ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"IMPLEMENTAR Y FORTALECER REDES COMPUTACIONALES MEDIANTE HERRAMIENTAS OPEN SOURCE PARA LAS PYMES EN ECUADOR"

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

LICENCIADO EN REDES Y SISTEMAS OPERATIVOS

ALLAN MARIO VEINTIMILLA VALVERDE

DARWIN ALEJANDRO MANJARREZ FAJARDO

GUAYAQUIL – ECUADOR

AÑO 2015

AGRADECIMIENTOS

Mis más sinceros agradecimientos a Dios por darme vida y por haber guiado mi camino en el cual he formado mi carácter, a mi madre mostrarme el significado del amor incondicional, a mi padre por su sabiduría y consejos a lo largo mi vida, a mi hermano por ser mi espejo y cuidar a la familia, a mi hijo por demostrarme que día a día se aprende cosas nuevas, a mi esposa por estar conmigo en este camino de ahora en adelante.

Allan Mario Veintimilla Valverde.

Mis más sinceros agradecimientos a Dios por bendecirme y otro muy especial para mis tías por la comprensión, paciencia y el ánimo recibidos.

Darwin Alejandro Manjarrez Fajardo

DEDICATORIA

El presente proyecto lo dedico a mi madre y a mi padre quienes han estado desde el comienzo de mi vida, quienes me han guiado con esmero y ahínco, colocando en mí la confianza para poder seguir mis metas y por la cual culminarla, aparte a mí esposa y a mi hijo demostrándole que todo se puede en la vida si uno se lo propone.

Allan Mario Veintimilla Valverde

A mis tías quienes me apoyaron todo el tiempo.

Darwin Alejandro Manjarrez Fajardo

TRIBUNAL DE EVALUACIÓN

Msc. José Roberto Patiño Sánchez	**Msc. Rayner Stalyn Durango Espinoza**
PROFESOR EVALUADOR	PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual."

.....

Allan Mario Veintimilla Valverde

.....

Darwin Alejandro Manjarrez Fajardo

RESUMEN

El presente proyecto integrador nos muestra y proporciona la importancia de su uso para integrar la idea de seguridad en las infraestructuras de redes en las pequeñas y medianas empresas, analizando vulnerabilidades de seguridad general que un administrador de redes y sistemas puede encontrar a lo largo de su carrera profesional, enfocado en recomendaciones en la seguridad de las redes computacionales y fortalecimiento en los sistemas.

Además tiene como objetivo el brindar buenas prácticas para la protección de entornos de sistemas operativos basados en GNU/Linux, tomando como punto de partida el modelo de defensa en profundidad, es decir asegurar los sistemas computacionales por capas. Dicho proyecto menciona también las debilidades de entornos generales, los cuales a partir de ese punto son asegurados y mostrados en la implementación y funcionamiento para mantener el entorno lo más seguro posible.

En definitiva este trabajo se recomienda a todos aquellos que deseen reforzar o asegurar sus sistemas computacionales e infraestructuras con herramientas de acceso o distribución libre, así como para los que necesiten una base desde la que partir a la hora de fortalecer un entorno Open Source.

ÍNDICE GENERAL

CAPÍTULO 1

1. ANTECEDENTES

A finales de 2014, el número de conexiones a Internet a nivel mundial estaba estimado en aproximadamente 3000 millones de usuarios. Dos tercios del total de los usuarios con acceso a internet, son personas de los países en desarrollo. El internet no es una red única, sino una colección de redes conectadas remotamente que son accesibles por miles de computadoras individuales, en una gran variedad de maneras y pueden tener accesos a todo tipo de información con tan solo una computadora y una conexión de red [1].

Por lo tanto, las personas y las organizaciones pueden llegar a cualquier punto de la Internet sin tener en cuenta las fronteras nacionales o geográficas o incluso la hora del día. Sin embargo, junto con la comodidad y el fácil acceso a la información llegan también los riesgos. Entre ellos se encuentran los riesgos de que la información importante se pierda, sea cambiado, robado o tal vez mal utilizada. Si la información se registra electrónicamente y está disponible en equipos en red, es más vulnerable que si la misma información está impresa en papel y almacenado en un archivador. Los intrusos no necesitan entrar en una oficina y manipular física o directamente las computadoras o servidores de la empresa. Ellos pueden robar u obtener información sin tocar un pedazo de papel. También pueden crear nuevos archivos electrónicos y ocultar las pruebas de su actividad no autorizada.

Y a la hora de administrar una empresa resulta indispensable conocer en todo momento las actividades de la misma, dando así una visión global de lo que sucede en su entorno, más cuando están en pleno proceso de invocación de productos o servicios, dado que es una necesidad para el crecimiento empresarial, y conjuntamente con la incorporación de las TIC, agentes externos pueden logran un mayor acceso a dicha información [2].

La necesidad de empezar a tomar en cuenta las nuevas plataformas y tecnologías para el crecimiento de las organizaciones privadas o públicas siempre está sujeta a su activo más importante, la información que almacenan.

En cuando más van creciendo las empresas en el mercado, mas es el interés de mantener reservada la seguridad de la misma, tanto física como lógica, creando así protocolos, políticas, estándares de accesos, etc.

Ante este punto de vista la concienciación sobre lo importante que es la seguridad informática en los empleados, directores o jefes de áreas e inclusive los propios gerentes de las organizaciones es un punto trivial hasta la actualidad. Y que la mejor forma para comenzar a mostrarle el interés debido, es que sean víctimas de un tipo de ataque informático.

Sin duda un ejemplo de que ni las organizaciones más sólidas en tema de tecnología y comunicaciones están exentos de ataques informáticos, está el caso del ataque ocurrido entre mediados y finales del año 2009 a un grupo de empresas mundiales entre ellas *Google*, *Adobe Systems*, *Juniper Network*, entre las empresas con mayor notoriedad, el cual de acuerdo a los datos recabados por investigaciones, se enfoca que el objetivo principal fue realizado con el fin de obtener de información de propiedad intelectual de las empresas e acceder a información almacenadas en los servidores de correo de Gmail de activistas de derechos humanos en China. A este ataque fue bautizado como "*Operación Aurora*" en el cual fue realizado por una serie de ataques de APT (de las siglas en inglés *Advanced Persistent Threats*) adjuntado al *Grupo Elderwood* que residían en Beijín, China.

El proceso consistía en un conjunto de ataques informáticos de forma continua, el cual requería un alto grado de cuidado y cautela, debido al tiempo que toma la búsqueda constante de vulnerabilidades en los sistemas, mediante el uso de perfeccionadas técnicas y uso de software malicioso, que apunta a que durante todo el proceso se mantiene un monitoreo y control de las acciones y datos extraídos en lo que dura el ataque [3] [4].

1.1. Seguridad Física del Entorno.

Para el tema de seguridad es necesario saber que por más que nuestra empresa sea la más segura desde el punto de vista de ataques externos, intrusos, virus, denegación de servicios, etc., la seguridad estaría comprometida en el caso de una eventualidad antrópica sino la tienen prevista

como por ejemplo un desastre natural (inundación, variación de energía, incendios).

En primer lugar, se puede visualizar la seguridad física como el procedimiento mediante el uso de cámaras, guardias de seguridad, Centro de Procesamiento de Datos aislados y asegurados que protegen las distintas capas o el acceso a sus contenedores. En el segundo lugar, la seguridad o protección física se podría entender como los mecanismos que son los utilizados para asegurar los sistemas o la información del acceso físico a un medio digital por parte de un usuario.

La utilización de sistemas de vigilancia, cámaras, guardas que protejan los datos de la empresa, es de vital importancia. Pero en el presente documento lo que compete es la segunda visión de seguridad física. La utilización de mecanismos que eviten que un usuario con acceso físico a los equipos pueda realizar cualquier tarea con ellos.

Hay que recalcar que cuando un usuario dispone de acceso físico a un equipo, en la mayoría de los casos, será difícil evitar que pueda utilizar el sistema para realizar tareas de dudosa moral. Más adelante en el documento se podrá estudiar mecanismos para evitar que un usuario pueda arrancar un sistema operativo si él no está autorizado, que pueda llegar a ser *root* con suma facilidad o proteger la información sensible mediante el uso del cifrado.

1.2. Seguridad Lógica.

La seguridad lógica se refiere al proceso de utilizar técnicas basadas en *software* para la autenticación de los privilegios de un usuario en una red informática específica o sistema. El concepto es una parte del campo más completa de la seguridad informática, que implica ambos métodos de hardware y software para asegurar un terminal o la red. Cuando se habla de la seguridad lógica, se debe considerar las diferentes técnicas utilizadas, que incluyen nombres de usuario y contraseñas, *token* de seguridad y autenticación de dos vías en un sistema.

Autenticación de contraseña es quizás el tipo más común y familiar de la seguridad lógica. Cualquiera que haya usado alguna vez un sitio de banca en línea o incluso un sistema de red social estará familiarizado con este concepto. Cuando una red se ha configurado para utilizar la autenticación de contraseña, los usuarios que intentan conectarse a un terminal específico en la red son los primeros obligados a demostrar sus credenciales mediante la introducción de un nombre de usuario y contraseña. La ventaja principal aquí es la simplicidad; los usuarios necesitan nada más que su nombre de usuario y la contraseña memorizada información para acceder al sistema. Una desventaja importante es que el equipo no tiene forma de comprobar si la persona que utiliza un nombre de usuario específico y una contraseña es el usuario autorizado; Por lo tanto, los usuarios sin escrúpulos pueden robar nombres de usuario y contraseñas para romper el sistema.

1.3. Sistemas Operativos.

El primer Sistema Operativo oficialmente lanzado data de los años 1970, el cual fue nombrado *UNIX* por la empresa de *Laboratorios Bell*, el cual fue ejecutado en una computadora *PDP-11* fabricada por la empresa *Digital Equipment Corp*, el cual era la primera de este tipo de computadoras en interconectar todos los elementos del sistema (Procesamiento, Periféricos y memoria) el cual tenía como características técnicas, bus de datos de 16bit bidireccional, el cual permitía recibir y enviar datos sin pasar por la memoria *RAM* [6] [7].

Fue entonces que la explotación comercial de los sistemas operativos comenzada a tener un lugar en mercado de la década de los 80 el cual estaba bajo restricciones de licencia de AT&T, posterior el sistema operativo fue incluido en las universidades de Estados Unidos para su desarrollo con fines educativos.

No fue hasta 1991 cuando Linus Torvalds comenzó Linux, un núcleo del sistema operativo, como un proyecto personal. Puso en marcha el proyecto porque quería ejecutar un sistema operativo basado en Unix sin gastar mucho dinero. Linux fue lanzado de forma gratuita al público para que cualquier

persona podía estudiarlo y hacer mejoras bajo la Licencia Pública General. Fue entonces así que aparece el sistema operativo denominado GNU/LINUX que consistía en una versión mejorada de UNIX, el cual está basado a un reemplazo no comercial compuesto principalmente de software libre. Posterior al lanzamiento con el pasar de los años fueron lanzadas múltiples distribuciones del GNU/LINUX que estaban enfocado a satisfacer las necesidades de ciertos usuarios, por ejemplo distribuciones de sistemas operativos para entornos de escritorio, sistemas operativos para entornos de trabajo y sistemas operativos para entornos de servidores [8].

Una de las principales cualidades que tiene el sistema operativo de GNU/LINUX para ser tomado como entornos para fortalecimiento es que tiene implementado un sistema de permisos, el cual no permite ejecutarse ninguna aplicación sin los permisos otorgados por los usuarios, también la disponibilidad de poder configurar todos los ficheros de directorios de sistema.

Hoy, Linux se ha convertido en un punto importante también en el mercado de sistemas operativos. Se ha adecuado para poder correr en una variedad de arquitecturas de sistemas, incluyendo Alfa HP / Compaq, SPARC de Sun y UltraSPARC, y los chips PowerPC de Motorola. Se ejecuta programas como Sendmail, Apache, y BIND, DNS que son software muy popular que se usa para ejecutar los servidores en Internet. Es importante recordar que el término “Linux” en realidad se refiere con la palabra “*núcleo”,* es al núcleo del sistema operativo. Este núcleo se encarga de controlar el procesador, la memoria de su ordenador, discos duros y periféricos. Eso es todo lo que Linux hace realmente es controlar el funcionamiento de su equipo y se asegura el comportamiento de todos sus programas [9].

Varias compañías y personas empaquetan los programas del kernel y varios programas juntos para hacer un sistema operativo. A cada paquete es que nosotros le llamamos una distribución Linux, entre las cuales están las tres principales distribuciones:

1.3.1. Debian

El Sistema Operativo denominado Debian pertenece a las distribuciones de GNU/LINUX, el cual es de distribución libre y gratuita, Debian fue una de las primeras distribuciones en agregar el sistema de gestión de paquetes, el cual permite a su administrador un control de las instalaciones que se hacen en el sistema. Su sistema está enfocado a una alta estabilidad y permite que se un sistema escalable, lo que permite en su instalación la configuración de diversas funciones de acuerdo a las necesidades del usuario [10].

Sus características de requerimiento para ejecutar el sistema operativo son:

Hardware	Requerimientos Mínimos
Arquitectura	Amd64, i386, ARM, IBM/Motorola power PC, MIPS, Sum SPARC, IBM S/390
Procesador	Pentium 4, 2.8GHz
Memoria RAM	512 megabytes
Disco Duro	10 Gigabytes
Conexión a Internet	256 Kbps

Tabla 1. Requerimientos de S.O Debian

1.3.2. Slackware

El Sistema Operativo denominado Slackware pertenece a las distribuciones de GNU/LINUX, el cual es una de las distribuciones más antiguas creada por Linus Torvalds y disponibles hasta la actualidad, el sistema operativo por defecto un kernel 2.6 Linux, Slackware mantiene un gestor de sistema de gestión de paquetes basado en menú, mantiene ciertos entornos gráficos basado para usuario final que permite que sea de fácil uso y a la vez muy estable para la operatividad. Su sistema está enfocado en varios

entornos de trabajo que puede ser para servidores, puestos de trabajo y máquinas de escritorio [11].

Sus características de requerimiento para ejecutar el sistema operativo son:

Hardware	Requerimientos Mínimos
Arquitectura	Amd64, inter 80x86, ARMs, IBM/Motorola POWER PC, MIPS, Sum SPARC, IBM
Procesador	586 2.4GHz o superiores
Memoria RAM	512 megabytes
Disco Duro	5 Gigabytes
Conexión a Internet	No

Tabla 2. Requerimiento de S.O Slackware

1.3.3. Red Had

El Sistema Operativo denominado Red Had pertenece a las distribuciones de GNU/LINUX, el cual está basado a entornos para servidores, fue creado por la empresa Red Had Enterprise Linux el cual es una versión comercial de GNU/Linux que mantiene como principal característica la asistencia de soporte técnico dedicado para responder a las innovaciones de TI, servicios y seguridad, también presenta mejoras de escalabilidad en su sistema de archivos, la interoperabilidad y fiabilidad entre dominios como por ejemplo el Active Directory de Microsoft [12].

Sus características de requerimiento para ejecutar el sistema operativo actual son:

Hardware	Requerimientos Mínimos
Arquitectura	Amd64, inter 80x86, POWER, Itanium 2, <z<
Procesador	386 2.4GHz o superiores

Memoria RAM	1 GB
Disco Duro	8 Gigabytes
Conexión a Internet	No

Tabla 3. Requerimientos de S.O Red Hat.

1.4. Justificación

Viendo el beneficio que contribuye de añadir seguridad a nuestras empresas en desarrollo, la seguridad informática se puede separar libremente en dos partes. La primera es la seguridad lógica que tienen que ver con el acceso y uso de los datos y programas en el entorno de programación convencional. La segunda es la seguridad física que tiene que ver con el acceso a áreas no autorizadas fuera del entorno habitual por medios físicos. El paso dado para protegerse de un ataque o un fracaso son las medidas de seguridad que podemos tener. Incluyen controles de validación de datos de entradas, políticas de uso correcto de software, políticas de uso adecuado de las tecnologías de la información, limitación de acceso físico a la sala de equipos computaciones, políticas de contraseñas, políticas de internet, entre otros.

1.5. Metodología

Para la realización de este proyecto se usará como base el Sistema Operativo Debian GNU/Linux para establecer pilares para un sistema robusto. Para ello vamos a usar el Esquema de Defensa en Profundidad, que abarcan aspectos comprendidos desde la seguridad física hasta la seguridad en las aplicaciones, además para administrar el sistema es indispensable conocer en todo momento la actividad del mismo. Para ello vamos a utilizar la herramienta de seguridad, la cual nos va a servir para tener una vista de todos los aspectos relativos a la seguridad de los servidores.

Para lograrlo se estableció el desarrollo de algunas fases:

- Organizar y explicar conceptos en torno a un modelo de defensa basado en la profundidad y unos principios técnicos.

- Recopilar la información requerida para iniciar las instalaciones.
- Preparar la infraestructura de virtualización.
- Preparar las máquinas de Testeo.
- Configurar y fortificar estos entornos siempre orientado a entornos GNU/LINUX.

El Esquema de Fortificación de Sistemas, la Instalación y configuración de herramientas Open Source para la gestión eficiente de seguridad para las Pymes en Ecuador se los realizarán de forma virtualizada, donde restringiremos y monitorizaremos el sistema brindándole una seguridad tanto física, sistemas operativos, software y datos.

Implementaremos un esquema que representa una configuración de firewall de doble enlace y con DMZ, un servidor centralizado de log y una red local con 3 máquinas en un entorno de servidor-cliente con sistema operativo Debían 8, donde instalaremos parámetros de seguridad integral para un manejo de esquemas de seguridad.

Una vez de haber analizado las alarmas de las diferentes herramientas de monitorización y haber configurado e implantado fortalecimiento en el entorno procederemos a sacar nuestras respectivas conclusiones de los resultados de cada tipo de herramienta, usos, ventajas, desventajas, aplicaciones y limitaciones, para que finalmente poder establecer claramente recomendaciones y observaciones para futuros usos.

CAPÍTULO 2

2. IMPLEMENTACION

2.1. Recomendaciones previas

Desde un enfoque de seguridad informática, en esta capa debemos preocuparnos por evitar que personas no autorizadas tengan acceso a las instalaciones. Entre otras medidas, encontramos cámaras de vigilancia, guardas de seguridad, entre otras.

