DISEÑO DE UN SISTEMA DE DETECCIÓN DE INTRUSOS SNORT Y UN SISTEMA TRAMPAS TIPO HONEYPOTS DE BAJA INTERACCIÓN EN LA RED DEL GRUPO DE INVESTIGACIÓN GEOMÁTICA

KELLY YADIRA MARTÍNEZ FLÓREZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2014

DISEÑO DE UN SISTEMA DE DETECCIÓN DE INTRUSOS SNORT Y UN SISTEMA TRAMPAS TIPO *HONEYPOTS* DE BAJA INTERACCIÓN EN LA RED DEL GRUPO DE INVESTIGACIÓN GEOMÁTICA

KELLY YADIRA MARTÍNEZ FLÓREZ

Proyecto de grado para optar al título de

ESPECIALISTA EN TELECOMUNICACIONES

Director:

OSCAR MAURICIO REYES TORRES

Dr. Ingeniería Electrónica

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y DE TELECOMUNICACIONES
ESPECIALIZACIÓN EN TELECOMUNICACIONES
BUCARAMANGA
2014

DEDICATORIA

A Dios por ser el refugio en cada momento de mi vida,

A mi ***Bibi***, por ser el esposo y el amigo incondicional, la guía para seguir creciendo como persona y profesional. Gracias ***Bibi*** por su apoyo y entrega.

A mi hermosa princesa ***Salomé***, por ser la alegría de mis días y el motivo para seguir luchando.

A mi familia por su cariño.

Kelly Yadira Martínez Flórez

AGRADECIMIENTOS

El autor expresa sus agradecimientos a:

A Dios Todopoderoso por su apoyo y compañía incondicionalmente.

Al grupo de investigación Geomática y su Director Prof. Hernán Porras Díaz, por brindarme su apoyo durante la especialización y a lo largo del desarrollo de esta monografía.

Al Director de la monografía, Prof. Oscar Mauricio Reyes, por el tiempo dedicado a su ayuda metodológica en la revisión y el cumplimiento del presente trabajo.

Al Ing. Elver Omar Gallo Lancheros, por su acompañamiento y quien dedicó parte de su tiempo para guiarme en el desarrollo de este proyecto.

Al Ing. Mauricio Moreno por su apoyo y a todas aquellas personas que de una u otra forma colaboraron en la realización de esta monografía.

TABLA DE CONTENIDO

Pág.

INTRODUCCIÓN 15

1 MARCO TEORICO 17
1.1 ATAQUES INFORMÁTICOS 17
1.1.1 Ataques de red 17
1.1.2 Ataques de autentificación 18
1.2 SISTEMAS DE DEFENSA DE ATAQUES INFORMÁTICOS 19
1.2.1 *Snort* 19
1.2.2 *Honeypots* 20

2 ANALISIS DE LA RED 22
2.1 TOPOLOGÍA LÓGICA DE LA RED DE GEOMÁTICA 22
2.2 TOPOLOGÍA FISICA DE LA RED DE GEOMÁTICA 23
2.2.1 *Rack* N° 1: *Rack* de Estaciones de Trabajo 27
2.2.2 Rack N°2: Data Center Geomática 28
2.3 ANÁLISIS DE TRÁFICO EN LA SUBRED DE SERVIDORES 30
2.3.1 Interfaz gráfica del análisis de tráfico 32
2.3.2 Resultados del análisis de tráfico en los servidores 32

3 REQUERIMIENTOS PARA *SNORT* Y *HONEYPOTS* DE BAJA INTERACCIÓN 37
***3.1* REQUERIMIENTOS PARA EL *SNORT* 37**
3.1.1 Requisitos de Hardware 37
3.1.2 Requisitos de Software 38
3.2 Requerimientos para el *Honeypots* 39
3.2.1 Captura de los datos 39
3.2.2 Recolección y análisis de datos 39
3.2.3 Requisitos de Hardware 39
3.2.4 Requisitos de Software 40

4 DISEÑO DEL *SNORT* Y EL *HONEYPOTS* DE BAJA INTERACCIÓN SOBRE LA RED DE GEOMÁTICA 42
4.1 DISEÑO DEL *SNORT* EN LA SUBRED DE SERVIDORES 43
4.2 DISEÑO DEL *HONEYPOTS* DE BAJA INTERACCIÓN EN LA SUBRED DE PRODUCCIÓN (192.168.85/24) 48
4.3 DISEÑO CONSOLIDADO DEL *SNORT* Y *HONEYPOTS* PARA LA RED DE GEOMÁTICA 53

5 VALIDACIÓN DEL DISEÑO 55
5.1 VALIDACIÓN DEL DISEÑO DEL *SNORT* 55

5.1.1 Inicialización del servicio *Snort* 55
5.1.2 Pruebas de Validación 55
5.2 VALIDACIÓN DEL DISEÑO DEL *HONEYPOTS* DE BAJA INTERACCIÓN 60
5.2.1 Inicialización de los servicios 60
5.2.2 Pruebas de validación.....62

6 CONCLUSIONES 70

7 RECOMENDACIONES 72

BIBLIOGRAFÍA.....75

ANEXOS.....78

TABLA DE FIGURAS

Pág.

Figura 1 Arquitectura Snort 20
Figura 2 Topología Lógica de la red de Geomática dentro de la red de la UIS 23
Figura 3 Topología Física de la red de Geomática 24
Figura 4 Diagrama de los componentes físicos de la red del grupo Geomática 26
Figura 5 Ubicación del equipo para monitorizar el tráfico en la subred de servidores 31
Figura 6 Interfaz gráfica del analizador de tráfico Ntop 32
Figura 7 Estadística de los tres primeros protocolos que más presentan tráfico en el servidor de Aplicaciones 33
Figura 8 Detalle de los protocolos que se usan en el servidor de Aplicaciones 34
Figura 9 Estadística de los tres primeros protocolos que más presentan tráfico en el servidor Web 35
Figura 10 Detalle de los protocolos que se usan en el servidor Web 36
Figura 11 Diseño del Snort en la subred 192.168.19.0/24 donde se encuentran los servidores del grupo Geomática 45
Figura 12 Diagrama de flujo del funcionamiento del Snort de Geomática 47
Figura 13 Diseño preliminar del Honeypots Virtual sobre un equipo real de la subred de producción 192.168.85.0/24 49
Figura 14 Diseño del Honeypots de baja interacción para la subred de producción 192.168.85.0/24 del grupo Geomática 51
Figura 15 Esquema unificado del diseño del Snort y Honeypots sobre la red del grupo Geomática 54
Figura 16 Ejecución del Snort 55
Figura 17 Ataque tipo ping al servidor 192.168.19.57 57
Figura 18 Alerta disparada por el Snort debido a un ataque de tipo Ping 58
Figura 19 Ataque de tipo Escaneo de direcciones IP con bandera FIN activada 58
Figura 20 Alerta generada por el Snort debido al ataque de Escaneo de direcciones IP 59
Figura 21 Alerta generada debido al ataque de tráfico UDP 59
Figura 22 Estructura del archivo /etc/default/honeyd 60
Figura 23 Inicialización del servicio farpd 61
Figura 24 Emulación de las máquinas trampa configurados en el Honeyd 62
Figura 25 Ataque de tipo Ping 64
Figura 26 Respuesta de las máquinas trampa al tipo de ataque Ping 65
Figura 27 Ataque de Escaneo de direcciones IP 65
Figura 28 Respuesta de los Honeypots al tipo de ataque Escaneo de direcciones IP 66
Figura 29 Ataque de Escaneo de Puertos con Advanced Port Scanner 66
Figura 30 Respuesta de los Honeypots al tipo de ataque Escaneo de Puertos 67
Figura 31 Validación del servicio web del Honeypots Windows XP 67

Figura 32 Validación del servicio ftp en el Honeypots Windows Server 2003.....68
Figura 33 Validación del servicio FTP en el Honeypots Linux Suse 69
Figura 34 Sitio oficial de Winpcap.....90
Figura 35 Instalación de la librería Winpcap 91
Figura 36 Sitio oficial del Snort 92
Figura 37 Instalación del Snort 2.9.6.2 92
Figura 38 Descarga del archivo de Reglas del Snort.....93
Figura 39 Configuración de Snort.conf – ipvar.....94
Figura 40 Configuración de Snort.conf - rutas de acceso 94
Figura 41 Configuración de Snort.conf – preprocesadores.....94
Figura 42 Configuración de Snort.conf – metadatos.....95
Figura 43 Configuración de Snort.conf - lista negra y blanca 95
Figura 44 Configuración de Snort.conf – salidas 96
Figura 45 Ubicación del Honeypots antes del Firewall 99
Figura 46 Ubicación del Honeypots después del Firewall.....100
Figura 47 Ubicación del Honeypots en la Zona Desmilitarizada.....101
Figura 48 Instalación del Honeyd.....102
Figura 49 Instrucción para la instalación de la herramienta farpd.....102
Figura 50 Instalación de la herramienta farpd.....103
Figura 51. Archivos de firmas del Honeypots 103
Figura 52 Archivo de configuración del Honeyd 105
Figura 53 Directorio donde se encuentran los servicios de la familia TCP/IP 105
Figura 54 Estructura del script iis.sh.....106
Figura 55 Archivo de configuración honeyd.conf 107

LISTA DE TABLAS

Pág.

Tabla 1. Ventajas y desventajas de los Honeypots.....21
Tabla 2. Descripción de las Salas de Cómputo del grupo Geomática 28
Tabla 3 Descripción Equipos que están conectados al Rack 2.....29
Tabla 4. Especificaciones técnicas del equipo donde se instaló el Snort 37
Tabla 5. Especificaciones técnicas del equipo donde se instaló el Honeypots Virtual.....40
Tabla 6. Preprocesadores del Snort 82
Tabla 7. Estructura del Encabezado de una regla 84
Tabla 8. Opciones Generales de las reglas de Snort.....85
Tabla 9. Opciones Payload de las reglas de Snort 85
Tabla 10. Opciones non-Payload de las reglas de Snort.....86
Tabla 11. Opciones de Post-detection de las reglas del Snort 87
Tabla 12. Módulos de salida del Snort.....89
Tabla 13. Respuesta del puerto frente a una actividad.....104
Tabla 14. Sistemas operativos y servicios montados en las máquinas trampa ...107

LISTA DE ANEXOS

Pág.

ANEXO A. GENERALIDADES DEL *SNORT* 78
ANEXO B. GENERALIDADES DEL *HONEYPOTS* 97

RESUMEN

TITULO: DISEÑO DE UN SISTEMA DE DETECCIÓN DE INTRUSOS *SNORT* Y UN SISTEMA TRAMPAS TIPO *HONEYPOTSS* DE BAJA INTERACCIÓN EN LA RED DEL GRUPO DE INVESTIGACIÓN GEOMÁTICA[*]

AUTOR: KELLY YADIRA MARTÍNEZ FLÓREZ.[**]

PALABRAS CLAVE: DETECCIÓN DE INTRUSOS, SISTEMAS DE TRAMPAS, SUBRED, VULNERABILIDAD, DISEÑO, INTERACCIÓN.

DESCRIPCIÓN:

El presente trabajo de monografía se deriva de la necesidad que tiene el grupo Geomática, de tomar medidas con respecto a la seguridad de la información que circula a través de su red interna y que es almacenada en los servidores que se encuentran ubicados en el DataCenter del grupo. Con base en la topología de red tipo estrella implementada en Geomática por parte de la división de servicios de información de la Universidad, se plantea el diseño de un sistema de detección de intrusos tipo *Snort* junto con un sistema de trampas de baja interacción tipo *Honeypots*, como una estrategia que permita contrarrestar algunas fallas de seguridad que se han venido presentado en la red como es el caso de pérdida de la información o el uso indebido de la misma.

Para tal fin, se llevó a cabo un análisis de la red con el propósito de identificar cuáles son los protocolos y puertos de tienen mayor tráfico en los servidores, con el objetivo de proponer algunas reglas que sirvan como base para la validación del diseño del sistema de detección de intrusos que se plantea para esta subred.

Por otra parte, en la subred de producción que comprende las salas de informática, los laboratorios de investigación y las impresoras, se propone el diseño de un sistema de trampas de baja interacción o *Honeypots*, con el fin de atraer a los posibles atacantes que se están conectando a través dela red interna con el propósito de realizar ataques a algunos de los equipos de la red o a los servidores.
De esta manera se plantean dos diseños, el primero consiste en un sistema de detección de intrusos para la subred donde se encuentran los servidores y el segundo es un sistema de trampas de baja interacción para la subred de producción.

[*] Monografía

[**] Facultad de Ingenierías Fisico-Mecanicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Especialización en Telecomunicaciones. Director: Ing. Oscar Mauricio Reyes.

ABSTRACT

TITLE: SYSTEM DESIGN *SNORT* INTRUSION DETECTION SYSTEM TYPE AND TRAPS *HONEYPOTSS* LOW INTERACTION NETWORK GEOMÁTICA RESEARCH GROUP[*].

AUTHOR: KELLY MARTINEZ YADIRA FLÓREZ [**].

KEYWORDS: INTRUSION DETECTION SYSTEMS CHEATING, SUBNET INFRINGEMENT, DESIGN, INTERACTION.

DESCRIPTION:

This working paper is derived from the need of the Geomatica group, taking measures against the security of the information flowing through its internal network and is stored on servers that are located in the Data Center of the group. Based on network topology like star Geomatica implemented by the information services division of the University are involved in designing a system of *Snort* intrusion detection system with type traps low interaction *Honeypots* type, as a strategy to counteract some security flaws that have been presented in the network as in the case of loss of information or misuse it.

To this end, we carried out an analysis of the network in order to identify the protocols and ports have increased traffic on the servers, in order to propose some rules that serve as a basis for validation of the system design are intrusion detection that arises for this subnet.

Moreover, the production subnet comprising computer labs, research labs and printers, designing a system traps or low interaction *Honeypots* is proposed in order to attract potential attackers who are connecting through the internal network for the purpose of carrying out attacks on some of the computers on the network or servers.

Thus arise two designs, the first consisting of an intrusion detection system for the subnet where the servers and the second is a low trap system interaction for the production subnet.

[*] Monograph

[**] Faculty of Mechanical Engineering-Physical. School of Electrical Engineering, Electronics and Telecommunications. Specialization in Telecommunications. Director: Eng. Oscar Mauricio Reyes.

INTRODUCCIÓN

Desde hace algunas décadas el uso del internet tenía propósitos de tipo militar, después se fue expandiendo hasta cubrir áreas como la investigación y las organizaciones empresariales; a partir de este este tiempo, se inicia una revolución en la tecnología que conlleva al uso del internet de forma masiva con la aparición del comercio electrónico y especialmente las redes sociales. De esta manera, se ha “abierto la puerta” para que cualquier equipo que esté conectado a internet o a una red interna pueda ser atacado. Cada vez se incrementa el número de ataques, la facilidad con la que son realizados y el gran daño que éstos producen. Por esta razón, se hace necesario concentrar los esfuerzos en el estudio y elaboración de estrategias que permitan tener un cierto grado de protección de la información, ya que hoy en día es el activo más importante que existe a nivel personal o empresarial.

La seguridad en una red depende de las vulnerabilidades que tengan los equipos que la conforman tanto a nivel de software como de hardware y son las que van a definir el tipo de ataque a la cual van estar expuestos. Pero también, son la guía para la elaboración de unas políticas de seguridad de forma adecuada y que cumpla con las necesidades específicas que tenga la red.

De igual forma como existen herramientas y métodos para el ataque, también existen para la defensa, por esta razón, se deben implementar herramientas que permitan conocer las vulnerabilidades que hay en el software o en los servicios de red con el fin de implementar medidas que conlleven a mitigar cualquier explotación de alguna de esas vulnerabilidades. Estas herramientas que sirven como mecanismo para la defensa en las redes pueden ser a nivel de hardware como el caso de un firewall, o a nivel de software como los sistemas de detección de intrusos, o incluso una combinación de los dos como es el caso de los sistemas de trampas.

La combinación de la tecnología de análisis de tráfico de red junto con un sistema de detección de intrusos y un sistema de trampas, proporciona al administrador de la red una visión detallada y realista de la vulnerabilidad existente en la red y de los tipos de ataques a lo que está expuesto, así como de las medidas adecuadas que debe implementar con el fin de establecer unas políticas de seguridad bien definidas y apropiadas para mitigar el daño que pueda sufrir tanto en su infraestructura lógica como física.

Por esta razón, este proyecto tiene el propósito de plantear un diseño preliminar de un sistema de detección de intrusos tipo *Snort* y un sistema de trampas de baja interacción tipo *Honeypots* sobre la red del grupo de investigación Geomática. Este diseño se hará con base en la topología existente en esta red, debido a que el grupo no cuenta con la autonomía ni el control sobre las políticas de diseño e implementación de la red que se lleva a cabo por parte de la división de servicios de información (DSI) de la Universidad.

1 MARCO TEORICO

Hoy en día el activo más importante de una organización es su información, por esta razón, se hace necesario implementar todos los mecanismos necesarios para proteger y salvaguardar todos los elementos que conforman la infraestructura informática de la empresa con el fin de garantizar su confidencialidad, integridad y disponibilidad. Por tal razón, el objetivo principal de la implementación de un sistema de seguridad, es tener un alto nivel de protección en su entorno lógico y físico, mediante planes, acciones y reglas que permitan contrarrestar la intrusión de cualquier atacante o analizar su modo de actuar.

1.1 ATAQUES INFORMÁTICOS

En la red existe una gran variedad de ataques que desde una simple intrusión a un sistema hasta gusanos informáticos que se encargan de robar información confidencial o incluso destruir el sistema. Dentro de los tipos de ataques que más se presentan se pueden clasificar en dos grandes categorías: los ataques de red y los de autentificación.

1.1.1 Ataques de red

Un ataque en una red se realiza con el propósito de encontrar vulnerabilidades y huecos de seguridad en los servicios y protocolos de un sistema con el fin de ejecutar cualquier acción que permita interrumpir o destruir el funcionamiento normal de estos servicios. Los ataques de red más comunes que se presentan son: el acceso no autorizado al sistema, el robo de información confidencial, la anulación de un servicio, la intrusión de un sistema, entre otros.

Los elementos que generalmente son más vulnerados en una red son: los enlaces de red, los servidores web, el DNS y la mayoría de protocolos de la familia TCP. Los ataques de red más comunes que se presentan son [1]:

- **Denegación de servicio (DOS *del inglés Denial Of Service*):** Este ataque es generalmente uno de los que más se usan, ya que compromete uno de los pilares fundamentales de la información, que es la disponibilidad de la misma. El objetivo de este ataque se basa en sobrecargar los límites de recursos que se establecen para un servicio determinado lo cual provoca la pérdida de la conectividad de la red por el consumo del ancho de banda. Esta interrupción

dificulta el acceso a los usuarios y su vez presenta una degradación de la calidad del servicio. Los tipos de ataques para esta vulnerabilidad son los de tipo TCP SYN Flood o TCP half-open.

- **Escaneo de Puertos (Scanning):** Este tipo de ataque realiza una búsqueda del estado de los puertos de una red o un equipo en particular, su intención es indagar qué puertos se encuentran abiertos, cerrados o protegidos por un firewall y cuáles tipos de servicios se encuentran activos en la red escaneada con el fin de encontrar alguna vulnerabilidad en los puertos que usan estos servicios.

- **Sniffing:** Esta técnica consiste en "escuchar" todo lo que pasa por una red con el fin de capturar o robar los datos que pasan por la misma. Esta práctica se realiza mediante un analizador de paquetes que se encarga de poner una o varias tarjetas de red en modo "promiscuo", es decir, la tarjeta captura, interpreta y almacena todos los paquetes que se desplazan por la red para su posterior análisis por parte del atacante, como por ejemplo, la extracción de contraseñas, mensajes de correo, datos bancarios, entre otros.

1.1.2 Ataques de autentificación

Estos ataques tienen como objetivo adquirir información de manera investigativa o maliciosa. La técnica que se utiliza para desarrollar este tipo de ataque es la suplantación de identidad o comúnmente denominado *Spoofing*. Esta suplantación se puede llevar a cabo en diferentes niveles: en la dirección IP, en el proveedor de servicios o DNS, en las tablas de enrutamiento o ARP y en el protocolo de correos SMTP. Los ataques de autentificación más comunes son [1]:

- **IP Spoofing:** La suplantación de la dirección IP es la variante más utilizada en este tipo de ataque. La técnica consiste en falsificar una dirección IP en cualquier protocolo TCP/IP bien sea ICMP, UDP o TCP de alguna máquina en la red, este enmascarado se realiza para que el equipo sea válido en la red y poder ejecutar el ataque con el fin de alterar, redireccionar o eliminar datos. Para llevar a cabo este tipo de ataque se requiere saber de antemano la dirección IP de la máquina que se va a tomar con el fin de suplantarla y de esta manera poder ingresar a la máquina víctima que generalmente es el servidor donde se aloja toda la información de la organización.

- **DNS Spoofing:** Este método es utilizado para falsificar la relación "nombre de dominio – IP" cuando se hace una consulta en el navegador por el nombre del

dominio, es decir, se alteran las direcciones IP de los servidores DNS de la víctima para que se redireccionen a servidores maliciosos. Un ejemplo sobre este tipo de ataques son las réplicas de una página web o sitios falsos, que generalmente son los portales web de las entidades bancarias, organizaciones gubernamentales, entre otros.

- **ARP Spoofing:** Esta técnica se usa para analizar los paquetes que circulan por una red con el fin de obtener información relevante para el atacante. Este tipo de ataques se realiza mediante el "envenenamiento" de tablas ARP lo cual permite falsificarla con el fin de obtener información como claves, mensajes de correo electrónico, nombres de usuarios, entre otros.

- **Mail Spoofing:** Este tipo de suplantación se realiza utilizando el protocolo SMTP que usa el puerto 25 de la familia TCP, el cual consiste en falsificación de correo electrónico a través otro correo electrónico, a este tipo de suplantación también se le conoce como SPAM.

1.2 SISTEMAS DE DEFENSA DE ATAQUES INFORMÁTICOS

1.2.1 *Snort*

Es un sistema multiplataforma diseñado para capturar el tráfico que circula a través de una red con el fin de comparar el contenido de esos paquetes contra una serie de reglas preestablecidas y así poder identificar si existe alguna amenaza o código malicioso en ellos [2].

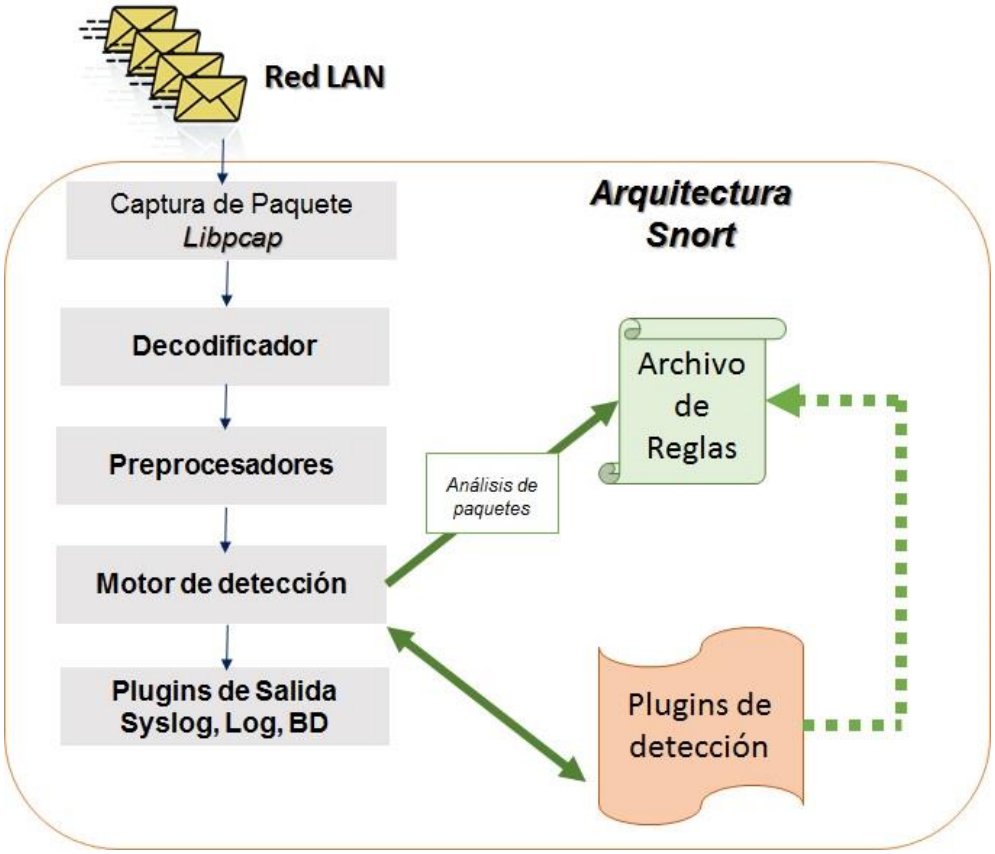
Los elementos que conforman el esquema básico del *Snort* son [3]:

- Módulo de captura del tráfico.
- Decodificador de paquetes.
- Preprocesadores.
- Motor de detección.
- Archivo de reglas.
- Sistema de alertas e informes.

En la figura 1 se muestra el esquema general del *Snort* junto con su estructura, en donde el primer componente se encarga de la captura del tráfico de la red, luego esta información es procesada a través de los módulos de decodificación y

preprocesadores, después realiza un chequeo a través del motor de detección y el archivo de reglas, para que finalmente genere unas alertas e informes.

Figura 1 Arquitectura Snort



Fuente: Basada [3]

La descripción de cada uno de los módulos junto con el proceso de instalación y configuración se encuentran en el Anexo A.

1.2.2 *Honeypots*

El fundador de esta tecnología Lance Spitzner define al *Honeypots* como un *recurso* que se usa para ser atacado, detectar el ataque e incluso desviarlo. El *Honeypots* no se usa para resolver algún problema de vulnerabilidad en la red, sino por el contrario, se utiliza para atraer a los atacantes ya que su función principal es la de recolectar toda la información posible sobre el ataque.

El *Honeypots* es un sistema que permite llevar a cabo la simulación de algún equipo o servicio informático cuyo objetivo principal es servir como señuelo para atraer a los atacantes con el fin de recolectar información valiosa acerca de las técnicas y herramientas que éstos utilizan, para luego ser analizadas y mejorar las reglas de los sistemas de detección y protección que existen en la red [4]. De igual manera como se simula un solo equipo, también se puede simular toda una red de computadoras con diferentes sistemas operativos cada uno con el fin de brindar un entorno más similar al que existe en la red actual, a este tipo de sistema se le denomina *Honeynet*.

- **Clasificación de *Honeypotss***

De acuerdo con el entorno donde se realiza la implementación del sistema, los *Honeypotss* se pueden clasificar en dos tipos: de producción y de investigación; mientras que, conforme al tipo de interacción que existe con el atacante, se pueden clasificar como *Honeypots* de baja y alta interacción [5], el detalle de cada uno de estos tipos de *Honeypots* se describen en el Anexo B.

- **Ventajas y desventajas de los *Honeypots***

Los *Honeypots* por su condición de herramientas de detección pasiva tienen una serie de ventajas y desventajas que se detallan en la Tabla 1:

Tabla 1. Ventajas y desventajas de los Honeypots

Ventajas	Desventajas
• Su implementación no requiere de grandes recursos del sistema como procesador, RAM y tarjeta de red. • Una de las grandes ventajas es que no generan grandes volúmenes de información. • Se les puede programar para reaccionar cuando presentan ataques. • Son muy utilizados para la práctica de computación forense, debido a que se basa en analizar el modus operandi del posible atacante para poder reconstruir su perfil.	• Para corregir o eliminar malas configuraciones. • Si se cuenta con una red vulnerable, al implementar un *Honeypots* este no solucionará las fallas que se están presentando. • Son muy pasivos, debido a que si no son atacados no arrojan ningún tipo de alerta.

