UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
UCI

Integración de Firewall y Proxy para instituciones de salud cubanas

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autoras: Dalay Fernández Rodríguez
Lidibet Minguez Boada

Tutor: Ing. Juan Carlos Pujol García

Ciudad de La Habana, Cuba
Julio, 2007

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de julio del año 2007.

Dalay Fernández Rodríguez

_____

Firma del Autor

Lidibet Minguez Boada

_____

Firma del Autor

Tutor: Ing. Juan Carlos Pujol García

_____

Firma del Tutor

Datos de contacto

Ing. Juan Carlos Pujol García.

Ingeniero, ISPJAE-1986, Especialista superior en informática de Softel. Profesor auxiliar, Carretera a San Antonio Km. 2 1/2 Reparto Torrens, Infraestructura productiva de la UCI, Ciudad Habana. Teléfono (53-7) 835-8258, email: juanca@softel.cu , labora actualmente como jefe del proyecto de Servicios remotos de la empresa Softel, su grupo que se dedica a la administración remota de servidores y aplicaciones informáticas, tanto basadas en plataforma libre como propietaria. Imparte la asignatura Sistemas de bases de datos en la UCI. Posee 21 años de experiencia en varios roles de la informática, como análisis, diseño, implementación, pruebas, despliegue, dirección de proyectos, así como en transmisión de datos, diseño y administración redes (tipo LAN y MAN), administración de servidores de varios tipos y tecnologías. Diseño de sistemas para procesamiento de señales bioeléctricas.

Agradecimientos

Muchas han sido las personas que de una forma u otra nos han ayudado a la realización de esta tesis: No podemos dejar de agradecerle infinitamente a nuestro tutor Juanca, por su dedicación, perseverancia y esmero durante todos estos meses de trabajo. A Jorge Gil Silva, administrador de los servidores y la red de la UCI, a Orestes Rodríguez Morales, jefe de conectividad de las redes y a Eduard Palomo Gené, administrador de red y responsable del nodo central de la UCI; quienes nos brindaron su ayuda desinteresada y significativa en la elaboración de este trabajo de diploma. A Adrián Peña y a Dayrel Almaguer, quienes desde el comienzo estuvieron a nuestro lado con muchísima paciencia ayudándonos en todo momento….mil gracias. A Abel Llerena por trabajar junto a nosotras en la implantación de este diseño. Y a Joel, que desde el GET ayudó a que este trabajo saliera lo mejor posible. Muchas gracias a todos por ayudar a que hoy seamos ingenieras.

de Dalay

A mis papitos lindos, por ayudarme en todo momento y ser mi mejor guía y ejemplo…por ser los mejores. A mis abuelitos que tanto quiero, por apoyarme siempre y poder darles la satisfacción de verme graduada. A mi tío Nelson, por quererme tanto y por poderle dar hoy un motivo más para estar orgulloso de mí. A mi tía Arlén, por darme tranquilidad, seguridad y su valiosa energía. A mi Roly, por su amor y comprensión, por quererme y apoyarme durante todo este tiempo. A Liensi y a mi negra Yaima, por haber sido más que amigas estos 5 años. A Sulia, Marta Isabel y Yelaine, mis amigas de siempre, gracias por su confianza y amistad. A Yirki y Papitico, por haber sido tan buenos y lindos conmigo, no los olvidaré. A Tita, tío Manoli y a mis hermanos Alberto y Adrián, por preocuparse y estar siempre pendiente de mi. Gracias a todos por quererme y estar a mi lado…

de Lidibet

Quiero agradecerle a mi mami y papi por confiar siempre en mí y por siempre estar a mi lado. A mi familia linda que siempre está pendiente de todo lo que me sucede. A mi Juank precioso que con tanta paciencia estuvo a mi lado durante estos cinco años. A Yanara por ser mi amiga y estar conmigo en los malos y buenos momentos aquí en la escuela. A Yirki y Yoander (mi pionono) por ser unos compañeros y amigos inigualables y a todas esas personas que de una forma u otra me ayudaron y me apoyaron durante el transcurso de mi carrera.

Dedicatoria

Dalay

A mi mamita y papito.

A mis abuelitos.

A mi bella familia que siempre confió en que llegaría a ser lo que hoy soy.

Lidibet

Yo le dedico mi tesis primeramente a mi mamá, a mi papá, a mis abuelitos queridos, a mis dos hermanitas y a toda mi familia que siempre estuvo apoyándome, dándome fuerzas y confiando siempre en mí.

Resumen

Con el creciente desarrollo de las tecnologías, las instituciones de salud cubanas han tomado en consideración el uso de las mismas para garantizar y controlar su buen funcionamiento. Estas instituciones se encuentran actualmente invirtiendo en un proceso de desarrollo informático que viene dado por el crecimiento del equipamiento instalado, las aplicaciones que utilizan, los servidores y la ínter conectividad de las redes. Debido a esto el presente trabajo está enmarcado en la realización de un diseño para aumentar la seguridad de las aplicaciones informáticas y agregar algunas funcionalidades a la conectividad de las LANs de las instituciones de salud que existen.

Para llevar a cabo esta solución se ha realizado un previo estudio de todo el marco teórico relacionado con este tema, como son los softwares utilizados para que funcionen como firewall y proxy y los diseños generales de seguridad perimetral que nos podemos encontrar. Se abordó lo referente a la problemática actual de las instituciones de salud cubanas, se realizaron recomendaciones para adaptar este diseño a cualquier institución así como la solución general. También se propusieron soluciones para cualquier escenario y el diseño de seguridad perimetral para dos instituciones de salud específicas como son el banco de sangre de 23 y 2 y el hospital Hermanos Ameijeiras, para esto se tuvieron en cuenta los resultados de una serie de pruebas realizadas en el laboratorio donde se probaron diferentes facilidades tanto de firewall como de proxy, que ayudaron a proponer la solución general de las dos instituciones antes mencionadas.

Palabras claves: firewall, proxy, redes, seguridad perimetral.

Índice

Introducción

Cuba cuenta con un sistema de atención a la salud considerado único en Latinoamérica, con cobertura total del país. Sin embargo muchas de las instituciones de salud cubanas, llámense éstas: bancos de sangre, policlínicos u hospitales, no cuentan con el desarrollo tecnológico y el recurso informático necesario para lograr una buena comunicación, un intercambio de información seguro y una navegación confiable en redes externas.

La incorporación de una computadora a la red y aún más a otra red externa hace necesario implementar un sistema de seguridad perimetral más eficiente, rápido y de mayor acceso a la información para con otras redes, en este caso específico nos referimos a la instalación de un firewall capaz de garantizar la no vulnerabilidad del sistema, realizando un filtrado de todos los paquetes enviados o recibidos de una red a otra, decidiendo que hacer con ellos, si se aceptan ya que no son un peligro, o si se deniegan por poder tentar contra la seguridad del sistema; y un proxy capaz de manejar las comunicaciones, registrando todas las operaciones realizadas, denegar servicios a sitios no permitidos, pedir identificación a la hora de conectarse con una red externa y sobre todo poder ofrecerle a sus usuarios la web que ellos necesitan lo más rápido posible. La integración de toda esta solución proporciona una efectiva protección de la información del usuario.

Situación Problémica

Actualmente las instituciones de salud cubanas se encuentran en un proceso de crecimiento informático, que incluye el incremento de su tamaño y su grado de ínter conectividad, por lo que el sistema de seguridad con que cuentan no es lo suficientemente seguro para garantizar su protección. Las aplicaciones informáticas no tienen un elevado nivel de seguridad y las funcionalidades de la conectividad no son las suficientes para poder conectarse a la red de salud pública.

Problema Científico

¿Cómo erradicar los problemas de la conexión de la red en las instituciones de salud cubanas mediante la instalación de un firewall y un proxy?

Objeto de estudio

Conectividad de las redes locales de las instituciones de salud cubanas a la red de salud pública.

Campo de acción

Integración de un firewall y un proxy para la seguridad perimetral de las instituciones de salud cubanas.

Idea a defender

La implantación de la solución general que incluye la integración de firewall más proxy en un equipo perimetral, garantizará la seguridad y confiabilidad de la red perimetral de las instituciones de salud cubanas.

Objetivo general

- Diseñar e implantar un sistema de seguridad o protección perimetral y de acceso (firewall más proxy) para las redes de área local de las instituciones de salud cubanas.

Objetivos específicos

Firewall

- Estudio y dominio de la tecnología firewall.

  - Estudio de nuevas alternativas de las tecnologías firewalls, diferenciarlos y ver particularidades de cada uno.
  - Aprender a hacer scripts de iptables.
  - Estudio del montaje del firewall.

- Selección del software de firewall específico.
  - Buscar softwares alternativos al iptables. Comparar, decidir y argumentar.
- Firewall robusto frente a tecnologías maliciosas.
  - Definir listado de tipos de ataques.
  - Generar scripts con técnicas de firewall avanzadas (para iptables) para combatir la mayoría de tipos de ataques que existen a una red. Buscar y adaptar.
- Pruebas a los diseños de firewall.
  - Hacer diseño de escenarios de pruebas después de haber diseñado el firewall, para probar y verificar que cumple con sus funcionalidades: permite todo lo que se quiere y deniega todo lo que no.
- Tipificación de diseños de firewall vs. escenarios típicos.
  - Describir más ampliamente los escenarios más típicos encontrados en las instituciones de salud.
  - Diseñar prototipos de scripts de firewalls para cada caso. (Patrones de diseño).
- Regularización o micro-metodología de diseño de firewall.
  - Levantamiento de requisitos y descripción de escenario.
  - Diseño de la solución. (Describir como hacerlo).
  - Adaptaciones a los scripts de firewall.
  - Confeccionar diferentes scripts típicos de firewall para escenarios tipificados, discutir las mejores alternativas analizándolas para finalmente realizar las pruebas.

Proxy

- Estudio y dominio de la tecnología proxy.

- Estudio del Squid.
- Estudio del monitoreo del proxy.
- Generar trazas de navegación en squid y poner a funcionar los logs (las trazas).

- Selección del software de proxy específico.

  - Buscar softwares alternativos al Squid. Comparar, decidir y argumentar.

- Configurar un servidor proxy con las funcionalidades básicas.

  - Lograr autenticación contra diferentes recursos: listas privadas, dominio Windows.
  - Revisión de trazas y logs de navegación por software amigable.
  - Manejo de navegación nacional e internacional.
  - Manejo de listas negras de sitios prohibidos por al menos un método.

- Configurar un servidor proxy con funcionalidades avanzadas.

  - Múltiples instancias del squid.
  - Proxy en cascada.

- Diseño de la solución proxy.

  - Alternativas de módulos frente a requisitos del lugar.
  - Análisis e incremento de la seguridad del proxy.

- Diseño de soluciones típicas con proxy.

  Firewall y Proxy

- Sostenibilidad de un diseño.

  - Definir listado de posibles situaciones adversas o siniestros.
  - Definir listado de posibles soluciones de emergencia y alternativas óptimas: obtener documentación técnica y de usuario.

- Integración del firewall con el proxy.

- Pruebas de las soluciones firewall y proxy.

  - Hacer pruebas piloto (adelantarse a los problemas que puedan ocurrir) de cada una de las variantes y soluciones.

Métodos Científicos de Investigación

- Métodos Teóricos.

  - Analítico - Sintético
  - Inductivo - Deductivo
  - Análisis Histórico Lógico

- Métodos Empíricos.

  - Observación

Capítulo 1 Fundamentación Teórica.

Capítulo 1

1.1 Introducción del capítulo.

En este capítulo se introducen conceptos básicos para la comprensión del tema a abordar. Se mencionan las tendencias y las tecnologías actuales más utilizadas que se tomaron en consideración. Se realiza un análisis de los tipos de ataques informáticos a los cuales están expuestas las redes. Por último se hace un estudio del estado del arte sobre algunos trabajos que abordan el tema.

1.2 Conceptos generales.

1.2.1 Firewall (Cortafuego).

El firewall es un dispositivo de hardware o software capaz de efectuar un filtrado sobre las conexiones que entran y salen de un determinado segmento de red. El firewall se coloca “en medio” de las comunicaciones entre una red local e Internet, es decir, separa una red protegida de una red sin protección, filtrando todo el tráfico que lo atraviesa y tomando decisiones de que hacer con él en función de reglas establecidas. Un firewall es parte de una política de seguridad completa que crea un perímetro de defensa diseñada para proteger las fuentes de información. **[1]**

Ventajas

Si no existiera el firewall todas las computadoras de la red estarían expuestas a ataques desde el exterior. Al manejar el acceso entre dos redes, la seguridad de toda la red depende de que tan fácil fuera violar la seguridad local de cada PC interna.

El firewall es el punto ideal para monitorear la seguridad de la red y generar alarmas de intentos de ataque, el administrador será el responsable de la revisión de estos monitoreos.

Desventajas

La limitación más grande que tiene un firewall sencillamente es el hueco que no se tapa y que coincidentemente o no, es descubierto por un intruso. Los firewalls no son sistemas inteligentes, ellos actúan de acuerdo a parámetros introducidos por su diseñador, por ende, si un paquete de información no se encuentra dentro de estos parámetros como una amenaza de peligro, simplemente lo deja pasar.

Otra limitación es que el firewall "no es contra humanos", es decir que si un intruso logra entrar a la organización y descubrir contraseñas o los huecos del firewall y difunde esta información, el firewall no se dará cuenta. **[2]**

1.2.2 Proxy (Servidor Intermediario).

Un proxy es un servidor intermediario para algún servicio de red, que consiste en permitir a los clientes realizar conexiones de red indirectas hacia otros servicios. Cuando un cliente navega a través de un proxy, cada petición que se hace al navegador se delega al propio servidor proxy y este es el que se encarga de acceder a algún recurso que se ha solicitado y se lo pasa a nuestro navegador.

El proxy abre una conexión con el servidor y le entrega los resultados al cliente. Si el protocolo involucra más de una consulta y su resultado, el proxy continúa interactuando entre el cliente y servidor. **[3]**

Al estar en medio de este tráfico puede realizar dos funciones muy importantes: controlar los accesos (permite o deniega según se disponga en sus normas) y hacer caché de peticiones de elementos (páginas web, imágenes). Íconos que una vez fueron pedidos se guardan en un espacio de disco de longitud variable al que se denomina caché. **[4]**

Los proxy en general, hacen posible que se logren varias cosas para garantizar el buen funcionamiento de una LAN manteniendo la seguridad.

Ventajas

- **Control:** Como actúa de intermediario es el que hace el trabajo real, por tanto se pueden limitar y restringir los derechos de los usuarios y dar permisos sólo al proxy.
- **Ahorro:** Sólo uno de los usuarios (el proxy) es el equipado para hacer el trabajo real.
- **Velocidad:** Si varios clientes piden el mismo recurso, el proxy puede hacer caché: guardar la respuesta de una petición para darla directamente cuando otro usuario la pida, así no tiene que volver a contactar con el destino, acaba más rápido y ahorra ancho de banda.
- **Filtrado:** El proxy puede negarse a responder algunas peticiones si detecta que están prohibidas.
- **Modificación:** Como intermediario que es, un proxy puede cambiar la información o modificarla siguiendo un algoritmo.

Cuando hablamos de proxy no solo nos referimos a cosas buenas ya que estos como intermediarios pueden provocar algunas contradicciones.

Desventajas

- **Abuso:** Al estar dispuesto a recibir peticiones de muchos usuarios y responderlas, es posible que haga algún trabajo que no le corresponda. Por tanto, ha de controlar quién tiene acceso y quién no a sus servicios, cosa que normalmente es muy difícil.
- **Carga:** Un proxy hace el trabajo de muchos usuarios.
- **Intromisión:** Es un paso más entre origen y destino, y algunos usuarios pueden no querer pasar por el proxy. Y menos si hace de caché y guarda copias de los datos.
- **Incoherencia:** Si hace de caché, es posible que se equivoque y dé una respuesta antigua cuando hay una más reciente en el recurso de destino.
- **Irregularidad:** Como representa a más de un usuario da problemas en muchos escenarios, en concreto los que presuponen una comunicación directa entre 1 emisor y 1 receptor. **[5]**

1.3 Lista de softwares similares de firewall y proxy.

En la actualidad existe una infinidad de tecnologías firewall y proxy, cada una con sus características propias, ventajas y desventajas. Se hará un análisis de todos estos softwares para más adelante decidir cuál es el más recomendable a utilizar de acuerdo a los requisitos y las necesidades de las instituciones de salud en Cuba.

1.3.1 Tipos de Firewalls que existen.

- ***Según lo que defienden:***

  - **Firewalls Perimetrales**

Los firewall perimetrales tradicionales protegen, como su propio nombre indica, el perímetro de la LAN corporativa: filtran e inspeccionan el tráfico que atraviesa la frontera entre la red interna e Internet. Sin embargo, los firewalls no están diseñados para proteger las conexiones en la red interna, a menos que se creen sub-redes virtuales. Los firewall de redes basados en host protegen redes completas y se instalan en sistemas operativos como Windows y Linux. Ejemplos de ellos son: Iptables para Linux y FW-1 Checkpoint para Windows. **[6]**

  - **Firewall Personales o basados en host**

Estos tipos de firewall son los que protegen un servidor o una estación de trabajo específica. Ejemplos de estos son: Agnitum Outpost Firewall, McAfee Firewall y Tiny Personal Firewall. **[7]**

- ***Por su funcionamiento:***

  - **Filtrado de Paquetes**

Este es el tipo más básico y sencillo de firewall, se basa en permitir o denegar el tráfico basado en los encabezados de segmentos y paquetes. Normalmente, se reducen a la

posibilidad de eliminar las conexiones a través de determinados puertos y a determinadas IP. Pueden dar una protección básica para eliminar una buena cantidad de ataques.

Ventajas

- Viene implementado prácticamente en la totalidad de los routers. **[8]**

Desventajas

- No ofrece autenticación, vulnerable a ataques como Spoofing (cambio de dirección del remitente). Difícil de administrar en ambientes complejos.

Statefull Firewall

Firewall con control de estado de la conexión. Un stateful firewall es un firewall que provee de herramientas como el seguimiento y control del flujo de una sesión de datos dentro y fuera de la red. **[9]**

El Stateful Firewall mantiene un registro de las conexiones, las sesiones y su contexto. Este módulo tiene su asiento entre la capa de Enlace y de Red. Esto significa que una vez establecida una conexión válida se puede enviar cualquier tipo de tráfico y el firewall no se dará cuenta. **[10]**

Ventajas

- Alto rendimiento. Opera solo a nivel de sesión y no de aplicación no tiene que inspeccionar todo el paquete de datos y por lo tanto con poco hardware maneja un buen ancho de banda.
- En lugar de abrir puertos de red permanentemente para ciertos protocolos que solicitan puertos arbitrarios, este tipo de firewalls solo abrirán estos puertos durante el tiempo suficiente para que el paquete pase.
- Permite definir un límite de tasa de conexión para defenderse de ataques DoS, tal como la inundación de paquetes SYN.

Desventajas

- No provee autenticación por usuario ni revisa toda la información del paquete, sólo los encabezados.

Stateless Firewall

Firewall sin control de estado de la conexión. Las primeras implementaciones de cortafuegos son considerados stateless, ya que al contrario de los stateful, no llevan un seguimiento de las sesiones, por lo cual cada paquete se analiza con las reglas definidas. Generalmente son más rápidos y consumen menos recursos, sin embargo, con el poder de cómputo actual, esto ya no es una ventaja significativa teniendo en cuenta la dificulta de configuración. **[9]**

- **Firewall de Aplicación**

Categoría que incluye a los proxy. Es capaz de inspeccionar hasta el nivel de aplicación. No solo la validez de la conexión sino todo el contenido de la trama. Es considerado como el más seguro. Todas las conexiones van a través del firewall. Un firewall de aplicación se distingue por el uso de los proxy para servicios como FTP, Telnet, etc. que previene el acceso directo a servicios al interior de la red.

Ventajas

- No permite conexiones directas (mantiene en secreto la identidad de los interlocutores), soporta autenticación a nivel de usuario y analiza los comandos de la aplicación dentro de la carga del paquete.

Desventajas

- Son más lentos (tienen que analizar todo) por lo tanto se requiere más cantidad de hardware para analizar el tráfico del canal. Pueden no soportar ciertos tipos de conexión. **[10]**

- **Firewall de Ficheros**

Firewall de fichero comúnmente referido a antivirus con protección en tiempo real del sistema de fichero.

