EVALUACIÓN DE SERVICIOS LINUX PARA LA EMPRESA METROLINEA S.A

OSCAR ALBERTO CASELLES ALVAREZ

BABINGTON LEONARDO ARENAS CALDERÓN

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2011

EVALUACIÓN DE SERVICIOS LINUX PARA LA EMPRESA METROLINEA S.A

OSCAR ALBERTO CASELLES ÁLVAREZ

BABINGTON LEONARDO ARENAS CALDERÓN

Trabajo de grado presentado como requisito
para optar al Título de Especialista en Telecomunicaciones

Directora
AMPARO JAIMES CABALLERO
Especialista en Telecomunicaciones

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2011

AGRADECIMIENTOS

A nuestra directora de proyecto ingeniera Amparo Caballero, cuya colaboración y orientación permitió la realización del presente trabajo.

A nuestros profesores de la especialización cuyo tiempo y conocimiento permitió ampliar nuestros horizontes.

DEDICATORIA

“Para Paula Andrea Vergara Herrera, cuya fé permitió creer en mí”

- O. Caselles

“A Dios, mis padres y mis maestros; quienes siempre me indicaron el camino”

- L. Arenas

TABLA DE CONTENIDO

INTRODUCCIÓN 15

PLANTEAMIENTO DEL PROBLEMA 16

JUSTIFICACIÓN DEL PROBLEMA 18

OBJETIVO GENERAL 19

OBJETIVOS ESPECÍFICOS 19

1. SISTEMA OPERATIVO LINUX 20

1.1. RESEÑA HISTÓRICA 20

1.2. DISTRIBUCIONES LINUX 21

1.3. ESTRUCTURA DE DIRECTORIOS LINUX (AMBIENTE OPERATIVO) 26

2. SERVICIOS LINUX 32

2.1. PROCESOS O DEMONIOS (DAEMONS) 32

2.2. ESTRUCTURA DE SERVICIOS (MODELO CLIENTE-SERVIDOR) 34

2.2.1. VENTAJAS Y DESVENTAJAS DEL MODELO CLIENTE-SERVIDOR 34

2.3. PROTOCOLOS DE INTERNET EN LINUX 36

2.3.1. PROTOCOLO HTTP 36

2.3.2. PROTOCOLO FTP 39

2.3.3. PROTOCOLO SMTP 42

2.3.4. PROTOCOLO SSH 44

2.4. HERRAMIENTAS DE CONFIGURACION DE SERVICIOS LINUX 47

2.4.1. **EJEMPLO DE CONFIGURACION DE UN SERVICIO LINUX** 48

3. **INSTALACIÓN DE LINUX** 56

4. ANÁLISIS DE LA RED DE METROLINEA S.A 64

4.1. TOPOGRAFÍA DE LA RED METROLINEA 64

4.2. CONCLUSIONES DEL ANÁLISIS DE LA RED METROLINEA 69

4.3. EMPLEO DE SERVICIOS LINUX PARA CORREGIR FACTORES DE

RIESGOS ENCONTRADOS EN RED DE METROLINEA S.A 71

5. EVALUACION DE SERVICIOS LINUX PARA METROLINEA 73

5.1. PROPUESTA DE SERVIDOR SQUID – WEB CACHE O PROXY 73

5.1.1. **LISTAS DE CONTROL ACL SQUID** 74

5.1.2. Tipos de ACL 75

5.1.3. Operador ACL – HTTP_ACCESS 78

5.2. PROPUESTA DE SERVIDOR SAMBA – SERVIDOR DE ARCHIVOS 81

6. INSTALACION Y CONFIGURACION DE SERVICIOS LINUX PARA METROLINEA 94

6.1. INSTALACION DE APLICATIVO WEBMIN 94

6.2. INSTALACION DE SERVIDOR SQUID 95

6.2.1. CONFIGURACION DEL SERVICIO SQUID 95

6.3. INSTALACION DE SERVIDOR SAMBA 109

6.3.1. CONFIGURACION DE SERVIDOR SAMBA 110

CONCLUSIONES..... 116

BIBLIOGRAFÍA..... 117

LISTA DE TABLAS

Tabla 1. Servicios Linux Internos y Externos **33**
Tabla 2. Esquema de Particionado Recomendado **60**
Tabla 3. Hosts de Metrolínea **65**
Tabla 4. Tipos de archivo **85**
Tabla 5. Tipos de Permisos **85**
Tabla 6. Grupos de Permisos **85**
Tabla 7. Permisos en Notación Octal **88**
Tabla 8 Valores usuales de la variable **92**
Tabla 9. Archivos Configuracion Squidguard **104**
Tabla 10. Usuarios Windows **110**
Tabla 11. Comando crear usuario y grupos **111**

LISTA DE FIGURAS

Figura 1. Kernel Linux 22
Figura 2. Jerarquía de directorios Unix/Linux 27
Figura 3. Estructura Cliente-Servidor 32
Figura 4. Comunicación Navegador/Servidor 37
Figura 5. Ejemplo de solicitud HTTP 39
Figura 6. Conexión Cliente/Servidor 40
Figura 7. Conexión FTP Servidor/Servidor 41
Figura 8. Conexión SSH Cliente/Servidor 47
Figura 9. Proceso de Configuración SSH 49
Figura 10. Parámetros Modificados en Nano 51
Figura 11. Conexión SSH 51
Figura 12. Ingreso a Webmin 52
Figura 13. Control de opciones Webmin para SSH 53
Figura 14. Opciones de configuración 54
Figura 15. Conexión SSH 54
Figura 16. Menú de Instalación Ubuntu Server 57
Figura 17. Preguntas del Instalador Ubuntu 57
Figura 18. Nombre Máquina y Reloj 58
Figura 19. Particionado de disco 59
Figura 20. Configuración de usuarios y contraseña 61
Figura 21. Cifrado de partición /home 62
Figura 22. Arranque y Fin de Instalación 63
Figura 23. Login de entrada Ubuntu 63
Figura 24. D-Link DIR-300 Wireless Access Point 67
Figura 25. Servidor Web Metrolínea 68
Figura 26. Topografía Red Metrolínea 69
Figura 27. ACL Squid 75
Figura 28. Servidor Samba Linux 82
Figura 29. Permisos de un Archivo 84
Figura 30. Opciones configuración Squid 97

Figura 31. Configuración ACL Permitidos 100
Figura 32. Agregar Clientes Permitidos 100
Figura 33. Restricciones del Proxy 101
Figura 34. Configuración Proxy Windows 102
Figura 35. Conexiones a Internet (Via Proxy) 102
Figura 36. Archivo Access.log 103
Figura 37. Listado de /var/lib/squidguard/db 106
Figura 38. Estructura Archivo squidGuard.conf 108
Figura 39. Log /var/log/squid/squidGuard.log 108
Figura 40. Login/Password acceso a share samba 113
Figura 41. Configuracion /etc/samba/smb.conf 113
Figura 42. Carpetas Linux Visibles en Windows (Via Samba) 115
Figura 43. Usuario de Grupo Jurídica - Acceso Denegado 115

RESUMEN EN ESPAÑOL

TITULO: EVALUACIÓN DE SERVICIOS LINUX PARA LA EMPRESA METROLINEA S.A[*]

AUTORES: CASELLES ÁLVAREZ, Oscar Alberto. [**]
ARENAS, Babington Leonardo

PALABRAS CLAVES: Linux, Servicios, Squid, Redes, Ubuntu, Servidores, Samba, Proxy.

DESCRIPCION:

Mediante el análisis a la red de Metrolínea S.A se establecerá el estado actual de la infraestructura de comunicaciones, en aras de proponer servicios linux para atender problemas identificables en dicho análisis. El empleo de herramientas informáticas permitirá evaluar características importantes, con el ánimo de orientar a los administradores de red en la criticidad de mantener una plataforma operativa y confiable; contando con los mecanismos y políticas para el control y regulación de la infraestructura e información.

El trabajo realizado consiste en la documentación, y recopilación de información de servicios basados en sistemas operativos Linux, convirtiéndose en una guía o manual para el personal técnico encargado de la red de área local de Metrolínea S.A. Es así, como el documento final se plantea como propuesta para Metrolínea S.A, en aras de incentivar la necesidad de invertir en tecnologías que permitan generar buenas prácticas en el uso de la información y la protección de la información de la empresa.

Es así, como se documentara los procedimientos que permitan el establecimiento de servicios probados y ensayados, para atender la criticidad de control y protección de la información, demostrando ser alternativas económicamente viables para su implementación. Concluyendo, que linux se presenta como una opción real y oportuna para atender las presentes y futuras necesidades de la red de Metrolínea S.A

---

[*] Monografía de Grado

[**] Facultad de Ingenierías Físico – Mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Director Amparo Jaimes Caballero

RESUMEN EN INGLES

TITTLE: EVALUATION OF LINUX SERVICES FOR METROLINEA[*]

AUTHORS: CASELLES ALVAREZ, Oscar Alberto. [**]
ARENAS, Babington Leonardo

KEY WORDS: Linux, Services, Squid, Networks, Ubuntu, Servers, Samba, Proxy

DESCRIPTION:
Conducting an analysis to Metrolinea's local area network, will reveal the actual conditions of its communications infrastructure, thus proposing linux as a method to fix those identifiable problems. The use of computational tools will allow the assessment of important characteristics, with the purpose of raising system administrator's awareness of having a functional and well thought infrastructure; to guarantee control and security of the company's data resources.

The assignment consists in the documentation and recompilation of information base on linux operative system, thus becoming a manual or guide for the technical personnel in charge of Metrolínea's local area network. This is how the final documentation of this assignment is considered as a proposal for Metrolínea, in the hope of promoting the investment in technologies that allow the good use of information, and protection of company's information.

Execution procedures will be documented, with the objective of attending known deficiencies in the control and protection of information. Linux will be presented as a method to mitigate known deficiencies in the company's network resources, while remaining a cost effective solution for implementation. Thus, linux will produce a suitable environment to experiment and test solutions; concluding that linux is indeed a viable solution for Metrolínea's actual and future needs.

---

[*] Monografía de Grado

[**] Facultad de Ingenierías Físico – Mecánicas. Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones. Director Amparo Jaimes Caballero

INTRODUCCIÓN

El establecimiento de soluciones para afrontar los problemas de protección y manejo de los recursos informáticos es de vital importancia dentro del entorno administrativo de toda empresa. La correcta utilización de los recursos de información, como lo es internet, al igual que la preservación de la información representan problemas latentes que debe ser atendido con la mayor prontitud posible.

Por consiguiente, tomando como objeto de estudio la empresa Metrolínea S.A se elaborara el estudio necesario para evaluar sus características y capacidades de infraestructura tecnológica. A fin brindar un análisis de su condición actual, en aras de proponer servicios Linux que permitan mejorar el aprovechamiento de servicios como la internet, y protección de información de esta entidad.

Los esfuerzos del presente proyecto permitirán el establecimiento de alternativas libres, como plataforma de desarrollo para el establecimiento de mecanismo, políticas y herramientas para el crecimiento tecnológico de Metrolínea S.A. Inversión que se traduce en la garantía que el know-how permanecerá dentro de la misma y que se honrara los principios de honestidad y transparencia de esta empresa.

La propuesta permitirá mantener la balanza económica de la empresa, mientras se minimizan los riesgos subyacentes al no contar con mecanismos de control y contingencias. Y a su vez, se pretenderá generar conciencia en la importancia de invertir en propuestas y elementos tecnológicos para la empresa Metrolínea S.A

PLANTEAMIENTO DEL PROBLEMA

El reciente origen de Metrolínea S.A como empresa, sugiere el típico paradigma de brindar poca o nada de importancia al factor de infraestructura tecnológica. Se puede decir, que mientras cada terminal pueda acceder a internet, imprimir y redactar documentos, no se declara importante o prioridad dentro de la organización. La premisa anterior sugiere desde luego una inversión casi nula en el presupuesto para el departamento de informática de la entidad, y por consiguiente acarreando futuros problemas en la integridad, seguridad y disponibilidad de la información de esta empresa.

La situación anterior, puede no presentarse como un riesgo inminente, sin embargo a medida que se incorporan más terminales, personal y aplicaciones; se convertirá en un factor de riesgo que evite las operaciones regulares de la entidad. Por consiguiente el establecimiento de políticas y mecanismos que permitan establecer controles y contingencias se hace fundamental en el desarrollo de la empresa.

Basados en la distribución espacial y cantidad de recursos tecnológicos con los que cuenta Metrolínea S.A se identifican dos (2) situaciones específicas, y que serán el foco de atención del presente proyecto. La primera, corresponde a la falta de controles de contenido para las terminales de la entidad; situación que facilita la fuga de trabajo por aquellos funcionarios que no hacen uso correcto de internet. La segunda, centralización y protección de la información; esta corresponde a que la gran mayoría de terminales dentro de la red funcionan aislados y no se cuenta con un repositorio principal de la información con la cual trabajan los empleados. Situación que genera la pérdida de información por motivos de monopolización por parte de funcionarios o por fallas electrónicas en los terminales.

Linux, al ser una alternativa abierta y de bajo costo, comparada con aquellas soluciones comerciales o propietarias, se plantea como una propuesta favorable a la hora de establecer servicios que permitan la administración y gestión de una red de área local. Especialmente, cuando se puede experimentar antes de invertir en soluciones tecnológicas.

JUSTIFICACIÓN DEL PROBLEMA

El presente trabajo de grado tiene como objetivo profundizar los temas adquiridos en las materias cursadas en la Especialización de Telecomunicaciones de la Universidad Industrial de Santander. Permitiendo a los autores colocar en práctica el conjunto de conocimientos que permitan afrontar una situación real dentro del ambiente tecnológico de una empresa determinada.

El empleo de metodologías de investigación, uso de conceptos y empleo de herramientas tecnológicas, hacen del presente trabajo un ambiente favorable para certificar y avalar la satisfactoria terminación del programa académico de la especialización. Permitiendo a la empresa objeto del presente trabajo beneficiarse de los resultados y conclusiones dispuestas en el presente informe.

OBJETIVO GENERAL

Evaluación de servicios Linux que permitan el aprovechamiento de los recursos informáticos encontrados en la red de Metrolínea S.A.

OBJETIVOS ESPECÍFICOS

- Documentar la topología de la red, permitiendo identificar las especificaciones de cada dispositivo involucrado y su criticidad dentro de la plataforma tecnológica de la entidad.

- Identificación de dos (2) factores que puedan ser atendidos o solucionados, por medio del empleo de soluciones informáticas; principalmente por medio del empleo de herramientas libres o Open Source.

- Registrar los procedimientos de instalación y configuración de servicios Linux para afrontar los factores identificados.

- Redactar el informe final del proyecto, en cumplimiento de las normas de trabajos escritos y demás requisitos de calidad académica.

1. SISTEMA OPERATIVO LINUX

1.1. RESEÑA HISTÓRICA[1]

Este sistema operativo nació en abril de 1991 como el pasatiempo de un estudiante finlandés de 21 años, llamado Linus Torvals, quien cursaba segundo año de Ciencias de los Computadores en la universidad de Helsinki en Finlandia. Este joven empezó a crear un programa con la intención inicial de producir una versión más sencilla del sistema operativo Unix, que pudiera utilizarse en computadores personales y que brindara más estabilidad y confiabilidad que los actuales sistemas DOS y Windows. (Unix es un sistema operativo para servidores y máquinas de alto rendimiento).

En enero de 1992 Linux tenía cerca de 100 usuarios y varios de ellos ya participaban en su desarrollo con mejoras y correcciones que enviaban a través de Internet ya que su código fuente ha permanecido abierto a sus seguidores desde un comienzo. Fue entonces cuando lanzó su primera versión, la 0.12 la cual incluía partes desarrolladas por otros programadores.

En 1994 se hizo pública la primera versión completa del sistema operativo: la 1.0. Esta versión ofreció soporte para redes y varias utilidades.

En 1998, los principales fabricantes de software como Corel, IBM, Netscape, Oracle, HP y Dell, anunciaron versiones para Linux de sus productos. Este respaldo fue clave para la consolidación de este sistema operativo que dominaba cerca del 17% del mercado con 7.5 millones de usuarios; de ellos unos 10.000 participaban en su desarrollo y optimización.

En la actualidad, programadores de todo el mundo participan en su desarrollo. Está siendo utilizado ampliamente en servidores de Internet, lo utilizan

Universidades alrededor del todo el mundo para sus redes y sus clases, en la industria como software de apoyo a su maquinaria, en cadenas de supermercados, estaciones de servicio y muchas instituciones del gobierno y militares en varios países. Obviamente, también es utilizado por miles de usuarios en sus computadores personales. El apoyo más grande, sin duda, ha sido Internet ya que a través de ella se ha podido demostrar que se puede crear un sistema operativo para todos los usuarios sin la necesidad de fines lucrativos.

Un ejemplo de la popularidad que ha alcanzado el sistema y la confianza que se puede depositar en él, es que incluso la NASA ha creído en su seguridad y eficacia utilizándolo en misiones espaciales y control de experimentos.

Torvals distribuyó Linux bajo un tipo de licencia llamada GPL (Licencia Pública General), que permite a cualquier persona bajar, usar y modificar el software sin ningún costo; la única condición que ha impuesto su creador, es que los cambios o mejoras que una persona o compañía realicen también deben ser públicos, es decir que debe quedar su código fuente disponible, de la misma forma que está disponible el código de Linux y que además no tienen permiso de realizar restricciones con respecto a la utilización de este programa modificado.

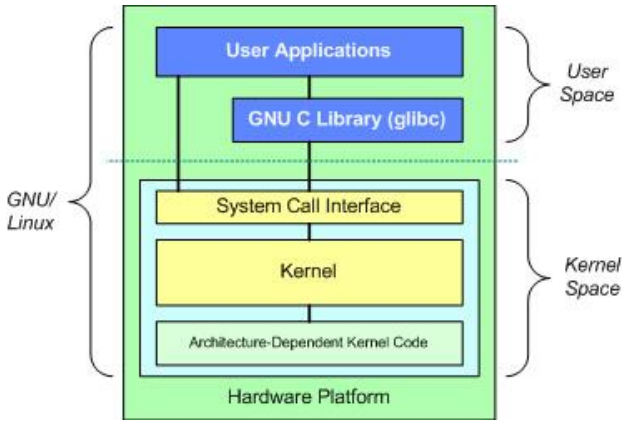
Este tipo de licencia llamada también de código abierto (Open Source), fue creada para mantener la libertad del software y evitar que alguien quiera apropiarse de la autoría intelectual de un determinado programa. La licencia advierte que el software debe ser gratuito y que el paquete final, ya modificado, también debe ser gratuito.

1.2. DISTRIBUCIONES LINUX

A diferencia del software comercial y sin duda a su naturaleza abierta, Linux puede conseguirse en versiones diferentes, dependiendo de las necesidades y/o orientación particular de una empresa o grupo de desarrolladores. Cada versión de Linux, corresponde a lo que se conoce como una distribución.

Una distribución Linux corresponde a una empresa o grupo de programadores que toman el kernel[2] Linux y juntan aplicaciones diversas para ofrecer un ambiente de aplicativos básicos con los cuales el usuario pueda administrar los recursos de un computador o máquina. Usualmente a lo anterior lo acompaña algún tipo de herramienta para la gestión de instalación de aplicaciones; bien sea compilando código fuente o instalando paquetes binarios para la administración automática de dependencias de librerías. Véase la siguiente figura:

Figura 1. Kernel Linux



Fuente: Referencia 2.

Las distribuciones Linux no tienen costo, a excepción si se desea obtener el soporte técnico para ellas. Es importante acotar que todos los códigos fuentes que conforman al sistema operativo y sus aplicativos pueden ser obtenidos sin ningún costo gracias a la existencia de licencias abiertas, tómese como ejemplo la GPL (General Public Licence).

Mientras el soporte técnico tiene un costo, este es de carácter opcional dentro de las distribuciones. Muchas de las grandes empresas han logrado obtener una utilidad de la naturaleza abierta de Linux, ofreciendo este tipo de servicios

técnicos; soporte que es crucial y bastante apreciado cuando se trabaja con redes de infraestructura y sistemas de producción.

Aquí comienza el primer atractivo de Linux, y esa es la libertad de elección. Para un determinado individuo o empresa la libertad de elección le permite ajustar su modelo económico a una solución en particular; en detalle le permite experimentar antes de invertir. Sin duda una característica bastante difícil de ignorar cuando se busca invertir en tecnología.

Sin duda, elegir la mejor distribución se vuelve una tarea agobiante, ya que cada una de ellas trae consigo un cumulo de beneficios y de inconvenientes. Por consiguiente cual elegir? Cada distribución busca plasmar en un sistema coherente y estructurado los intereses particulares del grupo de desarrolladores, tómese por ejemplo las siguientes descripciones a las distribuciones más conocidas:

- **DEBIAN LINUX[3]:** Es una comunidad conformada por desarrolladores y usuarios, que pretende crear y mantener un sistema operativo GNU basado en software libre precompilado y empaquetado, en un formato sencillo en múltiples arquitecturas de computador y en varios núcleos. Nace como una apuesta por separar en sus versiones el software libre del software no libre. El modelo de desarrollo del proyecto es ajeno a motivos empresariales o comerciales, siendo llevado adelante por los propios usuarios, aunque cuenta con el apoyo de varias empresas en forma de infraestructuras. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

Es ampliamente reconocida por su robustez, estabilidad y seguridad. Al no contar con una fecha regular para el lanzamiento de nuevas versiones, los desarrolladores y comunidad en general cuentan con el tiempo suficiente tiempo para depurar varios tipos de fallos. Sin duda una distribución bastante apropiada para ambientes de servidor de producción.

