FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

IMPLEMENTACIÓN DE UN SERVIDOR REDUNDANTE PARA LA GESTIÓN DE SERVICIOS DE UNA RED EN PRODUCCIÓN BASADO EN SISTEMAS UNIX

PROFESOR GUÍA
ING. FABIÁN WLADIMIRO BASANTES MORENO

AUTOR
CHRISTIAN ALEJANDRO MORA CAMPOVERDE

AÑO
2016

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación”

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes”

AGRADECIMIENTOS

“De antemano agradeciendo a Dios por darme la salud y la fuerza para realizar este trabajo, gracias a mi familia y amigos que de alguna manera estuvieron junto a mí en el desarrollo aportando opiniones que aunque no conlleven datos técnicos, siempre es bienvenido aceptar las críticas constructivas de las cuales he aprendido y por las cuales he llegado a finalizar el presente escrito”

DEDICATORIA

“Dedico este trabajo a todos mis compañeros de universidad, para que con él se puedan complementar ciertos conocimientos que nunca se llegan a obtener del todo, sino a base de experiencia; es por ello que este trabajo también abarca ciertos problemas laborales reales, de los cuales me he inspirado para desarrollar un modelo de monitoreo basado en software libre, que cualquier persona puede investigar, utilizar y aplicar”

INDICE DE CONTENIDO

INDICE DE IMAGENES

CAPÍTULO 1: SISTEMAS OPERATIVOS

1.1 DEFINICIÓN E HISTORIA

Se define como sistema operativo al conjunto de programas que interactúan entre sí para coordinar y dirigir servicios, procesos y aplicaciones para el usuario final; ya que en la actualidad, resulta sencillo encontrar varios de estos sistemas funcionales en dispositivos electrónicos con microprocesador incorporado, como por ejemplo un celular, un reproductor de DVD y hasta televisores Smart.

La necesidad de un ordenador que ejecute procesos básicos y complejos surgió en la década de 1940. Para esa época, el mundo se recuperaba de las guerras mundiales y para el progreso de una sociedad en ruinas, se empezó a idear maquinas que simplificaran los procesos y administración de una fábrica o empresa.

Para los años 50, las computadoras existían únicamente para propósitos científicos o militares, por lo que el acceso para el ciudadano promedio estaba restringido. Fortran Monitor System fue uno de los primeros sistemas operativos, junto al sistema de IBM, que posteriormente tomaría la delantera gracias a su evolución en el multiproceso. Con la evolución de la electrónica, las compuertas lógicas y los chips integrados, se logra avanzar en el manejo de hardware diferente y la interacción con los sistemas operativos y es en esta generación donde aparecen los primeros sistemas MS-DOS y UNIX. (FreeBSD, 1995)

Entre los años 60 y 80, con la aparición de Internet, se va haciendo necesaria la conectividad entre sitios remotos, se definen nuevos estándares LAN, WAN y Wireless y la necesidad de los sistemas operativos de interactuar con las redes de datos y el modelo OSI[1].

El desarrollo del tema, se centra específicamente en la distribución FreeBSD[2] de UNIX, puesto que la sumatoria de sistemas GNU[3] actuales, como de los grupos Open Source y las empresas que apoyan el desarrollo de software libre, tales como: IBM o SUN.

Con todo ello, se ha logrado que el software de código abierto en estas épocas sea tan capaz de rivalizar ante soluciones de código cerrado como WINDOWS, AVID[4], CISCO o AS400[5].

---

[1] Open System Interconnect.
[2] Berkeley Software Distribution.
[3] Referencia de software libre.
[4] Sistema de edición profesional licenciado.
[5] Sistema Operativo propietario de IBM.

Los sistemas GNU empezaron en los años 80; el desarrollo de kernel y el apoyo de codificaciones hechas por la comunidad estudiantil, que apoyaron el avance de nuevas versiones del sistema operativo, así como de su compatibilidad con los diferentes periféricos utilizados en aquella época. A principios de los años 90 ya se establecieron varias versiones del sistema libre, entre ellas, algunas licenciadas y otras que no fueron logrando tener a través de los años hasta la actualidad, un apoyo de un sinfín de organizaciones y comunidades tanto estudiantiles como científicas. (FreeBSD, 1995)

1.2 TIPOS Y DESCRIPCIÓN

Como todos los sistemas de uso libre, FreeBSD se compone de un "kernel" que es el núcleo que interactúa con el hardware del equipo para ser utilizado en las diferentes aplicaciones del usuario, sin embargo, también existe un componente llamado "*world*" en donde se alojan las aplicaciones propias del sistema y tanto el kernel como el *world* conforman el sistema operativo como tal. (FreeBSD, 1995)

Entre la variedad de sistemas operativos, tanto para ordenadores como para equipos móviles actualmente se tiene:

- *Windows*
- *Mac OS*
- *Linux*
- *Amiga OS*
- *Unix*
- *AS400*
- *Solaris*
- *Windows Phone*
- *Android*
- *Symbian*

En los sistemas BSD se puede destacar el extremado poco uso de los recursos del sistema, ya que utiliza apenas 14MB de memoria RAM para sus versiones x64 bits, mientras que para x32 bits ocupa tan solo 8MB, asimismo, las aplicaciones en un entorno KDE[6] no superan los 400MB y 600MB cuando se utilizan versiones gráficas gnome. (FreeBSD, 1995)

También se recalca la facilidad de acceso al código fuente del sistema y la actualización de sus puertos (ports), que te permiten instalar ciertas aplicaciones hechas para los sistemas BSD.

[6] Entorno gráfico para sistemas LINUX, tipo GNOME.

Para obtener acceso a estos puertos simplemente se ejecuta el comando *pkg_add*; aunque también existe *pkg_src*, que es un sistema de gestión de paquetes desarrollado por NetBSD para su uso con sistemas FreeBSD.

Las aplicaciones de Windows no son un problema en entornos BSD, pues simplemente se requiere la aplicación WINE para emular dichos entornos virtuales.

En el ámbito de drivers también se tiene una amplia gama de soporte para tarjetas de red, gráficas y de sonido; se han solventado problemas de compatibilidad ACPI[7] y gracias a los avances aportados por la compañía *iXsystems* se ha ido logrando poco a poco la convergencia de tecnologías con estos sistemas de uso libre. (FreeBSD, 1995)

Aunque *FreeBSD* es de uso libre, también tiene aplicativos de licenciamiento privado al igual que Linux o MAC OS, BHYVE[8] es un ejemplo de ello o como el caso de la compañía SUN, que al desarrollar el sistema de ficheros ZFS[9], que también funcionan sobre sistemas UNIX, es de uso licenciado.

FreeBSD Foundation, es una entidad apoyada por algunas empresas para el apoyo al desarrollo de uno de los sistemas más seguros que existen, como es el caso de *iXsystems*, Google, root labs, McAffe, etc.

1.3 FUNCIONES

Básicamente se realizan ciertas funciones principales en un sistema operativo para la correcta interacción entre el usuario y el ordenador, entre ellas:

- Administración de recursos
- Interfaz de usuario
- Administrador de archivos
- Administrador de tareas
- Servicios de soporte y actualización

Un sistema operativo hace referencia a tres procesos principales, que son:

- Sistema de archivos
- Intérprete de comandos
- Núcleo

[7] Advanced Configuration and Power Interface, que admite interfaces para periféricos externos.
[8] Una versión supervisora de sistemas BSD de uso licenciado.
[9] Zettabyte File Systems, un sistema de archivos introducido por la compañía SUN Microsystems.

En donde, el sistema de archivos administra y conserva la estructura arbórea de los archivos tanto del usuario como del sistema, el intérprete de comandos permite la interacción entre el hardware y el usuario, ya que al no hablar el mismo idioma, mediante los comandos se pueden dar instrucciones básicas de operación. Finalmente, es en el núcleo donde convergen todos los procesos de entrada y salida, la gestión de la memoria y la comunicación interna entre periféricos.

FreeBSD ofrece también alto rendimiento en compatibilidad con otros sistemas operativos, teniendo nuevas ventajas en desarrollo todavía, tales como:

- *Bounce buffering*, que se trata de la arquitectura ISA[10] de los ordenadores que limita el acceso a la memoria en los primeros 16 megabytes.
- *Buffer* de caché combinado con memoria virtual, que continuamente ajusta la cantidad de memoria utilizada por la demanda del usuario.
- Módulos de compatibilidad, que permiten la ejecución de programas de otros ambientes operativos, como Linux, NetBSD, SCO.
- Módulos de kernel dinámicos, que permiten el acceso a diferentes características de los puertos a utilizar, dándole escalabilidad al sistema sin la necesidad de reconfiguración de kernel.
- Librerías compartidas, que reducen el tamaño de los programas, pues se manejan a través de un sistema ELF que combina los archivos necesarios de sistema de los programas con otros de uso similar.

1.4 APLICACIONES

Actualmente, existen muchas empresas y negocios que utilizan software para distribuir servicios a través de su red interna, hacia redes externas o simplemente contar con un recurso operativo para hacer frente a una necesidad tecnológica.

Actualmente, varias aplicaciones se pueden instalar en un ordenador que disponga de un sistema operativo, mismo que interactúa con varios otros programas a la vez basados en códigos de programación, para cumplir una tarea.

Entre los varios servicios que pueden coexistir en un sistema operativo, se encuentran:

- Servicios de correo
- Servicios de telefonía
- Servicios de programación
- Servicios de interconexión

[10] Industry Standard Architecture.

- Servicios de transferencia de archivos
- Servicios de streaming
- Servicios de dominios
- Servicios de firewall
- Servicios de redundancia
- Servicios de edición
- Servicios de antivirus
- Servicios de transmisión

Existen muchas más aplicaciones que se les puede dar a los sistemas operativos, pero en este caso se han citado los más relevantes.

En la actualidad todas las organizaciones requieren de los servicios anteriormente indicados y adicionalmente, tener cierto control sobre los mismos. Para ello se creó el protocolo simple de comunicaciones de red o SNMP, permitiendo a los departamentos de TI una fácil y rápida administración de la red y sus recursos; ya que para el manejo de una red en producción se requiere:

- Identificar los servicios críticos de una topología de red y proyectar su escalabilidad.
- Disponer de soluciones económicas de control y monitoreo de alto rendimiento.
- Asegurar la disponibilidad de los servicios y asegurar los accesos no autorizados.
- Diferenciar el tipo de tráfico, para así optimizar recursos y garantizar la fiabilidad de los servicios.
- Analizar los beneficios de la convergencia entre nuevos protocolos de monitoreo y administración de seguridad que proporcionan los sistemas de uso libre.

El fin de un sistema operativo de monitoreo es simplificar la gestión de los departamentos de TI, al mantener el control y administración de sus servicios, monitoreando el tráfico, aplicando técnicas de calidad de servicio, estableciendo anchos de banda diferenciados, proporcionando guías rápidas de diagnóstico ante incidentes físicos o lógicos en la topología.

CAPÍTULO 2: SOFTWARE LIBRE

2.1 DEFINICIÓN E HISTORIA

El proyecto *FreeBSD*, tuvo sus inicios en el año 1993; en la universidad de *Berkeley* en California, en EEUU. El mismo fue desarrollado y encabezado por Jordan Hubbard, Nate Williams y Rod Grimes; quienes buscaban una solución parche al sistema lanzado 386BSD, después que el creador original Bill Jolitz se retirase al tener algunos inconvenientes con su patchkit 386BSD, pues existían ciertos problemas legales y pese al apoyo de los autores anteriormente indicados, Jolitz destitió del proyecto. Sin embargo, tanto Hubbard como Williams y Grimes, continuaron el proyecto aumentado codificación de kernel y compatibilidades varias. (Foundation, 2015)

FreeBSD se introdujo como nombre código al proyecto lanzado por David Greenman y Walnut Creek, mismos que fueron responsables de la distribución del sistema en Cd-rom y es, a finales del año 93, cuando sale a la luz FreeBSD 1.0, teniendo una gran acogida en el mundo Open Source.

En mayo de 1994 salió la segunda versión del proyecto llamado FreeBSD 1.1 y tuvo ese año varias dificultades legales con la compañía Novell, pues se alegó que FreeBSD estaba basado en Net/2 que era un código cerrado de la compañía AT&T, el cual fue cedido exclusivamente para ellos. No obstante, para julio de 1994, FreeBSD basó su desarrollo sin la cinta de Berkeley Net/2.

En junio de 1995 se dio a conocer FreeBSD 2.0.5, el cual todavía tenía errores por pulir; sin embargo, tuvo un éxito significativo y es a partir de agosto de 1996, donde se empiezan a desarrollar temas de seguridad, conectividad y compatibilidad.
Después de cuatro años de trabajo, se desarrolló una rama de código estable llamada 4.X_STABLE, la cual estaba incluida en la versión 5.0 de FreeBSD, la cual apareció a inicios del año 2003. (Foundation, 2015)

En esta versión, se incluía escenarios multiproceso adaptados al hardware de procesamiento, soporte de aplicaciones y comunicaciones avanzando su desarrollo a través de los años hasta ser en lo más actual su rama código RELENG_7; incluida en las versiones de FreeBSD 6.x en adelante.

FreeBSD (Berkeley Software Distribution), es parte de sistemas similares, como *OpenBSD* y *NetBSD*, dejando un legado importante para las siguientes generaciones de programación libre al disponer de código fuente libre al manejo de memoria virtual, archivo de paginación bajo demanda, *Fast FileSystem*, y el

protocolo TCP/IP; cabe recalcar que casi todas las implementaciones TCP/IP actuales derivan de la original implementación TCP/IP en 4.4BSD-Lite.

2.2 UTILIDADES

Los sistema GNU tienen varias utilidades, desafiando los servicios y funciones que ofertan los sistemas de código cerrado, tal como se describe anteriormente y aunque todavía no se llegan a las mismas prestaciones de un software licenciado, todavía se tiene un largo camino de apoyo para estos sistemas, incluyendo, foros abiertos donde la comunidad GNU puede intercambiar ideas, programación y nuevas funciones. (Foundation, 2015)

Entre las utilidades que podemos encontrar en los sistemas GNU se encuentran:

- Firefox (Software de navegación web)
- Gnome (Software de interfaz gráfica de SO)
- Open Office (Software editor de texto, hojas de cálculo, etc.)
- Vim (Software editor de código GNU)
- Squid (Software filtrador de contenido HTTP)
- Dovecot (Software interprete de correo IMAP)
- Postfix (Software interprete de correo POP)
- Opera (Software de navegación web)
- Wireshark (Software analizador de tramas LAN y direccionamiento)
- Waifu (Software de visualización de videos y series animadas)
- Iptables (Software de reglas y políticas de acceso en Capa 2 y 3)
- Ipfirewall (Software de reglas y políticas de acceso en Capa 2 y 3)
- Packet Filter (Software de reglas y políticas de acceso en Capa 2 y 3)
- Blender (Software de animación gratuito)
- Gedit (Software editor de texto GNU)
- VLC (Software reproductor de audio y video)
- Avidemux (Software editor de video de código abierto)
- Carpd (Software incluido en kernel para redundancia IP)
- Vrrpd (Software incluido en kernel para redundancia IP)
- Glbpd (Software incluido en kernel para redundancia IP)
- Scribes (Software para desarrollar scripts en sistemas GNU)
- Nano (Software de edición de texto para sistemas GNU)
- Thunderbird (Software organizador de correo electrónico)
- Tiger VNC (Software de control remoto gráfico de ordenadores)
- Gwibber (Software de redes sociales para sistemas GNU)
- Jabbim (Software multiplataforma para voz sobre IP)
- Sonata (Software reproductor de música y videos para GNU)
- Audacity (Software editor de audio para sistemas GNU)

2.3 VENTAJAS Y DESVENTAJAS

Existen varias características en los diversos sistemas operativos, lo que hace que suelan tener ventajas sobre otros o inclusive desventajas entre ellos:

2.3.1 VENTAJAS SOFTWARE LIBRE

- No existen costos de licenciamiento ya que cualquier usuario puede tener acceso al sistema operativo y su código fuente.
- Representa una cooperación colectiva de todas las personas interesadas en el sistema y su mejoramiento, expandiendo áreas de conocimiento público.
- Se puede acudir a foros en internet, para compartir ideas y problemas de funcionamiento así como las posibles soluciones a los mismos.
- Asegura la fiabilidad de la información ya que evita ataques traseros que involucran espionaje e introducción de códigos maliciosos.
- Contribuye al desarrollo de profesionales de TI[11] para la mejora continua del sistema.

2.3.2 DESVENTAJAS SOFTWARE LIBRE

- La migración de datos del usuario estándar resulta difícil, además de tener que integrar mediante comandos cualquier hardware adicional que desee agregar al sistema.
- No existen garantías por parte del autor, puesto que como es de uso libre, tendría que mantener un especial cuidado con la información y configuraciones disponibles.
- Para su implementación y configuración, se requieren cierto conocimiento básico de sistemas operativos y de lenguajes de programación.
- Se debe mantener un constante sondeo a la red de internet para vigilar posibles vulnerabilidades y actualizaciones.
- No existe soporte de ningún proveedor, pues actualmente ciertos sistemas y aplicaciones de software propietario resultan una mejor opción, al disponer de mejor funcionamiento y prestaciones.
- De igual forma, para las actualizaciones, se requieren conocimientos de ciertos comandos básicos para proceder con las mismas.

[11] Tecnologías de la Información.

2.3.3 VENTAJAS SOFTWARE PROPIETARIO

- Existe un control de calidad proporcionado por la empresa, cuyos recursos son en el mayor de los casos, más grandes, a diferencia de los desarrolladores de software libre, que trabajan de manera independiente.
- Los desarrolladores son personal altamente calificado e investigativo para innovación de funcionalidades.
- Software amigable al usuario, ofreciendo una interfaz gráfica fácil de manipular e instalaciones más sencillas y autómatas.
- Amplio campo de expansión de uso educativo, además de capacitaciones permanentes al personal de desarrollo.
- Distribución y publicidad rápida a través de los medios de comunicación.
- Se ofrece soporte en línea y a domicilio, también se ofrece un alto nivel de seguridad y se puede solicitar en cualquier momento responsabilidades en caso de falla, daño o pérdida de la información.

2.3.4 DESVENTAJAS SOFTWARE PROPIETARIO

- El aprendizaje del sistema resulta costoso para personas interesadas en capacitaciones acerca del mismo.
- Código fuente restringido y oculto al usuario estándar, haciendo que se requiera soporte en línea continuamente.
- Debido a la alta demanda del sistema el soporte técnico va decayendo.
- Se depende del proveedor para el lanzamiento de nuevas versiones, actualizaciones y parches.
- Se disminuye el avance tecnológico compartido ya que poca gente es capaz de acceder al costo económico que suponen las certificaciones actuales.

2.4 APLICACIONES

Mucha gente no es consciente del uso que se le da al software libre hoy en día; de hecho, se ha avanzado ampliamente en este aspecto. Sin embargo, no se cuentan con medios para la distribución de esta información a nivel mundial, por lo que únicamente las personas inmersas en campos como de la informática o redes de la información, tienen acceso a ella y están abiertos al diseño y la creatividad con dichos sistemas. (Tanenbaum, 2003)

Actualmente el software libre puede hacer frente a varios tipos de programas de código cerrado, pues de otro modo existiría únicamente el software licenciado con todas las desventajas que eso conlleva; no obstante con el software libre se han logrado abarcar temas extensos como:

- Navegación Web y correo electrónico
- Descargas y Gestores
- Mensajería Instantánea
- Reproducción de Audio y Video
- Edición de Audio y Video
- Dibujo y fotografías
- Manejo de archivos
- Virtualización

Existen varios programas de uso libre para todos los temas anteriormente expuestos, de hecho, en algunos casos, en los programas de licencia libre, podría resultar más sencilla su administración y uso.

De hecho, del sistema operativo en cuestión, utilizado para el desarrollo del proyecto, se puede indicar que además de ser de uso libre, ocupa un alto puesto en el ranking mundial de firewalls. (Foundation, 2015)

CAPÍTULO 3: FREEBSD

3.1 SISTEMA UNIX

Como se definió anteriormente, la idea del software libre surgió de la necesidad de trabajar con sistemas informáticos sin licenciamiento. Por otro lado, esto conlleva a un soporte deficiente, ya que es el mismo usuario, quien debe buscar soluciones al código o añadir software adicional, que se va desarrollando poco a poco.

A partir de la necesidad, de cooperación colectiva, de donde nace el software libre y para entender un poco mejor la evolución del sistema UNIX, se debe analizar sus inicios en los laboratorios Bell Telephone de AT&T, alrededor del año 1969.

En ésa época, varias compañías tenían ambiciosos planes de diseño de un sistema operativo para usuarios de hogar y pequeñas oficinas, un sistema fácil utilizar y amigable para personas sin conocimientos básicos de informática. Pero los costos y el personal calificado que el proyecto supondría, condujo a varios intentos fallidos de desarrollar tal sistema, pues, en esa época, *Bell Telephone* se sumaba a otras compañías como General Electric, que intentaban desarrollar el sistema operativo. (Foundation, 2015)

El primer ordenador, donde se hicieron las pruebas del sistema operativo UNIX, fue un DEC PDP7 de 18 bits, pues, en el año de desarrollo del proyecto, se hacían pruebas de código ensamblador y el núcleo de sistema operativo. Posteriormente, se implementaría la idea de un modelo de manejo de ficheros en modo jerárquico, un intérprete de comandos, un editor de texto, etc. Todo ello independiente al modelo de hardware que se esté usando, facilitando el uso y la portabilidad del sistema contribuyendo a su desarrollo. (Ecured, 2015)

Para la década de 1970, se empezó a documentar los avances en el desarrollo del sistema; lo que implica que dichos avances eran publicados libremente en manuales distribuidos como libros.

Para esos tiempos UNIX se empezaba a instalar por otro tipo de usuarios y poco a poco se fue elevando el número de instalaciones del sistema en ordenadores de cualquier tipo y como era un software de uso libre, la compañía no ofrecía soporte, lo que incremento la participación colectiva en el diseño de nuevos programas y corrección de errores. (Bach, 1986)

La Universidad de California, en Berkeley, fue donde inicialmente tuvo su desarrollo el sistema UNIX, a base de lenguajes de programación conjunta de estudiantes y

profesores, uno de estos estudiantes, era Bill Joy, quien empezó a distribuir el sistema como Berkeley Software Distribution o más bien conocido como *BSD*.