- ***Por su arquitectura:***

  - **Firewalls por hardware**

Es un equipo cerrado corriendo un sistema operativo propio. Suelen ser bastante efectivos, especialmente los de marcas reconocidas, como es el caso de Cisco con PIX. **[8]**

Ventajas

- Durabilidad. Resistencia mayor a roturas.
- Facilidad relativa de instalación (en modo simple).
- Bueno para redes complejas y grandes.
- Confiabilidad.
- Seguridad. Son dispositivos que existen esencialmente para proveer seguridad.

Desventajas

- Muy alto precio.
- Reponibilidad.
- No facilidad para montar características adicionales.
- No estabilidad de suministros homogéneos en el tiempo.
- Software de gestión más monitoreo centralizado, prohibitivamente caros.
- Protocolos para VPN y autentificación fijos y establecidos por el fabricante. Posibilidad de incompatibilidad con otros suministros y diseños.
- Conocimientos técnicos especializados necesarios para diseñar soluciones y administrarlos. **[6]**

  - **Firewalls por software**

El firewall por software corriendo en una PC, se trata de un software específico o incorporado al sistema en un ordenador dedicado. El caso más conocido es el uso de

iptables de Linux. También hay productos completos basados en Linux e incluso distribuciones "live", es decir, con arranque desde CD y sin necesidad de disco duro. Otra opción son los productos comerciales, como es el caso de ISA Server de Microsoft. **[8]**

Ventajas

- Flexibilidad en los diseños.
- Reponibilidad. Cualquier PC con varias NICs en emergencias.
- Posibilidad de gestión de ancho de banda flexible.
- VPN flexibles según software adicional que se instale.
- Facilidad de montar otras facilidades adicionales.
- Posibilidad de lograr homogeneidad en todas las instalaciones en que se instale.
- Se desarrolla una tecnología propia, basada en software libre.
- Confiabilidad.
- Seguridad. Son dispositivos que existen esencialmente para proveer seguridad.

Desventajas

- Durabilidad. Posibilidad de fallas de hardware (partes mecánicas móviles) si hay mala calidad en el hardware adquirido.
- Necesidad de conocimientos elementales de Linux - Informática para instalar.
- Posiblemente sensible a fallas en el suministro eléctrico, si no hay UPS más grupos electrógenos.
- Conocimientos técnicos especializados necesarios para diseñar soluciones y administrarlos. **[6]**

1.3.2 Tipos de Proxy que existen.

Los proxy pueden trabajar en dos capas del modelo OSI:

Capa de Sesión (capa 5): Tiene la capacidad de asegurar que, dada una sesión establecida entre dos PC, la misma se pueda efectuar para las operaciones definidas de principio a fin, reanudándolas en caso de interrupción. En muchos casos, los servicios de la capa de sesión son parcialmente, o incluso, totalmente prescindibles.

Capa de Aplicación (capa 7): Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos. Hay tantos protocolos como aplicaciones distintas y puesto que continuamente se desarrollan nuevas aplicaciones, el número de protocolos crece sin parar. **[11]**

- **Proxy a nivel de Sesión**

La funcionalidad total de un servidor proxy no radica únicamente en el caché y en la mejora de la calidad de navegación. Hay un factor muy importante que hay que tener en cuenta a la hora de decidir sobre la instalación de un servidor proxy: la seguridad.

Para esto existen implementaciones como el protocolo Socks, que en su sentido más general, puede ser utilizado de la misma forma y para los mismos fines que un servidor proxy. Es por eso que se habla de la existencia de una variedad de paquetes de software de proxy para Linux y que estos pueden ser a nivel de aplicación (como Squid) y otros a nivel de sesión (como Socks).

- **Proxy Socks**

Proporciona también la funcionalidad de un firewall, es decir, brinda una mayor seguridad para los usuarios internos de nuestra red, ocultando las verdaderas direcciones IP que formulan las solicitudes, y filtrando los paquetes que salen, pero sobre todo los que entran, para ser redirigidos hacia la PC local que originalmente hizo la petición.

Actualmente, las versiones de Socks casi exclusivamente utilizadas son la 4 y la 5. Socks5 proporciona mayor seguridad, ya que añade un mecanismo de autentificación, donde los clientes deben además proporcionar nombre de usuario y contraseña para acceder al servicio. Como un dato adicional, Socks es abreviatura de Sockets.

- **Proxy a nivel de Aplicación**

  - **Proxy Transparente**

Un proxy transparente combina un servidor proxy con NAT de manera que las conexiones son enrutadas dentro del proxy sin configuración por parte del cliente, y habitualmente sin que el propio cliente conozca de su existencia. Este es el tipo de proxy que utilizan los proveedores de servicios de Internet.

Muchas organizaciones usan los proxy para reforzar las políticas de uso de la red o para proporcionar seguridad y servicios de caché. Normalmente, un proxy web o NAT no son transparentes a la aplicación cliente: deben ser configuradas para usar el proxy manualmente, por lo tanto, el usuario puede evadir el proxy cambiando simplemente la configuración.

Ventajas

- Teóricamente aumenta la rapidez para proporcionar la página solicitada por el usuario directamente desde el proveedor, sin acceder a servidores remotos que siempre serán más lentos.

- Resultan de gran beneficio para los proveedores por el considerable ahorro de ancho de banda, y por tanto de costos económicos.

Desventajas

- A su vez le puede provocar al usuario mayor lentitud al cargar la página solicitada debido a un desajuste o una incorrecta configuración del proxy.

- Pueden afectar los servicios de búsqueda como son los directorios elaborados en páginas estáticas.

- **Proxy Reverso**

Un proxy reverso es un servidor proxy instalado en el domicilio de uno o más servidores web. Todo el tráfico entrante de Internet y con el destino de uno de esos servidores web pasa a través del servidor proxy. Hay varias razones para instalar un “proxy reverso ":

- **Seguridad:** el servidor proxy es una capa adicional de defensa y por lo tanto protege los servidores web.
- **Encriptación / Aceleración SSL (Security Sockets Layer):** cuando se crea un sitio web seguro, habitualmente la encriptación SSL no la hace el mismo servidor web, sino que es realizada por el proxy reverso, el cual está equipado con un hardware de aceleración SSL.
- **Distribución de Carga:** el proxy reverso puede distribuir la carga entre varios servidores web. En ese caso puede necesitar reescribir las URL de cada página web.
- **Caché de contenido estático:** puede descargar los servidores web almacenando contenido estático como imágenes y otro contenido gráfico.

- **Proxy NAT / Enmascaramiento**

Otro mecanismo para hacer de intermediario en una red es el NAT. La traducción de direcciones de red NAT también es conocida como enmascaramiento de IPs. Es una técnica mediante la cual las direcciones fuente o destino de los paquetes IP son rescritas, sustituidas por otras (de ahí el "enmascaramiento").

Funcionamiento

Esto es lo que ocurre cuando varios usuarios comparten una única conexión a Internet. Se dispone de una única dirección IP pública, que tiene que ser compartida. Dentro de la LAN los equipos emplean direcciones IP reservadas para uso privado y será el proxy el encargado de traducir las direcciones privadas a esa única dirección pública para realizar las peticiones, así como de distribuir las páginas recibidas a aquel usuario interno que la

solicitó. Esta situación es muy común en empresas y domicilios con varios ordenadores en red y un acceso externo a Internet.

Ventajas

- El acceso a Internet mediante NAT proporciona una cierta seguridad, puesto que en realidad no hay conexión directa entre el exterior y la red privada y así nuestros equipos no están expuestos a ataques directos desde el exterior.
- Mediante NAT también se puede permitir un acceso limitado desde el exterior, y hacer que las peticiones que llegan al proxy sean dirigidas a una PC concreta que haya sido determinada para tal fin en el propio proxy.
- La función de NAT reside en los cortafuegos y resulta muy cómoda porque no necesita de ninguna configuración especial en los equipos de la red privada que pueden acceder a través de él.

- **Proxy de web / Proxy caché de web**

Se trata de un proxy para una aplicación específica: el acceso a la web. Aparte de la utilidad general de un proxy, proporciona una caché para las páginas web y los contenidos descargados, que es compartida por todos los equipos de la red, con la consiguiente mejora en los tiempos de acceso para consultas coincidentes. Al mismo tiempo libera la carga de los enlaces hacia Internet.

Funcionamiento

- El cliente realiza una petición (mediante un navegador web) de un recurso de Internet (una página web o cualquier otro archivo) especificado por una URL.
- Cuando el proxy caché recibe la petición, busca la URL resultante en su caché local. Si la encuentra, devuelve el documento inmediatamente, si no es así, lo captura del servidor remoto, lo devuelve al que lo pidió y guarda una copia en su caché para futuras peticiones.

Ventajas

- **Ahorro de tráfico**: Las peticiones de páginas web se hacen al servidor proxy y no a Internet directamente. Por lo tanto, aligera el tráfico en la red y descarga los servidores destino, a los que llegan menos peticiones.
- **Velocidad en tiempo de respuesta**: El servidor proxy crea una caché que evita transferencias idénticas de la información entre servidores durante un tiempo (configurado por el administrador) así que el usuario recibe una respuesta más rápida.
- **Demanda a usuarios**: Puede cubrir a un gran número de usuarios para solicitar a través de él los contenidos web.
- **Filtrado de contenidos**: El servidor proxy puede hacer un filtrado de páginas o contenidos basándose en criterios de restricción establecidos por el administrador dependiendo valores y características de lo que no se permite, creando una restricción cuando sea necesario.
- **Modificación de contenidos**: Basándose en la misma función del filtrado, tiene el objetivo de proteger la privacidad en Internet, puede ser configurado para bloquear direcciones por expresiones regulares y modifica en la petición el contenido.

Desventajas

- Las páginas mostradas pueden no estar actualizadas si éstas han sido modificadas desde la última carga que realizó el proxy caché.
- El hecho de acceder a Internet a través de un proxy, en vez de mediante conexión directa, impide realizar operaciones avanzadas a través de algunos puertos o protocolos.
- Almacenar las páginas y objetos que los usuarios solicitan puede suponer una violación de la intimidad para algunas personas. **[5]**

Ejemplos

ABC Proxy: Permite el uso compartido de una sola conexión a Internet por todos los usuarios de la red local. Soporta los protocolos HTTP, HTTPS, IRC, SMTP, POP3 y muchos más. Incorpora caché de sitios visitados para una más rápida navegación, entre otras opciones.

CProxy Server: Conecta varios ordenadores a Internet a través de una única conexión, brinda velocidad y seguridad. Incluye una caché en la que se almacenarán los sitios a los que más frecuentemente acudamos en Internet. Soporta los protocolos HTTP, HTTPS, FTP, SOCKS entre otros. **[12]**

WebCleaner: es un proxy HTTP de filtrado HTML, lo que significa que se puede navegar de forma más segura y rápida en la página web, desactivando imágenes animadas, eliminando o añadiendo cabeceras HTTP y apartando elementos HTML indeseables. Puede detectar fallos de seguridad en los procesos HTML. **[13]**

SQUID: es un proxy a nivel de aplicación para HTTP, HTTPS y FTP. También puede ejecutar peticiones DNS bastante más rápido de lo que puede hacerlo la mayoría del software cliente. Squid es ideal para acelerar el acceso a URLs y para controlar el acceso a sitios web. Squid es el servidor proxy más popular y extendido entre los sistemas operativos basados en UNIX. Es muy confiable, robusto y versátil. Al ser software libre, además de estar disponible el código fuente, está libre del pago de costosas licencias por su uso o con restricción a determinado número de usuarios.

Ventajas

- Soporta muchísimos protocolos de aplicación (HTTP, FTP, etc.)
- Tiene un avanzado mecanismo de autentificación (o sea, cuándo y a quién permitir utilizar el proxy).
- Permite actuar como caché de Internet, copiando contenido en forma local para que se le pueda acceder más rápido.
- Es software libre.
- Restricción de accesos a determinadas URLs.
- Crear jerarquías de caché.
- Levantar múltiples instancias.
- Verificar ficheros de logs.
- Controla ancho de banda con la herramienta Delays Pools.
- Puede acelerar la caché.
- Permitir o denegar accesos de navegadores a ciertas páginas. **[14]**

Desventajas

- La máquina donde funcionará el proxy debe tener capacidad de almacenamiento acorde a la caché que se necesite o se quiera.
- Debe tener un buen poder de procesamiento, ya que no es solo un reenvío de paquetes TCP/IP.
- En la mayoría de las veces es más rápido hacer NAT que utilizar un proxy.
- Hay que configurar la utilización del proxy en cada cliente.

ISA Server: Es una de las soluciones más importantes y conocidas del sistema operativo Windows. ISA Server 2000 puede configurarse como una solución integrada de firewall y proxy caché, o puede implementarse como un firewall o caché dedicado. Cuando se busca una solución robusta de firewall, ISA Server puede asegurar las redes con la filtración dinámica de paquetes, así como la detección de intrusión, el fortalecimiento de la seguridad del sistema y filtros "inteligentes" para aplicaciones. O de lo contrario si lo que desea es una solución dedicada de memoria caché puede usar ISA Server para mejorar la red con una memoria caché avanzada.

Ventajas

- Rapidez y seguridad.
- Buena integración con el sistema operativo Linux.

Desventajas

- En cuanto al rendimiento y la continuidad del servicio al hacerle modificaciones a su configuración.
- No permite realizar control de ancho de banda. (Lo implementa por una función del sistema operativo la cual presentó algunos problemas que provocó su eliminación en la versión 2004)
- No aplica políticas a las conexiones VPN. **[15]**

1.4 Diseños generales de aseguramiento perimetral.

Existen diversas arquitecturas para llevar a cabo el aseguramiento perimetral. Es preciso aclarar que estos diseños de seguridad perimetral son generales y que a la hora de hacer un diseño específico para una institución, depende de un análisis previo de la caracterización del escenario (equivalente a la descripción del negocio en ingeniería de software), del levantamiento de requisitos que se le haga a la institución, del problema que presenta la red en ese momento y que para resolverlo se requiera de un nuevo diseño y por último de la complejidad e importancia de la red y los recursos que se deseen proteger.

Filtrado de paquetes:

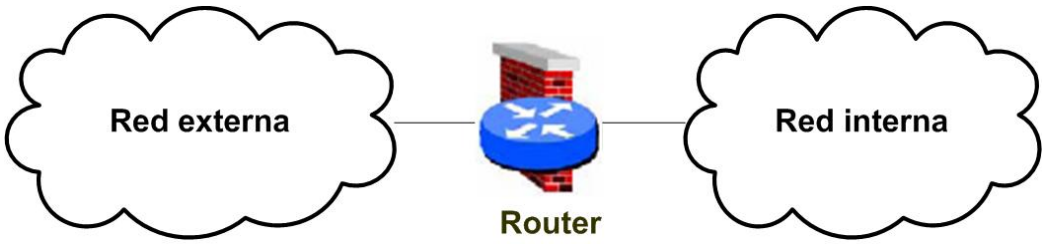

Figura1: Arquitectura Filtrado de paquetes.

Esta arquitectura es la más antigua de todas, basada simplemente en aprovechar la capacidad de algunos routers para hacer un encaminamiento selectivo, es decir, para bloquear o permitir el tránsito de paquetes mediante listas de control de acceso en función de ciertas características de las tramas, de forma que el router actúe como pasarela de toda la red.

En un cortafuegos de filtrado de paquetes los accesos desde la red interna al exterior que no están bloqueados son directos, por lo que esta arquitectura es la más simple de implementar y la más utilizada en organizaciones que no precisan grandes niveles de seguridad. El principal problema es que no disponen de un sistema de monitorización sofisticado, por lo que muchas veces el administrador no puede determinar si el router está siendo atacado o si su seguridad ha sido comprometida. Además, las reglas de

filtrado pueden llegar a ser complejas de establecer y por tanto es difícil comprobar la seguridad, habitualmente sólo se comprueba a través de pruebas directas, con los problemas de seguridad que esto puede implicar.

Dual-homed gateway:

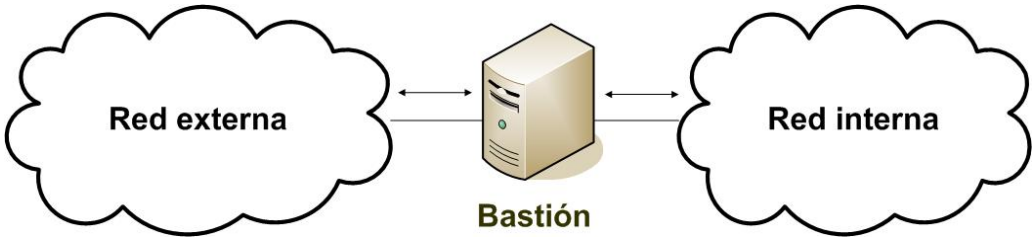

Figura 2: Arquitectura Dual-homed gateway.

Se trata de una PC con dos tarjetas de red, cada una de ellas conectada a una red diferente. El sistema ha de ejecutar al menos una aplicación proxy para cada uno de los servicios que se desee pasar a través del cortafuego y también es necesario deshabilitar la función de enrutamiento. Así, los sistemas externos verán a la PC a través de una de las tarjetas y los internos a través de la otra, pero entre las dos partes no puede existir ningún tipo de tráfico que no pase por el cortafuego. Todo el intercambio de datos entre las redes se ha de realizar a través de servidores proxy situados en la PC bastión.

La ventaja de estos sistemas es su sencillez, pues sólo requieren un ordenador. La desventaja es que sólo soportan servicios mediante proxy y no por filtrado de paquetes, ya que al tener la función de enrutamiento deshabilitada, se fuerza a que el tráfico deba ser tratado por una aplicación en el propio host.

Screened host:

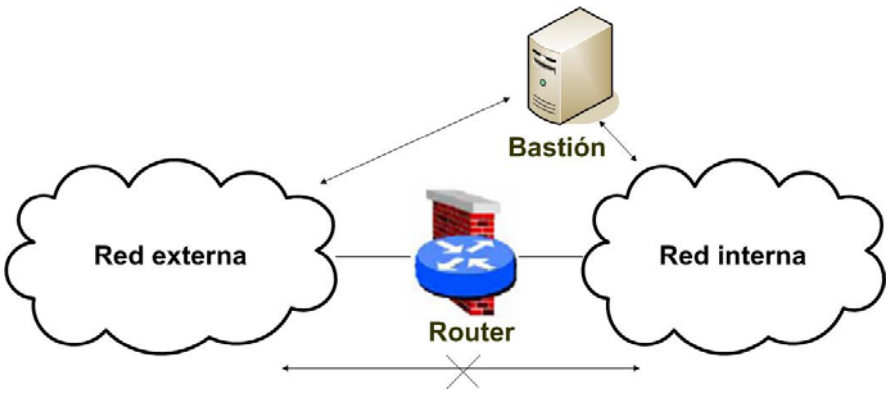

Figura 3: Arquitectura Screened host.

En este modelo la conexión entre las dos redes se produce mediante un router configurado para bloquear todo el tráfico entre la red externa y todos las PC de la red interna, excepto un único bastión, donde se instala todo el software necesario para la implementación del firewall. Esta topología nos permite soportar servicios tanto mediante proxy (en el bastión) como mediante filtro de paquetes (en el router). El problema de esta topología es que no hay nada previsto a nivel de seguridad entre el bastión y el resto de las PC internas, de modo que si un atacante logra entrar en el bastión, puede atacar la red interna, al igual que pueden producirse ataques internos hacia la PC bastión.

Esta arquitectura es un paso más en términos de seguridad de los cortafuegos al combinar un router con una PC bastión, el principal nivel de seguridad proviene del filtrado de paquetes, es decir, el router es la primera y más importante línea de defensa. En la PC bastión, único sistema accesible desde el exterior, se ejecutan los proxy de las aplicaciones, mientras que el router se encarga de filtrar los paquetes que se puedan considerar peligrosos para la seguridad de la red interna, permitiendo únicamente la comunicación con un reducido número de servicios.