Igualmente, es reconocida por su gestor de paquetes conocido como APT, que permite la instalación de programas distribuidos en paquetes con extensión ".deb". El gestor regula de forma automática las dependencias necesarias. Esto significa que si un paquete en particular requiere de otro para poder funcionar, el gestor automáticamente instala cualquier librería o programa para que el paquete a ser instalado funcione.

- **FEDORA LINUX:** Considerada la distribución de prueba de Red Hat Enterprise Linux, que corresponde a la propuesta comercial de Linux para ambientes corporativos de servidores. Fedora es una distribución reconocida por su innovación, y orientación hacia el entorno corporativo. Solida en seguridad al incorporar SELinux como gestor de seguridad integral del sistema operativo. Al igual que otras distribuciones, cuenta con su propio gestor de paquetes conocido como YUM (Yellow Dog Update) aplicativo que permite mejorar la instalación de paquetes con extensión ".rpm", haciendo una labor inteligente de solución de conflictos para la instalación de paquetes o librerías.

  El esquema propuesto por la empresa Red Hat Inc. Hace interesante el uso de una distribución como esta, ya que mucho del desarrollo y realimentación generada por los usuarios que usan Fedora Linux, permite depurar e incorporar funcionalidades a la distribución empresarial Red Hat Enterprise Linux. Esto permite una transición bastante sencilla desde el ambiente de escritorio hacia el nicho de servidores. Ambiente de donde Red Hat Inc, obtiene mucho de sus recursos, ya que estos se lucran de la

venta de servicios de soporte técnico para la distribución Red Hat Enterprise.

- **UBUNTU LINUX:** Reconocida por ser una de la distribución con mayor penetración en el mercado, se ubica como la alternativa para reemplazar el uso de Microsoft Windows y Microsoft Office. Se encuentra encaminada hacia el usuario de escritorio, que desea sencillamente trabajar, en lugar de preocuparse por los detalles técnicos de Linux. Se puede decir, que brinda una experiencia más estilizada de Linux, en comparación a otras distribuciones.

  Sin embargo, al ser una distribución basada en Debian, la hace heredera directa de todos sus beneficios, como estabilidad y robustez, con la diferencia que esta cuenta con una frecuencia de lanzamiento de versiones de cada seis meses. Algo que el usuario de escritorio agradece, ya que todas las aplicaciones de la distribución se encuentran con la última versión.

  Ubuntu sin embargo, ha incursionado hacia el ambiente de servidores, por lo cual es una estrategia bastante astuta, ya que en el corto plazo le permitirá que esos usuarios de escritorio, que representan aproximadamente el 50%, hagan en algún momento la transición hacia el entorno corporativo de servidores; permitiéndose posicionarse como una solución integral para el ambiente corporativo.

Concluyendo, ninguna es mejor que la otra. Realmente, cada una atiende una necesidad específica. Sin embargo, tomando en cuenta que todas las distribuciones siguen un concepto y modelo similar; la transición de una distribución a otra, solo se basa en las herramientas o comandos particulares de cada una de ellas. La recomendación general de los expertos, es usar aquella con la que se sienta más cómodo o se tenga un mejor desenvolvimiento.

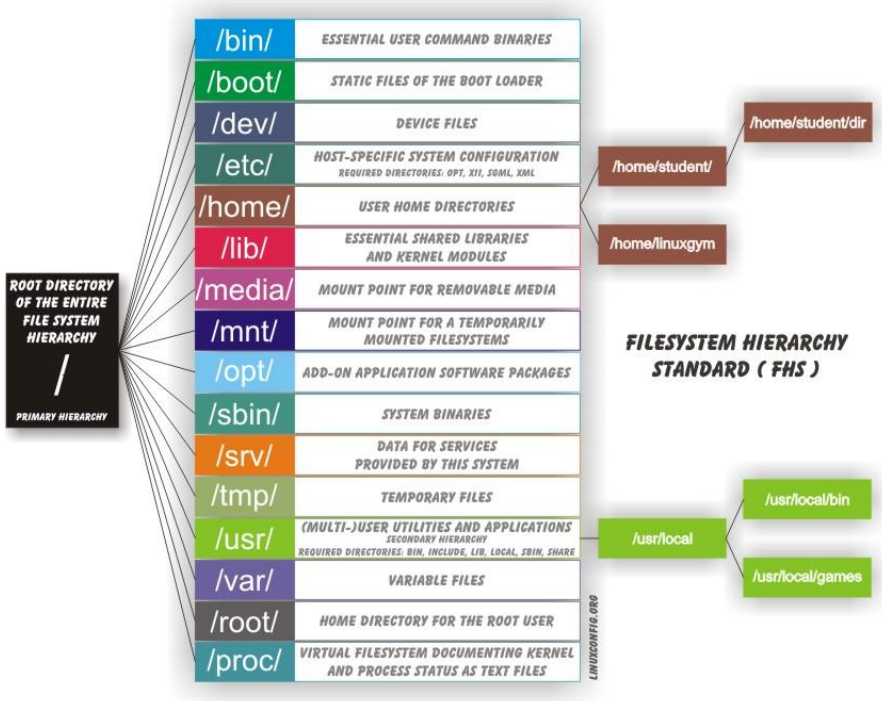
1.3. ESTRUCTURA DE DIRECTORIOS LINUX (AMBIENTE OPERATIVO)[4]

Los sistemas operativos basados en UNIX cuentan con una estructura definida de directorios que permite el establecimiento de áreas comunes para tareas específicas. En Linux este tipo de estructuras o jerarquías permite agrupar el sin numero de archivos en una estructura coherente para la administración de un sistema.

A diferencia de Windows, en Linux los archivos y carpetas se organizan de manera diferente, en Windows cuando se instala un programa se crea una carpeta donde estarán todos los archivos de programas, en Linux en cambio, la idea es agrupar en carpetas archivos con fines similares, es decir en un sector todos los ejecutables, en otro las librerías, en otro las configuraciones, etc.

La mayoría de los sistemas Unix tienen una distribución estándar de ficheros, de esta forma los recursos que el sistema posee y los archivos, serán fácilmente localizados. Esta distribución forma el árbol de directorios, el cual comienza con /, también conocido como directorio raíz. Directamente por debajo de este hay algunos subdirectorios muy importantes para el sistema: Estos son /bin, /etc, /dev y /usr, entre otros. Estos a su vez contienen otros directorios con archivos de configuración del sistema, programas, etc. Un ejemplo ocurre con el directorio /home que suele contener todos los archivos de configuración de programas para un usuario en particular. Dentro de la estructura /home/<usuario> se encuentran datos como posicionamiento de ventanas, preferencias, etc.

Figura 2. Jerarquía de directorios Unix/Linux



Fuente: http://www.cesarius.net/diagrama-de-jerarquia-de-carpetas-en-gnulinux/

A continuación se brindara una descripción corta de cada uno de los directorios claves dentro de un sistema operativo Linux.

- **/bin - Binarios de comandos esenciales de usuarios:** bin es la abreviación de binaries, o ejecutables. Es donde residen la mayoría de los programas esenciales del sistema. Use la orden ls -F /bin para listar los ficheros. Podrá ver algunas órdenes que reconocerá, como cp, ls y mv. Estos son los programas para estas órdenes. Cuando usa la orden cp está ejecutando el programa /bin/cp. Usando ls -F se verá que la mayoría (si no todos) los ficheros de /bin tienen un asterisco añadido al final de sus nombres. Esto indica que son ficheros ejecutables.

- **/boot - Archivos estáticos del cargador de arranque:** Este directorio contiene todo para arrancar excepto los archivos de configuración y el instalador de mapas. En su sentido más sencillo /boot es para cualquier cosa que se utiliza antes de que el kernel ejecute /sbin/init. Esto incluye sectores maestros de arranque (master boot sectors) guardados, archivos de mapeo de sectores y cualquier otra cosa que no es editada directamente a mano. Los programas necesarios para arreglar que el cargador de arranque sea capaz de arrancar un archivo (tal como el instalador de mapas [lilo]) estarán localizados en /sbin. Los archivos de configuración para cargadores de arranque podrían estar localizados en /etc.

- **/dev - Archivos de dispositivos:** Los archivos en /dev son conocidos como controladores de dispositivo (device drivers) son usados para acceder a los dispositivos del sistema y recursos, como discos duros, modems, memoria, etc. Por ejemplo, de la misma forma que puede leer datos de un fichero, puede leerla desde la entrada del ratón leyendo /dev/mouse.

- **/etc - Configuración del sistema local a la máquina:** Contiene una serie de ficheros de configuración del sistema. Estos incluyen /etc/passwd (la base de datos de usuarios), /etc/rc (guiones de inicialización del sistema), etc. La mayoría de archivos de configuración global para servicios o daemon se encuentran en este directorio.

- **/home - Directorios hogar de los usuarios (opcional):** Contiene los directorios "home" de los usuarios. Por ejemplo, /home/shrek es el directorio del usuario shrek. En un sistema recién instalado, no habrá ningún usuario en este directorio. En sistemas grandes (especialmente cuando los directorios /home son compartidos entre varias máquinas usando NFS) es útil subdividir los directorios hogar. La subdivisión puede ser llevada a cabo utilizando subdirectorios tales como /home/apoyo,

/home/huéspedes, /home/estudiantes, etc

- **/lib - Librerías compartidas y módulos de kernel esenciales:** Contiene las imágenes de las librerías compartidas. Estos ficheros contienen código que compartirán muchos programas. En lugar de que cada programa contenga una copia propia de las rutinas compartidas, estas son guardadas en un lugar común, en /lib. Esto hace que los programas ejecutables sean menores y reduce el espacio usado en disco.

- **/media - Medios extraíbles:** En esta carpeta se montan los diversos medios extraíbles (CD, DVD, ZIP, pendrives, etc.) y también particiones adicionales (particiones únicamente para datos, particiones de Windows, etc). Por ejemplo, un DVD puede estar montado en /media/cdrom0 o /media/cdrom1 (si hay dos unidades ópticas por ejemplo). Un pendrive en /media/pendrive, una partición de Windows en /media/ntfs.

- **/mnt - Punto de montaje para sistemas de archivos montados temporalmente:** Este directorio se ha provisto para que el administrador pueda montar temporalmente sistemas de archivos cuando lo necesite. El contenido de este directorio es un asunto local y no debe afectar la manera en la cual se ejecuta ningún programa.

- **/opt - Agregados de paquetes de software y aplicaciones:** Proporciona un área para almacenar habitualmente paquetes de software de una aplicación estática y amplia. Un paquete colocando archivos en el directorio /opt/ crea un directorio con el mismo nombre del paquete. Este directorio en su lugar guarda archivos, que de otra forma, estarían esparcidos por el sistema de archivos, dándole así al administrador del sistema una forma fácil de determinar el papel de cada archivo dentro de un paquete particular. Por ejemplo, si google-earth fuese el nombre de un paquete de software

particular localizado en el directorio /opt/, todos sus archivos podrían ser emplazados en directorios dentro de /opt/google-earth/, tales como /opt/google-earth/bin/ para binarios y /opt/google-earth/man/ para páginas de manual.

- **/sbin - Binarios del sistema:** Se usa para almacenar programas esenciales del sistema, que usará el administrador del sistema. Decidir qué cosa va en los directorios de /sbin es sencillo: Si un usuario necesitará ejecutarlo, debe de ir en otro lado. Si sólo será ejecutado por el administrador del sistema o por root como scripts de administración, entonces debe ir en /sbin (o en /usr/sbin o en /usr/local/sbin, si el archivo no es vital para la operación del sistema).

- **/tmp - Archivos temporales:** Muchos programas tienen la necesidad de generar cierta información temporal y guardarla en un fichero temporal. El lugar habitual para esos ficheros es en /tmp.

- **/usr - Utilidades y aplicaciones multiusuario:** Es un directorio muy importante. Contienen una serie de subdirectorios que contienen a su vez algunos de los más importantes y útiles programas y ficheros de configuración usados en el sistema. Se considera el área en donde la gran mayoría de aplicaciones se encuentran. Aplicaciones como ambientes gráficos como KDE, Gnome, Nautilus, Gimp y Openoffice. Aplicaciones adicionales que amplían la experiencia de un entorno informático.

- **/var – Directorio de variables temporales:** Contiene directorios que a menudo cambian su tamaño o tienden a crecer. El cache de varias aplicaciones se encuentra en este directorio. Al igual que algunos registros de sistema.

- **/var/log: Directorio de registros:** contiene varios ficheros de interés para el administrador del sistema, específicamente históricos del sistema, los cuales recogen errores o problemas con el sistema. Otros ficheros guardan las sesiones de presentación en el sistema, así como los intentos fallidos.

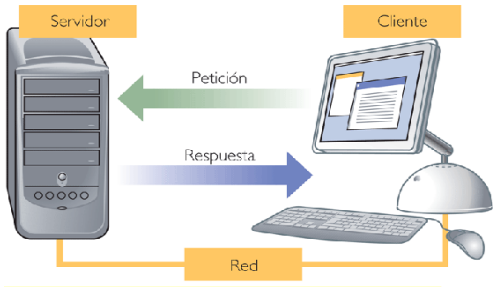
- **/root Directorio hogar de root (opcional):** El directorio / es tradicionalmente el directorio hogar del usuario root en los sistemas UNIX. /root se usa en muchos sistemas Linux y en algunos sistemas UNIX. El directorio hogar de la cuenta de el usuario root puede ser determinada por el desarrollador o por preferencias locales.

- **/proc Documentación del kernel y procesos:** Es un "sistema de ficheros virtual". Los ficheros que contiene realmente residen en memoria, no en un disco. Hacen referencia a varios procesos que corren en el sistema, y le permiten obtener información acerca de que programas y procesos están corriendo en un momento dado.

2. SERVICIOS LINUX

Los servicios Linux corresponden al conjunto de programas que se ejecutan en un sistema operativo Gnu/Linux con el propósito de ofrecer información a solicitudes especificas realizadas por programas denominados clientes. El procesamiento de las solicitudes es realizado por el programa servidor, quien realiza la mayoría del trabajo pesado, tómese por ejemplo la consulta a una página web. El programa servidor luego comunica el resultado de la solicitud, enviando una respuesta haciendo uso de la infraestructura de comunicaciones disponible.

Figura 3. Estructura Cliente-Servidor



Fuente: http://kalepedia.com

El principal propósito es el intercambio de información de una o varias solicitudes, dependiendo de la naturaleza del requerimiento; bien sea la consulta de un correo electrónico, búsqueda en bases de datos, o la consulta a una página web y cuya ventaja reside en la centralización de esta información en una máquina denominada “Servidor”.

2.1. PROCESOS O DEMONIOS (DAEMONS)

Los servicios en máquinas Unix/Linux se conocen como demonio, o daemon (de sus siglas en inglés Disk And Execution MONitor), es un tipo especial de proceso computacional que se ejecuta en segundo plano sin necesidad de ser controlado directamente por interacción del usuario, estos tipos de programas se ejecutan de

forma continua y son los encargados de escuchar y atender las solicitudes provenientes de aquellos aplicativos clientes.

Un servicio puede ser un daemon que maneja solicitudes bien seas externas y/o internas a la misma máquina. Esto es muy común en máquinas de sistema operativos derivados de Unix, ya que muchos de estos sistemas operativos se constituyen por aplicaciones particulares e independientes que se ejecutan para generar un Sistema General para el usuario. Comúnmente cada procesos tiene un número de puerto asignado. Tómese como ejemplo el puerto 80, que identifica generalmente al servicio de páginas web de un servidor.

La siguiente tabla numera algunos de los servicios que ofrecen servicios de forma interna y externa dentro de una máquina Linux:

Tabla 1. Servicios Linux Internos y Externos

INTERNOS	**EXTERNOS**
df: Reporta Uso del Sistema de Archivos del Sistema	httpd: Servidor de Páginas Web
dmesg: Imprime el buffer del kernel	squid: Cache de Páginas Web
Ls: listado archivos	ssh: Acceso Remoto Cifrado
Cron: Temporizador de procesos	Deluge: Atiende solicitudes de P2P

Fuente: Autores.

A lo anterior, dicha jerarquía u organización se le determina como el Modelo Cliente-Servidor. La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o

recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

2.2. ESTRUCTURA DE SERVICIOS (MODELO CLIENTE-SERVIDOR)

Los aplicativos clientes realizan generalmente las funciones de manejo de la interfaz de usuario, captura y validación de los datos de entrada y generación de consultas e informes sobre las bases de datos. Tómese como ejemplo el aplicativo de un navegador web; herramienta que permite al usuario la interacción necesaria para buscar la información requerida brindando una interfaz de usuario común. En ocasiones, el mismo aplicativo por medio de algún lenguaje de programación y de ejecución local (Ej: Javascript) valida los datos de entrada para la correspondiente solicitud.

Por otro lado, del modelo cliente/servidor el daemon httpd procesa la solicitud acorde a las especificaciones técnicas del protocolo http (Protocolo de Transferencia de Hipertexto) y brinda una respuesta estructurada a la solicitud requerida. En este proceso el daemon se encarga de gestión los periféricos compartidos, control de accesos concurrentes a bases de datos compartidas y el enlace de comunicaciones con otras redes de área local.

2.2.1. VENTAJAS Y DESVENTAJAS DEL MODELO CLIENTE-SERVIDOR[5]

Es claro que el esquema previamente mencionado hace parte de la gran mayoría de servicios que se encuentran internamente en empresas y en el internet. A lo cual se suma sus conocidas ventajas y desventajas, las cuales se pueden citar a continuación:

2.2.1.1. VENTAJAS

- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no Autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).

- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).

- Fácil mantenimiento: Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.

- Existen tecnologías, suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad del interfaz, y la facilidad de empleo.

2.2.1.2. DESVENTAJAS

- La congestión del tráfico ha sido siempre un problema en el paradigma de C/S. Cuando una gran cantidad de clientes envían peticiones simultaneas al mismo servidor, puede ser que cause muchos problemas para éste (a mayor número de clientes, más problemas para el servidor). Al contrario, en

las redes P2P como cada nodo en la red hace también de servidor, cuantos más nodos hay, mejor es el ancho de banda que se tiene.

- El paradigma de C/S clásico no tiene la robustez de una red P2P. Cuando un servidor está caído, las peticiones de los clientes no pueden ser satisfechas. En la mayor parte de redes P2P, los recursos están generalmente distribuidos en varios nodos de la red. Aunque algunos salgan o abandonen la descarga; otros pueden todavía acabar de descargar consiguiendo datos del resto de los nodos en la red.

- El software y el hardware de un servidor son generalmente muy determinantes. Un hardware regular de un computador personal puede no poder servir a cierta cantidad de clientes. Normalmente se necesita software y hardware específico, sobre todo en el lado del servidor, para satisfacer el trabajo. Por supuesto, esto aumentará el costo.

2.3. PROTOCOLOS DE INTERNET EN LINUX

Desde el origen del internet han existido un número considerable de servicios que han permitido el intercambio de información entre usuarios, todo gracias a la existencia de protocolos específicos para esta tarea.

2.3.1. PROTOCOLO HTTP[6]

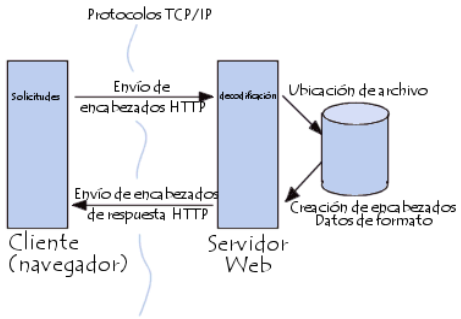
Desde 1990, el protocolo HTTP (Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet. La versión 0.9 sólo tenía la finalidad de transferir los datos a través de Internet (en particular páginas Web escritas en HTML). La versión 1.0 del protocolo (la más utilizada) permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME.

El propósito del protocolo HTTP es permitir la transferencia de archivos

(principalmente, en formato HTML). entre un navegador (el cliente) y un servidor web (denominado, entre otros, *httpd* en equipos UNIX) localizado mediante una cadena de caracteres denominada dirección URL.