Otro peligro a examinar es la red interna, es decir, personal autorizado realizando acciones no permitidas. La colocación de los cables no debe permitir escuchas indebidas, para ello es necesario restringir acceso a la sala de telecomunicaciones y así proteger la electrónica de la red para que no puedan ser intencionalmente dañados cuyo objetivo es provocar un ataque directamente a la disponibilidad del servicio.

En este capítulo se van a presentar una serie de aseguramientos para tratar de mitigar o al menos retrasar el acceso a un servidor estando físicamente en el emplazamiento del mismo. Se está suponiendo que el atacante ha saltado la primera capa de la protección física externa; es decir he eludido cámaras de vigilancia, guardas de seguridad, cerraduras electrónicas, sensores de presencia, entre otros.

Ahora el atacante está ante varios racks que albergan numerosos servidores físicos y tiene acceso físico a las maquinas. De igual modo, está equipado con herramientas para hardware y software que van desde un simple destornillador de tipo Philips que le ayudará a extraer un disco duro, hasta dispositivos USB o CD auto-arrancables con herramientas de extracción de dato, entre otros.

Se proponen a continuación algunas recomendaciones para ponerle las cosas difíciles a este atacante.

2.2. BIOS / UEFI

Cuando se inicia un sistema operativo lo primero que se ejecuta es el BIOS (Basic Input/Output System); luego el sistema operativo y finalmente las aplicaciones que poseamos instaladas en nuestra computadora. Es importante mencionar que ya se está utilizando el UEFI (Unified Extensible Firmware Interface), un nuevo estándar para PCs diseñado para reemplazar a BIOS.

La fortificación de esta capa es un punto muy relativo y depende del fabricante del hardware, aunque generalmente los sistemas BIOS/UEFI se asemejan entre sí.

Los puntos que habría que reforzar en todos los entornos serían los listados a continuación, obviamente sin dejar de incluir aquellas soluciones específicas de cada fabricante o modelo de hardware:

- Edición de opciones protegidas por contraseña.
- Deshabilitar la selección de medio de arranque.
- Deshabilitar, si fuera posible, los siguientes medios de arranque.
- Cualquier medio de tipo extraíble.
- Arranque desde todas las tarjetas de red del sistema usando sistemas PXE.

Si el atacante consiguiera acceso a puertos de periféricos externos, unidades ópticas, tarjetas de red, etc. Se encontraría con una barrera en caso de haber deshabilitado el arranque desde los medios mencionados.

Estableciendo una contraseña para la edición de opciones se erradicaría el problema. Es recomendado por otro lado establecer una contraseña si fuera posible para los propios configuradores y BIOS exclusivos para los RAID, evitando así la manipulación de la configuración de los discos físicos [14].

2.3. Gestor de arranque. Grub y Grub2

Un Gestor de Arranque es un programa que nos permite administrar el arranque del sistema operativo. Hoy es normal tener varios sistemas

operativos en un mismo ordenador y es el gestor de arranque quien nos permite seleccionar el sistema operativo que queremos arrancar

Los gestores de arranque se instalan en el master boot record (MBR) o también llamado sector cero del disco, siendo el sistema de arranque el primer programa de se ejecuta una vez completado el inicio normal de la BIOS.

Los gestores de arranque más conocidos son GRUB y LILO, siendo Grub más moderno y más flexible que Lilo, debido a que permite que el administrador ejecute desde la línea de comando de Grub cualquier comando. Entre todas las características de Grub debemos mencionar el arranque de sistemas operativos no multi-arranque, la posibilidad de incluir múltiples formatos ejecutables, una interfaz de línea de comando muy flexible y una agradable interfaz de usuario. Pero existen una serie de problemas de seguridad relacionados con Grub o su versión más moderna GRUB 2 como puede ser el salto de las credenciales del servidor. Como consecuencia de ello, se podría disponer de acceso a la visualización de datos sensibles, así como extraerlos para su posterior estudio o explotación. La extracción puede realizarse en soportes físicos insertados localmente en la máquina atacante, o bien volcándolos a servidores remotos haciendo uso de los comandos SCP o FTP por ejemplo [15].

2.3.1. Un Gestor de arranque no protegido

Desde un gestor de arranque no protegido podemos realizar acciones independientes a la versión de GRUB como son las siguientes:

- Ejecución de comandos GNU básicos.
- Acceso a una Shell como root.
- Arranque de un sistema editando las líneas de configuración.

Hay que recordar que aún no está arrancado el sistema operativo y no se ha solicitado credenciales; entonces es importante saber las formas de trabajar en la pantalla del gestor de arranque GRUB2 antes de incursionar en las vulnerabilidades que se podrían presentar.

La herramienta GRUB permite al usuario administrador trabajar de formas diferentes:

- Interfaz de menú
- Interfaz del editor de menú de entrada
- Interfaz de línea de comando

La interfaz de menú consiste en un menú de sistemas operativos (núcleos) arrancables que se muestra al inicio.

Además, desde esta interfaz se puede:

- Entrar en la interfaz del editor de menú: pulsar -e-
- Entrar en la interfaz de línea de comandos: pulsar -c-

Se accede desde la interfaz de menú pulsando -e-.

Desde este editor el usuario puede teclear:

- b: ejecuta el comando seleccionado y arranca el sistema operativo
- e: edita el comando seleccionado
- c: pasa a la interfaz de línea de órdenes
- o/O: abre una nueva línea después/antes de la actual
- d: borra la línea seleccionada

Se llega pulsando -c- desde la interfaz de menú. Aparece el prompt del GRUB:

Figura 2.1: Interfaz de Grub

El primero de los ataques consiste en visualizar la información sensible de la máquina atacada. En GRUB 2 haremos lo siguiente

Pulsaremos la tecla “C” donde aparecerá una consola con una breve ayuda para su uso y el prompt grub ver *figura* 2.2.

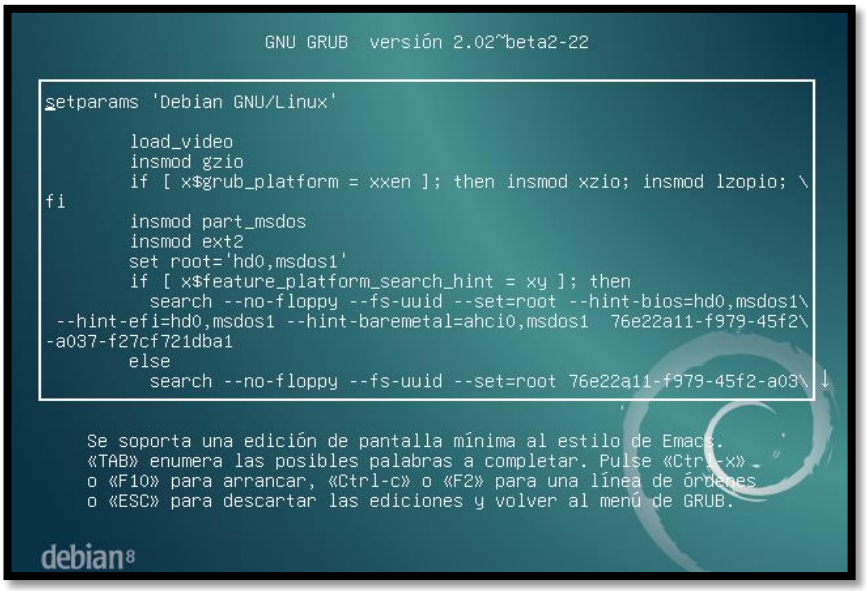

Figura 2.2: Prompt de GRUB 2

Escribir la palabra *help* o presionar la tecla *tab* donde aparecerá un listado de los comandos disponibles pero de todos los comandos mostrados vamos a ser uso de *ls* y *cat* para volcar listados de ficheros y sus contenidos. Un par de ficheros importantes pueden ser *passwd* y *shadow*. La ruta de estos ficheros podemos averiguarla escribiendo el comando *ls –l.*

Sin duda se puede obtener mucha información como son reglas de firewall, ficheros personales, scripts, configuraciones, usuarios, credenciales, servicios, etc. Aunque todo resultaría inútil si mostramos una pantalla del gestor de arranque con menos posibilidades de extracción de información.

Es el turno de obtener acceso *root* desde la terminal GRUB sin conocer las credenciales y para aquello haremos lo siguiente:

Cuando inicias, en la primera pantalla del Grub, seleccionas editar pulsando la letra e. Dicha acción abrirá una edición de la entrada de GRUB. Se debe arrancar el núcleo Linux en modo de escritura, mono-usuario y ejecutar una terminal sh. Las opciones anteriores son “rw single init=/bin/sh”

Una vez que el núcleo arranque aparecerá el símbolo #, que indicará que se tiene a disposición en *prompt* como súper-usuario, con las particiones montadas y con la posibilidad de realizar modificaciones sobre el sistema. A pesar de todo, la terminal no resulta del todo cómoda, pues le faltan acciones como puede ser el autocompletado; lo que se realiza normalmente disponiendo de todas sus características [16] [17].

2.3.2. Protección de Grub

En la primera versión del gestor de arranque se puede establecer una especie de contraseña maestra que ayudará a proteger la edición y el acceso a la consola de GRUB. Si aplicamos las medidas que vamos a mencionar para proteger GRUB conseguiremos los siguientes resultados.

- Impedir a alguien no autorizado el acceso a la línea de comandos del GRUB.
- Impedir a alguien no autorizado la edición de las entradas del GRUB.

- Impedir a alguien no autorizado la ejecución de todas las entradas del GRUB

Para obtener los resultados que acabamos de mencionar, tenemos que definir las políticas de uso y acceso:

- Definir los usuarios y las contraseñas de los usuarios que podrán modificar el Grub
- Cifrar las contraseñas de los usuarios
- Actualizar la configuración del Grub

En la segunda versión del gestor de arranque se modifican bastantes aspectos y se apuesta por ofrecer más flexibilidad a la hora de configurarlo. Todo ello mediante scripts para automatizar configuraciones y nuevas directivas de configuración entre otros cambios. A pesar de ello, el aspecto sigue siendo casi idéntico al de GRUB.

Centrando la atención en la protección mediante contraseña, la principal novedad radica en que es posible la creación de roles o grupos de usuarios con diferentes privilegios de GRUB2. Así pues, es posible disponer del rol por defecto super-usuario y agregar a él diferentes usuarios que adquieran la posibilidad de acceder a la terminal de GRUB y modificar entradas de arranque. Para esta configuración se centrará la atención únicamente en el rol super-usuarios [18].

Como ocurría en la primera versión de GRUB, puede establecerse la contraseña en texto plano o cifrado además de bloquear diferentes líneas de arranque. La diferencia es que si se escoge la opción de cifrado, ya no se establecerá md5, sino que en su defecto se utiliza pbkdf2. La herramienta a utilizar para generar la contraseña cifrado es grub-mkpasswd-pbkdf2. Ver *figura 2.3.*

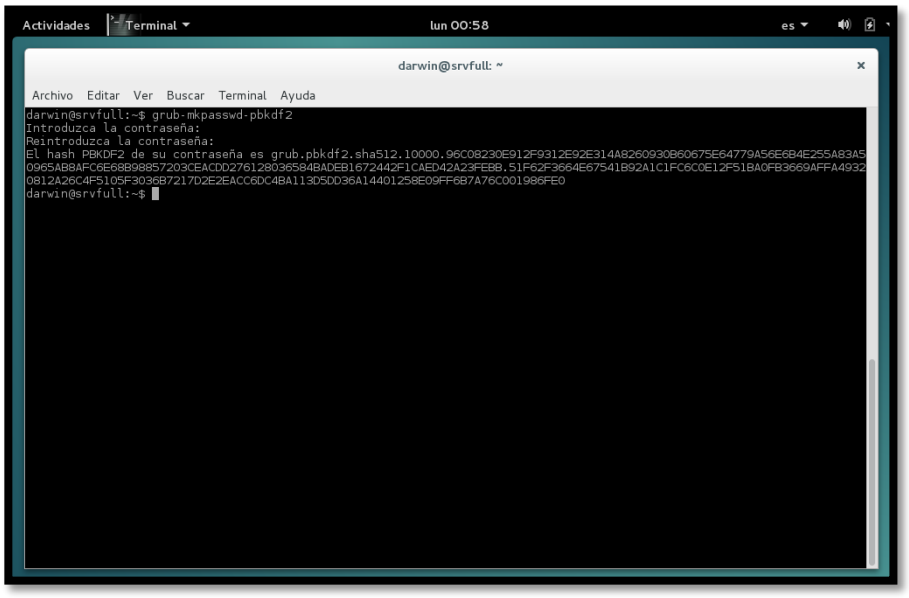

Figura 2.3: Generación de la contraseña pbkdf2

Una vez que tenemos ingresamos la línea de comando *grub-mkpasswd-pbkdf2,* nos va a solicitar que agreguemos la contraseña para poder generar el hash de pbkdf2, el cual lo debemos agregar al archivo de configuración.

A la hora de establecer la contraseña, en el caso de Debian en concreto, es recomendable hacerlo en el fichero /etc/grub.d/40_custom. Nuevamente, como recomendación, se ha utilizado una contraseña cifrada. Ver *figura 2.4*

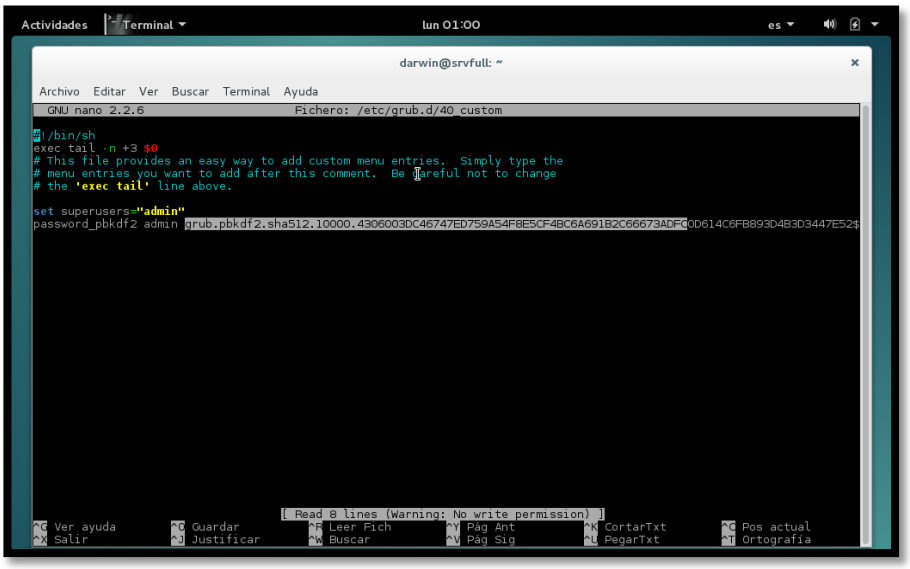

Figura 2.4: Modificación del archivo grub.d/40_custom

Para que la configuración sea efectiva es necesario regenerar el fichero de configuración de GRUB. Para ello se utiliza el comando update-grub, que leerá todos los scripts incluyendo el /etc/grub.d/40_custom como se muestra en la *figura 2.5.* De igual modo estará restringido el acceso a la edición de las líneas de arranque. En el caso del ejemplo, habría que introducir el nombre del usuario “admin” y la contraseña establecida.

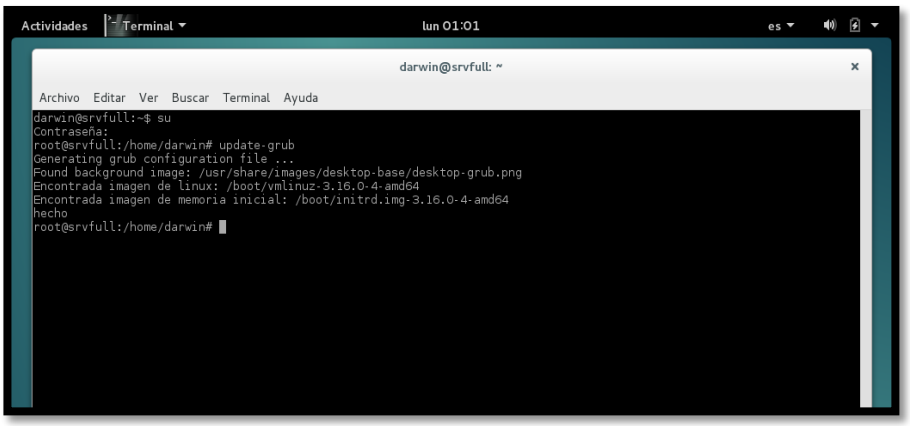

Figura 2.5: Actualizando el Grub - Comando Update Grub

Si se precisa del bloqueo de arranque de los núcleos Linux, sería necesario modificar el script que se encarga de detectar dichos sistemas y establecer

el parámetro—users. En el caso de Debian se trata del fichero /etc/grub.d/10_linux.

En el momento en que se reinicie la máquina se podrá comprobar que la consola no es accesible a menos que se introduzcan las credenciales establecidas. Ver *figura 2.6.*

Figura 2.6: Grub Protegido

2.4. Protección del sistema de ficheros

Los sistemas de ficheros nos permiten almacenar, recuperar y estructurar la información que se encuentra guardada en las unidades de almacenamiento, y luego nos permite representarla en forma gráfica o por texto mediante un gestor de archivo. Los ficheros están gestionados por el sistema operativo. La mayor cantidad de los sistemas operativos tienen su propio sistema de archivos.

La seguridad es muy importante en el sistema, la primera política se seguridad en el sistema de fichero, es dar permisos sean estos de lectura, escritura y ejecución a cada fichero y usuario.

La solución para proteger los discos en caso de robo es mediante el cifrado de discos o particiones. Existen diferentes métodos de cifrado como por ejemplo LUKS y TrueCrypt.

Luks (Linux Unified Key Setup)

Es una implementación de cifrado muy sencilla de utilizar y define un formato estándar de disco e independiente de la plataforma utilizada. La mayoría de las distribuciones disponen de soporte de cifrado. En el caso de Debian GNU/Linux para obtener un sistema con cifrado de datos siga los siguientes pasos durante la instalación.

- Seleccione el particionado de discos guiados con cifrado de datos
- Seleccione que el sistema será instalado en una misma partición.
- Borre datos de la partición cifrada
- Establezca la passphrase

Una vez finalizada la instalación y la máquina arranca, aparecerá el ya conocido gestor de arranque GRUB2. Pero para quedar de cara al sistema se debe introducir de forma correcta a frase de paso. Es recomendable tener presente para seguridad agregar la funcionalidad de cifrado a tu sistema GNU/Linux mediante la integración de dm-crypt y cryptsetup y LUKS.

