ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"Adaptación de un módulo que garantice la alta disponibilidad del IDS/IPS Snort"

TESINA DE SEMINARIO

Previa a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN SISTEMAS DE INFORMACIÓN

INGENIERO EN CIENCIAS COMPUTACIONALES
ESPECIALIZACIÓN SISTEMAS DE MULTIMEDIA

Presentada por:

NELSON FERNANDO HERRERA CONFORME

CARLOS XAVIER SÁNCHEZ QUIÑONEZ

MARCO ANTONIO RODRÍGUEZ POZO

Santiago de Guayaquil - Ecuador

Año: 2011

A G R A D E C I M I E N T O

Agradezco a Dios por la salud y la fuerza que me ha dado durante mi vida universitaria, a mi hermana Landys por todo su apoyo incondicional. Agradezco a todos aquellos que me ayudaron a realizarme como persona brindándome un apoyo moral.

Carlos Sánchez Q.

Agradezco a quienes siempre confiaron en mí y estuvieron para apoyarme; a mis padres quienes siempre han sido mi inspiración, ejemplo y gracias a ellos he llegado donde nunca imaginaba.

Nelson Herrera C.

Agradezco a Dios, a mis padres de manera especial a mi madre que siempre estuvo conmigo aconsejándome para terminar mi etapa universitaria, gracias mamá.

Marco Rodriguez P.

D E D I C A T O R I A

Dedico este trabajo a mi familia, en especial a mi hermana Landys, quien siempre estuvo pendiente de mis estudios desde mi niñez y dándome un apoyo incondicional para culminar exitosamente mis estudios.

Carlos Sánchez Q.

Dedico el presente trabajo a mi familia, de quienes recibí ayuda incondicional para culminar exitosamente mis estudios.

Nelson Herrera C.

Dedico este trabajo a mi familia, en especial a mis abuelos: Marina Campoverde y Segundo Pozo que fueron uno de los principales guias en casi toda mi vida academica, Dios los tenga en su gloria.

Marco Rodriguez P.

D E C L A R A C I Ó N E X P R E S A

"La responsabilidad del contenido de este Trabajo de Graduación, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la **Escuela Superior Politécnica del Litoral**"

(Reglamento de Graduación de la ESPOL)

Nelson Fernando Herrera C.

Carlos Xavier Sánchez Q.

Marco Antonio Rodriguez P.

TRIBUNAL DE SUSTENTACIÓN

Msc. Alfonso Aranda

PROFESOR DE LA MATERIA DE GRADUACIÓN

Msc. Vannesa Cedeño Mieles

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

RESUMEN

El proyecto consta, a demás de la Introducción, de seis capítulos más:

En el capítulo 1 se describe el análisis contextual, donde se plantea, los antecedentes y objetivos del proyecto.

En el capítulo 2 se presenta el marco teórico y los conceptos intrínsecos al tema del proyecto, detallando brevemente a los sistemas IDS/IPS, alta disponibilidad, sistema SNORT con sus respectivas especificaciones.

En el capítulo 3 se presenta una breve descripción y analisis de las especificaciones técnicas y las herramientas de propósito general que aportan de gran manera a la solución del proyecto.

En el capítulo 4 se describe la instalación y configuración de herramientas de uso general.

En el capítulo 5 se describe el diseño y metodología utilizada en el desarrollo del proyecto.

En el capítulo 6, se presenta las pruebas realizadas y se procura hacer el análisis respectivo del comportamiento del servicio de Snort y del resto de los programas interventores.

ÍNDICE DE CONTENIDO

ABREVIATURAS

IDS: Sistemas de Detección de Intrusos

IPS: Sistemas de Prevención de Intrusos

DBMS: Sistema de Gestión de Bases de Datos

DRBD: Dispositivo de Bloques Replicado y Distribuido

HA: Alta Disponibilidad

NIDS: Sistemas de Detección de Intrusos basados en Red

MV: Maquina virtual

DDos: Ataque de denegación de servicio

DNS: Sistema de nombres de dominio

FTP: Protocolo de transferencia de archivos

ICMP: Protocolo de Mensajes de Control de Internet

IMAP: Protocolo de acceso a mensajes de Internet.

POP3: Protocolo de la oficina de correo

RPC: Llamada a Procedimiento Remoto

SNMP: Protocolo Simple de Administración de Red.

IIS: Internet Information Services

P2P: Red punto a punto

CGI: Interfaz de entrada común.

ARP: Protocolo de resolución de direcciones

IT: Tecnologia de Información

HTTP: Protocolo de Transferencia de Hipertexto

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

El presente proyecto es una solución a uno de los problemas mas comunes que existe en las empresas las cuales dependen de un servicio o brindan un servicio a usuarios.

Snort es un detector de intrusos el cual es usado por muchas compañías para dar seguridad a su red, debido a que es un sistema muy importante es de vital importancia tener un plan de contingencia para que la red nunca deje de estar monitoreada.

Para nuestro caso vamos a usar una combinación de herramientas open source las cuales garantizarán la alta disponibilidad del (**IDS/IPS**) Snort.

CAPÍTULO 1

1. ANÁLISIS CONTEXTUAL

1.1. Antecedentes

Con el avance de la tecnología y la alta demanda de varios de los servicios ofrecidos por diferentes empresas y la necesidad de las mismas de ofrecer un servicio garantizado y sin interrupciones, es de vital importancia tener un plan de contingencia (**BACKUP**), el cual asegurará fiabilidad para los usuarios y tranquilidad para las personas encargadas del servicio ofrecido. Las fallas de hardware o software suelen suceder en cualquier momento sin previo aviso al igual que un desastre natural o una falla en el fluido electrico.

Para el caso de los sistemas que controlan y sensan el trafico en una red de computadores en busca de amenazas, estos sistemas se vuelven muy cotizados y requeridos por las compañías que buscan la protección de sus archivos y redes, un valor agregado que se ofrece junto a estos sistemas es la redundancia o alta disponibilidad ya que una falla podría perjudicar enormemente el trabajo o el servicio de la empresa. El presente proyecto plantea una forma de asegurar la

alta disponibilidad especificamente en uno de los sistemas IDS/IPS OpenSource, mas usados en la actualidad “**SNORT**” **[1]**

1.2. Objetivos del proyecto

1.2.1. Objetivos Generales

- Proveer alta disponibilidad al IDS/IPS “Snort”.

1.2.2. Objetivos específicos

- Adaptación de un módulo usando heartbeat para garantizar alta disponibilidad y drbd para la replicación de datos entre servidor primario y backup.
- Adaptación de un módulo usando acoplamiento de interfaces de red (**bonding**), el cual proveerá tolerancia a fallos en las interfaces de red.

CAPÍTULO 2

2. MARCO TEÓRICO

2.1. Sistemas IDS/IPS

2.1.1. Introducción

Los IDS/IPS o tambien conocidos como IDPS, nacen de la unión de dos importantes sistemas:

- IDS (Sistemas de detección de Intrusos)
- IPS (Sistemas de Prevencion de Intrusos)

Los IDPS están diseñados para cumplir dos funciones principales:

- Evitar una infiltración a los sistemas informaticos de una empresa, sirviendoce de un sistema captación de pruebas.
- Realizar seguimientos exhaustivos del ente infiltrado y de los pasos que sigue el mismo, para en un futuro realizar las acciones mas adecuadas.

De manera más sencilla los IDS/IPS fueron creados por tener la necesidad de proteger toda información digital que sea de importancia para una empresa o entidad, por estas caracteristicas a los IDS/IPS se los denominan sistemas “AntiHacker”. Más detalles en **[2]**.

2.1.2. Carácteristicas Generales

Un IDS/IPS es un sistema que implementa las características de los sistemas IDS como la de los IPS, pero por ser sistemas conjuntos sus características no son simplemente la unión de ambas, se definen características propias para estos sistemas, y se las detalla acontinuación:

- Detección de procesos hostiles
- Escaneos de puertos.
- Ataques a aplicaciones y servicios.
- Detección de debilidades en DNS, FTP, ICMP, IMAP, POP3, RPC, SNMP, entre otros protocolos de red.
- Detección de explotación de vulnerabilidades de software y servicios como por ejemplo IIS, Oracle, SQL, Exchange, etc.
- Detección de actividades relacionadas con mensajería y chats.
- Detección de actividades de redes P2P.

- Detección de código malicioso.
- Base de datos de exploits actualizada.

Ademas de las caracteristicas detalladas, los sistemas IDS/IPS, pueden trabajar conjuntamente con algunos sistemas informaticos, como por ejemplo: Gestores de Bases de Datos, Clusters de Replicacion, Clusters de Alta Disponibilidad, como apoyo o ayuda para el monitoreo, registro de fallas o vulnerabilidades, como por ejemplo algunos ofrecen comunicación con ciertos gestores de bases de datos como forma de salvar todos los eventos que registran de la red que monitorean.

2.1.3. Clases de IDS/IPS

Las clases están orientas al medio de conexión de la red o a la estructura lógica de la red, dependiendo del medio se dividen en cuatro clases como son:

- **Basados en la Red.-** Supervisa el tráfico de paquetes en la red.
- **Wireless.-** De manera similar al de basado en red, supervisa el tráfico inalámbrico con la meta de identificar la actividad sospechosa.
- **Análisis de conducta de la red.-** Monitorea la red centrandose en los incrementos inusuales de trafico en la red.

- **Basados en Host.-** Se enfoca en supervisar hosts especificos con potenciales amenazas.

2.1.4. Esquemas de Red con Snort

Se debe de colocar el IDS de forma que se garantice la interoperabilidad y la correlación en la red. Así la interoperabilidad permite que un sistema IDS pueda compartir u obtener información de otros sistemas como firewalls, routers y switches, lo que permite reconfigurar las características de la red de acuerdo a los eventos que se generan.

Se puede colocar el IDS de las siguientes formas:

- Delante del firewall.
- Detrás del firewall.
- Combinación de los dos casos.
- Firewall / NIDS

2.1.4.1. Delante del firewall

De esta forma el IDS puede comprobar todos los ataques producidos, aunque muchos de ellos no se hagan efectivos. Genera gran cantidad de información en los logs, que puede resultar contraproducente.

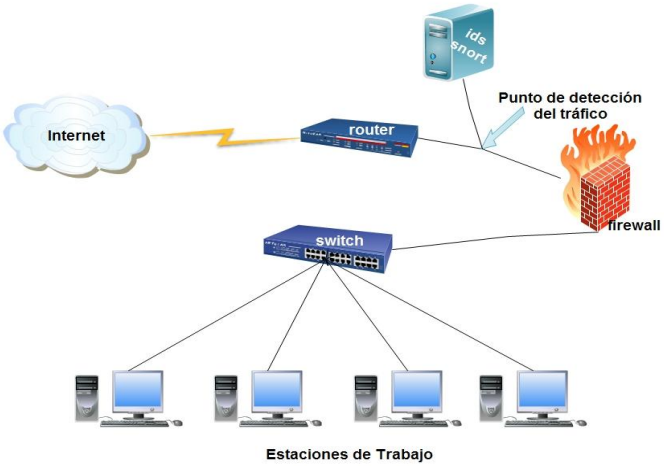


Figura 2.1.a: NIDS delante del Firewall

2.1.4.2. Detrás del firewall

Snort colocado detrás del firewall suele ser la ubicación característica, puesto que permite analizar, todo el tráfico que entra en la red (y que sobrepasa el firewall).

Además, permite vigilar el correcto funcionamiento del firewall. Monitoriza únicamente el tráfico que haya entrado realmente en la red y que no ha sido bloqueado por el firewall. Con lo cual la cantidad de logs generados es inferior a la producida en el caso anterior.

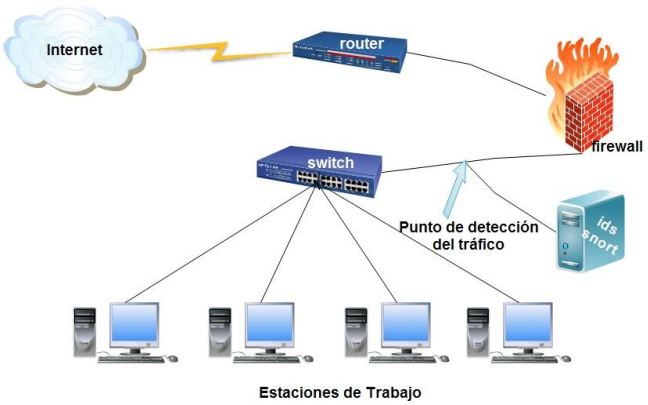


Figura 2.1.b: NIDS detrás del Firewall

2.1.4.3. Combinación de los dos casos anteriores

Combinando la colocación del IDS delante y detrás del firewall el control que se ejerce es mayor. Se puede efectuar una correlación entre ataques detectados en un lado y otro. El inconveniente es que se necesitan dos máquinas para implementarlo.