La descripción de cada uno de los tipos de *Honeypots* y el proceso de instalación y configuración se encuentran detallados en el Anexo B

2 ANALISIS DE LA RED

El primer paso que se hace antes de diseñar y hacer la respectiva validación del diseño del *Snort* y el *Honeypots*, es realizar un estudio sobre la topología de red existente en el grupo de investigación Geomática. Para esta actividad se contó con el apoyo de la documentación existente por parte de la División de Servicios de Información (DSI), acerca de los diseños de la topología de red interna del grupo y su posterior comprobación en los *Racks* del Edificio de Laboratorios Pesados de la UIS, acompañados por profesionales del grupo que están encargados del funcionamiento de estos equipos.

El estudio acerca de la topología lógica y física de la red del grupo se hizo para llevar a cabo las siguientes actividades:

- Registrar la forma en que está diseñada la red, es decir, conocer la topología lógica y física y los recursos tecnológicos que la componen.
- Conocer la cantidad y distribución de las subredes dentro de la red del grupo, con el fin de saber el rango de direcciones IP que gestionan y la carga de tráfico que opera en cada una de ellas.
- Conocer de primera mano la infraestructura física de los equipos que conforman la red, como por ejemplo, los Racks de comunicaciones, los *switches*, el tendido de fibra óptica que comunica directamente con el CENTIC, el *Data Center* en donde se encuentran todos los servidores y los equipos de cómputo e impresoras.
- Documentar la forma en cómo se encuentra distribuidas las salas de cómputo con el fin de analizar la cantidad de tráfico de red que consume cada una de ellas y los recursos informáticos que se utilizan.

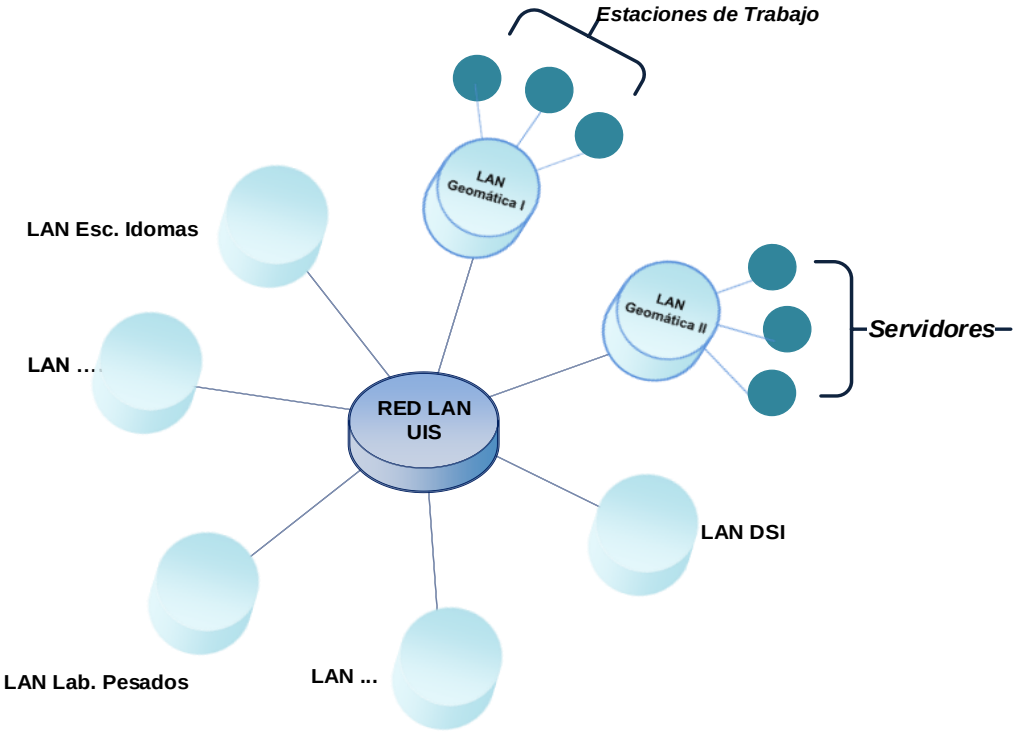
En las siguientes secciones se hace una descripción detallada de la topología lógica y física de la red que existe actualmente en el grupo Geomática, estos diseños fueron facilitados por el DSI dentro del marco de colaboración para realización de esta monografía.

2.1 TOPOLOGÍA LÓGICA DE LA RED DE GEOMÁTICA

En la figura 2 se muestra la topología tipo Estrella que está implementada para la red de datos de la Universidad, de igual manera, se conserva este tipo de topología para todas las subredes internas que se encuentran en cada uno de los Racks ubicados en los edificios de esta institución. Esta distribución de la red en forma de estrella tiene la ventaja de que cada nodo tiene una independencia física dentro del *switch*, es decir, si algún nodo presenta problemas o fallos no afectará el normal funcionamiento de la red, ni de los demás nodos conectado a ella.

A partir de esta gráfica se puede deducir que existen dos subredes para el grupo Geomática, una de ellas es donde se encuentran todas las estaciones de trabajo o también llamada red de producción y la otra es donde se encuentran interconectados todos los servidores del grupo.

Figura 2 Topología Lógica de la red de Geomática dentro de la red de la UIS

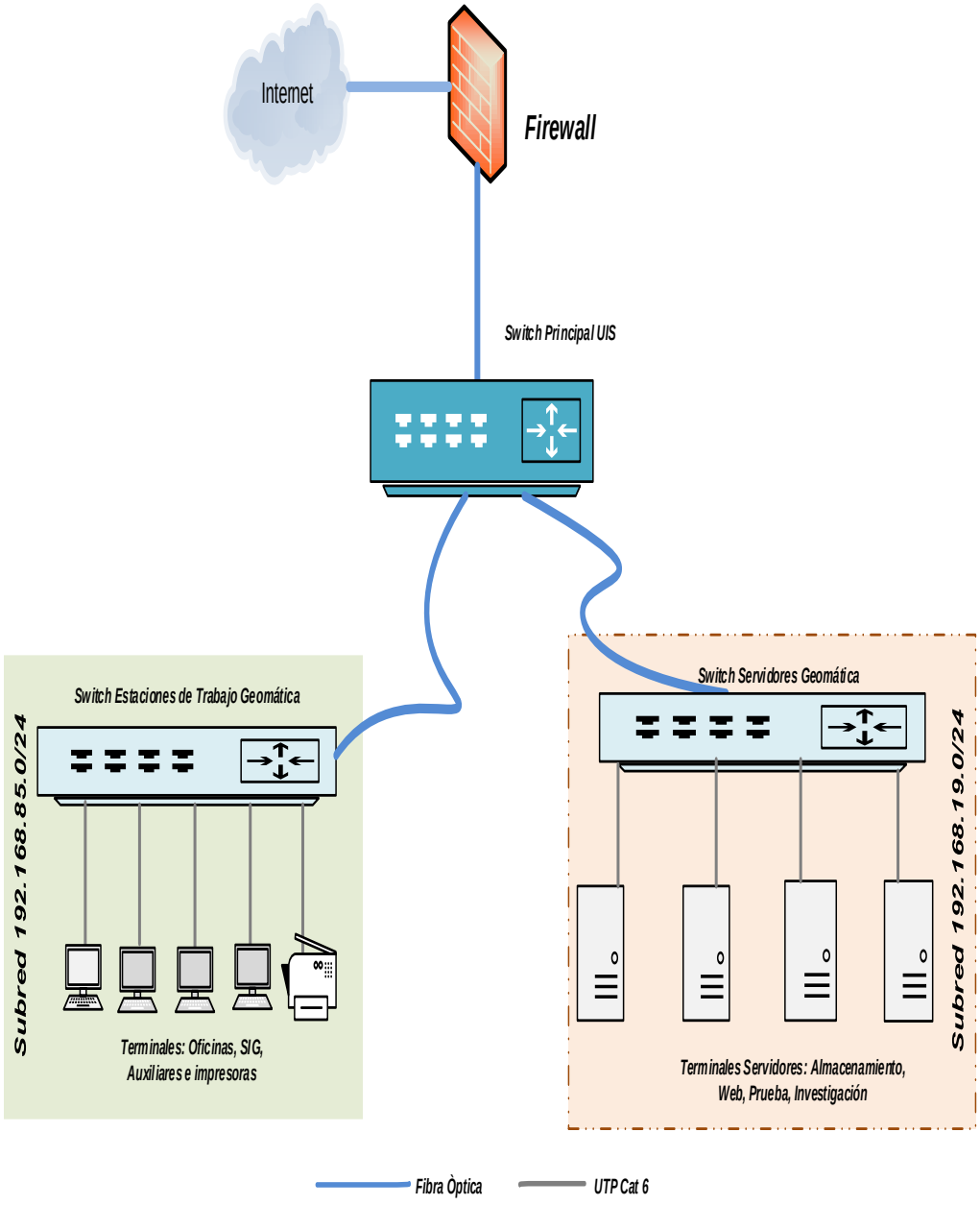


Fuente: Autor

2.2 TOPOLOGÍA FISICA DE LA RED DE GEOMÁTICA

La estructura física de la red interna de datos del grupo Geomática está compuesta por principalmente dos dispositivos activos de capa dos o de enlace, que para este caso son los *switches* que aparecen en la figura 3 y que están conectados directamente con el *backbone* principal del CENTIC mediante un tendido de fibra óptica multimodo. La función de cada uno de estos *switches* es la de administrar los equipos de cada subred por separado manteniendo la misma topología de tipo estrella que está implementada en la Universidad.

Figura 3 Topología Física de la red de Geomática

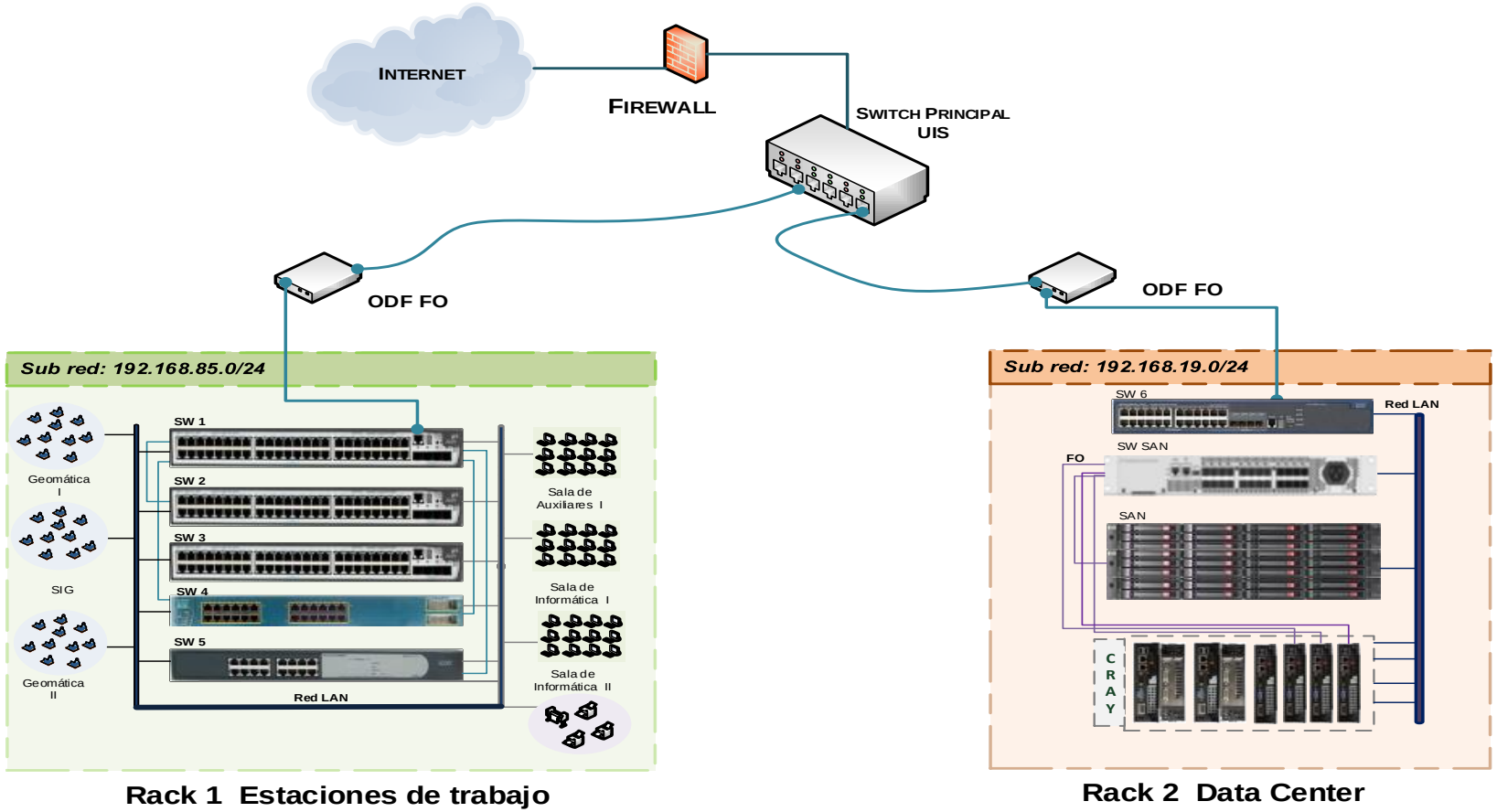


Fuente: Autor

En el lado izquierdo de la figura anterior se muestra la subred 192.168.85.0/24, que está compuesta por los equipos de cómputo de las Salas de Informática I y II, las salas de auxiliares, el laboratorio de SIG, las salas de investigación Geomática

I y II y las impresoras. Esta subred es la llamada red de producción del grupo, porque es aquí donde se llevan a cabo todos los proyectos de investigación y extensión, por esta razón, existen recursos compartidos en muchos de estos equipos con el fin de transferir información técnica y administrativa entre ellos y que en algunos casos consumen mucho espacio en disco, como es el caso de la cartografía Digital. Mientras que en el lado derecho de esta figura se visualiza la subred 192.168.19.0/24, que es donde se encuentran interconectados todos los servidores del grupo conformados por: el servidor web, el de aplicaciones y la unidad de almacenamiento SAN. En la figura 4 muestra la misma topología física descrita anteriormente, pero en este caso se hace un énfasis más real y detallado de cada uno de los componentes que la conforman

Figura 4 Diagrama de los componentes físicos de la red del grupo Geomática



Rack 1 Estaciones de trabajo

Rack 2 Data Center

Fuente: Autor

2.2.1 *Rack* N° 1: *Rack* de Estaciones de Trabajo

En el *rack* de comunicaciones que se encuentra en la parte izquierda de la figura 4 se encuentran conectados todos los equipos de las salas de investigación, laboratorios de informática, dispositivos inalámbricos e impresoras del grupo.

La descripción más detallada de cada uno de los componentes del rack de comunicaciones N° 1 es:

- **Un Cortafuegos (*Firewall*)**: La barrera de seguridad utilizada para proteger la red de posibles ataques, malware o intrusiones desde el exterior, es de marca ***SonicWALL Dynamic Dell***.

- **Un Módulo de Fibra Extremo del CENTIC:** Conexión con módulo *Extreme Networks 1000BASE-SX* Mini *GBic*, conector *LC*, fibra multimodo de 50/125 micras.

- **Un Módulo de Fibra *Extremo Rack* Geomática:** módulo *3COM 1000BASE-SX SFP Transceiver*, conector *LC* de fibra multimodo de 50/125 micras.

- **Una Referencia SW1: *switch* 48 puertos 10/100/1000 *POE*:** *switch* 3Com 4800G POE (*Power Over Ethernet*) de 48 Puertos 10/10/1000.

- **Referencias SW2 y SW3**: dos *switch* 3COM 4800G de 48 Puertos 10/100/1000.

- **Una Referencia SW4**: Un *switch* marca *Catalyst* 3550 de 24 puertos 10/100.

- **Una Referencia SW5**: swicht 3Com *Baseline* 2024 de 24 10BASE-T/,100BASE-TX y configuración de puertos auto detección *MDI/MDIX*

- **Módulos conexión en cascada con cada *switch 4800G*:** tres (3) módulos para la conexión en cascada de los *switch* 3COM 4800G de conexión local CX4.

Cada uno de estos elementos activos (switch capa 2) realizan conexión a cada una de las terminales de los puntos de red por medio de cable categoría 6 con conector RJ 45 y bajo el protocolo TCP/IP v4 en las oficinas del grupo, salas de investigación y salas de informática. En la Tabla 2 se describen las características de cada sala y la cantidad total de los puestos de trabajo que integran toda la red.

Tabla 2. Descripción de las Salas de Cómputo del grupo Geomática

Ubicación	Descripción
Sala Geomática 1	Esta sala está conformada por profesionales en áreas de Cartografía, Ambiental, Pavimentos, Hidráulica, Geotecnia y coordinación técnica e investigativa, en total existen doce (12) puestos de trabajo
Sala SIG	Esta sala tiene la función de servir de apoyo en la parte cartográfica y modelado estructural para el desarrollo de los proyectos de extensión del grupo de Geomática en el área de la Ingeniería Civil. Generalmente, la conforman estudiantes en la modalidad de auxiliares de la escuela de civil y está compuesta por doce (12) puestos de trabajo de alto desempeño gráfico.
Sala de Auxiliares	Esta área ocupada por estudiantes de Ingeniería Civil, Sistemas y Electrónica, en la modalidad de proyectos de grado y auxiliares de apoyo para los diferentes proyectos de investigación y extensión que el grupo de Geomática. Está compuesta por veintidós (22) puestos de trabajo.
Salas de Informática I y II	Son salas dedicadas a la academia en donde se imparten clases de pregrado, diplomados y postgrados para estudiantes tanto de ingeniería civil como de otras carreras. Cada uno de las salas cuenta con treinta y dos (32) máquinas.

2.2.2 Rack N°2: Data Center Geomática

El *rack* del *Data Center* de Geomática brinda conectividad a un conjunto de servidores de tipo *Blade* y a un almacenamiento tipo SAN como se visualiza en la tabla 3.

La conexión de este rack con la red de la Universidad hace a través de un tendido de fibra óptica de 12 hilos multimodo de 50/125 micras de 300 metros de recorrido, el cual va directamente desde el Backbone del CENTIC

al *rack* del Data Center de Geomática. Su conexión con el *switch* del Core del Centic se realiza por medio de un módulo *GBIC* 1000SX, de igual manera que con la bandeja de fibra que está el *Rack* del *Data Center* de Geomática.

El SW6 que aparece en la figura 4, hace referencia a un switch de 48 puertos 10/100/1000. Este elemento brinda interconexión por cada uno de sus puertos a las diferentes subredes que tienen asignadas los servidores del grupo. La descripción de cada uno de los equipos que conforman el Rack No.2 se representa en la tabla 3.

Tabla 3 Descripción Equipos que están conectados al Rack 2

Equipo	**Descripción Switch SAN**
HP StorageWorks 8/8 SAN Switch	Equipo conmutador que permite el acceso a la *SAN*, Lo cual ayuda a mejorar el rendimiento y la gestión de la infraestructura del almacenamiento. Actualmente tiene 8 puertos habilitados los cuales están conectados a los servidores que tienen módulo SFP para conexión con *Fiberchannel*. El *Switch* es administrado vía LAN por IP privada y conectado a un punto el SW6 con cable Ethernet categoría 6.
Equipo	**Descripción SAN**
HP MS P2000G3 FC/iSCSI y expansión D2700	Es una tecnología basada en red de almacenamiento que permite crear diferentes volúmenes de discos para ser presentados a los recursos necesarios y así poder utilizarlo por medio de la red. Actualmente se tiene creados tres discos virtuales con arreglos en RAID 5 y 6, los cuales están divididos en volúmenes con el fin de poder ser accedido a través de la red.
Servidores	**Descripción Nodos blades**
CRAY	
1. Nodo de Procesamiento en Paralelo	Blade asignado al procesamiento en paralelo, para el desarrollo de los proyectos en nubes de puntos, renderizado de elementos tridimensionales en Revit, tratamiento de imágenes. El sistema operativo instalado en este servidor es Red Hat.
2. Nodo Web	En este servidor se alojan las páginas y servicios web de todos los proyectos de Sistema de Información Geográfica que se desarrollan en Geomática. También en este servidor se encuentra instalado el servidor de mapas de ArcGIS – ArcGIS Server, para proveer los geo-servicios para los proyectos SIG

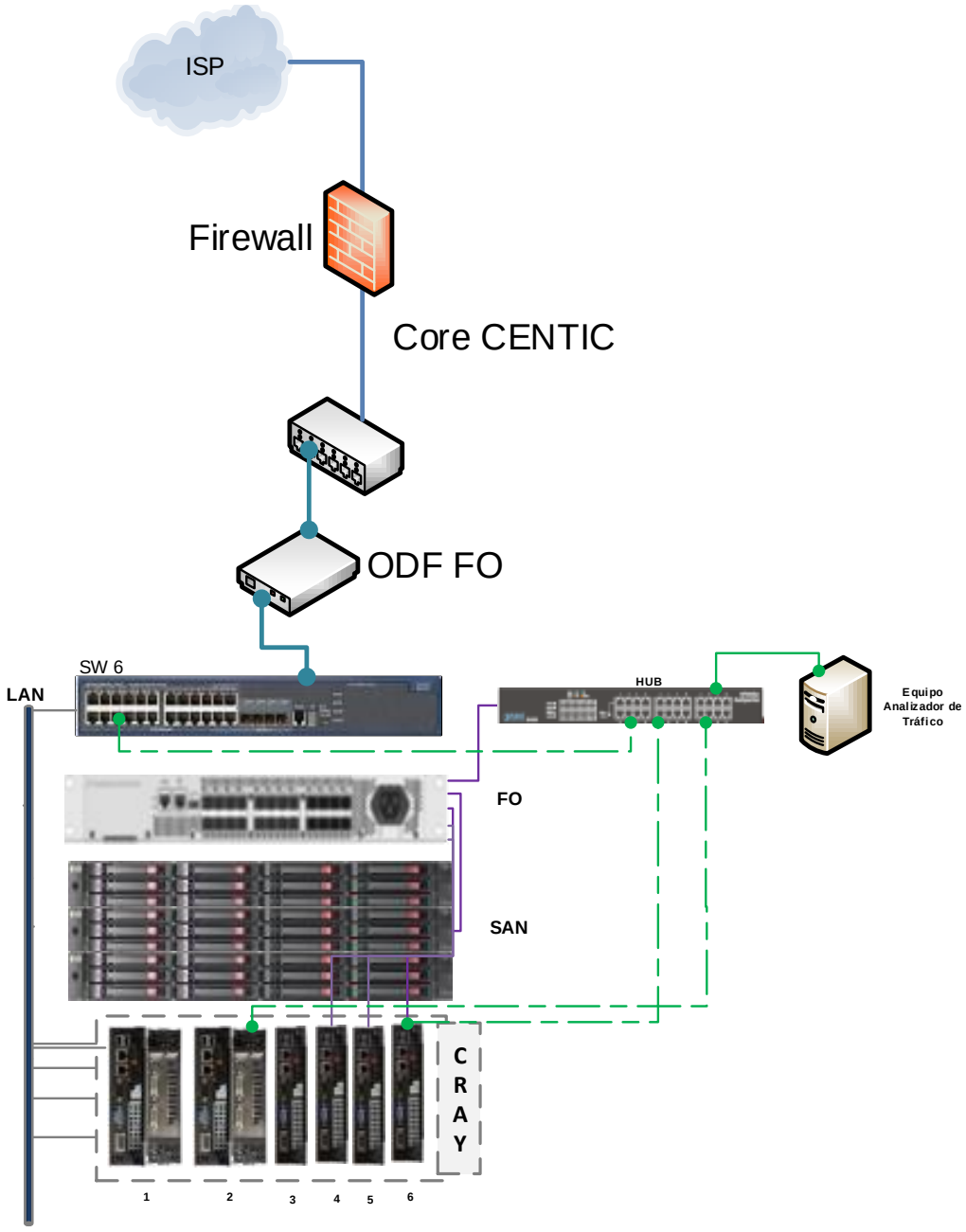
	desarrollados en el grupo. El sistema operativo instalado en este servidor es Windows Server 2008R2 y está destinado para trabajar exclusivamente en la red interna y no tiene salida al exterior.
3. Nodo Cloud 00	Son tres Blades asignados al proyecto Cloud Computing. Son nodos de virtualización que soportan la infraestructura como servicio (IaaS), El Nodo 00 (Doble conexión Fiberchannel) es el maestro, tiene las herramientas administrativas y el acceso a los discos de la SAN, los otro nodos son esclavos y en ellos se pueden levantar y migrar las máquinas virtuales en ejecución.
4. NodoCloud 2	
5. Nodo Oca	
6. Nodo Almacenamiento	El objetivo de este servidor es la administración del Directorio Activo del grupo, esto es, el manejo de las cuentas de usuarios, los grupos de trabajo, los archivos e impresoras compartidos, etc. Otra función de igual importancia es la administración para el control de licencias de los programas de AutoCAD, ArcGIS y Matlab, que se encuentran instalados en las salas de cómputo y en los equipos del grupo Geomática. Otra función importante es la de servir como puente de comunicación con la SAN para el almacenamiento de la información. El sistema operativo instalado en este nodo es Windows Server 2008R2 y la conexión con la SAN se hace a través de Fiber Channel.

2.3 ANÁLISIS DE TRÁFICO EN LA SUBRED DE SERVIDORES

El análisis de tráfico es realizado mediante sondas conectadas a un dispositivo activo de la red en modo promiscuo para visualizar los protocolos y comportamientos que está teniendo un sistema en especial. El objetivo de esta actividad se centra en la subred donde se encuentran los servidores del grupo Geomática.

Para llevar a cabo este análisis se utilizó la herramienta Ntop que permite hacer un análisis en tiempo real de una red, independientemente de los sistemas operativos que existan en los equipos que la conforman. Una vez definida la herramienta con la cual se va a ejecutar esta actividad, se configuró la infraestructura de red que tiene el Rack de servidores con el fin de compartir el medio de transmisión, es decir, se ubica un equipo con el Ntop instalado entre el Switch y el segmento de red de los servidores, tal como se muestra en la figura 5.

Figura 5 Ubicación del equipo para monitorizar el tráfico en la subred de servidores

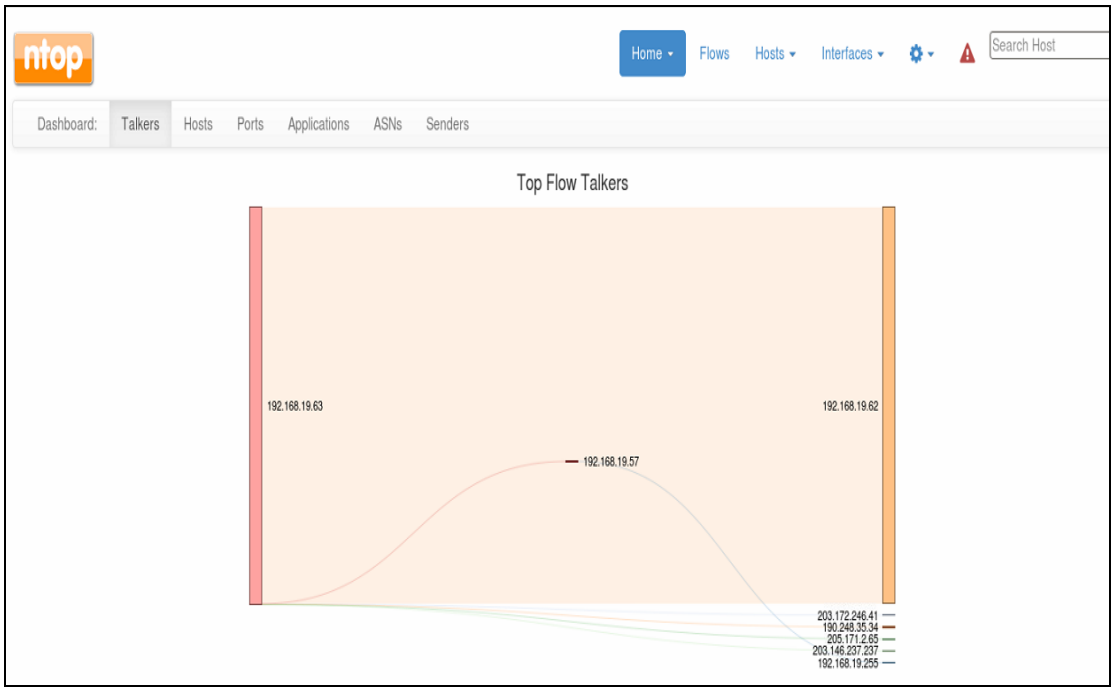


Fuente: Autor

2.3.1 Interfaz gráfica del análisis de tráfico

Esta herramienta de análisis tiene la característica de poder visualizar los resultados en tiempo real en internet a través de cualquier navegador web dentro de la red de la Universidad, que para este caso la dirección para acceder a la interfaz gráfica es 192.168.19.63:3000, en donde después de autenticarse se ingresa al entorno gráfico, en figura 6 se muestra el análisis que se está haciendo a las direcciones IP de cada uno de los servidores del grupo, sin embargo, esta herramienta también tiene la capacidad de hacer análisis sobre el tráfico que tiene un equipo en particular.