Comunicación entre el navegador y el servidor: La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

Figura 4. Comunicación Navegador/Servidor



Fuente: http://es.kioskea.net/contents/internet/http.php3

- El navegador realiza una **solicitud HTTP**
- El servidor procesa la solicitud y después envía una **respuesta HTTP**

En realidad, la comunicación se realiza en más etapas si se considera el procesamiento de la solicitud en el servidor.
Solicitud HTTP: Una solicitud HTTP es un conjunto de líneas que el navegador envía al servidor. Incluye:

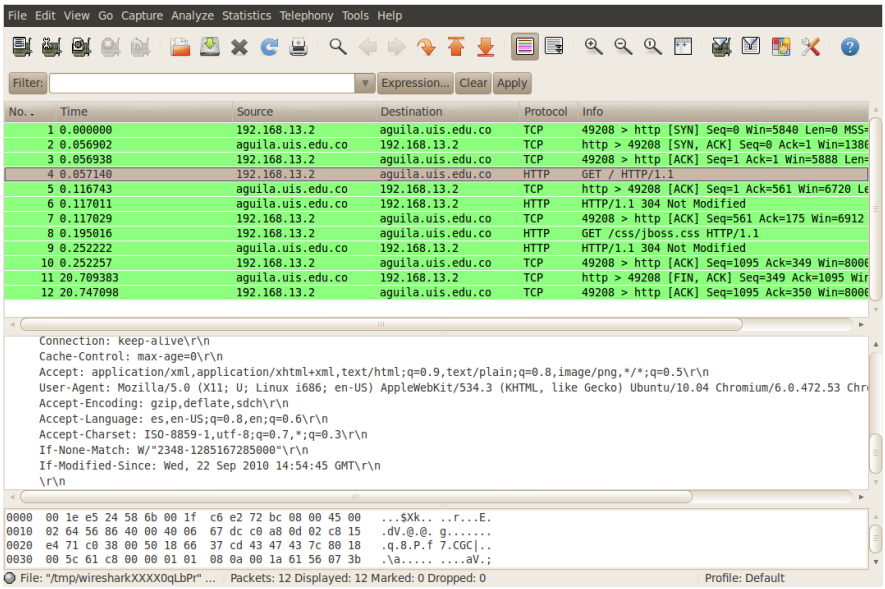
- **Una línea de solicitud**: es una línea que especifica el tipo de documento solicitado, el método que se aplicará y la versión del protocolo utilizada. La línea está formada por tres elementos que deben estar separados por un espacio:
    - el método
    - la dirección URL

    - la versión del protocolo utilizada por el cliente (por lo general, *HTTP/1.0*)
- **Los campos del encabezado de solicitud**: es un conjunto de líneas opcionales que permiten aportar información adicional sobre la solicitud y/o el cliente (navegador, sistema operativo, etc.). Cada una de estas líneas está formada por un nombre que describe el tipo de encabezado, seguido de dos puntos (:) y el valor del encabezado.
- **El cuerpo de la solicitud**: es un conjunto de líneas opcionales que deben estar separadas de las líneas precedentes por una línea en blanco y, por ejemplo, permiten que se envíen datos por un comando POST durante la transmisión de datos al servidor utilizando un formulario.

Por lo tanto, una solicitud HTTP posee la siguiente sintaxis (<*crlf*> significa retorno de carro y avance de línea):

MÉTODO VERSIÓN URL<*crlf*>
ENCABEZADO: Valor<*crlf*>
. . . ENCABEZADO: Valor<*crlf*>
Línea en blanco <*crlf*>
CUERPO DE LA SOLICITUD

Figura 5. Ejemplo de solicitud HTTP



Fuente: Autores.

2.3.2. PROTOCOLO FTP[7]

El protocolo FTP (*Protocolo de transferencia de archivos*) es, como su nombre lo indica, un protocolo para transferir archivos.

La implementación del FTP se remonta a 1971 cuando se desarrolló un sistema de transferencia de archivos (descrito en RFC141) entre equipos del Instituto Tecnológico de Massachusetts (*MIT, Massachusetts Institute of Technology*). Desde entonces, diversos documentos de RFC (petición de comentarios) han mejorado el protocolo básico, pero las innovaciones más importantes se llevaron a cabo en julio de 1973.

La función del protocolo FTP

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP.

El objetivo del protocolo FTP es:

- permitir que equipos remotos puedan compartir archivos
- permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor
- permitir una transferencia de datos eficaz

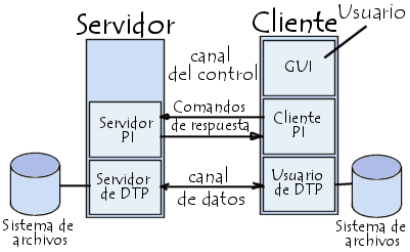
El modelo FTP

El protocolo FTP está incluido dentro del modelo cliente-servidor, es decir, un equipo envía órdenes (el cliente) y el otro espera solicitudes para llevar a cabo acciones (el servidor).

Durante una conexión FTP, se encuentran abiertos dos canales de transmisión:

- Un canal de comandos (canal de control) y un canal de datos.

Figura 6. Conexión Cliente/Servidor



Fuente: http://es.kioskea.net/contents/internet/ftp.php3

Por lo tanto, el cliente y el servidor cuentan con dos procesos que permiten la Administración de estos dos tipos de información:

- **DTP** (*Proceso de transferencia de datos*) es el proceso encargado de establecer la conexión y de administrar el canal de datos. El DTP del lado del servidor se denomina *SERVIDOR DE DTP* y el DTP del lado del cliente se denomina *USUARIO DE DTP*.
- **PI** (*Intérprete de protocolo*) interpreta el protocolo y permite que el DTP pueda ser controlado mediante los comandos recibidos a través del canal de control. Esto es diferente en el cliente y el servidor:
  - El SERVIDOR PI es responsable de escuchar los comandos que provienen de un USUARIO PI a través del canal de control en un

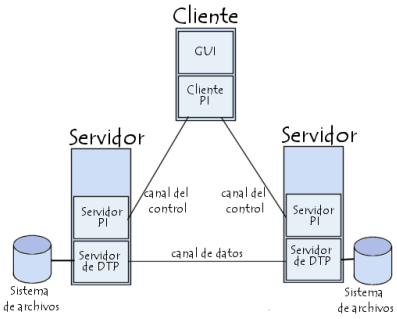
puerto de datos, de establecer la conexión para el canal de control, de recibir los comandos FTP del USUARIO PI a través de éste, de responderles y de ejecutar el SERVIDOR DE DTP.

- El USUARIO PI es responsable de establecer la conexión con el servidor FTP, de enviar los comandos FTP, de recibir respuestas del SERVIDOR PI y de controlar al USUARIO DE DTP, si fuera necesario.

Cuando un cliente FTP se conecta con un servidor FTP, el USUARIO PI inicia la conexión con el servidor de acuerdo con el protocolo Telnet. El cliente envía comandos FTP al servidor, el servidor los interpreta, ejecuta su DTP y después envía una respuesta estándar. Una vez que se establece la conexión, el servidor PI proporciona el puerto por el cual se enviarán los datos al Cliente DTP. El cliente DTP escucha el puerto especificado para los datos provenientes del servidor.

Es importante tener en cuenta que, debido a que los puertos de control y de datos son canales separados, es posible enviar comandos desde un equipo y recibir datos en otro. Entonces, por ejemplo, es posible transferir datos entre dos servidores FTP mediante el paso indirecto por un cliente para enviar instrucciones de control y la transferencia de información entre dos procesos del servidor conectados en el puerto correcto.

Figura 7. Conexión FTP Servidor/Servidor



Fuente: http://es.kioskea.net/contents/internet/ftp.php3

En esta configuración, el protocolo indica que los canales de control deben permanecer abiertos durante la transferencia de datos. De este modo, un servidor puede detener una transmisión si el canal de control es interrumpido durante la transmisión.

2.3.3. PROTOCOLO SMTP[8]

El **protocolo SMTP** (*Protocolo simple de transferencia de correo*) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

Éste es un protocolo que funciona en línea, encapsulado en una trama TCP/IP. El correo se envía directamente al servidor de correo del destinatario. El protocolo SMTP funciona con comandos de textos enviados al servidor SMTP (al puerto 25 de manera predeterminada). A cada comando enviado por el cliente (validado por la cadena de caracteres ASCII *CR/LF*, que equivale a presionar la tecla Enter) le sigue una respuesta del servidor SMTP compuesta por un número y un mensaje descriptivo.

A continuación se describe una situación en la que se realiza una solicitud para enviar correos a un servidor SMTP:

- Al abrir la sesión SMTP, el primer comando que se envía es el comando *HELO* seguido por un espacio (escrito <*SP*>) y el nombre de dominio de su equipo (para decir "hola, soy este equipo"), y después validado por Enter (escrito <*CRLF*>). Desde abril de 2001, las especificaciones para el protocolo SMTP, definidas en RFC 2821, indican que el comando HELO sea remplazado por el comando *EHLO*.
- El segundo comando es "*MAIL FROM:*" seguido de la dirección de correo electrónico del remitente. Si se acepta el comando, el servidor responde con un mensaje "*250 OK*".

- El siguiente comando es "*RCPT TO:*" seguido de la dirección de correo electrónico del destinatario. Si se acepta el comando, el servidor responde con un mensaje "*250 OK*".
- El comando *DATA* es la tercera etapa para enviar un correo electrónico. Anuncia el comienzo del cuerpo del mensaje. Si se acepta el comando, el servidor responde con un mensaje intermediario numerado *354* que indica que puede iniciarse el envío del cuerpo del mensaje y considera el conjunto de líneas siguientes hasta el final del mensaje indicado con una línea que contiene sólo un punto. El cuerpo del correo electrónico eventualmente contenga algunos de los siguientes encabezados:
    - Date (Fecha)
    - Subject (Asunto)
    - Cc
    - Bcc (Cco)
    - From (De)

  Si se acepta el comando, el servidor responde con un mensaje "*250 OK*".

A continuación se describe un ejemplo de transacción entre un cliente (C) y un servidor SMTP (S):

```
S: 220 smtp.commentcamarche.net SMTP Ready C: EHLO
machine1.commentcamarche.net S: 250 smtp.commentcamarche.net C: MAIL
FROM:<webmaster@commentcamarche.net> S: 250 OK C: RCPT
TO:<meandus@meandus.net> S: 250 C: RCPT TO:<tittom@tittom.fr> S: 550 No such
user here C: DATA S: 354 Start mail input; end with <CRLF>.<CRLF> C:
Subject: Hola C: Hola Meandus: C: ¿Cómo andan tus cosas? C: C: ¡Nos vemos
pronto! C: <CRLF>.<CRLF> S: 250 C: QUIT R: 221 smtp.commentcamarche.net
closing transmission
```

Las especificaciones básicas del protocolo SMTP indican que todos los caracteres enviados están codificados mediante el código ASCII de 7 bits y que el 8° bit sea explícitamente cero. Por lo tanto, para enviar caracteres acentuados es necesario recurrir a algoritmos que se encuentren dentro de las especificaciones MIME:

- **base64** para archivos adjuntos

- **quoted-printable** (abreviado *QP*) para caracteres especiales utilizados en el cuerpo del mensaje

2.3.4. PROTOCOLO SSH[9]

El protocolo **SSH** (*Secure Shell*) se desarrolló en 1995 por el finlandés Tatu Ylönen.

Es un protocolo que hace posible que un cliente (un usuario o incluso un equipo) abra una sesión interactiva en una máquina remota (servidor) para enviar comandos o archivos a través de un canal seguro.

- Los datos que circulan entre el cliente y el servidor están cifrados y esto garantiza su confidencialidad (nadie más que el servidor y el cliente pueden leer la información que se envía a través de la red). Como resultado, no es posible controlar la red con un rastreador.
- El cliente y el servidor se autentifican uno a otro para asegurarse que las dos máquinas que se comunican son, de hecho, aquellas que las partes creen que son.

El objetivo de la *versión 1* del protocolo (SSH1), propuesta en 1995, ofrecía una alternativa a las sesiones interactivas (shells) tales como *Telnet*, *rsh*, *rlogin* y *rexec*. Sin embargo, este protocolo tenía un punto débil que permitía a los hackers introducir datos en los flujos cifrados. Por este motivo, en 1997 se propuso la versión 2 del protocolo (SSH2) como un anteproyecto del IETF.

Secure Shell Versión 2 también incluye un protocolo SFTP (*Secure File Transfer Protocol; en castellano, Protocolo Seguro de Transferencia de Archivos*).

SSH es un protocolo, es decir, un método estándar que permite a los equipos establecer una conexión segura. Como tal, existe una variedad de implementaciones de clientes y servidores SSH. Algunas requieren el pago de una cuota, en tanto que otras son gratuitas o de *código abierto*.

Cómo funciona SSH

Una conexión SSH se establece en varias fases:

- En primera instancia, se determina la identidad entre el servidor y el cliente para establecer un canal seguro (capa segura de transporte).
- En segunda instancia, el cliente inicia sesión en el servidor.

Establecer un canal seguro

El establecimiento de una capa segura de transporte comienza con la fase de negociación entre el cliente y el servidor para ponerse de acuerdo en los métodos de cifrado que quieren utilizar. El protocolo SSH está diseñado para trabajar con un gran número de algoritmos de cifrado, por esto, tanto el cliente como el servidor deben intercambiar primero los algoritmos que admiten.

Después, para establecer una conexión segura, el servidor envía al cliente su clave de host. El cliente genera una clave de sesión de 256 bits que cifra con la clave pública del servidor y luego la envía al servidor junto con el algoritmo utilizado. El servidor descifra la clave de sesión con su clave privada y envía al cliente un mensaje de confirmación cifrado con la clave se sesión. Después de esto, las comunicaciones restantes se cifran gracias a un algoritmo de cifrado simétrico, mediante la clave de sesión compartida entre el cliente y el servidor.

La seguridad de la transacción se basa en la confianza del cliente y el servidor en que las claves host de cada una de las partes son válidas. Así, cuando se conecta por primera vez con el servidor, el cliente muestra generalmente un mensaje en el que le pide que acepte la comunicación (y posiblemente le presenta un hash de la clave host del servidor):

Para obtener una sesión segura propiamente dicha, es mejor pedirle directamente al administrador del servidor que valide la clave pública presentada. Si el usuario valida la conexión, el cliente guarda la clave host del servidor para evitar tener que repetir esta fase.

Por el contrario, dependiendo de su configuración, el servidor puede, a veces,

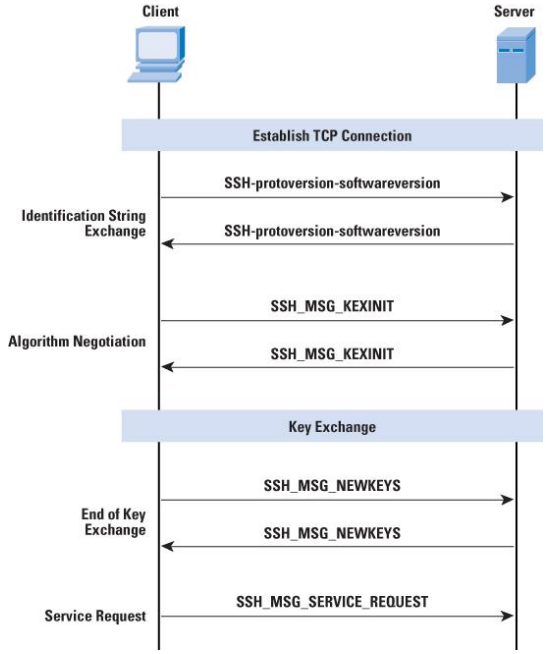
verificar que el cliente es quien dice ser. Si el servidor tiene un lista de hosts autorizados para la conexión, cifrará el mensaje utilizando la clave pública del cliente (que se encuentra en la base de datos de claves del host) para verificar si el cliente es capaz de descifrarla con su clave privada (esto se llama *challenge)*.

Autentificación

Una vez que se ha establecido la conexión segura entre el cliente y le servidor, el cliente debe conectarse al servidor para obtener un derecho de acceso. Existen diversos métodos:

- el método más conocido es la contraseña tradicional. El cliente envía un nombre de acceso y una contraseña al servidor a través de la conexión segura y el servidor verifica que el usuario en cuestión tiene acceso al equipo y que la contraseña suministrada es válida.
- un método menos conocido, pero más flexible, es el uso de claves públicas. Si el cliente elige la clave de autentificación, el servidor creará un *challenge* y le dará acceso al cliente si éste es capaz de descifrar el challenge con su clave privada.

Figura 8. Conexión SSH Cliente/Servidor



Fuente: http://www.ipjforum.org/?p=180

2.4. HERRAMIENTAS DE CONFIGURACION DE SERVICIOS LINUX

Como se estableció en los apartados anteriores, cada servicio corresponde a una funcionalidad específica dentro de un sistema de información, las cuales se encuentran amparadas en el desarrollo y ejecución de protocolos TCP/IP específicos para la comunicación de información que se desea transmitir o compartir. Por consiguiente, los programas informáticos pertenecientes a un sistema Linux, deben ser configurados previamente con parámetros iníciales que permitan la ejecución de sus labores de forma autónoma.

La ventaja de Linux, es que cuenta con varias formas de realizar la configuración inicial de parámetros. Estos parámetros pueden configurarse empleando líneas de comandos, editores de texto o de interfaces web. Conocer estas alternativas representa el primer escalón en la correcta y satisfactoria configuración de un servidor.

Con la finalidad de brindar un ejemplo real en la configuración de un servicio Linux, el presente apartado abordara dos (2) de las formas de realizar la configuración básica de un servicio Linux, tomando como objeto de configuración el servicio SSH (Secure Shell). Se deberá tener en cuenta que en una máquina Linux que se han configurado dos (2) usuarios:

xcasello: Administrador del Sistema
familiac: Usuario sin privilegios de administrador o “root”

2.4.1. EJEMPLO DE CONFIGURACION DE UN SERVICIO LINUX

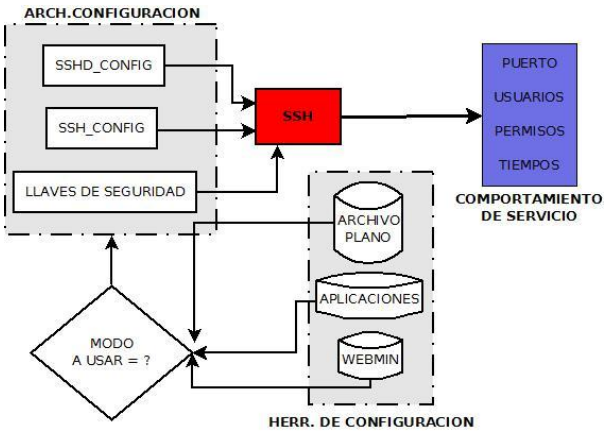
En el caso particular del SSHD (Secure Shell Daemon), proveniente del programa OpenSSH Server, este se encuentra regido por dos (2) archivos principales. Como se menciono en el numeral 1.3 los archivos de configuración se encuentran ubicados en “/etc”. Para el caso del SSH, contenemos los siguientes archivos, que se encuentran específicamente en el directorio “/etc/ssh”

- sshd_config: Corresponde al archivo de configuración global, que establece el comportamiento general del servidor. Características como número de puerto, lista de usuarios, llaves de seguridad, entre otros parámetros generales aplicables a todo el servidor.

- ssh_config: Este archivo permite asignar los parámetros de comportamiento encontrados en el “sshd_config”, pero para cada host que se conecte al SSH. Permitiendo condiciones específicas para cada host que se conecte, tómese por ejemplo: Un usuario “A” puede tener acceso a una sesión grafica, mientras otro usuario “B” se le es denegado por cuestiones de

seguridad. Esto permite una configuración fina y detallada de cada usuario o host que accede al servidor.

Figura 9. Proceso de Configuración SSH



Fuente: Autores.

La figura anterior resalta lo previamente expresado, en donde se establece que los servicios de Linux pueden ser configurados según la escogencia del usuario. Bien sea que haga por medio de un aplicativo, interfaz web o vía comando, las modificaciones a los parámetros de los servicios siempre llevan a una modificación de un archivo o grupo de archivos específicos.

A continuación se configurara dos (2) parámetros específicos dentro de un servidor SSH, haciendo uso de dos (2) herramientas o formas de configuración. Dichos parámetros corresponderá al puerto de escucha y al acceso a usuarios a dicho servicio. Los cuales comprenden a parámetros encontrados en el archivo "/etc/sshd_config". Los parámetros a modificar corresponderán a las opciones descritas como "Port" y "AllowUsers", los cuales se modificaran con los siguientes valores:

PARAMETROS	**VALOR**
Port 23	23

AllowUsers Familiac

Las modificaciones a los parámetros, modificaran el comportamiento del servidor SSH. Es así como se tendrá en primera instancia que al ejecutar una conexión ssh al servidor habrá que especificar el puerto (23), ya que por defecto un comando de conexión ssh lo hace hacia el puerto 22. Y en último el usuario “xcasello” quien es administrador del sistema no podrá conectarse, mientras un usuario “familiac” podrá hacerlo.

2.4.1.1. C

ONFIGURACION VIA ARCHIVOS DE TEXTO PLANO

La configuración de servicios por medio de archivos de texto es la forma más común de configurar una aplicación o un servicio, dentro de un ambiente Linux. Accediendo al archivo “/etc/sshd_config” por medio de cualquier aplicación de edición de texto, bien sea por forma de consola o modo grafico; podremos cambiar los valores de los parámetros “Port” y AllowUsers” del servicio SSH.