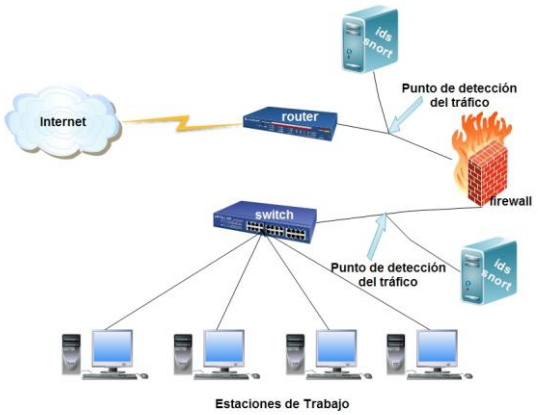


Figura 2.1.c: NIDS detrás y delante del Firewall

2.1.4.4. Firewall / NIDS

Otra opción es usar una única máquina que haga las funciones de firewall y de NIDS a la vez.

Figura 2.1.d: Firewall / NIDS

2.1.5. Sistemas IDS/IPS en el mercado

- **TopLayer.-**Es una empresa dedicada a proveer globalmente de sistemas de prevención de intrusos en redes (IPS), desarrolla y provee al mercado soluciones de infraestructura de red de seguridad que ayudan a las organizaciones comerciales y gubernamentales a proteger sus activos críticos en línea de las pérdidas y los riesgos asociados con las amenazas informáticas. Su familia de dispositivos IPS está diseñado con tres dimensiones de Protección (3DP) que proporcionan las capacidades de

protección más avanzadas contra ataques conocidos y desconocidos en las más altas tasas de rendimiento probado.

- **SourceFire 3D.-**Desarrollado por SourceFire Inc. En el 2001, es un sistema de detención de intrusos (IDS), que provee una solución de seguridad distribuidas en capas, Cada módulo del sistema SourceFire 3D se basa en las capacidades del anterior a fin de aumentar la protección de una organización de la red.

- **Snort.-**Sistema de detección de intrusos basado en red, con licencia GPL, actualmente es uno de los mayores IDS gratuitos usados en el mundo, entre sus principales funciones está la de identificar la actividad maliciosa, la información de registro acerca de dicha actividad e intentar bloquear o detener la actividad.

- **MacAfee Network Threat Response.-** Creado por MacAfee Company, constituye un sistema que captura, des construye y analiza un malware especifico en la red donde reside el sistema. De forma general se basa en tres características: Detectar un malware en la red, analiza el malware para ayudar al usuario a comprender la amenaza y previene escenarios y amanezcas a través de tecnologías que proporciona la propia compañía.

- **CoreImpact Pro.-** Desarrollado por Core Security Technologies, permite al usuario explorar y explotar las vulnerabilidades de

seguridad en las redes de computadores, terminales, aplicaciones de internet y redes inalámbricas.

2.2. Snort como IDS basado en red

2.2.1. Introducción

Snort es un sistema que fue creado inicialmente como un IDS, pero en los últimos años los creadores han agregado funcionalidades de los sistemas IPS, por esto se lo esta considerando un sistema IDS/IPS.

Snort está disponible bajo licencia GPL, es gratuito y funciona bajo plataformas Windows y UNIX/Linux. Es uno de los más usados y dispone de una gran cantidad de filtros o patrones ya predefinidos, así como de actualizaciones constantes de estos ante casos de ataques, barridos o vulnerabilidades nuevos que vayan siendo detectadas a través de los distintos boletines de seguridad, que se publican en la Web o por las organizaciones que realizan seguimientos de ataques.

2.2.2. Definición

Snort es un Sistema de detección de intrusos basado en red (NIDS). Implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques,

barridos, intentos aprovechar alguna vulnerabilidad, análisis de protocolos, conocidos. Todo en tiempo real, más detalles en **[3]**.

2.2.3. Características propias

Además de las características de los sistemas IDS/IPS, como se detalló anteriormente, cada sistema de estos implementa su propia arquitectura y por consiguiente tendrán características que otros no tienen, y esto es el caso de **Snort**, cuyas características son las siguientes:

- Implementa un lenguaje de creación de reglas flexibles, potente y sencilla.
- La instalacion incluye cientos de filtros o reglas para backdoor, ddos, finger, ftp, ataques web, CGI, escaneos Nmap, etc.
- FlexResp, que permite a Snort tener funcionalidades de Firewall.
- Da la opción de actualizar su base de filtros por medio del web.
- Permite trabajar en varios modos: Sniffer, Packed Logger, IDS de red, Inline.
- Contiene una gran variedad de Preprocesadores para varios protocolos, por ejemplo: Stream5, sfPortscan, RPC decode, HTTP Inspect, SMTP preprocessor, DNS, SSL/TLS, y mas.
- Permite a los usuarios el poder crear sus propias reglas.

2.2.4. Arquitectura

La arquitectura de Snort se basa en 3 modulos principalmente, distribuidos jerarquijamente, considerando el análisis y aceptación de un paquete en la red que monitorea. Estos modulos constituyen la base fundamental del funcionamiento y la herramienta más poderosa para el objetivo de Snort, la detección de toda anomalía y vulnerabilidad en la red.

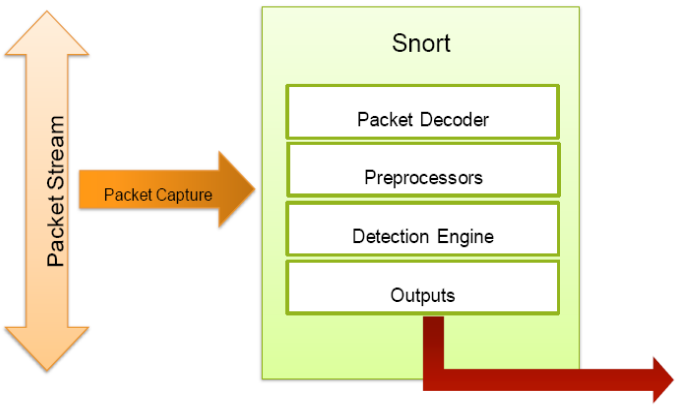


Figura 2.2.a: Arquitectura de snort.

Estructura Funcional del Decoder

(Packed Decoder)

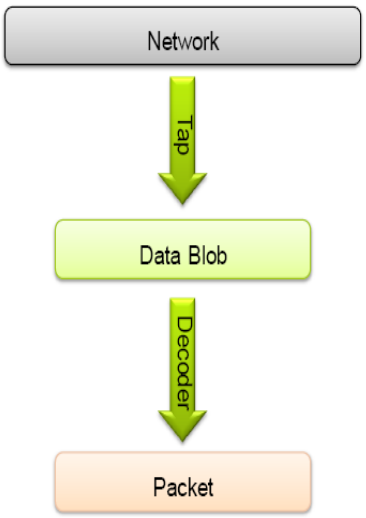


Figura 2.2.b: Estructura del Packed Decoder.

El decoder es la primera sección por la que pasa un paquete que ha sido captado por Snort, tiene a su cargo las siguientes operaciones:

- Recivir el bloque de datos.
- Agregar punteros criticos en ciertos sectores de los datos.
- Ethernet header
- IP header
- TCP header
- Payload
- Pequeños chequeos para purgar son hechos aquí.

Figura 2.3: Cabeceras de un paquete a ser procesado por Snort.

Preprocesadores

La sección de preprocesadores tiene a su cargo, el análisis de los paquetes que ya han pasado por el decoder, aquí se realizan las siguientes tareas:

- Detectar ataques y alguna actividad que no esta disponible en las reglas generales de Snort.
- Presentar los datos de una manera normalizada, para ello tiene una herramienta para normalizar los datos.
- Reensamblar mensajes para posteriormente devolver a la sección anterior para que comienze el ciclo de detección de Snort.

Figura 2.2.c: Orden de procesamiento por protocolo.

2.2.5. Metodologia de detección

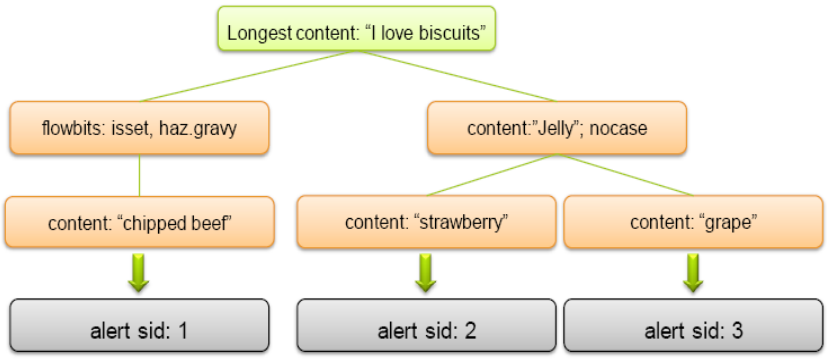


Figura 2.4: Arbol del algoritmo de Snort (detecta código malicioso).

En esta fase, la metodología de detección conforma principalmente el algoritmo por el cual detecta, si un paquete captado por Snort, tiene algún patrón de ataque comparando con la base de firmas de Snort, se basa en la revisión de la estructura del paquete inclusive en la parte de datos del paquete, si este tiene un fin dañino.

El procedimiento para la detección se realiza de la siguiente forma:

- Si se es posible, en todos los casos, se realiza un content match.
- El content match se realiza con la mayor longitud posible.
- Fija y compara similutes en la base de firmas de ataques de Snort a través del content match.

- Los chequeos que se realizan involucran los headers o el estado de los streams como por ejemplo (Flow, y Flowbits).

2.3. Virtualización

La virtualización **[4]** es un conjunto de tecnicas de abstracción de recursos computacuionales que a traves de una capa virtual permite que multiples maquinas virtuales con sistemas operativos heterogeneos pruedan ejecutarse individualmente en las misma maquina fisica.

Cada maquina virtual dispone de su propio hardware virtual (memoria RAM, CPU, etc), con el cual se gestiona el sistema operativo y las aplicaciones.

2.3.1. Sistema anfitrión o host

Es el sistema operativo de la maquina fisica en donde se ejecutarán sistemas de virtualización, como VirtualBox o Vmware.

2.3.2. Maquina Virtual

Es la representación de una máquina real que, a través del software adecuado, provee un ambiente operativo en el que se puede ejecutar o alojar un sistema operativo.

2.3.3. Software de Virtualización (Libre)

Los programas de código libre que funcionan tanto en Mac OS, en Windows como en GNU/Linux, son los siguientes:

Software	Características
Xen	✓ Proporciona aislamiento seguro ✓ Control de recursos ✓ Garantías de calidad de servicio ✓ Migración de máquinas virtuales en caliente
OpenVZ	✓ Proporciona mejor rendimiento ✓ Escalabilidad ✓ Densidad ✓ Administración de recursos dinámicos ✓ Facilidad de administración que las alternativas.
VirtualBox	✓ Modularidad. ✓ Uso de XML . ✓ Optimización del sistema virtual instalado.

Tabla 2.3.1 Software Virtualización Libre

2.3.4. Aceptación en el Mercado

Libre		
Software	**Aceptación del mercado**	**Desarrollado por**
KVM	50%	Red Hat
VirtualBox	28%	Oracle Corporation(todos)
Xen	16%	
OpenVZ	6%	SWsoft, Inc.(solo linux)

Tabla 2.3.2 Aceptación en el Mercado Software Virtualización Libres

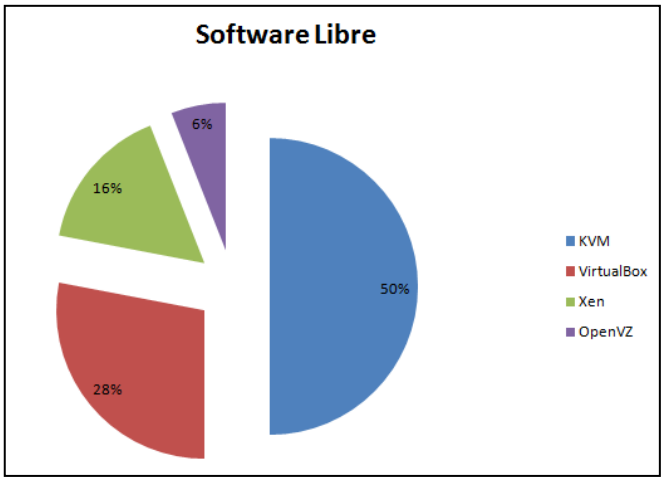


Figura 2.5 Aceptación en el Mercado Software Virtualización Libre

2.4. Alta Disponibilidad

2.4.1. Introducción

La mayoría de sistemas de información se encuentran en servidores. Estos servidores no se encuentran libres de inconvenientes, el hardware puede fallar o un error humano puede ocasionar que nuestro sistema quede fuera de servicio durante un tiempo.

2.4.2. Definición

La alta disponibilidad consiste en una serie de medidas tendientes a garantizar la disponibilidad del servicio, es decir, asegurar que el servicio funcione durante las veinticuatro horas los 365 dias del año.

2.4.3. Motivos para tener sistemas con alta disponibilidad

La falla de un sistema informático puede producir pérdidas en la productividad y de dinero. Por eso es necesario evaluar los riesgos ligados al funcionamiento incorrecto o falla de uno de los componentes de un sistema informático y crear recursos para poder anticipar los medios y medidas y asi evitar incidentes o restablecer el servicio en un tiempo aceptable.

