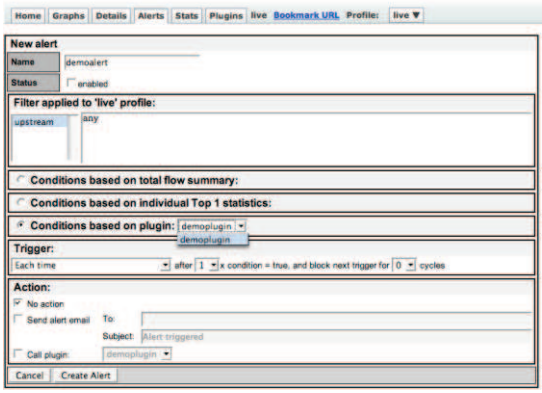


Figura C.28 Ventana de creación de un plugin

La subrutina 'alert_condition' es llamada después de que es aplicado el filtro. El archivo resultado es guardado en el directorio de la alerta. Y este archivo es pasado como parámetro a la subrutina. La subrutina devuelve 1 si las acciones programadas en ellas se cumplen, en caso contrario devuelve 0. El código necesario en esta subrutina es el siguiente:

```
sub alert_condition {
my $argref = shift;
my $alert     = $$argref{'alert'};
my $alertflows = $$argref{'alertfile'};
my $timeslot   = $$argref{'timeslot'};

syslog('info', "Alert condition called: alert: $alert, alertfile: $alertflows, timeslot: $timeslot");
      # Add your code here

return 1;
}
```

alert_action

Esta función es necesaria cuando se crea un plugin con el objetivo de observar por qué se ejecutó la alerta. El código de esta subrutina es el siguiente:

```
sub alert_action {
my $argref = shift;

my $alert    = $$argref{'alert'};
my $timeslot = $$argref{'timeslot'};

syslog('info', "Alert action function called: alert: $alert, timeslot: $timeslot");
      # Add your code here

return 1;
}
```

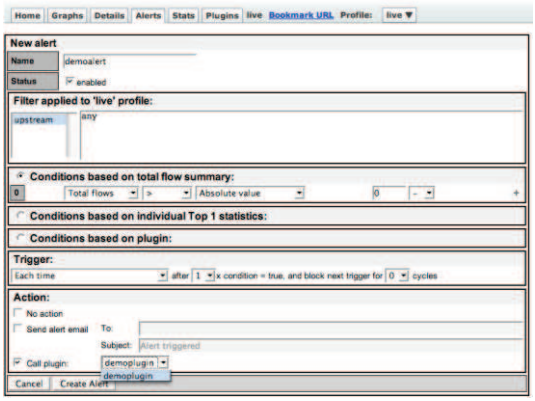


Figura C.29 Ventana de creación de un plugin seleccionando la opción "alert_action"

Creación de plugins Frontend.

Un plugin frontend es escrito como un módulo en php. Permite visualizar, en la interfaz web del software Nfsen, el resultado del módulo backend ejecutado. Cualquier plugin frontend escrito debe de contener el siguiente código:

```
<?php
/*
 * nameplugin_ParseInput is called prior to any output to the web browser and is intended for the plugin to parse possible form data. This function is called only, if this plugin is selected in the plugins tab. If required, this function may set any number of messages as a result of the argument parsing.The return value is ignored.
 */
functionnameplugin_ParseInput( $plugin_id ) {
        //your code here
} // End of nameplugin_ParseInput
/*
 * This function is called after the header and the navigation bar have  been sent to the browser. It's now up to this function what to display.
 * This function is called only, if this plugin is selected in the plugins tabIts return value is ignored.
 */
functionnameplugin_Run( $plugin_id ) {
        // your code here
} // End of nameplugin_Run
?>
```

La función utilizada en la creación de un plugin frontend, es 'nameplugin_Run', esta función recibe como parámetro el ID correspondiente del plugin. Y en base a este parámetro se asocia al plugin backend.

Configuración de plugin en Nfsen.

Para que un plugin pueda ser ejecutado por Nfsen, es necesario añadir al plugin en el archivo "nfsen.conf", buscar la siguiente línea:

```
#Example
@plugins = (
    # profile # module
[ '*', 'demoplugin' ],
);
```

En el arreglo "plugins" se guarda la información de cada plugin añadido al software Nfsen, con el siguiente formato:

- En el primer campo se observa al símbolo '*', este símbolo indica que el plugin es aplicable a cualquier profile y se actualiza periódicamente. Si se desea que el plugin sea ejecutado como una condición de alerta, se añade el símbolo '!'.
- En el segundo campo, se escribe el nombre del plugin a agregar en el Software Nfsen.

En este caso se añadió el plugin escaneo como un módulo que se actualiza periódicamente:

```
@plugins = (
   # profile    # module
[ '*', 'demoplugin' ],
[ '*',    'escaneo'],
);
```

Reiniciar el servicio de nfsen para que surjan efecto los cambios realizados.

```
#          /Listry-AIGC/nfsen-1.3.2/bin/nfsen reload
```

En el archivo "/var/log/messages", se observa que se han cargado los dos plugin exitosamente.

```
# tail -100 /var/log/messages | grep nfsen
Jan  9 23:01:54 localhost nfsen[6660]: Startup. Version: 1.3.2 $Id: nfsend 14 2009-06-10 08:07:06Z haag $
Jan  9 23:01:54 localhost nfsen[6662]: Comm server started: [6662]
Jan  9 23:01:54 localhost nfsen[6661]: nfsend: [6661]
Jan  9 23:01:54 localhost nfsen[6662]: Frontend module 'demoplugin.php' found
Jan  9 23:01:54 localhost nfsen[6662]: Loading plugin 'demoplugin': Success
Jan  9 23:01:54 localhost nfsen[6662]: demoplugin: Init
Jan  9 23:01:54 localhost nfsen[6662]: Initializing plugin 'demoplugin': Success
```

```
Jan  9 23:01:54 localhost nfsen[6662]: plugin 'demoplugin': Profile plugin: 1, Alert condition plugin: 1, Alert action plugin: 1
Jan  9 23:01:54 localhost nfsen[6662]: Frontend module 'escaneo.php' found
Jan  9 23:01:55 localhost nfsen[6662]: Loading plugin 'escaneo': Success
Jan  9 23:01:55 localhost nfsen[6662]: ** Important **: Plugin 'escaneo' is a legacy plugin.
Jan  9 23:01:55 localhost nfsen[6662]: Escaneo: Init
Jan  9 23:01:55 localhost nfsen[6662]: Initializing plugin 'escaneo': Success
```

Visualizar en la Ventana Plugin que se han cargado exitosamente ambos plugin.

Figura C.30 Ejemplo de ejecución de los dos plugins creados

OpenWebmail

El software OpenWebmail es una potente herramienta que permite la visualización y administración de correos electronicos. En el anexo B se describió como acceder a este software. En esta sección se describiran aspectos escenciales en el uso de este software.

Enviar correos con OpenWebmail.

- En el menu principal del software OpenWebmail, dar click en new.

Figura C.31 Menu del software OpenWebmail

- Seleccionar el destinatario y archivo a adjuntar, en caso de requerirse
- Al terminar de escribir el correo, dar click en enviar.

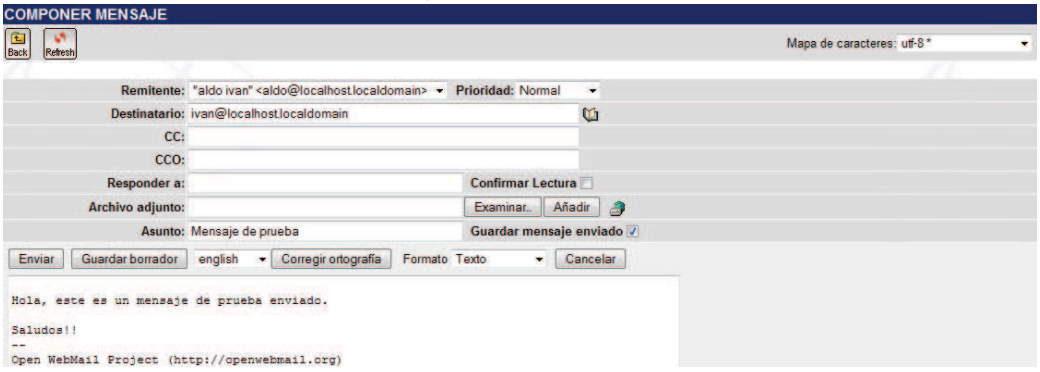


Figura C.32 Ventana de envió de un nuevo correo en OpenWebmail

Acceder con la cuenta “ivan” y visualizar el correo.

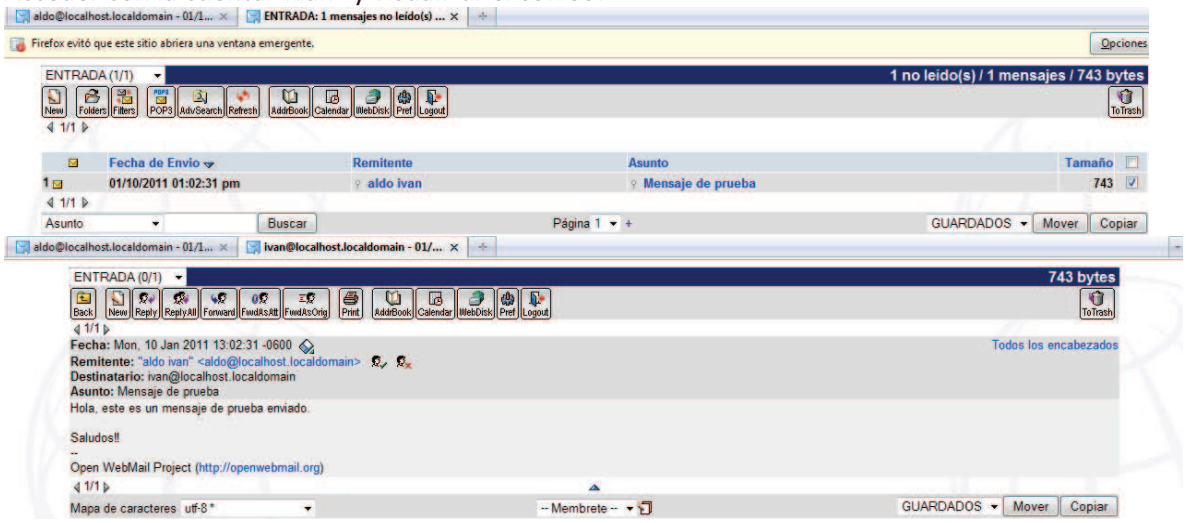


Figura C.33 Visualización de correos recibidos en OpenWebmail

El correo recibido puede ser agrupado en las carpetas: entrada, guardado, enviados, borrador, reenviar, papelera, spam, virus.

Búsqueda de correos

Otra característica importante del software OpenWebmail, es que permite realizar búsquedas muy detalladas de correos.

Al dar clic en “AdvSearch”, aparece una ventana que permite:

- Realizar búsquedas en correos enviados, recibidos, guardados, borradores o correos de detectados como SPAM o virus.
- Escribir el rango de fechas en la búsqueda.
- Seleccionar alguna parte específica del correo a realizar la búsqueda (remitente, destinatario, fecha, asunto, archivos adjuntados, contenido o todo).
- Acepta expresiones regulares para realizar búsqueda.

En este caso se creó un sencillo ejemplo que busca en los correos enviados por aldo@localhost.localdomain, todo correo que contenga la palabra "hola". El resultado se muestra en la figura.

Figura C.34 Busqueda avanzada de correos en OpenWebmail

Como resultado de la búsqueda se observa el correo enviado a ivan@localhost.localdomain. Este correo contiene la palabra "hola" en el cuerpo del mensaje.

Navicat.

Navicat es un software que permite la administración de bases de datos de forma muy simple. En el anexo B se explicó cómo acceder a este software y, crear bases de datos y tablas.

Acceder al contenido de tablas.