Este último personaje sería una influencia bastante importante para el sistema UNIX, puesto que puso en práctica códigos de programación como terminales de texto, interprete de controladores para hardware y gestores de archivos y unos años más tarde, puso en marcha la compatibilidad entre diferente arquitectura de hardware, para ello, fue necesario realizar ciertas pruebas en ordenadores de aquella época como IBM. Como se indicó anteriormente en la reseña histórica de los sistemas operativos, AT&T en los años 80 empezó a tomar parte en el desarrollo del proyecto UNIX, cerrando el acceso al código fuente, por lo que UNIX dejó de ser de uso libre; sin embargo pese a todos los problemas, se necesitaba continuar el desarrollo del proyecto sin contar con el código fuente de AT&T implementado en el entonces llamado UNIX, posteriormente se pondría en desarrollo otros sistemas parecidos a UNIX, a partir de código reescrito desde cero; mismos que, para la década de los 90, le servirían a Linus Torvalds para el desarrollo de lo que hoy conocemos como sistema LINUX. Historia del desarrollo de (Foundation, 2015)

Actualmente, varias empresas y fundaciones apoyan el desarrollo de UNIX de desarrollo libre como por ejemplo:

- UNIX International
- UNIX Systems Laboratories
- Open Software Foundation

Sin embargo, cabe recalcar que actualmente todavía se usan sistemas UNIX de código cerrado, es decir con licenciamiento, como AIX, SOLARIS, HPUX, IRIX, MAC; que son sistemas operativos diferentes. No obstante a pesar de ello, las empresas portadoras de este código licenciado suelen apoyar también, el desarrollo del software libre también. (Moritsugu, 2000)

3.2 SERVICIOS Y PROTOCOLOS APLICABLES

Hoy en día se utilizan varios servicios para diferentes plataformas de hardware, con diferentes sistemas operativos y por ello debe haber una cohesión entre sistemas, puesto que diferentes usuarios están sujetos a diferentes plataformas operativas. A continuación se describen los diferentes protocolos aplicables al sistema UNIX FreeBSD.

- TCP/IP
- IPX/SPX
- FTP
- SSH

- HTTP
- SNMP
- CARP[12]
- DLC[13]
- Apple Talk
- NetBIOS
- NetBEUI

Dado que varios de estos protocolos son aplicables a los sistemas UNIX, para este proyecto se ha tomado en cuenta realizar una interconexión multiplataforma, es decir, que varios sistemas operativos puedan coexistir en una red operativa, combinando accesos y servicios de diferentes tipos ya que se han citado los protocolos aplicables, también se describen los servicios aplicables.

- Servicios de correo
- Servicios de VPN
- Servicios de FTP
- Servicios de WEB
- Servicios de base de datos
- Servicios de video y audio

En una red en producción, es necesario diseñar el esquema básico de distribución de servicios a través de ella, para facilitar que los usuarios de la capa de acceso, tengan acceso rápido y discriminado a todos los servicios a través de la red.

3.3 RED OPERATIVA

En la actualidad, existen empresas con necesidades de expansión que suelen ofertar productos y/o servicios en diferentes puntos geográficos de un país, por lo que se requiere implementación WAN para interconectar dichos puntos. Al unir todas las posiciones geográficas, se necesita configurar varios protocolos sobre la red física existente que sean versátiles y adaptables al medio existente, aprovechando recursos, tiempo y dinero.

Un tipo de solución utilizada actualmente por las empresas, para reducir costos en conexiones dedicadas, es la implementación de redes VPN; que se utilizará para crear un túnel cifrado, alojando todos los protocolos aplicables a través de ella, permitiendo la interconectividad de puntos remotos sin la necesidad de costosas implementaciones dedicadas.

[12] Common Address Resolution Protocol, utilizado para redundancia lógica de único Gateway.
[13] Protocolo de encapsulamiento WAN, tipo PPP.

Dado que la topología de la red en producción del proyecto en cuestión se basa en el tipo estrella, esta concentra el flujo de comunicaciones hacia un punto central. Este modelo de topología red tiene ciertas ventajas como desventajas. Obviamente la primera desventaja radica en que ante la caída del nodo central toda la red estaría fuera de servicio; sin embargo, al disponer de un nodo central, se debe ahondar en temas como redundancia tanto eléctrica como lógica. (Atelin, 2006)

Precisamente, se menciona el tema de las redes VPN para reiterar el hecho de que para una PYME en crecimiento, supone un costo excesivo para la interconectividad entre puntos geográficos distantes. Por ello, la topología en estrella concentra sus enlaces físicos hacia un mismo nodo, sin la necesidad de adquirir más líneas dedicadas hacia nodos distantes.

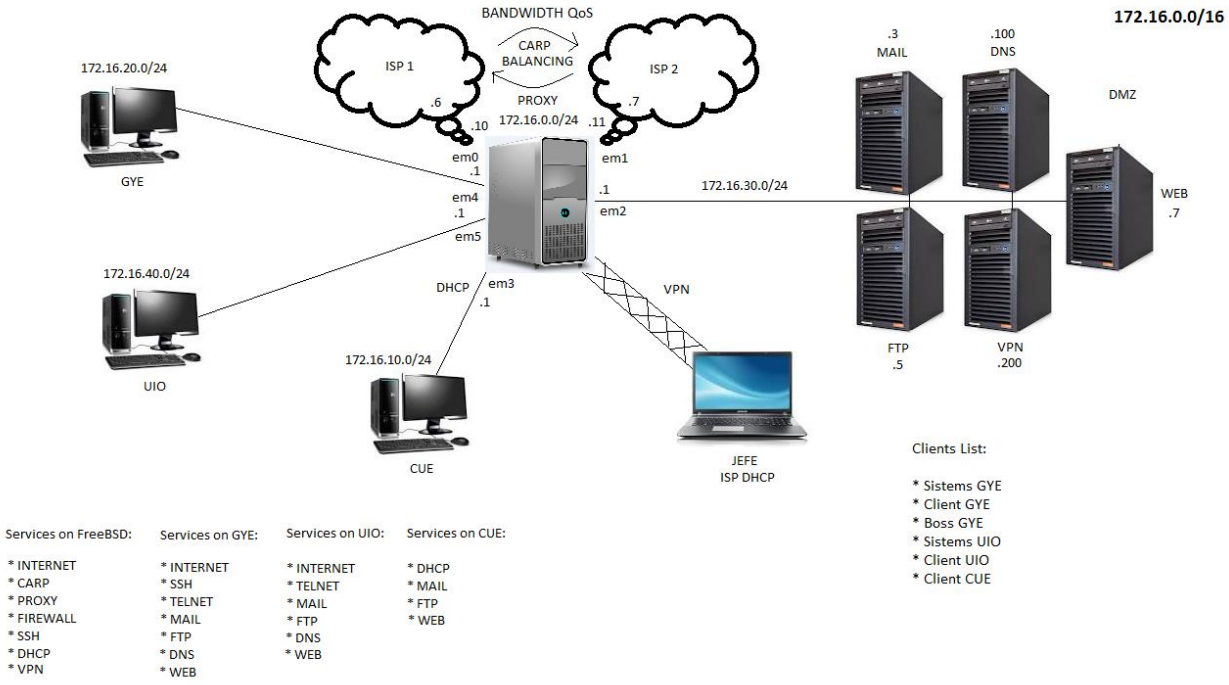

Figura 3.1: Red Operativa Nacional

En la figura 3.1, se puede apreciar claramente una red de topología estrella, con redundancia, que es la que se utilizará en el desarrollo del control y monitoreo de la red en producción. (Palmer, 2000)

Para este caso, se ha tomado en cuenta una empresa distribuidora de etiquetas, que tiene su sede en Guayaquil; cuenta con sucursal en Quito y planifica expandir las operaciones a la ciudad de Cuenca, donde actualmente solo existe un vendedor. En este nodo se concentrarán todas las comunicaciones y es el punto central de acceso remoto. Ciertas áreas de la compañía tienen acceso al servidor, para conocer la situación actual de la red, en cuanto a tráfico y prestaciones.

El protocolo simple de comunicaciones de red versión 3 (SNMPv3), resulta muy útil por su seguridad al momento de aplicarlo para cuestiones de monitoreo, sin embargo en su desarrollo todavía no está contemplada la compatibilidad entre sistemas operativos.

Existen físicamente dos enlaces en la topología, conectados al nodo central, que es la computadora central basada en UNIX, que gestiona todas las comunicaciones, se encarga de ofrecer servicios de la red DMZ, las reglas de acceso, monitoreo y control, balanceo de carga, etc. (Atelin, 2006)

Para todos estos requerimientos, se ha de equipar el ordenador central con tarjetas de red de alta capacidad, que permitan el flujo elevado de tráfico que supone una red operativa en pleno desarrollo.

Figura 3.2: Tarjetas de Red compatibles con sistemas UNIX. (Imágenes extraídas de Google)

Se cuenta con una tarjeta de red una INTEL 82598EB y tres NETGEAR GA302T, compatible con sistemas UNIX. Cabe recalcar que en el kernel del sistema se encuentra un listado con todas las tarjetas de red compatibles con el sistema; facilitando mucho la elección de hardware a operar.

El primer enlace hacia el proveedor de servicios primario, tiene un ancho de banda de 4096 kbps, mientras que la segunda conexión tiene un ancho de banda de 1024 kbps, ambas coexisten para brindar una capacidad de balanceo y redundancia, además de ofrecer servicios de Firewall ante la conexión pública al internet.

Como el nodo principal físicamente se encuentra en la sede Guayaquil, que es la matriz de la empresa, la conexión física hacia el primer ISP es por fibra óptica, mientras que la segunda es por radiofrecuencia.

Toda la red se basa en el protocolo IP, específicamente el encapsulamiento PPP[14] para utilizar la conexión a internet para la interoperabilidad; es decir que en cada agencia existe un modem con conexión telefónica configurado con encapsulamiento PPPoE[15] y así lograr una conexión con la matriz en Guayaquil y teniendo así acceso total a los servicios de la red empresarial.

3.3.1 MATRIZ

Como se describió anteriormente la matriz de la empresa en cuestión se encuentra en Guayaquil, siendo este un servidor basado en FreeBSD, compilación 10.2; que cuenta con servicios como:

- Firewall
- Packet Filter
- PROXY
- SSH
- DHCP
- VPN
- CARP

La matriz provee conectividad entre las redes de Guayaquil, Quito, Cuenca y la DMZ, introduciendo servicios de correo, transferencia de archivos, servicio de nombres de dominio y servicios VPN a través de la empresa.

También permite una conectividad VPN a través del internet hacia el servidor central para controlar las prestaciones de toda la red; además de emitir de forma automática informes de monitoreo de la red, puesto que se ha implementado el monitoreo con el protocolo SNMPv2 y SNMPv3.

El nodo central se basa enteramente en UNIX, básicamente la distribución libre de la Universidad de Berkeley en California (FreeBSD 10.2), integrando todos los protocolos anteriormente descritos y ofreciendo los servicios citados. El ordenador central utilizado para este propósito es un Intel Pentium de núcleo simple con frecuencia de 2.4GHZ, disco duro de 100GB y memoria RAM de 1024MB.

Cabe recalcar que la distribución de FreeBSD utilizada, se encuentra con entorno gráfico GNOME 3, lo que requiere mayor espacio de memoria RAM, en los casos en los que se utilice netamente comandos, suele requerirse únicamente 512MB de RAM. Sin embargo, se cuenta con 1GB; además de que existen otras distribuciones

[14] Protocolo de conectividad WAN con nombre de usuario y contraseña.
[15] Protocolo de conectividad WAN, similar a PPP, pero sobre Ethernet.

BSD, como OpenBSD o NetBSD, considerado este último como el más seguro en reglas de firewall y filtrado de paquetes.

3.3.2 DMZ

Una DMZ[16] es una red desmilitarizada, es decir con conectividad tanto al internet, como a la red privada empresarial; el objetivo de la red es proveer los diferentes servicios disponibles entre los diferentes usuarios que comparten recursos.

Además de estar disponible para la red privada todo el tiempo, es accesible desde el internet, es decir contiene direcciones IP públicas a las que se puede recurrir para obtener servicios de la DMZ desde fuera de la red corporativa, como por ejemplo un servidor web, que aloja la página principal de la compañía en el internet o un servidor de correo que corresponda a un dominio MX[17] para aceptar y enviar correos entre dominios en el internet.

Como el propósito del proyecto se basa en la funcionalidad del sistema UNIX y sus protocolos convergentes, no se aplicarán ejemplos de direcciones de dominio públicas, pues el objeto es simular los diferentes servicios DNS, VPN, MAIL, FTP, etc. dentro de una red operativa empresarial. Para el servicio de nombres de dominio, se implementa un ordenador, que está basado en un sistema Linux CentOS 6.7, el mismo que está configurado sobre una plataforma de hardware que contiene un procesador Intel Pentium de núcleo simple con una frecuencia de 2.4GHZ, 40GB de disco duro y 512MB de memoria RAM. El propósito del servidor es intercambiar direcciones IP por nombres de los servidores para los servicios que ofrece la compañía.

En la figura 3.1, se puede apreciar que se utilizará la red general 172.16.0.0/16 para distribuir direcciones entre las redes disponibles, cuya distribución para la red DMZ es la subred 172.16.30.0/24.

- DNS 172.16.30.100
- FTP 172.16.30.5
- VPN 172.16.30.200
- WEB 172.16.30.7
- MAIL 172.16.30.3

El servidor FTP, está basado en un sistema Linux CentOS 6.7, implementado sobre un hardware con procesador Intel Pentium de núcleo simple con frecuencia de

[16] Zona de red desmilitarizada.

[17] Dominio MX que responde al servidor de correo para la recepción de correos de dominios exteriores.

2.4GHZ, 80GB de disco duro y 512MB de memoria RAM; cuyo fin es el intercambio de archivos entre las diferentes sucursales de la empresa.

El servidor VPN, también está basado en sistema Linux CentOS 6.7, sobre un hardware con procesador de núcleo simple Intel Pentium de 2.4GHZ de frecuencia, con 50GB de disco duro y 512MB de memoria RAM; el servicio será utilizado para el acceso exclusivo de los administradores y el jefe de TI en casos de emergencia.

El servidor WEB, funciona sobre una plataforma CentOS 6.7; con un hardware con procesador de núcleo simple Intel Pentium de 2.4 GHZ de frecuencia, con 50GB de disco duro y 512MB de memoria RAM; el servicio será utilizado por toda la compañía para el ingreso de ventas e información de clientes a la base de datos.

El servidor MAIL, está basado también en plataforma CentOS 6.7; con un hardware con procesador de simple núcleo Intel Pentium de 2.4GHZ de frecuencia, con 80GB de disco duro y 512MB de memoria RAM; es el encargado de distribuir correos a través de todo el dominio privado.

Los servicios de la DMZ están disponibles para toda la red, tomando en cuenta las políticas de acceso establecidas en el servidor central, ya que es en el nodo de Guayaquil, donde se encuentra el Firewall también.

En los sistemas UNIX, existen dos módulos en kernel de ejecutan procesos de filtrado y bloqueo, que son IPFIREWALL (IPFW) y el Packet Filter (PF), ambos coexisten para cumplir objetivos de políticas de acceso, cifrado de información, traducciones de direcciones ip, QoS, balanceo de carga, etc.

Para comprender mejor la estructura de la topología del nodo central se indica la imagen a continuación.

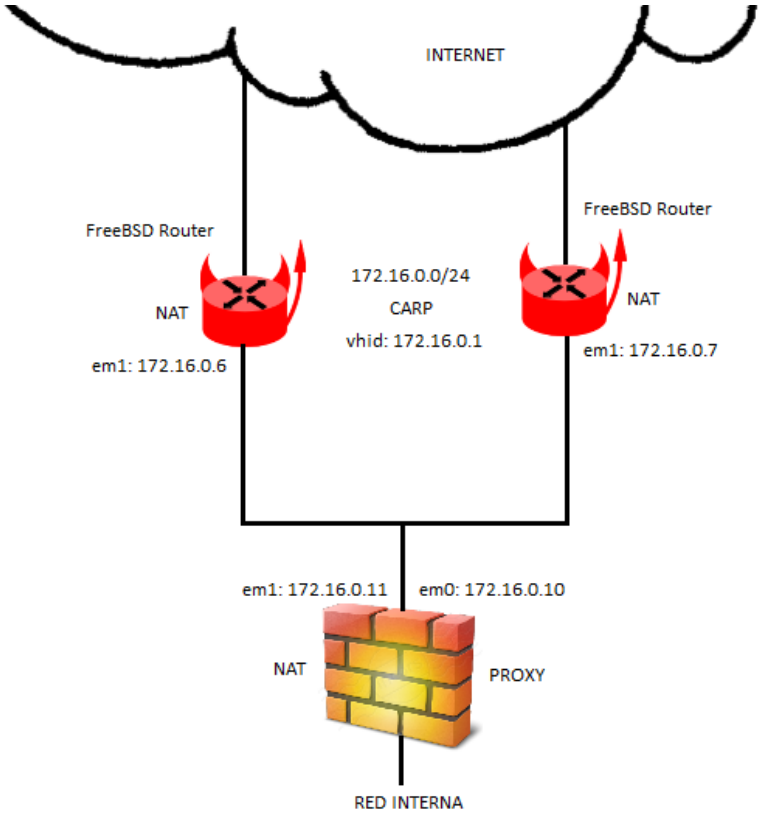

Figura 3.3: Distribución de redundancia para la red interna.

Como se aprecia en la figura el servicio de proxy también es algo embebido en el servidor central, puesto que forma parte de las políticas de acceso web que utilizan la conexión de internet de los empleados en todo el país.

3.3.3 RED GENERAL

La red general corresponde a las diferentes porciones de usuarios ubicados en diferentes locaciones geográficas, lo cual dificultaría la comunicación entre sí de no existir los sistemas de comunicación presentes para compartir recursos de forma económica, segura y fiable sin licenciamientos.

La red empresarial presente en varias ciudades del país, tiene la siguiente distribución de red:

- Quito 172.16.40.0/24
- Guayaquil 172.16.20.0/24
- Cuenca 172.16.10.0/24

Como la matriz es Guayaquil, únicamente el departamento de TI de esa locación podrá acceder al servidor y modificar su configuración; en Quito, en cambio, solamente tendrá acceso al nodo central para propósitos de monitoreo y reportes.

Como en la ciudad de Cuenca, es un proyecto recién en desarrollo, no se cuenta todavía con departamento de TI, por lo que los usuarios únicamente tienen acceso a internet filtrado y los servicios ofrecidos por la red desmilitarizada empresarial.

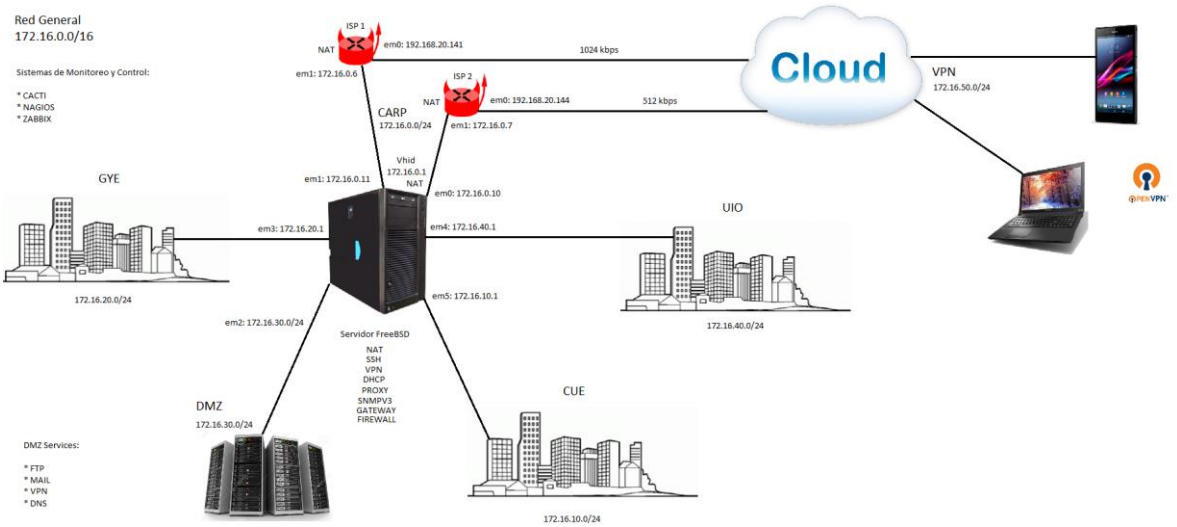

Figura 3.4: Topología de red empresarial a nivel nacional.

En la figura 3.1 se puede apreciar la topología general de la red en producción, así como los servicios ofrecidos tanto por el nodo central como de la red DMZ; también se indica el acceso remoto a través de VPN al servidor central para propósitos de monitoreo y control.

3.3.4 VPN

En el gráfico de la red general, se puede apreciar el objeto de la topología fuera del internet; es decir la configuración de un servicio vpn encriptado entre el cliente y el servidor que permita el acceso remoto del cliente a través del internet a un servidor de direccionamiento privado dentro una red corporativa.

Para el proyecto en cuestión se ha implementado un servidor VPN a base de certificados cifrados y socket seguros, llaves públicas y privadas con algoritmos de encripción que permite al usuario externo, conectarse al servidor de manera segura, sin poner en peligro su información y datos enviados.

El acceso remoto le permitirá al jefe de la compañía revisar, eventualmente, informes de monitoreo, configuraciones, prestaciones, LOGS[18], etc.

De modo fácil, rápido y seguro; se puede acceder a la red empresarial en cualquier momento y trabajar normalmente desde la distancia con un servicio VPN

[18] Eventos que se registran en un servicio interno del sistema.

multiplataforma, ya que desde el celular también se puede ingresar a la VPN y realizar pequeñas configuraciones rápidas desde allí, acortando el tiempo de respuesta ante eventos de emergencia como caídas de enlaces o cortes de servicio ahorrando en costosos enlaces dedicados. La red VPN en la compañía utiliza el espacio de direcciones 172.16.50.0/24.

Por propósitos de seguridad se verá reducida la máscara de subred en /30 para permitir únicamente dos hosts, que serían en este caso el jefe del departamento de TI y el gerente de la empresa.