Truecrypt

Es un proyecto open-source y entre sus principales ventajas destaca la posibilidad de ser utilizado con sistemas Linux, Windows y MacOS X. Emplea diferentes algoritmos como AES, Serpent y Twofish o una combinación de los mismos. TrueCrypt crea un volumen cifrado en la computadora donde puedes almacenar tus archivos.

2.4.1. Instalación de un sistema de cifrado de datos.

A continuación procederemos a la instalar de cero un sistema operativo precautelando antes que todo la seguridad de los datos que mantendremos de acuerdo a las necesidades específicas del administrador de red.

Continuando con el caso de un esquema de bajo costo para las pymes es recomendable instalar un sistema operativo de distribución gratuita, que permita la administración de los accesos y servicios, por lo que se ha optado en este caso Debian GNU/Linux para realizar las configuraciones de fortalecimiento inicial de un sistema operativo.

Antes que nada se sugiere tener previamente realizada una planificación del particionado de disco, es decir el tamaño, formatos, y características que procedamos a configurar como el ejemplo en la siguiente tabla.

USO	FORMATO	CIFRADO	TAMAÑO
/boot	Ext-4		250 MB
/	Ext-4		15GB
Swap	Swap	LUKS	8GB
/home	LVM-ext4	LUKS	20GB
/var	LVM-ext4		12GB
/opt /temp	LVM-ext4		20GB

Tabla 4. Esquema de partición de disco.

Nosotros mostraremos el aspecto que tiene la parte cifrada en el momento de la instalación del sistema operativo. De la misma manera se verá el resultado obtenido tras la instalación.

Una vez que arrancamos el boteo del disco con el sistema operativo, nos aparecerá la ventana de *setup* en el cual nos solicitara escoger el tipo de instalación que deseamos hacer, ya sea por medio de entorno grafico o mediante entorno de línea de comandos. En practica al momento de seleccionar el entorno de instalación, varia de la experticia y conocimiento que posean al momento de realizarlo.

Continuando con la instalación nos solicitara ingresar datos generales de configuración listados a continuación:

- Idioma
- Ubicación
- Teclado
- Nombre de Equipo
- Nombre de Dominio
- Clave de Super Usuario (root)
- Nombre de Usuario Inicial
- Particionado de Disco
- Gestor de Paquete
- Configuración de Proxy
- Instalación de Programas
- Gestor de Arranque GRUB

Una vez que legamos al paso de selección de disco duro, debemos continuar con la configuración manual de la partición para tener el control de configuración. Entonces el sistema nos va a mostrar la información de la capacidad y ubicación de los discos que tengamos instalados.

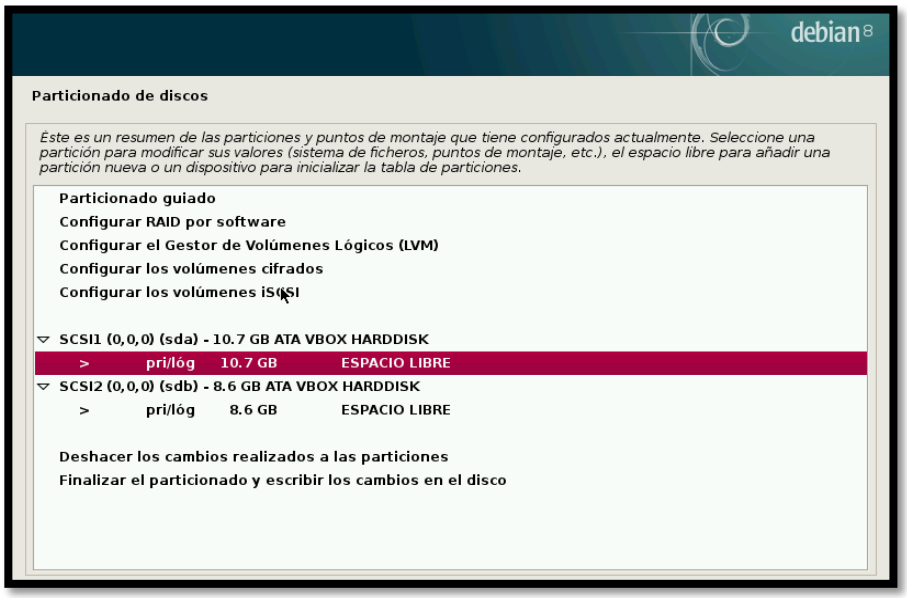

Figura 2.7: Partición del Disco Duro

Una vez que seleccionamos el disco duro libre que vamos a fraccionar, nos solicita un mensaje de advertencia que estamos de acuerdo con la modificación que se va a realizar para así poder comenzar la segmentación del disco. Ver *figura 2.7.*

Hecho esto, el sistema nos pide que ingresemos el tamaño de la fracción a crear, mostrando la cantidad que tenemos libre actualmente, posterior a eso selección de partición entre lógica y primaria, finalmente en donde va a estar ubicada la partición.

Bueno ya en este punto es donde ves todas las características que puedes configurar en la partición y las otras por defecto si no conoces datos específicos que quieras hacer sobre la partición que estás configurando. Lo que vamos a modificar nosotros es el punto de montaje, que lo que te permite es indicarle al sistema que directorio va a estar en esa partición. Ver *figura 2.8*

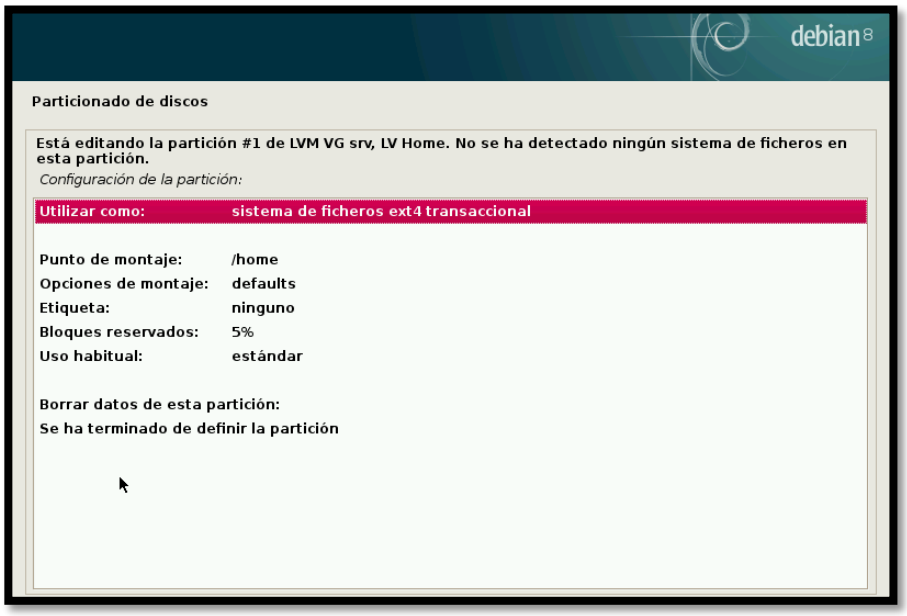

Figura 2.8: Configuraciones de partición.

Al seleccionar Punto de montaje el sistema mostrará las opciones que tiene para la creación de la partición específica, y basándonos previamente al esquema de particiones que íbamos a utilizar escogeremos el punto de montaje para ese directorio.

La *figura 2.9* siguiente muestra cada uno de los directorios con una información previa de su contenido y utilidad, ayudando así a mantener una mejor idea de la gestión del fichero de directorios a crear.

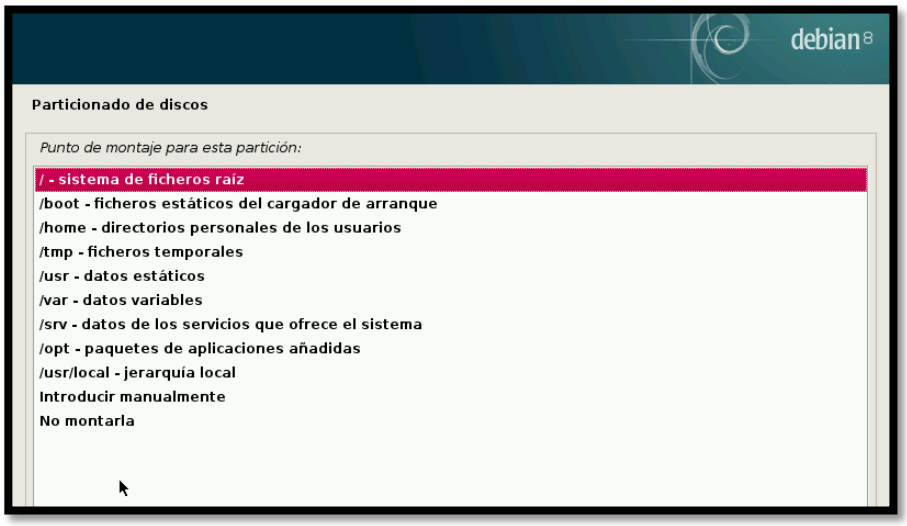

Figura 2.9: Tipos de Directorios Específicos Debian GNU/Linux.

Para ámbitos de fortalecimiento de la instalación de particiones es necesario saber cuál es la cantidad de información de maneja cada directorio y de acuerdo al estándar de la jerarquía del sistema de ficheros, por ejemplo para la partición del directorio “/boot” será sin cifrar debido a que contiene el gestor de arranque, pues si no el sistema no sería capaz de arrancar.

Es recomendable mantener siempre la instalación de los directorios estáticos con el espacio mínimo reducido puesto que un gestor de arranque no supera los 256Mb de información incluyendo espacio para las actualizaciones del kernel, lo cual ayuda a que no se intente instalar cualquier archivo malicioso debido a la falta de espacio de almacenamiento.

Los sistemas de cifrado pueden configurarse de diferentes modos. Se puede cifrar una partición y posteriormente formatearla y crear volúmenes lógicos y cifrar los discos lógicos.

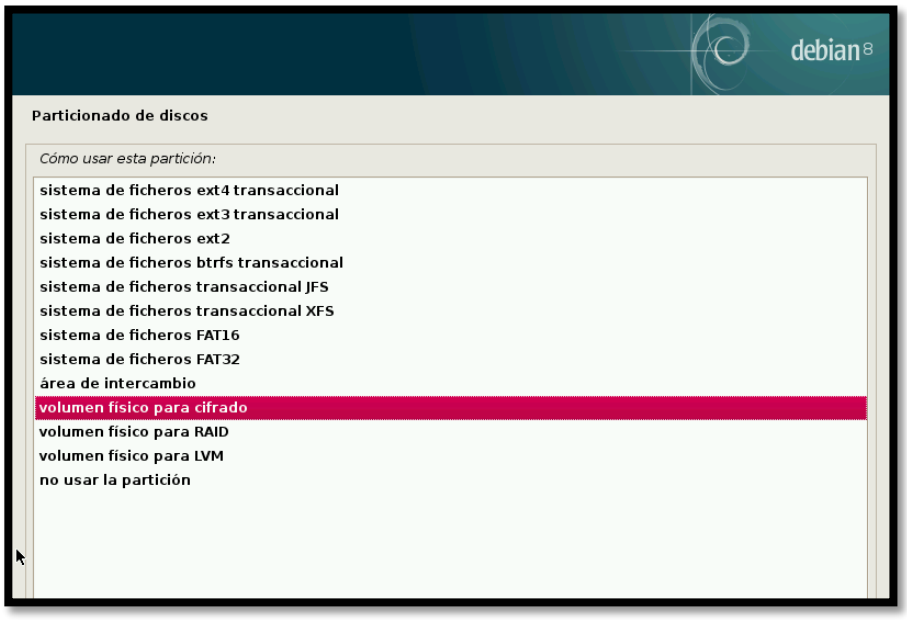

Figura 2.9: Tipos de Sistemas de Ficheros Debian GNU/Linux.

Como se aprecia en la *figura 2.10*, vamos a cambiar en la opción de la *Utilizar como:* debemos seleccionar *volumen físico para cifrado*, al seleccionar esto, cambiaran las opciones o características para cifrar la partición.

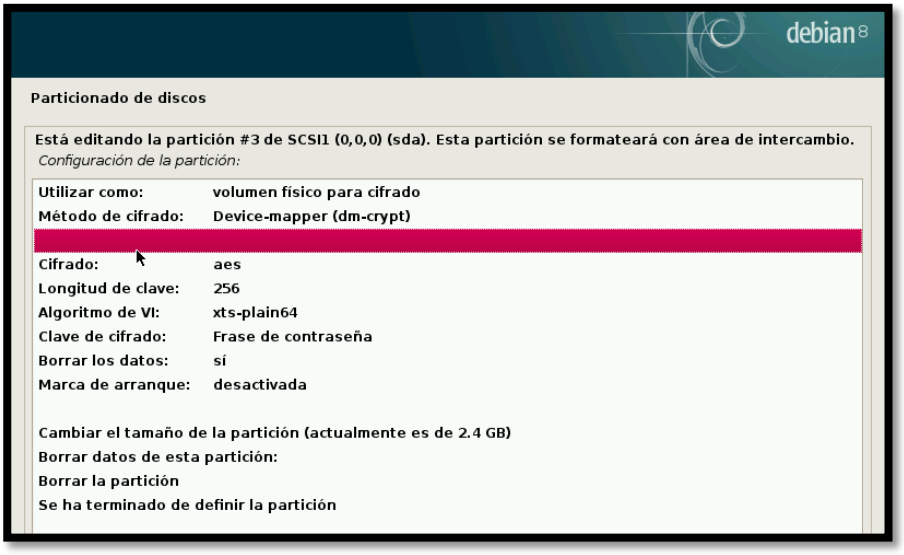

Figura 2.10: Configuración de parámetros de cifrado.

Dentro de las características de cifrado de una partición encontramos el método de cifrado que vamos a utilizar, para el cual vamos a seleccionar el cifrado Device-mapper para la utilización de LUKS, el tipo de cifrado usado será AES que ciertamente es un buen sistema de cifrado, longitud de clave puede estar entre 256 y 512 para buenas practicas siempre es recomendable el más robusto, el tipo de algoritmo de vector de iniciación se usara xts-plain64 que simplemente pasa el índice del sector de 64 bits directamente en el algoritmo de encadenamiento, el tipo de clave de cifrado utilizaremos la *Frase de contraseña,* el cual hace que sea más complicada la obtención de la clave por métodos de fuerza bruta ya que mantiene agrupación mayores a 16 caracteres en combinación con letras números y caracteres especiales.

Una vez que terminamos de configurar el método de cifrado tendremos una configuración similar a la *figura 2.11* anterior, para entonces seleccionamos terminar de definir la partición y continuamos.

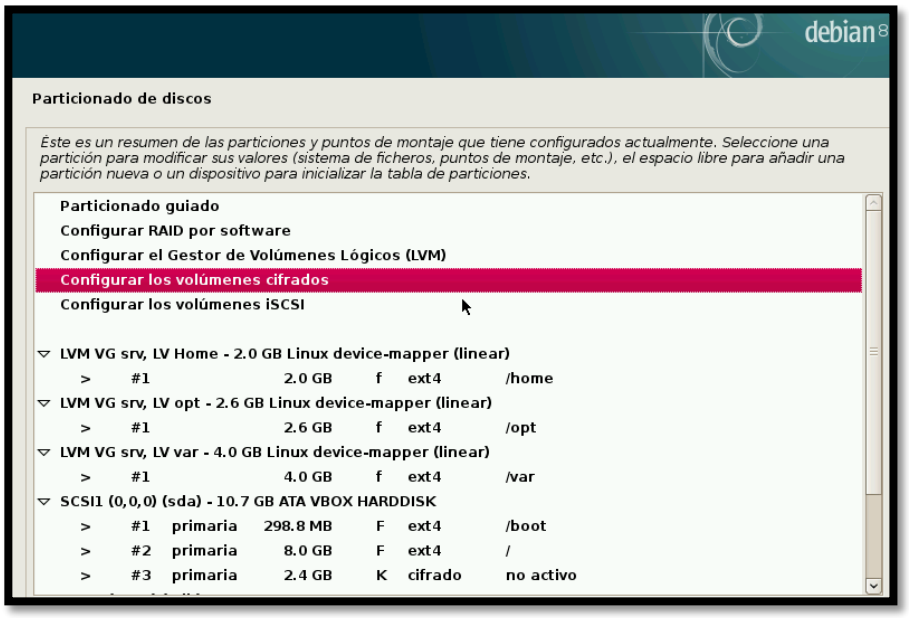

Figura 2.11: Configuración general de particiones.

Una vez que hemos terminado de configurar y tenemos el esquema las particiones como volúmenes físicos cifrados, el particionado quedaría de la siguiente forma, Ver *figura 2.12.*

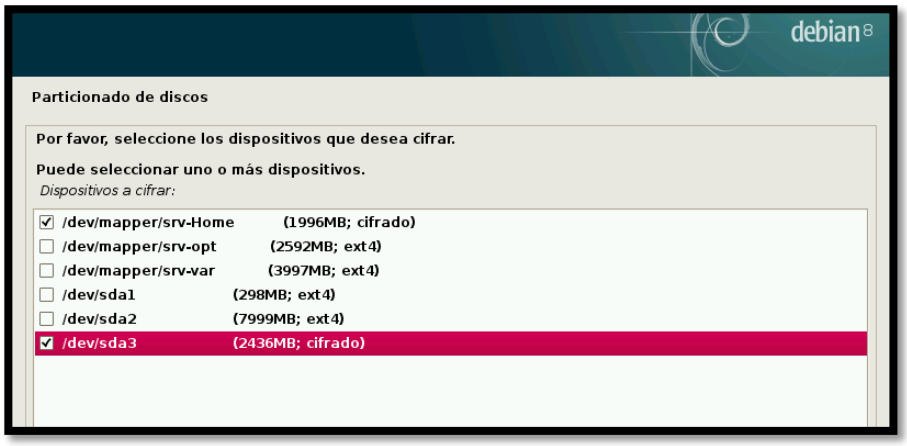

Figura 2.12: Particiones a cifrar.