Figura 6 Interfaz gráfica del analizador de tráfico Ntop



Fuente: Herramienta Ntop

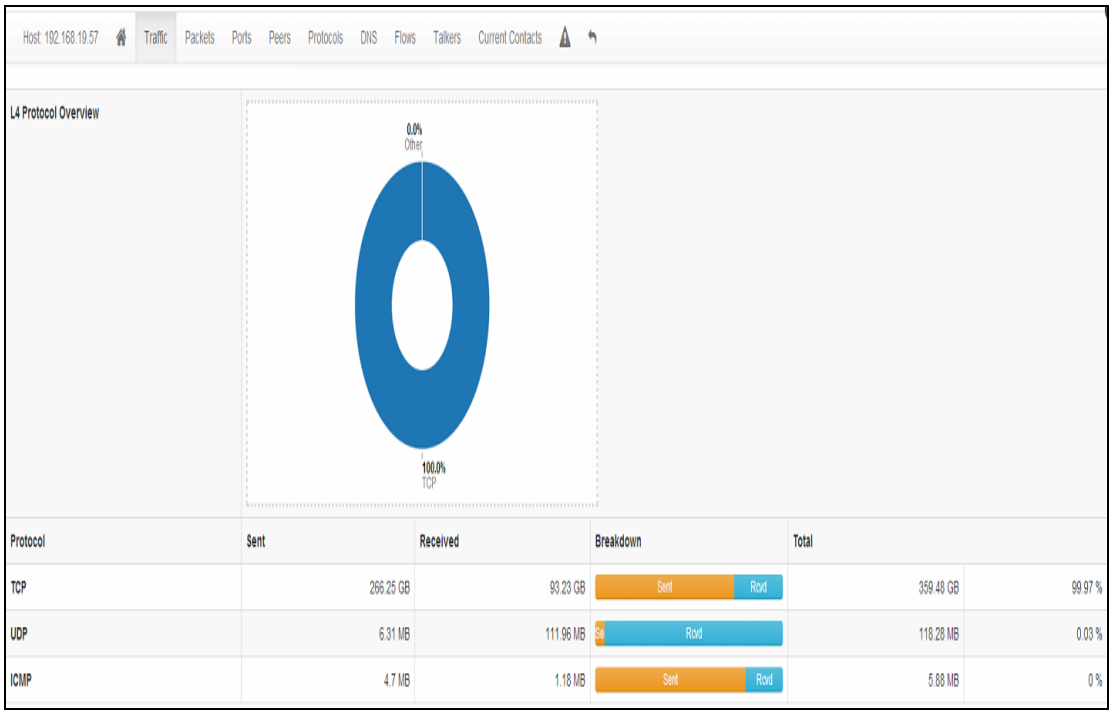
2.3.2 Resultados del análisis de tráfico en los servidores

Para evaluar los resultados del tráfico que existen en cada uno de los servidores, se realizó una monitorización durante veinticuatro (24) horas al

día durante ocho (8) días continuos. El resultado por cada uno de los servidores es el siguiente:

Servidor de aplicaciones y almacenamiento: este servidor tiene asignada la dirección IP 192.168.19.57 y es el servidor donde se encuentra instaladas todas las aplicaciones que tienen licencia educativa tipo servidor, es decir, desde este servidor se administran todas las licencias que se suministran a los equipos de la subred de producción. En la figura 7 se visualiza una estadística filtrada por tres principales puertos más usados en este servidor.

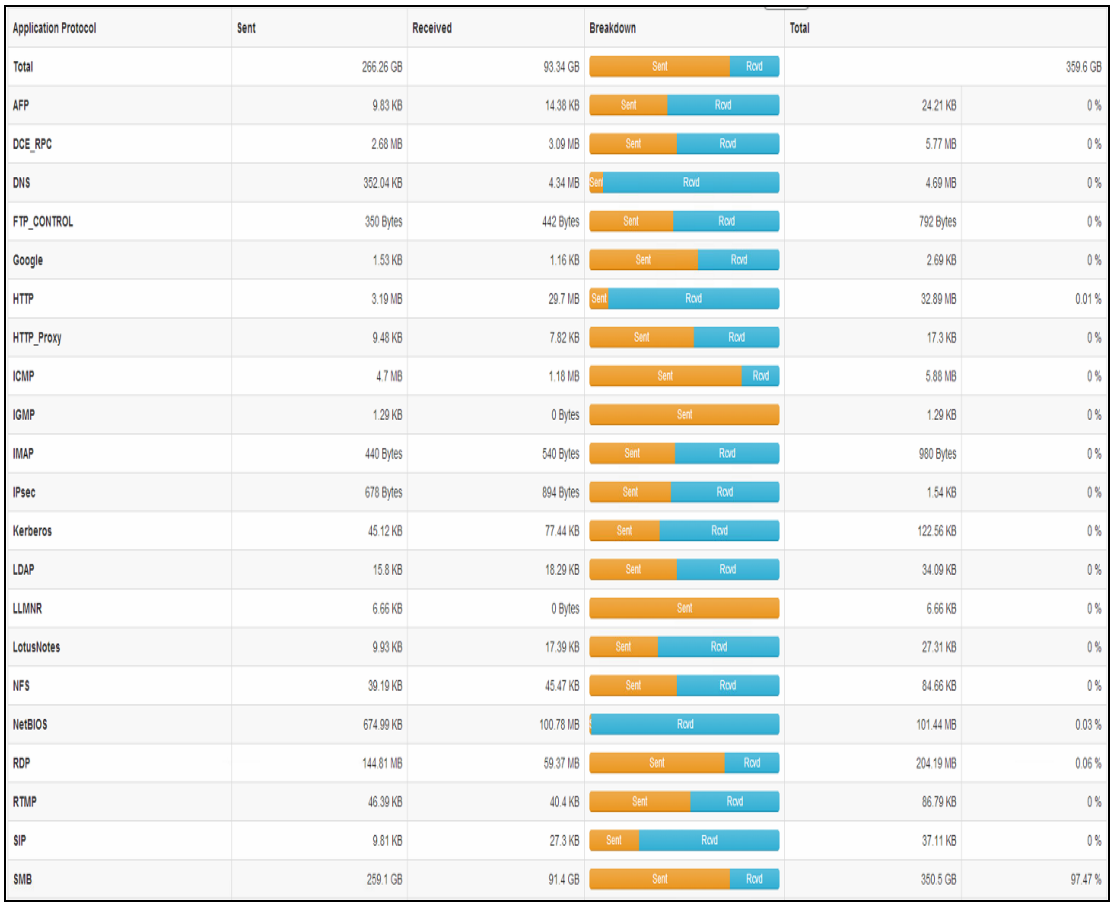
Figura 7 Estadística de los tres primeros protocolos que más presentan tráfico en el servidor de Aplicaciones



Fuente: Herramienta Ntop

Un resultado más detallado del análisis del tráfico de esta red se muestra en la figura 8.

Figura 8 Detalle de los protocolos que se usan en el servidor de Aplicaciones



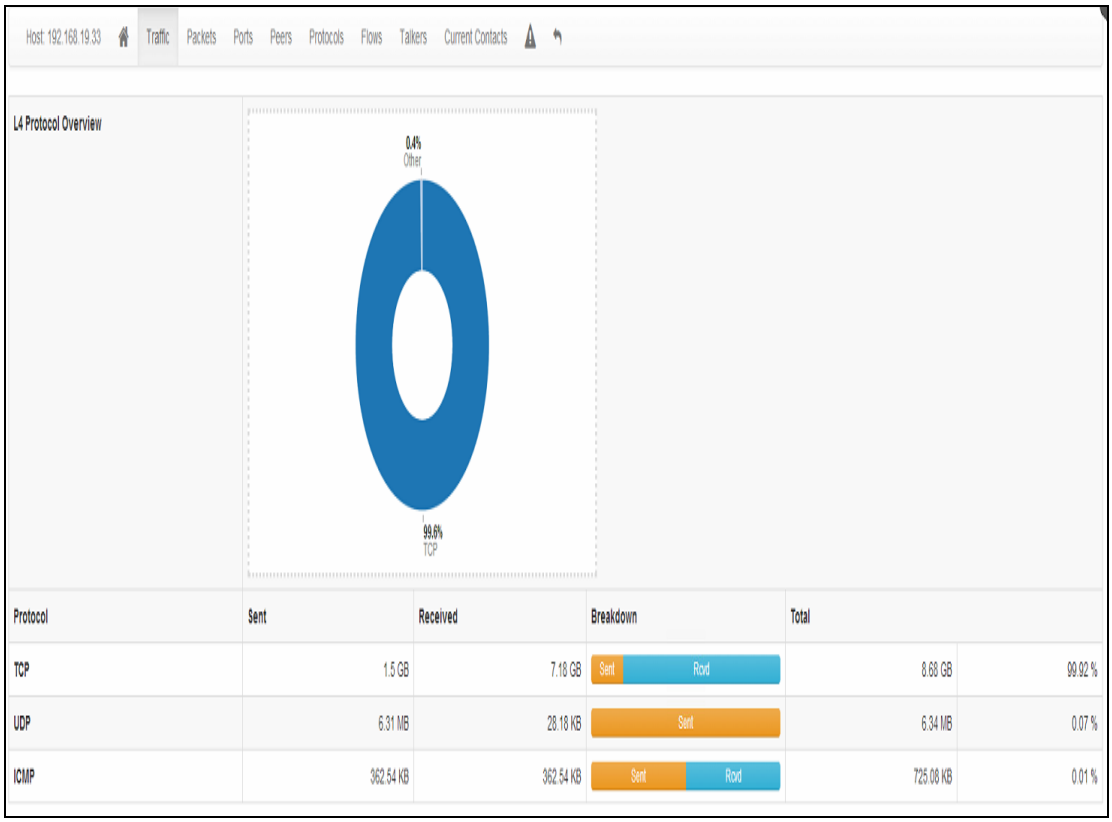
Application Protocol	Sent	Received	Breakdown	Total	
Total	266.26 GB	93.34 GB	Sent Rcvd	359.6 GB	
AFP	9.83 KB	14.38 KB	Sent Rcvd	24.21 KB	0 %
DCE_RPC	2.68 MB	3.09 MB	Sent Rcvd	5.77 MB	0 %
DNS	352.04 KB	4.34 MB	Sent Rcvd	4.69 MB	0 %
FTP_CONTROL	350 Bytes	442 Bytes	Sent Rcvd	792 Bytes	0 %
Google	1.53 KB	1.16 KB	Sent Rcvd	2.69 KB	0 %
HTTP	3.19 MB	29.7 MB	Sent Rcvd	32.89 MB	0.01 %
HTTP_Proxy	9.48 KB	7.82 KB	Sent Rcvd	17.3 KB	0 %
ICMP	4.7 MB	1.18 MB	Sent Rcvd	5.88 MB	0 %
IGMP	1.29 KB	0 Bytes	Sent	1.29 KB	0 %
IMAP	440 Bytes	540 Bytes	Sent Rcvd	980 Bytes	0 %
IPsec	678 Bytes	894 Bytes	Sent Rcvd	1.54 KB	0 %
Kerberos	45.12 KB	77.44 KB	Sent Rcvd	122.56 KB	0 %
LDAP	15.8 KB	18.29 KB	Sent Rcvd	34.09 KB	0 %
LLMNR	6.66 KB	0 Bytes	Sent	6.66 KB	0 %
LotusNotes	9.93 KB	17.39 KB	Sent Rcvd	27.31 KB	0 %
NFS	39.19 KB	45.47 KB	Sent Rcvd	84.66 KB	0 %
NetBIOS	674.99 KB	100.78 MB	Rcvd	101.44 MB	0.03 %
RDP	144.81 MB	59.37 MB	Sent Rcvd	204.19 MB	0.06 %
RTMP	46.39 KB	40.4 KB	Sent Rcvd	86.79 KB	0 %
SIP	9.81 KB	27.3 KB	Sent Rcvd	37.11 KB	0 %
SMB	259.1 GB	91.4 GB	Sent Rcvd	350.5 GB	97.47 %

Fuente: Herramienta Ntop

Con esta información recolectada por el *Ntop*, se evidencia que los protocolos más usados en el servidor de aplicaciones en su orden son: *TCP, UDP e ICMP*, el cual confirma que la mayoría de las aplicaciones instaladas utilizan el protocolo *TCP* para validar la licencias, aunque cada una de ellas establece la comunicación por un puerto diferente. Por otra parte, en cuanto al protocolo UDP la mayoría son de tipo peticiones, ya que por cada equipo que ejecute una aplicación que está administrada en este servidor, le envía un mensaje en este protocolo. Por esta característica, el protocolo *UDP* es candidato para hacer una validación de reglas en el *Snort*. Finalmente, este servidor envía gran cantidad de mensajes en el protocolo *ICMP*, que es el encargado de enviar mensajes con la finalidad de diagnosticar o controlar los servicios que ofrece hacia los equipos. Este último protocolo también es un fuerte candidato a validar con la prueba piloto para el diseño del *Snort y Honeypots*.

- **Servidor Web:** este servidor tiene asignada la dirección IP pública 192.168.19.33 y su dominio es *garza.uis.edu.co* y como su nombre lo dice es el encargado de alojar todas las páginas web y los servicios web tanto espaciales como alfanuméricos de los diferentes productos que tiene el grupo Geomática. Como en el caso del servidor anterior, en la figura 9 también muestra las estadísticas filtradas por los tres protocolos más usados en este servidor.

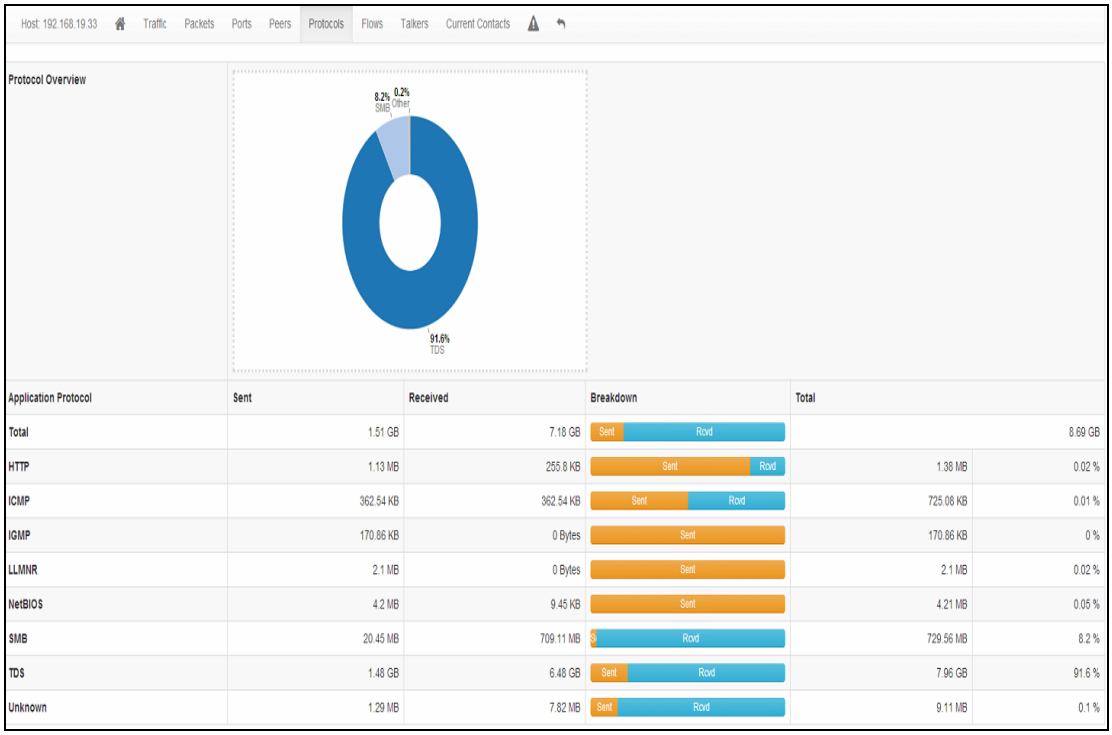
Figura 9 Estadística de los tres primeros protocolos que más presentan tráfico en el servidor Web



Fuente: Herramienta Ntop

El análisis más detallado de la actividad de cada uno de los protocolos de este servidor se visualiza en la figura 10.

Figura 10 Detalle de los protocolos que se usan en el servidor Web



Application Protocol	Sent	Received	Breakdown	Total	
Total	1.51 GB	7.18 GB	Sent / Rcvd		8.69 GB
HTTP	1.13 MB	255.8 KB	Sent / Rcvd	1.38 MB	0.02 %
ICMP	362.54 KB	362.54 KB	Sent / Rcvd	725.08 KB	0.01 %
IGMP	170.86 KB	0 Bytes	Sent	170.86 KB	0 %
LLMNR	2.1 MB	0 Bytes	Sent	2.1 MB	0.02 %
NetBIOS	4.2 MB	9.45 KB	Sent	4.21 MB	0.05 %
SMB	20.45 MB	709.11 MB	Rcvd	729.56 MB	8.2 %
TDS	1.48 GB	6.48 GB	Sent / Rcvd	7.96 GB	91.6 %
Unknown	1.29 MB	7.82 MB	Sent / Rcvd	9.11 MB	0.1 %

Fuente: Herramienta Ntop

Esta información suministrada a través de la interfaz del analizador presenta una gran actividad de peticiones en el protocolo *TCP* y de envío en el protocolo *HTTP*, es ocurre debido a la interacción con las páginas y servicios web, así como también los geoservicios que el grupo ofrece en cada uno de los proyectos que maneja. De igual manera, presenta gran actividad de petición en el protocolo *UDP*, por esta razón será tenido en cuenta para la prueba piloto de la validación del diseño.

3 REQUERIMIENTOS PARA *SNORT* Y *HONEYPOTS* DE BAJA INTERACCIÓN

3.1 REQUERIMIENTOS PARA EL *SNORT*

Para realizar el estudio acerca de los requerimientos del *Snort* de tipo *NIDS*, se toma como base la documentación realizada en el análisis de red con el fin de poder identificar la cantidad de tráfico que circula por la red, especialmente en la subred donde se encuentra la zona de servidores, es decir, la zona desmilitarizada, que en este caso corresponde a la familia de direcciones IP 192.168.19.0/24 de la UIS.

3.1.1 Requisitos de Hardware

El conocimiento del tráfico de la red va a determinar en parte la capacidad de procesamiento y de almacenamiento que debe tener el equipo en donde se va a instalar el *Snort*, principalmente si el *Snort* es de tipo NIDS, ya que éstos generan aproximadamente unos 170.000 paquetes por segundo en una red Ethernet de 100 Mbps, lo que hace que en muchos casos se pierdan paquetes con información relevante. De igual manera, consume muchos recursos en memoria RAM, debido a que debe mantener la información actualizada sobre el estado de cada una de las conexiones TCP que tiene abiertas.

Con base en el análisis de tráfico de red que se realizó en la subred en donde se encuentran los servidores, se recomienda un equipo con altos recursos de procesamiento y almacenamiento. Sin embargo, para efectos de llevar a cabo la validación del diseño del *Snort* se realizó la prueba piloto en un equipo de propiedad del grupo Geomática y sus especificaciones técnicas se muestran en la Tabla 4.

Tabla 4. Especificaciones técnicas del equipo donde se instaló el Snort

Workstation	**Especificaciones Técnicas**				
Equipo Base	Procesador	RAM	Disco Duro	Ethernet	Dirección IP
	CPU: Intel ® Core 2 Quad CPU 8400 @ 2.66 GHz	8 GB	180 GB	Intel® Gigabit Network Connection	192.168.19.63 (kiwi.uis.edu.co)

3.1.2 Requisitos de Software

La característica más importante del *Snort* de tipo NIDS es que se basa en una serie de programas escritos en C llamados plug-ins, los cuales están desarrollados bajo extensiones PCAP y que se usan para la captura de datos o para el sistema de salida e informes. Igualmente, el motor de detección hace uso del archivo de reglas para analizar el tráfico que circula por la red. Algunos requerimientos a nivel de software son:

- **Librería *PCAP***

Es una librería de código libre escrita en C y que es conocida como *Libpcap* para la versión de Linux/Unix y en la versión de Windows se llama W*inpcap*s. Esta librería es la encargada de capturar todos los paquetes que circulan por la red, además, contiene diferentes opciones como: captura de interfaces de red, configuración de la tarjeta de red en modo normal o promiscuo, rearme de paquetes fragmentados para obtener nuevamente el paquete original, filtrado de tráfico dependiendo del protocolo y manejo de archivos.

- **Librería *tcpdump***

Es un programa tipo consola de comandos que permite visualizar y analizar el contenido de los paquetes. También permite almacenar las tramas capturadas en un archivo para su posterior análisis. Usa la librería Libpcap para capturar tráfico no sólo de la red sino también de interfaces USB.

- **Librería de Adquisición de paquetes**

También llamada DAQ (del inglés Data Acquisition) es un paquete para interfaces I/O. Remplaza llamadas directas a funciones libpcap con una capa de abstracción que facilita las operaciones con diferentes interfaces de hardware y software.

- **Archivos de reglas**

Como se mencionó anteriormente, este archivo comprende el conjunto de reglas que usa el motor de detección del *Snort* para compararlas con los paquetes capturados y en caso de que exista alguna coincidencia, realiza alguna acción que también viene configurada en la regla que la detectó. Este archivo se puede descargar directamente desde el sitio oficial de *Snort*.

3.2 Requerimientos para el *Honeypots*

Este proyecto también pretende llevar a cabo el diseño y validación del *Honeypots* para la red del grupo Geomática, por lo tanto, se hace necesario hacer una prueba piloto en la red con el fin de verificar que el diseño esté cumpliendo con los objetivos planteados, para ello se necesita que el *Honeypots* cumpla con los siguientes requisitos:

3.2.1 Captura de los datos

Esta captura consiste en la monitorización y el registro de la actividad que arroje el Honeyd con respecto a los atacantes. Para efectos de la validación del diseño, se realiza un ejercicio de ataque a la red de tipo scanning con el fin de extraer y analizar cómo es la estructura de la información que el Honeyd captura.

3.2.2 Recolección y análisis de datos

Con respecto a la recolección de la información suministrada por el Honeyd se debe analizar el lugar en donde se almacena con el fin de que no vaya a quedar expuesta. El análisis de esta información se hace mediante el equipo en el cual se encuentra instalado el control de *Honeypots*.

3.2.3 Requisitos de Hardware

Con el fin de no representar un sobrecosto en la validación del diseño mediante una prueba piloto, se hace uso de las tecnologías de virtualización de sistemas, por esta razón, se lleva a cabo la instalación del Honeyd sobre una máquina virtual que a su vez está montada sobre un equipo real de la red. Por lo tanto, el requerimiento más importante que debe tener este equipo real es que tenga suficiente capacidad de memoria RAM, debido a que el software de virtualización monta las máquinas virtuales en la memoria del equipo. Igualmente, se necesita que el computador posea una tarjeta de red activa que permita una buena capacidad de flujo de información.

Para el caso de este proyecto se hace uso de un equipo que se encuentra en la red del grupo, en la Tabla 5 se detallan las especificaciones técnicas de este equipo.

Tabla 5. Especificaciones técnicas del equipo donde se instaló el Honeypots Virtual

Workstation	Especificaciones Técnicas				
Equipo Base	Procesador	RAM	Disco Duro	Ethernet	Dirección IP
	CPU: Intel ® Xeon @ 3.07 GHz	16 GB	300 GB	Intel® Gigabit Network Connection	192.168.85.235

3.2.4 Requisitos de Software

Como la implementación del Honeyd se realiza sobre máquinas virtuales, entonces la primera aplicación que se debe tener en cuenta es el software de virtualización, que para este caso, se utilizó el VMWare Workstation 9.

Para el ejercicio de elegir el sistema operativo ideal en el que se va a crear la máquina virtual donde se instala el Honeyd, se hicieron pruebas con diferentes versiones de Linux, como por ejemplo, el *RedHat*, el *Fedora* y el *Ubuntu*. Los dos primeros presentaron algunos problemas en la configuración y operación del servidor proxy ARP del Honeyd, por esta razón, se elige la versión de Ubuntu ya que presentaba una mejor estabilidad en la implementación del ARP.

Finalmente, para llevar a cabo la instalación del Honeyd requiere de una serie de librerías o dependencias como: *Libevent, Libdnet, Libpcap, Farpd y Nmap.* Estas librerías pueden ser descargadas desde la página oficial de Honeyd [6], las funciones que cada una de ellas realizan se detallan a continuación:

- **Libevent**

Es un API que proporciona un mecanismo para ejecutar una función de devolución de llamada cuando ocurre un evento específico en un descriptor de archivo o después de que se haya alcanzado un tiempo de espera. También admite devoluciones de llamada debido a las señales de los tiempos de espera regulares [7].

- **Libdnet**

Proporciona una interfaz simplificada, portátil para varias rutinas de redes de bajo nivel [8], incluyendo:

- La manipulación de direcciones de red.
- El kernel ARP (4) caché y ruta (4) búsqueda en la tabla y la manipulación.
- Los cortafuegos de red (filtro IP, ipfw, ipchains, pf, PktFilter, ...)
- Búsqueda de interfaz de red y la manipulación
- Túneles IP (BSD / Linux tun, Universal TUN / TAP device)
- Prima de paquetes IP y transmisión de tramas Ethernet

- **Libpcap**

Esta librería es la encargada de capturar los paquetes que corresponden al funcionamiento de red a nivel de usuario [9].

- **Farpd**

Esta herramienta se usa para contestar cualquier petición ARP de alguna dirección IP que coincida con el rango específico configurado dentro del sistema. El Honeyd realiza con esta librería un ARP-Spoofing de las IP que se configuran en el *Honeypots*, estas IP hacen parte del rango de estaciones inactivas en el segmento de red.

- ***Nmap***

Es una herramienta vital para la auditoria de redes en una organización, su objetivo es encontrar servicios o puertos abiertos, averiguar el sistema operativo, encontrar qué equipos se encuentran activos en una red mediante el ping; estas acciones las realiza con el fin de encontrar huecos de seguridad en un servidor o máquina [10].

4 DISEÑO DEL *SNORT* Y EL *HONEYPOTS* DE BAJA INTERACCIÓN SOBRE LA RED DE GEOMÁTICA

Este diseño se realiza con el fin plantear una alternativa que permita mitigar algunos problemas de seguridad tanto en la red como en las estaciones de trabajo y servidores, los cuales se han manifestado en pérdida de la información o infección por virus en algunos equipos de la subred de producción. De igual manera, el análisis hecho a los servicios que ofrece el grupo de servidores ha determinado las diferentes vulnerabilidades y huecos de seguridad que éstos poseen.