3.4 REDUNDANCIA

Este es un tema delicado debido a que gracias a la redundancia se pueden ofrecer servicios 24/7, lo que quiere decir que funcionan las 24 horas del día; por ello, se debe tomar en cuenta las acciones a realizar para mantener la disponibilidad de dichos servicios.

3.4.1 REDUNDANCIA LÓGICA

Básicamente, la redundancia lógica trata sobre mantener la disponibilidad de una conexión a internet, por ejemplo, de forma lógica, es decir, que ante un incidente físico en los enlaces existentes, no haga falta intervención del usuario para conmutar el circuito, sino que automáticamente se activa la redundancia y los usuarios de la capa de acceso no notan ninguna caída de servicio.
Esto se puede lograr en la actualidad con algunos protocolos tanto de desarrollo libre como licenciado, como por ejemplo:

De uso libre multiplataforma

- GLBP
- CARP
- VRRP

De uso licenciado

- HSRP
- BGP

3.4.2 REDUNDANCIA ELÉCTRICA

En los sistemas de telecomunicación actuales, la redundancia eléctrica es extremadamente crucial al momento de las transmisiones dedicadas.

Por ello se emplean equipos UPS[19] o estabilizadores de corriente que ayudan a brindar a los equipos la corriente necesaria para su correcto funcionamiento, ya que toda la red eléctrica pública es susceptible a fallas. Es en ese momento cuando los sistemas de corriente continua entran en acción y mantienen la disponibilidad del servicio hasta el restablecimiento normal de la energía eléctrica, no obstante los equipos también han tomado parte de este concepto añadiendo varias fuentes de poder a un mismo equipo servidor.

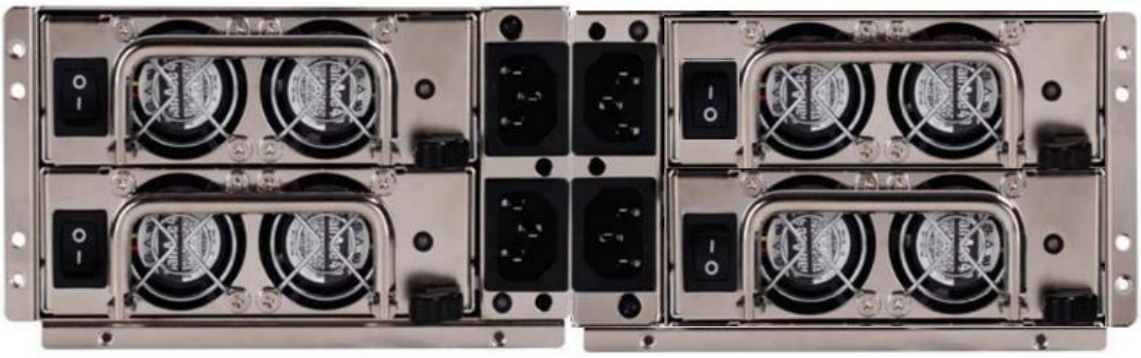

Figura 3.5: Fuentes de poder redundantes para servidores.

En la figura 3.5 se puede observar un grupo de fuentes redundantes para un servidor de broadcast de televisión; las cuales deberían conectarse a cuatro diferentes líneas de corriente con UPS diferentes en cada punto para mantener los servicios del nodo principal, ya que como se indicó al principio, el principal defecto de la topología de red en estrella, es que ante la caída del nodo central se produciría un DoS[20] en toda la red empresarial.

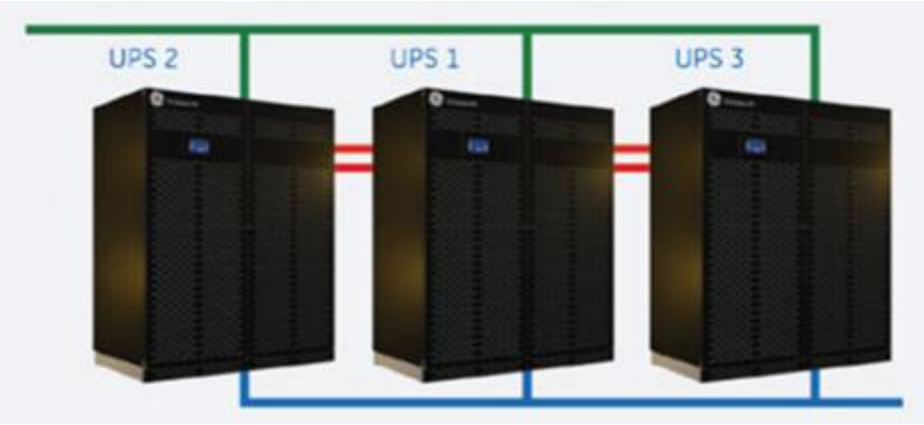

Figura 3.6: Sistema de UPS con redundancia.

[19] Uninterruptible Power Supply.
[20] Denied of Service, el resultado final de un ataque informático.

CAPÍTULO 4: MONITOREO, GESTIÓN Y CONTROL

4.1 GESTIÓN DE REDES

El tema central del proyecto se basa en los requerimientos actuales de desarrollo empresarial, para promover el trabajo remoto con el fin de ahorrar recursos económicos. Por eso, se requiere un completo dominio de la tecnología disponible y es por ello que existe el monitoreo de la misma.

Con el monitoreo se lograr visualizar el rendimiento de la red, las tazas de utilización de recursos, la horas pico de oferta de servicios y los accesos de las diferentes áreas de la compañía al servidor.

Es por esto, que en virtud al software libre, algunas compañías han desarrollado herramientas de monitoreo sin licenciamiento, que pueden ofrecer prestaciones cercanas a los programas de monitoreo más desarrollados, como son Solar Winds o *PRTG NETWORK MONITORING*.

Si bien es cierto estas herramientas ofrecen versatilidad en la convergencia de servicios, resultan soluciones muy costosas para pequeñas y medianas empresas que quieran participar del servicio SNMP, sin embargo existen herramientas sin licenciamiento que permiten mantener un registro de los sucesos en la red, en cuanto a tráfico de datos se refiere, para tener un control estadístico de la misma y así lograr programar cambios en la misma que proyecten siempre a una mejora de la topología actual.

4.1.1 ZABBIX

Creado por Alexei Vladishev en el año de 1998, es uno de los software de monitoreo de uso libre más utilizados en el mundo, por su escalabilidad y adaptación multiplataforma. Su código fuente está basado en el lenguaje C++ y su interfaz administrativa web, se basa en lenguaje PHP. (Vladishev, 2016)

La última versión de este sistema es la 3.0, compatible con todos los sistemas operativos y algunas distribuciones de Linux de tipo hogar y pequeñas empresas.
Para el proyecto en cuestión, se utilizará la versión 3.0.1 compatible con sistemas FreeBSD/UNIX.

Puesto que el nodo central será el servidor, se tomará en cuenta a los servidores de la DMZ, a los ISP´s y a los usuarios de capa de acceso, como clientes SNMP.
La versión 3 del protocolo SNMP con autenticación, es aplicable a ciertos sistemas operativos y puede coexistir con la versión 2, sin embargo supone cierta latencia en

la red general debido a los procesos de seguridad implícitos, como cifrado y algoritmos de encripción.

Cada agente de la comunidad SNMP, responde a las peticiones de información del servidor y se despliega toda la información requerida en una interfaz web de fácil administración.

Figura 4.1: Interfaz web gráfica administrable de Zabbix.

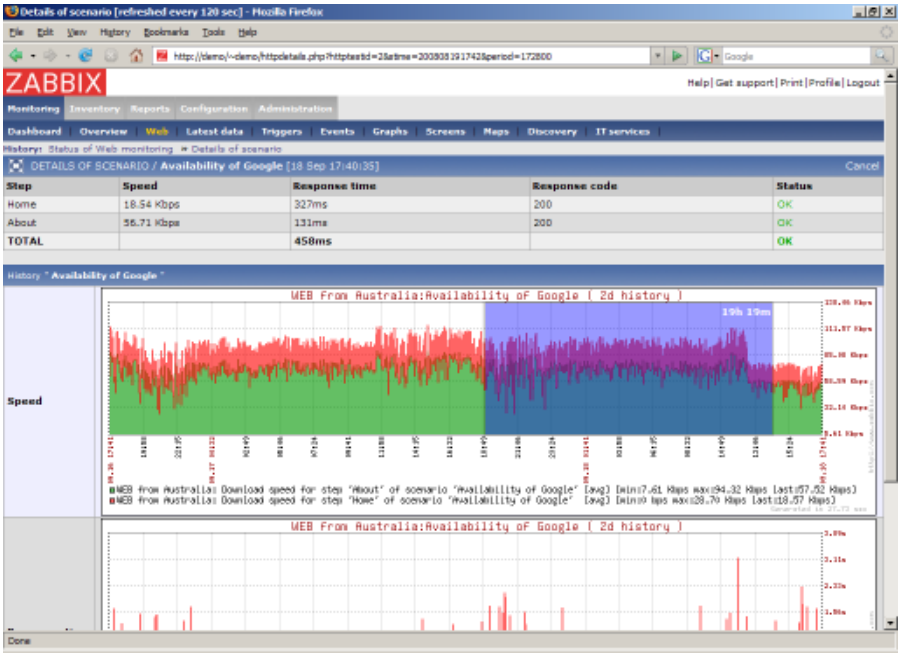

Figura 4.2: Interfaz web gráfica administrable de Zabbix.

4.1.2 CACTI

Proyecto iniciado por Ian Berry, en el año 2001, como resultado de un trabajo conjunto entre varias personas a través de los años, todas ellas, inspiradas por el desarrollo de software libre. Han utilizado la interfaz web de modo gráfico y lenguaje PHP para implementar un software de monitoreo gratuito, que al igual que ZABBIX, es bastante popular en el mundo de las redes en producción. (Pink, 2016)

Este software es multiplataforma y su versión más reciente es la 0.8.8g lanzada en el segundo mes del presente año.

Al igual que en ZABBIX, se utilizan ambas versiones del protocolo SNMP para monitorear toda la red operativa, es decir, las versiones 2 y 3; en este caso el nodo central también contendrá toda la información de los clientes SNMP que reporten utilización de recursos para posteriormente elaborar un informe detallado, con gráficas interpretativas del tráfico, anchos de banda, utilización de recursos y servicios, etc.

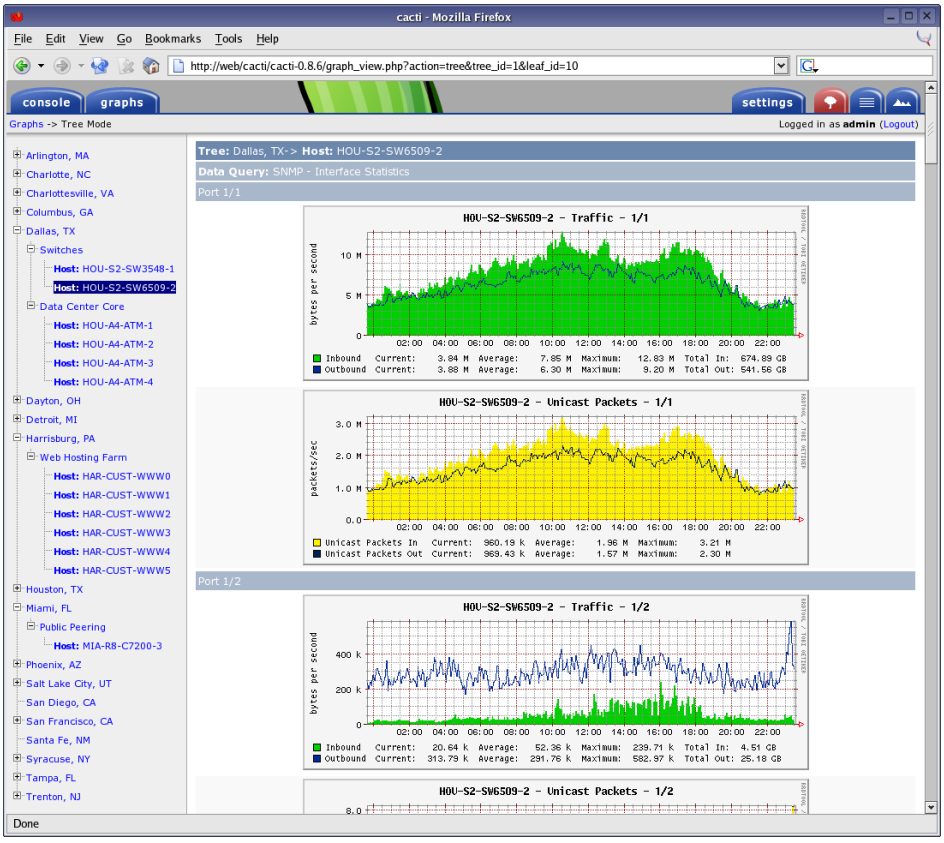

Figura 4.3: Interfaz web gráfica administrable de Cacti.

4.1.3 NAGIOS

Proyecto lanzado por primera vez en el año 1999, dirigido por Ethan Galstad y un grupo de alianza de software libre, al igual que, ZABBIX y CACTI; NAGIOS es un software de código libre, utilizado también para el monitoreo de red, utiliza protocolos PHP, HTTP, POP3, SNMP. (NAGIOS, 2009)

El software es multiplataforma también y con una característica adicional; es más compatible con los ambientes virtuales, pues ciertamente el proyecto que se basa en una simulación de red, requiere mayor interacción del programa de esta manera.

La última versión estable, está disponible en su página web y se trata de la versión Core 4.x.x que añade más complementos para los ambientes virtualizados, además de utilizar el protocolo SNMPv3 para añadir seguridad al monitoreo.

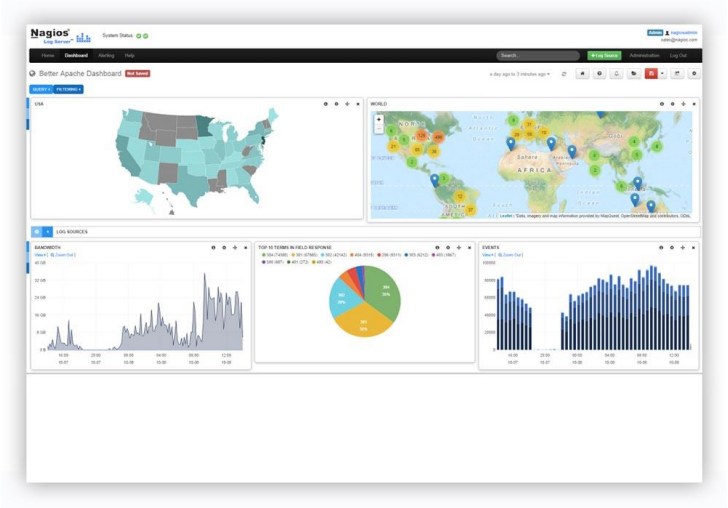

Figura 4.4: Interfaz web gráfica administrable de Nagios.

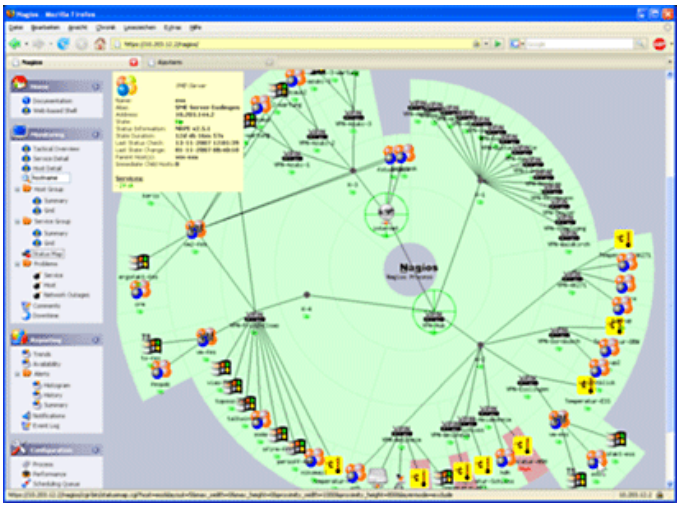

Figura 4.5: Interfaz web gráfica administrable de Nagios.

4.1.4 APLICACIÓN

Para la red operativa definida en capítulos anteriores y para el proyecto en cuestión, se ha aplicado la última versión más estable del sistema de monitoreo ZABBIX, que es la 3.0.1.

Zabbix es una plataforma de monitoreo, con interfaz gráfica web, con administración de base de datos integrada; soporta encripción, autenticación, certificados TLS, firmas digitales, etc. Todas estas características añadidas en esta versión para proteger el envío de información entre agente y servidor, evitando así ataques con analizador de protocolos, sobre todo por administrar actualmente el nuevo protocolo SNMPv3.

Utiliza los puertos 10050 y 10051 tanto en TCP como en UDP, por lo que se tendrían que crear permisos de tráfico por esos puertos en el firewall; también incluye características de predicción que proporciona acciones correctivas, como copiar datos de discos por llenarse, esto hace referencia de uso cuando las acciones correctivas del problema superan las diez horas y éstas acciones pueden llegar a modo de alertas por correo electrónico.

Dado que los historiales de monitoreo son largos y pesados; la plataforma elimina automáticamente de la base de datos, información obsoleta, obviamente configurable; es integrable además con todos los servicios de los sistemas operativos compatibles, que son:

- Linux
- FreeBSD
- OpenBSD
- IBM
- HP-UX
- Mac OS X
- Solaris
- Windows

De igual manera es compatible con algunos sistemas de bases de datos, como:

- MySQL
- PostgreSQL
- Oracle
- SQLite

El nodo central tendrá el rol de servidor de monitoreo, por lo que se encargará de administrar el protocolo SNMP para la interacción entre servidor y clientes; para

posteriormente elaborar un informe completo acerca de la utilización de recursos de la red, así como la presentación de gráficas de rendimiento y alertas de conectividad con los dispositivos añadidos al monitoreo.

Para la instalación del sistema, se requieren ciertos servicios añadidos al sistema FreeBSD, para la correcta ejecución de la plataforma web:

- Apache 2.4
- Php 5.6
- Mysql 5.6
- Zabbix Frontend
- Zabbix Server
- Zabbix Agent

Todos los servicios anteriormente indicados, interactúan entre sí, para proporcionar el seguimiento a los agentes que envían información continua al servidor central. El mismo que almacena los datos en la base de datos y a través de la interfaz web y por último es presentada al usuario.

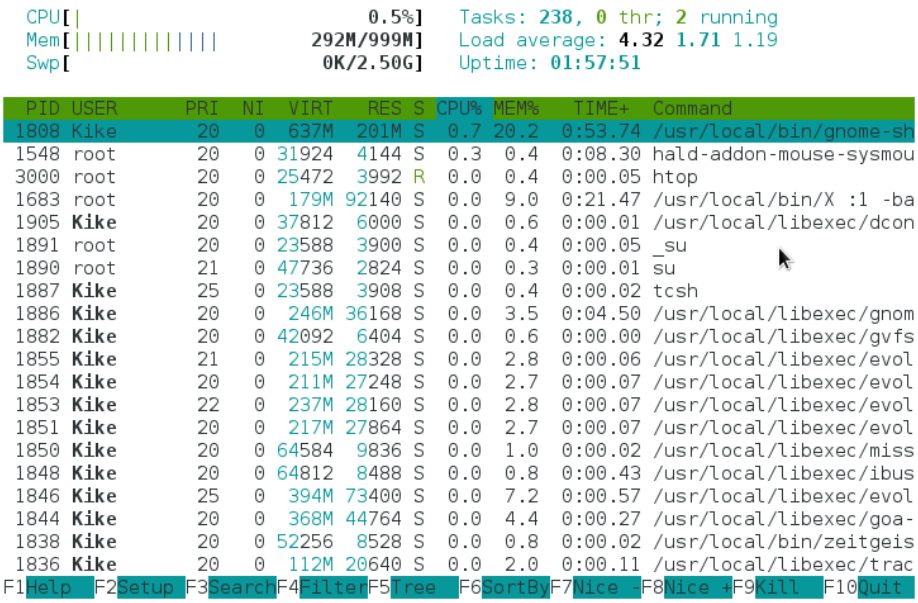

Figura 4.6: Herramienta HTOP para sistemas GNU.

Como primer punto, se tiene la instalación de los servicios, para ello, se debe tener actualizada la distribución FreeBSD a utilizar; por ello debemos ejecutar los siguientes comandos de actualización del sistema.

- Portsnap Fetch
- Portsnap Extract

- Portsnap Update

El primer comando permite actualizar todos los puertos del sistema UNIX. Se llaman puertos a los paquetes de programas instalables para el sistema, no tiene ninguna referencia con puertos de la capa transporte.

```
root@FreeBSD:~ # uname -a
FreeBSD FreeBSD 10.2-RELEASE-p12 FreeBSD 10.2-RELEASE-p12 #1: Wed May 11 00:33:1
1 ECT 2016     Kike@ToshibaBSD:/usr/obj/usr/src/sys/GENERIC  amd64
root@FreeBSD:~ # portsnap fetch
Looking up portsnap.FreeBSD.org mirrors... 7 mirrors found.
Fetching snapshot tag from your-org.portsnap.freebsd.org... done.
Fetching snapshot metadata... done.
Updating from Fri May 27 23:04:40 EDT 2016 to Thu Jun  2 20:31:08 EDT 2016.
Fetching 4 metadata patches... done.
Applying metadata patches... done.
Fetching 0 metadata files... done.
Fetching 463 patches.
(463/463) 100.00%  done.
done.
Applying patches...
done.
Fetching 26 new ports or files... done.
root@FreeBSD:~ #
```

Figura 4.7: Comandos de actualización del sistema FreeBSD.