A continuación vamos proceder a configurar el sistema de cifrado, que tiene un menú aparte en la opción que se llama *Configurar los volúmenes cifrados*. Después el sistema les pide que indiquen cuales particiones son las que se van a cifrar, para el cual debemos escoger las que están con los parámetros de cifrado. Ver *figura 2.13.*

Posterior se mostrará un mensaje indicando que se van sobrescribir datos aleatorios en las particiones mencionadas y comenzara el proceso de borrado y sobrescritura el cual puede demorar una cantidad de tiempo de acuerdo al volumen de las particiones creadas. Ver *figura 2.14*

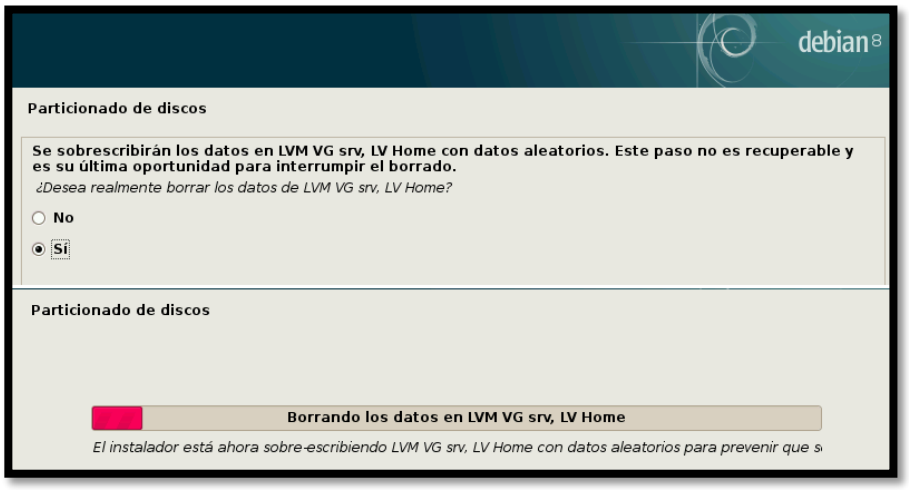

Figura 2.13: Sobrescritura de datos en particiones cifradas.

Una vez que se termina de realizar el borrado de la unidad, pedirá que ingresemos una frase de cifrado, *figura 2.15*, al ser frase puede contener caracteres de espaciado y especiales, los cuales debemos tener en cuenta que serán solicitados cada vez que se arranque el sistema por cada partición cifrada.

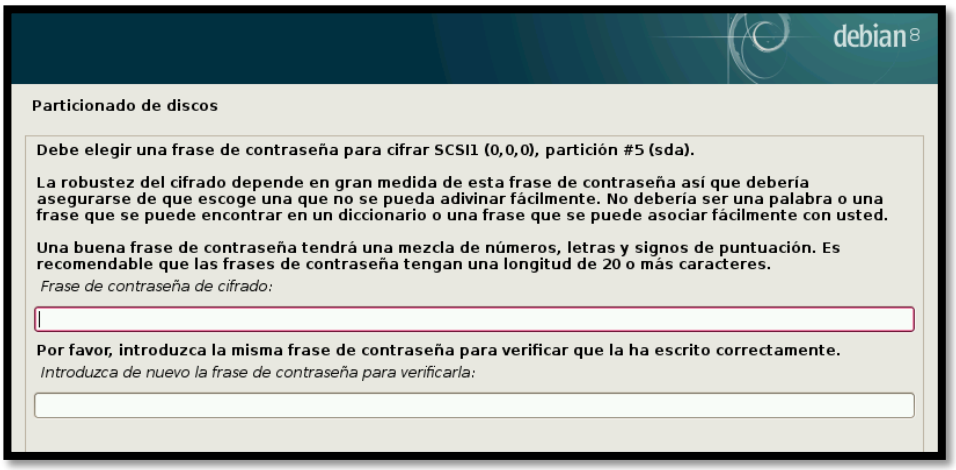

Figura 2.14. Gestionando Frase de contraseña

Ya con esto están todas las particiones cifrada, para lo cual como última opción es asignarles el punto de montaje de cada una de las particiones cifradas*, ver figura 2.16,* en el cual queda el esquema de partición final, y continuar con la instalación de los archivos de cada uno de los directorios.

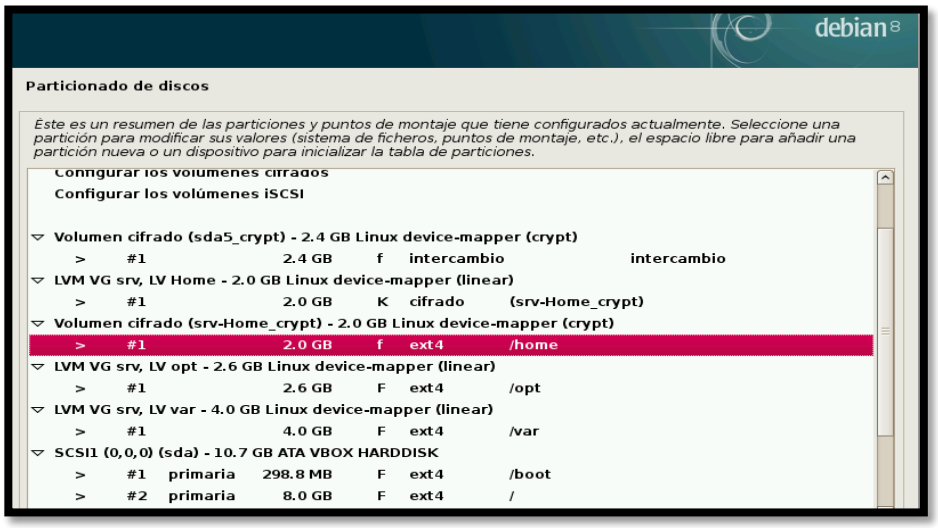

Figura 2.15: Partición cifrada con punto de montaje realizado.

2.5. Cifrado de ficheros

Hasta el momento se ha conseguido cifrar el sistema de ficheros o mejor dicho las particiones que lo contienen. El método resulta efectivo, pero aún puede existir alguna posibilidad más de conseguir acceso a información privilegiada estando físicamente en el emplazamiento del servidor.

En numerosas ocasiones, tanto usuarios como administradores de sistemas olvidan cerrar sus sesiones o simplemente no las bloquean. Si se diera este caso, todas las prácticas seguidas hasta el momento serán en vano. Por muy fuerte que sea el cifrado de las particiones y su frase de paso, se haya protegido el gestor de arranque, etc. No servirá de nada para esta situación.

Existen diversos métodos para tratar de mitigar estos problemas, pero el que aquí se presenta consiste en cifrar ficheros sensibles mediante el uso de GPG, GNU Privacy Guard.

2.5.1. GPG, Gnu Privacy Guard

GPG, GNU Privacy Guard, es una implementación del estándar OpenPGP, que a su vez nació como versión libre de PGP, Pretty Good Privacy. Se encuentra disponible por defecto en la mayoría de las distribuciones

basadas en Linux hoy en día y en caso contrario, está disponible en los repositorios oficiales de cada distribución.

Mediante el uso de la aplicación gpg se pueden realizar las siguientes acciones:

Firmado de ficheros mediante el uso de clave privada. Su finalidad consiste en determinar si el fichero firmado pertenece a quien dice ser.

Cifrado de ficheros mediante el uso de clave pública. Consiste en proteger un fichero cifrándolo por completo haciendo uso de una clave pública.

Cifrado de ficheros mediante el uso de una passphrase. Es exactamente igual que el caso anterior salvo que se usa cifrado simétrico. Una misma clave para cifrar y descifrar.

La última acción es la más conocida y sencilla de llevar a cabo, pero como contrapartida no resulta tan flexible como su homónima de clave pública.

Antes de continuar con la utilización de gpg, resulta necesario hacer una pequeña introducción al funcionamiento de la criptografía de clave pública.

2.5.2. Cifrado y descifrado de ficheros

La finalidad de cifrar es la de mantener la privacidad de los datos. Puede ser para compartir el mensaje o archivo con una persona de confianza o bien para protegerlos en un disco y desbloquearlos cuando sea necesario. El proceso de cifrado sería el siguiente:

Se cifra el contenido con la clave pública, por lo que cualquier persona con acceso a la clave pública puede cifrar el contenido de un mensaje o fichero.

El proceso de descifrado se realiza mediante la pareja de la clave pública que ha cifrado el mensaje o archivo. Es decir, es necesario utilizar la clave privada asociada a la clave pública que ha cifrado. Si la clave privada se generó con una passphrase asociada será necesario su introducción para acceder al mensaje o archivo.

2.5.3. Firmado y verificación de ficheros

El firmado de un contenido es una acción que puede realizarse tanto si el fichero o mensaje ha sido cifrado tanto como si no. Se utiliza para poder verificar que la fuente de la cual proviene un archivo o mensaje es realmente la que dice ser. El método para firmar es el siguiente:

Se firma el archivo o mensaje con la clave privada. Debe introducirse la *passphrase* que bloquea la clave en caso de disponer de ella.

Cualquier persona con acceso a la clave pública puede verificar la firma y comprobar si el origen del mensaje o archivo es correcto.

El caso de GPG no existe ninguna autoridad certificadora, CA como acurre con las implementaciones PKI, *Private Key Infraestructure*. Para GPG se usan las llamadas relaciones de confianza, que consiste en que los propios usuarios verifican que una clave pública accesible por todos pertenece a la persona indicada por la clave.

Las contraseñas de cifrado y de descifrado pueden ser iguales (criptografía simétrica) o no (criptografía asimétrica). GPG es una herramienta multiplataforma de cifrado/descifrado y es software libre licenciado bajo la GPL. Viene de serie en las principales distribuciones. GPG cifra los mensajes usando pares de claves individuales asimétricas generadas por los usuarios. También es posible añadir una firma digital criptográfica a un mensaje.

2.5.4. Otras Protecciones

Existen algunas opciones adicionales con las que es posible proteger aún más una máquina frente a ataques físicos. A continuación se enumeran algunas de ellas:

- Bloqueo y cierre automático de sesión.
- Variable de entorno TMOUT.
- Evitando el reinicio y apagado accidental.
- Deshabilitar dispositivos de almacenamiento USB.

2.6. Protección Perimetral

Tras haber estudiado ciertos métodos para proteger la capa física de un entorno, llega el momento de implementar protecciones en la capa más externa en cuanto a la red de comunicaciones se refiere. En este capítulo se

verán herramientas que ayudarán a establecer barreras para aumentar la seguridad de la red, habilitando al mismo tiempo accesos remotos seguros. Por otra parte, se expondrá cómo se puede monitorizar es estado de salud de las máquinas del entorno de red con lo que siempre se podrá obtener información de las máquinas, si responden, sus servicios están activos, etc.

La seguridad en el perímetro de la red consiste en implementar políticas de seguridad en los equipos de comunicación, los equipos son instalados entre la red interna y la externa, cuya objetivo de las políticas es permitir o denegar el acceso a los diferentes servicios de la red tanto para usuarios internos como externos.

Fortalecer el perímetro consta de 3 etapas:

- IPTABLES
- VPN

2.7. Iptables

Iptables es un firewall que está incursionado en el núcleo de Linux, que permite la gestión, configuración y manejo del filtrado de tráfico de red en una máquina Linux. Para ello se utiliza netfilter, que es un framework integrado en el núcleo Linux capaz de interceptar y manipular paquetes de datos de red en diferentes estados del procesamiento. Netfilter hace referencia al proyecto que brinda herramientas libres para administrar cortafuegos bajo Linux.

2.7.1. Función

El núcleo Linux tiene la habilidad de permitir pasar o no paquetes de datos de red por una serie de políticas o reglas que iptables nos permite configurar. Aquellas reglas se aglutinan en cadenas y las cadenas se ubican en las tablas. Las reglas son una aglutinación de parámetros configurados por el encargado del iptables, donde se busca que esas reglas coincidan con un datagrama de red según el protocolo, estado, IPs destino y origen, etc. Cuando el datagrama coincide con alguna regla, el iptables deberá aceptar o rechazar el paquete, dichas acciones son conocidas como ACCEPT y DROP.

Es importante acotar que cuando los paquetes ingresan a la red con iptables deben pasar por todas las reglas y cadenas que se encuentran configuradas por defecto. Si se presenta la ocasión de que no existe regla alguna que coincida con el paquete de datos de red, el paquete será sujeto a las reglas que el iptables tenga configuradas por defecto, y se entiende que se dejará pasar todos los paquetes que no coincidan con alguna cadena debido a que el iptables cuando no está configurado la acción a tomar siempre es ACCEPT.

2.7.2. Tablas

Hay 4 tablas ya configuradas por defecto, y contienen cadenas ya definidas. Es posible agregar más tablas mediante la incorporación de módulos en iptables. El encargado de la administración del iptables puede dentro de cualquier tabla eliminar o crear cadenas.

Las tablas por defecto son:

Filter.- En esta tabla se administra los filtros de paquetes. Es decir aquí se bloquean o dejan pasar paquetes de datos sometiéndose a las reglas que están incluidas en las cadenas predefinidas por defecto mencionadas a continuación.

Input.- En esta cadena se determinan las reglas para los paquetes recibidos en la máquina local, y por eso también es llamada algunas veces LOCAL_INPUT.

Output.- En esta cadena atraviesan los paquetes que han sido propagados desde la máquina local, y por eso también es llamada *LOCAL_OUTPUT*.

Forward.- Esta cadena permite enrutar los paquetes de datos que se dirigen a otros destinos.

Nat.- En esta tabla se configuran las reglas de traducción de direcciones o de los puertos de paquetes. Las cadenas por defecto son mencionadas a continuación.

Prerouting.- En esta cadena se verifica la dirección de red de los paquetes que entren al sistema configurado por Iptables antes de ser enrutados, se utiliza DNAT (Destination Nat).

Postrouting.- En esta cadena se utiliza SNAT (Source Nat), Por aquí ingresan los paquetes de datos después de ser enrutados, traduciendo las IPs locales en IP pública.

Output.- Cadena usada para hacer la traslación de dirección de redes en paquetes generados localmente.

Mangle.- En esta tabla se pueden alterar los paquetes y es responsable de adaptar opciones tales como calidad de servicio, marcado de paquetes y tiempo de vida. Las cadenas por defecto son mencionadas a continuación:

Raw.- Es una nueva tabla que nos permite configurar una acción NOTRACK y librarse que netfilter adapte conntrack al paquete de red y así eludir que efectué un seguimiento del paquete.

A continuación en la *figura2.17,* se muestra un diagrama de flujo correspondiente a los diferentes estados con los que trabaja iptables.

Cuando un paquete u otra comunicación llegan al kernel con iptables se sigue este camino.

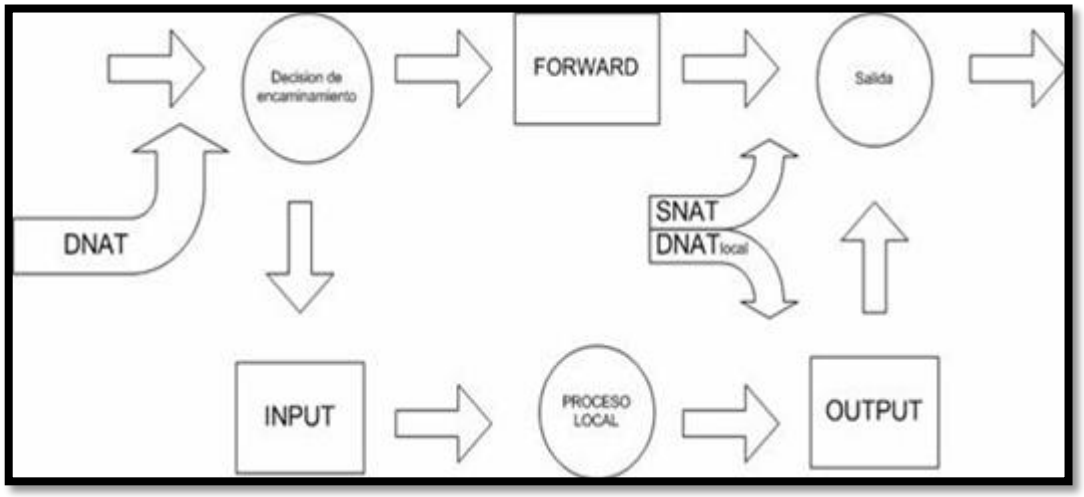

Figura 2.16: Flujograma de estados de Iptables.

2.7.3. Reglas

Hemos mencionado los diferentes tipos de cadenas que están incorporadas en las diferentes tablas, así como podemos crear tablas, iptables le brinda la oportunidad al usuario de poder crear o eliminar cadenas que desee dentro de la tabla, lo que permite aglutinar las reglas de manera lógica.

Cada cadena está formada por reglas. Los paquetes ingresan al sistema, son verificados por las reglas de la cadena correspondiente y además podemos unir las resoluciones de las reglas con otras cadenas.

2.7.4. Destinos de reglas

En las reglas se definen las características de debe llevar el paquete, como dirección IP, número de puerto, para que la regla pueda encajar y si no coincide el proceso seguirá con la regla posterior. En cambio si la regla encaja, las directrices de destino de las reglas se siguen y por ende algún otro proceso de la cadena se interrumpe.

Las siempre llevan un destino sean determinados por el usuario o algún destino ya admitido como son ACCEPT, DROP, QUEUE, o RETURN. Estos destinos los detallamos a continuación:

ACCEPT. Aquí netfilter acepta el paquete. Aquel paquete que es admitido en la cadena de INPUT es recibido por la máquina local y si aquel paquete es aceptado en la cadena OUTPUT es permitido abandonar el sistema y un paquete de red que es admitido en la cadena FORWARD se le permite ser enrutado a través del sistema.

DROP. Aquí netfilter descarta el paquete sin hacer algún tipo de proceso.

QUEUE. Aquí el paquete de red es encaminado a una cola y si no hay lectura de la cola, es igual a la regla DROP.

RETURN. Le permite al paquete de red que deje de circular por la cadena. Si la cadena es una cadena principal, al paquete se le adjudicará la sentencia por defecto como pueden ser ACCEPT, DROP o similar), pero si la cadena

es una subcadena de otra, el paquete de red seguirá por la cadena superior como que nada hubiese sucedido.

Hay otros destinos considerados de extensión disponibles. Aquí vamos a mencionar los más utilizados.

REJECT. Es similar a DROP aunque REJECT envía los paquetes con error al usuario que envió el paquete. Generalmente este destino es usado en las cadenas de INPUT y FORWARD.

LOG. Es una bitácora del paquete y la podemos utilizar en cualquier cadena de cualquier tabla y en algunas circunstancias es usada para analizar los fallos y qué paquetes están siendo eliminados. Es importante mencionar que la información de este destino es enviada al log del núcleo.

