UNIVERSIDAD INTERNACIONAL SEK

FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES

Trabajo de fin de carrera titulado:

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL ESQUEMA DE SEGURIDAD PERIMETRAL PARA LA RED DE DATOS DE LA UISEK – ECUADOR (CAMPUS MIGUEL DE CERVANTES)

Realizado por:

DAVID MAYORGA POLO

Como requisito para la obtención del título de
INGENIERO DE SISTEMAS EN INFORMÁTICA

QUITO, JULIO DE 2008

DECLARACIÓN JURAMENTADA

Yo, David Mayorga Polo, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en éste documento.

A través de la presente declaro que el patrimonio intelectual de éste trabajo le pertenece a la UNIVERSIDAD INTERNACIONAL SEK, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad Institucional vigente.

……

David Mayorga Polo

DECLARATORIA

El presente trabajo de investigación de fin de carrera, titulado

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL ESQUEMA DE SEGURIDAD PERIMETRAL PARA LA RED DE DATOS DE LA UISEK – ECUADOR (CAMPUS MIGUEL DE CERVANTES)

Realizado por el alumno

DAVID MAYORGA POLO

Como requisito para la obtención del título de

INGENIERO DE SISTEMAS EN INFORMÁTICA

Ha sido dirigido por el profesor

Ing. SANTIAGO MENA VELASCO

Quien considera que constituye un trabajo original de su autor.

…….

Ing. SANTIAGO MENA VELASCO

Director

Los profesores informantes

Ing. Gonzalo Ruales Ruales

Ing. Wilmer López Cevallos

Después de revisar el trabajo escrito presentado,

Lo han calificado como apto para su defensa oral ante el tribunal examinador.

……..……

Ing. GONZALO RUALES RUALES

……..……

Ing. WILMER LÓPEZ CEVALLOS

Quito, a 15 de Julio de 2008

AGRADECIMIENTOS

Esta tesis de grado mas que el contenido en si, el esfuerzo está dedicado a las personas mas queridas en mi vida, mi esposa María Del Pilar, que me apoyó en todo momento en el proceso, sin dejarme desanimar por lo lejos que se veía el camino, a mis padres Francisca y Hernán, mi hermana Sandra, que aunque a la distancia supieron guiarme y hasta darme “el empujón” para enfocarme en el importante proceso de la tesis de grado. También agradezco a Ramiro y Pilar por su apoyo y preocupación en todo el proceso de mi trabajo de investigación.

Les agradezco inmensamente por la guía y el esfuerzo puesto en mi trabajo a mi tutor, Ing. Santiago Mena, que siempre estuvo desde el principio con toda la voluntad de ayudarme y sacar adelante este difícil trabajo, a mi decana Ing. Viviana Guerrón que siempre me animó y me ayudó, aún en días de difícil trabajo para ella, al Ing. Wilmer López, persona muy conocedora de su profesión, nunca me negó un pedido de ayuda.

No faltaría mas agradecerles a todos los docentes mencionados por la agradable amistad que hemos entablado desde antes y durante este difícil proceso de la obtención de mi título universitario de Ingeniero de Sistemas.

RESÚMEN EJECUTIVO

Las Instituciónes educativas universitarias, además de perseguir el noble fin de educar a la gente, también tienen su parte administrativa, en la que se persiguen otros objetivos que vayan de la mano con el principal, que es educar. Las Universidades también son empresas, como cualquier otra, y necesitan proteger sus activos críticos de cualquier eventualidad que pueda suceder y afectar a su negocio. En la UISEK Ecuador Campus Miguel de Cervantes se consideran parte de los activos críticos los sistemas informáticos, no solo porque alojen información sensible, sino también porque son parte importante del proceso de educación de los alumnos y actualmente no se les da el suficiente resguardo, por lo que la presente investigación propone un esquema de seguridad perimetral para la red de datos del Campus; además del desarrollo de un manual de políticas de seguridad, así como un plan de contingencia ante posibles desastres y un esquema de monitoreo proactivo de la seguridad perimetral establecida.

The university educative Institutions, besides persecuting the noble aim to educate people, also have their administrative part, in which other objectives are persecuted that complements the main one, which it is to educate. The Universities also are companies, like any other, and need to protect their critic assets of any eventuality that can happen and affect its business. In the SEK International University Campus Miguel de Cervantes the IT systems are considered part of the critic assets, not only because they host sensible information, but also because they are an important part of the process of educating students and at the moment it does not have the sufficient security, reason why the present investigation proposes a perimetral security scheme for the network data of the Campus; besides the development of security policies, as well as a contingency plan to execute in a event of a disaster and a proactive monitoring scheme of the established perimetral security.

TABLA DE CONTENIDOS

INDICE DE IMÁGENES

INDICE DE TABLAS

CAPITULO I

1.1.- INTRODUCCIÓN

El desarrollo tecnológico, el nivel de competitividad de las organizaciónes, la estrategia de negocios electrónicos y el mercado global, hace que las redes de datos de las Organizaciónes estén expuestas a varios peligros tales como intentos de accesos no autorizados, ataques internos y externos, hackers, entre otros. Este alto nivel de exposición se traduce en amenazas que afectan el nivel de confiabilidad de las redes de datos de las Organizaciónes y por consecuencia también tendrá repercusión en los resultados del negocio.

En este contexto y como parte del proceso que han venido desarrollando las Organizaciónes para asegurar las redes de datos, surge la seguridad informática, como una necesidad de resguardar o proteger la información ante los daños a los que puede estar sujeta por acción de cualquier individuo o grupo que pueda tener acceso malintencionado a ella. Estos daños pueden comprender la pérdida total, parcial o la modificación de la información, y las consecuencias pueden generar problemas de grandes proporciones. Generalmente las empresas son los objetivos más apetecidos para realizar cualquier tipo de ataque informático, ya que al atacante le resulta atractivo y retador el hecho de provocar una pérdida económica que incluso podría ocasionar la quiebra de la Organización. El objetivo de la seguridad informática es proteger la integridad, autenticidad y confidencialidad de la información y de los sistemas informáticos de la Organización, de manera que, los recursos tecnológicos de ésta sean totalmente fiables. Actualmente el ritmo de crecimiento de los ataques es muy acelerado y cada día se puede encontrar una vulnerabilidad nueva que ha sido descubierta, en cualquier sistema.

Otro punto muy importante en la seguridad de una red de datos, es la seguridad perimetral, que es la encargada de proteger el perímetro o límites de una red de las amenazas externas. En toda red de datos debería existir este tipo de seguridad de lo contrario, existiría un riesgo latente que la expondría a todo tipo de amenazas que circulan libremente por el Internet.

En términos generales se entiende como perímetro a los límites que encierran a una porción de espacio, de igual forma se aplica en redes de datos a seguridad perimetral, es decir, es el borde o límite de la red que está fortalecido "como la muralla de un castillo" para que las amenazas externas no puedan acceder al espacio considerado como privado, en éste caso a la red de datos interna.

La seguridad perimetral no es controlada y monitoreada por un solo componente, son una serie de recursos de hardware y software que conviven y se complementan para proteger diferentes puntos de la red creando el perímetro de seguridad. Uno de los principios de la seguridad perimetral parte de que toda información que viaja por la red es sospechosa.

La seguridad perimetral ha tenido un desarrollo considerable desde hace aproximadamente 15 años, antes solo se utilizaba un firewall, que sirve para proteger el perímetro de la red de manera óptima y no requería mayor control después de haberlo implementado, ahora no es tan sencillo, debido a la creciente necesidad de expansión de las redes empresariales fuera del perímetro físico. Ahora los usuarios necesitan estar conectados a la red de la Organización desde cualquier parte del mundo, un claro ejemplo son las redes privadas virtuales o VPN's (Virtual Private Network), es por esto que la seguridad perimetral también debe controlar éste tipo de conexiones a la red y poder filtrar los accesos genuinos de los maliciosos, por ello un solo componente que se ocupe de monitorear y controlar el perímetro no es suficiente y se requieran mayor cantidad de sensores en la red, óptimamente distribuidos de acuerdo a un análisis y diseños previos.

Sin embargo, a pesar de que las Organizaciónes han impulsado proyectos de aseguramiento de las redes de datos, todavía se evidencia que este tipo de esfuerzos es más de carácter reactivo que preventivo, es decir, se corrige el problema cuando ya ha ocurrido, por ello, la implementación de un esquema de seguridad perimetral en la red de datos de las Organizaciónes o Instituciónes Educativas como la Universidad Internacional SEK – Ecuador (Campus Miguel de Cervantes), se considera como un factor crítico para incrementar la seguridad para proteger la información importante de la Institución.

1.2.- ANTECEDENTES

La seguridad informática empieza a tomar forma desde los años setenta cuando se empieza a conocer sobre los primeros casos de intervención ilegal a los sistemas telefónicos, en ese tiempo la telefonía era la principal víctima; más tarde los sistemas informáticos empiezan a ser el blanco de intentos de ataques y el FBI (Federal Bureau of Investigation) empieza a tomar cartas en el asunto. Por el año de 1980, las brechas de seguridad de los sistemas informáticos empiezan a ser más comunes y en Estados Unidos se crea una nueva legislación para castigar severamente a individuos que ataquen a sistemas informáticos.
El gobierno de Estados Unidos, como contramedida para frenar éstos ataques crea el CERT (Computer Emergency Response Team), que comprendía un equipo de especialistas que se dedicaría a investigar las brechas de seguridad producidas en los sistemas informáticos atacados.

Con el crecimiento del número de piratas informáticos por 1987 aparecen los primeros virus de computadora, que rápidamente se esparcen y el número de virus existentes se dispara, para 1991 IBM y Symantec habían detectado más de 1000 virus esparcidos por todo el mundo.

A partir de 1995 la situación se agudiza, los hackers empezaron a atacar los sistemas informáticos militares y de gobierno con la intención causar daño y robar información,

atacaron a empresas de telefonía y debido a los ataques éstas sufrieron pérdidas millonarias, los ataques crecían cada vez mas y los sistemas informáticos cada vez parecían mas débiles, por esto el gobierno de Estados Unidos creo el Centro de Protección de Infraestructura Nacional, con el objetivo de proteger de los hackers la tecnología, sistemas de telecomunicaciones y sistemas de transporte de Estados Unidos.

En el año de 1999 surge el comercio electrónico y muchas de las empresas grandes construyen sitios para vender sus productos en Internet, los ataques a estos sitios web crecen en un 79% debido a la vulnerabilidad que conlleva estar en Internet, a partir de aquí empieza a nacer otro tipo de malware o software malicioso que se esparce rápidamente por el Internet.

Ecuador ha venido impulsando proyectos para incrementar la seguridad de datos principalmente en el ámbito de la banca, que siempre se ha considerado como un sector fuerte y de mucha influencia en la economía del país. Cerca a los años 80 aparecen los primeros cajeros automáticos para retiro de dinero efectivo en el país, prácticamente no se evidencia problemas de seguridad con el uso de éstos, pero el cambio fue grande para el sector bancario ya que la implementación de cajeros automáticos llevó a los bancos a crear las primeras redes LAN (Local Area Network) dentro de cada agencia y/o sucursal, hasta aquí todavía los sistemas bancarios no evidencian problemas en cuanto a seguridad perimetral se refiere, ya que siempre que se detecto algún tipo de estafa a los bancos se consideró como un ataque interno ya que las personas que llevaron a cabo el fraude tuvieron ayuda de algún empleado dentro de la institución financiera.

En el año 1995 la banca empezó a sufrir ataques de clonación de tarjetas en los cajeros automáticos lo que obligó a mejorar la tecnología de éstos y de las funciones del software que los manejaban. En el año 1996 cuando empiezan a aparecer los primeros sistemas web de banca electrónica, los bancos llegan a estar más propensos a los ataques ya que se encuentran en la red pública más grande del mundo, el Internet, sin embargo y debido a la globalización éste es un riesgo que hay que asumir para poder seguir siendo competitivo.

Por ésta misma época empieza el boom del Internet en el país, los ISP (Internet Service Provider) o proveedores de servicio de Internet son débiles en cuanto a seguridad se refiere, con el tiempo van mejorando notablemente al implementar medidas de seguridad

como firewalls, filtros de spam para los servidores de correo, configuración de servicios innecesarios en los servidores, entre otras.

Se puede considerar que las Instituciónes en Ecuador han conseguido desarrollar un sentido de la seguridad informática mas reactiva que preventiva en todo lo referente a ataques considerados como externos, todavía se piensa que en el país no existen personas con tanto nivel de conocimiento para realizar con éxito un ataque externo sobrepasando el perímetro de las redes aseguradas.

Por otro lado las Instituciónes se han venido desarrollando muy bien en lo referente a planes de contingencia generales y en los enfocados a posibles desastres naturales. Debido a la erupción que tuvo el volcán Pichincha en el año de 1996, muchas instituciónes se vieron ante la preocupación de que sus centros de cómputo estaban en alto de riesgo de desaparecer, a partir de ahí se empezaron a crear planes de contingencia y nuevos centros de cómputo alternos para garantizar la continuidad del negocio.

La Universidad Internacional SEK (UISEK) tiene una trayectoria de 15 años en el Ecuador, en sus inicios no contaba con una infraestructura tecnológica muy grande, pero con los años ésta ha ido creciendo y mejorado notablemente. En períodos pasados se ha evidenciaban varios problemas en la red de datos, generalmente relacionados con virus, cortes de energía, software malicioso instalado en algunas estaciones de trabajo, caídas frecuentes del servidor, etc.; todo esto debido a una escasa planificación para implementar y mantener la red informática, así como la falta de un plan de seguridad perimetral que resguarde la información y las configuraciones de los equipos informáticos tanto a nivel de estaciones de usuarios como de los servidores de aplicaciones. Los problemas ocasionados por ésta situación direon como resultado un efecto negativo en el aprovechamiento de los alumnos y en el desempeño del personal administrativo.

1.3.- SITUACIÓN ACTUAL

Actualmente la información es el activo más valorado, tanto por empresas como por particulares, y por lo tanto se la debe proteger ante riesgos existentes en el entorno, tanto interno como externo. Los riesgos internos son las amenazas propias de la Organización o del entorno cercano donde se maneja la información, tales como los empleados, y los riesgos externos son amenazas del entorno ajeno, en definitiva, cualquier individuo que tenga acceso a Internet.
Siempre van a existir amenazas latentes en cualquier ámbito, interno o externo, es por eso que la seguridad nunca debe ser poco estimada ya que siempre pueden haber intereses maliciosos para obtener o alterar dicha información.

Hoy en día cualquier persona puede ser capaz de comprometer la seguridad de una red empresarial o de una computadora personal, independientemente del lugar donde se encuentre, solo se necesita una computadora y acceso al Internet, cosas que la mayoría de personas pueden tener acceso fácilmente. En Internet existe infinidad de información para usuarios principiantes, intermedios y avanzados sobre cómo aprender técnicas que desafían y comprometen la seguridad informática en cualquier ámbito. Hay varios motivos por los cuales se desafía a la seguridad informática de cualquier persona u Organización, por simple diversión o curiosidad o por fines maliciosos, cualquiera de éstos dos motivos producen el mismo resultado, el daño a la víctima del ataque, así sea económico, moral, o simplemente una pérdida de tiempo.

Hoy en día hay una cantidad considerable de sitios y foros en Internet que se dedican a alertar a la comunidad sobre nuevos virus o ataques que aparecen, es como una carrera entre los grupos que crean el malware versus la comunidad que trata de defenderse ante estos ataques.

Los programas y herramientas maliciosas también llamado malware, han evolucionado rápidamente desde que apareció el primer virus para computadoras personales en 1987. A la par ha tenido que surgir el campo de la seguridad informática como una necesidad urgente de evitar el caos en los sistemas informáticos.

Es sorprendente el grado de evolución que va a tener el malware en el 2008, tal como se puede ver en éste fragmento de noticia sobre un resumen de seguridad del año:

"2008 será un buen año, pero para los virus, troyanos y malware en general. Según Kaspersky Labs la cifra rondará los 20 millones de nuevo malware para este año. Además de para el malware, también será el año de los creadores de virus, cada vez más preparados y dispuestos a poner sus cualidades del lado de los malos. El ejemplo más reciente, según Kaspersky, es el caso del famoso gusano Zhelatin cuyos variantes no sólo se tornaron más complejos durante el primer trimestre del año sino también desplegaron un amplio espectro de comportamientos peligrosos y vías de propagación.

En 2007, un total de 2.227.415 nuevos programas nocivos (virus, gusanos y troyanos), aparecieron en Internet, ocupando un espacio de 354GB. Se trataba de un número cuatro veces superior al que se había registrado en 2006. A finales del año pasado, muchos expertos declararon que esta excepcional situación en el flujo de programas nocivos dio como resultado la imposibilidad por parte de algunas compañías de antivirus a responder con efectividad ante tal cantidad de amenazas. Por otra parte, el número de programas maliciosos detectados en el primer trimestre de 2008, ya ha sobrepasado el número de programas detectados en 2007"[1]

A partir del lanzamiento de un nuevo software o nueva tecnología, los hackers inmediatamente empiezan a analizar las vulnerabilidades que pudieran explotar, y cuando las detectan las atacan y posteriormente las compañías fabricantes de éstos nuevos

[1] Fuente: http://www.vnunet.es/es/vnunet/news/2008/04/14/el_malware_crecera_diez_veces_en_2008

productos desarrollan los parches para corregir los llamados “huecos” de seguridad; es una lucha constante que por supuesto también genera beneficios a ciertos grupos (compañías fabricantes de soluciones antivirus) por lo que se presume que para ellos no sería rentable cortar el problema de raíz.

En el mundo globalizado en el que vivimos es necesario que la información esté a disposición de sus dueños en cualquier parte del mundo y a toda hora y para esto, es necesario estar interconectados mediante el Internet. A partir de aquí el nivel de seguridad decae considerablemente y se está expuesto a que alguien encuentre cualquier vulnerabilidad en el sistema informático de la Organización.

Por vulnerabilidad se entiende la exposición latente a un riesgo. En el área de informática, existen varios riesgos tales como: ataques de virus, códigos maliciosos, gusanos, caballos de Troya, puertas traseras, etc. Con la adopción de Internet como instrumento de comunicación y colaboración, los riesgos han evolucionado y ahora, las empresas deben enfrentar ataques de negación de servicio y amenazas combinadas; es decir, la integración de herramientas automáticas maliciosas, accesos no autorizados a los sistemas y capacidad de identificar y explotar las vulnerabilidades de los sistemas operativos o aplicaciones para dañar los recursos informáticos.

Las redes de datos están implementadas hoy en día en varios escenarios, desde una red de datos casera con no más de 3 computadoras, hasta una red de datos corporativa con decenas o cientos de usuarios haciendo uso de ella. En general son usadas para compartir recursos de una manera eficiente y ordenada, y aunque generalmente se les otorgan ciertos niveles de seguridad que ayudan a impedir el acceso no autorizado a la red, es muy común encontrar que no se realiza un monitoreo continuo del estado de seguridad de la red, pensando que las herramientas instaladas protegerán a la empresa sin que el usuario tenga que intervenir, pero se debería mantenerlas actualizadas y si es necesario complementarlas con otras herramientas o reemplazarlas si ya no satisfacen el nivel de seguridad deseado, ya que es muy posible que las técnicas de seguridad implantadas en cierto período de tiempo, al día de hoy ya son obsoletas.

Actualmente en la UISEK Campus Miguel de Cervantes existen documentos de inventario de equipos informáticos, un documento de reestructuración de la red que ya ha sido implementado y refleja la infraestructura actual, a fecha 08/Julio/2008, además se tienen dos documentos de políticas de respaldos y uso de los centros de cómputo, respectivamente. Los servicios de red actualmente no fallan como en ocasiones anteriores, pero todavía se evidencia cierto nivel de vulnerabilidad en todo lo relacionado a virus, saturación de servicios de red debido a inexistencia de filtros de contenidos, seguridad física, control de acceso, etc…

La infraestructura actual de la red no fue planeada con un enfoque de seguridad a futuro, ya que concuerda con el esquema típico de una red, que consiste en la protección de un firewall para toda la red privada, sin embargo, el crecimiento de la Institución puede llevar a la modificación del esquema de red actual, siendo necesario tomarlos en cuenta para realizar actualizaciones planeadas o estimadas de la infraestructura IT.

Actualmente no existe un esquema de seguridad perimetral establecido y previamente analizado para suplir las necesidades de seguridad de la red actual; existen ciertos puntos de control pero no valoraciones previas que justifiquen dichos puntos, son solo controles implementados esporádicamente.

1.4.- JUSTIFICACIÓN E IMPORTANCIA

La seguridad perimetral en una red de datos es de vital importancia para proteger la integridad, confidencialidad y autenticidad de la información, si no se aplicarían normas de seguridad en el perímetro de la red existiría un riesgo que en cualquier momento podría ser aprovechado generando consecuencias muy graves para cualquier empresa.

Es muy importante conocer a fondo todo el esquema y los componentes de la red de datos, a partir de lo cual, deberían implementar los controles necesarios para bloquear o filtrar los accesos desde las redes no confiables hacia la red privada. Todo esto conlleva a la elaboración de un esquema de seguridad perimetral que debe ser muy bien analizado y evaluado, ya que una mala planeación puede llevar a la saturación de los recursos de red y afectar al negocio.

La seguridad perimetral en cualquier red de datos está conformada por componentes de hardware y software, entre los cuales podemos destacar:

- Ruteadores
- Firewall
- IDS (Intrusion Detection System) o IPS (Intrusion Prevention System)
- Antivirus
- Antispam
- Software de filtrado de contenidos web

El uso de estas herramientas permitirá tener un control preventivo sobre el perímetro de la red.

Es importante también tener un plan de mantenimiento y monitoreo del perímetro de seguridad después de haber implementado el esquema, esto permitirá tener un control sobre los puntos críticos de la red y realizar un monitoreo periódico para cuantificar el nivel de seguridad existente en la red de datos.

Al implementar un esquema de seguridad perimetral se logrará tener una red eficiente y segura para las operaciones normales de la Organización.

En el campo de la seguridad informática es difícil llegar a un nivel del 100% de efectividad, pero con la implementación de un esquema de seguridad para los sistemas informáticos se logrará tomar acciones preventivas en lugar de reactivas.

Al analizar y reestructurar la seguridad perimetral de la UISEK - Ecuador (Campus Miguel de Cervantes), se evitará que la Institución esté expuesta al monitoreo frecuente en busca de vulnerabilidades.

1.5.- DEFINICIÓN DEL TEMA

Análisis, diseño e implementación del esquema de seguridad perimetral para la red de datos de la UISEK – Ecuador (Campus Miguel de Cervantes).

1.6.- OBJETIVOS

1.6.1.- Objetivos generales:

- Analizar, diseñar e implementar un esquema de seguridad perimetral en la red de datos de la UISEK – Ecuador (Campus Miguel de Cervantes).

1.6.2.- Objetivos específicos:

- Analizar el riesgo actual de la red de datos de la UISEK – Ecuador (Campus Miguel de Cervantes).
- Analizar y definir las necesidades de seguridad perimetral, en base a los resultados obtenidos del análisis de riesgo actual.
- Diseñar el esquema de seguridad definiendo políticas de seguridad, mecanismos de defensa, dispositivos de seguridad, software y hardware a utilizarse usando la información resultante del análisis previo sobre las necesidades de seguridad perimetral para la red.
- Analizar y diseñar un plan de mantenimiento y monitoreo del perímetro de seguridad de la red
- Implementar el esquema de seguridad perimetral en la red de datos de la UISEK – Ecuador (Campus Miguel de Cervantes).

1.7.- METODOLOGÍA

A continuación se describirá los aspectos metodológicos con los cuales se efectuó la investigación del proyecto.

1.7.1. Tipo de Estudio

1.7.1.1. Estudio Exploratorio:

Se ha realizado este tipo de estudio por las siguientes razones:

- Destaca aspectos fundamentales de una problemática determinada.
- Al realizar una investigación para encontrar resultados a la problemática, permite abrir mas temas de investigación que se consideren necesarios.

1.7.2. Método de investigación

1.7.2.1. Método Analítico Sintético

Hemos partido de las necesidades generales para poder llegar a lo más específico.

1.7.3. Fuentes y técnicas de recolección de información

La fuente de información es primaria ya que se obtuvo la ayuda directamente del administrador de sistemas de la UISEK Campus Miguel de Cervantes.

Además se ha obtenido información secundaria de artículos, libros de seguridades, documentos electrónicos de libre distribución en el Internet, foros.

1.7.4.- Estándar utilizado

En el campo de la tecnología la mejor manera de llegar a un alto nivel de calidad de servicio es manejar un equipo de personas para administrar la infraestructura tecnologica en base a reglas o guiándose por ciertas premisas, esto es lo que se usará para mejorar el nivel de seguridad de la infraestructura de red y comunicaciones de la Universidad Internacional SEK Ecuador.

La norma de estandarización ISO/IEC 13335:3 contempla todo lo relacionado a las técnicas de administración de seguridad IT en cualquier tipo de Organización, sea comercial, educativa, industrial, etc… y la norma ISO/IEC 13335:4 contempla el proceso de selección de contramedidas de seguridad que se adequen a la Organización a la que se aplica dicha norma.

Las normas de estandarización mencionadas nos darán un enfoque global sobre la implementación de un sistema de gestión de seguridad informática, que cumplirá con reglas internacionales y es por esto que serán aplicadas para implementar y administrar un sistema de seguridad informática en la Universidad Internacional SEK Ecuador.

1.8.- ESTRUCTURA DEL DOCUMENTO

CAPITULO 1: INTRODUCCIÓN

En este capítulo se tratarán los detalles formales de la Tesis, esto es, la introducción, antecedentes, justificación e importancia, los objetivos del proyecto en específico y se determinará la metodología a utilizar.

CAPITULO 2: MARCO TEORICO

En este capítulo se tratará todo lo relacionado a conceptos base sobre seguridad informática y perimetral, definición de información, además se hará una reseña histórica sobre la evolución de las amenazas informáticas y las contramedidas que se han venido desarrollando a lo largo de 20 años atrás.

CAPITULO 3: ANÁLISIS DE RIESGOS Y NECESIDADES

Se recopilará información sobre la infraestructura de red de la Universidad Internacional SEK – Ecuador (Campus Miguel de Cervantes) para su posterior análisis de riesgo y selección de medidas de seguridad, también se creará un manual de políticas de seguridad IT y un plan de contingencia ante desastres.

CAPITULO 4: DISEÑO DEL ESQUEMA DE SEGURIDAD PERIMETRAL

Se diseñará el esquema de seguridad perimetral usando como base la información obtenida del análisis de riesgos y se hará un plan de implementación y pruebas.

CAPITULO 5: IMPLEMENTACIÓN Y PRUEBAS

Se ejecutará el proceso de implementación del esquema de seguridad perimetral propuesto y se realizaran pruebas para verificar el correcto funcionamiento del mismo.

CAPITULO 6: DISEÑO DEL ESQUEMA DE MANTENIMIENTO Y MONITOREO DEL PERÍMETRO DE SEGURIDAD DE LA RED

Se creará un esquema para el mantenimiento y monitoreo de los componentes críticos, tanto de hardware como de software, que conforman el perímetro de seguridad de la red de la Institución universitaria.

CAPITULO 7: CONCLUSIONES Y RECOMENDACIONES

Se analizará el funcionamiento de los esquemas propuestos y se propondrán recomendaciones según lo experimentado.

CAPITULO 8: BIBLIOGRAFÍA

CAPITULO 9: GLOSARIO DE TÉRMINOS

CAPITULO 10: ANEXOS

CAPITULO II

MARCO TEÓRICO

2.1.- INTRODUCCIÓN

En este capítulo se va a tratar todo lo relacionado a conceptos bases sobre seguridad informática, desde el concepto de información hasta los componentes dentro de un esquema de seguridad perimetral.
También se entrará en el concepto de políticas de seguridad y procedimientos, de una manera general.

2.2.- DEFINICIÓN DE INFORMACIÓN

Se define información como *"Agrupación de datos con el objetivo de lograr un significado específico más allá de cada uno de éstos. Un ejemplo 2, 0, 0 y 1 son datos; y 2001 es información"*[2].

Existe información que debe o puede ser pública, es decir, puede ser visualizada por cualquier persona (por ejemplo, las estadísticas sobre la inflación de un país); y aquella que

[2] Fuente: http://www.xpress.com.mx/glosario_i.jsp

debe ser privada y solo puede ser visualizada por un grupo selecto de personas (por ejemplo, antecedentes médicos de una persona). En ésta última se debería maximizar esfuerzos para preservarla de ese modo reconociendo que dicha información es:

- **Crítica**: Indispensable para garantizar la continuidad operativa del negocio.
- **Valiosa**: Para la Organización es un activo corporativo con valor.
- **Sensitiva**: Debe ser conocida solo por las personas que necesitan la información.

Toda Organización debe garantizar ciertos aspectos de la información que maneja, debido a que constituye una ventaja fundamental para el negocio.

Es imprescindible garantizar estos tres aspectos en cuanto a la información que maneja:

- **Disponibilidad:** Que siempre esté disponible cuando se la necesita.
- **Integridad:** La información siempre debe ser auténtica y completa.
- **Confidencialidad:** La información debe ser solo vista por aquellas personas que están autorizadas para hacerlo.

2.3.- CONCEPTO DE SEGURIDAD INFORMÁTICA

La seguridad informática nació alrededor del año de 1980 como una necesidad de evitar y contrarrestar los efectos que producen los ataques informáticos. Todo esto empezó a tener más importancia cuando la información usada por los sistemas informáticos era cada vez más privilegiada o de suma importancia y llegaba a convertirse en un activo de una Organización o también de un particular, ahí es cuando la seguridad informática crece debido a la necesidad de proteger información muy importante que podría afectar al

negocio de una Organización. Actualmente en el mundo, la mayoría de Organizaciónes procesan los datos que utilizan para el desarrollo de su negocio en equipos informáticos, entidades privadas y públicas manejan información muy sensible no solo para ellos sino incluso para países enteros, por ejemplo, el registro civil maneja todos los datos de los ciudadanos de un país, qué pasaría si esos datos son modificados o eliminados por un pirata informático?, se desataría un caos de grandes proporciones, es éste el nivel de importancia que hay que otorgarle a la seguridad informática en cualquier Organización que considere a su información como un activo privilegiado.

2.3.1.- Amenazas informáticas

Basta con ver algunas estadísticas sobre amenazas informáticas del año 2007 para darse cuenta del daño que puede producir un ataque informático (*Ver Anexo 1*).

En el año 2007 *"hubo una pérdida de 66'930.950 USD de entre 194 empresas"*[3], debido a varios tipos de ataques las redes informáticas de dichas empresas, la más común fue contaminación por fraude financiero informático, virus en los sistemas informáticos, y el acceso no autorizado a información, entre otros.

Para que estos ataques hayan llegado a tener acceso al sistema informático, implica que la red o la infraestructura de los sistemas informáticos de la Organización, o los dos, poseían una vulnerabilidad, que muchas veces son difíciles de encontrar pero para tratar de contrarrestar esto se crean esquemas de seguridad que a corto o largo plazo incrementan continuamente el nivel de protección en la red y en los sistemas informáticos.

Las Organizaciónes cada día le están dando mas importancia a la inversión en seguridad aunque esto represente un gasto fuerte que a la final da resultados intangibles y esto hace

[3] Fuente: Anexo 1

que las personas que toman las decisiones sobre presupuestos nunca estén muy convencidos de invertir en seguridad, pero a la larga mediante reportes mundiales sobre seguridad y estadísticas de Organizaciónes afectadas por ataques informáticos, se dan cuenta que gracias a tener implementados esquemas de seguridad en la red no han sido parte de las estadísticas.

Aunque nunca se puede llegar a un nivel de 100% de seguridad, con un esquema bien implementado y políticas de seguridad estrictas se puede llegar fácilmente hasta un 99%.

2.3.2.- Tipos de ataques informáticos

Podemos definir como ataques, todas aquellas acciones que suponen una violación de seguridad de nuestra red que afecten la confidencialidad, integridad o disponibilidad de la información albergada en ella.

Estas acciones se pueden clasificar de modo genérico según los efectos causados:

- **Intercepción:** Cuando se desvía información a otro punto que no sea el destinatario original.
- **Modificación:** Alguien no autorizado consigue acceso a algún tipo de información en un sistema o base de datos y es capaz de manipularla.
- **Suplantación:** Es conocido como ”phishing” y consiste en crear portales exactos a otros y de ésta manera engañar a los usuarios para que den información confidencial, generalmente se albergan en servidores remotos a los que se ha tenido acceso no autorizado.
- **Autentificación:** Cuando un ataque suplanta a una persona con autorización.
- **Explotación de errores:** Suceden en el momento que se encuentran vulnerabilidades en los sistemas operativos, protocolos de red o aplicaciones.

- **Ataques de denegación de servicio (DoS):** Consiste en saturar un servidor con peticiones falsas hasta dejarlo fuera de servicio.

La red de la UISEK está expuesta a todos estos tipos de ataques, de usuarios internos o externos, se podría decir que el mayor peligro se encuentra en los usuarios internos, debido a que éstos se encuentran dentro de la Institución y por consiguiente están dentro del perímetro de seguridad de la red, es decir, no tienen restricción alguna en cuanto a seguridad perimetral se refiere.
Actualmente los ataques internos a nivel mundial producen más daño que los externos y son más frecuentes.

2.3.2.1.- Ataques internos:

Debido a la naturaleza de la Institución, el perfil que mas adecua a un atacante interno en la red de la UISEK, es el propio alumno, con más probabilidad de que sea un estudiante de Ingeniería de Sistemas, solo por el hecho de experimentar, aprender, curiosear o por otros motivos, ya se convierte en una potencial amenaza. Otro tipo de perfil que también se considera una amenaza interna y de mayor nivel que un estudiante, son los docentes de la facultad de Sistemas de la Universidad, así como los administradores de red de la misma, éstos tienen mayores bases de conocimiento que los estudiantes, y en el caso de los administradores, tienen el poder total sobre la red ya que ellos conocen usuarios y claves con todos los privilegios, en teoría los administradores deben tener un concepto de ética mucho mayor que los demás usuarios, ya que a ellos se les otorga el control total de la red, pero mundialmente en empresas ha habido casos de ataques internos fulminantes por parte de administradores de sistemas que han sido despedidos o simplemente están insatisfechos con algún aspecto de la empresa, por lo que también se han convertido en sujeto de amenaza para la red.
Existe una técnica de ataque interno que está preocupando bastante en el ámbito de la seguridad informática, ésta es la ingeniería social.

2.3.2.1.1.- Ingeniería social:

Consiste en un tipo de ataque persuasivo donde la única vulnerabilidad es la confianza plena en todas las personas de la Institución. El ataque consiste en persuadir y engañar, generalmente a usuarios de computadoras del área administrativa o incluso a los administradores de la red, con el fin de obtener la clave para un usuario con todos los privilegios, engañándolos con cualquier excusa, o convenciéndolos de ingresar éstos datos para que las otras personas utilicen el computador. Aquí ningún dispositivo de seguridad es eficiente ya que todo recae sobre el poder de convencimiento que tenga el atacante y con eso basta para burlar sistemas de seguridad multimillonarios de una manera sencilla, es por eso que también hay que capacitar a la gente para que este tipo de ataques no progresen.

Existen muchas modalidades de ataques internos que tienen que ver con el descuido de los empleados, por ejemplo cuando se levantan de su escritorio para ir a hacer alguna otra gestión y dejan su computador sin bloquear, es fácil que otra persona vaya al escritorio y use el computador para sacar información o ejecutar programas maliciosas. En éste caso no ha sido burlado ningún sistema de seguridad, ha sido un total descuido por parte del empleado.

Éstos tipos de ataques son probable que se presenten en el ambiente de la UISEK, debido a que los alumnos siempre van a necesitar solicitar servicios o equipos a los administradores o docentes, y si existe algún tipo de amistad o compañerismo, es probable que se los trate de convencer para obtener los datos buscados o por simples descuidos de las personas administrativas con sus computadoras, sería muy fácil que otra persona utilice ese computador para sacar información.

2.3.2.2.- Ataques externos:

El ámbito de ataques externos es más amplio y genérico que el de los ataques internos, ya que prácticamente cualquier red puede ser víctima de ataques provenientes de afuera sin importar el tipo de red que sea o si contiene información crítica en sus servidores.

Los ataques externos son mas fáciles de detectar y repeler que los ataques internos, ya que con solo revisar los logs del firewall o de otras herramientas de seguridad, se puede saber desde donde se realizó el ataque, que tipo de ataque fue y a que parte de la red se quiso afectar.

2.3.2.2.1.- Ataques de virus informáticos:

Se define como programas que contienen código malicioso que al ejecutarlos de diferentes maneras, desarrollan un comportamiento anormal en el computador, existen varios tipos:

- **Bombas lógicas:** Diseñados para activarse ante la ocurrencia de un determinado evento.
- **Troyanos:** Suelen propagarse como parte de programas de uso común y se activan cuando los mismos se ejecutan.
- **Gusanos:** Se autoduplican causando diversos efectos.
- **Keyloggers:** Son aplicaciones destinadas a registrar todas las teclas que el usuario aplasta en su computador.

Estos tipos de ataques también se pueden dar internamente, son más fáciles de evitar teniendo un antivirus actualizado diariamente, pero igual existe una amenaza latente ya que no se pueden desactivar puertos USB y CD-ROM en una Institución educativa como una

Universidad. Éstos producen daños molestos con pérdida de información a veces y siempre con una pérdida de tiempo tremenda, es importante tener buenos antivirus actualizados siempre para poder reducir la probabilidad de infección al mínimo. Incluso este tipo de ataques pueden llegar desde el exterior muy fácilmente por medio de e-mails, mensajería instantánea, etc.

2.3.2.2.2.- Puertas traseras:

Son aplicaciones cliente/servidor en el que un usuario puede tener control remotamente de otro computador y realizar diversas funciones como bloquear el teclado, abrir el micrófono, inhabilitar el cd-rom, tener acceso a los discos duros, etc.

2.3.2.2.3.- Ataques de Sniffing:

Consiste en interceptar el tráfico de una red, analizarlo y obtener información tal como usuarios, passwords, etc.

Este tipo de ataque generalmente sucede en redes inalámbricas no cifradas, como es el caso de la red inalámbrica de la UISEK, es una red abierta, no cifrada y sin clave de acceso. Para realizar éste ataque se necesita un software analizador de paquetes, que hoy en día los hay gratis en Internet, una computadora con tarjeta inalámbrica y estár dentro del rango de la red inalámbrica que va a ser atacada. En la UISEK sería muy sencillo perpetrar este ataque desde adentro o en los sectores alrededor de la Universidad sin levantar sospechas, puede ser un alumno con su laptop que esté conectado a la red wi-fi, nadie podrá imaginarse que esté ejecutando un programa analizador de paquetes en su laptop, de ésta manera podrá interceptar toda la información que otros usuarios conectados a la red wi-fi ingresen, por ejemplo en formularios en páginas de Internet, etc.

2.4.- CONCEPTO DE SEGURIDAD PERIMETRAL

La seguridad perimetral es una rama de la seguridad informática que se ocupa de vigilar el perímetro o “borde” de la red, es decir, es una defensa ante las amenazas externas que intentan filtrarse.

Se puede hacer una analogía con una muralla fortificada cuyo objetivo es restringir el paso a los enemigos, eso es básicamente lo que busca la seguridad perimetral, limitar los accesos solamente los paquetes confiables que circulan por la red y restringir aquellos que pueden hacer daño.

La seguridad perimetral no es una tecnología, se considera como un sistema que se compone de varios elementos de tecnología, de hardware y software, que actúan de manera conjunta para detectar y tomar acción ante cualquier intento de acceso a la red privada, es decir, se parte de que todo paquete que pasa por la red es sospechoso.

El desarrollo de un esquema de seguridad perimetral se divide en 6 etapas que son:

- Análisis de riesgo de la red de datos tomando en cuenta los recursos existentes.
- Análisis de necesidades de seguridad para la red de datos.
- Diseño del nuevo esquema de seguridad perimetral.
- Implementación del esquema de seguridad perimetral.
- Pruebas de funcionamiento.

La ejecución de estas etapas conlleva un proceso de análisis que tiene que ser bien realizado para que la implementación del esquema de seguridad tenga un funcionamiento óptimo, ya que de lo contrario puede ocasionar varios inconvenientes en términos de rendimiento y funcionamiento normal de la red. Es por eso que es importante conocer qué ventajas y posibles desventajas trae la implementación de éste tipo de esquema y sobre todo evaluar si es factible y oportuno implementarlo haciendo un análisis previo de las necesidades tecnológicas y de seguridad de la Organización.

La seguridad perimetral ha crecido en gran medida en los últimos 10 años, aunque se considera como una tarea que requiere atención constante ya que cuando se aseguran ciertos aspectos en una red, los piratas informáticos ya están esforzándose por encontrar otras vulnerabilidades.

2.4.1.- Objetivos de seguridad perimetral

Los objetivos de la seguridad perimetral son:

- Proteger el perímetro de la red privada ante amenazas externas.
- Filtrar eficientemente los accesos solicitados hacia la red privada.
- Tomar acción ante cualquier amenaza antes de que acceda a la red privada.

2.4.2.- Componentes de seguridad perimetral

Los componentes esenciales que pueden existir en el perímetro de una red son:

2.4.2.1.- Ruteadores de perímetro:

Los ruteadores direccionan el tráfico de red hacia, desde y dentro de la red. Los ruteadores de perímetro (también se los conoce como ruteadores de frontera o límite) son los últimos ruteadores que están justo antes de una red no confiable, como el Internet. Debido a que todo el tráfico de Internet de una Organización pasa por estos ruteadores, se lo utiliza como un primer y último filtro, por eso se los considera críticos para la defensa.

2.4.2.2.-Firewalls:

El firewall es un dispositivo que protege de amenazas externas a uno o varios equipos dentro de una red

Existen 2 tipos de firewall, los personales que protegen a un solo equipo en el cual está instalado, o los de red que protegen a los equipos de toda una red de datos.

2.4.2.2.1.- Firewalls de red:

Existen 3 tipos de firewalls de red, los filtros de paquetes, los stateful y los deep-packet inspection.

Los filtros de paquetes son los firewall de más simple tipo y sirven para controlar el acceso a determinados segmentos de red definiendo que tipo de tráfico es permitido y que otro tipo no es permitido. Los filtros de paquete analizan el tráfico en la capa 4 del modelo OSI (Transporte):

- Dirección de origen
- Dirección de destino
- Puerto de origen
- Puerto de destino
- Protocolo

Este tipo de firewalls no analizan ciertos campos de Capa 3 y 4 como el número de secuencia y banderas de control TCP.

Los firewall stateful analizan toda conexión que pasa por su interfaz de red, además de analizar los campos del encabezado del paquete (filtro de paquetes) también analiza el estado de la conexión (establecida, cerrada, reset, negociando), que sirve para detectar otros tipo de ataques que se basan en el estado de la conexión de los paquetes.

Los firewall deep-packet inspection o de inspección profunda de paquetes analizan la información en la Capa 7 (Aplicación).

Existen aplicaciones que requieren un manejo especial de los paquetes de datos cuando pasan por un firewall. Estos incluyen aplicaciones y protocolos que tienen embebidos información de direccionamiento IP en sus paquetes de datos o abren canales secundarios en puertos asignados dinámicamente (P2P). Usando inspección de aplicaciones se puede identificar los puertos asignados dinámicamente por la misma y permitir o denegar el intercambio de datos por esos puertos en una conexión específica.

2.4.2.3.- NAT (Network Address Translation):

La mayoría de firewalls existentes ofrecen el servicio de NAT, que consiste en enmascarar o disfrazar la dirección IP de los hosts protegidos o que están detrás del firewall a una dirección IP pública, por ejemplo, una red corporativa con 30 hosts con diferentes IP privadas dentro de la red interna, saldrán a navegar por Internet con una sola dirección IP pública.

2.4.2.4.- IDS (Intrusión Detection Systems):

Un IDS es un dispositivo que detecta intentos de acceso malicioso a la red privada o a un host. Se los utiliza como sensores en varios puntos estratégicos de la red. Existen tres tipos básicos de IDS:

- **NIDS (Network Based IDS):** Es un IDS basado en red que controla el tráfico en busca de actividades sospechosas.
- **HIDS (Host Based IDS):** Es un IDS que protege un solo ordenador, por lo tanto el procesamiento del CPU es mucho menor a un NIDS.
- **DIDS (Distributed IDS):** Es un IDS que funciona en una arquitectura cliente-servidor, está compuesto por varios NIDS que actúan como sensores centralizando la información de posibles ataques en una unidad central que almacena los datos en una base de datos.

El funcionamiento de este tipo de herramientas se basa en el análisis del tráfico de red, el cual al entrar en contacto con el IDS es comparado con firmas de ataques conocidos o comportamientos sospechosos, como puede ser el escaneo de puertos, paquetes malformados, etc… y no solo se analiza que tipo de tráfico es sino también su contenido y su comportamiento.
Normalmente esta herramienta se integra con un firewall y al trabajar conjuntamente pueden convertirse es una herramienta muy poderosa de seguridad.
En un sistema pasivo, el sensor detecta la posible intrusión, almacena la información y envía una señal de alerta al administrador de la red, en cambio en sistemas reactivos el IDS responde a la actividad sospechosa reprogramando el firewall para que bloquee el tráfico de donde proviene el posible ataque.

2.4.2.5.- IPS (Intrusión Prevention System):

Un IPS es un sistema que establece políticas de seguridad para proteger un equipo o una red. A diferencia con un IDS que se ocupa de alertar al administrador sobre actividad

sospechosa o toma acción ante la detección de una posible intrusión, un IPS ya tiene preestablecidas políticas de seguridad que no dejarían acceder cierto tipo de tráfico o a ciertos patrones que se detecten dentro de los paquetes, a la red privada, es un tipo de protección proactiva a diferencia del IDS que es reactiva.

2.4.2.6.- VPN (Virtual Private Networks):

Una VPN es una sesión de red protegida formada a través de un canal no protegido como es el Internet. Las Organizaciónes crean VPN's para ofrecer integridad de datos, autenticación y encriptación de datos para asegurar la confidencialidad de los paquetes enviados sobre una red no protegida. Una VPN permite a un usuario externo unirse a una red privada a que participe en ésta sin estar conectado físicamente o internamente a la misma. El uso de este tipo de conexiones está diseñado también para ahorrar costos en líneas dedicadas externas.

2.4.2.6.1.- Clasificación de las VPN:

- **Site-to-Site:** Permiten a las Organizaciónes establecer túneles VPN entre 2 o más ubicaciones (por ejemplo, oficina principal y sucursal) para que los usuarios se comuniquen mediante un medio compartido como el Internet.
- **Remote-access:** Permite a los usuarios trabajar desde ubicaciones remotas (casa, hotel, etc…) como si estuvieran fisicamente conectados a la red privada de la Organización.

2.4.2.7.- SSL VPN's:

Las VPN’s basadas en SSL (Secure Socket Layer) están teniendo bastante demanda hoy en día, debido a su facilidad de uso. La característica mas popular de éste tipo de VPN’s es que el usuario solo necesita ejecutar un browser y conectarse a la dirección de la VPN, ésta facilidad de uso es debido a que la mayoría de firewalls aceptan conexión a SLL (Puerto 443) y no es necesario abrir otros puertos para establecer la comunicación.

2.4.2.8.- Arquitectura de software:

La arquitectura de software consiste en un conjunto de patrones definidos que proporcionan el marco de referencia para guiar la construcción de un software. Existen varios tipos de arquitecturas:

- **Monolítica:** El software se compone de un solo módulo donde se encuentra todo el código, es poco organizado.
- **Cliente-Servidor:** El software realiza todo el cómputo en un solo computador (Servidor) y entrega datos ya procesados a los clientes conectados, para que estos no realicen ningún proceso de datos.
- **Arquitectura de 3 niveles:** El computo se divide en 3 niveles, persistencia (Almacenamiento de datos), negocio (procesamiento de datos) y presentación (interfaz de usuario).

Una arquitectura de software segura se trata de aplicaciones que están alojadas en la red interna de la Organización, generalmente con una arquitectura de 3 niveles:

- **1era Capa:** Front End o interfaz de usuario.
- **2da Capa:** Lógica de negocio de la aplicación (Código fuente).
- **3ra Capa:** Back End o bases de datos.

La arquitectura de software tiene un papel importante cuando se habla de una infraestructura segura porque el primer objetivo de la seguridad perimetral es proteger los datos y servicios de los sistemas de información.

2.4.2.9.- DMZ (Zonas desmilitarizadas) y subredes monitoreadas:

Una DMZ es un área insegura, en el caso de una red es todo lo que está antes del firewall, y una subred monitoreada es una red pequeña que está aislada de la red interna pero también tiene la protección del firewall, éste tipo de redes se usan para separar servidores que necesitan ser accesibles desde Internet de los servidores que contienen sistemas que se usan solo internamente en la Organización.

2.5.- CONCEPTO DE ESTRATEGÍAS DE DEFENSA

Una buena arquitectura de defensa trata la seguridad de la red por capas, ésta estrategia se llama “Defensa en profundidad”, y ayuda a proteger los recursos de red por etapas, es decir, si la primera capa de seguridad falla no está todo perdido ya que el atacante deberá comprometer la segunda capa y así sucesivamente.

La defensa en profundidad comprende el perímetro, la red interna y el factor humano. Cada uno de éstos contiene varios componentes que por si solos no son suficientes para asegurar una red, es por eso que cada uno debe ser complementario a los demás para formar una estructura de defensa óptima.

2.5.1.- Componentes de la estrategia “Defensa En Profundidad”

2.5.1.1.- El perímetro:

El perímetro de una red contiene todos o algunos de éstos dispositivos:

- Filtro de paquetes estáticos
- Firewall Stateful
- Proxy
- IDS e IPS
- Dispositivo de VPN
- Etc…

Todos estos dispositivos contribuyen a la seguridad en el perímetro de la red, trabajan de forma armónica entre todos para brindar un nivel de seguridad eficiente.

2.5.1.2.- Red interna:

La red interna se encuentra protegida por el perímetro y donde están todos los servidores, estaciones e infraestructura, con la que la compañía desarrolla su negocio.

Además del perímetro, la red interna debe ser asegurada desde adentro ya que aquí también hay un riesgo, tales como los propios empleados, y no necesariamente tienen que tener un fin malicioso, un empleado descuidado implica un riesgo para la seguridad de los sistemas de la Organización.

En la red interna se puede encontrar los siguientes componentes de seguridad:

- Filtros de ingreso y salida en cada router.
- Firewalls internos para segregar servicios.
- IDS's.
- Antivirus.
- Firewall personales.
- Sistemas operativos "endurecidos".
- Auditorías.

2.5.1.3.- El factor humano:

Siempre se considera el aspecto técnico cuando se trata de armar o actualizar una infraestructura de seguridad, pero no se pone mucha atención al aspecto humano cuando se implementa una solución.

Las políticas de seguridad son un factor muy importante cuando se quiere implementar una infraestructura de seguridad de Defensa en profundidad. Estos son los factores claves para tener una buena política de seguridad:

- **Autoridad:** Quién es responsable?
- **Alcance:** A quién afecta?
- **Expiración:** Cuando se termina?
- **Especificidad:** Que es requerido?
- **Claridad:** Todos pueden entenderlo?

El factor humano es tal vez el pilar fundamental para que la estrategia de Defensa en profundidad tenga éxito, es muy importante implementar políticas y educar a los empleados a ser cuidadosos.

2.6.- CONCEPTO DE POLÍTICAS DE SEGURIDAD

Una política de seguridad es una definición de lo que una Organización, sistema u otra entidad considera como “ser seguro”. Para una Organización se refiere a las normas de comportamiento de sus miembros así como los dispositivos de seguridad impuestos para los adversarios, como puertas con cerraduras electrónicas, etc… Para los sistemas informáticos se refiere a las normas de seguridad impuestas en dispositivos de software y hardware que no permiten el acceso a ellos por personas ajenas a la Organización.

Un grupo de políticas de seguridad consta en un documento que establece como una empresa planea proteger sus activos físicos, tecnológicos y de información.
Es un marco de trabajo dentro del cual una Organización establece los niveles necesarios de seguridad de la información para lograr los niveles de confiabilidad deseados.

2.7.- ESTRATEGÍAS DE ANÁLISIS DE RIESGOS

Existen varios tipos de estrategias de análisis de riesgos según la norma ISO/IEC 13335:3 que tienen que ver con la profundidad de análisis que se va a aplicar a los sistemas IT de la Organización. Esto depende de varios factores como el nivel de importancia que tengan los sistemas IT para el neogcio, la magnitud de la infraestructura, el nivel de seguridad requerida, la situación económica la Organización, etc…

Es posible combinar las diferentes estrategias de análisis de riesgos para aplicarlas en los sistemas IT, ésta es una buena opción si existen, por ejemplo, servidores de misión crítica que necesiten un mayor nivel de seguridad que otros, de ésta manera los riesgos se analizan de una forma acorde a los requisitos de seguridad de cada servidor.