Una vez descargados los paquetes para el sistema, el segundo comando extrae e instala los archivos de todos los paquetes de actualización; para posteriormente, verificarlos y compilarlos con el tercer comando.

```
root@FreeBSD:~ # portsnap extract
/usr/ports/.arcconfig
/usr/ports/.gitattributes
/usr/ports/.gitignore
/usr/ports/CHANGES
/usr/ports/CONTRIBUTING.md
/usr/ports/COPYRIGHT
/usr/ports/GIDs
/usr/ports/Keywords/desktop-file-utils.ucl
/usr/ports/Keywords/fc.ucl
/usr/ports/Keywords/fcfontsdir.ucl
/usr/ports/Keywords/fmtutil.ucl
/usr/ports/Keywords/fontsdir.ucl
/usr/ports/Keywords/glib-schemas.ucl
/usr/ports/Keywords/info.ucl
/usr/ports/Keywords/kld.ucl
/usr/ports/Keywords/rmtry.ucl
/usr/ports/Keywords/sample.ucl
/usr/ports/Keywords/shared-mime-info.ucl
/usr/ports/Keywords/shell.ucl
/usr/ports/Keywords/terminfo.ucl
```

Figura 4.8: Comandos de actualización del sistema FreeBSD.

Ahora se tiene la instalación del servicio web, para los sistemas UNIX, se utiliza el bien conocido demonio APACHE, cuya programación permite visualizar paginas HTML en la web e interactuar con las mismas, a través de texto, imágenes o links hacia otras páginas. La versión por defecto que viene instalada en el sistema es la 1.8; sin embargo, la plataforma de monitoreo requiere una versión más actual del demonio APACHE para poder interpretar el contenido PHP en la web.

```
root@FreeISP2:~ # pkg search apache
apache-ant-1.9.4               Java- and XML-based build tool, conceptually simi
lar to make
apache-forrest-0.9             Tool for rapid development of small sites
apache-mode.el-2.0             Emacs major mode for editing Apache configuration
 files
apache-openoffice-4.1.2_5      Integrated wordprocessor/dbase/spreadsheet/drawin
g/chart/browser
apache-openoffice-devel-4.2.1735889,4 Integrated wordprocessor/dbase/spreadsheet
/drawing/chart/browser (developer version)
apache-poi-3.14                Java API To Access Microsoft Format Files
apache-solr-5.2.1_1            High performance search server built using Lucene
 Java
apache-solr3-3.6.2             High performance search server built using Lucene
 Java
apache-spark-1.6.1             Fast big data processing engine
apache-struts-1.2.9            Apache Struts framework
apache-xml-security-c-1.7.2_1  Apache XML security libraries - C++ version
apache22-2.2.31                Version 2.2.x of Apache web server with prefork M
PM.
apache22-event-mpm-2.2.31      Version 2.2.x of Apache web server with event MPM
.
apache22-itk-mpm-2.2.31        Version 2.2.x of Apache web server with itk MPM.
apache22-peruser-mpm-2.2.31    Version 2.2.x of Apache web server with peruser M
PM.
apache22-worker-mpm-2.2.31     Version 2.2.x of Apache web server with worker MP
M.
apache24-2.4.20_1              Version 2.4.x of Apache web server
apachetop-0.12.6_4             Apache RealTime log stats
mkapachepw-1.121               Group & Password Management Tool For Apache
```

Figura 4.9: Búsqueda de paquetes del demonio APACHE.

Como se puede apreciar en el gráfico, se procederá con la versión 2.4 de APACHE para continuar con la instalación; posteriormente a ello, se debe instalar el servicio PHP.

```
root@FreeISP2:~ # pkg search php
beautifyphp-0.5.0              PEAR beautifier for PHP4
gitphp-0.2.8_1                 Web based git repository browser written in PHP
ja-php5-mecab-0.5.0            PHP5 extension for MeCab Morphological Analyzer
kdevelop-php-1.7.1             PHP support for KDevelop
kdevelop-php-docs-1.7.1        PHP documentation for KDevelop
libmrss-php-0.19.2_2           PHP library for parsing, writing, and creating RS
S
mod_php55-5.5.36               PHP Scripting Language
mod_php56-5.6.22               PHP Scripting Language
mod_php70-7.0.7                PHP Scripting Language
myphpmoney-1.3.r3,1            PHP script for managing your accounts
mysqlphp2postgres-0.95         Convert MySQL calls in a PHP page into PostgreSQL
 calls
p5-Apache-Session-PHP-0.05_1   Glue Apache::Session with PHP::Session
p5-HTML-WikiConverter-PhpWiki-0.51_1 Convert HTML to PhpWiki markup
p5-PHP-Serialization-0.34_1    Converting the output of PHP serialize() into the
 Perl
p5-PHP-Session-0.27_1          Read / write PHP session files
```

Figura 4.10: Búsqueda de paquetes del demonio PHP-FPM.

Una vez instalado el servicio PHP, se deben modificar ciertos parámetros en el archivo de configuración de inicio del servicio, el cual se llama php.ini.

```
root@FreeBSD:/usr/local/etc # ls | grep php
php
php-fpm.conf
php-fpm.conf.default
php.conf
php.ini
php.ini-development
php.ini-production
root@FreeBSD:/usr/local/etc # ee php.ini
```

Figura 4.11: Ubicación del archivo php.ini en sistemas FreeBSD.

```
=====line 665 col 0 lines from top 665 ======================================
; to proxy requests or to process the POST data in a memory efficient fashion.
; http://php.net/enable-post-data-reading
;enable_post_data_reading = Off

; Maximum size of POST data that PHP will accept.
; Its value may be 0 to disable the limit. It is ignored if POST data reading
; is disabled through enable_post_data_reading.
; http://php.net/post-max-size
post_max_size = 16M

; Maximum execution time of each script, in seconds
; http://php.net/max-execution-time
; Note: This directive is hardcoded to 0 for the CLI SAPI
max_execution_time = 300

; Maximum amount of time each script may spend parsing request data. It's a good
; idea to limit this time on productions servers in order to eliminate unexpecte
; long running scripts.
; Note: This directive is hardcoded to -1 for the CLI SAPI
; Default Value: -1 (Unlimited)
; Development Value: 60 (60 seconds)
; Production Value: 60 (60 seconds)
; http://php.net/max-input-time
max_input_time = 300

[Date]
; Defines the default timezone used by the date functions
; http://php.net/date.timezone
date.timezone = America/Bogota

; http://php.net/date.default-latitude
;date.default_latitude = 31.7667

; http://php.net/date.default-longitude
;date.default_longitude = 35.2333

; Always populate the $HTTP_RAW_POST_DATA variable. PHP's default behavior is
; to disable this feature and it will be removed in a future version.
; If post reading is disabled through enable_post_data_reading,
; $HTTP_RAW_POST_DATA is *NOT* populated.
; http://php.net/always-populate-raw-post-data
always_populate_raw_post_data = -1
```

Figura 4.12: Cambios necesarios en el archivo php.ini.

```
=====line 280 col 0 lines from top 280 ======================================
    # Controls who can get stuff from this server.
    #
    Require all granted
</Directory>

#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
<IfModule dir_module>
    DirectoryIndex index.php index.html index.html.var index.htm index.php3 inde
</IfModule>
<IfModule ssl_module>
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
</IfModule>

<FilesMatch "\ php $">
        SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$">
        SetHandler application/x-httpd-php-source
</FilesMatch>

Include etc/apache24/Includes/*.conf
# If you include a trailing / on /webpath then the server will
# require it to be present in the URL.  You will also likely
# need to provide a <Directory> section to allow access to
# the filesystem path.
Alias /zabbix /usr/local/www/zabbix3
<Directory "/usr/local/www/zabbix3">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
    Allow from all
</Directory>
```

Figura 4.13: Cambios necesarios en el archivo httpd.conf.

Posteriormente se continúa con la instalación del servicio de base de datos; para converger con la plataforma web en PHP, se requiere la versión 5.6 de MYSQL, con ello, el servidor FRONTEND de ZABBIX, puede recopilar la información de cada agente y almacenarla en la base de datos además de presentarla en la página web.

```
root@FreeISP2:~ # pkg search mysql
mysql2odbc-0.99.2_5            Openlink MySQL-ODBC Gateway
mysql2pgsql-1.2,1              Convert a MySQL dump to a PostgreSQL dump
mysql55-client-5.5.46          Multithreaded SQL database (client)
mysql55-server-5.5.46          Multithreaded SQL database (server)
mysql56-client-5.6.30          Multithreaded SQL database (client)
mysql56-q4m-0.9.13_1           Message queue that works as a pluggable storage e
ngine of MySQL
mysql56-server-5.6.30          Multithreaded SQL database (server)
mysql57-client-5.7.12          Multithreaded SQL database (server)
mysql57-server-5.7.12          Multithreaded SQL database (server)
mysqlbackup-2.8                Creates MySQL backups on a periodic basis
```

Figura 4.14: Búsqueda de paquetes del demonio MYSQLD.

Una vez instalado el demonio MYSQLD, es necesario crear la base de datos, ZABBIX, con un usuario y contraseña y con ello gestor los datos recogidos desde los agentes hacia el servidor; sin olvidar dar permisos de acceso a la base de datos desde la plataforma web y a los usuarios que vayan a manipular el sistema.

```
root@FreeISP2:~ # mysqladmin -u root password kkike775
Warning: Using a password on the command line interface can be insecure.
root@FreeISP2:~ # mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.30 Source distribution

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database zabbix character set utf8 collate utf8_bin;
Query OK, 1 row affected (0.09 sec)

mysql> grant all privileges on zabbix.* to zabbix@localhost identified by 'zabbi
x';
Query OK, 0 rows affected (0.05 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Figura 4.15: Creación de base de datos, usuario y contraseña de Zabbix.

```
root@FreeBSD:/usr/local/share/zabbix3/server/database/mysql # ls
data.sql        images.sql      schema.sql
root@FreeBSD:/usr/local/share/zabbix3/server/database/mysql # cat schema.sql ima
ges.sql data.sql | mysql -u zabbix -p zabbix
Enter password:
root@FreeBSD:/usr/local/share/zabbix3/server/database/mysql #
```

Figura 4.16: Comandos de asociación de tablas a la base de datos MYSQL.

```
root@FreeBSD:~ # service apache24 status
apache24 is running as pid 1413.
root@FreeBSD:~ # service php-fpm status
php_fpm is running as pid 1093.
root@FreeBSD:~ # service mysql-server status
mysql is running as pid 1311.
root@FreeBSD:~ #
```

Figura 4.17: Comandos para verificar que los servicios se encuentren activos.

Una vez actualizado el sistema, se podrá proceder con la instalación del servidor ZABBIX así como el cliente y la interfaz gráfica FRONTEND.

```
root@FreeISP2:~ # pkg search zabbix
zabbix2-frontend-2.0.16_1      Enterprise-class open source distributed monitori
ng (frontend)
zabbix2-proxy-2.0.16_1         Enterprise-class open source distributed monitori
ng (proxy)
zabbix2-server-2.0.16_1        Enterprise-class open source distributed monitori
ng (server)
zabbix22-agent-2.2.11_1        Enterprise-class open source distributed monitori
ng (agent)
zabbix22-frontend-2.2.11_1     Enterprise-class open source distributed monitori
ng (frontend)
zabbix22-proxy-2.2.11_1        Enterprise-class open source distributed monitori
ng (proxy)
zabbix22-server-2.2.11_1       Enterprise-class open source distributed monitori
ng (server)
zabbix24-agent-2.4.7_1         Enterprise-class open source distributed monitori
ng (agent)
zabbix24-frontend-2.4.7_1      Enterprise-class open source distributed monitori
ng (frontend)
zabbix24-proxy-2.4.7_1         Enterprise-class open source distributed monitori
ng (proxy)
zabbix24-server-2.4.7_1        Enterprise-class open source distributed monitori
ng (server)
zabbix3-agent-3.0.1_2          Enterprise-class open source distributed monitori
ng (agent) LTS
zabbix3-frontend-3.0.1_2       Enterprise-class open source distributed monitori
ng (frontend) LTS
zabbix3-proxy-3.0.1_2          Enterprise-class open source distributed monitori
ng (proxy) LTS
zabbix3-server-3.0.1_2         Enterprise-class open source distributed monitori
ng (server) LTS
root@FreeISP2:~ #
```

Figura 4.18: Búsqueda de paquetes asociados a ZABBIX.

Una vez instalados todos los servicios, se debe modificar el archivo de configuración principal del servidor, para asociarlo a la base de datos creada y al usuario y contraseña para administrarla.

```
=====line 74 col 1 lines from top 74 ========================================
#       If set to empty string, socket is used for PostgreSQL.
#
# Mandatory: no
# Default:
DBHost=localhost

### Option: DBName
#       Database name.
#       For SQLite3 path to database file must be provided. DBUser and DBPasswor
#
# Mandatory: yes
# Default:
# DBName=

DBName=zabbix

### Option: DBSchema
# Mandatory: no
# Default:
# DBUser=

DBUser=zabbix

### Option: DBPassword
#       Database password. Ignored for SQLite.
#       Comment this line if no password is used.
#
# Mandatory: no
# Default:
DBPassword=zabbix

### Option: DBSocket
#       Path to MySQL socket.
#
```

Figura 4.19: Cambios necesarios en el archivo zabbix_server.conf.

```
root@FreeBSD:~ # chown -R zabbix /usr/local/www/apache24
root@FreeBSD:~ # chown -R zabbix /usr/local/www/zabbix3
root@FreeBSD:~ # chown -R www:www /usr/local/www/zabbix3
root@FreeBSD:~ # chown -R www:www /usr/local/www/apache24
root@FreeBSD:~ # chmod 777 /usr/local/www/apache24
root@FreeBSD:~ # chmod 777 /usr/local/www/zabbix3
root@FreeBSD:~ #
```

Figura 4.20: Comandos para administrar permisos al usuario y a la web.

```
root@FreeBSD:~ # sockstat -4 -l
USER     COMMAND    PID   FD PROTO  LOCAL ADDRESS         FOREIGN ADDRESS
www      httpd      54946 4  tcp4   *:80                  *:*
www      httpd      54009 4  tcp4   *:80                  *:*
www      httpd      14031 4  tcp4   *:80                  *:*
www      httpd      13464 4  tcp4   *:80                  *:*
www      httpd      11924 4  tcp4   *:80                  *:*
www      httpd      11513 4  tcp4   *:80                  *:*
www      httpd      8804  4  tcp4   *:80                  *:*
www      httpd      3801  4  tcp4   *:80                  *:*
www      httpd      3776  4  tcp4   *:80                  *:*
zabbix   zabbix_ser 2539  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2538  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2537  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2536  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2535  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2534  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2533  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2532  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2531  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2530  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2529  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2528  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2527  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2517  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2516  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2515  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2514  5  tcp4   127.0.0.1:10051       *:*
zabbix   zabbix_ser 2512  5  tcp4   127.0.0.1:10051       *:*
root     sendmail   2090  4  tcp4   127.0.0.1:25          *:*
www      httpd      1426  4  tcp4   *:80                  *:*
zabbix   zabbix_age 1017  5  tcp4   *:10050               *:*
zabbix   zabbix_age 1016  5  tcp4   *:10050               *:*
zabbix   zabbix_age 1015  5  tcp4   *:10050               *:*
zabbix   zabbix_age 1014  5  tcp4   *:10050               *:*
zabbix   zabbix_age 1007  5  tcp4   *:10050               *:*
root     syslogd    877   7  udp4   *:514                 *:*
unbound  unbound    728   5  udp4   127.0.0.1:53          *:*
unbound  unbound    728   6  tcp4   127.0.0.1:53          *:*
root@FreeBSD:~ #
```

Figura 4.21: Comando para verificar los puertos abiertos para cada servicio.