- Seleccionar la base de datos a usar y dar doble clic en cualquier tabla creada. En este caso se accedió al contenido de la tabla 'epuertos'

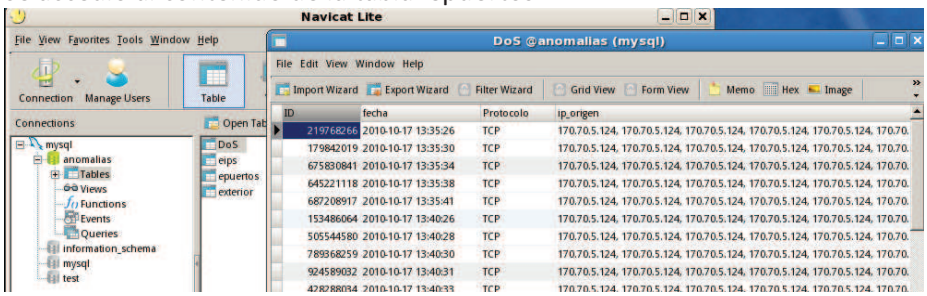


Figura C.35 Ejemplo de datos contenidos en la tabla "DoS"

- Es posible modificar los registros de la tabla al dar click en el registro y escribir el nuevo valor.

Realizar consultas a tablas.

Navicat permite de manera muy eficaz el realizar consultas en tablas de la siguiente forma:

- En el menú principal del software, seleccionar la opción query->new query
- Aparece una ventana en donde se realizan consultas con la sintaxis utilizada en MySQL.
- En este caso se realizó un query que muestre los registros en la tabla 'DoS' con fecha 2010-10-17 entre las 13:35 y 13:40.
- El resultado de este query se muestra en la figura.

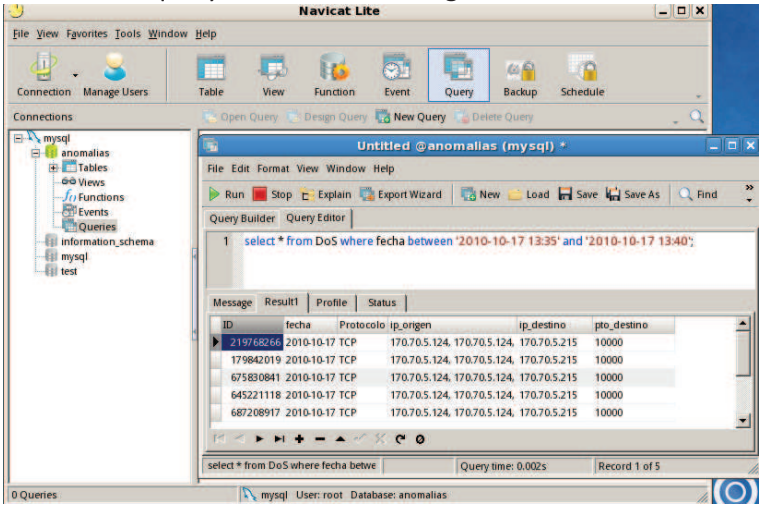


Figura C.36 Consulta realizada a la tabla "DoS" en Navicat

Además, el software Navicat permite:

- Crear funciones
- Realizar vistas en tablas
- Crear respaldos de bases de datos
- Programar tareas que se ejecutan a cierta hora
- Entre otras acciones.

D

Código del plugin "escaneo"