La norma de estandarización ISO/IEC 13335:3 propone las siguientes estrategias de análisis de riesgos:

- Análisis de riesgos base
- Análisis de riesgos informal
- Análisis de riesgos detallado
- Análisis de riesgos combinado

2.7.1.- Análisis de riesgos base

El objetivo del análisis de riesgos base es establecer un mínimo de contramedidas para proteger todos o parte de los sistemas IT de una Organización. Con ésta estrategia es posible aplicar protección base de una forma horizontal, y si es necesario, aplicar un análisis de riesgo detallado a los sistemas IT de misión crítica. El uso de esta estrategia reduce la inversión que la Organización tiene que realizar para mejorar la seguridad en sus sistemas.

2.7.2.- Análisis de riesgos informal

Esta estrategia se usa para realizar análisis de riesgos que no están basados en métodos estrucuturados o guías de estándares, es simplemente un método personalizado o llevado a cabo por personas experimentadas que no necesitan guiarse punto por punto por metodologías ya establecidas.

2.7.3.- Análisis de riesgos detallado

El análisis de riesgos detallado involucra un nivel de profundidad alto al identificar y valorar activos, al analizar amenazas y vulnerabilidades. Los resultados al aplicar ésta estrategia son usados para analizar los riesgos a detalle a los que están sometidos los sitemas IT e identificar justificadamente las contramedidas a implementar.

2.7.4.- Análisis de riesgos combinado

La estrategia combinada inicia con un análisis de riesgos detallado en todos los sistemas IT de la Organización que se consideran críticos para el negocio, y para los demás sistemas se les aplica la estrategia de protección base, de ésta manera se identifica claramente los puntos que requieren mayor seguridad y los que no la requieren en un nivel alto, y se minimiza el trabajo en seleccionar las contramedidas, asegurándonos que los activos críticos están bien protegidos.

CAPITULO III

ANÁLISIS DE RIESGOS Y NECESIDADES

3.1.- METODOLOGÍA DE ANÁLISIS DE RIEGOS SEGÚN LA NORMA ISO/IEC 13335:3 “TÉCNICAS PARA LA ADMINISTRACIÓN DE SEGURIDAD IT”

La administración de seguridad IT incluye el análisis de requerimientos de seguridad, un plan para satisfacer dichos requerimientos, la implementación del plan y el mantenimiento y administración de las seguridades implementadas.

3.1.1.- Objetivos de seguridad:

Es importante definir los objetivos de seguridad para los sistemas IT de la UISEK.

- Aumentar el nivel de disponibilidad, integridad y fiabilidad de los sistemas IT del Campus Miguel de Cervantes, a un porcentaje aceptable.
- Garantizar la confidencialidad de la información que manejan los sistemas IT del Campus Miguel de Cervantes.

3.1.2.- Estrategias de seguridad:

- Aplicar el método de “Baseline Approach” para realizar el análisis de riesgos de la Institución.
- Aplicar medidas de seguridad convenientes para reducir al máximo el porcentaje de vulnerabilidad en los sistemas IT del Campus.
- Desarrollar un manual de políticas de seguridad IT para el Campus.
- Desarrollar un manual de procedimientos para operar los sistemas IT del Campus.
- Desarrollar un plan de contingencia para manejo de incidentes.

3.1.3.- Manual de políticas de Seguridad IT:

3.1.3.1.- Introducción

En una Organización la gestión de seguridad puede ser una compleja y difícil tarea de realizar, para tener éxito es necesario emplear mecanismos reguladores de las funciones y actividades desarrolladas por cada uno de los empleados de la Institución. El documento que se presenta como políticas de seguridad, pretende ser el medio de comunicación en el cual se establecen las normas y controles que regulen la forma en que la Institución, prevenga, proteja y maneje los riesgos de seguridad en diversas circunstancias. Las normas y políticas expuestas en este documento sirven de referencia, en ningún momento pretenden ser normas absolutas, las mismas están sujetas a cambios realizables en cualquier momento, siempre y cuando se tengan presentes los objetivos de seguridad. Toda persona que utilice los servicios que ofrece la red, deberá conocer y aceptar el reglamento vigente sobre su uso, el desconocimiento del mismo, no exonera de responsabilidad al

usuario, ante cualquier eventualidad que involucre la seguridad de la información o de la red Institucional.

En términos generales el manual de normas y políticas de seguridad informática, engloba los procedimientos más adecuados, tomando como lineamientos principales cuatro criterios:

- Seguridad Organizacional
- Seguridad Lógica
- Seguridad Física
- Seguridad Legal

3.1.3.2.- Alcance y propósito de las políticas de Seguridad IT.-

Las políticas pretenden un marco de referencia y procedimientos a seguir para cumplir los objetivos de la Institución en cuanto a seguridad informática se refiere. El alcance comprende a todo el personal docente, administrativo, de mantenimiento, alumnos, así como la aplicación de las políticas a todos los sistemas e infraestructura IT del Campus Miguel de Cervantes de la Universidad SEK Ecuador.

3.1.3.3.- Objetivos de seguridad:

- Controlar el uso de los sistemas IT del Campus Miguel de Cervantes para prevenir cualquier incidente que pueda ser dañino para las actividades de la Institución.
- Establecer directrices para el buen uso de los sistemas IT del Campus Miguel de Cervantes.

3.1.3.4.-Seguridad Organizaciónal

Art 1.- Los servicios de la red Instituciónal son de exclusivo uso académico, de investigación, técnicos y para gestiones administrativas.

Art 2.- El administrador de sistemas es el encargado de mantener en buen estado los servidores y demás equipos informáticos y en general toda la infraestructura de red de datos del Campus.

Art 3.- Los usuarios tendrán el acceso a Internet, siempre y cuando se cumplan los requisitos mínimos de seguridad para acceder a este servicio y se acaten las disposiciones de conectividad que el administrador de sistemas previamente haya realizado.

Art 4.- Las actividades académicas (clases, exámenes, prácticas, tareas, etc.) en los centros de cómputo, tienen la primera prioridad, por lo que a cualquier usuario utilizando otro servicio sin estos fines, se le podrá solicitar dejar libre la estación de trabajo, si así, fuera necesario.

3.1.3.4.1.- Control de activos

Art 5.- El administrador de sistemas será el único responsable por salvaguardar los activos físicos que se encuentren dentro de los centros de cómputo y los activos de información que se encuentren dentro de dichos equipos.

3.1.3.4.2.- Seguridad del personal

Art 6.- La Universidad Internacional SEK Ecuador no se hace responsable por daños causados provenientes de sus empleados a la información o activos de procesamiento, propiedad de la Institución, daños efectuados desde sus instalaciones de red a equipos informáticos externos.

3.1.3.4.3.- Reportes de incidencias de seguridad:

Art 7.- En caso de suscitarse un incidente de seguridad en la infraestructura de red o equipos del Campus Miguel de Cervantes, así sea por daño, modificación, actualización o cambio, éste será reportado al Gerente de la UISEK Ecuador por parte del administrador de red, que previamente deberá ser documentado realizando una revisión de los registros o Logs, con el objetivo de verificar la situación y dar una respuesta congruente y acorde al problema, ya sea está en el ámbito legal o cualquier situación administrativa.

3.1.3.5.- Seguridad Lógica

3.1.3.5.1.- Control de acceso

Art 8.- Son usuarios de la red Institucional los docentes, administrativos, alumnos, y toda aquella persona considerada como empleado y que utilice los servicios de la red Institucional de la Universidad Internacional SEK Ecuador.

Art 9.- Se asignará una cuenta de acceso a todo usuario de la red Institucional, siempre y cuando se identifique previamente el objetivo de su uso o permisos específicos a los que este accederá.

Art 10.- Los alumnos, son usuarios limitados, estos tendrán acceso únicamente a los servicios de Internet y recursos compartidos de la red.

Art 11.- Se consideran usuarios externos, a cualquier entidad o persona, que tenga una relación con la Institución fuera del ámbito de empleado o estudiante y siempre que tenga una vinculación con los servicios de la red Institucióonal.

Art 12.- El acceso a la red por parte de terceros es estrictamente restrictivo y permisible únicamente mediante firma impresa y documentación de aceptación de confidencialidad hacia la Institución y comprometido con el uso exclusivo del servicio para el que le fue provisto el acceso.

Art 13.- No se proporcionará el servicio solicitado por un usuario, departamento o facultad, sin antes haberse completado todos los procedimientos de autorización necesarios.

Art 14.- Se creará una cuenta temporal del usuario, en caso de olvido o extravío de información de la cuenta personal, para brindar al usuario que lo necesite, siempre y cuando se muestre un documento de identidad personal.

Art 15.- La longitud mínima de caracteres permisibles en una contraseña es de 6 caracteres, los cuales tendrán una combinación alfanumérica.

Art 16.- La longitud máxima de caracteres permisibles en una contraseña es de 12 caracteres, siendo esta una combinación de mayúsculas y minúsculas.

3.1.3.5.2.- Responsabilidades del usuario

Art 17.- El usuario es responsable exclusivo de mantener a salvo su contraseña.

Art 18.- El usuario será responsable del uso que haga de su cuenta de acceso a los sistemas o servicios tecnológicos.

Art 19.- El usuario es responsable de evitar la práctica de establecer contraseñas relacionadas con alguna característica de su persona o relacionado con su vida.

Art 20.- Cualquier usuario que encuentre un hueco o falla de seguridad en los sistemas informáticos de la Institución, está obligado a reportarlo al administrador de sistemas.

3.1.3.5.3.- Control de acceso a la red

Art 21.- El acceso a la red interna, se permitirá siempre y cuando se cumpla con los requisitos de seguridad necesarios, y éste será permitido mediante un mecanismo de autenticación.

Art 22.- Cualquier alteración del tráfico entrante o saliente a través de los dispositivos de acceso a la red, será motivo de verificación y tendrá como resultado directo la realización de una auditoria de seguridad.

Art 23.- El administrador de sistemas deberá emplear dispositivos de red para el bloqueo, enrutamiento, o el filtrado de tráfico evitando el acceso o flujo de información, no autorizada hacia la red interna o desde la red interna hacia el exterior.

Art 24.- Los accesos a la red interna, desde una red externa perteneciente a la Institución, se harán mediante un mecanismo de autenticación seguro y el tráfico entre ambas redes o sistemas será cifrado con una encriptación de 128 bit, esto es mediante una red privada virtual (VPN o Virtual Private Network)

Art 25.- Se registrara todo acceso a los dispositivos de red, mediante archivos de registro o Logs, de los dispositivos que provean estos accesos.

Art 26.- Se efectuara una revisión de Logs de los dispositivos de acceso a la red como mínimo cada 2 semanas

3.1.3.5.4.- Control de acceso al sistema operativo

Art 27.- Al terminar una sesión de trabajo en cualquier computador, el usuario deberá obligatoriamente cerrar su sesión.

3.1.3.5.5.- Servidores

Art 28.- El acceso a la configuración del sistema operativo de los servidores, es únicamente permitido al administrador de sistemas.

Art 29.- Todo servicio provisto o instalado en los servidores, correrá o será ejecutado bajo cuentas restrictivas, nunca bajo cuentas de administrador.

Art 30.- Los servidores, al igual que las estaciones de trabajo, tendrán instalado y configurado correctamente software antivirus actualizable y activada la protección en tiempo real.

3.1.3.6.- Seguridad Física

3.1.3.6.1.- Seguridad de los equipos

Art 31.- El cableado de red, se instalará físicamente separado de cualquier otro tipo de cables, sobre todo de corriente o energía eléctrica, para evitar interferencias.

Art 32.- Si los servidores presentan problemas de hardware, éstos deberán ser reparados por el administrador de sistemas, de no poder ser así, deberán ser retirados sus medios de almacenamiento.

Art 33.- Los equipos o activos críticos de información y proceso, deberán ubicarse en áreas aisladas y seguras, protegidas con un nivel de seguridad.

3.1.3.6.2.- Controles generales

Art 34.- Deberá llevarse un control exhaustivo del mantenimiento preventivo y otro para el mantenimiento correctivo que se les haga a los equipos.

Art 35.- Toda centro de cómputo debe poseer herramientas auxiliares (extintores, alarmas contra incendios, lámpara de emergencia), necesarias para salvaguardar los recursos tecnológicos y la información.

Art 36.- Toda visita a los centros de cómputo de procesamiento de datos (sala de servidores) deberá ser registrada mediante un formulario de accesos.

Art 37.- La sala o cuarto de servidores, deberá estar recubierta de material aislante o protegido contra el fuego, esta sala deberá ser utilizada únicamente por equipos de tipo servidor.

Art 38.- El suministro de energía eléctrica debe hacerse a través de un circuito exclusivo para los equipos de cómputo, o en su defecto el circuito que se utilice no debe tener conectados equipos que demandan grandes cantidades de energía.

Art 39.- El suministro de energía eléctrica debe estar debidamente polarizado, no siendo conveniente la utilización de polarizaciones locales de tomas de corriente, sino que debe existir una red de polarización.

Art 40.- Las instalaciones de las áreas de trabajo deben contar con una adecuada instalación eléctrica, y proveer del suministro de energía mediante una estación de alimentación ininterrumpida o UPS para poder proteger la información.

Art 41.- Los centros de cómputo deberán poseer información en carteles, sobre accesos, alimentos o cualquier otra actividad contraria a la seguridad de la misma.

3.1.3.7.- Seguridad Legal

3.1.3.7.1.- Auditoría de Sistemas

Art 42.- Se debe efectuar una auditoria de seguridad a los sistemas de acceso a la red, trimestral, enmarcada en pruebas de acceso tanto internas como externas, desarrolladas por personal técnico especializado o en su defecto personal capacitado en el área de seguridad.

Art 43.- Toda auditoria a los sistemas, estará debidamente aprobada, y tendrá el sello y firma de la gerencia de la Institución.

Art 44.- Cualquier acción que amerite la ejecución de una auditoria a los sistemas informáticos deberá ser documentada y establecido el objetivos de la misma, así como razones para su ejecución, personal involucrada en la misma y sistemas implicados.

Art 45.- La auditoria no deberá modificar en ningún momento el sistema de archivos de los sistemas implicados, en caso de haber necesidad de modificar algunos, se deberá hacer un respaldo formal del sistema o sus archivos.

Art 46.- Las herramientas utilizadas para la auditoria deberán estar separadas de los sistemas de producción y en ningún momento estas se quedaran al alcance de personal ajeno a la elaboración de la auditoria.

3.1.4.- Plan de contingencia ante desastres

3.1.4.1.- Resumen ejecutivo

La UISEK – Ecuador (Campus Miguel de Cervantes) busca poder reanudar operaciones en el menor tiempo posible instalando un mini – rack con lo mínimo necesario en una ubicación diferente dentro del campus, bajo el siguiente esquema:

- Activar el mayor número posible de equipos.
- Dar servicio a los procesos identificados como críticos, con el menor tiempo y restricciones posibles.
- Mantener el flujo de información permanente sobre el estado de la contingencia.

3.1.4.2.- Procesos críticos del negocio

En ésta sección se describirá los procesos críticos para la continuidad del negocio después de un desastre ocurrido en las instalaciones de la UISEK Campus Miguel de Cervantes, los cuales se definieron a partir de conversaciones y un análisis con el administrador de la red del Campus Miguel de Cervantes.

ÁREA DE NEGOCIO	VITAL(S/N)	PRIORIDAD (MAYOR = 1)
CCSISTEMAS1	S	1
CCSISTEMAS2	S	1
CCFINANZAS	S	1
CCCOMUNICACION	S	1
ADMINISTRACION	S	1
CCTELECOMUNICACIONES	S	1
ESTUDIO DE RADIO	S	1
ESTUDIO TV	S	1
LAB. AMBIENTAL	S	1
AREA DE SERVIDORES	S	1
CCBIBLIOTECA	S	2
DECANATO SIS.	S	2
DECANATO FIN.	S	2
DECANATO COM.	S	2
TUTORIAS FIN.	N	3
TUTORIAS AMB.	N	3
AULAS AMBIENTAL	N	4
ENFERMERIA	N	5

Tabla 3.1.4.2

3.1.4.3.- Definición de contingencia

Un Plan de contingencias es un instrumento de gestión para la buena gestión de las Tecnologías de la Información y las Comunicaciones en el dominio del soporte y el desempeño.

Este plan contiene las medidas técnicas, humanas y organizativas necesarias para garantizar la continuidad del negocio y las operaciones de la UISEK Campus Miguel de Cervantes.

3.1.4.4.- Criterio de desastre

Los siguientes criterios han sido establecidos de acuerdo a los procesos vitales del negocio, para identificar una situación de Contingencias en la UISEK Campus Miguel de Cervantes.

Cualquier interrupción en las operaciones del computador, que potencialmente pudiera exceder de un (1) día, desde el momento que una evaluación de la situación es realizada y una situación de Contingencia es declarada.

El número de días identificado aquí, se basa en el período de tiempo más corto, en que la pérdida de las funciones vitales afectan de manera significativa los procesos de negocio de la UISEK Campus Miguel de Cervantes.

- Fuego en la sala del computador.
- Daños causados por agua (inundaciones) y humo.
- Problemas eléctricos en las salas de cómputo.
- Problemas técnicos de hardware.

Un desastre que potencialmente pudiera impedir el acceso a la edificación de la UISEK Campus Miguel de Cervantes y por ende las operaciones del computador.

- Movimientos sísmicos
- Erupciones volcánicas.
- Otros desastres naturales.

3.1.4.5.- Autoridad para declarar un desastre

Las personas autorizadas para declarar la situación de contingencia son:

- **Líder:** Administrador de Sistemas
- **Alterno:** Decano de la facultad de Sistemas

3.1.4.6.- Equipos de trabajo y responsabilidades

Esta sección identifica a los equipos de personas involucradas en la recuperación de operaciones del negocio y las responsabilidades de cada miembro. Las actividades serán divididas en categorías de antes, durante y después de una situación de Contingencia.

Factores considerados para la confirmación de equipos:

- Todo equipo debe estar conformado por un líder y un alterno.
- Cada equipo debe contar con mínimo de 3 personas, incluidos el líder y el alterno.

- Ninguna persona debe estar participando en más de 1 equipo, si sus tareas son concurrentes.
- Todas las personas identificadas en el Plan de Contingencia deben conocer sus responsabilidades.

De ésta manera se minimiza las posibilidades de no operatividad de los equipos debido a la ausencia de sus integrantes y/o al desconocimiento de sus responsabilidades.

En la próxima sección se presentan los equipos, con sus respectivos integrantes y se presenta la información relativa a los números telefónicos y direcciones de cada miembro de éstos.

3.1.4.6.1.- Equipo Gerencial

El equipo gerencial es el que se encarga de evaluar y declarar la situación de contingencia.

- **Líder:** Administrador de Sistemas
- **Alterno:** Decano de la faultad de Sistemas

3.1.4.6.1.1.- Responsabilidades Previas a un Desastre

- Comprender el criterio de desastre del plan de contingencia.
- Mantener actualizado el plan de contingencia.
- Coordinar la realización de las diferentes pruebas anuales.

3.1.4.6.1.2.- Responsabilidades Durante un Desastre

- Estudiar y evaluar el alcance del desastre en cada área.
- Establecer seguridad en el área afectada.
- Coordinar las actividades con las compañías de seguros, policías, bomberos y servicios médicos.
- Determinar si el área afectada es segura para el personal.
- Proveer facilidades al personal encargado de la recuperación.
- Asegurar que los equipos de trabajo realicen las tareas asignadas.
- Proveer transporte para equipos, personas y suministros.
- Coordinar pagos de facturas.

3.1.4.6.1.3.- Responsabilidades Después de un Desastre

- Vigilar por la actualización del plan en caso de haber surgido modificaciones durante la situación de contingencia.

3.1.4.6.2.- Equipo de seguridad

- **Líder:** Decano de otra facultad
- **Alterno:** Decano de otra facultad

3.1.4.6.2.1.- Responsabilidades Previas a un Desastre

- Definir estrategia para proteger los activos de la universidad, tanto humanos como materiales.

- Identificar y evaluar riesgos que amenacen la continuidad operacional de la universidad.
- Formular políticas para la protección y resguardo del patrimonio de la universidad.
- Elaborar y mantener un plan de respuesta a incidentes.

3.1.4.6.2.2.- Responsabilidades Durante un Desastre

- Facilitar la evacuación de los ocupantes de las instalaciones de la universidad.
- Reportar el estado de situación de la emergencia al Rectorado de la universidad.
- Proteger que los bienes de la universidad y controlar el acceso de los predios de la misma.
- Brindar apoyo logístico a los equipos de trabajo de la contingencia y coordinar los recursos para la reanudación de las operaciones de la misma.

3.1.4.6.2.3.- Responsabilidades Después de un Desastre

- Evaluar los daños y coordinar operaciones de restauración.
- Proteger las áreas afectadas.
- Realizar la actualización de los planes de emergencias.

3.1.4.7.- Procedimientos de respaldo para contingencia

Actualmente existe un solo tipo de respaldo: Semanal.

En éste proceso semanal se realiza un respaldo de todo el sistema operativo del servidor de Internet.

3.1.4.8.- Requerimientos de Hardware y de Software

Se ha estimado que un equipo con las siguientes características, sería suficiente para restaurar los procesos vitales del negocio de la Universidad SEK Campus Miguel de Cervantes:

- Procesador Intel Pentium 4 (Como mínimo 2 Ghz de velocidad de reloj)
- Memoria RAM de 1GB
- 1 Unidad de CD-ROM
- 1 Unidad de DVD-ROM
- 2 tarjetas de red internas Ethernet 10/100 Mbps
- 1 Disco Duro con capacidad de 80 GB (Mínimo)

El software requerido es:

- Sistema Operativo Linux (Kernel 2.6.X)
- Webmin
- Squid
- Iptables
- DHCP

3.1.4.9.- Procedimiento de restauración

La secuencia lógica de restauración para la nueva PC es la siguiente:

- Instalación del sistema operativo Linux
- Configuración de tarjetas de red (IP pública y privada)
- Instalación de webmin
- Instalación de SQUID
- Instalación de Firewall
- Instalación de servicio DHCP

3.1.4.10.- Implementación del plan

El propósito de ésta sección es seguir paso a paso el procedimiento para declarar una situación de contingencia y dar paso a la recuperación de los procesos vitales del negocio.

Se mostrarán varias listas de tareas que tendrán que ser ejecutadas para que el plan de contingencia sea exitoso.

3.1.4.10.1.- Evaluación y declaración de contingencia

TAREA (EV = EVALUACIÓN)	DESCRIPCIÓN
EV1	Establecer seguridad en el área afectada según (Equipo de seguridad)
EV2	Reportar situación de daños al Líder del equipo Gerencial
EV3	Equipo Gerencial convocará a todos los demás equipos a una reunión
EV4	Equipo Gerencial verificará la asistencia de todos los equipos, así como de sus miembros
EV5	Evaluar daños y estimar tiempo de fuera de servicio
EV6	Realizar declaración de contingencia a una autoridad superior (Rectorado)
EV7	Entregar un informe preliminar al Rectorado sobre evaluación de daños

Tabla 3.1.4.10.1

3.1.4.10.2.- Instalación de nuevo Rack en ubicación alterna

TAREA (IN = INSTALACIÓN)	DESCRIPCIÓN
IN1	Equipo gerencial trasladará equipos sin daño mayor a la nueva ubicación del Rack (Facultad de Ciencias Ambientales, Piso 1)
IN2	Instalación de 1 switch con puerto de fibra óptica (1 ya existente) en nuevo Rack.
IN3	Instalación de 1 switch con puerto de fibra óptica en Facultad de Finanzas en ubicación alterna al daño
IN4	Instalación de 1 switch con puerto de fibra óptica en Facultad de comunicación en ubicación alterna al daño
IN5	Conexión de enlace de fibra óptica alterno #1, entre switch #1 (nuevo Rack) y switch de Facultad de Finanzas
IN6	Conexión de enlace de fibra óptica alterno #2, entre switch #2 (nuevo Rack) y switch de Facultad de Comunicación

TAREA (IN = INSTALACIÓN)	DESCRIPCIÓN
IN7	Instalación del nuevo servidor de contingencia en nuevo cuarto de servidores (Facultad de Ciencias Ambientales) (Detalle: Ver Capítulos 11 y 12)
IN8	Conexión de Banda Ancha existente (Access RAM) al nuevo servidor de contingencia
IN9	Coordinación del equipo de seguridad para brindar protección al nuevo cuarto de servidores (Facultad de Ciencias Ambientales)
IN10	Verificación de todas las instalaciones y configuraciones anteriores
IN11	Inicio de las operaciones de negocio

Tabla 3.1.4.10.2

3.1.4.11.- Mantenimiento del Plan de Contingencia

El propósito de esta sección es definir el esquema de actualización del Plan de Contingencia de la UISEK – Ecuador (Campus Miguel de Cervantes).

El mantenimiento del plan tiene un nivel de importancia crítico para poder ejecutarlo exitosamente en una situación de contingencia, de otra manera, puede convertirse en perjudicial para la Institución.

- **Cambios en la Organización:** Conformación de los equipos de trabajo definidos en el plan, y cambios en la estructura Organizaciónal de la Institución que afecten a la vigencia de la información y las responsabilidades asociadas a los roles establecidos en cada equipo.

- **Cambios en los procedimientos de recuperación:** Cambios y/o modificaciones en los procedimientos de recuperación y en los recursos de Hardware y Software existentes.

3.1.4.12.- Recomendaciones para ejecutar el Plan de contingencia (IMPORTANTE)

- Instalación de 2 nuevos enlaces alternos de fibra óptica que vayan desde la Facultad de Ciencias Ambientales hasta cada una de las otras facultades.
- Compra de 2 switches alternos con puertos de fibra óptica y guardarlos en un lugar seguro.

3.1.5.- Metodología propuesta

En la UISEK Ecuador Campus Miguel de Cervantes no existen servicios críticos o información sensible que estén albergados en los sistemas IT de la Institución, es por esto que se propone aplicar el método Baseline Approach para analizar los riesgos de seguridad a la que la Institución es propensa. Ésta metodología propone guías generales para la selección de contramedidas de seguridad que mas se adecuen a los sitemas IT de la Organización. Éste tipo de metodología se aplica en Organizaciónes en la que los sistemas IT no requieran un nivel alto de seguridad debido a que éstos no tienen un rol crítico que pueda afectar el negocio significativamente, incluso no es necesario realizar un análisis de riesgos exhaustivo, sin embargo, para poner en conocimeinto el estado de seguridad actual de los sistemas IT del Campus Miguel de Cervantes se realizará el análisis de riesgos.

El proceso de realizar el análisis de riesgos se compone de varias etapas:

- Identificación de activos
- Valoración y dependencias de activos
- Análisis de amenazas

- Análisis de vulnerabilidades
- Análisis de riesgos

3.1.6.- Identificación de activos

Se consideran activos importantes dentro del ámbito informático de la UISEK ECUADOR Campus Miguel de Cervantes, según conversaciones y análisis conjunto con el administrador de sistemas:

CÓDIGO (AC = ACTIVO)	ACTIVOS IMPORTANTES
AC1	Servidor de Internet
AC2	Servidor de dominio
AC3	Switches de enlace a los edificios de facultades colindantes
AC4	Switches de enlace a todos los centros de cómputo del ed. principal
AC5	Ruteador de Internet
AC6	Computadora de asistente administrativa
AC7	Enlace de fibra óptica al edificio de Facultad de Medio Ambiente
AC8	Enlace de par trenzado apantallado al edificio de Facultad de Comunicación

Tabla 3.1.6

3.1.7.- Valoración de activos

Es importante valorar los activos según su importancia en el negocio, la valoración se realiza según el impacto negativo que tendría la pérdida, daño o destrucción de dicho activo. *(Refiérase a la tabla 3.1.6 para identificar los activos)*

CÓDIGO (AC = ACTIVO)	VALORACIÓN
AC1	Alto
AC4	Alto
AC5	Alto
AC7	Alto
AC8	Alto
AC6	Medio
AC3	Medio
AC2	Bajo

Tabla 3.1.7

3.1.8.- Análisis de amenazas

Siempre existen amenazas latentes donde exista una infraestructura de sistemas IT, es importante identificar la probabilidad de que éstas ocurran en nuestro entorno, de qué forma ocurrirían y a que activos afectarían.

La escala a usarse para las causas de las amenzas es A (accidental), M (ambiental), D (deliberado), y ALTA, MEDIA, BAJA para la probabilidad de que éstas ocurran

PROBABILIDAD Y CAUSA DE EXPLOTACIÓN DE AMENAZAS		
AMENAZA	CAUSA	PROBABILIDAD
Terremoto	M	Media
Inundación	A,M,D	Alta
Relámpagos	M	Media
Bomba	A,D	Baja

PROBABILIDAD Y CAUSA DE EXPLOTACIÓN DE AMENAZAS		
Armas De Fuego	A,D	Media
Fuego	A,M,D	Alta
Daño Voluntario	D	Alta
Falla De Suministro Eléctrico	A	Media
Fallo De Hardware	A	Media
Variación De Voltaje	A	Alta
Temperatura Muy Alta	A,M	Alta
Humedad Muy Alta	M	Baja
Carga Electrostática	A,D	Media
Robo	D	Media
Error Operacional	A,D	Media
Error De Mantenimiento	A,D	Media
Error De Software	A,D	Media
Software Mal Utilizado	A,D	Media
Software Instalado Por Personas No Autorizadas	D	Alta
Suplantación De Identidad	D	Alta
Software Mailicioso	D	Alta
Acceso A La Red Por Personas No Autorizadas	D	Alta
Sobrecarga De Tráfico	A,D	Media
Fallo De Componentes De Red	A	Alta
Mal Uso De Los Recursos	A,D	Media
Falta De Personal	A	Baja

Tabla 3.1.8.1

Estos serían los activos afectados en el supuesto caso de que una amenaza se convierta en un hecho real:

ACTIVOS AFECTADOS POR EXPLOTACIÓN DE AMENAZAS	
AMENAZA	**ACTIVOS AFECTADOS**
Terremoto	Todos

ACTIVOS AFECTADOS POR EXPLOTACIÓN DE AMENAZAS	
Inundación	Servidores, Switches, Equipo Asistente Adm., Ruteador
Relámpagos	Servidores, Switches, Ruteador
Bomba	Todos
Armas de fuego	Servidores, Switches, Ruteador
Fuego	Todos
Daño voluntario	Todos
Falla de suministro eléctrico	Servidores, Switches, Equipo Asistente Adm, Ruteador
Fallo de hardware	Todos
Variación de voltaje	Servidores, Switches, Equipo Asistente Adm, Ruteador
Temperatura muy alta	Servidores, Switches
Humedad muy alta	Servidores, Switches
Carga electrostática	Servidores, Switches, Equipo Asistente Adm.
Robo	Todos
Error operacional	Servidores, Switches, Equipo Asistente Adm.
Error de mantenimiento	Servidores, Switches, Equipo Asistente Adm.
Error de software	Servidores, Equipo Asistente Adm.
Software mal utilizado	Servidores, Equipo Asistente Adm.

Tabla 3.1.8.2

3.1.9.- Análisis de vulnerabilidades

Existen acciones de toda índole que pueden explotar ciertas vulnerabilidades, en diferentes ámbitos:

AMBIENTE E INFRAESTRUCTURA	
VULNERABILIDAD	**ACCIÓN QUE LA EXPLOTARÍA**
Carencia de protección física de las puertas de los centros de cómputo	Robo

AMBIENTE E INFRAESTRUCTURA	
Control de acceso inadecuado al Campus	Daño voluntario, Robo
Área propensa a inundación	Inundación
Carencia de detectores de humo	Fuego
Carencia de sistema de control de temperatura	Temperatura muy alta
Susceptibilidad a polvo	Polvo
Susceptibilidad a los ataques internos	Software instalado por personas no autorizadas

Tabla 3.1.9.1

HARDWARE	
VULNERABILIDAD	**ACCIÓN QUE LA EXPLOTARÍA**
Fallos de hardware por variación de temperatura	Temperatura muy alta
Fallos de hardware por voltaje inestable	Variaciones de voltaje
Susceptibilidad a la radiación electromagnética	Manejor erróneo al instalar componentes de hardware

Tabla 3.1.9.2

COMUNICACIONES	
VULNERABILIDAD	**ACCIÓN QUE LA EXPLOTARÍA**
Carencia de autenticación en la red	Acceso a la red por personas no autorizadas
Transferencia de contraseñas	Suplantación de identidad
Inadecuada administración de la red	Sobrecarga de tráfico
Red pública no protegida	Uso de software malicioso

Tabla 3.1.9.3

PERSONAL	
VULNERABILIDAD	**ACCIÓN QUE LA EXPLOTARÍA**
Ausencia de personal	Falta de personal
Trabajo de terceros no supervisado	Robo

PERSONAL	
Entrenamiento de seguridad insuficiente	Error operacional
Uso incorrecto de hardware y software	Error operacional, Mal uso de los recursos

Tabla 3.1.9.4

3.1.10.- Análisis de riesgos

SERVIDOR DE INTERNET				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Temperatura muy alta	5	5	25	1
Daño voluntario	5	4	20	2
Variación de voltaje	5	4	20	3
Suplantación de identidad	5	4	20	4
Software mailicioso	5	4	20	5
Falla de suministro eléctrico	4	4	16	6
Fallo de hardware	4	4	16	7
Software instalado por personas no autorizadas	5	3	15	8
Acceso a la red por personas no autorizadas	5	3	15	9
Sobrecarga de tráfico	5	3	15	10
Robo	4	3	12	11
Fallo de componentes de red	3	4	12	12
Inundación	5	2	10	13
Fuego	5	2	10	14
Humedad muy alta	5	2	10	15
Error de software	5	2	10	16
Error operacional	4	2	8	17
Error de mantenimiento	4	2	8	18
Software mal utilizado	4	2	8	19
Mal uso de los recursos	3	2	6	20

SERVIDOR DE INTERNET				
Terremoto	5	1	5	21
Bomba	5	1	5	22
Relámpagos	3	1	3	23
Armas de fuego	3	1	3	24
Carga electrostática	3	1	3	25

Tabla 3.1.10.1

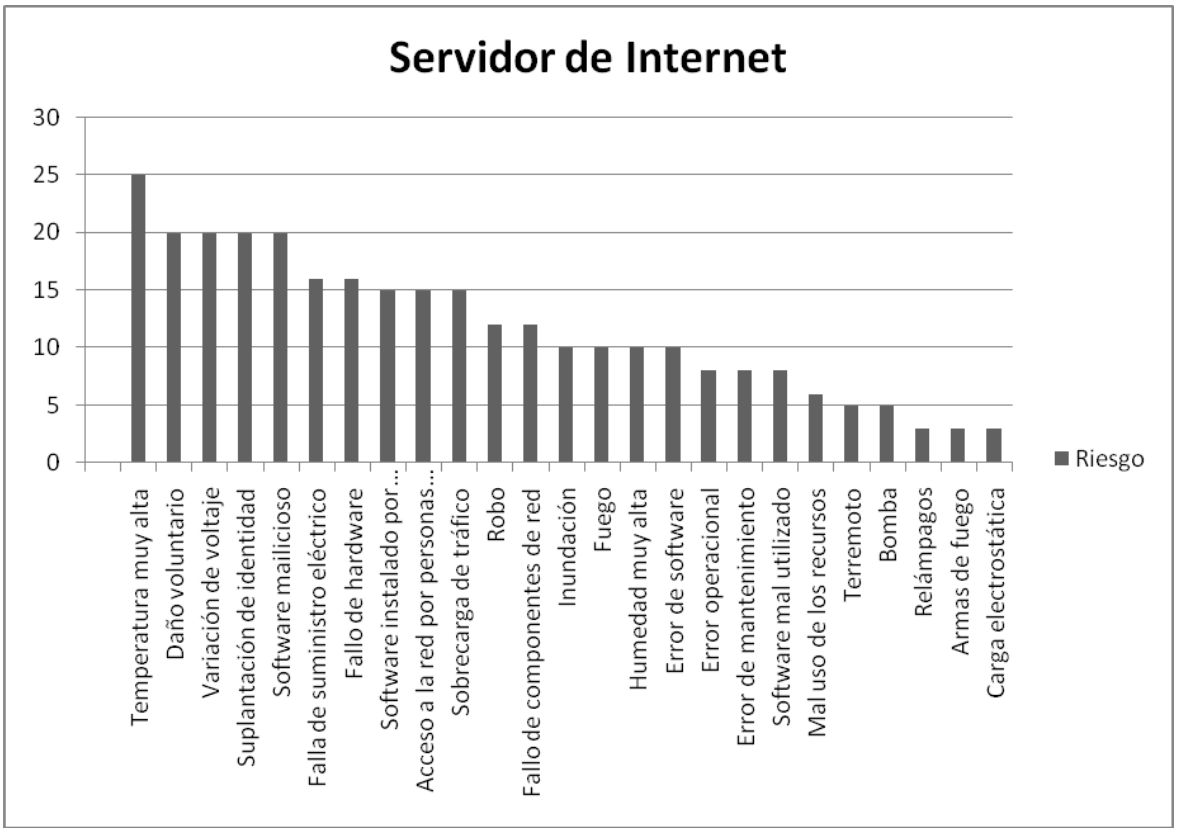


Figura 3.1.10.1

SERVIDOR DE DOMINIO				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Temperatura muy alta	5	5	25	1
Suplantación de identidad	5	5	25	2
Daño voluntario	5	4	20	3
Software mailicioso	5	4	20	4

SERVIDOR DE DOMINIO				
Software instalado por personas no autorizadas	5	4	20	5
Acceso a la red por personas no autorizadas	5	4	20	6
Falla de suministro eléctrico	4	4	16	7
Variación de voltaje	5	3	15	8
Fallo de hardware	4	3	12	9
Robo	4	3	12	10
Error operacional	4	3	12	11
Error de mantenimiento	4	3	12	12
Inundación	5	2	10	13
Fuego	5	2	10	14
Relámpagos	5	2	10	15
Sobrecarga de tráfico	3	3	9	16
Error de software	4	2	8	17
Software mal utilizado	4	2	8	18
Mal uso de los recursos	4	2	8	19
Armas de fuego	4	2	8	20
Carga electrostática	4	2	8	21
Fallo de componentes de red	3	2	6	22
Humedad muy alta	5	1	5	23
Terremoto	5	1	5	24
Bomba	5	1	5	25

Tabla 3.1.10.2

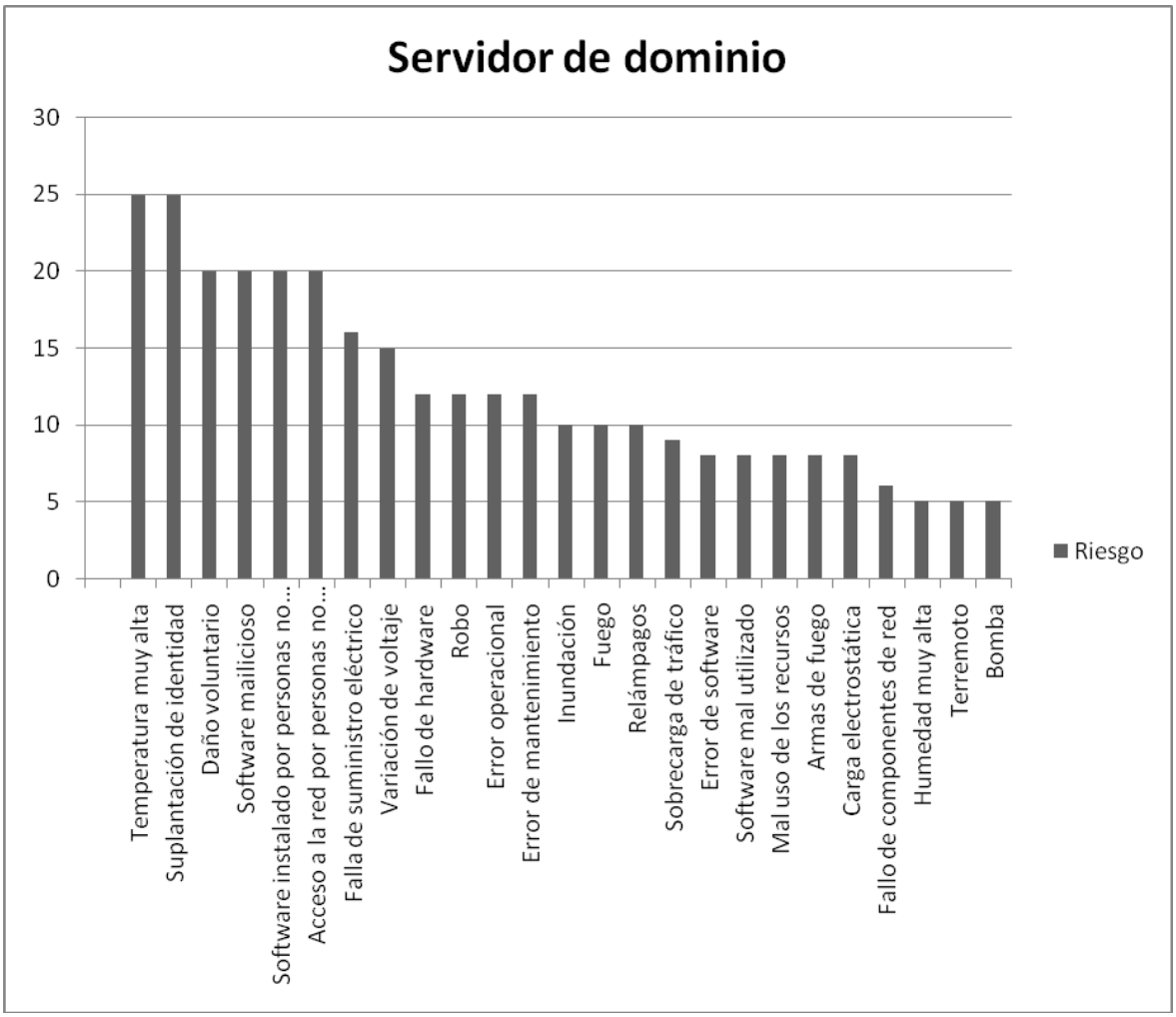


Figura 3.1.10.2

SWITCHES DE ENLACE A FACULTADES COLINDANTES				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Temperatura muy alta	5	5	25	1
Variación de voltaje	4	4	16	2
Robo	5	3	15	3
Inundación	5	3	15	4
Carga electrostática	4	3	12	5
Daño voluntario	5	2	10	6
Fallo de hardware	5	2	10	7
Fuego	5	2	10	8
Relámpagos	5	2	10	9
Fallo de componentes de red	5	2	10	10

SWITCHES DE ENLACE A FACULTADES COLINDANTES				
Mal uso de los recursos	3	3	9	11
Falla de suministro eléctrico	4	2	8	12
Error operacional	4	2	8	13
Error de mantenimiento	3	2	6	14
Sobrecarga de tráfico	2	3	6	15
Armas de fuego	5	1	5	16
Terremoto	5	1	5	17
Bomba	5	1	5	18
Software mailicioso	1	4	4	19
Acceso a la red por personas no autorizadas	1	4	4	20
Suplantación de identidad	1	3	3	21
Software instalado por personas no autorizadas	1	3	3	22
Error de software	1	3	3	23
Software mal utilizado	1	2	2	24
Humedad muy alta	1	1	1	25

Tabla 3.1.10.3

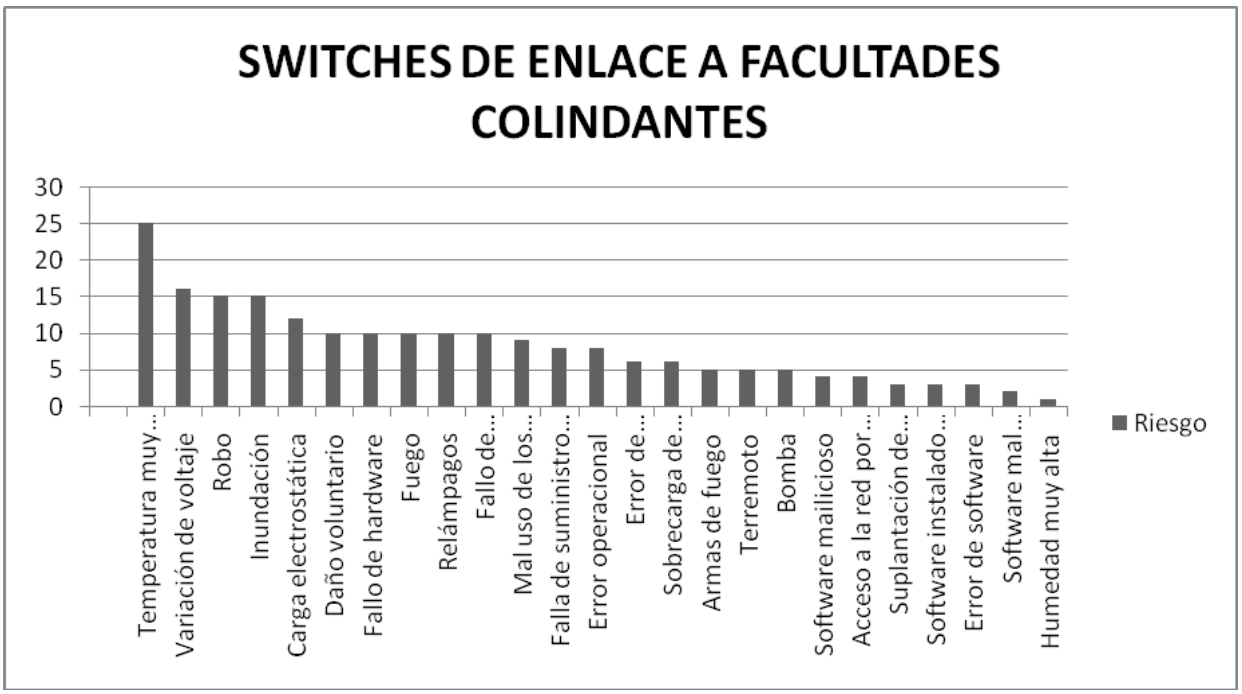


Figura 3.1.10.3

SWITCHES DE ENLACE A CENTROS DE CÓMPUTO ED. PRINCIPAL				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Temperatura muy alta	5	5	25	1
Variación de voltaje	4	4	16	2
Robo	5	3	15	3
Inundación	5	3	15	4
Carga electrostática	4	3	12	5
Daño voluntario	5	2	10	6
Fallo de hardware	5	2	10	7
Fuego	5	2	10	8
Relámpagos	5	2	10	9
Fallo de componentes de red	5	2	10	10
Mal uso de los recursos	3	3	9	11
Falla de suministro eléctrico	4	2	8	12
Error operacional	4	2	8	13
Error de mantenimiento	3	2	6	14
Sobrecarga de tráfico	2	3	6	15
Armas de fuego	5	1	5	16
Terremoto	5	1	5	17
Bomba	5	1	5	18
Software mailicioso	1	4	4	19
Acceso a la red por personas no autorizadas	1	4	4	20
Suplantación de identidad	1	3	3	21
Software instalado por personas no autorizadas	1	3	3	22
Error de software	1	3	3	23
Software mal utilizado	1	2	2	24
Humedad muy alta	1	1	1	25

Tabla 3.1.10.4

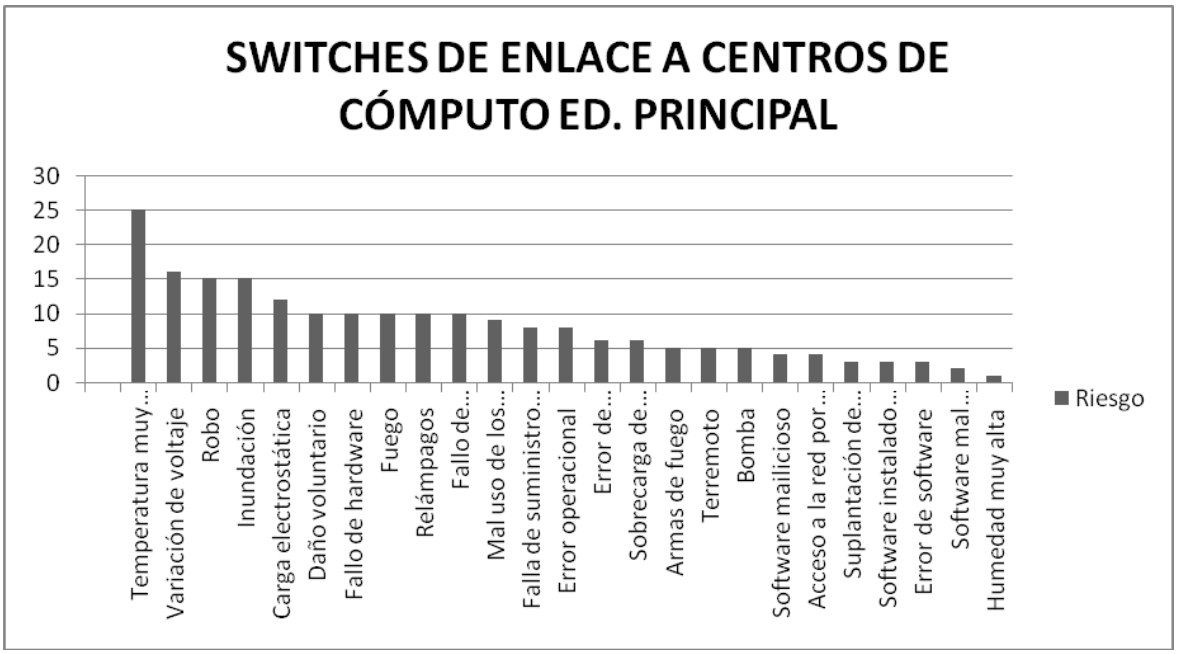


Figura 3.1.10.4

RUTEADOR DE INTERNET				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Temperatura muy alta	5	5	25	1
Variación de voltaje	4	4	16	2
Software mailicioso	4	4	16	3
Daño voluntario	5	3	15	4
Fuego	5	3	15	5
Relámpagos	5	3	15	6
Acceso a la red por personas no autorizadas	3	4	12	7
Robo	5	2	10	8
Carga electrostática	5	2	10	9
Falla de suministro eléctrico	5	2	10	10
Fallo de hardware	4	2	8	11
Mal uso de los recursos	4	2	8	12
Error operacional	4	2	8	13
Error de mantenimiento	4	2	8	14
Suplantación de identidad	2	3	6	15
Software instalado por personas no autorizadas	2	3	6	16

RUTEADOR DE INTERNET				
Inundación	5	1	5	17
Armas de fuego	5	1	5	18
Terremoto	5	1	5	19
Bomba	5	1	5	20
Humedad muy alta	5	1	5	21
Fallo de componentes de red	2	2	4	22
Sobrecarga de tráfico	2	2	4	23
Error de software	2	2	4	24
Software mal utilizado	2	1	2	25