```
=====line 1 col 0 lines from top 1 ========================================
hostname="FreeBSD"
ifconfig_em0="inet 172.16.0.10 netmask 255.255.255.0"
ifconfig_em0_ipv6="inet6 accept_rtadv"
dhcpd_enable="YES"
inetd_enable="YES"
portmap_enable="YES"
gateway_enable="YES"
hcpd_ifaces="em3"
defaultrouter="172.16.0.1"
gdm_enable="YES"
gnome_enable="YES"
dbus_enale="YES"
hald_enable="YES"
sshd_enable="YES"
moused_enable="YES"
ntpd_enable="YES"
powerd_enable="YES"
firewall_enable="YES"
mysql_enable="YES"
apache24_enable="YES"
mysql_args="--bind-address=127.0.0.1"
zabbix_server_enable="YES"
zabbix_agentd_enable="YES"
ntpdate_enable="YES"
ntpdate_program="/usr/sbin/ntpdate"
ntpdate_flags="-u ru.pool.ntp.org"
php_fpm_enable=YES
```

Figura 4.22: Configuración por defecto en el archivo rc.conf.

Al final de la instalación de servicios y respectivas configuraciones, se tendrá la interfaz gráfica web de ZABBIX, en plena operación.

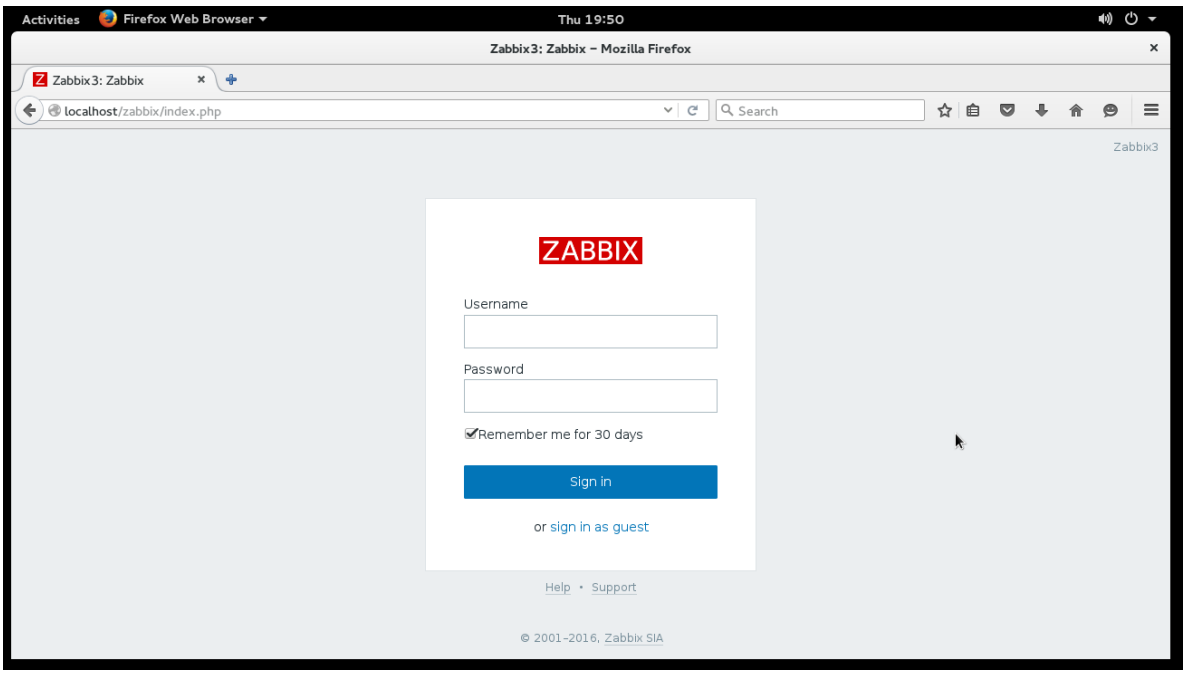

Figura 4.23: Interfaz web gráfica de ZABBIX FRONTEND.

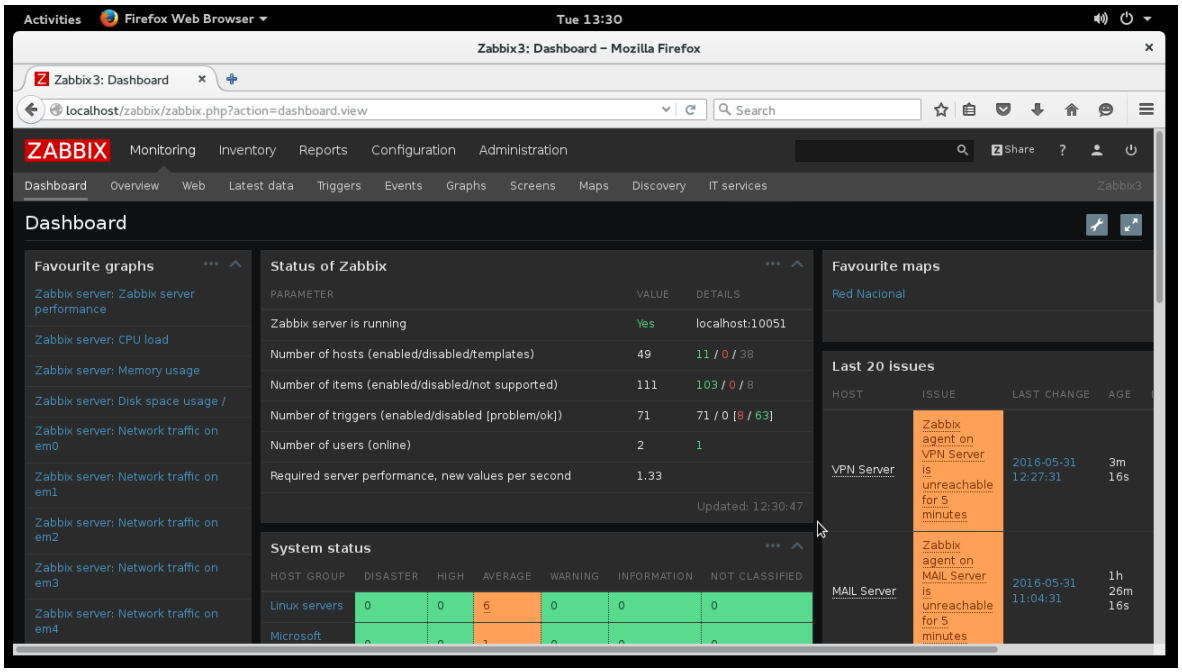

Figura 4.24: Dashboard de la página de monitoreo en la plataforma Zabbix.

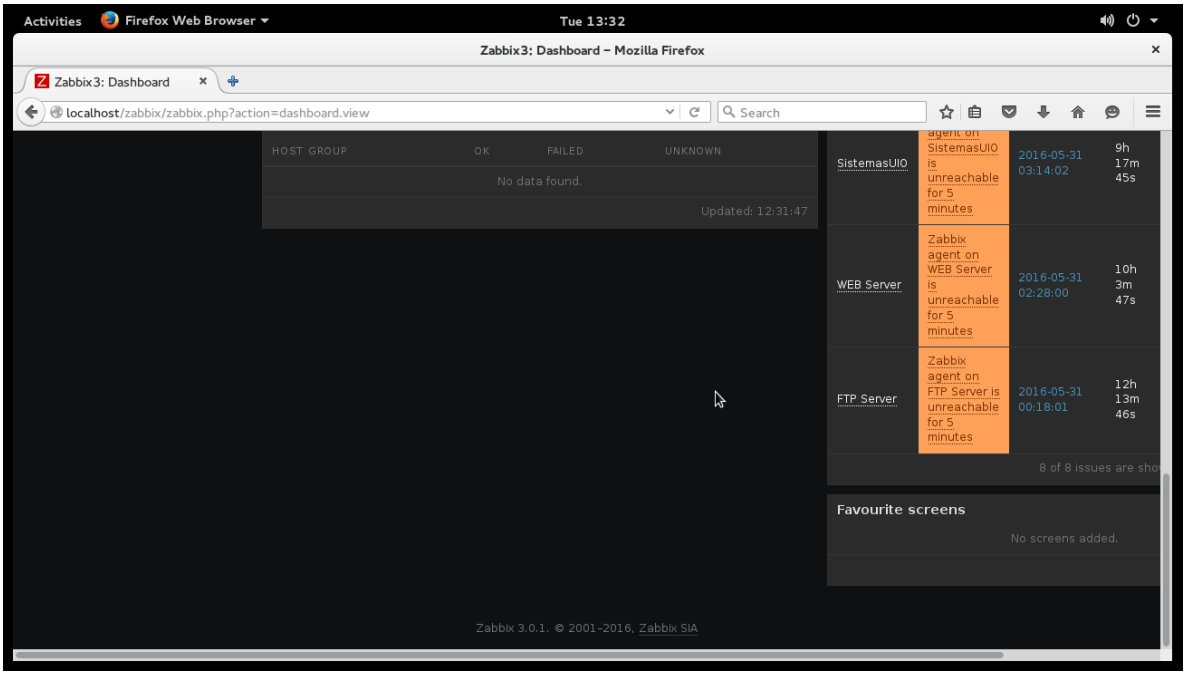

Figura 4.25: Dashboard de la página de monitoreo en la plataforma Zabbix.

Se procede agregar los hosts al sistema, asociados a un grupo de notificadores de conectividad así como de esquemas de la topología donde se pueden apreciar alarmas al verificar el mapa completo de la red.

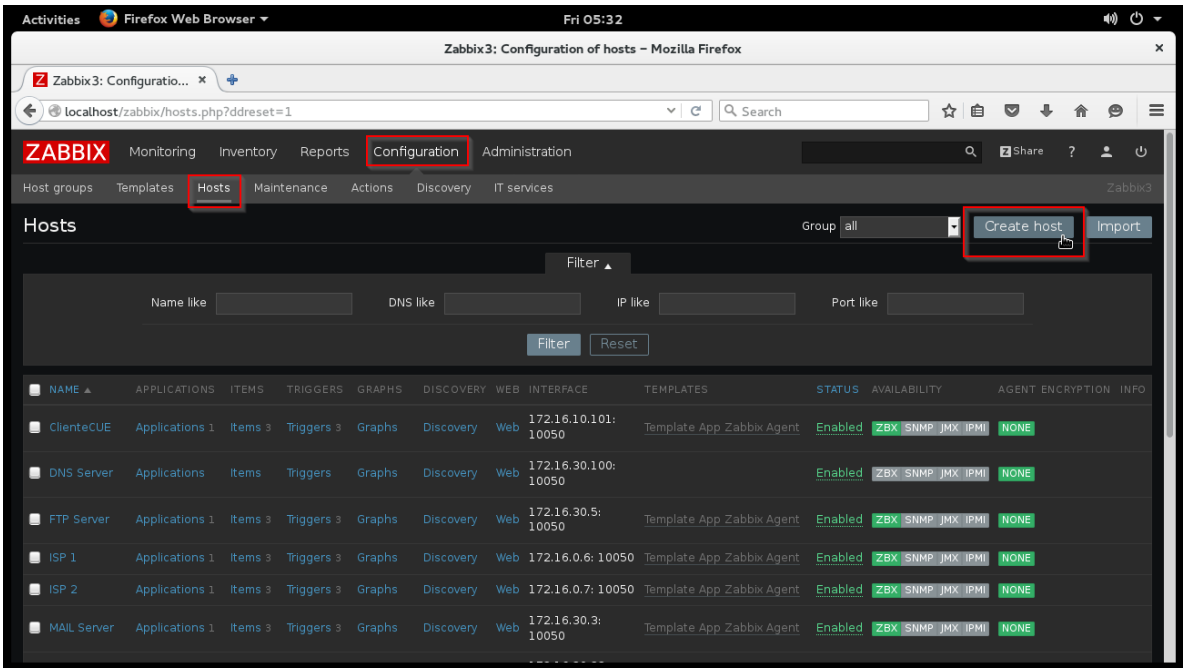

Figura 4.26: Página de administración de hosts o terminales clientes Zabbix.

En la topología actual, se pueden apreciar varios sistemas operativos, mismos que interactúan entre ellos a través del protocolo SNMP para dar seguimiento a todos los hosts que el administrador desee agregar a la red monitoreada.

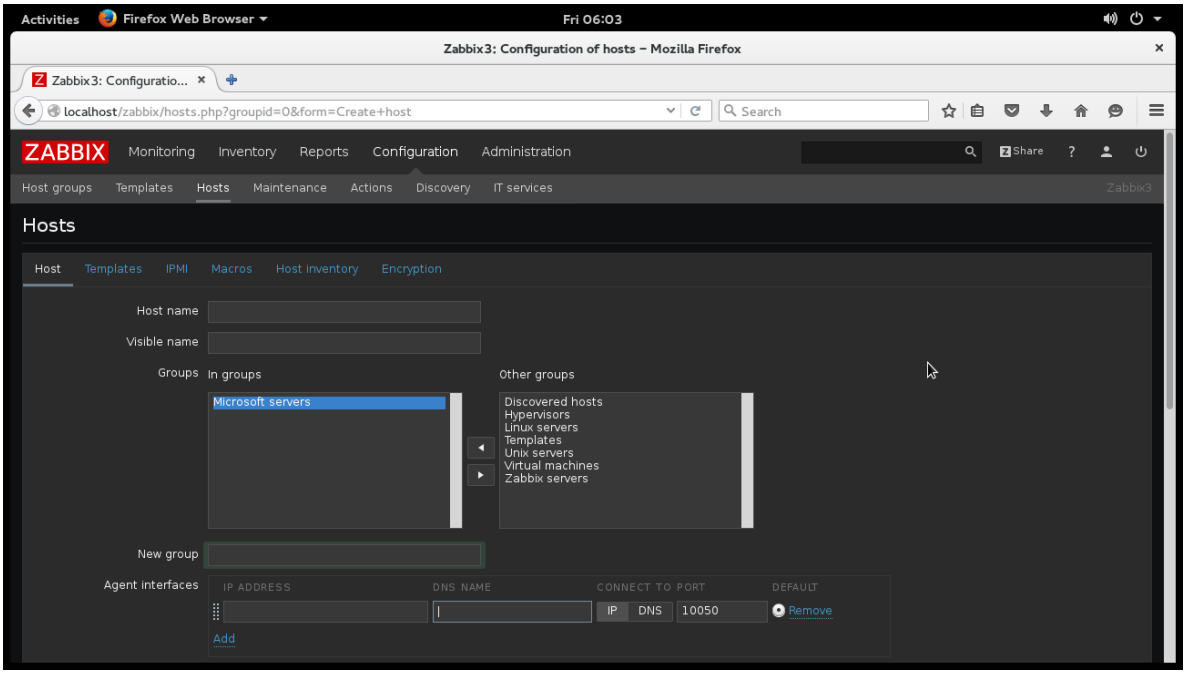

Figura 4.27: Página de hosts nuevos para ser monitoreados como clientes Zabbix.

También existe la pestaña Dashboard, misma que indica brevemente el estado de cualquier alarma o avería en el sistema de red monitoreado, además de mostrar gráficos de tráfico en las interfaces, utilización de servicios y cambios de configuración.

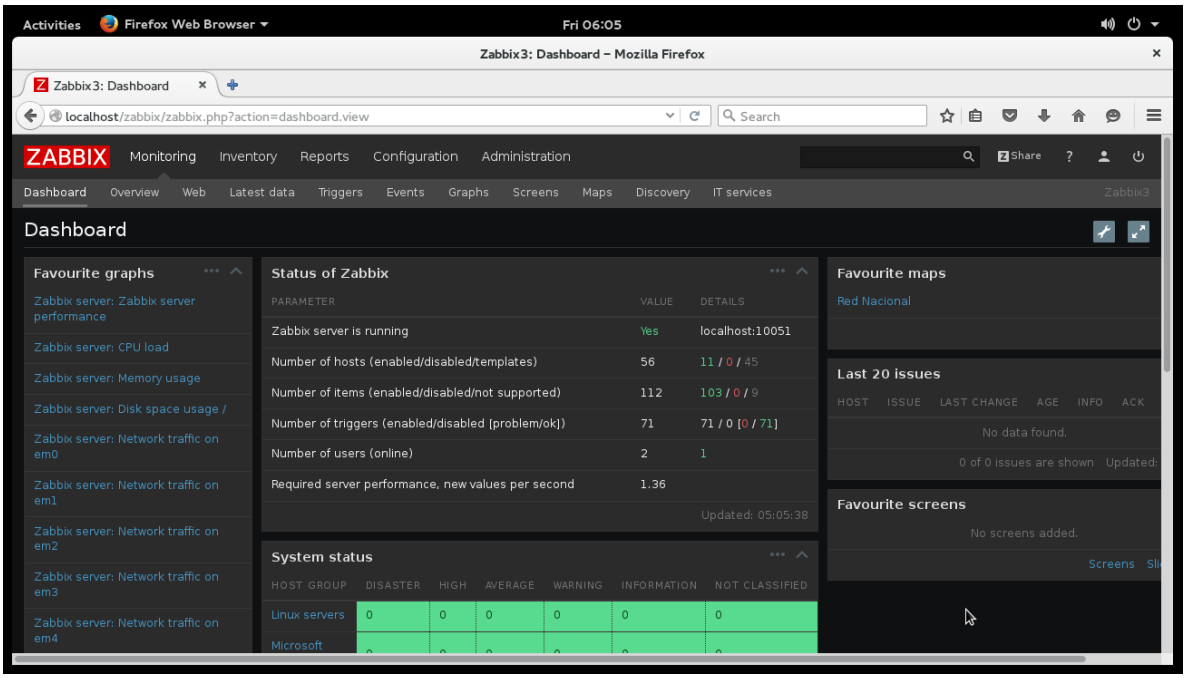

Figura 4.28: Dashboard principal de monitoreo en la plataforma Zabbix.

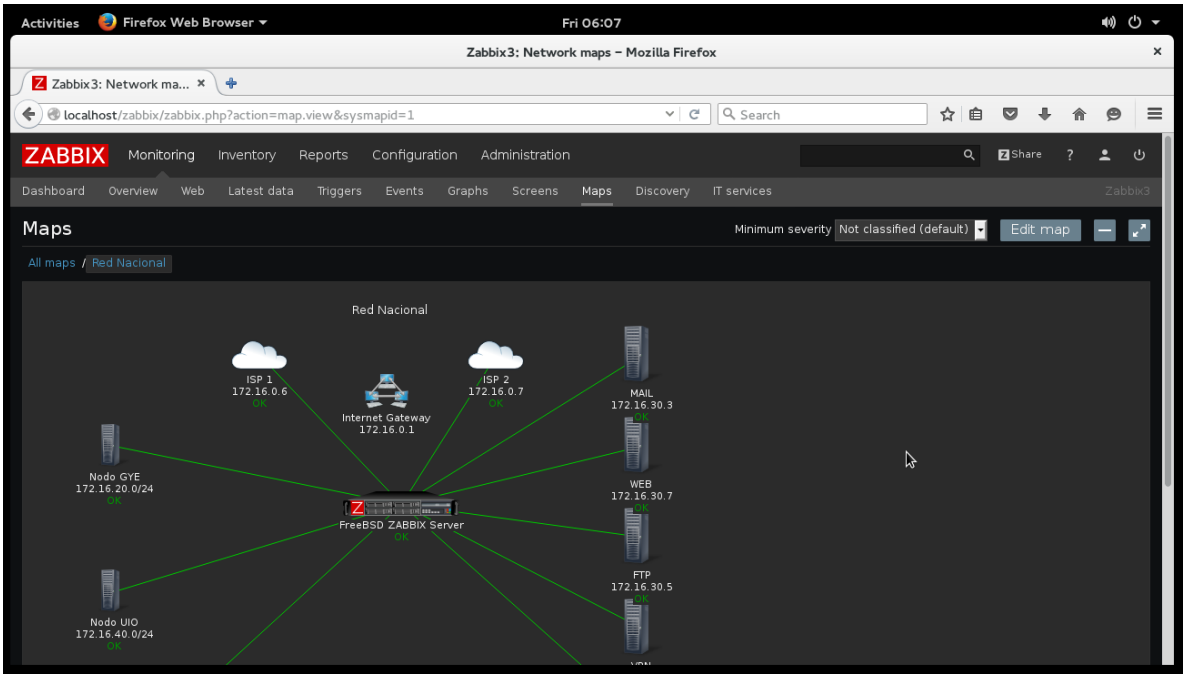

Figura 4.29: Página de administración de mapas de topologías de Zabbix.

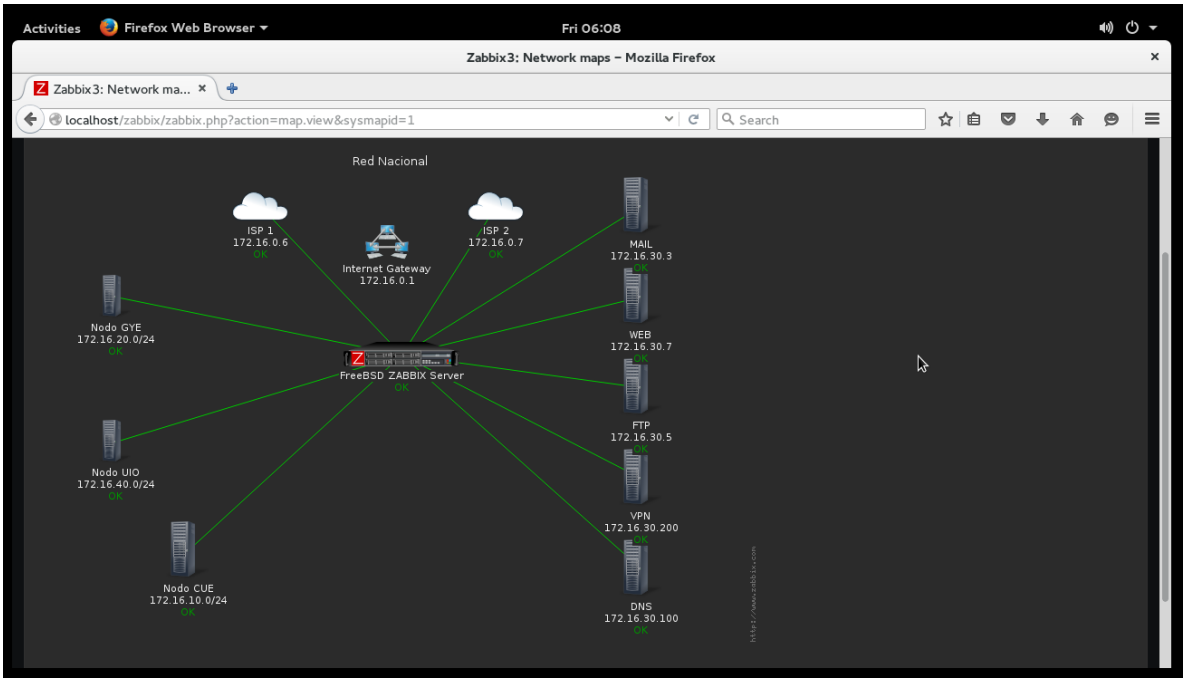

Figura 4.30: Página de administración de mapas de topologías de Zabbix.

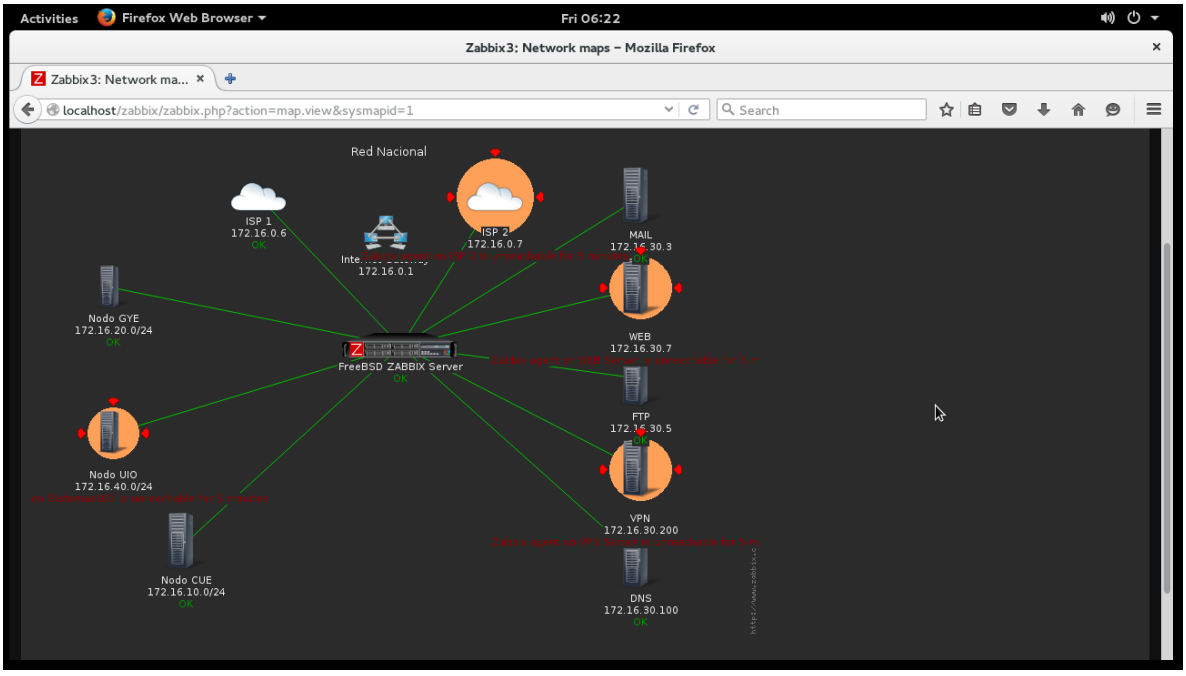

Figura 4.31: Página de administración de mapas de topologías de Zabbix.

En las figuras anteriores, se puede apreciar las diferentes alarmas que se pueden dar al monitorear la red en producción, de este modo el administrador de la red, puede tomar rápidas decisiones para corregir cualquier problema que se suscite, en cuanto a tráfico o procesamiento alto se refiera.

Dado que la compañía utiliza servicios web para ofertar sus productos, registrar cliente y pedidos; el servidor también es capaz de monitorear la utilización de la página web, así como del servicio que debe estar levantado y la disponibilidad del hardware.

En el proyecto se utiliza como referencia la página web *www.tesisudla.html*; cuyo código de hipertexto, está ubicado en el servidor web de la red DMZ, para compartir el servicio web a toda la red. El propósito de Zabbix en este escenario es entregar al usuario un esquema de utilización de recursos del servidor web, así como de su disponibilidad para mantener la página web de la empresa operativa para toda la red de la capa de acceso.

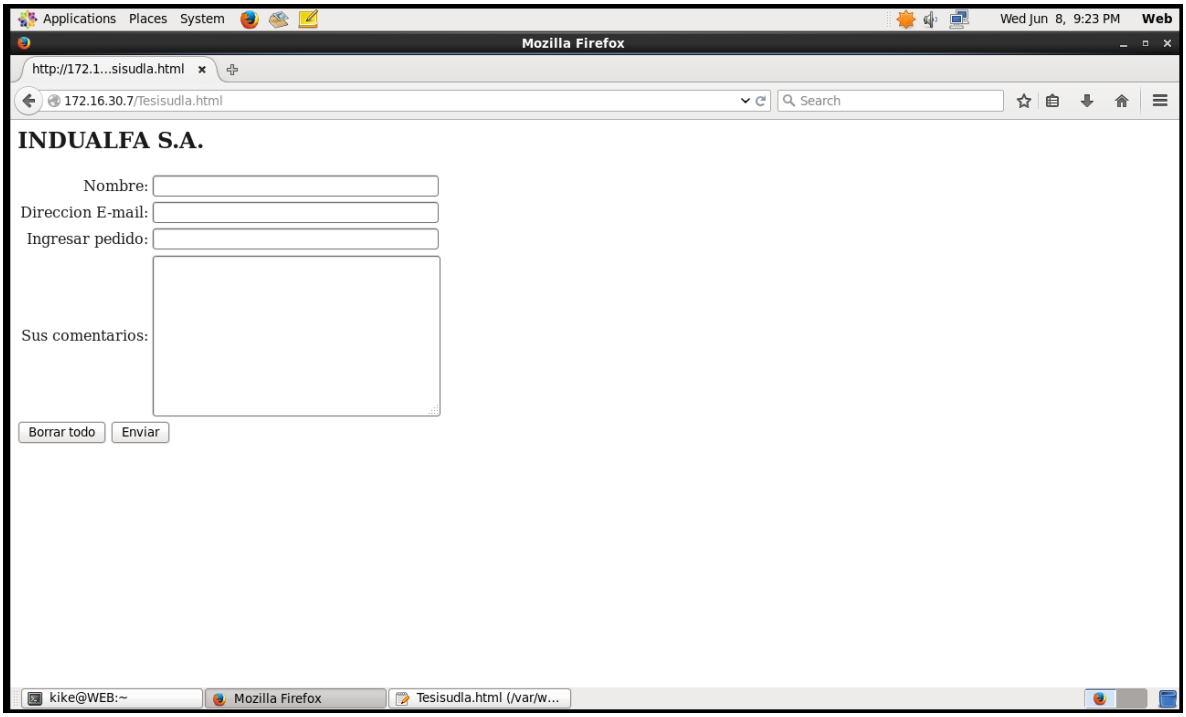

Figura 4.32: Página web principal de la PYME.

En la figura anterior, se puede apreciar un formulario de registro de pedidos para clientes de la empresa, cuyos datos se alojan en una base de datos; también es por donde se registran pedidos al exterior para llenar stock o registrar la información de clientes y sus compras, así como descuentos y direcciones de trabajo.

Para todo ello existe también un script, que respalda semanalmente la base de datos, de modo que se evita la pérdida de datos.

La redundancia también está implementada para este propósito, pues todo el tiempo la empresa debe estar conectada al internet porque además de tener contacto con clientes y proveedores a través de ese medio, se realizan también todas las

compras y pagos se realizan a través del nuevo sistema de pago electrónico, de modo que el internet es un recurso que no puede faltar ni desestabilizarse. Es por ello que la empresa justifica con esto sus dos enlaces físicos redundantes.

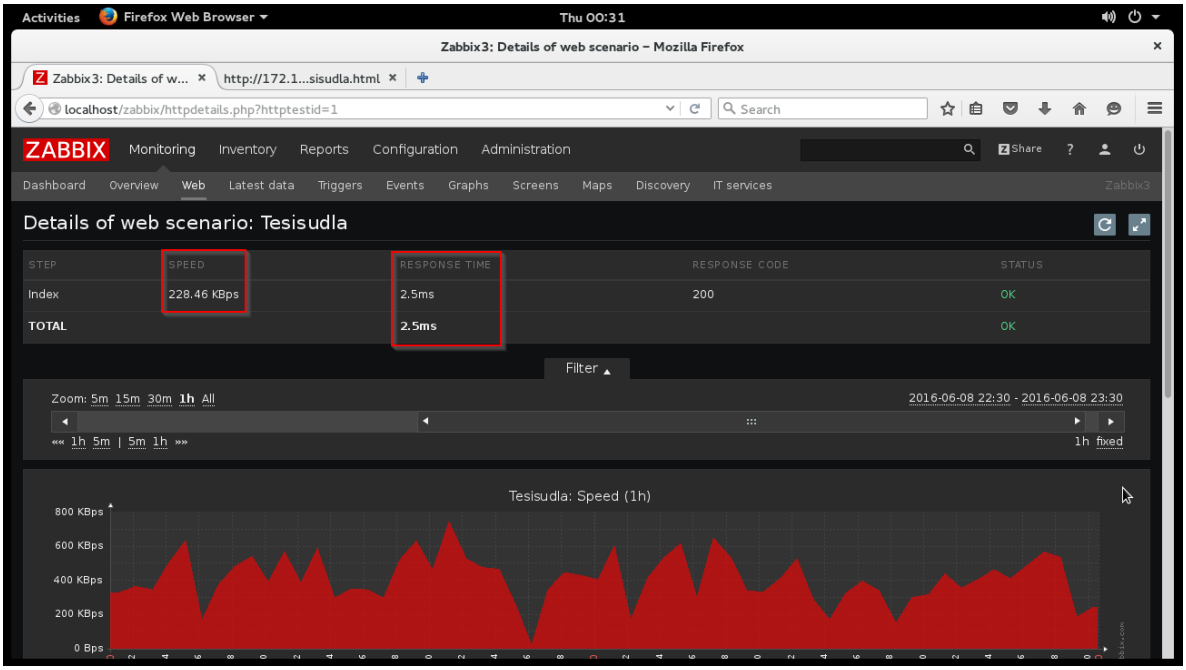

Figura 4.33: Monitor web de Zabbix para la página web de la PYME.

Figura 4.34: Monitor web Zabbix para la página web de la PYME.

4.2 OPTIMIZACIÓN

La optimización trata acerca del ahorro de los recursos disponibles en una red, ya sean estos de voz, video o datos; la ingeniería de la red demanda una buena gestión, pues en momento críticos se requieren acciones rápidas para mantener la disponibilidad de los servicios a la capa de acceso.

La optimización tiene su objeto con técnicas propuestas por el administrador de la red, como son:

- Priorizar el tráfico
- Usar VLAN´s
- Usar Proxy
- Aplicar QoS
- Implementar firewall

Como se indicó en el punto 4.1.4, el monitoreo logra adaptarse a todos estos servicios, pues muchas veces las redes operativas actuales utilizan estos servicios en sus organizaciones, mejorando así el desarrollo diario de la misma.

Asimismo, se deben programar mantenimientos periódicos a la misma, como revisar semanalmente los informes elaborados por los programas de monitoreo, revisar cableado estructurado y de ser necesario aplicar nuevas configuraciones que simplifiquen el procesamiento de los paquetes y las tramas a fin de aliviar la carga de la red.

Identificar la utilización de los servicios en hora punta, para así poder desarrollar planes y esquemas de prevención de DoS, lo cual retardaría mucho la recuperación de los servicios ante una caída de ese tipo.

Verificar continuamente la configuración de las interfaces de red inmersas en la conexión, pues muchas veces existen errores de configuración en ellas, lo que hace que los dispositivos no negocien correctamente la comunicación *full-duplex* o la sincronización de velocidad de transmisión de dato. En algunos dispositivos, como los Switches *Catalyst* de Cisco, que tienen seguridad de puertos, solo permiten la conexión de un solo equipo, o asignación de VLANS a las subredes para la comunicación en capa 2.

Un buen manejo de la red es la implementación de VLANS, puesto que limitamos a los dispositivos a la comunicación en capa 2, evitando la examinación de paquetes ip; apresurando las tramas Ethernet a través de la red; el protocolo RSTP de CISCO, permite en algunos dispositivos, priorizar la utilización de vlans y hacer la entrega de información inclusive más seguro que el protocolo TCP/IP.

4.3 SEGURIDAD

Uno de los aspectos más importantes de una red en producción; pues es aquí donde se aplican las normas de seguridad necesarias para proteger la información de la empresa; que viene a significar, su más valioso recurso.

Es importante tener siempre en una red empresarial una política de seguridad sustentada en normas de privacidad e integridad; mediante el uso de mecanismos de seguridad, se pueden mitigar varios posibles ataques a nuestra red ya que mediante la definición de las políticas de seguridad, se garantiza la disponibilidad de los recursos.

Entre los servicios descritos en la norma X.800, se tiene:

- Autenticación
- Control de Acceso
- Confidencialidad
- Integridad
- No repudio

Tomando en cuenta que cada servicio tiene su propio e indistinto mecanismo de seguridad, como pueden ser:

- Firmas digitales
- Cifrado
- Control de acceso
- Integridad
- Enrutamiento
- Notarización

En cuanto a los servicios, el no repudio hace referencia a garantizar que el receptor no rechace la comunicación desde el transmisor; tomando siempre en cuenta que cada servicio, tiene su propio mecanismo de seguridad.

La política de seguridad empresarial, es un documento que comunica a los usuarios tanto internos como externos, la forma de proceder con el manejo de la información, así como de la utilización de recursos informáticos de la misma. Siendo también un conjunto de requisitos definidos por los responsables de TI de la compañía, que estipulan lo que está permitido y lo que no.

Puesto que la seguridad tiene tres elementos principales a cuales enfocarse, se debe mantener una comunicación continua y constante con los usuarios finales, ya que es bien conocido que la vulnerabilidad viene de los eslabones más débiles de

la cadena empresarial, siendo estos los empleados administrativos y esto puede afectar gravemente a la disponibilidad e integridad de los servicios e información.

En el siguiente esquema se puede apreciar la equivalencia de mecanismo de seguridad a utilizar para los diferentes servicios de los que se dispone y que se desea proteger:

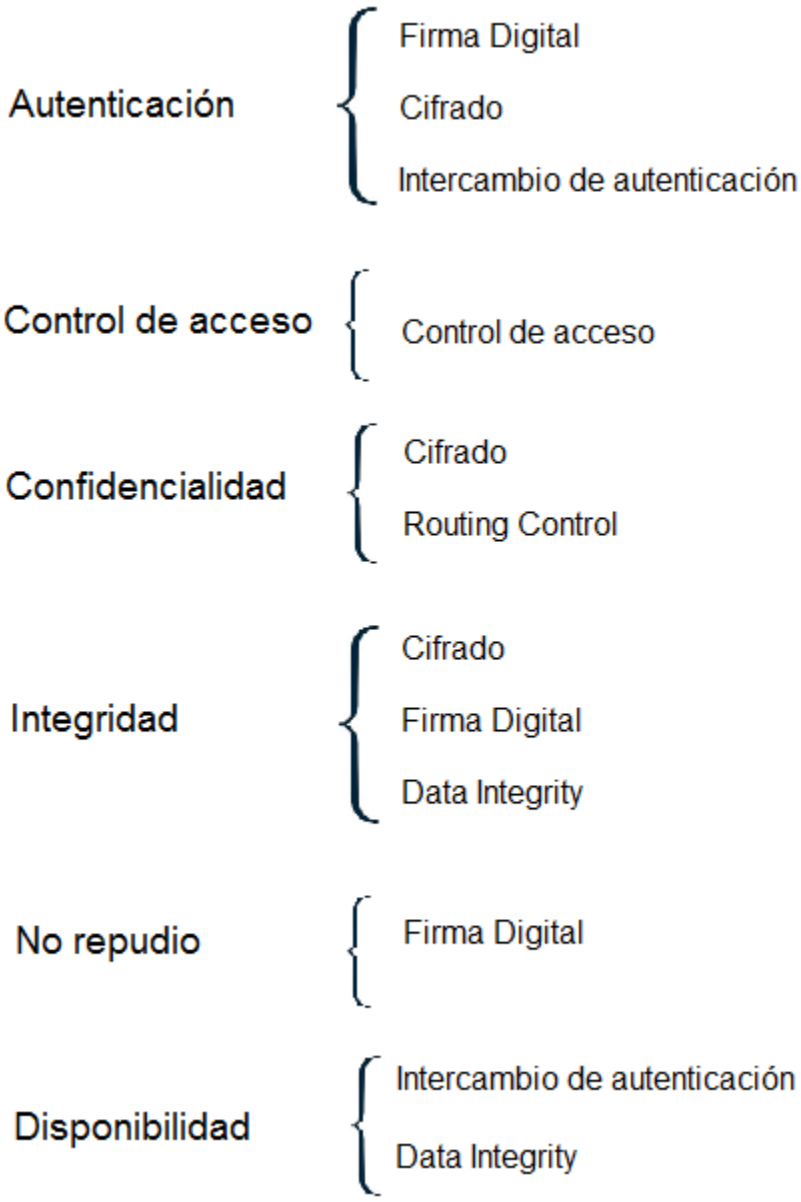

Figura 4.35: Esquema de la norma de seguridad X.800.

Para la encripción de los datos se puede usar algoritmos como:

- DES
- RC5
- IDEA
- AES

Para la protección de usuarios finales se puede utilizar técnicas como:

- Antivirus o Firewall
- Contraseñas Seguras
- Radius
- Active Directory
- Bloqueo de USB

Para la protección de servicios disponibles a nivel de red, se aplicarían métodos como:

- Firewalls
- Sistemas RAID
- Sistemas Logsys
- Actualizaciones
- Disaster Recovery

Los firewalls son útiles herramientas de seguridad, por sus características de filtrado de puertos y servicios, sin embargo hoy en día resulta bastante difícil elegir la solución correcta a nuestro sistema por su ambiente operativo y su costo; ya que en el mercado actual se tienen características costosas de las diferentes soluciones de firewall más relevantes, que son:

- CISCO ASA
- FORTINET
- CHECKPOINT

Algunas de las características que ofrecen algunas de estas potentes herramientas de seguridad, son:

- Filtrado de paquetes; que trabaja con los paquetes en in y out con dirección ip de origen y destino.
- Stateful inspections; que trabaja en las 3 primeras capas del modelo OSI, con la capacidad de recordar configuraciones prestablecidas en el firewall, en si un modo de firewall más inteligente.

- Firewall proxy; que trabaja a nivel de aplicación, manejando conexiones tcp a través de los diferentes puertos de servicios, además de proporcionar web filtering.

En el modelo de red empresarial propuesto, se implementa un esquema de seguridad de dos sentidos, ya que como se definió anteriormente, los sistemas UNIX cuentan con doble módulo de seguridad en su kernel, que son el IPFW el PF. Cada módulo debe activarse y compilarse mediante kernel para funcionar y es así como se tiene un esquema de seguridad doble en el servidor principal.