Módulo “escaneo.pm”

```
#/usr/bin/perl
#######          ****************************************************************
#######          ****************************************************************
#######          **********                Plugin escaneo.pm           *********
#######          **********  Creado por: Aldo Iván Girón Capistrán  *********
#######          **********  Última fecha de modificación: 15/10/2010 *********
#######          ****************************************************************
#######          ****************************************************************

#######                    Objetivo del plugin
####### Verificar si se encuentran anomalías en la red que presenten el comportamiento de una
computadora o computadoras infectadas por algún malware.

###### El plugin verifica comportamientos anómalos verificando si se ha realizado en la red:
######           * Escaneo de puertos
######           * Escaneo de Ip'S
######           * Envió de información hacia el exterior y que presente un comportamiento anormal
######           * Ataque de negación de servicios (DoS)

######  El plugin escaneo.pm consta de las siguientes subrutinas.
######      * run:            Encargada de ejecutar el plugin cada 5 minutos.
######      * init:           Encargada de cargar el plugin al momento de iniciar el software nfsen.
######      * epuertos:       Encargada de verificar si se realizó un escaneo de puertos
######      * eips:           Encargada de verificar si se realizó un escaneo de ip's
######      * exterior:       Encargado de verificar si hubo fuga información hacia el exterior dela
red con un comportamiento anormal
#####       * DoS:            Encargado de verificar si se ha efectuado un ataque denegación de
servicios (DoS) o denegación de servicios distribuido (DDoS).
######      * guarda:         En caso de encontrar un comportamiento anormal, esta subrutina se
encargará de guardar los flujos anormales detectados en el archivo “anomalías.txt”
######      * envia_correo:  Esta subrutina verificará si existe el archivo “anomalías.txt”. Si
existe este archivo la subrutina se encargará de notificar mediante él envió de un correo
electrónicomostrando el contenido de este archivo
######      * compara_ip:    Subrutina ocupada por “eips”. Esta subrutina se encarga de verificar si
se presenta un escaneo de IP’S en los registros analizados.
######      * compara:    Subrutina ocupada por “epuertos”Esta subrutina se encarga de verificar si
se presenta un escaneo de puertos en los registros analizados.
######      * compara_exterior: Subrutina ocupada por “exterior”Esta subrutina se encarga de
verificar si se presenta una fuga de información en los registros analizados.
######      * compara_dos:   Subrutina ocupada por “DoS”Esta subrutina se encarga de verificar si se
presenta un ataque de negación de servicios en los registros analizados.
######      * separa:     Subrutina encargada de separar la información recibida por el colector
nfdump de la siguiente forma:     (protocolo) , (ip ori : pto ori) , (ip dst : pto dst)
######      * agrupa:     Subrutina ocupada por “epuertos” y “DoS”. Esta subrutina se encarga de
separar  la información en un formato que permita la detección de un escaneo de puertos o ataque de
negación de  servicios. La subrutina “agrupa” separa la información de la siguiente forma:
######                    xxx.xxx.xxx.xxx :  xxxxx    ->  xxx.xxx.xxx.xxx :  xxxxx        xx
######                             (ip ori)    , (pto ori)  ,     (ip dst)     , (pto dst) , (trafico)
######      * agrupa_ip:  Subrutina ocupada por “eips” y “exterior”. Esta subrutina se encarga de
separar  la información en un formato que permita la detección de un escaneo de IP’S o detectar fuga
de información. La subrutina “agrupa_ip” separa la información de la siguiente forma:
######   xxx.xxx.xxx.xxx :  xxxxx     ->   xxx.xxx  .    xxx   .   xxx    :  xxxxx        xx
######    (ip ori)        , (pto ori)  ,   (ip dst1) , (ip dst2), (ip dst3), (pto dst)  (trafico)
######      * Cleanup:    Subrutina especial, utilizada por el software nfsen para permitir al plugin
limpiar datos cuando se finaliza la ejecución del software Nfsen

package escaneo;
### Declaración de modulos a utilizar.
use strict;
use Switch;
use NfSen;
use NfConf;
use Sys::Syslog;

Sys::Syslog::setlogsock('unix');
use Mysql;
use Notification;
### Declaración de variables locales dentro del plugin, que actuaran como variables globales para el
uso de las demás subrutinas
my ( $nfdump, $PROFILEDATADIR, $LOGFILE, $NOTIFY );
my (@registros,@guarde,@anomalias,@encontre);
my @registros =();

#############################################################################################
######      Subrrutina run:           Encargada de ejecutar el plugin cada 5 minutos.

sub run {
### Guardamos información recibida por subrutina init y notifícanos en syslog el que seestá
ejecutando el plugin.
my $profile = shift;
my $timeslot = shift;
syslog('debug',"Plugin escaneo run: Profile: $profile, Time: $timeslot");

### Inicializamos el arreglo encontre con -1: En este arreglo se guardará el número de la línea en el
cual se encontró alguna anomalía para evitar revisar información repetida
```

```
        @encontre =(-1);

### Leemos información proporcionada por una módulo de nfsen
        my %profileinfo      = NfProfile::ReadProfile('live');
        #my $netflow_sources ="$PROFILEDATADIR/live/Anexo";

### Accedemos a la ruta donde se instaló nfdump
        my $netflow_sources ="$PROFILEDATADIR/live/$profileinfo{'sourcelist'}";

### El arreglo “registros” contendrá toda la información que ha sido interpretada por el colector
nfdump, y ha sido guardada en un archivo nfcapd, este arreglo se modificará cada 5 minutos,
almacenando en él la nueva información capturada por nfdump y guardada en el archivo nfcapd
@registros = `$nfdump -M $netflow_sources -r nfcapd.$timeslot`;

### Guardamos la información contenida en el arreglo registros en un archivo llamada salida.txt
        if (open (LOG, "> /Listry-AIGC/salida.txt")){ ;
                print LOG "$nfdump -M $netflow_sources -r nfcapd.$timeslot\n\nDatos guardados.\n\n";
        print LOG @registros;
        close LOG ;
        } else {
        syslog('debug', "Escaneo: unable to open $LOGFILE") ;
        }
### Si en ejecuciones posterioresdel plugin se detectaron anomalías, se creará un archivo llamado
“anomalias.txt”. Al realizar un nuevo análisis en caso de tener este archivo se borrará con el
objetivo de no mostrar información repetida
        if (open (VERIFICA,"</Listry-AIGC/anomalias.txt")){
                close VERIFICA;
                system ("rm /Listry-AIGC/anomalias.txt");
        }
        if (open (VERIFICA,"</Listry-AIGC/anomalias_php.txt")){
close VERIFICA;
system ("rm /Listry-AIGC/anomalias_php.txt");
}
### Mandamos llamar a las subrutinas epuertos, eips, exterior y DoS, cada vez que alguna de estas
subrutinas termine volvemos a iniciar el arreglo encontre en -1
        &epuertos(@registros);
        @encontre =(-1);
        &eips(@registros);
        @encontre =(-1);
&exterior(@registros);
        @encontre =(-1);
&DoS(@registros);

### Al haber terminado todas las subrutinas su función especifica verificamos si existen anomalías
que se han guardado en el archivo anomalias.txt, en caso de encontrar este archivo llamamos a la
subrutina envia_correo
        if (open (VERIFICA,"</Listry-AIGC/anomalias.txt")){
                close VERIFICA;
                &envia_correo();
        }

} # Fin de la subrutina run

#############################################################################################
###  Subrutina init:         Encargada de cargar el plugin al momento de iniciar el software nfsen.

sub Init {
syslog("info", "Escaneo: Init");
# Inicializamos variables que contendrán información necesaria parala ejecución de la subrutina run
$nfdump          = "$NfConf::PREFIX/nfdump";
    $PROFILEDATADIR =  "$NfConf::PROFILEDATADIR";
    $LOGFILE         = "$NfConf::VARDIR/tmp/escaneo.log" ;
    $NOTIFY          = 1 ;
return 1;
} # Fin subrrutina Init

#############################################################################################
###  Subrutina epuertos:     Encargada de verificar si se realizó un escaneo de puertos

sub epuertos{
### Declaración variables locales ocupadas en esta subrutina.
        my ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico);
        my ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;

### Obtenemos la información enviada por la subrutina run
        @registros=@_;

### Inicializamos arreglos vacíos, el arreglo “anomalias” contendrá la información encontrada del
análisis y que se ha considerado como anormal.El arreglo “guarde” contiene elementos separados por la
subrutina separa.
        @guarde=();
        @anomalias=();
        $linea=0;
```

```
## Obtenemos el número total de elementos almacenados en el arreglo “registros” y se guarda esta
referencia en la variable “valor”
        $valor=$#registros;
        #print "$#registros";

### Recorremos todos los datos contenidos en el arreglo “registros”, haciendo referencia a cada uno
de ellos en la variable “linea”
foreach $linea (@registros){
                #print "$linea\t $contador\n";

## Llamamos a la subrutina “separa”,  pasando como argumento la variable “linea”, además el arreglo
“guarde”. Esta subrutinadevuelveel arreglo “guarde” que contiene la información separada
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }   ###Fin ciclo foreach

### Inicializamos contadores requeridos
        $contador =0;
        $contador_anom=0;
        $valor=$#guarde;
        my ($cont_encontre, $cont_linea ,$banderin);

### Este ciclo while se encargara de verificar si existe algún número guardado en arreglo
“encontre”que corresponda con alguna línea del arreglo “registros”, en caso de encontrar algún número
igual no permitirá que se realice el análisis, debido a que esa línea ya ha sido revisada y se
encontró con una anomalía
### Este ciclo while recorrerá todos los elementos que se tengan en nuestro arreglo registros
        ## Inicio 1er while
        while($contador<=$valor){
                $cont_encontre=$#encontre;
                $cont_linea=0;
                $banderin=1;
                while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                $anomalo=$encontre[$cont_linea];
                #print"$cont_tmp == $anomalo \t";
                        if ($cont_tmp == $anomalo){
                                #print "\n No entre\n";
                                $banderin=0;
                        }
                $cont_linea ++;
                }   #### Fin ciclo lineas encontradas

### En caso de que el ciclo anterior haya encontrado alguna línea ya revisada y con comportamiento
anómalo no se entra al if, Si el registro no ha sido revisado se procederá a revisar esta línea con
las demáselementosdel arreglo.
                if ($banderin==1){

## Guardamos el valor temporal del arreglo “guarde” en la variable “linea”
                $linea=$guarde[$contador];
                #print "\n\n$contador\t$linea  \n";

## Mandamos llamar a la subrutina “agrupa”, que encargará de separar los elementos de la forma
indicada en el inicio del plugin
                ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico)=&agrupa($linea);
                #print "\n\n$contador\t$ip_ori:$port_ori --> $ip_dest:$port_dest\n";

## Asignamos el varal de la variable “contador” a la variable “cont_tmp”, este contador temporal se
encargara de buscar alguna anomalía desde este elemento en adelante
                $cont_tmp=$contador;

### En este arreglo se almacenaran todos los elementos temporales que hayan sido encontrados como
anómalos
                @tmp=();

### Cada que se encuentre un elemento anormal en este arreglo se incrementará la variable
“contador_anom”
                $contador_anom=0;

### En este ciclo iremos comparando los elementos que se encuentren por debajo del contador
“cont_tmp”
                ### Inicio 2 ciclowhile
                while($cont_tmp<=$valor){
                        $cont_tmp ++;
                        $cont_encontre=$#encontre;
                        $cont_linea=0;
                        $banderin=1;

#Esta parte del codigo ya se ha explicado.
                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";
                                if ($cont_tmp == $anomalo){
                                        #print "\n No entre\n";
                                        $banderin=0;
                                }
                        $cont_linea ++;
                        }   #### Fin ciclo lineas encontradas
```

```
                        if ($banderin==1){
                        #print "$contador = $cont_tmp\t";

## Guardamos el elemento temporal a comparar
$linea=$guarde[$cont_tmp];
                        #print "\t$cont_tmp\t$linea";

## Llamamos a la subrutina agrupa. Esta subrutina se encargará de agrupar la línea que contiene el
elemento temporal. Esta subrutina nos devuelve variables que contiene la información
agrupada($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp)=&agrupa($linea);
                        #print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp -->
                        $ip_dest_tmp:$port_dest_tmp";

## Mandamos llamar a la subrutina “compara” que se encargará de verificar si hay un incremento en el
puerto destino del elemento actual con el elemento temporal comparado, en caso de encontrar esta
anomalía, se guardara esta línea en el arreglo “tmp”, además de incrementar el contador_anom.Esta
subrutina devuelve la variable “port_dst” para no perder referencia de este elemento.
                        ($port_dest,$contador_anom,@tmp)=&compara($ip_ori, $ip_dest, $port_dest,
$ip_ori_tmp, $ip_dest_tmp, $port_dest_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
                #$cont_tmp ++;
                        }## Fin if coincide
                }   #### Fin 2 ciclo while iteración con valores temporales
                #print "\n\n";

### En caso de que la variable“contador_anom” sea mayor o igual que diez, nos indica que se ha
detectado un escaneo de puertos.Al detectarse esta anomalía se guardara el contenido del arreglo“tmp”
en el arreglo “anomalias”
                if ($contador_anom >= 10){
                        push (@anomalias,@tmp);
                }
                }   #Fin if si coincide línea con alguna anomalía
        $contador ++;
        }   ###Fin ciclo 1er while recorre todos los datos encontrados

### Si existe el arreglo “anomalias” se ejecutará la subrutina guarda,pasando como argumento a esta
subrutina la bandera “impirmo”, esta bandera indicaráque anomalía se ha detectado. Además se pasa
como argumento el arreglo “anomalias”.
        if (@anomalias){
                my $imprimo=0;
                &guarda($imprimo,@anomalias);
        }
}   ### Fin subrutina e_puertos

#############################################################################################################
###     Subrutina eips:      Encargada de verificar si se realizó un escaneo de IP'S
sub eips{
### Declaración variables locales ocupadas en esta subrutina.
        my ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico);
my ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;

### Obtenemos la información enviada por la subrutina run
        @registros=@_;

### Inicializamos arreglos vacíos, el arreglo “anomalias” contendrá la información que se ha
detectado anormal del análisis realizado, el arreglo guarde contiene elementos separados por la
subrutina separa.
        @guarde=();
        #@anomalias=();
        $linea=0;

## Obtenemos el número total de elementos almacenados en la arreglo “registros”
        $valor=$#registros;
        #print "$#registros";

### Recorremos todos los datos contenidos en el arreglo “registros”.haciendo referencia a cada uno de
ellos en la variable “linea”
foreach $linea (@registros){
                #$contador ++;

## Llamamos a la subrutina “separa”,  pasando como argumento la variable “linea”, además el arreglo
“guarde”. Esta subrutinadevuelveel arreglo “guarde” que contiene la información separada
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }   ###Fin ciclo foreach

### Inicializamos contadores requeridos
        my $i;
        my ($cont_encontre, $cont_linea ,$banderin);

## Este ciclo hará que todo el procedimiento se repita dos veces por este motivo:
##      * En la 1er iteración buscara escaneo de IP's con modificación en el 4 octeto
##      * En la 2 iteración buscara escaneo de IP's con modificación en el 3er octeto
        ## Inicio ciclo for recorre 2 veces todo el procedimiento
```

```
        for ($i=1; $i<=2; $i++){
                @anomalias=();

### Inicializamos contadores requeridos
                $contador =0;
                $contador_anom=0;
                $valor=$#guarde;
                #print "$i\n";
### Este ciclo while recorrerá todos los elementos que se tengan en nuestro arreglo registros
                ### Incio primer ciclo while
while($contador<=$valor){

                        $cont_encontre=$#encontre;
                        $cont_linea=0;
                        $banderin=1;
### Este ciclo while se encargara de verificar si existe algún número guardado en arreglo
“encontre”que corresponda con alguna línea del arreglo “registros”, en caso de encontrar algún número
igual no permitirá que se realice el análisis, debido a que esa línea ya ha sido revisada y se
encontró con una anomalía
### Este ciclo while recorrerá todos los elementos que se tengan en nuestro arreglo registros
                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                                $anomalo=$encontre[$cont_linea];
                                #print"$cont_tmp == $anomalo \t";
                                if ($cont_tmp == $anomalo){
                                        #print "\n No entre\n";
                                        $banderin=0;
                                }
                        $cont_linea ++;
                        }   #### fin ciclo lineas encontradas
### En caso de que el ciclo anterior haya encontrado alguna línea ya revisada y con comportamiento
anómalo no se entra al if, Si el registro no ha sido revisado se procederá a revisar esta línea con
las demáselementosdel arreglo.
                        if ($banderin==1){

## Guardamos el valor temporal del arreglo “guarde” en la variable “linea”
                                $linea=$guarde[$contador];

## Mandamos llamar a la subrutina “agrupa_ip”, que encargará de separar los elementos de la forma
indicada en el inicio del plugin
        ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico)=&agrupa_ip($linea);
                                #print "\n$contador\t$ip_ori:$port_ori -->
                                $ip_dest1.$ip_dest2.$ip_dest3:$port_dest\n";

## Asignamos el varal de la variable “contador” a la variable “cont_tmp”, este contador temporal se
encargara de buscar alguna anomalía desde este elemento en adelante
                                $cont_tmp=$contador;

### En este arreglo se almacenaran todos los elementos temporales que hayan sido encontrados como
anómalos
                                @tmp=();

### Cada que se encuentre un elemento anormal en este arreglo se incrementará la variable
“contador_anom”
                                $contador_anom=0;

### En este ciclo iremos comparando los elementos que se encuentren por debajo del contador
“cont_tmp”
                        ### Inicio 2 while
                                while($cont_tmp<=$valor){
                                        $cont_tmp ++;
                                        $cont_encontre=$#encontre;
                                        $cont_linea=0;
                                        $banderin=1;

#Esta parte del codigo ya se ha explicado.
                                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                                                $anomalo=$encontre[$cont_linea];
                                                #print"$cont_tmp == $anomalo \t";

                                                if ($cont_tmp == $anomalo){
                                                        #print "\n No entre\n";
                                                        $banderin=0;
                                                }
                                        $cont_linea ++;
                                        }   #### fin ciclo lineas encontradas

                                        if ($banderin==1){

## Guardamos el elemento temporal a comparar
                $linea=$guarde[$cont_tmp];

## Llamamos a la subrutina agrupa_ip. Esta subrutina se encargará de agrupar la línea que contiene el
elemento temporal. Esta subrutina nos devuelve variables que contiene la información agrupada
($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico_tmp)=&ag
rupa_ip($linea);