Tabla 3.1.10.5

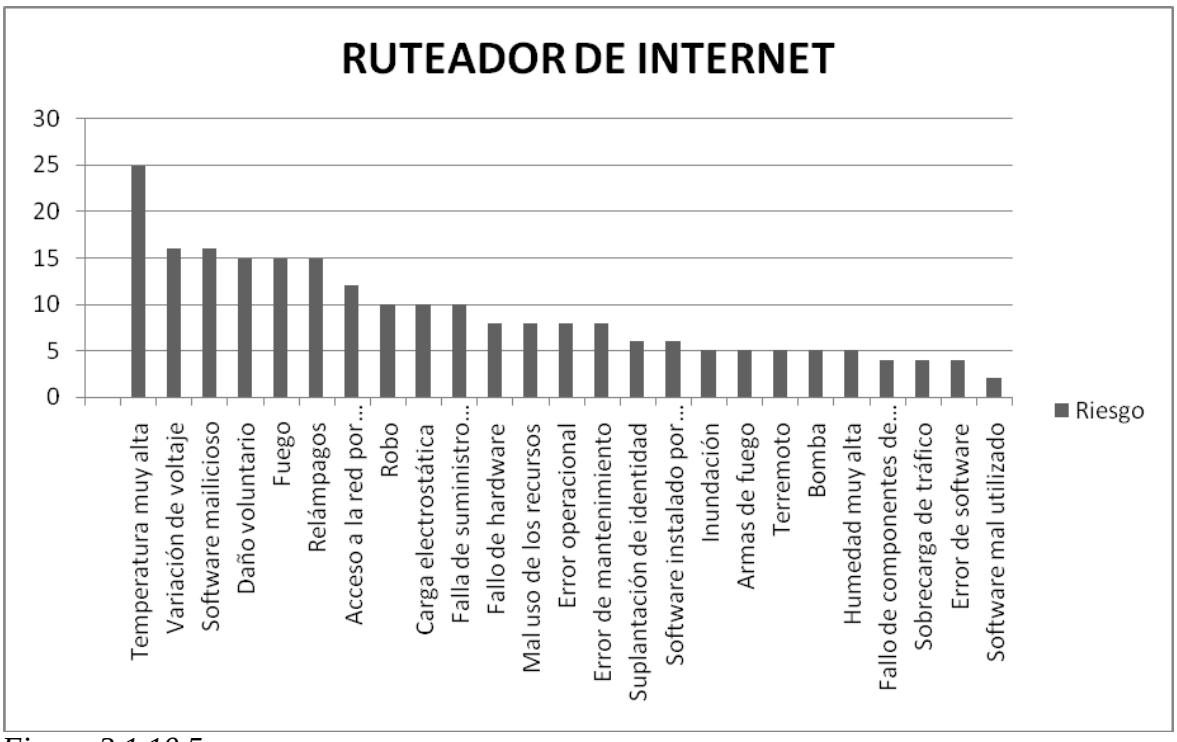


Figura 3.1.10.5

ENLACE DE FIBRA ÓPTICA A FAC. MEDIO AMBIENTE				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Daño voluntario	5	4	20	1
Robo	5	3	15	2
Mal uso de los recursos	5	2	10	3
Temperatura muy alta	5	1	5	4
Fuego	5	1	5	5
Error operacional	5	1	5	6
Error de mantenimiento	5	1	5	7
Inundación	5	1	5	8
Terremoto	5	1	5	9
Bomba	5	1	5	10
Acceso a la red por personas no autorizadas	1	4	4	11
Variación de voltaje	3	1	3	12
Software mailicioso	1	3	3	13
Relámpagos	1	3	3	14
Carga electrostática	1	3	3	15
Falla de suministro eléctrico	1	3	3	16
Fallo de hardware	1	3	3	17
Suplantación de identidad	1	3	3	18
Error de software	1	3	3	19
Software instalado por personas no autorizadas	1	2	2	20
Armas de fuego	2	1	2	21
Fallo de componentes de red	1	2	2	22
Sobrecarga de tráfico	1	2	2	23
Software mal utilizado	1	2	2	24
Humedad muy alta	1	1	1	25

Tabla 3.1.10.6

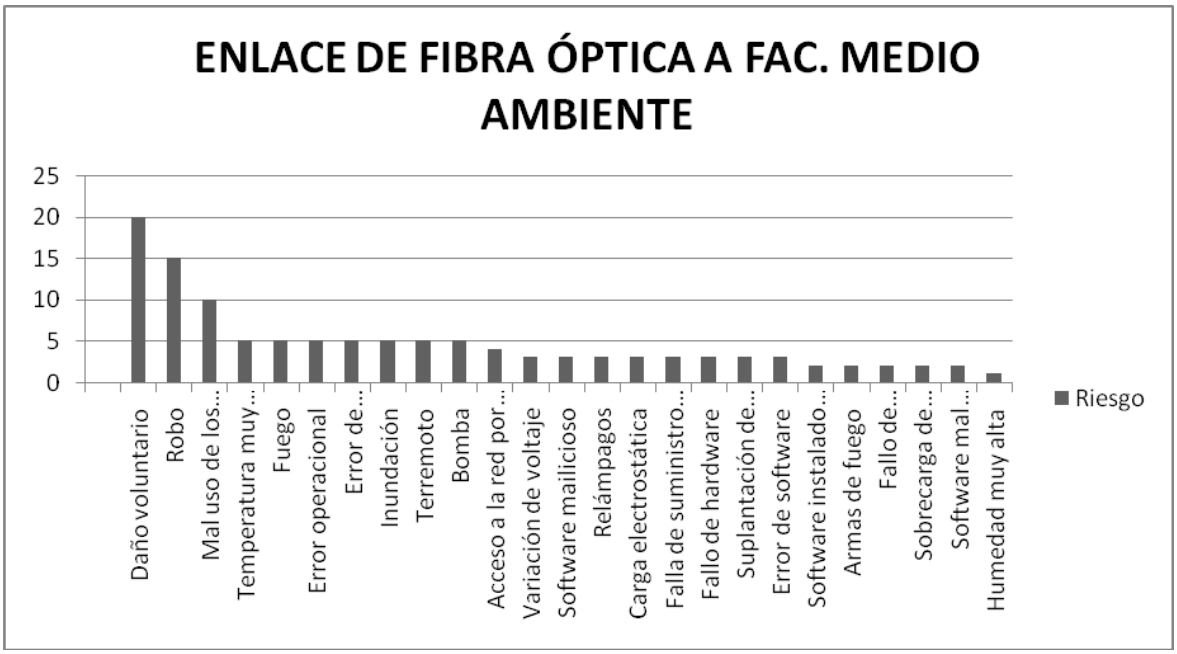


Figura 3.1.10.6

ENLACE DE PAR TRENZADO A FAC. COMUNICACIÓN				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Daño voluntario	5	4	20	1
Robo	5	3	15	2
Mal uso de los recursos	5	2	10	3
Temperatura muy alta	5	1	5	4
Fuego	5	1	5	5
Error operacional	5	1	5	6
Error de mantenimiento	5	1	5	7
Inundación	5	1	5	8
Terremoto	5	1	5	9
Bomba	5	1	5	10
Acceso a la red por personas no autorizadas	1	4	4	11
Variación de voltaje	3	1	3	12
Software mailicioso	1	3	3	13
Relámpagos	1	3	3	14
Carga electrostática	1	3	3	15
Falla de suministro eléctrico	1	3	3	16

ENLACE DE PAR TRENZADO A FAC. COMUNICACIÓN				
Fallo de hardware	1	3	3	17
Suplantación de identidad	1	3	3	18
Error de software	1	3	3	19
Software instalado por personas no autorizadas	1	2	2	20
Armas de fuego	2	1	2	21
Fallo de componentes de red	1	2	2	22
Sobrecarga de tráfico	1	2	2	23
Software mal utilizado	1	2	2	24
Humedad muy alta	1	1	1	25

Tabla 3.1.10.7

Figura 3.1.10.7

COMPUTADORA DE ASISTENTE ADMINISTRATIVA				
Amenaza	**Valor de activo (A)**	**Probab. ocurrencia (B)**	**Riesgo (A x B)**	**Ranking**
Suplantación de identidad	5	5	25	1
Daño voluntario	5	4	20	2
Acceso a la red por personas no autorizadas	5	4	20	3
Software mailicioso	5	4	20	4

COMPUTADORA DE ASISTENTE ADMINISTRATIVA				
Variación de voltaje	5	3	15	5
Software instalado por personas no autorizadas	5	3	15	6
Fallo de componentes de red	5	3	15	7
Fallo de hardware	4	3	12	8
Robo	5	2	10	9
Mal uso de los recursos	5	2	10	10
Falla de suministro eléctrico	5	2	10	11
Error operacional	4	2	8	12
Error de mantenimiento	4	2	8	13
Relámpagos	4	2	8	14
Carga electrostática	4	2	8	15
Error de software	4	2	8	16
Software mal utilizado	4	2	8	17
Sobrecarga de tráfico	3	2	6	18
Fuego	5	1	5	19
Inundación	5	1	5	20
Terremoto	5	1	5	21
Bomba	5	1	5	22
Armas de fuego	5	1	5	23
Temperatura muy alta	2	1	2	24
Humedad muy alta	1	1	1	25

Tabla 3.1.10.8

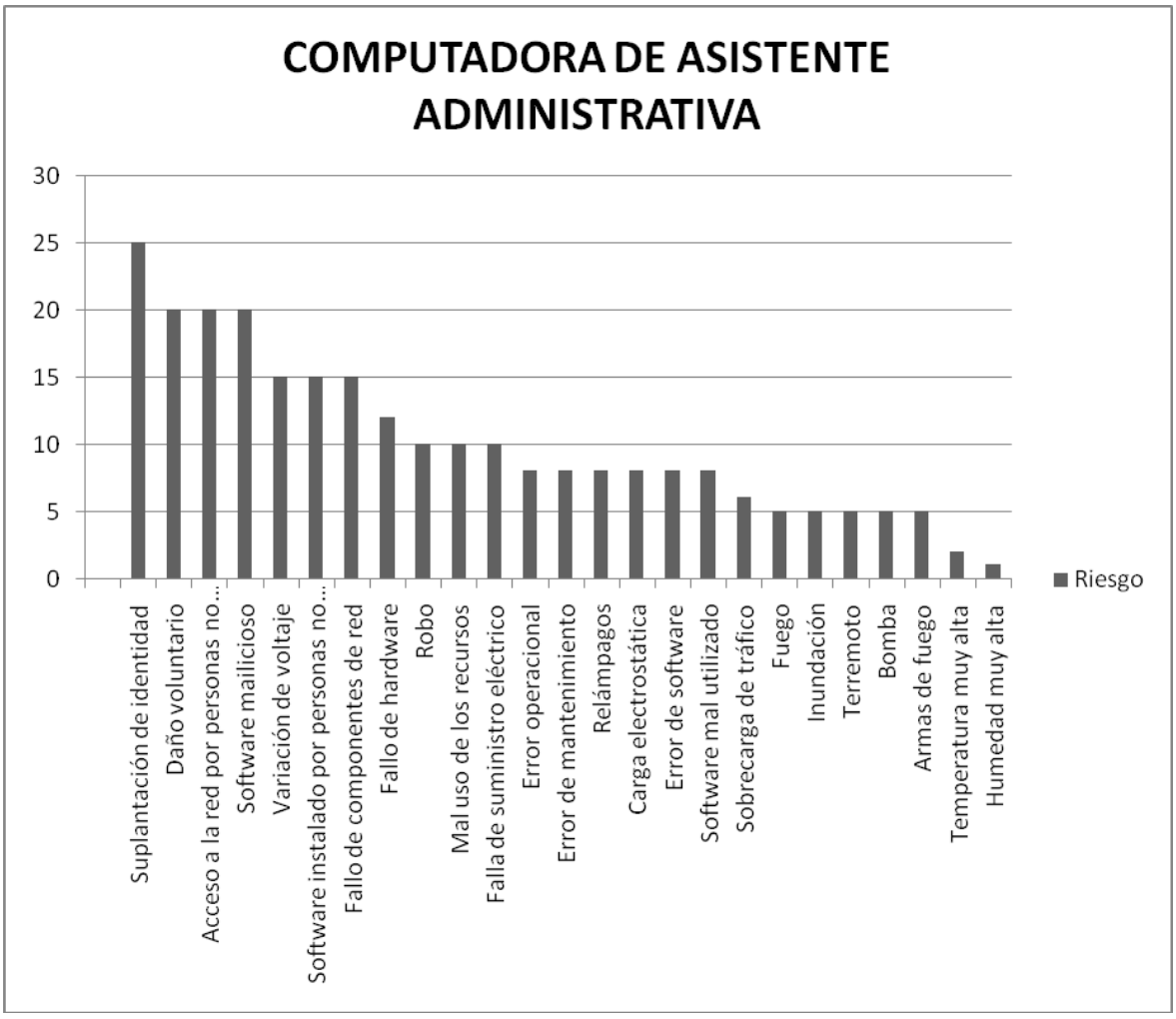


Figura 3.1.10.8

3.2.- SELECCIÓN DE CONTRAMEDIDAS DE SEGURIDAD SEGÚN LA NORMA ISO/IEC 13335:4 “SELECCIÓN DE MEDIDAS DE SEGURIDAD”

3.2.1.- Esquema general:

La red de datos está distribuida en 3 edificios:

- Edificio principal
- Facultad de Medio Ambiente
- Facultad de Comunicación

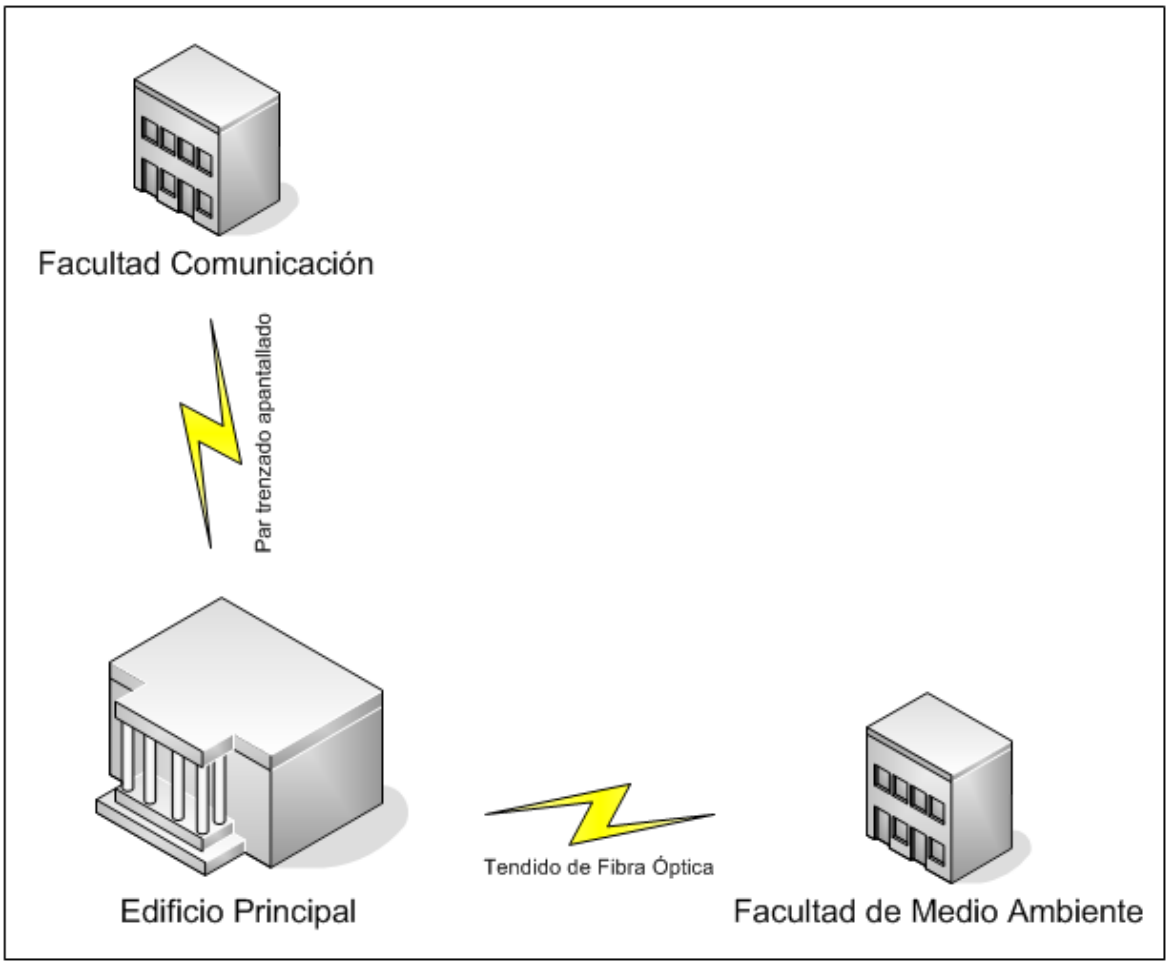


Figura 3.2.1

3.2.2.- Topología de red existente:

En el Campus Miguel de Cervantes de la Universidad Internacional SEK Ecuador existe una red de datos de tipo LAN, ésta es de topología de estrella, donde todas las estaciones de trabajo y servidores están conectadas independientemente a un punto en común que son los concentradores o switches.

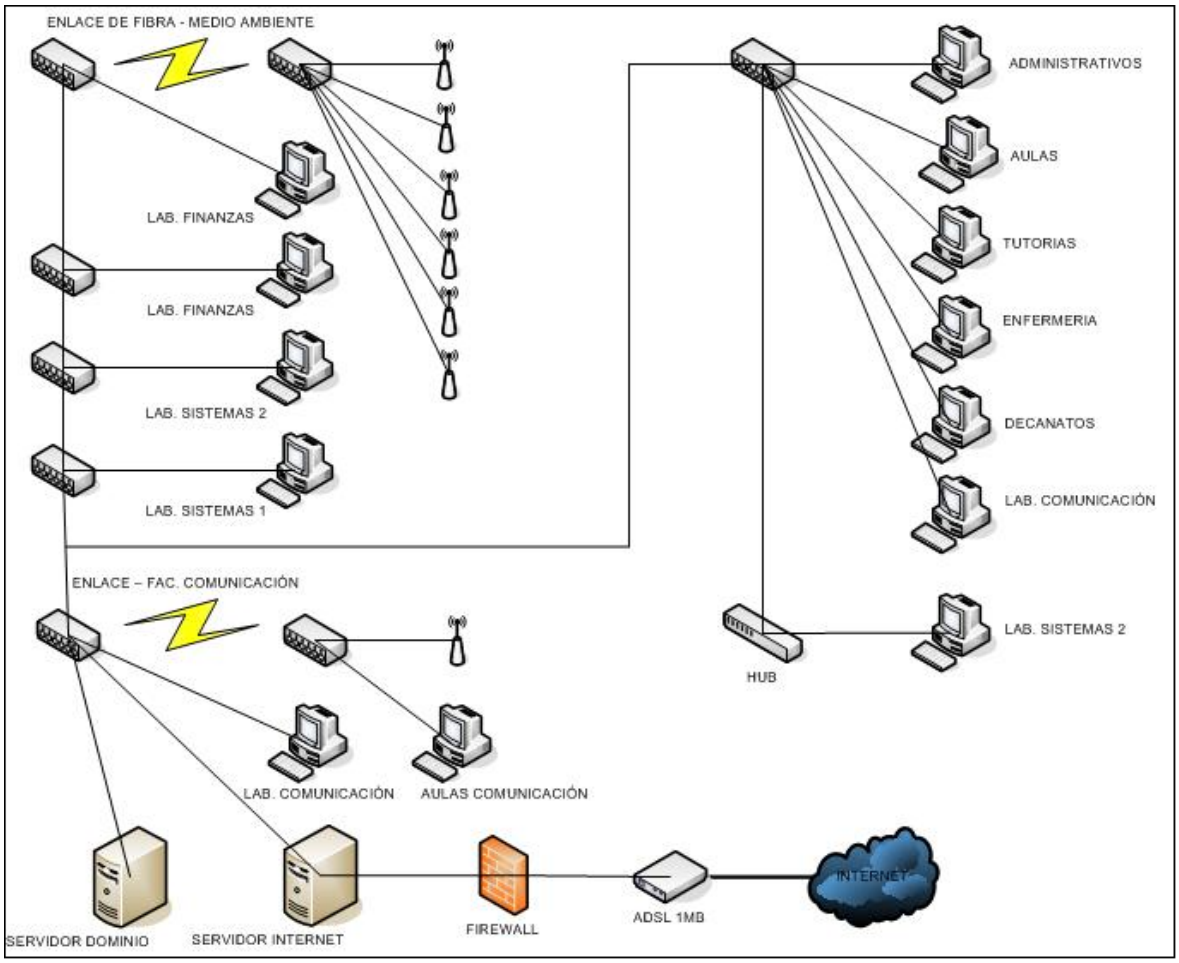


Figura 3.2.2

3.2.3.- Perímetro físico:

Los 3 edificios pertenecientes a la UISEK Campus Miguel de Cervantes se encuentran ubicados en el sector de Carcelén, en las calles Alberto Einstein s/n y 5ta transversal. El terreno está delimitado enteramente por una cerca de malla de acero y está alejado de los demás edificios colindantes y del tráfico vehicular.

Los edificios de la UISEK Campus Miguel de Cervantes son de múltiple ocupación, es decir, además de albergar la infraestructura y servidores informáticos también se ocupan para dar clases a estudiantes, para tareas administrativas, etc…

3.2.4.- Control de acceso físico:

El acceso a los edificios está restringido a estudiantes, ex estudiantes, docentes y todo personal que pertenece a la Institución. A la entrada de la Institución existe una garita controlada por 2 guardias de seguridad que verifican la identidad de cada persona que ingresa a la Institución. Existe un control para diferenciar estudiantes, docentes y administrativos de las demás personas que ingresan, mediante un adhesivo que debe estar visible en el parabrisas de los automóviles, de ésta manera el ingreso es más rápido y así se identifica quién pertenece a la Institución, los que no posean este adhesivo se les tomará ciertos datos para el ingreso y se les preguntará el motivo de su visita, queda a criterio del guardia la autorización de entrada. Además existe una segunda garita controlada por otro guardia de seguridad en la parte de arriba a la entrada del terreno propiamente dicho que ejerce los mismos controles de seguridad.
Los 3 guardias de seguridad permanecen las 24 horas del día y hacen rondas por todo el perímetro físico.

Los 3 edificios del Campus Miguel de Cervantes están hechos de hormigón armado por lo que se considera como una estructura robusta. Las puertas de las aulas son de madera y tienen 2 cerraduras cada una, así como las puertas exteriores son de hierro y son aseguradas mediante candados grandes. Las ventanas son de vidrio y están protegidas externamente por rejillas de hierro soldadas a la estructura de los edificios.

El edificio principal y el edificio de la Facultad de Medio ambiente tienen sistema de alarma contra robos, éstas son activadas al cierre de horario de la Institución y los encargados de realizar ésta tarea, así como la desactivación, es el personal de mantenimiento.
El edificio perteneciente a la Facultad de Comunicación no tiene sistema de alarma.

3.2.5.- Protecciones en lugares específicos:

Los cuartos que albergan los sistemas IT en cada uno de los edificios tienen extintores en caso de producirse un incendio, aunque no poseen ningún tipo de dispositivos de detección de humo. Los extintores contienen PQS (Polvo Químico Seco), pesan 15 libras y están colgados en la pared en un lugar de fácil accesibilidad

Tampoco se dispone de una impermeabilización de los cuartos o de algún sistema de detección de fuga de líquidos en caso de inundación parcial o completa, así como de un sistema de control de temperatura. Se dispone de sistemas UPS para garantizar el funcionamiento de los sistemas IT en períodos de corte de energía.

3.2.6.- Segmentos de red:

El único segmento de red para todo el campus es el 192.168.2.x

3.2.7- Red inalámbrica:

La red inalámbrica del Campus Miguel de Cervantes dispone del servicio DHCP que asigna direcciones IP automáticamente para cada equipo que se conecte, el acceso es libre sin necesidad de contraseña, el rango va desde 192.168.2.140 a la 192.168.2.240

La red cumple con el estándar 802.11/g y tiene el nombre de UISEK.

EQUIPO	SERIE	UBICACIÓN	IP
CISCO AP1130	FTX1019T2WS	LABORATORIO DE SITEMAS 1	192.168.2.241
DLINK DWL-2100AP	DR9T266004634	COMUNICACIÓN	192.168.2.242
3COM OFFICE CONECT WL-525	M3BA5CDC6A18C	SECRETARIA	192.168.2.243
3COM OFFICE CONECT WL-252	M3BA69EB08703	BIBLIOTECA	192.168.2.244
DLINK DI-624	F31G164001275	TELECOMUNICACIONES	192.168.2.245
3COM OFFICE CONECT WL-450	0444/75VF6BEB2ADE7	MEDIO AMBIENTE	192.168.2.246
CISCO AP1130		AULA MBA 1	192.168.2.248

Tabla 3.2.7

3.2.8.- Edificio principal:

Aquí es donde se encuentra la mayor parte de la infraestructura de red de la Universidad, y es también la más crítica debido a que los servidores y el armario o rack principal de comunicaciones se encuentran alojados en éste edificio. Existen conexiones físicas e inalámbricas para conectarse a la red.

El edificio principal está dividió en varias secciones donde hay conectividad:

SECCIÓN	# EQUIPOS	DIRECCIONES IP
Laboratorio “Sistemas 1”	11	(*Ver Anexo 3)*
Laboratorio “Sistemas 2”	24	(*Ver Anexo 4)*
Laboratorio “Finanzas”	26	(*Ver Anexo 5)*
Lab. “Comunicación”	11	(*Ver Anexo 6)*
Biblioteca	10	(*Ver Anexo 7)*
Administrativos	4	(*Ver Anexo 8)*
Decanatos	2	(*Ver Anexo 9)*

SECCIÓN	# EQUIPOS	DIRECCIONES IP
Tutorías	4	(*Ver Anexo 10*)
Aulas	12	
“Telecomunicaciones”	5	Red Interna
Enfermería	1	No tiene asignada

Tabla 3.2.8

En total el edificio principal tiene 105 PC’s activas.

3.2.8.1.- Rack principal:

En el centro de cómputo “Sistemas 1” se encuentra el armario o rack que concentra todo el cableado estructurado del Campus y tiene alojado el servidor de Internet que administra toda la red de datos. El armario esta debidamente organizado y etiquetado para poder identificar cualquier falla de la red de una manera más sencilla, podemos ver que todos los switch están conectados en cascada y el cable que tiene ésta función es de color anaranjado a diferencia del resto, es decir todas las computadoras existentes en el Campus estarían en la misma red.

El rack se distribuye de la siguiente manera:

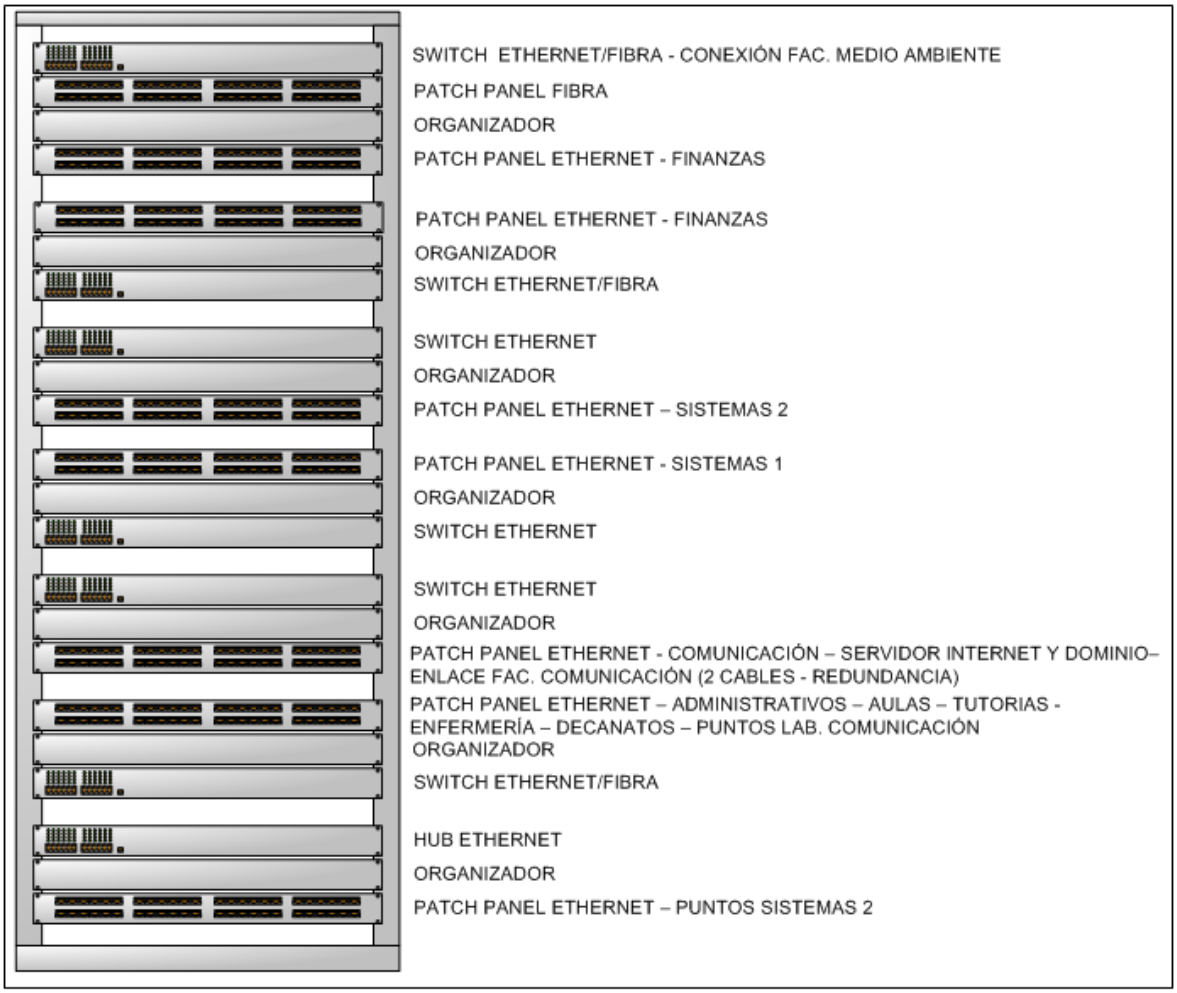


Figura 3.2.8.1

3.2.8.2.- Servidores y direccionamiento IP:

En la misma sala de equipos, en una mesa, hay 3 servidores:

NOMBRE	FUNCIÓN	MARCA	S.O.	SERVICIOS
CAYAMBE	Internet	HP PROLIANT DL 380 G1	LINUX CENTOS 5.1	Firewall, Proxy, DHCP, WebSense
UIO	Soft. Bibliotea y tutorías a distancia		LINUX CENTOS 5.1	Moodle

NOMBRE	FUNCIÓN	MARCA	S.O.	SERVICIOS
SERVER02MC	Dominio	DELL POWER EDGE 860	WINDOWS 2003 SERVER	Active Directory, Panda Business Security

Tabla 3.2.8.2.1

NOMBRE	IP PÚBLICA	IP PRIVADA
CAYAMBE	157.100.115.50	192.168.2.253

Tabla 3.2.8.2.2

NOMBRE	IP	DOMINIO
SERVER02MC	192.168.2.254	UISEKMC

Tabla 3.2.8.2.3

3.2.8.3.- Configuración del Firewall:

La configuración actual del Firewall restringe el reenvío de todo tráfico que salga de la red local hacia el exterior, y solo se autoriza la salida de ciertas direcciones IP y puertos mediante reenvío para que no pasen por el firewall. *(Para mayor detalle ver Anexo 11)*

Las direcciones IP permitidas son éstas:

SERVICIO	IP
BIBLIOSEK	192.168.2.251
PC ADMINISTRADOR SISTEMAS	192.168.2.1
PC ASISTENTE ADMINISTRATIVA	192.168.2.211

Tabla 3.2.8.3.1

Los puertos permitidos son éstos:

PUERTO	SERVICIO
80 TCP, UDP	HTTP
81 TCP, UDP	HOSTS2 NAME SERVER
25 TCP, UDP	SMTP
21 TCP	FTP
53 TCP, UDP	DNS
110 TCP, UDP	POP3
22 TCP	SSH
23 TCP	TELNET
443 TCP	HTTPS
10000 TCP	WEBMIN
3128 TCP, UDP	PROXY FIJO
8080 TCP, UDP	PROXY FIJO

Tabla 3.2.8.3.2

Todo tráfico que salga por el puerto 80 (HTTP) es redireccionado al puerto 8080 (Proxy Fijo)

3.2.8.4.- Configuración del Proxy:

En la configuración actual del proxy se establecen 7 acl's (Access Control List) con sus rangos de direcciones IP *(Para mayor detalle ver Anexo 12)*:

NOMBRE ACL	RANGO IP
ADMINISTRATIVOS	192.168.2.1 – 192.168.2.19
FINANZAS	192.168.2.20 – 192.168.2.49
COMUNICACIÓN	192.168.2.50 – 192.168.2.64
SISTEMAS1	192.168.2.65 – 192.168.2.88
SISTEMAS2	192.168.2.89 – 192.168.2.99
DHCP	192.168.2.100 – 192.168.2.255
PROHIBIDOS	"url_regex sexo hi5 sex chicas porno ares hardcore mamadas tetas x55.iloveim.com iloveim"

Tabla 3.2.8.4.1

Con las listas de control de acceso establecidas se deniega o autoriza el acceso a Internet y se deniega el acceso a los sitios web que contengan palabras establecidas en la lista de control de acceso "Prohibidos":

ACL	PERMISO	DESCRIPCIÓN
LOCALHOST	ALLOW	Permite acceso
ADMINISTRATIVO	ALLOW	Permite acceso
FINANZAS	ALLOW !PROHIBIDOS	Permite acceso, pero no a sitios web que contengan palabras contenidas en el ACL “Prohibidos”
SISTEMAS1	ALLOW !PROHIBIDOS	Permite acceso, pero no a sitios web que contengan palabras contenidas en el ACL “Prohibidos”
SISTEMAS2	ALLOW !PROHIBIDOS	Permite acceso, pero no a sitios web que contengan palabras contenidas en el ACL “Prohibidos”
COMUNICACIÓN	ALLOW !PROHIBIDOS	Permite acceso, pero no a sitios web que contengan palabras contenidas en el ACL “Prohibidos”
DHCP	ALLOW !PROHIBIDOS	Permite acceso, pero no a sitios web que contengan palabras contenidas en el ACL “Prohibidos”

Tabla 3.2.8.4.2

3.2.8.5.- Puntos de acceso inalámbricos:

En el edificio principal existen 5 dispositivos de punto de acceso inalámbricos distribuidos de la siguiente manera:

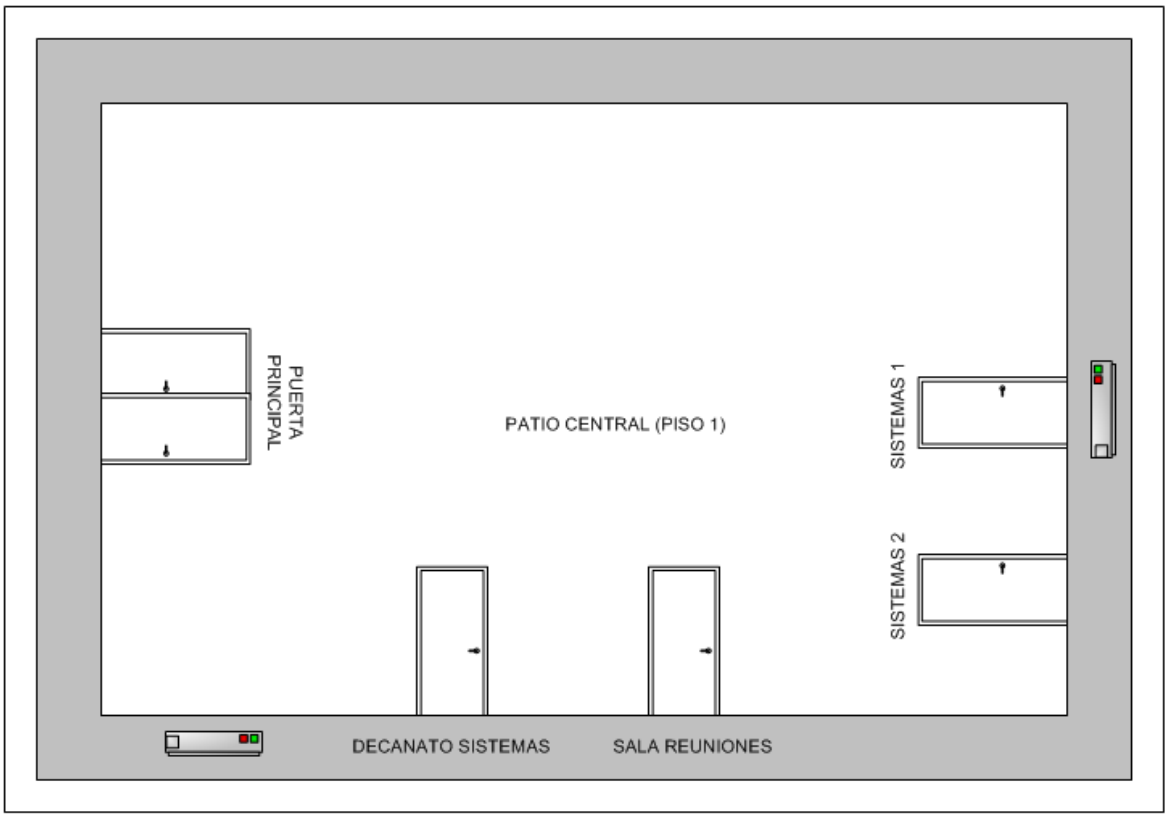


Figura 3.2.8.5.1

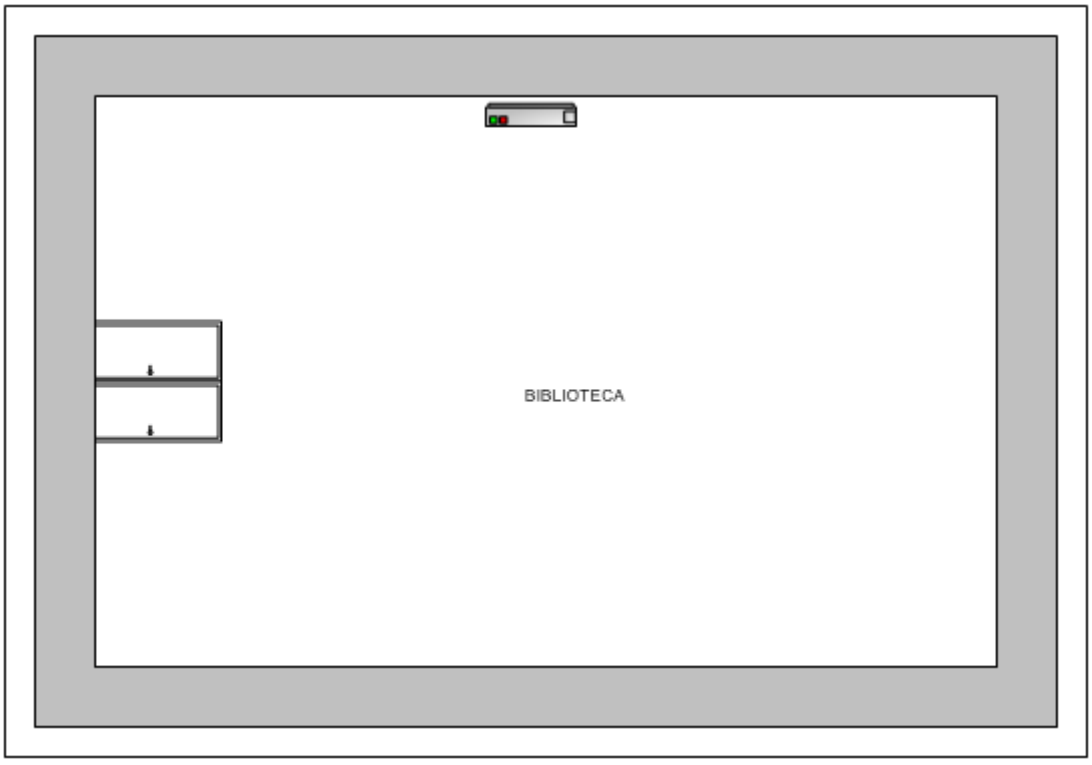


Figura 3.2.8.5.2

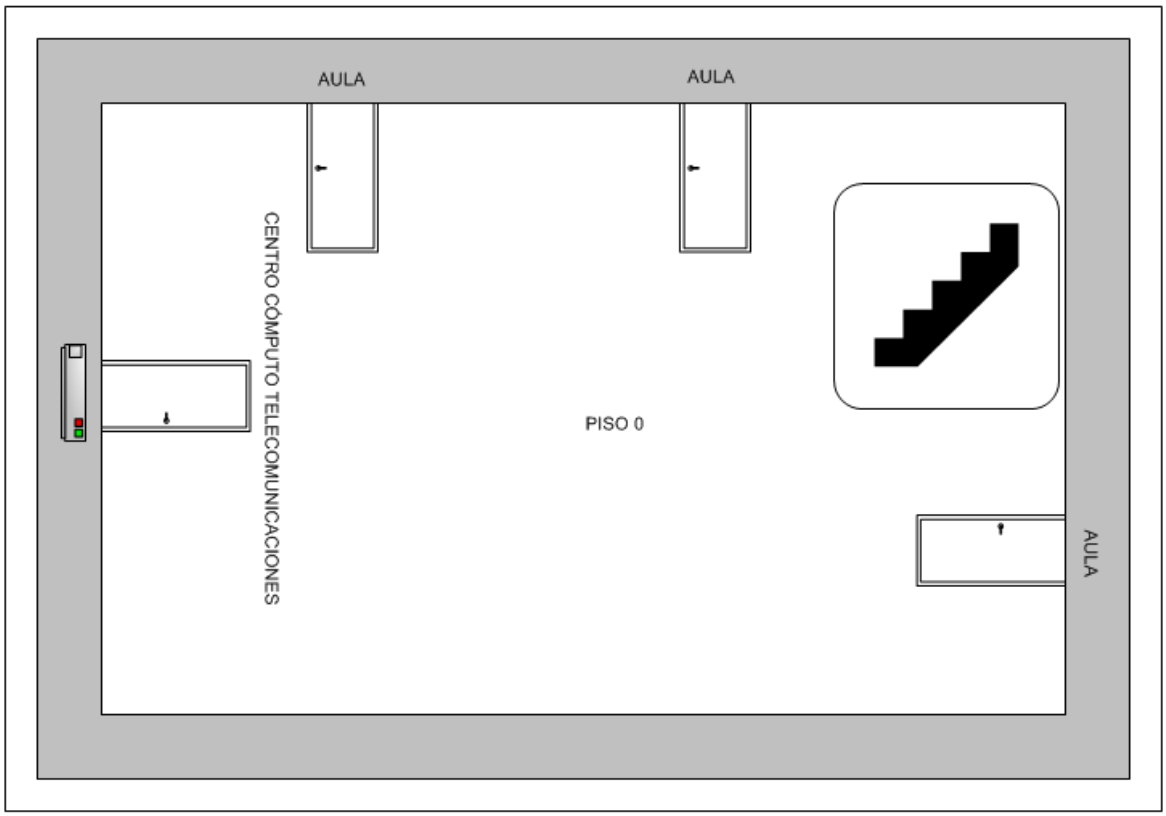


Figura 3.2.8.5.3

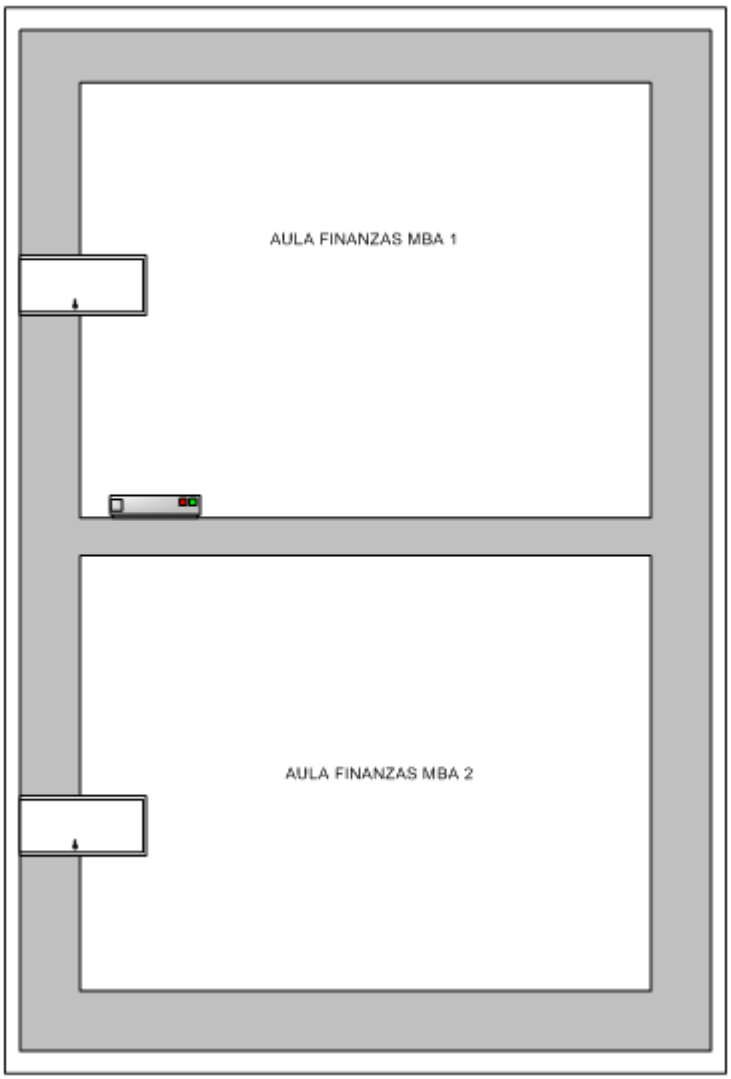


Figura 3.2.8.5.4

3.2.9.- Facultad de Medio Ambiente:

No existe infraestructura física de red además de la existente en la sala de equipos de la facultad, ya que en éste edificio toda la red es inalámbrica.

3.2.9.1.- Rack:

En la facultad existe un cuarto de equipos, de donde conecta el enlace de fibra óptica hacia el edificio principal para poder tener todos los servicios de red, en este rack también se conectan puntos de acceso inalámbricos para poder dar servicio de Internet a toda la facultad. El rack se distribuye de la siguiente manera:

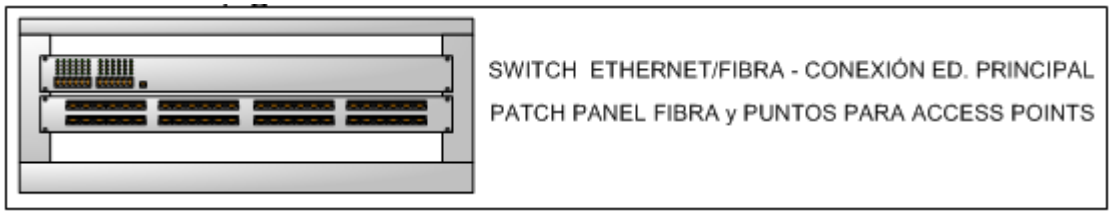


Figura 3.2.9.1

3.2.9.2.- Puntos de acceso inalámbricos:

Existe un punto de acceso inalámbrico en el Hall de la entrada a la facultad, otro está en la sala de equipos, y hay 2 puntos de acceso por cada piso, es decir 4 puntos de acceso, repartidos entre el segundo y tercer piso. En total existen 6 puntos de acceso inalámbricos en la Facultad de Medio Ambiente.

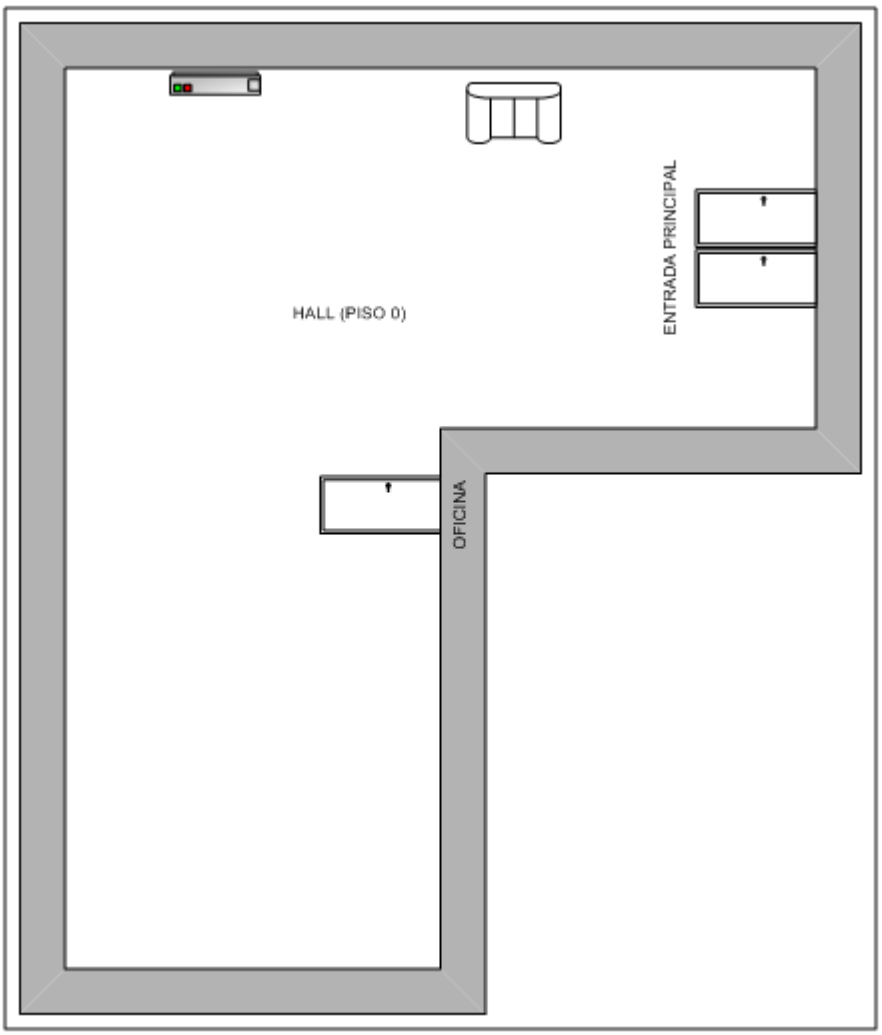


Figura 3.2.9.2

3.2.10.- Facultad de Comunicación:

En el edificio existe infraestructura de red física e inalámbrica, éste tiene un enlace de par trenzado apantallado hacia el edificio principal, para poder tener todos los servicios de red, el enlace no supera los 100 mts.

3.2.10.1.- Rack:

Existe un rack en el edificio de donde conecta el enlace al edificio principal y también existen puntos para el access point y todos los computadores de la facultad. El rack se distribuye de la siguiente manera:

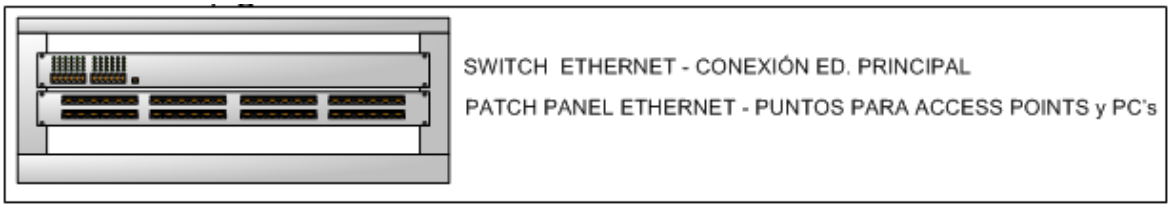


Figura 3.2.10.1

3.2.10.2.- Puntos de acceso inalámbricos:

Existe un punto de acceso en el aula de diseño gráfico, donde está el rack de comunicaciones, éste cubre óptimamente las necesidades de conexión inalámbrica en toda la facultad.

3.2.11.- Selección de contramedidas de seguridad

3.2.11.1.- Manejo de incidentes

Todas las personas pertenecientes a la Institución deben reportar los incidentes de seguridad que se presenten. Debe existir un formato de reporte para poder describir ciertos datos importantes del incidente.

Debe existir un proceso definido a llevar cabo cuando se presenten incidentes de seguridad, para tener uan buena capacidad de reacción al mismo. Información sobre dichos incidents debe recolectada y evaluada para prevenir incidents similares en el futuro.

3.2.11.2.- Personal

Es importante minimizar los riesgos de seguridad resultado de errors intencionales o no intencionales que impliquen el incumplimiento de reglas de seguridad aplicadas al personal de la Institución.

3.2.11.2.1.- Medidas de seguridad para personal permanente y temporal

Todos los empleados deben tener conocimiento de sus roles de seguridad y responsabilidades (si aplica), así como todos los procedimientos de seguridad relevantes, que los mismos deberían llevar a cabo, y dichos procedimientos deberían estar plasmados

en un documento. Los empleados deben firmar un acuerdo de confidencialidad antes de ser contratados por la Institución, si es necesario.

3.2.11.2.2.- Medidas de seguridad para personal de sistemas

Todo empleado que use, desarrolle y tenga acceso a los sistemas IT de la Institución, debe recibir entrenamiento regular sobre amenazas y vulnerabilidad actuales en el tema de seguridad informática, de ésta forma el empleado esta conciente de la importancia de la seguridad en los sistemas IT de la Institución. También es importante el entrenamiento para usar las areas donde haya sistemas IT, para evitar errores.

3.2.11.3.- Tareas operacionales

Es necesario tener procedimientos seguros al momento del mantenimiento o modificación de algunos aspectos de los sistemas IT, éstas medidas también deben ser llevadas a cabo en combinación con medidas de seguridad físicas.

3.2.11.3.1.- Control de cambios

El control de cambios es el proceso de crear un historial de cambios realizados a los sistemas IT. El objetivo principal de seguridad es asegurar que los cambios no repercutan en el rendimiento de las medidas de seguridad de dichos sistemas.

3.2.11.3.2.- Manejo de capacidades

El manejo de capacidades debe ser usado para evitar fallas debido a que el equipo no tiene

capacidad suficiente (RAM, Disco duro, Procesador, etc…). Futuros requerimientos de capacidad deben ser tomados en cuenta.