El diseño que se plantea para la red del grupo Geomática se basa principalmente en dos argumentos importantes como son: la topología de la red y los resultados del análisis de tráfico de red que se llevaron a cabo con la herramienta *Ntop*. Es necesario aclarar que este diseño debe adaptarse a la topología de la red que ha sido implementada en el grupo por parte del DSI, es decir, que el grupo no cuenta con la autonomía ni el control sobre la distribución de la red y los elementos que las interconectan como son: el Rack, los switches, el cableado, entre otros.

Teniendo en cuenta los argumentos mencionados anteriormente y sabiendo que el diseño que se debe plantear no debe incurrir en grandes costos adicionales, pero tampoco comprometer la calidad de la solución, se planteó que la mejor opción en donde se debe implementar el *Snort* es en la subred en donde se encuentran los servidores o zona desmilitarizada, es decir la 192.168.**19**.0/24 , debido a que en esta zona es donde encuentran todos los servicios (web, remoto, smb, entre otros) que se exponen hacia el exterior y por consiguiente existen muchas “puertas abiertas” que hacen que estén expuestos constantemente a ataques tanto del exterior como desde la red interna de la UIS.

Mientras que el *Honeypots* se ubicará en la subred de producción del grupo, es decir la 192.168.**85**.0/24, ya que varios de estos equipos han presentado evidencia de haber sido atacados por el hecho de compartir archivos entre ellos a través de la red, por lo tanto, se requiere entrar a analizar cuáles son los métodos que están usando los atacantes y sobretodo con qué intención es que están haciendo este tipo de ataque que generalmente consiste en pérdida de información.

4.1 DISEÑO DEL *SNORT* EN LA SUBRED DE SERVIDORES

El *Snort* en su definición más elemental, es un sistema multiplataforma diseñado para analizar el tráfico que circula a través de una red, para compararlo con una serie de reglas predeterminadas y generar las respectivas alarmas.

Existen diversas posibilidades en donde se pueda ubicar el *Snort* dependiendo del objetivo que se desee lograr y en función del tráfico que se quiere monitorizar. La ubicación del *Snort* se debe hacer de tal manera que garantice la interoperabilidad de la red, es decir, que el *Snort* pueda obtener o compartir con otros sistemas como switches, routers y firewall, sin crear traumatismo en la red. Los diferentes lugares dentro de una red en donde se pueden ubicar el *Snort* son:

- **Delante del *firewall*:**

En este lugar el *Snort* puede comprobar todos los ataques que provienen del exterior de la red, aunque muchos de ellos no se hagan efectivos porque quedan en el firewall. La desventaja es que genera gran volumen de información en los archivos logs debido a la cantidad de tráfico que captura.

- **Detrás del *firewall*:**

Es una de las ubicaciones que presenta mejor desempeño, debido a que el *Snort* puede analizar todo el tráfico que entra a la red y que no haya sido bloqueado por el firewall, lo que hace que no se genere gran volumen de información. Dentro de este esquema se pueden presentar las diferentes opciones de ubicación que son:

  - *Detrás de un Hub*: une las conexiones y no altera los paquetes que llegan.
  - *Detrás de un switch*: en donde el switch almacena el paquete antes de reenviarlo al *Snort*.
  - *Modo bridge*: de esta manera establece comunicación entre dos adaptadores de una red.

- **Combinación de los anteriores:**

Cuando se ubica el *Snort* delante y detrás del firewall el control es mucho mayor, ya que se puede implementar una correlación entre ataques detectados a lado y lado. La desventaja es que se hace necesario tener dos equipos para implementarlo.

- ***Firewall/Snort*:**

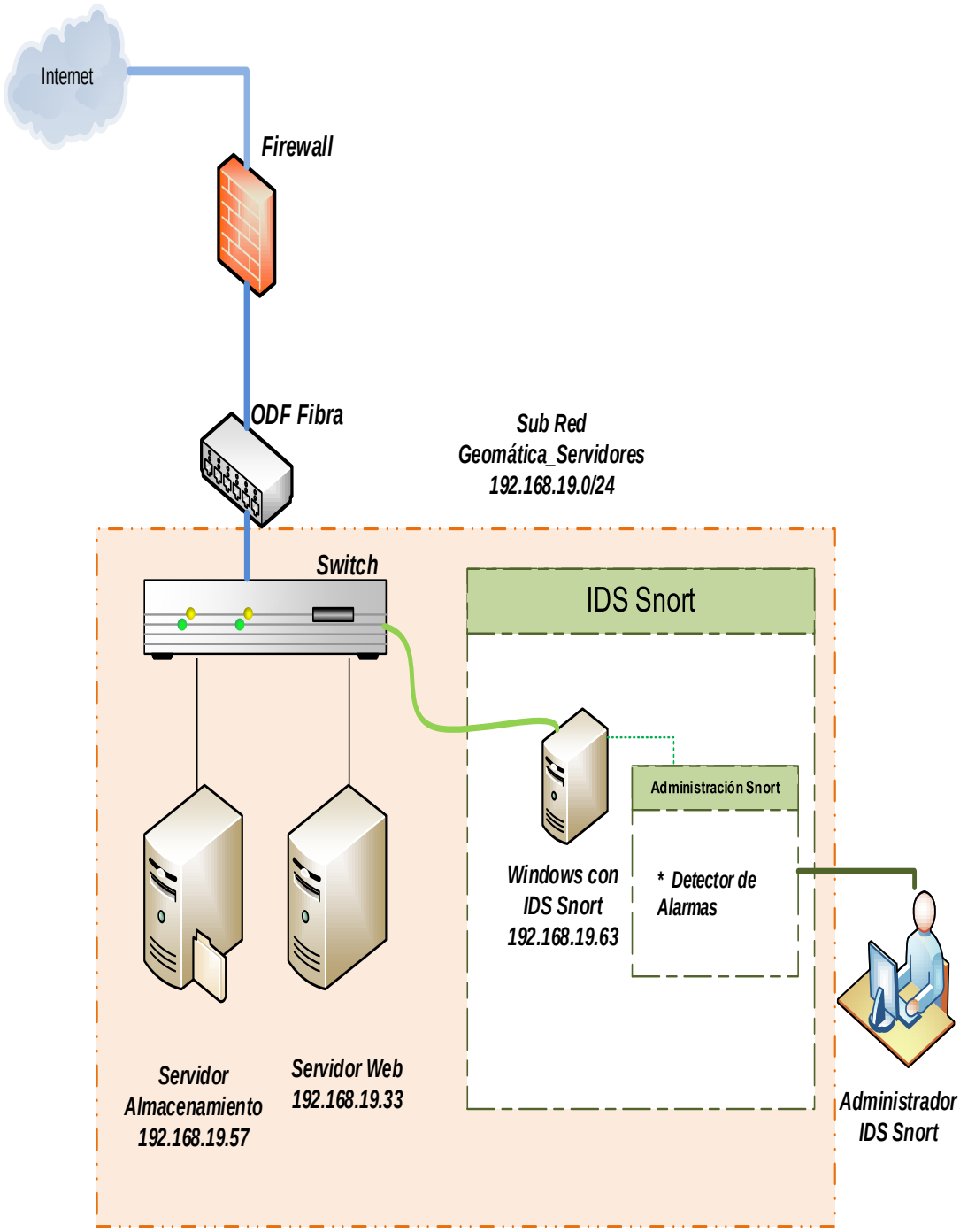
En esta opción de implementar en un mismo equipo el firewall y el *Snort* se obtienen buenos resultados.

- **Zona desmilitarizada (DMZ):**

O también llamada red perimetral, es una subred localizada entre la red interna y externa de una organización. Su objetivo es permitir que la conexión desde la red interna y la externa a la DMZ estén permitidas. Esta subred generalmente se usa para ubicar los servidores (Web, DNS, correo) debido a que éstos son necesarios que sea accedido desde el exterior de la red.

Con los argumentos expuestos al comienzo de esta sección junto con las posibles ubicaciones que puede tener el *Snort* dentro de una red, una solución preliminar que se plantea para el grupo Geomática es que el *Snort* se debe ubicar dentro de la zona desmilitarizada, es decir, en la zona donde se encuentran los servidores debido a la gran cantidad de información y servicios que son necesarios tener disponible todo el tiempo por parte de los servidores; además, permitirá apoyar los sistemas de filtrado que existen en sus sistemas operativos y llevar a cabo la monitorización de los accesos no autorizados en cada uno de ellos. En la figura 11 muestra el diseño del *Snort* que se propone para la subred 192.168.**19**.0/24 DMZ del grupo.

Figura 11 Diseño del Snort en la subred 192.168.19.0/24 donde se encuentran los servidores del grupo Geomática



Para llevar a cabo la validación del diseño que se plantea, se hace necesario contar con un equipo que tenga buenos recursos en hardware tal como se presentó en la Tabla 4, ya que se necesita una buena capacidad de almacenamiento para gestionar la cantidad de archivos *logs* que se generan debido al gran tráfico que tienen los servidores; de igual manera, debe tener una buena capacidad de procesamiento, porque por cada paquete que llega a la red, existe un consumo considerable en memoria RAM por parte del decodificador, el preprocesador y el motor de ejecución del *Snort*.

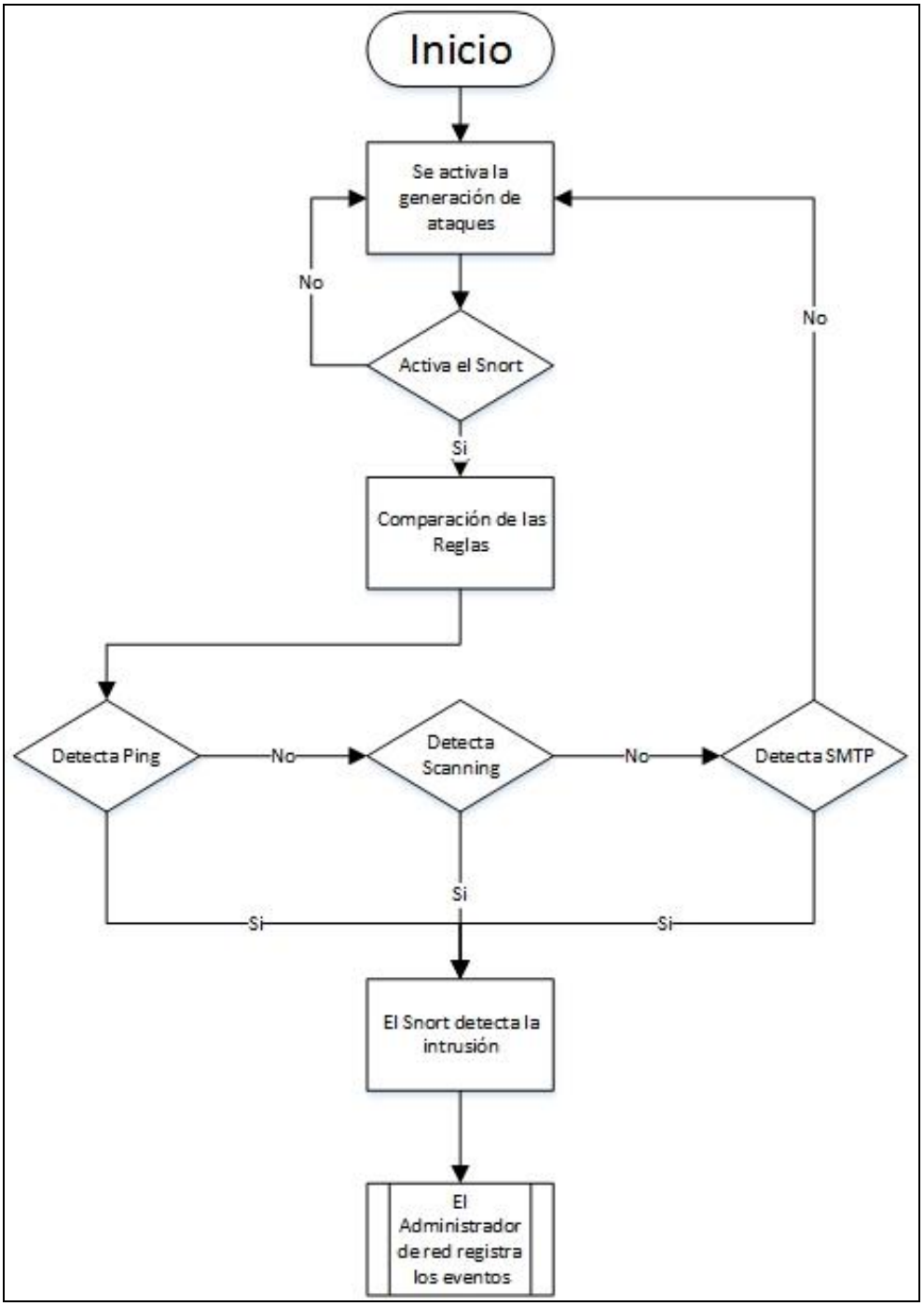
Este *Snort* será de tipo *NIDS*, es decir, tendrá la función de capturar y analizar los paquetes que llegan a cada uno de los servidores con el fin de mitigar cualquier intento de intrusión, estas intrusiones se pueden producir de varias maneras como:

- Los atacantes que provienen desde internet.
- Usuarios autorizados del sistema que intentan ganar privilegios adicionales para los cuales no están permitidos.
- Usuarios autorizados que hacen mal uso de los privilegios o recursos que se le han sido asignados.

Para efectos de la prueba piloto con el fin de validar el diseño del *Snort*, se implementaron una serie de reglas sobre algunos de los tipos de ataques más comunes que se presentan en esta subred, sin embargo, para una futura implementación será necesario de un análisis de tráfico más exhaustivo con el fin de definir la mayor cantidad de reglas posibles y realizar las validaciones de las mismas para que no se lleguen a presentar falsos positivos.

El funcionamiento del *Snort* inicia su actividad en el momento en que se activa la generación del ataque con algún tipo de mensaje programado como amenaza, después de que el paquete le ha sido descifrado su protocolo por medio del decodificador, los preprocesadores se encargan de reconstruirlo, para que luego el motor de detección se encargue de compararlos con las reglas que fueron definidas, que para efecto de la validación se implementaron reglas para el escaneo de puertos, el ping y SMTP. Si se llega a presentar alguna correlación entre el paquete y algunas de esas reglas, entonces el *Snort* disparará la alerta que tenga programada, de lo contrario, se descarta el paquete. En la figura 12 se muestra el diagrama de flujo que obedece al comportamiento del *Snort* diseñado para el grupo Geomática.

Figura 12 Diagrama de flujo del funcionamiento del Snort de Geomática



4.2 DISEÑO DEL *HONEYPOTS* DE BAJA INTERACCIÓN EN LA SUBRED DE PRODUCCIÓN (192.168.85/24)

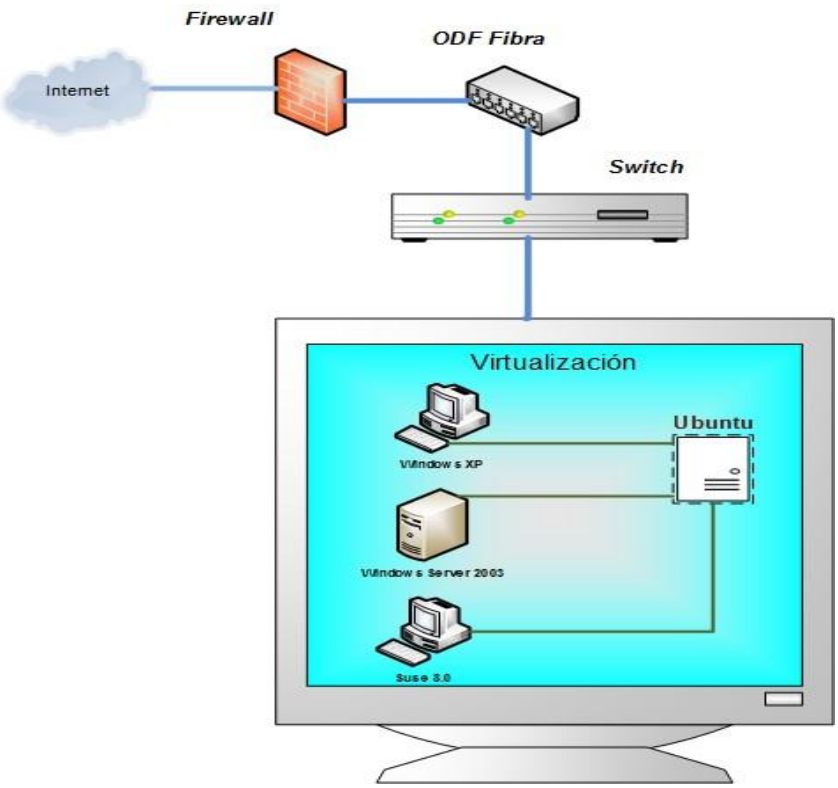
El *Honeypots* desde el punto de vista conceptual, es un sistema que está conformado por un conjunto de máquinas bien sea reales o virtuales llamadas *máquinas trampa*, las cuales presentan una serie de vulnerabilidades en su sistema operativo y están monitorizadas por un equipo de control central que se encarga de almacenar toda actividad que se lleve a cabo dentro del rango de direcciones IP asignadas a estas máquinas.

En cada una de estas máquinas trampa se les configura un sistema operativo en particular, de igual manera, también se les pueden habilitar varios servicios como por ejemplo, el de transferencia de archivos o ftp, el de correos, el web, entre otros; incluso se puede colocar deliberadamente información sensible aunque falsa, de manera que se muestren más atractivas para los atacantes.

Con el fin de no generar sobrecostos en una futura implementación de este diseño, sin comprometer la calidad de la solución y dado que las máquinas trampa no requieren grandes recursos de hardware, se plantea un esquema de diseño basado en un *Honeypots* Virtual que se monta sobre un software de virtualización dentro una máquina real que pertenece a la red de producción del grupo. De igual manera que con el *Snort*, el diseño del *Honeypots* se debe plantear sobre la topología de red existente en el grupo.

En la figura 13 muestra un esquema preliminar únicamente de la arquitectura del *Honeypots* virtual que funciona sobre la máquina real.

Figura 13 Diseño preliminar del Honeypots Virtual sobre un equipo real de la subred de producción 192.168.85.0/24



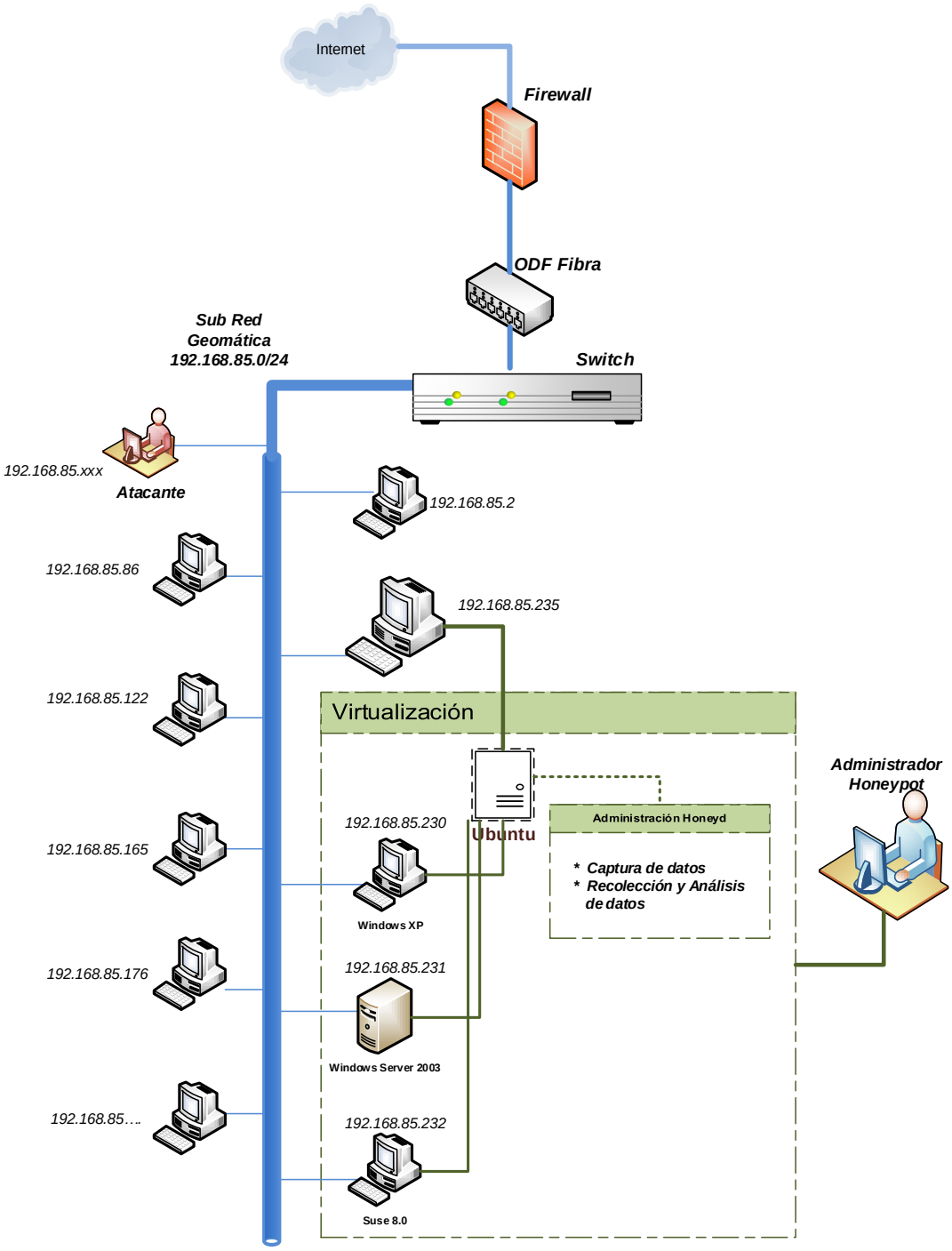
Para el diseño de este *Honeypots* Virtual se plantea seleccionar un equipo dentro de la red que tenga de buenos recursos de hardware, ya que este diseño pretende implementar todo un conjunto de máquinas trampas con el fin de que ocupen varias direcciones IP que no están activas en la red. Sin embargo, para efectos de las pruebas de validación, el diseño contemplará la implementación de tres máquinas trampas a las cuales se les asignarán direcciones IP propias de la red. Estas direcciones fueron escogidas aleatoriamente dentro del rango de IPs que no se están usando y son las que van desde la 192.168.85.230 hasta la 192.168.85.232.

La máquina virtual que contendrá al *Honeypots* o comúnmente llamado equipo de control, funciona en modo de puente Ethernet y opera de forma transparente a nivel de IP, de tal manera que permite conectar las máquinas trampa a la red, pero al mismo tiempo las mantiene aisladas del resto de equipos que están conectados, sin tener que implementar una subred exclusiva para ellas. Este aislamiento de las máquinas trampa se realiza con

el fin de que el equipo de control pueda tener un registro de todas las conexiones entrantes y/o salientes que ocurren entre ellas y así poder monitorizar cualquier conexión que se intente hacer desde la red interna del grupo.

En la figura 14 se muestra el diseño del *Honeypots* que se plantea para el grupo Geomática, el cual se escogió un equipo de la sala Geomática II cuya especificaciones técnicas se mencionaron anteriormente en la Tabla 5. Como lo muestra este esquema, el *Honeypots* está montado sobre una máquina virtual en este equipo y funciona en conjunto dentro de la misma red de producción ya que pertenece al mismo rango de direcciones IP destinadas para esta subred (192.168.85.0/24).

Figura 14 Diseño del Honeypots de baja interacción para la subred de producción 192.168.85.0/24 del grupo Geomática



Este *Honeypots* divide de manera lógica la red del grupo, separando las máquinas trampa del resto de los equipos creando una red privada entre ellas, tal como se muestra en la figura 14. Esta separación hace que las máquinas trampa no capturen todo el tráfico que circula por el resto de la red, es decir, solamente se tendrá en cuenta el tráfico entrante y/o saliente que pase entre ellas. Por esta razón, el tráfico que exista entre los demás equipos de la red con el servidor o hacia afuera no será detectado por la red de máquinas trampa, mientras que, si desde el la red se realiza un ataque como por ejemplo, el escaneo de direcciones IP a toda la red 192.168.85.0/24, entonces tanto los equipos de la red como las máquinas trampa responderán a esas conexiones, haciéndole creer al atacante que todos son equipos reales.

El equipo de control se puede implementar sobre un sistema operativo Linux, que para efectos de la validación se escogió el Ubuntu 13.10, de igual manera, las máquinas trampa se pueden configurar con cualquier sistema operativo, que como se mencionó anteriormente, para realizar las pruebas se establecieron tres máquinas trampa, una con sistema operativo Windows XP, otra con Linux Suse y para incitar más a un ataque, la tercera máquina trampa está configurada como Windows Server 2003, además a cada una de estas máquinas se les configura algunos servicios como el de *FTP*, web, *ICMP*, entre otros.

Al momento de implementar el *Honeypots* hay que tener en cuenta que se debe limitar en el router la cantidad de tráfico saliente por parte de las máquinas trampa, ya que si un atacante logra capturar el equipo donde se encuentra instalado el *Honeypots*, puede luego utilizar estas máquinas como arma de ataque hacia otros sistemas.

El funcionamiento de este *Honeypots* inicia su actividad desde el momento en que atacante intenta hacer algún tipo de conexión con una o varias de estas máquinas trampa. Entonces cuando se envía un paquete desde el equipo del atacante que también se encuentra conectado a la red interna del grupo, la librería Libcap del *Honeypots* realiza la verificación del paquete, es decir, identifica a qué tipo de protocolo pertenece TCP, UDP o ICMP y dependiendo del tipo escoge alguna de las siguientes actividades:

- Si es otro protocolo que no es reconocido por el *Honeypots*, guarda el registro y rechaza la solicitud.
- Si es una petición ICMP, sólo responde con un eco (como es el caso del ping).
- Pero si se trata de una petición TCP o UDP, consulta la configuración del honeyd.conf y si está configurada, se presenta emulando el servicio solicitado a través de la librería Libnet.

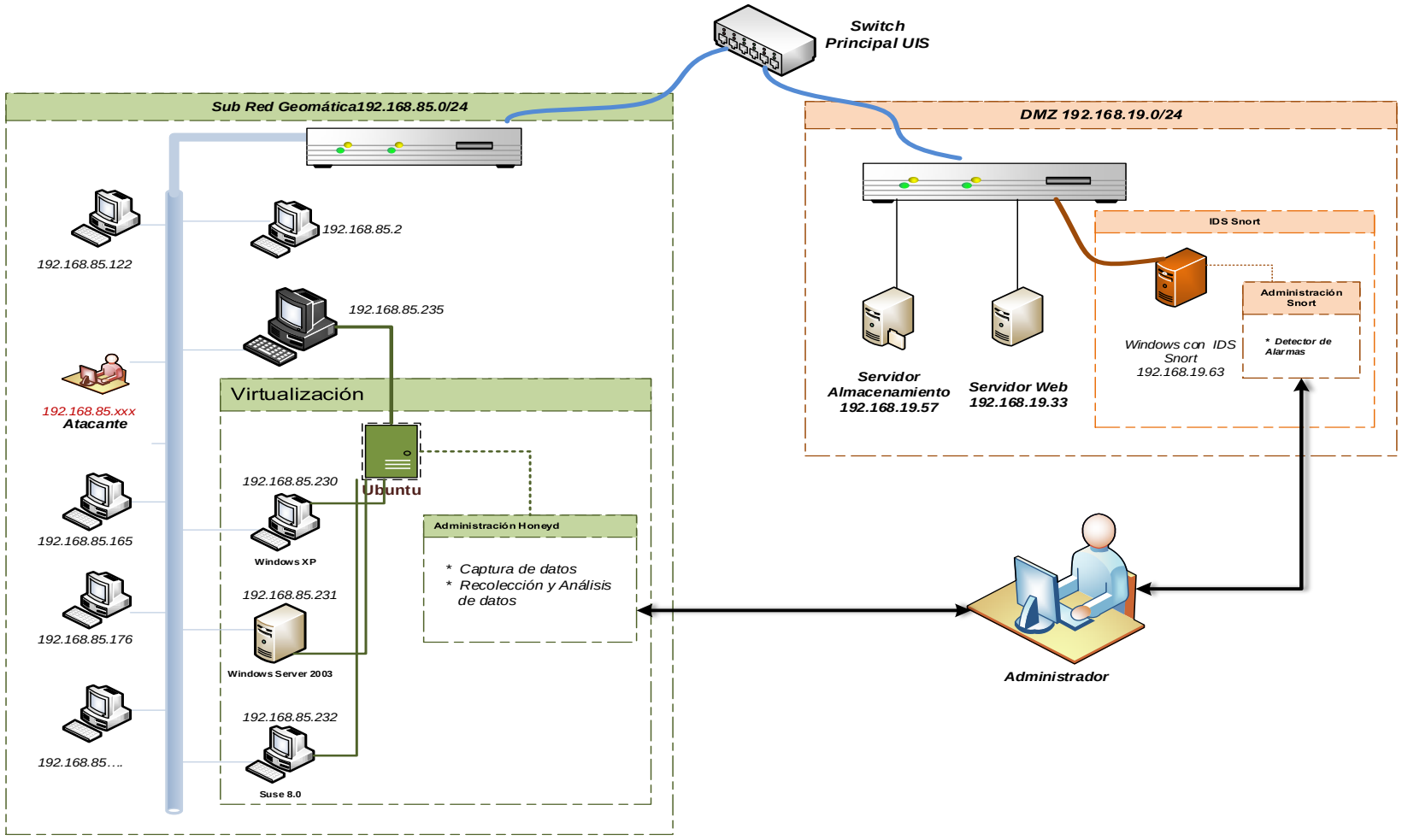
Como se mencionó anteriormente, este diseño contempla la emulación de tres máquinas trampa, sin embargo, para tener mayores probabilidades de que un atacante caiga en algunas de estas máquinas se sugiere establecer un número mayor de máquinas trampa dentro del *Honeypots*.