Las causas de las fallas pueden ser las siguientes:

- Causas físicas (de origen natural o delictivo)
- Desastres naturales (inundaciones, terremotos, incendios)
- Ambiente (condiciones climáticas adversas, humedad, temperatura)
- Fallas de la red de comunicaciones
- Cortes de energía
- Causas humanas (intencionales o accidentales):
    - Error de diseño (software, aprovisionamiento de red insuficiente)
- Causas operativas (vinculadas al estado del sistema en un momento dado):
    - Errores de software
    - Falla del software

2.4.4. Diferencia entre fiabilidad y disponibilidad

El término "**disponibilidad**" hace referencia a la probabilidad de que un servicio funcione adecuadamente en cualquier momento.

El término "**fiabilidad**", que se utiliza en algunos casos, se refiere a la probabilidad de que un sistema funcione normalmente durante un período de tiempo dado. Esto se denomina "continuidad del servicio".

2.4.5. Programas que proveen alta disponibilidad

2.4.5.1. Heartbeat

Es un daemon que provee una infraestructura de cluster de alta disponibilidad con sus clientes. Se comunica en todo momento con los nodos del cluster mediante pings y notificaciones cada cierto tiempo que indican si el servidor está vivo o no. Más detalles en **[5]**.

Características

Sus más importantes características son:

- Máximo número de nodos no establecidos. **Heartbeat** puede ser usado tanto para clusters grandes como clusters de menor tamaño.

- Motorización de recursos: recursos pueden ser reiniciados o movidos a otro nodo en caso de fallo.
- Mecanismo de cercado para remover nodos fallidos en el cluster
- Gestión de recursos basado en directivas, inter-dependencia de recursos y restricciones.
- Varios scripts de recursos (para Apache, DB2, Oracle, PostgreSQL, etc) incluidos.

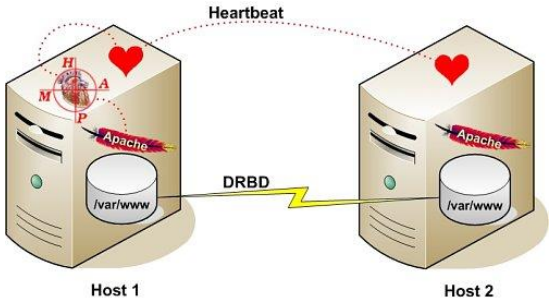


Figura 2.6 Esquema de heartbeat y drbd

2.4.5.2. DRBD

Software que permite hacer réplica de los datos de una partición entre varias máquinas, más detalle en **[6]** y **[7]**.

DRBD es frecuentemente usado junto con el Hearbeat, a pesar de que se puede integrar con otros marcos de gestión de clúster. Tambien puede integrarse con la virtualización de soluciones

como Xen.

Las principales características de DRBD son:

- **Tiempo real.** La replicación se realiza de manera continua.
- **Transparencia.** Las aplicaciones que estén almacenando datos en la unidad no sabrán que está se está replicando en varias localizaciones.
- **Síncrono o asíncrono.** Con replicación síncrona la aplicación que escribe es notificada que la escritura se ha completado solo después de que ésta haya sido replicada. Con replicación asíncrona la notificación de que la escritura se ha completado será entregada a la aplicación cuando ésta haya sido realizada de manera local, esto es, antes de que la escritura haya sido propagada al resto de ordenadores.

2.4.5.3. VMware HA

VMware High Availability (HA) **[8]** permite a las compañias alta disponibilidad para cualquier aplicación corriendo en una máquina virtual.

El esquema general sucede cuando el servidor físico falla, las máquinas virtuales afectadas a este fallo, se reinician automáticamente en otros servidores espejos con las mismas capacidades o mejores para servir dando el servicio. Ademas, si en

una máquina virtual se produce un fallo a nivel del sistema operativo, dicho problema será detectado por VMware HA y la máquina virtual afectada se reiniciará en el mismo servidor físico.

Con VMware HA las compañias IT pueden:

- Proteger aplicaciones que no cuentan con una opción de failover.
- Establecer una linea de defensa de la infraestructura IT, que no está atada a una distribución específica de sisteme operativo.
- Evitar altos costos de soluciones parecidas que dependen de un hardware específico.

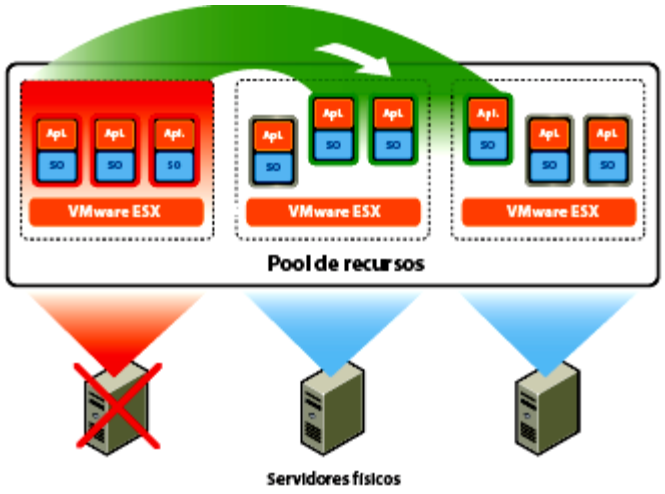


*Figura 2.7: Esquema de Servidores con Vmware HA.**[8]***

2.4.6. Acoplamiento de tarjetas de red (Bonding)

Bonding consiste en hacer trabajar varias tarjetas de red como si fuera una, lo que conlleva a que comparten la misma dirección MAC, que puede ser cualquiera del juego de tarjetas que se esta uniendo. El resultado final es un aumento de la velocidad bastante considerable, y mediante una interfaz virtual todas las tarjetas de red obedecen a una sola ip.

Hay que tener en cuenta si se esta usando bonding en un servidor y dentro de una red hay switch conectado, es necesario que los puertos implicados soporten el protocolo 802.3ad (Dynamic Link Aggregation) que en el caso de switches Cisco equivale a la configuracion llamada EtherChannel, caso contrario el switch puede tener fallas al recibir tramas con la misma direccion MAC origen en diversos puertos, ocasionando que se bloqueen o presenten errores

El uso de bonding ofrece los siguientes beneficios:

- **Ancho de banda:** el ancho de banda de la interfaz virtual será la suma de los anchos de banda de las interfaces reales (tarjetas de red instaladas).

- **Balanceo de carga:** tendremos balanceo de carga del tráfico de red entre todas las interfaces reales (por defecto Round Robin).

- **Redundancia:** si una tarjeta de red falla los datos viajarán por las que estén en buen estado.

El comportamiento de las interfaces de red depende de la modalidad que este disponible, generalmente, se habla de los siguientes modos:

- ❖ Servicio en caliente.
- ❖ En espera de balanceo de carga.

Para mayor informacion sírvase ver la siguiente referencia bibliográfica **[18]** y **[19]**.

El bonding puede ser utilizado siempre que se necesite de enlaces redundantes, tolerancia a fallos o redes con balanceo de carga. Es la mejor manera de tener un segmento de red de alta disponibilidad.

2.4.6.1. Modelos de implementación visto graficamente

Modelos para Balanceo de Carga

Modelo 1

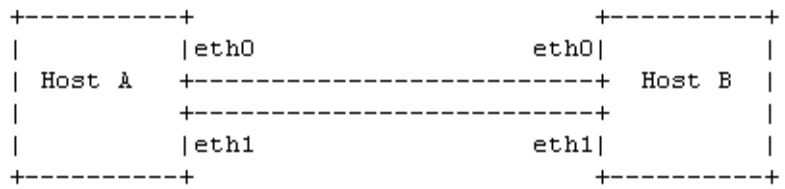


Figura 2.8: Conexión de Host a Host (duplica velocidad de transmisión)

Modelo 2

Figura 2.9: Conexión de Host a Switch (duplica velocidad de transmisión)

2.4.6.2. Modelos de implementación para la Alta Disponibilidad

Modelo 1

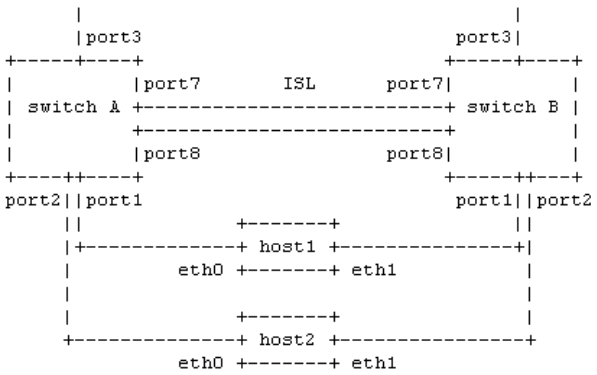


Figura 2.10: Modelo de multiples host y switch's (crear un solo punto de falla).

Modelo 2

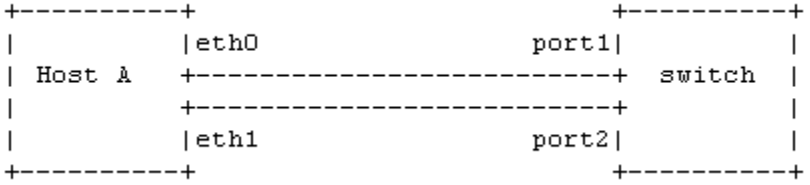


Figura 2.11: Modelo con multiples tarjetas de red conectados a un switch.

CAPÍTULO 3

3. ESPECIFICACIONES Y HERRAMIENTAS PARA LA SOLUCIÓN

3.1. Especificaciones Técnicas

El servidor es la parte más importante de nuestro proyecto ya que se encarga de receptar y procesar la información.

Figura 3.1: Esquema de los servidores

Las características o requerimientos básicos que debe tener el servidor para un buen rendimiento y funcionamiento son las siguientes, ver la tabla 3.1ª

No	Dispositivo	Requerimiento	
		Mínimo	Recomendado
1	Procesador	Pentium IV de 32bits	Intel Dual Core de 64bits
2	RAM	2 GB	4 GB
3	Disco Duro	160 GB	350 GB
4	Tarjeta de Red	10/100 Mbps	10/100/1000 Mbps
5	Tarjeta Analógica	2 puertos	4 puertos

Tabla 3.1a Características del servidor

En la tabla 3.1b se presentan los componentes de software:

No	Componente	Nombre
1	Plataforma	Linux
2	Distribución	Centos 5.5
3	IDS/IPS SNORT	Snort 2.8.6.1
4	DBMS	MySql 5.0
5	Servidor Web	Apache Tomcat 5.0

Tabla 3.1b Componentes del servidor

3.2. Herramientas de proposito general

En esta sección se especifica el aporte que nos brinda cada herramienta basada en software para la implementación de nuestro proyecto.

3.2.1. MYSQL

MySQL es una herramienta opensource **[9]**, se la utiliza en conjunto con el sistema Snort (IDS/IPS) para manejar la interaccion con la base de datos, este gestor de base de datos nos permitirá registrar: alertas, eventos y anomalías las mismas que son capturadas por el sistema Snort.

3.2.2. SNORT

Es un **IDS/IPS** con el cual se está implementando la solución del proyecto, este sistema nos permite registrar, alertar y responder ante cualquier anomalía previamente definida como patrones que corresponden a ataques, vulnerabilidades y amenazas que se presentan en el proceso de escaneo y barrido de puertos en un dominio de colision determinado.

3.2.3. HEARTBEAT

Es un servicio que trabaja enviando latidos (pings), los cuales verifican si el **servidor principal (Master)** esta activo o no, estos pings enviados por heartbeat requieren una respuesta por parte del **servidor principal**

(Master), si al cabo de un cierto tiempo el servidor no responde dichos pings, heartbeat determina que ese servidor se encuentra inactivo, y automáticamente activa al **servidor secundario (Mirror)** que actua como backup, para que asuma el control de la red.

3.2.4. DRBD

El Dispositivo de Bloques Replicado y Distribuido, nos permite realizar la replica de la data del **servidor principal (Master)** al **servidor secundario (Mirror)** y viceversa esto depende de que, servidor se vea afectado para que se realice la replica de la información. Esta replica es transparente para las demás aplicaciones en los sistemas de los equipos.

3.2.5. MONIT

Es un software **[17]** que nos permite realizar un monitoreo exhaustivo de los servicios de mysqld y snortd en cada servidor respectivamente.

3.2.6. Network mapper o nmap

Nmap es una de las utilidades de codigo libre mas integras para el escaneo de redes y por tanto, su uso es el mas extendido. Esta diseñado para permitir a los administradores de sistemas y en general, a quien lo desee, escanear redes a gran escala para determinar que dispositivos estan activos y/o disponibles y que tipo de servicio (nombre, version, etc.)

estan ofreciendo en ese momento. De esta forma es posible explorar las redes y auditar su seguridad.