```
root@FreeBSD:~ # cat /etc/ipfw.conf
ipfw -q -f flush

ipfw -q add 1 allow ip from any to any via lo0
ipfw -q add 2 allow ip from any to any via tap0

ipfw -q add 3 allow tcp from any to any established
ipfw -q add 4 allow udp from any to me dst-port 1194
ipfw -q add 5 allow udp from me 1194 to any
ipfw -q add 6 allow all from any to any out keep-state

ipfw -q add 7 allow ip from any to me
ipfw -q add 8 allow ip from me to any
ipfw -q add 9 allow icmp from any to any

ipfw -q add 10 allow tcp from any to any 20 in
ipfw -q add 11 allow tcp from any to any 20 out
ipfw -q add 12 allow tcp from any to any 21 in
ipfw -q add 13 allow tcp from any to any 21 out
ipfw -q add 14 allow tcp from any to any 22 in
ipfw -q add 15 allow tcp from any to any 22 out
ipfw -q add 16 allow tcp from any to any 25 in
ipfw -q add 17 allow tcp from any to any 25 out
ipfw -q add 18 allow tcp from any to any 80 in
ipfw -q add 19 allow tcp from any to any 80 out
ipfw -q add 20 allow tcp from any to any 53 in
ipfw -q add 21 allow udp from any to any 53 in
ipfw -q add 22 allow tcp from any to any 53 out
ipfw -q add 23 allow udp from any to any 53 out
ipfw -q add 24 allow tcp from any to any 110 in
ipfw -q add 25 allow tcp from any to any 110 out
ipfw -q add 26 allow tcp from any to any 443 in
ipfw -q add 27 allow tcp from any to any 443 out
ipfw -q add 28 allow tcp from any to any 55000-65000 in
ipfw -q add 29 allow tcp from any to any 55000-65000 out
ipfw -q add 30 allow tcp from any to any 10050 in
ipfw -q add 31 allow tcp from any to any 10050 out
ipfw -q add 32 allow udp from any to any 10051 in
ipfw -q add 33 allow udp from any to any 10051 out
ipfw -q add 34 deny tcp from any to any
ipfw -q add 35 deny udp from any to any
root@FreeBSD:~ #
```

Figura 4.36: Figura de reglas para el FIREWALL.