3.2.11.3.3.- Documentación

Todos los aspectos de la configuración de los equipos IT y las operaciones deben ser documentados para asegurar la continuidad y consistencia de los sistemas IT. Los aspectos de seguridad también deben ser documentados.

3.2.11.3.4.- Monitoreo de cambios de seguridad

Los cambios en las amenazas, vulnerabilidades y riesgos deben ser monitoreados. El monitoreo debe incluir los aspectos existentes y nuevos que se encuentren.

3.2.11.3.5.- Auditoría de accesos

La capacidad de manejo de acceso de usuarios de servidores, redes y aplicaciones debe ser utilizada para tener un historial detallado de eventos de seguridad relevantes. Esto incluye detalles de eventos no autorizados y eventos normales que posiblemente deben ser analizados posteriormente. Los logs de auditoria deben ser revisados regularmente para detector actividades no autorizadas y permitir aplicar medidas correctivas a dichas actividades.

3.2.11.3.6.- Pruebas de seguridad

Aplicar pruebas de penetración en la red de la Institución es importante para asegurar el funcionamiento seguro de todos los equipos IT. Es necesario también aplicar las pruebas

para asegurar que el nivel de seguridad requerido ha sido alcanzado cuando se aplique un plan de implementación de la seguridad.

3.2.11.3.7.- Sistema de gestor de seguridad

La implementación de un sistema gestor de la seguridad es una parte importante para unir datos de varios puntos de la red y tener un monitoreo permanente.

3.2.11.3.8.- Actualización de Plan de Contingencia

Un plan de contingencia debe ser siempre actualizado mediante pruebas en un ambiente simulado para constatar que el plan funciona bajo condiciones reales, y también se debe actualizar cuando en la Institución haya cambios en aspectos Organizaciónales, físicos, etc…

3.2.11.4.- Seguridad física

Se debe considerar medidas de seguridad en el ambiente físico de la Institución, debido a que los riesgos también se presentan en éste ámbito y pueden ser los mas dañinos.

3.2.11.4.1.- Protección física en zonas de seguridad

Las areas consideradas como seguras en los edificios de la Institución, como los cuartos donde se albergan servidores, racks, etc… deben ser protegidas mediante un control de acceso físico, puertas y ventanas de material resistente, logs de acceso, etc…

3.2.11.4.2.- Protección contra fuego

Los equipos y areas cercanas a las zonas de seguridad, incluyendo los accesos, deben ser protegidas contra fuego desde cualquier punto dentro del edificio o desde los edificios colindantes. Se debe incluir detectores de humo y fuego, así como alarmas y equipos de supresión del mismo, como sprinklers. Es importante asegurarse que el material de supresión no sea dañino para los equipos IT al tratar de apagar fuego.

3.2.11.4.3.- Protección contra agua

Las zonas seguras deben ser situadas en un sitio que no sea propenso a inundación o derrame de líquidos, o en su defecto se debe impermeabilizar totalmente la zona en caso que se suscite una inundación o derrame de líquidos.

3.2.11.4.4.- Protección contra desastres naturales

Los edificios que albergen zonas seguras deben tener capacidad de protección contra relámpagos, así como los equipos IT deben ser también protegidos contra estos efectos. Protecciones contra otro tipo de desastres naturales se puede lograr evitando areas propensas a éstos y teniendo como apoyo un plan de contingencia probado.

3.2.11.4.5.- Protección contra robos

Para lograr un control sobre los activos importantes de la Institución, es necesario tener un inventario de los mismos. Los guardias de seguridad o recepcionistas deben verificar a toda persona que entre o salga de la insitución y deben autorizar o desautorizar la salida de equipos IT.

3.2.11.4.6.- Control de poder y temperatura

Todo equipo IT debe ser protegido contra fallos de suministro eléctrico mediante reguladores de voltaje y UPS's. Es necesario también implementar un sistema de control de temperatura con aire acondicionado para evitar fallas de hardware.

3.2.11.4.7.- Cableado

El cableado eléctrico y de comunicaciones deber ser protegido contra interferencias, daños, sobrecarga y debe ser físicamente protegido contra daño accidental o intencional.

CAPITULO IV

DISEÑO DEL ESQUEMA DE SEGURIDAD PERIMETRAL

4.1.- ESQUEMA DE SEGURIDAD PERIMETRAL PROPUESTO

Además de incrementar el nivel de seguridad lógica, física, Organizaciónal, legal, y de crear un marco de seguridad informática con un manual de políticas IT y un plan de contingencia ante desastres, es importante también crear un esquema de seguridad perimetral eficiente que se acople a las necesidades de la Institución y complemente el marco de seguridad informática previamente creado.

El esquema de seguridad perimetral propuesto es el siguiente:

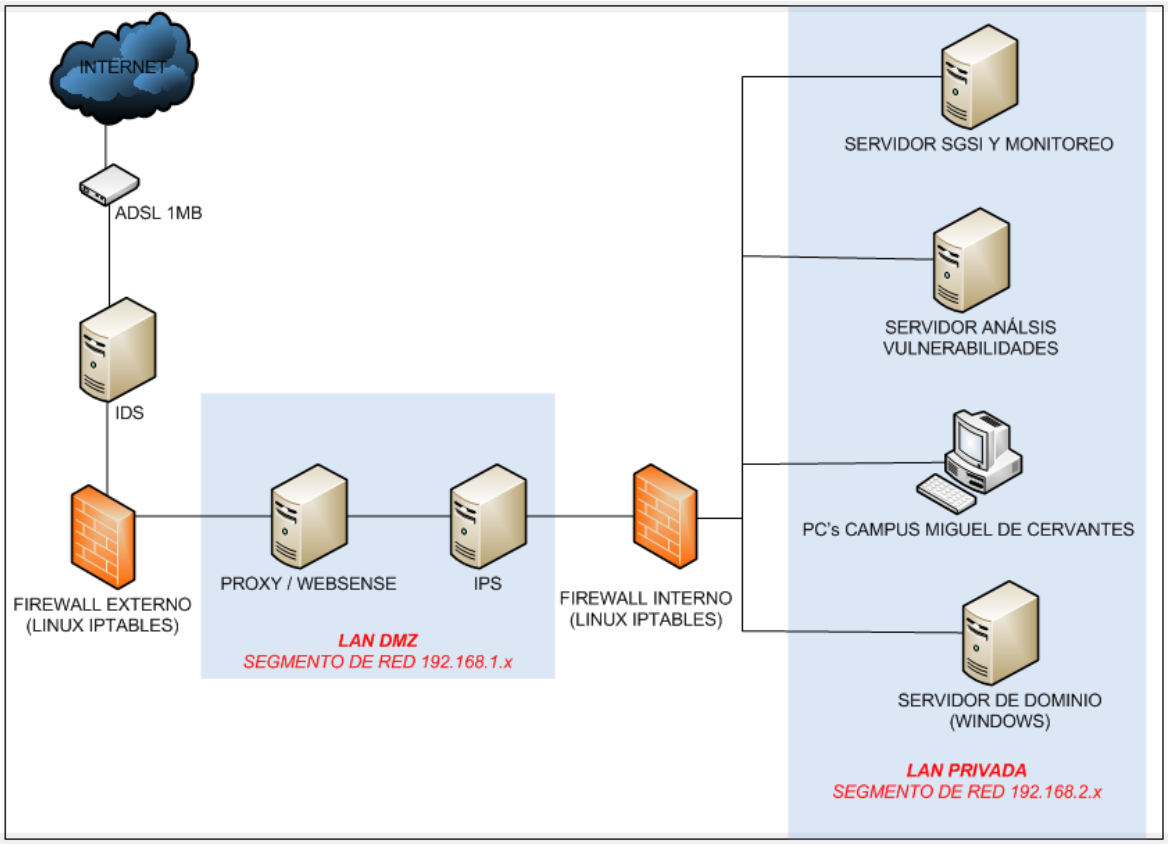


Figura 4.1

Este esquema permite incrementar el nivel de seguridad perimetral reorganizando los componentes críticos de la red y aumentando otros, basándose en una estructura de seguridad perimetral definida.

La estructura de “muro doble” consiste en añadir 2 firewalls para poder diferenciar 3 segmentos de red:

- Red externa
- Red semi-privada (DMZ)
- Red privada

De ésta manera aumenta el nivel de seguridad en la red, ya que en un supuesto caso de un ataque, tendrán que ser comprometidos 2 firewalls para poder ingresar a la red privada.

4.1.1.- Red externa

La red externa es el Internet, donde no hay ningún tipo de restricción de tráfico ni garantiza ningún nivel de seguridad, aquí se va a exponer el servidor IDS.

4.1.2.- Red Semi-privada (DMZ)

La red semi-privada o DMZ es un segmento de red donde se ubican los servidores que tengan que recibir cierto tipo de tráfico entrante de la red externa, por ejemplo, servidores proxy, de correo, filtro de contenidos, ftp, servidor web, etc…., es decir, es un segmento con un nivel de seguridad medio ya que permite el ingreso de ciertos paquetes filtrados por el firewall externo. Para las necesidades de la red de la UISEK Ecuador Campus Miguel de Cervantes, es necesario poner el servidor proxy/websense y un servidor IPS en éste segmento.

4.1.3.- Red privada

La red privada tiene un nivel alto de seguridad y confiabilidad, ya que la mayor parte del tráfico es saliente y el tráfico entrante es muy limitado y proviene solo de los servidores de la red semi-privada (DMZ) que ya ha sido filtrado por el firewall externo y adicionalmente serán filtrados por el firewall interno.

4.1.3.- Firewall externo

Este componente permite el tráfico a los puertos establecidos como permitidos en su configuración para q cualquier usuario de la red insegura pueda acceder estrictamente solo a los servicios de los servidores dentro de la DMZ. El firewall externo va a ser implementado con éstos datos:

NOMBRE	INTERFAZ 1	INTERFAZ 2
COTOPAXI	157.100.115.50	192.168.1.253

Tabla 4.1.3.1

El tráfico que entre desde la interfaz 1 (INTERNET) a la interfaz 2 (DMZ) por los puertos listados será permitido:

PUERTO	SERVICIO
80 TCP, UDP	HTTP
22 TCP	SSH
443 TCP	HTTPS
3128 TCP, UDP	Proxy fijo
8080 TCP, UDP	Proxy fijo
53 TCP, UDP	DNS

Tabla 4.1.3.2

El tráfico que vaya desde la interfaz 2 (DMZ) a la interfaz 1 (INTERNET) por los puertos listados será permitido:

PUERTO	SERVICIO
80 TCP, UDP	HTTP
25 TCP, UDP	SMTP
53 TCP, UDP	DNS
110 TCP, UDP	POP3
443 TCP	HTTPS

Tabla 4.1.3.3

4.1.4.- Firewall interno

El firewall interno tiene un nivel de seguridad alto debido a que es el componente que protege la red local y prácticamente deberá solo permitir el acceso del tráfico para navegar en Internet. El firewall interno va a ser implementado con éstos datos:

NOMBRE	INTERFAZ 1	INTERFAZ 2
CHIMBORAZO	192.168.1.252	192.168.2.253

Tabla 4.1.4.1

El tráfico que entre desde la interfaz 1 (DMZ) a la interfaz 2 (RED LOCAL) por los puertos listados será permitido:

PUERTO	SERVICIO
80 TCP, UDP	HTTP
22 TCP	SSH
443 TCP	HTTPS
3128 TCP, UDP	Proxy fijo
8080 TCP, UDP	Proxy fijo
53 TCP, UDP	DNS
110 TCP, UDP	POP3

Tabla 4.1.4.2

El tráfico que vaya desde la interfaz 2 (RED LOCAL) a la interfaz 1 (DMZ) por los puertos listados será permitido:

PUERTO	SERVICIO
80 TCP, UDP	HTTP
25 TCP, UDP	SMTP
53 TCP, UDP	DNS
110 TCP, UDP	POP3
22 TCP	SSH
443 TCP	HTTPS
10000 TCP	Webmin
3128 TCP, UDP	Proxy fijo
8080 TCP, UDP	Proxy fijo

Tabla 4.1.4.3

4.1.5.- IPS

El servidor IPS, que se encuentra dentro de la DMZ, va a servir como un control adicional en caso que un atacante logre penetrar el firewall externo ya que en ese momento el IPS emitirá una alerta y va a descartar los paquetes sospechosos. El servidor IPS se va a configurar para que recolecte información sobre eventos sospechosos y los guarde en una base de datos, de la cual el servidor SGSI (Sistema de Gestión de Seguridad Informática) se alimentará para realizar los reportes.

4.1.6.- IDS

El servidor IDS se ubicará fuera de la DMZ, en la red externa, con el fin de que se pueda captar la mayoría de tráfico sospechoso que intente entrar a la DMZ a través del firewall externo, de ésta manera en caso de que ocurra un ataque quedará registrado en el repositorio del IDS todo el detalle de las alertas que se produjeron, incluso para guardar datos estadísticos de intentos de ataque a la red de Campus, exitosos o no.

4.1.7.- Monitor de red

Se establecerá un servidor de monitoreo de red junto con el SGSI, ya que se consideran complementarios. En el monitor de red se instalarán varias herramientas para verificar el tráfico, disponibilidad, cuellos de botellas en la red, etc... Es muy importante realizar un seguimiento del tráfico de la red ya que gran parte del mal uso de los recursos se puede verificar con el monitoreo frecuente.

4.1.8.- Sistema de Gestión de Seguridad Informática

Se entiende por SGSI como todo un proceso de implementación de seguridad informática en una red y un esquema de monitoreo que recolecte información del estado de la seguridad establecida. Existen aplicaciones que combinan todas las herramientas necesarias en una sola aplicación para centralizar el manejo de todos los componentes de seguridad que existan en la red. Este servidor va a funcionar como una consola central que recolecta datos de los puntos de control de seguridad implementados, éste equipo estará localizado dentro de la red privada ya que va a manejar información que solo ciertos usuarios de la red interna pueden tener acceso.

4.2.- PLAN DE IMPLEMENTACIÓN Y PRUEBAS

El esquema de seguridad perimetral propuesto anteriormente no es posible implementarlo al momento debido a que la Institución prefiere esperar un tiempo prudencial para analizar la posibilidad de implementar el esquema cuando las facultades y alumnos del Campus Juan Montalvo estén próximos a incorporarse al Campus Miguel de Cervantes, sin embargo, parte de la implementación va a ser posible al momento:

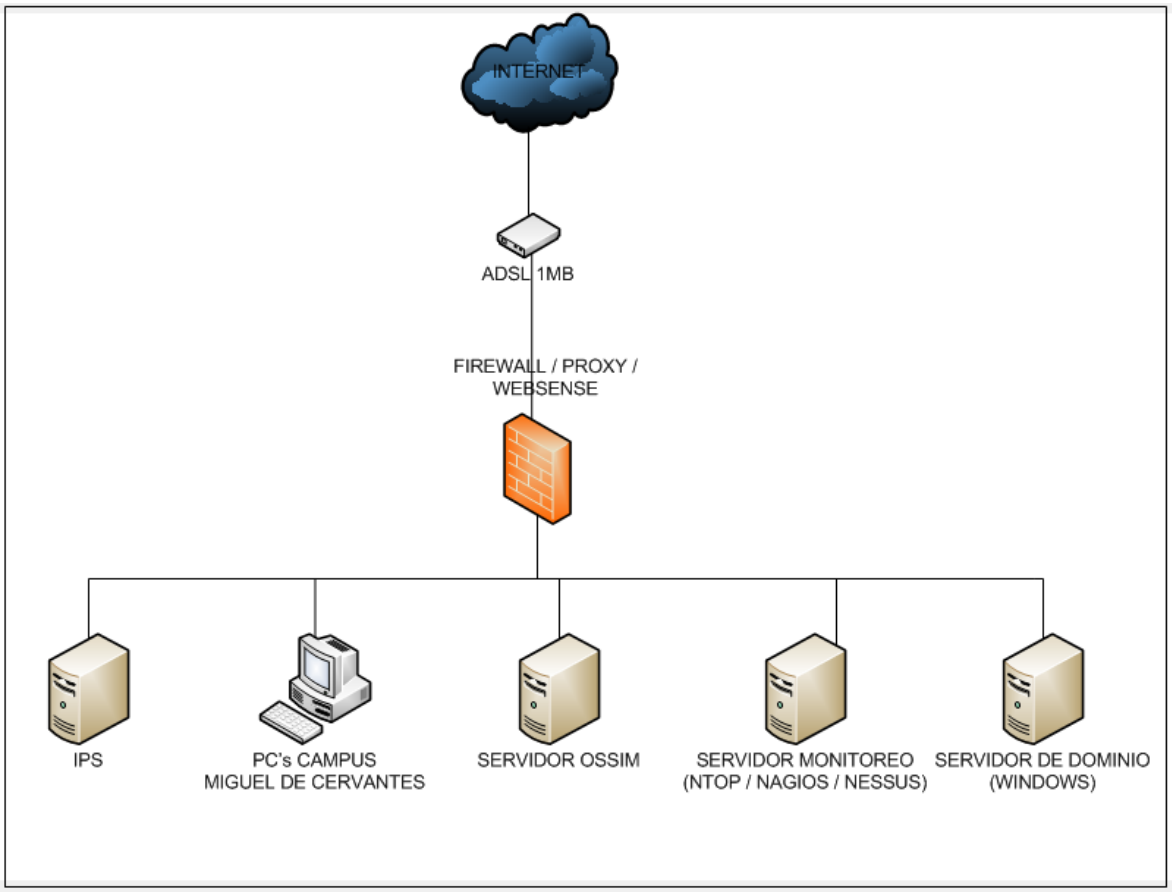


Figura 4.2

4.2.1.- Plan de Implementación

4.2.1.1.- Servidor OSSIM (Open Source Security Information Management)

OSSIM es una herramienta Open Source que sirve para administrar y monitorear un la seguridad en una red informática de una manera centralizada. Se compone de varias herramientas de seguridad que actúan como sensores, idependientes o no, recolectando información para posterior procesamiento y generación de reportes de incidentes de seguridad. Las herramientas contenidas en el sistema OSSIM son las siguientes:

HERRAMIENTA	DESCRIPCIÓN
SNORT	IPS
NESSUS	Detector de vulnerabilidades
NTOP	Monitor de red
NAGIOS	Monitor de disponibilidad de servicios de red
OSIRIS, SNARE	Host IDS
SPADE	Detector de anomalías
NMAP	Análisis de puertos en hosts de red
ACID/BASE	Analizador forense
OSVDB	Base de datos de vulnerabilidades

Tabla 4.2.1.1

Los componentes del sistema OSSIM son módulos independientes y pueden ser configurados cada uno por separado, además según las necesidades de monitoreo de la red, es posible separar cada herramienta en distintos equipos de hardware o ponerlos a todos en un solo equipo.

La arquitectura del sistema OSSIM consiste en 4 elementos:

- Sensores
- Servidor de Gestión
- Base de datos
- Frontend

4.2.1.1.1.- Sensores

Los sensores se despliegan para monitorizar actividad en la red e incluyen:

- Monitores y detectores pasivos (no afectan al tráfico) de bajo nivel que colectan datos buscando patrones.
- Analizadores activos que buscan vulnerabilidades en la red.

4.2.1.1.2.- Servidor de gestión

El Servidor de Gestión centraliza la información recibida de los sensores y realiza reportes y gráficos explicativos de la información recolectada.

4.2.1.1.3.- Base de datos

La Base de Datos almacena eventos e información útil para la gestión del sistema.

4.2.1.1.4.- Frontend

El FrontEnd o Consola es la aplicación web de visualización donde se despliega de manera gráfica los resultados y configuraciones del sistema OSSIM.

Se va a implementar el SGSI (Servidor de Gestión de Seguridad Informática) OSSIM en un equipo destinado solo para este servicio, con los siguientes datos:

SISTEMA OPER.	IP ESTÁTICA
LINUX DEBIAN	192.168.2.91

Tabla 4.2.1.1.1

La instalación del sistema OSSIM se realiza mediante un CD al arrancar el computador y éste mismo instala el sistema operativo Linux Debian y la herramienta OSSIM.

4.2.1.2.- Servidor de monitoreo (NTOP / NAGIOS / NESSUS)

La herramienta NAGIOS en el servidor de monitoreo sirve para tener un control sobre los equipos críticos de la red, asi como de sus servicios, y se enfoca mas a monitorear la disponibilidad de éstos.

El programa NTOP es una herramienta de monitoreo de red que verifica el tráfico de distintas maneras, por host, por servicio, etc…

Nessus analiza hosts específicos o una red completa de éstos para encontrar vulnerabilidades en ellos.

Las tres herramientas fueron elegidas ya que OSSIM se integra de manera muy simple con éstas.

Se va a utilizar un equipo destinado a las aplicaciones de monitoreo de red y analizadores pasivos con los siguientes datos:

SISTEMA OPER.	IP ESTÁTICA
LINUX CENTOS 5	192.168.2.65

Tabla 4.2.1.2

Para empezar se va a instalar el sistema operativo Linux Centos 5 con un perfil de servidor con interfaz gráfica, seguido de esto se instala la herramienta NTOP mediante el comando ”yum” (Yellow Dog Update Manager) que se conecta a un repositorio de paquetes Linux e instala el software automáticamente.

Acto seguido se instala el paquete NAGIOS mediante la descompresión de un archivo .tar, y se crea un usuario y un password para poder acceder a la interfaz GUI (Graphic User Interface). Es necesario configurarlo para que monitoré los equipos que lo necesiten, para realizar esta tarea modificaremos los archivos localhost.cfg y switches.cfg y añadiremos los siguientes equipos, que previamente consideramos como críticos para la continuidad de las operaciones en el Campus Miguel de Cervantes:

- Servidor de dominio
- Servidor de internet (CAYAMBE)
- Servidor de monitoreo
- Servidor IPS
- Servidor OSSIM
- Switch de personal administrativo

Para la instalación de Nessus también se usa el comando “yum” (Yellow Dog Update Manager).

4.2.1.3.- Servidor IPS (SNORT)

El equipo destinado al servidor IPS va a ejecutar solamente este servicio, ya que al ser un analizador de paquetes activo, necesita algunos recursos libres en el equipo. El sistema IPS elegido es SNORT, debido a que es una herramienta Open Source con licencia GPL

(General Public License), por lo tanto, su uso y modificación del código fuente es libre y además se integra a OSSIM sin problemas.

Estos son los datos para implementar el servidor IPS:

SISTEMA OPER.	IP ESTÁTICA
LINUX CENTOS 5	192.168.2.66

Tabla 4.2.1.3

La herramienta SNORT es por defecto un IDS, pero se la va a configurar acitvando el modo “inline” que le permite actuar como un IPS.

4.2.1.4.- Integración de servidor de monitoreo e IPS al sistema OSSIM

Luego de tener instaladas las herramientas en sus respectivos servidores hay que configurar el servidor OSSIM para que recolecte información de éstos servidores. Esto se hace en su mayor parte en el menú Configuration → Main, aquí se define las direcciones IP para cada herramienta y la ruta de directorios, por defecto viene configurado como localhost, ya que OSSIM ya viene compilado con todas estas herramientas, pero para el caso de la implementación de este esquema de seguridad perimetral, es mejor tener las herramientas en diferentes servidores.

4.2.2.- Plan de pruebas

Es importante relizar pruebas luego de tener implementados los servidores de monitoreo para comprobar el buen funcionamiento de las herramientas y corregir cualquier error que se produzca, asi como verificar que la implementación del esquema de seguridad perimetral no afecte al rendimiento de los servicios de red.

4.2.2.1.- Nagios

Para comprobar el buen funcionamiento de la herramienta luego de creados los servidores, equipos críticos y switches, y luego de haber personalizado las notificaciones en caso de alarma, se procederá de la siguiente manera para simular una de ellas:

- Apagar servidor OSSIM
- Verificar que el servicio NAGIOS sube sin ningún error.
- Verificar el correo entrante de la dirección mail que ingresamos en la configuración de NAGIOS para constatar si el correo de notificación mencionando que el servidor OSSIM no responde, efectivamente llega.

4.2.2.2.- Ntop

Para verificar que la herramienta esta funcionando bien ejecutaremos las siguientes tareas:

- Verificar que el servicio NTOP sube sin problemas.
- Navegar en Internet por alrededor de 15 minutos.

- Verificar en la administración de NTOP en http://192.168.2.65:3000 que en las estadísticas de tráfico de red por datos enviados y recibidos aparezca en la lista la dirección ip de la computadora en la cual estábamos navegando y presente la mayoría de tráfico por HTTP.

4.2.2.3.- Nessus

La herramienta Nessus actúa como una anlizador pasivo, es decir, se ejecuta cuando el usuario lo considere necesario hasta que finalize el proceso y se desactive de nuevo.

Para asegurarse de que la herramienta esta funcionando correctamente debemos ejecutar estas tareas:

- Verificar que el servicio NESSUSD sube sin ningún error.
- Realizar un scaneo de un host en la red.

4.2.2.5.- Servidor IPS

El IPS Snort es una herramienta activa de análisis de paquetes, es decir, una vez que se ejecuta no termina hasta que el usuario le indique. Para verificar el buen funcionamiento de ésta herramienta se ejecutará estas tareas:

- Verificar que el servicio snort este activado.
- Ejecutar el comando snort –vd y verificar si captura el tráfico de red en pantalla.

4.2.2.5.- OSSIM

El sistema OSSIM tiene una interfaz GUI (Graphic User Interface) amigable y fácil de manejar. Cuando en el sistema se hayan configurado cada una de las herramientas, será necesario ejecutar éstas tareas para verificar su funcionamiento:

- Ingresar desde cualquier host a http://192.168.2.91 con el username y password correspondientes.
- Ingresar al menú Monitors → Network y verificar que NTOP se despliega sin problema.
- Ingresar al menú Monitors → Availability y verificar que el resúmen inicial de NAGIOS que aparece concuerde con los equipos ingresados previamente.
- Ingresar al menú Events → Vulnerabilities y verificar que al escanear cualquier host, Nessus despliega resultados.

CAPITULO V

IMPLEMENTACIÓN Y PRUEBAS

5.1.- EJECUCIÓN DEL PLAN DE IMPLEMENTACIÓN

5.1.1.- Servidor OSSIM

La implementación de éste servidor se llevó a cabo de manera bastante simple ya que se realiza mediante un CD que se ejecuta al iniciarse el equipo:

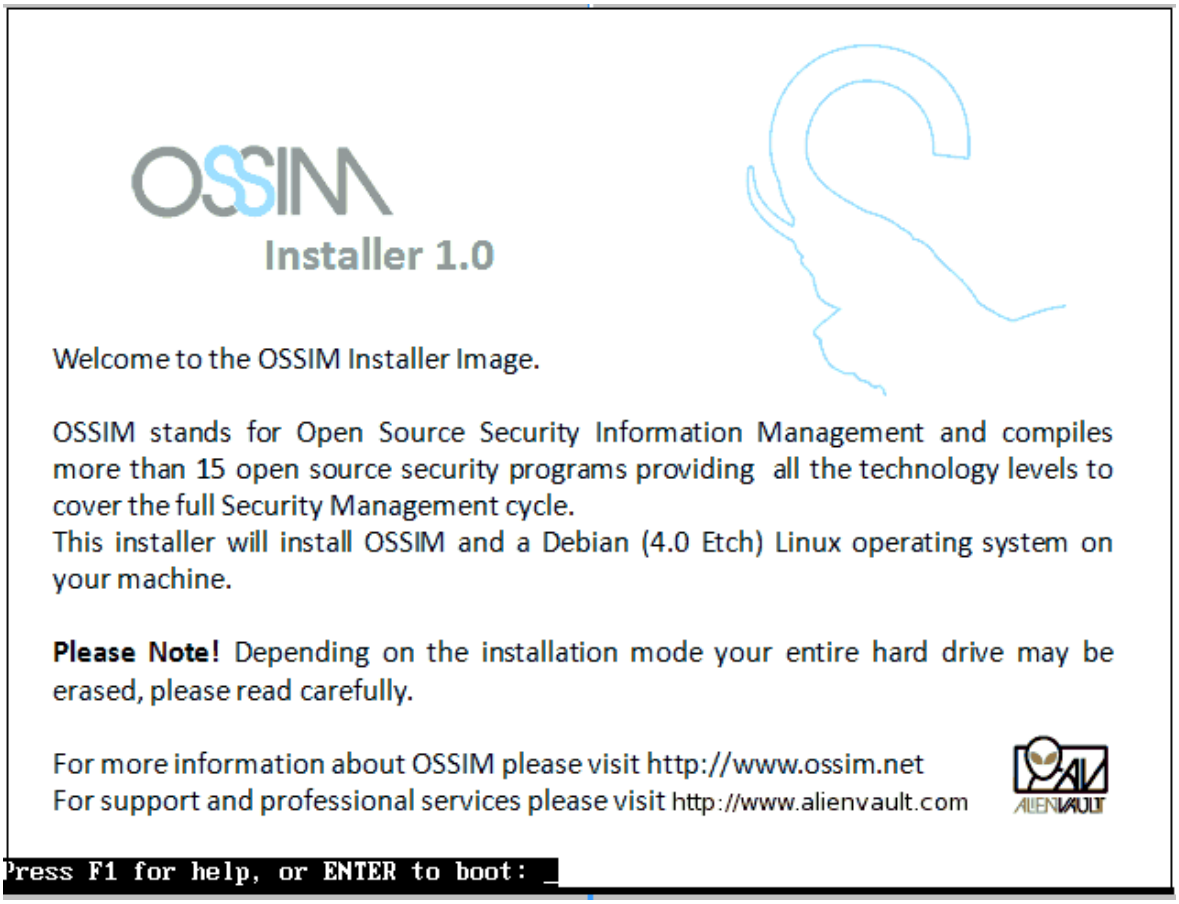


Figura 5.1.1.1

Una de las partes importantes al instalar el sistema es ingresar una dirección ip estática, que previamente se verifica con el adminsitrador de sistemas para cerciorarnos de que efectivamente esta libre de uso, en este caso se usó la 192.168.2.91:

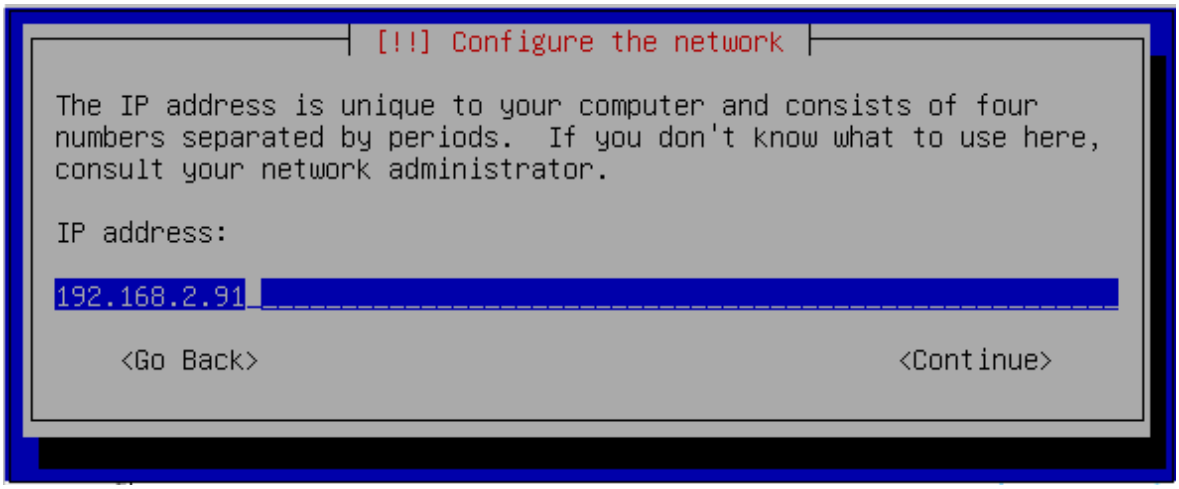


Figura 5.1.1.2

También es importante verificar que se tenga la dirección ip del Gateway, en este caso es 192.168.2.253:

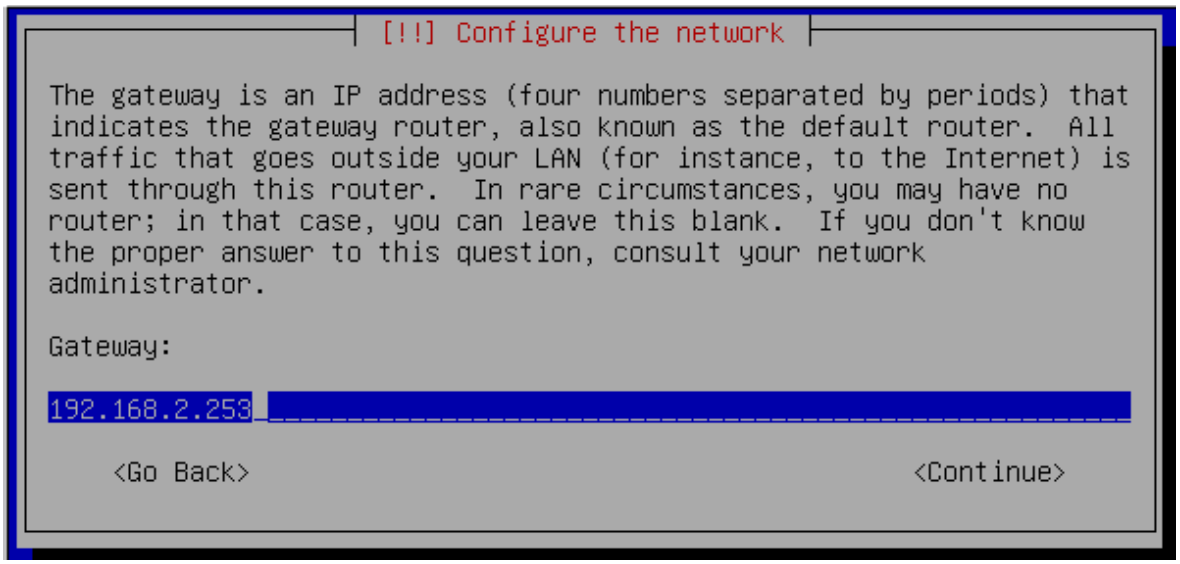


Figura 5.1.1.3

Luego de ingresar otros datos, el sistema procederá a la copia de los archivos de Linux Debian y de OSSIM y la instalación terminará pidiendo el reinicio del equipo, luego de esto el sistema estará instalado completamente.

5.1.2.- Servidor de monitoreo (NTOP / NAGIOS / NESSUS)

En el servidor de monitoreo primeramente se debió instalar el sistema operativo Linux CENTOS 5 con perfil de servidor con interfaz gráfica, ésta concluyó sin ningún problema. Es importante desactivar el Firewall asi como la opción de SELinux (Security enhanced Linux), ya que causan problemas al intentar monitorear la seguridad en una red.

5.1.2.1.- Ntop

Para la instalación de NTOP se usó el comando “yum install ntop”, de ésta manera el paquete se descarga de Internet y el administrador de actualizaciones (yum) lo instala. Luego de instalado se ejecuta el comando “ntop” para iniciar el servico, si es la primera vez que lo ejecutamos nos pedirá una contraseña para el usuario admin, luego de ingresarla ejecutamos el comando “chkconfig ntop on” para que la herramienta se convierta en servicio y se inicie cada vez que se inicia el equipo.

5.1.2.2.- Nagios

El comando “yum install nagios” fue necesario para instalar la herramienta de monitoreo de disponibilidad NAGIOS, luego de esto se debió ingresar los equipos a monitorear mediante los archivos de configuración contenidos en el directorio “/etc/nagios/”, por ejemplo, para ingresar los datos del servidor CAYAMBE para que la herramienta sea capaz de monitorearlo a el y sus servicios, fue necesario entrar al archivo “localhost.cfg” y añadir éste código:

```
define host
{
        use   linux-server
        host_name            cayambe
        alias                cayambe
        address              192.168.2.253
}

define service
{
        use                  local-service
```

```
    host_name               cayambe
service_description         PING
check_command               check_ping!100.0,20%!500.0,60%
}
```

Con esta configuración NAGIOS es capaz de monitorear al servidor CAYAMBE con el servicio de ping, en el caso de que el servidor se caiga por alguna razón el sistema lo detectará ya que no habrá respuesta a la solicitud de ping.

Luego se subió el servicio NAGIOS y se ingresó a la interfaz gráfica mediante la dirección: http://192.168.2.65/nagios, en el menú Config se ingresó un dirección de correo electrónico para que los correos de alerta de servicios se notifiquen también por ese medio.

5.1.2.3.- Nessus

Para instalar Nessus se descargó el paquete rpm de Internet y se lo instaló haciendo doble click sobre él. Acto seguido se subió el servicio nessusd para que esté activo.

5.1.3.- Servidor IPS

En el servidor destinado a IPS se instaló en sistema operativo Linux Centos 5 sin ningún problema. Se descargó el paquete de Snort comprimido en un archivo .tar, y se ejecuto el comando "./configure –enable-inline", de ésta forma se instala la herramienta en el modo "inline", que convierte al Snort común IDS en IPS.

5.1.4.- Integración de servidor de monitoreo e IPS a sistema OSSIM

En el sistema OSSIM se configura cada herramienta, en caso de que éstas estén en servidores separados, para el caso del servidor IPS ingresamos éstos datos:

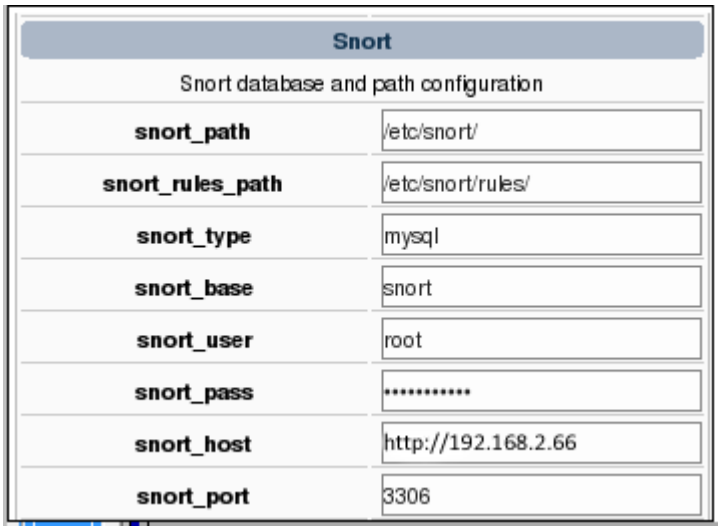


Figura 5.1.4.1

Para la configuración de NTOP y NAGIOS se usaron éstos datos:

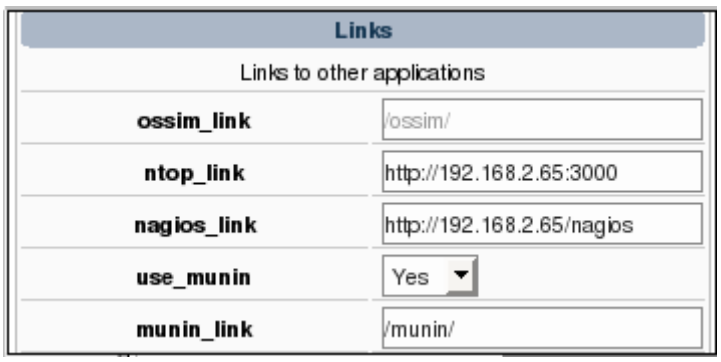


Figura 5.1.4.2

En la configuración de Nessus se usaron los siguientes datos:

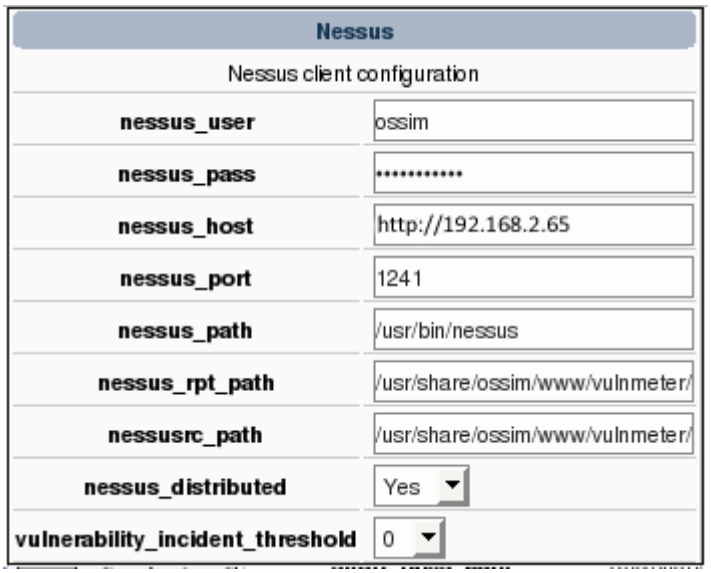


Figura 5.1.4.3

5.2.- EJECUCIÓN DEL PLAN DE PRUEBAS

Luego de tener todas las herramientas instaladas en los diferentes servidores, procedemos a ejecutar las pruebas planeadas para verificar el buen funcionamiento de las herramientas:

5.2.1.- Nagios

TAREAS	REALIZADO (S/N)	OBSERVACIONES
Apagar servidor OSSIM	S	
Verificar que el servicio NAGIOS sube sin ningún error	S	El servicios sube sin problemas
Verificar el correo entrante de la dirección mail que ingresamos en la configuración de NAGIOS para constatar si el correo de notificación mencionando que el servidor OSSIM no responde, efectivamente llega:	S	El correo electrónico se envía desde el sistema y se recibe sin problemas:

Tabla 5.2.1

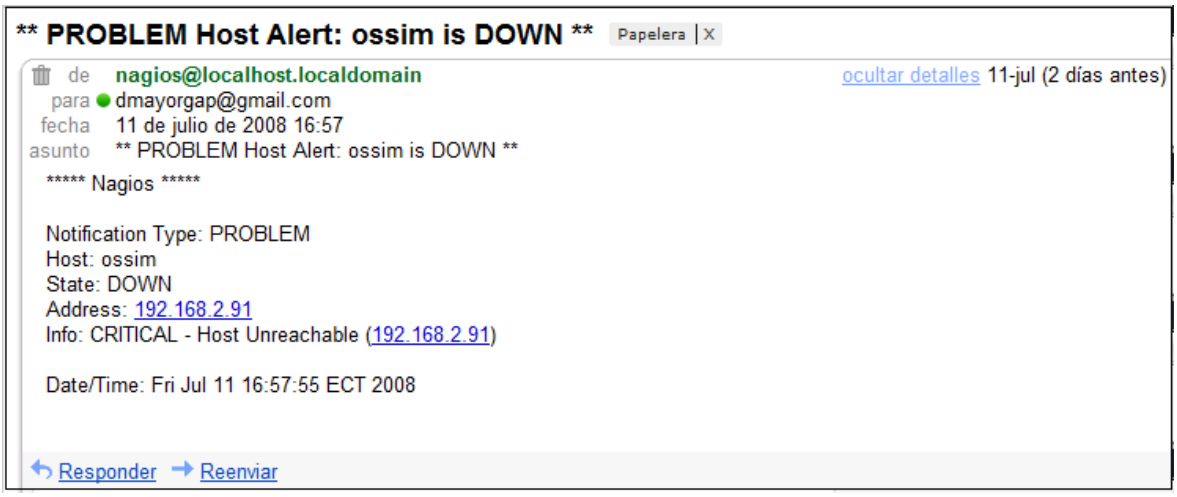


Figura 5.2.1.1

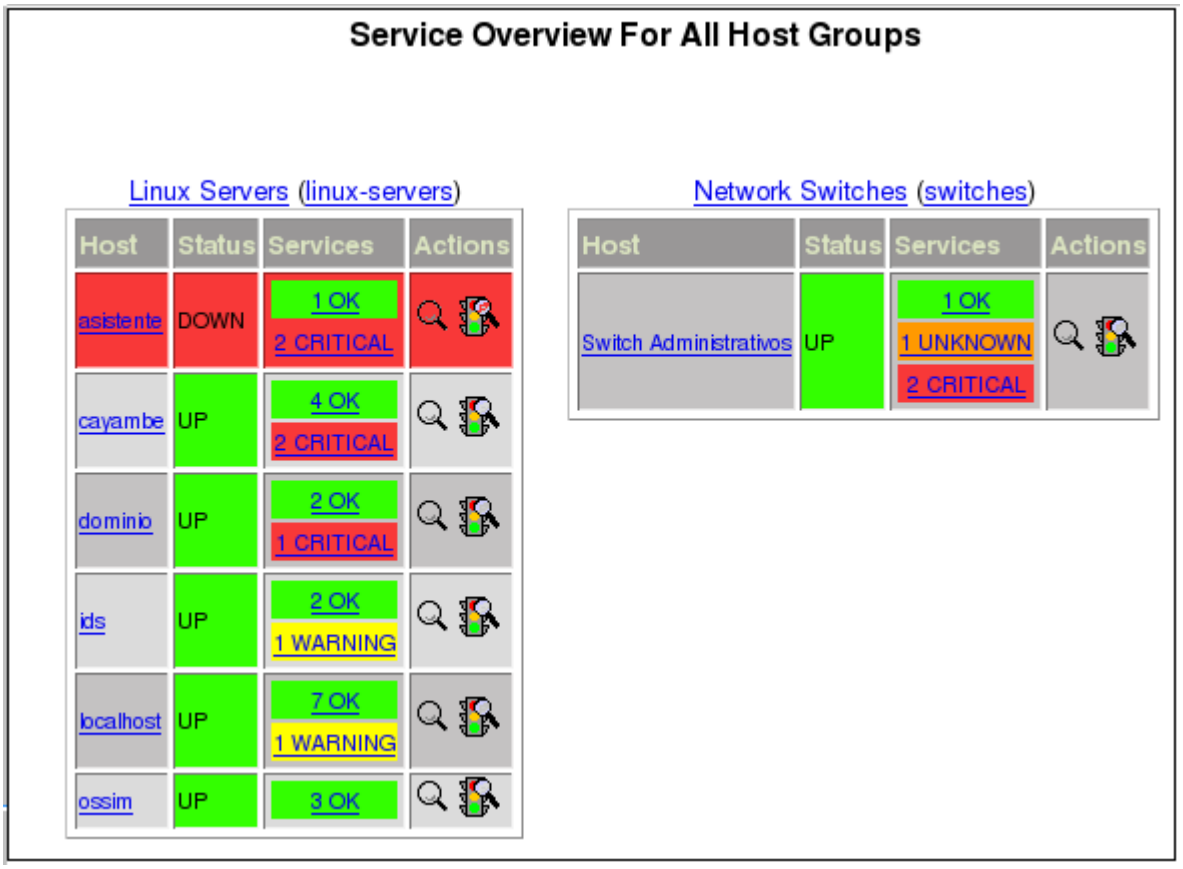
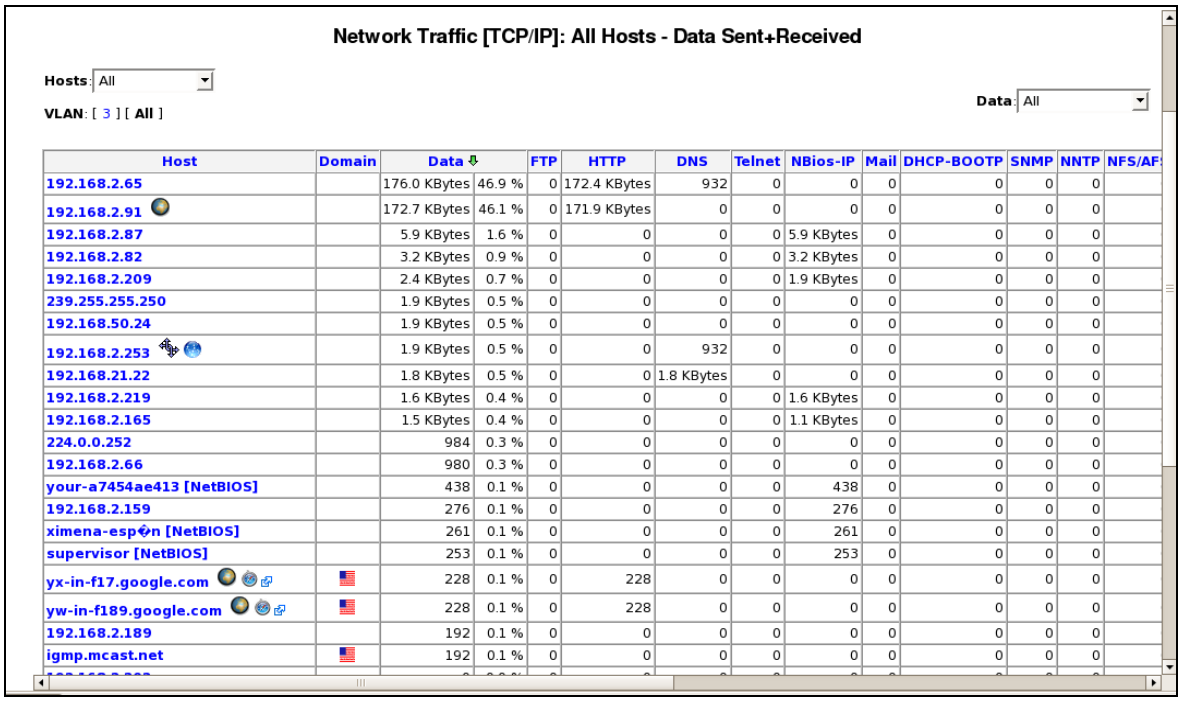

Service Overview For All Host Groups
Linux Servers (linux-servers)
Host
Status
Services
Actions
asistente
DOWN
1 OK
2 CRITICAL
cayambe
UP
4 OK
2 CRITICAL
dominio
UP
2 OK
1 CRITICAL
ids
UP
2 OK
1 WARNING
localhost
UP
7 OK
1 WARNING
ossim
UP
3 OK
Network Switches (switches)
Host
Status
Services
Actions
Switch Administrativos
UP
1 OK
1 UNKNOWN
2 CRITICAL

Figura 5.2.1.2

5.2.2.- Ntop

TAREAS	REALIZADO (S/N)	OBSERVACIONES
Verificar que el servicio NTOP sube sin problemas	S	El servicio tiene un error al subir automáticamente, según investigaciones en foros de Internet, es un bug del programa, hay que subir el servicio manualmente ejecutando el comando “ntop”.
Navegar en Internet por alrededor de 15 minutos.	S	
Verificar en la administración de NTOP en http://192.168.2.65:3000 que en las estadísticas de tráfico de red por datos enviados y recibidos aparezca en la lista la dirección ip de la computadora en la cual estábamos navegando y presente la mayoría de tráfico por HTTP	S	Los reportes de NTOP se despliegan sin problema.

Tabla 5.2.2



Network Traffic [TCP/IP]: All Hosts - Data Sent+Received

Hosts: All

VLAN: [3] [All]

Data: All

Host	Domain	Data		FTP	HTTP	DNS	Telnet	NBios-IP	Mail	DHCP-BOOTP	SNMP	NNTP	NFS/AF
192.168.2.65		176.0 KBytes	46.9 %	0	172.4 KBytes	932	0	0	0	0	0	0	
192.168.2.91		172.7 KBytes	46.1 %	0	171.9 KBytes	0	0	0	0	0	0	0	
192.168.2.87		5.9 KBytes	1.6 %	0	0	0	0	5.9 KBytes	0	0	0	0	
192.168.2.82		3.2 KBytes	0.9 %	0	0	0	0	3.2 KBytes	0	0	0	0	
192.168.2.209		2.4 KBytes	0.7 %	0	0	0	0	1.9 KBytes	0	0	0	0	
239.255.255.250		1.9 KBytes	0.5 %	0	0	0	0	0	0	0	0	0	
192.168.50.24		1.9 KBytes	0.5 %	0	0	0	0	0	0	0	0	0	
192.168.2.253		1.9 KBytes	0.5 %	0	0	932	0	0	0	0	0	0	
192.168.21.22		1.8 KBytes	0.5 %	0	0	1.8 KBytes	0	0	0	0	0	0	
192.168.2.219		1.6 KBytes	0.4 %	0	0	0	0	1.6 KBytes	0	0	0	0	
192.168.2.165		1.5 KBytes	0.4 %	0	0	0	0	1.1 KBytes	0	0	0	0	
224.0.0.252		984	0.3 %	0	0	0	0	0	0	0	0	0	
192.168.2.66		980	0.3 %	0	0	0	0	0	0	0	0	0	
your-a7454ae413 [NetBIOS]		438	0.1 %	0	0	0	0	438	0	0	0	0	
192.168.2.159		276	0.1 %	0	0	0	0	276	0	0	0	0	
ximena-esp@n [NetBIOS]		261	0.1 %	0	0	0	0	261	0	0	0	0	
supervisor [NetBIOS]		253	0.1 %	0	0	0	0	253	0	0	0	0	
yx-in-f17.google.com		228	0.1 %	0	228	0	0	0	0	0	0	0	
yw-in-f189.google.com		228	0.1 %	0	228	0	0	0	0	0	0	0	
192.168.2.189		192	0.1 %	0	0	0	0	0	0	0	0	0	
igmp.mcast.net		192	0.1 %	0	0	0	0	0	0	0	0	0	

Figura 5.2.2

5.2.3.- Nessus

TAREAS	REALIZADO (S/N)	OBSERVACIONES
Verificar que el servicio NESSUSD sube sin ningún error.	S	El servicio sube sin ningún problema
Realizar un scaneo de un host en la red.	S	El escaneo se realizó sin ningún problema.