Screened subnet:

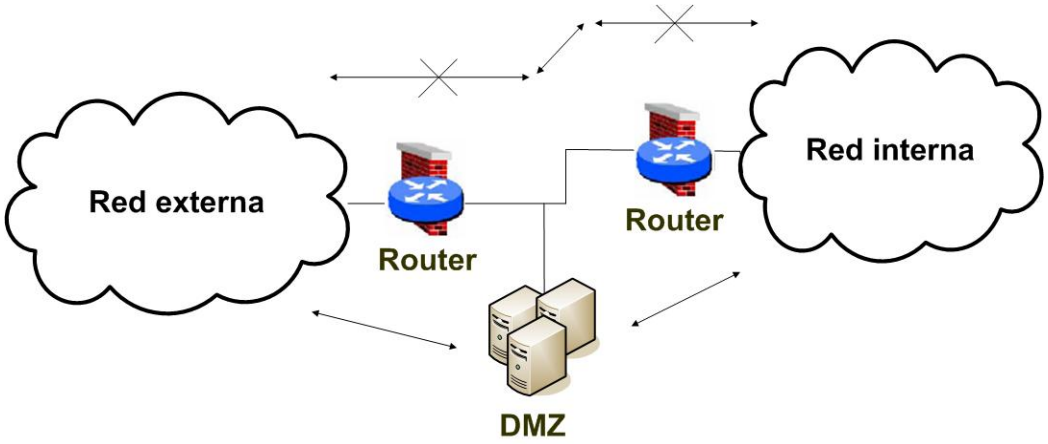

Figura 4: Arquitectura Screened subnet.

En este modelo se sitúa una red entre las dos redes a conectar. A ésta red se le conoce como red perímetro o zona desmilitarizada (DMZ) y se conecta a las otras dos mediante sendos routers. La DMZ añade un nivel de seguridad en las arquitecturas de cortafuegos, de forma que se consiguen reducir los efectos de un ataque exitoso a la PC bastión. Como este bastión es un objetivo interesante para muchos piratas, la arquitectura DMZ intenta aislarla en una red perimétrica de forma que un intruso que accede a esta PC no consiga un acceso total a la subred protegida.

Los routers se configuran mediante reglas de filtrado, para que tanto los nodos de la red interna como los de la externa, sólo puedan comunicarse con PCs de la red perímetro. Esto permite a la red interna ser efectivamente invisible a la externa.

Screened subnet es la arquitectura más segura, pero también la más compleja; se utilizan dos routers, denominados exterior e interior, conectados ambos a la red perimétrica. El router exterior tiene como misión bloquear el tráfico no deseado en ambos sentidos, mientras que el interior hace lo mismo pero con el tráfico entre la red interna y la perimétrica. De esta forma, un atacante necesitaría romper la seguridad de ambos routers para acceder a la red protegida. **[16]**

1.5 Características de equipos perimetrales.

Los equipos perimetrales son los encargados de garantizar la seguridad en una red, para esto, en ellos se implementan un grupo de facilidades, las cuales varían conforme al diseño de red utilizado. **[17]**

Las principales facilidades que se implementan en estos equipos son:

- Firewall por software.
- Servidor proxy.
- Programa gestor de ancho de banda.
- Programa medidor de ancho de banda, monitorizador del tráfico en una interfaz de red.
- Programa para analizar las trazas que generen tanto el firewall como el proxy.
- Servidor web para mostrar en formato HTML los reportes de las trazas.
- Software de autenticación para firewall.
- Software de autenticación para proxy.
- Programas detectores de intrusos.
- Antivirus. **[18]**

1.6 Tipos de ataques informáticos.

Desde el momento que nos conectamos a Internet, nuestro equipo se encuentra vulnerable a los diversos tipos de ataques informáticos existentes. Un Firewall evidentemente no protege contra ataques que no pasan por él, esto incluye todo tipo de ataques internos dentro del perímetro de seguridad. **[19]**

Clasificación de diferentes tipos de ataques contra sistemas de información:

Estas acciones se pueden clasificar de modo genérico en cuatro categorías generales según los efectos causados:

- **Interrupción:** cuando un recurso del sistema es destruido o se vuelve no disponible. Este es un ataque contra la disponibilidad. Ejemplos de este ataque son: la destrucción de un elemento hardware, como un disco duro y cortar una línea de comunicación.
- **Intercepción:** una entidad no autorizada consigue acceso a un recurso. Este es un ataque contra la confidencialidad. La entidad no autorizada podría ser una persona, un programa o un ordenador. Ejemplos de este ataque son: pinchar una línea para hacerse con datos que circulen por la red y la intercepción de datos.
- **Modificación:** una entidad no autorizada no sólo consigue acceder a un recurso, sino que es capaz de manipularlo. Este es un ataque contra la integridad. Ejemplos de este ataque son: alterar un programa para que funcione de forma diferente y modificar el contenido de mensajes que están siendo transferidos por la red.
- **Fabricación:** una entidad no autorizada inserta objetos falsificados en el sistema. Este es un ataque contra la autenticidad. Ejemplos de este ataque son: la inserción de mensajes falsos en una red y añadir registros a un archivo. **[20]**

Estos ataques también se pueden ordenar por modalidades según la forma de actuar:

- **Escaneo de puertos:** esta técnica consiste en buscar puertos abiertos, y fijarse en los que puedan ser receptivos o de utilidad.
- **Autentificación:** cuando un atacante suplanta a una persona con autorización.

- **Explotación de errores:** suceden en el momento que se encuentran agujeros de seguridad en los sistemas operativos, protocolos de red o aplicaciones.
- **Denegación de Servicio (DoS):** consiste en saturar un servidor con pedidos falsos hasta dejarlo fuera de servicio.
- **Modificación-Daño:** Consiste en una modificación no autorizada de datos o software instalado en el sistema victima (incluyendo borrado de archivos). **[19]**

Herramientas y aplicaciones de software usadas para llevar a cabo los ataques:
Programas Malignos (Código malicioso)

- **Virus**

Los Virus Informáticos pueden ser ingresados al sistema por un dispositivo externo (diskettes) o a través de la red (e-mails, chat u otros protocolos) sin intervención directa del atacante. Dado que el virus tiene como característica propia su auto-reproducción, no necesita de mucha ayuda para propagarse a través de una LAN o WAN, infectando cualquier tipo de archivo ejecutable, sin conocimiento del usuario, si es que no está instalada una protección antivirus en los servidores, estaciones de trabajo y los servidores de e-mail. **[2]**

- **Gusanos**

Es un código maligno cuya principal misión es reenviarse a sí mismo. Son códigos víricos que, en principio, no afectan a la información de los sitios que contagian, aunque consumen amplios recursos de los sistemas, y los usan para infectar a otros equipos. A diferencia de la mayoría de virus, los gusanos se propagan o reproducen por sí mismos de una unidad de disco a otra o a través de la red, mediante el correo electrónico u otro mecanismo de transporte sin necesidad de modificar su PC u ocultarse bajo otros programas para propagarse. **[19]**

- **Caballos de Troya (Troyanos)**

Pequeño programa malicioso capaz de alojarse en computadoras dentro de una aplicación, una imagen, un archivo de música u otro elemento de apariencia inocente, que

se instala en el sistema al ejecutar el archivo que lo contiene y permitir el acceso a usuarios externos, a través de una red local o de Internet, con el fin de recabar información. No se replica por sí mismo, pero su funcionalidad maliciosa está escondida en otros programas que parecen tener alguna utilidad, por lo que suele pasarse a otros equipos. **[2]**

Ejecución de comandos

- **Desbordamiento de buffer**

La explotación de un desbordamiento de buffer es un ataque que altera el flujo de una aplicación sobrescribiendo partes de la memoria.

- **Ataques de formato de cadena**

Alteran el flujo de una aplicación utilizando las capacidades proporcionadas por las librerías de formato de cadenas para acceder a otro espacio de memoria. **[21]**

Ataques Remoto

Se define Ataque Remoto como un ataque iniciado contra una PC sobre la cual el atacante no tiene control físico. Esta PC es distinta a la usada por el atacante y será llamada "víctima". Cada uno de los ataques abajo descriptos será dirigido remotamente.

- **TCP Connect Scanning**

Es la forma básica del escaneo de puertos TCP. La ventaja de esta técnica es su gran velocidad y que no necesita de privilegios especiales; y la desventaja radica en que es fácilmente detectable por el administrador del sistema.

- **TCP SYN Scanning**

La principal ventaja de esta técnica de escaneo es que pocos sitios están preparados para registrarlos. La desventaja es que en algunos sistemas Unix, se necesitan privilegios

de administrador para construir estos paquetes SYN. La aplicación del servidor escucha todo lo que ingresa por los puertos.

- **TCP FIN Scanning- Stealth Port Scanning**

Los paquetes FIN podrían ser capaces de pasar sin ser advertidos.

- **Fragmentation Scanning**

Es una modificación de técnica de escaneo. En lugar de reenviar paquetes completos de sondeo, los mismos se particionan en un par de segmentos IP. **[2]**

- **Eavesdropping-Packet Sniffing**

Muchas redes son vulnerables al eavesdropping, o a la pasiva intercepción (sin modificación) del tráfico de red. En Internet esto es realizado por packet sniffers, que son programas que monitorean los paquetes de red que están direccionados a la computadora donde están instalados. **[22]**

ATAQUES DE AUTENTICACIÓN

Este tipo de ataque tiene como objetivo engañar al sistema de la víctima para ingresar al mismo. Generalmente este engaño se realiza tomando las sesiones ya establecidas por la víctima u obteniendo su nombre de usuario y password. **[2]**

- **IP Spoofing**

Los protocolos de red también son vulnerables al spoofing. Con el IP spoofing, el atacante genera paquetes de Internet con una dirección de red falsa en el campo ¨From¨, pero que es aceptada por el destinatario del paquete. **[22]**

- **DNS Spoofing**

Consigue mediante la manipulación de paquetes UDP pudiéndose comprometer el servidor de nombres de dominios. Capacidad de un servidor de nombres para resolver una petición de dirección IP a partir de un nombre que no figura en su base de datos.

ATAQUES DE DENEGACIÓN DE SERVICIO

Los ataques por **denegación de servicio** implican una interrupción de los recursos de un sistema, lo que es suficiente para evitar que funcione normalmente. Los ataques por DoS pueden deberse a un ataque directo, o bien pueden estar provocados por virus, gusanos o troyanos. Los ataques por **denegación distribuida de servicio** implican la instalación en varios equipos de programas conocidos como zombis antes del ataque.

Los ataques por denegación de servicio y por denegación distribuida de servicio son los tipos más comunes de ataques en Internet. Se debe asegurar que siempre se esté al corriente sobre estos ataques y sobre cómo protegerse de los mismos. **[23]**

- **Jamming o Flooding**

Este tipo de ataque desactiva o satura los recursos del sistema. Por ejemplo, un atacante puede consumir toda la memoria o espacio en disco disponible, así como enviar tanto tráfico a la red que nadie más puede utilizarla.

- **Ping de la muerte**

Muchos servidores de Internet han sido dados de baja por el "ping de la muerte", una versión-trampa del comando ping. Mientras que el ping normal simplemente verifica si un sistema esta enlazado a la red, el ping de la muerte causa el reinicio o el apagado instantáneo del equipo. **[22]**

- **Syn Flood**

Se basa en un ‘saludo’ incompleto entre las dos PC. Aprovecha la mala implementación del protocolo TCP.

- **Connection Flood**

Existe un límite máximo en el número de conexiones simultáneas, una vez que se ataque ese límite no se admitirán conexiones nuevas. Si el atacante establece mil conexiones y no realiza ninguna petición sobre ellas monopolizará la capacidad del servidor.

- **Net Flood**

La red víctima no puede hacer nada, aunque filtre el tráfico en sus sistemas sus líneas estarán saturadas con tráfico malicioso, incapacitándolas para procesar tráfico útil. El atacante envía tantos paquetes de solicitud de conexión que las conexiones auténticas no pueden competir.

- **Land Attack**

Consiste en enviar a algún puerto abierto de un servidor un paquete maliciosamente construido con la dirección y puerto origen igual que la dirección y puerto destino.

- **Smurf o Broadcast Storm**

Es un ataque bastante simple y devastador que consiste en recolectar una serie de direcciones broadcast para mandar una secuencia de peticiones ICMP a cada una de ellas varias veces, falsificando la dirección IP de origen (máquina víctima) o colocar como dirección de retorno de los paquetes una dirección de broadcast.

- **OOB, Supernuke o Winnuke**

Hace que lo equipos que escuchan por el puerto NetBios sobre TCP/UDP queden fuera de servicio o disminuyan su rendimiento al enviarle paquetes UDP manipulados. **[2]**

1.7 Conclusiones del capítulo.

Al concluir este capítulo se han sentado las bases para el entendimiento de la tecnología firewall y proxy así como la necesidad de un alto nivel de seguridad en las redes informáticas.

Capítulo 2 Diseño.

Capítulo 2

2.1 Introducción del capítulo.

En este capítulo se realiza un análisis de las tecnologías expuestas en el capítulo anterior, realizando la selección de las más adecuadas según los objetivos planteados para el desarrollo del trabajo. Se explican los procedimientos y las herramientas que serán utilizadas, el por qué se escogió dicha herramienta y sus ventajas. Se explica la solución general del diseño realizado para la seguridad perimetral de las instituciones de salud cubanas y en específico la del banco de sangre de 23 y 2 y la del hospital Hermanos Ameijeiras.

2.2 Descripción y análisis de la problemática general.

2.2.1 Caracterización de la conectividad y seguridad de las instituciones de salud cubanas.

Las instituciones de salud en Cuba, en general casi todas, están en estos momentos en un proceso inversionista de crecimiento en cuanto a la cantidad de equipamiento informático instalado, su conectividad interna o cubrimiento de sus redes locales, el software o aplicaciones que utilizan y su conectividad a la red de salud pública.

Se hace necesario aumentar aún más la seguridad informática de estas instituciones, pues incrementa su tamaño y grado de ínter conectividad. Aumentan también los servidores, las aplicaciones y la importancia de la información que se procesa, almacena y transmite. Además de que cada día aparecen nuevos tipos de ataques externos con un nivel superior de sofisticación técnica.

En una etapa “cero” los policlínicos docentes, por ejemplo, se puede decir que poseen solo un enlace vía PPP telefónico conmutado, o un sencillo enlace arrendado, con de 2 a 4 PCs en un solo local, comúnmente en la biblioteca.

Un prototipo típico de instalación de salud cubana, podría ser un policlínico docente común, con por ejemplo: de dos a cuatro decenas de PCs conectadas en red, bien sea cableada o bien inalámbrica, dos servidores, uno de ellos de bases de datos, switch(s) capa 2 no administrable(s) y una conexión por modem-router a la red de Infomed con protocolo xDSL o con frame relay. En ese enlace arrendado las velocidades de acceso típicas por ahora varían entre 64 y 256 kbit/s.

2.2.2 Caracterización de las aplicaciones informáticas en las instituciones de salud cubanas.

Desde el punto de vista que interesa analizar para este trabajo, las categorías de aplicaciones informáticas que habrá en una primera etapa de desarrollo serán:

- Aplicaciones de gestión cliente servidor (con arquitectura de dos capas) procesando información médica y/o de gestión médico-organizativa, fundamentalmente almacenada en un servidor local de bases de datos y con ocasional acceso a bases de datos centrales externas vía webservices.
- Aplicaciones médicas de gestión, centralizadas en el nodo central del país y a las que se accede vía web desde la institución de salud.
- Aplicaciones de visualización y TxD de imágenes médicas.

En desarrollos a mediano y largo plazo, irán apareciendo más aplicaciones de tres capas, servidores de base de datos más “cargados” y potentes, habrá mayor interrelación entre las aplicaciones locales y las remotas e irá variando la proporción de equipos con software libre a favor de este último.

2.2.3 Definición de los flujos de información, volumen, importancia y necesidad de inmediatez.

Los flujos de información previstos por los canales arrendados, son los generados por las aplicaciones informáticas anteriormente descritas, más los provocados por el trabajo asistencial, docente e investigativo diario de médicos y técnicos de salud, no recogidos en el funcionamiento de las anteriores aplicaciones, como puede ser: navegación (nacional e internacional) para búsqueda de información científico - técnica, acceso web a la universidad virtual, correo electrónico, (con posible envío de imágenes médicas), descarga de files vía web o FTP, etc.

La mayoría de las sesiones TCP serán originadas desde adentro de la instalación local, lo que en cierta medida simplifica la colocación de reglas en el firewall.

Para realizar la gestión del ancho de banda conviene identificar los diferentes flujos que pueden circular por el canal arrendado, con vistas clasificarlos por diferentes criterios, agruparlos y aplicarles prioridades.

Identificación de principales flujos típicos por canal arrendado. (Nótese que no son protocolos, ni similares y si flujos de datos o tráfico).

Flujo

- Navegación nacional ordinaria.
- Navegación internacional directa desde el proxy de la institución.
- Navegación en webmail Infomed.
- Navegación internacional a través de un proxy en cascada con el de Infomed.
- FTP con Infomed y descarga de ficheros vía HTTP.
- Email tipo SMTP.
- TxD de imágenes.
- Cliente de webservices local a servidor web central.
- Navegación en aplicación web central.
- Tráfico administrativo SSH y otros.
- Tráfico de gestión de servidores de bases de datos (SQL / mySQL / postgreSQL).

- Tráfico de replicación de bases de datos.
- Tráfico de actualización de parches y antivirus. (Centralizado)
- Tráfico de actualización de parches y antivirus. (Individual)
- Publicación web (u otros) de servicios de la institución.
- Todo el resto del tráfico no especificado previamente.

Ejemplo: Tipos de tráficos existentes por tipos de instituciones.

- Un banco de sangre no tiene tráfico de imágenes.
- Un hospital que tiene servidor de email propio no debería tener tráfico HTTP a webmail en Infomed, pero si tiene tráfico tipo SMTP.
- Una institución pequeña que no tiene hecho el plan de seguridad informática no está aún preparada para navegar por Internet, por tanto, no posee tráfico web internacional.
- Una institución que tiene equipos médicos generadores de imágenes y da servicio a externos puede tener tráfico de imágenes fuerte. Igual un consumidor de esas imágenes, o un centro consultor (de diagnósticos).
- Un policlínico puede tener un tráfico web a aplicaciones centrales como el acceso al sistema de Atención Primaria de Salud que posiblemente no serán voluminosas en tráfico pero que si requieren inmediatez o interactividad alta.

Algunos de estos flujos deberán circular por canales encriptados o con algún tipo de protección criptográfica, otros no lo necesitan y de hacerlo aumentaría el volumen de tráfico innecesariamente. **[24]**

2.3 Recomendaciones para adaptar el diseño a cualquier institución de salud.

Se propone una secuencia de pasos para la configuración, montaje y explotación de la solución. Estos pasos no son una guía rígida o inflexible, pueden ser adaptados a las condiciones de cada caso.

1. Caracterización del escenario

Se hace un estudio de la red y se le caracteriza, de ser posible se selecciona uno de los escenarios que se tipifican a continuación, como descripción general de la misma y se especifican las particularidades que la diferencian. Se documenta.

2. Levantamiento de requisitos

Se investigan y enumeran los requisitos que se necesitan para conocer el funcionamiento de la red y así poder crear las reglas de permisividad y negación del firewall y el proxy, para poder ofrecer las funcionalidades que se esperan de la red. En el levantamiento de requisitos son fuente de información tanto los usuarios de la red como los diseñadores de las soluciones de software montadas y/o los manuales de explotación de los mismos. Se documenta el levantamiento.

3. Selección y diseño de la solución

Se hace una propuesta de hardware, de arquitectura de solución y se programan las reglas en lenguaje, script o formato específico seleccionado para lograr las facilidades tanto del firewall como del proxy, guiándose estrictamente por el levantamiento de requisitos que se realizó previamente. Se documenta detalladamente todo el diseño de la solución, para poder someterlo a análisis, crítica de terceros especialistas y evaluación por el control de calidad.

4. Compra y montaje del firewall y el proxy

Adquisición y montaje del equipamiento, implementación (activación) del grupo de reglas para el firewall y el proxy según corresponda. Debe tenerse en cuenta de que si la red ya estaba en funcionamiento esto no deberá causar o habrá al menos que minimizar las interrupciones en los servicios que ya se brindan.

5. Pruebas del firewall y el proxy

Pruebas de que permiten el paso a aquellos protocolos y direcciones que se deben permitir. El conjunto de reglas de filtraje del firewall y del proxy no deben obstaculizar el funcionamiento normal o flujo de información permitido. Se prueban detalladamente cada uno de los flujos que se documentaron durante el levantamiento de requisitos.

Pruebas de que no dejan pasar a aquellos que deben limitar. Se diseñan y realizan todas las pruebas que sean posibles (imposible probar todo) de los flujos prohibidos, por ejemplo se prueba la visibilidad de puertos desde direcciones externas, intentos de tráficos, intentos de navegar en sitios no permitidos, verificación de que no todos los usuarios tienen permiso para navegar, etc.

6. Administración continúa

El firewall y el proxy no son dispositivos que pueden ser diseñados, instalados y olvidados, necesitan ser administrados, es decir hay que dedicarle un mínimo de atención continua a través del tiempo, como revisión de trazas, actualización del software, actualización de las reglas ante requisitos cambiantes, colocación de nuevas reglas ante vulnerabilidades detectadas, etc.