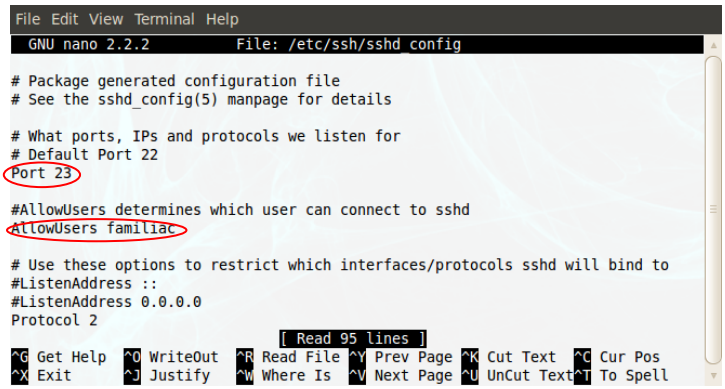
Habiendo accedido a una consola, donde tengamos acceso ejecutamos los siguientes comandos y modificamos las variables previamente mencionadas:

a.) E

l comando “nano”, abre un editor de texto en consola, que permitirá ir al parámetro determinado y cambiar su correspondiente valor. El comando “sudo” que antecede al comando “nano” se emplea para poder acceder a permisos del usuario “root” o administrador para poder modificar el archivo; ya que este tipo de alteraciones requieren de privilegios de administrador.

sudo nano /etc/ssh/sshd_config

Figura 10. Parámetros Modificados en Nano

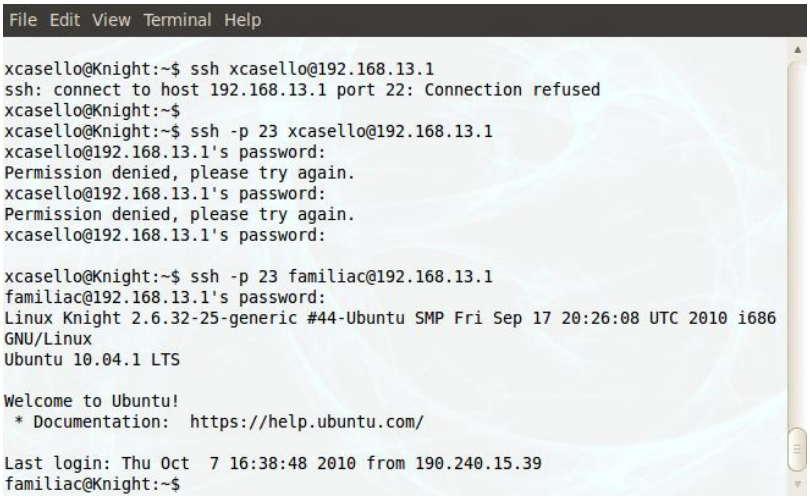


Fuente: Autores.

b.) N

ano tiene el comportamiento de un editor de texto normal. Una vez modificado los valores de las variables “Port” y “AllowUsers”, tecleamos la siguiente combinación de teclas CTRL+x. y en la Pregunta tecleamos “y” + ENTER. Esto guardara los cambios realizados al archivo.

Figura 11. Conexión SSH



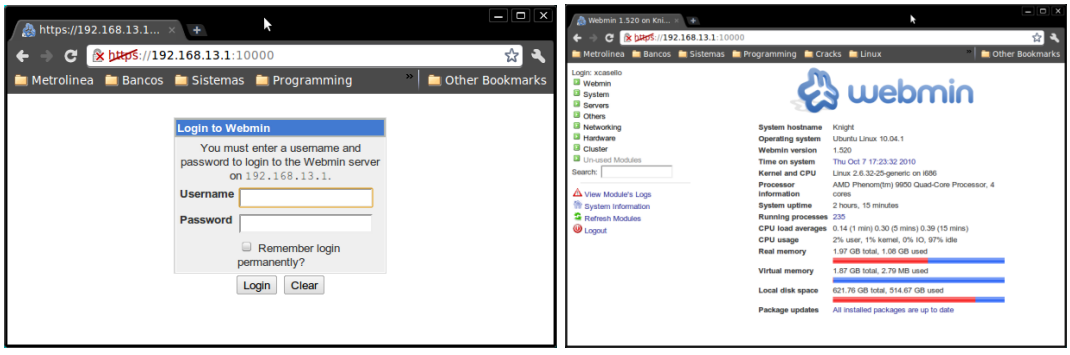
Fuente: Autores.

La figura 11. Muestra la modificación de comportamiento del servidor ssh. Como se observa la primera conexión no fue establecida debido a que el puerto de ssh no esta configurado como el puerto 22 (por defecto). La segunda conexión se establece al ser especificado el puerto 23 (modificado), pero el usuario “xcasello” que es root (Administrador) no puede acceder al servicio ssh. Se intenta una tercera conexión al puerto 23 y con el usuario “familiac”, teniendo un perfecto establecimiento de una sesión ssh en el equipo 192.168.13.1.

2.4.1.2. CONFIGURACION VIA WEBMIN

Webmin es una herramienta de gestión de red, la cual permite controlar y gestionar un gran número de servicios Linux dentro de un servidor o cluster de servidores. Consiste en un programa que establece una interface grafica de tipo web, con el cual se puede parametrizar los valores de determinados servicios. Webmin es un hibrido entre interface grafica web y scripts Perl para la configuración de un sistema Linux completo. Webmin, es un proceso que se ejecuta en el puerto 10000 (por defecto). Al establecer una conexión HTTP al puerto 10000 mediante un navegar web, recibimos una pagina de bienvenida, en la cual es necesario autenticarse como administrador para poder hacer uso de ella.

Figura 12. Ingreso a Webmin

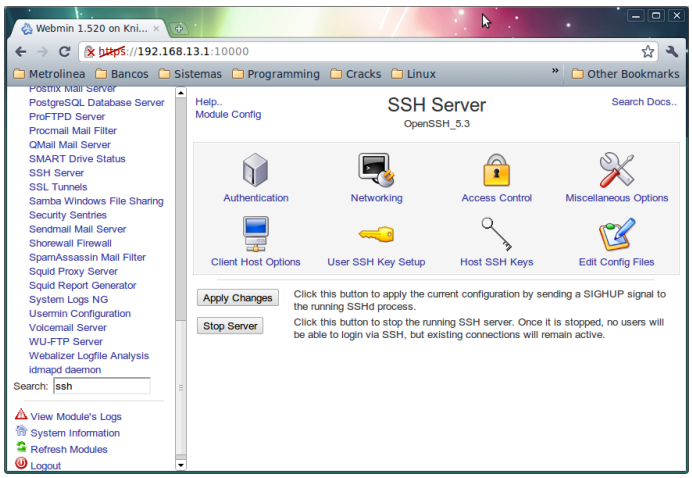


Fuente: Autores.

Por medio de esta interface se pueden realizar los cambios necesarios a los parámetros modificados en el numeral 2.4.1.1. Con el fin de evaluar que el comportamiento del servidor ha sido modificado. Por consiguiente, se procederá a revertir los valores de defecto para servidor ssh. Lo cual significa acceso de ssh para el administrador del sistema “xcasello” y establecimiento de conexión por el puerto 22.

Habiendo accedido a Webmin, buscamos en el borde izquierdo el conjunto de opciones “Servers” y ubicamos a “SSH Server”.

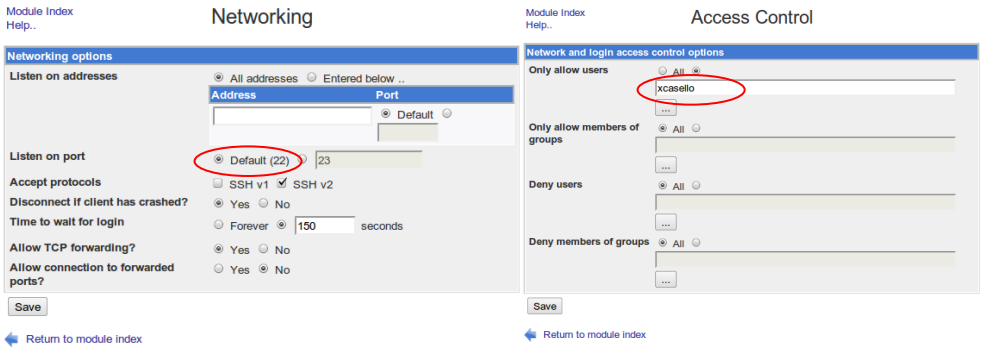
Figura 13. Control de opciones Webmin para SSH



Fuente: Autores.

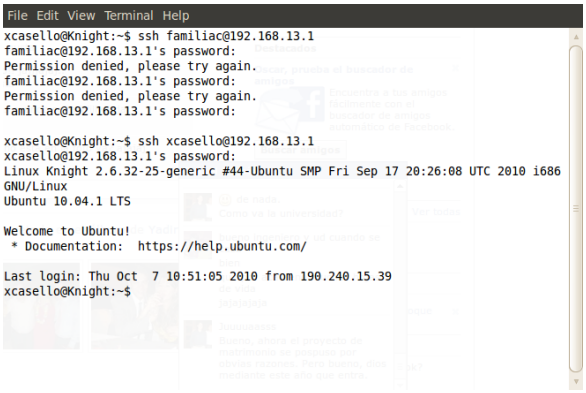
De la anterior figura podemos ver dos opciones que nos interesan, la primera corresponde a “Networking” y la otra a “Access Control”. Dentro de estas opciones se modificara el puerto y las personas que pueden acceder al servidor SSH.

Figura 14. Opciones de configuración



Fuente: Autores.

Figura 15. Conexión SSH



Fuente: Autores.

Como se observa en la Figura anterior, la conexión establecida para el servicio SSH se establece sin necesidad de especificar un puerto, ya que esta se hace empleando el puerto 22 por defecto. En segunda instancia, el usuario “familiac” es denegado acceso al servidor, mientras el usuario “xcasello” entra sin dificultad.

Tener presente las formas o mecanismos para realizar cambios a un servicio es de gran importancia. Sin embargo, más allá del empleo de estas herramientas, el comprender los parámetros que definen el comportamiento de un servicio, son fundamentales a la hora de implementar Linux; estos parámetros son los que

brindan la flexibilidad necesaria para abordar las necesidades especificas de un ambiente informático.

Es importante mencionar, que un servicio SSH hace referencia al protocolo empleado, y como tal existen numerosos programas que lo implementan para ofrecer mayor o mejor soporte a ciertas características del protocolo. Características que pueden ir a un mayor detalle del protocolo o sencillamente limitarse a permitir una configuración más directa y sencilla del servicio. Es así como un mismo protocolo SSH puede tener varios programas que lo ejecuten, cada uno con sus ventajas o especificaciones.

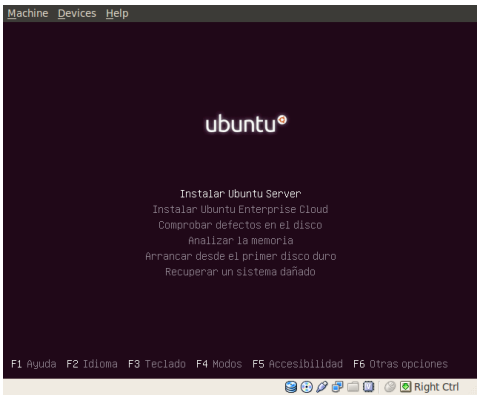
3. INSTALACIÓN DE LINUX

Para los fines prácticos del presente documento, se describirá la instalación de la distribución “Ubuntu Server Edition 10.04”. Se ha optado por la presente distribución, por motivos de facilidad en su gestor de instalación de aplicaciones y su fuerte comunidad de soporte. La instalación se realizara sobre una máquina virtual, empleando el software Virtualbox.

Ubuntu Server Edition 10.04 se puede obtener por medio de internet visitando la dirección electrónica http://www.ubuntu.com. A la hora de seleccionar que tipo de imagen a descargar, es importante conocer el tipo de hardware en el cual se va a instalar; bien sea un procesador con soporte de 32bits o para 64 bits. La recomendación es elegir la versión de 32 bits. Una vez descargada, se graba la imagen en un CD o DVD según corresponda y se procede a iniciar la máquina para que arranque desde la unidad de DVD.

Si el procedimiento anterior ha sido ejecutado satisfactoriamente, se obtendrá una pantalla en donde se pregunte el lenguaje de instalación. Se selecciona el idioma que deseamos, en este caso la instalación será en “español”. Esta opción determinara el lenguaje del instalador de Ubuntu. Obteniendo una pantalla similar a la siguiente imagen:

Figura 16. Menú de Instalación Ubuntu Server

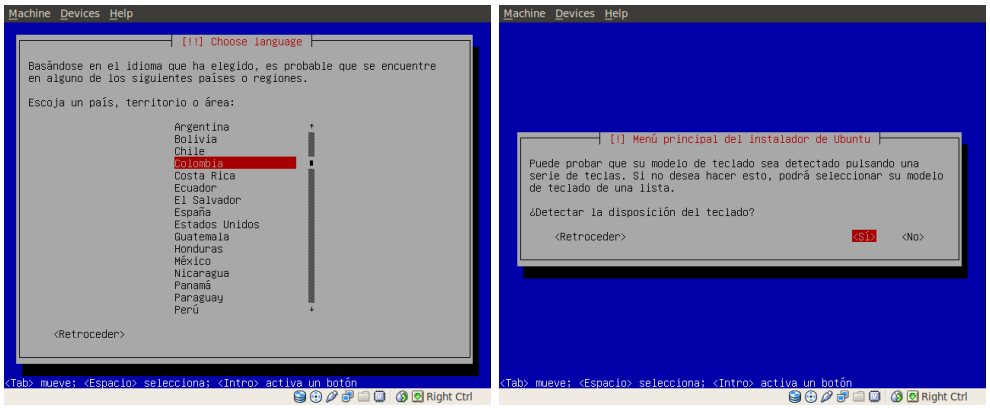


Fuente: Autores.

Se recomienda antes de dar inicio a la instalación comprobar la integridad del disco de instalación, para evitar posibles fallos en el proceso de instalación.

A continuación se procede a dar enter en la opción de “Instalación de Ubuntu Server”, esto conllevara a un cuestionario guiado que permitirá la instalación y configuración inicial de Ubuntu Server.

Figura 17. Preguntas del Instalador Ubuntu

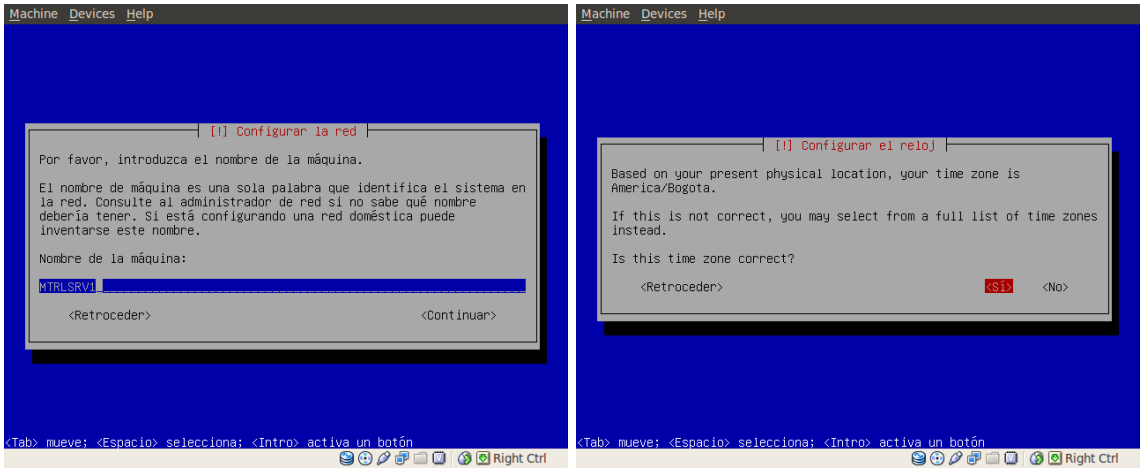


Fuente: Autores.

En la comprobación del modelo de teclado, se puede obviar si se conoce la distribución del teclado a emplear en el servidor. Para la presente instalación se eligió “latam” que corresponde a la distribución de un teclado latinoamericano. En caso de desconocer el tipo de tecleado, se sugiere emplear el modo de deducción automático.

Una vez cargadas las librerías y dependencias que requiere el instalador, proceso que se visualiza por una barra de progreso; se hará necesario asignar un nombre a la máquina Linux. Se asigna el nombre “MTRLSRV1” y se continúa con la siguiente opción del instalador, confirmando que el reloj del sistema se encuentra configurado en la zona de “America/Bogota”.

Figura 18. Nombre Máquina y Reloj

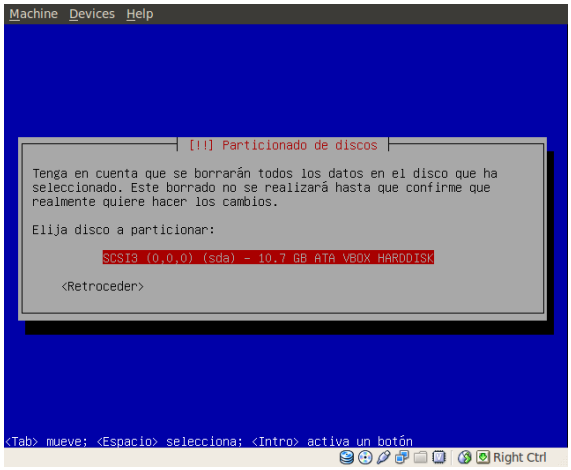


Fuente: Autores.

Continuando con el proceso de instalación, se hace necesario elegir el método de particionado del disco, para alojar el sistema operativo. La opción de LVM se encuentra disponible en la forma normal ó cifrada. LVM (Lógica Volume Management) corresponde a una forma de gestionar discos duros de la máquina, permitiendo agrupar discos físicos en grupos virtuales de discos. Esto permite crear volúmenes lógicos o particiones que permitan variar su tamaño (aumentar o

disminuir la capacidad) y adicionar discos en el caso de existir una falla, daños, etc. Si bien LVM representa una forma de dinámica de administrar nuestros discos, se optara por una instalación más tradicional o estática para cada una de las particiones del sistema operativo. Razón por la cual en el menú de partición de discos se selecciona la opción “Guiado – utilizar todo el disco”. Dependiendo de cuantos discos se tengan conectados a la máquina, podremos determinar en cuál de todos ellos se desea la instalación. Se recomienda usar el primer disco duro del sistema, para garantizar su correcto funcionamiento al iniciar la máquina. En el caso particular del servidor MTRLSRV1 se elige el disco “sda” con capacidad de 10.7 GBytes.

Figura 19. Particionado de disco



Fuente: Autores.

De este procedimiento se obtiene la asignación automática de dos (2) particiones, una correspondiente a la partición “/” que corresponde a la porción lógica del disco en donde todo el sistema Ubuntu Linux será instalado, denominada partición raíz. Y por otro lado se obtiene una partición “swap” que es de intercambio; una memoria virtual o de acceso rápido que es empleada por programas para no sobrecargar la memoria RAM del servidor.

Es importante resaltar que el presente esquema de particiones no es estándar para todos los casos. Dependiendo del tipo de uso, cantidad de usuarios, y servicios a emplear este esquema de particionado puede no ser el más indicado. Sin embargo, para una instalación sencilla se recomienda usar las previas dos particiones mencionadas, en el caso descrito anteriormente se empleara como forma didáctica y bajo ninguna circunstancia debería usarse para servidores de producción.

A continuación se describen las condiciones de las particiones, según su uso particular, en específico para servidores squid y samba.

Para un servidor squid, debido a su alto tráfico y volúmenes de información que maneja por conexión. Se recomienda tener una partición separada para el almacenamiento de logs y cache squid. Considerando que el spool de squid y los registros se encuentran en la ruta "/var" Dicha partición se recomienda se haga en otro disco, aparte de aquel donde se encuentra la raíz de sistema y otros directorios. El tamaño de la partición, es relativo al tamaño de la empresa y cantidad de conexiones, sin embargo a continuación se realiza una ponderación, para ambos tipos de servidores.

Tabla 2. Esquema de Particionado Recomendado

PARTICION	CAPACIDAD PROXY	CAPACIDAD SAMBA
/boot	250 MB	150 MB
/	20 GB	20 GB
/usr	40 GB	40 GB
/var	50 GB	> 100 GB

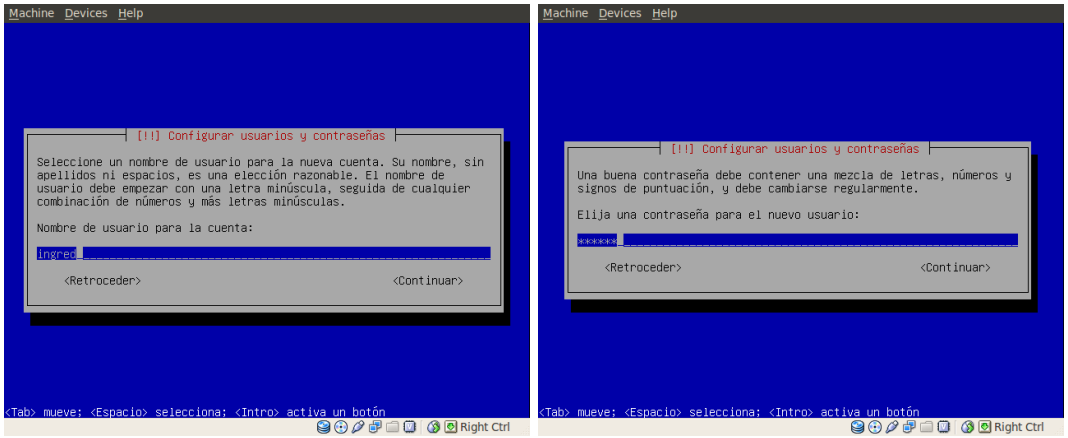
Fuente: Autores

Se recomienda enfáticamente, asignar a un disco diferente del principal, a aquellas particiones que tengan altas probabilidades de crecimiento. De forma tal,

que dicha partición pueda ser copiada a un disco de mayor capacidad, sin pasar por el proceso entero de instalación o ajustes de particiones.