Tabla 5.2.3

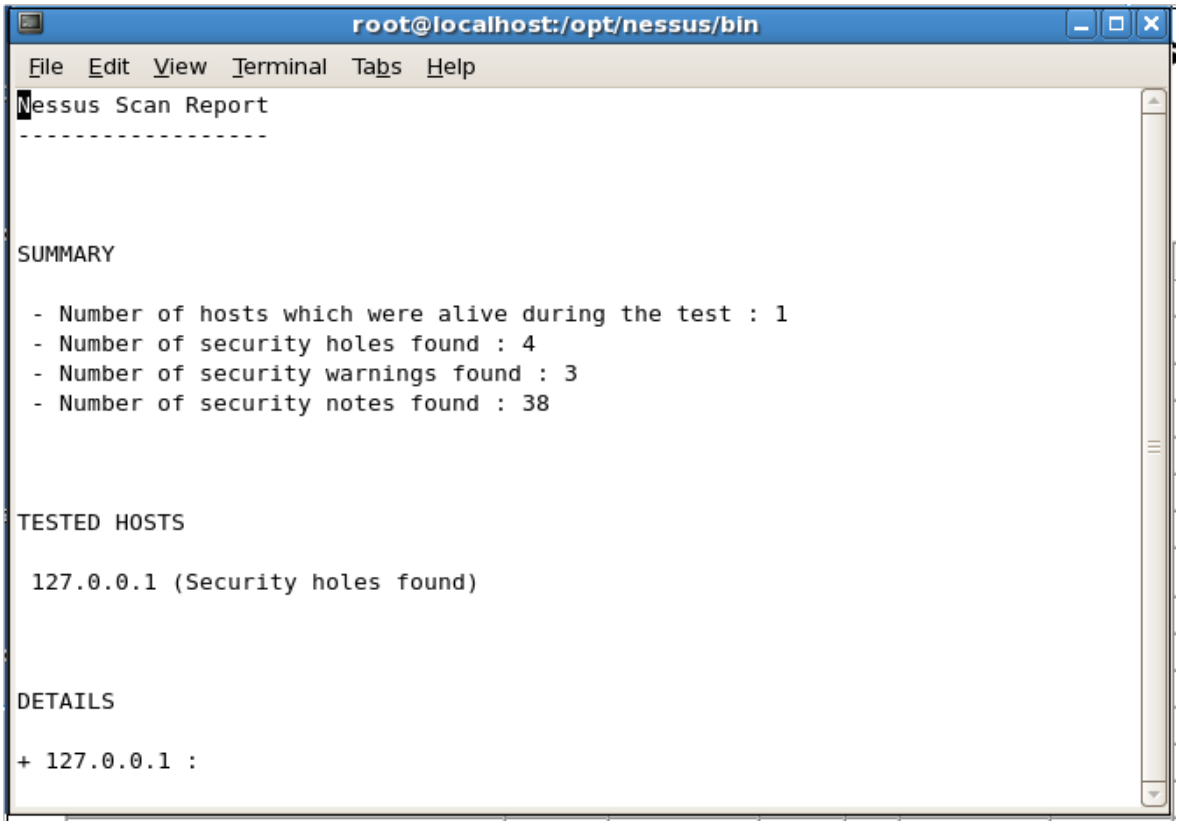

root@localhost:/opt/nessus/bin
File Edit View Terminal Tabs Help
Nessus Scan Report
SUMMARY
- Number of hosts which were alive during the test : 1
- Number of security holes found : 4
- Number of security warnings found : 3
- Number of security notes found : 38
TESTED HOSTS
127.0.0.1 (Security holes found)
DETAILS
+ 127.0.0.1 :

Figura 5.2.3

5.2.4.- IPS

TAREAS	REALIZADO (S/N)	OBSERVACIONES
Verificar que el servicio snort este activado	S	El servicio sube sin ningún problema
Ejecutar el comando snort –vd y verificar si captura el tráfico de red en pantalla	S	La herramienta captura el tráfico de paquetes sin ningún problema.

Tabla 5.2.4

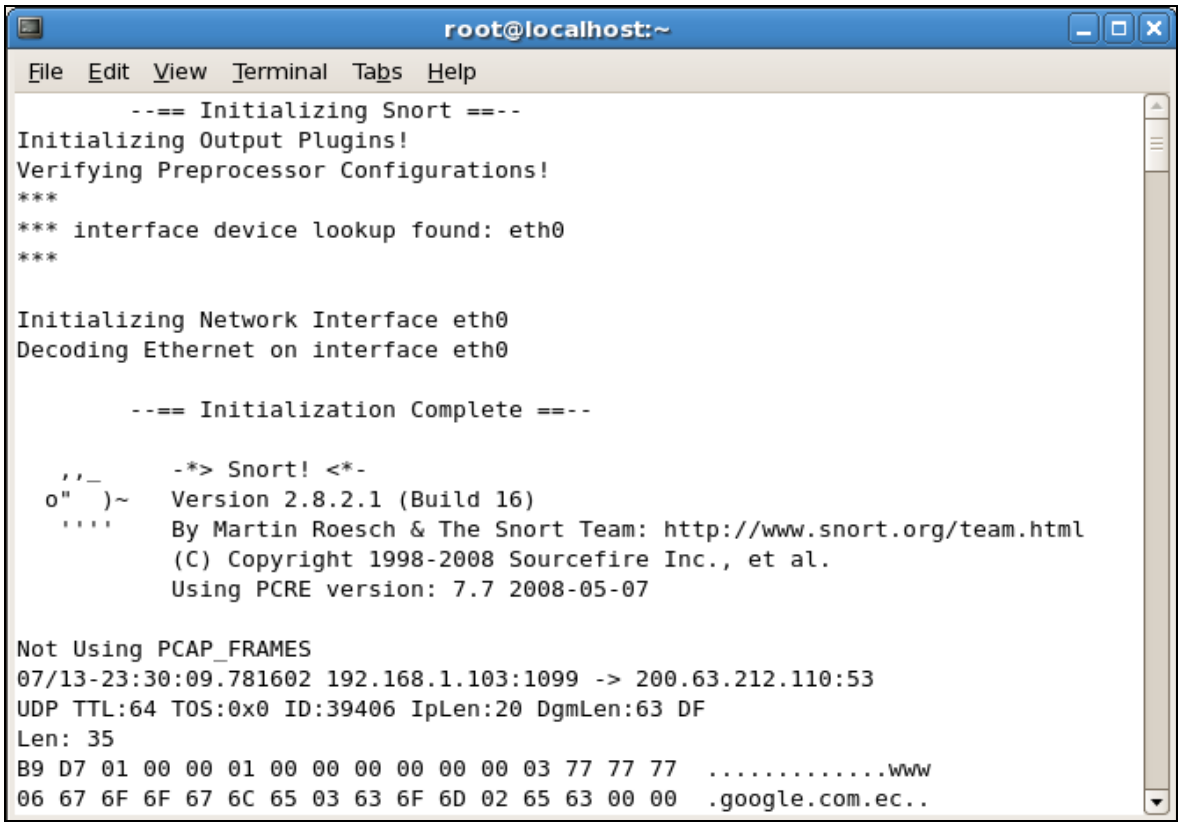


Figura 5.2.4

5.2.5.- OSSIM

TAREAS	REALIZADO (S/N)	OBSERVACIONES
Ingresar desde cualquier host a http://192.168.2.91 con el username y password correspondiente	S	El ingreso se dio sin ningún problema
Ingresar al menú Monitors → Network y verificar que NTOP se despliega sin problema	S	NTOP se despliega sin ningún incoveniente.
Ingresar al menú Monitors → Availability y verificar que el resúmen inicial de NAGIOS que aparece concuerde con los equipos ingresados previamente	S	NAGIOS se despliega sin problema y la configuración de equipos concuerda.
Ingresar al menú Events → Vulnerabilities y verificar que al escanear cualquier host, Nessus despliega resultados	S	NESSUS despliega el reporte sin problemas.

Tabla 5.2.5

Figura 5.2.5.1

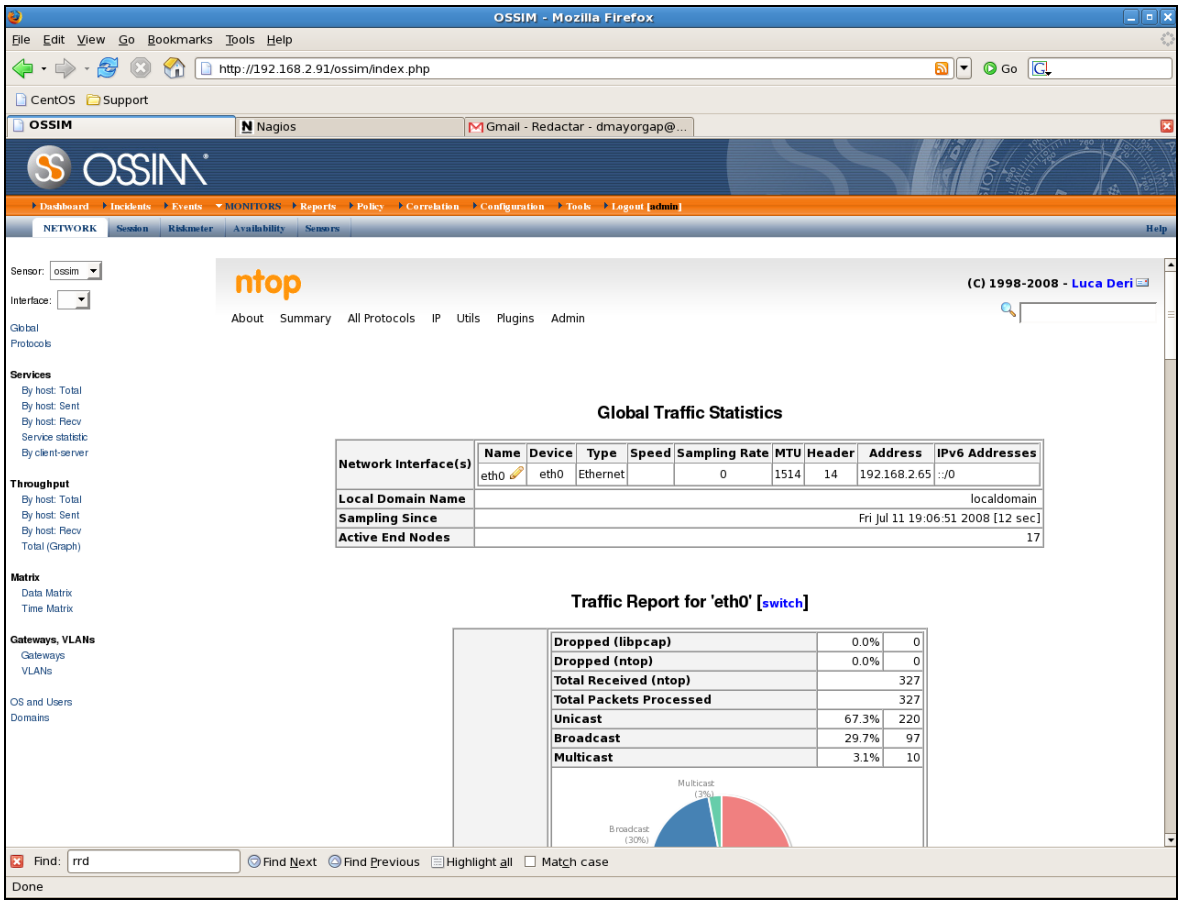

OSSIM - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://192.168.2.91/ossim/index.php
CentOS Support
OSSIM
Nagios
Gmail - Redactar - dmayorgap@...
OSSIM
Dashboard Incidents Events MONITORS Reports Policy Correlation Configuration Tools Logout [admin]
NETWORK Session Riskmeter Availability Sensors Help
Sensor: ossim
Interface:
Global
Protocols
Services
By host: Total
By host: Sent
By host: Recv
Service statistic
By client-server
Throughput
By host: Total
By host: Sent
By host: Recv
Total (Graph)
Matrix
Data Matrix
Time Matrix
Gateways, VLANs
Gateways
VLANs
OS and Users
Domains
ntop
(C) 1998-2008 - Luca Deri
About Summary All Protocols IP Utils Plugins Admin
Global Traffic Statistics
Network Interface(s)
Name Device Type Speed Sampling Rate MTU Header Address IPv6 Addresses
eth0 eth0 Ethernet 0 1514 14 192.168.2.65 ::/0
Local Domain Name localdomain
Sampling Since Fri Jul 11 19:06:51 2008 [12 sec]
Active End Nodes 17
Traffic Report for 'eth0' [switch]
Dropped (libpcap) 0.0% 0
Dropped (ntop) 0.0% 0
Total Received (ntop) 327
Total Packets Processed 327
Unicast 67.3% 220
Broadcast 29.7% 97
Multicast 3.1% 10
Multicast (3%)
Broadcast (30%)
Find: rrd
Find Next
Find Previous
Highlight all
Match case
Done

Figura 5.2.5.2

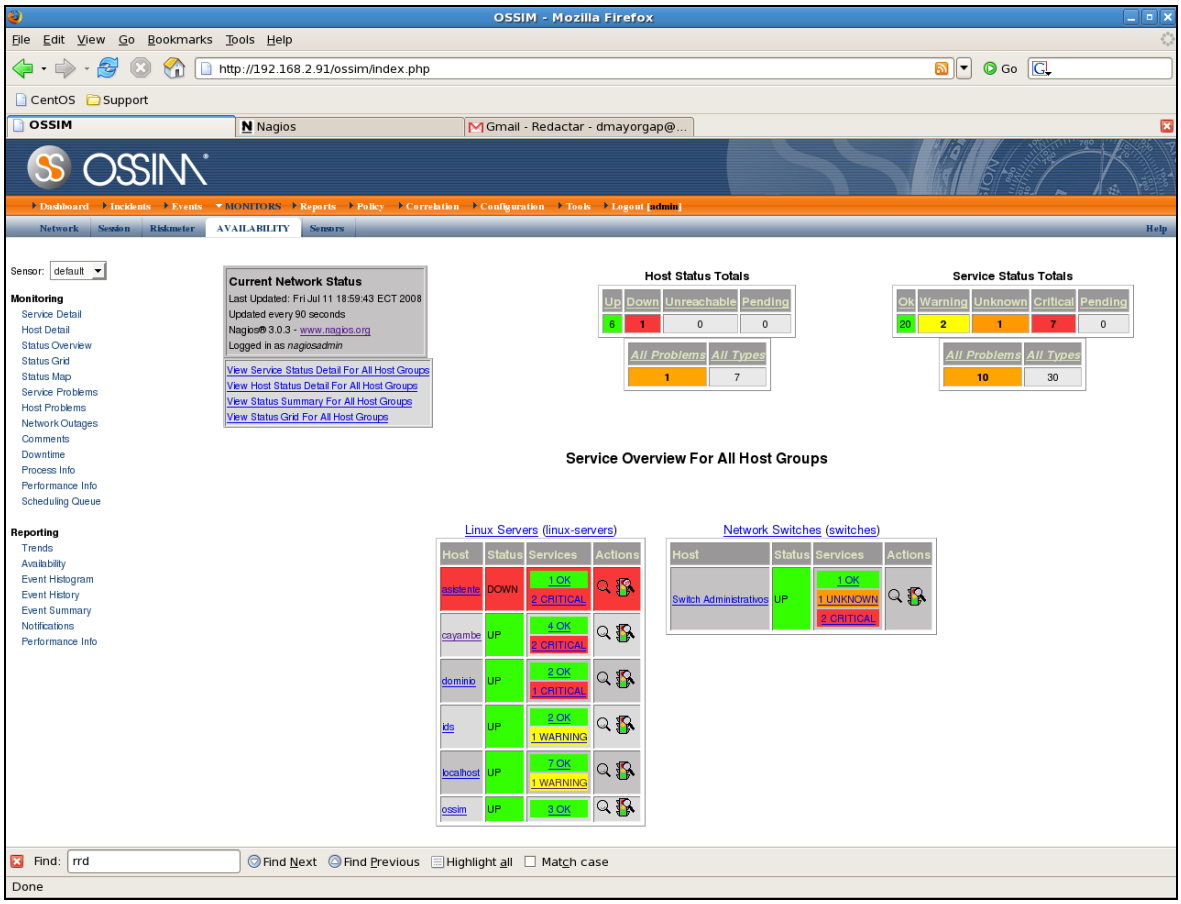


Figura 5.2.5.3

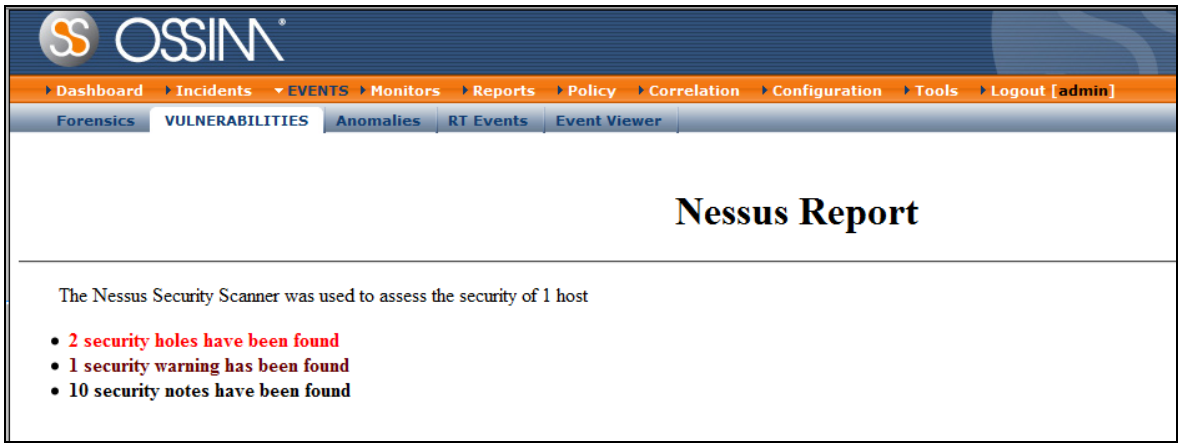


Figura 5.2.5.4

CAPITULO VI

DISEÑO DEL ESQUEMA DE MANTENIMIENTO Y MONITOREO DEL PERIMETRO DE SEGURIDAD DE LA RED

6.1.- ESQUEMA DE MONITOREO PROPUESTO

Se ha implementado un servidor destinado al monitoreo de la red de la UISEK Ecuador Campus Miguel de Cervantes, es importante diseñar un esquema de monitoreo de la red para tener la certeza de que el esquema de seguridad perimetral está funcionando correctamente y todavía se considera como “seguro”, ya que éste tiene un ciclo de vida que va acorde al avance de la tecnología en lo referente a seguridad y a ataques informáticos.

6.1.1.- Componentes de monitoreo

El servidor de monitoreo contiene las siguientes herramientas:

- NTOP
- NAGIOS

6.1.1.1.- Ntop

La herramienta NTOP sirve para el monitoreo del tráfico de red, es muy útil cuando se tiene tiempos de respuesta altos en la red y se sospecha que, por ejemplo, una estación de trabajo está consumiendo la mayor parte del ancho de banda y está causando un cuello de botella en la red. NTOP es fácil de usar y presenta información bastante útil como:

- **Sesiones TCP activas:** Detalle del origen y destino de transmisión de datos de cada host.
- **Matriz de conexión:** Detalle de las conexiones de hosts a servidores y la cantidad de tráfico intercambiado.
- **Monitoreo de los hosts específicos:** Detalle de la cantidad de tráfico intercambiada por un host específico.

La presentación de información de NTOP es intuitiva ya que los reportes presentan gráficos estadísticos y hay la posibilidad de guardar un histórico por fechas.

Se propone seguir el siguiente esquema de monitoreo para la herramienta NTOP:

- Monitorear las sesiones TCP activas diariamente en la hora pico del uso de red.
- Monitorear la matriz de conexión cuando se detecte que la red tiene tiempos de respuesta altos.
- Monitorear la matriz de tiempos semanalmente para verificar que el uso de la red sea mínimo fuera del horario laborable.

6.1.1.2.- Nagios

La herramienta NAGIOS sirve para monitorear la disponibilidad de hosts y servicios en una red, incluso se pueden monitorear componentes de red como switches. El uso de la herramienta es simple y no requiere mucho tiempo para familiarizarse con ella. El esquema de monitoreo de disponibilidad establecido es el siguiente:

EQUIPO	SERVICIOS	ALARMA
CAYAMBE	• HTTP → • PARTICION USUARIO ROOT → • SSH → • PROCESOS → • USUARIOS CONECTADOS → • PING →	• Servicio no esté subido • 80% lleno • Servicio no esté subido • Mas de 100 Procesos • Mas de 20 Usuarios • No responda
SRV. MONITOREO	• HTTP → • PARTICION USUARIO ROOT → • SSH → • PROCESOS → • USUARIOS CONECTADOS → • PING →	• Servicio no esté subido • 80% lleno • Servicio no esté subido • Mas de 100 Procesos • Mas de 20 Usuarios • No responda
SRV. IPS	• HTTP → • PROCESOS → • PING →	• Servicio no esté subido • Mas de 100 Procesos • No responda
SRV. OSSIM	• HTTP → • PROCESOS → • PING →	• Servicio no esté subido • Mas de 100 Procesos • No responda
EQUIPOS ASISTENTE ADM.	• HTTP → • PROCESOS → • PING →	• Servicio no esté subido • Mas de 100 Procesos • No responda
SWITCH ADMINISTRATIVOS	• PING →	• No responda

Tabla 6.1.1.2

6.1.1.2.1.- Tipos de alarmas

Se han configurado alarmas visuales y por correo electrónico en el servidor de monitoreo, las alarmas visuales se dividen en tres tipos:

COLOR	ESTADO
ROJO	Crítico
NARANJA	Desconocido
AMARILLO	Advertencia
VERDE	Ok

Tabla 6.1.1.2.1

Generalmente las alarmas de color AMARILLO se refieren a servicios no críticos pero que igualmente no responden y las de color NARANJA son activadas por problemas desconocidos, generalmente por problemas de falta de plugins en NAGIOS.

Si alguna alarma es de color ROJO, la configuración por defecto es que también se envie una alarma por correo electrónico, pero esto es configurable para los demás tipos de alarmas visuales.

Las notificaciones por correo electrónico son muy útiles ya que en caso de que algún incidente crítico ocurra, por ejemplo, si el ping al servidor Cayambe no tiene respuesta, se mostrará una alarma visual con un recuadro rojo en la consola de administración con el nombre del servidor Cayambe indicando que no responde al ping y puede estar caído, y también se enviará un correo electrónico a la dirección establecida para informar del incidente.

Ya que se trata de una herramienta automatizada, el único requisito para tener un buen monitoreo de disponibilidad de los equipos críticos, es actulizar la configuración de NAGIOS cuando se implementen nuevos equipos en la red considerados como críticos y tener actualizada la dirección mail definida para las alarmas. De todas maneras se propone un esquema de monitoreo de la disponibilidad:

- Monitorear diariamente todos los servicios de todos los hosts en busca de alarmas tipo AMARILLAS y NARANJAS**.**
- Tratar de resolver las alertas NARANJAS antes de la amarillas ya que se trata de un mal funcionamiento de algún plugin de NAGIOS.
- Resolver las alertas AMARILLAS aunque sean solo de advertencia para asegurar un monitoreo óptimo en todos los servicios.
- Crear un reporte de disponibilidad mensualmente del servidor CAYAMBE para verificar que el porcentaje no sea menor al 90%.

6.1.1.3.- Nessus

Es necesario escanear frecuentemente los hosts de toda la red para buscar vulnerabilidades en cada uno de ellos, generalmente éstas son causadas por la falta de actulización del sistema operativo de cada equipo, es por esto que se propone el siguiente esquema de monitoreo:

- Escanear mensualmente, fuera de horario laborable, todos los hosts de la red del Campus Miguel de Cervantes.
- Actualizar los sistemas operativos de los hosts con mayor número de vulnerabilidades.
- Actualizar semanalmente los plugins de Nessus para tenerlos actualizados y poder detectar las vulnerabilidades mas recientes.

CAPITULO VII

CONCLUSIONES Y RECOMENDACIONES

7.1.- CONCLUSIONES

- El análisis de riesgos realizado en el Campus Miguel de Cervantes evidenció varias fallas de seguridad, en el aspecto físico, de control de acceso, personal, manejo de incidentes, control de cambios, documentación y monitoreo.
- El riesgo más alto que existe en los sistemas IT del Campus es la alta temperatura a la que están sometidos los equipos debido a la falta de un sistema de control de temperatura.
- Se mejoró sustancialmente el esquema de seguridad informático en el Campus Miguel de Cervantes, ya que se combinó la seguridad activa mediante servidores dedicados a detectar e informar sobre posibles ataques y vulnerabilidades, con políticas de seguridad que marcan pautas para comunicar los incidentes de seguridad y tomar acciones inmediatas.
- Se aumentó el porcentaje de disponibilidad de la red de datos del Campus implementado servidores de monitoreo y un plan de contingencia para restaurar las operaciones en el menor tiempo posible en caso de desastres naturales.
- La seguridad de los sistemas informáticos del Campus pasaron de un ámbito reactivo, sin ninguna planificación en caso de incidentes, al ámbito proactivo resolviendo posibles amenazas antes de que ocurran.
- Se redujeron los tiempos de notificación, análisis y detección de fallas cuando se produce un incidente en la red debido a problemas de hardware o servicios de red gracias a las herramientas gráficas de monitoreo de tráfico y disponibilidad.
- Se aumentó el nivel de reacción ante cualquier falla de los componentes de red mediante las notificaciones por correo electrónico al administrador de red.

7.2.- RECOMENDACIONES

- Se recomienda implementar el esquema de seguridad perimetral propuesto para mejorar la organización y seguridad en la red del Campus.
- Es importante realizar pruebas de penetración en la red del Campus para tener un mejor conocimiento de los ataques que se perfilen como muy probables para la red del Campus.
- Es muy importante implementar la medidas de seguridad seleccionadas de la norma ISO/IEC 13335:4 que se proponen en este documento.
- Se recomienda implementar un sistema de control de temperatura ya que la probabilidad de que ocurra un fallo en algún equipo de la red por éste motivo es alta.
- Es recomendable empezar a utilizar el manual de políticas de seguridad desarrollado en éste trabajo de investigación.
- Se recomienda implementar un SMTP Gateway en el segmento de red DMZ para adicionar seguridad en caso de implementar un servidor de correo electrónico en la red del Campus.
- Se recomienda segmentar la red de datos por facultad para mejorar el tráfico de red en caso de que el Campus de Guapulo se traslade al Campus de Carcelén.
- Es muy importante destinar más recursos por parte de las autoridades de la Institución para optimizar y asegurar los activos críticos del Campus para minimizar el nivel de afectación al negocio en caso de incidentes.
- Es altamente recomendable desarrollar documentación sobre configuraciones y control de cambios de la red para evitar fallas por el desconocimiento de información sobre ésta.

CAPITULO VIII

BIBLIOGRAFIA

- INSIDE Network Perimeter Security por Stephen Northcutt, Lenny Zeltser, Scott Winters, Karen Kent, Ronald W. Ritchey. USA, Sams Publishing, 2005. 768p.

- INTERNATIONAL Organization for Standarization. ISO/IEC 27001-2005: Requirements. 2005. 42p.

- INTERNATIONAL Organization for Standarization. ISO/IEC 27002-2005: Code of practice for information security management. 2da Edición. 2005. 128p.

- INTERNATIONAL Organization for Standarization. ISO/IEC 13335-3: Techniques for the Management of IT Security. 1ra Edición. 1998. 64p.

- INTERNATIONAL Organization for Standarization. ISO/IEC 13335-4: Selection of Safeguards. 1ra Edición. 2000. 80p.

- LÓPEZ, Wilmer. Memoria Técnica Reestructuración del Sistema de Cableado Estructurado Universidad Internacional SEK Campus “Miguel de Cervantes”. <MEMORIA_TECNICA_RED_CARCELEN.pdf>. 2007. 23p.

- NAGIOS. Service and Network Monitor Program. [en línea] < http://www.nagios.org/docs/> [consulta : 11 julio 2008]

- NTOP. Netowork Monitor Program. [en línea] < http://www.ntop.org/documentation.html> [consulta : 09 julio 2008]

- OSSIM. Open Source Security Information Management. [en línea] <http://www.ossim.net/dokuwiki/doku.php?id=documentation:general_description > [consulta : 02 junio 2008]

- SANTOS, Omar. End-to-End Network Security Defense-in-Depth. Indianapolis, USA, Cisco Press, 2008. 469p.

- TENABLE Network Security. Nessus Vulnerability Scanner. [en línea] < http://www.nessus.org/documentation/> [consulta : 11 julio 2008]

CAPITULO IX

GLOSARIO DE TÉRMINOS

CERT: Las siglas significan Computer Emergency Response Team, es un programa conformado por especialistas que se dedican a investigar las brechas de seguridad informática existentes.

Confidencialidad: es la propiedad de un documento o mensaje que únicamente está autorizado para ser leído o entendido por algunas personas o entidades.
Se dice que un documento o mensaje es confidencial si éste sólo está autorizado a ser leído o entendido por un destinatario designado.

Deep-packet firewall: Es un tipo de firewall que analiza los datos y cabezeras que forman parte de los paquetes que viajan a través de la red.

FBI: Las siglas significan Federal Bureau of Investigation, es el brazo investigador del Departamento de Justicia de los Estados Unidos. La principal responsabilidad del FBI es encargarse de la inteligencia en suelo estadounidense.

Firewall: Elemento de hardware o software utilizado en una red de datos para controlar las comunicaciones, filtrando los accesos que entran y salen según las políticas de red que haya definido el departamento de tecnología de la organización.

Gusano: Es un virus informático que tiene la propiedad de duplicarse a sí mismo. Los gusanos utilizan las partes automáticas de un sistema operativo que generalmente son invisibles al usuario. A diferencia de un virus, un gusano no precisa alterar los archivos de programas, sino que reside en la memoria y se duplica a sí mismo. Los gusanos siempre dañan la red (aunque sea simplemente consumiendo ancho de banda), mientras que los virus siempre infectan o corrompen los archivos de la computadora que atacan.

IDS (Intrusion Detection System): Un sistema de detección de intrusos es un programa usado para detectar accesos desautorizados a un computador o a una red.

IPS (Intrusion Prevention System): Es un programa usado para prevenir accesos desautorizados a un computador o a una red.

ISP: Acrónimo de Internet Service Provider, es una empresa dedicada a conectar a Internet a los usuarios o las distintas redes que tengan, y dar el mantenimiento necesario para que el acceso funcione correctamente.

LAN: Significa Local Area Network, es la interconexión de varios ordenadores y periféricos, su extensión esta limitada físicamente a un entorno de hasta 100 metros. Su aplicación más extendida es la interconexión de computadores personales y estaciones de trabajo en oficinas, hogares etc., para compartir recursos e intercambiar datos y aplicaciones.

Malware: La palabra malware proviene de una agrupación de las palabras (malicious software). Existen muchísimos tipos de malware, aunque algunos de los más comunes son los virus, los gusanos, los troyanos, los programas de spyware/adware. Estos programas tienen como objetivo recopilar información sobre un usuario, un PC, etc., o con los peores fines también se utilizan para dañar los computadores así como la información contenida en ella.

Ruteador: Un ruteador es un dispositivo hardware o software de interconexión de redes de computadoras que opera en la capa tres (nivel de red) del modelo OSI. Este dispositivo interconecta segmentos de red o redes enteras. Transmite paquetes de datos entre redes tomando como base la información de la capa de red.

Software: Es el conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

Spam: Se llama spam o correo basura a los mensajes no solicitados, habitualmente de tipo publicitario, enviados en grandes cantidades que perjudican de alguna o varias maneras al receptor, la manera mas común para enviar éste tipo de mensajes es por correo electrónico.

Sprinkler: Es una medida de protección activa contra incendios que consiste en una instalación de tuberías de agua a las cuales los dispositivos sprinklers están conectados, cuando éstos detectan fuego se activan y lanzan agua.

Stateful firewall: Es un tipo de firewall que analiza el estado de las conexiones de red, generalmente TCP y UDP, que se transmiten a través de la red.

Troyano: Es un programa malicioso capaz de alojarse en computadoras y permitir el acceso a usuarios externos, a través de una red local o de Internet, con el fin de recabar información o controlar remotamente a la máquina anfitriona. Un troyano no es en sí un virus, aún cuando teóricamente pueda ser distribuido y funcionar como tal. La diferencia fundamental entre un troyano y un virus consiste en su finalidad. Para que un programa sea un "troyano" solo tiene que acceder y controlar la máquina anfitriona sin ser advertido, normalmente bajo una apariencia inocua. Al contrario que un virus, que es un huésped destructivo, el troyano no necesariamente provoca daños porque no es su objetivo.

Virus: es un programa que se copia automáticamente y que tiene por objeto alterar el normal funcionamiento de la computadora, sin el permiso o el conocimiento del usuario. Aunque popularmente se incluye al "malware" dentro de los virus, en el sentido estricto de esta ciencia los virus son programas que se replican y ejecutan por sí mismos. Los virus, habitualmente, reemplazan archivos ejecutables por otros infectados con el código de éste. Los virus pueden destruir, de manera intencionada, los datos almacenados en un ordenador, aunque también existen otros más "benignos", que solo se caracterizan por ser molestos.

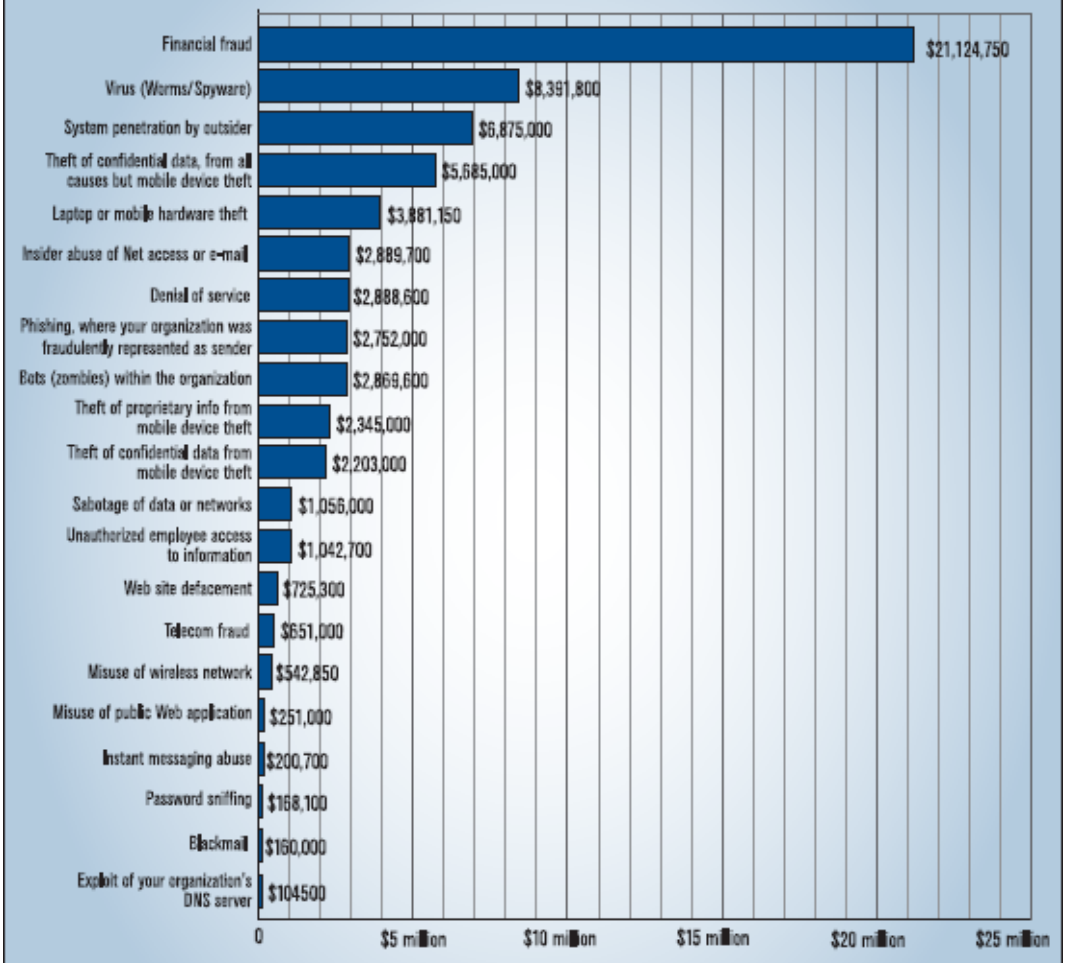
CAPITULO X

ANEXOS

ANEXO 1

Monto de pérdidas por ataques informáticos

Figure 16. Dollar Amount Losses by Type of Attack



Total Losses for 2007 = $66,930,950

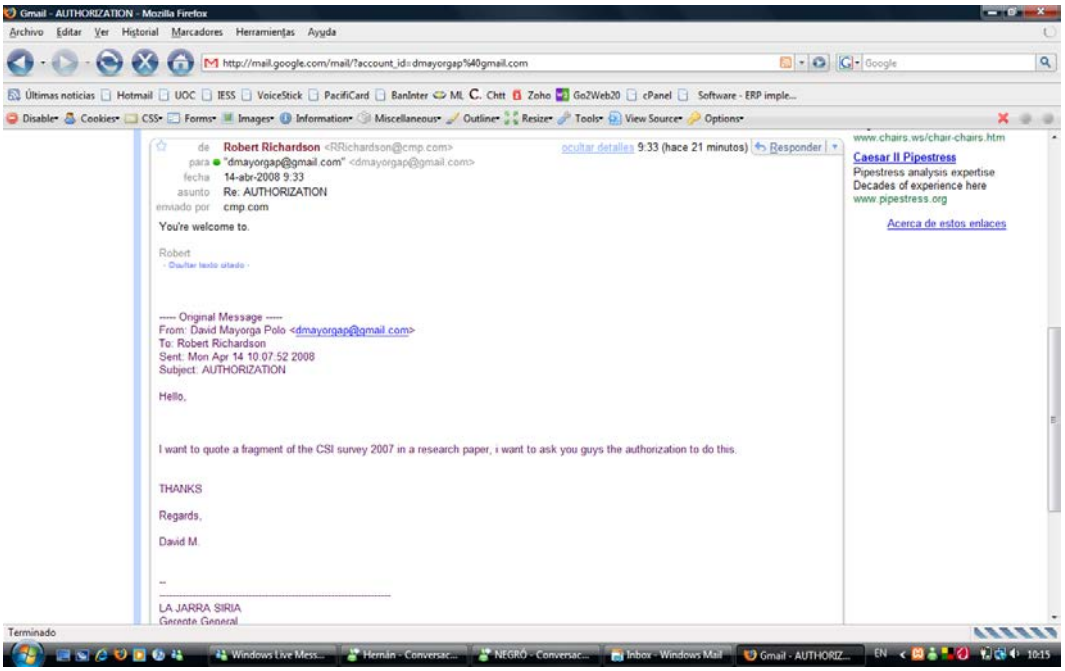
(Numbers above do not equal total due to rounding.)

CSI 2007 Computer Crime and Security Survey
Source: Computer Security Institute

2007: 194 Respondents

ANEXO 2

Autorización para publicar ANEXO 1


Gmail - AUTHORIZATION - Mozilla Firefox
Archivo Editar Ver Historial Marcadores Herramientas Ayuda
http://mail.google.com/mail/?account_id=dmayorgap%40gmail.com
Robert Richardson <RRichardson@cmp.com>
ocultar detalles 9:33 (hace 21 minutos) Responder
para "dmayorgap@gmail.com" <dmayorgap@gmail.com>
fecha 14-abr-2008 9:33
asunto Re: AUTHORIZATION
enviado por cmp.com
You're welcome to.
Robert
----- Original Message -----
From: David Mayorga Polo <dmayorgap@gmail.com>
To: Robert Richardson
Sent: Mon Apr 14 10:07:52 2008
Subject: AUTHORIZATION
Hello,
I want to quote a fragment of the CSI survey 2007 in a research paper, i want to ask you guys the authorization to do this.
THANKS
Regards,
David M.
LA JARRA SIRIA
Gerente General
www.chairs.ws/chair-chairs.htm
Caesar II Pipestress
Pipestress analysis expertise
Decades of experience here
www.pipestress.org
Acerca de estos enlaces
Terminado

ANEXO 3

Nombres y direcciones IP para equipos de Laboratorio SISTEMAS 1

LABORATORIO SISTEMAS 1	
EQUIPO	**IP**
SIS1-01	192.168.2.90
SIS1-02	192.168.2.91
SIS1-03	192.168.2.92
SIS1-04	192.168.2.93
SIS1-05	192.168.2.94
SIS1-06	192.168.2.95
SIS1-07	192.168.2.96
SIS1-08	192.168.2.97
SIS1-09	192.168.2.98
SIS1-10	192.168.2.99
SIS1-11	192.168.2.100

Tabla 3.2.8.1.1

ANEXO 4

Nombres y direcciones IP para equipos de Laboratorio SISTEMAS 2

LABORATORIO SISTEMAS 2	
EQUIPO	**IP**
SIS2-01	192.168.2.65
SIS2-02	192.168.2.66
SIS2-03	192.168.2.67
SIS2-04	192.168.2.68
SIS2-05	192.168.2.69
SIS2-06	192.168.2.70
SIS2-07	192.168.2.71
SIS2-08	192.168.2.72
SIS2-09	192.168.2.73
SIS2-10	192.168.2.74
SIS2-11	192.168.2.75
SIS2-12	192.168.2.76
SIS2-13	192.168.2.77
SIS2-14	192.168.2.78
SIS2-15	192.168.2.79
SIS2-16	192.168.2.80
SIS2-17	192.168.2.81
SIS2-18	192.168.2.82
SIS2-19	192.168.2.83
SIS2-20	192.168.2.84
SIS2-21	192.168.2.85
SIS2-22	192.168.2.86
SIS2-23	192.168.2.87
SIS2-24	192.168.2.88

Tabla 3.2.8.1.2

ANEXO 5

Nombres y direcciones IP para equipos de Laboratorio FINANZAS

LABORATORIO DE FINANZAS	
EQUIPO	**IP**
FIN01	192.168.2.20
FIN02	192.168.2.21
FIN03	192.168.2.22
FIN04	192.168.2.23
FIN05	192.168.2.24
FIN06	192.168.2.25
FIN07	192.168.2.26
FIN08	192.168.2.27
FIN09	192.168.2.28
FIN10	192.168.2.29
FIN11	192.168.2.30
FIN12	192.168.2.31
FIN13	192.168.2.32
FIN14	192.168.2.33
FIN15	192.168.2.34
FIN16	192.168.2.35
FIN17	192.168.2.36
FIN18	192.168.2.37
FIN19	192.168.2.38
FIN20	192.168.2.39
FIN21	192.168.2.40
FIN22	192.168.2.41
FIN23	192.168.2.42
FIN24	192.168.2.43
FIN25	192.168.2.44
FIN26	192.168.2.45

Tabla 3.2.8.1.3

ANEXO 6

Nombres y direcciones IP para equipos de Laboratorio COMUNICACIÓN

LABORATORIO DE COMUNICACIÓN	
EQUIPO	**IP**
COM1	192.168.2.50
COM2	192.168.2.51
COM3	192.168.2.52
COM4	192.168.2.53
COM5	192.168.2.54
COM6	192.168.2.55
COM7	192.168.2.56
COM8	192.168.2.57
COM9	192.168.2.58
COM10	192.168.2.59
COM11	192.168.2.60

Tabla 3.2.8.1.4

ANEXO 7

Nombres y direcciones IP para equipos de BIBLIOTECA

BIBLIOTECA	
EQUIPO	**IP**
BIB-01	192.168.2.2
BIB-02	Red Inalámbrica
BIB-03	Red Inalámbrica
BIB-04	Red Inalámbrica
BIB-05	Red Inalámbrica
BIB-06	Red Inalámbrica
BIB-07	Red Inalámbrica
BIB-08	Red Inalámbrica
BIB-09	Red Inalámbrica
BIB-10	Red Inalámbrica

Tabla 3.2.8.1.5

ANEXO 8

Nombres y direcciones IP para equipos de personal ADMINISTRATIVO

ADMINISTRATIVOS	
EQUIPO	**IP**
Supervisor	192.168.2.1
Reservado	192.168.2.5
Supervisor Ambiental	192.168.2.9
Secretaría	192.168.2.211

Tabla 3.2.8.1.6

ANEXO 9

Nombres y direcciones IP para equipos de DECANATOS

DECANATOS	
EQUIPO	**IP**
Dec. Sistemas	192.168.2.3
Dec. Finanzas	192.168.2.4

Tabla 3.2.8.1.7

ANEXO 10

Nombres y direcciones IP para equipos de aulas de TUTORÍAS

TUTORÍAS	
EQUIPO	**IP**
Tutoría 1	192.168.2.6
Tutoría 2	192.168.2.7
Tutoría 3	192.168.2.10
Tutoría 4	192.168.2.11