#print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp --> $ip_dest_tmp1 a.$ip_dest_tmp2 b.$ip_dest_tmp3
c:$port_dest_tmp";
```

```
## Mandamos llamar a la subrutina “compara_ip” que se encargará de verificar si hay un incremento en
el tercer o cuarto octeto de la IP destino (dependiendo de la iteración del ciclo for) del elemento
actual con el elemento temporal comparado, en caso de encontrar esta anomalía, se guardara esta línea
en el arreglo “tmp”, además de incrementar el contador_anom.Esta subrutina devuelve la variable
“ip_dest2” e “ip_dest3” para no perder referencia de estos elementos.

($ip_dest2,$ip_dest3,$contador_anom,@tmp)=&compara_ip($i,$ip_ori, $ip_dest1,$ip_dest2, $ip_dest3,
$port_dest, $ip_ori_tmp, $ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,
$port_dest_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
                                } ## Fin if
                }   #### Fin 2 ciclo while iteración con valores temporales

### En caso de que la variable“contador_anom” sea mayor o igual que diez, nos indica que se ha
detectado un escaneo de IP’S.Al detectarse esta anomalía se guardara el contenido del arreglo“tmp” en
el arreglo “anomalias”
                                if ($contador_anom >= 10){
                                        push (@anomalias,@tmp);
                                }
                        }   #fin if si conincide con linea con alguna anomalia
                        $contador ++;

                }   ###Fin 1er ciclo while recorre todos losdatos encontrados

### Si existe el arreglo “anomalias” se ejecutará la subrutina guarda,pasando como argumento a esta
subrutina la bandera “impirmo”, esta bandera indicaráque anomalía se ha detectado. Además se pasa
como argumento el arreglo “anomalias”.
        if (@anomalias){
my $imprimo=1;
&guarda($imprimo,@anomalias);
                }
        }#fin ciclo for  recorre 2 veces
} #######fin sub escaneo ips

#############################################################################################
## Subrutina exterior:       Encargado de verificar si se ha enviado información hacia el exterior
de la red con un comportamiento anormal.
### No se comenta esta subrutina debido a que tiene una estructura similar eips, solamente cambia en
llamar  a la subrutina“compara_exterior”.

sub exterior{
        my ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico);
        my
($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;
        @registros=@_;
        @guarde=();
        @anomalias=();
        $linea=0;
        $valor=$#registros;
        #print "$#registros";

foreach $linea (@registros){
                #print "$linea\t $contador\n";
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }###Fin ciclo for

        $contador =0;
        $contador_anom=0;
        $valor=$#guarde;
        my ($cont_encontre, $cont_linea ,$banderin);

        ### Inicio 1er ciclo while recorre todos los datos encontrados
        while($contador<=$valor){
                $cont_encontre=$#encontre;
                $cont_linea=0;
                $banderin=1;

                while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";

                        if ($cont_tmp == $anomalo){
                                #print "\n No entre\n";
                                $banderin=0;
                        }
                        $cont_linea ++;
                }   #### Fin ciclo lineas encontradas

                ### Inicio if en caso de que el elemento actual no haya sido detectado como anomalo
                if ($banderin==1){
                                $linea=$guarde[$contador];
        ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico)=&agrupa_ip($linea);
```

```
                                #print "\n$contador\t$ip_ori:$port_ori -->
$ip_dest1.$ip_dest2.$ip_dest3:$port_dest\n";
                                $cont_tmp=$contador;
                                @tmp=();
                                $contador_anom=0;

                                ## Inicio 2 ciclo while iteraccion con valores temporales
                                while($cont_tmp<=$valor){
                                        $cont_tmp ++;
                                        $cont_encontre=$#encontre;
                                        $cont_linea=0;
                                        $banderin=1;

                                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                                          $anomalo=$encontre[$cont_linea];
                                          #print"$cont_tmp == $anomalo \t";

                                          if ($cont_tmp == $anomalo){
                                                  #print "\n No entre\n";
                                                  $banderin=0;
                                          }
                                          $cont_linea ++;
                                        }   #### Fin ciclo lineas encontradas

                                if ($banderin==1){
                $linea=$guarde[$cont_tmp];

                                        #print "\t$cont_tmp\t$linea";

        ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,$port_dest_tmp,$trafico
_tmp)=&agrupa_ip($linea);

                                        #print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp -->
$ip_dest_tmp1 a.$ip_dest_tmp2 b.$ip_dest_tmp3 c:$port_dest_tmp";

                                        ($port_ori,$contador_anom,@tmp)=&compara_exterior($ip_ori,
$port_ori, $ip_dest1,$ip_dest2, $ip_dest3, $ip_ori_tmp, $port_ori_tmp,
$ip_dest_tmp1,$linea,$cont_tmp,$contador_anom,$contador,$trafico,$trafico_tmp,@tmp);
                                } ## Fin if
                }#### fin 2 ciclo while iteraccion con valores temporales

                                if ($contador_anom >= 10){
                                        push (@anomalias,@tmp);
                                }
                        }    #Fin if si conincide con linaco con alguna anomalia
                        $contador ++;
                        }     ###Fin 1er ciclo while recorre todos losdatos encontrados

        if (@anomalias){
my $imprimo=2;
&guarda($imprimo,@anomalias);
}
}   ##### Fin subrrutina exterior.

#############################################################################################################
## Subrrutina DoS:            Encargado de verificar si se ha efectuado un ataque denegación de
servicios (DoS) o denegación de servicios distribuido (DDoS).
### No se comenta esta subrutina debido a que tiene una estructura similar a la subrutina “epuertos”,
solamente cambia en llamar a la función “compara_dos”.

sub DoS{
        my ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico);
        my ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp);
        my ($var,$valor, $linea, $contador, $contador_anom,$cont_tmp,$anomalo);
        my @tmp;
        $contador=0;
        @registros=@_;
        @guarde=();
        @anomalias=();
        $linea=0;
        $valor=$#registros;
        #print "$#registros";
foreach $linea (@registros){
                #print "$linea\t $contador\n";
                @guarde=&separa($linea,@guarde);
                #print "\nHola imprimo \t@guarde\n";
                #print "$linea";
        }   ###Fin ciclo foreach
        $contador =0;
        $contador_anom=0;
        $valor=$#guarde;
        my ($cont_encontre, $cont_linea ,$banderin);

        ## Inicio 1er while
        while($contador<=$valor){
                $cont_encontre=$#encontre;
                $cont_linea=0;
                $banderin=1;
```

```
                while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";
                        if ($cont_tmp == $anomalo){
                                #print "\n No entre\n";
                                $banderin=0;
                        }
                        $cont_linea ++;
                }   #### Fin ciclo lineas encontradas
                if ($banderin==1){
                $linea=$guarde[$contador];
                #print "\n\n$contador\t$linea  \n";

                ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico)=&agrupa($linea);
                #print "\n\n$contador\t$ip_ori:$port_ori --> $ip_dest:$port_dest\n";
                $cont_tmp=$contador;
                @tmp=();
                $contador_anom=0;
                ### Incio 2 while
                while($cont_tmp<=$valor){
                        $cont_tmp ++;
                        $cont_encontre=$#encontre;
                        $cont_linea=0;

                        $banderin=1;
                        while (($cont_linea <=$cont_encontre)&& ($banderin==1)){
                        $anomalo=$encontre[$cont_linea];
                        #print"$cont_tmp == $anomalo \t";
                                if ($cont_tmp == $anomalo){
                                        #print "\n No entre\n";
                                        $banderin=0;
                                }
                        $cont_linea ++;
                        }   #### fin ciclo lineas encontradas
                        if ($banderin==1){
$linea=$guarde[$cont_tmp];
                        #print "\t$cont_tmp\t$linea";
                ($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp)=&agrupa($linea);
                        #print"\t$cont_tmp =-= $ip_ori_tmp:$port_ori_tmp -->
$ip_dest_tmp:$port_dest_tmp";
        ($port_dest,$contador_anom,@tmp)=&compara_dos($port_dest,$ip_dest,$trafico,$port_dest_tmp,$i
p_dest_tmp,$trafico_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
                        }  ## Fin if
                }   #### Fin 2 ciclo while iteraccion con valores temporales
                 if ($contador_anom >= 50){
                 #if ($contador_anom >= 25){
                        push (@anomalias,@tmp);
                 }
                 }   #Fin if si conincide con linaco con alguna anomalia
        $contador ++;
        }   ###Fin ciclo 1er while recorre todos los datos encontrados
        if (@anomalias){
                 my $imprimo=3;
                 &guarda($imprimo,@anomalias);
        }
}       ### Fin subrrutina DoS

#############################################################################################
### Subrutina obten_fecha_id encargada de obtener fecha e ID para insertar en  BD Mysql

sub obten_fecha_id{
        my ($segundos, $minutos, $horas, $dia_mes, $mes, $anyo, $fecha_unix, $fecha);
## Obtenemos fecha y le asignamos el formato correcto
        $fecha_unix = time ();
        ($segundos, $minutos, $horas, $dia_mes, $mes, $anyo, undef, undef, undef) = localtime
($fecha_unix);
        $mes ++;
if (($dia_mes >= 0)&&($dia_mes < 10)){
                $dia_mes ='0'.$dia_mes;
        }
if (($mes >= 0)&&($mes < 10)){
                $mes ='0'.$mes;
        }
if (($segundos >= 0)&&($segundos < 10)){
                $segundos ='0'.$segundos;
        }
if (($minutos >= 0)&&($minutos < 10)){
                $minutos ='0'.$minutos;
        }
if (($horas >= 0)&&($horas < 10)){
                $horas ='0'.$horas;
        }
if ($anyo < 1900){
                $anyo += 1900
        }
## Guardamos en variable “fecha”: año, mes, día, hora y minutos
        $fecha="$anyo-$mes-$dia_mes $horas:$minutos:$segundos";
        #print "$fecha\n";
```

```
## Generamos un número aleatorio que servirá como referencia del registro en las tablas de MySQL,
este número lo asignamos a la variable "ID"
        my $rango=999999999;
        my $ID=int(rand($rango));
        #print "$ID\n";
return ($ID, $fecha)
}

#############################################################################################
### Subrutina guarda:        En caso de encontrar un comportamiento anormal, esta rutina se encarga
de guardar los datos encontrados en el archivo "anomalias.txt" y generar el archivo
"anomalias_php.txt". Este archivo genera el formato observado en la interfaz web del módulo
escaneo.php

sub guarda{

        my($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes);
        my($ip_ori_tmp,$port_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$trafico_tmp);
        my($linea,$imprimo, $fecha,$contador,$var,$espe,$red_serv);
## Se almacenan en variables datos proporcionados por la subrutina que ha mandado a llamar a esta
subrutina
        $imprimo=shift;
        $contador =0;
        my @tmp=@_;
        my $var_tmp="";
        my $query="";
        my($ID,$fecha);

# MYSQL CONFIG VARIABLES
my ($host,$database,$user,$pw,$connect,$execute);
 $host = "localhost";
 $database = "anomalias";
 $user = "root";
 $pw = "Aldeano";
# PERL MYSQL CONNECT()
$connect = Mysql->connect($host, $database, $user, $pw);
# SELECT DB
 $connect->selectdb($database);

## Abrimos archivos
open (ESCRIBE,">>/Listry-AIGC/anomalias.txt");
open (PHP,">>/Listry-AIGC/anomalias_php.txt");
### Mediante switch verificaremos los contenidos de la variable "imprimo", dependiendo del valor
contenido en esta variable, seejecutara el case que le corresponda y guardará datos del arreglo "tmp"
en los archivos, además de indicar que anomalía se ha detectado.
        switch ($imprimo){
                ## Se encontró un escaneo de puertos
                case 0{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\t\tEscaneo de puertos encontrado:\n\t\t\t\t$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\tTrafico\n";
                        print PHP"<h3><font color='red'>\t\t\tEscaneo de puertos
encontrado:\n\t\t\t\t$fecha</font></h3>";
                        print PHP"<table with border=4, width='90%'>";
                        print PHP"<tr aling='center' background BGCOLOR='#F54C58'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                        $espe=0;
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_ori;
                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                #$port_tmp="$port_tmp, $port_dest"
                                if ($var eq $ip_ori){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $port_dest";
                                        }
                                        else {
                                                $var_tmp=$port_dest;
                                        }
                                }# fin if
                                else {
```

```
                                        $var=$ip_ori;
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                        ($ID,$fecha)=obten_fecha_id();
                                        $query="insert into epuertos values
('$ID','$fecha','$protocolo', '$ip_ori', '$ip_dest', '$var_tmp')";
                                        $execute = $connect->query($query);
                                        #print "\n$query\n";
                                        print ESCRIBE "\nSe observa que la ip origen: $ip_ori esta
buscando algun puerto disponible dentro de la ip destino: $ip_dest que pueda infectar. \nEl escaneo
de puertos se esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos'
con ID= $ID\n\n";
                                        print PHP"</table>";
                                        print PHP "<h4>\nSe observa que la ip origen: $ip_ori esta
buscando algun puerto disponible dentro de la ip destino: $ip_dest \nque pueda infectar. El escaneo
de puertos se esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos'
con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                                        $espe=$contador;
                                        print ESCRIBE "\n\t\t\t\tEscaneo de puertos
encontrado:\n\t\t\t\t\t$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\tTrafico\n";
                                        print PHP"<h3><font color='red'>\t\t\tEscaneo de puertos
encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#F54C58'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        $var_tmp=$port_dest;
                                }  ## Fin else
                $contador++;
                }       ## Fin For
                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                         ($ID,$fecha)=obten_fecha_id();
                          $query="insert into epuertos values ('$ID','$fecha','$protocolo',
'$ip_ori', '$ip_dest', '$var_tmp')";
                         $execute = $connect->query($query);
                         #print "\n$query\n";
                         print ESCRIBE "\nSe observa que la ip origen: $ip_ori esta buscando algun
puerto disponible dentro de la ip destino: $ip_dest que pueda infectar. \nEl escaneo de puertos se
esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos' con ID=
$ID\n\n";
                         print PHP"</table>";
                         print PHP "<h4>\nSe observa que la ip origen: $ip_ori esta buscando algun
puerto disponible dentro de la ip destino: $ip_dest \nque pueda infectar. El escaneo de puertos se
esta realizando de forma secuencial\nSe ha insertado la anomalia en tabla 'epuertos' con ID= $ID\nY
se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                }       ### Fin case 0

                ## Se encontró un escaneo de Ip's
                case 1{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\t\tEscaneo de Ip'S encontrado:\n\t\t\t\t\t$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\t Trafico\n";
                         print PHP"<h3><font color='purple'>\t\t\tEscaneo IP'S
encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                         print PHP"<table with border=4, width='90%'>";
                         print PHP"<tr aling='center' background BGCOLOR='#501287'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                         $espe=0;
```

```
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_ori;
                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                if ($var eq $ip_ori){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5></td><td><h5 aling='center'>$trafico</h5></td></tr>";

                                        #$var_tmp="$var_tmp, $ip_dest";
                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $ip_dest";
                                        }
                                        else {
                                                $var_tmp=$ip_dest;
                                        }
                                }# fin if

                                else {
                                        $var=$ip_ori;
                                        $linea=$tmp[$espe];

        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        ($ID,$fecha)=obten_fecha_id();
                                        $query="insert into eips values
('$ID','$fecha','$protocolo', '$ip_ori', '$var_tmp', '$port_dest')";
                                        $execute = $connect->query($query);
                                        #print "\n$query\n";
                                        print ESCRIBE "\nSe observa que la ip $ip_ori esta buscando
alguna otra ip dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma
secuencial con una variacion en el tercer o cuarto octeto dependiendo \ndel caso encontrado\n Se ha
insertado la anomalia en tabla 'eips' con ID= $ID.\n\n";
                                        print PHP "</table>";
                                        print PHP "<h4>\nSe observa que la ip $ip_ori esta buscando
alguna otra ip dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma
secuencial con una variacion en el tercer o cuarto octeto dependiendo\ndel caso encontrado\n Se ha
insertado la anomalia en tabla 'eips' con ID= $ID\nY se ha notificado via email a:
'aldo\@localhost.localdomain' sobre la anomalia de dectectada.\n\n<\h4>";

                                        $espe=$contador;
                                        print ESCRIBE "\n\t\t\t\tEscaneo de Ip'S
encontrado:\n\t\t\t\t\t$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\t Trafico\n";
                                        print PHP"<h3><font color='purple'>\t\t\tEscaneo IP'S
encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#501287'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5></td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        $var_tmp=$ip_dest;
                                }  ## Fin else
                $contador++;
                }       ## Fin For

                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        ($ID,$fecha)=obten_fecha_id();
                        $query="insert into eips values ('$ID','$fecha','$protocolo', '$ip_ori',
'$var_tmp', '$port_dest')";
                        $execute = $connect->query($query);
                        #print"\n$query\n";
                        print ESCRIBE "\nSe observa que la ip $ip_ori esta buscando alguna otra ip
dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma secuencial con una
variacion en el tercer o cuarto octeto dependiendo \ndel caso encontrado\nSe ha insertado la anomalia
en tabla 'eips' con ID= $ID.\n\n";
```

```