3.2.7. Wireshark

Es un analizador de paquetes de red. Esto quiere decir que captura datos, en los que procesa y analiza su información para mostrarla de forma detallada. Por el contrario de otros programas costosos, propietarios o ambos, Wireshark es uno de los mejores analizadores de paquetes Open Source hoy en día.

Caracteristicas

- Disponible para Unix y Windows.
- Captura de paquetes en tiempo real desde la interfaces de red.
- Muestra y almacena la informacin de los paquetes capturados.
- Es posible importar y exportar datos de los paquetes hacia o desde otros programas.
- Tambien estan disponibles la busqueda de paquetes y creacion de estadisticas.

3.2.8. Base analisys and security engine o BASE

BASE es un front end de Snort (y en general, de una variedad de herramientas de monitorización como firewalls o demas IDSs), escrito en PHP, que realiza busquedas enlas bases de datos que almacenan los eventos de seguridad registrados, procesándolos para mostrarlos comodamente a través de su interfaz.

Un diagrama general de los elementos que lo conforman se puede apreciar en la *Figura 3.2*

BASE es el sucesor de otro sistema denominado Analysis Console for Intrution Databases (ACID).

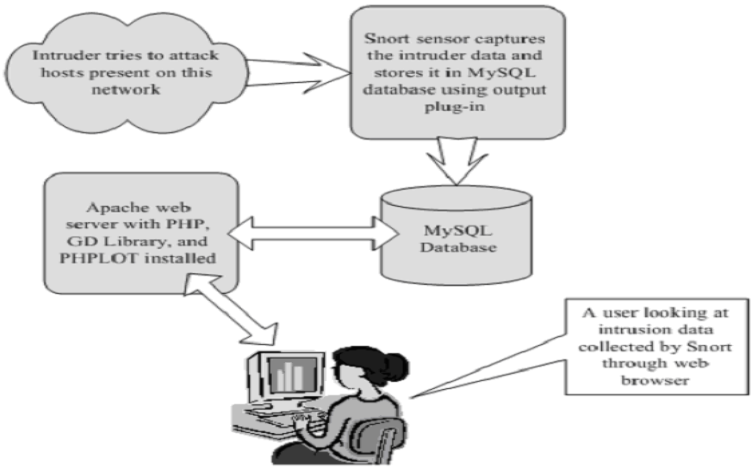


Figura 3.2: Diagrama de un escenario de congifuración Snort+BASE

3.2.9. VIRTUALBOX

Herramienta de virtualización **[12]** la cual hacemos uso para efectos de implementación del proyecto, con esta herramienta montamos nuestro un ambiente virtualizado de dos equipos uno actua como servidor principal y el otro como servidor secundario.

CAPÍTULO 4

4. AMBIENTE DE CONFIGURACIÓN

4.1. Instalacion de herramientas y programas de uso general

Para el caso de este proyecto, Snort registrará la información de las alertas en una base de datos MySQL, denominada snort. La elección de MySQL radica en que es el gestor de base de batos Open Source más popular y por ende, en donde Sourcefire brinda mayor soporte. Para ello será necesaria la instalación Snort compilado con MySQL, Apache2, php5 y otros paquetes complementarios para su funcionamiento adecuado.

La instalación del resto de estas herramientas la podemos hacer de forma sencilla desde la terminal con yum.

#yum install nmap

#yum install wireshark

4.2. Configuración de Snort + BASE

Para la instalación y la respectiva configuración de Snort + BASE sírvase revisar el **Anexo A**.

4.3. Instalación y configuración de Heartbeat - DRBD

Para la instalación y configuración de un cluster de alta disponibilidad con heartbeat y drbd revisar el **Anexo B**.

4.4. Instalación y configuración de Monit

Para el monitoreo de los servicios de mysqld y snortd utilizamos el software monit, ver mas detalles de su configuración e instalación en el **Anexo C**.

4.5. Acoplamiento de tarjetas de Red (Bonding)

En el **Anexo D** se especifica la configuración del bypass que provee alta disponibilidad a nivel de hardware, para ello se utiliza el concepto de Bonding Linux.

CAPÍTULO 5

5. DISEÑO E IMPLEMENTACIÓN

5.1. Alta Disponibilidad (Hearbeat – DRBD / Snort)

Modelo de Cluster Activo - Pasivo

Para conseguir alta disponibilidad necesitamos que el propio sistema realice todos estos pasos de manera automática, dando al usuario la sensación de que el servicio no ha sufrido ninguna interrupción (el tiempo sin servicio sería muy pequeño).

Para la implementación de este modelo contamos con dos servidores conectados entre si, tanto el servidor principal como el secundario poseen las mismas características y configuraciones ya que el **servidor secundario (Mirror)** es espejo del **servidor principal (Master)**, ambos servidores se monitorizan constantemente, de tal modo que uno realiza el trabajo mientras el segundo queda a la espera, comprobando en todo momento que el servidor principal se este ejecutando. Si el servidor

principal presenta alguna falla ya sea por software o hardware, el servidor secundario toma el control y continúa ofreciendo los servicios que le han sido asignado. La falla ocasionada es transparente para el usuario administrador.

El proceso antes mencionado se puede realizar gracias a las bondades que nos ofrece **Heartbeat**, el cual está específicamente diseñado para funcionar como racimo de alta disponibilidad para cualquier tipo de servicio.

DRBD trabaja en conjunto con heartbeat para dar alta disponibilidad, el servicio de mysqld, snortd, drbd, monit y httpd están asociados con heartbeat, esto implica que después de que se levante el servicio de heartbeat ya sea en el nodo principal o nodo secundario tomará unos segundos para que inicien los servicios antes mencionados. La misión de drbd es, sincronizar los servidores para que la información que se encuentra en el **servidor principal (Master)**, sea replicada constantemente al **servidor secundario (Mirror)**.

El servicio web (**httpd**) lo utilizamos para efectos de pruebas, este servicio nos permitirá observar que servidor posee control de la red, para ello tenemos que digitar en el browser la dirección IP **(192.168.0.126)**.

La dirección IP **(192.168.0.126)**, es una ip virtual por la cual responderán ambos servidores (Principal, Mirror) cada vez que se realice una petición via http.

Al digitar esta dirección en el browser, presionando la tecla enter se presenta un mensaje en una ventana con la siguiente descripción:

Servidor Principal

- Este es el nodo 1, servidor principal ejecutándose.

Servidor Secundario

- Este es el nodo2, servidor secundario tomando control de la red.

Monit será el encargado de estar monitoreando con intervalos de tiempo de 3 seg el o los servicios que le sean asignados **(mysqld, snortd)**, en caso de que algunos de ellos falle, monit intentara reiniciar el servicio por 3 ocasiones consecutivas, el servicio de heartbeat es parado e inmediatamente toma el control el servidor secundario (Mirror) hasta que el administrador de sistemas pueda darle solución al problema ocurrido.

Para mayor detalle ver los siguientes diagramas

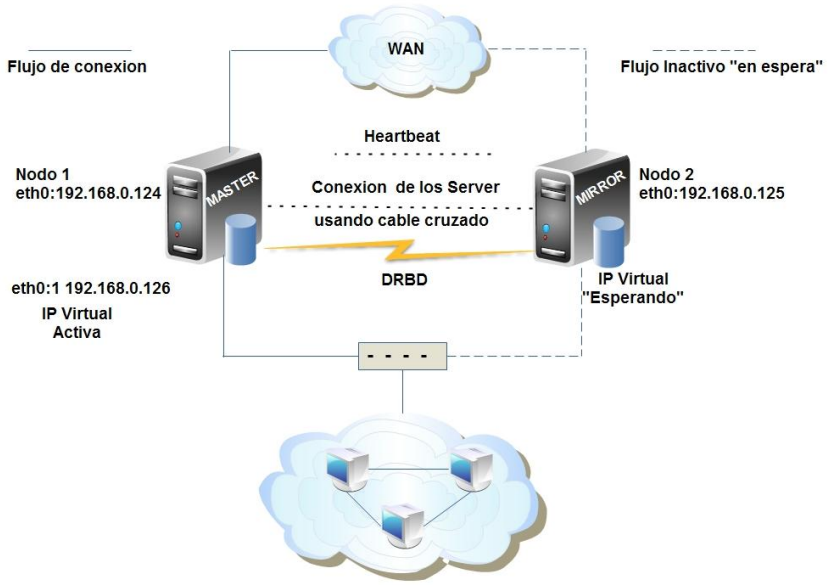


Figura.5.1a: Esquema funcionamiento normal HA

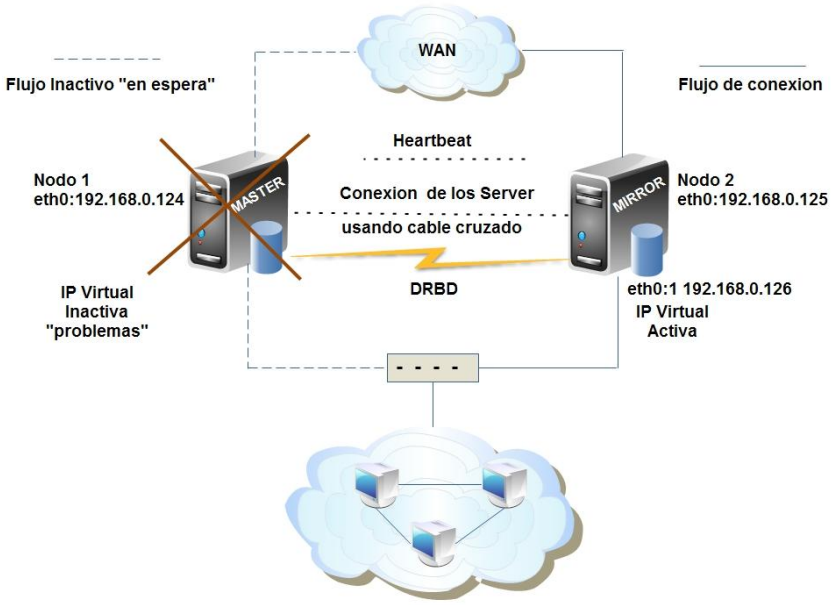


Figura.5.1b: Esquema funcionamiento en modo backup "HA"

5.2. Alta disponibilidad con bonding

Para la implementación de esta solución utilizamos el parámetro **mode** con valor 1 (**mode=1**) más detalle **(27)**, el cual proporciona tolerancia a fallos.

Todo el tráfico se transmite a través de una tarjeta de red y solo se utilizará la otra interfaz en caso de que falle la primera, de esta manera estamos logrando alta disponiblidad a nivel de hardware.

CAPÍTULO 6

6. ESCENARIOS DE PRUEBA

Los siguientes escenarios fueron realizados en un ambiente virtualizado con virtualbox, usando dos computadores, donde el sistema operativo host en ambos era Windows 7 y el **sistema operativo guest** usado centos 5.5, 700 MB RAM, disco duro 20GB y tres tarjetas de red (dos para alta disponibildad entre servidores y una para coneccion a la red). *Ver figura 5.1a y5.1b*

6.1. Basados en software

6.1.1. Servicios o procesos no disponibles

Es de conocimiento general que en cualquier sistema operativo sus servicios o procesos pueden no estar disponibles por diferentes razones.

Como se explico en la sección 5.1 los procesos mysqld y snortd están siendo monitoreados.

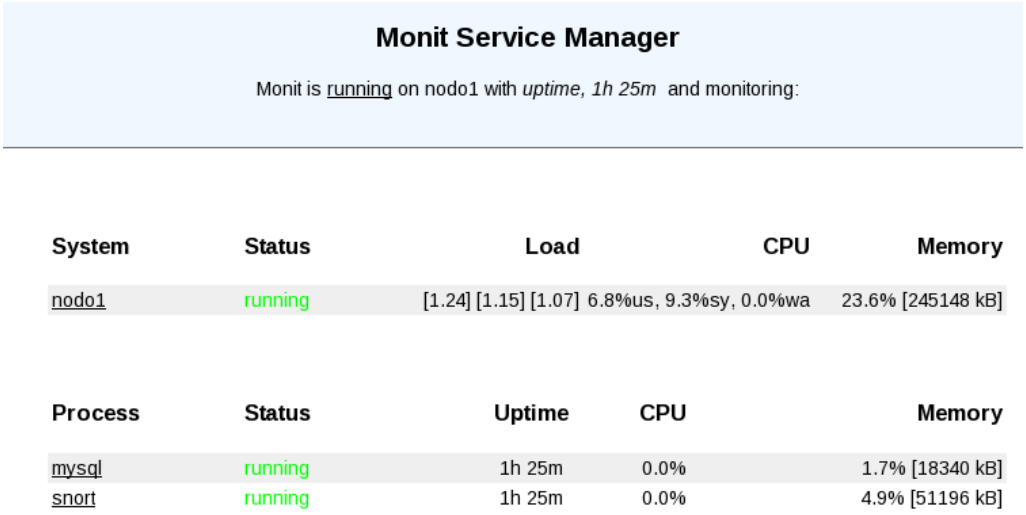


Figura 6.1 Esquema de servicios monitoreados

En este escenario el servidor principal tiene uno o los dos procesos no disponibles, el monitor de servicios (monit) intenta reiniciarlos por tres veces consecutivas al no lograr su objetivo ejecuta un script que detiene el servicio de heartbeat; el servidor secundario detecta que no hay comunicación con el servidor principal e inmediatamente procede a tomar el control y reanudar las actividades.