Tabla 3.2.8.1.8

ANEXO 11

Configuración de Firewall de Servidor CAYAMBE

```
#!/bin/sh
# iptables, by Technion
# $Id: iptables,v 1.33 2002/11/20 23:22:16 technion Exp $
# chkconfig: 2345 08 80
# description: Script for setting IPTABLES rules
# processname: iptables
# Traduccion al espanol:
# Ing. Wilmer Lopez


#  Si se ejecuta en Redhat, poner en YES, sino, en NO
REDHAT="YES"

# Informacion de Red, necesita ser ajustada por usted.
INTERNALIF="eth0"
INTERNALNET="192.168.XXX.XXX/24"
INTERNALBCAST="192.168.XXX.XXX"
EXTERNALIF="eth1"
MYADDR="XXX.XXX.XXX.XXX"  # Sólo se necesita en caso de hacerse DNAT.
                          # Port Forwarding
                          # Sino, dejar comentado.

# Caminos - No tocar si no se conoce.
DMESG="/bin/dmesg"
IPTABLES="`which iptables`"
MODPROBE="/sbin/modprobe"

# Esta parte permite al usuario de redhat llamar al script con
# los parametros: start/stop/restart
if [ X"$REDHAT" = X"YES" ]; then
       . /etc/rc.d/init.d/functions
       case "$1" in
              stop)
                     action "Shutting down firewall:" echo
                     $IPTABLES -F
                     $IPTABLES -P FORWARD DROP
                     exit 0
                     ;;
              status)
                     echo "The status command is not supported for iptables"
                     exit 0
                     ;;
              restart|reload)
                     $0 stop
                     exec $0 start
                     ;;
              start)
                     action "Starting Firewall:" echo
                         ;;
                 *)
                         echo "Usage: firewall (start|stop|restart)"
                         exit 1
        esac
fi


#####################################################################
#Insertando los modulos. Debe hacerse automaticamente si se requiere
dmesg -n 1 #No saca la bandera de copyright
/sbin/modprobe ip_tables
/sbin/modprobe iptable_filter
/sbin/modprobe ip_conntrack
/sbin/modprobe ip_conntrack_ftp
#
## Vaciar las reglas que puedan existir
```

```
#
# Paquetes provenientes del exterior
$IPTABLES -F INPUT
# Paquetes de la red interna hacia el exterior
$IPTABLES -F OUTPUT
# Forwarding/enmascaramiento
$IPTABLES -F FORWARD
#Tabla de NAT
$IPTABLES -t nat -F

#
#Deshabilitando ataques spoofing
#Comente esta linea si se usa IPSEC
echo 2 > /proc/sys/net/ipv4/conf/all/rp_filter

#No responder a pings de broadcast
echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts

#No responder a los pings - Activar si se requiere
#echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all

#Habilitando el forwarding
echo 1 >/proc/sys/net/ipv4/ip_forward

#Normalmente usted no tiene control de la ruta que un paquete sigue.
#IP Source Routing (SRR) es una forma de especificarle a un paquete la ruta
#exacta que un paqeute debe seguir. Habilitarlo es una mala idea, en forma
#general. Se desactiva así:
echo 0 >/proc/sys/net/ipv4/conf/all/accept_source_route

#Anular los TimeStamps. Se puede averiguar el uptime de un sistema
echo 0 > /proc/sys/net/ipv4/tcp_timestamps

#Habilitar SYN Cookies, previene algunos DoS
#echo 1 > /proc/sys/net/ipv4/tcp_syncookies

#Dehabilitar redirects. Habilitar si no se actua como router
echo 0 >/proc/sys/net/ipv4/conf/all/accept_redirects

#Habilitar proteccion contra mensaje invalido
#Algunos routers envian mensajes invalidos que son guardados en
#log logs por el kernel. Este punto le indica al kernel no guardar
#estos datos y se protege contra un posible ataque de negacion
#debido a tener el filesystem lleno
#echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses

#Permitir direcciones dinamicas, Bueno para clientes dialup y xDSL cuyas IPs
#pueden variar. No es necesario para clientes dedicados
#echo "1" > /proc/sys/net/ipv4/ip_dynaddr

#Guardar a los marcianos. Paquetes que llegan en una interfaz no esperada
#Algunas tarjetas de red se niegan a trabajar. Comentar si es el caso
#echo 1 >/proc/sys/net/ipv4/conf/all/log_martians

#Definir rango de puertos locales a ser usados por nuestras aplicaciones
echo "32768 61000" >/proc/sys/net/ipv4/ip_local_port_range

#Reducir posibilidad de DoS al reducir los timeouts
#Tiempo en el que linux tratará de finalizar una conexion
echo 30 > /proc/sys/net/ipv4/tcp_fin_timeout
#Tiempo para finalizar una conexión no activa
echo 1800 > /proc/sys/net/ipv4/tcp_keepalive_time
echo 1 > /proc/sys/net/ipv4/tcp_window_scaling
echo 0 > /proc/sys/net/ipv4/tcp_sack
#Maximo de conexiones SYN
echo 1280 > /proc/sys/net/ipv4/tcp_max_syn_backlog
```

```
##Activando reglas basicas

#Eliminar todo paquete sin sentido, incluye:
#-Paquetes muy pequeños para tener un encabezamiento completo ICMP/UDP/TCP
#- Paquetes TCP y UDP con cero (invalidos) puertos de origen y destino
#-Combinaciones ilegales de banderas TCP
#-Opciones TCP e IP con cero (invalido) o sobredimensionada
#      o opciones despues de la opcion END-OF-OPTIONS
#-Fragmentos con longitud ilegal (ej., Ping of Death).
#Tomado de: http://www.linux-mag.com/2000-01/bestdefense_02.html


#Matar paquetes con combinaciones de banderas invalidas
$IPTABLES -A INPUT -m state --state INVALID -j DROP
$IPTABLES -A FORWARD -m state --state INVALID -j DROP


# Permitir todas las conexiones en la interfaz local
#
$IPTABLES -A INPUT -i lo -j ACCEPT

#Eliminar conexiones a la interfaz local desde el mundo exterior
$IPTABLES -A INPUT -d 127.0.0.0/8 -j REJECT


#Permitir trafico ilimitado de la red interna que usan direcciones validas
$IPTABLES -A INPUT -i $INTERNALIF -s $INTERNALNET -j ACCEPT
#
#IPv6 tunneling
#$IPTABLES -A INPUT -p ipv6 -j ACCEPT

#IPSEC tunneling
#$IPTABLES -A INPUT -p 50 -j ACCEPT
#Permitir trafico del dispositivo IPSEC a la red interna
#$IPTABLES -A FORWARD -i ipsec0 -o $INTERNALIF -j ACCEPT
#Permitir negociacion de claves
#$IPTABLES -A INPUT -p udp --dport 500 -j ACCEPT

#Kill anything from outside claiming to be from internal network
$IPTABLES -A INPUT -i $EXTERNALIF -s $INTERNALNET -j REJECT

##ICMP
#No enviar pings yendo hacia dentro
#$IPTABLES -A FORWARD -p icmp --icmp-type echo-request -o $INTERNALIF -j REJECT
#Protección contra ping flood
$IPTABLES -A INPUT -p icmp --icmp-type echo-request -m limit --limit 10/s -j
ACCEPT
#$IPTABLES -A INPUT -p icmp --icmp-type echo-request -j DROP
#Denegar icmp a direcciones broadcast
#$IPTABLES -A INPUT -p icmp -d $INTERNALBCAST -j DROP

#Permitir el resto de los ICMP
$IPTABLES -A INPUT -p icmp -j ACCEPT

##Permitir conexiones establecidas desde dentro
$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

#No reenviar trafico SMB
$IPTABLES -A FORWARD -o $EXTERNALIF -p tcp --dport 137 -j REJECT
$IPTABLES -A FORWARD -o $EXTERNALIF -p tcp --dport 138 -j REJECT
$IPTABLES -A FORWARD -o $EXTERNALIF -p tcp --dport 139 -j REJECT
$IPTABLES -A FORWARD -o $EXTERNALIF -p udp --dport 137 -j REJECT
$IPTABLES -A FORWARD -o $EXTERNALIF -p udp --dport 138 -j REJECT
$IPTABLES -A FORWARD -o $EXTERNALIF -p udp --dport 139 -j REJECT
$IPTABLES -A INPUT -i $EXTERNALIF -p udp --dport 137 -j REJECT
#
#Permitir el resto del trafico salir
#$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -j ACCEPT
```

```
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -s 192.168.XXX.XXX -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -s 192.168.XXX.XXX -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -s 192.168.XXX.XXX -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 80 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 80 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 81 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 81 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 25 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 25 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 53 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 53 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 110 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 110 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 22 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 23 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 443 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 10000 -j ACCEPT
#$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 1433 -#j
ACCEPT
#$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 1433 -#j
ACCEPT
#$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 3389 -#j
ACCEPT
#$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 3389 -#j
ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 21 -j ACCEPT
#$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 2000 -#j
ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 3128 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 3128 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p tcp --dport 8080 -j ACCEPT
$IPTABLES -A FORWARD -o $EXTERNALIF -i $INTERNALIF -p udp --dport 8080 -j ACCEPT



#Permitir las respuestas entrar
$IPTABLES -A FORWARD -i $EXTERNALIF -m state --state ESTABLISHED,RELATED -j
ACCEPT

#Permitir DHCP - Descomentar sólo si se usa DHCP y no funciona normalmente
#$IPTABLES -A INPUT -p udp -d 255.255.255.255 --dport 68 -j ACCEPT

#Permitir a esta maquina ser un servidor DHCP para su red interna
#$IPTABLES -A INPUT -i $INTERNALIF -p tcp --sport 68 --dport 67 -j ACCEPT
#$IPTABLES -A INPUT -i $INTERNALIF -p udp --sport 68 --dport 67 -j ACCEPT

#Permitir paquetes de DNS. Algunas verciones de IPTABLES parece que tienen
#errores aqui.
#Comentar si fuese necesario
# cat /etc/resolv.conf | \
# awk '/^nameserver/ {print $2}' | \
# xargs -n1 $IPTABLES -A INPUT -p udp --sport 53 -j ACCEPT -s

#De aqui en adelante, trabajamos con intentos de conexion.
#El -m limit es una protecciones contra DoS
#Primero permitimos una cierta cantidad de conexiones/Segundo
#DROP - Descartamos al resto, de tal forma que no nos neguemos nosotros mismos
#el servicio
##Servicios basicos. Descomentar para permitirlos
#Estos son puertos que estamos abriendo al exterior!!
#Si los vamos a usar en la red interna, no es necesario abrirlos aqui
#Solo para abrirlos a la red externa, a la internet

# ftp-data
#$IPTABLES -A INPUT -p tcp  --dport 20 -j ACCEPT

# ftp
```

```
#$IPTABLES -A INPUT -p tcp  --dport 21 -j ACCEPT

# ssh
$IPTABLES -A INPUT -p tcp --dport 22 -j ACCEPT

#telnet
#$IPTABLES -A INPUT -p tcp --dport 23 -j ACCEPT

# smtp  Una conexion por segundo
$IPTABLES -A INPUT -p tcp --dport 25 --syn -m limit --limit 8/s \
        --limit-burst 10 -j ACCEPT
$IPTABLES -A INPUT -p tcp --dport 25 --syn -j DROP
$IPTABLES -A INPUT -p tcp --dport 25 -j ACCEPT

#nessus
#$IPTABLES -A INPUT -p tcp --dport 1241 -j ACCEPT

# DNS
$IPTABLES -A INPUT -p tcp --dport 53 -j ACCEPT
$IPTABLES -A INPUT -p udp --dport 53 -j ACCEPT

# http
$IPTABLES -A INPUT -p tcp --dport 80 -j ACCEPT

# POP-3
$IPTABLES -A INPUT -p tcp --dport 110 -j ACCEPT

# identd
#$IPTABLES -A INPUT -p tcp --dport 113 -j ACCEPT

# https
$IPTABLES -A INPUT -p tcp --dport 443 -j ACCEPT

#VNC Server
#$IPTABLES -A INPUT -p tcp --dport 5801 -j ACCEPT
#$IPTABLES -A INPUT -p tcp --dport 5901 -j ACCEPT
#$IPTABLES -A INPUT -p tcp --dport 6001 -j ACCEPT

#Webmin
$IPTABLES -A INPUT -p tcp --dport 10000 -j ACCEPT

#
##DNAT
#Modificar las direcciones y descomentar para hacer forwarding (DNAT)

#Enviar peticiones web a una maquina interna (192.168.0.10 ?)
#Send mail to an internal machine
$IPTABLES -A PREROUTING -t nat -i $EXTERNALIF -p tcp -d $MYADDR --dport 80 \
                      -j DNAT --to 192.168.2.251:80
$IPTABLES -A FORWARD -i $EXTERNALIF -p tcp -d 192.168.2.251 --dport 80 -j ACCEPT


$IPTABLES -A PREROUTING -t nat -i $EXTERNALIF -p tcp -d $MYADDR --dport 2000 \
                      -j DNAT --to 192.168.2.251:22
$IPTABLES -A FORWARD -i $EXTERNALIF -p tcp -d 192.168.2.251 --dport 22 -j ACCEPT

#Smtp a un servidor interno

#$IPTABLES -A PREROUTING -t nat -i $EXTERNALIF -p tcp -d $MYADDR --dport 25 \
#                       -j DNAT --to 192.168.0.10:25
#$IPTABLES -A FORWARD -i $EXTERNALIF -p tcp -d 192.168.0.10 --dport 25 -j ACCEPT


##Puertos que deben ser denegados y guardados
$IPTABLES -A INPUT -p tcp --dport 1433 -m limit -j LOG \
                          --log-prefix "Firewalled packet: MSSQL "

$IPTABLES -A INPUT -p tcp --dport 1433 -j DROP
```

```
$IPTABLES -A INPUT -p tcp --dport 6670 -m limit -j LOG \
                          --log-prefix "Firewalled packet: Deepthrt "
$IPTABLES -A INPUT -p tcp --dport 6670 -j DROP
$IPTABLES -A INPUT -p tcp --dport 6711 -m limit -j LOG \
                          --log-prefix "Firewalled packet: Sub7 "
$IPTABLES -A INPUT -p tcp --dport 6711 -j DROP
$IPTABLES -A INPUT -p tcp --dport 6712 -m limit -j LOG \
                          --log-prefix "Firewalled packet: Sub7 "
$IPTABLES -A INPUT -p tcp --dport 6712 -j DROP
$IPTABLES -A INPUT -p tcp --dport 6713 -m limit -j LOG \
                          --log-prefix "Firewalled packet: Sub7 "
$IPTABLES -A INPUT -p tcp --dport 6713 -j DROP

$IPTABLES -A INPUT -p tcp --dport 12345 -m limit -j LOG \
                          --log-prefix "Firewalled packet: Netbus "
$IPTABLES -A INPUT -p tcp --dport 12345 -j DROP
$IPTABLES -A INPUT -p tcp --dport 12346 -m limit -j LOG \
                          --log-prefix "Firewalled packet: Netbus "
$IPTABLES -A INPUT -p tcp --dport 12346 -j DROP
$IPTABLES -A INPUT -p tcp --dport 20034 -m limit -j LOG \
                          --log-prefix "Firewalled packet: Netbus "
$IPTABLES -A INPUT -p tcp --dport 20034 -j DROP
$IPTABLES -A INPUT -p tcp --dport 31337 -m limit -j LOG \
                          --log-prefix "Firewalled packet: BO "
$IPTABLES -A INPUT -p tcp --dport 31337 -j DROP
$IPTABLES -A INPUT -p tcp --dport 6000  -m limit -j LOG \
                          --log-prefix "Firewalled packet: XWin "
$IPTABLES -A INPUT -p tcp --dport 6000  -j DROP

$IPTABLES -A INPUT -p udp --dport 1:10000 -j DROP

#No guardar ident porque es muy usado por servidores irc
$IPTABLES -A INPUT -p tcp --dport 113 -j REJECT

#No guardar IGMP, muchas personas reciben enorme cantidad de este
$IPTABLES -A INPUT -p igmp -j REJECT

#No guardar http ni https pues navegar por largos periodos causan muchas
#conexiones caidas y llenan nuestros logs.
$IPTABLES -A INPUT -p tcp --dport 80 -j REJECT
$IPTABLES -A INPUT -p tcp --dport 443 -j REJECT

##Reglas de captura generales
#iptables viene a estas si no han matcheado en ninguna de las previas
#No tiene sentido guardarlas, no es más que ruido
$IPTABLES -A INPUT -p tcp --syn -m limit --limit 5/minute -j LOG  \
       --log-prefix "Firewalled packet:"
$IPTABLES -A FORWARD -p tcp --syn -m limit --limit 5/minute -j LOG \
       --log-prefix "Firewalled packet:"
#Reject
$IPTABLES -A INPUT -p tcp -j REJECT --reject-with tcp-reset
$IPTABLES -A INPUT -p all -j DROP

$IPTABLES -A FORWARD -p tcp -j REJECT --reject-with tcp-reset
$IPTABLES -A FORWARD -p all -j DROP

#Accept it anyway if it's only output
$IPTABLES -A OUTPUT -j ACCEPT

#Aqui dos ejemplos de como hacer redireccionamiento con squid, escoger uno de
ellos. Primero leer el mini howto sobre squid y redireccionamiento

#al box remoto - Esta alternativa es poco usada, requiere de dos maquinas, una el
linux que redireccionaremos y otra el box remoto hacia donde se enviaran los
paquetes
#iptables -t nat -A PREROUTING -i $INTERNALIF -s ! 216.219.18.5 -p tcp --dport 80
-j DNAT --to 216.219.18.5:3128
```

```
#iptables -t nat -A POSTROUTING -o $INTERNALIF -s 10.10.0.0/24 -d 216.219.18.5 -j
MASQUERADE
#iptables -A FORWARD -s 10.10.0.0/24 -d 216.219.18.5 -i eth0 -o eth0 -p tcp --
dport 3128 -j ACCEPT

#Al box local

$IPTABLES -t nat -A PREROUTING -i $INTERNALIF -p tcp --dport 80 -j REDIRECT --to-
port 8080

#Enmascarar conexiones internar yendo hacia fuera.
#Comentar si no se desea proxy transparente
#Recordar que si se permiten conexiones desde el interior, los clientes
#no usarian el proxy haciendonos perder ancho de banda por no cache

$IPTABLES -A POSTROUTING -t nat -o $EXTERNALIF -j MASQUERADE

exit 0
```