Retomando el proceso de instalación, es hora de crear usuarios y asignar las correspondientes contraseñas para estos. Ubuntu tiene la particularidad de no asignar una clave al usuario “root”, por el contrario un usuario normal es asignado para que por medio del comando sudo se accedan a los permisos de administrador ó “root”. Se asigna de esta forma, un usuario con nombre “Ingeniero de Red” a nuestro sistema, haciendo uso del instalador, quedando registrado como usuario “ingred”, con asignación de clave “123456”.

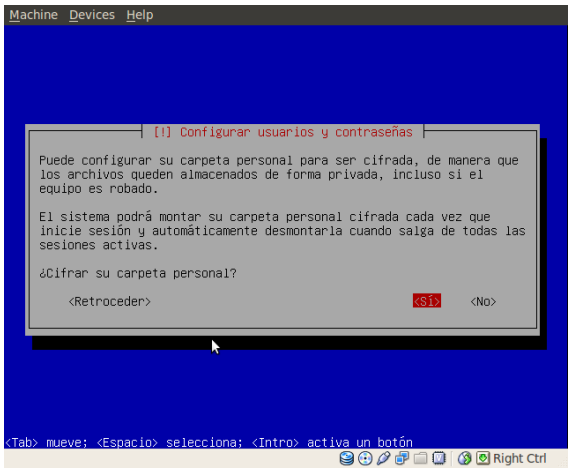
Figura 20. Configuración de usuarios y contraseña



Fuente: Autores.

A su vez, se cifra la partición “/home” para brindar seguridad adicional al sistema. Opción que se presenta en el cuestionario del instalador.

Figura 21. Cifrado de partición /home



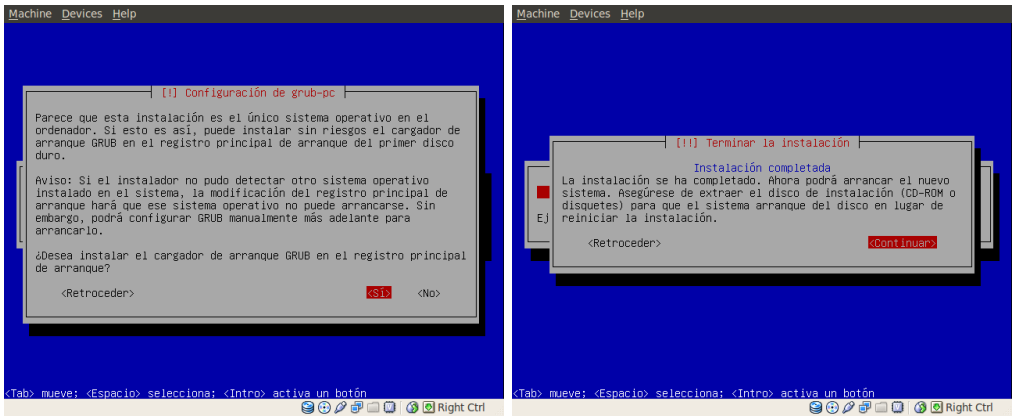
Fuente: Autores.

En la pantalla de “Configurar el gestor de paquetes” no se especifica mayor información y se continúa. En la siguiente pantalla se plantean tres opciones para administrar las actualizaciones en el sistema, se indica “Sin actualizaciones automáticas”. Esta elección es recomendada para garantizar que nuestro sistema continúe funcionando sin alteraciones.

La siguiente pantalla hace referencia a los servicios que se desean instalar automáticamente con la presente instalación. En este caso no se habilita ningún servidor o servicio, puesto el propósito del presente capitulo es la instalación del sistema operativo exclusivamente.

Por último, se configura el arrancador encargado de iniciar Linux cada vez que se energice el computador. El MBR (Master Boot Record) o Registro Maestro de Arranque es la porción de un disco duro que indica la forma o que sistema operativo se ha de ejecutar. Para el caso particular de MTRLSRV1 se elige el disco “sda”, que corresponde al único disco duro disponible. Con este último paso se habrá instalado Ubuntu Server Edition 10.04 de forma satisfactoria.

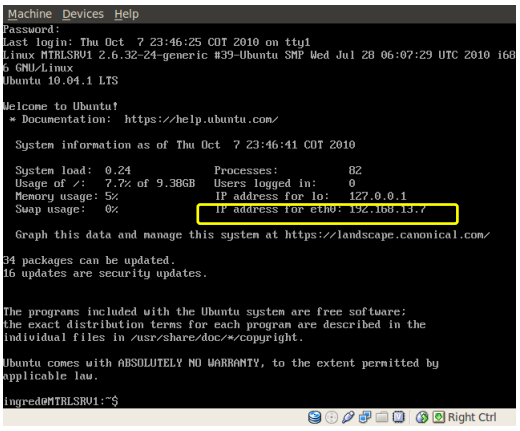
Figura 22. Arranque y Fin de Instalación



Fuente: Autores.

Al finalizar la instalación se procede a revertir la configuración del equipo para que en el arranque no se ejecute la unidad de DVD, y por consiguiente se haga desde el primer disco duro de la máquina. El producto final de los pasos anteriores deberá llevarnos a una pantalla de login, en donde nuestro usuario “ingred” pueda acceder por medio de la clave “123456”.

Figura 23. Login de entrada Ubuntu



Fuente: Autores.

4. ANÁLISIS DE LA RED DE METROLINEA S.A

Conforme al alcance del presente trabajo, se realizara el análisis y valoración de la infraestructura de comunicaciones de la red Metrolínea. Se emplearan herramientas de descubrimiento que permitan identificar la importancia y criticidad de cada elemento dentro de la red de área local (LAN) de la empresa y por consiguiente obtener conclusiones de su condición y futuras propuestas de mejoramiento.

4.1. TOPOGRAFÍA DE LA RED METROLINEA

La topografía de una red, hace referencia a la forma en que diferentes dispositivos se encuentran distribuidos para generar la infraestructura de comunicación que permitan las comunicaciones y distribución de recursos y/o servicios.

La topografía de una red puede ser el listado estructurado de cada uno de los dispositivos, o una representación grafica que permita al administrador de red conocer la interacción de cada uno de los elementos. Este tipo de abstracción permite tener un panorama general de toda la red que se desea administrar o analizar.

Para generar el mapa conceptual o topografía de la red local de Metrolínea, se optó por emplear herramientas de análisis en entorno Linux, siendo el foco del presente documento, la propuesta de servicios de ambientes operativos open source o libres (Linux). Cabe resaltar, que estas herramientas son menos intuitivas que aquellas encontradas en entornos operativos Windows. Sin embargo, la información generada posee el mismo valor y por consiguiente útil para los fines prácticos del análisis.

Empleando un computador de prueba bajo sistema operativo Linux, se efectúa el conexionado a la red de Metrolínea, accediendo a un puerto Ethernet de un Switch

disponible dentro de la empresa. Este primer procedimiento permite establecer que la red se encuentra trabajando bajo el protocolo DHCP (Protocolo de Configuración Dinámica de Host).

Por medio del uso del comando “ifconfig” se evidencia que el equipo tiene asignada una IP de 192.168.1.107, con submascara 255.255.255.0. Ubicando al computador de análisis en la subred 192.168.1.0.

Por medio de la ejecución del comando “*arp-scan –localnet*” se logra usar el protocolo ARP (Protocolo Resolución de Direcciones) de la red, para obtener el listado completo de hosts disponibles en la subred 192.168.1.0. Los datos obtenidos permiten vincular cada host con su correspondiente dirección IP y MAC (Control de Acceso al Medio). Es así, como se obtiene el siguiente listado representativo:

Tabla 3. Hosts de Metrolínea

IP	MAC	DESCRIPCION
192.168.1.1	00:22:b0:8e:ce:97	(Unknown)
192.168.1.101	00:17:a4:d9:e9:a2	Global Data Services
192.168.1.103	00:1e:68:e3:9c:10	(Unknown)
192.168.1.104	00:16:76:18:54:37	Intel Corporation
192.168.1.106	00:16:76:18:54:0c	Intel Corporation
192.168.1.113	00:22:19:fd:92:2b	(Unknown)
192.168.1.120	00:1d:72:45:4e:dd	(Unknown)
192.168.1.122	00:13:8f:cc:c3:03	Asiarock Incorporation
192.168.1.124	00:16:76:18:4f:ca	Intel Corporation
192.168.1.125	00:16:76:18:52:35	Intel Corporation
192.168.1.126	00:16:76:18:4f:83	Intel Corporation
192.168.1.127	00:16:76:18:6f:de	Intel Corporation
192.168.1.129	00:1f:29:ad:bf:37	(Unknown)
192.168.1.131	00:17:a4:d9:f9:61	Global Data Services
192.168.1.133	00:1c:c0:fb:4f:c5	(Unknown)
192.168.1.135	00:16:76:18:4f:1b	Intel Corporation
192.168.1.136	00:16:76:18:7e:fa	Intel Corporation

192.168.1.137	00:16:76:18:4d:b6	Intel Corporation
192.168.1.139	00:13:8f:05:50:eb	Asiarock Incorporation
192.168.1.140	00:12:3f:82:3a:91	Dell Inc
192.168.1.141	00:24:21:34:82:8c	(Unknown)
192.168.1.142	00:13:8f:cc:c3:16	Asiarock Incorporation
192.168.1.143	00:11:09:ab:3b:ba	Micro-Star International
192.168.1.144	00:13:8f:03:b4:20	Asiarock Incorporation
192.168.1.145	00:13:8f:cc:c1:03	Asiarock Incorporation
192.168.1.146	00:19:d1:b1:35:6b	Intel Corporation
192.168.1.149	00:c0:9f:e6:34:6d	QUANTA COMPUTER, INC.
192.168.1.102	00:1e:64:43:bc:da	(Unknown)
192.168.1.158	00:24:81:42:68:b5	(Unknown)
192.168.1.105	00:21:00:4d:53:0c	(Unknown)
192.168.1.110	00:26:c6:7f:73:48	(Unknown)
192.168.1.223	00:c0:ee:2d:12:c4	KYOCERA CORPORATION
192.168.1.224	00:15:60:0b:da:87	Hewlett Packard
192.168.1.100	30:7c:30:64:03:5a	(Unknown)

Fuente: Autores.

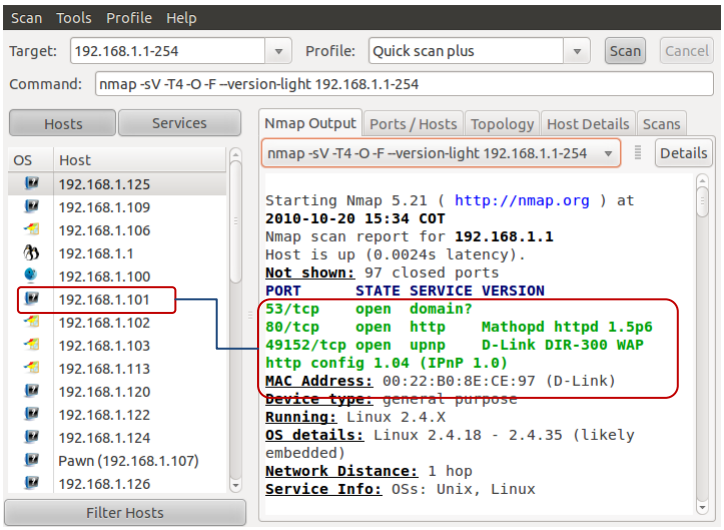
Complementando el comando “arp-scan --localnet”, se emplea la herramienta Zenmap (NMAP – Network Mapper GUI), para obtener las características individuales de cada host; especialmente en aquellos puertos que se encuentren abiertos o disponibles. Estos puertos, permitirán identificar el propósito específico de cada host dentro de la red.

Se asigna un rango de escaneo para toda la subred 192.168.1.0 por medio de Zenmap. Obteniendo como resultado una lista de host similares a “arp-scan”, pero con información detallada de cada uno de los puertos que se encuentran abiertos por host.

Como se mencionó en el numeral 2.1, cada puerto tiene asignado una función o proceso particular. Puerto que permite identificar el propósito de la máquina dentro

de la red. Es así, como se logra identificar que el host 192.168.1.1 y 192.168.1.224 desarrollan funciones muy específica dentro de la red de Metrolínea.

Figura 24. D-Link DIR-300 Wireless Access Point

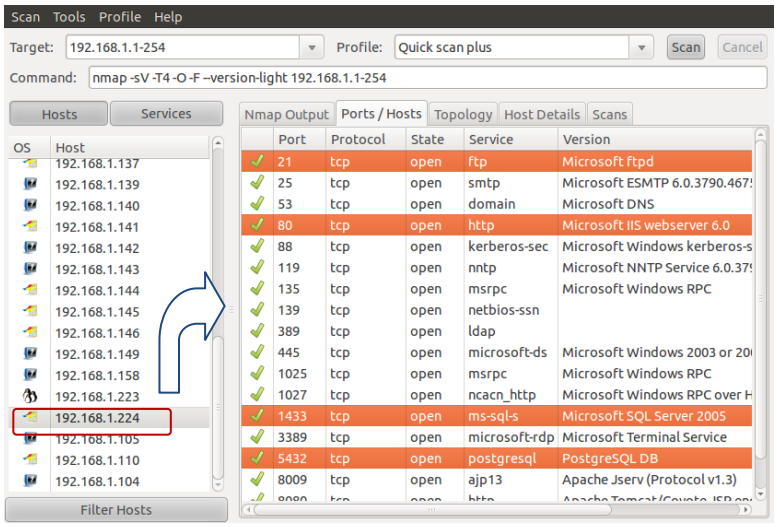


Fuente: Autores.

La figura 24. Revela que el cliente 192.168.1.101 está operando como punto de acceso inalámbrico. Permitiendo de esta forma la comunicación para dispositivos móviles que trabajen con el estándar IEEE 802.11 o WIFI.

Por otro lado, la figura 25. Identifica al cliente 192.168.1.224 como el servidor más relevante dentro de la empresa, al estar ejecutando servicios como servidor de páginas web, Servicio FTP, PostSQL, y Microsoft Terminal Service.

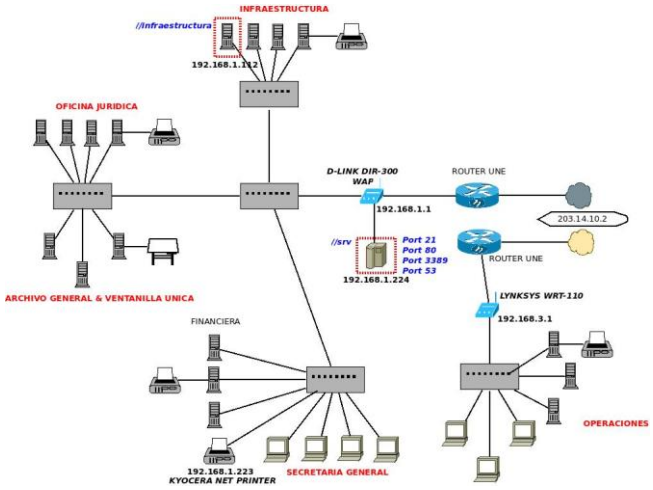
Figura 25. Servidor Web Metrolínea



Fuente: Autores.

Realizando un análisis similar para cada uno de los clientes encontrados, se logra construir una conceptualización grafica de la topografía de la red de Metrolínea, obteniendo una ilustración presentada en la figura 26. La aplicación zenmap no genera la figura de la topología de la red. Esta grafica es la abstracción deducida que se hace de los datos obtenidos por zenmap; resultados que se pueden consultar en el anexo digital.

Figura 26. Topografía Red Metrolínea



Fuente: Autores.

Se resalta que todos los hallazgos realizados por estas herramientas han sido condensados en la figura 26 del presente documento, y las cuales son detalladas y ampliadas en la sección 4.2

4.2. CONCLUSIONES DEL ANÁLISIS DE LA RED METROLINEA

Los datos obtenidos de la aplicación Zenmap permitieron concluir que Metrolínea S.A basa su infraestructura en una disposición bastante sencilla. Sin embargo, esta sencillez resalta factores necesarios de atención, al tratarse de aspectos que compromenten la integridad, seguridad y disponibilidad de la información. A continuación se resaltan las características más relevantes obtenidas dentro del análisis ejecutado:

- Carencia de dispositivos de capa 3 (Switches Administrables). Los resultados del análisis evidencian la carencia de etiquetas que sugieran la existencia de dispositivos activos de red en capa 3. Esta premisa, permite suponer, que todo terminal se encuentra configurado en una única subred.

Suposición que se logro corroborar, al verificarse la configuración de varios terminales, en diferentes dependencias de la entidad.

- Cuatro de cinco impresoras son compartidas vía Windows Share, a excepción de aquella encontrada bajo IP 192.168.1.223.

- El Access Point inalámbrico (192.168.1.1) se encuentran dentro de la única subred disponible para la red y funcionando como puerta de enlace para todas las comunicaciones que salen hacia internet. Igualmente se resalta que todo cliente interno o invitado que hace uso de la red WIFI accede inmediatamente a internet y a los servicios compartidos de Windows de toda la red sin ningún tipo de restricción o protección.

- El host 192.168.1.112 se encuentra operando como servidor de archivos para toda la empresa; al emplearse como carpeta compartida, denominada "//infrestructura". El volumen, tipos de archivos y la cantidad de estos documentos permiten colocar en evidencia la importancia de este cliente dentro de la red local de Metrolínea. Este hallazgo permite evidenciar una vulnerabilidad que coloca en riesgo la información de la empresa; al demostrarse que cualquier usuario puede leer, escribir y ejecutar los archivos existentes.

- El servidor 192.168.1.224 es el único host ofreciendo servicios de importancia para la red. Dentro de ellos acceso a una base de datos y el hosting de la pagina web para el sistema de gestión de calidad de la empresa. Lo anterior se corrobora usando un navegador de internet para acceder a la ip de la máquina, obteniendo una página de bienvenida al "Control de Documentos de Metrolínea".

- La carencia de un servidor específico para la pagina web la empresa, al

igual que de correos electrónicos. Da en evidencia que el domino (www.metrolinea.gov.co) esta ejecutándose como hosting de la pagina web y servicio de correo electrónicos de los empleados.

- El departamento de operaciones se encuentra aislado completamente de la intranet institucional de Metrolínea S.A al encontrarse geográficamente por fuera de las oficinas principales de Metrolínea S.A. No existe un canal de comunicación que los vincule y por lo tanto se toma como otra red de área local independiente.

4.3. EMPLEO DE SERVICIOS LINUX PARA CORREGIR FACTORES DE RIESGOS ENCONTRADOS EN RED DE METROLINEA S.A

Basados en las conclusiones obtenidas del análisis aplicado a la red de Metrolínea, se pueden identificar dos (2) factores significativos. Los cuales serán el centro de desarrollo de los objetivos del presente informe, al tratarse de factores que colocan en riesgo la disponibilidad e integridad de la información de esta empresa.

El primer factor identificable es la falta de control de acceso a internet. Todo cliente interno o invitado de la red de la empresa puede hacer uso ilimitado y sin control de internet. Lo anterior genera un foco de vulnerabilidad al ser un entorno favorable para la propagación de virus informáticos, colocando en riesgo la integridad y disponibilidad de la información.

El segundo factor corresponde a la falta de un repositorio central para el almacenamiento de documentos importantes de la empresa. Como se identificó en el análisis, muchos de los usuarios emplean al servidor 192.168.1.112 para acceder a la carpeta “//infraestructura”, en aras de compartir archivos o de usarlo como punto de almacenamiento de red. El uso de una carpeta compartida no permite la ejecución de permisos de archivos o directorio por usuario o grupos de

usuarios. Permitiendo que un usuario borre y/o modifique archivos de otra persona, al igual que elevar las posibilidades de infección de estos archivos al encontrarse en un host en donde varios empleados tienen acceso.

Por otra parte el host 192.168.1.112 al ser un sencillo computador de escritorio, carece de los mecanismos para tener redundancias y manejar el adecuado proceso de ejecución de respaldo o copias de seguridad. Sin mencionar, que dicho equipo no se encuentran en una localidad con la seguridad física suficiente para evitar su posible hurto.

5. EVALUACION DE SERVICIOS LINUX PARA METROLINEA

Basados en el análisis realizado, se evaluaran aquellos servicios linux que permitan implementar los mecanismos para garantizar la disponibilidad e integridad de la información. En el presente capitulo se brindara información de las características generales de los servicios y la sustentación de cómo estos servicios sirven para subsanar los riesgos encontrados.

5.1. PROPUESTA DE SERVIDOR SQUID – WEB CACHE O PROXY

Como se estableció en el numeral 4.3, Metrolínea carece de los mecanismos para limitar el uso de internet, permitiendo a sus empleados o invitados a usar la red sin ningún tipo de restricciones. Es así, como se evalúa la implantación de un servidor squid o proxy para atender los requerimientos de control de contenidos.

Un servidor de web cache o proxy corresponde a una máquina o servicio que funciona como apoderado o intermediario de una conexión de tipo HTTP. El servidor proxy escucha las solicitudes generadas para la adquisición de contenido web (Protocolo HTTP) y se encarga de traer el material consultado, sin embargo, este almacena los contenidos de forma temporal, para que en caso de volver a ser solicitados; estos puedan ser obtenido y distribuidos de forma local dentro de una red local y no tener que recurrir nuevamente a la fuente (internet).