ULOG. También es una bitácora pero se diferencia con el destino LOG en que aquí se realizan varias transmisiones de los paquetes de red que concuerden con esta regla mediante un socket netlink.

DNAT. Aquí en este destino se produce la traducción de la dirección de red de destino, es usado sólo en las cadenas OUTPUT y POSTROUTING dentro de la tabla nat, La decisiones tomadas se les recuerda a los paquetes de red que pertenecen a la misma conexión.

SNAT. Aquí en este destino se produce la traducción de la dirección de red de origen, es usado sólo en la cadena POSTROUTING dentro de la tabla nat.

MASQUERADE. Aquí usamos SNAT para las direcciones dinámicas, que en su mayoría son los ISPs. Es importante mencionar que en este destino se debe calcular la Ip origen en la cual se hace NAT precisando en la Ip de salida cuando la regla coincide.

2.7.5. Comandos de Iptables

Habíamos detallado los diferentes tipos de tablas, cadenas y reglas. Se había mencionado que las reglan estás incursionadas en las cadenas, y de la misma manera las tablas están formadas por cadenas. Iptables posee

parámetros y comandos que autorizan fijar la conducta de una o varias reglas. Esto quiere decir que el usuario puede añadir una regla, alterar una regla actual o descartar una cadena.

A continuación se detalla los comandos más utilizados de iptables es el comando:

***service iptables start*.** Permite iniciar Iptables

service iptables stop. Permite parar Iptables

service iptables restart. Permite reiniciar Iptables

***a*.** Nos permite crear una regla a la cadena conveniente.

***t*.** Se emplea para señalar la tabla con la que se está laborando en el instante de emitir el comando.

***j*.** Señala la acción que se le adapta a un paquete de red en caso de encajar con la regla.

***d*.** Descarta la regla de la cadena optada.

***r*.** Nos permite cambiar la regla de una cadena determinada.

***e*.** Modifica el nombre de la cadena.

***i*.** incluye una nueva regla antes de otra.

***l*.** Nos muestra la lista de las reglas de una cadena específica y si no se define una cadena, el comando mostrará todas las cadenas de la tabla.

***n*.** Crear una nueva cadena.

2.7.6. Parámetros

Todas las reglas en iptables poseen declarado su estado por los parámetros, que forman parte primordial.

A continuación se detalla los parámetros más utilizados.

o. Indica la interfaz de salida como pueden ser (eth0, eth1, eth2...).

sport. Indica el puerto de origen.

i. Indica la interfaz de entrada como pueden ser (eth0, eth1, eth2...).

p. Señala el protocolo del paquete a comprobar como por ejemplo: tcp, udp, icmp o all, siendo all la configuración por defecto.

dport. Indica el puerto de destino.

line-numbers. Adhiere el número que le pertenece a cada regla dentro de la cadena cuando son listadas las reglas.

2.7.7. Agregar reglas con Iptables

La configuración de una regla elemental sería:

iptables –t <nombre de la tabla> -A <cadena> <opciones> -j <acción>

Aquí mencionaremos algunos ejemplos:

iptables -t filter -A INPUT -p tcp -dport 23 -j DROP

La tabla filter es por defecto, la cadena input nos permite filtrar los paquetes de red que hacen uso del protocolo tcp que ingresan por el puerto 23 (Telnet) y mediante la acción DROP los paquetes son descartados.

iptables -A INPUT -p tcp -dport telnet -j ACCEPT

Esta regla indica lo inverso a la regla anterior, aquí se ha eliminado el parámetro –t filter ya que es una tabla por defecto y por ende no altera la regla. La diferencia está en que está regla acepta los paquetes.

iptables -A INPUT -i eth0 -p tcp -dport http -j ACCEPT

Aquí tenemos un ejemplo donde se está aceptamos tráfico http y estamos indicando mediante el parámetro –i la interfaz de entrada que este caso es eth0.

iptables –A FORWARD –i eth1 –o eth0 –p tcp—dport 80 –j DROP

Esta regla cumple la función de rechazar el tráfico HTTP que ingrese por la interfaz interna eth1 y salga por la interfaz externa eth0, mediante el

parámetro –p nos indica el tipo de protocolo que es este caso es tcp, y además nos indica el puerto destino mediante –dport.

#iptables –t nat –A POSTROUTING –o eth0 –j MASQUERADE

Aquí se ha señalado que a todos los paquetes de red que salgan por la interfaz externa eth0, se les aplique SNAT. De esa manera el paquete podría encaminarse por Internet sin inconvenientes.

2.7.8. Listando reglas con Iptables

Sí se anhela conseguir el listado de las reglas debemos usar el parámetro –l que normalmente es acompañada por –n para que los resultados se muestren de forma numérica, y –v para que muestre una información más detallada.

Por Ejemplo:

#iptables –t nat –nvl—line-numbers

Se muestra las reglas que se han designado en la tabla nat y además mediante el parámetro—line-numbers nos permite observar los números agremiado a cada regla.

2.7.9. Modificando la regla por defecto

Para cambiar esa conducta vamos a utilizar la siguiente sintaxis

Iptables –t <tables> -P <cadena> <Política>

Por Ejemplo

iptables -P INPUT DROP

En esta instrucción hemos cambiado la política por defecto, como todos sabemos es ACCEPT, y ha sido cambiado a DROP y a su vez estamos denegando el trafico entrante que vaya dirigido a la máquina local donde se encuentra configurado el iptables.

2.7.10. Borrar reglas con Iptables

Para rehabilitar la configuración del firewall será vital eliminar reglas, a continuación exponemos dos sintaxis:

Iptables –t <tables> -F

Está sintaxis nos indica que estamos eliminando todas las reglas de determinada tabla.

Por ejemplo:

iptables –t nat –F

Aquí estamos borrando todas las reglas de la tabla nat.

#Iptables –t <tables> -D <cadena> --line-numbers

Está sintaxis nos expone como eliminar regla de una cadena según el número de la regla en la cadena, y como se mencionó anteriormente para obtener los números ligados a cada regla usamos el parámetro –line-numbers.

2.7.11. Guardar información de Iptables

Cuando la máquina en la que se configura iptables se reinicia, las reglas desaparecen y vuelven a la configuración por defecto, debido a que las reglas son guardadas en la RAM. Para hacer que las reglas tengan consecuencia en cualquier instante que se reinicie la máquina se requiere que el script se ejecute desde /etc/rc.local o guardarlas en el fichero /etc/sysconfig/iptables.

Importante: También es posible realizar esta tarea usando los comandos *iptables-save, iptables-restore* e *iptables-apply*, además si se desea asignar el fichero /etc/sysconfig/iptables a otras máquinas, es importante escribir /sbin/ service iptables restart para que las siguientes reglas tengan efecto.

2.7.12. Creando un firewall con iptables doble enlace

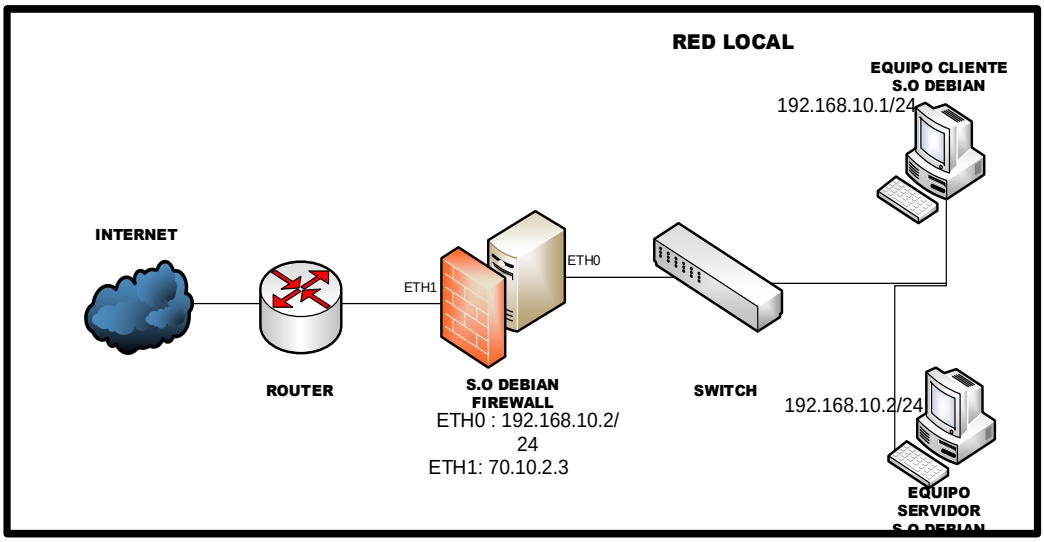

Figura 2.17: Esquema de firewall doble enlace.

Para la implementación nos basaremos en el esquema de la *figura 2.18, que* se trata de un firewall de dos patas. La máquina que actúa como router y firewall ejecuta Debian 8 dando conectividad a la red local hacia internet.

Para el esquema mostrado nos podemos dar cuenta que máquina que está actuando como firewall y router al mismo tiempo tiene asociada la IP 70.10.2.3 para la interfaz externa y la IP 193.168.10.2/24 para la interfaz interna, mientras la red interna tiene asociada la red 192:168.10.1/24.

Versión con drop por defecto

En esta sección vamos permitir tipos de tráfico. Con DROP por defecto se va a preservar la máquina, dándole salida hacia internet para que la máquina pueda actualizarse con las consultas de DNS, Correo, etc.

A continuación se muestra un script en bash, haciendo usos de variables en bash.

```
#!/bin/bash

# Borrando reglas anteriores

iptables –F
```

```
iptables –t nat –F
iptables –Z
iptables –X

#Estableciendo política por defecto a DROP
iptables –P INPUT DROP
iptables –P OUTPUT DROP
iptables –P FORWARD DROP
iptables –t nat –p PREROUTING DROP
iptables –t nat –p POSTROUTING DROP

#Convertir la máquina Debian en un Router
echo 1 > /proc/sys/net/ipv4/ip_forward

#Declarar las constantes y variables
LAN= “192.168.10.0/24”
IFLAN=”eth1”
IFEXT=”eth0”

#Tráfico entrada y salida desde la máquina
#Permitir el trafico loopback en la máquina local
iptables –A INPUT –i lo –j ACCEPT
iptables –A OUTPUT –o lo –j ACCEPT

#Se permiten consultas DNS hacia internet
```

```
iptables –A INPUT –i $IFEXT –p udp—sport 53 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p udp—dport 53 –j ACCEPT

#Se permiten consultas HTTP y HTTPS hacia internet

iptables –A INPUT –i $IFEXT –p tcp—sport 80 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p tcp—dport 80 –j ACCEPT

iptables –A INPUT –i $IFEXT –p tcp—sport 443 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p tcp—dport 443 –j ACCEPT

#Se permite consultas DHCP

iptables –A INPUT –i $IFEXT –p udp –sport 67:68 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p udp—dport 67:68 –j ACCEPT

#Se permite solo el tráfico ICMP saliente y su respuesta

iptables –A INPUT –i $IFEXT –p icmp –m state –state ESTABLISHED, RELATED -j ACCEPT

iptables –A OUTPUT –o $IFEXT –p icmp –j ACCEPT

#La máquina Debian podrá realizar conexiones SSH

iptables –A INPUT –i $IFEXT –p tcp—sport 22 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p tcp—dport 22 –j ACCEPT

#La máquina Debian es un servidor SSH

iptables –A INPUT –p tcp—sport 22 –j ACCEPT

iptables –A OUTPUT –p tcp—dport 22 –j ACCEPT


#Tráfico Forward

#La máquina Debian tendrá permitido solo el tráfico ICMP saliente.

iptables –A FORWARD –p icmp -i $IFLAN –o $IFEXT  –j ACCEPT

iptables –A FORWARD –p icmp –o $IFLAN -i $IFEXT –j ACCEPT
```

#Se permite el tráfico HTTP y HTTPS

iptables –A FORWARD –p tcp—dport 80 -i $IFLAN –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 80 -o $IFLAN -i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 443 -i $IFLAN –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 443 –o $IFLAN -i $IFEXT –j ACCEPT

#La Máquina es servidor HTTP

iptables –A FORWARD –p tcp—sport 80 -i $IFLAN –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 80 -o $IFLAN -i $IFEXT –j ACCEPT

#Se permite consultas DNS

iptables –A FORWARD –p udp—dport 53 -i $IFLAN –o $IFEXT –j ACCEPT

iptables –A FORWARD –p udp—sport 53 -o $IFLAN -i $IFEXT –j ACCEPT

#NAT

#Hacemos DNAT para Windows XP. Puerto 80.

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 80 –j DNAT—to-destination 192.168.10.1

#Hacemos SNAT para la red interna

Iptables –t nat –A PREROUTING –s $LAN –o $IFEXT –j MASQUERADE

Bueno ya contamos con un esquema de firewall de 2 patas, solo hemos aplicado reglas para las interfaces, pero es importante mencionar que el esquema anterior es para una red doméstica, imaginemos que tenemos una red con varios servidores, entonces la configuración anterior no es muy segura, para darle mayor seguridad debemos de aplicar reglas con subredes/ hosts e interfaces, pero en un ambiente en entornos con servidores se debe implementar una configuración de firewall más avanzado y crear una DMZ.

2.7.13. Configurando un firewall de tres enlaces

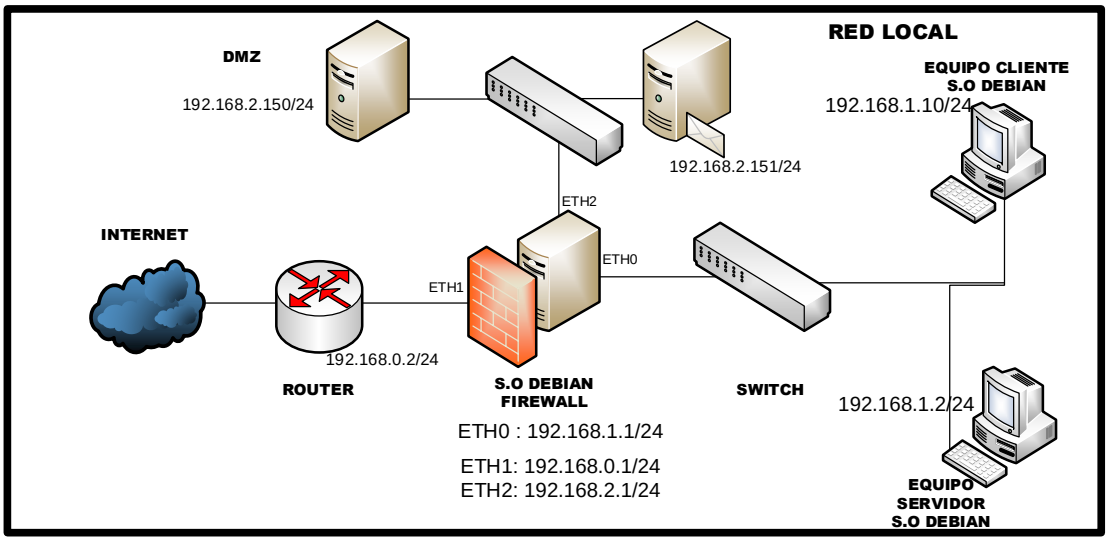

Figura 2.18. Esquema de Firewall de tres enlaces.

Ahora bien de acuerdo a esquema de red de la *figura 2.19* vamos a montar un esquema de firewall con tres enlaces donde vamos a configurar una red similar a la siguiente imagen. El firewall está ejecutando una Debian 8. Los elementos importantes son la máquina que hace de router y el servidor de la DMZ.

Vamos a considerar varias características previas, que nuestro firewall debe cumplir.

Políticas Drop.

Se permitirá el tráfico de loopback en la máquina firewall.

La máquina firewall podrá realizar conexiones SSH hacia internet.

La máquina firewall es un servidor SSH.

Accesible desde la LAN.

Accesible desde las direcciones IP 8.8.8.8 de Internet.

La máquina firewall tendrá permitido el tráfico ICMP. Responderá por todas las interfaces a dicho tráfico.

Desde la máquina firewall se pondrán iniciar conexiones SSH hacia el servidor de la DMZ.

La máquina firewall tiene conectividad como cliente a recursos HTTP, HTPPS y DNS.

La red DMZ tendrá acceso como cliente a los recursos HTTP, HTTPS y DNS.

Desde la red DMZ se acepta el tráfico ICMP saliente de tipo 0, 3, 8 y 11.

Desde la red DMZ se ofrecerán los siguientes servicios:

Servidor web en la IP 192.168.2.150. Se ofrecen webs seguras.

Servidor en la IP 192.168.2.151 ofrece servicios de correo SMTP, SMTPS, POP3, POP3S, IMAP3 e IMAPS.

Desde la red LAN se permitirá el acceso a recursos HTTP, HTTPS, DNS, FTP.

Desde la red LAN se acepta el tráfico ICMP saliente de tipo 0, 3, 8 y 11.

A continuación se muestra a modo de script cómo quedarían las reglas anteriores mediante iptables.