2.4 Solución general propuesta.

Después de haber hecho un análisis del estado del arte del tema a tratar se propone la solución general al problema.

2.4.1 Selección de los softwares específicos a utilizar.

De la lista de tipos de firewall y proxy vista en el capítulo anterior, se hizo una comparación de cada una de las alternativas o variantes de estas tecnologías, para decidir cuál escoger y así llevar a cabo el diseño de la red de las instituciones de salud.

Selección del Firewall

IPTABLES

Después de realizar el análisis de la comparación de los tipos de firewall existente, se decidió escoger la alternativa de Iptables, también conocido como Netfilter, del cual su antecesor era el nombrado Ipchains. Iptables cumple las condiciones requeridas: protege el perímetro de la red asignada, funciona con control de estado de la conexión, su arquitectura está basada en software y es uno de los mejores cortafuegos disponibles, incluyendo soluciones comerciales y desde luego, la mejor solución gratuita por su potencia y versatilidad.

Iptables permite configurar un firewall cuya función consistirá básicamente en analizar todo el flujo de tráfico entrante y saliente y/o enrutado a través de nuestra red y tomar decisiones sobre cada paquete en base a unas reglas definidas.

Iptables es un sistema de firewall vinculado al kernel de Linux que se ha extendido enormemente a partir del kernel 2.4 de este sistema operativo. A diferencia de un firewall tradicional, que puede levantarse o bajarse como un servicio, o que puede caerse debido a un error de programación, iptables es un firewall que está integrado al kernel, es decir, es parte del sistema operativo.

Lo que en este caso se hace, en lugar de levantar el servicio, es aplicar reglas. Para esto se aplica el comando iptables con el cual añadimos, borramos o creamos reglas. Iptables

es una aplicación en línea de comandos que gestiona el filtrado de paquetes en sistemas Linux. Por lo tanto un firewall de iptables no es más que un simple script de shell en el que se van a ejecutar las reglas a aplicar.

La regla o política por defecto que se escogió fue DROP o DENEGAR, puesto que es la más segura y recomendable, ya que deniega todo y solo permite pasar lo que esté explícitamente permitido en el firewall, convirtiéndose este en un auténtico ‘’muro’’ infranqueable.

Selección del Proxy

SQUID

Indiscutiblemente para el diseño de esta solución, el software ideal para cumplir con las tareas indicadas sobre proxy es el Squid, por ser hoy en día el más popular y ampliamente utilizado en los sistemas operativos como Linux y derivados de Unix, es muy robusto, confiable y versátil y se distribuye bajo la licencia de software libre, característica muy importante debido a que es uno de los requisitos no funcionales que exigen las instituciones de salud cubanas. El Squid tiene una amplia variedad de utilidades, desde acelerar un servidor web guardando en caché peticiones repetidas, hasta caché de web y otras búsquedas para un grupo de gente que comparte recursos de la red. Otras posibilidades que añade son: seguridad filtrando el tráfico debido a que se puede permitir o denegar el acceso a determinadas URL, pedir autenticación, analizar los log que genera para mantener un control de las URL visitadas, acelerar la caché para garantizar la velocidad cuando se entrega un sitio que está en la caché y así optimizar la utilización del ancho de banda, puede trabajar debajo o en paralelo con otro proxy y además permite levantar en una misma máquina varias instancias con configuraciones totalmente diferentes. Es en definitiva, el mejor de los proxy-caché para compartir el acceso a la red de redes por varios clientes.

2.4.2 Integración de los softwares en el equipo perimetral.

La integración de varios programas en un mismo equipo, lleva consigo un análisis previo de las características de cada uno por separado, para saber si existe algún tipo de problema entre ellos que pueda entorpecer su buen funcionamiento y la instalación de cada uno por separado, para lograr que cumplan todas las funcionalidades que se desean sin interferir unos con otros. Es aquí donde cada cual hace lo que le corresponde.

La computadora que se asigne como equipo perimetral tendrá integrado una amplia variedad de softwares para facilitar o hacer posible el buen funcionamiento de la red.

Los programas seleccionados para integrar son:

- El **Squid** como proxy con todas sus funcionalidades incluyendo el módulo de autenticación smb_auth con la previa instalación del servidor Samba para lograr que cada usuario del proxy se autentique.
- El **Iptables** como firewall que no es más que un script en el que se van ejecutando una serie de reglas, las cuales pueden ser muy seguras.
- El **SARG** como software que analiza los log que genera el squid para mostrarlos de una forma más organizada y comprensible para el administrador del proxy.
- El **Apache** para mostrar todos los reportes en HTML generados por el SARG.

Todos estos programas funcionando en una misma computadora constituye una solución factible para garantizar, sin requerir más equipamiento, la seguridad y el buen funcionamiento de una red, permitiendo así, que todo el tráfico que pase por este equipo perimetral sea controlado y que cada software por si solo haga lo que le corresponde, sin interferir en el trabajo de los otros programas que están en el equipo perimetral.

2.5 Tipificación de soluciones.

En este epígrafe está enmarcado todo el análisis requerido para proponer una solución de diseño típica para cualquier institución de salud cubana. Teniendo en cuenta la descripción de los posibles escenarios y patrones de solución, la descripción específica de cada uno de los parámetros que muestran como está conformada la red físicamente, la captura de los requisitos necesarios para satisfacer las necesidades de la institución y por último el diseño típico para las posibles soluciones.

2.5.1 Descripción de posibles escenarios y patrones de solución.

Los siguientes escenarios y patrones de solución son los válidos o posibles a encontrar en las instituciones de salud cubanas, como por ejemplo: bancos de sangre, policlínicos y hospitales.

Firewall

Escenario # 1

Red local pequeña o mediana que navega y utiliza servicios de Internet pero que no ofrece ningún servicio público.

Patrón # 1

Firewall con dos NICs, por tanto dos zonas de seguridad. A una llámesele zona insegura o Internet y a la otra zona segura o interna.

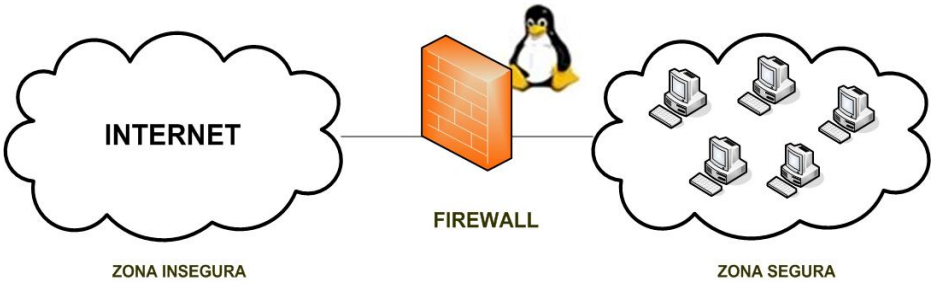

Figura 5: Escenario y patrón 1.

Escenario # 2

Red local empresarial que tiene pocos servicios públicos como un servidor web estático y un servidor de correo propio que interactúa con Internet.

Patrón # 2

Firewall con 3 NICs, tres zonas de seguridad. Una llamada zona insegura o Internet, otra zona muy segura o de servidores y una tercera zona de seguridad que llamaremos zona segura o interna.

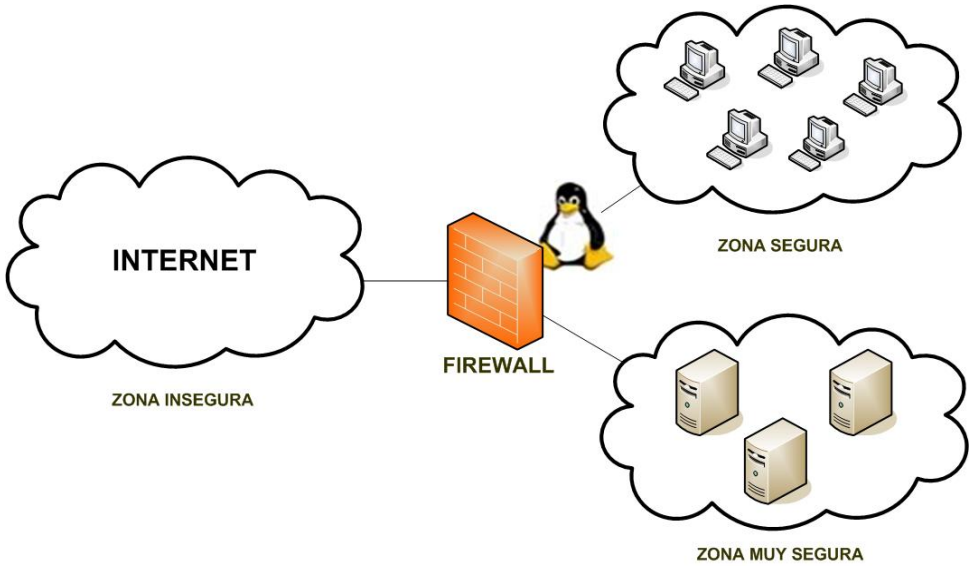

Figura 6: Escenario y patrón 2.

Además en la tipificación también se incluyen los scripts de Iptables, obteniéndose patrones de scripts a aplicar en cada caso. De cualquier forma estos se comportan, precisamente, solo como eso: patrones, pues se pretende particularizar cada solución a una institución concreta.

En cada institución se realizará un levantamiento de requisitos y se hará un diseño específico para cada una basado en los patrones de su tipo.

Proxy

Escenario # 1

Red local pequeña o mediana que navega a través del proxy y utiliza servicios de Internet.

Patrón # 1

Proxy con 2 NICs, dos zonas de seguridad, una zona insegura o zona Internet y una zona segura o zona de red interna.

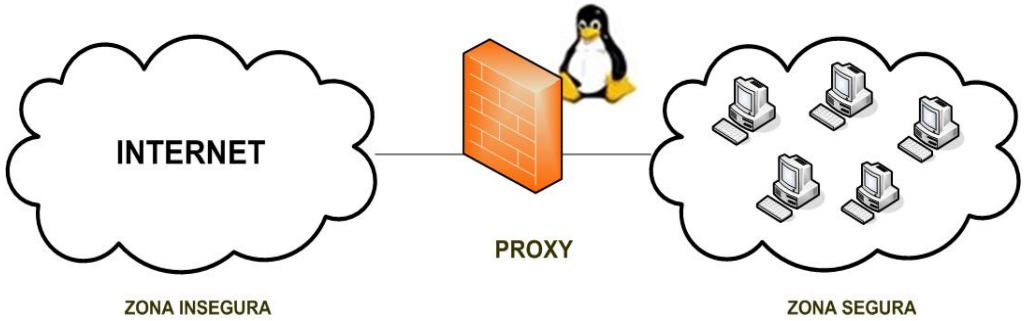

Figura 7: Escenario y patrón 1.

Escenario # 2

Red local pequeña o mediana que navega a través de un proxy en cascada con otro proxy y utiliza servicios de Internet.

Patrón # 2

2 Proxy, con uno de estos en cascada, con 2 NICs cada uno, dos zonas de seguridad, una zona insegura o zona Internet y una zona segura o zona de red interna.

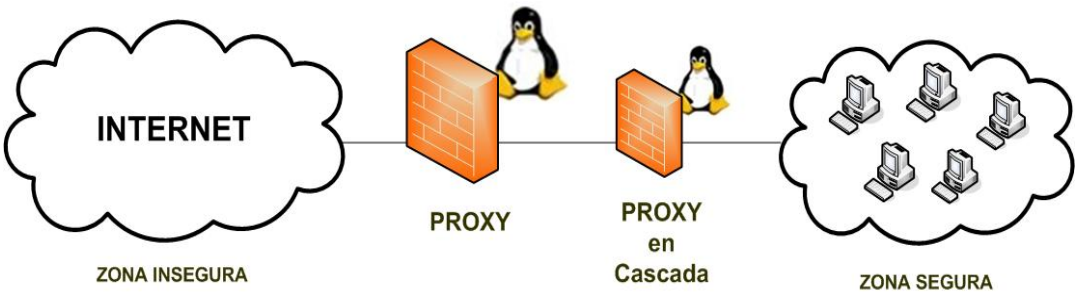

Figura 8: Escenario y patrón 2.

Escenario # 3

2 Proxy, uno de estos con dos instancia, recogiendo en la primera el escenario # 1 y en la segunda el escenario # 2, cada proxy con 2 NICs, dos zonas de seguridad.

Patrón # 3

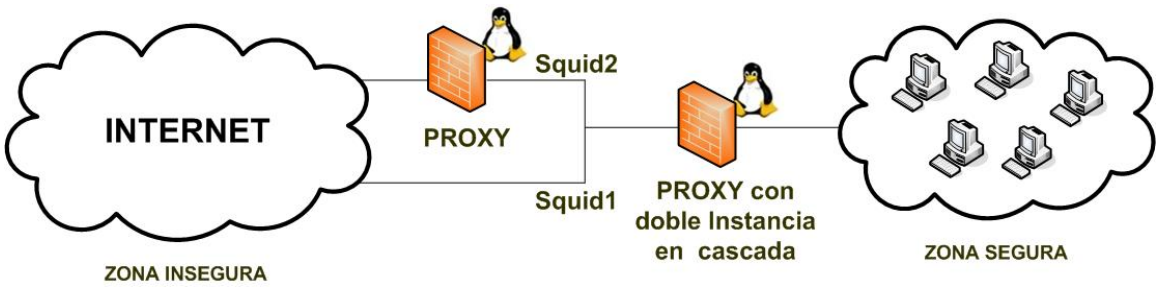

Figura 9: Escenario y patrón 3.

2.5.1.1 Parámetros a seguir en cualquier descripción típica de escenarios.

Para asignarle un patrón de solución a un escenario de una institución específica de salud, es necesario tener en cuenta la descripción de algunos parámetros que muestran como está conformada la red, o sea, que características presenta esa institución. A continuación se muestran algunos de estos posibles parámetros:

- Cantidad de PC con que cuenta la red LAN.
- Tipo de conexión.
- Ancho de banda de la conexión.
- Si existe DHCP.
- Si existe Switch, qué capa es y si es administrable.
- Si existe facilidad de VLAN.
- Equipo de conectividad.
- Si existe administración local permanente.
- Sistema operativo de las PC y servidores.
- Tipo de aplicaciones que corren en la red local.
- Tipo de autentificación de las aplicaciones con la base de datos.
- Si existe proxy.
- Si existe servidor de correo.
- Si existe servidor de dominio, etc.

2.5.2 Levantamiento de posibles requisitos típicos a encontrar en cualquier institución de salud.

Después de haber caracterizado el escenario de la institución que se visite, se realiza la captura de los requisitos para conocer las necesidades que se deseen satisfacer con la solución propuesta. Se toman como parámetros algunos requisitos genéricos enmarcados a continuación, que muestran el funcionamiento de la red.

- Protección especial para los servidores (si existen) aislándolos del resto de la LAN.
- Acceso desde las estaciones de trabajo a los servidores.
- Navegación de usuarios de la red local.
- Aplicaciones existentes en las instituciones de salud.
- Generales de seguridad.
- Requisitos NO Funcionales del equipo perimetral (equipamiento), etc.

2.5.3 Diseño típico de posibles soluciones.

Una vez realizada la caracterización del escenario y el levantamiento de los requisitos de la institución, se selecciona la solución específica que requiere la misma teniendo en cuenta las soluciones dadas a los posibles escenarios vistos anteriormente.

Esto es un proceso de selección y adaptación que se propone para la solución del problema. Se detallan las zonas que tiene la institución para conocer cuales requieren una mayor seguridad y se describen las características específicas que tiene el equipo perimetral.

Definición de las zonas:

Se definen las zonas de seguridad que necesite la institución y se especifica cada una.

- **Zona insegura** (Zona 1) (Zona Internet) (Red del MINSAP).
- **Zona segura** (Zona 2) (Zona LAN de usuarios).
- **Zona muy segura** (Zona 3) (Zona servidores de base de datos).

Características del equipo perimetral:

Después de conocer las zonas que se van a asegurar, se describen las características que tendrá el equipo perimetral:

- **Sistema Operativo** (Ej: Linux Debian-Sarge. Sin modo gráfico).
- **Software de firewall** (Ej: Iptables (Netfilter)).
- **Software de proxy** (Ej: Squid).
- **Cantidad de NICs con que cuenta la PC** (Ej: PC con 3 NICs (sin monitor, ni mouse, ni audio)).
- **Interfases de la tarjeta de red** (Ej: Una tarjeta de red haciendo interfase con 3 zonas.).

Especificación de las zonas:

- **Zona 1**

- Direcciones IP oficiales.
- Direcciones IP especificadas por el proveedor.

- **Zona 2**

- Direcciones IP privadas. Subred IP diferente a la de zona 3.
- Red.
- Dirección IP de la interfase del equipo perimetral.

- **Zona 3**

- Direcciones IP privadas. Subred IP diferente a la de Zona 2. Cross-over al servidor de base de datos.
- Red.
- Dirección IP de la interfase del equipo perimetral.

Configuración del Iptables:

- Política de firewall.
- Reglas necesarias para cada subred.

Configuración del Squid:

- Squid con conexión directa a Internet.
- Squid en cascada con otro proxy.
- Squid con múltiples instancias.

2.6 Perdurabilidad, sostenibilidad y robustez del diseño.

Se tomaron medidas para que el diseño (o la solución) sea perdurable en el tiempo, o sea que perdure en la institución.

Estas medidas serán necesarias debido a que permitirá:

- Extender la vida útil de la solución implantada.
- Comprobar que se ejecuta de forma adecuada.
- Asegurarse de que sigue proporcionando un ambiente seguro y apropiado para la empresa.
- Optimizar su operación y sus servicios.
- Realizar cualquier actualización necesaria.
- Asegurar que los componentes siguen funcionando e interactuando entre sí.

Al realizarse una afinación periódica de la solución se es capaz de evaluar de manera adecuada la carga de operación que se está soportando o su capacidad para enfrentar y anticipar problemas en el futuro. Al modelar su desempeño con respecto a un número cada vez mayor de cargas de operación medidas, se podría tener un buen panorama general sobre los signos vitales.

Posibles situaciones adversas o siniestros:

- Caída de la red por fallas eléctricas.
- Rotura del disco duro.
- Desconfiguración de algún software, etc.

Soluciones de emergencias y alternativas óptimas propuestas frente a posibles averías:

- Documentación de usuario bien detallada (manuales de usuario de las soluciones que se implanten).
- Documentación técnica (manuales de instalación, montaje y configuración).
- CD de instalación con software necesarios.

2.6.1 Scripts de iptables contra tipos de ataques.

De todos los tipos de ataques expuestos en el capítulo anterior, se han escogido aquellos a los que se puede ser más vulnerables y se han programado scripts de iptables con técnicas de firewall avanzadas para contrarrestar la mayoría de las amenazas o ataques que existen a una red.

Protege contra Ping de la Muerte

(Se podría restringir más, aceptando de sólo las IP que se desee, no de todas)

iptables -A INPUT -p icmp --icmp-type echo-request -m limit --limit 1/s -j ACCEPT

Protege contra Footprinting (Primera etapa de ataque)

iptables -A INPUT -p icmp -m icmp --icmp-type address-mask-request -j DROP

iptables -A INPUT -p icmp -m icmp --icmp-type timestamp-request -j DROP

Protege contra Traceroute

iptables -A OUTPUT -p icmp -m icmp --icmp-type destination-unreachable -j DROP

iptables -A OUTPUT -p icmp -m icmp --icmp-type time-exceeded -j DROP

Protege contra escaneo

iptables -A INPUT -p tcp -m tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

iptables -A FORWARD --protocol tcp --tcp-flags ALL SYN, ACK -j DROP

iptables -A FORWARD -p tcp --tcp-flags SYN, ACK, FIN, RST RST -m limit --limit 1/s -j ACCEPT

Protege contra tipos de paquetes SYN

(Se podría restringir más, aceptando de sólo las IP que se desee, no de todas)

iptables -A INPUT -p tcp --syn -m limit --limit 1/s -j ACCEPT

Protege contra ataques DoS

iptables -A FORWARD -p tcp -m limit --limit 1/s -j ACCEPT

Protege contra escaneo de puertos UDP

iptables -A INPUT -p udp -m udp --sport 32769:65535 --dport 33434:33523 -j DROP

Protege contra ataque Jamming o Flooding

iptables -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \ ALL RST --dport ssh -j ACCEPT

Protege contra Spoofing de IP

iptables -A INPUT –i $EXTIF -s 10.0.0.0/8 -j DROP
iptables -A INPUT -i $EXTIF -s 172.16.0.0/12 -j DROP
iptables -A INPUT -i $EXTIF -s 192.168.0.0/16 -j DROP

Protege contra Smurf o Broadcast Storm

iptables -A INPUT -p icmp --icmp-type 8 -j DROP

Guarda los log de los paquetes

iptables -A FORWARD -m limit --limit 3/minute --limit-burst 3 -j LOG \
--log-level DEBUG --log-prefix "IPT FORWARD packet died:”

2.7 Casos de implantación.

Las instituciones seleccionadas para implantar el diseño de red realizado en este trabajo de diploma han sido: el banco de sangre de 23 y 2 en el Vedado y el hospital Hermanos Ameijeiras. La solución ha sido probada exitosamente en diversas simulaciones en un laboratorio. En fase posterior se pasó a probarla en condición de piloto en el banco de sangre de 23 y 2, institución pequeña-mediana que resuelve ahora esta solución: protección perimetral, con todas las facilidades de un firewall de estado (statefull firewall) y en el hospital Hermanos Ameijeiras con la solución proxy y sus funcionalidades básicas y avanzadas.