Esta tarea donde el servidor secundario es ahora el encargado de realizar las actividades con total normalidad, también conocido como **failover**, se lo realiza con un tiempo de respuesta aproximadamente de 10 seg, cabe recalcar que durante este tiempo se sincronizaron los datos y se iniciaron los servicios necesarios. Este tiempo puede variar y es proporcional a la cantidad de datos a sincronizar.

Una vez que el servidor primario se encuentre nuevamente activo, el servidor secundario establece comunicación con este e inmediatamente detiene los servicios que se estaban ejecutando y da paso a la sincronización de los datos. El servidor primario inicia los servicios necesarios y vuelve todo a la normalidad; el tiempo que toma esta tarea, también conocida como **failback**, es aproximadamente de 30 seg.

6.1.2. Error humano

Unas de las fallas más comunes de la inoperancia de un sistema computacional es el error humano, usuarios administradores con poca experiencia, errores involuntarios tales como:

Líneas de código incompletas, caracteres especiales, olvidar guardar los cambios realizados, procesos incompletos, etc son ejemplos de situaciones que pueden afectar de una u otra manera a cualquier sistema dentro de una empresa.

Dado los acontecimientos antes mencionados el o los servicios en el servidor primario no se encuentran disponibles, el procedimiento que tomarán los servidores para seguir con su normal funcionamiento será igual al descrito en la sección 6.1.1

6.2. Basados en hardware

La mayoría de soluciones o sistemas usados en una empresa se encuentran implementados en equipos los cuales son propensos a fallos, lo que ocasionaría la interrupción de las actividades normales. Por esta razón es una buena práctica o se acostumbra a que las partes mas sensibles de los equipos tales como: discos duros, fuentes de poder, tarjetas de red, etc sean redundantes, es decir, tolerantes a fallos.

6.2.1. Tarjeta de red defectuosa

Una parte vital de nuestra solución es la comunicación directa entre los servidores que se realiza mediante un cable de red. Si una de las tarjetas de red presentara algún fallo no se podría seguir trabajando normalmente. Para dar solución a este problema se ha implementado un bypass en las tarjetas de red de cada servidor a nivel de software también conocido como bonding.

Si una de las tarjetas de red de cualquiera de los dos servidores presentara algún problema, bonding manejara el siguiente procedimiento. Las dos tarjetas de red se encuentran acopladas a una interfaz virtual con una sola ip.

Esta interfaz verifica por intervalos de tiempo el estado de las tarjetas y si

alguna se encuentra inactiva redirige el tráfico a la que este funcional garantizando el servicio continuo.

6.2.2. Cable de red defectuoso

Tropiesos, tiempo de uso, desastres naturales, daño intensional, etc son razones por las cuales un cable de red puede presentar fallos, por ende la interrupción de las actividades normales. Bonding un bypass a nivel de software ofrece la solución, sírvase revisar la sección 6.2.1.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1) Con la solución propuesta se cumple en su totalidad con el objetivo propuesto. Es decir, contamos con el servicio las 24 horas del día los 365 días del año; por ende se cuenta con un sistema tolerante a fallos, fiable y confiable.

2) Se pone en evidencia que si se desea tener un servicio fiable, es necesario hacer uso de una solución de alta disponibilidad.

3) En la actualidad disponer o hacer uso de los servicios de un sistema tiene una importancia muy significativa, es por eso que la alta disponibilidad o tolerancia a fallos es un requerimiento muy tomado en cuenta al momento de adquirir o implementar un sistema computacional en una empresa.

4) Las soluciones Opensource como puede ser snort está actualmente muy valorada y suponen un ahorro de costes considerablemente respecto a las soluciones comerciales como pueden ser las de Cisco por ejemplo.

5) Cuando se habla de alta disponibilidad hay que tener muy claro de su significado, no hay que confundir o pensar que solo es tener dos computadores, donde uno es el servidor primario que en caso de que algo le suceda inmediatamente entra a funcionar el servidor secundario o de respaldo. Hay que tener en cuenta que componentes internos de los

servidores como discos duros, fuentes de poder, etc también son instalados para ser tolerantes a fallos.

6) La implementación realizada para dar alta disponibilidad a un servicio como es en el caso de snort, se realizaron concentrándonos en dos puntos vitales en el mundo de la informática, esto es dar alta disponibilidad, a nivel de hardware y software, para los servidores que provean el servicio de snort, cubriendo de esta manera los dos aspectos por el cual el servicio de snort puede verse afectado, de manera directa e indirecta, aunque no se abarco todos los escenarios para estos dos niveles, se ha encontrado varias soluciones a los problemas más comunes que podrían afectar el correcto funcionamiento de los servidores.

7) Durante el desarrollo de este proyecto se puede concluir que debido a la importancia que toma día a día la alta disponibilidad, es más frecuente encontrar todo tipo de soluciones: gratis, de bajo costo e incluso unas de valores sumamente elevados.

Recomendaciones

1) Como se menciono anteriormente existen muchas soluciones que ofrecen alta disponibilidad de servicios pero hay que tener en cuenta que no todas se adaptan o son las adecuadas para nuestra infraestructura. Por tal razón siempre hay que analizar muy profundo de lo que se requiere hacer. No necesariamente una empresa grande va a requerir una solución costosa que incluya beneficios que nunca serán usados.

2) Es importante resaltar que un adecuado posicionamiento del IDS/IPS snort en la red, influenciaría mucho para que realice mejor su trabajo teniendo una mayor eficiencia en la recuperación de fallas, eventos, alertas o posibles ataques.

3) Tener en cuenta que al usar bonding en una configuracion donde hay switches de por medio, estos deben ser compatible con el protocolo 802.3ad (Dynamic Link Aggregation) que en el caso de switches Cisco equivale a la configuracion llamada EtherChannel.

4) El alcance de alta disponibilidad no solo se limita a los servidores comprometidos, es recomendable que los elementos como switeches y routers a los que se encuentran conectados estos servidores tambien tengan alta disponiblidad. Ya que si solo se cuenta con un solo switch la alta disponibilidad entre los servidores pierde su efectividad cuando suceda algun fallo en dicho switch.

5) Se recomienda instalar todas las dependencias necesarias para no tener problemas con la instalación y la configuración de las herramientas basadas en software.

6) Como es de conocimiento en los escenarios de pruebas se hiso referencia al sistema operativo guest (Centos 5.5), para la instalación y configuración tal como se especifica en la sección de anexos, el usuario debe tener conocimiento básicos de Linux.

GLOSARIO DE TÉRMINOS

TOLERANTE A FALLOS

Es la propiedad que permite a un sistema continuar operando adecuadamente en caso de una falla en alguno de sus componentes.

FAILOVER

Es un modo de operación de respaldo en el cual las funciones de un componente del sistema (heartbeat, mysqld, snortd, etc) son asumidas por un segundo componente del sistema cuando el primero no se encuentra disponible debido a una fallo ya se por hardware o software.

SISTEMA OPERATIVO CENTOS

Es una distribución Linux para propósitos generales basadas en RPM.

MYSQL

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

LICENCIA GNU

Es una licencia creada por la Free Software Foundation, y esta orientada principalmente a proteger la libre distribución, modificación y uso de software.

CLUSTER

Se define al conjunto o conglomerados de computadores con componentes comunes de hardware y que se comportan como si fueran un solo computador.

BACKUP

Se define a todo el proceso que forma en un plan de contingencia para un sistema, también se define a todos los datos de copia o respaldo de un sistema o programa.

BONDING

Acoplamiento de interfaces de red.

SISTEMA OPERATIVO GUEST

Sistema operativo de la maquina virtual

BACKDOOR (Puerta trasera)

En un sistema informático es una secuencia especial dentro del código de programación mediante la cual se puede evitar los sistemas de seguridad del algoritmo para acceder al sistema.

FAILBACK

Se define al proceso de retorno a producción en la ubicación original de un objeto después de un desastre o de un proceso de mantenimiento que ha sido programado.

GPL

General Public License, se define al tipo de licencia creada por la Free Software Foundation en 1989 (la primera versión), la cual está orientada principalmente a proteger la libre distribución, modificación y uso de software.

SNIFFER

Conocido también como packet sniffer, es un program que captura las tramas de una red de computadores.

PACKED LOGGER

Se define a todo software que copia los paquetes transmitidos en una red.

BYPASS

Es un sistema informático que modifica el flujo normal de datos hacia una ruta alternativa si se produce una caída de corriente o algún otro problema.

ANEXOS

ANEXO A

INSTALACIÓN Y CONFIGURACIÓN DE SNORT + BASE

La instalación y compilación de Snort se la realiza de forma manual, para ello se debe de instalar en primer lugar mysql, para indicarle a Snort que almacene sus alertas y eventos en dicha base de datos, adicionalmente se deben instalar los paquetes necesarios para que la instalación sea exitosa.

Instalación

Instalamos los paquetes necesarios desde la terminal, como root ejecutamos los siguientes comandos:

[root@nodo1 ~] # yum -y install mysql mysql-bench mysql-server mysql-devel php-mysql httpd gcc pcre-devel php-gd gd mod_ssl glib2-devel gcc-c++ bison flex php-pear

Iniciamos el servicio de base de datos (mysql) y se lo configura para que arranque en el inicio:

[root@nodo1 ~] # service mysqld start

[root@nodo1 ~] # chkconfig mysqld on

Creamos nuestro directorio donde pondremos todos los paquetes que vayamos a utilizar.

[root@nodo1 ~]# mkdir /root/snortInstall

Procedemos a descargar las fuentes desde el sitio oficial de snort y realizamos la respectiva descompresión.

[root@nodo1 ~] # cd snortInstall

[root@nodo1 snortInstall] # wget http://www.snort.org/dl/snort-2.8.6.1.tar.gz

[root@nodo1 snortInstall] #tar -zxvf snort-2.8.6.1.tar.gz

[root@nodo1 snortInstall] #cd snort-2.8.6.1

Procedemos a compilar, para nuestra instalación.

[root@nodo1 snortInstall] # cd snort-2.8.6.1

[root@nodo1 snort-2.8.6.1] #. /configure –with-mysql –enable-dynamicplugin

[root@nodo1 snort-2.8.6.1]# make

[root@nodo1 snort-2.8.6.1]# make install

Configuración

Finalizada la instalación de snort, se procede con los siguientes pasos.

Creamos el grupo de usuario **snort** al cual le asignamos el usuario **snort**.

[root@nodo1 ~] # groupadd snort

[root@nodo1 ~] # useradd -g snort snort -s /sbin/nologin

Se crea el directorio de trabajo de snort:

[root@nodo1 ~] # mkdir /etc/snort

Se crea el directorio donde se van a guardar las firmas:

[root@nodo1 ~] # mkdir /etc/snort/rules

Se crea el directorio donde va a guardar el registro de actividad:

[root@nodo1 ~] # mkdir /var/log/snort

[root@nodo1 ~] # adduser snort

[root@nodo1 ~] # chown snort /var/log/snort

[root@nodo1 ~] # cp /root/snortInstall/snort-2.8.6.1/etc/* /etc/snort/

Ahora vamos a descargas las reglas, para descargas las reglas necesitamos tener nuestra cuenta en www.snort.org, una ves hecho esto nos logeamos y nos bajamos las reglas en la ruta **[root@nodo1 snortInstall] #.**

Nos ubicamos en la carpeta **/root/snortInstall** para proceder a descomprimir las reglas y copiarlas al directorio de trabajo de snort.

[root@nodo1 ~] # cd /root/snortInstall/

[root@nodo1 snortInstall] # tar xfz snortrules-snapshot-CURRENT.tar.gz

[root@nodo1 snortInstall] # cp rules/* /etc/snort/rules/

Ahora vamos a configurar nuestro snort

[root@nodo1 ~] # cd /etc/snort/

En el archivo **snort.conf** modificamos algunos parametros, tales como poner nuestra red interna, en mi caso **192.168.0.0/24**, ademas de ello modifcamos la ruta donde se encuentran las reglas.

[root@nodo1 snort] # vi snort.conf

…

var HOME_NET 192.168.0.0/24

var EXTERNAL_NET !$HOME_NET

var RULE_PATH /etc/snort/rules

…

Configuremos nuestra base de datos mysql para registrar todos los eventos del snort

#mysql -u root –p

#mysql> create database snort;

#mysql>grant INSERT,SELECT on root.* to snort@localhost;

#mysql>SET PASSWORD FOR

snort@localhost=PASSWORD('password_from_snort.conf');

#mysql>grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to

snort@localhost;

#mysql>grant CREATE, INSERT, SELECT, DELETE, UPDATE on snort.* to

snort;

#mysql>exit

Ejecutar el siguiente comado para crear las tablas.