                        print PHP"</table>";
                        print PHP "<h4>\nSe observa que la ip $ip_ori esta buscando alguna otra ip
dentro de la red que pueda infectar!\nEl escaneo de Ip's realizado es de forma secuencial con una
variacion en el tercer o cuarto octeto dependiendo\ndel caso encontrado\nSe ha insertado la anomalia
en tabla 'eips' con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la
anomalia de dectectada.\n\n</h4>";
                }       ### Fin case 1

                ## Se encontrófuga de información hacia el exterior con un comportamiento anormal
                case 2{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\t\tEnvio de informacion hacia el exterior
encontrado:\n\t\t\t\t\t$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\tTrafico\n";
                        print PHP"<h3><font color='cyan'>\t\t\tEnvio de informacion hacia el
exterior encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                        print PHP"<table with border=4, width='90%'>";
                        print PHP"<tr aling='center' background BGCOLOR='#24E1DB'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                        $espe=0;
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_ori;

                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                if ($var eq $ip_ori){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";

                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $port_ori";
                                        }
                                        else {
                                                $var_tmp=$port_ori;
                                        }
                                        #$var_tmp="$var_tmp, $port_ori";
                                }# fin if

                                else {
                                        $var=$ip_ori;
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                        if ($ip_ori =~ /(\d+.\d+).(\d+).(\d+)/){
                                                $red_serv="$1.$2";
                                                #print "$red_serv ";
                                        }
                                        if (($red_serv eq '172.16.5')||($red_serv eq '172.16.10')){
                                                ($ID,$fecha)=obten_fecha_id();
                                                $query="insert into exterior values
('$ID','$fecha','$protocolo', '$ip_ori','$var_tmp', '$ip_dest')";
                                                $execute = $connect->query($query);
                                                #print "\n$query\n";
                                                print ESCRIBE "\nSe observa que la ip origen:
$ip_ori perteneciente a la red de servidores: $red_serv.0/24 esta enviando informacion con un trafico
mayor a 5 MB hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori
utiliza puertos de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en
tabla 'exterior' con ID= $ID.\n\n";
                                                print PHP"</table>";
                                                print PHP"<h4>\nSe observa que la ip origen:
$ip_ori perteneciente a la red de servidores: $red_serv.0/24 esta enviando \ninformacion con un
trafico mayor a 5 MB hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen
$ip_ori utiliza puertos origen de forma secuencial en el envio de la informacion\nSe ha insertado la
anomalia en tabla 'exterior' con ID= $ID\nY se ha notificado via email a:
'aldo\@localhost.localdomain' sobre la anomalia de dectectada.\n\n</h4>";
                                        }
                                        else {
                                                ($ID,$fecha)=obten_fecha_id();
                                                $query="insert into exterior values
('$ID','$fecha','$protocolo', '$ip_ori','$var_tmp', '$ip_dest')";
                                                $execute = $connect->query($query);
                                                #print "\n$query\n";
```

```
                                                print ESCRIBE "\nSe observa que la ip origen:
$ip_ori perteneciente a una red de usuarios esta enviando informacion con un trafico mayor a 5 MB
hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos
de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior'
con ID= $ID.\n\n";
                                                print PHP"</table>";
                                                print PHP "<h4>\nSe observa que la ip origen:
$ip_ori perteneciente a una red de usuarios esta enviando \ninformacion con un trafico mayor a 5 MB
hacia Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos
origen de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla
'exterior' con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la
anomalia de dectectada.\n\n</h4>";
                                        }
                                        $espe=$contador;
                                        ($ID,$fecha)=obten_fecha_id();
                                        print ESCRIBE "\t\t\t\tEnvio de informacion hacia el
exterior encontrado:\n\t\t\t\t\t$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\tTrafico\n";
                                        print PHP"<h3><font color='cyan'>\t\t\tEnvio de informacion
hacia el exterior encontrado:\n\t\t\t\t\t$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#24E1DB'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        #print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='10'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                $var_tmp=$port_ori;
                                }  ## Fin else
                $contador++;
                }       ## Fin For
                $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                        if ($ip_ori =~ /(\d+.\d+).(\d+).(\d+)/){
                                $red_serv="$1.$2";
                                #print "$red_serv ";
                        }

                        if (($red_serv eq '172.16.5')||($red_serv eq '172.16.10')){
                                ($ID,$fecha)=obten_fecha_id();
                                $query="insert into exterior values ('$ID','$fecha','$protocolo',
'$ip_ori','$var_tmp', '$ip_dest')";
                                 $execute = $connect->query($query);
                               #print "\n$query\n";
print ESCRIBE "\nSe observa que la ip origen: $ip_ori perteneciente a la red de servidores:
$red_serv.0/24 esta enviando informacion con un trafico mayor a 5 MB hacia Ip's fuera del rango
permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos de forma secuencial en el
envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior' con ID= $ID.\n\n";
                                print PHP"</table>";
                                print PHP"<h4>\nSe observa que la ip origen: $ip_ori perteneciente
a la red de servidores: $red_serv.0/24 esta enviando \ninformacion con un trafico mayor a 5 MB hacia
Ip's fuera del rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos origen
de forma secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior'
con ID= $ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                        }
                        else {
                                ($ID,$fecha)=obten_fecha_id();
                                $query="insert into exterior values ('$ID','$fecha','$protocolo',
'$ip_ori','$var_tmp', '$ip_dest')";
                                $execute = $connect->query($query);
                                #print "\n$query\n";
                                print ESCRIBE "\nSe observa que la ip origen: $ip_ori perteneciente
a una red de usuarios esta enviando informacion con un trafico mayor a 5 MB hacia Ip's fuera del
rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos de forma secuencial en
el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior' con ID= $ID.\n\n";
                                print PHP"</table>";
                                print PHP"<h4>\nSe observa que la ip origen: $ip_ori perteneciente
a una red de usuarios esta enviando \ninformacion con un trafico mayor a 5 MB hacia Ip's fuera del
```

```
rango permitido. \nTambien se detecta que la Ip origen $ip_ori utiliza puertos origen de forma
secuencial en el envio de la informacion\nSe ha insertado la anomalia en tabla 'exterior' con ID=
$ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                        }
                }       #### Fin case 2

                ## Ataque denegación de servicios encontradó
                case 3{
                        ($ID,$fecha)=obten_fecha_id();
                        print ESCRIBE "\n\t\t\tAtaque denegacion de servicios
encontrado:\n\t\t\t\t\$fecha\n";
                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip destino    \t:
Pto destino\tPaquetes\t Trafico\n";
                        print PHP"<h3><font color='blue'>\t\t\tAtaque denegacion de servicios
encontrado:\n\t\t\t\t\$fecha</font></h3>";
                        print PHP"<table with border=4, width='90%'>";
                        print PHP"<tr aling='center' background BGCOLOR='#08088A'><td><h3><font
color='white', aling='center'>Protocolo</font></h3></td><td><h3><font color='white',
aling='center'>IP origen</font></h3></td><td><h3><font color='white', aling='center'>Puerto
origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                        $espe=0;
                        $linea=$tmp[0];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        $var=$ip_dest;
                foreach(@tmp){
                $linea=$tmp[$contador];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);

                                if ($var eq $ip_dest){
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";


                                        if ($var_tmp){
                                                $var_tmp="$var_tmp, $ip_ori";
                                        }
                                        else {
                                                $var_tmp=$ip_ori;
                                        }
                                        #$var_tmp="$var_tmp, $ip_ori";
                                }# fin if

                                else {
                                        $var=$ip_dest;
                                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        ($ID,$fecha)=obten_fecha_id();
                                        $query="insert into DoS values
('$ID','$fecha','$protocolo', '$var_tmp', '$ip_dest', '$port_dest')";
                                        $execute = $connect->query($query);
                                        #print "\n$query\n";
                                        print ESCRIBE "\nSe observa que la Ip destino $ip_dest Esta
recibiendo multiples conexiones sobre le puerto $port_dest de una o varias Ip origen.\nCon un trafico
mayor a 50 MB. Con el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla 'DoS' con
ID= $ID.\n\n";
                                        print PHP"</table>";
                                        print PHP"<h4>\nSe observa que la Ip destino $ip_dest Esta
recibiendo multiples conexiones sobre le puerto $port_dest de una <br/>o varias Ip origen. Con un
trafico mayor a 50 MB. Con el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla
'DoS' con ID= $ID \nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia
de dectectada..\n\n</h4>";
                                        $espe=$contador;
                                        print ESCRIBE "\n\t\t\tAtaque denegacion de servicios
encontrado:\n\t\t\t\t\$fecha\n";
                                print ESCRIBE "Protocolo\tIp origen\t:   Pto origen\t   -->\t   Ip
destino    \t:  Pto destino\tPaquetes\t Trafico\n";
                                        print PHP"<h3><font color='blue'>\t\t\tAtaque de negacion
de servicios encontrado:\n\t\t\t\t\$fecha</font></h3>";
                                        print PHP"<table with border=4, width='90%'>";
                                        print PHP"<tr aling='center' background
BGCOLOR='#08088A'><td><h3><font color='white', aling='center'>Protocolo</font></h3></td><td><h3><font
color='white', aling='center'>IP origen</font></h3></td><td><h3><font color='white',
aling='center'>Puerto origen</font></h3></td><td width='40'><h3><font color='white', aling='center'>-
></font></h3></td><td><h3><font color='white', aling='center'>IP
Destino</font></h3></td><td><h3><font color='white', aling='center'>Puerto
Destino</font></h3></td><td><h3><font color='white',
```

```
aling='center'>Paquetes</font></h3></td><td><h3><font color='white',
aling='center'>Trafico</font></h3></td></tr>";
                                        $linea=$tmp[$espe];
        $ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                                        print ESCRIBE "  $protocolo   \t$ip_ori\t:\t$port_ori\t   -
->\t  $ip_dest    \t:\t$port_dest\t  $paquetes\t\t  $trafico\n";
                                        print PHP"<tr aling='center'><td><h5
aling='center'>$protocolo</h5></td><td><h5 aling='center'>$ip_ori</h5></td><td><h5
aling='center'>$port_ori</h5></td><td width='40'><h5 aling='center'>-></h5></td><td><h5
aling='center'>$ip_dest</h5></td><td><h5 aling='center'>$port_dest</h5></td><td><h5
aling='center'>$paquetes</h5</td><td><h5 aling='center'>$trafico</h5></td></tr>";
                                $var_tmp=$ip_ori;
                                }  ## Fin else
                               $contador++;
                }       ## Fin For

                        $linea=$tmp[$espe];
        ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes)=&agrupa($linea);
                        ($ID,$fecha)=obten_fecha_id();
                        $query="insert into DoS values ('$ID','$fecha','$protocolo', '$var_tmp',
'$ip_dest', '$port_dest')";
                        $execute = $connect->query($query);
                        #print "\n$query\n";
                        print ESCRIBE "\nSe observa que la Ip destino $ip_dest Esta recibiendo
multiples conexiones sobre le puerto $port_dest de una o varias Ip origen. Con un trafico mayor a 50
MB.\nCon el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla 'DoS' con ID=
$ID.\n\n";
                        print PHP"</table>";
                        print PHP"<h4>\nSe observa que la Ip destino $ip_dest Esta recibiendo
multiples conexiones sobre le puerto $port_dest de una o<br/>varias Ip origen. Con un trafico mayor a
50 MB. Con el objetivo de saturar la conexion\nSe ha insertado la anomalia en tabla 'DoS' con ID=
$ID\nY se ha notificado via email a: 'aldo\@localhost.localdomain' sobre la anomalia de
dectectada.\n\n</h4>";
                }       #### Fin case 3
        }    #### Fin switch
close (ESCRIBE);
        close (PHP);
}      ### Fin subrrutina guarda

##############################################################################################################
### Subrutina envia_correo: Si existe el archivo“anomalias.txt” se notificarámedianteél envió de un
correo electrónico que contendrá el contenido este archivo.
sub envia_correo{
        my @arreglo;
        my $tmp;

### Abrimos una tubería hacia sendmail e imprimiremos cabeceras correspondientes para el envió del
correo, además de la información contenida en el archivo “anomalias.txt”
open (MAIL,"|/usr/sbin/sendmail -t");
print MAIL "Content-Type: text/plain; charset=iso-8859-1\n";
print MAIL "To: aldo\@localhost.localdomain\n";
print MAIL "From: aldeano_demon_nfsen\@localhost.localdomain\n";
print MAIL "Subject: Asunto del mensaje-> Alerta!!!!!    Anomalias detectadas\n";
print MAIL "Content-Type: text/plain; charset=iso-8859-1\n";

open (ARCHIVO,"</Listry-AIGC/anomalias.txt");
                        @arreglo=<ARCHIVO>;
close(ARCHIVO);

foreach $tmp (@arreglo){
print MAIL "$tmp";
                }
print MAIL "\n\n\n";
        close (MAIL);
}       ### Fin subrrutina envia correo

##############################################################################################################
## Subrutina compara_ip: Ocupada por subrutina eips. Esta subrutina verificará si se presenta un
escaneo de IP’S
## Ej.   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy:zz    ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy:zz
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+1:zz  ->xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+1.yyy:zz
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+2:zz  ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+2.yyy:zz
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+3:zz  ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+3.yyy:zz
##                                                    .
##       xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy.yyy+n:zz  ->   xxx.xxx.xxx.xxx:ww => yyy.yyy.yyy+n.yyy:zz

sub compara_ip {
### Inicializamos variables ocupadas y almacénanos en ellas los datos provenientes de la subrutina
eips
my($i,$ip_ori, $ip_dest1,$ip_dest2, $ip_dest3, $port_dest, $ip_ori_tmp,
$ip_dest_tmp1,$ip_dest_tmp2,$ip_dest_tmp3,
$port_dest_tmp,$linea,$cont_tmp,$contador_anom,$contador,@tmp);
my $linea;
my ($var_ipori, $var_ipori_tmp, $var_ipdst, $var_ipdst_tmp);

        $i=shift;
        $ip_ori= shift;
        $ip_dest1=shift;
```

```
        $ip_dest2=shift;
        $ip_dest3=shift;
        $port_dest=shift;
        $ip_ori_tmp=shift;
        $ip_dest_tmp1=shift;
        $ip_dest_tmp2=shift;
        $ip_dest_tmp3=shift;
        $port_dest_tmp=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        my @tmp=@_;

### Creamos una variable que contendrá todala dirección IP (xxx.xxx.xxx.xxx) y se comparara esta
variable con255.255.255.255, en caso que se presente este dato salimos.
        my $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3";

        if($var_ipdst eq '255.255.255.255'){
                #print "No entre";
        }

## Después de comprobar que no sea un broadcast, comparamos con switch indicándonos que estamos
buscando.
#       * Case 1: Se buscara escaneo de IP’S con variación en el 4 octeto
#       * Case 2: Se buscara escaneo de IP’S con variación en el 3 octeto
        else {
                switch ($i){
                        case 1{
                                #print "\nEntre caso 1";
## En caso de encontrar que el valor del 4 octeto es 255, incrementamos en 1 el valor del 3er octeto
                                if ($ip_dest3 == 255) {
                                        $ip_dest2 ++;
                                        $ip_dest3=0;
                                #print"\n\n\n\t\t $ip_dest1.$ip_dest2.$ip_dest3:$port_dest   ";
                                }
### En caso de que sea un valor normal incrementaremos en 1 el valor del 4 octeto para proceder a la
comparación
                                else {
                                        $ip_dest3 ++;
                                }


### Creamos variables temporales que tendrán la dirección IP a comparar, además de la dirección IP
temporal.
                                $var_ipori=$ip_ori;
                                $var_ipori_tmp=$ip_ori_tmp;
                                $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3:$port_dest";

        $var_ipdst_tmp="$ip_dest_tmp1.$ip_dest_tmp2.$ip_dest_tmp3:$port_dest_tmp";
        #print "$var_ipdst:$port_dest  ===  $var_ipdst_tmp:$port_dest_tmp   ";

### En caso de que las variables IP normales y temporales coincidan en sus valores se aumenta
contador temporal y se guarda la línea que contiene la anomalía en el arreglo “tmp”, además de
guardar el número de la línea en el arreglo “encontre”
                                if (($var_ipori eq $var_ipori_tmp) && ($var_ipdst eq
$var_ipdst_tmp)){
                                        if(@tmp){
                                                push(@tmp,$linea);
                                                $contador_anom ++;
                                                push(@encontre,$cont_tmp);
                                        }
                                        else {
                                                push(@tmp,$guarde[$contador],$linea);
                                                $contador_anom +=2;
                                                push(@encontre,$cont_tmp);
                                        }
                                }

## En caso contrario decrementamos el cuarto octeto
                                else {
                                        #$continua=0;
                                        #print "Diferentes\t";
                                        $ip_dest3 --;
                                }
                        } #fin case 1

### Mismo procedimiento que case 1: La única diferencia es el incremento o decremento del 3er octeto,
dependiendo de la acción efectuada.
                        case 2{
                                #print "\nEntre caso 2";