2.7.1 Banco de sangre de 23 y 2 en el Vedado.

Al diseño que existía anteriormente en esta institución (hecho por la empresa Softel) se le hicieron modificaciones para su mejora, con el objetivo de que las reglas de filtraje del firewall fueran menos permisibles y proporcionaran una mayor seguridad a la red del lugar. Este diseño consta solamente con la solución de firewall, debido a que esta institución es pequeña-mediana y no necesita de un proxy propio ya que todos sus usuarios navegan mediante el proxy que se encuentra en la red de Infomed.

2.7.1.1 Ejemplo específico del diseño del banco de sangre de 23 y 2.

Escenario y Requisitos

- **Descripción del Escenario**
  - LAN de entre 30 y 60 PC.
  - Conexiones a la LAN alambradas de 100 Mbits/seg.
  - Switch capa 2 no administrable. Sin facilidades de VLAN.
  - Conexión arrendada la red pública, configurada para conectarse solo con la red del MINSAP (Infomed). Conexión HDSL de 256 kbits/s.
  - Modem - Router. - asincrónico HDSL.
  - No hay administración local permanente.
  - Servidor de dominio Windows 2000 Server. La mayoría de las estaciones de trabajo con Windows XP.

- Existe aplicaciones cliente servidor de dos capas. Con un servidor de base de datos MS-SQL 2000 Server Enterprise.
- Conexión a la base de datos por autentificación estándar de SQL-2000. En las estaciones clientes con Windows XP corren aplicaciones basadas en Visual Basic 6 y Delphi 5/6.
- No hay servidor proxy en este momento. No hay servidor email actualmente.
- Corre una aplicación que está conectada directamente a la base de datos por SQL y además se conecta vía HTTP a webservices localizados en el nodo central de Infomed y transfiere información.
- Hay DHCP en la red.
- No se conoce el listado del conjunto de subredes IP que conforman la red de salud pública. Red de 35000 usuarios, Al no tener listados, conocimientos, control de ella, etc. se debe considerar insegura.
- Los administradores de la red de Infomed expresan que realizan un control de filtraje entre la Internet exterior y la red privada.

- **Levantamiento o captura de Requisitos**

Suministro de equipamiento (hardware)

- Conexión a UPS más grupo electrógeno capaz de suministrar servicio ininterrumpido confiable.
- PC de hardware muy confiable en cuanto a fallas para 24 horas por 365 días de funcionamiento, con categoría de misión crítica.
- Disco duro de 10 Gbytes, lector de CD, lector de diskettes, 3 NICs de 100 Mbit/s, 256 Mbytes de RAM, teclado, un puerto USB. No mouse ni monitor.
- Todas las NICs deben tener drivers para Linux (Debian). No tienen que ser iguales necesariamente. Puede ser una onboard y dos adicionales.
- Cross-overs, o latiguillos invertidos (5 ‘).
- Latiguillos de red (normales) (5 ‘).
- Las PC de todos los bancos de sangre deben ser iguales entre sí.

Sistema de Bancos de Sangre

- Protección especial para el servidor de base de datos aislándolo del resto de la LAN.
- Acceso desde las estaciones de trabajo al servidor de base de datos vía conexión puerto 1433, con solo autenticación estándar de MS-SQL.
- Acceso vía HTTP sin proxy desde la red local a web-services ubicados en Infomed.
- Acceso desde ARSA a administrar el servidor de base de datos (transferir ficheros y conexión SQL).
- Acceso desde ARSA a administrar el Firewall (por protocolo ssh).

Usuarios de la red local

- Los usuarios de la red local navegan en la red privada de Infomed.
- Los usuarios de la red local utilizan el proxy de Infomed.
- Los usuarios de la red local usan el email de Infomed.

Generales de seguridad

- Todo aquello que no esté explícitamente especificado se prohíbe.
- Protecciones estándares contra ataques ICMP.
- Protecciones estándares contra ataques TCP y UDP.
- Seguridad máxima posible.
- Se debe dejar abierto en el diseño la posibilidad para que, con el mismo hardware y sistema operativo, se pueda incorporar próximamente la gestión de ancho de banda, detección de intrusos y VPN.

Diseño

- **Definiciones**

Se definen tres zonas de seguridad.

1. Red de Infomed. (Zona Internet, Zona insegura o Zona 1)
2. LAN de usuarios. (Zona de seguridad 2)
3. Server de base de datos. (Zona de seguridad 3, o Zona de base de datos)

Al no conocerse el listado del conjunto de subredes IP que conforman la red de salud pública. Se asume que todo el exterior es Internet e inseguro.

Se realizará la implementación con un firewall por software con las siguientes características:

- PC con 3 NICs (sin monitor, ni mouse, ni audio)
- Sistema Operativo: Linux Debian-Sarge. Sin modo gráfico.
- Software de firewall: Iptables (Netfilter).
- Una tarjeta de red haciendo interfase con cada zona.

➢ **Especificaciones de las zonas**

Zona 1: Eth0

- Direcciones IP oficiales. Subred para comunicarse con router. Cross-over al router.
- Direcciones IP especificadas por el proveedor.

Zona 2: Eth1

- Direcciones IP privadas. Subred IP diferente a la de zona 3. Conexión al switch de la LAN.
- Red 192.168.1.0/24 (255.255.255.0).
- Dirección IP de la interfase del firewall 192.168.1.100.

Zona 3: Eth2

- Direcciones IP privadas. Subred IP diferente a la de Zona 2. Cross-over al servidor de base de datos.
- Red 192.168.2.0/28 (255.255.255.240).
- Dirección IP de la interfase del firewall: 192.168.2.1.
- Dirección IP del servidor de base de datos: 192.168.2.2.

➢ **Configuración del Iptables**

- Todo tráfico negado por defecto.
- Dnat de zona 2 a zona 1 para acceder a red externa.
- Snat de zona 1 a zona 2 entrante para administrar. Estrechamente filtrado por reglas explícitas.
- Ruteo entre zona 2 y zona 3. Filtrado por reglas.
- Paso de conexiones establecidas y relacionadas (statefull filtering).
- Conjunto de reglas que satisfagan los requisitos de diseño.

Reglas de objeto a objeto (subredes/IP):

De subred IP 192.168.1.0/24 a la dirección IP 192.168.2.2

- Puerto 1433 TCP (MS-SQL).
- ICMP necesario.

De subred IP 192.168.1.0/24 al resto del mundo pero solo por la interfase Eth0

- Puertos 80 TCP y 443 TCP (HTTP y HTTPS - navegación saliente).
- Puertos 21 TCP y 20 TCP (FTP saliente).
- Puerto 53 TCP y UDP (interrogación DNS externo).
- ICMP necesario.

De subred IP 192.168.1.0/24 al proxy de Infomed**.**

- Puertos 3128 TCP.
- ICMP necesario.

De subred IP 192.168.1.0/24 al SMTP servidor de Infomed**.**

- Puerto 25 TCP.

De subred IP 192.168.1.0/24 al POP3 servidor de Infomed**.**

- Puerto 110 TCP.

De 200.55.161.62 al firewall (INPUT)

- Puerto 22 TCP.

- ICMP necesario.

De 200.55.161.62 al servidor de base de datos

- Puerto 22 TCP (mapeado como 1022 en el firewall).
- Puerto 1433 TCP (mapeado como 2433 en el firewall).

2.7.1.2 Script de iptables hecho para el banco de sangre de 23 y 2.

Para mejorar el diseño de seguridad que existía en esta institución se reafinó la programación del script de Iptables, es decir, se transformaron las reglas de filtraje para proporcionarle una mayor restricción en las conexiones.

```
#!/bin/sh
echo Comenzando firewall Banco Sangre 23 y 2.
echo IP Forwarding.
echo 1 > /proc/sys/net/ipv4/ip_forward
echo Seteo de Constantes.
EXTIF="eth0"                 # interfase externa
INTIF1="eth1"                # interfase interna de la LAN de usuarios
INTIF2="eth2"                # interfase interna de la base de datos
EIP="201.220.193.6"          # dirección IP externa
IIP1="192.168.1.100"         # dirección IP interna 1
IIP2="192.168.2.1"           # dirección IP interna 2
LOCALLAN="192.168.1.0/24"  # PC clientes
DBSERVER="192.168.2.2"     # servidor de bases de datos
ARSA="200.55.161.62"         # IP real de ARSA-Softel
echo Limpieza de reglas previas.
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
echo Reglas drop por defecto
```

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
echo Permitir todo lo que entra y sale del Localhost
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
echo Permitir todo lo establecido y relacionado
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
#De subred IP 192.168.1.0/24 a IP address 192.168.2.2
#Puerto 1433 TCP (MS-SQL)
#ICMP necesario.
iptables -A FORWARD -s 192.168.1.0/24 -p TCP -d 192.168.2.2 --dport 1433 -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -p ICMP -d 192.168.2.2 -j ACCEPT
#De subred IP 192.168.1.0/24 a resto del mundo pero solo por la interfase Eth0, la externa
#Puertos 80 TCP y 443 TCP (HTTP y HTTPS - navegación saliente)
#Puertos 21 TCP y 20 TCP (FTP saliente)
#Puerto 53 TCP y UDP (interrogación DNS externo)
#ICMP necesario.
iptables -A FORWARD -s 192.168.1.0/24 -p TCP -o $EXTIF -m multiport --dport 80,443,21,20,53 -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -p UDP -o $EXTIF --dport 53 -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -p ICMP -o $EXTIF -j ACCEPT
#De subred IP 192.168.1.0/24 a proxy de Infomed.
#Puertos 3128 TCP
#ICMP necesario
iptables -A FORWARD -s 192.168.1.0/24 -p TCP --dport 3128 -o $EXTIF -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -p ICMP -o $EXTIF -j ACCEPT
#De subred IP 192.168.1.0/24 a SMTP server de Infomed.
#Puerto 25 TCP
iptables -A FORWARD -s 192.168.1.0/24 -p TCP --dport 25 -o $EXTIF -j ACCEPT
```

```
#De subred IP 192.168.1.0/24 a POP3 server de Infomed.
#Puerto 110 TCP
iptables -A FORWARD -s 192.168.1.0/24 -p TCP --dport 110 -o $EXTIF -j ACCEPT
#De 200.55.161.62 a Firewall (INPUT)
#Puerto 22 TCP
#ICMP necesario
iptables -A INPUT -s 200.55.161.62 -p TCP --dport 22 -i $EXTIF -j ACCEPT
iptables -A INPUT -s 200.55.161.62 -p ICMP -i $EXTIF -j ACCEPT
#De 200.55.161.62 a DB-Server
#Puerto 22 TCP (mapeado como 1022 en el Firewall)
#Puerto 1433 TCP (mapeado como 2433 en el firewall)
iptables -A FORWARD -s 200.55.161.62 -d 192.168.2.2 -p TCP --dport 22 -j ACCEPT
iptables -A FORWARD -s 200.55.161.62 -d 192.168.2.2 -p TCP --dport 1433 -j ACCEPT
#Reglas SNAT
iptables -t nat -A POSTROUTING -o $EXTIF -j SNAT --to $EIP
#Reglas DNAT
iptables -t nat -A PREROUTING -p tcp -s $ARSA -d $EIP --dport 2433 -j DNAT --to-
destination 192.168.2.2:1433
```

2.7.2 Hospital Hermanos Ameijeiras.

A esta institución se le propuso una solución de diseño específica para su red, pues con el diseño que tenían no controlaban la navegación de todos sus usuarios y precisamente lo que se quiere es mantener un control absoluto y detallado de la navegación de cada usuario mediante su autenticación, ya sean los que tienen permiso para navegar directamente en internet como los que navegan debajo del proxy de Infomed. También es preciso delimitar la navegación a determinados sitios que no están permitidos y llevar un control de la navegación mediante las trazas que genera el proxy.

Escenario

Diseño que tenía el Hospital Hermanos Ameijeiras (HHA)

Acceso actual a Internet:

- Canal de 1 Mbit/s.
- Línea frame relay.
- Subred de 32 direcciones IP oficiales.
- Router Telindus 1421. configurado con NAT – PAT mapeando hacia IP privadas internas.

Otras características:

- No hay firewall.
- No hay reglas de filtraje en el router, el filtraje corre por cuenta del “extremo de Infomed”, administrado por esta misma red.
- ISA Server local del hospital actuando como proxy directo a Internet con IP: 172.16.1.4 para la subred del hospital y con IP: 201.220.212.98 para conectarse a Internet, con este proxy navegan 142 usuarios del hospital.
- Los 174 usuarios restantes salen directamente por el router y se conectan mediante el servidor de Infomed a donde deseen, por lo que el control de la navegación no lo tiene el ISA Server del hospital sino el servidor de Infomed.

Protocolos por los cuales se accede a la red externa:

- 80……..http (e/s)
- 443……https (s)
- 20,21……ftp (s)
- 25……..smtp (e/s)

Para servidor de correo nada más (IP específica) (hacer NAT con ellas)

- 110……pop3 (s)
- 3128……proxy (s)
- 103,104,105…imágenes (e/s)
- 53……dns (e/s)
- 80…… mrtg (e)

Desde dentro cualquier IP y desde fuera IP específicas (puntuales)

- 80…… sarg (e)

Desde dentro y fuera IP específicas

- 1433……sql (f)

Para acceder al SQL desde Softel

- 22……ssh (e)

Navegación en el hospital:

Requisitos

- Protección especial para el servidor aislándolo del resto de la LAN.
- Un grupo de usuarios de la red local navegan mediante el proxy de Infomed y otro grupo navegan directamente en Internet por el propio proxy de la institución.
- Cada usuario que navegue tiene que autenticarse.
- Todo lo que no esté explícitamente especificado se prohíbe.
- Prohibir la navegación a determinados sitios.
- Llevar un control del toda la navegación.
- Seguridad máxima posible.

Diseño Implantado como solución a sus necesidades

Definiciones:

Se definen tres zonas de seguridad:

1. **1.** Infomed (Zona Internet, Zona insegura o Zona 1)
2. **2.** LAN HHA (Zona segura, zona 2)
3. **3.** Servidores (Zona muy segura, zona 3)

Se realizará la implementación de un equipo perimetral con las siguientes características:

- PC con 2 NICs (sin monitor, ni mouse, ni audio).
- Sistema operativo: Linux Debian-Sarge. Con modo gráfico.
- Software de firewall: Iptables (Netfilter).
- Software de proxy: Squid.
- Una tarjeta de red haciendo interfase con cada zona.

Especificaciones de las zonas:

Zona 1: Infomed

Eth0
Direcciones IP oficiales. Subred para comunicarse con el router. Cross-over al router.
Direcciones IP especificadas por el proveedor (Infomed).

Zona 2: LAN HHA

Eth1
Direcciones IP privadas. Subred IP diferente a la de zona 3. Conexión al switch de la LAN.
Red: 172.16.0.0.
Dirección IP de la interfase del proxy: 172.16.1.2.

Zona 3: Servidores

Eth2

Direcciones IP privadas. Subred IP diferente a la de Zona 2.

Red: 172.16.1.0.

Dirección IP de la interfase del proxy: 172.16.1.3.

Configuración del Iptables:

Todo tráfico negado por defecto.

Dnat de zona 2 a zona 1 para acceder a red externa.

Snat de zona 1 a zona 2 entrante para administrar.

Ruteo entre zona 2 y zona 3 filtrado por reglas.

Paso de conexiones establecidas y relacionadas (statefull filtering)

Conjunto de reglas que satisfagan los requisitos de diseño.

Reglas de objeto a objeto (subredes/IP):

#Puerto 80 (HTTP) entrada-salida
iptables -A FORWARD -s $IIP1 -p TCP -o $EXTIF --dport 80 -j ACCEPT

#Puerto 443 (HTTPS) salida
iptables -A FORWARD -s $IIP1 -p TCP -o $EXTIF --dport 443 -j ACCEPT

#Puertos 20 y 21 (FTP saliente) salida
iptables -A FORWARD -s $IIP1 -p TCP -o $EXTIF --dport 20,21 -j ACCEPT
iptables -A FORWARD -s $IIP1 -p ICMP -o $EXTIF -j ACCEPT

#Puerto 25 (smtp) entrada-salida
iptables -A FORWARD -s $IIP1 -p TCP --dport 25 -o $EXTIF -j ACCEPT

#Puerto 110 (pop3) salida
iptables -A FORWARD -s $IIP1 -p TCP --dport 110 -o $EXTIF -j ACCEPT

#Puertos 3128 (proxy) entrada
iptables -A FORWARD -s $IIP1 -p TCP --dport 3128 -o $EXTIF -j ACCEPT
iptables -A FORWARD -s $IIP1 -p ICMP -o $EXTIF -j ACCEPT

#Puertos 103, 104, 105 (transferencia de imágenes) entrada-salida
iptables -A FORWARD -s $IIP1 -o $EXTIF -j ACCEPT

#Puerto 53 (DNS) salida-forward
iptables -A FORWARD -s $IIP1 -p TCP -o $EXTIF --dport 53 -j ACCEPT
iptables -A FORWARD -s $IIP1 -p UDP -o $EXTIF --dport 53 -j ACCEPT
iptables -A FORWARD -s $IIP1 -p ICMP -o $EXTIF -j ACCEPT

#Puerto 80 (MRTG) entrada
iptables -A FORWARD -s $IIP1 -j ACCEPT

#Puerto 80 (SARG) entrada
iptables -A FORWARD -s $IIP1 -j ACCEPT

#Puerto 1433 (acceso al MS-SQL desde softel) forward
iptables -A FORWARD -s $SOFTEL -p TCP -d $IIP2 --dport 1443 -j ACCEPT
iptables -A FORWARD -s $SOFTEL -p ICMP -d $IIP2 -j ACCEPT

#Puerto 22 (ssh-acceso al firewall desde softel) entrada
iptables -A FORWARD -s $SOFTEL -p TCP -d $IIP2 --dport 22 -j ACCEPT

#Reglas SNAT
iptables -t nat -A POSTROUTING -o $EXTIF -j SNAT --to $EIP

#Reglas DNAT
iptables -t nat -A PREROUTING -p tcp -s $ Infomed -d $EIP -j DNAT --to-destination $IIP2

Configuración del Proxy:

En el montaje del nuevo servidor proxy para el hospital Ameijeiras se decidió levantar dos instancias del Squid, logrando que los dos grupos de usuarios que existen naveguen por el propio proxy de la institución. Por una instancia navegan los 142 usuarios que tienen permiso para acceder directamente a internet y por la otra los 174 que tienen que navegar por debajo del proxy de Infomed, garantizando que estas dos listas de usuarios se autentiquen, no puedan acceder a determinados sitios que estén prohibidos por la institución y logrando el control de la navegación mediante las trazas generadas por el proxy.

Con el diseño anterior no se lograba esto, pues con el ISA Server solo se controlaban los usuarios que navegan por él, los que salían directamente por el router eran controlados por el proxy de Infomed.

2.8 Conclusiones del capítulo.

A lo largo de este capítulo, debido la problemática general que tienen actualmente las instituciones de salud cubanas, se dejó propuesta una secuencia de pasos para adaptar este diseño a cualquier institución, ya sean bancos de sangre, policlínicos u hospitales, definiéndose la solución general del diseño para la seguridad perimetral, que incluye la selección del firewall y el proxy y la integración de estos en un equipo perimetral, además de resolver los problemas de seguridad del hospital Hermanos Ameijeiras y la del banco de sangre de 23 y 2 con un diseño propio para cada institución.

Capítulo 3 Pruebas.

Capítulo 3

3.1 Introducción del capítulo.

En este último capítulo se mostrarán todas las pruebas realizadas en el laboratorio para comprobar la solución del diseño, el análisis de los resultados de estas y por último la prueba que incluye la integración de todos los software en el equipo perimetral.

3.2 Descripción de pruebas de integración de las facilidades propuestas y sus configuraciones.

Estas son pruebas que se hicieron en el laboratorio para demostrar que se cumplen cada una de las facilidades requeridas por las instituciones a las cuales se les proponga el diseño de seguridad perimetral:

- Validar diseños típicos.
- Validar que funcionan los script de Iptables.
- Validar configuraciones del Squid en cascada.
- Autenticación mediante el método SMB.
- Validar que el proxy levante múltiples instancias.
- Visualizar de forma entendible para el usuario las trazas generadas por el proxy.
- Negar la navegación en sitios prohibidos.
- Validar la prueba de integración con la solución general dada al banco de sangre de 23 y 2 y al hospital Hermanos Ameijeiras.

3.2.1 Diseño de cada una de las pruebas de validación de los diseños que se hicieron.

Todas estas pruebas fueron hechas en PCs de laboratorio con una tarjeta de red simulando dos interfases de red.

3.2.1.1 Pruebas de firewall.

Prueba 1

Objetivo: Filtrar todos los puertos que estén levantados en un servidor, menos uno en específico.

Consiste en: Validar que cuando una PC cliente que se encuentre en una red externa use el Nmap para mapear los puertos de un servidor de la red protegida por el firewall (script de Iptables con sus reglas de filtraje), ésta solo pueda ver el puerto 80 y que los demás aunque estén abiertos no se vean.

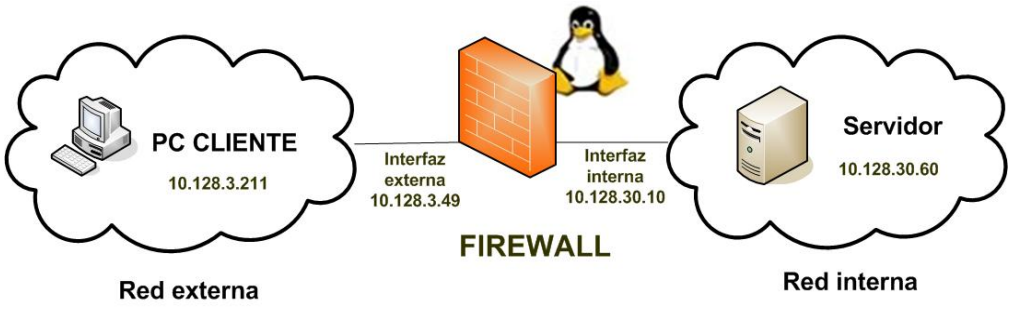

Figura10: Prueba 1 de firewall.

Configuración de cada PC en particular:

PC Cliente

Tener instalado el WinPcap (Nmap) para poder mapear los puertos del servidor para ver los que están abiertos.

Address	10.128.3.211

Netmask	255.255.255.0
Network	10.128.3.0
Broadcast	10.128.3.255
Gateway	10.128.3.49

Firewall

Tener instalado el Iptables y corriendo con permiso de ejecución el fichero del script.

	Eth0(Interfaz externa)	Eth0:1(Interfaz interna)
Address	10.128.3.49	10.128.30.10
Netmask	255.255.255.0	255.255.255.0
Network	10.128.3.0	10.128.30.0
Broadcast	10.128.3.255	10.128.30.255
Gateway	10.128.3.254	

Servidor

Address	10.128.30.60
Netmask	255.255.255.0
Network	10.128.30.0
Broadcast	10.128.30.255
Gateway	10.128.30.10

Script de iptables