4.3 DISEÑO CONSOLIDADO DEL *SNORT* Y *HONEYPOTS* PARA LA RED DE GEOMÁTICA

En la figura 15 se muestra el diseño unificado que se planteó para la red del grupo Geomática, que como se mencionó anteriormente está limitado a las características propias que tiene la red, es decir, este modelo fue diseñado de tal manera que se adaptara a la topología existente en la red, debido a que no se tiene la autonomía suficiente para generar otros modelos que impliquen una reestructuración o reorganización de la red. Esta gráfica está compuesta por el diseño del *Snort* para la subred de servidores (lado derecho de la gráfica) y el diseño del *Honeypots* de baja interacción para la subred de producción (lado izquierdo de la gráfica).

Esto significa que a pesar de que es una misma red, los diseños planteados se manejan por separado, debido no es viable colocar el *Honeypots* en la misma subred donde se encuentra el *Snort*, ya que se generarían muchos falsos positivos debido a las reglas del *Snort*. Por otra parte, si se instala el *Snort* en la subred donde se encuentra el *Honeypots*, la mayoría de los atacantes quedarían filtrados por las reglas del *Snort* perdiendo la posibilidad de ingresar a la red para ser atraídos por el *Honeypots* y poder analizar su comportamiento.

Figura 15 Esquema unificado del diseño del Snort y Honeypots sobre la red del grupo Geomática



5 VALIDACIÓN DEL DISEÑO

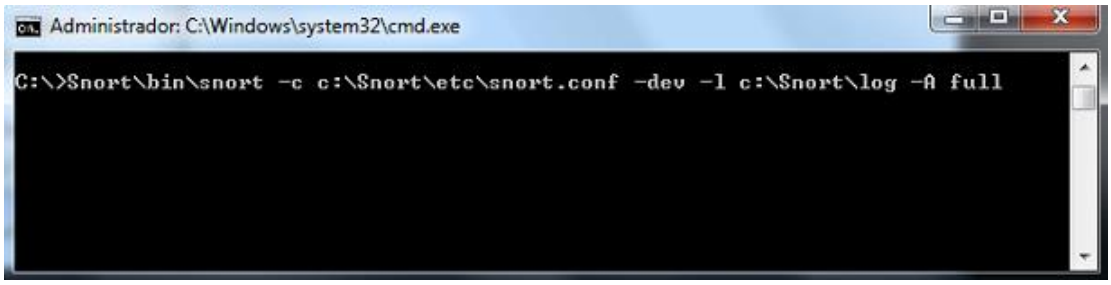
5.1 VALIDACIÓN DEL DISEÑO DEL *SNORT*

5.1.1 Inicialización del servicio *Snort*

Al terminar la etapa de instalación y configuración del *Snort* y con el objetivo de hacer las pruebas de validación del diseño, se requiere que el servicio de *Snort* se encuentre inicializado y para ello se abre una ventana de comandos de Windows en donde se digita la siguiente instrucción que aparece en la figura 16.

*C:\Snort\bin**Snort** -c c:\Snort\etc\Snort.conf -dev -l c:\Snort\log –A full*

Figura 16 Ejecución del Snort



Fuente: *cmd.exe Windows* para la ejecución de la herramienta *Snort*

5.1.2 Pruebas de Validación

5.1.2.1 Requerimientos de software

Para realizar las pruebas con el fin de validar las reglas que fueron implementadas a manera de prototipo, se utilizaron algunos programas para simular el ataque al *Snort*. Estos programas se ejecutaron bajo entorno Windows ya que es el sistema operativo en el cual está instalado el *Snort*. Algunos de ellos son:

- **Consola o Terminal:** es una aplicación que se encarga de interpretar comandos propios de cada sistema operativo. Para abrir la consola en Windows se ejecuta la instrucción *cmd.exe*, mientras que en Linux, comúnmente existe un ícono que se llama *Terminal*. A través de esta consola

se pueden hacer diferentes tipos de ataques que van desde una comprobación de conexión a un equipo como lo es el *ping*, hasta envío de virus o gusanos informáticos.

- **Advanced IP Scanner:** es una herramienta gratuita que sirve para hacer una exploración en la red, en estas exploraciones detecta direcciones IP, dispositivos que están conectados, puertos abiertos y carpetas compartidas por parte de algunos equipos; incluso también puede ejecutar comandos de consola como: ping, telnet, ssh, entre otros.

- **Advanced Port Scanner:** es una herramienta gratuita que se encarga de hacer un examen completo de todos los puertos de conexión y su descripción en uno o varios equipos de forma simultánea y en pocos segundos.

- **Nmap:** es una herramienta de código abierto que sirve para el rastreo y descubrimiento de puertos en una red, así como el estado de actividad que tengan (abiertos, cerrados o sin acceso). Por esta razón, esta herramienta es usada para el descubrimiento de los servidores que hay en una red, un vez identificado un equipo, puede determinar qué servicios está ejecutando y hasta qué sistema operativo tiene.

5.1.2.2 Plan de pruebas de ataque

Con el fin de llevar a cabo la prueba piloto sobre la validación del diseño del *Snort* a través de la validación de sus reglas, se hizo el ejercicio de realizar dos tipos de ataque más comunes que se presentan generalmente en el área de los servidores, estos ataques son :

- **Ping:** esta instrucción generalmente se ejecuta desde una terminal, su función principal es la de comprobar el estado de conexión que hay desde la máquina atacante con la máquina objetivo. De igual manera, esta instrucción puede diagnosticar el estado, velocidad y calidad de la red.

  Cuando se envía un ping a una máquina remota, lo que sucede es que desde la máquina local se le envían paquetes del protocolo ICMP de solicitud de eco y respuesta de eco. La respuesta de eco también es un mensaje ICMP enviado por la máquina remota como respuesta a ese ping.

- **Escaneo de direcciones IP:** esta actividad es realizada por el atacante para identificar cuáles son las direcciones IP que se encuentran activas en la red. El programa que se usó para hacer este escaneo es el Advanced IP Scanner.

5.1.2.3 Ejecución de ataques

Para llevar a cabo la ejecución del ataque del sistema *Snort*, el primer paso es definir qué tipo de ataque va a realizar con el fin de crear la regla de validación que va a controlar ese ataque, esto se realiza en el archivo de reglas del *Snort*. Luego, se usa el programa con el cual se va a ejecutar el ataque y poder ver a través de la consola de alertas el informe del ataque. Los ataques fueron los siguientes:

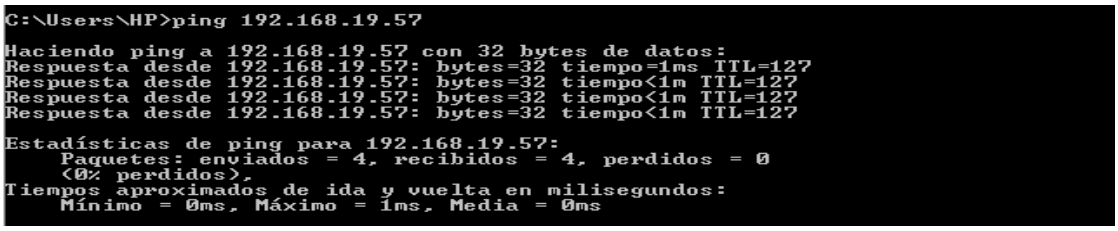
- **Ping:**

Antes de ejecutar el ataque de tipo ping se crea la regla que va a contrarrestar este procedimiento. La regla para el ataque a esta subred es la siguiente:

alert icmp any any -> any any (msg:"ICMP PRUEBA DE REGLA PING"; sid:1000001)

Este ataque de tipo ping se realizó al servidor de aplicaciones del grupo, el cual tiene asignado la dirección IP 192.168.19.57 y fue ejecutado desde dos máquinas diferentes: una fuera de la subred con IP 192.168.24.210 y la otra dentro de la red de producción de Geomática con IP 192.168.85.245, como se muestra en la figura 17.

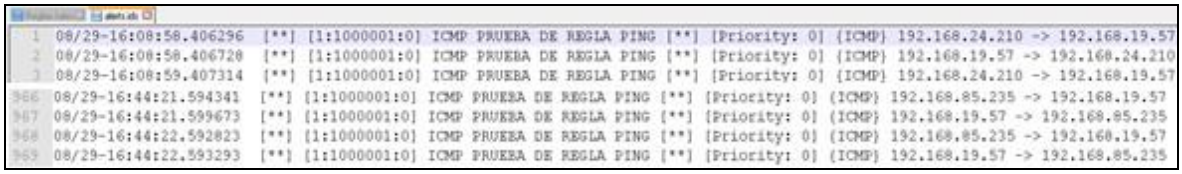
Figura 17 Ataque tipo ping al servidor 192.168.19.57



Fuente: cmd.exe Windows para la ejecución de la herramienta Ping

En la figura 18 se visualiza la alerta que lanzó el *Snort* desde la interfaz de su sistema de alertas e informes.

Figura 18 Alerta disparada por el Snort debido a un ataque de tipo Ping



Fuente: Archivo alert.ids de la herramienta Snort

- **Escaneo de direcciones IP**

Para este tipo de ataque se utilizó una herramienta para el escaneo de direcciones IP de tipo *Nmap* llamada *Zenmap*, a través de esta herramienta se hizo la validación de la regla para los ataques en la red de tipo escaneo y que vienen con la bandera FIN activada, esta bandera tiene la función de cerrar la conexión después de haber transferido el paquete, lo cual se usa para no dejar conexiones abiertas de modo que no quede ningún rastro del atacante. La regla para este tipo de ataque es:

alert tcp any any -> 192.168.19.0/24 any (msg:"Paquete FIN detectado dentro de la red"; flags: F; sid: 139;)

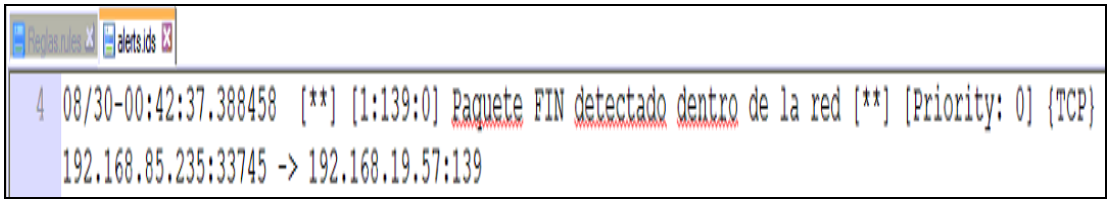
En la figura 19 se muestra el ataque realizado a través de la herramienta Zenmap, en donde se hace ataque de escaneo al puerto 139 del servidor que está en la dirección IP 192.168.19.57.

Figura 19 Ataque de tipo Escaneo de direcciones IP con bandera FIN activada

Fuente: Herramienta *Zanmap*

Una vez realizado el ataque anterior, se revisa el archivo *alerts.ids* que es donde se guardan las alertas que dispara el *Snort*, en la figura 20 se visualiza la alerta emitida por el anterior ataque.

Figura 20 Alerta generada por el Snort debido al ataque de Escaneo de direcciones IP



Fuente*:* Archivo alert.ids de la herramienta Snort

- **Alarma de tráfico UDP**

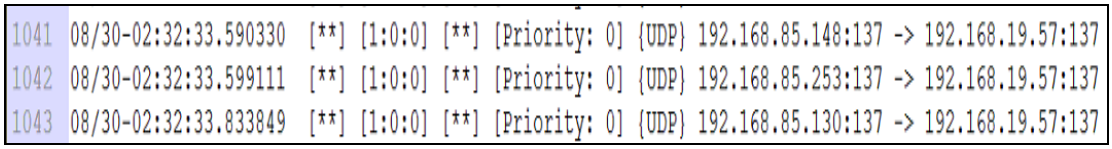
El ataque de denegación de servicios tiene diferentes formas de realizarse, una de ellas es el llamado *inundación UDP* (UDP flood), el cual consiste en generar grandes cantidades de paquetes UDP, ya que aprovechan la naturaleza sin conexión de este protocolo. La regla para este tipo de ataque es:

alert udp any any -> 192.168.19.0/24 1:1024

Para realizar este ataque se abre varias veces una página web con dirección *http://emergenciasantander.uis.edu.co* que se encuentra en el servidor web con dirección IP 192.168.19.33.

En la figura 21 se muestra en el archivo de alertas el registro del ataque que se realizó anteriormente.

Figura 21 Alerta generada debido al ataque de tráfico UDP



Fuente: *Archivo alert.ids de la herramienta Snort*

5.2 VALIDACIÓN DEL DISEÑO DEL *HONEYPOTS* DE BAJA INTERACCIÓN

5.2.1 Inicialización de los servicios

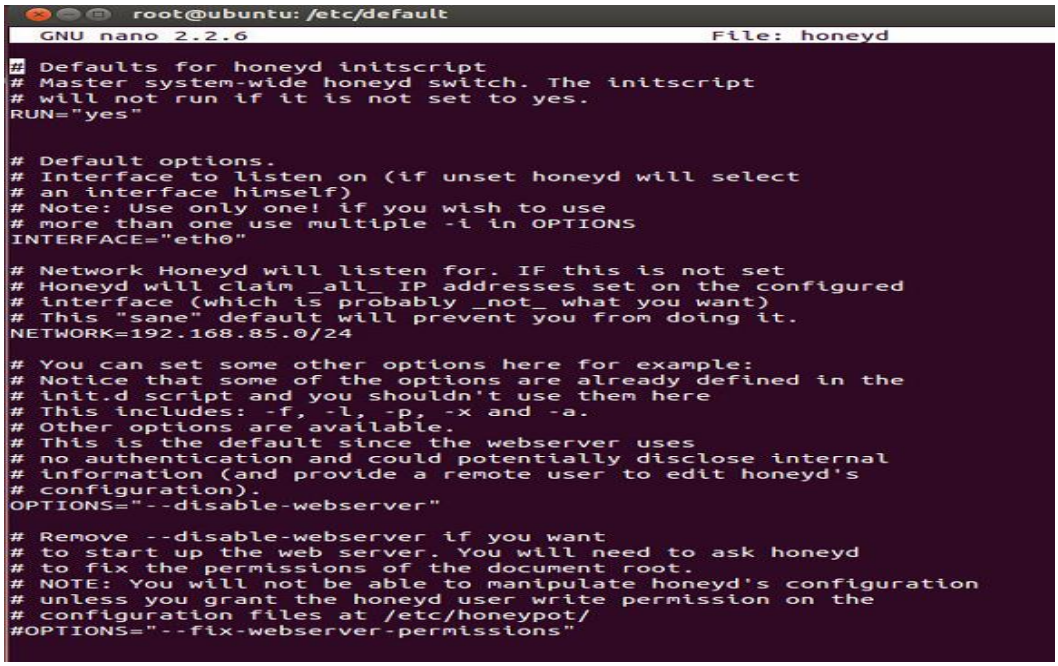
Una vez finalizada la etapa de instalación y configuración del Honeyd, se inicia el servicio o "demonio", pero antes hay que configurar algunos parámetros para su funcionamiento. Este archivo de configuración se edita de la con la siguiente instrucción:

root@ubuntu:$ nano /etc/default/honeyd

Al ejecutar la anterior instrucción en la consola, se abre el archivo honeyd, tal como se muestra en la figura 22 los parámetros que se deben modificar son los siguientes:

RUN="yes" para iniciar el demonio.
INTERFACE="eth0" interfaz utilizada por el Honeyd.
NETWORK= "192.168.85.0/24" para seleccionar el rango de direcciones IP de la red.

Figura 22 Estructura del archivo /etc/default/honeyd



Fuente: *Archivo configuración herramienta* honeyd

Luego, se ejecuta el arranque del servicio o demonio del Honeyd con la siguiente instrucción:

root@ubuntu:$ /etc/init.d/honeyd start

De igual manera, se debe inicializar el servicio del *farpd*, que es el que permite reconocer el ARP Spoofing el rango de direcciones IP que se necesita que simule el Honeyd. Con la instrucción que aparece a continuación se lleva a cabo la inicialización de este servicio.

farpd -d -i eth X *192.168.85.229-192.168.85.232*

-d indica la habilitación de la información de las peticiones que escucha y acepta.
-i eth0 interfaz por la cual escucha la maquina donde se encuentra el *Honeypots*
19.168.85.230-192.85.232 rango para configurar ARP-Spoofing que es el mismo rango que está especificado en el archivo de configuración del Honeyd.

La ejecución del comando anterior y la respuesta a éste se visualiza en la figura 23.

Figura 23 Inicialización del servicio *farpd*

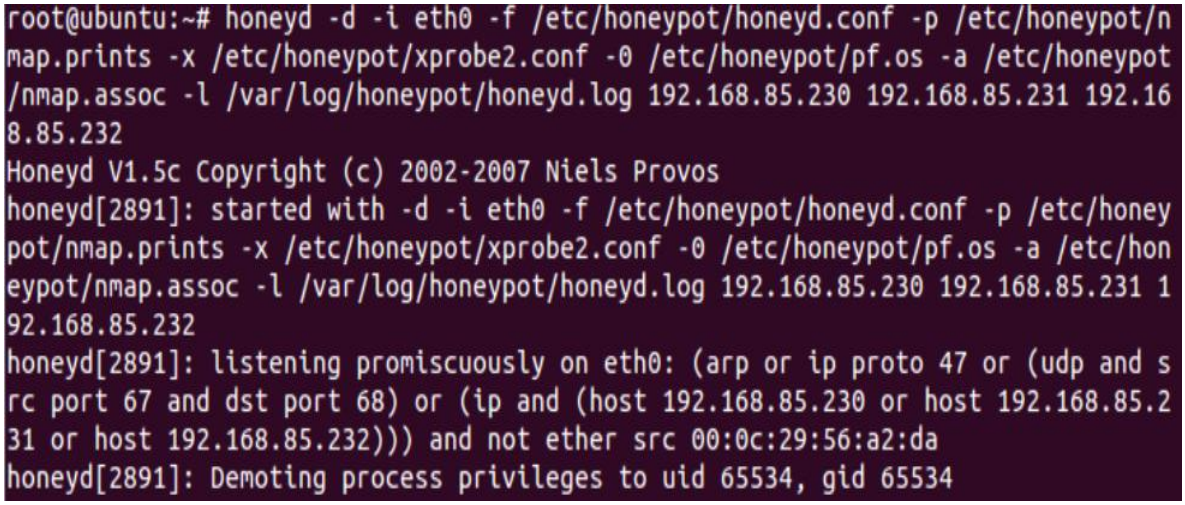
```
root@ubuntu:/etc/default# farpd -d -i eth0 192.168.85.230-192.1
68.85.232
arpd[2974]: listening on eth0: arp and (dst net 192.168.85.230/
31 or dst net 192.168.85.232/32) and not ether src 00:0c:29:56:
a2:da
```

Fuente*:* Herramienta ***farpd***

Finalmente, con la instrucción siguiente se lleva a cabo la emulación de las máquinas trampa definidas en el archivo de configuración, por parte del Honeyd. La ejecución de esta instrucción y la respuesta por parte del Honeyd se muestra en la figura 24.

honeyd -d -i eth0 -f /etc/Honeypots/honeyd.conf -p /etc/Honeypots/nmap.prints –x /etc/Honeypots/xprobe2.conf -0 /etc/Honeypots/pf.os -a /etc/Honeypots/nmap.assoc -l /var/log/Honeypots/honeyd.log

Figura 24 Emulación de las máquinas trampa configurados en el Honeyd



Fuente: Vista ejecución *honeyd*

5.2.2 Pruebas de validación

5.2.2.1 Requerimientos de software

Para llevar a cabo el plan de pruebas se utilizaron programas propios del sistema operativo tanto en Linux como en Windows, como es el caso de la consola o terminal. Sin embargo, algunas otras pruebas de ataque se requirieron de otras herramientas adicionales que se detallan a continuación:

- Consola o Terminal
- Advanced IP Scanner
- Advanced Port Scanner
- Nmap

Estas fueron las mismas herramientas que se usaron en los requerimientos del software para la validación del *Snort* (sección 5.3.2.1).

5.2.2.2 Plan de pruebas de ataque

Para llevar a cabo la prueba piloto sobre validación en la implementación del diseño del *Honeypots* de baja interacción para el grupo Geomática, se escogieron

los tipos de ataque más comunes que se presentan en una red. Los ataques que se describen a continuación, los dos primeros ya fueron explicados en el plan de pruebas de ataque del *Snort* (sección 5.3.2.2):

- **Ping**
- **Escaneo de direcciones IP**
- **Escaneo de Puertos:** es una acción que realiza el atacante con el fin de analizar el estado (abierto, cerrado o protegido) en que se encuentran los puertos de una máquina en la red. Al conocer el estado de los puertos, puede descubrir qué servicios está ofreciendo esa máquina e identificar las posibles vulnerabilidades de seguridad y hasta qué sistema operativo tiene instalado. Para realizar esta actividad se usó el programa Advanced Port Scanner.

  Para efectos de la validación de la prueba piloto, los puertos que se habilitaron en las máquinas trampa fueron:

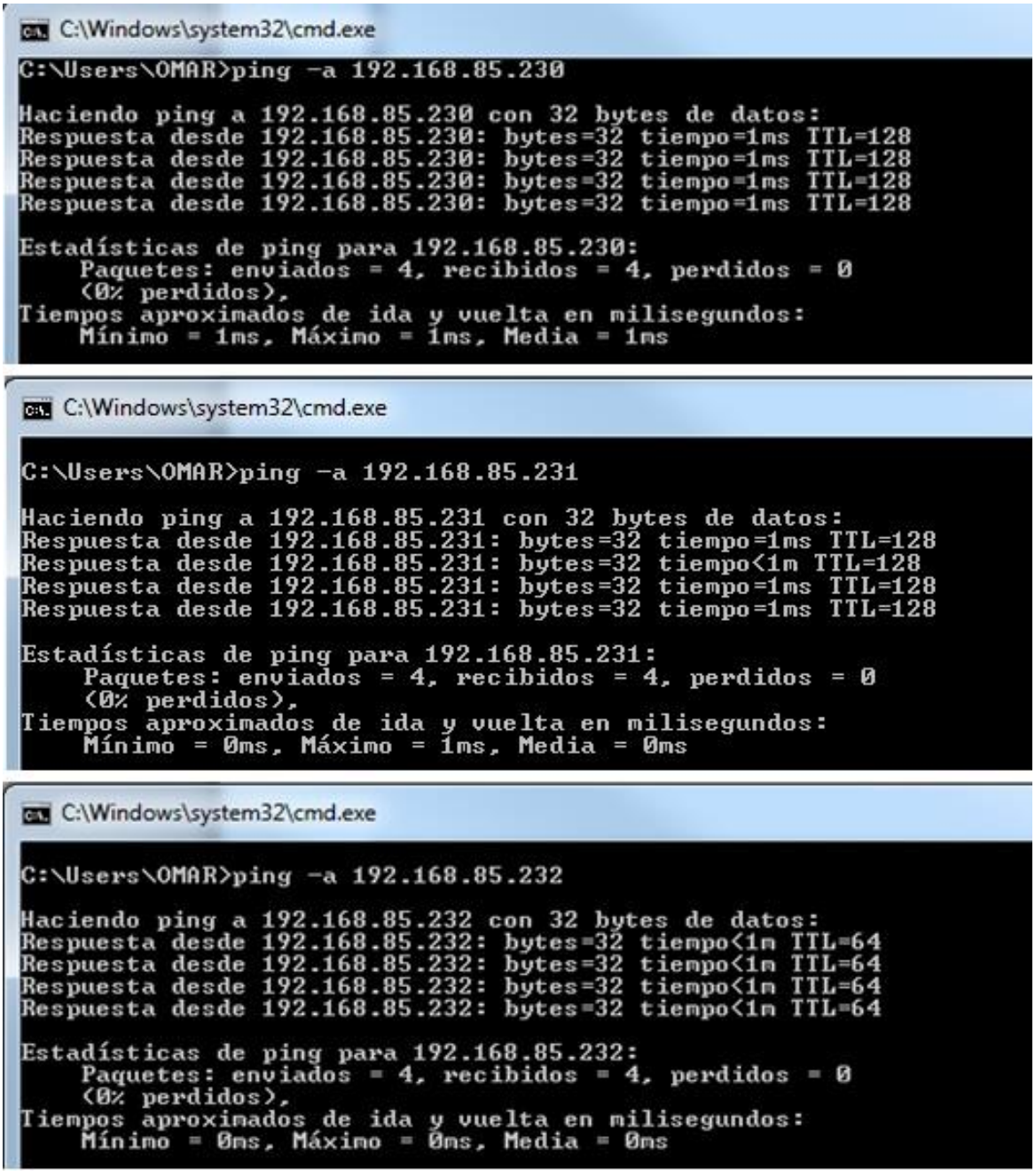
  - Servicio icmp: que no utiliza puertos.
  - Servicio web: que escucha por el puerto 80.
  - Servicio ftp: que escucha por el puerto 21.

5.2.2.3 Ejecución de ataques

- **Ping**

El ataque de tipo ping se realizó desde dos máquinas diferentes en la red tal como se muestra en la figura 25.

Figura 25 Ataque de tipo Ping



Fuente: cmd.exe *Windows* para la ejecución de la herramienta *Ping*

La respuesta que emitieron las máquinas trampa se almacenaron en la bitácora que se muestra en la figura 26.