                                $ip_dest2 ++;
                                #print" $ip_dest1.$ip_dest2.$ip_dest3:$port_dest ===
                                $ip_dest_tmp1.$ip_dest_tmp2.$ip_dest_tmp3:$port_dest_tmp  ";

                                $var_ipori=$ip_ori;
                                $var_ipori_tmp=$ip_ori_tmp;
                                $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3:$port_dest";
```

```
                        $var_ipdst_tmp="$ip_dest_tmp1.$ip_dest_tmp2.$ip_dest_tmp3:$port_dest_tmp";
                        #print "$var_ipdst:$port_dest  ===  $var_ipdst_tmp:$port_dest_tmp   ";
                                if (($var_ipori eq $var_ipori_tmp) && ($var_ipdst eq
$var_ipdst_tmp)){
                                        if(@tmp){
                                                push(@tmp,$linea);
                                                $contador_anom ++;
                                                push(@encontre,$cont_tmp);
                                        }
                                        else {
                                                push(@tmp,$guarde[$contador],$linea);
                                                $contador_anom +=2;
                                                push(@encontre,$cont_tmp);
                                        }
                                }
                                else {
                                        #$continua=0;
                                        #print "Diferentes\t";
                                        $ip_dest2 --;
                                }
                        }
                        default {

                                #print "\nopcion no valida";
                        }
                }   #fin switch
        }  #fin else
### Devolvemos valores
return ($ip_dest2,$ip_dest3,$contador_anom,@tmp);
}   #fin subrrutina compara_ip para el escaneo ip's

#################################################################################################
### Inicio subrutina compara: Ocupada por la subrutina “epuertos”. Esta subrutina se encarga de
verificar si se presenta un escaneo de puertos

##Ej             xxx.xxx.xxx.xxx:ww       =>      yyy.yyy.yyy.yyy:ww
##              xxx.xxx.xxx.xxx:ww       =>      yyy.yyy.yyy.yyy:ww+1
##              xxx.xxx.xxx.xxx:ww       =>      yyy.yyy.yyy.yyy:ww+2
##              xxx.xxx.xxx.xxx:ww       =>      yyy.yyy.yyy.yyy:ww+3
##                                       .
##              xxx.xxx.xxx.xxx:ww       =>      yyy.yyy.yyy.yyy:ww+n

sub compara {
### Declaración de variables locales y se almacenara en ellas valores obtenidos de la subrutina
“epuertos”.
my ($ip_ori,$ip_dest,$port_dest,$ip_ori_tmp,$ip_dest_tmp,$port_dest_tmp,$cont_tmp,$contador);
        my (@anomalias,@tmp);
        my $linea;
        $ip_ori= shift;
        $ip_dest=shift;
        $port_dest=shift;
        $ip_ori_tmp=shift;
        $ip_dest_tmp=shift;
        $port_dest_tmp=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        @tmp=();
        @tmp=@_;
        #print "\n$ip_ori:$port_ori-->$ip_dest:$port_dest\t\t$ip_ori_tmp:$port_ori_tmp -->
$ip_dest_tmp:$port_dest_tmp\n Linea =$linea";

## Incrementamos el puerto destino para proceder a su comparación con los valores temporales
        $port_dest ++;
        #print"\t$port_dest == $port_dest_tmp ";

### Verificamos si se presenta un incremento en el puerto destino, además si direcciones IP orígenes
y destinoson iguales, en caso de que esto se presente esto, guardamos la anomalía detectada en el
arreglo “tmp”, además guardamos en el arreglo“encontre” el número dela línea. También incrementamos
el valor de “contador_anomalo”.
        if (($ip_ori eq $ip_ori_tmp) && ($ip_dest eq $ip_dest_tmp) &&($port_dest eq
$port_dest_tmp)){
                #$continua=1;
                #print "Iguales\t $cont_tmp\t$linea \n";
                if(@tmp){
                        push(@tmp,$linea);
                        $contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
                else {
                        #print "\n$contador Arreglo tmp vacio $guarde[$contador]\n";
                        push(@tmp,$guarde[$contador],$linea);
                        $contador_anom +=2;
                        #print "@tmp\n";
                        push(@encontre,$cont_tmp);
                }
        }
```

```
## En caso contrario todo normal, decrementamos el puerto destino
        else {
                #$continua=0;
                #print "Diferentes\t";
                $port_dest --;
        }
### Devolvemos valores
return ($port_dest,$contador_anom,@tmp);
}   #### Fin subrrutina compara (creada para verificar si existe un escaneo de puertos)

#################################################################################################
### Subrutina Compara_exterior: Ocupada por la subrutina “exterior”. Esta subrutina se encarga de
verificar si se presenta una fuga de información hacia el exterior con comportamiento anómalo
## Ej            xxx.xxx.xxx.xxx:ww       =>      aaa.aaa.aaa.aaa:zz
##               xxx.xxx.xxx.xxx:ww+1     =>      aaa.aaa.aaa.aaa:zz
##               xxx.xxx.xxx.xxx:ww+2     =>      aaa.aaa.aaa.aaa:zz
##               xxx.xxx.xxx.xxx:ww+3     =>      aaa.aaa.aaa.aaa:zz
##                                        .
##               xxx.xxx.xxx.xxx:ww+n     =>      aaa.aaa.aaa.aaa:zz
## Donde aaa.aaa.aaa.aaa Es una Ip exterior a la red normal

sub compara_exterior{
## Declaración de variables locales a ocupar y se les asigna la información recibida de subrutina
exterior.
my($ip_ori, $port_ori, $ip_dest1, $ip_dest2, $ip_dest3, $ip_ori_tmp, $port_ori_tmp,
$ip_dest_tmp1,$linea,$cont_tmp,$contador_anom,$contador,$trafico,$trafico_tmp,@tmp);
        $ip_ori= shift;
        $port_ori=shift;
        $ip_dest1=shift;
        $ip_dest2=shift;
        $ip_dest3=shift;
        $ip_ori_tmp=shift;
        $port_ori_tmp=shift;
        $ip_dest_tmp1=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        $trafico=shift;
        $trafico_tmp=shift;
        my @tmp=@_;
        my $red_serv;

### Se incrementa el valor del puerto origen.
        $port_ori ++;

## Se crea una variable especial que contendrá la ip_dst yyy.yyy.yyy.yyy y se compara con
255.255.255.255, en caso de ser verdadero termina la subrutina y se decremento el puerto origen.
        my $var_ipdst="$ip_dest1.$ip_dest2.$ip_dest3";

        if ($ip_ori_tmp =~ /(\d+.\d+).(\d+).(\d+)/){
                $red_serv="$1.$2";
                #print "$red_serv ";
        }
        if ($trafico_tmp =~ /(\d+|\d+.\d+)(\s)(K|M|G|T)/){
                switch ($3){
                        case 'K'{
                                $trafico_tmp= $1*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'M'{
                                $trafico_tmp= $1*1024*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'G'{
                                $trafico_tmp= $1*1024*1024*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'T'{
                                $trafico_tmp= $1*1024*1024*1024*1024;
                                #print "$trafico_tmp \t= $1 * $3\n";
                        }
                } #Fin switch
        } ### Fin if

        if($var_ipdst eq '255.255.255.255'){
                #print "\t $var_ipdst  --->>  No entre";
                $port_ori --;
        }

## En caso contrario se compara el 1 y 2 octeto con direcciones IP destino válidas y si se presenta
esto salimos de la subrutina.
        elsif ( ($ip_dest_tmp1 eq '255.255') || ($ip_dest_tmp1 eq '172.16') || ($ip_dest_tmp1 eq
'172.16') || ($ip_dest_tmp1 eq '172.17')) {

                #print "\t $var_ipdst --->><<----- EnTrE \t $ip_dest_tmp1 \n ";
                $port_ori --;
```

```
        }
## En caso que no se cumpla la condición anterior se tiene una dirección IP exterior, comparamos si
tiene un comportamiento mencionado en el ejemplo, de ser así almacenamos el valor de la línea que
presenta este comportamiento anormal, además de almacenar el número de esta línea en el arreglo
“encontre” e incrementamos el contador.
        #elsif (($ip_ori eq $ip_ori_tmp) && ($port_ori eq $port_ori_tmp)) {
        elsif ((($red_serv eq '172.16.5')||($red_serv eq '172.16.10'))&&($ip_ori eq
$ip_ori_tmp)&&($port_ori eq $port_ori_tmp)&&($trafico_tmp >= 5242880) || (($ip_ori eq
$ip_ori_tmp)&&($port_ori eq $port_ori_tmp)&&($trafico_tmp >= 5242880))) {
                #print"\t Encontre anomalia $var_ipdst --->> \t $ip_dest_tmp1   .   $ip_dest_tmp2 \n
";
                if(@tmp){
                        push(@tmp,$linea);
                        $contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
                else {
                        push(@tmp,$guarde[$contador],$linea);
                        $contador_anom +=2;
                        push(@encontre,$cont_tmp);
                }
        }
## No se detectó nada, se decremento el valor del puerto origen
        else { $port_ori --;}
## Devolvemos valores requeridos.
return ($port_ori,$contador_anom,@tmp);
}  ### Fin subrrutina compara_exterior

#############################################################################################################
### Subrutina Compara_dos: Ocupada por la subrutina“DoS”. Esta subrutina se encarga de verificar si
se presenta un ataque denegación de servicios o un ataque de negación de servicios distribuido.
## Ej
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##                                       .
##              aaa.aaa.aaa.aaa:ww       =>      xxx.xxx.xxx.xxx:zz
##Donde:  aaa.aaa.aaa.aaa es cualquier ip.
##              ww cualquier puerto origen igual en todas las ip aaa.aaa.aaa.aaa
##              xxx.xxx.xxx.xxx. es cualquier ip destino igual.
##              zz cualquier puerto destino
##              Además debe de cumplir que el tráfico sea mayor a 50 M

sub compara_dos{
##Declaración de variables locales a ocupar y se les asigna la información recibida de subrutina DoS.

my ($port_dest,$ip_dest,$trafico,$port_dest_tmp,$ip_dest_tmp,$trafico_tmp,$cont_tmp,$contador);
        my (@anomalias,@tmp);
        my $linea;
        $port_dest=shift;
        $ip_dest=shift;
        $trafico=shift;
        $port_dest_tmp=shift;
        $ip_dest_tmp=shift;
        $trafico_tmp=shift;
        $linea=shift;
        $cont_tmp=shift;
        my $contador_anom=shift;
        $contador=shift;
        @tmp=@_;

## El valor almacenado del tráfico en cada flujo se convertirá a bytes.
        if ($trafico_tmp =~ /(\d+|\d+.\d+)(\s)(K|M|G|T)/){

                switch ($3){
                        case 'K'{
                            $trafico_tmp= $1*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'M'{
                            $trafico_tmp= $1*1024*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'G'{
                            $trafico_tmp= $1*1024*1024*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                        case 'T'{
                            $trafico_tmp= $1*1024*1024*1024*1024;
                            #print "$trafico_tmp \t= $1 * $3\n";
                        }
                } #Fin switch
        } ### Fin if
        else {
                #print "$trafico_tmp \n";
        }
```

```
## Empezamos comparación. En caso de encontrar broadcats salimos.
        if ($ip_dest_tmp eq '255.255.255.255'){
                #print "No entro\n";
        }

##  Verificamos si presenta un comportamiento anormal: Dirección IP destino actual sea igual a la
dirección IP destino temporal, además de que los puertos origen sean los mismos y el tráfico sea
mayor a 50 M, en caso de presentarse esta anomalía se guarda esta linea como un comportamiento
anormal
        elsif (($ip_dest eq $ip_dest_tmp) &&($port_dest eq $port_dest_tmp) && ($trafico_tmp >=
52428800)){
                #$continua=1;
                #print "Iguales\t $cont_tmp\t$linea \n";
                if(@tmp){
                        push(@tmp,$linea);
                        $contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
                else {
                        push(@tmp,$guarde[$contador],$linea);
                        $contador_anom +=2;
                        #print "Se incremento mi contador en =\t$contador_anom\n";
                        #push(@tmp,$linea);
                        #$contador_anom ++;
                        push(@encontre,$cont_tmp);
                }
        }

## En caso contrario todo normal, salimos
        else {
                #print "Diferentes\t";
        }

### Devolvemos valores
return ($port_dest,$contador_anom,@tmp);
#return (@anomalias);

}       ### Fin subrrutina compara_dos

#############################################################################################################
####                              Subrutina separa:
### Se encarga de separar la información recibida por el colector nfdump de la siguiente forma:
######                            (protocolo) , (ip ori : pto ori) , (ip dst : pto dst)
### Y almacenarla en el arreglo guarda.
sub separa{
        my ($linea, $var, $banderin);
        my @ip;

        $linea=shift;
        @ip=@_;
        $banderin=1;
        #print "$linea\n Recibi arreglo";
                        #print "\n$linea";

### Por medio de expresiones regulares separamos la información de la forma indicada y la guardamos
en el arreglo temporal llamado “ip”
                        if( $linea =~ /ICMP (\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+.\d+)(\s+)(\d+)(\s+)(\d+)/){

                $var= "ICMP $2 -> $5 $7 $9";
                                push(@ip,$var);
                                $banderin=0;
                }

                        if( $linea =~  /([a-zA-Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)/){
                        #$var= "$1 $3 -> $6 $8 $10 $12 ";
                               $var= "$1 $3 -> $6 $8 $10 $12 $14";
                push(@ip,$var);
                                $banderin=0;
                }

                        if( ($banderin==1) && ($linea =~  /([a-zA-
Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s+)(\d+)/)){
                        $var= "$1 $3 -> $6 $8 $10 $12 ";
                               #$var= "$1 $3 -> $6 $8 $10 $12 $14";
                push(@ip,$var);
                                $banderin=0;
                }

                        if( ($banderin==1) && ($linea =~  /([a-zA-
Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)-
>(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+)(\s+)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
                        $var= "$1 $3 -> $6 $8 $10 $12 ";
                               #$var= "$1 $3 -> $6 $8 $10 $12 $14";
                push(@ip,$var);
                                $banderin=0;
```

```
                }

                        if( ($banderin==1) && ($linea =~  /([a-zA-
Z]+)(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)->(\s+)(\d+.\d+.\d+.\d+:\d+)(\s+)(\d+)(\s+)(\d+)/)){

                $var= "$1 $3 -> $6 $8 $10";
                                push(@ip,$var);
                        }
        #print "\nHola imprimo \t@ip";

### Regresamos el arreglo que contiene toda la infroamcion separada
        return (@ip);
} ####fin subrrutina.


#################################################################################################
#####                             Subrutina agrupa
###Ocupada por la subrutina “epuertos” y “DoS”, la información obtenida por la subrutina “separa”, se
agrupará de la siguiente forma y será pasada como argumento a la subrutina “compara” o “compara_dos”
con el objetivo de verificar si se ha realizado un escaneo de puertos o un ataque de negación de
servicios.
######   xxx.xxx.xxx.xxx :  xxxxx    ->  xxx.xxx.xxx.xxx :  xxxxx        xx
######    (ip ori)    , (pto ori)  ,     (ip dst)     , (pto dst) , (trafico)

sub agrupa {
### Declaramos variables en donde se guardara información recibida y variables que almacenaran la
información separada por las expresiones regulares, con el formato indicado
        my ($linea,$ip_ori,$ip_dest,$port_ori,$port_dest,$trafico,$protocolo,$paquetes);
        $linea=shift;
        my $banderin=1;

        if( $linea =~ /ICMP(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+.\d+)(\s)(\d+)(\s)(\d+)/){
                $ip_ori=$2;     $port_ori=$3;   $ip_dest=$6;    $port_dest=$7;  $trafico=$11;
        $protocolo="ICMP";       $paquetes=$9;
                #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico\t";
                $banderin=0;
        }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
                $ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico="$14 $16";
        $protocolo="$1"; $paquetes="$10 $12";
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico $14\t";
                $banderin=0;
         }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+)/)){
                $ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico="$14";
        $protocolo="$1"; $paquetes="$10 $12";
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico $14\t";
                $banderin=0;
         }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
                $ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico="$12 $14";
        $protocolo="$1"; $paquetes=$10;
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico $14\t";
                $banderin=0;
         }

        if( ($banderin==1) && ($linea =~ /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)(\d+)(\s)(\d+)/)){
$ip_ori=$3;     $port_ori=$4;   $ip_dest=$7;    $port_dest=$8;  $trafico=$12;       $protocolo="$1";
        $paquetes=$10;
                 #print "$ip_ori:$port_ori -> $ip_dest:$port_dest \tTrafico= $trafico\t";
        }

## Devolvemos estas variables que contiene la informaciÃ³n agrupada.
return ($ip_ori,$port_ori,$ip_dest,$port_dest,$trafico,$protocolo,$paquetes);

}    #### fin subrrituna agrupa

#################################################################################################
###                             Subrutina agrupa_ip
###   Ocupada por la subrutina “eips” y “exterior”, la información obtenida por la subrutina “separa”
se agrupará de la siguiente forma y será pasada como argumento a la subrutina
“compara_ip”o“compara_exterior” respectivamente  con el objetivo de verificar si se ha realizado un
escaneo de IP’S o se ha enviado información hacia el exterior.
######   xxx.xxx.xxx.xxx :  xxxxx     ->   xxx.xxx  .    xxx   .   xxx    :  xxxxx        xx
######   (ip ori)        , (pto ori)  ,   (ip dst1) , (ip dst2), (ip dst3), (pto dst)  (trafico)

sub agrupa_ip{
### Declaramos variables en donde se guardara información recibida y variables que almacenaran la
información separada por las expresiones regulares, con el formato indicado
```

```
        my ($linea, $ip_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_ori,$port_dest,$trafico);
        $linea=shift;
        my $banderin=1;

       if( $linea =~ /ICMP(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+.\d+)(\s)(\d+)(\s)(\d+)/){
                $ip_ori=$2;     $port_ori=$3;   $port_dest=$9;  $trafico=$13;
                $ip_dest1=$6;    $ip_dest2=$7;    $ip_dest3=$8;

               #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico\t";
                $banderin=0;
        }

        if( ($banderin==1) && ($linea =~ /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){
               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico="$16 $18";
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico\t";
                $banderin=0;
        }

        if( ($banderin==1) && ($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)(\s)(\d+)/)){
               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico="$16";
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico $16\t";
                $banderin=0;
       }

        if( ($banderin==1) &&($linea =~  /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+)(\s)(\d+|\d+.\d+)(\s)(K|M|G|T)/)){

               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico="$14 $16";
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico $16\t";
                $banderin=0;
       }

        if( ($banderin==1) && ($linea =~ /([a-zA-Z]+)(\s)(\d+.\d+.\d+.\d+):(\d+)(\s)-
>(\s)(\d+.\d+).(\d+).(\d+):(\d+)(\s)(\d+)(\s)(\d+)/)){

               $ip_ori=$3;     $port_ori=$4;   $port_dest=$10;  $trafico=$14;
                $ip_dest1=$7;    $ip_dest2=$8;    $ip_dest3=$9;
                #print "$ip_ori:$port_ori -> $ip_dest1.$ip_dest2.$ip_dest3:$port_dest \tTrafico=
$trafico\t";

        }

return ($ip_ori,$port_ori,$ip_dest1,$ip_dest2,$ip_dest3,$port_dest,$trafico);
}    ################fin subbrutina agrupa_ip#############################
# The Cleanup function is called, when nfsend terminates. It's purpose is to give the
# plugin the possibility to cleanup itself. It's return value is discard.
sub Cleanup {
syslog("info", "Plugin escaneo Cleanup");
# not used here
}

1;
```