ANEXO 12

Configuración Proxy de Servidor CAYAMBE

```
#      WELCOME TO SQUID 2
#      ------------------
#
#      This is the default Squid configuration file. You may wish
#      to look at the Squid home page (http://www.squid-cache.org/)
#      for the FAQ and other documentation.
#
#      The default Squid config file shows what the defaults for
#      various options happen to be.  If you don't need to change the
#      default, you shouldn't uncomment the line.  Doing so may cause
#      run-time problems.  In some cases "none" refers to no default
#      setting at all, while in other cases it refers to a valid
#      option - the comments for that keyword indicate if this is the
#      case.
#
# This line was added to make rpm -U Do The Right Thing (interim fix)


# NETWORK OPTIONS
# ------------------------------------------------------------------------------

#  TAG: http_port
#      Usage: port
#             hostname:port
#             1.2.3.4:port
#
#      The socket addresses where Squid will listen for HTTP client
#      requests.  You may specify multiple socket addresses.
#      There are three forms: port alone, hostname with port, and
#      IP address with port.  If you specify a hostname or IP
#      address, Squid binds the socket to that specific
#      address.  This replaces the old 'tcp_incoming_address'
#      option.  Most likely, you do not need to bind to a specific
#      address, so you can use the port number alone.
#
#      The default port number is 3128.
#
#      If you are running Squid in accelerator mode, you
#      probably want to listen on port 80 also, or instead.
#
#      The -a command line option will override the *first* port
#      number listed here.   That option will NOT override an IP
#      address, however.
#
#      You may specify multiple socket addresses on multiple lines.
#
#      If you run Squid on a dual-homed machine with an internal
#      and an external interface we recommend you to specify the
#      internal address:port in http_port. This way Squid will only be
#      visible on the internal address.
#
#Default:
# http_port 3128
http_port 192.168.2.253:3128
http_port 192.168.2.253:8080
#  TAG: https_port
#        Usage:  [ip:]port cert=certificate.pem [key=key.pem] [options...]
#
#        The socket address where Squid will listen for HTTPS client
#        requests.
#
#        This is really only useful for situations where you are running
#        squid in accelerator mode and you want to do the SSL work at the
```

```
#        accelerator level.
#
#      You may specify multiple socket addresses on multiple lines,
#      each with their own SSL certificate and/or options.
#
#      Options:
#
#         cert=     Path to SSL certificate (PEM format)
#
#         key=             Path to SSL private key file (PEM format)
#                   if not specified, the certificate file is
#                   assumed to be a combined certificate and
#                   key file
#
#         version=  The version of SSL/TLS supported
#                       1  automatic (default)
#                       2  SSLv2 only
#                       3  SSLv3 only
#                       4  TLSv1 only
#
#         cipher=   Colon separated list of supported ciphers
#
#         options=  Varions SSL engine options. The most important
#                   being:
#                       NO_SSLv2  Disallow the use of SSLv2
#                       NO_SSLv3  Disallow the use of SSLv3
#                       NO_TLSv1  Disallow the use of TLSv1
#                   See src/ssl_support.c or OpenSSL documentation
#                   for a more complete list.
#
#Default:
# none

#  TAG: ssl_unclean_shutdown
#      Some browsers (especially MSIE) bugs out on SSL shutdown
#      messages.
#
#Default:
# ssl_unclean_shutdown off

#  TAG: icp_port
#      The port number where Squid sends and receives ICP queries to
#      and from neighbor caches.  Default is 3130.  To disable use
#      "0".  May be overridden with -u on the command line.
#
#Default:
# icp_port 3130

#  TAG: htcp_port
# Note: This option is only available if Squid is rebuilt with the
#       --enable-htcp option
#
#      The port number where Squid sends and receives HTCP queries to
#      and from neighbor caches.  Default is 4827.  To disable use
#      "0".
#
#Default:
# htcp_port 4827

#  TAG: mcast_groups
#      This tag specifies a list of multicast groups which your server
#      should join to receive multicasted ICP queries.
#
#      NOTE!  Be very careful what you put here!  Be sure you
#      understand the difference between an ICP _query_ and an ICP
#      _reply_.  This option is to be set only if you want to RECEIVE
#      multicast queries.  Do NOT set this option to SEND multicast
#      ICP (use cache_peer for that).  ICP replies are always sent via
```

```
#      unicast, so this option does not affect whether or not you will
#      receive replies from multicast group members.
#
#      You must be very careful to NOT use a multicast address which
#      is already in use by another group of caches.
#
#      If you are unsure about multicast, please read the Multicast
#      chapter in the Squid FAQ (http://www.squid-cache.org/FAQ/).
#
#      Usage: mcast_groups 239.128.16.128 224.0.1.20
#
#      By default, Squid doesn't listen on any multicast groups.
#
#Default:
# none

#  TAG: udp_incoming_address
#  TAG: udp_outgoing_address
#      udp_incoming_address      is used for the ICP socket receiving packets
#                          from other caches.
#      udp_outgoing_address      is used for ICP packets sent out to other
#                          caches.
#
#      The default behavior is to not bind to any specific address.
#
#      A udp_incoming_address value of 0.0.0.0 indicates Squid
#      should listen for UDP messages on all available interfaces.
#
#      If udp_outgoing_address is set to 255.255.255.255 (the default)
#      it will use the same socket as udp_incoming_address. Only
#      change this if you want to have ICP queries sent using another
#      address than where this Squid listens for ICP queries from other
#      caches.
#
#      NOTE, udp_incoming_address and udp_outgoing_address can not
#      have the same value since they both use port 3130.
#
#Default:
# udp_incoming_address 0.0.0.0
# udp_outgoing_address 255.255.255.255


# OPTIONS WHICH AFFECT THE NEIGHBOR SELECTION ALGORITHM
# -----------------------------------------------------------------------------

#  TAG: cache_peer
#      To specify other caches in a hierarchy, use the format:
#
#            cache_peer hostname type http_port icp_port
#
#      For example,
#
#      #                                        proxy  icp
#      #          hostname             type     port   port  options
#      #          -------------------- -------- -----  -----  -----------
#      cache_peer parent.foo.net       parent    3128  3130  [proxy-only]
#      cache_peer sib1.foo.net         sibling   3128  3130  [proxy-only]
#      cache_peer sib2.foo.net         sibling   3128  3130  [proxy-only]
#
#           type:  either 'parent', 'sibling', or 'multicast'.
#
#      proxy_port:  The port number where the cache listens for proxy
#                   requests.
#
#       icp_port:  Used for querying neighbor caches about
#                  objects.  To have a non-ICP neighbor
#                  specify '7' for the ICP port and make sure the
#                  neighbor machine has the UDP echo port
```

```
#                 enabled in its /etc/inetd.conf file.
#
#         options: proxy-only
#                 weight=n
#                 ttl=n
#                 no-query
#                 default
#                 round-robin
#                 multicast-responder
#                 closest-only
#                 no-digest
#                 no-netdb-exchange
#                 no-delay
#                 login=user:password | PASS | *:password
#                 connect-timeout=nn
#                 digest-url=url
#                 allow-miss
#                 max-conn
#                 htcp
#                 carp-load-factor
#
#                 use 'proxy-only' to specify objects fetched
#                 from this cache should not be saved locally.
#
#                 use 'weight=n' to specify a weighted parent.
#                 The weight must be an integer.  The default weight
#                 is 1, larger weights are favored more.
#
#                 use 'ttl=n' to specify a IP multicast TTL to use
#                 when sending an ICP queries to this address.
#                 Only useful when sending to a multicast group.
#                 Because we don't accept ICP replies from random
#                 hosts, you must configure other group members as
#                 peers with the 'multicast-responder' option below.
#
#                 use 'no-query' to NOT send ICP queries to this
#                 neighbor.
#
#                 use 'default' if this is a parent cache which can
#                 be used as a "last-resort." You should probably
#                 only use 'default' in situations where you cannot
#                 use ICP with your parent cache(s).
#
#                 use 'round-robin' to define a set of parents which
#                 should be used in a round-robin fashion in the
#                 absence of any ICP queries.
#
#                 'multicast-responder' indicates the named peer
#                 is a member of a multicast group.  ICP queries will
#                 not be sent directly to the peer, but ICP replies
#                 will be accepted from it.
#
#                 'closest-only' indicates that, for ICP_OP_MISS
#                 replies, we'll only forward CLOSEST_PARENT_MISSes
#                 and never FIRST_PARENT_MISSes.
#
#                 use 'no-digest' to NOT request cache digests from
#                 this neighbor.
#
#                 'no-netdb-exchange' disables requesting ICMP
#                 RTT database (NetDB) from the neighbor.
#
#                 use 'no-delay' to prevent access to this neighbor
#                 from influencing the delay pools.
#
#                 use 'login=user:password' if this is a personal/workgroup
#                 proxy and your parent requires proxy authentication.
#                 Note: The string can include URL escapes (i.e. %20 for
```

```
#                spaces). This also means % must be written as %%.
#
#                use 'login=PASS' if users must authenticate against
#                the upstream proxy. This will pass the users credentials
#                as they are to the peer proxy. This only works for the
#                Basic HTTP authentication sheme. Note: To combine this
#                with proxy_auth both proxies must share the same user
#                database as HTTP only allows for one proxy login.
#                Also be warned this will expose your users proxy
#                password to the peer. USE WITH CAUTION
#
#                use 'login=*:password' to pass the username to the
#                upstream cache, but with a fixed password. This is meant
#                to be used when the peer is in another administrative
#                domain, but it is still needed to identify each user.
#                The star can optionally be followed by some extra
#                information which is added to the username. This can
#                be used to identify this proxy to the peer, similar to
#                the login=username:password option above.
#
#                use 'connect-timeout=nn' to specify a peer
#                specific connect timeout (also see the
#                peer_connect_timeout directive)
#
#                use 'digest-url=url' to tell Squid to fetch the cache
#                digest (if digests are enabled) for this host from
#                the specified URL rather than the Squid default
#                location.
#
#                use 'allow-miss' to disable Squid's use of only-if-cached
#                when forwarding requests to siblings. This is primarily
#                useful when icp_hit_stale is used by the sibling. To
#                extensive use of this option may result in forwarding
#                loops, and you should avoid having two-way peerings
#                with this option. (for example to deny peer usage on
#                requests from peer by denying cache_peer_access if the
#                source is a peer)
#
#                use 'max-conn' to limit the amount of connections Squid
#                may open to this peer.
#
#                use 'htcp' to send HTCP, instead of ICP, queries
#                to the neighbor.  You probably also want to
#                set the "icp port" to 4827 instead of 3130.
#
#                use 'carp-load-factor=f' to define a parent
#                cache as one participating in a CARP array.
#                The 'f' values for all CARP parents must add
#                up to 1.0.
#
#
#      NOTE: non-ICP/HTCP neighbors must be specified as 'parent'.
#
#Default:
# none

#  TAG: cache_peer_domain
#      Use to limit the domains for which a neighbor cache will be
#      queried.  Usage:
#
#      cache_peer_domain cache-host domain [domain ...]
#      cache_peer_domain cache-host !domain
#
#      For example, specifying
#
#             cache_peer_domain parent.foo.net .edu
#
#      has the effect such that UDP query packets are sent to
```

```
#      'bigserver' only when the requested object exists on a
#      server in the .edu domain.  Prefixing the domainname
#      with '!' means the cache will be queried for objects
#      NOT in that domain.
#
#      NOTE:  * Any number of domains may be given for a cache-host,
#               either on the same or separate lines.
#             * When multiple domains are given for a particular
#               cache-host, the first matched domain is applied.
#             * Cache hosts with no domain restrictions are queried
#               for all requests.
#             * There are no defaults.
#             * There is also a 'cache_peer_access' tag in the ACL
#               section.
#
#Default:
# none

#  TAG: neighbor_type_domain
#      usage: neighbor_type_domain neighbor parent|sibling domain domain ...
#
#      Modifying the neighbor type for specific domains is now
#      possible.  You can treat some domains differently than the the
#      default neighbor type specified on the 'cache_peer' line.
#      Normally it should only be necessary to list domains which
#      should be treated differently because the default neighbor type
#      applies for hostnames which do not match domains listed here.
#
#EXAMPLE:
#      cache_peer  parent cache.foo.org 3128 3130
#      neighbor_type_domain cache.foo.org sibling .com .net
#      neighbor_type_domain cache.foo.org sibling .au .de
#
#Default:
# none

#  TAG: icp_query_timeout (msec)
#      Normally Squid will automatically determine an optimal ICP
#      query timeout value based on the round-trip-time of recent ICP
#      queries.  If you want to override the value determined by
#      Squid, set this 'icp_query_timeout' to a non-zero value.  This
#      value is specified in MILLISECONDS, so, to use a 2-second
#      timeout (the old default), you would write:
#
#            icp_query_timeout 2000
#
#Default:
# icp_query_timeout 0

#  TAG: maximum_icp_query_timeout      (msec)
#      Normally the ICP query timeout is determined dynamically.  But
#      sometimes it can lead to very large values (say 5 seconds).
#      Use this option to put an upper limit on the dynamic timeout
#      value.  Do NOT use this option to always use a fixed (instead
#      of a dynamic) timeout value. To set a fixed timeout see the
#      'icp_query_timeout' directive.
#
#Default:
# maximum_icp_query_timeout 2000

#  TAG: mcast_icp_query_timeout  (msec)
#      For Multicast peers, Squid regularly sends out ICP "probes" to
#      count how many other peers are listening on the given multicast
#      address.  This value specifies how long Squid should wait to
#      count all the replies.  The default is 2000 msec, or 2
#      seconds.
#
#Default:
```

```
# mcast_icp_query_timeout 2000

#  TAG: dead_peer_timeout (seconds)
#      This controls how long Squid waits to declare a peer cache
#      as "dead."  If there are no ICP replies received in this
#      amount of time, Squid will declare the peer dead and not
#      expect to receive any further ICP replies.  However, it
#      continues to send ICP queries, and will mark the peer as
#      alive upon receipt of the first subsequent ICP reply.
#
#      This timeout also affects when Squid expects to receive ICP
#      replies from peers.  If more than 'dead_peer' seconds have
#      passed since the last ICP reply was received, Squid will not
#      expect to receive an ICP reply on the next query.  Thus, if
#      your time between requests is greater than this timeout, you
#      will see a lot of requests sent DIRECT to origin servers
#      instead of to your parents.
#
#Default:
# dead_peer_timeout 10 seconds

#  TAG: hierarchy_stoplist
#      A list of words which, if found in a URL, cause the object to
#      be handled directly by this cache.  In other words, use this
#      to not query neighbor caches for certain objects.  You may
#      list this option multiple times.
#We recommend you to use at least the following line.
hierarchy_stoplist cgi-bin ?

#  TAG: no_cache
#      A list of ACL elements which, if matched, cause the request to
#      not be satisfied from the cache and the reply to not be cached.
#      In other words, use this to force certain objects to never be cached.
#
#      You must use the word 'DENY' to indicate the ACL names which should
#      NOT be cached.
#
#We recommend you to use the following two lines.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY


# OPTIONS WHICH AFFECT THE CACHE SIZE
# -----------------------------------------------------------------------------

#  TAG: cache_mem   (bytes)
#      NOTE: THIS PARAMETER DOES NOT SPECIFY THE MAXIMUM PROCESS SIZE.
#      IT ONLY PLACES A LIMIT ON HOW MUCH ADDITIONAL MEMORY SQUID WILL
#      USE AS A MEMORY CACHE OF OBJECTS. SQUID USES MEMORY FOR OTHER
#      THINGS AS WELL. SEE THE SQUID FAQ SECTION 8 FOR DETAILS.
#
#      'cache_mem' specifies the ideal amount of memory to be used
#      for:
#            * In-Transit objects
#            * Hot Objects
#            * Negative-Cached objects
#
#      Data for these objects are stored in 4 KB blocks.  This
#      parameter specifies the ideal upper limit on the total size of
#      4 KB blocks allocated.  In-Transit objects take the highest
#      priority.
#
#      In-transit objects have priority over the others.  When
#      additional space is needed for incoming data, negative-cached
#      and hot objects will be released.  In other words, the
#      negative-cached and hot objects will fill up any unused space
#      not needed for in-transit objects.
#
```

```
#      If circumstances require, this limit will be exceeded.
#      Specifically, if your incoming request rate requires more than
#      'cache_mem' of memory to hold in-transit objects, Squid will
#      exceed this limit to satisfy the new requests.  When the load
#      decreases, blocks will be freed until the high-water mark is
#      reached.  Thereafter, blocks will be used to store hot
#      objects.
#
#Default:
# cache_mem 8 MB
cache_mem 16 MB
#  TAG: cache_swap_low    (percent, 0-100)
#  TAG: cache_swap_high   (percent, 0-100)
#
#      The low- and high-water marks for cache object replacement.
#      Replacement begins when the swap (disk) usage is above the
#      low-water mark and attempts to maintain utilization near the
#      low-water mark.  As swap utilization gets close to high-water
#      mark object eviction becomes more aggressive.  If utilization is
#      close to the low-water mark less replacement is done each time.
#
#      Defaults are 90% and 95%. If you have a large cache, 5% could be
#      hundreds of MB. If this is the case you may wish to set these
#      numbers closer together.
#
#Default:
# cache_swap_low 90
# cache_swap_high 95

#  TAG: maximum_object_size      (bytes)
#      Objects larger than this size will NOT be saved on disk.  The
#      value is specified in kilobytes, and the default is 4MB.  If
#      you wish to get a high BYTES hit ratio, you should probably
#      increase this (one 32 MB object hit counts for 3200 10KB
#      hits).  If you wish to increase speed more than your want to
#      save bandwidth you should leave this low.
#
#      NOTE: if using the LFUDA replacement policy you should increase
#      this value to maximize the byte hit rate improvement of LFUDA!
#      See replacement_policy below for a discussion of this policy.
#
#Default:
# maximum_object_size 4096 KB

#  TAG: minimum_object_size      (bytes)
#      Objects smaller than this size will NOT be saved on disk.  The
#      value is specified in kilobytes, and the default is 0 KB, which
#      means there is no minimum.
#
#Default:
# minimum_object_size 0 KB

#  TAG: maximum_object_size_in_memory  (bytes)
#        Objects greater than this size will not be attempted to kept in
#        the memory cache. This should be set high enough to keep objects
#        accessed frequently in memory to improve performance whilst low
#        enough to keep larger objects from hoarding cache_mem .
#
#Default:
# maximum_object_size_in_memory 8 KB

#  TAG: ipcache_size      (number of entries)
#  TAG: ipcache_low (percent)
#  TAG: ipcache_high      (percent)
#      The size, low-, and high-water marks for the IP cache.
#
#Default:
# ipcache_size 1024
```

```
# ipcache_low 90
# ipcache_high 95

#  TAG: fqdncache_size    (number of entries)
#      Maximum number of FQDN cache entries.
#
#Default:
# fqdncache_size 1024

#  TAG: cache_replacement_policy
#      The cache replacement policy parameter determines which
#      objects are evicted (replaced) when disk space is needed.
#
#          lru       : Squid's original list based LRU policy
#          heap GDSF : Greedy-Dual Size Frequency
#          heap LFUDA: Least Frequently Used with Dynamic Aging
#          heap LRU  : LRU policy implemented using a heap
#
#      Applies to any cache_dir lines listed below this.
#
#      The LRU policies keeps recently referenced objects.
#
#      The heap GDSF policy optimizes object hit rate by keeping smaller
#      popular objects in cache so it has a better chance of getting a
#      hit.  It achieves a lower byte hit rate than LFUDA though since
#      it evicts larger (possibly popular) objects.
#
#      The heap LFUDA policy keeps popular objects in cache regardless of
#      their size and thus optimizes byte hit rate at the expense of
#      hit rate since one large, popular object will prevent many
#      smaller, slightly less popular objects from being cached.
#
#      Both policies utilize a dynamic aging mechanism that prevents
#      cache pollution that can otherwise occur with frequency-based
#      replacement policies.
#
#      NOTE: if using the LFUDA replacement policy you should increase
#      the value of maximum_object_size above its default of 4096 KB to
#      to maximize the potential byte hit rate improvement of LFUDA.
#
#      For more information about the GDSF and LFUDA cache replacement
#      policies see http://www.hpl.hp.com/techreports/1999/HPL-1999-69.html
#      and http://fog.hpl.external.hp.com/techreports/98/HPL-98-173.html.
#
#Default:
# cache_replacement_policy lru

#  TAG: memory_replacement_policy
#      The memory replacement policy parameter determines which
#      objects are purged from memory when memory space is needed.
#
#      See cache_replacement_policy for details.
#
#Default:
# memory_replacement_policy lru


# LOGFILE PATHNAMES AND CACHE DIRECTORIES
# -----------------------------------------------------------------------------

#  TAG: cache_dir
#      Usage:
#
#      cache_dir Type Directory-Name Fs-specific-data [options]
#
#      You can specify multiple cache_dir lines to spread the
#      cache among different disk partitions.
#
```

```
#      Type specifies the kind of storage system to use. Only "ufs"
#      is built by default. To eanble any of the other storage systems
#      see the --enable-storeio configure option.
#
#      'Directory' is a top-level directory where cache swap
#      files will be stored.  If you want to use an entire disk
#      for caching, this can be the mount-point directory.
#      The directory must exist and be writable by the Squid
#      process.  Squid will NOT create this directory for you.
#
#      The ufs store type:
#
#      "ufs" is the old well-known Squid storage format that has always
#      been there.
#
#      cache_dir ufs Directory-Name Mbytes L1 L2 [options]
#
#      'Mbytes' is the amount of disk space (MB) to use under this
#      directory.  The default is 100 MB.  Change this to suit your
#      configuration.  Do NOT put the size of your disk drive here.
#      Instead, if you want Squid to use the entire disk drive,
#      subtract 20% and use that value.
#
#      'Level-1' is the number of first-level subdirectories which
#      will be created under the 'Directory'.  The default is 16.
#
#      'Level-2' is the number of second-level subdirectories which
#      will be created under each first-level directory.  The default
#      is 256.
#
#      The aufs store type:
#
#      "aufs" uses the same storage format as "ufs", utilizing
#      POSIX-threads to avoid blocking the main Squid process on
#      disk-I/O. This was formerly known in Squid as async-io.
#
#      cache_dir aufs Directory-Name Mbytes L1 L2 [options]
#
#      see argument descriptions under ufs above
#
#      The diskd store type:
#
#      "diskd" uses the same storage format as "ufs", utilizing a
#      separate process to avoid blocking the main Squid process on
#      disk-I/O.
#
#      cache_dir diskd Directory-Name Mbytes L1 L2 [options] [Q1=n] [Q2=n]
#
#      see argument descriptions under ufs above
#
#      Q1 specifies the number of unacknowledged I/O requests when Squid
#      stops opening new files. If this many messages are in the queues,
#      Squid won't open new files. Default is 64
#
#      Q2 specifies the number of unacknowledged messages when Squid
#      starts blocking.  If this many messages are in the queues,
#      Squid blocks until it receives some replies. Default is 72
#
#      When Q1 < Q2 (the default), the cache directory is optimized
#      for lower response time at the expense of a decrease in hit
#      ratio.  If Q1 > Q2, the cache directory is optimized for
#      higher hit ratio at the expense of an increase in response
#      time.
#
#      The coss store type:
#
#      block-size=n defines the "block size" for COSS cache_dir's.
#      Squid uses file numbers as block numbers.  Since file numbers
```

```
#      are limited to 24 bits, the block size determines the maximum
#      size of the COSS partition.  The default is 512 bytes, which
#      leads to a maximum cache_dir size of 512<<24, or 8 GB.  Note
#      you should not change the coss block size after Squid
#      has written some objects to the cache_dir.
#
#      Common options:
#
#      read-only, this cache_dir is read only.
#
#      max-size=n, refers to the max object size this storedir supports.
#      It is used to initially choose the storedir to dump the object.
#      Note: To make optimal use of the max-size limits you should order
#      the cache_dir lines with the smallest max-size value first and the
#      ones with no max-size specification last.
#
#      Note that for coss, max-size must be less than COSS_MEMBUF_SZ
#      (hard coded at 1 MB).
#
#Default:
# cache_dir ufs /var/spool/squid 100 16 256
cache_dir ufs /var/spool/squid 700 16 256

#  TAG: cache_access_log
#      Logs the client request activity.  Contains an entry for
#      every HTTP and ICP queries received. To disable, enter "none".
#
#Default:
# cache_access_log /var/log/squid/access.log

#  TAG: cache_log
#      Cache logging file. This is where general information about
#      your cache's behavior goes. You can increase the amount of data
#      logged to this file with the "debug_options" tag below.
#
#Default:
# cache_log /var/log/squid/cache.log

#  TAG: cache_store_log
#      Logs the activities of the storage manager.  Shows which
#      objects are ejected from the cache, and which objects are
#      saved and for how long.  To disable, enter "none". There are
#      not really utilities to analyze this data, so you can safely
#      disable it.
#
#Default:
# cache_store_log /var/log/squid/store.log

#  TAG: cache_swap_log
#      Location for the cache "swap.state" file. This log file holds
#      the metadata of objects saved on disk.  It is used to rebuild
#      the cache during startup.  Normally this file resides in each
#      'cache_dir' directory, but you may specify an alternate
#      pathname here.  Note you must give a full filename, not just
#      a directory. Since this is the index for the whole object
#      list you CANNOT periodically rotate it!
#
#      If %s can be used in the file name it will be replaced with a
#      a representation of the cache_dir name where each / is replaced
#      with '.'. This is needed to allow adding/removing cache_dir
#      lines when cache_swap_log is being used.
#
#      If have more than one 'cache_dir', and %s is not used in the name
#      these swap logs will have names such as:
#
#            cache_swap_log.00
#            cache_swap_log.01
#            cache_swap_log.02
```

```
#
#      The numbered extension (which is added automatically)
#      corresponds to the order of the 'cache_dir' lines in this
#      configuration file.  If you change the order of the 'cache_dir'
#      lines in this file, these log files will NOT correspond to
#      the correct 'cache_dir' entry (unless you manually rename
#      them).  We recommend you do NOT use this option.  It is
#      better to keep these log files in each 'cache_dir' directory.
#
#Default:
# none

#  TAG: emulate_httpd_log on|off
#      The Cache can emulate the log file format which many 'httpd'
#      programs use.  To disable/enable this emulation, set
#      emulate_httpd_log to 'off' or 'on'.  The default
#      is to use the native log format since it includes useful
#      information Squid-specific log analyzers use.
#
#Default:
# emulate_httpd_log off

#  TAG: log_ip_on_direct  on|off
#      Log the destination IP address in the hierarchy log tag when going
#      direct. Earlier Squid versions logged the hostname here. If you
#      prefer the old way set this to off.
#
#Default:
# log_ip_on_direct on

#  TAG: mime_table
#      Pathname to Squid's MIME table. You shouldn't need to change
#      this, but the default file contains examples and formatting
#      information if you do.
#
#Default:
# mime_table /etc/squid/mime.conf

#  TAG: log_mime_hdrs     on|off
#      The Cache can record both the request and the response MIME
#      headers for each HTTP transaction.  The headers are encoded
#      safely and will appear as two bracketed fields at the end of
#      the access log (for either the native or httpd-emulated log
#      formats).  To enable this logging set log_mime_hdrs to 'on'.
#
#Default:
# log_mime_hdrs off

#  TAG: useragent_log
#      Squid will write the User-Agent field from HTTP requests
#      to the filename specified here.  By default useragent_log
#      is disabled.
#
#Default:
# none

#  TAG: referer_log
#      Squid will write the Referer field from HTTP requests to the
#      filename specified here.  By default referer_log is disabled.
#
#Default:
# none

#  TAG: pid_filename
#      A filename to write the process-id to.  To disable, enter "none".
#
#Default:
# pid_filename /var/run/squid.pid
```

```
#  TAG: debug_options
#      Logging options are set as section,level where each source file
#      is assigned a unique section.  Lower levels result in less
#      output,  Full debugging (level 9) can result in a very large
#      log file, so be careful.  The magic word "ALL" sets debugging
#      levels for all sections.  We recommend normally running with
#      "ALL,1".
#
#Default:
# debug_options ALL,1

#  TAG: log_fqdn    on|off
#      Turn this on if you wish to log fully qualified domain names
#      in the access.log. To do this Squid does a DNS lookup of all
#      IP's connecting to it. This can (in some situations) increase
#      latency, which makes your cache seem slower for interactive
#      browsing.
#
#Default:
# log_fqdn off

#  TAG: client_netmask
#      A netmask for client addresses in logfiles and cachemgr output.
#      Change this to protect the privacy of your cache clients.
#      A netmask of 255.255.255.0 will log all IP's in that range with
#      the last digit set to '0'.
#
#Default:
# client_netmask 255.255.255.255


# OPTIONS FOR EXTERNAL SUPPORT PROGRAMS
# ------------------------------------------------------------------------------

#  TAG: ftp_user
#      If you want the anonymous login password to be more informative
#      (and enable the use of picky ftp servers), set this to something
#      reasonable for your domain, like wwwuser@somewhere.net
#
#      The reason why this is domainless by default is the
#      request can be made on the behalf of a user in any domain,
#      depending on how the cache is used.
#      Some ftp server also validate the email address is valid
#      (for example perl.com).
#
#Default:
# ftp_user Squid@

#  TAG: ftp_list_width
#      Sets the width of ftp listings. This should be set to fit in
#      the width of a standard browser. Setting this too small
#      can cut off long filenames when browsing ftp sites.
#
#Default:
# ftp_list_width 32

#  TAG: ftp_passive
#      If your firewall does not allow Squid to use passive
#      connections, turn off this option.
#
#Default:
# ftp_passive on

#  TAG: ftp_sanitycheck
#      For security and data integrity reasons Squid by default performs
#      sanity checks of the addresses of FTP data connections ensure the
#      data connection is to the requested server. If you need to allow
```

```
#      FTP connections to servers using another IP address for the data
#      connection turn this off.
#
#Default:
# ftp_sanitycheck on

#  TAG: ftp_telnet_protocol
#      The FTP protocol is officially defined to use the telnet protocol
#      as transport channel for the control connection. However, many
#      implemenations are broken and does not respect this aspect of
#      the FTP protocol.
#
#      If you have trouble accessing files with ASCII code 255 in the
#      path or similar problems involving this ASCII code you can
#      try setting this directive to off. If that helps, report to the
#      operator of the FTP server in question that their FTP server
#      is broken and does not follow the FTP standard.
#
#Default:
# ftp_telnet_protocol on

#  TAG: cache_dns_program
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
#      Specify the location of the executable for dnslookup process.
#
#Default:
# cache_dns_program /usr/lib/squid/dnsserver

#  TAG: dns_children
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
#      The number of processes spawn to service DNS name lookups.
#      For heavily loaded caches on large servers, you should
#      probably increase this value to at least 10.  The maximum
#      is 32.  The default is 5.
#
#      You must have at least one dnsserver process.
#
#Default:
# dns_children 5

#  TAG: dns_retransmit_interval
#      Initial retransmit interval for DNS queries. The interval is
#      doubled each time all configured DNS servers have been tried.
#
#
#Default:
# dns_retransmit_interval 5 seconds

#  TAG: dns_timeout
#      DNS Query timeout. If no response is received to a DNS query
#      within this time all DNS servers for the queried domain
#      are assumed to be unavailable.
#
#Default:
# dns_timeout 2 minutes

#  TAG: dns_defnames      on|off
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
#      Normally the 'dnsserver' disables the RES_DEFNAMES resolver
#      option (see res_init(3)).  This prevents caches in a hierarchy
#      from interpreting single-component hostnames locally.  To allow
#      dnsserver to handle single-component names, enable this
```

```
#      option.
#
#Default:
# dns_defnames off

#  TAG: dns_nameservers
#      Use this if you want to specify a list of DNS name servers
#      (IP addresses) to use instead of those given in your
#      /etc/resolv.conf file.
#      On Windows platforms, if no value is specified here or in
#      the /etc/resolv.conf file, the list of DNS name servers are
#      taken from the Windows registry, both static and dynamic DHCP
#      configurations are supported.
#
#      Example: dns_nameservers 10.0.0.1 192.172.0.4
#
#Default:
# none

#  TAG: hosts_file
#      Location of the host-local IP name-address associations
#      database.  Most Operating Systems have such a file: under
#      Un*X it's by default in /etc/hosts.  MS-Windows NT/2000 places
#      it in %SystemRoot%(by default
#      c:\winnt)\system32\drivers\etc\hosts, while Windows 9x/ME
#      places it in %windir%(usually c:\windows)\hosts
#
#      The file contains newline-separated definitions, in the
#      form ip_address_in_dotted_form name [name ...] names are
#      whitespace-separated.  lines beginnng with an hash (#)
#      character are comments.
#
#      The file is checked at startup and upon configuration.  If
#      set to 'none', it won't be checked.  If append_domain is
#      used, that domain will be added to domain-local (i.e. not
#      containing any dot character) host definitions.
#
#Default:
# hosts_file /etc/hosts

#  TAG: diskd_program
#      Specify the location of the diskd executable.
#      Note that this is only useful if you have compiled in
#      diskd as one of the store io modules.
#
#Default:
# diskd_program /usr/lib/squid/diskd

#  TAG: unlinkd_program
#      Specify the location of the executable for file deletion process.
#
#Default:
# unlinkd_program /usr/lib/squid/unlinkd

#  TAG: pinger_program
# Note: This option is only available if Squid is rebuilt with the
#       --enable-icmp option
#
#      Specify the location of the executable for the pinger process.
#
#Default:
# pinger_program /usr/lib/squid/pinger

#  TAG: redirect_program
#      Specify the location of the executable for the URL redirector.
#      Since they can perform almost any function there isn't one included.
#      See the FAQ (section 15) for information on how to write one.
#      By default, a redirector is not used.
```

```
#
#Default:
# none

#  TAG: redirect_children
#      The number of redirector processes to spawn. If you start
#      too few Squid will have to wait for them to process a backlog of
#      URLs, slowing it down. If you start too many they will use RAM
#      and other system resources.
#
#Default:
# redirect_children 5

#  TAG: redirect_rewrites_host_header
#      By default Squid rewrites any Host: header in redirected
#      requests.  If you are running an accelerator this may
#      not be a wanted effect of a redirector.
#
#Default:
# redirect_rewrites_host_header on

#  TAG: redirector_access
#      If defined, this access list specifies which requests are
#      sent to the redirector processes.  By default all requests
#      are sent.
#
#Default:
# none

#  TAG: auth_param
#      This is used to define parameters for the various authentication
#      schemes supported by Squid.
#
#      format: auth_param scheme parameter [setting]
#
#      The order in which authentication schemes are presented to the client is
#      dependant on the order the scheme first appears in config file. IE
#      has a bug (it's not rfc 2617 compliant) in that it will use the basic
#      scheme if basic is the first entry presented, even if more secure
#      schemes are presented. For now use the order in the recommended
#      settings section below. If other browsers have difficulties (don't
#      recognise the schemes offered even if you are using basic) either
#      put basic first, or disable the other schemes (by commenting out their
#      program entry).
#
#      Once an authentication scheme is fully configured, it can only be
#      shutdown by shutting squid down and restarting. Changes can be made on
#      the fly and activated with a reconfigure. I.E. You can change to a
#      different helper, but not unconfigure the helper completely.
#
#      Please note that while this directive defines how Squid processes
#      authentication it does not automatically activate authentication.
#      To use authenticaiton you must in addition make use of acls based
#      on login name in http_access (proxy_auth, proxy_auth_regex or
#      external with %LOGIN used in the format tag). The browser will be
#      challenged for authentication on the first such acl encountered
#      in http_access processing and will also be rechallenged for new
#      login credentials if the request is being denied by a proxy_auth
#      type acl.
#
#      === Parameters for the basic scheme follow. ===
#
#      "program" cmdline
#      Specify the command for the external authenticator.  Such a program
#      reads a line containing "username password" and replies "OK" or
#      "ERR" in an endless loop.
#
#      By default, the basic authentication sheme is not used unless a
```

```
#      program is specified.
#
#      If you want to use the traditional proxy authentication, jump over to
#      the helpers/basic_auth/NCSA directory and type:
#            % make
#            % make install
#
#      Then, set this line to something like
#
#      auth_param basic program /usr/libexec/ncsa_auth /usr/etc/passwd
#
#      "children" numberofchildren
#      The number of authenticator processes to spawn.
#      If you start too few Squid will have to wait for them to process a
#      backlog of usercode/password verifications, slowing it down. When
#      password verifications are done via a (slow) network you are likely to
#      need lots of authenticator processes.
#      auth_param basic children 5
#
#      "realm" realmstring
#      Specifies the realm name which is to be reported to the client for
#      the basic proxy authentication scheme (part of the text the user
#      will see when prompted their username and password).
#      auth_param basic realm Squid proxy-caching web server
#
#      "credentialsttl" timetolive
#      Specifies how long squid assumes an externally validated
#      username:password pair is valid for - in other words how often the
#      helper program is called for that user. Set this low to force
#      revalidation with short lived passwords.  Note that setting this high
#      does not impact your susceptability to replay attacks unless you are
#      using an one-time password system (such as SecureID). If you are using
#      such a system, you will be vulnerable to replay attacks unless you
#      also use the max_user_ip ACL in an http_access rule.
#      auth_param basic credentialsttl 2 hours
#
#      "casesensitive" on|off
#      Specifies if usernames are case sensitive. Most user databases are
#      case insensitive allowing the same username to be spelled using both
#      lower and upper case letters, but some are case sensitive. This
#      makes a big difference for user_max_ip ACL processing and similar.
#      auth_param basic casesensitive off
#
#      === Parameters for the digest scheme follow ===
#
#      "program" cmdline
#      Specify the command for the external authenticator.  Such a program
#      reads a line containing "username":"realm" and replies with the
#      appropriate H(A1) value base64 encoded or ERR if the user (or his H(A1)
#      hash) does not exists.  See rfc 2616 for the definition of H(A1).
#
#      By default, the digest authentication scheme is not used unless a
#      program is specified.
#
#      If you want to use a digest authenticator, jump over to the
#      helpers/digest_auth/ directory and choose the authenticator to use.
#      It it's directory type
#            % make
#              % make install
#
#      Then, set this line to something like
#
#      auth_param digest program /usr/libexec/digest_auth_pw /usr/etc/digpass
#
#
#      "children" numberofchildren
#      The number of authenticator processes to spawn (no default). If you
#      start too few Squid will have to wait for them to process a backlog of
```

```
#      H(A1) calculations, slowing it down.  When the H(A1) calculations are
#      done via a (slow) network you are likely to need lots of authenticator
#      processes.
#      auth_param digest children 5
#
#      "realm" realmstring
#      Specifies the realm name which is to be reported to the client for the
#      digest proxy authentication scheme (part of the text the user will see
#      when prompted their username and password).
#      auth_param digest realm Squid proxy-caching web server
#
#      "nonce_garbage_interval" timeinterval
#      Specifies the interval that nonces that have been issued to clients are
#      checked for validity.
#      auth_param digest nonce_garbage_interval 5 minutes
#
#      "nonce_max_duration" timeinterval
#      Specifies the maximum length of time a given nonce will be valid for.
#      auth_param digest nonce_max_duration 30 minutes
#
#      "nonce_max_count" number
#      Specifies the maximum number of times a given nonce can be used.
#      auth_param digest nonce_max_count 50
#
#      "nonce_strictness" on|off
#      Determines if squid requires strict increment-by-1 behaviour for nonce
#      counts, or just incrementing (off - for use when useragents generate
#      nonce counts that occasionally miss 1 (ie, 1,2,4,6)).
#      auth_param digest nonce_strictness off
#
#      "check_nonce_count" on|off
#      This directive if set to off can disable the nonce count check
#      completely to work around buggy digest qop implementations in certain
#      mainstream browser versions. Default on to check the nonce count to
#      protect from authentication replay attacks.
#      auth_param digest check_nonce_count on
#
#      "post_workaround" on|off
#      This is a workaround to certain buggy browsers who sends an incorrect
#      request digest in POST requests when reusing the same nonce as aquired
#            earlier in response to a GET request.
#      auth_param digest post_workaround off
#
#      === NTLM scheme options follow ===
#
#      "program" cmdline
#      Specify the command for the external ntlm authenticator. Such a
#      program participates in the NTLMSSP exchanges between Squid and the
#      client and reads commands according to the Squid ntlmssp helper
#      protocol. See helpers/ntlm_auth/ for details. Recommended ntlm
#      authenticator is ntlm_auth from Samba-3.X, but a number of other
#      ntlm authenticators is available.
#
#      By default, the ntlm authentication scheme is not used unless a
#      program is specified.
#
#      auth_param    ntlm    program    /path/to/samba/bin/ntlm_auth    --helper-
protocol=squid-2.5-ntlmssp
#
#      "children" numberofchildren
#      The number of authenticator processes to spawn (no default). If you
#      start too few Squid will have to wait for them to process a backlog
#      of credential verifications, slowing it down. When crendential
#      verifications are done via a (slow) network you are likely to need
#      lots of authenticator processes.
#      auth_param ntlm children 5
#
#      "max_challenge_reuses" number
```

```
#      The maximum number of times a challenge given by a ntlm authentication
#      helper can be reused. Increasing this number increases your exposure
#      to replay attacks on your network. 0 (the default) means use the
#      challenge is used only once. See also the max_ntlm_challenge_lifetime
#      directive if enabling challenge reuses.
#      auth_param ntlm max_challenge_reuses 0
#
#      "max_challenge_lifetime" timespan
#      The maximum time period a ntlm challenge is reused over. The
#      actual period will be the minimum of this time AND the number of
#      reused challenges.
#      auth_param ntlm max_challenge_lifetime 2 minutes
#
#      "use_ntlm_negotiate" on|off
#      Enables support for NTLM NEGOTIATE packet exchanges with the helper.
#      The configured ntlm authenticator must be able to handle NTLM
#      NEGOTIATE packet. See the authenticator programs documentation if
#      unsure. ntlm_auth from Samba-3.0.2 or later supports the use of this
#      option.
#      The NEGOTIATE packet is required to support NTLMv2 and a
#      number of other negotiable NTLMSSP options, and also makes it
#      more likely the negotiation is successful. Enabling this parameter
#      will also solve problems encountered when NT domain policies
#      restrict users to access only certain workstations. When this is off,
#      all users must be allowed to log on the proxy servers too, or they'll
#      get "invalid workstation" errors - and access denied - when trying to
#      use Squid's services.
#      Use of ntlm NEGOTIATE is incompatible with challenge reuse, so
#      enabling this parameter will OVERRIDE the max_challenge_reuses and
#      max_challenge_lifetime parameters and set them to 0.
#      auth_param ntlm use_ntlm_negotiate off
#
#Recommended minimum configuration:
#auth_param digest program <uncomment and complete this line>
#auth_param digest children 5
#auth_param digest realm Squid proxy-caching web server
#auth_param digest nonce_garbage_interval 5 minutes
#auth_param digest nonce_max_duration 30 minutes
#auth_param digest nonce_max_count 50
#auth_param ntlm program <uncomment and complete this line to activate>
#auth_param ntlm children 5
#auth_param ntlm max_challenge_reuses 0
#auth_param ntlm max_challenge_lifetime 2 minutes
#auth_param ntlm use_ntlm_negotiate off
#auth_param basic program <uncomment and complete this line>
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
auth_param basic casesensitive off

#  TAG: authenticate_cache_garbage_interval
#      The time period between garbage collection across the username cache.
#      This is a tradeoff between memory utilisation (long intervals - say
#      2 days) and CPU (short intervals - say 1 minute). Only change if you
#      have good reason to.
#
#Default:
# authenticate_cache_garbage_interval 1 hour

#  TAG: authenticate_ttl
#      The time a user & their credentials stay in the logged in user cache
#      since their last request. When the garbage interval passes, all user
#      credentials that have passed their TTL are removed from memory.
#
#Default:
# authenticate_ttl 1 hour

#  TAG: authenticate_ip_ttl
```

```
#      If you use proxy authentication and the 'max_user_ip' ACL, this
#      directive controls how long Squid remembers the IP addresses
#      associated with each user.  Use a small value (e.g., 60 seconds) if
#      your users might change addresses quickly, as is the case with
#      dialups. You might be safe using a larger value (e.g., 2 hours) in a
#      corporate LAN environment with relatively static address assignments.
#
#Default:
# authenticate_ip_ttl 0 seconds

#  TAG: external_acl_type
#      This option defines external acl classes using a helper program to
#      look up the status
#
#        external_acl_type   name   [options]   FORMAT..   /path/to/helper   [helper
arguments..]
#
#      Options:
#
#        ttl=n             TTL in seconds for cached results (defaults to 3600
#                    for 1 hour)
#        negative_ttl=n
#                    TTL for cached negative lookups (default same
#                    as ttl)
#        children=n Concurrency level / number of processes spawn
#                    to service external acl lookups of this type.
#                    Note: see compatibility note below
#        cache=n    result cache size, 0 is unbounded (default)
#        protocol=3.0      Use URL-escaped strings instead of quoting
#
#      FORMAT specifications
#
#        %LOGIN     Authenticated user login name
#        %IDENT     Ident user name
#        %SRC       Client IP
#        %DST       Requested host
#        %PROTO     Requested protocol
#        %PORT             Requested port
#        %METHOD    Request method
#        %{Header}  HTTP request header
#        %{Hdr:member}     HTTP request header list member
#        %{Hdr:;member}
#                    HTTP request header list member using ; as
#                    list separator. ; can be any non-alphanumeric
#                    character.
#
#      In addition, any string specified in the referencing acl will
#      also be included in the helper request line, after the specified
#      formats (see the "acl external" directive)
#
#      The helper receives lines per the above format specification,
#      and returns lines starting with OK or ERR indicating the validity
#      of the request and optionally followed by additional keywords with
#      more details.
#
#      General result syntax:
#
#        OK/ERR keyword=value ...
#
#      Defined keywords:
#
#        user=             The users name (login)
#        error=     Error description (only defined for ERR results)
#
#      Keyword values need to be enclosed in quotes if they may contain
#      whitespace, or the whitespace escaped using \. Any quotes or \
#      characters within the keyword value must be \ escaped.
#
```

```
#      If protocol=3.0 then URL escaping of the strings is used instead
#      of the above described quoting format.
#
#      Compatibility Note: The children= option was named concurrency= in
#      Squid-2.5.STABLE3 and earlier and such syntax is still accepted to
#      keep compatibility within the Squid-2.5 release. However, the meaning
#      of concurrency= option has changed in Squid-3 and the old syntax of
#      the directive is therefore depreated from Squid-2.5.STABLE4 and later.
#      If you want to be able to easily downgrade to earlier Squid-2.5
#      releases you may want to continue using the old name, if not
#      please use the new name.
#
#Default:
# none


# OPTIONS FOR TUNING THE CACHE
# ------------------------------------------------------------------------------

#  TAG: wais_relay_host
#  TAG: wais_relay_port
#      Relay WAIS request to host (1st arg) at port (2 arg).
#
#Default:
# wais_relay_port 0

#  TAG: request_header_max_size  (KB)
#      This specifies the maximum size for HTTP headers in a request.
#      Request headers are usually relatively small (about 512 bytes).
#      Placing a limit on the request header size will catch certain
#      bugs (for example with persistent connections) and possibly
#      buffer-overflow or denial-of-service attacks.
#
#Default:
# request_header_max_size 10 KB

#  TAG: request_body_max_size    (KB)
#      This specifies the maximum size for an HTTP request body.
#      In other words, the maximum size of a PUT/POST request.
#      A user who attempts to send a request with a body larger
#      than this limit receives an "Invalid Request" error message.
#      If you set this parameter to a zero (the default), there will
#      be no limit imposed.
#
#Default:
# request_body_max_size 0 KB

#  TAG: refresh_pattern
#      usage: refresh_pattern [-i] regex min percent max [options]
#
#      By default, regular expressions are CASE-SENSITIVE.  To make
#      them case-insensitive, use the -i option.
#
#      'Min' is the time (in minutes) an object without an explicit
#      expiry time should be considered fresh. The recommended
#      value is 0, any higher values may cause dynamic applications
#      to be erroneously cached unless the application designer
#      has taken the appropriate actions.
#
#      'Percent' is a percentage of the objects age (time since last
#      modification age) an object without explicit expiry time
#      will be considered fresh.
#
#      'Max' is an upper limit on how long objects without an explicit
#      expiry time will be considered fresh.
#
#      options: override-expire
#               override-lastmod
```

```
#             reload-into-ims
#             ignore-reload
#
#            override-expire enforces min age even if the server
#            sent a Expires: header. Doing this VIOLATES the HTTP
#            standard.  Enabling this feature could make you liable
#            for problems which it causes.
#
#            override-lastmod enforces min age even on objects
#            that were modified recently.
#
#            reload-into-ims changes client no-cache or ``reload''
#            to If-Modified-Since requests. Doing this VIOLATES the
#            HTTP standard. Enabling this feature could make you
#            liable for problems which it causes.
#
#            ignore-reload ignores a client no-cache or ``reload''
#            header. Doing this VIOLATES the HTTP standard. Enabling
#            this feature could make you liable for problems which
#            it causes.
#
#      Basically a cached object is:
#
#            FRESH if expires < now, else STALE
#            STALE if age > max
#            FRESH if lm-factor < percent, else STALE
#            FRESH if age < min
#            else STALE
#
#      The refresh_pattern lines are checked in the order listed here.
#      The first entry which matches is used.  If none of the entries
#      match the default will be used.
#
#      Note, you must uncomment all the default lines if you want
#      to change one. The default setting is only active if none is
#      used.
#
#Suggested default:
refresh_pattern ^ftp:           1440   20%    10080
refresh_pattern ^gopher:  1440   0%     1440
refresh_pattern .         0      20%    4320

#  TAG: quick_abort_min   (KB)
#  TAG: quick_abort_max   (KB)
#  TAG: quick_abort_pct   (percent)
#      The cache by default continues downloading aborted requests
#      which are almost completed (less than 16 KB remaining). This
#      may be undesirable on slow (e.g. SLIP) links and/or very busy
#      caches.  Impatient users may tie up file descriptors and
#      bandwidth by repeatedly requesting and immediately aborting
#      downloads.
#
#      When the user aborts a request, Squid will check the
#      quick_abort values to the amount of data transfered until
#      then.
#
#      If the transfer has less than 'quick_abort_min' KB remaining,
#      it will finish the retrieval.
#
#      If the transfer has more than 'quick_abort_max' KB remaining,
#      it will abort the retrieval.
#
#      If more than 'quick_abort_pct' of the transfer has completed,
#      it will finish the retrieval.
#
#      If you do not want any retrieval to continue after the client
#      has aborted, set both 'quick_abort_min' and 'quick_abort_max'
#      to '0 KB'.
```

```
#
#      If you want retrievals to always continue if they are being
#      cached set 'quick_abort_min' to '-1 KB'.
#
#Default:
# quick_abort_min 16 KB
# quick_abort_max 16 KB
# quick_abort_pct 95

#  TAG: negative_ttl      time-units
#      Time-to-Live (TTL) for failed requests.  Certain types of
#      failures (such as "connection refused" and "404 Not Found") are
#      negatively-cached for a configurable amount of time.  The
#      default is 5 minutes.  Note that this is different from
#      negative caching of DNS lookups.
#
#Default:
# negative_ttl 5 minutes

#  TAG: positive_dns_ttl  time-units
#      Upper limit on how long Squid will cache positive DNS responses.
#      Default is 6 hours (360 minutes). This directive must be set
#      larger than negative_dns_ttl.
#
#Default:
# positive_dns_ttl 6 hours

#  TAG: negative_dns_ttl  time-units
#      Time-to-Live (TTL) for negative caching of failed DNS lookups.
#      This also makes sets the lower cache limit on positive lookups.
#      Minimum value is 1 second, and it is not recommendable to go
#      much below 10 seconds.
#
#Default:
# negative_dns_ttl 1 minute

#  TAG: range_offset_limit(bytes)
#      Sets a upper limit on how far into the the file a Range request
#      may be to cause Squid to prefetch the whole file. If beyond this
#      limit Squid forwards the Range request as it is and the result
#      is NOT cached.
#
#      This is to stop a far ahead range request (lets say start at 17MB)
#      from making Squid fetch the whole object up to that point before
#      sending anything to the client.
#
#      A value of -1 causes Squid to always fetch the object from the
#      beginning so it may cache the result. (2.0 style)
#
#      A value of 0 causes Squid to never fetch more than the
#      client requested. (default)
#
#Default:
# range_offset_limit 0 KB


# TIMEOUTS
# -----------------------------------------------------------------------------

#  TAG: forward_timeout   time-units
#      This parameter specifies how long Squid should at most attempt in
#      finding a forwarding path for the request before giving up.
#
#Default:
# forward_timeout 4 minutes

#  TAG: connect_timeout   time-units
#      This parameter specifies how long to wait for the TCP connect to
```

```
#      the requested server or peer to complete before Squid should
#      attempt to find another path where to forward the request.
#
#Default:
# connect_timeout 1 minute

#  TAG: peer_connect_timeout     time-units
#      This parameter specifies how long to wait for a pending TCP
#      connection to a peer cache.  The default is 30 seconds.   You
#      may also set different timeout values for individual neighbors
#      with the 'connect-timeout' option on a 'cache_peer' line.
#
#Default:
# peer_connect_timeout 30 seconds

#  TAG: read_timeout      time-units
#      The read_timeout is applied on server-side connections.  After
#      each successful read(), the timeout will be extended by this
#      amount.  If no data is read again after this amount of time,
#      the request is aborted and logged with ERR_READ_TIMEOUT.  The
#      default is 15 minutes.
#
#Default:
# read_timeout 15 minutes

#  TAG: request_timeout
#      How long to wait for an HTTP request after initial
#      connection establishment.
#
#Default:
# request_timeout 5 minutes

#  TAG: persistent_request_timeout
#      How long to wait for the next HTTP request on a persistent
#      connection after the previous request completes.
#
#Default:
# persistent_request_timeout 1 minute

#  TAG: client_lifetime   time-units
#      The maximum amount of time a client (browser) is allowed to
#      remain connected to the cache process.  This protects the Cache
#      from having a lot of sockets (and hence file descriptors) tied up
#      in a CLOSE_WAIT state from remote clients that go away without
#      properly shutting down (either because of a network failure or
#      because of a poor client implementation).  The default is one
#      day, 1440 minutes.
#
#      NOTE:  The default value is intended to be much larger than any
#      client would ever need to be connected to your cache.  You
#      should probably change client_lifetime only as a last resort.
#      If you seem to have many client connections tying up
#      filedescriptors, we recommend first tuning the read_timeout,
#      request_timeout, persistent_request_timeout and quick_abort values.
#
#Default:
# client_lifetime 1 day

#  TAG: half_closed_clients
#      Some clients may shutdown the sending side of their TCP
#      connections, while leaving their receiving sides open.     Sometimes,
#      Squid can not tell the difference between a half-closed and a
#      fully-closed TCP connection.  By default, half-closed client
#      connections are kept open until a read(2) or write(2) on the
#      socket returns an error.  Change this option to 'off' and Squid
#      will immediately close client connections when read(2) returns
#      "no more data to read."
#
```

```
#Default:
# half_closed_clients on

#  TAG: pconn_timeout
#      Timeout for idle persistent connections to servers and other
#      proxies.
#
#Default:
# pconn_timeout 120 seconds

#  TAG: ident_timeout
#      Maximum time to wait for IDENT lookups to complete.
#
#      If this is too high, and you enabled IDENT lookups from untrusted
#      users, you might be susceptible to denial-of-service by having
#      many ident requests going at once.
#
#Default:
# ident_timeout 10 seconds

#  TAG: shutdown_lifetime time-units
#      When SIGTERM or SIGHUP is received, the cache is put into
#      "shutdown pending" mode until all active sockets are closed.
#      This value is the lifetime to set for all open descriptors
#      during shutdown mode.  Any active clients after this many
#      seconds will receive a 'timeout' message.
#
#Default:
# shutdown_lifetime 30 seconds


# ACCESS CONTROLS
# -----------------------------------------------------------------------------

#  TAG: acl
#      Defining an Access List
#
#      acl aclname acltype string1 ...
#      acl aclname acltype "file" ...
#
#      when using "file", the file should contain one item per line
#
#      acltype is one of the types described below
#
#      By default, regular expressions are CASE-SENSITIVE.  To make
#      them case-insensitive, use the -i option.
#
#      acl aclname src      ip-address/netmask ... (clients IP address)
#      acl aclname src      addr1-addr2/netmask ... (range of addresses)
#      acl aclname dst      ip-address/netmask ... (URL host's IP address)
#      acl aclname myip     ip-address/netmask ... (local socket IP address)
#
#      acl aclname srcdomain   .foo.com ...    # reverse lookup, client IP
#      acl aclname dstdomain   .foo.com ...    # Destination server from URL
#      acl aclname srcdom_regex [-i] xxx ...   # regex matching client name
#      acl aclname dstdom_regex [-i] xxx ...   # regex matching server
#        # For dstdomain and dstdom_regex  a reverse lookup is tried if a IP
#        # based URL is used. The name "none" is used if the reverse lookup
#        # fails.
#
#      acl aclname time     [day-abbrevs]  [h1:m1-h2:m2]
#          day-abbrevs:
#             S - Sunday
#             M - Monday
#             T - Tuesday
#             W - Wednesday
#             H - Thursday
#             F - Friday
```

```
#           A - Saturday
#         h1:m1 must be less than h2:m2
#      acl aclname url_regex [-i] ^http:// ...# regex matching on whole URL
#      acl aclname urlpath_regex [-i] \.gif$ ...     # regex matching on URL path
#      acl aclname urllogin [-i] [^a-zA-Z0-9] ...    #   regex   matching   on   URL
login field
#      acl aclname port     80 70 21 ...
#      acl aclname port     0-1024 ...        # ranges allowed
#      acl aclname myport   3128 ...          # (local socket TCP port)
#      acl aclname proto    HTTP FTP ...
#      acl aclname method   GET POST ...
#      acl aclname browser  [-i] regexp ...
#        # pattern match on User-Agent header (see also req_header below)
#        acl aclname referer_regex  [-i] regexp ...
#          # pattern match on Referer header
#          # Referer is highly unreliable, so use with care
#      acl aclname ident    username ...
#      acl aclname ident_regex [-i] pattern ...
#        # string match on ident output.
#        # use REQUIRED to accept any non-null ident.
#      acl aclname src_as   number ...
#      acl aclname dst_as   number ...
#        # Except for access control, AS numbers can be used for
#        # routing of requests to specific caches. Here's an
#        # example for routing all requests for AS#1241 and only
#        # those to mycache.mydomain.net:
#        # acl asexample dst_as 1241
#        # cache_peer_access mycache.mydomain.net allow asexample
#        # cache_peer_access mycache_mydomain.net deny all
#
#      acl aclname proxy_auth username ...
#      acl aclname proxy_auth_regex [-i] pattern ...
#        # list of valid usernames
#        # use REQUIRED to accept any valid username.
#        #
#        # NOTE: when a Proxy-Authentication header is sent but it is not
#        # needed during ACL checking the username is NOT logged
#        # in access.log.
#        #
#        # NOTE: proxy_auth requires a EXTERNAL authentication program
#        # to check username/password combinations (see
#        # auth_param directive).
#        #
#        # WARNING: proxy_auth can't be used in a transparent proxy. It
#        # collides with any authentication done by origin servers. It may
#        # seem like it works at first, but it doesn't.
#
#      acl aclname snmp_community string ...
#        # A community string to limit access to your SNMP Agent
#        # Example:
#        #
#        #    acl snmppublic snmp_community public
#
#      acl aclname maxconn number
#        # This will be matched when the client's IP address has
#        # more than <number> HTTP connections established.
#
#      acl aclname max_user_ip [-s] number
#        # This will be matched when the user attempts to log in from more
#        # than <number> different ip addresses. The authenticate_ip_ttl
#        # parameter controls the timeout on the ip entries.
#        # If -s is specified the limit is strict, denying browsing
#        # from any further IP addresses until the ttl has expired. Without
#        # -s Squid will just annoy the user by "randomly" denying requests.
#        # (the counter is reset each time the limit is reached and a
#        # request is denied)
#        # NOTE: in acceleration mode or where there is mesh of child proxies,
#        # clients may appear to come from multiple addresses if they are
```

```
#       # going through proxy farms, so a limit of 1 may cause user problems.
#
#     acl aclname req_mime_type mime-type1 ...
#       # regex match against the mime type of the request generated
#       # by the client. Can be used to detect file upload or some
#       # types HTTP tunelling requests.
#       # NOTE: This does NOT match the reply. You cannot use this
#       # to match the returned file type.
#
#     acl aclname req_header header-name [-i] any\.regex\.here
#       # regex match against any of the known request headers.  May be
#       # thought of as a superset of "browser", "referer" and "mime-type"
#       # acls.
#
#     acl aclname rep_mime_type mime-type1 ...
#       # regex match against the mime type of the reply recieved by
#       # squid. Can be used to detect file download or some
#       # types HTTP tunelling requests.
#       # NOTE: This has no effect in http_access rules. It only has
#       # effect in rules that affect the reply data stream such as
#       # http_reply_access.
#
#     acl aclname rep_header header-name [-i] any\.regex\.here
#       # regex match against any of the known response headers.
#       # Example:
#       #
#       # acl many_spaces rep_header Content-Disposition -i [[:space:]]{3,}
#
#     acl acl_name external class_name [arguments...]
#       # external ACL lookup via a helper class defined by the
#       # external_acl_type directive.
#
#Examples:
#acl myexample dst_as 1241
#acl password proxy_auth REQUIRED
#acl fileupload req_mime_type -i ^multipart/form-data$
#acl javascript rep_mime_type -i ^application/x-javascript$
#
#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
acl Administrativo src 192.168.2.1-192.168.2.19
acl finanzas src 192.168.2.20-192.168.2.49
acl comunicacion src 192.168.2.50-192.168.2.64
acl sistemas2 src 192.168.2.65-192.168.2.88
acl sistemas1 src 192.168.2.89-192.168.2.99
acl DHCP src 192.168.2.100-192.168.2.255
acl prohibidos url_regex sexo hi5 sex chicas porno ares hardcore mamadas tetas
x55.iloveim.com iloveim

#  TAG: http_access
#     Allowing or Denying access based on defined access lists
#
#     Access to the HTTP port:
```

```
#      http_access allow|deny [!]aclname ...
#
#      NOTE on default values:
#
#      If there are no "access" lines present, the default is to deny
#      the request.
#
#      If none of the "access" lines cause a match, the default is the
#      opposite of the last line in the list.  If the last line was
#      deny, the default is allow.  Conversely, if the last line
#      is allow, the default will be deny.  For these reasons, it is a
#      good idea to have an "deny all" or "allow all" entry at the end
#      of your access lists to avoid potential confusion.
#
#Default:
# http_access deny all
#
#Recommended minimum configuration:
#
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager
# Deny requests to unknown ports
http_access deny !Safe_ports
# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports
#
# We strongly recommend the following be uncommented to protect innocent
# web applications running on the proxy server who think the only
# one who can access services on "localhost" is a local user
#http_access deny to_localhost
#
# 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

# And finally deny all other access to this proxy
http_access allow localhost
http_access allow Administrativo
http_access allow finanzas !prohibidos
http_access allow sistemas1 !prohibidos
http_access allow sistemas2 !prohibidos
http_access allow comunicacion !prohibidos
http_access allow DHCP !prohibidos
#http_access deny all

#  TAG: http_reply_access
#        Allow replies to client requests. This is complementary to http_access.
#
#        http_reply_access allow|deny [!] aclname ...
#
#        NOTE: if there are no access lines present, the default is to allow
#      all replies
#
#        If none of the access lines cause a match the opposite of the
#        last line will apply. Thus it is good practice to end the rules
#        with an "allow all" or "deny all" entry.
#
#Default:
# http_reply_access allow all
#
#Recommended minimum configuration:
#
# Insert your own rules here.
```

```
#
#
# and finally allow by default
http_reply_access allow all

#  TAG: icp_access
#      Allowing or Denying access to the ICP port based on defined
#      access lists
#
#      icp_access  allow|deny [!]aclname ...
#
#      See http_access for details
#
#Default:
# icp_access deny all
#
#Allow ICP queries from everyone
icp_access allow all

#  TAG: miss_access
#      Use to force your neighbors to use you as a sibling instead of
#      a parent.  For example:
#
#            acl localclients src 172.16.0.0/16
#            miss_access allow localclients
#            miss_access deny  !localclients
#
#      This means only your local clients are allowed to fetch
#      MISSES and all other clients can only fetch HITS.
#
#      By default, allow all clients who passed the http_access rules
#      to fetch MISSES from us.
#
#Default setting:
# miss_access allow all

#  TAG: cache_peer_access
#      Similar to 'cache_peer_domain' but provides more flexibility by
#      using ACL elements.
#
#      cache_peer_access cache-host allow|deny [!]aclname ...
#
#      The syntax is identical to 'http_access' and the other lists of
#      ACL elements.  See the comments for 'http_access' below, or
#      the Squid FAQ (http://www.squid-cache.org/FAQ/FAQ-10.html).
#
#Default:
# none

#  TAG: ident_lookup_access
#      A list of ACL elements which, if matched, cause an ident
#      (RFC 931) lookup to be performed for this request.  For
#      example, you might choose to always perform ident lookups
#      for your main multi-user Unix boxes, but not for your Macs
#      and PCs.  By default, ident lookups are not performed for
#      any requests.
#
#      To enable ident lookups for specific client addresses, you
#      can follow this example:
#
#      acl ident_aware_hosts src 198.168.1.0/255.255.255.0
#      ident_lookup_access allow ident_aware_hosts
#      ident_lookup_access deny all
#
#      Only src type ACL checks are fully supported.  A src_domain
#      ACL might work at times, but it will not always provide
#      the correct result.
#
```

```
#Default:
# ident_lookup_access deny all

#  TAG: tcp_outgoing_tos
#      Allows you to select a TOS/Diffserv value to mark outgoing
#      connections with, based on the username or source address
#      making the request.
#
#      tcp_outgoing_tos ds-field [!]aclname ...
#
#      Example where normal_service_net uses the TOS value 0x00
#      and normal_service_net uses 0x20
#
#      acl normal_service_net src 10.0.0.0/255.255.255.0
#      acl good_service_net src 10.0.1.0/255.255.255.0
#      tcp_outgoing_tos 0x00 normal_service_net 0x00
#      tcp_outgoing_tos 0x20 good_service_net
#
#      TOS/DSCP values really only have local significance - so you should
#      know what you're specifying. For more, see RFC 2474
#
#      The TOS/DSCP byte must be exactly that - a byte, value  0 - 255, or
#      "default" to use whatever default your host has.
#
#      Processing proceeds in the order specified, and stops at first fully
#      matching line.
#
#Default:
# none

#  TAG: tcp_outgoing_address
#      Allows you to map requests to different outgoing IP addresses
#      based on the username or sourceaddress of the user making
#      the request.
#
#      tcp_outgoing_address ipaddr [[!]aclname] ...
#
#      Example where requests from 10.0.0.0/24 will be forwareded
#      with source address 10.1.0.1, 10.0.2.0/24 forwarded with
#      source address 10.1.0.2 and the rest will be forwarded with
#      source address 10.1.0.3.
#
#      acl normal_service_net src 10.0.0.0/255.255.255.0
#      acl good_service_net src 10.0.1.0/255.255.255.0
#      tcp_outgoing_address 10.0.0.1 normal_service_net
#      tcp_outgoing_address 10.0.0.2 good_service_net
#      tcp_outgoing_address 10.0.0.3
#
#      Processing proceeds in the order specified, and stops at first fully
#      matching line.
#
#Default:
# none

#  TAG: reply_header_max_size    (KB)
#      This specifies the maximum size for HTTP headers in a reply.
#      Reply headers are usually relatively small (about 512 bytes).
#      Placing a limit on the reply header size will catch certain
#      bugs (for example with persistent connections) and possibly
#      buffer-overflow or denial-of-service attacks.
#
#Default:
# reply_header_max_size 20 KB

#  TAG: reply_body_max_size      bytes allow|deny acl acl...
#        This option specifies the maximum size of a reply body in bytes.
#      It can be used to prevent users from downloading very large files,
#      such as MP3's and movies. When the reply headers are recieved,
```

```
#      the reply_body_max_size lines are processed, and the first line with
#      a result of "allow" is used as the maximum body size for this reply.
#      This size is checked twice. First when we get the reply headers,
#      we check the content-length value.  If the content length value exists
#      and is larger than the allowed size, the request is denied and the
#      user receives an error message that says "the request or reply
#      is too large." If there is no content-length, and the reply
#      size exceeds this limit, the client's connection is just closed
#      and they will receive a partial reply.
#
#      WARNING: downstream caches probably can not detect a partial reply
#      if there is no content-length header, so they will cache
#      partial responses and give them out as hits.  You should NOT
#      use this option if you have downstream caches.
#
#      If you set this parameter to zero (the default), there will be
#      no limit imposed.
#
#Default:
# reply_body_max_size 0 allow all


# ADMINISTRATIVE PARAMETERS
# ------------------------------------------------------------------------------

#  TAG: cache_mgr
#      Email-address of local cache manager who will receive
#      mail if the cache dies.  The default is "root".
#cache_mgr root
#
#Default:
# cache_mgr root

#  TAG: cache_effective_user
#      If you start Squid as root, it will change its effective/real
#      UID/GID to the user specified below.  The default is to change
#      to UID to "squid".  If you define cache_effective_user, but not
#      cache_effective_group, Squid sets the GID to the effective
#      user's default group ID (taken from the password file) and
#      supplementary group list from the from groups membership of
#      cache_effective_user.
#cache_effective_user squid
#
#Default:
# cache_effective_user squid

#  TAG: cache_effective_group
#      If you want Squid to run with a specific GID regardless of
#      the group memberships of the effective user then set this
#      to the group (or GID) you want Squid to run as. When set
#      all other group privileges of the effective user is ignored
#      and only this GID is effective. If Squid is not started as
#      root the user starting Squid must be member of the specified
#      group.
#cache_effective_group squid
#
#Default:
# cache_effective_group squid

#  TAG: visible_hostname
#      If you want to present a special hostname in error messages, etc,
#      define this.  Otherwise, the return value of gethostname()
#      will be used. If you have multiple caches in a cluster and
#      get errors about IP-forwarding you must set them to have individual
#      names with this setting.
#
#Default:
# none
```

```
#  TAG: unique_hostname
#      If you want to have multiple machines with the same
#      'visible_hostname' you must give each machine a different
#      'unique_hostname' so forwarding loops can be detected.
#
#Default:
# none

#  TAG: hostname_aliases
#      A list of other DNS names your cache has.
#
#Default:
# none


# OPTIONS FOR THE CACHE REGISTRATION SERVICE
# ----------------------------------------------------------------------------
#
#      This section contains parameters for the (optional) cache
#      announcement service.  This service is provided to help
#      cache administrators locate one another in order to join or
#      create cache hierarchies.
#
#      An 'announcement' message is sent (via UDP) to the registration
#      service by Squid.  By default, the announcement message is NOT
#      SENT unless you enable it with 'announce_period' below.
#
#      The announcement message includes your hostname, plus the
#      following information from this configuration file:
#
#            http_port
http_port 8080
#            icp_port
#            cache_mgr
#
#      All current information is processed regularly and made
#      available on the Web at http://www.ircache.net/Cache/Tracker/.

#  TAG: announce_period
#      This is how frequently to send cache announcements.  The
#      default is `0' which disables sending the announcement
#      messages.
#
#      To enable announcing your cache, just uncomment the line
#      below.
#
#Default:
# announce_period 0
#
#To enable announcing your cache, just uncomment the line below.
#announce_period 1 day

#  TAG: announce_host
#  TAG: announce_file
#  TAG: announce_port
#      announce_host and announce_port set the hostname and port
#      number where the registration message will be sent.
#
#      Hostname will default to 'tracker.ircache.net' and port will
#      default default to 3131.  If the 'filename' argument is given,
#      the contents of that file will be included in the announce
#      message.
#
#Default:
# announce_host tracker.ircache.net
# announce_port 3131
```

```
# HTTPD-ACCELERATOR OPTIONS
# -----------------------------------------------------------------------------

#  TAG: httpd_accel_host
#  TAG: httpd_accel_port
#      If you want to run Squid as an httpd accelerator, define the
#      host name and port number where the real HTTP server is.
#
#      If you want IP based virtual host support specify the
#      hostname as "virtual". This will make Squid use the IP address
#      where it accepted the request as hostname in the URL.
#
#      If you want virtual port support specify the port as "0".
#
#      NOTE: enabling httpd_accel_host disables proxy-caching and
#      ICP.  If you want these features enabled also, set
#      the 'httpd_accel_with_proxy' option.
#
#Default:
# httpd_accel_port 80

#  TAG: httpd_accel_single_host  on|off
#      If you are running Squid as an accelerator and have a single backend
#      server set this to on. This causes Squid to forward the request
#      to this server, regardles of what any redirectors or Host headers
#      say.
#
#      Leave this at off if you have multiple backend servers, and use a
#      redirector (or host table or private DNS) to map the requests to the
#      appropriate backend servers. Note that the mapping needs to be a
#      1-1 mapping between requested and backend (from redirector) domain
#      names or caching will fail, as cacing is performed using the
#      URL returned from the redirector.
#
#      See also redirect_rewrites_host_header.
#
#Default:
# httpd_accel_single_host off

#  TAG: httpd_accel_with_proxy   on|off
#      If you want to use Squid as both a local httpd accelerator
#      and as a proxy, change this to 'on'. Note however your
#      proxy users may have trouble to reach the accelerated domains
#      unless their browsers are configured not to use this proxy for
#      those domains (for example via the no_proxy browser configuration
#      setting)
#
#Default:
# httpd_accel_with_proxy off

#  TAG: httpd_accel_uses_host_header   on|off
#      HTTP/1.1 requests include a Host: header which is basically the
#      hostname from the URL.  The Host: header is used for domain based
#      virutal hosts. If your accelerator needs to provide domain based
#      virtual hosts on the same IP address you will need to turn this
#      on.
#
#      Note Squid does NOT check the value of the Host header matches
#      any of your accelerated server, so it may open a big security hole
#      unless you take care to set up access controls proper.  We recommend
#      this option remain disabled unless you are sure of what you
#      are doing.
#
#      However, you will need to enable this option if you run Squid
#      as a transparent proxy.  Otherwise, virtual servers which
#      require the Host: header will not be properly cached.
#
```

```
#Default:
# httpd_accel_uses_host_header off
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on



# MISCELLANEOUS
# -----------------------------------------------------------------------------

#  TAG: dns_testnames
#      The DNS tests exit as soon as the first site is successfully looked up
#
#      This test can be disabled with the -D command line option.
#
#Default:
# dns_testnames netscape.com internic.net nlanr.net microsoft.com

#  TAG: logfile_rotate
#      Specifies the number of logfile rotations to make when you
#      type 'squid -k rotate'.  The default is 10, which will rotate
#      with extensions 0 through 9.  Setting logfile_rotate to 0 will
#      disable the rotation, but the logfiles are still closed and
#      re-opened.  This will enable you to rename the logfiles
#      yourself just before sending the rotate signal.
#
#      Note, the 'squid -k rotate' command normally sends a USR1
#      signal to the running squid process.  In certain situations
#      (e.g. on Linux with Async I/O), USR1 is used for other
#      purposes, so -k rotate uses another signal.  It is best to get
#      in the habit of using 'squid -k rotate' instead of 'kill -USR1
#      <pid>'.
#
#logfile_rotate 0
#
#Default:
# logfile_rotate 0

#  TAG: append_domain
#      Appends local domain name to hostnames without any dots in
#      them.  append_domain must begin with a period.
#
#      Be warned there are now Internet names with no dots in
#      them using only top-domain names, so setting this may
#      cause some Internet sites to become unavailable.
#
#Example:
# append_domain .yourdomain.com
#
#Default:
# none

#  TAG: tcp_recv_bufsize  (bytes)
#      Size of receive buffer to set for TCP sockets.  Probably just
#      as easy to change your kernel's default.  Set to zero to use
#      the default buffer size.
#
#Default:
# tcp_recv_bufsize 0 bytes

#  TAG: err_html_text
#      HTML text to include in error messages.  Make this a "mailto"
#      URL to your admin address, or maybe just a link to your
#      organizations Web page.
#
#      To include this in your error messages, you must rewrite
```

```
#      the error template files (found in the "errors" directory).
#      Wherever you want the 'err_html_text' line to appear,
#      insert a %L tag in the error template file.
#
#Default:
# none

#  TAG: deny_info
#      Usage:   deny_info err_page_name acl
#      or       deny_info http://... acl
#      Example: deny_info ERR_CUSTOM_ACCESS_DENIED bad_guys
#
#      This can be used to return a ERR_ page for requests which
#      do not pass the 'http_access' rules.  A single ACL will cause
#      the http_access check to fail.  If a 'deny_info' line exists
#      for that ACL Squid returns a corresponding error page.
#
#      You may use ERR_ pages that come with Squid or create your own pages
#      and put them into the configured errors/ directory.
#
#      Alternatively you can specify an error URL. The browsers will
#      get redirected (302) to the specified URL. %s in the redirection
#      URL will be replaced by the requested URL.
#
#      Alternatively you can tell Squid to reset the TCP connection
#      by specifying TCP_RESET.
#
#Default:
# none

#  TAG: memory_pools      on|off
#      If set, Squid will keep pools of allocated (but unused) memory
#      available for future use.  If memory is a premium on your
#      system and you believe your malloc library outperforms Squid
#      routines, disable this.
#
#Default:
# memory_pools on

#  TAG: memory_pools_limit(bytes)
#      Used only with memory_pools on:
#      memory_pools_limit 50 MB
#
#      If set to a non-zero value, Squid will keep at most the specified
#      limit of allocated (but unused) memory in memory pools. All free()
#      requests that exceed this limit will be handled by your malloc
#      library. Squid does not pre-allocate any memory, just safe-keeps
#      objects that otherwise would be free()d. Thus, it is safe to set
#      memory_pools_limit to a reasonably high value even if your
#      configuration will use less memory.
#
#      If not set (default) or set to zero, Squid will keep all memory it
#      can. That is, there will be no limit on the total amount of memory
#      used for safe-keeping.
#
#      To disable memory allocation optimization, do not set
#      memory_pools_limit to 0. Set memory_pools to "off" instead.
#
#      An overhead for maintaining memory pools is not taken into account
#      when the limit is checked. This overhead is close to four bytes per
#      object kept. However, pools may actually _save_ memory because of
#      reduced memory thrashing in your malloc library.
#
#Default:
# none

#  TAG: forwarded_for     on|off
#      If set, Squid will include your system's IP address or name
```

```
#      in the HTTP requests it forwards.  By default it looks like
#      this:
#
#            X-Forwarded-For: 192.1.2.3
#
#      If you disable this, it will appear as
#
#            X-Forwarded-For: unknown
#
#Default:
# forwarded_for on

#  TAG: log_icp_queries   on|off
#      If set, ICP queries are logged to access.log. You may wish
#      do disable this if your ICP load is VERY high to speed things
#      up or to simplify log analysis.
#
#Default:
# log_icp_queries on

#  TAG: icp_hit_stale     on|off
#      If you want to return ICP_HIT for stale cache objects, set this
#      option to 'on'.  If you have sibling relationships with caches
#      in other administrative domains, this should be 'off'.  If you only
#      have sibling relationships with caches under your control,
#      it is probably okay to set this to 'on'.
#      If set to 'on', your siblings should use the option "allow-miss"
#      on their cache_peer lines for connecting to you.
#
#Default:
# icp_hit_stale off

#  TAG: minimum_direct_hops
#      If using the ICMP pinging stuff, do direct fetches for sites
#      which are no more than this many hops away.
#
#Default:
# minimum_direct_hops 4

#  TAG: minimum_direct_rtt
#      If using the ICMP pinging stuff, do direct fetches for sites
#      which are no more than this many rtt milliseconds away.
#
#Default:
# minimum_direct_rtt 400

#  TAG: cachemgr_passwd
#      Specify passwords for cachemgr operations.
#
#      Usage: cachemgr_passwd password action action ...
#
#      Some valid actions are (see cache manager menu for a full list):
#            5min
#            60min
#            asndb
#            authenticator
#            cbdata
#            client_list
#            comm_incoming
#            config *
#            counters
#            delay
#            digest_stats
#            dns
#            events
#            filedescriptors
#            fqdncache
#            histograms
```

```
#            http_headers
#            info
#            io
#            ipcache
#            mem
#            menu
#            netdb
#            non_peers
#            objects
#            offline_toggle *
#            pconn
#            peer_select
#            redirector
#            refresh
#            server_list
#            shutdown *
#            store_digest
#            storedir
#            utilization
#            via_headers
#            vm_objects
#
#      * Indicates actions which will not be performed without a
#        valid password, others can be performed if not listed here.
#
#      To disable an action, set the password to "disable".
#      To allow performing an action without a password, set the
#      password to "none".
#
#      Use the keyword "all" to set the same password for all actions.
#
#Example:
# cachemgr_passwd secret shutdown
# cachemgr_passwd lesssssssecret info stats/objects
# cachemgr_passwd disable all
#
#Default:
# none

#  TAG: store_avg_object_size    (kbytes)
#      Average object size, used to estimate number of objects your
#      cache can hold.  See doc/Release-Notes-1.1.txt.  The default is
#      13 KB.
#
#Default:
# store_avg_object_size 13 KB

#  TAG: store_objects_per_bucket
#      Target number of objects per bucket in the store hash table.
#      Lowering this value increases the total number of buckets and
#      also the storage maintenance rate.  The default is 50.
#
#Default:
# store_objects_per_bucket 20

#  TAG: client_db   on|off
#      If you want to disable collecting per-client statistics,
#      turn off client_db here.
#
#Default:
# client_db on

#  TAG: netdb_low
#  TAG: netdb_high
#      The low and high water marks for the ICMP measurement
#      database.  These are counts, not percents.  The defaults are
#      900 and 1000.  When the high water mark is reached, database
#      entries will be deleted until the low mark is reached.
```

```
#
#Default:
# netdb_low 900
# netdb_high 1000

#  TAG: netdb_ping_period
#      The minimum period for measuring a site.  There will be at
#      least this much delay between successive pings to the same
#      network.  The default is five minutes.
#
#Default:
# netdb_ping_period 5 minutes

#  TAG: query_icmp  on|off
#      If you want to ask your peers to include ICMP data in their ICP
#      replies, enable this option.
#
#      If your peer has configured Squid (during compilation) with
#      '--enable-icmp' that peer will send ICMP pings to origin server
#      sites of the URLs it receives.  If you enable this option the
#      ICP replies from that peer will include the ICMP data (if available).
#      Then, when choosing a parent cache, Squid will choose the parent with
#      the minimal RTT to the origin server.  When this happens, the
#      hierarchy field of the access.log will be
#      "CLOSEST_PARENT_MISS".  This option is off by default.
#
#Default:
# query_icmp off

#  TAG: test_reachability on|off
#      When this is 'on', ICP MISS replies will be ICP_MISS_NOFETCH
#      instead of ICP_MISS if the target host is NOT in the ICMP
#      database, or has a zero RTT.
#
#Default:
# test_reachability off

#  TAG: buffered_logs     on|off
#      cache.log log file is written with stdio functions, and as such
#      it can be buffered or unbuffered. By default it will be unbuffered.
#      Buffering it can speed up the writing slightly (though you are
#      unlikely to need to worry unless you run with tons of debugging
#      enabled in which case performance will suffer badly anyway..).
#
#Default:
# buffered_logs off

#  TAG: reload_into_ims   on|off
#      When you enable this option, client no-cache or ``reload''
#      requests will be changed to If-Modified-Since requests.
#      Doing this VIOLATES the HTTP standard.  Enabling this
#      feature could make you liable for problems which it
#      causes.
#
#      see also refresh_pattern for a more selective approach.
#
#Default:
# reload_into_ims off

#  TAG: always_direct
#      Usage: always_direct allow|deny [!]aclname ...
#
#      Here you can use ACL elements to specify requests which should
#      ALWAYS be forwarded directly to origin servers.  For example,
#      to always directly forward requests for local servers use
#      something like:
#
#            acl local-servers dstdomain my.domain.net
```

```
#            always_direct allow local-servers
#
#     To always forward FTP requests directly, use
#
#            acl FTP proto FTP
#            always_direct allow FTP
#
#     NOTE: There is a similar, but opposite option named
#     'never_direct'.  You need to be aware that "always_direct deny
#     foo" is NOT the same thing as "never_direct allow foo".  You
#     may need to use a deny rule to exclude a more-specific case of
#     some other rule.  Example:
#
#            acl local-external dstdomain external.foo.net
#            acl local-servers dstdomain  .foo.net
#            always_direct deny local-external
#            always_direct allow local-servers
#
#     This option replaces some v1.1 options such as local_domain
#     and local_ip.
#
#Default:
# none

#  TAG: never_direct
#     Usage: never_direct allow|deny [!]aclname ...
#
#     never_direct is the opposite of always_direct.  Please read
#     the description for always_direct if you have not already.
#
#     With 'never_direct' you can use ACL elements to specify
#     requests which should NEVER be forwarded directly to origin
#     servers.  For example, to force the use of a proxy for all
#     requests, except those in your local domain use something like:
#
#            acl local-servers dstdomain .foo.net
#            acl all src 0.0.0.0/0.0.0.0
#            never_direct deny local-servers
#            never_direct allow all
#
#     or if Squid is inside a firewall and there are local intranet
#     servers inside the firewall use something like:
#
#            acl local-intranet dstdomain .foo.net
#            acl local-external dstdomain external.foo.net
#            always_direct deny local-external
#            always_direct allow local-intranet
#            never_direct allow all
#
#     This option replaces some v1.1 options such as inside_firewall
#     and firewall_ip.
#
#Default:
# none

#  TAG: header_access
#     Usage: header_access header_name allow|deny [!]aclname ...
#
#     WARNING: Doing this VIOLATES the HTTP standard.  Enabling
#     this feature could make you liable for problems which it
#     causes.
#
#     This option replaces the old 'anonymize_headers' and the
#     older 'http_anonymizer' option with something that is much
#     more configurable. This new method creates a list of ACLs
#     for each header, allowing you very fine-tuned header
#     mangling.
#
```

```
#      You can only specify known headers for the header name.
#      Other headers are reclassified as 'Other'. You can also
#      refer to all the headers with 'All'.
#
#      For example, to achieve the same behaviour as the old
#      'http_anonymizer standard' option, you should use:
#
#            header_access From deny all
#            header_access Referer deny all
#            header_access Server deny all
#            header_access User-Agent deny all
#            header_access WWW-Authenticate deny all
#            header_access Link deny all
#
#      Or, to reproduce the old 'http_anonymizer paranoid' feature
#      you should use:
#
#            header_access Allow allow all
#            header_access Authorization allow all
#            header_access WWW-Authenticate allow all
#            header_access Cache-Control allow all
#            header_access Content-Encoding allow all
#            header_access Content-Length allow all
#            header_access Content-Type allow all
#            header_access Date allow all
#            header_access Expires allow all
#            header_access Host allow all
#            header_access If-Modified-Since allow all
#            header_access Last-Modified allow all
#            header_access Location allow all
#            header_access Pragma allow all
#            header_access Accept allow all
#            header_access Accept-Charset allow all
#            header_access Accept-Encoding allow all
#            header_access Accept-Language allow all
#            header_access Content-Language allow all
#            header_access Mime-Version allow all
#            header_access Retry-After allow all
#            header_access Title allow all
#            header_access Connection allow all
#            header_access Proxy-Connection allow all
#            header_access All deny all
#
#      By default, all headers are allowed (no anonymizing is
#      performed).
#
#Default:
# none

#  TAG: header_replace
#      Usage:   header_replace header_name message
#      Example: header_replace User-Agent Nutscrape/1.0 (CP/M; 8-bit)
#
#      This option allows you to change the contents of headers
#      denied with header_access above, by replacing them with
#      some fixed string. This replaces the old fake_user_agent
#      option.
#
#      By default, headers are removed if denied.
#
#Default:
# none

#  TAG: icon_directory
#      Where the icons are stored. These are normally kept in
#      /usr/share/squid/icons
#
#Default:
```

```
# icon_directory /usr/share/squid/icons

#  TAG: short_icon_urls
#      If this is enabled Squid will use short URLs for icons.
#
#      If off the URLs for icons will always be absolute URLs
#      including the proxy name and port.
#
#Default:
# short_icon_urls off

#  TAG: error_directory
#      Directory where the error files are read from.
#      /usr/lib/squid/errors contains sets of error files
#      in different languages. The default error directory
#      is /etc/squid/errors, which is a link to one of these
#      error sets.
#
#      If you wish to create your own versions of the error files,
#      either to customize them to suit your language or company,
#      copy the template English files to another
#      directory and point this tag at them.
#
#error_directory /etc/squid/errors
#
#Default:
# error_directory /etc/squid/errors

#  TAG: maximum_single_addr_tries
#      This sets the maximum number of connection attempts for a
#      host that only has one address (for multiple-address hosts,
#      each address is tried once).
#
#      The default value is one attempt, the (not recommended)
#      maximum is 255 tries.  A warning message will be generated
#      if it is set to a value greater than ten.
#
#      Note: This is in addition to the request reforwarding which
#      takes place if Squid fails to get a satisfying response.
#
#Default:
# maximum_single_addr_tries 1

#  TAG: snmp_port
#      Squid can now serve statistics and status information via SNMP.
#      A value of "0" disables SNMP support. If you wish to use SNMP,
#      set this to "3401" to use the normal SNMP support.
#
#Default:
# snmp_port 0

#  TAG: snmp_access
#      Allowing or denying access to the SNMP port.
#
#      All access to the agent is denied by default.
#      usage:
#
#      snmp_access allow|deny [!]aclname ...
#
#Example:
# snmp_access allow snmppublic localhost
# snmp_access deny all
#
#Default:
# snmp_access deny all

#  TAG: snmp_incoming_address
#  TAG: snmp_outgoing_address
```

```
#      Just like 'udp_incoming_address' above, but for the SNMP port.
#
#      snmp_incoming_address     is used for the SNMP socket receiving
#                        messages from SNMP agents.
#      snmp_outgoing_address     is used for SNMP packets returned to SNMP
#                        agents.
#
#      The default snmp_incoming_address (0.0.0.0) is to listen on all
#      available network interfaces.
#
#      If snmp_outgoing_address is set to 255.255.255.255 (the default)
#      it will use the same socket as snmp_incoming_address. Only
#      change this if you want to have SNMP replies sent using another
#      address than where this Squid listens for SNMP queries.
#
#      NOTE, snmp_incoming_address and snmp_outgoing_address can not have
#      the same value since they both use port 3401.
#
#Default:
# snmp_incoming_address 0.0.0.0
# snmp_outgoing_address 255.255.255.255

#  TAG: as_whois_server
#      WHOIS server to query for AS numbers.  NOTE: AS numbers are
#      queried only when Squid starts up, not for every request.
#
#Default:
# as_whois_server whois.ra.net
# as_whois_server whois.ra.net

#  TAG: wccp_router
#      Use this option to define your WCCP ``home'' router for
#      Squid.   Setting the 'wccp_router' to 0.0.0.0 (the default)
#      disables WCCP.
#
#Default:
# wccp_router 0.0.0.0

#  TAG: wccp_version
#      According to some users, Cisco IOS 11.2 only supports WCCP
#      version 3.  If you're using that version of IOS, change
#      this value to 3.
#
#Default:
# wccp_version 4

#  TAG: wccp_incoming_address
#  TAG: wccp_outgoing_address
#        wccp_incoming_address   Use this option if you require WCCP
#                        messages to be received on only one
#                        interface.  Do NOT use this option if
#                        you're unsure how many interfaces you
#                        have, or if you know you have only one
#                        interface.
#
#      wccp_outgoing_address     Use this option if you require WCCP
#                        messages to be sent out on only one
#                        interface.  Do NOT use this option if
#                        you're unsure how many interfaces you
#                        have, or if you know you have only one
#                        interface.
#
#        The default behavior is to not bind to any specific address.
#
#        NOTE, wccp_incoming_address and wccp_outgoing_address can not have
#        the same value since they both use port 2048.
#
#Default:
```

```
# wccp_incoming_address 0.0.0.0
# wccp_outgoing_address 255.255.255.255


# DELAY POOL PARAMETERS (all require DELAY_POOLS compilation option)
# -----------------------------------------------------------------------------

#  TAG: delay_pools
#      This represents the number of delay pools to be used.  For example,
#      if you have one class 2 delay pool and one class 3 delays pool, you
#      have a total of 2 delay pools.
#
#Default:
# delay_pools 0

#  TAG: delay_class
#      This defines the class of each delay pool.  There must be exactly one
#      delay_class line for each delay pool.  For example, to define two
#      delay pools, one of class 2 and one of class 3, the settings above
#      and here would be:
#
#Example:
# delay_pools 2      # 2 delay pools
# delay_class 1 2    # pool 1 is a class 2 pool
# delay_class 2 3    # pool 2 is a class 3 pool
#
#      The delay pool classes are:
#
#            class 1             Everything is limited by a single aggregate
#                        bucket.
#
#            class 2      Everything is limited by a single aggregate
#                        bucket as well as an "individual" bucket chosen
#                        from bits 25 through 32 of the IP address.
#
#            class 3             Everything is limited by a single aggregate
#                        bucket as well as a "network" bucket chosen
#                        from bits 17 through 24 of the IP address and a
#                        "individual" bucket chosen from bits 17 through
#                        32 of the IP address.
#
#      NOTE: If an IP address is a.b.c.d
#            -> bits 25 through 32 are "d"
#            -> bits 17 through 24 are "c"
#            -> bits 17 through 32 are "c * 256 + d"
#
#Default:
# none

#  TAG: delay_access
#      This is used to determine which delay pool a request falls into.
#      The first matched delay pool is always used, i.e., if a request falls
#      into delay pool number one, no more delay are checked, otherwise the
#      rest are checked in order of their delay pool number until they have
#      all been checked.  For example, if you want some_big_clients in delay
#      pool 1 and lotsa_little_clients in delay pool 2:
#
#Example:
# delay_access 1 allow some_big_clients
# delay_access 1 deny all
# delay_access 2 allow lotsa_little_clients
# delay_access 2 deny all
#
#Default:
# none

#  TAG: delay_parameters
#      This defines the parameters for a delay pool.  Each delay pool has
```

```
#      a number of "buckets" associated with it, as explained in the
#      description of delay_class.  For a class 1 delay pool, the syntax is:
#
#delay_parameters pool aggregate
#
#      For a class 2 delay pool:
#
#delay_parameters pool aggregate individual
#
#      For a class 3 delay pool:
#
#delay_parameters pool aggregate network individual
#
#      The variables here are:
#
#            pool         a pool number - ie, a number between 1 and the
#                         number specified in delay_pools as used in
#                         delay_class lines.
#
#            aggregate    the "delay parameters" for the aggregate bucket
#                         (class 1, 2, 3).
#
#            individual   the "delay parameters" for the individual
#                         buckets (class 2, 3).
#
#            network             the "delay parameters" for the network buckets
#                         (class 3).
#
#      A pair of delay parameters is written restore/maximum, where restore is
#      the number of bytes (not bits - modem and network speeds are usually
#      quoted in bits) per second placed into the bucket, and maximum is the
#      maximum number of bytes which can be in the bucket at any time.
#
#      For example, if delay pool number 1 is a class 2 delay pool as in the
#      above example, and is being used to strictly limit each host to 64kbps
#      (plus overheads), with no overall limit, the line is:
#
#delay_parameters 1 -1/-1 8000/8000
#
#      Note that the figure -1 is used to represent "unlimited".
#
#      And, if delay pool number 2 is a class 3 delay pool as in the above
#      example, and you want to limit it to a total of 256kbps (strict limit)
#      with each 8-bit network permitted 64kbps (strict limit) and each
#      individual host permitted 4800bps with a bucket maximum size of 64kb
#      to permit a decent web page to be downloaded at a decent speed
#      (if the network is not being limited due to overuse) but slow down
#      large downloads more significantly:
#
#delay_parameters 2 32000/32000 8000/8000 600/8000
#
#      There must be one delay_parameters line for each delay pool.
#
#Default:
# none

#  TAG: delay_initial_bucket_level     (percent, 0-100)
#      The initial bucket percentage is used to determine how much is put
#      in each bucket when squid starts, is reconfigured, or first notices
#      a host accessing it (in class 2 and class 3, individual hosts and
#      networks only have buckets associated with them once they have been
#      "seen" by squid).
#
#Default:
# delay_initial_bucket_level 50

#  TAG: incoming_icp_average
#  TAG: incoming_http_average
```

```
#  TAG: incoming_dns_average
#  TAG: min_icp_poll_cnt
#  TAG: min_dns_poll_cnt
#  TAG: min_http_poll_cnt
#      Heavy voodoo here.  I can't even believe you are reading this.
#      Are you crazy?  Don't even think about adjusting these unless
#      you understand the algorithms in comm_select.c first!
#
#Default:
# incoming_icp_average 6
# incoming_http_average 4
# incoming_dns_average 4
# min_icp_poll_cnt 8
# min_dns_poll_cnt 8
# min_http_poll_cnt 8

#  TAG: max_open_disk_fds
#      To avoid having disk as the I/O bottleneck Squid can optionally
#      bypass the on-disk cache if more than this amount of disk file
#      descriptors are open.
#
#      A value of 0 indicates no limit.
#
#Default:
# max_open_disk_fds 0

#  TAG: offline_mode
#      Enable this option and Squid will never try to validate cached
#      objects.
#
#Default:
# offline_mode off

#  TAG: uri_whitespace
#      What to do with requests that have whitespace characters in the
#      URI.  Options:
#
#      strip:  The whitespace characters are stripped out of the URL.
#            This is the behavior recommended by RFC2396.
#      deny:   The request is denied.  The user receives an "Invalid
#            Request" message.
#      allow:  The request is allowed and the URI is not changed.  The
#            whitespace characters remain in the URI.  Note the
#            whitespace is passed to redirector processes if they
#            are in use.
#      encode:      The request is allowed and the whitespace characters are
#            encoded according to RFC1738.  This could be considered
#            a violation of the HTTP/1.1
#            RFC because proxies are not allowed to rewrite URI's.
#      chop:  The request is allowed and the URI is chopped at the
#            first whitespace.  This might also be considered a
#            violation.
#
#Default:
# uri_whitespace strip

#  TAG: broken_posts
#      A list of ACL elements which, if matched, causes Squid to send
#      an extra CRLF pair after the body of a PUT/POST request.
#
#      Some HTTP servers has broken implementations of PUT/POST,
#      and rely on an extra CRLF pair sent by some WWW clients.
#
#      Quote from RFC 2068 section 4.1 on this matter:
#
#        Note: certain buggy HTTP/1.0 client implementations generate an
#        extra CRLF's after a POST request. To restate what is explicitly
#        forbidden by the BNF, an HTTP/1.1 client must not preface or follow
```

```
#       a request with an extra CRLF.
#
#Example:
# acl buggy_server url_regex ^http://....
# broken_posts allow buggy_server
#
#Default:
# none

#  TAG: mcast_miss_addr
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
#      If you enable this option, every "cache miss" URL will
#      be sent out on the specified multicast address.
#
#      Do not enable this option unless you are are absolutely
#      certain you understand what you are doing.
#
#Default:
# mcast_miss_addr 255.255.255.255

#  TAG: mcast_miss_ttl
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_TTL option
#
#      This is the time-to-live value for packets multicasted
#      when multicasting off cache miss URLs is enabled.  By
#      default this is set to 'site scope', i.e. 16.
#
#Default:
# mcast_miss_ttl 16

#  TAG: mcast_miss_port
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
#      This is the port number to be used in conjunction with
#      'mcast_miss_addr'.
#
#Default:
# mcast_miss_port 3135

#  TAG: mcast_miss_encode_key
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
#      The URLs that are sent in the multicast miss stream are
#      encrypted.  This is the encryption key.
#
#Default:
# mcast_miss_encode_key XXXXXXXXXXXXXXXX

#  TAG: nonhierarchical_direct
#      By default, Squid will send any non-hierarchical requests
#      (matching hierarchy_stoplist or not cachable request type) direct
#      to origin servers.
#
#      If you set this to off, Squid will prefer to send these
#      requests to parents.
#
#      Note that in most configurations, by turning this off you will only
#      add latency to these request without any improvement in global hit
#      ratio.
#
#      If you are inside an firewall see never_direct instead of
#      this directive.
#
```

```
#Default:
# nonhierarchical_direct on

#  TAG: prefer_direct
#      Normally Squid tries to use parents for most requests. If you for some
#      reason like it to first try going direct and only use a parent if
#      going direct fails set this to on.
#
#      By combining nonhierarchical_direct off and prefer_direct on you
#      can set up Squid to use a parent as a backup path if going direct
#      fails.
#
#      Note: If you want Squid to use parents for all requests see
#      the never_direct directive. prefer_direct only modifies how Squid
#      acts on cachable requests.
#
#Default:
# prefer_direct off

#  TAG: strip_query_terms
#      By default, Squid strips query terms from requested URLs before
#      logging.  This protects your user's privacy.
#
#Default:
# strip_query_terms on

#  TAG: coredump_dir
#      By default Squid leaves core files in the directory from where
#      it was started. If you set 'coredump_dir' to a directory
#      that exists, Squid will chdir() to that directory at startup
#      and coredump files will be left there.
#
#Default:
# coredump_dir none
#
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid

#  TAG: redirector_bypass
#      When this is 'on', a request will not go through the
#      redirector if all redirectors are busy.  If this is 'off'
#      and the redirector queue grows too large, Squid will exit
#      with a FATAL error and ask you to increase the number of
#      redirectors.  You should only enable this if the redirectors
#      are not critical to your caching system.  If you use
#      redirectors for access control, and you enable this option,
#      users may have access to pages they should not
#      be allowed to request.
#
#Default:
# redirector_bypass off

#  TAG: ignore_unknown_nameservers
#      By default Squid checks that DNS responses are received
#      from the same IP addresses they are sent to.  If they
#      don't match, Squid ignores the response and writes a warning
#      message to cache.log.  You can allow responses from unknown
#      nameservers by setting this option to 'off'.
#
#Default:
# ignore_unknown_nameservers on

#  TAG: digest_generation
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#      This controls whether the server will generate a Cache Digest
#      of its contents.  By default, Cache Digest generation is
```

```
#      enabled if Squid is compiled with USE_CACHE_DIGESTS defined.
#
#Default:
# digest_generation on

#  TAG: digest_bits_per_entry
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#      This is the number of bits of the server's Cache Digest which
#      will be associated with the Digest entry for a given HTTP
#      Method and URL (public key) combination.  The default is 5.
#
#Default:
# digest_bits_per_entry 5

#  TAG: digest_rebuild_period    (seconds)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#      This is the number of seconds between Cache Digest rebuilds.
#
#Default:
# digest_rebuild_period 1 hour

#  TAG: digest_rewrite_period    (seconds)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#      This is the number of seconds between Cache Digest writes to
#      disk.
#
#Default:
# digest_rewrite_period 1 hour

#  TAG: digest_swapout_chunk_size      (bytes)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#      This is the number of bytes of the Cache Digest to write to
#      disk at a time.  It defaults to 4096 bytes (4KB), the Squid
#      default swap page.
#
#Default:
# digest_swapout_chunk_size 4096 bytes

#  TAG: digest_rebuild_chunk_percentage(percent, 0-100)
# Note: This option is only available if Squid is rebuilt with the
#       --enable-cache-digests option
#
#      This is the percentage of the Cache Digest to be scanned at a
#      time.  By default it is set to 10% of the Cache Digest.
#
#Default:
# digest_rebuild_chunk_percentage 10

#  TAG: chroot
#      Use this to have Squid do a chroot() while initializing.  This
#      also causes Squid to fully drop root privileges after
#      initializing.  This means, for example, that if you use a HTTP
#      port less than 1024 and try to reconfigure, you will get an
#      error.
#
#Default:
# none

#  TAG: client_persistent_connections
#  TAG: server_persistent_connections
```

```
#      Persistent connection support for clients and servers.  By
#      default, Squid uses persistent connections (when allowed)
#      with its clients and servers.  You can use these options to
#      disable persistent connections with clients and/or servers.
#
#Default:
# client_persistent_connections on
# server_persistent_connections on

#  TAG: detect_broken_pconn
#      Some servers have been found to incorrectly signal the use
#      of HTTP/1.0 persistent connections even on replies not
#      compatible, causing significant delays. This server problem
#      has mostly been seen on redirects.
#
#      By enabling this directive Squid attempts to detect such
#      broken replies and automatically assume the reply is finished
#      after 10 seconds timeout.
#
#Default:
# detect_broken_pconn off

#  TAG: balance_on_multiple_ip
#      Some load balancing servers based on round robin DNS have been
#      found not to preserve user session state across requests
#      to different IP addresses.
#
#      By default Squid rotates IP's per request. By disabling
#      this directive only connection failure trigers rotation.
#
#Default:
# balance_on_multiple_ip on

#  TAG: pipeline_prefetch
#      To boost the performance of pipelined requests to closer
#      match that of a non-proxied environment Squid can try to fetch
#      up to two requests in parallell from a pipeline.
#
#      Defaults to off for bandwidth management and access logging
#      reasons.
#
#Default:
# pipeline_prefetch off

#  TAG: extension_methods
#      Squid only knows about standardized HTTP request methods.
#      You can add up to 20 additional "extension" methods here.
#
#Default:
# none

#  TAG: request_entities
#      Squid defaults to deny GET and HEAD requests with request entities,
#      as the meaning of such requests are undefined in the HTTP standard
#      even if not explicitly forbidden.
#
#      Set this directive to on if you have clients which insists
#      on sending request entities in GET or HEAD requests.
#
#Default:
# request_entities off

#  TAG: high_response_time_warning     (msec)
#      If the one-minute median response time exceeds this value,
#      Squid prints a WARNING with debug level 0 to get the
#      administrators attention.  The value is in milliseconds.
#
#Default:
```

```
# high_response_time_warning 0

#  TAG: high_page_fault_warning
#      If the one-minute average page fault rate exceeds this
#      value, Squid prints a WARNING with debug level 0 to get
#      the administrators attention.  The value is in page faults
#      per second.
#
#Default:
# high_page_fault_warning 0

#  TAG: high_memory_warning
#      If the memory usage (as determined by mallinfo) exceeds
#      value, Squid prints a WARNING with debug level 0 to get
#      the administrators attention.
#
#Default:
# high_memory_warning 0

#  TAG: store_dir_select_algorithm
#      Set this to 'round-robin' as an alternative.
#
#Default:
# store_dir_select_algorithm least-load

#  TAG: forward_log
# Note: This option is only available if Squid is rebuilt with the
#       -DWIP_FWD_LOG option
#
#      Logs the server-side requests.
#
#      This is currently work in progress.
#
#Default:
# none

#  TAG: ie_refresh  on|off
#      Microsoft Internet Explorer up until version 5.5 Service
#      Pack 1 has an issue with transparent proxies, wherein it
#      is impossible to force a refresh.  Turning this on provides
#      a partial fix to the problem, by causing all IMS-REFRESH
#      requests from older IE versions to check the origin server
#      for fresh content.  This reduces hit ratio by some amount
#      (~10% in my experience), but allows users to actually get
#      fresh content when they want it.  Note that because Squid
#      cannot tell if the user is using 5.5 or 5.5SP1, the behavior
#      of 5.5 is unchanged from old versions of Squid (i.e. a
#      forced refresh is impossible).  Newer versions of IE will,
#      hopefully, continue to have the new behavior and will be
#      handled based on that assumption.  This option defaults to
#      the old Squid behavior, which is better for hit ratios but
#      worse for clients using IE, if they need to be able to
#      force fresh content.
#
#Default:
# ie_refresh off

#  TAG: vary_ignore_expire on|off
#      Many HTTP servers supporting Vary gives such objects
#      immediate expiry time with no cache-control header
#      when requested by a HTTP/1.0 client. This option
#      enables Squid to ignore such expiry times until
#      HTTP/1.1 is fully implemented.
#      WARNING: This may eventually cause some varying
#      objects not intended for caching to get cached.
#
#Default:
# vary_ignore_expire off
```

```
#  TAG: sleep_after_fork  (microseconds)
#      When this is set to a non-zero value, the main Squid process
#      sleeps the specified number of microseconds after a fork()
#      system call. This sleep may help the situation where your
#      system reports fork() failures due to lack of (virtual)
#      memory. Note, however, that if you have a lot of child
#      processes, these sleep delays will add up and your
#      Squid will not service requests for some amount of time
#      until all the child processes have been started.
#
#Default:
# sleep_after_fork 0

#  TAG: relaxed_header_parser    on|off
#      Set this to off if you want Squid to be strict about
#      the HTTP protocol syntax and reject non-compliant requests
#      or responses.
#
#      In the default "on" setting Squid accepts certain forms
#      of non-compliant HTTP messages where it is unambigous
#      what the sending application indended even if the message
#      is not correctly formatted. The messages is then normalized
#      to the correct form when forwarded by Squid.
#
#Default:
# relaxed_header_parser on
```