El principal beneficio de un servidor proxy es la reducción de utilización del ancho de banda de la conexión de internet, mejoras en los tiempos de respuestas a solicitudes de páginas frecuentemente consultadas, y por último la habilidad de funcionar como control de acceso para ciertos protocolos y contenidos, por consiguiente es ideal para empresas que desean establecer políticas de acceso al contenido web. Un servidor proxy toma control sobre las conexiones web que se establecen, permitiendo o denegando el acceso según un criterio establecido. Estos criterios pueden agruparse en varias opciones que se describen a

continuación:

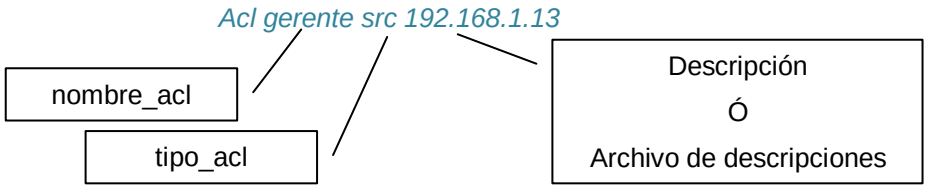
- Permitir un a un usuario o grupos de usuarios acceder a paginas, aplicaciones, archivos y/o servicios específicos. Mientras a otro grupo de usuarios le es denegado el acceso.
- Control de tiempo de navegación de los usuarios, limitando el uso por franjas de horarios.
- Monitoreo de todos las solicitudes que realizan los usuarios, con el fin de llevar un indicador de productividad de cada empleado.
- Funcionamiento de firewall o muro de seguridad interno para las empresas, permitiendo el filtrado de conexiones a usuarios internos.
- Evitar la descarga especifica de ciertos archivos, bien sea por criterio de extensión de archivos, tamaños, etc.

Es así, como un proxy permite encaminar y controlar toda actividad de una red, con el objetivo claro de administrar los recursos que se solicitan al interior o exterior de la red local. Todo proxy, independiente de su sistema operativo o implementación particular, estructura su comportamiento en las denominadas “listas de control o ACL”. Estas listas son la piedra angular que regulan y determinan el comportamiento específico de cada una de las solicitudes realizadas por cada uno de los host de una red.

5.1.1. **LISTAS DE CONTROL ACL SQUID**

Squid basa su habilidad de funcionar como control de acceso empleando un conjunto de listas de control. Estas listas de control reconocidas como “acls”, permite el establecimiento de clases. Dichas clases puede ser un grupo de hosts, un rango de ip, una subred, etc. Tómese por ejemplo los siguientes ejemplos, basados en una estructura definida de listas acls.

Figura 27. ACL Squid



Fuente: Autores.

5.1.2. Tipos de ACL[10]

a) src

Especifica una dirección origen de una conexión en formato IP/máscara.

Por ejemplo, se utilizará una acl de tipo src para especificar la red local:

```
acl red_local src 192.168.1.0/24
```

También se pueden especificar rangos de direcciones mediante una acl de tipo src:

```
acl jefes src 192.168.1.10-192.168.1.25/32
```

b) dst

Especifica una dirección destino de una conexión en formato IP/máscara.

```
acl google_es dst 216.239.0.0/24
```

También se puede especificar hosts concretos mediante una acl de tipo dst:

```
acl google_es2 dst 216.239.59.104/32 216.239.39.104/32 216.239.57.104/32
```

Las definiciones son idénticas a las acl de tipo src salvo que se aplican al destino de las conexiones, no al origen.

c) srcdomain y dstdomain

Estos tipos de acl especifican un nombre de dominio. En el caso de srcdomain es el dominio origen y se determina por resolución DNS inversa de la IP de la máquina, es decir, se tendrá que configurar un DNS para la red local.

En el caso de dstdomain el nombre del dominio se comprueba con el dominio que se haya especificado en la petición de página web. Por ejemplo:

```
acl google_com dstdomain google.com
```

d) srcdom_regex y dstdom_regex

Especifican una expresión regular que verifican los dominios origen o destino. La expresión regular hace distinción entre mayúsculas y minúsculas salvo que incluyamos la opción "-i" que evita dicha distinción. Por ejemplo:

```
acl google_todos dstdom_regex -i google\..*
```

Observamos como al incluir "-i" se indica que no haga distinción entre mayúsculas y minúsculas.

e) time

Este tipo de acl permite especificar una franja horaria concreta dentro de una semana. La sintaxis es la siguiente:

```
acl nombre_acl_horaria time [dias-abrev] [h1:m1-h2:m2]
```

Donde la abreviatura del día es: S - Sunday (domingo), M - Monday (lunes), T - Tuesday (martes), W - Wednesday (miércoles), H - Thursday (jueves), F - Friday (viernes), A - Saturday (sábado).

Por otro lado, la primera hora especificada debe ser menor que la segunda, es decir h1:m1 tiene que ser menor que h2:m2 (h1:m1<h2:m2). Por ejemplo:

acl horario_laboral time M T W H F 8:00-15:00

f) url_regex

Permite especificar expresiones regulares para comprobar una url completa, desde el http:// inicial. Por ejemplo, se establece una acl que se verifique con todos los servidores cuyo nombre sea adserver:

url_regex serv_publicidad ^http://adserver.*

En otro ejemplo se puede usar una acl que verifique las peticiones de ficheros mp3:

url_regex ficheros_mp3 -i mp3$

g) referer_regex

Define una acl que se comprueba con el enlace que se ha pulsado para acceder a una determinada página. Cada petición de una página web incluye la dirección donde se ha pulsado para acceder. Si se escribe la dirección en el navegador entonces se estará haciendo una petición directa. Por ejemplo, se establece una acl para todas las páginas a las que se hallan accedido pulsando en una ventana de búsqueda de google:

acl pincha_google referer_regex http://wwww.google.*

h) req_mime

Las acl de tipo req_mime se utilizan para comprobar el tipo de petición mime que realiza un cliente, y se puede utilizar para detectar ciertas descargas de ficheros o ciertas peticiones en túneles HTTP. Esta acl sólo comprueba las peticiones que realiza el cliente, no comprueba la respuesta del servidor. Esto es importante para tener claro qué se esta ejecutando. Por ejemplo,

```
acl subida req_mime_type -i ^multipart/form-data$

acl javascript req_mime_type -i ^application/x-javascript$
acl estilos req_mime_type -i ^text/css$
acl audiompeg req_mime_type -i ^audio/mpeg$
```

i) rep_mime_type

Este tipo de acl se utiliza para verificar el tipo de respuesta recibida por el proxy. Este tipo de acl, analiza una respuesta del servidor por lo que sólo le afectas las reglas de respuesta como http_reply_access y no las reglas http_access que se aplican a las peticiones. Por ejemplo,

```
acl javascript rep_mime_type -i ^application/x-javascript$
acl ejecutables rep_mime_type -i ^application/octet-stream$
acl audiompeg rep_mime_type -i ^audio/mpeg$
```

En complemento a una acl, tiene varios operadores que permiten admitir o denegar una solicitud en particular a una clase de “acls”. El operador-acl se conoce como http_access.

5.1.3. Operador ACL – HTTP_ACCESS

El operador-acl que permite o deniega accesos a una o más ACL, contempla la

siguiente sintaxis:

http_access allow|deny [!]acl ...

Si no hay ninguna línea de acceso la acción predeterminada es denegar la petición. Si una petición no ha verificado ninguna línea de acceso, la acción que se realiza es la opuesta a la última línea de la lista. Si la última línea deniega entonces el valor predeterminado es permitir. Por este motivo es conveniente incluir una línea "deny all" o "allow all" al final de las listas de accesos para tener las cosas claras.

Es importante la forma en la cual se añaden los distinto parámetros http_access para determinar la forma en la que se comprueba. En primer lugar, todas las acl incluidas en una cláusula http_access se comprueban y todas ellas tendrán que verificarse conjuntamente, es decir, como si estuvieran unidas por un operador AND. Después, los sucesivos parámetros http_access se evalúan individualmente, es decir, como si estuvieran unidos mediante un operador OR.

Por ejemplo,

```
acl red1 src 192.168.0.0/24
acl red2 src 192.168.1.0/24
http_access allow red1 red2
```

Permitiría el acceso a todas aquellas conexiones que procedieran a la vez de red1 y de red2. Para permitir acceso a las dos redes se podría usar:

```
acl red1 src 192.168.0.0/24
acl red2 src 192.168.1.0/24
  http_access allow red1
  http_access allow red2
```

o también, de forma más simple:

```
acl redes src 192.168.0.0/24 192.168.1.0/24
http_access allow redes
```

Como se observa, squid cuenta con un gran número de opciones con las cuales experimentar, y de esta forma ajustarlas a las necesidades especificas de nuestro entorno informático.

En el caso particular de Metrolínea S.A, se ha establecido, según criterio del administrador de la red, que dichas restricciones a implementar por squid se realicen de una forma escalonada; contrario al planteamiento inicial de los autores de limitar e ir habilitando según un cuadro de necesidades. Sin embargo, este argumento es válido, considerando que internet ha sido un privilegio que todo usuario ha tenido desde hace tiempo y por lo tanto debe abordarse con cierto grado de diplomacia, para evitar inconvenientes con los usuarios. Es así, como se plantea inicialmente dos (2) tipos de configuraciones básicas o condiciones de funcionamiento:

- Permitir acceso a internet única y exclusivamente a los equipos de la empresa. Equipos portátiles y dispositivos móviles personales estarán sin conexión a internet.

- Bloqueo de todo sitio cuya insinuación tenga que ver con violencia, sexo o paginas carentes de componentes productivos para la empresa.

La configuración de estas condiciones serán cubiertas en el capitulo siguiente; las cuales quedaran consignadas como guía para el administrador de red, y cuyo

comportamiento podrá ser modificado según criterios posteriores al presente documento.

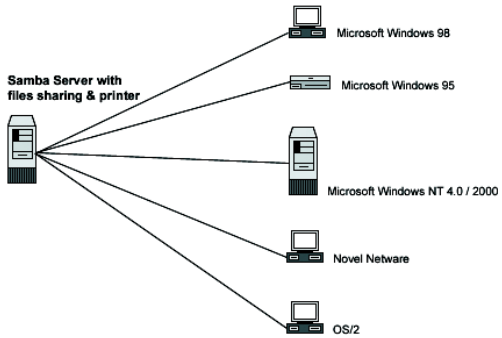
5.2. PROPUESTA DE SERVIDOR SAMBA – SERVIDOR DE ARCHIVOS[11]

Como segundo factor a ser corregido, está el empleo de unidades compartidas para uso de intercambio de archivos. Como se describió en el numera 4.2 el host 192.168.1.112 está operando como servidor de archivos, sin contar con los mecanismo para tener un adecuado permiso de acceso a los archivos y respaldo de los mismos. Es así como un servidor Samba se propone a ser implementado para que funcione como servidor de archivos dentro de la empresa.

Samba es una implementación libre del protocolo de archivos compartidos de Microsoft Windows (antiguamente llamado SMB, renombrado recientemente a CIFS) para sistemas de tipo UNIX. De esta forma, es posible que ordenadores con GNU/Linux, Mac OS X o Unix en general se vean como servidores o actúen como clientes en redes de Windows. Samba también permite validar usuarios haciendo de Controlador Principal de Dominio (PDC), como miembro de dominio e incluso como un dominio Active Directory para redes basadas en Windows; aparte de ser capaz de servir colas de impresión, directorios compartidos y autentificar con su propio archivo de usuarios.

Samba puede configurar directorios Unix y GNU/Linux (incluyendo sus subdirectorios) como recursos para compartir a través de la red. Para los usuarios de Microsoft Windows, estos recursos aparecen como carpetas normales de red; permitiendo colocar a disposición de una red Windows, un sistema de archivos linux para almacenar información, con las ventajas de seguridad y capacidad de entornos Linux. En especial el estricto control de permisos de lectura, escritura y ejecución para archivos.

Figura 28. Servidor Samba Linux



Fuente: http://www.faqs.org/docs/securing/chap29sec280.html

5.2.1 PERMISOS DE ARCHIVOS EN LINUX[12]

Para entender el concepto de permisos se debe tomar en cuenta que cada usuario puede pertenecer a uno o más grupos. Cada usuario pertenece por lo menos a un grupo, que es establecido en el momento en que el usuario se crea. El administrador del sistema puede agregar al usuario a otros grupos, si es necesario. Estos grupos son necesarios para poder establecer una política de acceso más organizada dado que en cualquier momento se podría dar a un archivo el acceso a personas de un grupo determinado. Lo único que se tendría que hacer es agregar a los usuarios que se quieran dar permisos a ese grupo.

Para cada objeto (archivo) que se encuentre en el sistema, GNU/Linux se cuenta con una información relevante a el, el cual corresponde a el dueño de archivo (owner), el grupo del archivo (group) y los bits de modo o también llamados permisos de archivo.

Al ser linux sistemas operativos multiusuario, para que se puedan proteger los archivos se estableció un mecanismo por el cual se otorgan permisos a un determinado usuario y/o grupo. Esto permite, por ejemplo, que si existe un archivo

creado por un usuario en particular, este será propiedad del usuario y también tendrá el grupo del usuario. Se permite que los archivos sean compartidos entre usuarios y grupos de usuarios. Por ejemplo si shrek quisiera puede prohibir los accesos a un archivo determinado que le pertenezca a todos los usuarios que no pertenezcan a su grupo de usuarios.

Los permisos están divididos en tres tipos: lectura, escritura y ejecución (rwx). Estos permisos pueden estar fijados para tres clases de usuario: el propietario del archivo, el grupo al que pertenece el archivo y para todo el resto de los usuarios. El permiso de lectura permite a un usuario leer el contenido del archivo o en el caso de que el archivo sea un directorio, la posibilidad de ver el contenido del mismo. El permiso de escritura permite al usuario modificar y escribir el archivo. En el caso de un directorio permite la crear nuevos archivos en él o borrar archivos existentes. El permiso de ejecución permite al usuario ejecutar el archivo, si tiene algo para ejecutarse. Para los directorios permite al usuario cambiarse a él con el comando **cd**.

Para poder interpretar los permisos de archivos se observa el siguiente listado:

```
[shrek@pantano:~]$ ls -la
total 13
drwxr-sr-x   2 shrek    user       1024 May  2 09:04 .
drwxrwsr-x   4 root     staff      1024 Apr 17 21:08 ..
-rw-------   1 shrek    user       2541 May  2 22:04 .bash_history
-rw-r--r--   1 shrek    user        164 Apr 23 14:57 .bash_profile
-rw-r--r--   1 shrek    user         55 Apr 23 14:44 .bashrc
-rwxrwxr-x   1 shrek    user          0 Apr 14 19:29 a.out
-rwxrwxr-x   1 shrek    user         40 Apr 30 12:14 hello.pl
-r--------   1 shrek    user         64 Apr 29 14:04 hola
-rwxrw-r--   1 shrek    user        337 Apr 29 13:57 lista
-rw-rw-r--   1 shrek    user        40  Apr 30 12:31 listador
```

```
-rw-rw-r--   1 shrek    user         0 May  2 09:04 null
-rwxrwxr-x   1 shrek    user       175 Apr 30 12:30 prue.pl
-rwxrwxr-x   1 shrek    user        56 Apr 23 15:08 que.sh
```

Como se puede apreciar en el listado, también están el directorio actual, representado por un punto "." y el directorio padre representado por dos puntos ".." . Ambos poseen permisos y atributos que son mostrados. Para entender que significa, se explicara que significan los primeros 10 dígitos de un archivo. Tómese como ejemplo el siguiente archivo:

```
-rw-r--r--   1 shrek    user       337 Apr 29 13:57 lista
```

Figura 29. Permisos de un Archivo

Fuente: http://www.ant.org.ar/cursos/curso_intro/x1439.html

Para comprender un poco mas que significa cada uno de estos caracteres al inicio, se emplearan unas tablas. Primero se observan aquellos caracteres que podrían aparecer en el primer lugar, que en el ejemplo anterior es un solo guión. Esto indica que es un archivo común. La tabla siguiente explica el significado del primer símbolo de acuerdo al tipo de archivo.

Tabla 4. Tipos de archivo

Contenido	Significado
-	Archivo común
d	Directorio
c	Dispositivo de caracteres (tty o impresora)
b	Dispositivo de Bloque (usualmente disco rígido o CD-ROM)
l	Enlace simbólico
s	Socket
p	Pipe

Fuente: http://www.ant.org.ar/cursos/curso_intro/x1439.html

Los siguientes 9 símbolos se toman en grupos de tres y cada grupo pertenece a una clase de permisos, y se muestran a continuación.

Tabla 5. Tipos de Permisos

Permiso	Significado
r	Permiso de lectura
w	Permiso de escritura
x	Permiso de ejecución

Fuente: http://www.ant.org.ar/cursos/curso_intro/x1439.html

Tabla 6. Grupos de Permisos

Columnas	Se aplica a	Significado

Columnas	Se aplica a	Significado
2,3,4	owner	Establece los permisos para el dueño del archivo
5,6,7	group	Establece los permisos para el grupo del archivo
8,9,10	other	Establece los permisos para los usuarios que no entran en las categorías anteriores

Fuente: http://www.ant.org.ar/cursos/curso_intro/x1439.html

De esta forma se puede interpretar el listado generado a partir del comando **ls -la** (listar archivos). Como se expresó, el primer símbolo indica que el archivo es un archivo común. El primer grupo de tres símbolos representa los permisos para el dueño del archivo (owner) que en este caso posee permisos de lectura, escritura y ejecución. El segundo grupo de tres símbolos representa los permisos para el grupo al cual pertenece el archivo (group), que en este caso tienen permisos de lectura y escritura. El tercer grupo de tres símbolos representa los permisos para todo el resto de los usuarios (other) en este caso es solo de lectura.

El número que sigue (1) representa el número de nombres que el archivo posee. Esto indica la cantidad de enlaces que existen a este archivo, comúnmente cuando se emplean simbólicos y duros. A continuación esta el nombre del dueño del archivo y del grupo al cual pertenece el archivo. El "337" representa el tamaño del archivo expresado en bytes. Lo siguiente es la fecha y hora de modificación del archivo e inmediatamente después esta el nombre del mismo.

Los permisos de los archivos también dependen del directorio donde estén guardados. En un ejemplo común se podría dar el caso de un archivo que posea todos los permisos, tanto para el usuario, grupo y otros pero no se podrá acceder

a él si no se cuenta con permisos de lectura y ejecución en el directorio que los contiene.

Esto funciona en el caso que se desee restringir el acceso a un directorio determinado y a todos los archivos que este contiene. En lugar de cambiar los permisos uno por uno solo habría que eliminar los permisos necesarios para que se prohíba el acceso mismo al directorio y con esto no se podrá ingresar para usarlos. Esto también está dado para toda la ruta del archivo. Es decir que no solo el último directorio, el cual lo contiene, tiene que tener los permisos necesarios, sino que todos los directorios que lo preceden también.

El comando chmod se emplea para realizar el cambio de permisos a todos los objetos de un sistema linux. El comando chmod se emplea utilizando símbolos como a,u,g,o que representan a todos (a "all"), al usuario (u), al grupo (g) y a todos los demás (o). Existen símbolos para agregar (+) quitar (-) o dejar invariantes los permisos (=). Además tendrán que usarse los símbolos característicos para cada tipo de permiso. Para el permiso de lectura (r), para el permiso de escritura (w) y para el permiso de ejecución (x). Solo el dueño del archivo puede cambiarlo con él; excepción del root que también lo puede hacer. Para ejemplificar un cambio de permisos usaremos el archivo lista.

```
[shrek@pantano:~]$ ls -l lista
total 1
-rwxrw-r--   1 shrek    user        337 Apr 29 13:57 lista
[shrek@pantano:~]$ chmod a-r lista
[shrek@pantano:~]$ ls -l lista
total 1
--wx-w----   1 shrek    user        337 Apr 29 13:57 lista
```

De esta forma se ha eliminado a todos los grupos y usuarios los permisos de lectura. Algunos ejemplos más:

```
[shrek@pantano:~]$ chmod u+r lista
[shrek@pantano:~]$ ls -l lista
total 1
-rwx-w----   1 shrek    user        337 Apr 29 13:57 lista
[shrek@pantano:~]$ chmod o+w lista
[shrek@pantano:~]$ ls -l lista
total 1
-rwx-w-w--   1 shrek    user        337 Apr 29 13:57 lista
[shrek@pantano:~]$ chmod og-w lista
[shrek@pantano:~]$ ls -l lista
total 1
-rwx------   1 shrek    user        337 Apr 29 13:57 lista
```

Ahora bien, esta es la forma simbólica. Pero existe una forma un poco más sistemática que es la forma de representación octal. El comando **chmod** permite establecer los permisos de un archivo por medio de un número octal.

En la representación octal solo se usan 8 números (0,1,2,3,4,5,6,7), para establecer los permisos. De esta forma habrá que sumar los dígitos octales de acuerdo a una tabla que se dará a continuación. Dado que no se realiza acarreo, la suma será trivial.