```
#!/bin/sh
echo Prueba de Firewall-Script de Iptables.
echo IP Forwarding.
echo 1 > /proc/sys/net/ipv4/ip_forward
# sustituir o agregar
# net/ipv4/ip_forward = 1
# en el file /etc/sysctl.conf
echo Limpieza de reglas previas.
iptables -F
iptables -X
```

```
iptables -Z
iptables -t nat -F
echo Reglas por defecto.
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
echo Permitido todo lo establecido y relacionado.
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
echo Regla Forward.
iptables -A FORWARD -d 10.128.30.60 -p TCP --dport 80 -j ACCEPT
iptables -A FORWARD -d 10.128.30.60 -p ICMP -j ACCEPT
```

Prueba 2

Objetivo: Filtrar todos los puertos que estén levantados en un servidor, menos uno en específico y poder dar ping al mismo servidor pero que todos los puertos estén filtrados.

Consiste en: Validar que cuando un cliente que esté en una red externa trate de dar Nmap al servidor que se encuentra en la red protegida, por las reglas de filtraje del script de Iptables, sólo pueda ver el puerto 80 y que cuando otro cliente en la misma red se conecte al este servidor no pueda ver ninguno, que todos estén filtrados, pero que sí se le permita dar ping al servidor.

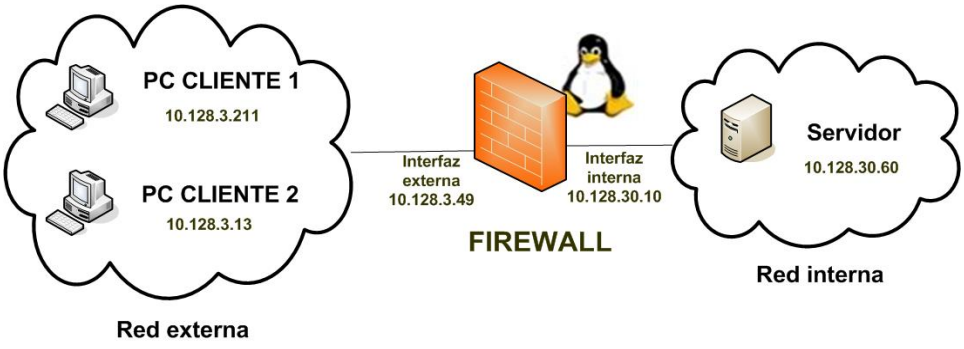

Figura11: Prueba 2 de firewall.

PC Cliente 2

Address	10.128.3.13
Netmask	255.255.255.0
Network	10.128.3.0
Broadcast	10.128.3.255
Gateway	10.128.3.49

Script de iptables

```
#!/bin/sh
echo Prueba2 de Firewall-Script de Iptables.
echo IP Forwarding.
echo 1 > /proc/sys/net/ipv4/ip_forward
# sustituir o agregar
# net/ipv4/ip_forward = 1
# en el file /etc/sysctl.conf
echo Limpieza de reglas previas.
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
echo Reglas por defecto.
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
echo Permitido todo lo establecido y relacionado.
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
echo Reglas Forward para Cliente 1.
iptables -A FORWARD -s 10.128.3.211 -d 10.128.30.60 -p TCP --dport 80 -j ACCEPT
iptables -A FORWARD -s 10.128.3.211 -d 10.128.30.60 -p ICMP -j ACCEPT
echo Reglas Forward para Cliente 2.
iptables -A FORWARD -s 10.128.3.13 -d 10.128.30.60 -p ICMP -j ACCEPT
```

Prueba 3

Objetivo: Comprobar script con reglas de transformaciones NAT.

Consiste en: Validar que la regla SNAT (IP fuente del paquete) haga que un paquete que salga de una red interna (con IP falsas) hacia una red externa (con IP verdaderas) cuando vire no sepa el IP privado, que llegue al firewall y éste se encargue de hacerlo llegar al servidor interno.

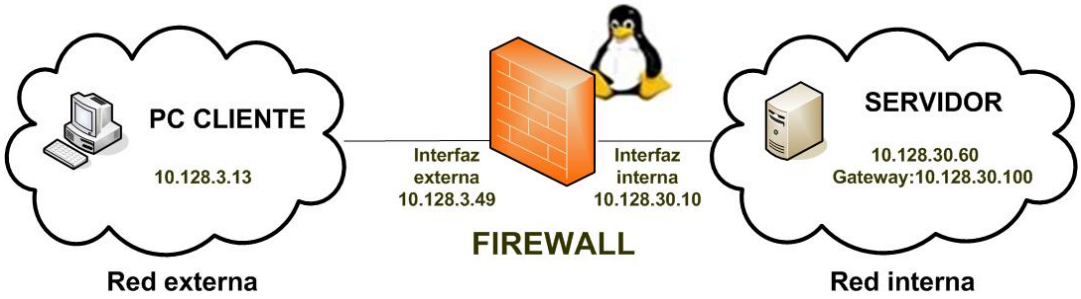

Figura12: Prueba 3 de firewall.

Script de iptables

```
#!/bin/sh
echo Prueba3 de Firewall-Script de Iptables.
echo IP Forwarding.
echo 1 > /proc/sys/net/ipv4/ip_forward
# sustituir o agregar
# net/ipv4/ip_forward = 1
# en el file /etc/sysctl.conf
echo Limpieza de reglas previas.
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
echo Reglas por defecto.
iptables -P INPUT DROP
```

```
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
echo Permitido todo lo establecido y relacionado.
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
echo Regla Forward.
iptables -A FORWARD -d 10.128.30.60 -p TCP --dport 80 -j ACCEPT
iptables -A FORWARD -d 10.128.30.60 -p ICMP -j ACCEPT
echo Transformaciones NAT
iptables -t nat -A POSTROUTING -d 10.128.30.60 -j SNAT --to 10.128.30.10
```

3.2.1.2 Pruebas de proxy.

Prueba 1

Objetivo: Verificar facilidades del proxy como: la autenticación contra un servidor de dominio y la no navegación a sitios prohibidos.

Consiste en:

1-Validar que cuando un usuario solicite un recurso que se encuentra en una red externa, el proxy le pida usuario y contraseña, verifica si esta en la lista de usuarios del servidor de dominio, si pertenece a esa lista el proxy obtiene el pedido y se lo hace llegar al cliente, sino le deniega el acceso.

2- Validar que cuando un usuario solicite un recurso que no está permitido, el proxy le deniegue el acceso.

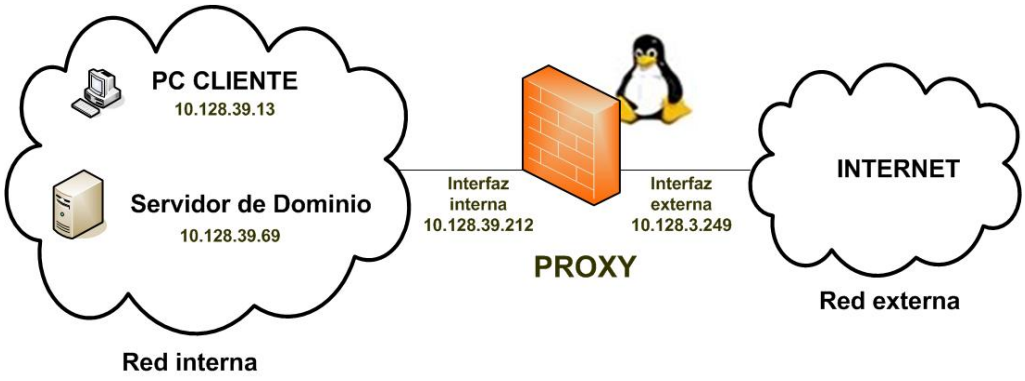

Figura13: Prueba 1 de proxy.

PC Cliente

Configurar el navegador Internet Explorer con el IP del proxy, el puerto y las direcciones específicas para las cuales no se tiene que usar el proxy.

Address	10.128.39.13
Netmask	255.255.255.0
Gateway	10.128.39.254
DNS	10.0.0.3 y 10.0.0.4

Servidor de Dominio

- Crear una lista de usuarios que pertenezcan a ese dominio.
- Crear un archivo llamado "proxyauth" en el recurso compartido NETLOGON.
- Colocar sólo la palabra "allow" en este archivo.
- Asignar acceso de "lectura" al archivo "proxyauth" para todos los usuarios o grupo que se desee permitir acceder al proxy.

Address	10.128.39.69
Netmask	255.255.255.0
Gateway	10.128.39.254
DNS	10.0.0.3 y 10.0.0.4

Proxy

- Sistema operativo GNU/Linux con Debian Sarge 3.1
- Software para el proxy Squid -2.5 Stable 9
- Software para lograr la autenticación smb_auth-0.05
- smb_auth necesita Samba para utilizar SMB. Instalar Samba con samba-common, smbclient.
- Simular dos interfases:

	Eth0(cara para la subred)	Eth:1(cara para internet)
Address	10.128.39.212	10.128.3.249
Netmask	255.255.255.0	255.255.255.0
Network	10.128.39.0	10.128.3.0
Broadcast	10.128.39.255	10.128.3.255
Gateway		10.128.3.254

Configurando Squid como Proxy convencional con parámetros básicos y logrando dos facilidades, autenticación y sitios prohibidos.

Como ejemplo están las líneas relevantes del propio archivo squid.conf ya diseñado:

Parámetro http_port:

http_port 3128

Este puerto porque Squid lo utiliza por defecto para atender peticiones, sin embargo se puede especificar que lo haga por cualquier otro.

Parámetro cache_mem:

cache_mem 16 MB

Aunque por defecto se establecen 8 MB, puede especificarse una cantidad mayor si así se considera necesario, por lo que si posee un servidor con al menos 128 MB de RAM, establezca 16 MB como valor para este parámetro.

Parámetro cache_dir:

cache_dir ufs /var/spool/squid 700 16 256

Se utiliza para establecer el tamaño que se desee que tenga la caché de Squid en el disco duro. Se puede incrementar el tamaño de la caché hasta donde se desee. Mientras más grande, más objetos se almacenarán en esta y por lo tanto se utilizará menos ancho de banda.

Los números *16* y *256* significan que el directorio de la caché contendrá 16 subdirectorios con 256 niveles cada uno. No es necesario modificar estos números.

Si se especifica un determinado tamaño de caché y este excede al espacio real disponible en el disco duro, Squid se bloqueará inevitablemente.

Lugar del disco duro donde Squid guardará los log:

cache_access_log /var/log/squid/access.log

cache_log /var/log/squid/cache.log

Programa a utilizar para lograr la autenticación:

auth_param basic program /usr/local/bin/smb_auth -W LIDY (específico el programa)

Es necesario establecer listas de control de acceso que definan la autenticación para una determinada lista de usuarios y para otra lista con sitios prohibidos. A cada lista se le asignará una regla de control de acceso que permitirá o denegará el acceso a Squid.

Para esto:

1- Se crean dos ficheros en cualquier parte del disco duro:

touch /etc/squid/permitidos #estos son los usuarios con permiso de navegar en Internet
touch /etc/squid/sitiosdenegados #aquí esta el listado de sitios prohibidos

2- Ha estos dos fichero se les debe dar permiso de lectura y escritura

chmod 600/etc/Squid/permitidos
chmod 600/etc/Squid/sitiosdenegados

Lo que tiene permitidos:

La lista de usuarios que pertenecen al dominio de la institución que tienen permisos:

Ejemplo de “permitidos”

Lidy
dalay

Sitios denegados:

Ejemplo de “sitiosdenegados”

www.sitioporno.com
www.otrositioporno.com
sitioindeseable.com
otrositioindeseable.com
sex
porn
mp3
xxx
adult

3- En el Squid.conf

```
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

# Example rule allowing access from your local networks. Adapt
# to list your (internal) IP networks from where browsing should
# be allowed
#acl our_networks src 192.168.1.0/24 192.168.2.0/24
#http_access allow our_networks
http_access allow localhost

acl autenticados proxy_auth REQUIRED
acl userspermitidos proxy_auth "/etc/squid/permitidos"
acl sitiosdenegados url_regex "/etc/squid/sitiosdenegados"
http_access allow autenticados userspermitidos !sitiosdenegados