#!/bin/bash

Borrando reglas anteriores

iptables –F

iptables –t nat –F

iptables –Z

iptables –X

Estableciendo política por defecto a DROP

iptables –P INPUT DROP

iptables –P OUTPUT DROP

iptables –P FORWARD DROP

iptables –t nat –p PREROUTING DROP

iptables –t nat –p POSTROUTING DROP

Convertir la máquina Debian en un Router

echo 1 > /proc/sys/net/ipv4/ip_forward

Declarar las constantes y variables

LAN= “192.168.1.0/24”

DMZ= “192.168.2.0/24”

IFLAN=”eth0”

IFEXT=”eth1”

IFDMZ=”eth2”

Permitir el trafico loopback en la máquina local

iptables –A INPUT –i lo –j ACCEPT

iptables –A OUTPUT –o lo –j ACCEPT

Se permiten consultas DNS hacia internet

iptables –A INPUT –i $IFEXT –p udp—sport 53 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p udp—dport 53 –j ACCEPT

Se permiten consultas HTTP y HTTPS hacia internet

iptables –A INPUT –i $IFEXT –p tcp—sport 80 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p tcp—dport 80 –j ACCEPT

iptables –A INPUT –i $IFEXT –p tcp—sport 443 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p tcp—dport 443 –j ACCEPT

Se permite consultas DHCP

iptables –A INPUT –i $IFEXT –p udp –sport 67:68 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p udp—dport 67:68 –j ACCEPT

Se permite solo el tráfico ICMP saliente y su respuesta

iptables –A INPUT –i $IFEXT –p icmp –m state –state ESTABLISHED, RELATED -j ACCEPT

iptables –A OUTPUT –o $IFEXT –p icmp –j ACCEPT

Se permite tráfico ICMP por todas las interfaces

iptables –A INPUT –p icmp –j ACCEPT

iptables –A OUTPUT –p icmp –j ACCEPT

La máquina Debian podrá realizar conexiones SSH

iptables –A INPUT –i $IFEXT –p tcp—sport 22 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –p tcp—dport 22 –j ACCEPT

Se permite e tráfico SSH hacia la DMZ

iptables –A INPUT –i $IFDMZ -s $DMZ –p tcp—sport 22 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –d $DMZ –p tcp—dport 22 –j ACCEPT

La Máquina es servidor SSH

iptables –A INPUT –i $IFLAN –p tcp—dport 22 –j ACCEPT

iptables –A OUTPUT –o $IFLAN –p tcp—sport 22 –j ACCEPT

iptables –A INPUT –i $IFEXT –s 8.8.8.8.8 –p tcp—dport 22 –j ACCEPT

iptables –A OUTPUT –o $IFEXT –d 8.8.8.8 –p tcp—sport 22 –j ACCEPT

DMZ

Se permite todo tipo de tráfico ICMP con origen DMZ y sólo alguno con destino DMZ.

iptables –A FORWARD –i $IFDMZ -s $DMZ –p icmp –j ACCEPT

iptables –A FORWARD –o $IFDMZ –d $DMZ –p icmp –icmp-type destination-unreachable –j ACCEPT

iptables –A FORWARD –o $IFDMZ –d $DMZ –p icmp –icmp-type time-exceeded –j ACCEPT

iptables –A FORWARD –o $IFDMZ –d $DMZ –p icmp –icmp-type echo-reply –j ACCEPT

iptables –A FORWARD –o $IFDMZ –d $DMZ –p icmp –icmp-type echo-request –j ACCEPT

Se permite el tráfico HTTP y HTTPS

iptables –A FORWARD –p tcp—dport 80 -i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 80 -o $IFDMZ -i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 443 -i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 443 –o $IFDMZ -i $IFEXT –j ACCEPT

Se permite consultas DNS

iptables –A FORWARD –p udp—dport 53 -i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p udp—sport 53 -o $IFDMZ -i $IFEXT –j ACCEPT

#La IP 192.168.2.150 es servidor web (http, https)

iptables –A FORWARD –p tcp—sport 80 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 80 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 443 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 443 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

La IP 192.168.10.151 es servidor de correo

iptables –A FORWARD –p tcp—sport 25 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 25 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 465 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 465 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 110 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 110 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 995 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 995 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 220 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 220 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 993 –s $DMZ-i $IFDMZ –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 993 -d $DMZ -o $IFDMZ –i $IFEXT –j ACCEPT

LAN

Se permite todo tipo de tráfico ICMP con origen LAN y solo alguno con destino LAN

iptables –A FORWARD –i $IFLAN –s $LAN –p icmp –j ACCEPT

iptables –A FORWARD –o $IFLAN –d $LAN –p icmp –icmp-type destination-unreachable –j ACCEPT

iptables –A FORWARD –o $IFLAN –d $LAN –p icmp –icmp-type time-exceeded –j ACCEPT

iptables –A FORWARD –o $IFLAN –d $LAN –p icmp –icmp-type echo-reply –j ACCEPT

iptables –A FORWARD –o $IFLAN –d $LAN –p icmp –icmp-type echo-request –j ACCEPT

Se permite el tráfico HTTP y HTTPS

iptables –A FORWARD –p tcp—dport 80 -i $IFLAN –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 80 -o $IFLAN -i $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 443 -i $IFLAN –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—sport 443 –o $IFLAN -i $IFEXT –j ACCEPT

Máquina es servidor HTTP

iptables –A FORWARD –p tcp—sport 80 -i $IFLAN –o $IFEXT –j ACCEPT

iptables –A FORWARD –p tcp—dport 80 -o $IFLAN -i $IFEXT –j ACCEPT

NAT

Hacemos DNAT para Windows Puerto 80.

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 80 –j DNAT—to-destination 192.168.2.150

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 443 –j DNAT—to-destination 192.168.2.150

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 25 –j DNAT—to-destination 192.168.2.151

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 465 –j DNAT—to-destination 192.168.2.151

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 110 –j DNAT—to-destination 192.168.2.151

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 995 –j DNAT—to-destination 192.168.2.151

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 220 –j DNAT—to-destination 192.168.2.151

Iptables –t nat –A PREROUTING –i $IFEXT –p tcp—dport 993 –j DNAT—to-destination 192.168.2.151

Hacemos SNAT para la red interna, DMZ y LAN

Iptables –t nat –A PREROUTING –s $LAN –o $IFEXT –j MASQUERADE

Iptables –t nat –A PREROUTING –s $DMZ –o $IFEXT –j MASQUERADE

2.8. VPN

Es una red privada edificada en el interior de un esquema de red pública, como puede ser Internet. La VPN apoya a las instituciones a agrandar la

conectividad de manera protegida, económica y nos ayuda a mejorar la velocidad.

Una red VPN nos brinda la cúspide nivel de seguridad mediante IPsec o SSL, y tecnologías de autenticación, permitiendo proteger los paquetes que viajan a través de la VPN

Los dos tipos de Implementaciones para VPN son:

VPN punto a punto. Se usa para enlazar 2 sedes remotas.

VPN de acceso remoto. También conocida como Road Warrior. Este esquema es usado cuando equipos remotos desean conectarse a la red interna.

Dependiendo del esquema de implementación y tecnología usada para instaurar un entorno VPN se alcanzará beneficios que los vamos a detallar a continuación:

- ***Autenticación***. El usuario para poder acceder a los recursos de red debe primero identificarse.
- ***Confidencialidad***. Mediante el intercambio de claves de sesión se podrá cifrar la comunicación para que aquellos extremos que intervienen en la VPN puedan descifrar los datos.
- ***Integridad***. Se puede verificar que los datos no hayan sido alterados mediante el uso de varios algoritmos de hash.
- ***No repudio***. Al existir un proceso de autenticación, es lógico que todas las acciones que se realicen queden registradas y asociadas a un determinado usuario o máquina.

Existen diversas implementaciones para crear VPN. En este caso se ha elegido PPTP dado su popularidad y OpenVPN. Se trata de dos métodos muy conocidos y extendidos en entornos GNU/Linux.

PPTP, Point-to-point Tunneling Protocol

PPTP, Point-to-point Tunneling Protocol, es usado en soluciones VPN, es muy fácil de configurar y es muy compatible con diferentes plataformas. Para PPTP usamos una conexión TCP, escuchando el servidor por el puerto 1723. Dicha conexión es utilizada como inicialización del protocolo GRE, Generic Routing Encapsulation, destinado al enrutado del tráfico tunelizado. Los mecanismos anteriores permiten la creación de un túnel, pero no el cifrado ni autenticación del mismo. Para esta última tarea se utiliza e protocolo PPP, Point-to-Point Protocol.

PPP ofrece diferentes mecanismos de autenticación. Los más destacado y utilizados son MSCHAPv2 y EAP-TLS. El primero para clave precompartida o preshared key y el segundo utilizando una infraestructura PKI, Public Key Infraestructure.

Openvpn

OpenVPN es una solución de tecnología VPN equiparable a la tan extendida IPsec. Se trata de un proyecto open-source que utiliza como base las librerías OpenSSL para implementar sus mecanismos de cifrado y autenticación de usuarios o máquinas. A diferencia de IPsec, OpenVPN es una aplicación que se ejecuta en el nivel del usuario sin ser necesarias modificaciones a nivel de kernel y pila IP.

2.9. Monitoreo de la red

A la hora de administrar cualquier tipo de sistema resulta indispensable conocer en todo momento la actividad de la misma. Por ejemplo, inicios de sección, accesos a determinados recursos e intentos de ataque, toda esta información dará al administrador de sistema una visión global de lo que sucede en su entorno. A ese registro de eventos o sucesos conocemos como logging. Son diferente los servicios y utilidades del sistema entre otros, los componentes que generan dichos eventos.

2.9.1. Consideraciones Previas

Existe una serie de aspectos que deberían considerarse antes de configurar sistemas para la gestión de logs o eventos.

Almacenamiento Independiente.

Los logs se almacenan de forma predeterminada en ficheros de texto plano. Su tamaño y crecimiento dependerá del número de aplicaciones que generen logs, carga de la máquina y los servicios ofrecidos, además de la información que se esté registrando. Es recomendable por lo tanto aislar el sistema de ficheros en el que se almacenan los logs para evitar colapso en particiones raíz.

Rotación de logs.

Por defecto los logs, cuando son almacenados en ficheros de texto planos, se almacenan de forma secuencial. Es decir, el fichero de log irá creciendo indefinidamente. Ese comportamiento por defecto realiza generación de ficheros de gran capacidad, perdida de rendimiento, además de perdida de eficiencia en caso de necesidad de realizar búsquedas.

Métodos Logging de Aplicaciones.

Es necesario tener en cuenta como ciertas aplicaciones registran sus eventos en el sistema. Como se verá posteriormente existen diferentes ficheros de log estándar donde se almacenan cierta información sobre diferentes tipos de eventos disparados por una aplicación o mecanismo de sistema. Dependiendo del tipo de aplicación es posible que la información se reparta de forma integral en ficheros utilizando un mecanismo de log local como rsyslog, que se utilicen mecanismos y ficheros propios de la aplicación para registrar evento, o bien que sea una combinación de ambos. Es necesario atender a este tipo de excepciones sobre todo a la hora de rotar logs o implementar un sistema centralizado de logs.

Alerta

Es posible realizar análisis en los logs de forma periódica en busca de ciertos patrones, siendo posible el envío a correos ante determinadas situaciones.

Infraestructura de Logs.

Dependiendo del número de máquinas a analizar mediante log es posible que sea prácticamente imposible revisarlos en cada uno de ellas. Es por ello que existe la posibilidad de centralización de logs, almacenamiento mediante otros mecanismos y no en ficheros de texto, análisis mediante aplicaciones web, etc.

2.10. Rsyslog

Este servicio se encarga del manejo de logs del sistema. Su función consiste en organizar los eventos generados por determinados servicios o aplicaciones en diferentes contenedores. Lo contenedores podrán ser ficheros de texto o bases de datos. La organización por su parte se realizará atendiendo a una serie de parámetros indicados por la propia aplicación que lance el evento, *rsyslog* recogerá entonces esa información y dependiendo de su configuración almacenará dicho evento en ficheros.

Existen varias alternativas para la gestión de logs en un sistema operativo Linux, sin embargo se elige rsyslog por su estandarización en las distribuciones más importantes. Entre ellas Debian y Red Hat.

2.10.1. Clasificación de mensajes Facility y Severity.

Para una organización de logs, rsyslog recibirá mensajes por parte de las aplicaciones que utilicen la API syslog. Para ello, como ya se ha comentado, dichas aplicaciones especificaran una serie de parámetros en los eventos que generen, con la finalidad de que syslog pueda clasificarlos mediante reglas que se configuren.

Facility

Debe entenderse como clasificación para los mensajes. Existen una serie de facilities predeterminadas como cron, kern, daemon, mail, authpriv, auth, que serán utilizadas por la aplicación dependiendo el tipo de mensaje que se genere. Además de esas facilities predeterminadas existen otras que podrán ser utilizadas libremente por el administrador de sistemas. Son las

facilities local*N* donde *N* equivale a los niveles de 0 a 7 de acuerdo a la tabla a continuación.

Level numbers	Severity level	Description
0	Kernel	Mensajes del kernel
1	User level	Mensajes de nivel de usuario
2	Mail system	Sistema del correo
3	System daemon	Demonios del sistema
4	Security/authorization messages	Mensajes de autorización y seguridad.
5	Messages internal syslog	Mensajes internos de syslog
6	Line printer subsystem	Subsistema de impresión.
7	Network new subsystem	Subsistema de noticias de red.

Tabla 5. Niveles de Facility Log

Severity

Indica el nivel de gravedad o seriedad de un evento. Este indicativo ayuda a diferenciar los mensajes informativos de los mensajes de emergencia. Algunas de las severities existentes se muestran en la tabla siguiente:[25]

Level numbers	Severity level	Description
0	Emergencies	System is unusable
1	Alert	Immediate action is needed
2	Critical	Critical conditions
3	Error	Error conditionds
4	Warning	Warning conditions
5	Notification	Normal but significant conditions
6	Informational	Informational messages only.

7	Debugging	Debugging messages only.

Tabla 6. Niveles de Severity Log

Cada entrada en la configuración de syslog está compuesta por una acción facility.priority. Podemos utilizar ciertos comodines para la configuración:

(,): Separar una instalación o prioridad en la misma regla.

(*): Todas las posibilidades.

(=): Otorga exclusividad a una instalación o prioridad .

(!): Excluir una instalación o prioridad a una regla.

(!=) : Excluir solamente la prioridad seleccionada .

(;): Separa un selector facility.priority por la misma regla de salida.

Las acciones posibles podrían ser:

Usuarios

Archivos

Los servidores remotos

Por ejemplo la siguiente configuración:

Local1.info */var/log/int_info.log*

En la siguiente línea de comando se indica que todos los eventos con facility *local1* y severity *info* sean almacenados de forma secuencial en el fichero /var/log/int_info.log.

2.11. Parámetros de Rsyslog

Para configurar el comportamiento de rsyslog se utiliza principalmente el fichero */etc/rsyslog.conf.*

Para el caso de Debian de forma adicional pueden utilizarse los ficheros contenidos en /etc/rsyslog.d para crear configuraciones fragmentadas o por servicios.

El fichero principal de configuraciones se divide en cuatro partes:

Directivas Globales: Se define el comportamiento del demonio rsyslog. Es donde se cargan los módulos para agregar características, se establecen los permisos de los ficheros de log creados.

Plantillas o template. Permite modificar la salida de los mensajes de log atendiendo a diferentes parámetros como son el nombre del programa, host, etc. También es posible utilizarlas con la finalidad de crar ficheros de log dinámicos por cada máquina.

Canales de salida, output Channels: definen canales que podrán ser utilizados posteriormente en los filtros o clasificación de los mensajes en lugar de utilizar un nombre de fichero por ejemplo.

Reglas: Se indica la acción a realizar cuando se hace coincidir una facility y severity.

Las acciones pueden ser el almacenamiento en ficheros, en la maquina remota con rol de servidor rsyslog, templates, etc.

Las siguientes líneas de la *figura 2.20* corresponden a reglas, las mencionadas combinaciones de filtros y acciones.

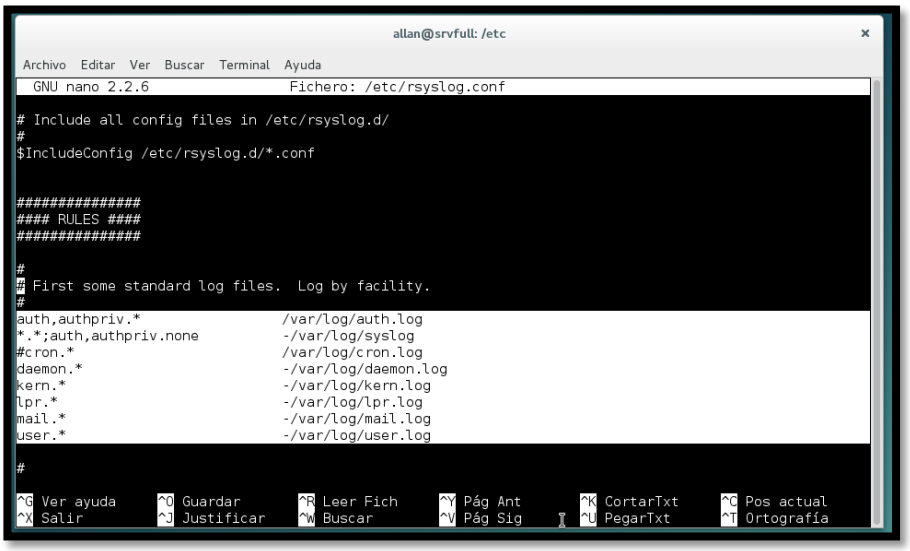

Figura 2.19: Fichero de configuración de Rsyslog.

Explicación:

En la primera línea se indica que los eventos relacionados con la autenticación del sistema se almacenen en el fichero /var/log/auth.log. Es decir, son los eventos que genera las aplicaciones con facility auth o authpriv y todas las severity de ambas facility.

En la segunda línea se introduce el símbolo “ ; ” que delimita conjuntos de filtros facility.severity. El símbolo “ , ” por su parte permite agrupar varias facilities para indicar una sola severity. El carácter “ * ” comodín que agrupa todas. Por otra parte el símbolo “-” indica que no se utilice la sincronización de forma inmediata para la escritura en el disco.

La cuarta línea se trata de todos los eventos de la facility daemon serán escritos en el fichero /var/log/daemon.log y así sucesivamente los demás facilities.

Para estos instantes una vez que tenemos ya gestionado y direccionado los almacenamientos de los logs dependiendo a la configuración. Dicho almacenamiento se realizara de forma secuencial y en principio no existirá un límite para el crecimiento de los ficheros. Por lo cual vamos a proceder a implementar un mecanismo para rotar los ficheros de logs.

2.12. Rotación de logs

Existe una herramienta open source llamada logrotate que está instalada por defecto en Debian. En este punto vamos a explicar el funcionamiento de logrotate mediante algunos de los ficheros de configuración, que conllevará a una rotación de los logs del sistema desde el momento en el que este se instala.

La configuración de logrotate está compuesta por el fichero principal /etc/logrotate.conf y los ficheros contenidos dentro del directorio /etc/logrotate.d/. En dichos ficheros se indican características como el intervalo entre la rotación de los logs, el tamaño máximo que deben ocupar, el número de logs almacenados tras cada rotación, etc.