Tabla 7. Permisos en Notación Octal

Número octal	Permiso
4000	Establece el número de identificación de usuario al ejecutarse SUID [a]
2000	Establece el número de identificación de grupo al ejecutarse SGID[a]

Número octal	Permiso
1000	Establece el *bit adhesivo*[a]
0400	Lectura por parte del dueño
0200	Escritura por parte del dueño
0100	Ejecución por parte del dueño
0040	Lectura por parte del grupo
0020	Escritura por parte del grupo
0010	Ejecución por parte del grupo
0004	Lectura por parte de los otros
0002	Escritura por parte de los otros
0001	Ejecución por parte de los otros
Notas: a. Se explica mas adelante	

Fuente: http://www.ant.org.ar/cursos/curso_intro/x1439.html

Para dar un ejemplo de la suma que se tendrá que realizar, tomamos un archivo con los permisos expresados en forma simbólica y realicemos la conversión. Para representar -rwxr-x---

```
  0400     Lectura por parte del dueño
+ 0200     Escritura por parte del dueño
+ 0100     Ejecución por parte del dueño
+ 0040     Lectura por parte del grupo
+ 0010     Ejecución por parte del grupo
----------------------------------------
 0750     Resultado
```

De esta forma si lo que se desea es cambiar los permisos de un archivo, solo se tendría que efectuar la suma necesaria y establecerlo con el comando chmod. Si quisiéramos cambiar los permisos para que el dueño tenga permisos de lectura y escritura y que el grupo y otros solo tengan permisos de lectura, la sintaxis es

```
[shrek@pantano:~]$ chmod 0644 lista
[shrek@pantano:~]$ ls -l lista
total 1
-rw-r--r--   1 shrek    user        337 Apr 29 13:57 lista
```

Por otro lado es importante saber cómo se cambia de usuario en un archivo. Para esto se usa el comando **chown** y su sintaxis es similar a la de **chmod** pero con la variante que se dan los nombres del usuario y del grupo. Si se desea cambiar el nombre de usuario del archivo lista tendremos:

```
[root@pantano:/home/shrek]# ls -l lista
total 1
-rw-r--r--   1 shrek    user        337 Apr 29 13:57 lista
[root@pantano:/home/shrek]# chown fiona lista
[root@pantano:/home/shrek]# ls -l lista
total 1
-rw-r--r--   1 fiona    user        337 Apr 29 13:57 lista
```

Si se desea cambiar también el nombre del grupo, se tendría que poner un punto entre el nombre de usuario y el grupo

```
[root@pantano]# ls -l lista
total 1
-rw-r--r--   1 shrek    user        337 Apr 29 13:57 lista
[root@pantano]# chown fiona.ventas lista
[root@pantano]# ls -l lista
total 1
-rw-r--r--   1 fiona    ventas      337 Apr 29 13:57 lista
```

Por supuesto que tanto el usuario como el grupo al que se hacen referencia tendrán que existir en el sistema, sino se producirá un error. En el caso que solo se quiera cambiar el grupo y no el usuario, se tendrá que poner un punto delante del nombre del grupo, omitiendo poner el nombre del algún usuario. O si se quiere, se podrá poner el nombre de usuario que estaba anteriormente.

```
[root@pantano]# ls -l lista
total 1
-rw-r--r--   1 shrek    user        337 Apr 29 13:57 lista
[root@pantano]# chown .ventas lista
[root@pantano]# ls -l lista
total 1
-rw-r--r--   1 shrek    ventas      337 Apr 29 13:57 lista
```

Explicaremos algunos puntos sobre permisos que son de gran utilidad para la seguridad de nuestro sistema. Umask es la abreviatura de *user file-creation mode mask* o *máscara del modo de creación de archivos de usuario* y es un número octal de cuatro dígitos que se utilizan para fijar los permisos de los archivos recién creados. Esto puede ocasionar confusión pero en realidad es una utilidad que permite el uso del sistema por múltiples usuarios sin que peligre la privacidad. En la mayoría de los sistemas Unix/Linux los archivos que son creados por el usuario, poseen permisos 0666 que dan permiso de lectura y escritura a cualquier usuario. En relación con los programas, estos se crean con 0777 donde cualquier usuario puede leer, escribir y ejecutar el programa. Normalmente el administrador del sistema aplica una *máscara* al usuario en el archivo .bash_profile y esta es usada para la creación de archivos haciendo una operación simple "AND" bit por bit con el complemento del valor umask bit por bit. La función umask esta integrada al intérprete de comandos. Para ejemplificar el proceso tomemos un archivo creado por el usuario.

```
  0666      Modo predeterminado de creación de archivos
- 0022      Umask
```

0644 Modo resultante

El modo resultante es que el dueño tiene permisos de lectura y escritura y los demás y el grupo solo de lectura.

0666 Modo predeterminado de creación de archivos
- 0077 Umask

0600 Modo resultante

El modo resultante es que el dueño tiene permisos de lectura y escritura y los demás y el grupo no tienen ningún permiso. Una forma de darse cuenta de la forma en que funciona es tener en cuenta que el valor 2 inhabilita el permiso de escritura mientras que el valor 7 inhabilita los permisos de lectura escritura y ejecución. A continuación daremos una tabla con los valores comúnmente usados para él.

Tabla 8 Valores usuales de la variable

Umask	**Accesos del usuario**	**Accesos del grupo**	**Accesos de los otros**
0000	Todos	Todos	Todos
0002	Todos	Todos	Lectura y ejecución
0007	Todos	Todos	Ninguno
0022	Todos	Lectura y ejecución	Lectura y ejecución
0027	Todos	Lectura y ejecución	Ninguno
0077	Todos	Ninguno	Ninguno

Fuente: Fuente: http://www.ant.org.ar/cursos/curso_intro/x1439.html

Como se puede apreciar, Linux cuenta con una fuerte arquitectura de permisos para archivos y directorios. Permitiendo tener un control bastante detallado y riguroso de aquellos usuarios o grupos de usuarios que hacen uso de estos.

Para el caso particular de Metrolínea se desea establecer un servidor samba que ofrezca carpetas por cada dependencia que existe en la compañía. La finalidad es generar un repositorio central por departamentos, y así evitar el entre cruzamiento de información entre dependencias. Si el intercambio fuese necesario, se emplearía una carpeta de “intercambio” para dichos propósitos. De lo anterior se desprenden las siguientes condiciones generales:

- Los permisos de escritura serán agrupados por los grupos de la dependencia, es decir, un usuario podrá escribir solo en la carpeta que pertenece a su grupo.
- Tan solo un usuario podrá tener permisos en todas las carpetas, este será el usuario “sistemas”.
- Los clientes Windows serán configurados para que ingresen a la unidad de red samba de forma directa, sin ninguna digitación de claves.

6. INSTALACION Y CONFIGURACION DE SERVICIOS LINUX PARA METROLINEA

6.1. INSTALACION DE APLICATIVO WEBMIN

Como se menciono en el literal 2.4.1 del presente documento, linux posee varias formas de realizar la configuración de servicios. Es así como Webmin será la puerta de entrada a la configuración de los servicios, excepto en aquellos casos en donde se haga necesario la edición o creación de ciertos parámetros vía comando o editores de texto.

Para realizar la instalación se procede a ejecutar los siguientes comandos:

nano /etc/apt/sources.list

Agregar al final de archivo la siguiente línea y guardar el documento:

deb http://download.webmin.com/download/repository sarge contrib

Continuar con la ejecución de los siguientes comandos:

cd
wget http://www.webmin.com/jcameron-key.asc
sudo apt-key add jcameron-key.asc
sudo apt-get update
sudo apt-get install webmin

Para comprobar la instalación de Webmin, accedemos a la IP que tiene configurada la maquina virtual. En el caso particular de este ejemplo, se tiene una

IP 192.168.13.2. Es necesario sin embargo especificar la siguiente dirección para tener acceso webmin https://192.168.13.2:10000. Nótese que el valor “10000” que hace referencia al puerto donde se encuentra corriendo el proceso webmin.

6.2. INSTALACION DE SERVIDOR SQUID

En el capítulo 3 del presente documento se realizó la instalación plena de un sistema linux Ubuntu Server Edition 10.04. Lo cual significa una instalación del sistema base Ubuntu, sin ninguno de los servicios activados. El procedimiento para la instalación y activación de servidor squid se realiza por medio del siguiente comando:

sudo apt-get install squid

El comando anterior requiere de autenticación como administrador de sistema, para lo cual se continúa a ingresar la clave previamente asignada, “123456”. Si la instalación de squid fue correcta, se deberá tener un puerto “3128” disponible.

Para realizar esta verificación procedemos a instalar y usar el programa nmap, por medio de los siguientes comandos:

sudo apt-get install nmap
nmap p 3128 –T4 192.168.13.2

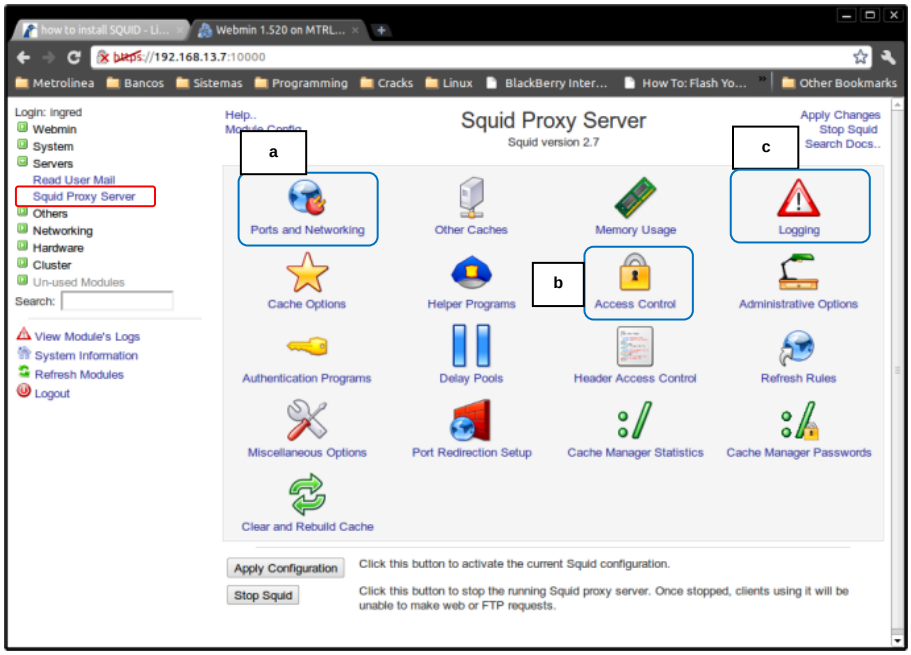
Habiendo usado nmap deberíamos obtener el resultado de “3128/tcp open squid-http”. Esto corrobora que el servidor squid se encuentra instalado y ejecutándose como un proceso en el puerto.

6.2.1. CONFIGURACION DEL SERVICIO SQUID

Por medio de la aplicación y servicio Webmin que ha sido instalado en el servidor local. Accedemos a la dirección web https://192.168.13.2:10000 por medio de un navegador web. Al ser preguntados por el usuario y clave, proporcionamos el usuario “ingred” y la clave “123456”. Téngase en cuenta que esta consulta HTTP se puede realizar en cualquier computador que se encuentre en el segmento de red, en donde se encuentra el servidor linux de prueba. Igualmente, se deberá acceder al servidor desde otro equipo, ya que el servidor que está ejecutando los procesos de squid no tiene entorno grafico instalado.

Una vez en Webmin, buscamos en las opciones de la franja izquierda de la pantalla, la opción de “Servers” y “Squid Proxy Server”.

Figura 30. Opciones configuración Squid



Fuente: Autores

Nótese que en la figura 31, se indican unas etiquetas con los literales a,b y c. Estas hacen referencia a las opciones que se explicaran a continuación y que son las responsables del comportamiento del servidor squid.

a) Ports and Networking Options

En esta sección se puede realizar el cambio del puerto predefinido para el servidor squid. Igualmente, se puede determinar puertos adicionales para escuchar solicitudes provenientes de la red. Se cambiara el puerto “3128” que viene por defecto al puerto “3130”. El siguiente cambio se realizara para determinar que webmin está ejecutando los cambios al daemon de squid. Se procede a emplear el comando nmap –p 3130 –T4 192.168.13.2”, para el cual se deberá tener una respuesta igual a la siguiente: “3130/tcp open unknown”.

Por consiguiente, el puerto “3130” será el puerto que se indique en la configuración de hosts para el acceso a internet.

b) Access Control (ACLs)

En esta opción se establecen las listas de acceso (ACLs) que deseamos sean tomadas en consideración por squid. Esta función de Webmin, permite la asignación de una forma grafica a la tradicional instrucción de “acl nombre_acl tipo_de_acl descripción” que se incluye en el archivo de configuración “/etc/squid/squid.conf”. Igualmente se tiene el control sobre los operadores-acls, establecidos por las instrucciones http_access.

c) Logging (Registros)

Una parte importante en el funcionamiento y gestión de un servidor squid corresponde a los registros de acceso. En squid, el registro de toda solicitud, y la acción tomada por squid se registra en el archivo “/var/log/squid/access.log”. Esta información nos permite depurar las acciones tomadas por squid, y a la vez de contar con la información de minería para entender y conocer los hábitos de navegación de los usuarios.

La configuración que se realizara, permitirá la navegación a aquellos computadores de la empresa y los cuales se encuentren registrados en los accesos concedidos por el servidor squid. Sin embargo, estarán sujetos al control de contenido que evite el acceso a páginas cuya naturaleza sea de sexo, violencia entre otras.

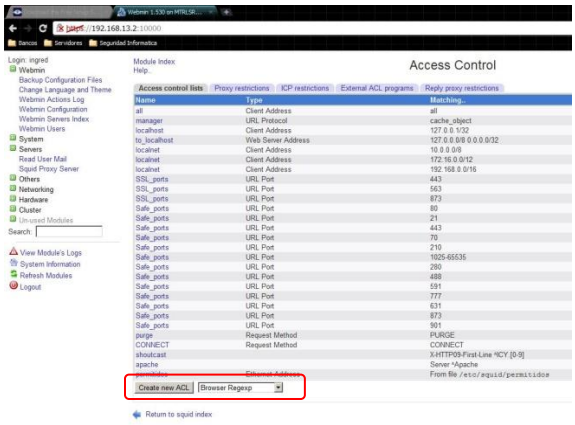
Considerando que los terminales de la red de Metrolínea se encuentran conectados a Switches de capa 2 y debido a que estos pueden llegar a sufrir perdida de energía, forzando a reasignar direcciones IP. Se emplearan las direcciones MAC de los equipos para construir el listado de clientes o host permitidos para navegar en internet.

Con la finalidad de atender el requerimiento de control de acceso a internet se opta por emplear el uso de direcciones MAC para asignar una lista de control principal, bajo el archivo “/etc/squid/permitidos”. Todo host que se encuentre bajo la lista “permitidos” tendrá autorización para navegar, pero estará regido por las normas de control de contenido. Por otro lado, aquellos que no se encuentren en la lista “permitidos” la conexión será inmediatamente rechazada por squid.

Para efectos del ambiente simulador o virtual que fue configurado se tendrán dos terminales. El primero identificado con MAC 00:1F:3C:45:92:F9 representara al grupo de computadores permitidos para la navegación en internet. Se interpreta como un computador de Metrolínea, sin embargo, estará regido por el control de contenidos de squid. Significa que podrá navegar, pero no en aquellos sitios web que se encuentren restringidos. El segundo, será un host con MAC 3C:F7:2A:57:A0:53 y será el host “no permitido” en la red de Metrolínea. Como se estableció, este será automáticamente rechazado por squid para la navegación, al no encontrarse en el listado “/etc/squid/permitidos”.

Se deberá acceder por webmin a la siguiente ruta “Servers->Squid Proxy Server->Access Control. Una vez localizados en el área de “Access Control” se seleccionara en la parte inferior de listado la opción “Etherner Address” que se encuentra disponible el combo box y se da click en la botón de “Create New ACL”. Tal y como se indica en la siguiente figura.

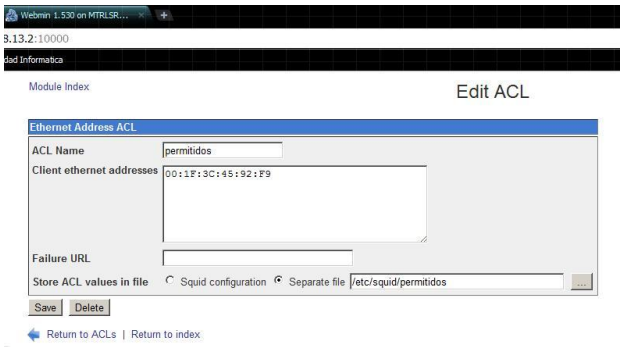
Figura 31. Configuración ACL Permitidos



Fuente: Autores

En las opciones que se presentan después de presionar “Create new ACL” es donde se configuraran cada uno de los clientes que estarán autorizados para acceder a internet. En el caso particular de este ejercicio se configuran las opciones que se observan en la figura siguiente, y que corresponden a nuestro cliente con MAC 00:1F:3C:45:92:F9.

Figura 32. Agregar Clientes Permitidos

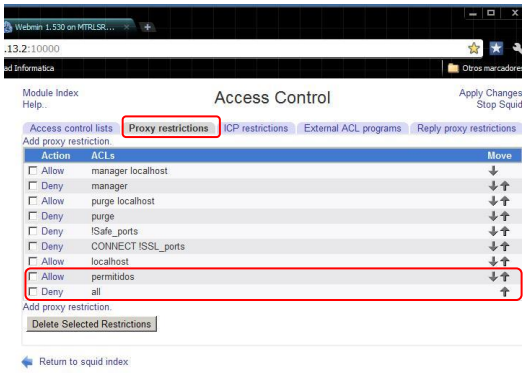


Fuente: Autores

Una vez creada la ACL, se deberá habilitar su uso, y esto se realiza bajo la misma ruta de webmin “Servers->Squid Proxy Server->Access Control, pero se

selecciona de la pagina la pestaña descrita como “Proxy Restrictions”. Continuamos a dar click en “Add Proxy Restriction” y se selecciona la opción “permitidos” y en la segunda columna “allow”. Tal y como se describe en la figura a continuación:

Figura 33. Restricciones del Proxy

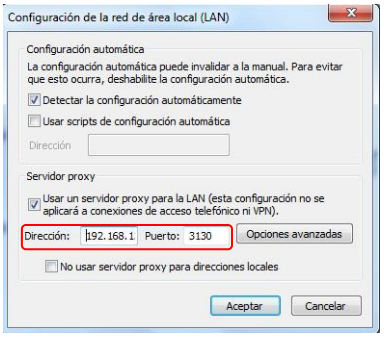


Fuente: Autores

Las acciones previas han colocado la lista “permitidos” como “allow”, que significa que permite conexiones. Sin embargo, es importante recordar que la lista “permitidos” debe quedar antes de la sentencia “deny all”. De lo contrario, todas las conexiones serán rechazadas.

Para probar que squid esté realizando control de acceso, se configuran ambos clientes para que empleen navegación vía proxy. En clientes Windows esta opción puede ser cambiada entrando a internet explorar y siguiendo la ruta Herramientas->Opciones de Internet->Conexiones->Configuración LAN->Servidor Proxy.

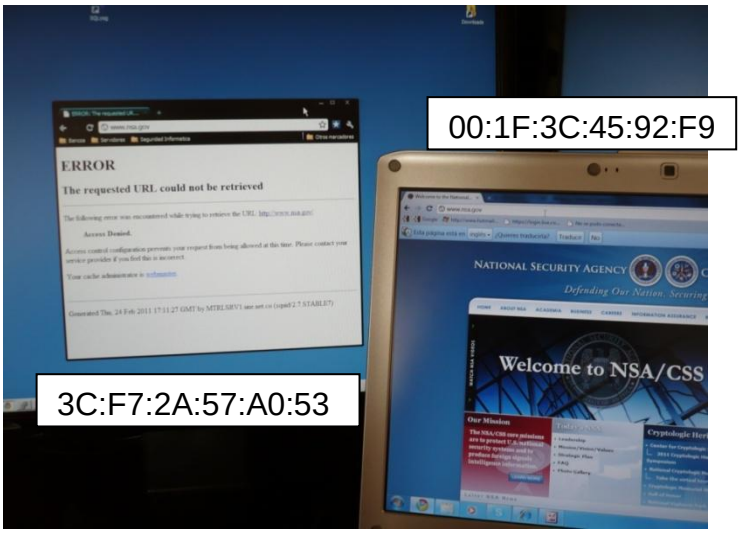
Figura 34. Configuración Proxy Windows



Fuente: Autores

Ambos clientes de prueba navegaran hacia la página www.nsa.gov y tan solo uno de ellos podrá acceder a la pagina. Mientras uno de ellos recibe una página de error de squid.