Figura 26 Respuesta de las máquinas trampa al tipo de ataque Ping

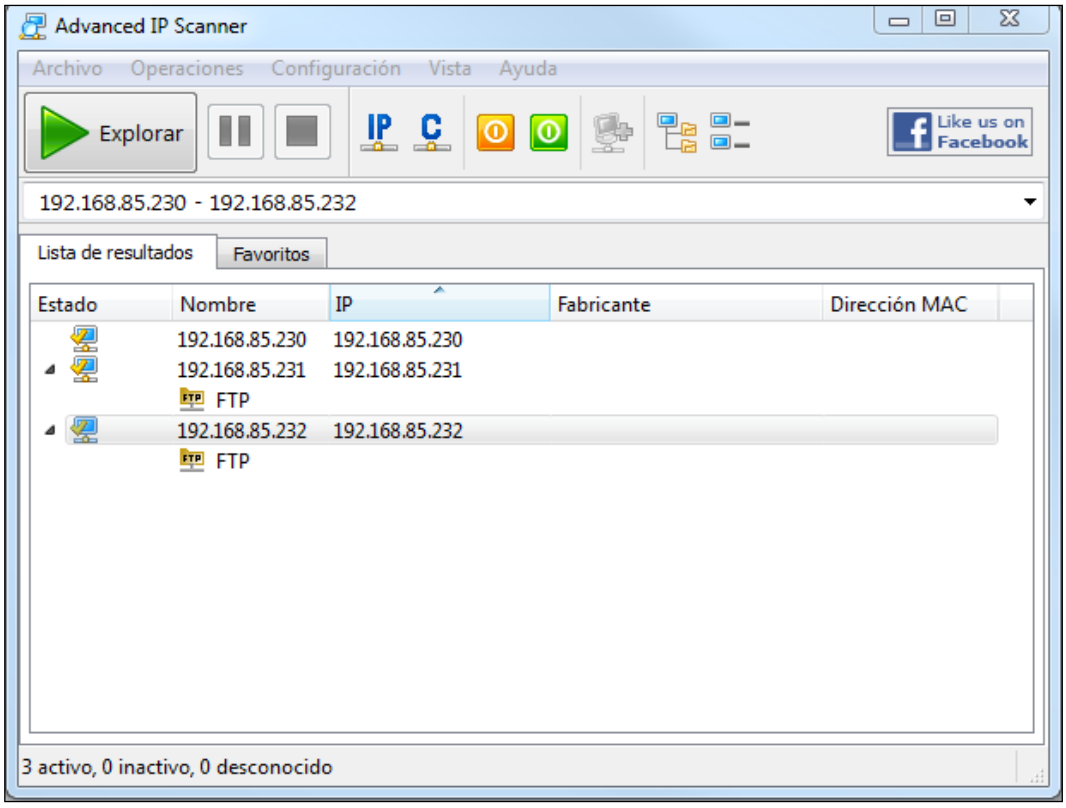
```
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.230 -> 192.168.85.235
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.232 -> 192.168.85.235
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.231 -> 192.168.85.235
```

Fuente: alerta *honeyd*

- **Escaneo de direcciones IP**

Como se mencionó anteriormente, para este tipo de ataque se utilizó la herramienta Advanced IP Scanner, el cual permite filtrar el rango de direcciones IP que se desea escanear, la ejecución de esta actividad se visualiza en la figura 27.

Figura 27 Ataque de Escaneo de direcciones IP



Fuente: Herramienta *IP Scanner*

El registro de la actividad del escaneo es almacenado en la bitácora del *Honeypots* y se visualiza en la figura 28

Figura 28 Respuesta de los Honeypots al tipo de ataque Escaneo de direcciones IP

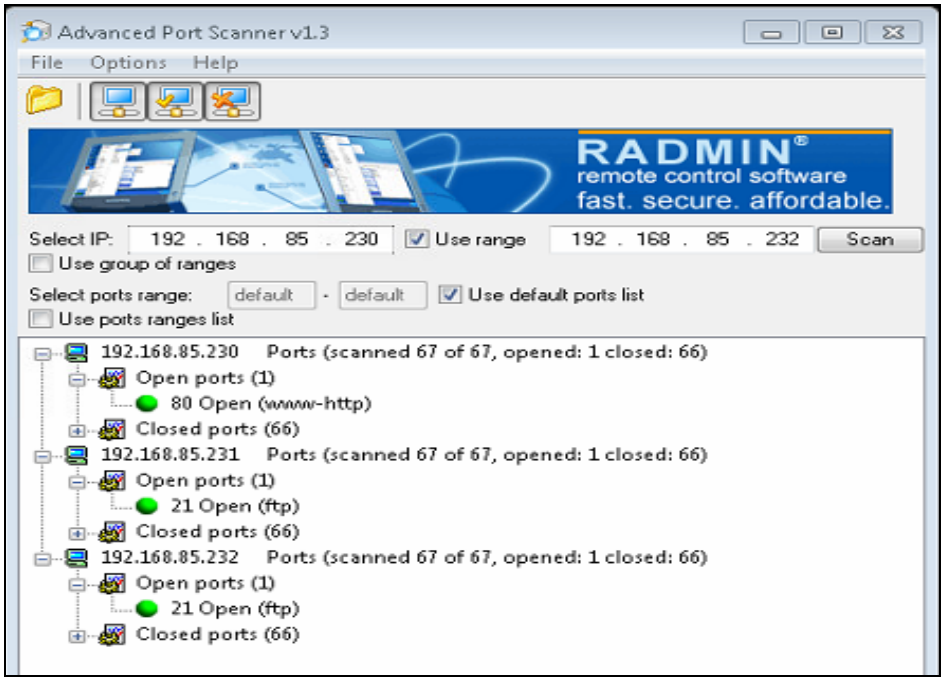
```
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.230 -> 192.168.85.235
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.232 -> 192.168.85.235
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.231 -> 192.168.85.235
```

Fuente: archivo alerta *honeyd*

Escaneo de puertos

La herramienta Advanced Port Scanner también permite hacer el análisis de los puertos filtrados mediante un rango de direcciones IP. En la figura 29 se muestra la ejecución de este ataque.

Figura 29 Ataque de Escaneo de Puertos con Advanced Port Scanner



Fuente: Herramienta *IP Scanner*

En la figura 30 se visualiza el registro que queda almacenado en la bitácora del *Honeypots* después de haber hecho este ataque.

Figura 30 Respuesta de los Honeypots al tipo de ataque Escaneo de Puertos

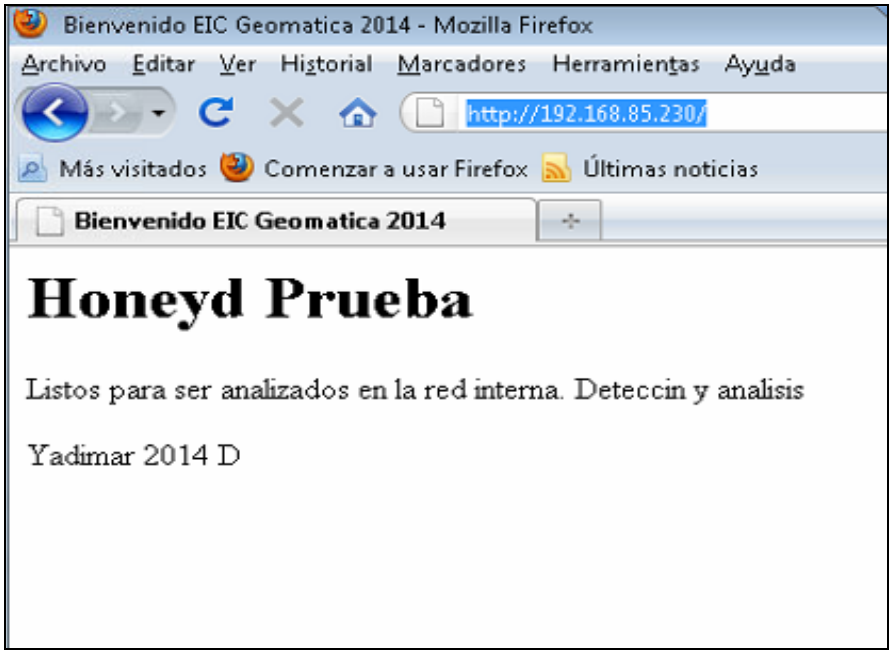
```
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.230 -> 192.168.85.235
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.232 -> 192.168.85.235
honeyd[3453]: Sending ICMP Echo Reply: 192.168.85.231 -> 192.168.85.235
```

Fuente: archivo alerta *honeyd*

Una vez realizado el escaneo de puertos, el *Honeypots* le presenta al atacante la siguiente información:

- La máquina trampa Windows XP que se encuentra en la dirección IP 192.168.85.230 presenta el puerto 80 abierto, esto quiere decir que existe un servicio web corriendo en este *Honeypots* y la validación de este servicio se hace a través de un navegador como se muestra en la figura 31.

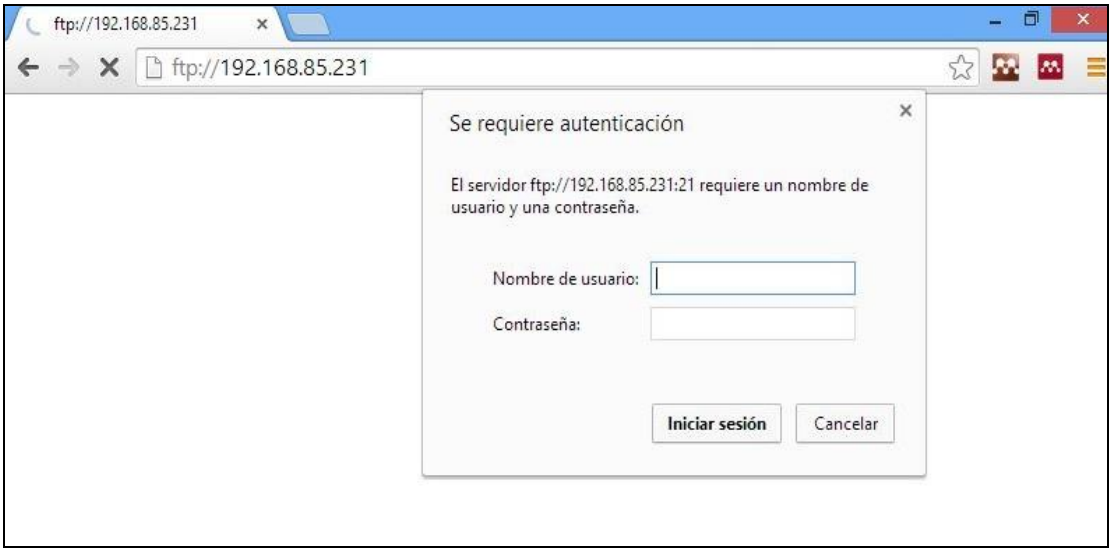
Figura 31 Validación del servicio web del Honeypots Windows XP



Fuente: archivo alerta *honeyd*

- La máquina trampa Windows Server 2003 que se encuentra en la dirección IP 192.168.85.231 presenta el puerto 21 abierto, esto quiere decir que existe un servicio ftp corriendo en este *Honeypots* y la validación de este servicio se hace a través de un navegador como se muestra en la figura 32.

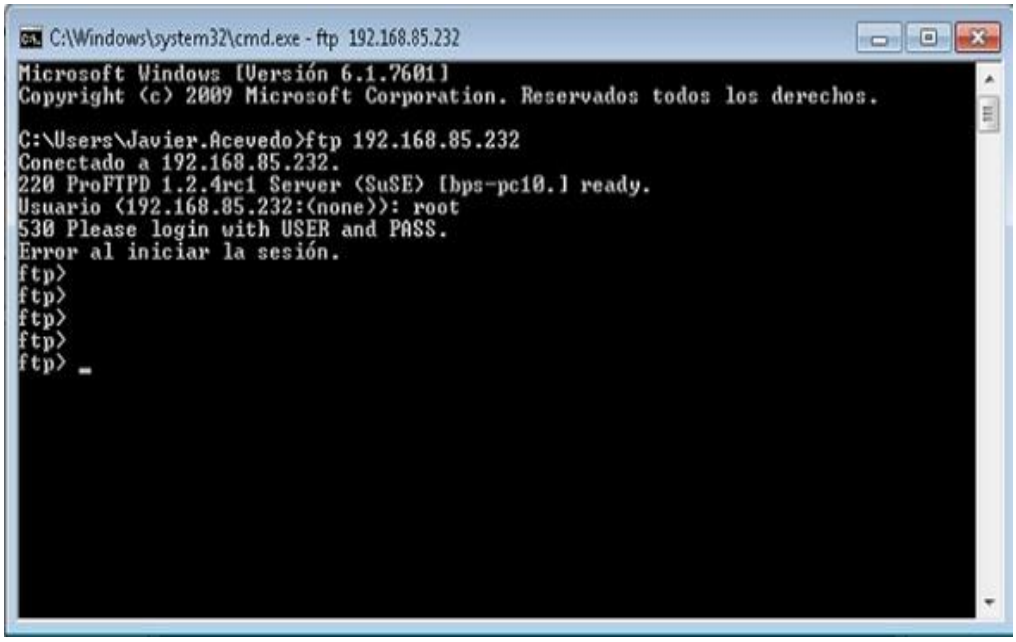
Figura 32 Validación del servicio ftp en el Honeypots Windows Server 2003



Fuente: archivo alerta *honeyd*

- La máquina trampa Linux Suse que se encuentra en la dirección IP 192.168.85.232 presenta el puerto 21 abierto, esto quiere decir que existe un servicio ftp corriendo en este *Honeypots* y la validación de este servicio se hace a través de un navegador como se muestra en la figura 33.

Figura 33 Validación del servicio FTP en el Honeypots Linux Suse



Fuente: *cmd.exe Windows* para la ejecución del *telnet*

6 CONCLUSIONES

Como resultado del diseño del sistema de detección de intrusos *Snort* y el *Honeypots* de baja interacción que se planteó para el grupo de investigación Geomática se han obtenido las siguientes conclusiones:

Análisis de tráfico

- Mediante el análisis de tráfico de red se logró detectar cuáles protocolos y puertos son los que más se usan en los servidores del grupo, los cuales sirvieron de guía para la implementación de las reglas de prueba en el sistema *Snort*.

***Snort*:**

- La ventaja de poseer versiones para diferentes sistemas operativos, la existencia de una buena documentación para la instalación y configuración, hace que esta herramienta sea una de las más populares dentro de los sistemas de detección de intrusos.

- La característica de modularidad que tiene el sistema *Snort*, es decir, que cada componente se pueda configurar por separado mediante una serie de archivos, como el archivo de captura de paquetes o el de firmas, le permite al administrador de red escalar en sus sistemas de seguridad o adaptarlos ante cualquier cambio en el entorno de su red.

- La flexibilidad en la creación de reglas es una de las ventajas que posee este sistema de detección basado en firmas, ya que permite configurar el *Snort* de acuerdo con las necesidades particulares de cada organización, como en el caso de este proyecto, que se crearon firmas basadas en el análisis de tráfico de red de los servidores y que después fueron probadas en la etapa de validación.

- El diseño del *Snort* planteado en este proyecto sirve como base para la implementación de un completo sistema de detección de intrusos para la red del grupo Geomática, debido a que no cuenta con un sistema de seguridad más que los que ofrecen los firewalls del sistema operativo de sus servidores.

***Honeypots*:**

- La herramienta *Honeypots* representa una solución de bajo costo debido a que es de código abierto y que no necesita de un equipo dedicado para su funcionamiento ya que se puede instalar sobre una máquina virtual.

- Una de las ventajas importantes que tiene el *Honeypots* es la de servir como señuelo para atraer a los atacantes y de esta manera desviar la atención sobre otros equipos en la subred de producción del grupo, con esta información recolectada se puede tratar de hacer un perfil del atacante con el fin de conocer cuál es el objetivo que pretende alcanzar.

Generales:

- El objetivo de plantear una combinación entre el *Snort* y el *Honeypots* se hizo con el fin de que fueran complementarias, es decir, mediante el análisis de tráfico de red se detectaron algunas vulnerabilidades que luego fueron implementadas como reglas en el *Snort*, pero existen otras vulnerabilidades que no las detectó el analizador, para eso se usó el *Honeypots*, para detectarlas y poderlas implementar también en el archivo de reglas del *Snort*, como en el caso de la validación de la regla que se disparaba debido a la bandera de cierre de conexión en el protocolo UDP.

- La propuesta de este proyecto de combinar una herramienta de detección de intrusos como el *Snort* con un sistema de trampas como el *Honeypots*, es un primer paso hacia los objetivos que el grupo Geomática quiere alcanzar con respecto a la seguridad de su información que últimamente ha sido objeto de ataques especialmente las de tipo de uso indebido.

7 RECOMENDACIONES

Una vez realizado el análisis de la red, el diseño y las pruebas piloto de cada una de las herramientas como el *Snort* y el *Honeypots* de baja interacción se identificaron algunas recomendaciones importantes como:

- Las reglas que fueron implementadas para la prueba piloto se realizaron con éxito debido a la facilidad de sintaxis que posee, sin embargo, al momento de la implementación del sistema, es necesario realizar un análisis del tráfico de red más profundo y en un mayor lapso de tiempo con el fin de detectar todas las posibles vulnerabilidades que pueda tener y así poder generar un conjunto de reglas mucho más completo.

- Para el mejoramiento de la seguridad en la red de grupo Geomática en cuanto a su infraestructura física, se hace necesario instalar un firewall físico con el fin de controlar el tráfico entre los equipos de la subred de producción y los servidores.

- Con el fin de implementar buenas prácticas en el manejo de las subredes, se recomienda separar la subred de producción de tal manera que en una subred queden solamente los equipos de las salas de informática I y II y en la otra queden solamente los equipos que están destinados para los proyectos de investigación y extensión del grupo (sala SIG, sala de Auxiliares y Salas Geomática I y II).

- Para un mejor desempeño del *Honeypots* sobre las direcciones IP que están inactivas en la subred de producción, se hace necesario realizar una reasignación de direcciones IP de manera consecutiva a todos los equipos de la red, de tal manera que el grupo de direcciones IP que no se usen queden también dentro de un rango consecutivo.

REFERENCIAS BIBLIOGRAFICAS

[1] A. A. Ramírez García, “Estudio de una plataforma de detección de intrusos Open Source,” Escuela Técnica de Ingenieria de Telecomunicaciones de Barcelona, 2009.

[2] C. Riley, “Seguridad *Snort*,” *Linux Magazine*, Madrid, España, pp. 60–66, 17-Nov-2009.

[3] D. S. Suárez Romero, “Implementación de *Snort* como sistema de detección de instrusos en una red,” Universidad Francisco de Paula Santander, 2014.

[4] J. I. Avilés Monroy and M. R. Pazmiño Castro, “Captura y análisis de los ataques informáticos que sufren las redes de datos de la ESPOL, implantando una Honeynet con miras a mejorar la seguridad informática en redes de datos del Ecuador,” Escuela Superior Politécnica del Litoral, 2009.

[5] P. A. Mora, “*Honeypotss*,” Escuela Politécnica del Ejército, 2006.

[6] D. Monkey.org, “Honeyd.” [Online]. Available: http://www.honeyd.org/. [Accessed: 22-Aug-2014].

[7] N. Mathewson and N. Provos, “Libevent.” [Online]. Available: http://libevent.org. [Accessed: 15-Aug-2014].

[8] “Libdnet.” [Online]. Available: https://code.google.com/p/libdnet. [Accessed: 15-Aug-2014].

[9] “Tcpdump & Libpcap.” [Online]. Available: http://www.tcpdump.org. [Accessed: 25-Aug-2014].

[10] G. Lyon, “Nmap.” [Online]. Available: http://nmap.org. [Accessed: 20-Aug-2014].

[11] M. de la Herrán Brickmanne, “Detector de intrusiones ligero para redes.,” Madrid, España, 2010.

[12] J. García Alfaro, “Detección de ataques en red con *Snort*,” Barcelona, España, 2003.

[13] M. I. Giménez García, “Utilización de Sistemas de Detección de Intrusos como Elemento de Seguridad Perimetral,” Universidad de Almería, 2008.

[14] L. C. Giralte, “Diseño de Sistemas Distribuidos de Detección de Anomalías de Red,” Madrid, España.

[15] P. Agarwal and S. Satapathy, “Implementation of Signature-based Detection System using *Snort* in Windows,” *Int. J. Innov. Adv. Comput. Sci.*, vol. 3, no. 3, pp. 120–127, 2014.

[16] J. A. Rojas and H. C. Manta, “Sistema dectores de intrusos y análisis de funcionamento del proyecto de código abierto *Snort*,” *Redes Ing.*, vol. 2, no. 1, pp. 100–112, 2011.

[17] M. Roesch, “*Snort*.” [Online]. Available: www.*Snort*.org. [Accessed: 20-Aug-2014].

[18] D. F. Torres García and P. S. Zambrano Núñez, “Implementación de un sistema de detección y análisis no autorizadas utilizando *Honeypotss* caso práctico DESITEL - ESPOCH,” Escuela Sueperior Politécnica de Chimborazo, 2011.

[19] J. C. Ángeles García, “Cómputo forense mediante la tecnología *Honeypots*,” Instituto Politécnico Nacional, 2010.

BIBLIOGRAFÍA

A. A. Ramírez García, “Estudio de una plataforma de detección de intrusos Open Source,” Escuela Técnica de Ingenieria de Telecomunicaciones de Barcelona, 2009.

C. Riley, “Seguridad Snort,” Linux Magazine, Madrid, España, pp. 60–66, 17-Nov-2009.

D. F. Torres García and P. S. Zambrano Núñez, “Implementación de un sistema de detección y análisis no autorizadas utilizando Honeypotss caso práctico DESITEL - ESPOCH,” Escuela Sueperior Politécnica de Chimborazo, 2011.

D. Monkey.org, “Honeyd.” [Online]. Available: http://www.honeyd.org/. [Accessed: 22-Aug-2014].

D. S. Suárez Romero, “Implementación de Snort como sistema de detección de instrusos en una red,” Universidad Francisco de Paula Santander, 2014.

G. Lyon, “Nmap.” [Online]. Available: http://nmap.org. [Accessed: 20-Aug-2014].

J. A. Rojas and H. C. Manta, “Sistema dectores de intrusos y análisis de funcionamento del proyecto de código abierto Snort,” Redes Ing., vol. 2, no. 1, pp. 100–112, 2011.

J. C. Ángeles García, “Cómputo forense mediante la tecnología Honeypots,” Instituto Politécnico Nacional, 2010.

J. García Alfaro, “Detección de ataques en red con Snort,” Barcelona, España, 2003.

J. I. Avilés Monroy and M. R. Pazmiño Castro, “Captura y análisis de los ataques informáticos que sufren las redes de datos de la ESPOL, implantando una Honeynet con miras a mejorar la seguridad informática en redes de datos del Ecuador,” Escuela Superior Politécnica del Litoral, 2009.

“L. C. Giralte, “Diseño de Sistemas Distribuidos de Detección de Anomalías de Red,” Madrid, España.

Libdnet.” [Online]. Available: https://code.google.com/p/libdnet. [Accessed: 15-Aug-2014].

M. de la Herrán Brickmanne, “Detector de intrusiones ligero para redes.,” Madrid, España, 2010.

M. I. Giménez García, “Utilización de Sistemas de Detección de Intrusos como Elemento de Seguridad Perimetral,” Universidad de Almería, 2008.

M. Roesch, “Snort.” [Online]. Available: www.Snort.org. [Accessed: 20-Aug-2014].

N. Mathewson and N. Provos, “Libevent.” [Online]. Available: http://libevent.org. [Accessed: 15-Aug-2014].

P. A. Mora, “Honeypotss,” Escuela Politécnica del Ejército, 2006.

P. Agarwal and S. Satapathy, “Implementation of Signature-based Detection System using Snort in Windows,” Int. J. Innov. Adv. Comput. Sci., vol. 3, no. 3, pp. 120–127, 2014.

“Tcpdump & Libpcap.” [Online]. Available: http://www.tcpdump.org. [Accessed: 25-Aug-2014].

ANEXOS

ANEXO A. GENERALIDADES DEL *SNORT*

De existir una coincidencia con alguna de esas reglas, el sistema disparará una acción predeterminada que depende de la configuración de dicha regla.
El *Snort* tiene una característica importante es que puede operar de tres formas diferentes[11]:

- Como un sniffer: este sistema se puede comportar como una aspiradora de paquetes, mostrando en la consola y en tiempo real qué está ocurriendo con el tráfico de la red.
- Como un registrador de paquetes: que permite guardar en una bitácora los registros de las intrusiones con el fin de analizarlas posteriormente.
- Como un detector de intrusos basado en red: comúnmente llamados NIDS (del inglés Network Intrusion Detection System), cuya función es la de examinar todos los paquetes que recibe la interfaz de red del equipo. Cada paquete es analizado y si coincide con alguna de las reglas establecidas por el administrador, el sistema dispara una acción que depende de la configuración de la regla y que puede ser la anotación del tráfico o hacer una copia local de los paquetes recibidos o hacer sonar una alarma, entre otros.

A.1 Entorno

El *Snort* es un programa es de libre distribución escrito bajo la licencia GNU/GPL. Su código fuente está escrito en C y fue liberado con el fin de que pueda ser implementado en múltiples plataformas como Linux, Windows, Solaris, MacOS X, entre otros. Su pequeño tamaño, licencia abierta y código multiplataforma hace que tenga el calificativo de ligero o *lightweight* en comparación con los demás NIDS que existen [11].

A.2 Historia

Su primera versión fue desarrollada en 1998 por Marty Roesch bajo el nombre de APE que funcionaba solamente para el sistema operativo Linux y estaba basado en la librería Libcap que es la encargada de la captura de paquetes. En 1999 se lanza la versión bajo el nombre de *Snort* donde se incluía el analizador de firmas y la capacidad de trabajar en múltiples sistemas operativos. También surgió la

versión comercial del *Snort* que se llama Sourcefire con el fin de dar soporte a las empresas [12].

A.3 Funciones

Su función principal es la de monitorizar todo el tráfico de la red en búsqueda de cualquier tipo de intrusión. De igual manera, se puede implementar un motor de detección de ataques y barrido de puertos que le permite registrar, alertar y responder ante cualquier anomalía dependiendo de las reglas que tenga configuradas por parte del administrador. Otras funciones principales que el *Snort* realiza son:

- Coloca una interfaz de red a escuchar en modo promiscuo, es decir, está constantemente atendiendo tanto a los paquetes que llegan a la interfaz de red como a los que no le corresponde escuchar, para luego registrar en una bitácora dicho tráfico de forma similar al programa tcpdump [11].
- Con el comportamiento de la interfaz de red en modo promiscuo se utiliza también para analizar y depurar el tráfico y los protocolos de red. Con el apoyo de una base de reglas sofisticadas es posible detectar y alertar sobre el tráfico no deseado que esté circulando por la red.

A.4 Arquitectura

Como se mencionó anteriormente, *Snort* es un sistema que posee un conjunto de características que lo hacen una herramienta para la detección de intrusos muy potente, como la captura del tráfico de la red, el análisis y registro de los paquetes capturados y la detección de paquetes maliciosos.

El *Snort* está conformado por un conjunto de elementos, la mayoría de los cuales se pueden configurar de acuerdo a las necesidades de la red, entre los cuales se destacan el procesador, que es el que permite que *Snort* manejar de forma más eficiente el contenido de los paquetes antes de pasarlos al elemento de detección, y el sistema de notificaciones y alertas que permite definir todos los aspectos relacionados con el almacenamiento de la información recolectada.

A.5 Componentes del *Snort*

Los módulos que componen el sistema de detección de intrusos *Snort* son los siguientes:

A.5.1 Módulo de captura del tráfico

Este módulo consta de un dispositivo que puede ser un software o un hardware, el cual cumple la función de un sniffer, es decir, es el que se encarga de realizar la captura del tráfico que circula a través de la red, aprovechando al máximo los recursos de procesamiento y minimizando la pérdida de paquetes.
Esta captura de tráfico la lleva a cabo mediante la librería Libcap, esta librería tiene la característica de que es multiplataforma y su función principal consiste en la captura de *paquetes raw* que circula por una red a través de la interfaz de red. Un paquete raw es aquel que no tiene ningún tratamiento por parte de algún sistema, es decir, que mantiene su forma original tal y como ha viajado a través de la red. Este paquete contiene toda la información de cabecera del protocolo de salida intacta e inalterada por el sistema operativo [3].

La utilización de la librería Libcap no es el método más eficiente para la captura de paquetes, puesto que sólo puede tratar un paquete a la vez lo que se traduce en un cuello de botella en donde el ancho de banda de la red sea alto. Existen otros métodos de captura de paquetes de una interfaz de red como son: el filtro de paquetes Berkeley (BFP), el interfaz de proveedor de enlace de transmisión (DLP) y la herramienta SOCK_PACKET en el kernel de Linux[3].