```
root@FreeBSD:~ # cat /etc/pf.conf
ext1_if="em0"
ext2_if="em1"
extgye_if="em4"
extuio_if="em5"
extcue_if="em3"
vpn_port="1194"
vpn_if="tun0"
vpn_net="172.16.50.0/24"
redcue="{172.16.10.0/24}"
redgye="{172.16.20.0/24}"
reduio="{172.16.40.0/24}"
nat on $ext1_if from $vpn_net to any -> ($ext1_if)
nat on $ext1_if from $vpn_net to any -> $ext1_if
nat on $ext1_if from $redgye to any -> ($ext1_if) static-port
nat on $ext1_if from $extgye_if:network to any -> ($ext1_if)
nat on $ext1_if from $reduio to any -> ($ext1_if) static-port
nat on $ext1_if from $extuio_if:network to any -> ($ext1_if)
nat on $ext1_if from $redcue to any -> ($ext1_if) static-port
nat on $ext1_if from $extcue_if:network to any -> ($ext1_if)
#rdr on $ext1_if proto { tcp udp } from any -> $ext1_if port 1194
rdr on $extgye if inet proto tcp from any to any port 80 -> 127.0.0.1 port 3128
rdr on $extuio_if inet proto tcp from any to any port 80 -> 127.0.0.1 port 3128
#rdr on $extuio_if inet proto tcp from any to any port 443 -> 127.0.0.1 port 312
8
rdr on $extcue_if inet proto tcp from any to any port 80 -> 127.0.0.1 port 3128
#rdr on $extcue_if inet proto tcp from any to any port 443 -> 127.0.0.1 port 312
8
#pass on $extgye_if inet proto tcp from any to 127.0.0.1 port 3128
#pass on $extuio_if inet proto tcp from any to 127.0.0.1 port 3128
#pass on $extcue_if inet proto tcp from any to 127.0.0.1 port 3128
tcp_services="{ ssh,smtp, domain, www, ftp, pop3, auth, pop3s, 10050, 10051 }"
udp_services="{ domain, openvpn, 1194, 10050, 10051 }"
#antispoof quick for $ext1_if inet
#antispoof quick for $ext2_if inet
set skip on lo0
#pass in quick on $vpn_if inet from any to any keep state
#pass out quick on $vpn_if inet from any to any keep state
#pass in on $ext1_if proto udp from any to port 1194 keep state
#pass quick on $vpn_if
#pass in quick log on $ext1_if proto udp from any to port $vpn_port
#pass in quick log on $vpn_if proto { tcp udp } from any to any
pass out proto tcp to any port $tcp_services keep state
pass proto udp to any port $udp_services keep state
root@FreeBSD:~ #
```

Figura 4.37: Figura de reglas para el PACKET FILTER.

También se puede cifrar la conexión entre agente y servidor, lo que da un valor agregado a la seguridad ya establecida, se puede seleccionar un host cualquiera e implementarle un grupo de monitoreo SNMPv3.

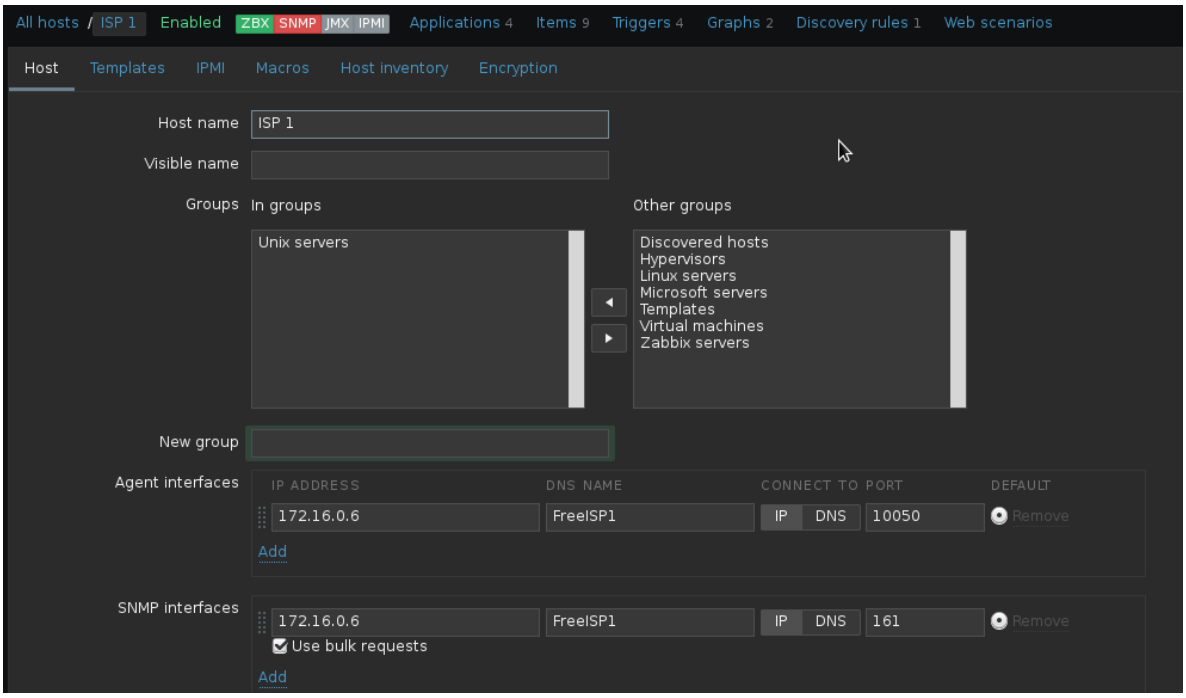

Figura 4.38: Figura de configuración SNMP para los hosts.

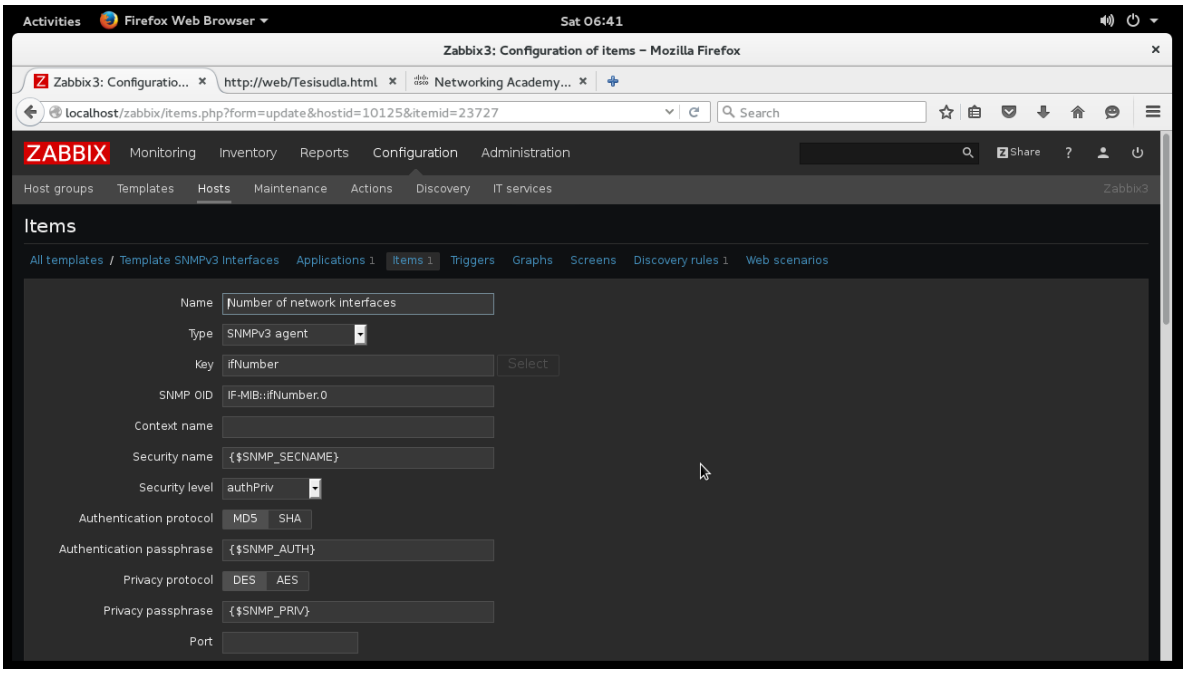

Figura 4.39: Figura de configuración SNMPv3 en cliente.

La comunicación entre agente y gestor, también resulta protegida, pues se pueden utilizar diferentes métodos de encripción, como certificados TLS o llaves PSK.

Primeramente se genera la llave secreta en el servidor, donde se genera automáticamente la clave PSK.

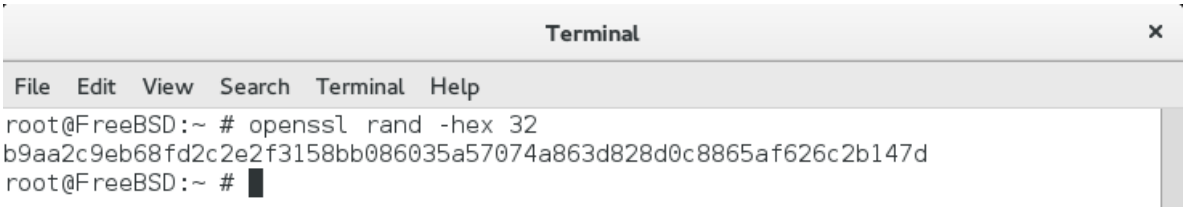

Figura 4.40: Comando para generar llave secreta PSK entre cliente y servidor.

Posteriormente se incluye la llave secreta en un archivo secreto en el cliente.

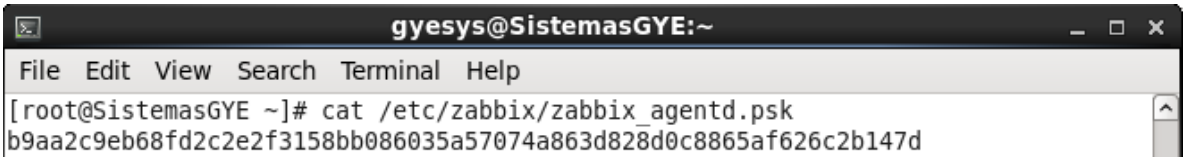

Figura 4.41: Verificación del archivo PSK en el cliente.

Como se puede apreciar en la figura anterior, el archivo de extensión PSK, incluye exactamente la misma clave PSK generada en el servidor; finalmente se configura el archivo *zabbix_agentd.conf* para que acepte la seguridad TLS por medio de PSK, como se puede apreciar en la siguiente figura, se modifican ciertos parámetros para que el servidor pueda interpretar la seguridad.

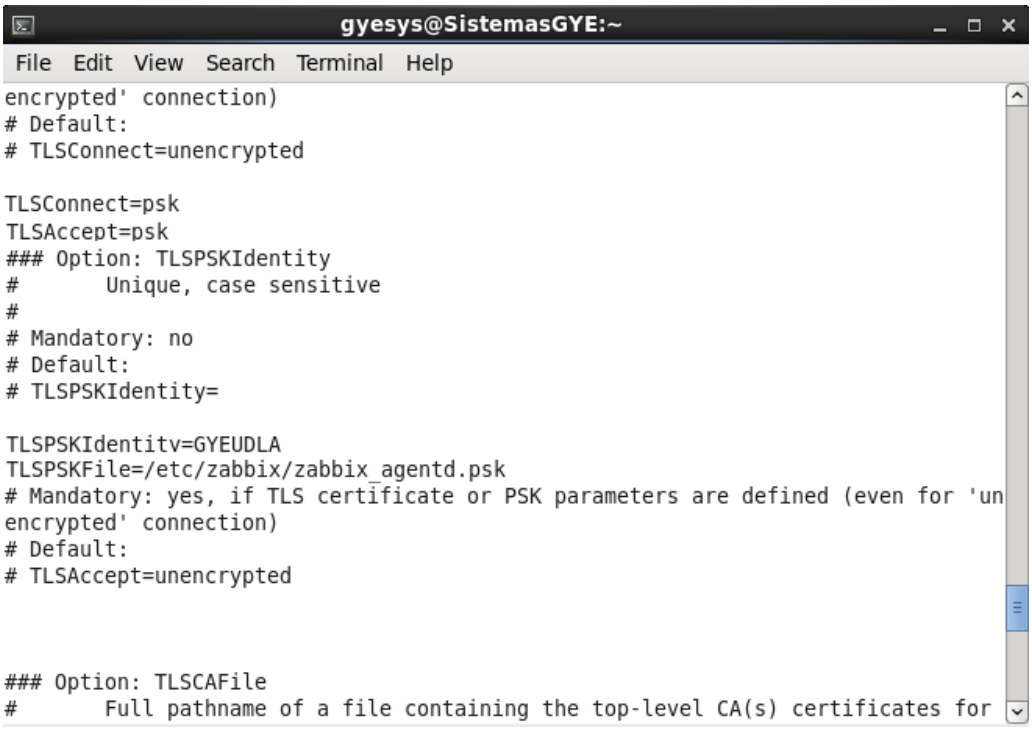

Figura 4.42: Parámetros necesarios para la configuración PSK.

Se aplica la configuración generada en la plataforma Zabbix, en la cual se agrega la seguridad del host y luego se prueba la configuración PSK.

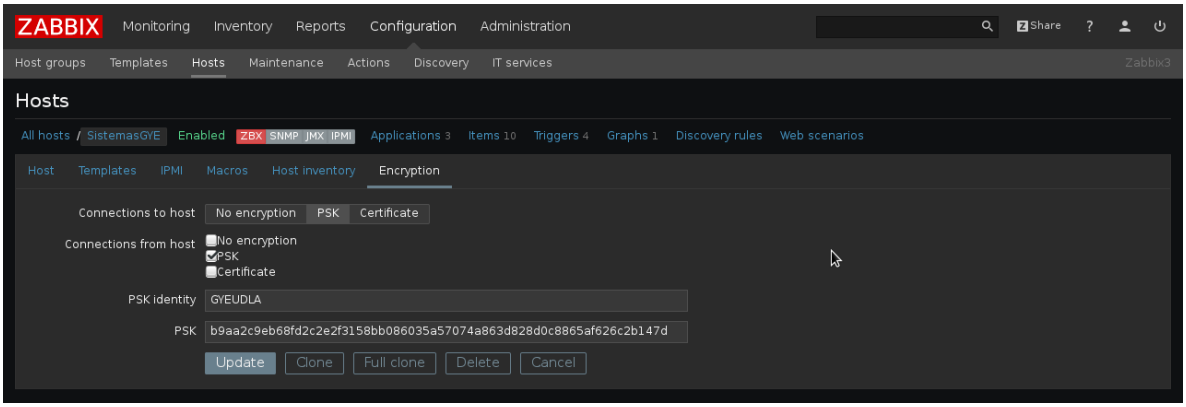

Figura 4.43: Configuración PSK para el agente desde el servidor.

4.4 APLICACIONES

En esta sección se definen las herramientas de seguridad aplicadas al sistema FreeBSD, ya que como cualquier empresa PYME, requiere cierto nivel de seguridad profesional para asegurar su recurso más valioso, la información.

Al tener todos los servicios en una red desmilitarizada, se garantiza la disponibilidad de los servicios inmersos en ella, también mediante un script implementado en el sistema principal, se copian los archivos nuevos y configuraciones adicionales generadas cada día en el mismo hacia otro disco duro, teniendo un espejo en caso de daños en el disco principal.

UNIX mantiene la integridad de los datos mediante el uso de un sistema de archivos diferente, además de interactuar únicamente con archivos específicos de configuración del sistema y que están sujetos a políticas de escritura y lectura.
La confidencialidad está asegurada en la conexión VPN mediante mecanismos de cifrado, encripción y la utilización de puertos SSL.

Se tiene instalado el servicio SYSLOG, para asegurar que todos los eventos de ingreso y modificación del sistema, se encuentren registrados para su eventual auditoría.

El nodo principal se encuentra minimizado, lo que quiere decir que únicamente se encuentran abiertos los puertos necesarios de los servicios utilizados, evitando así ataques de fuerza bruta y de hombre en la mitad.

El nodo principal también cuenta con el servicio de PROXY, para web filtering y control de acceso a sesiones VTY; es decir a las sesiones que se realizan de manera remota, utilizando por supuesto el protocolo SSH a través del puerto 2222.

Existe también un balanceo de carga entre ambos servidores ISP, lo cual si bien es cierto vendría por parte del proveedor; es importante acotar, pues este no es simplemente un esquema de redundancia, sino que también proporciona balanceo en caso de saturación del enlace principal.

4.5 NUEVOS PROTOCOLOS

Actualmente existen nuevos protocolos aplicables a los diferentes sistemas operativos, para proporcionar seguridad en las comunicaciones, tales como:

- SNMPv3
- RMON
- IPv6

En el esquema de seguridad, se ha implementado SNMPv2 para el monitoreo, el cual se encuentra embebido en el monitoreo interno entre agentes y servidor; adicionalmente, todos los parámetros de seguridad anteriormente descritos. Sin embargo cabe recalcar la utilidad de los nuevos protocolos citados anteriormente así como de su utilidad y su costo de implementación.

Mediante el uso de herramientas de monitoreo de ancho de banda, se puede constatar la utilización de recursos que supone el monitoreo continuo de Zabbix para con sus agentes.

```
root@FreeBSD:~ # vnstat -u -i em4
root@FreeBSD:~ # vnstat
Database updated: Sun Jun 26 16:33:08 2016

   em4 since 06/22/16

          rx:  2.83 MiB      tx:  30.78 MiB      total:  33.60 MiB

   monthly
                     rx      |     tx      |    total    |   avg. rate
     ------------------------+-------------+-------------+---------------
       Jun '16     2.83 MiB |   30.78 MiB |   33.60 MiB |    0.12 kbit/s
     ------------------------+-------------+-------------+---------------
     estimated        2 MiB |      35 MiB |      37 MiB |