Una vez establecida una configuración utilizando los ficheros de logrotate, será necesario configurar una tarea automática para ejecutar logrotate y l configuración establecida. Por defecto se utiliza el fichero /etc/cron.daily/logrotate que contiene las siguientes líneas. Ej. *Figura.2.21*

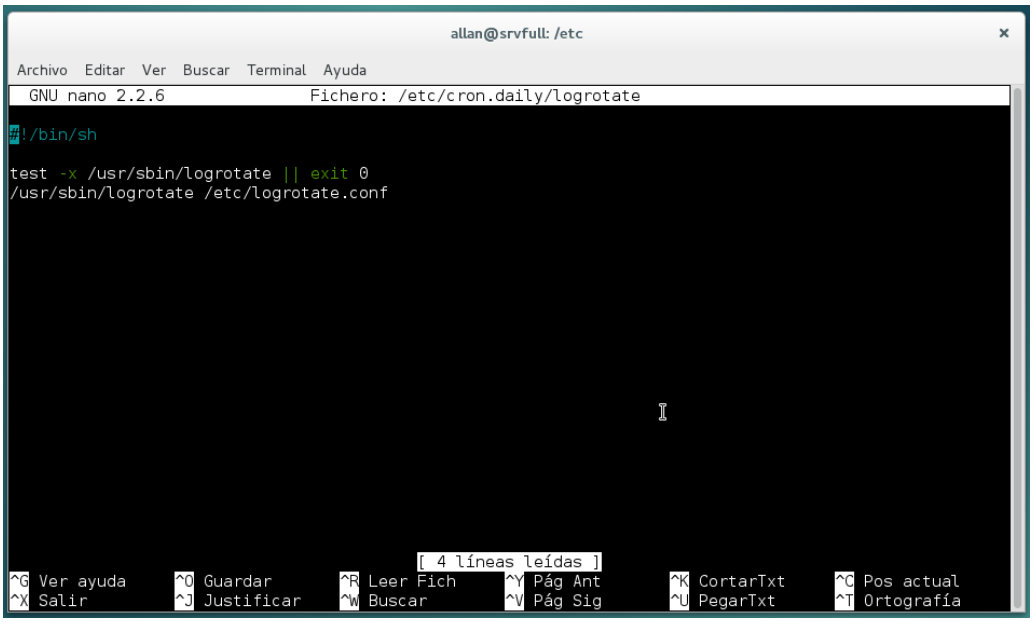

Figura 2.20: Fiche de configuración de tareas de Logrotate.

Es decir, de forma diaria se ejecutará el script mostrando en el que se ejecuta la aplicación logrotate utilizando la configuración contenida en /etc/logrotate.conf.

2.12.1. Ficheros de configuración general de Logrotate.

Existe una configuración global que afectará a todos los ficheros contenidos bajo el directorio */var/log*, sin embargo la mayoría de las aplicaciones que escriben sus propios ficheros log o utilizan rsyslog agregan configuraciones propias para la rotación de logs, para ellos debemos configurare el archivo de la siguiente ruta: */etc/logrotate.conf* mostrado en la *figura 2.22*

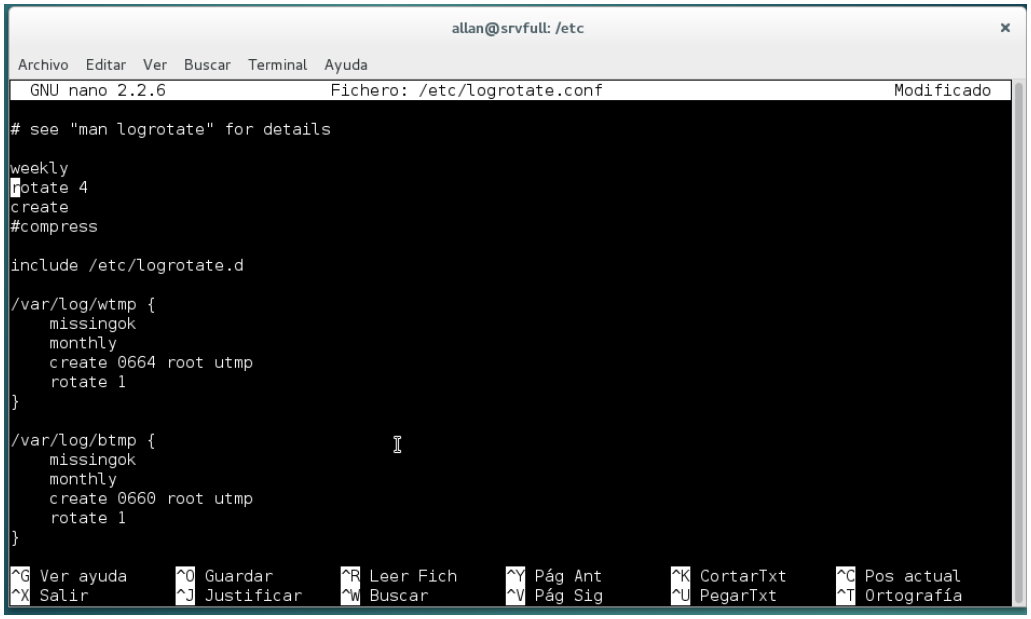

Figura 2.21. Fichero de configuración de Logrotate.

En este fichero se define una configuración global, siendo el siguiente:

Los tres primeros parámetros corresponden a la configuración general para todos los ficheros contenidos en /var/log.

Weekly: indica que los ficheros se rotan semanalmente. Esto hará que cada semana el fichero de log actual sea renombrado.

Rotate 4: indica que como máximo existirán cuatro ficheros de log rotados.

Create: indica que al renombrar el fichero de log se creará uno nuevo para que rsyslog continúe trabajando con normalidad.

Por su parte, la línea ***include /etc/logrotate.d*** permite leer la configuración de los ficheros contenidos en el directorio indicado. Dentro de él se encontrarán configuraciones segmentadas y propias de algunas aplicaciones o servicios, como por ejemplo ocurre con el servidor web apache.

Los dos últimos bloques establecen las opciones para rotar los ficheros binarios de log ***/var/log/wtmp y /var/log/btmp.*** En ambos casos se rotarán mensualmente, almacenando solo una rotación e ignorando en caso que el fichero de log no exista con la directiva *missingok*. Adicionalmente se crearán los ficheros con los permisos indicados mediante *créate.*

Alguna de las opciones que también se pueden incluir y que no son usadas en la configuración global son:

Compress. El cual hace que antiguas versiones de logs sean comprimidas para reducir el espacio.

Copy. Crea una copia del log generado, ideal para realizar backup de los logs o cuando estos son consultados por otras aplicaciones, y se quiere evitar al máximo la interacción con el log original.

Size. Establece el tamaño del archivo que se tiene en cuenta para poder realizar la roacion del log, podríamos definir tamaños con las letras M (megabytes) y K (kilobytes) haciendo que se pueda mantener un control del tamaño de los logs.

El administrador podrá crear sus propias configuraciones o bien modificar las ya existentes. Cuando se trata de almacenamiento de log se debe tener en cuenta el periodo de almacenamiento de los mismos según dictan ciertas leyes de protección de datos en caso de manejar datos de usuarios, acceso público, etc. [26].

2.13. Implementando Rsyslog

Para entender cómo funciona se muestra un ejemplo para un fichero de log llamado /var/log/int_info.log, que nunca debería superar los 1k. Para *rsyslog* se creará un fichero de configuración /etc/rsyslog.d/conf con el siguiente contenido ver *figura 2.23*.

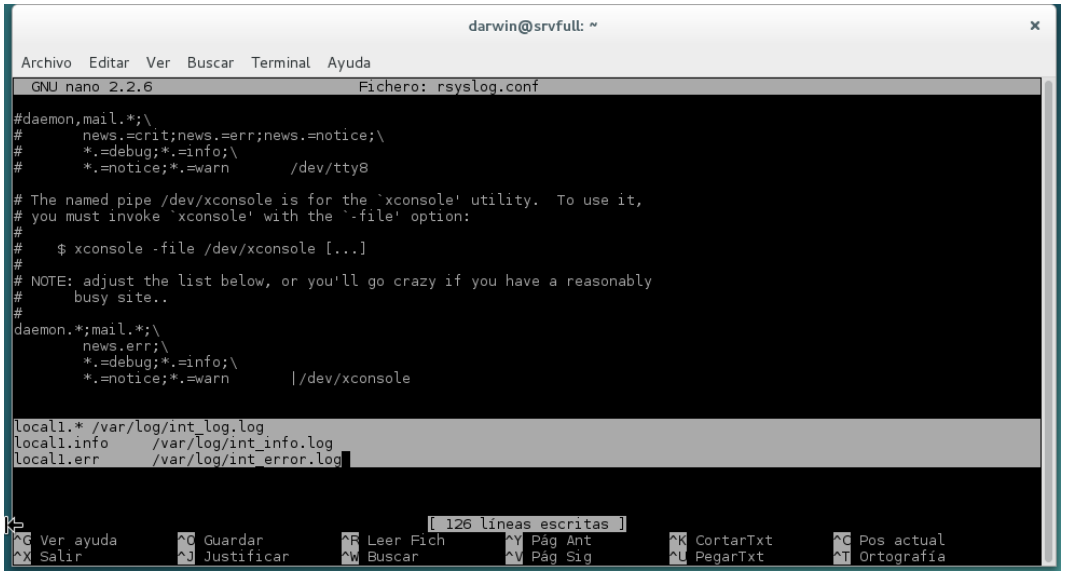

Figura 2.22. Creando Políticas de registro de Log.

Para logrotate se creará el fichero de configuración /etc/logrotate.d/integradora. Ver *figura 2.24*

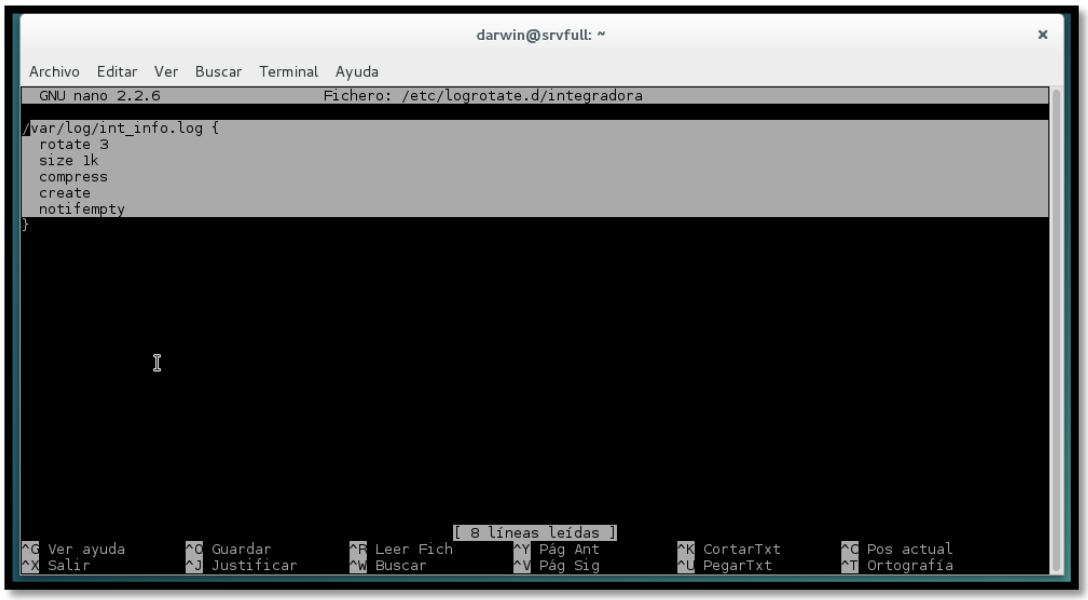

Figura 2.23. Configuración de parámetros para un logrotate específico.

Es decir, cuando rsyslog escriba 1k en el fichero de log se lanzará *logrotate* utilizando la configuración /etc/logrotate.d/integradora, mostrando el siguiente resultado. Ver *figura 2.25*

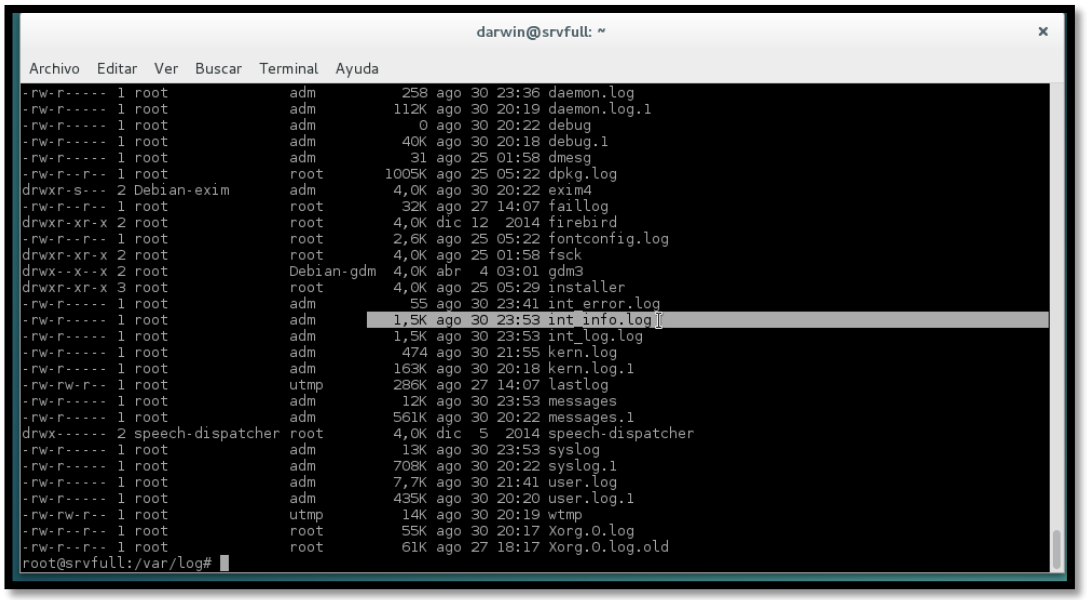

Figura 2.24. Verificando los ficheros de log.

Con la configuración establecida el fichero de log será rotado si el tamaño alcanza más de 1k., y será rotado hasta tres veces, creara un nuevo log y le realizará compresión de paquete. El resultado obtenido será el siguiente. Ver *figura 2.26.*

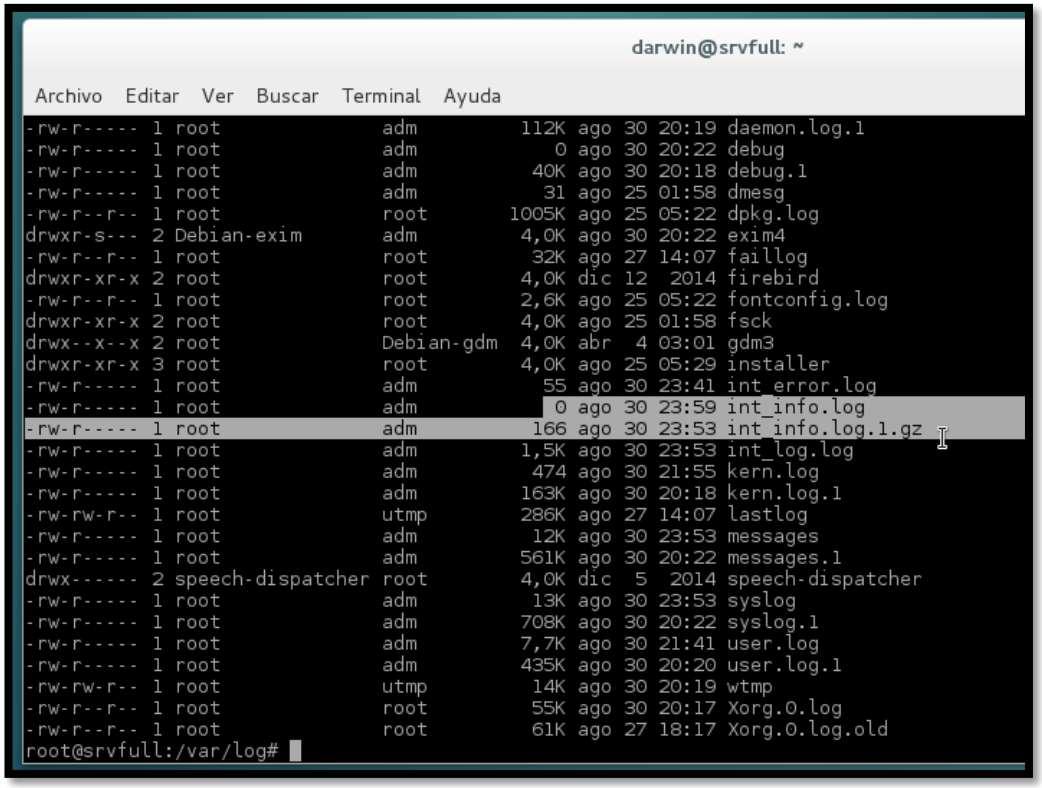

Figura 2.25: Comprobación de Log rotados.

Por lo tanto ya se ha conseguido que los log vayan rotando conforme a una serie de características, incluso pudiendo ser controlando el tamaño en tiempo real según esta última configuración. Incluso en entornos pequeños puede resultar pesado e inabordable, por lo que resulta necesario centralizar los *logs.*

2.14. Logging remoto o centralizado

Una de las características esenciales en los sistemas de logging como rsyslog es sin duda la posibilidad de poder enviar los eventos hacia otra máquina de forma remota y en tiempo real. Esto permitirá crear un sistema centralizado de logs.

Utilizar esta opción en rsyslog resulta muy beneficiosos, tan solo habilitando el modulo correspondientes para permitir conexiones remotas en la máquina que actué como servidor de logging, así como el envío de logs hacia la maquina remota desde el cliente.

Para lo cual procedemos a la configuración de logging remoto de un entorno construido por un modelo servidor cliente.

Datos:

Maquina A. actuará como servidor centralizado rsyslog. Se utilizará UDP para la recepción de mensajes utilizando el puerto estándar, 514. La dirección IP de esta máquina será 192.168.9.50. El hostmane de esta máquina es Test 1

Maquina B: Será configurada como cliente remoto rsyslog. La dirección IP en este caso será 192.168.9.25. El hostname para esta máquina será Tes2.