A.5.2 Decodificador de paquetes

Este motor está establecido alrededor de las capas de pila de protocolos de Enlace de Datos y TCP/IP, se encarga de tomar los paquetes que recoge le Libcap y los guarda en una estructura de datos en la que se apoyan el resto de capas. Cada instrucción en el decodificador establece un orden sobre los datos del paquete añadiendo estructuras de datos sobre el tráfico de la red. *Snort* es capaz de decodificar los paquetes que pertenezcan a los protocolos de Ethernet, SLIP y PPP [3].

Una vez que se capture el paquete, el *Snort* mediante una serie de paquetes de decodificadores, en los cuales cada uno de ellos se encarga de descifrar los elementos de protocolos específicos. Trabaja sobre la pila de protocolos de red que comienza con el nivel más bajo que es el protocolo de Enlace de Datos y así sucesivamente descifra cada protocolo conforme asciende en la pila de protocolos de red [13].

A.5.3 Preprocesadores

Son componentes que se encargan de tomar la información que viaja a través de la red en forma de paquetes individuales y desordenados. Después de leer todo el tráfico de la red e interpretarlo es capaz de tener el control sobre todos los paquetes con el fin de poderlos ordenar y darle sentido esa información que llega a la interfaz de red.
Estos preprocesadores fueron incluidos desde la versión 1.5 y no dependen reglas ya que el conocimiento sobre la intrusión depende del módulo del preprocesador, y se ejecutan siempre que llegue un paquete sin tratar o paquete raw que los verifica mediante un conjunto de programas escritos en C o *plug-ins* específicos como por ejemplo el plug-in para llamadas RPC o el plug-in de escaneo de puertos, entre otros. Estos programas verifican los paquetes en busca de ciertos comportamientos con el fin de determinar su tipo, para luego ser enviado al motor de detección.

Esta característica de utilizar diferentes plug-in para cada comportamiento hace que el *Snort* sea una de las herramientas de detección con mejor escalabilidad, debido a que estos preprocesadores se pueden activar o desactivar a través de su archivo de configuración, sin embargo, *Snort* activa automáticamente varios preprocesadores para [2]:

- El tratamiento y recomposición del tráfico fragmentado.
- La inspección del flujo con control de estado.
- La monitorización el rendimiento.
- La decodificación del tráfico RPC.
- La monitorización y recomposición de las cadenas de caracteres enviadas a través de servicios como FTP, Telnet, SMTP, DNS, HTTP, POP3, IMAP y SMB, entre otros.
- Explorar el tráfico no sólo a nivel de red sino que también a nivel de la aplicación.
- El escaneo de puertos.

Incluso existe un preprocesador especialmente diseñado para el troyano *Back Orifice*. Cada preprocesador tiene su propio conjunto de opciones y configuraciones, las cuales vienen establecidas por defecto, pero si se quiere aprovechar al máximo la potencialidad del *Snort*, es necesario llevar a cabo una óptima configuración del preprocesador; por ejemplo, el preprocesador *sfPosrtscan* puede generar muchos falsos positivos a un NIDS si se configura de forma inadecuada [14].

En resumen, estos preprocesadores de *Snort* son pequeños programas escritos en C que toman decisiones de qué hacer con el paquete. Estos son compilados junto al *Snort* en forma de librería y son ejecutados después de que se realiza la decodificación para luego llamar al motor de detección. Si el número de preprocesadores es muy alto el rendimiento del *Snort* puede disminuir

considerablemente. En la Tabla 6 se visualiza algunos de los preprocesadores del *Snort* y su descripción [3].

Tabla 6. Preprocesadores del Snort

Preprocesador	Descripción
stream4	Proporciona un flujo de ensamblado TCP y capacidades de análisis para rastrear hasta 100.000 conexiones simultáneas.
stream5	Módulo de reensamblado que permite rastrear comunicaciones TCP y UDP.
sfportscan	Desarrollado por Sourcefire para detectar el primer paso de un ataque: el escaneo de puertos.
rpc_decode	Analiza múltiples registros RPC fragmentados en un único registro.
performance monitor	Mide el tiempo real del funcionamiento del *Snort*.
smtp	Es un decodificador SMTP para los clientes de correo electrónico.
http_inspect	Es un decodificador genérico para analizar el tráfico http. Analiza respuestas tanto de los clientes como de los servidores.
ftp/telnet	Permite decodificar el tráfico ftp y telnet para buscar cualquier actividad anormal.
ssh	Permite analizar el tráfico ssh de los clientes y servidores.
dce/rpc	Analiza el tráfico SMB que es el encargado de compartir archivos y carpetas en Windows.
dns	Analiza el tráfico de DNS para detectar diferentes tipos de ataques.

A.5.4 Motor de detección

El motor de detección es el núcleo del *Snort* en su versión en modo de detección de intrusos o IDS (sigla del inglés Intrusion Detection System). Su función principal es la detectar cualquier actividad de intrusión existente en un paquete basándose en el archivo de reglas del *Snort*. Las reglas son leídas desde una estructura de datos interna o archivos de textos y comparadas con cada paquete. Si existe una correlación con alguna de estas reglas, el motor de detección se encargará de avisar al sistema de alertas indicando la regla que se ha saltado, con el fin de que se ejecute la acción que la regla tenga configurada que se debe llevar a cabo. Si

por el contrario no existe alguna coincidencia con alguna regla, este paquete será descartado.

Es en el motor de detección en donde se consume la mayor cantidad de tiempo en la ejecución del *Snort*, los factores que influyen en los tiempos de respuestas y en la carga del motor de detección son [3]:

- Los recursos de hardware que posea el equipo.
- La definición de las reglas.
- La velocidad interna del bus en el equipo *Snort*.
- La cantidad de tráfico en la red.

Por ejemplo, si el tráfico de la red es demasiado alto en el momento en que está funcionando el *Snort* de tipo NIDS, se pueden descartar paquetes y no se conseguirá respuestas en tiempo real.

El motor de detección puede aplicar las reglas en diferentes partes del paquete. La estructura en la que está compuesta un paquete es la siguiente [3]:

- La cabecera IP: el motor de detección se pueden aplicar reglas de que correspondan a esta parte.
- La cabecera de la capa de Transporte: aquí se incluyen las cabeceras TCP, UDP e ICMP.
- La cabecera de la capa de Aplicación: incluye las cabeceras DNS, FTP, SNMP y SMPT.
- Payload del paquete: se puede aplicar una regla para que el motor de detección use para encontrar una cadena dentro del paquete.

A.5.5 Archivos de reglas

Las reglas son una serie de firmas que se usan como patrones que se buscan dentro de los paquetes que son analizados por el motor de detección del *Snort*. Una regla está divida en dos partes: la cabecera o encabezado y las opciones.

- **Cabecera de una regla:** en donde se establece el origen y destino de la comunicación con el fin de realizar una acción determinada, es decir, es donde se indica qué acción debe ejecutar en caso de que se cumpla dicha regla. Estas acciones pueden ser la generación de un archivo o la generación de una alerta. En esta parte también se le configura el tipo de paquete (TCP, UDP, ICMP) y la dirección de origen y destino del paquete, entre otros.

En la tabla Tabla 7 se muestra un ejemplo de la estructura del encabezado de una regla y la definición de cada uno de los componentes son [15],[16]:

Tabla 7. Estructura del Encabezado de una regla

Acción	Protocolo	Red origen	Puerto origen	Dirección	Red destino	Puerto destino
alert	tcp	$EXTERNAL_NET	Any	→	$HOME_NET	53

- Acción: permite indicar la acción que se va a realizar sobre dicho paquete. Los posibles valores son:
  - *alert*: genera una alerta usando el método de alerta seleccionado.
  - *log*: registra el paquete.
  - *pass*: ignora el paquete.
  - *activate*: alerta y luego activa una regla dinámica.
  - *dynamic:* permanece inactivo hasta que se active por una regla, es decir, actúa como inspector de reglas.
- Protocolo: establece el protocolo de comunicaciones que se va a utilizar. Los posibles valores son: TCP, UDP, IP e ICMP.
- Red origen: es la dirección de red desde la cual se generó el paquete.
- Puerto origen: es el puerto en el cual se generó el paquete.
- Dirección: establece el sentido de la comunicación. Las posibles opciones son: <-, ->, <>.
- Red destino: es el destino al cual llega el paquete, o sea la red atacada. Define la dirección de red del usuario.
- Puerto destino: es el puerto al cual llega el paquete.

- **Opciones de una regla:** es el núcleo del motor de detección, combina la facilidad de uso con poder y flexibilidad. Todas las opciones de las reglas están separadas entre sí por un punto y coma (;). Las palabras clave se separan de sus argumentos con dos puntos (:). Un ejemplo de una opción de regla se muestra a continuación [16]:

  (*msg:"ICMP Echo Reply"; icode:0; itype:0; classtype:misc-activity; sid:408; rev:5;*)

El código anterior corresponde a una regla para detectar un Ping realizado a alguno de los equipos de la red. Genera el mensaje: "**ICMP Echo Reply"**, cuando

el paquete tienen **icode:0; itype:0; classtype:misc-activity;** corresponde a la categorización de *Snort* para el tipo de ataque correspondiente a un Ping; **sid:408** se refiere al número de identificación de la regla *Snort* y **revit:5** es el número de revisiones que ha tenido la regla.
Hay cuatro categorías principales en las opciones de una regla:

General o Metadata: proporciona información de la regla, pero no tiene ningún efecto en la detección. En la Tabla 8 se detallan las opciones generales [16].

Tabla 8. Opciones Generales de las reglas de Snort

Palabra clave	Descripción
msg	Es el mensaje que se mostrará al producir una alerta.
reference	Permite incluir referencias de la intrusión, para conocer el tipo.
gid	Para identificar qué parte del *Snort* genera el evento en una regla particular.
sid	Identificador único de la regla usada por el *Snort*.
rev	Para conocer el número de revisiones de la regla.
classtype	Para categorizar el tipo de ataque en una regla.
priority	Indica el nivel de prioridad de la regla utilizado por *Snort*.
metadata	Permite incluir información adicional sobre la regla.

- Payload: buscan concordancia o patrones dentro de una carga útil del paquete y pueden ser interrelacionados. En la Tabla 9 se relacionan algunas opciones payload [16].

Tabla 9. Opciones *Payload* de las reglas de *Snort*

Palabra Clave	Descripción
content	Establece el contenido específico que se busca en la carga útil de un paquete.

rawbytes	Permite ver el paquete ignorando cualquier decodificación hecha por le preprocesador.
depth	Especifica los primeros bytes de la carga útil en que se deben buscar coincidencias con el patrón content.
offset	Especifica dónde empieza la búsqueda en un patrón within.
distance	Indica a partir de qué punto debe empezar a buscar el patrón de coincidencia del paquete.
within	Es un modificador de contenido que asegura que la mayoría de N bytes coinciden con el patrón content.
unicontent	Indica una dirección URL dentro del paquete.
isdataat	Especifica que la carga útil tiene datos en una ubicación específica.
pcre	Permite a las reglas tener la compatibilidad con Perl.
byte_test	Revisa un byte especificado por un valor.
byte_jump	Permite leer una porción de datos y saltar una posición adelante.
ftpbounce	Detecta ataques FTP.
asnl	Decodifica una porción del paquete y busca síntomas maliciosos.
cvs	Detecta cadenas de texto inválidas.

- Non-payload: buscan que no existan concordancias dentro de la carga útli o buscan patrones dentro de los demás campos del paquete que no sean de carga útil. En la Tabla 10 se especifican algunas opciones de este tipo [16].

Tabla 10. Opciones non-*Payload* de las reglas de *Snort*

Palabra clave	Descripción
fragoffset	Compara el campo offset del fragmento IP con un valor decimal.
til	Usado para verificar el valor TT1.

tos	Verifica el campo IP TOS para un valor específico.
id	Verifica el campo IP ID para un valor específico.
ipopts	Verifica que una opción específica esté presente en el paquete.
fragbits	Verifica si los bits de fragmentación y reservado están activados en la cabecera IP.
dsize	Confirma el tamaño de la carga útil del paquete.
flow	Aplica la regla solo en ciertas direcciones del flujo de tráfico.
flowbits	Conoce el estado a través de la sesión del protocolo de transporte.
Seq	Especifica el número de secuencia TCP.
ack	Especifica el número ACK TCP.
windows	Especifica el tamaño de la ventana TCP.
itype	Especifica el campo type de ICMP.
icode	Especifica el campo code de ICMP.
icmp_id	Especifica el campo Id de ICMP.
icmp_seq	Especifica el campo Seq de ICMP.
ip_proto	Verifica la cabecera del protocolo IP
sameip	Verifica si la dirección IP origen es igua a la dirección IP destino.

- Post-detection: especifican la acción a realizar después de que una alerta fue detectada. En la Tabla 11 se relacionan la mayoría de las opciones de este tipo [16].

Tabla 11. Opciones de *Post-detection* de las reglas del *Snort*

Palabra clave	Descripción
logto	Registra todos los paquetes en un archivo de registro.
session	Extrae los datos de usuario de la sesión TCP.
resp	Para cerrar las sesiones cuando se activa una alerta.
react	Permite la posibilidad de reaccionar ante una coincidencia de tráfico con una regla, por medio del cierre de conexión y el envío de un aviso.

tag	Permite a las reglas registrar más paquetes que simplemente el que lanza la alerta.
activates	Permite especificar una regla para agregar, cuando una red específica produce el evento.
activated_by	Especifica dinámicamente una regla cuando la regla activate es lanzada.
count	Permite detallar cuántos paquetes han sido habilitados después que la regla es activada. Debe usarse con activated_by.