Módulo “escaneo.php”

```
<?php
/*****************************************************************
*****************************************************************
**********           Plugin escaneo.php              ********
**********  Creado por: Aldo Iván Girón Capistrán      ********
**********  Última fecha de modificación: 1/08/2010  ********
*****************************************************************
*****************************************************************

*                    Objetivo del plugin
* Mostrar los resultados obtenidos por escaneo.pm en el navegador web
*/

function escaneo_ParseInput( $plugin_id ) {
/*
      Esta función no es ocupada en este plugin, pero debe de existir.
      */
} // Fin escaneo_ParseInput

/*
        Función que se ejecutará al momento de mandar llamar al plugin en el navegador web
*/
function escaneo_Run( $plugin_id ) {
```

```
global $VARDIR;
// Verificamos que estemos en el profile live
if ($_SESSION['profile'] == 'live') {

                print "<h2><b><font color='red', aling='center'>\t\t\tObjetivo del
plugin:<br/>Analizar el ultimo archivo nfcapd obtenido en busqueda de anomalias tipicas de algun
malware.</font></b></h2>";

// Abrimos el archivo a mostrar en caso de haber detectado anomalías y mostramos el contenido del
archivo
                if (fopen("/Listry-AIGC/anomalias_php.txt","r")){

                        print "<h2><b><font color='green', aling='center'>Se encontraron las
siguientes anomalias:</font></b></h2>";
                        echo "<pre>";

                        $lines = file("/Listry-AIGC/anomalias_php.txt");

                        foreach ($lines as $line){
                                echo "$line";
                        }
                        echo "</pre>";
                }
// En caso de no existir el archivo notificamos que no se encontró ninguna anomalía
                else {
                        print "<h2><b><font color='green', aling='center'>No se encontraron
anomalias en el analisis realizado</font></b></h2>";
                }
//Mostramos el archivo que se capturo con ayuda de nfcapd.
                print "\n\n";
                print "<h2><b><font color='green', align='center'>Se analizo el
archivo:</font></b></h2>\n";
        echo "<pre>";
        #$logfile = "$VARDIR/tmp/trackstats.log";
        $logfile="/Listry-AIGC/salida.txt";
        $lines = file($logfile) ;

        foreach ($lines as $line) {
echo htmlspecialchars($line) ;
         }
        echo "</pre>";
        } else {
print "<h3>plugin not applicable for profile: " .$_SESSION['profile'] . "</h3>" ;
       }
} // Fin Escaneo_Run
?>
```