# And finally deny all other access to this proxy
http_access deny all
```

Parámetro cache_mgr:

cache_mgr lminguez@estudiantes.uci.cu

Para si algo ocurre con la caché, como por ejemplo que mueran los procesos, se enviará un mensaje de aviso a esta cuenta de correo.

Usuario y grupo con el que se ejecutará Squid:

Se puede usar el usuario que se quiera. En este trabajo se utilizó el que nos monta Debian por defecto:

cache_effective_user proxy
cache_effective_group proxy

HTTPD-ACCELERATOR OPTIONS (Para acelerar la caché):

Opciones para proxy convencional:

httpd_accel_host virtual
httpd_accel_port 0
httpd_accel_with_proxy on

Prueba 2

Objetivo: Comprobar la capacidad que tiene un servidor proxy de navegar en cascada con otro proxy.

Consiste en: Validar que un usuario para obtener un recurso que se encuentra en una red externa tiene que conectarse primeramente al servidor propio de la red interna, autenticarse y luego éste le delega el pedido al proxy que se encuentra en otra red y finalmente éste es quien realmente obtiene el recurso de la red externa, para luego a través del proxy en cascada hacérselo llegar al cliente.

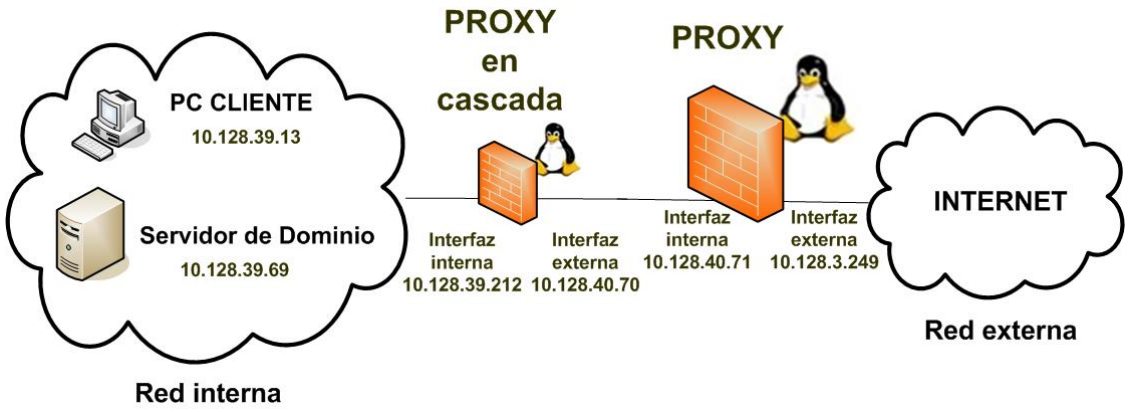

Figura14: Prueba 2 de proxy.

Para esto se configuraron cuatro máquinas:

-El cliente y el servidor de dominio mantienen la misma configuración.

-La configuración del proxy1 mantiene su configuración solo varían tres de sus parámetros dentro del squid.conf:

Puerto para atender peticiones:

http_port 8080

En él se le especifica cual es el servidor padre, es decir debajo de cuál servidor va a trabajar, los puertos por los que escucha el padre tanto el HTTP como el ICP y además especificándole la opción de que no guarde en caché los objetos traídos, sino que sólo atienda los pedidos.

cache_peer 10.128.30.86 parent 8080 3130 proxy-only

Es importante descomentar esta directiva para que el proxy navegue en cascada, de no ser así al parecer hace cascada pero en realidad se conecta directo a internet.

never_direct allow all

No se le define una cache_dir porque este no guarda en caché los objetos traídos solo atiende los pedidos.

No se especifica ni el parámetro cache_mgr, ni el HTTPD-ACCELERATOR OPTIONS (Para acelerar la cache).

Además tener en cuenta que las interfases también varían:

	Eth0(cara para la subred)	Eth:1(cara para el proxy padre)
Address	10.128.39.212	10.128.40.70
Netmask	255.255.255.0	255.255.255.0
Network	10.128.39.0	10.128.40.0
Broadcast	10.128.39.255	10.128.40.255
Gateway		10.128.40.254

Cuando se navega a través del Squid1:

El usuario hace la petición de un recurso de Internet especificado por una URL, el squid1 le pide autenticación, verifica si es un usuario de la lista de los usuarios restringidos, si lo es le hace el pedido al squid2, el squid2 verifica si el pedido viene del IP que tiene en su fichero de permitir por IP, si es así verifica si tiene el recurso en la caché si lo tiene se lo envía rápidamente al squid1 y este se lo hace llegar al cliente inmediatamente, sino lo tiene o no está actualizado en la caché se conecta a Internet y se lo hace llegar al squid1 para que este responda el pedido.

Si el usuario no es de la lista restringida que tiene el squid1, este ni siquiera se conecta al otro servidor (squid2), inmediatamente le deniega el acceso.

Configuración del proxy2:

Este servidor proxy no se encontrará en la institución, así que su configuración será la que el administrador del mismo le asigne, configurando los parámetros básicos será necesarios, pero eso sí, tiene que tener una ACL donde se le especifique quién es el servidor hijo y con su respectiva regla de acceso.

- Puerto por el que escucha las peticiones.
- Memoria para los objetos en tránsito, los hot y los negativamente guardados en caché.

- Tamaño de la caché en el disco duro para Squid.
- Parámetro para saber donde Squid guarda los log en el disco duro.
- Una ACL para especificar el IP del proxy que tiene que navegar a través de él.
- acl permitirip src "/etc/squid/permitidos", donde permitidos tendrá el IP del servidor hijo, o de los servidores porque pueden ser más de uno.
- La regla de control de acceso aplicada a la ACL para permitirla.
- http_access allow permitirip
- Parámetro para si algo ocurre con la caché, se envíe un mensaje de aviso a la cuenta que se especifique.
- Especificar el usuario y el grupo con el que se ejecutará Squid.
- Por último las líneas pertinentes que aceleran la caché para un proxy convencional.

Sus interfases también varían:

	Eth0(cara para el proxy hijo)	Eth0:1(cara para internet)
Address	10.128.40.71	10.128.3.54
Netmask	255.255.255.0	255.255.255.0
Network	10.128.40.0	10.128.3.0
Broadcast	10.128.40.255	10.128.3.255
Gateway		10.128.3.254

Cuando se navega a través del Squid2:

El squid1 envía el pedido del recurso solicitado al squid2, este verifica si el IP que tiene en su fichero permitir por IP coincide con el IP del cual viene el pedido, si coincide, busca en la caché y se lo envía, sino se conecta a Internet, localiza el recurso, se lo envía al squid1 y hace una copia en la caché. Si no coincide el IP le deniega el servicio.

Prueba 3

Objetivo: Comprobar que en una misma PC se pueden levantar dos instancias del proxy con configuraciones diferentes.

Consiste en: Validar que dos listas diferentes de usuarios logren obtener recursos de una red externa por vías diferentes, ambas primeramente se autentican para luego navegar por diferentes instancias: una lista de usuarios lo hace por una instancia del proxy que se conecta directamente a la red externa o Internet y la otra lista de usuarios navega por la otra instancia que se conecta en cascada con otro proxy que está fuera de la red para navegar en Internet.

Proxy con doble instancia. Descripción del escenario de la doble instancia que recoge las dos pruebas anteriores.

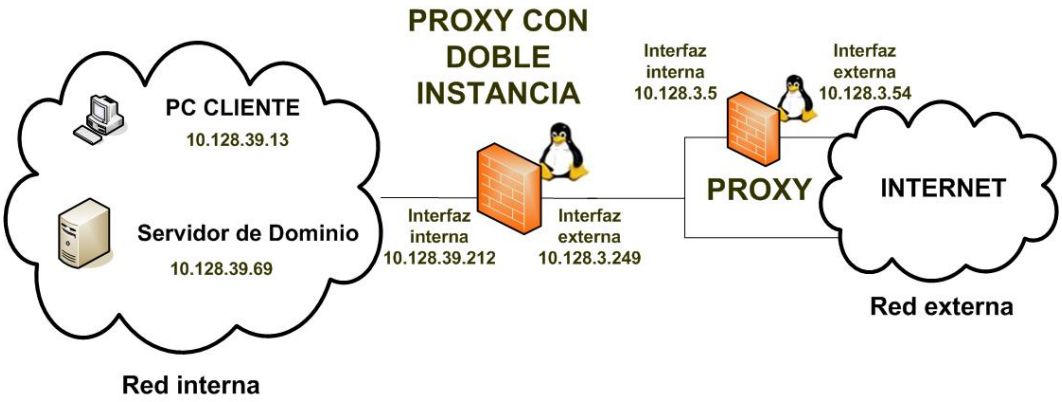

Figura15: Prueba 3 de proxy.

Una red LAN con una PC servidor proxy caché, con dos instancias del Squid levantadas, (squid1 y squid2), donde squid1 va a tener salida directa a Internet mientras que squid2 tiene que conectarse a otro servidor proxy caché (squid3) para poder navegar en Internet.

Todas las peticiones que salen desde la LAN pasan por las dos instancias montadas en el servidor en dependencia del usuario que haga el pedido, ya que unos usuarios navegarán a través del squid1 directamente a Internet y otros navegarán a través del squid2, el cual se conectará en cascada al squid3 para poder navegar en Internet.

- Estas dos instancias se levantan por medio de dos script de booteo.
- Las configuraciones de los ficheros squid.conf de ambas instancias son diferentes:

La instancia 1 tendrá una configuración del fichero squid.conf tal como la del proxy de la primera prueba, que funciona directo a Internet pidiendo autenticación y restringiendo el acceso a sitios prohibidos.

Cambiándole sólo parámetros tales como:

- Parámetro para saber donde Squid guarda los log en el disco duro, pues cada Squid tendrá ficheros diferentes.

cache_access_log /var/log/squid/squid_1-access.log
cache_log /var/log/squid/squid_1-cache.log

- Definir la ubicación del archivo squid_1.pid
pid_filename /var/run/squid_1.pid

La instancia 2 tendrá una configuración del fichero squid.conf tal como la del proxy de la segunda prueba que funciona en cascada con otro proxy.

Cambiándole sólo parámetros tales como:

- Parámetro para saber donde Squid guarda los log en el disco duro, que no es el mismo fichero donde los guarda el squid_1.

cache_access_log /var/log/squid/squid_2-access.log
cache_log /var/log/squid/squid_2-cache.log

- Definir la ubicación del archivo squid_2.pid
pid_filename /var/run/squid_2.pid

- Además de que los pedidos que pasan por esta instancia son realizados por usuarios diferentes a los que hacen los pedidos a la instancia1.

En el proxy con las dos instancias se simularán dos interfases:

	Eth0(cara para la subred)	Eth0:1(cara para el proxy padre y para Internet)
Address	10.128.39.212	10.128.3.249
Netmask	255.255.255.0	255.255.255.0
Network	10.128.39.0	10.128.3.0
Broadcast	10.128.39.255	10.128.3.255
Gateway		10.128.3.254

Prueba 4

Objetivo: Visualizar las trazas generadas por el proxy, de forma entendible para el administrador del servidor y así tener un control de los sitios que son visitados por los usuarios.

Consiste en: Validar que cuando el usuario navegue a través del proxy los log que genere éste, se muestren de forma entendible en formato HTML.

El SARG es la herramienta seleccionada para este trabajo, es utilizada en las tres pruebas mencionadas anteriormente.

Para su uso eficiente los pasos a seguir son:

1. La instalación de la herramienta como otro software cualquiera.
2. La configuración del fichero sarg.conf, modificando los parámetros siguientes:

Parámetro para darle la ubicación del fichero del cual debe obtener los log:
access_log /var/log/squid/squid-access.log

Fichero en el cual debe mostrar los log:
output_dir /var/www/squid-reports

Y los demás parámetros que vienen sin comentarios en el fichero de configuración.

3. Utilizar y modificar acorde a las necesidades del administrador el fichero crontabs para que el sitio donde se muestran los logs refresque diariamente.

3.2.2 Análisis de los resultados de las pruebas.

Las pruebas realizadas en el laboratorio ya explicadas detalladamente en los epígrafes anteriores, muestran que los objetivos trazados para desarrollar este trabajo de diploma han sido cumplidos.

Los resultados de estas pruebas han mostrado que para cualquier escenario se puede montar un firewall con reglas adaptadas a las necesidades de cada institución garantizando la seguridad requerida. Con la programación de los scripts de iptables para los tres tipos de escenarios vistos en las pruebas de firewall, se comprobaron las reglas de filtrado con las que se decide si una conexión determinada puede establecerse o no, además de decidir si un paquete pasa, se modifica, se convierte o se descarta.

Por cada escenario también se hicieron pruebas con la solución de proxy, donde se comprobaron múltiples funcionalidades de este, tales como: la autenticación de cada usuario para permitirle el acceso a determinado recurso de la red, la negación a determinadas URLs que no están permitidas, la navegación del proxy en cascada con otro proxy, levantamiento de múltiples instancias garantizando que en una misma PC se encuentren dos proxy funcionando con configuraciones diferentes y el analizador de los reportes que genera el proxy para llevar un control de los recursos solicitados por el usuario.

Después de haber realizado un análisis de todas las pruebas, se puede concluir que los resultados fueron satisfactoriamente los esperados.

3.3 Diseño de pruebas de implantación.

Prueba con el equipo perimetral ya funcionando con la integración de todos los softwares que se requieren: el firewall con sus reglas de filtraje, el proxy con sus funcionalidades básicas y avanzadas, la herramienta para analizar los reportes que genera cada proxy (ya que en el equipo perimetral se encuentran levantadas dos instancias) y por último el servidor web para visualizar las trazas en formato HTML.

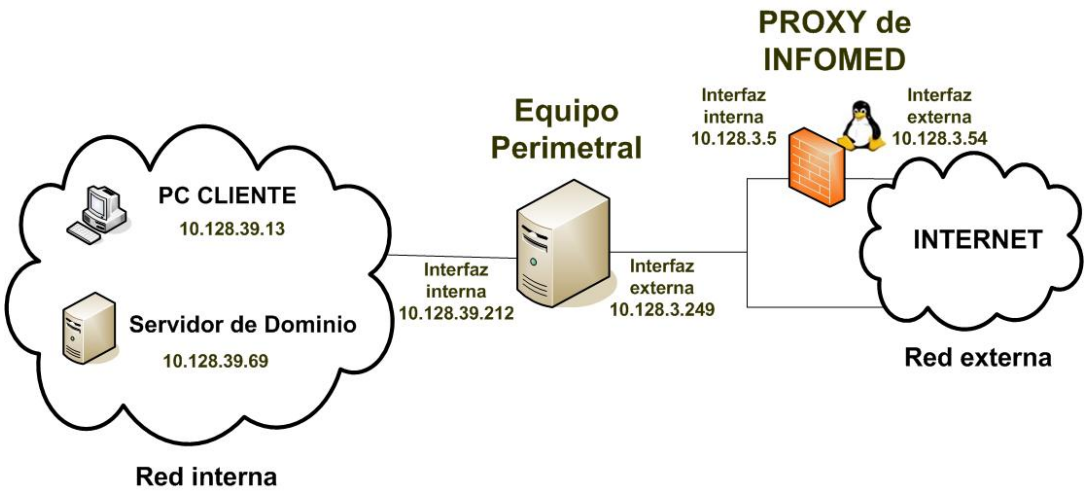

Figura16: Prueba de integración.

En esta prueba se mantiene la configuración tanto de la PC cliente como la del servidor de dominio vista en la prueba1 de proxy.

Montaje del equipo perimetral con los softwares necesarios:

- Sistema Operativo: Linux (con ambiente gráfico).
- Distribución: Debian (Sarge 2.6.8).
- Softwares:
  - Firewall: Iptables (filtrado de paquetes IP con status + nat).
  - Proxy: Squid 2.5.
  - Analizador de reportes de logs: Sarg-2.2.3.1.
  - Autenticación por dominio: Smb_auth-0.05 y Samba.

3.4 Conclusiones del capítulo.

Con este capítulo de pruebas se logró comprobar el buen funcionamiento de todas las facilidades que se necesitaban cumplir para la solución que se le ofreció al banco de sangre de 23 y 2 y al hospital Hermanos Almeijeiras que incluye la solución integradora con firewall y proxy.

Conclusiones

Se desarrolló una solución de firewall más proxy para la seguridad perimetral de las redes de las instituciones de salud cubanas, basada en software libre, equipamiento relativamente barato y adaptado a las características propias de cada institución.

Se dejó propuesta una secuencia de pasos para ser utilizados a la hora de realizar el diseño de seguridad perimetral específico de cualquier institución de salud. Se definió la solución de diseño general, que incluye la selección del firewall y el proxy y la integración de estos en un equipo perimetral, para resolver los problemas de seguridad del hospital Hermanos Ameijeiras y del banco de sangre de 23 y 2.

Una vez terminado este trabajo de diseño de seguridad perimetral, se considera que se ha confirmado la hipótesis planteada y que se han cumplido todos los objetivos propuestos.

Recomendaciones

Se desea hacer varias recomendaciones o sugerencias para este trabajo, puesto que existen otras vulnerabilidades que, aunque no fueron mencionadas anteriormente por no ser objetivo de este trabajo se quisieran referenciar.

- Instalar y utilizar programas detectores de intrusos.
- Enlazar el firewall dinámicamente con detectores de intrusos.
- Ponerle detectores de antivirus al proxy.
- Desarrollar más las facilidades que se brinden en el equipo mixto (firewall más proxy) de VPN.
- Instalar algún software de autenticación para firewall, tal y como la nueva tecnología NuFW (Now User Filtering Works) que es una extensión del Netfilter.
- Implementar un sistema analizador de trazas para iptables.

Bibliografía

1. Marcus Goncalves, Manual de firewall, 970-10-3536-4, Septiembre 2001.

2. Seguridad Informática-Firewall, http://www.segu-info.com.ar, Octubre 2005.

3. Comunicación de Datos. Seguridad, CC51C.

4. Pello Xabier Altadill Izura, Proxy Cache. Guías rápidas de sistemas, 2003.

5. Ventajas y Desventajas del Proxy y Proxy Caché de web, http://es.wikipedia.org/wiki/Proxy, 2006.

6. Juan Carlos Pujol García, Selección y configuración de firewalls, Softel, 2006.

7. Tipos de Firewall, http://descargas.orange.es/descargas/buscador.php, 2007.

8. Lista de firewall, http://www.elistas.net/lista/infohackers, 2007.

9. Instalación de un firewall, http://www.ceyusa.com/documentos, 2005.

10. Ingeniería de redes, http://www.interlan.com.co, 2007.

11. Los proxy trabajan en dos capas del modelo OSI, http://es.wikipedia.org/wiki/Modelo_OSI, 2004.

12. Ejemplos de Proxy, http://descargas.orange.es/descargas/programas.

13. Proxy WebCleaner, http://software.elpais.com/ie/38424-WebCleaner.

14. Squid: Ventajas y desventajas que admite, http://www.squid-cache.org.

15. Microsoft Corporation, ¿Qué es ISA-Server?, http://www.microsoft.com/spain/isaserver.

16. Conferencia de Firewall. Sistema de prevención de incidentes de seguridad, 2007.

17. Equipos para el aseguramiento de redes perimetrales, http://www.activar.com.mx/seguridad_perimetral.htm, enero-2007.

18. Equipos de seguridad, http://www.directindustry.es/fabricante-industrial, 2007.

19. Diferentes tipos de intrusiones o ataques, http://www.terra.es/tecnologia/seguridad, abril-2007.

20. Delitos Informáticos. Clasificación y tipos de ataques contra sistemas de información, http://www.delitosinformaticos.com, febrero-2007.

21. Tipos de ataques en aplicaciones web, http://climbo.wordpress.com, 2007.

22. Hackers_Criminales informáticos - TIPOS DE ATAQUES, http://www.monografias.com/trabajos23/hackers, febrero-2007.

23. Microsoft Corporation, Guía de operaciones de seguridad, http://www.microsoft.com/spain/technet/seguridad, marzo-2007.

24. Juan Carlos Pujol, Abel Llerena y Dayrel Almaguer, Seguridad y eficiencia en comunicaciones informáticas en pymes cubanas (5ta Semana Tecnológica de Fordes-MIC), abril-2007.

Anexos

Manual técnico para usuarios con la integración de todos los software en el equipo perimetral.

1. **Configuración básica del proxy Squid con las facilidades de autenticación y listas negras:**

Instalando Squid:

La PC donde se montó el proxy tiene instalado Debian Sarge 3.1, con un repositorio actualizado del cual se puede instalar fácilmente la versión Squid 2.5 (viene incluido en casi todas las distribuciones actuales), apt - get install squid.

Si no se tiene un repositorio actualizado, se baja de Internet el fichero fuente del Squid y se guarda en squid en /usr/local/src. Descomprimimos con tar –zxvf Squid.tar.gz y se compila Squid con: make y make install.

Configurando Squid:

Squid se configura desde un único archivo llamado Squid.conf que se encuentra en /etc/Squid/Squid.conf, aquí todos los parámetros están muy bien explicados.

Se debe evitar dejar espacios vacíos en lugares indebidos. El siguiente es un ejemplo de como se debe descomentar un parámetro:

Opción correctamente descomentada

http_access 3128

Existen un gran número de parámetros, de los cuales recomendamos configurar los siguientes:

- http_port
- cache_mem
- cache_dir
- cache_access_log
- cache_log
- Especificar el programa para lograr autenticación

- Una lista de control de acceso (ACL) para pedir que el usuario se autentique.
- Una lista de control de acceso para especificar un fichero donde estarán los usuarios con permiso para navegar en internet porque no todos van a navegar.
- Una lista de control de acceso para especificar un fichero donde estarán una serie de sitios prohibidos que el usuario no podrá acceder.
- Las reglas de control de acceso (http_access) requeridas para cada ACL donde se permite o se deniega el acceso a squid.
- cache_mgr
- cache_effective_user
- cache_effective_group
- httpd_accel_host
- httpd_accel_port
- httpd_accel_with_proxy

Parámetro http_port:

¿Que puerto utilizar para Squid? Squid por defecto utilizará el puerto 3128 para atender peticiones, sin embargo se puede especificar que lo haga en cualquier otro puerto o bien que lo haga en varios puertos a la vez, se puede especificar que Squid escuche peticiones por el puerto 8080.

Siendo así se localiza la sección de definición de http_port y se especifica:

Default: http_port 3128

http_port 3128

http_port 8080

Parámetro cache_mem:

El parámetro cache_mem establece la cantidad ideal de memoria para lo siguiente: objetos en tránsito, objetos Hot y objetos negativamente almacenados en la caché.

Los datos de estos objetos se almacenan en bloques de 4 Kb. El parámetro cache_mem especifica un límite máximo en el tamaño total de bloques acomodados, donde los objetos en tránsito tienen mayor prioridad. Sin embargo los objetos Hot y los negativamente almacenados en la caché podrán utilizar la memoria no utilizada hasta que esta sea

requerida. De ser necesario, si un objeto en tránsito es mayor a la cantidad de memoria especificada, Squid excederá lo que sea necesario para satisfacer la petición.
Por defecto se establecen 8 MB. Puede especificarse una cantidad mayor si así se considera necesario, dependiendo esto de los hábitos de los usuarios o necesidades establecidas por el administrador. Si se posee un servidor con al menos 128 MB de RAM, establezca 16 MB como valor para este parámetro:
cache_mem 16 MB

Parámetro cache_dir:

¿Cuánto desea almacenar de Internet en el disco duro? Este parámetro se utiliza para establecer que tamaño se desea que tenga la caché en el disco duro para Squid. Por defecto Squid utilizará una caché de 100 MB, de modo tal que encontrará la siguiente línea:
cache_dir ufs /var/spool/squid 100 16 256
Se puede incrementar el tamaño de la caché hasta donde se desee. Mientras más grande la caché, más objetos se almacenarán en éste y por lo tanto se utilizará menos el ancho de banda. La siguiente línea establece una caché de 700 MB:
cache_dir ufs /var/spool/squid 700 16 256
Los números 16 y 256 significan que el directorio de la caché contendrá 16 subdirectorios con 256 niveles cada uno. No se modifican estos números, no hay necesidad de hacerlo. Es muy importante considerar que si se especifica un determinado tamaño de caché y este excede al espacio real disponible en el disco duro, Squid se bloqueará inevitablemente. Se debe ser cauteloso con el tamaño de caché especificado.

Parámetros para guardar los log que se genera el Squid en el disco duro:

Por defecto Squid los escribe en tres lugares que se definen en estas líneas:
cache_log /var/log/squid/cache.log
cache_access_log /var/log/squid/access.log
cache_store_log /var/log/squid/store.log

Controles de acceso:

Es necesario establecer Listas de Control de Acceso que definan una red o bien ciertas PC en particular. A cada lista se le asignará una regla que permitirá o denegará el acceso a Squid.

Listas de control de acceso:

Regularmente una lista de control de acceso se establece siguiendo la siguiente sintaxis:
acl [nombre de la lista] tipo [lo que compone a la lista]
tipo: (src**, url_regex**, **proxy_auth**) depende de la ACL que quieras declarar
Lo que compone la lista puede ser un número IP, un fichero, una URL, etc.

Reglas de Control de Acceso:

Estas definen si se permite o no el acceso a Squid. Se aplican a las Listas de Control de Acceso. Deben colocarse en la sección de reglas de control de acceso definidas por el administrador, es decir, a partir de donde se localiza la siguiente leyenda:
INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
La sintaxis básica es la siguiente:
http_access [deny o allow] [lista de control de acceso]
deny para denegar todo y allow para permitir todo

Para especificar el programa y lograr autenticación con las listas (ACL) y las reglas de control de acceso (http_access) requeridas:

Autenticación con el Squid:

Es muy útil poder establecer un sistema de autenticación para acceder hacia Internet. Esto permite controlar quienes accederán o no, sin importar desde que PC de la red local lo hagan. Squid permite autenticación con nombre de usuario y contraseña por vía de una aplicación externa, usando la ACL proxy_auth y authenticate_program; se fuerza a un cliente a verificar nombre de usuario y contraseña antes de que obtenga acceso a Internet.
Hay varios programas de autenticación disponibles que Squid puede usar:

- LDAP : Usa Linux Lightweight Directory Access Protocol
- NCSA : Usa un archivo estilo NCSA con username y password

- SMB : Usa el server SMB server como SAMBA Windows NT
- MSNT : Usa la autenticación de dominio de Windows NT
- PAM : Usa Linux Pluggable Authentication Modules
- getpwam: Usa el archivo de contraseñas de Linux.

Programa para la autenticación:

El programa seleccionado es el SMB, para este método Squid utiliza el módulo smb_auth que autentica los usuarios del proxy contra un servidor SMB como Windows NT ó Samba.

Requerimientos para este programa:

- Squid 2.x
- smb_auth-0.05 (se descarga de: http://www.hacom.nl/~richard/software/smb_auth.html)
- smb_auth necesita Samba para utilizar SMB. Instalar Samba con samba-common, smbclient.

Instalación del smb_auth:

- Se chequea el Makefile. Asegurarse que SAMBAPREFIX e INSTALLBIN estén seteados correctamente antes de ejecutar make.
- Ejecutar "make", después "make install". Esto instalará smb_auth y smb_auth.sh en el directorio INSTALLBIN.

Configuración del controlador primario de dominio:

Para tener control de acceso al proxy por usuario y grupo, smb_auth lee el archivo \netlogon\proxyauth en uno de los controladores de dominio usando las credenciales suministradas. Si al leer el archivo devuelve "allow" entonces se permitirá el acceso, en caso contrario el acceso será denegado.

- Crear un archivo llamado "proxyauth" en el recurso compartido NETLOGON del controlador primario de dominio. En caso de tener uno ó más controladores de de dominio, se supone que replicará este recurso compartido a los controladores del dominio. Si se prefiere, se puede cambiar la ubicación de este archivo usando la opción -S de smb_auth.

- Colocar sólo la palabra "allow" en este archivo.
- Asignar acceso de "Lectura" al archivo "proxyauth" para todos los usuarios o grupo que se desee permitir acceder al proxy.
- Si se desea permitir el acceso desde múltiples dominios al proxy, repita los pasos de arriba para los otros dominios.

Configurando Squid:

Como ejemplo, están las líneas relevantes del propio archivo squid.conf:

auth_param basic program /usr/local/bin/smb_auth -W LIDY (se especifica el programa)

acl autenticados proxy_auth REQUIRED (se pide autenticación)

http_access allow autenticados

Si realmente se desea especificar la dirección IP de un controlador de dominio se utiliza:

smb_auth -W nombredominio -U <dirección IP>

Solucionando problemas:

Se puede ejecutar smb_auth desde la línea de comandos usando las mismas opciones que en su squid.conf. Para depurar la autentificación se puede adicionalmente usar la opción -d que mostrará información después de cada paso, así se puede determinar que paso está fallando.

No usar la opción -d en el squid.conf, esto corrompe la comunicación entre Squid y smb_auth.

Se necesita introducir un nombre-de-usuario y clave (separados por el carácter espacio) a la entrada estándar de smb_auth. Luego de autentificar este nombre-de-usuario y clave, smb_auth continuará aceptando esta combinación nombre-de-usuario/clave hasta que se cierre la entrada estándar presionando Ctrl-C.

Esta es la salida de una autentificación exitosa, así se conoce como debería verse la salida:

smb_auth -W LIDY –d

lidibet xxxxxxxx

Domain name: LIDY

Pass-through authentication: no

Query address options:

Domain controller IP address: 10.128.3.69
Domain controller NETBIOS name: VEGA
Contents of //VEGA/NETLOGON/proxyauth: allow
OK

Una lista de control de acceso para especificar un fichero donde estarán los usuarios con permiso para navegar en Internet y además una ACL para los sitios prohibidos.

Se crean dos ficheros en cualquier parte del disco duro:
touch /etc/squid/permitidos #estos son los usuarios con permiso de navegar en internet
touch /etc/squid/sitiosdenegados #aquí esta el listado de sitios prohibidos

A estos dos fichero se les debe dar permiso de lectura y escritura:
chmod 600/etc/Squid/permitidos
chmod 600/etc/Squid/sitiosdenegados

¿Qué tiene ¨permitidos¨?:
La lista de usuarios que pertenecen al dominio de la institución que tienen permisos

Ej “permitidos”: Lidy
dalay
Ej “sitiosdenegados”: www.sitioporno.com
sitioindeseable.com
sex
xxx
adult

En el Squid.conf:
INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
Example rule allowing access from your local networks. Adapt
to list your (internal) IP networks from where browsing should
be allowed
#acl our_networks src 192.168.1.0/24 192.168.2.0/24
#http_access allow our_networks

http_access allow localhost

acl autenticados proxy_auth REQUIRED
acl userspermitidos proxy_auth "/etc/squid/permitidos"
acl sitiosdenegados url_regex "/etc/squid/sitiosdenegados"
http_access allow autenticados userspermitidos !sitiosdenegados

And finally deny all other access to this proxy
http_access deny all

Parámetro cache_mgr:
Por defecto, si algo ocurre con la caché, como por ejemplo que muera el procesos, se enviará un mensaje de aviso a la cuenta *webmaster* del servidor. Puede especificarse una distinta si acaso se considera conveniente.
cache_mgr lminguez@estudiantes.uci.cu

Usuario y grupo con el que se ejecutará Squid:
Se puede usar el usuario que se quiera. Aquí se utiliza el que monta Debian por defecto: proxy. Nunca utilizar a root por motivos obvios.
Las líneas a tocar son estas dos:
cache_effective_user proxy
cache_effective_group proxy

HTTPD-ACCELERATOR OPTIONS (Para acelerar la caché)
En esta opción se deben habilitar los siguientes parámetros para navegar más rápido cuando los objetos ya están en la caché de Squid y así se optimiza enormemente la utilización del ancho de banda:
Opciones para proxy convencional:
httpd_accel_host virtual
httpd_accel_port 0
httpd_accel_with_proxy on
Una vez terminada la configuración, se ejecuta el siguiente comando para iniciar por primera vez Squid:

/etc/init.d/Squid start

Si se necesita reiniciar para probar cambios hechos en la configuración, ejecutar lo siguiente:

/etc/init.d/Squid restart.

2. Para levantar dos instancias del squid en una misma PC.

Cliente:

Cualquier sistema operativo (Ej: Windows XP, Windows Server 2000, Linux).

Utilizar el navegador que se desee (Internet Explore, Mozilla Firefox), lo que hay que configurarlo y decirle quien es el proxy, cual es el puerto por el que escucha y para que peticiones no pasar por él.

Proxy:

Sistema Operativo Linux.

Software necesario para la instalación del proxy.

Dos Squid levantados con configuraciones diferentes (Squid1 y Squid2).

Dos tarjetas de red o dos interfases con direcciones de IP diferentes: una mirando para la red del cliente y otra con el IP real para internet.

Levantando las dos instancias del squid en el servidor:

1. Se instala el Squid.
2. Se detiene el Squid: /etc/init.d/squid stop
3. Se elimina la configuración actual:

   update-rc.d -f squid remove

   rm /etc/init.d/squid

   rm /etc/squid/squid.conf

   rm /etc/default/squid

4. Se crean los dos ficheros de configuración squid_1 y squid_2:

   mv /var/spool/squid /var/spool/squid_1

   cp -r /var/spool/squid_1 /var/spool/squid_2

   chown -R proxy.proxy /var/spool/squid_1

   chown -R proxy.proxy /var/spool/squid_2

   cp ./config/squid_1.conf /etc/squid/

cp ./config/squid_2.conf /etc/squid/

5. Se copian los script en el directorio especificado para levantar las dos instancias:

cp ./script/squid_1 /etc/init.d/

cp ./script/squid_2 /etc/init.d/

6. Se asignan los permisos necesarios sólo para el usuario root.

chmod 0755 /etc/init.d/squid_1

chown root.root /etc/init.d/squid_1

chmod 0755 /etc/init.d/squid_2

chown root.root /etc/init.d/squid_2

Configuración del Squid_1:

Puerto por el que escucha las peticiones: http_port 3128

Memoria para los objetos en tránsito, los hot y los negativamente guardados en caché:

cache_mem 16 MB

Tamaño de la caché en el disco duro para Squid:

cache_dir ufs /var/spool/squid_1 700 16 256

Parámetro para saber donde Squid guarda los log en el disco duro:

cache_access_log /var/log/squid/squid_1-access.log

cache_log /var/log/squid/squid_1-cache.log

Definir la ubicación del archivo squid_1.pid

pid_filename /var/run/squid_1.pid

Parámetro para especificarle el programa que usará para la autenticación.

auth_param basic program /usr/lib/squid/smb_auth -W lidy

Dos ACL una para pedir autenticación y otra para llamar a un fichero donde están los usuarios permitidos para navegar en internet.

acl autenticados proxy_auth REQUIRED

acl userspermitidos proxy_auth "/etc/squid/permitidos_squid_1"

Una regla de control de acceso aplicada a las ACL para permitirles o no el acceso a Squid en el orden requerido.

http_access allow autenticados userspermitidos

Parámetro para si algo ocurre con la caché, se envíe un mensaje de aviso a la cuenta que se especifique.

cache_mgr lminguez@estudiantes.uci.cu

Especificar el usuario y el grupo con el que se ejecutara Squid.

cache_effective_user proxy

cache_effective_group proxy

Por último las líneas pertinentes que aceleran la caché para un proxy convencional.

httpd_accel_host virtual

httpd_accel_port 0

httpd_accel_with_proxy on

Configuración del Squid_2:

Puerto por el que escucha que no debe ser el mismo que el del squid1.

http_port 8080

Se especifica cual es su servidor padre, es decir debajo de cual servidor va a trabajar, los puertos por el que escucha el padre tanto el http como el icp y además especificándole la opción de que no guarde en caché los objetos traídos, sino que solo los atienda los pedidos.

cache_peer 10.128.30.86 parent 8080 3130 proxy-only

never_direct alow all

cache_mem 16 MB

Parámetro para saber donde Squid guarda los log en el disco duro, que no es el mismo fichero donde los guarda el squid_1.

cache_access_log /var/log/squid/squid_2-access.log

cache_log /var/log/squid/squid_2-cache.log

Definir la ubicación del archivo squid_2.pid

pid_filename /var/run/squid_2.pid

Parámetro para especificarle el programa que usara para la autenticación.

auth_param basic program /usr/lib/squid/smb_auth -W lidy

Dos ACL, una para pedir autenticación y otra para llamar a un fichero donde están los usuarios permitidos para navegar detrás del proxy padre (squid3) y salir a Internet, esta lista no es la misma que se especifica en squid1. Aquí se agrega una nueva acl para denegar el acceso a varios sitios que están prohibidos.

acl autenticados_2 proxy_auth REQUIRED

acl userspermitidos_2 proxy_auth "/etc/squid/permitidos_squid_2"

acl sitiosdenegados_2 url_regex "/etc/squid/sitiosdenegados_squid_2"

http_access allow autenticados_2 userspermitidos_2 !sitiosdenegados_2

Una regla de control de acceso aplicada a las ACL para permitir o no el acceso a Squid en el orden requerido.
http_access allow autenticados_2 userspermitidos_2 !sitiosdenegados_2

Especificar el usuario y el grupo con el que se ejecutará Squid.
cache_effective_user Proxy
cache_effective_group proxy

Configuración del Squid3:

Este servidor proxy no se encontrará en la institución, así que su configuración será la que el administrador del mismo le asigne, configurando los parámetros básicos será necesario, pero eso sí, tiene que tener una acl donde se le especifique quien es el servidor hijo y con su respectiva regla de acceso para permitirle el acceso.
Puerto por el que escucha las peticiones.
Memoria para los objetos en transito, los hot y los negativamente guardados en cache.
Tamaño de la caché en el disco duro para Squid.
Parámetro para saber donde Squid guarda los log en el disco duro.
Una ACL para especificar el IP del proxy que tiene que navegar a través de el.
acl permitirip src "/etc/squid/permitidos" donde permitidos tendrá el IP del servidor hijo, o de los servidores porque pueden ser más de uno.
La regla de control de acceso aplicada a la ACL para permitirla.
http_access allow permitirip
Parámetro para si algo ocurre con la caché, se envíe un mensaje de aviso a la cuenta que especifiques.
Especificar el usuario y el grupo con el que se ejecutara Squid.
Por último las líneas pertinentes que aceleran la caché para un proxy convencional.

3. Instalación del software para analizar las trazas que generan los dos Squid y mostrarla en formato HTML:

Instalando SARG:

Apt - get install sarg

Otra forma es obteniendo la fuente de Internet, descomprimirlo y con la secuencia mundialmente conocida de configure, make y make install se instala rápidamente.

Una vez instalado pasamos a la configuración:

Recordar que en el proxy hay levantadas dos instancias del squid, las cuales generan reportes diferentes, por tanto hay que crear dos ficheros de configuración para el sarg y lo necesario para que funcione.

1. Eliminando configuración actual:

rm -r /var/www/squid-reports/

rm /etc/squid/sarg.conf

2. Creando nuevos ficheros de configuración para squid_1 y squid_2

cp -r ./site/* /var/www/

cp ./config/sarg_squid_1.conf /etc/squid/

chmod 0644 /etc/squid/sarg_squid_1.conf

chown root.root /etc/squid/sarg_squid_1.conf

cp ./config/sarg_squid_2.conf /etc/squid/

chmod 0644 /etc/squid/sarg_squid_2.conf

chown root.root /etc/squid/sarg_squid_2.conf

3. Modificandolos ficheros sarg_squid_1.conf y sarg_squid_2.conf:

El parámetro para darle la ubicación del fichero del cual debe obtener los log para cada sarg son ficheros diferentes:

access_log /var/log/squid/squid_1-access.log

access_log /var/log/squid/squid_2-access.log

Decirle el fichero en el cual debe mostrar los log:

output_dir /var/www/squid_1-reports

output_dir /var/www/squid_2-reports
Y los demás parámetros que vienen descomentados en el fichero de configuración.

4. Especificando el fichero donde estarán los script de booteo para levantar las dos instancias del sarg para cada squid y dándole los permisos que les corresponden.
cp ./script/sarg_squid_1.sh /etc/squid/
chmod 0755 /etc/squid/sarg_squid_1.sh
chown root.root /etc/squid/sarg_squid_1.sh
cp ./script/sarg_squid_2.sh /etc/squid/
chmod 0755 /etc/squid/sarg_squid_2.sh
chown root.root /etc/squid/sarg_squid_2.sh

Contenido de estos dos script:
sarg_squid_1.sh:
/usr/bin/sarg -f /etc/squid/sarg_squid_1.conf
sarg_squid_2.sh:
/usr/bin/sarg -f /etc/squid/sarg_squid_2.conf

5. Especificar el fichero del cron que se utilizara para refrescar el sitio donde se muestran los reportes que genera el squid. Este fichero se configura de acuerdo al tiempo en que se quiera refrescar el sitio.
cp ./cron/root /var/spool/cron/crontabs/root
chmod 0600 /var/spool/cron/crontabs/root
chown root.crontab /var/spool/cron/crontabs/root
/etc/init.d/cron restart.

4. Instalación del servidor web Apache para que se muestren los reportes generados por el Sarg:

Apt - get install Apache2.

5. Instalación del firewall (Iptables):

Por último una vez terminado todo lo referente al proxy es necesario continuar con lo necesario para que funcione el firewall (Iptables).

Ante todo configurar correctamente las tarjetas de red de la PC donde vamos a montar el firewall. Para configurar las tarjetas por separado: etc/network/interfaces. El iptables ya viene instalado por defecto.

Ficheros a salvar / restaurar. Tenerlos al final bien guardado las reglas que se encuentran en /etc/iptables: el fichero "iptables" que se encuentra en /etc/init.d/ y el fichero "interfaces" que se encuentra en /etc/network.

Después de instalar, crear el directorio "iptables" en "/etc/": mkdir /etc/iptables.

Copiar las reglas salvadas para el directorio creado anteriormente. Verificar dueño, grupo y que tengan permiso de ejecución para el dueño:

chown -R root.root /etc/iptables

chmod 755 -R /etc/iptables/*

Crear el directorio "iptables" en "/var/lib/": mkdir /var/lib/iptables.

Crear el fichero saved-rules: touch /var/lib/iptables/saved-rules. Verificar dueño y grupo y darle los permisos necesarios:

chown -R root.root /var/lib/iptables

chmod 600 /var/lib/iptables/saved-rules

Copiar el fichero "iptables" salvado para "/etc/init.d/". Editarlo y verificar que el nombre del fichero de reglas sea el que es para la ocasión. Verificar dueño y grupo y darle los permisos necesarios:

chown -R root.root /etc/init.d/iptables

chmod 755 /etc/init.d/iptables

Para crear los enlaces en el run-level usar la linea de comando:

update-rc.d iptables start 19 2 3 4 5. El número 19 se usó para la prioridad en los run-level.

Para actualizar las reglas ejecutar: /etc/init.d/iptables update.

Por último para salvar las reglas ejecutar: /etc/init.d/iptables save.

Glosario de Términos y Abreviaturas

CD: Compact Disc - Disco Compacto.

DoS: Denial of Service - Denegación de Servicio.
DNS: Domain Name Server - Servidor de Nombres de Dominio.

FTP: File Transfer Protocol – Protocolo de Transferencia de Ficheros.

HTML: HyperText Markup Language - Lenguaje de Marcas HiperTextuales.
HTTP: HyperText Transfer Protocol - Protocolo de Transferencia de HiperTexto.
HTTPS: Versión segura del protocolo HTTP que utiliza un cifrado basado en las Secure Socket Layers (SSL).

Infomed: Red Telemática de Salud Cubana.
IP: Internet Protocol - Protocolo de Internet.
ICMP: Internet Control Message Protocol - Protocolo de Mensajes de Control de Internet.
IRC: Internet Relay Chat - Comunicación en tiempo real basado en texto.

Logs: Registro de actividad de un sistema, que generalmente se guarda en un fichero de texto.
LAN: Local Area Network - Red de Área Local.

Nmap: Programa de consola que muestra todos los puertos que están levantados en la terminal deseada.
NIC: Network Interface Controller - Controlador de Interfaz de Red.
NAT: Network Address Translation - Traducción de Dirección de Red.

OSI: Open System Interconnection - Modelo de referencia de Interconexión de Sistemas Abiertos.

PC: Personal Cumputer – Computadora Personal.

Ping: Herramienta de diagnóstico de redes TCP/IP, que ofrece información útil sobre la presencia en red de de otro ordenador.

POP3: Post Office Protocol - Protocolo más común para descarga de correo electrónico desde un servidor.

PPP: Point to Point Protocol – Protocolo Punto a Punto. Mecanismo para crear y ejecutar IP.

Routers: Interconecta segmentos de red, o algunas veces hasta redes enteras.

SMTP: Simple Mail Transfer Protocol - Protocolo simple de transferencia de correo electrónico.

Telnet: Protocolo que sirve para acceder mediante una red a otra PC, para manejarla como si estuviéramos sentados delante de ella.

TCP: Transmission Control Protocol - Protocolo de Control de Transmisión.

TxD: Transmite Datos de Señal de salida.

UDP: User Datagram Protocol **-** Protocolo de Datagramas de Usuario.

UPS: Protege los datos proporcionando batería de reserva para trabajo en red cuando falla el fluido eléctrico.

URL: Uniform Resource Locutor - Localizador Uniforme de Recurso.

VLAN: Virtual LAN - Red de área local Virtual.

VPN: Virtual Private Network - Red Privada Virtual.

WAN: Wide Area Network – Red de Area extendida.