En resumen, un ejemplo de una regla sería:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 53 \
(msg:"DNS EXPLOIT named 8.2->8.2.1"; flow:to_server.estabilished; \
content:"../../../"; reference:bugtraq,788; reference:cve,1999-0833; \
classtype:attemped-admin; sid:258; rev:6; )
```

La regla anterior genera la alerta *DNS_EXPLOIT named 8.2 -> 8.2.1* cuando en una comunicación establecida con el servidor DNS se encuentre el texto *"../../../"*. Los campos *reference* permiten indicar el tipo de vulnerabilidad que se ha detectado.

Existen cuatro categorías de reglas para evaluar un paquete. Estas categorías están divididas a su vez en dos grupos, las que tienen contenido y las que no. Las categorías son las siguientes [14]:

- **Reglas de protocolo:** son aquellas que son dependientes del protocolo que se está analizando, por ejemplo, en el protocolo Http está la palabra reservada *uricontent*.
- **Reglas de contenido genéricas:** estas reglas permiten especificar patrones para buscar en el campo de datos del paquete, los patrones de búsqueda que pueden ser binarios o en modo ASCII. Este tipo de reglas son muy útiles para cuando se buscan los *exploits* ya que comúnmente terminan en cadenas de tipo *"/bin/sh"*.
- **Reglas de paquetes malformados:** este tipo regla describe características sobre los paquetes, específicamente sobre sus cabeceras, las cuales indican que si se están produciendo o no algún tipo de anomalía. Estas reglas no

examinan el contenido, ya que primero revisan las cabeceras en busca de incoherencias u otro tipo de anomalía

- **Reglas IP:** estas reglas se aplican directamente sobre la capa IP y son comprobadas para cada datagrama IP. Luego, si el datagrama pertenece al protocolo TCP, UDP o ICMP entonces se realiza un análisis del datagrama con su correspondiente capa de protocolo.

A.5.6 Sistema de alertas e informes

El sistema de alertas está compuesto por módulos de salida o plug-ins, en donde cada uno de ellos hace diferente tipo de operación dependiendo de qué tratamiento se le debe dar a la salida generada por el sistema de loggin y alerta del *Snort*. El módulo de salida que utilice el *Snort* depende del formato en que se deseen los datos que van desde archivos de registros o log hasta información de forma estructurada para algún motor de base de datos como MySQL o Postgres.

En la Tabla 12 se muestra los diferentes tipos de módulos de salida que se ejecutan en *Snort* [3]:

Tabla 12. Módulos de salida del *Snort*

Módulo	Descripción
syslog	Envía alarmas al syslog.
alert_fast	Modo de alerta rápida el cual retorna información acerca del tiempo, mensaje de alerta, clasificación, prioridad de la alerta, IP, puerto de origen y destino.
alert_smb	Permite a *Snort* realizar llamadas al cliente de SMB y enviar mensajes de alerta a hosts de Windows.
alert_unixsock	Envía las alertas a través de un socket, para que las escuche otra aplicación.
log_tcpdump	Asocia los paquetes a un archivo con formato tcpdump.
database	Admite cuatro tipos de salida a base de datos: MySQL, PostgreSQL, Oracle y unixODBC.
csv	El módulo de salida csv permite escribir datos de alerta en un formato que se pueda importar fácilmente a una base de datos.
unified	Es el formato binario básico para registrar los datos para exportarlos a otros programas.
log_Null	Para provocar alertas sobre ciertos tipos de tráfico pero que no sean registradas.
Eventlog	Registra las alertas para mostrarse a través de un visor de

	sucesos en Windows.

A.6 Instalación del *Snort*

Para llevar a cabo la instalación del *Snort* se hizo sobre el equipo destinado para hacer las pruebas el cual tiene instalado un sistema operativo Windows 7 Pro, es necesario descargar la herramienta Winpcap que es la librería de Libpcap de captura de paquetes para entornos Windows. En la figura 34 se muestra la página oficial en donde se puede hacer la descarga.

Figura 34 Sitio oficial de Winpcap

Fuente: Sitio web www.winpcap.org

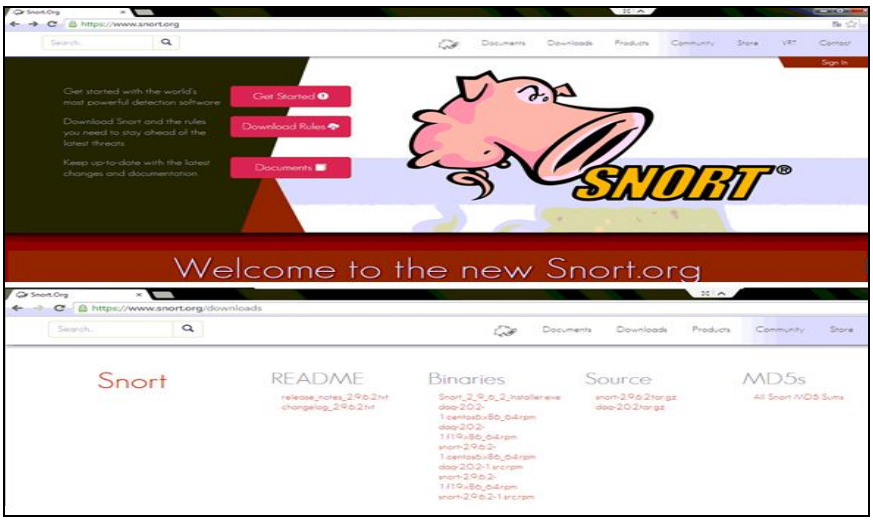
Luego, en la figura 35 se visualiza el inicio y el terminado de la instalación del Winpcap, que para realizar esta prueba se trabajó con la última versión 4.1.3

Figura 35 Instalación de la librería Winpcap

Fuente: Instalador WinPcap 4.1.3

Una vez instalado la librería de captura de paquetes para Windows, se procede a instalar el *Snort*, que se descarga de su sitio oficial en internet [17], tal como se muestra en la figura 36.

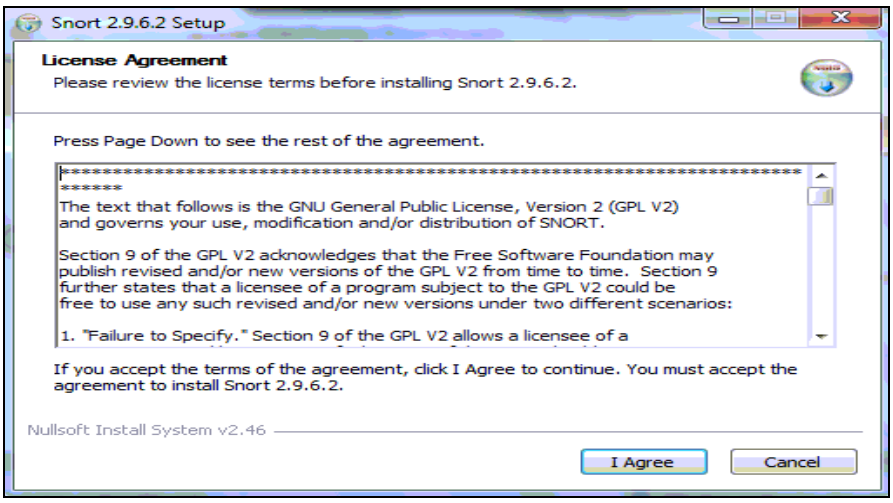
Figura 36 Sitio oficial del Snort



Fuente: *Sitio web www.snort.org*

En la figura 37 se muestra el inicio de la instalación del *Snort*, que para este caso se utilizó la versión 2.9.6.2.

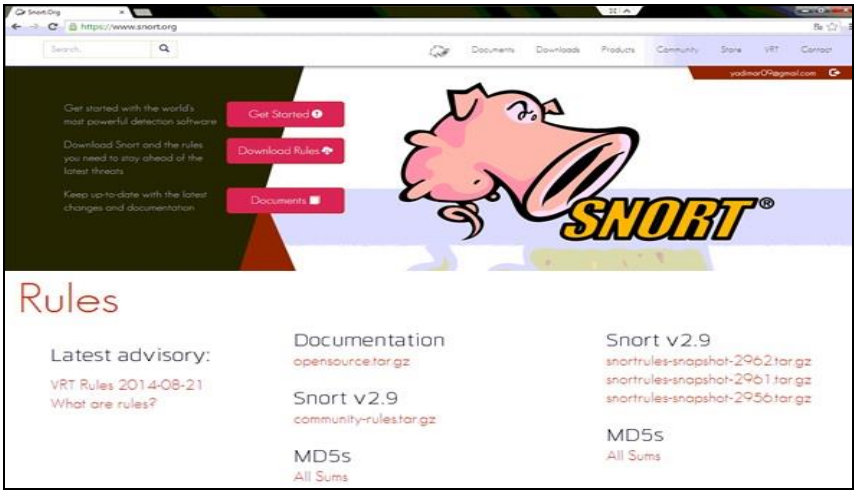
Figura 37 Instalación del Snort 2.9.6.2



Fuente: Instalador Snort

De igual manera, se lleva a cabo la descarga del archivo de reglas desde la página oficial del *Snort* [17], como se visualiza en la figura 38. Una vez se ha descargado este archivo, se agrega a la carpeta C:/*Snort*/Rules.

Figura 38 Descarga del archivo de Reglas del Snort

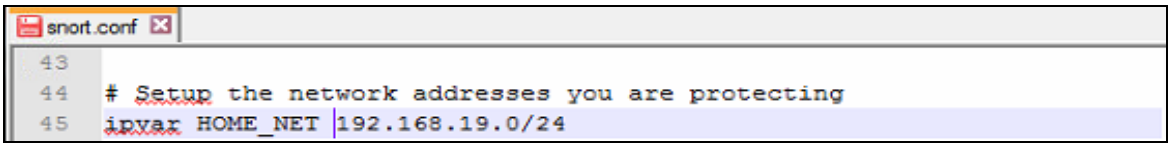


Fuente: *Sitio web www.snort.org*

A.7 Configuración del *Snort*

El primer paso consiste en modificar el archivo *Snort*.conf, el cual en la línea "ipvar" con el fin de indicar cuál es la red en donde se estará ejecutando el *Snort*, tal como se muestra en la figura 39.

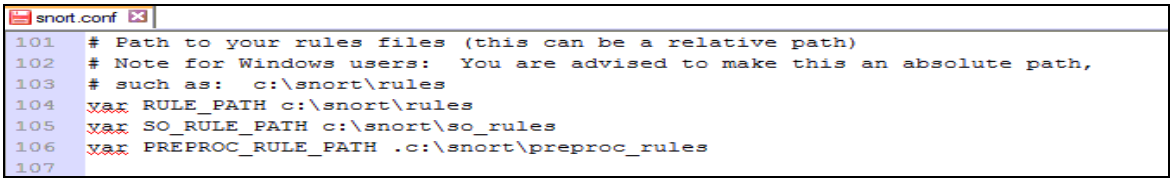
Figura 39 Configuración de Snort.conf – ipvar



Fuente: herramienta snort

Igualmente, se modifican las rutas de acceso del *Snort* para la ejecución correcta en Windows ya que por defecto vienen configuradas para Linux. Ver figura 40.

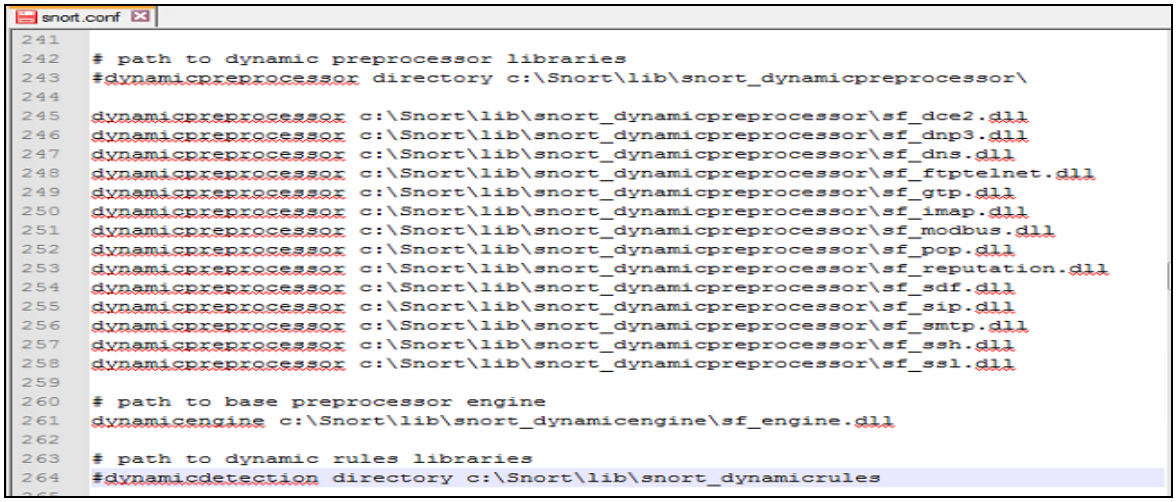
Figura 40 Configuración de Snort.conf - rutas de acceso



Fuente: herramienta snort

También se cambian las rutas para la librería del preprocesador dinámico, base del procesador y las reglas dinámicas, como se muestra en la figura 41.

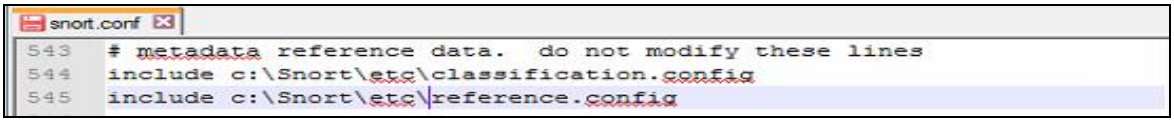
Figura 41 Configuración de Snort.conf – preprocesadores



Fuente: herramienta snort

La ruta de acceso a los metadatos, también se modifican de acuerdo con la ruta que haya quedado instalado en su equipo, ver la figura 42.

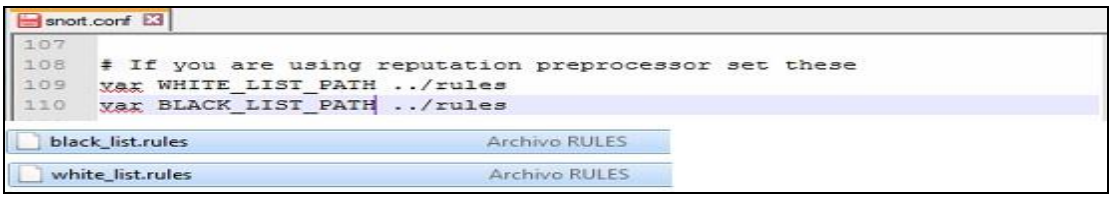
Figura 42 Configuración de Snort.conf – metadatos



Fuente: herramienta snort

Se necesitan dos archivos *white_list.rules* y *black_list.rules*, en el directorio de reglas existe uno de la siguiente forma *blacklist.rules*, este es modificado y se crea archivo *while_list.rules*, como se visualiza la figura 43.

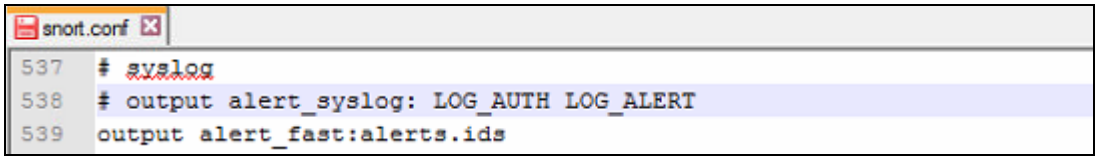
Figura 43 Configuración de Snort.conf - lista negra y blanca



Fuente: herramienta snort

Finalmente, se agrega la siguiente línea de comandos para configurar la salida de eventos como se muestra en la figura 44.

Figura 44 Configuración de Snort.conf – salidas



Fuente: herramienta snort

ANEXO B. GENERALIDADES DEL *HONEYPOTS*

En las máquinas que se usan como señuelo se coloca intencionalmente información sensible aunque falsa o se deja algún hueco de seguridad para que sea más atractivo para el atacante, que no dudará en exponer sus técnicas y métodos para llevar a cabo el ataque y es en ese momento donde el *Honeypots* se encarga de recolectar toda la información acerca del ataque.

Dentro de las funciones principales del *Honeypots* se destacan: la desviación de la atención del atacante de la red real, capturar nuevos virus o gusanos para llevar a cabo su análisis posterior, construir los perfiles de los atacantes y sus métodos, conocer las nuevas vulnerabilidades que tienen los sistemas que conforma la LAN, entre otros [5].

B.1 Clasificación

Existen diferentes tipos de clasificación de los *Honeypotss*, entre los que se destacan:

B.1.1 *Honeypotss* de Producción

Son usados para proteger la red y los equipos que conforman una organización en su ambiente real de operación, su objetivo principal es la de minimizar el riesgo de un ataque informático, por esta razón, se implementan de forma paralela a la red de datos con el fin de obtener una constante monitorización sobre la misma y así complementar de manera oportuna la actualización de los sistemas de detección en la red [18].

B.1.2 *Honeypotss* de Investigación

Este tipo de *Honeypots* no se implementan para proteger la red, sino por el contrario, su objetivo es el de exponer la red para recopilar todo tipo de información sobre los distintos atacantes o actividad maliciosa presente en la red, de tal manera que permita analizar nuevas herramientas de ataque, patrones de comportamiento y amenazas de todo tipo. Este *Honeypots* trabaja a partir de

servicios reales, además permite que el atacante tome el control de la máquina con el fin de poder estudiar su comportamiento[5].

B.1.3 *Honeypots* de baja interacción

Generalmente estos *Honeypotss* son implementados sobre máquinas virtuales para emular servicios informáticos como FTP o sistemas operativos. Se caracterizan por el poco grado de libertad que tiene el atacante con el sistema y por esta razón registran información limitada sobre el mismo, lo cual limita la capacidad de poder llevar a cabo un análisis completo del atacante. La ventaja que tienen estos sistemas es su simplicidad, ya que son fáciles de implementar y el grado de riesgo sobre la red es mínimo, por esta razón, son implementados en redes de producción [5].

Estos tipos de *Honeypots* son diseñados para capturar ataques con herramientas automatizadas en vez de atacantes reales, porque un intruso experimentado podría fácilmente detectar que no está en un entorno real y rápidamente abandonaría el ataque; mientras que, las herramientas automáticas de ataque están programadas para realizar una actividad específica como por ejemplo una prueba de escaneo o el análisis de tráfico malicioso en la red, por tal razón, no tienen la capacidad de sospechar nada extraño en el entorno en que se encuentra y de esta manera son “atrapadas” por este *Honeypots*.

Entre los *Honeypots* de baja interacción más comunes se encuentran: Honeyd, Nepenthes, Honeytrap y Tini *Honeypots* [4].

B.1.4 *Honeypots* de alta interacción

Este tipo de *Honeypots* también se implementa en redes de producción, en donde el grado de interacción del atacante con el sistema es total, por lo que se tiene la posibilidad de recolectar gran información acerca del atacante y sus técnicas.

Debido a que este tipo de *Honeypots* cuenta con servicios reales y sistemas operativos reales montados en hardware real, aumenta el riesgo de que un atacante pueda capturar estos sistemas para luego utilizarlos como herramientas de ataque a otros sistemas dentro de la red. Por esta razón, cualquier tipo de interacción con este *Honeypots* se considera sospechosa en principio y de esta manera logra prevenir al sistema de cualquier tipo de ataque [19].

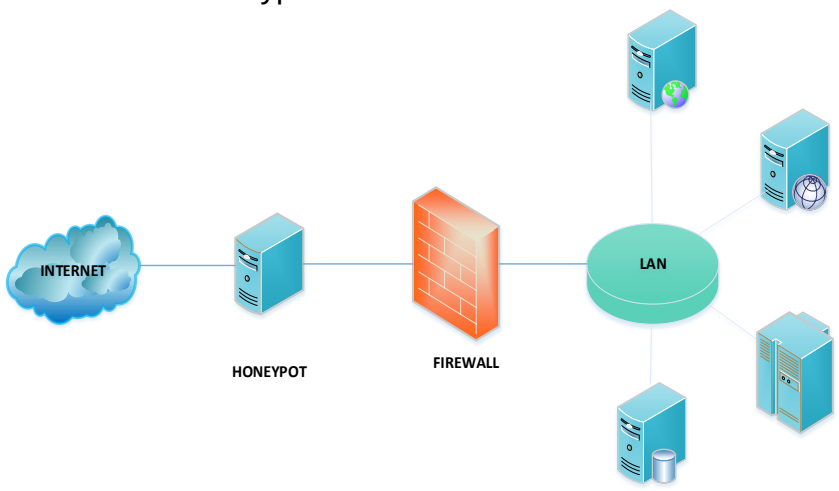
B.2 Ubicación de los *Honeypots*

Según el lugar dentro de la red donde se ubique el *Honeypots* permitirá explotar mejor su efectividad debido a su naturaleza pasiva. Por esta razón, hay que tener en cuenta que se debe ubicar en un lugar en donde no sea tan obvio de tal manera que un atacante experimentado lo descubra rápidamente, ni tampoco en un lugar de difícil acceso que haga que el atacante pierda su interés.

Los *Honeypotss* se utilizan para la detección de ataques que provienen del exterior de la red como por ejemplo, de internet o también de para prevenir ataques que provienen de la red interna o intranet. Por lo tanto, existen diferentes lugares dentro de la red donde se puede ubicar el *Honeypots*, todo dependerá del objetivo que se requiera. Los posibles lugares donde se puede ubicar el *Honeypots* son [5]:

- ***Antes del Firewall***: este tipo de ubicación permite reducir el riesgo de algún peligro sobre el resto de la red, tal como se muestra en la figura 45. Sin embargo, esta ubicación hará que exista mucho tráfico en la red debido a la facilidad que tiene de ser atacado y generará en gran consumo de ancho de banda. La desventaja de esta ubicación es que no puede detectar los ataques que se realizan al interior de la red.

Figura 45 Ubicación del Honeypots antes del Firewall

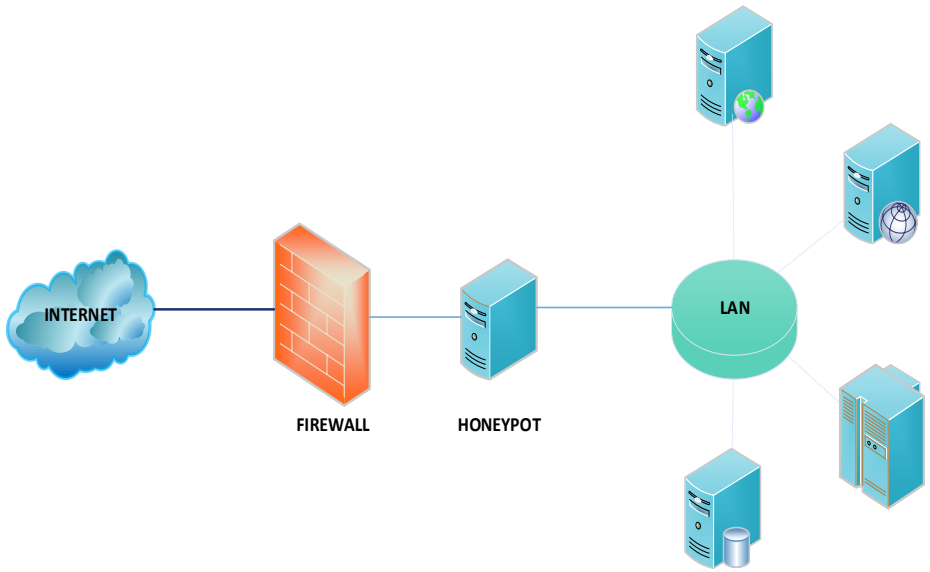


Fuente: Autor

- ***Detrás del Firewall*:** esta ubicación permite la detección de ataques internos en la red, así como el descubrimiento de configuración errónea del firewall

e incluso la detección de ataques que provienen del exterior, ver figura 46. La desventaja que tiene esta ubicación es que queda condicionado por las reglas de filtrado del firewall, y si por el contrario, se disminuye la seguridad en el firewall para que pueda ser accesible el *Honeypots*, se incrementará el riesgo de algún tipo de ataque sobre la red interna.

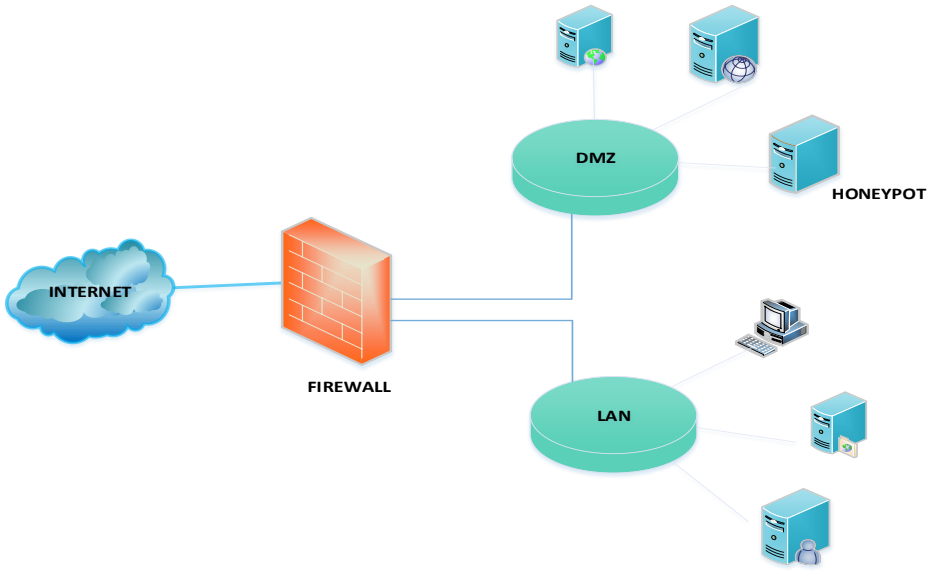
Figura 46 Ubicación del Honeypots después del Firewall



Fuente: Autor

- ***En la zona desmilitarizada*:** dentro de esta zona permite ubicar el *Honeypots* junto con los servidores de producción lo que permite aislarlo del resto de la red, ver figura 47. Esta ubicación permite al *Honeypots* detectar ataques externos e internos, aunque éste último no es del todo tan eficiente ya que no comparten el mismo segmento de red.

Figura 47 Ubicación del Honeypots en la Zona Desmilitarizada



Fuente: Autor

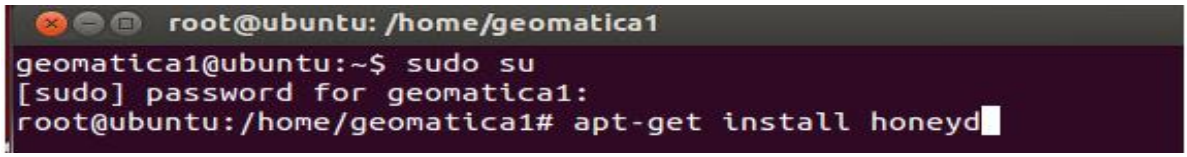
B.3 Instalación del *Honeypots* de baja interacción

Para llevar a cabo la instalación del Honeyd sobre la máquina virtual con sistema operativo Linux Ubuntu 13.10, es necesario descargar el instalador que aparece en la página oficial del Honeyd [6]. Luego, se debe abrir una consola o terminal para autenticarse como *superusuario* del sistema mediante los siguientes comandos:

geomatica1@ubuntu:~$ ***sudo su***
*[sudo] password for geomatica1: "**xxxxxx**"*
root@ubuntu: significa que ya se encuentra como "superusuario del sistema"

Una vez se ingresa como superusuario, se hace la instalación del Honeyd y las librerías necesarias para su correcto funcionamiento. Se inicia entonces con la instrucción que se muestra en la figura 48. Para la versión de este sistema operativo Linux (Ubuntu 13.10) el proceso de instalación del Honeyd es muy simple, es por esta razón fue que se escogió esta versión para hacer la prueba piloto.

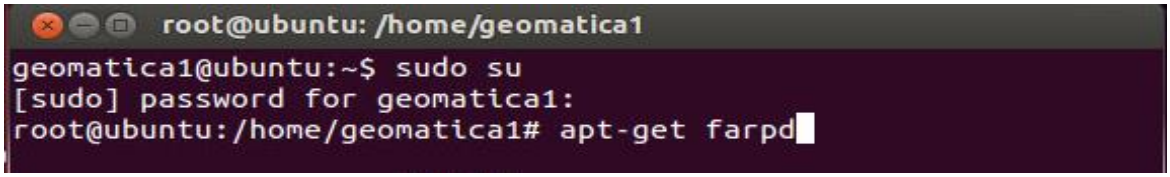
Figura 48 Instalación del Honeyd



Fuente: Herramienta *Honeyd*

Con esta instrucción anterior se inicia la descarga de los paquetes de instalación y librerías desde los repositorios de este sistema operativo. Además de la instalación del Honeyd, también se debe instalar la herramienta *farpd*, como se muestra en la figura 49 la cual es necesaria para reconocer el ARP Spoofing del rango de direcciones IP que se necesita que simule el Honeyd. De esta manera, cuando se inicie en modo activo estará escuchando en todo momento y cuando alguna máquina solicite una petición ARP que esté dentro de las IP configuradas, el *Honeypots* simulará ser un equipo activo y se le presentará al atacante como si fuera un equipo real.

Figura 49 Instrucción para la instalación de la herramienta *farpd*



Fuente: Herramienta *farpd*

Durante la instalación del Honeyd, también se configuran algunos parámetros que son los que se detallan a continuación:

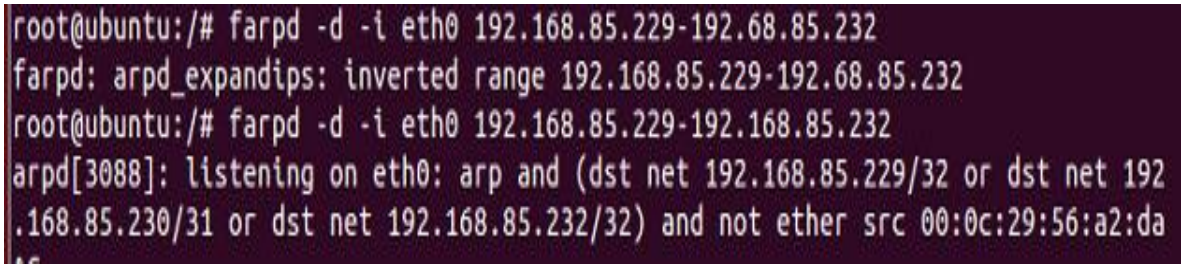
El parámetro *-d* muestra la habilitación de la información de las peticiones que escucha y acepta.

El parámetro *-i ethX* es la interfaz por la cual escucha la máquina donde se encuentra el *Honeypots*.

El parámetro *-"IP-IP"* es el rango de direcciones IP para configurar ARP Spoofing, que coinciden con el rango de IP que está configurado el Honeyd.

La ejecución del comando que se muestra en la figura 50 está indicando que permanece escuchando y enviando respuestas con la dirección MAC de su propia máquina.

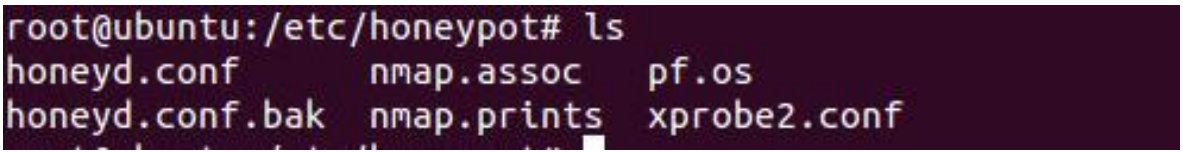
Figura 50 Instalación de la herramienta **farpd**



Fuente: Herramienta *farpd*

Después de la instalación se ingresa al directorio del *Honeypots* /etc/*Honeypots*/, tal como se muestra en la **¡Error! No se encuentra el origen de la referencia.** , en este directorio es donde se encuentran los archivo de firmas.

Figura 51. Archivos de firmas del *Honeypots*



Fuente: Herramienta *honeypot*

Los archivos de firmas que aparecen en el directorio del *Honeypots* son:

nmap: este archivo contiene las huellas digitales “fingerprints” utilizada por Honeyd para emular los diferentes sistemas operativos.

pf.os: este archivo ayuda a la identificación de sistema operativo de una máquina.

xprobe2: este archivo determina cómo el Honeyd reacciona ante huellas digitales de los paquetes ICMP.

honeyd.conf: es el archivo principal de la configuración del Honeyd, también se encuentra en el mismo directorio del *Honeypots* y es donde se encuentra la configuración de los sistemas operativos y servicios a emular. El detalle de este archivo se muestra en la figura 52.

La descripción de los comandos que aparecen dentro de este archivo es la siguiente:

- *create:* con esta sintaxis se le asigna un nombre a la máquina que se va a emular.
- *set win2k personality "Microsoft Windows Server 2003 Standart Edition"*: esta línea indica el sistema operativo que se va a emular.
- *set win2k default tcp/udp/icmp action open*: Esta sentencia indica qué acción se desea que tenga los paquetes TCP, UDP o ICMP y las banderas pueden ser open, block o reset.

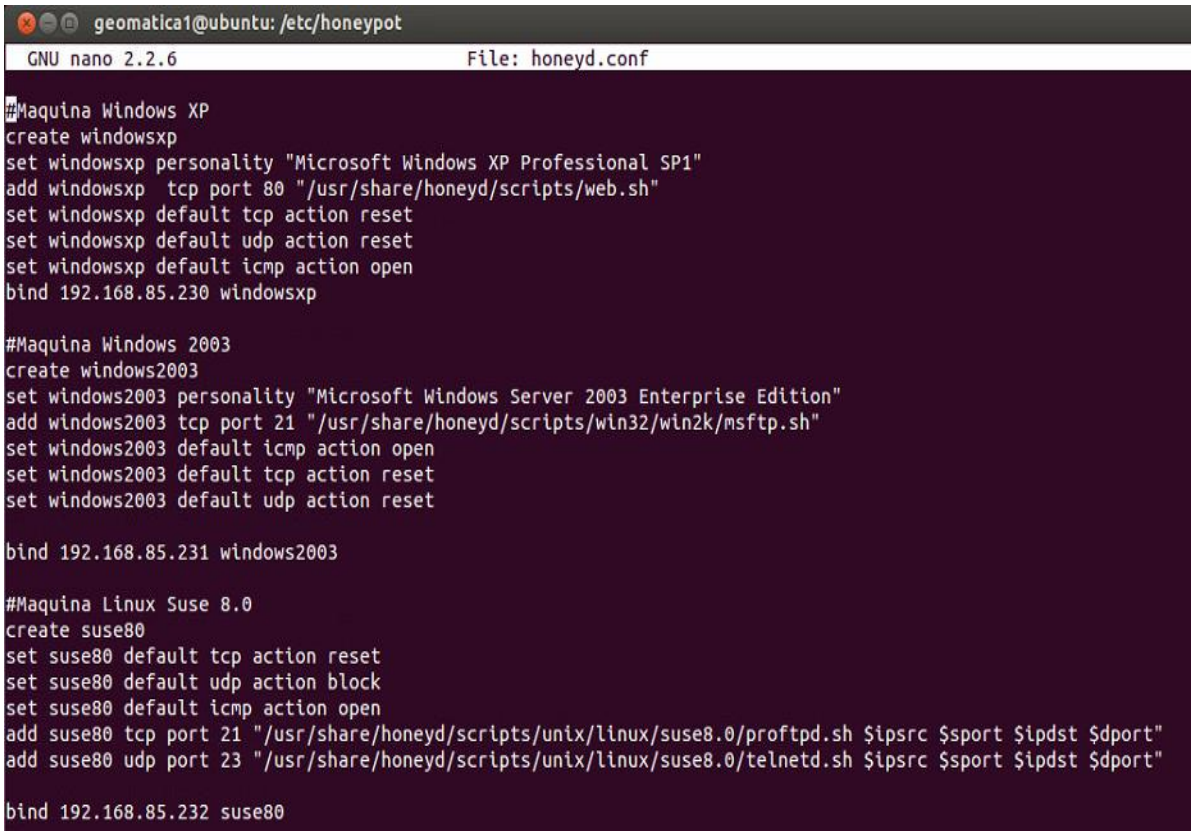
  En la Tabla 13 se visualizan las posibles respuestas que cada uno de los protocolos de la familia TCP/IP puede llevar a cabo frente a una actividad de Open, Block o Reset.

Tabla 13. Respuesta del puerto frente a una actividad

Puerto	Respuesta del Puerto		
	OPEN	**BLOCK**	**RESET**
TCP	Syn/Ack, si se establece la conexión	No responde o paquete perdido	Responde con RST
UDP	No hay respuesta	No responde o paquete perdido	Respuesta con puerto ICMP o envía mensaje de error
ICMP	Respuesta de paquete con actividad	No responde o paquete perdido	

- *add default default tcp port:* Este comando indica que si un atacante se conecta por el puerto 110, va a poder interactuar con el servicio mediante el protocolo tcp.
- *bind 192.168.85.229 win2k:* para este ejercicio es la dirección IP que identifica el sistema operativo que se muestra en la red con el fin de atraer a algún atacante.

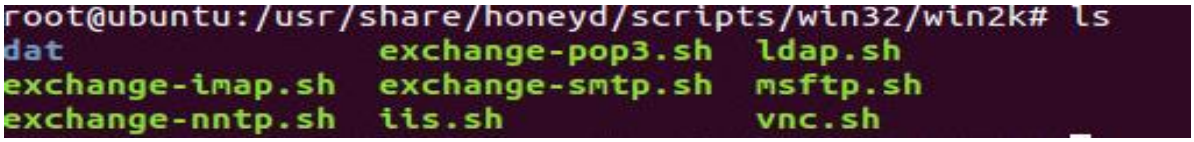
Figura 52 Archivo de configuración del Honeyd



Fuente: Archivo de configuración del *Honeyd*

También existe otro directorio importante que es donde se encuentran alojados los scripts de los diferentes servicios de la familia TCP/IP como el imap, nntp, pop3, smtp, iis, entre otros. En la figura 53 se visualiza la ruta donde se encuentran estos archivos.

Figura 53 Directorio donde se encuentran los servicios de la familia TCP/IP



Fuente: Herramienta *Honeyd*

De manera ilustrativa en la figura 54 se ve la estructura que contiene el script iis.sh que se encuentra en la ruta de la figura anterior.

Figura 54 Estructura del script iis.sh

Fuente: Archivo configuración *script ISS.SH de Honeyd*

B.4 Configuración del *Honeypots* de baja interacción

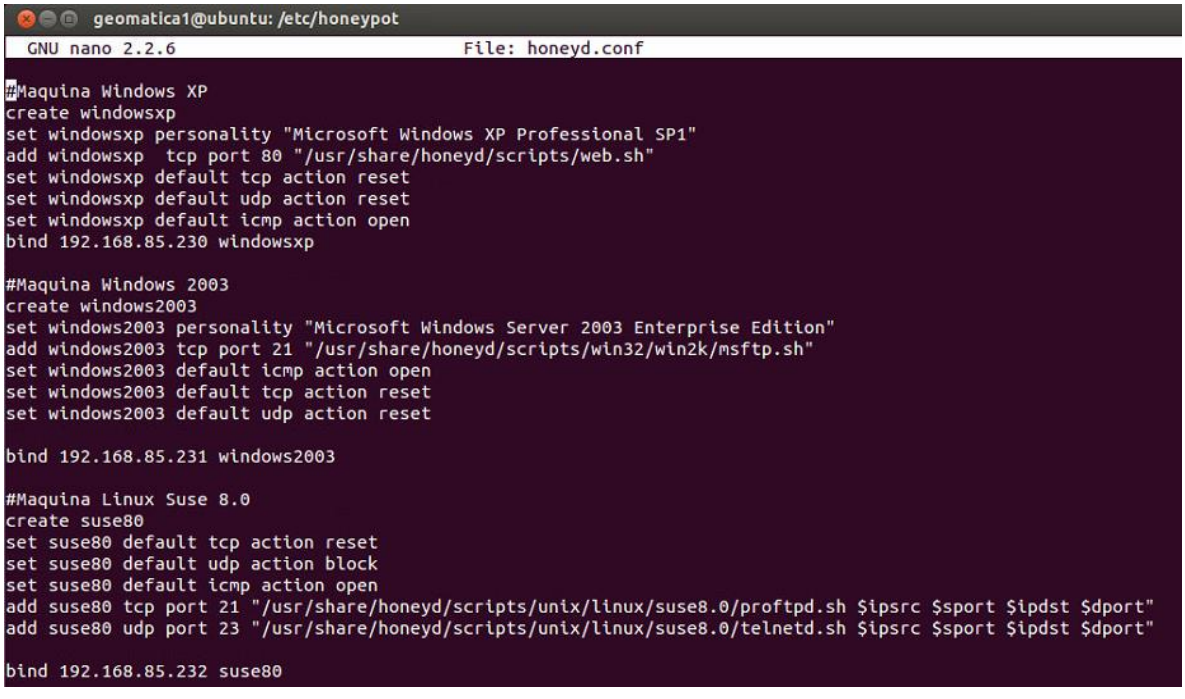
Después de llevar a cabo el proceso de instalación del Honeyd, se realiza la configuración de máquinas trampa de la siguiente manera: se emularon tres máquinas, cada una con un sistema operativo diferente, una dirección IP diferente que está dentro del rango de IPs que controla el *Honeypots* y a cada una de ellas se le asignó diferentes tipos de servicios que son los que se le van a presentar al atacante en el momento en que éste intente ingresar a su sistema. En la Tabla ***14*** se describe el detalle de máquinas trampa con sus respectivos servicios instalados.

***Tabla 14**. Sistemas operativos y servicios montados en las máquinas trampa*

Sistema Operativo	Servicio
Windows XP	Web = Puerto 80 TCP = Reset UDP= Reset ICMP= Open
Windows Server 2003	MFTP = Puerto 21 TCP = Reset UDP= Reset ICMP= Open
Linux Suse 8.0	TCP = Reset UDP= Block ICMP= Open Proftpd= Puerto 21 Telnet= Puerto 23

Estos tres sistemas operativos emulados junto con los servicios que ofrece se deben implementar en el archivo de configuración del Honeyd tal como se muestra en la figura 55.

Figura 55 Archivo de configuración honeyd.conf



Fuente: configuración honeyd.conf