Bibliografia y Referencias

Libros:

- Seguridad Informática
  Juan José Nombela
  Editorial Paraninfo
  1997

- Security Engineering. A guide to building dependable distributed system
  Ross J. Anderson.
  Publishing house: Wiley Computer
  2001

- Secure Computers and Networks
  Analysis, design, and implementation
  Eric A. Fisch.
  Gregory B. White
  Publishing house: CRC Press LLC
  2000

- Fundamentos de seguridad informática
  María Jaquelina López Barrientos
  Cintia Quezada Reyes
  Facultad de Ingenieria, Universidad Nacional Autonoma de México
  2006

Apuntes de clases:

- Apuntes de clase de Redes de Datos.
  Profesor: M.C. Alejandro Velázquez Mena

- Apuntes de clase de Administración de redes.
  Profesor: M.C. Alejandro Velázquez Mena

- Apuntes de clase de Criptografía.
  Profesor: M.C. Jaquelina López Barrientos.

- Apuntes de clase de seguridad informática I
  Profesor: M.C. Cintia Quezada Reyes.

- Apuntes de clase de seguridad informática II
  Profesor: ING Jesús Ramírez Pichardo.

Referencias:

NetFlow v9 vs. NetFlow v5: What are the differences?
http://www.plixer.com/blog/netflow/netflow-v9-vs-netflow-v5/
Última fecha de revisión: Abril 2010

Cisco IOS NetFlow White Papers
http://www.cisco.com/en/US/products/ps6601/prod_white_papers_list.html
Última fecha de revisión: Abril 2010

Monitorización de redes y análisis del tráfico mediante Cisco Netflow
http://docsharing.wordpress.com/2007/12/09/monitorizacion-de-redes-y-analisis-del-trafico-mediante-cisco-netflow/
Última fecha de revisión: Abril 2010

Argus and Netflow
http://www.qosient.com/argus/argusnetflow.htm
Última fecha de revisión: Mayo 2010

cflowd: Traffic Flow Analysis Tool
http://www.caida.org/tools/measurement/cflowd/
Última fecha de revisión: Mayo 2010

SourceForge.net NFDUMP
http://nfdump.sourceforge.net/
Última fecha de revisión: Mayo 2010

NfSen - Netflow Sensor
http://nfsen.sourceforge.net/
Última fecha de revisión: Mayo 2010

Free NetFlow Tools
http://www.networkuptime.com/tools/netflow/ehnt.html
Última fecha de revisión: Mayo 2010

flowd
http://www.mindrot.org/flowd.html
Última fecha de revisión: Mayo 2010

FlowScan - Network Traffic Flow Visualization and Reporting Tool
http://www.caida.org/tools/utilities/flowscan/
Última fecha de revisión: Mayo 2010

ntop
http://www.ntop.org/
Última fecha de revisión: Mayo 2010

Welcome to Stager
http://software.uninett.no/stager/
Última fecha de revisión: Mayo 2010

CERT NetSA Security Suite Monitoring for Large-Scale Networks
http://tools.netsa.cert.org/silk/
Última fecha de revisión: Julio 2010

CERT
http://www.cert.org/
Última fecha de revisión: Julio 2010

SANS
http://www.sans.org/
Última fecha de revisión: Julio 2010

insecure.org
http://insecure.org/
Última fecha de revisión: Julio 2010

Proyecto Malware - UNAM-CERT
http://www.malware.unam.mx/
Última fecha de revisión: Julio 2010

Definiciónm de MAN
http://www.mastermagazine.info/termino/5664.php
Última fecha de revisión: Mayo 2010

vsantivirus estar informado para estar seguro
http://www.vsantivirus.com/
Última fecha de revisión: Agosto 2010

El Virus Conficker: Una Guía Para Conocerlo
http://peleandomecontodos.blogspot.com/2009/03/el-virus-conficker-una-guia-para.html
Última fecha de revisión: Septiembre 2010

Comportamiento de Virus en plataformas Windows
http://www.elhacker.net/comportamiento-virus.htm
Última fecha de revisión: Septiembre 2010

Network Working Group R. Shirey
http://www.ietf.org/rfc/rfc2828.txt
Última fecha de revisión: Junio 2010

Top 10 del Malware en América Latina
http://threatpost.com/es_la/blogs/top-10-del-malware-en-america-latina-110209
Última fecha de revisión: Septiembre 2010

RFC 1135 - Helminthiasis of the Internet
http://www.faqs.org/rfcs/rfc1135.html
Última fecha de revisión: Junio 2010

Cisco Systems NetFlow Services Export Version 9
http://www.ietf.org/rfc/rfc3954.txt
Última fecha de revisión: Septiembre 2010

What is IPFIX vs. NetFlow v9?
http://www.plixer.com/blog/netflow/what-is-ipfix-vs-netflow-v9/
Última fecha de revisión: Septiembre 2010

Segu-Info SEGURIDAD EN LA INFORMACIÓN 11 años educando seguridad
http://www.segu-info.com.ar/
Última fecha de revisión: Septiembre 2010

"PIRATAS INFORMATICOS"
http://usuarios.lycos.es/RAMCHOS/piratas_inf.html#TIPOS
Última fecha de revisión: Agosto 2010

SEGURIDAD INFORMATICA
http://www.seguridadinformatica.es/
Última fecha de revisión: Septiembre 2010