Configuración de máquina A

En el fichero de configuración por defecto */etc/rsyslog.conf. Ver figura 2.27* se muestra comentadas las líneas que proporcionan la funcionalidad de recepción de *logs.* Una vez descomentadas quedarían de la siguiente manera:

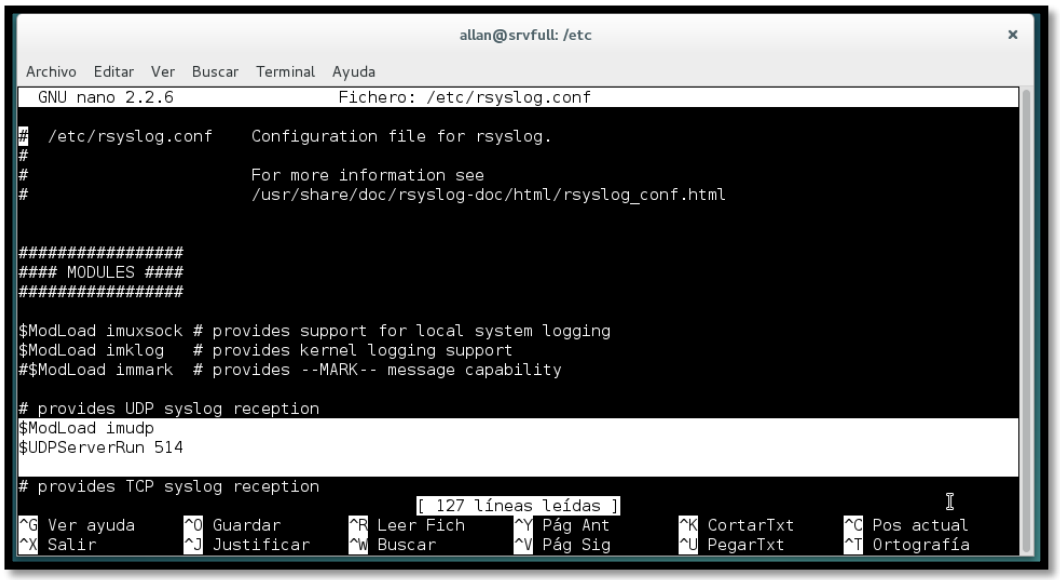

Figura 2.26: Habilitando el servicio de Rsyslog remoto.

Tras reiniciar el servicio rsyslog debería existir un socket de escucha para las conexiones rsyslog entrantes. Puede comprobarse mediante el comando *netstat* en la consola de terminal. Ver *figura 2.28.*

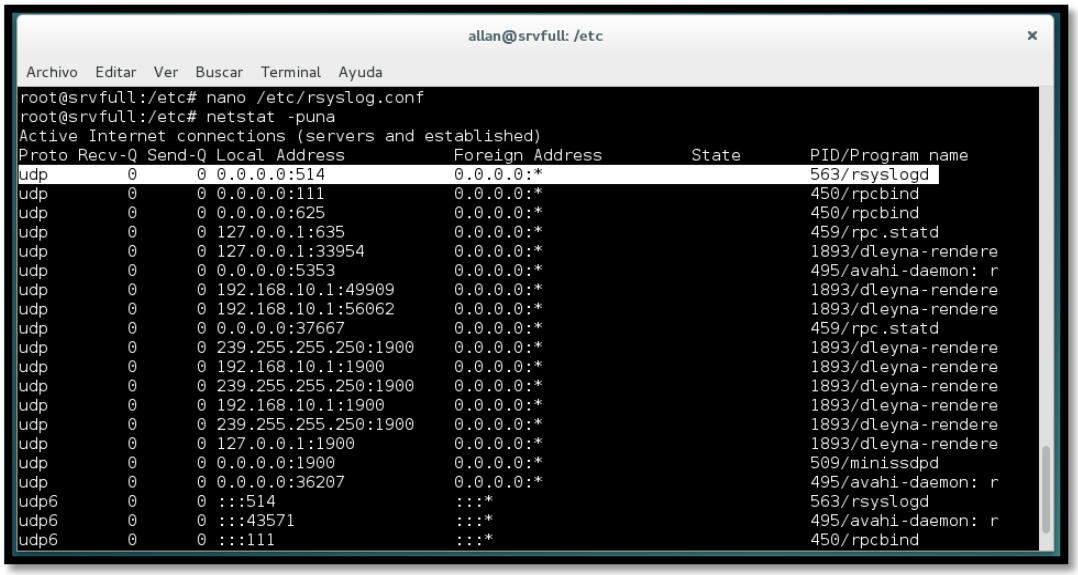

Figura 2.27: Servicio UDP de Rsyslog activo.

Configuración de maquina B

La configuración de la maquina cliente dependerá de los logs que deseen enviarse al sistema remoto, atendiendo a sus facilities por ejemplo para activar el envío de todos los mensajes a la maquina remota. Para ese caso debería agregarse la siguiente línea al fichero /etc/rsyslog.conf. Ver *figura 2.29*

Figura 2.28: Configurando Rsyslog en máquina cliente.

Es decir, se enviaran los eventos de todas las facilities a la máquina remota 192.168.9.50. Por supuesto en la maquina local seguirán almacenándose los ficheros de acuerdo a las configuración establecida. Una vez reiniciado el *rsyslogd* podría verificarse el funcionamiento utilizando el comando *logger* en la máquina B, creando un log de tipo local1.info. “Login de Pc” para verificar que se está creando el log y constar que la gestión de envío de log hacia servidor centralizado. Ver *figura 2.30.*

Figura 2.29: Comando Logger para generación de log de pruebas.

Una vez esto ingresamos a la maquina 1 para verificar que se encuentra ya el registro del log, que de acuerdo a nuestras reglas que creamos al ser un archivo de tipo local1 debe almacenarse en nuestro log *int_info.log.*

Para ellos nos vamos a la ruta de almacenamiento de log /var/log/int_info.log y consultamos con el comando *cat al archivo int_info.log.*

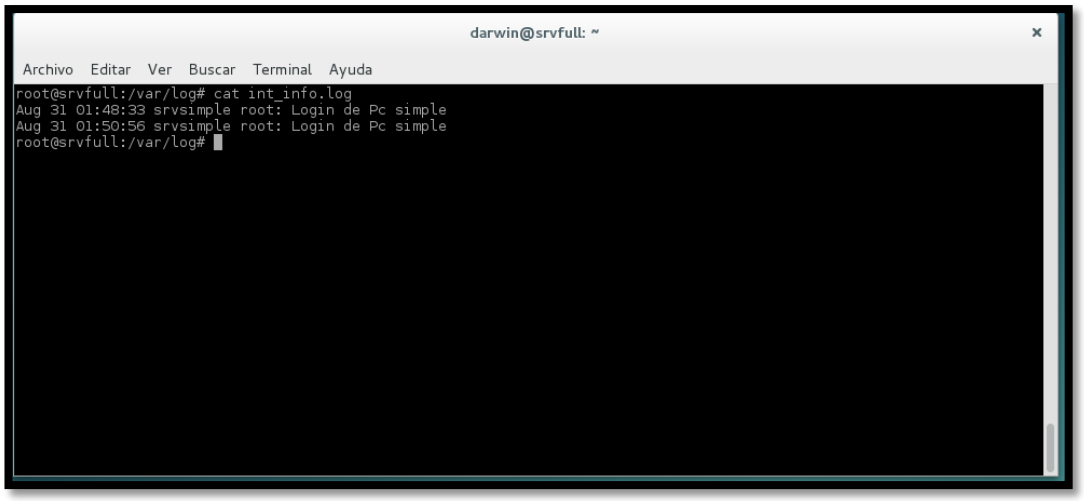

Figura 2.30: Verificación de Log Generados en máquina cliente.

Como se muestra en la *figura 2.31* tenemos que se ha registrado un log con origen en el equipo *srvsimple* y el usuario que se estaba root empleando al momento de crearlo en nuestra maquina B para la práctica.

Se entiende que las configuraciones y parámetros a establecer estarán basados a las necesidades de cada administrador de sistemas.

CAPÍTULO 3

3. RESULTADOS DE ASEGURAR NUESTROS EQUIPOS

La idea de la selección de un sistema operativo de distribución GNU/Linux es por su principal beneficio de controlar el funcionamiento de su equipo y asegurar el comportamiento de todos sus programas con los sistemas de gestión de permisos.

Al usar herramientas Open Source nos permite adaptarnos a las necesidades actuales, y una ventaja de aprender en comunidad, sin la necesidad de presupuestar el coste de mantenimiento de software y personal encargado, dado que se puede obtener repositorios oficiales con esquemas, consejos o platillas de trabajo para poder implementar en nuestras empresas.

3.1. Protección arranque

Normalmente evitamos el acceso no autorizado a nuestros equipos, creamos contraseñas seguras, ACL, configuramos firewall; etc.; pero muy poco le prestamos atención a proteger el arranque de nuestros equipos.

Atendiendo a los resultados obtenidos en la sección anterior con los ataques, para evitar esto resulta necesario asegurar GRUB mediante la utilización con contraseñas para evitar el acceso a las entradas de arranque de GRUB, así como el acceso a la consola de GRUB. Éste es uno de los puntos donde se diferencian GBRUB y GRUB2.

En definitiva, si no protegemos el gestor de arranque, estaremos dejando un agujero de seguridad muy grande permitiendo al usuario malintencionado tener control absoluto del equipo de manera fácil y rápida.

3.2. Firewall

Ahora que ya poseemos información para empezar a configurar las reglas que decidirán la aprobación o la negación de entrada y salida de paquetes de red de nuestra máquina local, de nuestra propia red local o a través de Internet. Ahora queda establecer políticas de acuerdo a los requerimientos y procedimientos establecidos que se rigen en la pymes que requiera implementarse.

Es muy importante tener mucha atención con el ordenamiento de las reglas porque iptables interpreta de manera secuencial las cadenas de reglas. Por ejemplo si la primera regla es descartar cualquier paquete desde la cadena INPUT, las posteriores reglas no serán analizadas y se descartará cualquier paquete.

3.3. VPN

La evolución de la tecnología nos presenta nuevas formas de trabajo en las empresas, brindando un revestimiento de forma local o remota, como por ejemplo actualmente las organización están implementado el trabajo remoto, y por ello que se implementa la VPN para que los usuarios fuera de la empresa puedan acceder desde cualquier lugar al entorno de trabajo y de esa manera no perjudica la seguridad de la empresa.

Entre las ventajas y funcionalidades que se pueden destacar están las siguientes:

Solución robusta y segura. Se utilizan tecnologías más que testadas con lo que se asegura que los procesos de cifrado, autenticación e interacción con el sistema sean muy sólidos.

Gran flexibilidad de configuración. Es posible utilizar scripts y diversas configuraciones según el momento o estado de la conexión. Esto último resulta muy útil para los clientes cuando se encuentran en entornos públicos, así como entornos empresariales donde las configuraciones de los firewall son muy estrictas

3.4. LOG

Cada reporte de las herramientas de seguridad, nos facilita la gestión y los eventos, además nos permite ahorro de tiempo y esfuerzo por parte de los Administradores de los sistemas.

El saber gestionar el almacenamiento de información que se genera en los equipos ayuda a mantener un estado mejor rendimiento de discos y tiempo de respuesta de los procesos o servicios que utilizamos concurrentemente,

también ayuda a mantener una organización de estado y novedades que puedan estar afectando al entorno de red.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. La protección del sistema de arranque ayuda a prevenir de accesos físico desde medios removibles y control de los parámetros de arranque y acceso y en integración con un cifrado de sistemas de ficheros, permite la proteger los datos de los sistemas de archivos incluso si es removido físicamente los discos
2. La implementación de seguridad en el perímetro de la red, consiste en implementar políticas de seguridad en los equipos de comunicación, los equipos son instalados entre la red interna y la externa, por lo cual la adquisición de un equipo firewall genera costos elevados para una red pequeña que desea mantener una seguridad del tráfico de la red, por lo cual implementar *Iptables* reduce costos y se mantiene el mismo servicio de filtrado de paquetes.
3. Saber que monitorizar el contenido de los logs de los servidores a medida que progresan es fundamental para detectar y solucionar problemas de forma preventiva, además el detectar velocidades de crecimiento del log inusuales es una característica que le ayudará a saber si su servidor está funcional y proporcionando servicio a un ritmo normal.

Recomendaciones

4. La implementación no garantiza que el sistema resultado de la integración sea 100% seguro. Para aumentar el porcentaje debemos usar un sistema de mejora continua en referente a las nuevas técnicas que van apareciendo en las tecnologías de la información.

BIBLIOGRAFÍA

[1] Brahima Sanou, (2014, Septiembre). International Telecomunication Union, Facts and Figures, [Online] http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx.

[2] Mohanbir Sawhney, Revista Ekos página 149, http://www.ekosnegocios.com/negocios/REV_paginaEdicion.aspx?edicion=223, fecha de publicación Nov 2012

[3] David Drummond, (2010, Enero), Desarrollo Corporativo y Jefe de Asuntos Legales Google Inc., [Online] http://googleamericalatinablog.blogspot.com.es/2010/01/un-nuevo-enfoque-para-china.html

[4] Wikipedia, (2011, Agosto). Operación Aurora [Online] https://en.wikipedia.org/wiki/Operation_Aurora

[5] Segu-Info (2012, Marzo) Seguridad Física. [Online] http://www.segu-info.com.ar/fisica/seguridadfisica.html

[6] Wikipedia, (2014, Diciembre) PDP-11. [Online]. https://es.wikipedia.org/wiki/PDP-11

[7] Wikipedia, (2015, Septiembre 5). Historia de los sistemas operativos. [Online]. https://es.wikipedia.org/wiki/Historia_de_los_sistemas_operativos

[8] Unix The Open Group. (2012). History and Timeline. [Online]. http://www.unix.org/what_is_unix/history_timeline.html

[9] Wikilibros, (2011, Enero 3). Seguridad Informática [Online]. https://es.wikibooks.org/wiki/Seguridad_inform%C3%A1tica/Lo_que_afecta_a_las_distribuciones_de_Gnu/Linux

[10] Debian, (2015, Julio 9). Acerca de Debian [Online]. https://www.debian.org/intro/about#what

[11] Slackware, (2005, Mayo 11). Slackware Linux Essentials. [Online]. http://slackbook.org/html/installation-requirements.html

[12] Wikipedia, (2015, Agosto 26). Red Hat. [Online]. https://es.wikipedia.org/wiki/Red_Hat

[13] Guillermo Rigotti, Universidad Nacional del Centro de Buenos Aires, (2012, Diciembre 11). Capas del Modelo OSI. [Online]. www.exa.unicen.edu.ar/catedras/comdat1/material/ElmodeloOsi.pdf

[14] Red Hat (2007, Marzo 3) Seguridad del BIOS y del gestor de arranque. [Online].

http://lists.openshift.redhat.com/docs/manuals/enterprise/RHEL-5-manual/es-ES/Deployment_Guide/s1-wstation-boot-sec.html#

[15] Pedro Fábrega Martínez, (2009, Marzo 31) Gestor de arranque de Grub [Online] http://www.bdat.net/documentos/grub/c13.html

[16] Elias Hidalgo, Linuxzone (2012, Mayo 5) Añade seguridad a Ubuntu protegiendo el Grub. [Online]

http://linuxzone.es/2012/05/05/anade-mas-seguridad-a-ubuntu-protegiendo-el-grub/

[17] Ubuntu Community Wiki, (2015, Marzo 30) Grub2 [Online] https://help.ubuntu.com/community/Grub2

[18] P. Fábrega Martínez, (2009, Marzo 31), Seguridad en el arranque [Online] http://www.bdat.net/documentos/grub/x337.html

[19] Guillermo Grandes (2014, Octubre 06). Diagrama Linux netfilter iptables [Online]https://commons.wikimedia.org/wiki/File:Diagrama_linux_netfilter_iptables.png

[20] Thomas M. Eastep (2015, Marzo 3) Netfilter Overview [Online]. http://www.shorewall.net/NetfilterOverview.html

[21] Rubén Velasco (2014, Abril) Configuración de firewall en linux con iptables. [Online]. http://www.redeszone.net/gnu-linux/iptables-configuracion-del-firewall-en-linux-con-iptables/

[22] Wikipedia, (2015, Septiembre 6) Iptables [Online] https://es.wikipedia.org/wiki/Netfilter/iptables

[23] Archlinux, (2015, Julio 19) Iptables [Online] https://wiki.archlinux.org/index.php/Iptables_(Espa%C3%B1ol)

[24] Daniel Omar Rodríguez, (2008, Enero 31), Mejores prácticas y herramientas para monitoreo de bitácoras de Unix [Online].

[25] Rsyslog (2013, Mayo 24) Newbie guide to rsyslog [Online]. http://www.rsyslog.com/guides-for-rsyslog/

[26] Linux Config.org (2014, marzo) logrotate - manual page. [Online]. http://linuxconfig.org/logrotate-8-manual-page.

ANEXOS

A: Abreviaturas

ACL: Access Control List / Lista de Control de Accesos.
AES: Advanced Encryption Standard / Estándar Avanzado de Encriptación.

APT: Advanced Packaging Tool (Herramienta Avanzada de Empaquetado).

APT: Advanced Persistent Theat (Amenaza persistente avanzada).

CLI: Command-Line Interface / Interfaz de Línea de Comandos.

DHCP: Dynamic Host Configuration Protocol / Protocolo de Configuración Dinámica de Host.

DMZ: Demilitarized Zone / Zona Desmilitarizada.

DNS: Domain Name System / Sistema de Nombres de Dominio.

FTP: File Transfer Protocol / Protocolo de Transferencia de Archivos.

GPG: GNU Privacy Guard
GPL: General Public License / Licencia Pública General.

HTTP: Hypertext Transfer Protocol / Protocolo de Transferencia de Hipertexto.

HTTPS: Hypertext Transfer Protocol Secure / Protocolo Seguro de Transferencia de Hipertexto.

ICMP: Internet Control Message Protocol / Protocolo de Mensajes de Control de Internet.

IP: Internet Protocol / Protocolo de Internet.

LAN: local area network / Red de Área Local.

LILO: Linux Loader / Arranque Lilux.

MBR: Master Boot Record / Registro Maestro de Arranque.

NAT: Network Address Translation / Conversión de Direcciones de Red.

OSSIM: Open Source Security Information Management

POP3: Protocolo de Oficina Postal Version 3 / Protocolo de Oficina de Correo.

PPP: Point-to-Point Protocol / Protocolo Punto-a-Punto.

PYMES: Pequeña y Mediana Empresa.

RAM: Random Access Memory / Memoria de Acceso Aleatoria.

ROOT: Nombre de cuenta de usuario que posee todos los derecho de gestión de un sistema operativo.

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

TIC: Tecnologías d la información y Comunicación.

UDP: User Datagram Protocol / Protocolo de Datagrama de Usuario.

USB: Universal Serial Bus / Bus Universal en Serie.

WAN: Wide Area Networks / Red de Área Amplia.