#mysql -u root -p < ~/snortInstall/snort-2.6.1/schemas/create_mysql snort

Ingrese el password del usuario root

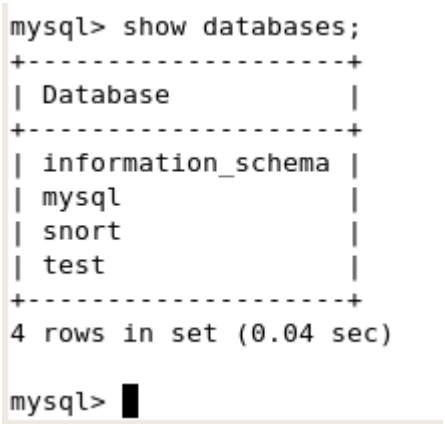
Ahora necesitas chequear y estar seguro de que la base **snort** fue creada correctamente.

#mysql -u root –p

Ingrese el password del usuario root

#mysql>SHOW DATABASES;

(Tú deberías ver lo siguiente)



```
mysql> use snort;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+------------------+
| Tables_in_snort  |
+------------------+
| data             |
| detail           |
| encoding         |
| event            |
| icmphdr          |
| iphdr            |
| opt              |
| reference        |
| reference_system |
| schema           |
| sensor           |
| sig_class        |
| sig_reference    |
| signature        |
| tcphdr           |
| udphdr           |
+------------------+
16 rows in set (0.01 sec)

mysql>
```

Ahora vamos a hacer que el snort registre todo en la base de datos mysql, para ello nuevamente editamos el archivo **snort.conf**, descomentamos la linea de output database para que quede de la siguiente manera.

[root@nodo1 snort] # vi snort.conf

…

output database: log, mysql, user=snort password=contraseña dbname=snort sensor_name=LAN host=localhost

Ahora configuramos a snort para qué arranque como daemon, para ello ejecutamos los siguientes comandos.

Se copia el ejecutable a su directorio de trabajo:

[root@nodo1 ~] # cp /usr/local/bin/snort /usr/sbin

Se copia el script de inicio del servidor:

[root@nodo1 ~] # cp /root/snortInstall/snort-2.8.6.1/rpm/snortd /etc/init.d/

[root@nodo1 ~] #chmod 755 /etc/init.d/snortd

[root@nodo1 ~] # ln -s /usr/local/bin/snort /usr/sbin/snort

[root@nodo1 ~] # chkconfig --add snortd

[root@nodo1 ~] # chkconfig snortd on

*Referencia Bibliografica **[10]** y **[11]***

Instalación y configuración de BASE

#tar -xvzf base-1.4.3.1.tar.gz

#mv base-1.4.3.1 /var/www/web

#chmod 757 /var/www/web/base-1.3.8

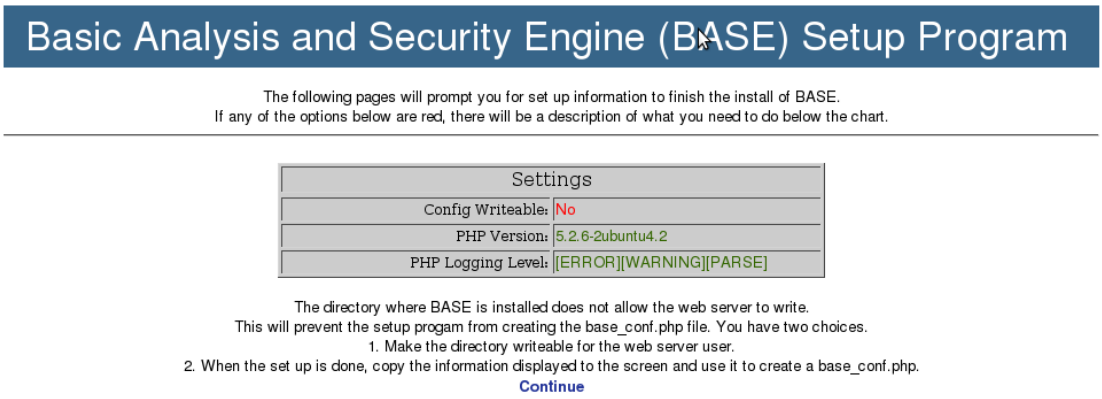


Figura Anexo A 1.1: Configuración Web de BASE – Inicio

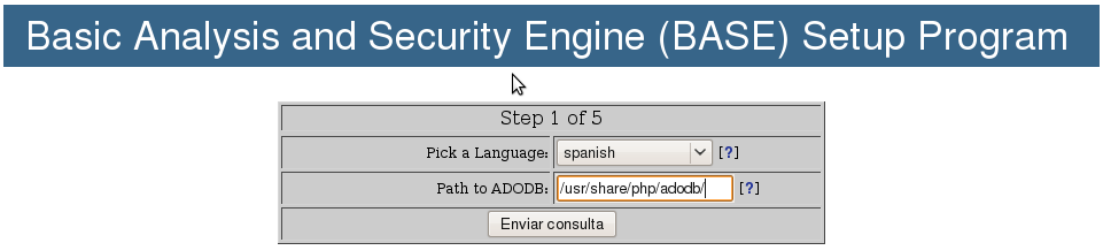


Figura Anexo A. 1.2: Configuración Web de BASE - Paso 1

Configuración via web

Para completar la instalación de BASE, hemos de acceder, a través del navegador a la ruta en la que está contenido en el fíchero de configuración de BASE, de esta forma se nos muestra una pantalla el inicio de este programa (en la Figura anexo A **1.1** se inicia el programa pero se nos muestra dos errores, generados por no darle correctamente los permisos de escritura a la carpeta correspondiente).

La instalación se completa tras cinco pasos: el primero sirve para la selección del idioma y de la ruta del paquete ADODB (Figura anexo A **1.2**), en el segundo paso se

selecciona el tipo de gestor de base de datos (MySQL, en nuestro caso), y la información de nuestra basede datos snort ya creada (Figura anexo A **1.3**), en la tercera etapa, se autentica el usuario del sistema (Figura anexo A **1.4**)y en el cuarto se crean las tablas que soportan la funcionalidad de BASE(Figura anexo A **1.5**), nornalmente, si se ha configurado correctamente tendremos el front-end funcionando adecuadamente (Figura anexo A **1.6**).

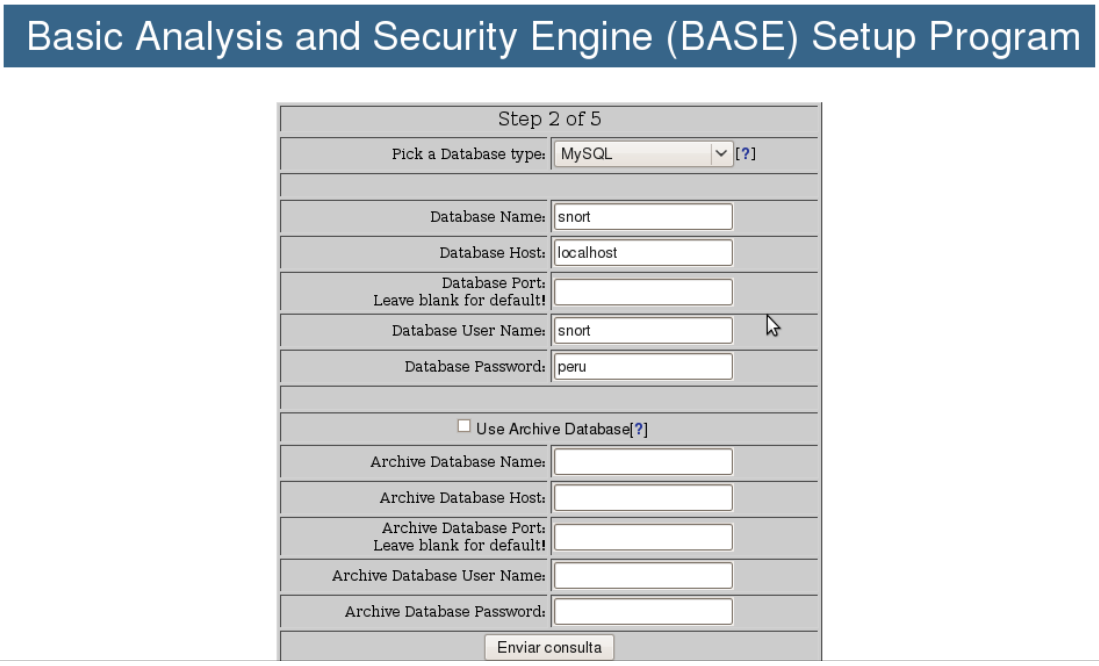


Figura anexo A 1.3: Configuración Web de BASE - Paso 2

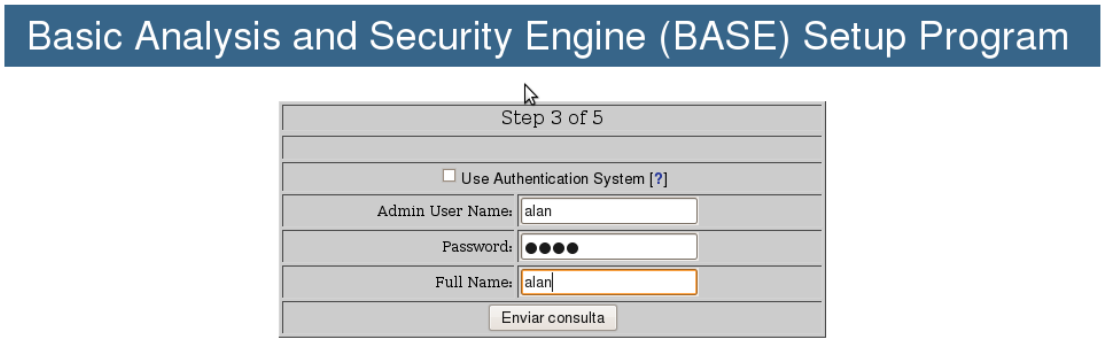


Figura anexo A 1.4: Configuración Web de BASE - Paso 3

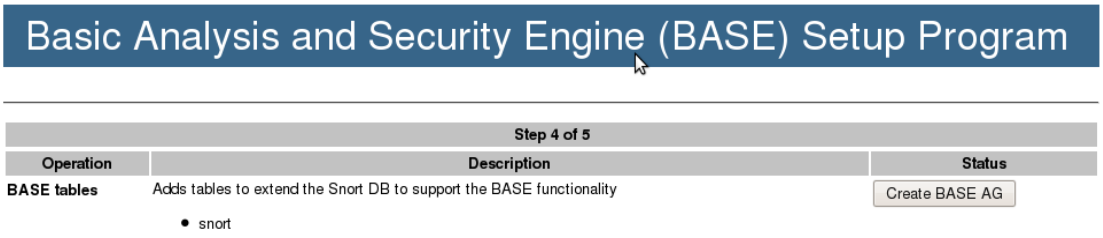


Figura anexo A 1.5: Configuración Web de BASE - Paso 4

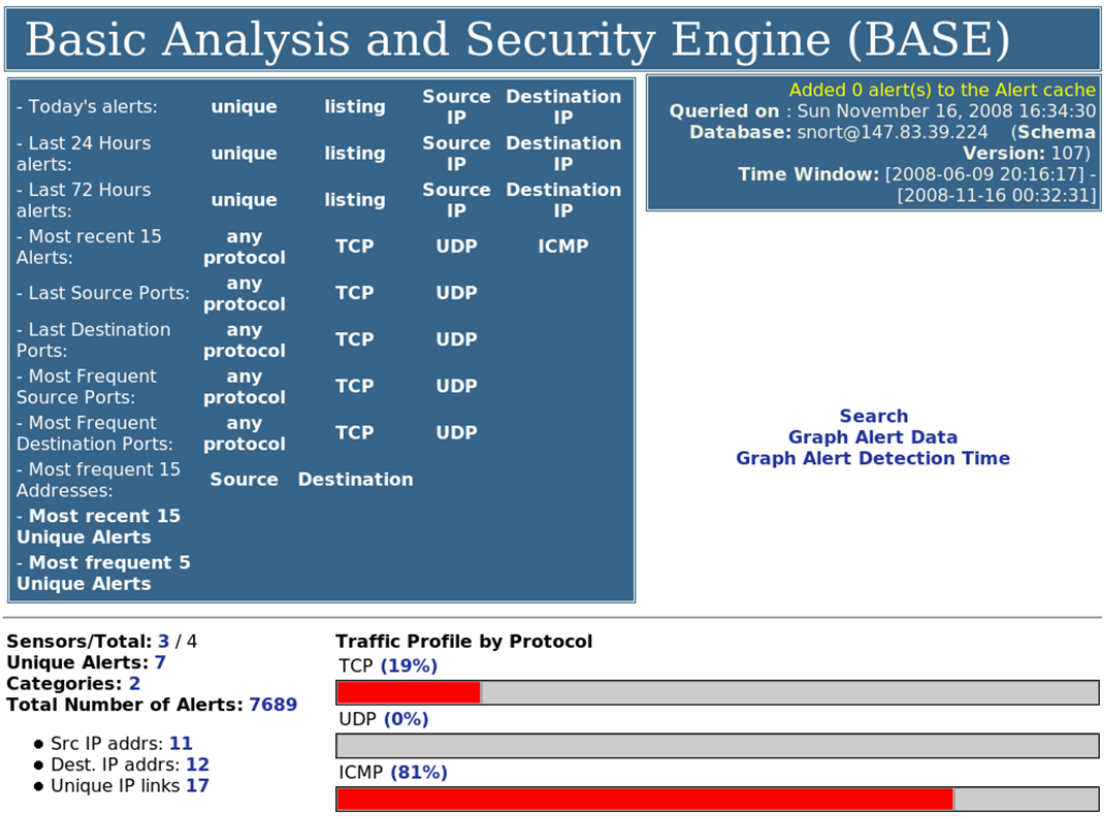


Figura anexo A 1.6: Configuración Web de BASE - Paso 5

Nota: Cabe recalcar que la instalación y configuración de snort se debe realizar en ambos servidores (Master y Mirror).