Trabajo de fin de carrera

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL ESQUEMA DE SEGURIDAD PERIMETRAL PARA LA RED DE DATOS DE LA UISEK – ECUADOR (CAMPUS MIGUEL DE CERVANTES)

INTRODUCCIÓN

- El desarrollo tecnológico hace que las organizaciones estén expuestas.
- El alto nivel de exposición puede afectar al negocio.
- Carácter mas reactivo que preventivo.

ANTECEDENTES

- En los años 70 aparecen los primeros casos de intervención telefónica.
- Por el año 1987 empiezan a aparecer los primeros virus.
- Por el año 1999 surge el E-Commerce y las empresas empiezan a exponerse en Internet.

ANTECEDENTES ECUADOR

- En 1996 la banca empieza a ofrecer servicios de banca electrónica.

- Este mismo año el servicio de Internet llega al Ecuador, los ISP no son seguros.

- La seguridad de los sistemas IT es mas enfocada a desastres naturales.

SITUACIÓN ACTUAL

- La información es el activo mas valorado por las organizaciones.
- Los ataques informáticos generaron alrededor de 66 millones de dólares en pérdidas en el 2007 (Muestra: 194 empresas).
- Internet incrementó el nivel de facilidad para perpetrar cualquier tipo de ataque informático.

SEGURIDAD INFORMÁTICA

Disponibilidad

Integridad

Confidencialidad

TIPOS DE AMENAZAS

- Ataques Internos y Externos
- Los ataques internos son más frecuentes y causan mas daño que los externos.
- Ingeniería Social

SEGURIDAD PERIMETRAL

- Concepto emergente que surgió por la rápida expansión de las redes internas hacia Internet.

- Consiste en vigiliar el perímetro o borde externo de la red privada ante las amenazas de la red externa (Internet).

OBJETIVOS Y COMPONENTES

- Proteger la red interna de las posibles intrusiones de la red externa.
- Filtrar los accesos hacia la red privada.
- Tomar acción ante cualquier intrusión

- Ruteadores, Firewall, IDS, VPN, DMZ

IMPORTANCIA

- La seguridad informática también es parte crítica del negocio de una organización
- La seguridad perimetral se enfoca en garantizar la integridad de los sistemas IT.
- La seguridad informática se enfoca también en el ”factor humano”.

OBJETIVOS
TRABAJO DE INVESTIGACIÓN

- Análisis de riesgos
- Selección de contramedidas de seguridad
- Diseño e implementación del esquema de seguridad perimetral.
- Diseño de un plan de monitoreo para el esquema de seguridad perimetral.

ANÁLISIS DE RIESGOS

- ISO/IEC 13335:3 – Técnicas para la administración de seguridad IT.

- Baseline Approach
- Manual de políticas de seguridad IT
- Plan de contingencia
- Análisis de riesgos

POLÍTICAS DE SEGURIDAD

- Seguridad Organizacional
- Seguridad Lógica
- Seguridad Física
- Seguridad Legal

PLAN DE CONTINGENCIA

- Reanudar operaciones en el menor tiempo posible.

- Equipo Gerencial
- Equipo de Seguridad

CRITERIO DE DESASTRE

- Cualquier interrupción en las operaciones del computador, que potencialmente pudiera exceder de un (1) día, desde el momento que una evaluación de la situación es realizada y una situación de Contingencia es declarada.

ANÁLISIS DE RIESGOS

- Identificación de activos
- Valoración de activos
- Análisis de amenazas
- Análisis de vulnerabilidades
- Análisis de riesgos encontrados

ACTIVOS CRÍTICOS

- Servidor de Internet
- Switches de enlace a centros de cómputo de Ed. Principal
- Enlace de fibra óptica
- Enlace de par trenzado
- Ruteador de Internet
- Equipo de Asistente Administrativa (M)
- Switches de enlace a edificios colindantes (M)
- Servidor de dominio (B)

ANÁLISIS DE AMENAZAS

- Alta probabilidad:
  - Inundación
  - Fuego
  - Daño voluntario
  - Temperatura muy alta
  - Variación de voltaje
  - Fallo de componentes de red
  - Suplantación de identidad

ANÁLISIS DE AMENAZAS

- Amenazas con alto índice de posible daño:
  - Inundación
  - Falla de suministro eléctrico
  - Variación de voltaje
  - Error de mantenimiento

ANÁLISIS DE VULNERABILIDADES

- Ambiente e infraestructura:
  - Puertas de centros de cómputo
  - Área propensa a inundación
  - Control de temperatura
  - Control contra incendios
  - Ataques internos

ANÁLISIS DE VULNERABILIDADES

- Hardware:
  - Fallos por temperatura
  - Fallos por voltaje inestable
  - Carga electrostática

ANÁLISIS DE VULNERABILIDADES

- Comunicaciones:
  - Carencia de autenticación por persona
  - Transferencia de contraseñas

ANÁLISIS DE VULNERABILIDADES

- Personal:
  - Trabajos de terceros sin suficiente control
  - Capacitación sobre seguridad informática ,insuficiente

ANÁLISIS DE RIESGOS

- Servidor de Internet:
  - Temperatura muy alta
  - Daño voluntario
  - Suplantación de identidad
  - Fallo de hardware

ANÁLISIS DE RIESGOS

- Servidor de dominio:
  - Temperatura muy alta
  - Suplantación de identidad
  - Daño voluntario
  - Software instalado por personas no autorizadas

ANÁLISIS DE RIESGOS

- Switches de enlace a facultades colindantes:
  - Temperatura muy alta
  - Daño voluntario
  - Robo
  - Variación de voltaje

ANÁLISIS DE RIESGOS

- Switches de enlace a centros de cómputo del edificio principal:
  - Temperatura muy alta
  - Daño voluntario
  - Robo
  - Variación de voltaje

ANÁLISIS DE RIESGOS

- Ruteador de Internet:
  - Temperatura muy alta
  - Variación de voltaje
  - Daño voluntario
  - Relámpagos

ANÁLISIS DE RIESGOS

- Enlace de fibra óptica:
  - Daño voluntario
  - Robo
  - Error de mantenimiento
  - Fuego

ANÁLISIS DE RIESGOS

- Enlace de par trenzado:
  - Daño voluntario
  - Robo
  - Error de mantenimiento
  - Inundación

ANÁLISIS DE RIESGOS

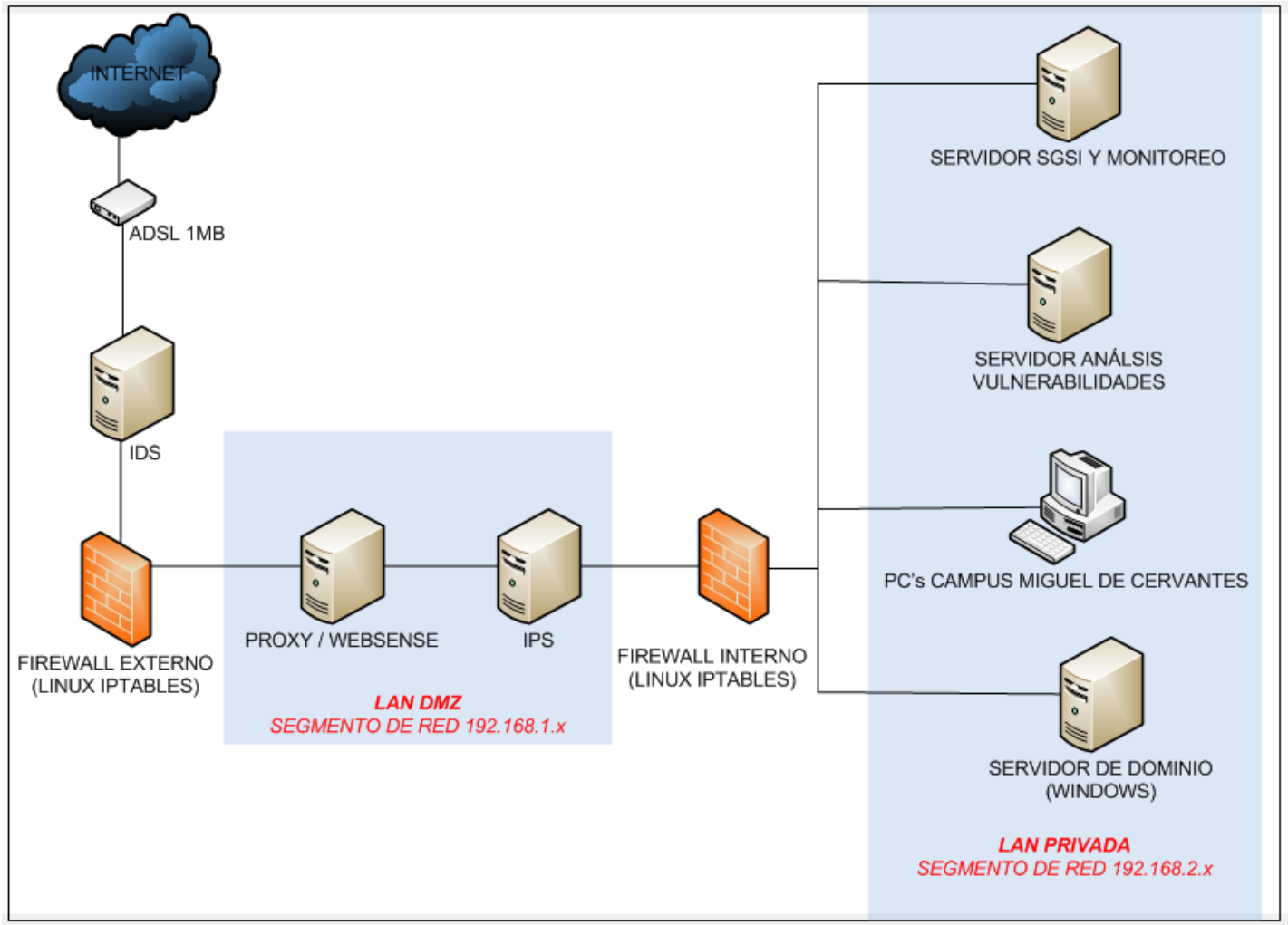
- Equipo asistente administrativa:
  - Suplantación de identidad
  - Acceso a la red por personas no autorizadas
  - Software malicioso
  - Mal uso de los recursos

CONTRAMEDIDAS

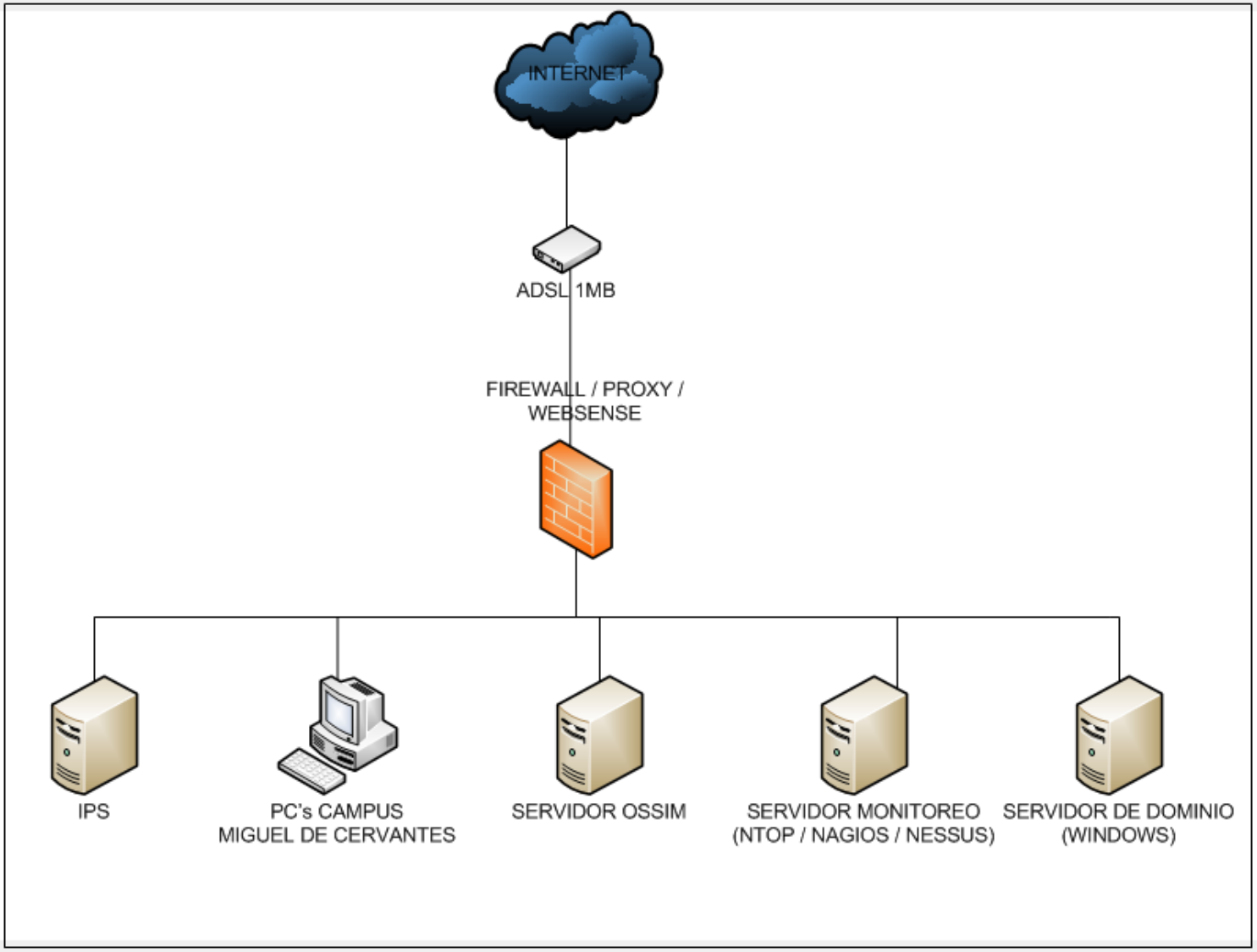
- ISO/IEC 13335:4 – Selección de contramedidas de seguridad.

- Manejo de incidentes
- Personal
- Tareas operacionales
- Seguridad física

ESQUEMA DE SEG. PERIMETRAL (PROPUESTO)



IMPLEMENTACIÓN



OSSIM

- Sistema de Gestión de Seguridad Informática

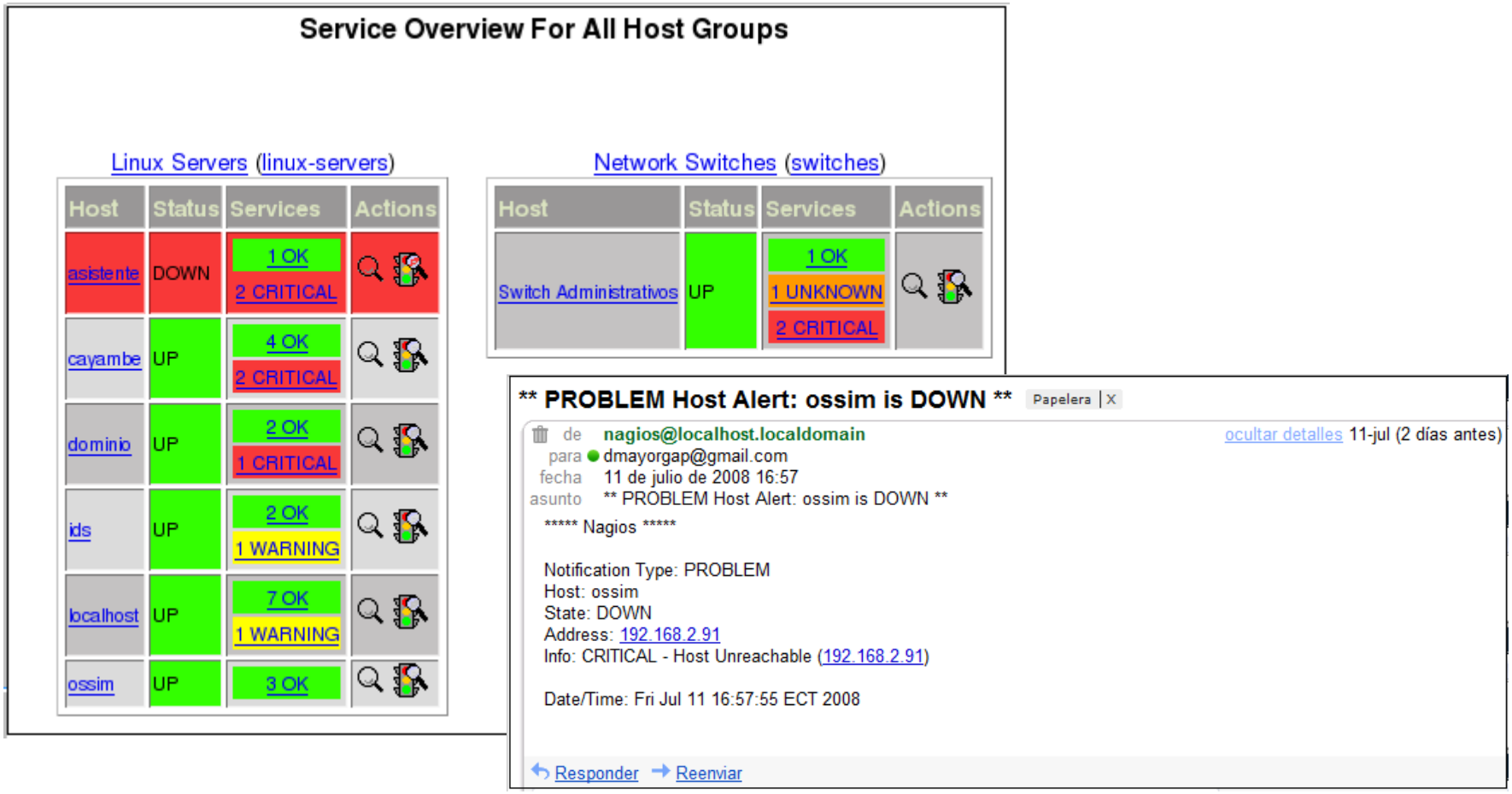
- Sensores
- Servidor de Gestión
- Base de datos
- Frontend

SERVIDOR DE MONITOREO

- **NAGIOS:** Aplicación de monitoreo de disponibilidad.
- **NTOP:** Herramienta de monitoreo de tráfico.
- **NESSUS:** Analizador pasivo de vulnerabilidades.

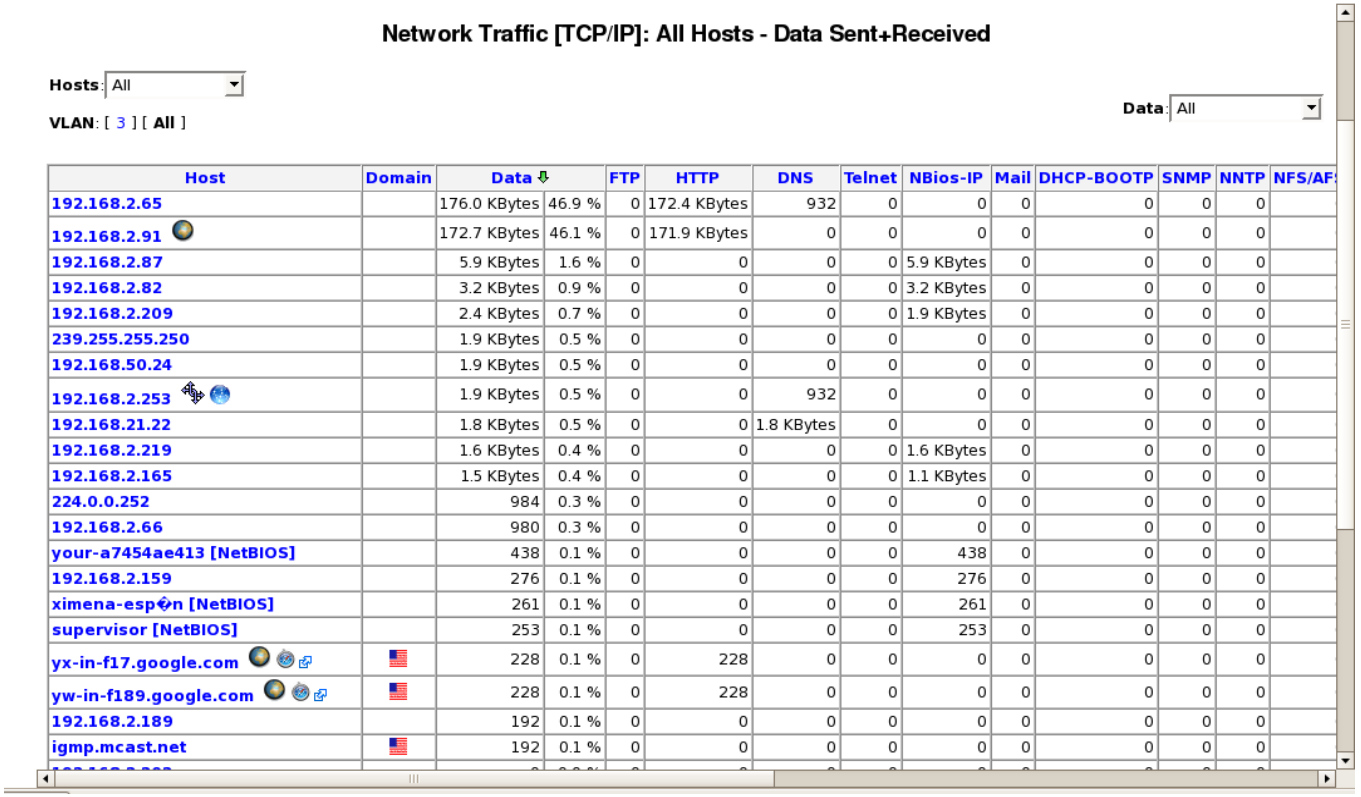
NAGIOS

- Alarmas visuales y por e-mail



NTOP

- Información detallada de tráfico de red:



Network Traffic [TCP/IP]: All Hosts - Data Sent+Received

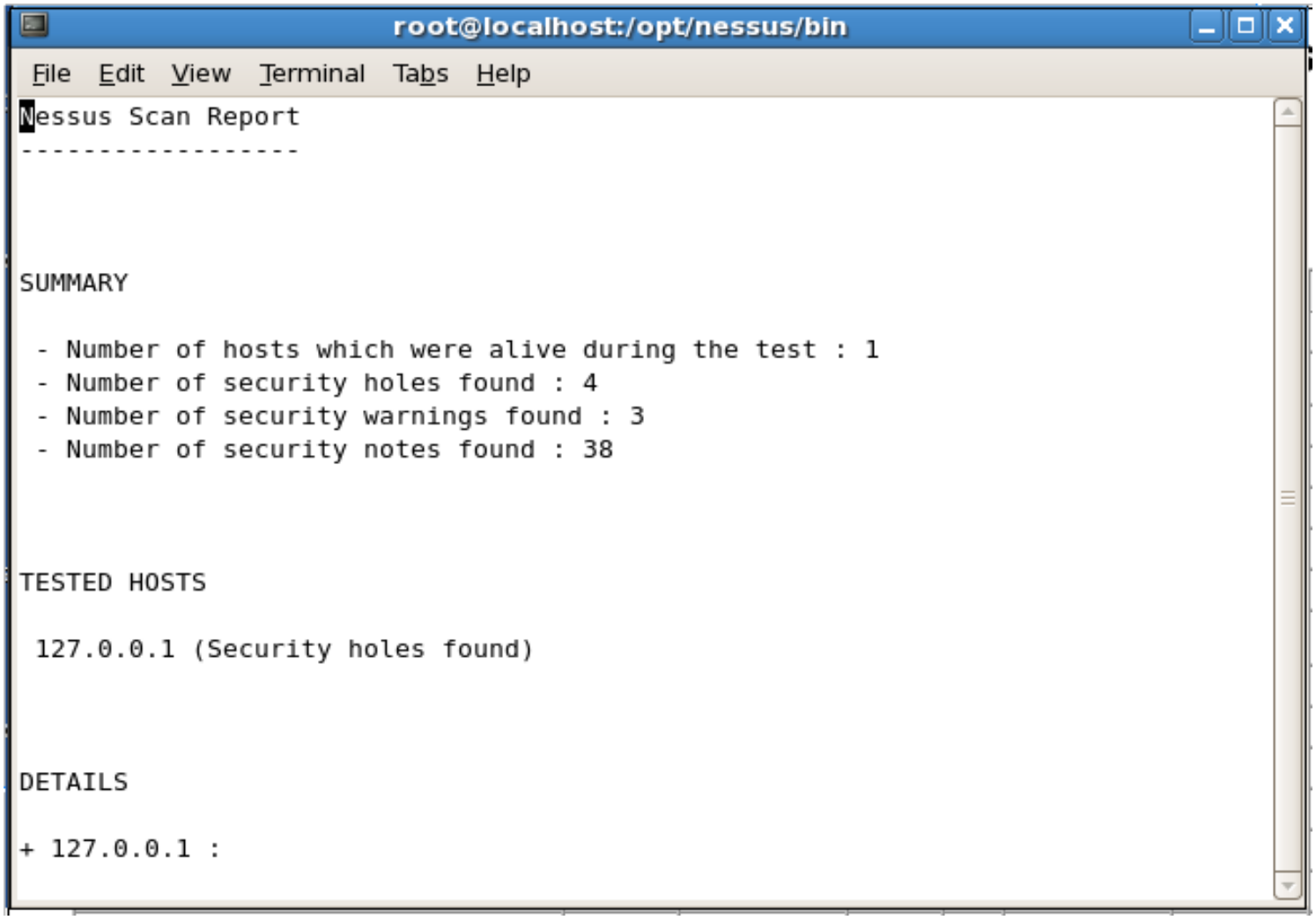
Hosts: All

VLAN: [3] [All]

Data: All

Host	Domain	Data		FTP	HTTP	DNS	Telnet	NBios-IP	Mail	DHCP-BOOTP	SNMP	NNTP	NFS/AF
192.168.2.65		176.0 KBytes	46.9 %	0	172.4 KBytes	932	0	0	0	0	0	0	
192.168.2.91		172.7 KBytes	46.1 %	0	171.9 KBytes	0	0	0	0	0	0	0	
192.168.2.87		5.9 KBytes	1.6 %	0	0	0	0	5.9 KBytes	0	0	0	0	
192.168.2.82		3.2 KBytes	0.9 %	0	0	0	0	3.2 KBytes	0	0	0	0	
192.168.2.209		2.4 KBytes	0.7 %	0	0	0	0	1.9 KBytes	0	0	0	0	
239.255.255.250		1.9 KBytes	0.5 %	0	0	0	0	0	0	0	0	0	
192.168.50.24		1.9 KBytes	0.5 %	0	0	0	0	0	0	0	0	0	
192.168.2.253		1.9 KBytes	0.5 %	0	0	932	0	0	0	0	0	0	
192.168.21.22		1.8 KBytes	0.5 %	0	0	1.8 KBytes	0	0	0	0	0	0	
192.168.2.219		1.6 KBytes	0.4 %	0	0	0	0	1.6 KBytes	0	0	0	0	
192.168.2.165		1.5 KBytes	0.4 %	0	0	0	0	1.1 KBytes	0	0	0	0	
224.0.0.252		984	0.3 %	0	0	0	0	0	0	0	0	0	
192.168.2.66		980	0.3 %	0	0	0	0	0	0	0	0	0	
your-a7454ae413 [NetBIOS]		438	0.1 %	0	0	0	0	438	0	0	0	0	
192.168.2.159		276	0.1 %	0	0	0	0	276	0	0	0	0	
ximena-esp&#65533;n [NetBIOS]		261	0.1 %	0	0	0	0	261	0	0	0	0	
supervisor [NetBIOS]		253	0.1 %	0	0	0	0	253	0	0	0	0	
yx-in-f17.google.com		228	0.1 %	0	228	0	0	0	0	0	0	0	
yw-in-f189.google.com		228	0.1 %	0	228	0	0	0	0	0	0	0	
192.168.2.189		192	0.1 %	0	0	0	0	0	0	0	0	0	
igmp.mcast.net		192	0.1 %	0	0	0	0	0	0	0	0	0	

NESSUS

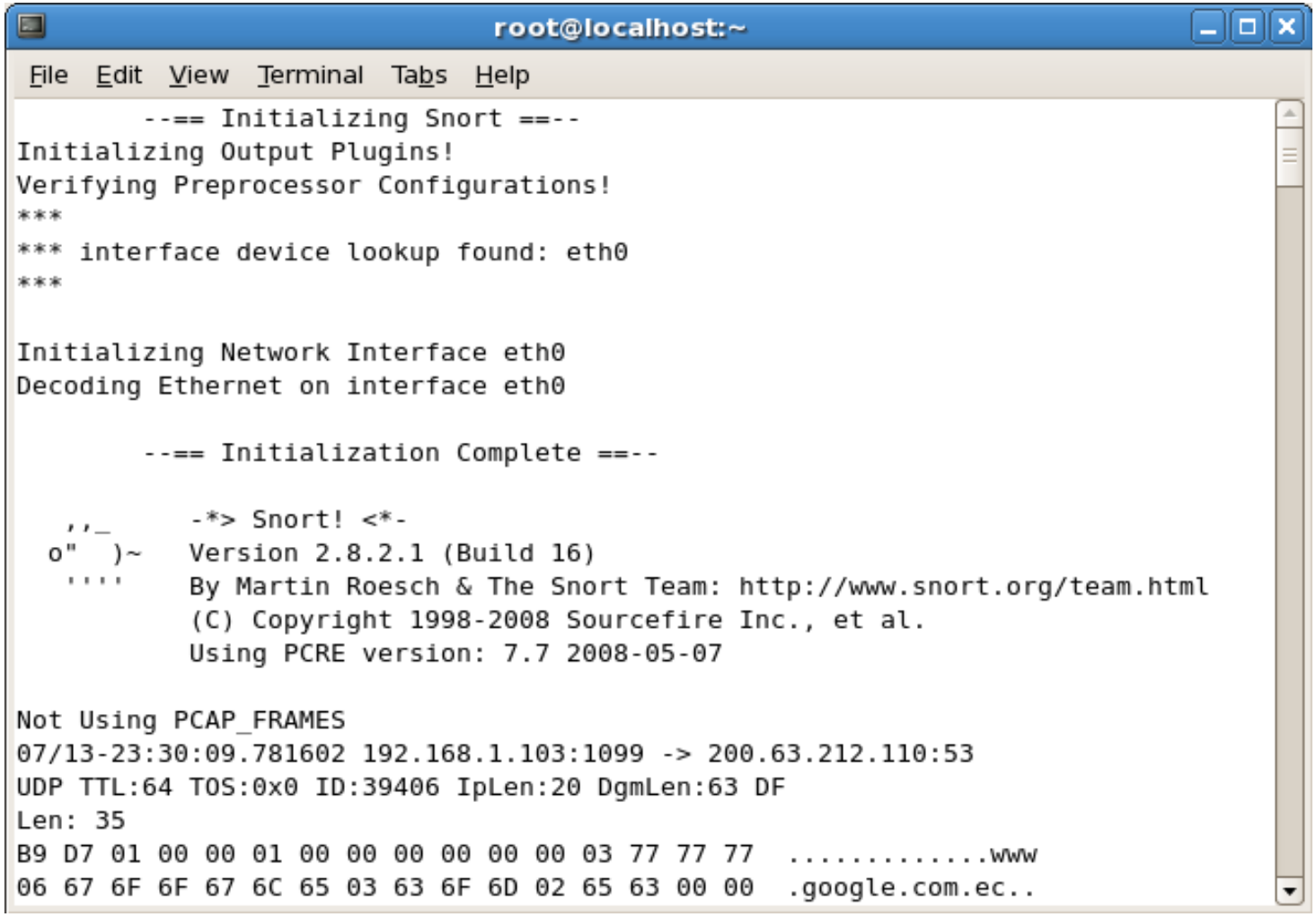

root@localhost:/opt/nessus/bin
File Edit View Terminal Tabs Help
Nessus Scan Report
SUMMARY
- Number of hosts which were alive during the test : 1
- Number of security holes found : 4
- Number of security warnings found : 3
- Number of security notes found : 38
TESTED HOSTS
127.0.0.1 (Security holes found)
DETAILS
+ 127.0.0.1 :

SERVIDOR IPS

- Sistema de prevención de intrusos.
- Snort-Inline

- Detecta y descarta paquetes sospechosos comparando con firmas establecidas.

SNORT


root@localhost:~
File Edit View Terminal Tabs Help
--== Initializing Snort ==--
Initializing Output Plugins!
Verifying Preprocessor Configurations!
***
*** interface device lookup found: eth0
***
Initializing Network Interface eth0
Decoding Ethernet on interface eth0
--== Initialization Complete ==--
-*> Snort! <*-
Version 2.8.2.1 (Build 16)
By Martin Roesch & The Snort Team: http://www.snort.org/team.html
(C) Copyright 1998-2008 Sourcefire Inc., et al.
Using PCRE version: 7.7 2008-05-07
Not Using PCAP_FRAMES
07/13-23:30:09.781602 192.168.1.103:1099 -> 200.63.212.110:53
UDP TTL:64 TOS:0x0 ID:39406 IpLen:20 DgmLen:63 DF
Len: 35
B9 D7 01 00 00 01 00 00 00 00 00 00 03 77 77 77 .....www
06 67 6F 6F 67 6C 65 03 63 6F 6D 02 65 63 00 00 .google.com.ec..

INTEGRACIÓN

Snort	
Snort database and path configuration	
snort_path	/etc/snort/
snort_rules_path	/etc/snort/rules/
snort_type	mysql
snort_base	snort
snort_user	root
snort_pass	••••••••••
snort_host	http://192.168.2.66
snort_port	3306

Links	
Links to other applications	
ossim_link	/ossim/
ntop_link	http://192.168.2.65:3000
nagios_link	http://192.168.2.65/nagios
use_munin	Yes
munin_link	/munin/

Nessus	
Nessus client configuration	
nessus_user	ossim
nessus_pass	••••••••••
nessus_host	http://192.168.2.65
nessus_port	1241
nessus_path	/usr/bin/nessus
nessus_rpt_path	/usr/share/ossim/www/vulnmeter/
nessusrc_path	/usr/share/ossim/www/vulnmeter/
nessus_distributed	Yes
vulnerability_incident_threshold	0

ESQUEMA DE MONITOREO

- NTOP:
  - Monitorear las sesiones TCP activas diariamente en la hora pico del uso de red.
  - Monitorear la matriz de conexión cuando se detecte que la red tiene tiempos de respuesta altos.
  - Monitorear la matriz de tiempos semanalmente para verificar que el uso de la red sea mínimo fuera del horario laborable.

ESQUEMA DE MONITOREO

- NAGIOS:
  - Monitorear diariamente en busca de alarmas tipo AMARILLAS y NARANJAS**.**
  - Resolver las alertas NARANJAS antes de la amarillas.
  - Resolver las alertas AMARILLAS para asegurar un monitoreo óptimo.
  - Reporte de disponibilidad mensualmente del servidor CAYAMBE para verificar que el porcentaje no sea menor al 90%.

ESQUEMA DE MONITOREO

- NESSUS:
  - Escanear mensualmente, fuera de horario laborable, todos los hosts de la red del Campus Miguel de Cervantes.
  - Actualizar los sistemas operativos de los hosts con mayor número de vulnerabilidades.
  - Actualizar semanalmente los plugins de Nessus para tenerlos actualizados y poder detectar las vulnerabilidades mas recientes.

CONCLUSIONES

- El análisis de riesgos evidenció varias fallas de seguridad, en el aspecto físico, de control de acceso, personal, manejo de incidentes, control de cambios, documentación y monitoreo.
- El riesgo más alto que existe en los sistemas IT del Campus es la alta temperatura a la que están sometidos los equipos.
- Se mejoró el esquema de seguridad informática en el Campus combinando la seguridad activa mediante servidores dedicados a detectar posibles ataques y vulnerabilidades, con políticas de seguridad que marcan pautas para comunicar los incidentes de seguridad y tomar acciones inmediatas.

CONCLUSIONES

- Se aumentó el porcentaje de disponibilidad de la red de datos implementado servidores de monitoreo y un plan de contingencia para restaurar las operaciones en el menor tiempo posible en caso de desastres naturales.
- La seguridad de los sistemas informáticos del Campus pasaron de un ámbito reactivo, sin ninguna planificación en caso de incidentes, al ámbito proactivo resolviendo posibles amenazas antes de que ocurran.
- Se redujeron los tiempos de notificación, análisis y detección de fallas cuando se produce un incidente en la red debido a problemas de hardware o servicios de red gracias a las herramientas gráficas de monitoreo de tráfico y disponibilidad.
- Se aumentó el nivel de reacción ante cualquier falla de los componentes de red mediante las notificaciones por correo electrónico al administrador de red.

RECOMENDACIONES

- Se recomienda implementar el esquema de seguridad perimetral propuesto para mejorar la organización y seguridad en la red del Campus.
- Es importante realizar pruebas de penetración en la red del Campus para tener un mejor conocimiento de los ataques que se perfilen como muy probables para la red.
- Es muy importante implementar las medidas de seguridad seleccionadas en este documento.
- Se recomienda implementar un sistema de control de temperatura ya que la probabilidad de que ocurra un fallo en algún equipo de la red por éste motivo es alta.

RECOMENDACIONES

- Es recomendable empezar a utilizar el manual de políticas de seguridad desarrollado en éste trabajo de investigación.
- Se recomienda implementar un SMTP Gateway en el segmento de red DMZ para adicionar seguridad en caso de implementar un servidor de correo electrónico en la red del Campus.
- Se recomienda segmentar la red de datos por facultad para mejorar el tráfico de red en caso de que el Campus de Guapulo se traslade al Campus de Carcelén.
- Es muy importante destinar más recursos por parte de las autoridades de la Institución para optimizar y asegurar los activos críticos del Campus para minimizar el nivel de afectación al negocio en caso de incidentes.
- Es altamente recomendable desarrollar documentación sobre configuraciones y control de cambios de la red para evitar fallas por el desconocimiento de información sobre ésta.

PREGUNTAS