   daily
                     rx      |     tx      |    total    |   avg. rate
     ------------------------+-------------+-------------+---------------
      06/22/16     1.98 MiB |   29.07 MiB |   31.05 MiB |    2.94 kbit/s
         today      863 KiB |    1.71 MiB |    2.56 MiB |    0.35 kbit/s
     ------------------------+-------------+-------------+---------------
     estimated        --     |      --     |      --     |
root@FreeBSD:~ #
```

Figura 4.44: Gráfica de consumo de la interfaz hacia el agente en un minuto con VNSTAT.

Como se puede apreciar en la figura 4.44, el agente consume menos del 3% de su ancho de banda hacia el servidor para transmitir la información censada, por lo que no se ve comprometido el rendimiento de la red hacia la capa de distribución.

Dado que la carencia principal del protocolo SNMP hasta su segunda versión, es la autenticación, se utilizará la versión 3 para solventar este inconveniente; puesto que incluye autentificación, control de acceso, privacidad, control remoto seguro, etc.

SNMPv3 utiliza el mecanismo de autenticación USM, lo que ratifica que el mensaje recibido por la entidad real del host y que no fue suplantado, modificado o repetido ya que tanto el servidor como el agente comparten una llave secreta pre configurada fuera del entorno SNMPv3 y que tampoco se almacena en la MIB, por lo que no es posible obtener la clave secreta de manera remota.

Los protocolos de autenticación utilizados en este punto, son extremadamente seguros, puesto que se aseguran de que los mensajes que transmiten los agentes, están dentro de los parámetros de tiempo establecidos por descarte, retardos o ataques con mensajes repetidos, también se sincroniza la comunicación entre emisor y receptor, de modo que los mensajes se encriptan, evitando su análisis por posibles intrusos utilizando algoritmos de encripción como el CBC, DES o DES-56.

Por otra parte el agente también está protegido ante ataques posibles intrusos, pues en la versión 3 del protocolo SNMP, incluye mecanismos de certificación para permitir el acceso del gestor hacia las MIB, ya que en este punto se encuentran configuradas las claves secretas y es aquí donde se detallan los privilegios de acceso para los diferentes gestores que puedan solicitar información de monitoreo.

El protocolo RMON define funciones de MIB más amplias que SNMP, como obtener información propia de la red, es decir, si en SNMP se monitorizaban dispositivos internos de la red en producción, en RMON se monitoriza toda la red en su conjunto. Funciona a base de sondas insertadas en toda la red, mismas que indican al gestor información de la red aun cuando este se encuentra desconectado de la misma; también en casos de fallo, las sondas almacenan la información suficiente del tráfico cursante en ese momento, para proporcionar información útil de diagnóstico y reparación. (Jimenez, 2014)

Tiene 3 versiones que describen sus funciones específicas:

- RMONv1: Estadísticas para una red LAN/WAN
- RMONv2: Estadísticas para el nivel de aplicación de la red.
- SMON: Versión de RMON para el análisis de redes conmutadas.

Finalmente se tiene la última versión del protocolo de direccionamiento IP, que es IPv6; el cual consta de 128 bits, una trama más simple y mayor espacio de direccionamiento, ya que para el caso de IPv4, se tiene 4300 millones de direcciones posibles, mientras que para IPv6, se tendrán 340 billones de direcciones posibles.

En cuestión de enrutamiento, se disminuye la latencia de la red, se proporciona mayor seguridad en cuanto la autenticación y confidencialidad y la flexibilidad de configuración al enrutar redes remotas.

Las solicitudes entre hosts, se realizan de manera multicast, al realizar la configuración DHCPv6 mediante SLAAC[21]; además cada dispositivo tendría una dirección global única IPv6, lo que haría posible localizarlo en todo momento.

[21] StateLess Address AutoConfiguration, es un protocolo de configuración dinámica en DHCP para IPv6.

CAPÍTULO 5: RIESGOS Y SEGURIDAD

5.1 RIESGOS DE SEGURIDAD

Según la teoría del hacking ético, solo se necesita conocer el nombre de una compañía o únicamente disponer de una dirección de correo, para poder empezar a realizar un estudio de vulnerabilidades.

Resulta muy útil aplicar los métodos de seguridad descritos en el capítulo anterior y también de implementar políticas de seguridad robustas en firewall; para el nodo central, se han considerado 2 módulos de firewall, que protegen los accesos internos y externos a los recursos de la red.

Resultan muy fáciles los ataques a la información, cuando existen huecos de seguridad, como por ejemplo puertos abiertos innecesariamente en el firewall, no establecer políticas de acceso por niveles. El no dar capacitación al personal acerca de la seguridad a los empleados de la empresa, también constituye un grave riesgo de seguridad.

Otro riesgo de seguridad, es el INTERNET; puesto que es una puerta abierta a cualquier tipo de ataques, ya que existen usuarios malintencionados que buscan dañar, sustituir o sustraer información de la compañía en cuestión y es trabajo del departamento de TI de la empresa, proporcionar soluciones técnicas a largo plazo para los diferentes ataques que puedan suscitarse.

El hecho de disponer de sistemas de claves sin estar sujeto a cambios, también representa una amenaza de seguridad ya que la misma clave podría ser descifrada.

No disponer de un recurso de monitoreo de servicios y disponibilidad puede ser un grave riesgo de seguridad, ya que fácilmente se pueden detectar los ataques de fuerza bruta con un simple monitoreo de la interfaz. Pues ante una actividad inusual, se puede dar de baja el puerto o simplemente agregarle una configuración de seguridad adicional, como en el caso de los switches CISCO CATALYST, que poseen seguridad de puertos por dirección MAC.

5.2 FALENCIAS EN PROTOCOLOS

Cada protocolo lleva en su versión, las características pensadas para el momento de su desarrollo; conforme avanzan las pruebas o implementaciones, van surgiendo necesidades que cubrir y es en las siguientes versiones, donde se aplican estos parches a la codificación, solventando dichas necesidades.

Entre los protocolos predecesores los actuales, se tienen:

- CMIP
- SNMPv1
- SNMPv2
- SSHv1
- TELNET
- HTTP

El protocolo CMIP, al igual que el protocolo SNMP, trabaja en la capa de aplicación y permite gestionar los dispositivos de la red mediante agentes y gestores.

El modelo de control de CMIP, es orientado a conexión, lo que quiere decir que el host debe enviar la información requerida por el protocolo; esto incluye información acerca de clases, instancias, estados, herencias, pasarelas, etc. Lo que demanda gran parte del ancho de banda, imposibilitando a otros servicios, como ftp, mail, web, etc.

A diferencia de CMIP, que incluye ciertos mecanismos de seguridad, la primera versión de SNMP descrita en 1988, no incluía ningún mecanismo de seguridad, los agentes transmitían la información a los gestores en texto plano; lo que quiere decir que la información transmitida, podía ser interceptada, modificada o repetida.
Los agentes únicamente comprueban que la petición de información corresponda a la misma comunidad de monitoreo, que una vez comprobado, entrega la información de las MIB, sin petición de ningún tipo de privilegios.

En la versión dos del protocolo SNMP, se mejoran ciertos aspectos del control de la red, como por ejemplo el envío de grandes cantidades de información a través de la red, como tablas de enrutamiento o las tablas de una base de datos, más convergencia en redes extensas y simplicidad de configuración. Sin embargo incluía métodos de seguridad deficientes, tales como una clave pública compartida en las MIB o sistemas de claves aleatorias o control total de una MIB, defectos que son completamente superados en la versión 3, considerada la versión actual desde el año 2004.

El protocolo TELNET, que es ampliamente utilizado para la gestión remota de servidores a nivel de CLI, sin embargo tiene un grave problema de seguridad, el cual se basa en el intercambio de información en texto plano; lo que facilita que cualquier atacante pueda observar las claves y usuarios sin ningún tipo de encripción ni autenticación.

Dados los problemas de TELNET, se crea el protocolo SSH, que en su primera versión utilizaba varios algoritmos de encripción patentados para cifrar la

información y ya no viaje en texto plano; no obstante con la expiración de algunas patentes y una deficiencia de seguridad detectada, se hizo necesario crear una segunda versión que solvente estos inconvenientes, en la cual se incluye el cifrado en todas las conexiones establecidas, con algoritmos de cifrado que impiden la lectura de la información intercambiada, de su modificación o repetición; además incluyen mecanismos de certificación de autoridad, que certifican que la petición de conectividad viene de un host autorizado.

El protocolo HTTP en sus inicios también presentaba deficiencias de seguridad, ya que no incluye seguridad, pues es vulnerable a los ataques como Man in the middle, que suplanta identidad y puede tener acceso a los datos, también el atacante puede estar observando la conexión TCP entre el ordenador y la página web, que suelen contener información personal delicada a la que podrían tener acceso.

HTTPS es la solución a esa falencia de seguridad, puesto que a través del puerto 443 soporta los ataques de fuerza bruta o de hombre en la mitad, lo que certifica al acceder a un sitio web con esas características, que es segura la información intercambiada en ese momento, protegiendo así los datos del usuario.

La navegación mediante HTTPS, también incluye certificados para los usuarios web de la empresa y control de su acceso al internet; se considera un entorno web más seguro a través de sockets que imposibilitan el espionaje de la conexión en ese momento además de evitar modificación de información enviada y recibida desde el ordenador.

5.3 TÉCNICAS DE SEGURIDAD

En las empresas actualmente es vital tener establecido un esquema de seguridad de varias direcciones; es decir, que la política de seguridad sea capaz de mitigar cualquier evento adverso a la actividad normal de la red.

Comenzando por el eslabón más débil de la cadena informática, se empieza lógicamente por los usuarios sin conocimientos de seguridad informática, instruyendo a los mismos acerca de claves complejas, cambios de las mismas cada cierto tiempo, cierre de sesiones y el intercambio de información delicada.

La separación del control de un sistema de seguridad a nivel modular; lo que implica separar los niveles de autoridad de acceso a nivel de los usuarios, que lo usuario tengan diferentes tipos de acceso y los usuarios y claves sean controlados en una base de datos protegida por contraseña.

Implementación de cifrado y encriptación en todas las conexiones entre hosts y servidores, de modo que no se pueda interceptar, modificar, repetir o suplantar los mensajes que se envían a través de la red.

La utilización de firewalls para las conexiones salientes y entrantes ante la red pública de internet, también se aumenta la seguridad mediante la implementación de listas de acceso para las diferentes áreas de la empresa.

Un servidor proxy también es sugerido para proteger la confidencialidad de la empresa, ya que se puede controlar el acceso de web de los usuarios evitando así comunicaciones externas no deseadas, además de equilibrar la utilización de los recursos al no permitir la saturación de los canales de red. La implementación de certificados para las conexiones hacia los clientes también es una solución factible, puesto que con ese método se certifica que la máquina remota es realmente quien es y no se trate de un atacante. La actualización del software presente en el cliente es crucial para mantener la disponibilidad de los servicios, se puede contar con antivirus, firewalls y sistemas operativos que deben estar actualizados continuamente.

El software de monitoreo propuesto en el proyecto, proporciona una excelente técnica de seguridad, como es el envío de correos electrónicos ante cualquier falla de la red programable; adicionalmente si se dispone de una conexión hacia la red 2G, también se puede configurar el envío de mensajes de texto a través de la red celular, a cualquier numero móvil programable.

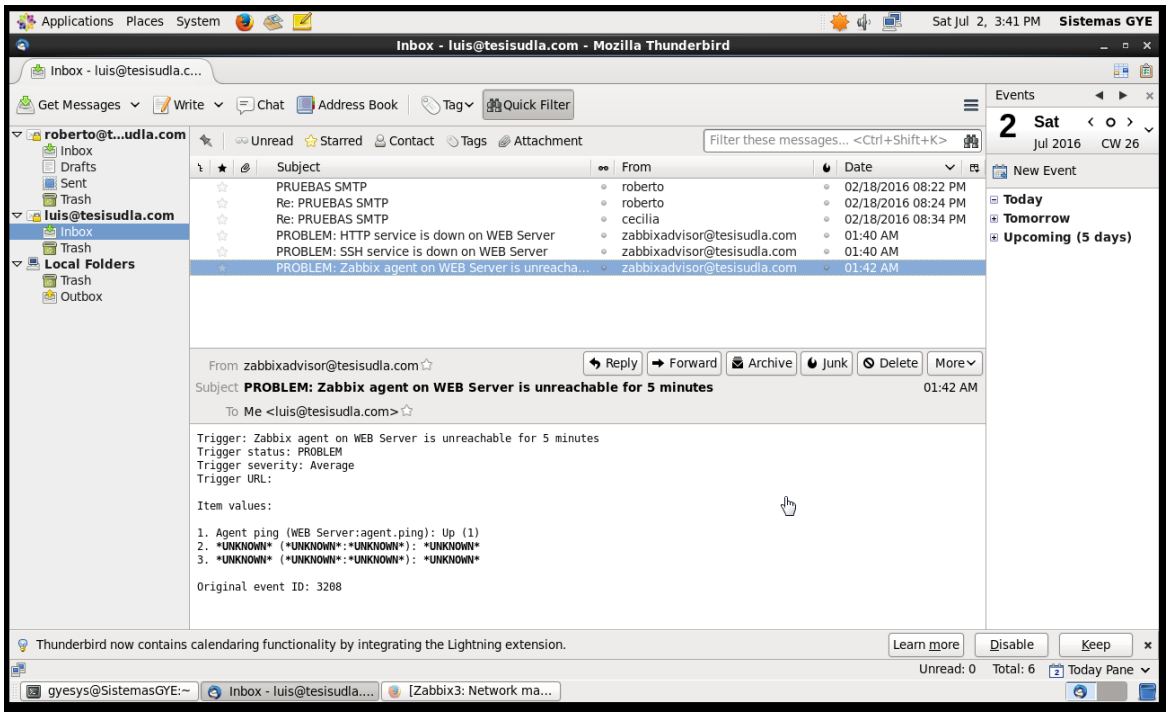

Figura 5.1: Figura de buzón de correos con alertas de Zabbix.

Para entender mejor la figura 5.1, se simuló un apagado del servidor WEB, mismo que se encuentra ubicado en la red DMZ; según el tiempo establecido, el servidor envía un correo electrónico a los destinatarios pre configurados, detallando el tipo

de alerta y el suceso. En este caso en el buzón del departamento de TI, se encuentran tres alertas que indican:

- Que el agente dejo de responder hace 5 minutos.
- Que el servicio SSH en el agente ha sido apagado.
- Que el servicio HTTP en el servidor WEB ha sido apagado.

Existen varias alertas que se pueden configurar, sin embargo, las anteriormente expuestas resultan un muy ilustrativo ejemplo de las alertas por correo electrónico de Zabbix.

5.4 INFORMES DE MONITOREO

La interpretación de los informes de monitoreo es clave, ya que le permite al administrador de la red, tomar decisiones acerca de las posibles fallas que se puedan producir en la topología, para dar soluciones rápidas y confiables que solventen las necesidades de la capa de acceso.

Estos informes nos proporcionan datos necesarios para realizar cambios futuros o inmediatos en la configuración actual o posibles criterios de diseño que abarquen la implementación de protocolos de ruteo o priorización de tráfico, así como de balanceo de carga y QoS. En el proyecto en cuestión, el servidor Zabbix, es capaz de proporcionar un esquema de disponibilidad de servicios para la red operativa.

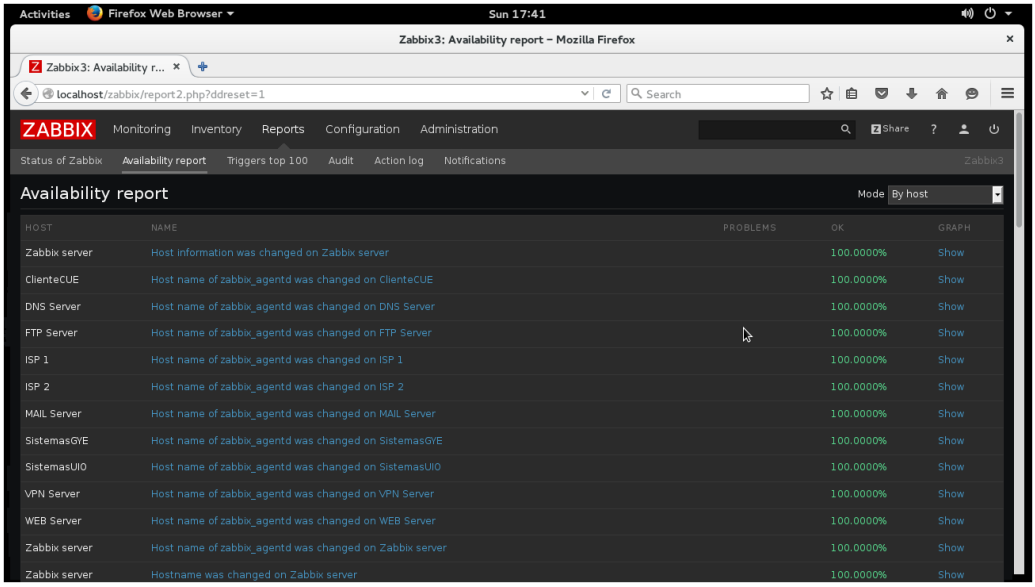

Figura 5.2: Informe de disponibilidad o SLA proporcionado por el servidor.

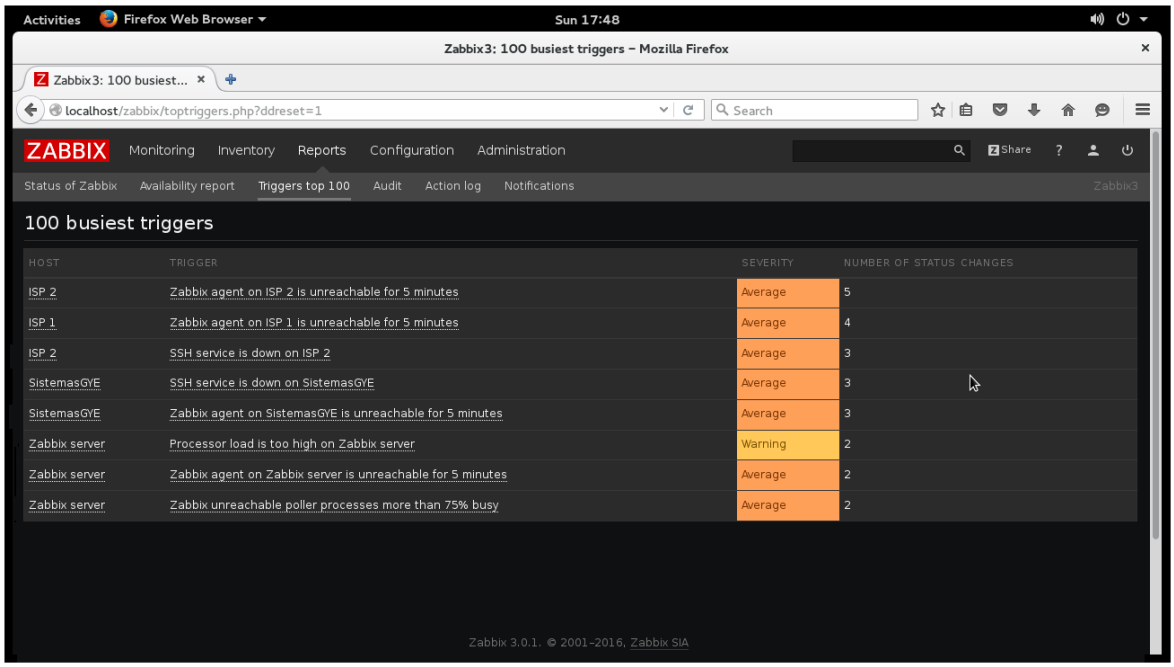

Figura 5.3: Informe de sucesos y alertas en la red proporcionado por el servidor.

La figura 5.2, representa la disponibilidad de los servicios configurados en la red desde la primera ejecución del sistema Zabbix. Estos parámetros son modificables, es decir pueden incluirse servicios adicionales, así como las diferentes representaciones de la disponibilidad.

Zabbix también maneja historiales de navegación y descargas para monitorear la actividad de la página web; por defecto, el historial del monitoreo de conexiones, dura tres meses; sin embargo, se puede configurar el servidor para ampliar o disminuir ese plazo.

SLA[22] es un nivel de servicio contratado actualmente por las empresas para mantener la disponibilidad de los servicios contratados, ya que en muchos casos el proveedor suele tener inconvenientes de conectividad y la caída del servicio cuenta como un descuento en la facturación mensual, si el servicio cae demasiado tiempo.

Es decir, que por un contrato de SLA de 96%, se asegura un tiempo máximo de caída de servicio, así como del tiempo para resolución de problemas físicos en la red.

Ciertas empresas, como las entidades bancarias, requieren niveles de servicio más altos, ya que sus servicios no pueden caer por más de dos horas, por lo que se contratan más puntos por encima de 96%, por ejemplo, 97.5%, 98% o 98.5%.

[22] Service Level Agreement.

Tomando en cuenta que cada punto más cercano al 100%, supone millones de dólares en contrato de nivel de servicio.

Por otro lado, la figura 5.2, representa las alarmas que se han tenido últimamente, de manera continua, para poder elaborar algún plan de acción de mitigación.

5.5 CONCLUSIONES Y RECOMENDACIONES

De los servicios de las redes en la actualidad, se puede concluir la importancia del control y administración de estos recursos; pues una buena gestión de los mismos conlleva a tener una red operativa sin inconvenientes no previsibles.

Con el sistema de monitoreo en cuestión, se puede llevar un control general y continuo de la red en producción, basando su funcionamiento en alertas visuales, alertas vía correo electrónico o SMS, utilizando la red celular desplegada.

Es muy útil contar con herramientas capaces de redundancia, control de acceso, QoS, el protocolo de encapsulación 802.1Q, enlaces Etherchannel y la ejecución de herramientas de diagnóstico para mantener un buen funcionamiento de la topología; sea cual fuere esta, el sistema es acoplable y escalable al diseño, tomando en cuenta que actualmente, herramientas de este tipo, suelen tener costos muy elevados.

Por otra parte, no se cuenta actualmente con personal capacitado para manejar estos sistemas, de modo que, las compañías PYME, se ven obligadas a gastar montos importantes de dinero por soluciones que se pueden proporcionar con software de uso libre, sin costos excesivos, sino únicamente de implementación y mantenimiento, para el correcto funcionamiento del sistema y la red administrada.

Una clara muestra de lo anteriormente expuesto, seria en base a la experiencia laboral; donde el cliente como tal, solicita redundancia para su red interna, ya que por cuestiones del negocio mismo, no se puede permitir la perdida de conexión hacia su granja de servicios o DMZ.

Resulta muy costosa una implementación de redundancia lógica para el cliente, pues al disponer de equipos CISCO para la interconectividad, se requieren licencias para la utilización de ciertas características del router. En este caso, tanto el cliente como el ofertante incurren en gastos de dicha implementación.

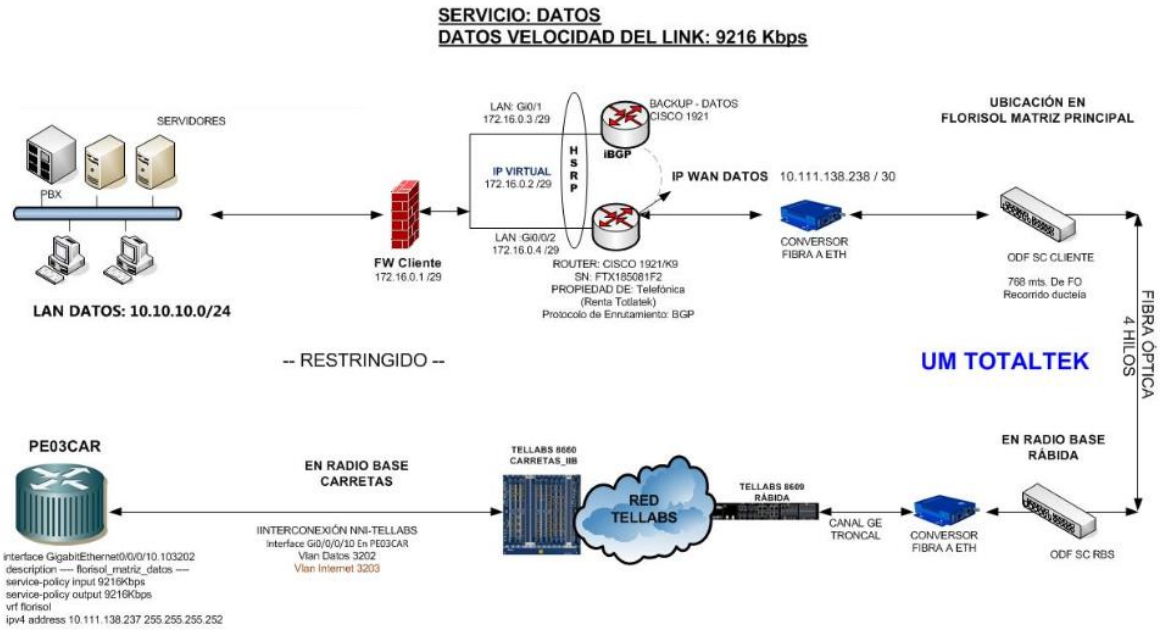

Figura 5.4: Topología de redundancia lógica ofrecida por Movistar para el cliente.

Como se puede apreciar en la figura 5.4, HSRP[23] se encuentra implementado en la solución ofrecida al cliente, sin embargo el costo de esta solución es relativamente elevado a comparación de las soluciones GNU.

Los router CISCO tienen un sistema operativo similar a UNIX, donde se ejecutan ciertos comandos para configurar las interfaces de red y se implementan protocolos de ruteo para alcanzar redes distantes. Sin embargo ciertos protocolos, como BGP[24], HSRP, OSPFv3[25], EIGRP[26], etc. Son de uso licenciado, es decir, que se debe realizar un pago económico para admitir el uso de este protocolo en ese dispositivo.

El nodo central del proyecto en cuestión, es capaz de realizar la misma característica que HSRP, por lo que únicamente es necesario contar con dos equipos de medio rendimiento, para ofrecer la misma solución de modo gratuito. No obstante, para las redes empresariales del mundo contemporáneo, se requieren medidas de seguridad más robustas; como es el caso de la nueva generación de seguridad con redes MPLS[27].

MPLS, es la nueva generación de redes inteligentes, cuyas características de rapidez y adaptación, la hacen ideal para cualquier entorno de redes WAN.

[23] Hot Standby Router Protocol, característica de los routers CISCO para ofrecer redundancia lógica.
[24] Border Gateway Protocol, es un protocolo de ruteo entre equipos CISCO.
[25] Open Shortest Path First, en su version 3, es un protocol de ruteo de equipos CISCO para IPv6.
[26] Enhanced Internal Gateway Routing Protocol, es un protocolo de ruteo entre equipos CISCO.
[27] MultiProtocol Label Switching, es un protocolo de capa 2, que encapsula las tramas de red.

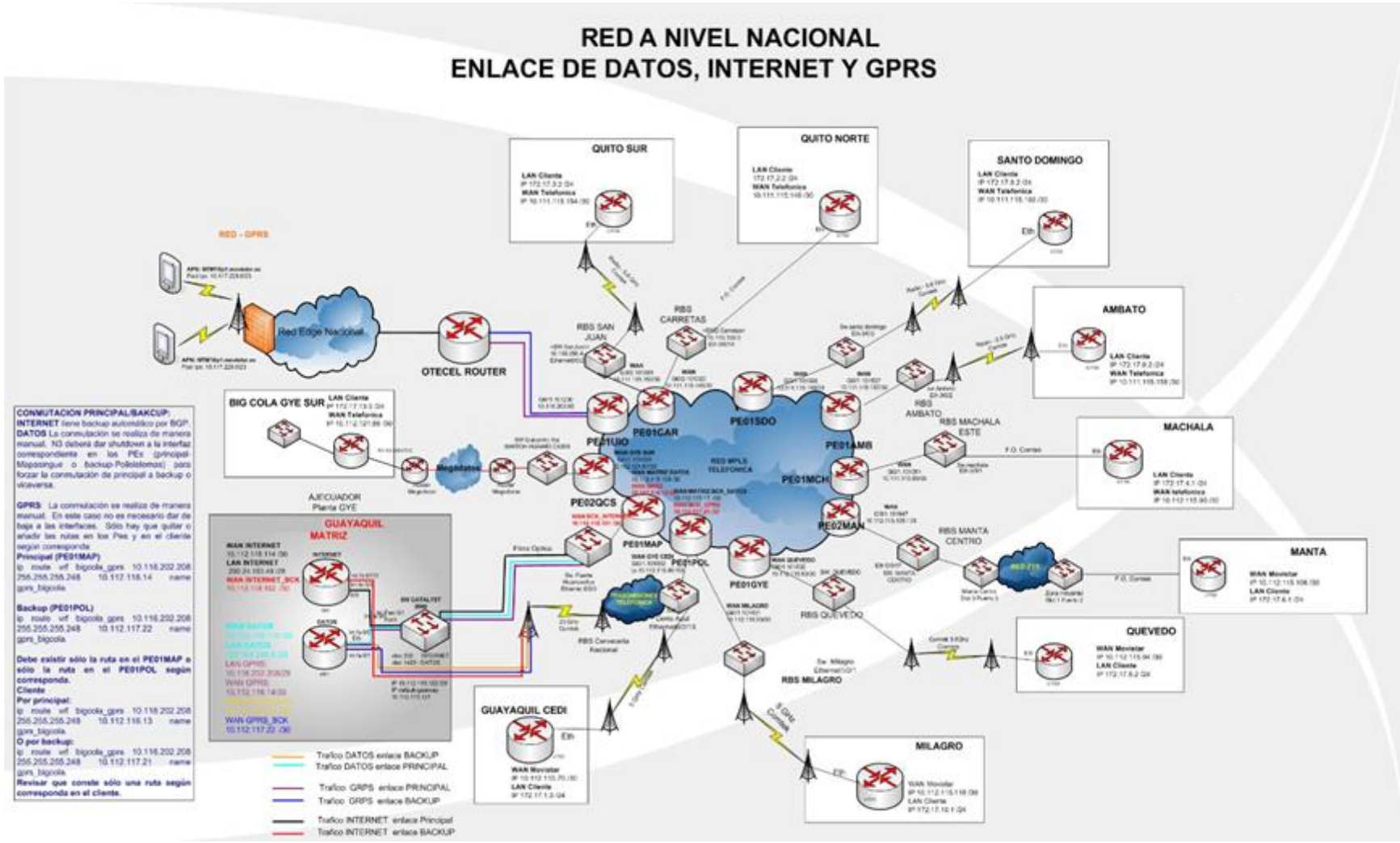

Figura 5.5: Esquema de topología MPLS a nivel empresarial.

Como se aprecia en la última figura del presente texto, se trata de una red empresarial a nivel nacional. Cuya infraestructura, abarca varias conexiones alrededor del territorio nacional; incluyendo tecnologías WAN como radiofrecuencia, fibra óptica, cable coaxial, etc. MPLS es una red versátil que se adapta a cualquier tecnología de transmisión.

Al solo incluir la revisión de etiquetas de las tramas, resulta más rápido el envío de tramas, sin tener que analizar la capa 3 del mensaje y tener que decidir su destino a través de un camino especifico. La característica principal de MPLS, reduce drásticamente la latencia de red, optimizando el rendimiento del equipo al realizar la decisión de envío tan solo una vez. (S21sec, 2008)

En cuanto a la seguridad, MPLS se acopla perfectamente con las VLAN, encapsulando las tramas etiquetadas y enviándolas a través de la red, una vez allí, la trama lleva varias cabeceras de diferentes clientes, lo que hace imposible reconstruir la información exacta y en orden con un analizador de protocolos. (Bili´c, 2016)

Entre las recomendaciones, se puede citar nuevamente lo indicado en el capítulo 5, puesto que en él, se detallan las técnicas de seguridad que se deben aplicar para cualquier sistema aplicado a una red operativa.

Una recomendación de un contexto un poco lejano, sería fomentar en el país la capacitación de estos sistemas, ya que con ella, se ahorrarían cantidades importantes de dinero al tener mano de obra nacional, que realice dichos esquemas de red, sin tener que recurrir a soluciones costosas proporcionadas por países extranjeros.

Finalmente se puede decir que, *FreeBSD* es el sistema madre de todas sus derivaciones actuales, pues la mayoría de sistemas desarrollados a través de la historia, parten del código fuente desarrollado en la universidad de *Berkeley*.

BIBLIOGRAFÍA

Atelin, P. &. (2006). *Redes informaticas: conceptos fundamentales: normas, arquitectura, modelo OSI, TCP/IP, Ethernet, WIFI.* Barcelona-España: ENI.

Bach, M. J. (1986). *The design of the UNIX operating system* (Vol. 5). Englewood Cliffs.

Bili´c, D. G. (26 de Octubre de 2016). *Welivesecurity*. Obtenido de http://www.welivesecurity.com/la-es/2015/10/26/ataques-proveedores-redes-mpls/

Ecured. (7 de Junio de 2015). *Historia y Desarollo de los sistemas BSD.* Obtenido de http://www.ecured.cu/FreeBSD

Foundation, F. (8 de Junio de 2015). *Reseñas historica del sistema.* Obtenido de https://www.freebsd.org/doc/es/books/handbook/history.html.

FreeBSD. (22 de Abril de 1995). *Historia de los sistemas UNIX & BSD.* Obtenido de https://www.frebsd.org/doc/es/articles/explaining-bsd/article.html

Jimenez, J. (26 de Junio de 2014). Obtenido de Monitoreo de Redes: https://oposcaib.wikispaces.com/file/view/29+-+SNMP,+CMIS-CMIP,+RMON.pdf

Moritsugu, S. (2000). *Practical Unix.* Que Publishing.

Moya, J. M. (2006). *Redes y servicios de telecomunicaciones.* Paraninfo.

NAGIOS. (2009). *Nagios Core*. Obtenido de https://www.nagios.org/about/history/

Palmer, M. &. (2000). *Redes informaticas: guia practica.*

Pink, A. (2016). *Amerika Pink*. Obtenido de http://america.pink/cacti-software_819819.html

S21sec. (3 de Noviembre de 2008). *Your Cybersecurity Company*. Obtenido de http://blog.s21sec.com/2008/11/seguridad-mpls-i_03.html

Tanenbaum, A. S. (2003). *Sistemas operativos modernos.* Pearson Education.

Vladishev, A. (2016). *Open Source*. Obtenido de Your Mind: http://opensourceday.com/osd-2016/speakers/alexei-vladishev-zabbix