ANEXO B

MySQL – Snort - HA con DRDB y Heartbeat en CentOS 5.5

A continuación se presenta la instalación y configuración del cluster de alta disponibilidad.

En cada servidor configuramos el arhivo hosts como se especifica en la imagen adjunta.

Desde la terminal ejecutamos **#vi /etc/hosts**

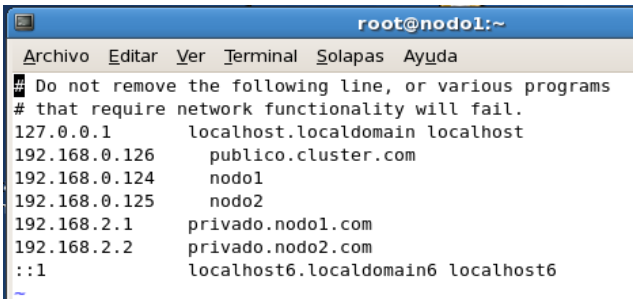


Figura anexo B 1.1: Archivo hosts

DRBD

Instalación

Ejecutamos desde la terminal los siguientes comandos:

 # yum -y install drbd

 # yum –y install kmod-drbd82.x86_64

 # modprobe drbd

Configuración

1. Se necesita una partición o espacio dedicado en cada uno de los servidores para el intercambio de datos entre los dos nodos. Este espacio puede ser un disco duro adicional o una partición.

2. Una vez instalado el paquete de DRBD, se procede a hacer las configuraciones de archivos que se han añadido con la instalación.

3. Se configura el archivo **drbd.conf** que se encuentra en la ruta **/etc/drbd.conf**

   Ejecutamos desde la termial el siguiente comando:

   #vi /etc/drbd.conf

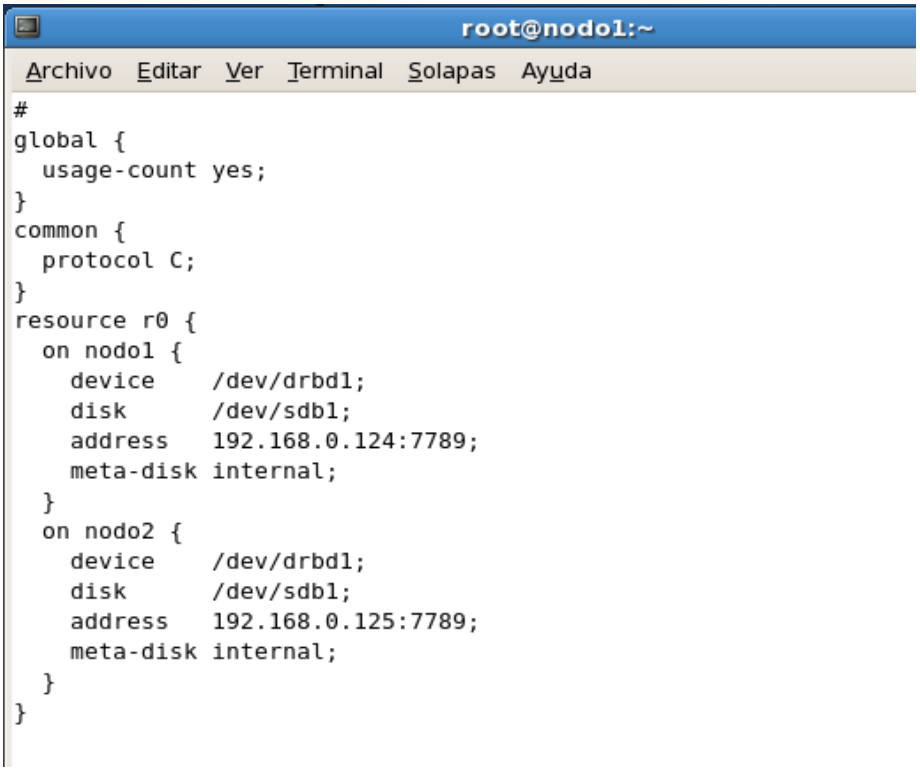


Figura anexo B 1.2: Archivo drbd.conf

El archivo *drbd.conf* se divide en tres secciones: Global, Common y Resource.

En **Goblal** se encuentra **usage-count**, que sirve solo para estadísticas de uso de DRBD. **Common** tiene **protocol C**, que es la manera de cómo se replican los datos. Hay otras opciones más como **protocol A** y **protocol B** que son las otras formas de replicación. En la sección **Resource** se encuentran el nombre de los dos nodos (nodo1 y nodo2), la partición en donde se replicarán los datos e ips de cada nodo con puerto de comunicación.

4. Los pasos anteriores se deben hacer iguales en los dos servidores, lo único que cambia es la ip y el nombre del nodo.

5. Se procede a setear el **resource r0** para que pueda ser usado para almacenar metadata y poder hacer la replicación de los datos. Primero se crea la metadata con el comando drbadadm **create-md resource r0**, una vez creado se procede a adjuntar. Esto se realiza con el comando **drbdadm up.**

6. Para empezar la sincronización se debe ejecutar el siguiente comando solo una vez y en un solo nodo. El que se haya escogido para que sea el primario.

   #drbdadm -- --overwrite-data-of-peer primary resource

7. Se procede a crear el filesystem para el dispositivo creado (**/dev/drbd1**) con el siguiente comando.

   #mke2fs -j /dev/drbd1

8. Se crea un directorio por ejemplo **mkdir -p /root/datos** donde van a estar los datos a replicar o se usa un directorio ya existente.

9. Se monta el directorio mencionado en el paso anterior al filesystem creado en el paso 7 con el siguiente comando **mount -t ext3 /dev/drbd1 /root/datos**

Una vez realizado esto, cualquier archivo nuevo añadido o cambio en los archivos que se encuentren en **/root/datos** serán replicados el momento que ocurra un fallo en el nodo.

*Referencia bibliográfica **[13]** y **[15]***

MySQL

Para configurar MySQL con las particiones de DRBD es simple en nuestro archivo **/etc/my.cfg**, tenemos una directiva que nos dice donde se almacena la data de MySQL, esta directiva es **datadir=/var/lib/mysql** en este caso el datadir apunta al directorio **/var/lib/MySQL**, lo que haremos ahora es simplemente mover el directorio **/var/lib/MySQL**, a **/root/datos/mysql** y luego crearemos un enlace simbólico lo que será suficiente para que la data almacenada por MySQL se escriba en nuestro volumen lógico. Para esto debemos detener el servicio de MySQL previamente.

Nodo1

[root@nodo1 /]# service mysqld stop

[root@nodo1 /]# mkdir /root/datos/mysql

[root@nodo1 /]# chown -R mysql.mysql /root/datos/mysql

[root@nodo1 /]# ln -s /var/lib/mysql /root/datos/mysql

Nodo2

[root@nodo1 /]# service mysqld stop

[root@nodo1 /]# chown -R mysql.mysql /root/datos/mysql

[root@nodo1 /]# ln -s /var/lib/mysql /root/datos/mysql

Una vez realizado esto ya estan preparados nuestros dos nodos, por lo que procederemos a iniciar Mysql en el Nodo1.

HEARTBEAT

A continuación se realizaran los pasos necesarios para instalar y configurar la herramienta que nos permitirá dar alta disponibilidad en los servicios de mysqld y snortd.

Instalación

En ambos servidores instalamos heartbeat y sus utilidades después de estos pasos necesitaras reiniciar las maquinas virtuales.

 # yum -y install gnutls*

 # yum -y install ipvsadm*

 # yum -y install heartbeat*

 # yum -y install heartbeat.x86_64

Configuración

Ahora es momento de meternos con la configuración de heartbeat, prácticamente tendremos que configurar 3 archivos de heartbeat los cuales se encuentran en **/etc/ha.d/...**

/etc/ha.d/ha.cf

En este archivo configuraremos el modo de conexión de ambos equipos.

/etc/ha.d/haresources

Este es el archivo que configuraremos los servicios que serían activados en el Servidor secundario en caso que el Servidor principal sufra algún conflicto.

/etc/ha.d/authkeys

Justamente este archivo es el encargado de la seguridad del sistema, en lo referente a encriptación de los paquetes transmitidos, como también del sistema en si, verificación de archivos alterados, etc.

Empezamos mirando el primer archivo.
#vi /etc/ha.d/ha.cf

#/etc/ha.d/ha.cf contenido

debugfile /var/log/ha-debug

logfile /var/log/ha-log

logfacility local0

keepalive 2

deadtime 30

warntime 10

initdead 120

udpport 694

bcast eth0 # Linux

auto_failback on

ping 192.168.2.1 # El gateway

apiauth ipfail gid=haclient uid=hacluster

node nodo1

node nodo2

vi /etc/ha.d haresources

/etc/ha.d/haresources contenido

nodo1 LVSSyncDaemonSwap::master IPaddr2::192.168.0.126/24/eth0:1

rbddisk::r0 Filesystem::/dev/drbd1::/datos::ext3 mysqld snortd monit

vi /etc/ha.d/authkeys

chmod 600 /etc/ha.d/authkeys

#/etc/ha.d/authkeys contenido

auth 2

2 sha1 BigSecretKeyks9wjwlf9gskg905snvl

*Referencia bibliográfica **[16]** y **[14]***

Una vez configurado los archivos procedemos a desvincular del inicio del sistema los servicios que se encuentran involucrados en el cluster de alta disponibilidad, ejecutamos los siguientes comandos desde la terminal:

#service mysqld stop

#service snortd stop

#service monit stop

#chkconfig mysqld off

#chkconfig snortd off

Seteamos los permisos para que inicie el servicio de heartbeat cuando inicia el sistema, ejecutamos desde la terminal el siguiente commando:

#chkconfig heartbeat on

Nota: Cabe recalcar que todas las configuraciones antes realizadas se deben hacer en los dos servidores (Master y Mirror).

ANEXO C

INSTALACION Y CONFIGURACION DE MONIT

Instalar Monit y configurarlo para monitorear los servicios del servidor web (http), mysql y snort.

Instalación

Para instalar monit tienes que estar como root, en el servidor ejecutamos los siguientes comandos desde la terminal:

[root@nodo1 ~] # cd /usr/src

[root@nodo1 src] # wget http://mmonit.com/monit/dist/monit-5.1.1.tar.gz

[root@nodo1 src] # tar zxvf monit-5.1.1.tar.gz

[root@nodo1 src] # cd monit-5.1.1

[root@nodo1 monit-5.1.1] # ./configure --without-ssl --prefix=/usr

[root@nodo1 monit-5.1.1] # make && make install

Monit se encuentra instalado, ahora configuramos a monit para qué arranque como daemon, para ello ejecutamos los siguientes comandos desde el mismo directorio donde descomprimimos monit.