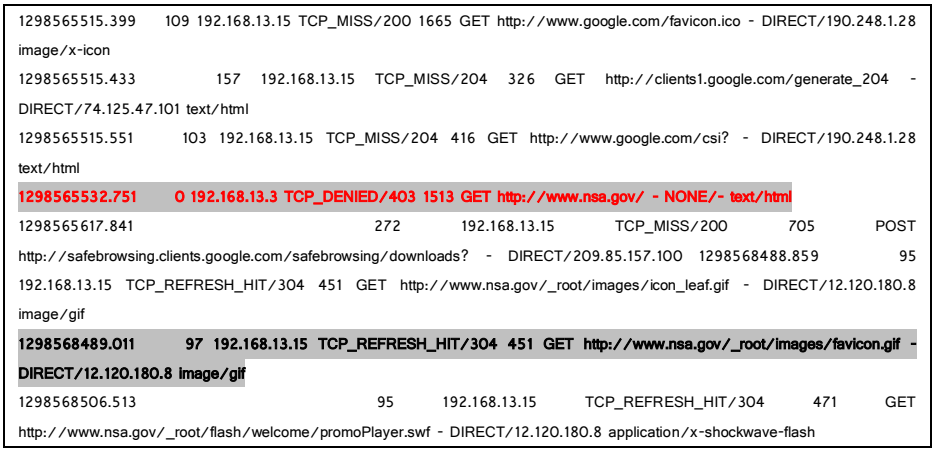
Figura 35. Conexiones a Internet (Via Proxy)



Fuente: Autores

La prueba de conexión o rechazo se puede evidenciar si se revisa el archivo encontrado en “/var/log/squid/access.log”. En donde se pueden identificar todos los tipos de conexiones provenientes de los clientes.

Figura 36. Archivo Access.log



Fuente: Autores

Si bien se ha comprobado el control de acceso, es necesario establecer el control de contenido. El control de contenido se encargara de restringir el acceso a paginas cuya insinuación sea de sexo, violencia o otra en no conformidad con las políticas de la empresa.

Como se describió en los apartados anteriores, squid cuenta con una gran cantidad de opciones que permite restringir acceso a páginas y/o contenidos. Los sitios o características a ser restringidas pueden ser configurados de forma manual por medio de un archivo plano de texto. Sin embargo, esta práctica es obsoleta y no cumple con los mínimos requerimientos para el control de contenido de una empresa. Existen varias alternativas, como el bajar lista de sitios restringidos, conocidos como “blacklist” o “listas negras”, e indicar al servidor squid que niegue el acceso a estos sitios. Estas listas pueden ser gratis o comerciales, entre mayor el número de sitios registrados contenga la lista, mejor será para la

empresa; sin mencionar el inmenso beneficio de no alimentar estas listas de forma manual.

En Linux existen dos programas que sirven para ampliar las capacidades de squid, a la vez que simplificando su uso para la administración del contenido que se desea restringir. Se pueden considerar como plugins para squid, entre los más reconocidos se encuentra DANSGUARD y SQUIDGUARD.

SquidGuard complementa a squid para que toda las peticiones que se realizan al servidor sean pasadas al modulo de Squidguard, y este funcione como control de contenido. Es decir, si una persona solicita una página, cuyo contenido sea de sexo, por ejemplo: “www.playboy.com”, squidguard automáticamente niegue dicha solicitud.

Con el objetivo de cumplir con el objetivo de brindar un adecuado control de contenidos para squid, se procede a instalar “Squidguard” en la maquina virtual, y configuramos para que funcione como aplicación de soporte a squid. Se ejecutan los siguientes comandos en la terminal:

sudo apt-get install squidguard

Una vez instalado, deberemos indicar al servidor squid que squidguard será el encargado de funcionar como control de contenidos. Se debe tener presente que existen dos (2) archivos que regulan el comportamiento de squidguard y estos corresponden a los siguientes:

Tabla 9. Archivos Configuracion Squidguard

ARCHIVO DE CONFIGURACION	DESCRIPCION
/etc/squid/squidGuard.conf	Archivo de configuración de squidguard

/var/lib/squidguard/db	Base de datos o lugar donde residen las listas negras o blacklist controladas por squidguard.

Fuente: Autores

Como se menciono previamente, el éxito de un control de contenidos se encuentra en las listas de sitios restringidos o listas negras. Existen varios sitios que brindan estas listas de forma gratuita, entre los más conocidos se encuentran:

- Shallas Blacklist: Contiene aproximadamente una lista de 1.7 millones de sitios registrados, cada sitio contenido dentro de varias categorías. Se encuentra libre para uso siempre y cuando sean para propósitos no comerciales y/o personal.
- Université Toulouse blacklist: Listado de aproximadamente 1.5 millones de sitios.
- URLBlacklist: Es una lista comercial, que contiene aproximadamente 2.0 millones de sitios registrados. Cuenta con actualizaciones periódicas.

Con lo anterior en mente, es necesario descargar la lista de alguno de estos sitios. Para la configuración de la maquina virtual emplearemos la lista dispuesta en Shillas Blacklist; la cual será suficiente para los propósitos demostrativos.

Ejecutamos la descarga y cambiamos los permisos de los archivos para que puedan ser leídos por squidguard. Se emplean los siguientes comandos en el terminal:

sudo -i

cd /var/lib/squidguard/db

wget –c http://www.shallalist.de/Downloads/shallalist.tar.gz

tar –xvzf shallalist.tar.gz

chown –R proxy:proxy /var/lib/squidguard/db

chmod –R 755 /var/lib/squidguard/db

Hasta el momento solo se ha cargado la lista, distribuida en diferentes categorías. Esto se aprecia al realizar un listado del directorio /var/lib/squidguard/db:

Figura 37. Listado de /var/lib/squidguard/db

```
drwxr-x--- 2 proxy proxy 4,0K 2011-02-23 15:12 adv
drwxr-x--- 2 proxy proxy 4,0K 2010-10-10 02:17 aggressive
drwxr-x--- 2 proxy proxy 4,0K 2011-02-14 17:07 alcohol
drwxr-x--- 2 proxy proxy 4,0K 2011-02-02 18:45 anonvpn
drwxr-x--- 6 proxy proxy 4,0K 2011-01-20 17:26 automobile
drwxr-x--- 2 proxy proxy 4,0K 2011-02-19 16:35 chat
drwxr-x--- 2 proxy proxy 4,0K 2011-02-23 04:46 costtraps
drwxr-x--- 2 proxy proxy 4,0K 2011-02-14 16:29 dating
drwxr-x--- 2 proxy proxy 4,0K 2011-02-23 15:17 downloads
drwxr-x--- 2 proxy proxy 4,0K 2011-02-23 16:07 drugs
drwxr-x--- 2 proxy proxy 4,0K 2010-05-24 11:14 dynamic
```

Fuente: Autores

Ahora es necesario realizar unos ajustes en el archivo de configuración de squid. Como se menciono, este reside en /etc/squid/squid.conf. Para realizar cambios de manera más rápida introducimos los siguientes parámetros al archivo de configuración de squid por medio de un editor de textos.

url_rewrite_program /usr/bin/squidGuard –c /etc/squid/squidGuard.conf
http_access allow localnet
visible_hostname localhost
#http_access allow permitidos

Finalmente, se edita el archivo /etc/squid/squidGuard.conf para vincular las categorías de las listas negras que fueron descargadas. Squidguard se encargara de administrar todas las solicitudes. En el caso particular de nuestro alcance se

tiene un estructuración del archivo squidGuard.conf de la siguiente forma:

Figura 38. Estructura Archivo squidGuard.conf

```
src permitidos {
    ip 192.168.13.15
}

dest porn {
        domainlist porn/domains
        urllist porn/urls
}

dest violence {
        domainlist violence/domains
        urllist violence/urls
}

acl {
    permitidos {
    pass !porn !violence all
}

    default {
    pass none
    redirect http://www.metrolinea.gov.co
}

}
```

Fuente: Autores

Si se observar el log de squidguard “/var/log/squid/squidGuard.log” se obtienen los siguientes datos:

Figura 39. Log /var/log/squid/squidGuard.log

```
2011-02-24 20:50:14 [9198] init domainlist
/var/lib/squidguard/db/porn/domains
2011-02-24 20:50:14 [9198] loading dbfile
/var/lib/squidguard/db/porn/domains.db
2011-02-24 20:50:14 [9199] init domainlist
/var/lib/squidguard/db/porn/domains
2011-02-24 20:50:14 [9199] loading dbfile
/var/lib/squidguard/db/porn/domains.db
```

```
2011-02-24 20:50:14 [9199] init urllist
/var/lib/squidguard/db/porn/urls
2011-02-24 20:50:14 [9199] loading dbfile
/var/lib/squidguard/db/porn/urls.db
2011-02-24 20:50:14 [9199] squidGuard 1.2.0 started
(1298598614.924)
2011-02-24 20:50:14 [9199] squidGuard ready for requests
(1298598614.927)
2011-02-24 20:50:14 [9198] init urllist
/var/lib/squidguard/db/porn/urls
2011-02-24 20:50:14 [9198] loading dbfile
/var/lib/squidguard/db/porn/urls.db
2011-02-24 20:50:14 [9198] squidGuard 1.2.0 started
(1298598614.923)
2011-02-24 20:50:14 [9198] squidGuard ready for requests
(1298598614.928)
2011-02-24 20:50:14 [9195] init urllist
/var/lib/squidguard/db/porn/urls
2011-02-24 20:50:14 [9195] loading dbfile
/var/lib/squidguard/db/porn/urls.db
2011-02-24 20:50:14 [9195] squidGuard 1.2.0 started
(1298598614.915)
2011-02-24 20:50:14 [9195] squidGuard ready for requests
(1298598614.929)
```

Squidguard permitirá la navegación a las personas que se encuentran en la categoría de “permitidos”, siempre y cuando la dirección de destino no corresponda a las contenidas en la lista negra que se bajo de Shillas Blacklist. Cualquier otra conexión será redireccionada a la página www.metrolinea.gov.co.

La ventaja de utilizar squidguard es la facilidad con la que se configura un archivo para el control de sitios y de accesos; a diferencia a la edición del archivo “/etc/squid/squid.conf” ya que este archivo se encuentra sumamente poblado por comentarios.

6.3. INSTALACION DE SERVIDOR SAMBA

Gracias al fuerte administrador de paquetes contenido en la distribucion Ubuntu, logramos realizar la tarea de instalación por medio del comando

sudo apt-get install samba smbfs

Al igual que squid, encontraremos el archivo de configuración en la carpeta “/etc”, pero específicamente en el archivo “/etc/samba/smb.conf”. La configuración del

presente archivo se realizara por medio de manipulación directa sobre los archivos de configuración, puesto que Webmin no posee un modulo que permita este tipo de configuración de forma visual.

6.3.1. CONFIGURACION DE SERVIDOR SAMBA

El propósito de la presente configuración es establecer tres (3) áreas dentro del servidor Ubuntu (Linux) para que se ofrezcan como recursos compartidos dentro de una red Windows. La finalidad será crear un arreglo de usuarios que tengan características y permisos específicos para cada una de estas áreas ofrecidas.

Tabla 10. Usuarios Windows

USUARIO	DEPENDENCIA
Rgonzalez	Jurídica
Ocaselles	Operaciones
sistemas	Sistemas
colaya	Jurídica
lgualdron	Operaciones

Fuente: Autores.

Se procede a crear primeros los usuarios en el servidor linux, los cuales posteriormente se emparejaran con el nombre de usuarios de las máquinas Windows de donde accederán. Es así como se procede crear los usuarios y agregarlos a un grupo específico de usuarios dentro de linux.

Cada usuario pertenece a una dependencia, en la cual tan solo los miembros de esa dependencia podrán ver y escribir los documentos que se encuentran compartidos bajo el recurso de la dependencia a la que pertenecen. A excepción de un recurso compartido que se llamara “intercambio” en los cuales todos los usuarios tendrán acceso de lectura y escritura. De igual forma el usuario sistemas tendrá la propiedad de escribir en cualquiera de las carpetas compartidas.

Por medio del comando “adduser” se crean los usuarios, todos ellos tendrán asignada la clave “12345”. Y posteriormente con el comando addgrp crearemos las dependencias a los cuales pertenece cada usuario.

Tabla 11. Comando crear usuario y grupos

COMANDO PARA USUARIO	COMANDO PARA GRUPOS
adduser rgonzalez	*addgroup operaciones*
adduser ocaselles	*addgroup jurídica*
adduser sistemas	*addgroup sistemas*
adduser colaya	
adduser lgualdron	

Fuente: Autores.

Hay dos instancias que se han de realizar. La primera generar los usuarios y los grupos. Y asignar los correspondientes usuarios a los diferentes grupos. Segundo corresponde a la creación de las carpetas que serán compartidas, con los correspondientes permisos de archivos. Y finalmente la parametrizacion de los valores en el archivo de configuración de samba.

Para vincular los usuarios a los diferentes grupos se emplean los siguientes comandos:

adduser rgonzalez jurídica

adduser ocaselles operaciones

adduser lgualdron operaciones

adduser sistemas operaciones

adduser sistemas juridica

Para poder albergar los archivos dentro del sistema linux, nos dirigimos a la carpeta “/” y se crea un directorio “/sambafiles”. Dentro de esta carpeta creamos

los directorios “jurídica”, “operaciones”, “sistemas” y “intercambio”. Empleando los siguientes comandos asignando los permisos de las carpetas.

Chmod g+rwx jurídica

Chmod o-rwx jurídica

Chmod g+rwx sistemas

Chmod o-rwx sistemas

Chmod g+rwx operaciones

Chmod o-rwx operaciones

Chmod a+rwx intercambio

Chgrp jurídica jurídica

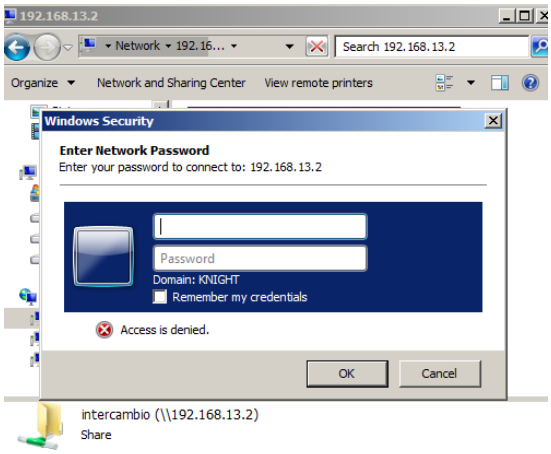
Chgrp operaciones operaciones

Chgrp sistemas sistemas

El fuerte de linux se encuentra en el estricto control sobre los permisos que cada archivo o directorio tiene, y la interrelación que estos usuarios contienen. Es así, como editando “/etc/samba/smb.conf”, se logra compartir en una red las diferentes carpetas, todas centralizadas en una maquina. Disposición que facilita la realización de copias de seguridad.

Se procede a vincular los usuarios a una clave del servicio samba, para lo cual se recomienda que la clave del usuario samba, sea la misma que se asigno al usuario Linux. Y similarmente para la sesión de usuarios Windows. De lo contrario siempre se presentara un login de usuario y password.

Figura 40. Login/Password acceso a share samba



Fuente: Autores

Se ejecutan los siguientes comandos para crear los usuarios en samba.

sudo smbpasswd -a rgonzalez

sudo smbpasswd -a ocaselles

sudo smbpasswd -a lgualdron

sudo smbpasswd -a colaya

sudo smbpasswd -a sistemas

Figura 41. Configuracion /etc/samba/smb.conf

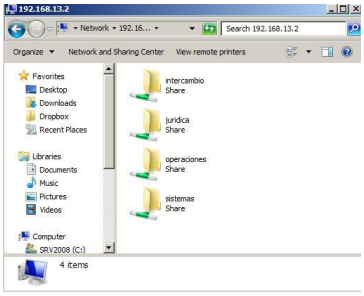
```
[global]
        workgroup = METROLINEA
        server string = %h server (Samba, Ubuntu)
        map to guest = Bad User
        obey pam restrictions = Yes
        pam password change = Yes
        passwd program = /usr/bin/passwd %u
        passwd chat = *Enter\snew\s*\spassword:* %n\n *Retype\snew\s*\spassword:*
%n\n *password\supdated\ssuccessfully* .
        unix password sync = Yes
        syslog = 0
        log file = /var/log/samba/log.%m
        max log size = 1000
```

```
        dns proxy = No
        usershare allow guests = Yes
        panic action = /usr/share/samba/panic-action %d

[printers]
        comment = All Printers
        path = /var/spool/samba
        create mask = 0700
        printable = Yes
        browseable = No
        browsable = No

[print$]
        comment = Printer Drivers
        path = /var/lib/samba/printers

[INTERCAMBIO]
        comment = Para uso de todos los usuarios
        path = /opt/samba/intercambio
        read only = No
        directory mask = 0775

[JURIDICA]
        comment = Juridica
        path = /opt/samba/juridica
        read only = No

[OPERACIONES]
        comment = Operaciones
        path = /opt/samba/operaciones
        read only = No

[SISTEMAS]
        comment = Sistemas
        path = /opt/samba/sistemas
        read only = No
```

Fuente: Autores

Para corroborar que samba está ofreciendo las carpetas que se crearon anteriormente. Empleamos un cliente Windows, y se digita la dirección \\192.168.13\2 en el explorador de archivos de Windows.

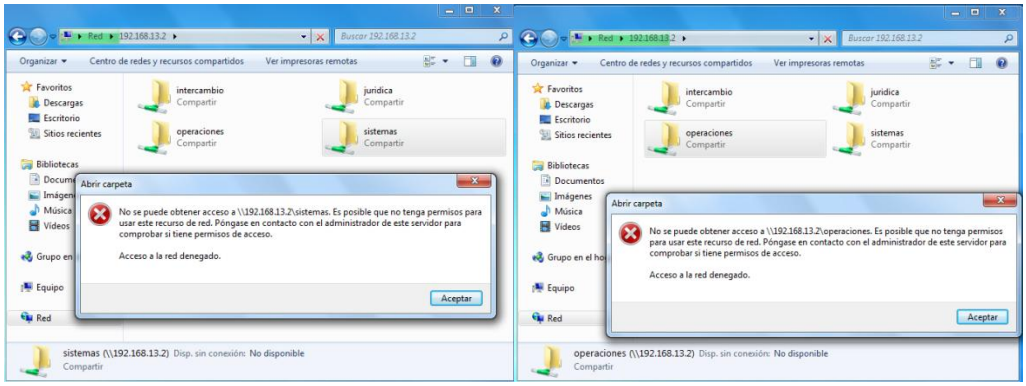
Figura 42. Carpetas Linux Visibles en Windows (Via Samba)



Fuente: Autores

Cada carpeta puede ser visualizada, tan solo por el usuario que pertenece al grupo de la carpeta. Es decir, solo los usuarios pertenecientes a jurídica pueden ver el share “jurídica”. Si un usuario del grupo jurídica intentara abrir las carpetas share de “operaciones” o “sistemas”, recibiría un acceso denegado, tal y como se ilustra en la figura.

Figura 43. Usuario de Grupo Jurídica - Acceso Denegado



Fuente: Autores

La ventaja de tener un servidor samba es el contar con un servicio de Windows, pero con los controles de Linux. Esto genera un punto central de almacenamiento, para la creación de copias de seguridad.

CONCLUSIONES

El empleo de herramientas de diagnostico permitieron realizar una correcta apreciación sobre el estado actual de la red informática de Metrolínea, permitiendo la propuesta de servicios linux que permitan mejorar las condiciones de uso y preservación de la información.

La propuesta de emplear sistemas operativos linux para gestionar servicios específicos de la red, se presenta como una realidad efectiva en la relación cost/beneficio. Con la actual ventaja de poder ser ensayados en ambientes aislados, antes de pensar en la implementación.

La configuración de servicios linux, revela la necesidad de generar un mejor planeamiento de los recursos actuales. Puesto que la administración, e inventario de la misma no tienen ningún tipo de control, haciéndola propicia a adquirir problemas significativos de seguridad.

BIBLIOGRAFÍA

[1] COLEGIO NARIÑO, “Rincón Tecnológico”. [en línea]
<http://fundacioncolegionarino.com/rincon.html>, [citado el 20 de agosto de 2009]

[2] IBM. “Anatomy of The Linux Kernel”, [en línea]
<http://www.ibm.com/developerworks/linux/library/l-linux-kernel/index.html>, [citado 20 de agosto de 2009]

[3]LINUX ZONE. “Linux Distributions”. [en linea]
< http://www.linuxzone.es/distribuciones-principales/debian/>, [citado 20 de agosto de 2009]

[4] http://mabavi.wordpress.com/linux-carpetas-del-sistema-ubuntu/

[5] WIKIPEDIA, “Cliente-Servidor”, [en línea]
<http://es.wikipedia.org/wiki/Cliente-servidor>, [citado el 26 agosto de 2009]

[6] KIOSKEA.NET, “Protocolo HTTP”, [en línea]
<http://es.kioskea.net/contents/internet/http.php3>, [citado el 26 agosto de 2009]

[7] KIOSKEA.NET, “Protocolo FTP”, [en línea]
<http://es.kioskea.net/contents/internet/ftp.php3>, [citado el 26 agosto de 2009]

[8] KIOSKEA.NET, “Protocolo SMTP”, [en línea]
<http://es.kioskea.net/contents/internet/smtp.php3>, [citado el 26 agosto de 2009]

[9] KIOSKEA.NET, “Protocolo SSH”, [en línea]
<http://es.kioskea.net/contents/internet/ssh.php3>, [citado el 26 agosto de 2009]

[10] ADMINISTRACION DE LINUX, “Control de Acceso a la Web”, [en línea] <http://dns.bdat.net/documentos/squid/x30.html>, [citado el 02 octubre de 2010]

[11] WIKIPEDIA, “Samba”, [en línea] <http://es.wikipedia.org/wiki/Samba_(programa)>, [citado el 12 octubre de 2010]

[12] CURSO DE INTRODUCION A GNU/LINUX, “Permisos de Archivos”, [en línea] <http://www.ant.org.ar/cursos/curso_intro/x1439.html>, [citado el 02 febrero de 2011]