[root@nodo1 src] # cp contrib/rc.monit /etc/init.d/monit

[root@nodo1 src] # chmod 755 /etc/init.d/monit

[root@nodo1 src] # ln -s /usr/local/bin/monit /usr/bin/monit

[root@nodo1 src] # chkconfig --add monit

[root@nodo1 src] # chkconfig monit on

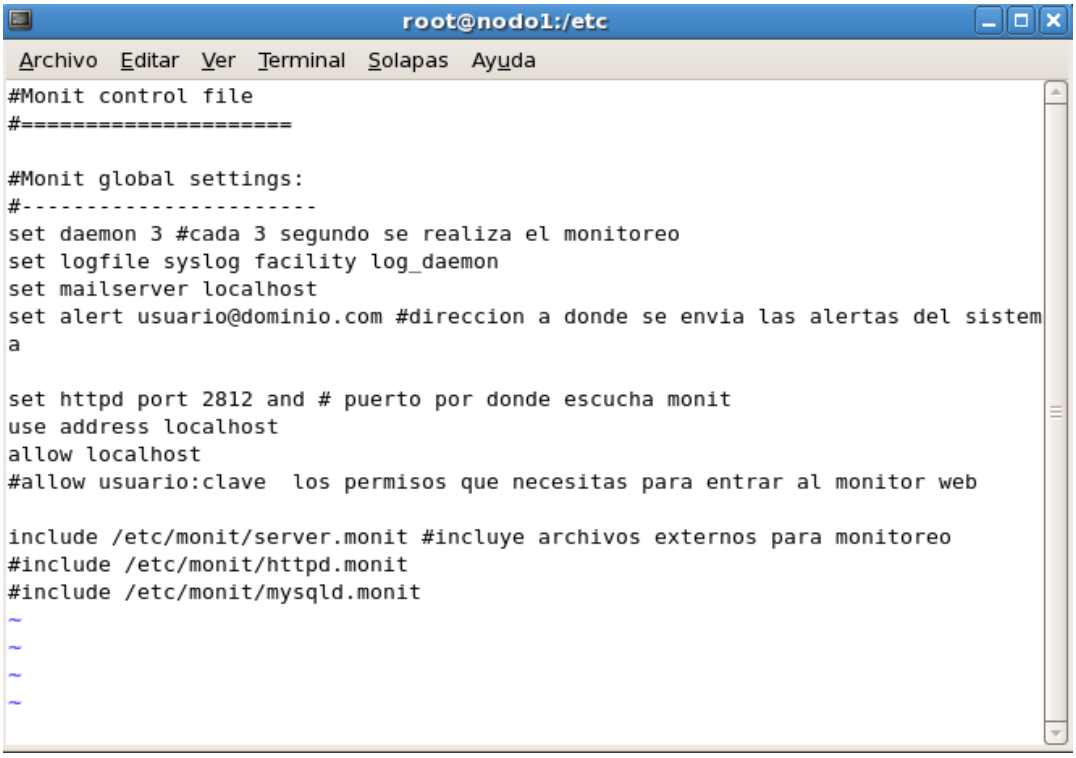
*Referencia Bibliografica **[17]***

Configuración

Una vez instalado monit como daemon editamos el archivo de configuración general de monit desde la terminal:

#vi /etc/monitrc

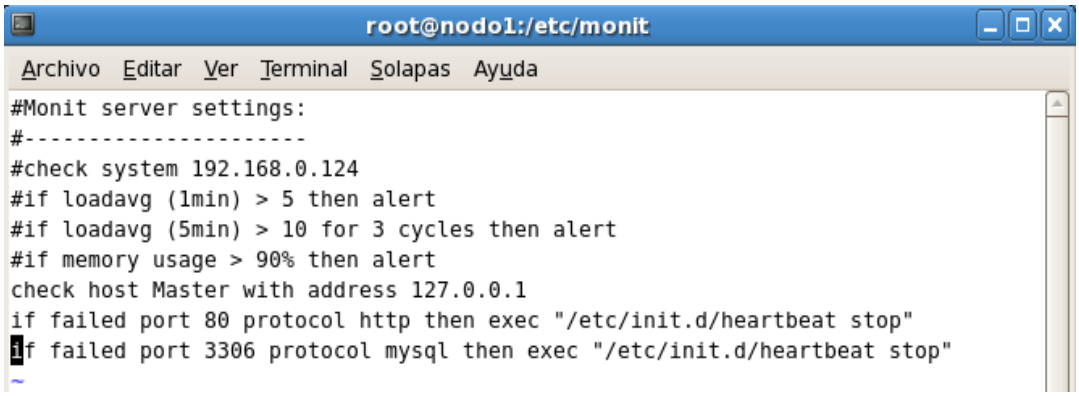
Realizando las respectivas configuraciones en el archivo nos quedaría algo asi como se puede apreciar en la imagen adjunta.



Como podrán observar este archivo de configuración permite establecer cada cuanto tiempo realizar la revisión, el correo donde se senvian las alertas, servicios que activaremos (con include), y acceso al puerto 2812 vía web con usuario y clave (no olvides abrir este puerto en tu firewall).

Ejecutamos desde la terminal **#mkdir etc/monit**, para crear la carpeta que almacenara los archivos de cada servicio que se desea monitorear con monit, para nuestro caso crearemos un archivo de nombre **#touch etc/monit/server.monit** encargado de monitorear el puerto 80 y el 3306 localmente, si llegase a fallar el puerto se para el servicio de heartbeat.

A continuación se presenta una imagen con las especificaciones del archivo antes mencionado.



Incluimos el archivo **#include /etc/monit/server.monit** al archivo de configuración arrancamos el servicio de monit con **#service monit start** para que empiece a monitorear los servicios que les han sido asignados.

Nota: Cabe recalcar que la instalación y configuración de monit debe estar en ambos servidores (Master y Mirror).

ANEXO D

BAYPASS CON ACOPLAMIENTO DE TARJETAS DE RED (BONDING)

A continuación se especifican los pasos que se deben seguir para la **configuración del Baypass con bonding.**

Para la respectiva configuración asumimos que ya se encuentra instalado y configurado snort con el gestor de bases de datos mysql.

Paramos el servicio de snort ejecutando el comando desde la terminal:

[root@nodo1 ~] #service snortd stop

Se procede a crear la interface lógica bond0:

[root@nodo1 ~] # vi /etc/sysconfig/network-scripts/ifcfg-bond0

Editamos el script con las siguientes lineas de código y lo guardamos los cambios.

DEVICE=bond0

BOOTPROTO=none

ONBOOT=yes

IPADDR=192.168.0.128

NETMASK=255.255.255.0

GATEWAY=192.168.0.1

TYPE=BOND

Editamos la interface eth0 con las siguientes lineas de código y luego guardamos los cambios.

[root@nodo1 ~] # vi /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

BOOTPROTO=none

ONBOOT=yes

SLAVE=yes

MASTER=bond0

TYPE=Ethernet

Editamos la interface eth1 con las siguientes lineas de codigo y luego guardamos los cambios.

[root@nodo1 ~] # vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1

BOOTPROTO=none

ONBOOT=yes

SLAVE=yes

MASTER=bond0

TYPE=Ethernet

Configuramos el archivo **modprobe.conf** con las siguientes líneas de código, para ello procedemos a editar el archivo ejecutando el comando:

[root@nodo1 ~] # vi /etc/modprobe.conf

----

alias scsi_hostadapter ata_piix

alias scsi_hostadapter1 ahci

alias snd-card-0 snd-intel8x0

options snd-card-0 index=0

options snd-intel8x0 index=0

remove snd-intel8x0 { /usr/sbin/alsactl store 0 >/dev/null 2>&1 || : ; };
/sbin/modprobe -r --ignore-remove snd-intel8x0

alias bond0 bonding

options bond0 mode=0 miimon=100

alias eth0 e1000

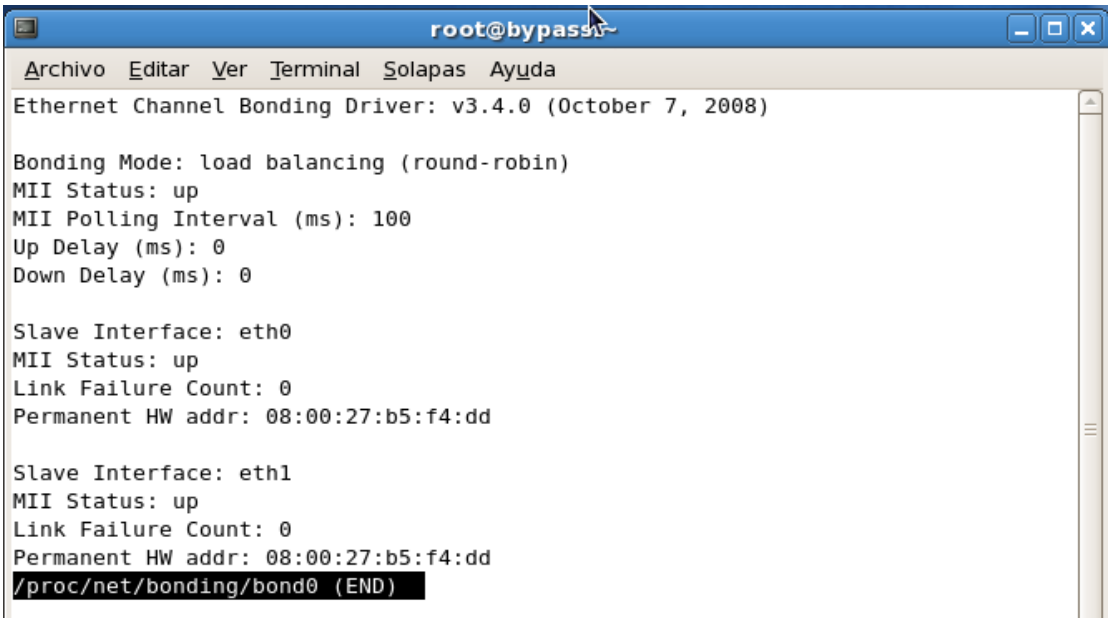
alias eth1 e1000

----

*Referencia Bibliografica **[18]***

Procedemos a chequear el status de las interfaces físicas relacionadas con la interface lógica (bond0) que se creo:

[root@nodo1 ~] # less /proc/net/bonding/bond0



Editamos el archivo de configuracion de snort **(vi /etc/init.d/snortd)** y le cambiamos la interface eth0 por bond0 y listo reiniciamos el servicio de snortd:

```
if [ "$INTERFACE"X = "X" ]; then
   INTERFACE="-i bond0"
else
   INTERFACE="-i $INTERFACE"
```

[root@nodo1 ~] #service snortd restart

Nota: Las dos interfaces eth0 y eth1 deben tener las mismas MAC address

BIBLIOGRÁFIA

Basicamente se ha utilizado Internet como fuente de información del proyecto. A continuación se detallan los enlaces usados.

[1] Sitio oficial de Snort

http://www.snort.org/

[En línea] [Citado el: 1 de Agosto del 2010]

[2] IDS / IPS

http://www.kineticsl.com/html/idsips.html

[En linea] [Citado el: 5 de Agosto del 2010]

[3] Sistemas de Deteción de intrusos y Snort.

http://www.maestrosdelweb.com/editorial/snort/

[En línea] [Citado el: 5 de Agosto del 2010]

[4] Wikipedia - Virtualización

http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n

[En línea] [Citado el: 10 de Agosto del 2010]

[5] Alta disponibilidad con heartbeat

http://blogs.ua.es/labseps/2010/07/13/alta-disponibilidad-con-heartbeat/

[En línea] [Citado el: 15 de Agosto del 2010]

[6] Sitio oficial de DRBD

http://www.drbd.org/home/what-is-drbd/

[En línea] [Citado el: 17 de Agosto del 2010]

[7] DRBD how to Red Hat / Centos

http://www.adminso.es/wiki/images/3/31/Pfc_Fransico_cap4.pdf

[En línea] [Citado el: 18 de Agosto del 2010]

[8] VMware High Availability

http://www.vmware.com/files/lasp/pdf/products/09Q1_VM_HA_DS_ES_A4_R2.pdf

[En línea] [Citado el: 20 de Agosto del 2010]

[9] "¿Qué es MySQL?:: Bases de datos MySQL".

http://www.esepestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm

[En línea] [Citado el: 22 de Agosto del 2010]

[10] IDS con Snort sobre Centos 5

http://fruizondre.wordpress.com/2009/10/27/ids-con-snort-sobre-centos-5/

[En línea] [Citado el: 26 de Agosto del 2010]

[11] IDS con Snort sobre Centos 5

http://www.snort.org/assets/145/Install_Snort_2.8.6_on_CentOS_5.5.pdf

[En línea] [Citado el: 29 de Agosto del 2010]

[12] VirtualBox

http://en.wikipedia.org/wiki/VirtualBox

[En línea] [Citado el: 5 de Septiembre del 2010]

[13] DRBD: RAID1 en red entre varios equipos

http://redesteleco.com/drbd-raid1_en_red_entre_varios_equipos

[En línea] [Citado el: 6 de Febrero del 2011]

[14] MySQL HA with DRBD and Heartbeat on CentOS 5.5

http://planetmysql.ru/2010/07/21/mysql-ha-with-drdb-and-heartbeat-on-centos-5-5/

[En línea] [Citado el: 6 de Febrero del 2011]

[15] DRBD how to Red Hat / Centos

http://wiki.itlinux.cl/doku.php?id=cluster:drbd

[En línea] [Citado el: 11 de Febrero del 2011]

[16] Cómo configurar un agrupamiento (cluster) de alta disponibilidad con Heartbeat en CentOS 5.5

http://www.alcancelibre.org/staticpages/index.php/como-cluster-heartbeat-centos

[En línea] [Citado el: 12 de Febrero del 2011]

[17] Instalar y cunfigurar monit para Linnux

http://agarzon.php.com.ve/instalar-y-configurar-monit-para-linux/

[En línea] [Citado el: 15 de Mayo del 2011]

[18] Acoplamiento de tarjetas de red (bonding)

http://blogofsysadmins.com/como-configurar-acoplamiento-de-tarjetas-de-red-bonding

[En línea] [Citado el: 2 de Junio del 2011].

[19] Linux Ethernet Bondig Driver mini-howto

http://www.kernel.org/pub/linux/kernel/people/marcelo/linux-2.4/Documentation/networking/bonding.txt

[En línea] [Citado el: 4 de Junio del 2011].